

*IBM SPSS Modeler 17.1 - Guida per
script Python ed automazione*

IBM

Nota

Prima di utilizzare queste informazioni e il relativo prodotto, leggere le informazioni in "Informazioni particolari" a pagina 331.

Informazioni sul prodotto

Questa edizione si applica alla versione 17, release 1, modifica 0 di IBM(r) SPSS(r) Modeler ed a tutte le release e modifiche successive se non diversamente indicato nelle nuove edizioni.

Indice

| | |
|--|-----------|
| Capitolo 1. Script e linguaggio di script | 1 |
| Panoramica sugli script | 1 |
| Tipi di script | 1 |
| Script del flusso | 1 |
| Esempio di script del flusso: addestramento di una rete neurale | 3 |
| Script autonomi | 4 |
| Esempio di script autonomo: salvataggio e caricamento di un modello | 4 |
| Esempio di script autonomo: generazione di un modello di selezione funzioni. | 4 |
| Script dei Supernodi. | 5 |
| Esempio di script di un Supernodo | 6 |
| Esecuzioni di cicli ed esecuzione condizionale nei flussi | 6 |
| Esecuzione di cicli nei flussi | 7 |
| Esecuzione condizionale nei flussi | 10 |
| Esecuzione e interruzione degli script. | 11 |
| Trova e sostituisci | 12 |
| | |
| Capitolo 2. Linguaggio di script | 15 |
| Panoramica sul linguaggio di script | 15 |
| Python e Jython | 15 |
| Script Python. | 16 |
| Operazioni | 16 |
| Elenchi | 16 |
| Stringhe | 17 |
| Contrassegni | 19 |
| Sintassi delle istruzioni | 19 |
| Identificativi | 19 |
| Blocchi di codice | 19 |
| Passaggio di argomenti ad uno script. | 20 |
| Esempi | 20 |
| Metodi matematici | 21 |
| Utilizzo di caratteri Non-ASCII. | 22 |
| Programmazione orientata agli oggetti | 23 |
| Definizione di una classe | 24 |
| Creazione di un'istanza della classe | 24 |
| Aggiunta di attributi ad un'istanza della classe | 24 |
| Definizione dei metodi e degli attributi della classe | 24 |
| Variabili nascoste | 25 |
| Ereditarietà | 25 |
| | |
| Capitolo 3. Script in IBM SPSS Modeler | 27 |
| Tipi di script | 27 |
| Flussi, flussi SuperNodo e diagrammi | 27 |
| Flussi | 27 |
| Flussi SuperNodo | 27 |
| Diagrammi | 27 |
| Esecuzione di un flusso | 27 |
| Contesto di script | 28 |
| Riferimento a nodi esistenti | 29 |
| Ricerca di nodi | 29 |
| Impostazione delle proprietà | 30 |

| | |
|--|----|
| Creazione di nodi e modifica dei flussi | 31 |
| Creazione di nodi | 31 |
| Collegamento e scollegamento di nodi | 31 |
| Importazione, sostituzione ed eliminazione di nodi | 32 |
| Attraversamento dei nodi in un flusso | 33 |
| Cancellazione o rimozione di elementi | 34 |
| Acquisizione delle informazioni relative ai nodi | 34 |

| | |
|---|-----------|
| Capitolo 4. API di script | 37 |
| Introduzione all'API di script | 37 |
| Esempio: ricerca di nodi utilizzando un filtro personalizzato | 37 |
| Metadati: informazioni sui dati | 37 |
| Accesso agli oggetti generati. | 40 |
| Gestione degli errori | 41 |
| Parametri stream, sessione e Supernodo | 42 |
| Valori globali | 46 |
| Utilizzo di più flussi: script autonomi | 47 |

| | |
|---|-----------|
| Capitolo 5. Suggerimenti per gli script | 49 |
| Modifica dell'esecuzione del flusso | 49 |
| Esecuzione di cicli sui nodi | 49 |
| Accesso a oggetti nel IBM SPSS Collaboration and Deployment Services Repository | 50 |
| Generazione di una password codificata. | 51 |
| Controllo degli script | 51 |
| Script dalla riga di comando. | 52 |
| Compatibilità con le versioni precedenti | 52 |
| Accesso ai risultati dell'esecuzione del flusso | 52 |
| Modello di contenuto tabella | 53 |
| Modello di contenuto XML | 54 |
| Modello di contenuto JSON | 56 |
| Modello di contenuto delle statistiche di colonne e modello di contenuto delle statistiche pairwise | 57 |

| | |
|---|-----------|
| Capitolo 6. Argomenti della riga di comando | 61 |
| Modalità di richiamo del software | 61 |
| Utilizzo degli argomenti della riga di comando | 61 |
| Argomenti di sistema | 62 |
| Argomenti dei parametri | 63 |
| Argomenti per la connessione del server. | 64 |
| Argomenti per la connessione a IBM SPSS Collaboration and Deployment Services Repository. | 65 |
| Argomenti per la connessione a IBM SPSS Analytic Server | 65 |
| Combinazione di più argomenti | 66 |

| | |
|---|-----------|
| Capitolo 7. Guida alle proprietà | 67 |
| Panoramica sui riferimenti alle proprietà | 67 |
| Sintassi delle proprietà | 67 |
| Esempi di proprietà dei nodi e dei flussi | 68 |

| | |
|---|----|
| Panoramica sulle proprietà dei nodi | 69 |
| Proprietà comuni dei nodi | 69 |

Capitolo 8. Proprietà del flusso 71

Capitolo 9. Proprietà dei nodi origine 75

| | |
|--|----|
| Proprietà comuni dei nodi origine | 75 |
| Proprietà asimport | 79 |
| Proprietà del nodo cognosimport | 79 |
| Proprietà databasenode | 81 |
| Proprietà datacollectionimportnode | 83 |
| Proprietà excelimportnode | 85 |
| Proprietà evimportnode | 86 |
| Proprietà fixedfilenode | 87 |
| Proprietà del nodo gsdata_import | 89 |
| Proprietà sasimportnode | 89 |
| Proprietà simgennode | 90 |
| Proprietà statisticsimportnode | 92 |
| Proprietà del nodo tmlimport | 92 |
| Proprietà userinputnode | 93 |
| Proprietà variablefilenode | 94 |
| Proprietà xmlimportnode | 97 |
| Proprietà dataviewimport | 97 |

Capitolo 10. Proprietà dei nodi Operazioni su record 99

| | |
|---------------------------------------|-----|
| Proprietà appendnode | 99 |
| Proprietà aggregatenode | 99 |
| Proprietà balancenode | 100 |
| Proprietà derive_stfnode | 101 |
| Proprietà distinctnode | 103 |
| Proprietà mergenode | 104 |
| Proprietà rfmaggregatenode | 106 |
| Proprietà Rprocessnode | 107 |
| Proprietà del nodo Campione | 108 |
| Proprietà selectnode | 110 |
| Proprietà sortnode | 110 |
| Proprietà streamingts | 110 |

Capitolo 11. Proprietà dei nodi Operazioni su campi 115

| | |
|---|-----|
| Proprietà anonymizenode | 115 |
| Proprietà autodatapreprenode | 116 |
| Proprietà astimeintervalsnode | 119 |
| Proprietà binningnode | 119 |
| Proprietà derivenode | 122 |
| Proprietà ensemblenode | 124 |
| Proprietà fillernode | 125 |
| Proprietà filternode | 126 |
| Proprietà historynode | 127 |
| Proprietà partitionnode | 127 |
| Proprietà del nodo Ricodifica | 128 |
| Proprietà reordernode | 129 |
| Proprietà reprojectnode | 130 |
| Proprietà restructurenode | 130 |
| Proprietà rfmanalysisnode | 131 |
| Proprietà settoflagnode | 132 |
| Proprietà statisticstransformnode | 133 |
| Proprietà timeintervalsnode | 133 |
| Proprietà transposenode | 137 |

| | |
|------------------------------|-----|
| Proprietà typenode | 138 |
|------------------------------|-----|

Capitolo 12. Proprietà dei nodi Grafici 145

| | |
|---|-----|
| Proprietà comuni dei nodi Grafici | 145 |
| Proprietà collectionnode | 146 |
| Proprietà distributionnode | 147 |
| Proprietà evaluationnode | 148 |
| Proprietà graphboardnode | 150 |
| Proprietà histogramnode | 152 |
| Proprietà multiplotnode | 153 |
| Proprietà plotnode | 154 |
| Proprietà timeplotnode | 156 |
| Proprietà webnode | 157 |

Capitolo 13. Proprietà dei nodi Modelli 161

| | |
|---|-----|
| Proprietà comuni nodi modellazione | 161 |
| Proprietà anomalydetectionnode | 161 |
| Proprietà apriorinode | 163 |
| Proprietà associationrulesnode | 164 |
| Proprietà autoclassifiernode | 166 |
| Impostazione delle proprietà degli algoritmi .. | 168 |
| Proprietà autoclusternode | 169 |
| Proprietà autonumericnode | 170 |
| Proprietà bayesnetnode | 172 |
| Proprietà buildr | 173 |
| Proprietà c50node | 174 |
| Proprietà carmanode | 175 |
| Proprietà cartnode | 176 |
| Proprietà chaidnode | 178 |
| Proprietà coxregnode | 180 |
| Proprietà decisionlistnode | 182 |
| Proprietà discriminantnode | 183 |
| Proprietà factornode | 185 |
| Proprietà featureselectionnode | 186 |
| Proprietà genlinnode | 188 |
| Proprietà glmnode | 191 |
| Proprietà gle | 195 |
| Proprietà kmeansnode | 199 |
| Proprietà knnnode | 200 |
| Proprietà kohonenode | 202 |
| Proprietà linearnode | 203 |
| Proprietà linearasnode | 204 |
| Proprietà logregnode | 205 |
| Proprietà lsvmnode | 210 |
| Proprietà neuralnetnode | 211 |
| Proprietà neuralnetworknode | 213 |
| Proprietà questnode | 215 |
| Proprietà randomtrees | 217 |
| Proprietà regressionnode | 218 |
| Proprietà sequencenode | 220 |
| Proprietà slrmnode | 222 |
| Proprietà statisticsmodelnode | 222 |
| Proprietà stpnode | 223 |
| Proprietà svmnode | 227 |
| Proprietà tcmnode | 228 |
| Proprietà timeseriesnode | 231 |
| Proprietà treeas | 233 |
| Proprietà twostepnode | 235 |
| Proprietà twostepAS | 236 |

Capitolo 14. Proprietà dei nodi dei nugget del modello 239

| | |
|--|-----|
| Proprietà applyanomalydetectionnode | 239 |
| Proprietà applyapriorinode | 239 |
| Proprietà applyassociationrulesnode | 240 |
| Proprietà applyautoclassifiernode | 240 |
| Proprietà applyautoclusternode | 241 |
| Proprietà applyautonumericnode | 241 |
| Proprietà applybayesnetnode | 241 |
| Proprietà applyc50node | 241 |
| Proprietà applycarmanode | 242 |
| Proprietà applycartnode | 242 |
| Proprietà applychaidnode | 242 |
| Proprietà applycoxregnode | 243 |
| Proprietà applydecisionlistnode | 243 |
| Proprietà applydiscriminantnode | 243 |
| Proprietà applyfactornode | 243 |
| Proprietà applyfeatureselectionnode | 244 |
| Proprietà applygeneralizedlinearnode | 244 |
| Proprietà applyglmnode | 244 |
| proprietà applygle | 245 |
| Proprietà applykmeansnode | 245 |
| Proprietà applyknnnode | 245 |
| Proprietà applykohonenode | 245 |
| Proprietà applylinearnode | 246 |
| Proprietà applylinearasnode | 246 |
| Proprietà applylogregnode | 246 |
| Proprietà applysvmnode | 246 |
| Proprietà applyneuralnetnode | 247 |
| Proprietà applyneuralnetworknode | 247 |
| Proprietà applyquestnode | 247 |
| Proprietà applyr | 248 |
| Proprietà applyrandomtrees | 248 |
| Proprietà applyregressionnode | 249 |
| Proprietà applyselflearningnode | 249 |
| Proprietà applysequencenode | 249 |
| Proprietà applysvmnode | 249 |
| Proprietà applystpnode | 250 |
| Proprietà applytcmnode | 250 |
| Proprietà applytimeseriesnode | 250 |
| Proprietà applytreeas | 251 |
| Proprietà applytwestepnode | 251 |
| Proprietà applytwestepAS | 251 |

Capitolo 15. Proprietà dei nodi Modelli database 253

| | |
|--|-----|
| Proprietà dei nodi Modelli Microsoft | 253 |
| Proprietà dei nodi Modelli Microsoft | 253 |
| Proprietà dei nugget del modello Microsoft | 255 |
| Proprietà dei nodi Modelli Oracle | 257 |
| Proprietà dei nodi Modelli Oracle | 257 |
| Proprietà dei nugget del modello Oracle | 263 |
| Proprietà dei nodi Modelli IBM DB2 | 264 |
| Proprietà dei nodi Modelli IBM DB2 | 264 |
| Proprietà dei nugget del modello IBM DB2 | 269 |
| Proprietà dei nodi Modelli IBM Netezza Analytics | 270 |
| Proprietà dei nodi Modelli Netezza | 270 |
| Proprietà dei nugget del modello Netezza | 280 |

Capitolo 16. Proprietà dei nodi Output 283

| | |
|--|-----|
| Proprietà analysisnode | 283 |
| Proprietà dataauditnode | 284 |
| Proprietà matrixnode | 285 |
| Proprietà meansnode | 287 |
| Proprietà reportnode | 289 |
| Proprietà routputnode | 290 |
| Proprietà setglobalsnode | 290 |
| Proprietà simevalnode | 291 |
| Proprietà simfitnode | 292 |
| Proprietà statisticsnode | 292 |
| Proprietà statisticsoutputnode | 294 |
| Proprietà tablenode | 294 |
| Proprietà transformnode | 296 |

Capitolo 17. Proprietà dei nodi di esportazione. 299

| | |
|---|-----|
| Proprietà comuni dei nodi di esportazione | 299 |
| Proprietà asexport | 299 |
| Proprietà del nodo di esportazione Cognos | 299 |
| Proprietà databaseexportnode | 301 |
| Proprietà datacollectionexportnode | 305 |
| Proprietà excelexportnode | 305 |
| Proprietà outputfilenode | 306 |
| Proprietà sasexportnode | 307 |
| Proprietà statisticsexportnode | 308 |
| Proprietà del nodo tmlexport | 308 |
| Proprietà xmlexportnode | 309 |

Capitolo 18. Proprietà dei nodi IBM SPSS Statistics 311

| | |
|---|-----|
| Proprietà statisticsimportnode | 311 |
| Proprietà statisticstransformnode | 311 |
| Proprietà statisticsmodelnode | 312 |
| Proprietà statisticsoutputnode | 312 |
| Proprietà statisticsexportnode | 313 |

Capitolo 19. Proprietà dei Supernodi 315

Appendice A. Riferimento dei nomi del nodo 317

| | |
|---|-----|
| Nomi dei nugget del modello | 317 |
| Per evitare nomi di modelli duplicati | 319 |
| Nomi dei tipi di output | 319 |

Appendice B. Migrazione da script legacy a script Python 321

| | |
|--|-----|
| Panoramica sulla migrazione di script Legacy | 321 |
| Differenze generali | 321 |
| Contesto di script | 321 |
| Comandi o funzioni | 321 |
| Valori letterali e commenti | 322 |
| Operatori | 322 |
| Istruzioni condizionali e cicli | 323 |
| Variabili | 324 |
| Tipi di nodo, output e modello | 324 |
| Nomi proprietà | 324 |
| Riferimenti a nodi | 324 |
| Ottenimento ed impostazione di proprietà | 325 |

| | | | |
|---|-----|--|------------|
| Modifica dei flussi | 325 | Operazioni di output di documento | 329 |
| Operazioni nodo | 326 | Altre differenze tra script legacy e script Python | 329 |
| Esecuzione di cicli | 326 | Informazioni particolari | 331 |
| esecuzione di flussi | 327 | Marchi | 332 |
| Accesso ad oggetti attraverso il file system ed il repository | 328 | Indice analitico. | 335 |
| Operazioni di flusso | 328 | | |
| Operazioni del modello | 329 | | |

Capitolo 1. Script e linguaggio di script

Panoramica sugli script

Gli script di IBM® SPSS Modeler sono un potente strumento per automatizzare i processi nell'interfaccia utente. Tramite gli script è possibile eseguire gli stessi tipi di azioni eseguite con il mouse o la tastiera, nonché automatizzare le attività ripetitive o la cui esecuzione manuale richiederebbe un tempo molto maggiore.

È possibile utilizzare gli script per:

- Imporre un ordine specifico per l'esecuzione dei nodi in un flusso.
- Impostare le proprietà di un nodo ed eseguire le derivazioni utilizzando un sottoinsieme di CLEM (Control Language for Expression Manipulation).
- Specificare una sequenza automatica di operazioni che in genere richiedono l'intervento dell'utente, per esempio la creazione e la verifica di un modello.
- Impostare processi di grande complessità per i quali sono necessari interventi sostanziali da parte dell'utente, per esempio le procedure di convalida incrociata che richiedono più processi di creazione e verifica dei modelli.
- Impostare i processi di manipolazione dei flussi, ad esempio recuperare un flusso di addestramento per un modello, eseguirlo e creare il flusso di verifica del modello corrispondente in modo automatico.

In questo capitolo sono fornite descrizioni approfondite ed esempi di script a livello di flusso, script autonomi e script all'interno di Supernodi nell'interfaccia IBM SPSS Modeler. Per ulteriori informazioni sul linguaggio di script, la sintassi e i comandi, consultare i capitoli che seguono.

Nota: non è possibile importare ed eseguire gli script creati in IBM SPSS Statistics all'interno di IBM SPSS Modeler.

Tipi di script

IBM SPSS Modeler utilizza tre tipi di script:

- Gli **script del flusso** sono archiviati come proprietà di stream e quindi salvati e caricati con un flusso specifico. Per esempio, è possibile scrivere uno script del flusso che automatizza il processo di addestramento e applicazione di un nugget del modello. È anche possibile specificare che, ogni volta che viene eseguito un determinato stream, venga eseguito lo script anziché il contenuto dell'area del flusso.
- Gli **script autonomi** non sono associati ad alcun flusso particolare e vengono salvati in file di testo esterni. È possibile utilizzare uno script autonomo, per esempio, per manipolare insieme più flussi.
- Gli **script del Supernodo** vengono archiviati come proprietà del flusso Supernodo. Gli script del Supernodo sono disponibili solo nei Supernodi terminali. È possibile utilizzare uno script del Supernodo per controllare la sequenza di esecuzione del contenuto del Supernodo. Per i Supernodi non terminali (origine o di elaborazione), è possibile definire le proprietà del Supernodo o direttamente i nodi che esso contiene nello script del flusso.

Script del flusso

È possibile utilizzare gli script per personalizzare le operazioni all'interno di un flusso specifico e salvarli insieme al flusso. Gli script del flusso possono essere utilizzati per specificare un particolare ordine di esecuzione per i nodi terminali all'interno di un flusso. La finestra di dialogo di script del flusso consente di modificare lo script salvato insieme al flusso corrente.

Per accedere alla scheda dello script del flusso nella finestra di dialogo Proprietà flusso:

1. Dal menu Strumenti, scegliere:

Proprietà flusso > Esecuzione

2. Fare clic sulla scheda **Esecuzione** per utilizzare gli script per il flusso corrente.

Le icone della barra degli strumenti nella parte superiore della finestra di dialogo dello script del flusso consentono di eseguire le operazioni riportate di seguito:

- Importare il contenuto di uno script autonomo preesistente nella finestra.
- Salvare lo script come file di testo.
- Stampare uno script.
- Accodare lo script di default.
- Modificare uno script (annullare l'operazione, tagliare, copiare, incollare ed altre funzioni di modifica comuni).
- Eseguire l'intero script corrente.
- Eseguire righe selezionate di uno script.
- Arrestare uno script durante l'esecuzione. Questa icona è abilitata solo durante l'esecuzione di uno script.
- Verificare la sintassi dello script e, se vengono rilevati errori, visualizzarli nel riquadro inferiore della finestra di dialogo.

A partire dalla versione 16.0, SPSS Modeler utilizza il linguaggio di script Python. Tutte le versioni precedenti utilizzavano un linguaggio di script univoco di SPSS Modeler, ora indicato come Script . In base al tipo di script utilizzato, nella scheda **Esecuzione**, selezionare la modalità di esecuzione **Predefinita (script facoltativo)**, quindi selezionare **Python** o **Legacy**.

Inoltre, è possibile specificare se lo script debba essere o meno eseguito all'esecuzione del flusso. È possibile selezionare **Esegui questo script** per eseguire lo script a ogni esecuzione del flusso rispettando l'ordine di esecuzione dello script. L'automazione a livello di flusso garantita in questo modo consente di accelerare la creazione del modello. Tuttavia, l'impostazione di default ignora questo script durante l'esecuzione del flusso. Anche se si seleziona l'opzione **Ignora questo script**, è sempre possibile eseguire lo script direttamente da questa finestra di dialogo.

L'editor di script include le seguenti funzioni che rendono più semplice la creazione di script:

- Evidenziazione della sintassi: parole chiave, valori letterali (come stringhe e numeri) e commenti sono evidenziati.
- Numerazione delle righe.
- Corrispondenza del blocco: quando il cursore viene posizionato all'inizio di un blocco di programma, viene evidenziato anche il blocco finale corrispondente.
- Suggerimenti per il completamento automatico.

Gli stili di testo e colori utilizzati dal programma di evidenziazione della sintassi possono essere personalizzati utilizzando le preferenze di visualizzazione di IBM SPSS Modeler. È possibile accedere alle preferenze di visualizzazione selezionando **Strumenti > Opzioni > Opzioni utente** e facendo clic sulla scheda **Sintassi**.

È possibile accedere ad un elenco di completamenti della sintassi suggeriti selezionando **Suggerimento automatico** dal menu di contesto oppure premendo Ctrl + Spazio. Utilizzare i tasti cursore per spostarsi verso l'alto e verso il basso all'interno dell'elenco, quindi premere Invio per inserire il testo selezionato. Premere Esc per uscire dalla modalità di suggerimento automatico senza modificare il testo esistente.

La scheda **Debug** visualizza i messaggi di debug e può essere utilizzata per valutare lo stato dello script una volta eseguito lo script. La scheda **Debug** è composta da un'area di testo di sola lettura e da un

campo di testo di input a riga singola. L'area di testo visualizza il testo inviato dagli script all'output standard o all'errore standard, ad esempio mediante il testo del messaggio di errore. Il campo del testo di input accetta l'input da parte dell'utente. Tale input viene valutato all'interno del contesto dello script eseguito più recentemente all'interno della finestra di dialogo (detto *contesto di script*). L'area di testo contiene i comandi e l'output risultante, in modo che gli utenti possano visualizzare una traccia dei comandi. Il campo del testo di input contiene sempre il prompt dei comandi (--> per gli script legacy).

Nelle seguenti circostanze viene creato un nuovo contesto di script:

- Viene eseguito uno script utilizzando il pulsante "Esegui questo script" oppure il pulsante "Esegui righe selezionate".
- Il linguaggio di script viene modificato.

Se viene creato un nuovo contesto di script, l'area di testo viene svuotata.

Nota: L'esecuzione di un flusso all'esterno del pannello dello script non modifica il contesto dello script del pannello dello script. I valori delle variabili create come parte di tale esecuzione non saranno visibili all'interno della finestra di dialogo dello script.

Esempio di script del flusso: addestramento di una rete neurale

È possibile utilizzare un flusso per addestrare una rete neurale durante l'esecuzione. La verifica del modello prevede in genere l'esecuzione del nodo di creazione modelli per aggiungere il modello al flusso, l'esecuzione delle connessioni appropriate e l'esecuzione del nodo Analisi.

Con uno script di IBM SPSS Modeler, è possibile automatizzare il processo di verifica del nugget del modello creato. Per esempio, il seguente script del flusso per il flusso di esempio *druglearn.str* (disponibile nella cartella */Demos/streams/* dell'installazione di IBM SPSS Modeler) può essere eseguito dalla finestra di dialogo Proprietà flusso (**Strumenti > Proprietà flusso > Script**):

```
stream = modeler.script.stream()
neuralnetnode = stream.findByType("neuralnetwork", None)
results = []
neuralnetnode.run(results)
appliernode = stream.createModelApplierAt(results[0], "Drug", 594, 187)
analysisnode = stream.createAt("analysis", "Drug", 688, 187)
typenode = stream.findByType("type", None)
stream.linkBetween(appliernode, typenode, analysisnode)
analysisnode.run([])
```

L'elenco riportato di seguito descrive ogni riga in questo esempio di script.

- La prima riga definisce una variabile che punta al flusso corrente.
- Nella riga 2, lo script rileva il nodo builder Rete neurale.
- Nella riga 3, lo script crea un elenco in cui è possibile archiviare i risultati dell'esecuzione.
- Nella riga 4, viene creato il nugget del modello Rete Neurale. Tale elemento viene archiviato nell'elenco definito alla riga 3.
- Nella riga 5, per il nugget del modello viene creato un nodo Applicazione del modello che viene posizionato nell'area di disegno del flusso.
- Nella riga 6, viene creato un nodo di analisi denominato Drug.
- Nella riga 7, lo script trova il nodo Tipo.
- Nella riga 8, lo script collega il nodo Applicazione del modello creato alla riga 5 tra il nodo Tipo ed il nodo Analisi.
- Infine, viene eseguito il nodo Analisi per produrre il report di analisi.

È possibile utilizzare uno script per creare ed eseguire un flusso nuovo, partendo da un'area vuota. Per ulteriori informazioni sul linguaggio di script in generale, vedere Panoramica sul linguaggio di script.

Script autonomi

Nella finestra di dialogo Script autonomo è possibile creare o modificare uno script salvato come file di testo. Nella finestra viene visualizzato il nome del file e sono disponibili funzionalità per il caricamento, il salvataggio, l'importazione e l'esecuzione degli script.

Per accedere alla finestra di dialogo dello script autonomo:

Dal menu principale, scegliere:

Strumenti > Script autonomi

Per gli script autonomi e del flusso sono disponibili la stessa barra degli strumenti e le stesse opzioni di controllo della sintassi degli script. Per ulteriori informazioni, consultare l'argomento "Script del flusso" a pagina 1.

Esempio di script autonomo: salvataggio e caricamento di un modello

Gli script autonomi sono utili per la manipolazione degli stream. Si supponga di avere due flussi, uno che crea un modello e un altro che utilizza grafici per analizzare l'insieme di regole generato dal primo flusso mediante i campi di dati esistenti. Uno script autonomo per questa situazione potrebbe essere simile al seguente:

```
taskrunner = modeler.script.session().getTaskRunner()

# Modify this to the correct Modeler installation Demos folder.
# Note use of forward slash and trailing slash.
installation = "C:/Program Files/IBM/SPSS/Modeler/16/Demos/"

# First load the model builder stream from file and build a model
druglearn_stream = taskrunner.openStreamFromFile(installation + "streams/druglearn.str", True)
results = []
druglearn_stream.findByType("c50", None).run(results)

# Save the model to file
taskrunner.saveModelToFile(results[0], "rule.gm")

# Now load the plot stream, read the model from file and insert it into the stream
drugplot_stream = taskrunner.openStreamFromFile(installation + "streams/drugplot.str", True)
model = taskrunner.openModelFromFile("rule.gm", True)
modelapplier = drugplot_stream.createModelApplier(model, "Drug")

# Now find the plot node, disconnect it and connect the
# model applier node between the derive node and the plot node
derivenode = drugplot_stream.findByType("derive", None)
plotnode = drugplot_stream.findByType("plot", None)
drugplot_stream.disconnect(plotnode)
modelapplier.setPositionBetween(derivenode, plotnode)
drugplot_stream.linkBetween(modelapplier, derivenode, plotnode)
plotnode.setPropertyValue("color_field", "%C-Drug")
plotnode.run([])
```

Nota: Per ulteriori informazioni sul linguaggio di script in generale, vedere Panoramica sul linguaggio di script.

Esempio di script autonomo: generazione di un modello di selezione funzioni

Iniziando con un'area vuota, questo esempio crea un flusso che genera un Modello di selezione funzioni, applica il modello e crea una tabella che elenca i 15 campi più importanti relativi all'obiettivo specificato.

```

stream = modeler.script.session().createProcessorStream("featureselection", True)

statisticsimportnode = stream.createAt("statisticsimport", "Statistics File", 150, 97)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/customer_dbase.sav")

typenode = stream.createAt("type", "Type", 258, 97)
typenode.setKeyedPropertyValue("direction", "response_01", "Target")

featureselectionnode = stream.createAt("featureselection", "Feature Selection", 366, 97)
featureselectionnode.setPropertyValue("top_n", 15)
featureselectionnode.setPropertyValue("max_missing_values", 80.0)
featureselectionnode.setPropertyValue("selection_mode", "TopN")
featureselectionnode.setPropertyValue("important_label", "Check Me Out!")
featureselectionnode.setPropertyValue("criteria", "Likelihood")

stream.link(statisticsimportnode, typenode)
stream.link(typenode, featureselectionnode)
models = []
featureselectionnode.run(models)

# Assumes the stream automatically places model apply nodes in the stream
applynode = stream.findByType("applyfeatureselection", None)
tablenode = stream.createAt("table", "Table", applynode.getXPosition() + 96, applynode.getYPosition())
stream.link(applynode, tablenode)
tablenode.run([])

```

Questo script crea un nodo origine nel quale leggere i dati, utilizza un nodo Tipo per impostare il ruolo (direzione) del campo response_01 su Obiettivo, quindi crea ed esegue un nodo Selezione funzioni. Inoltre, lo script connette i nodi e le posizioni nell'area del flusso per generare un layout leggibile. Il nugget del modello così ottenuto viene quindi connesso a un nodo Tabella, che elenca i 15 campi più importanti come determinato dalle proprietà selection_mode e top_n. Per ulteriori informazioni, consultare l'argomento "Proprietà featureselectionnode" a pagina 186.

Script dei Supernodi

È possibile creare e salvare script all'interno di qualsiasi Supernodo terminale utilizzando il linguaggio di script di IBM SPSS Modeler. Questi script sono disponibili solo per i Supernodi terminali e vengono spesso utilizzati durante la creazione di modelli di stream o per imporre un ordine di esecuzione speciale per il contenuto del Supernodo. Gli script del Supernodo consentono anche l'esecuzione di più di uno script all'interno di un flusso.

Per esempio, si supponga che sia stato necessario specificare l'ordine di esecuzione di un flusso complesso e che il Supernodo contenga più nodi tra cui un nodo Calcola globali, che deve essere eseguito prima di creare un nuovo campo utilizzato in un nodo Plot. In tal caso, è possibile creare uno script del Supernodo che esegue prima il nodo Calcola globali. I valori calcolati da questo nodo, quali la media o la deviazione standard, possono quindi essere utilizzati quando viene eseguito il nodo Plot.

All'interno di uno script del Supernodo è possibile specificare le proprietà del nodo analogamente agli altri script. In alternativa, è possibile modificare e definire le proprietà di qualsiasi Supernodo o dei suoi nodi incapsulati direttamente da uno script del flusso. Per ulteriori informazioni, consultare l'argomento Capitolo 19, "Proprietà dei Supernodi", a pagina 315. Questo metodo funziona per i Supernodi origine e di elaborazione e per i Supernodi terminali.

Nota: poiché solo i Supernodi terminali possono eseguire i propri script, la scheda Script della finestra di dialogo Supernodo è disponibile solo per i Supernodi terminali.

Per aprire la finestra di dialogo Script Supernodo dall'area principale:

Selezionare un Supernodo terminale nell'area dello script e, dal menu Supernodo, scegliere:

Script Supernodo...

Per aprire la finestra di dialogo Script Supernodo dall'area del Supernodo in modalità Zoom avanti:

Fare clic con il pulsante destro del mouse sull'area del Supernodo e dal menu di scelta rapida scegliere:

Script Supernodo...

Esempio di script di un Supernodo

Lo script del Supernodo riportato di seguito dichiara l'ordine in cui devono essere eseguiti i nodi terminali all'interno del Supernodo. Questo ordine assicura che il nodo Calcola globali venga eseguito per primo, in modo che i valori calcolati da questo nodo possano successivamente essere utilizzati quando viene eseguito un altro nodo.

```
execute 'Set Globals'  
execute 'gains'  
execute 'profit'  
execute 'age v. $CC-pep'  
execute 'Table'
```

Esecuzioni di cicli ed esecuzione condizionale nei flussi

Dalla Versione version 16.0 in poi, SPSS Modeler consente di creare alcuni script di base all'interno di un flusso selezionando i valori all'interno di varie finestre di dialogo invece di dover scrivere istruzioni direttamente nel linguaggio di script. I due tipi principali di script che è possibile creare in questo modo sono cicli semplici e un modo per eseguire i nodi se una condizione è stata soddisfatta.

È possibile combinare le regole sia dell'esecuzione di cicli che dell'esecuzione condizionale all'interno di un flusso. Ad esempio, si supponga di avere i dati relativi alle vendite di automobili dai produttori di tutto il mondo. È possibile impostare un ciclo per elaborare i dati in un flusso, identificando i dettagli per paese di produzione ed creare output di dati in grafici diversi che mostrano i dettagli come ad esempio il volume di vendite per modello, i livelli di emissione sia per produttore che per dimensione del motore e così via. Se si fosse interessati ad analizzare solo le informazioni Europee, si potrebbero anche aggiungere condizioni nell'esecuzione del ciclo che forniscano grafici creati per produttori situati in America e Asia.

Nota: Poiché sia l'esecuzione di cicli che l'esecuzione condizionale sono basate su script in background, questi vengono applicati solo ad un flusso totale quando viene eseguito.

- **Esecuzione di cicli** È possibile utilizzare l'esecuzione di cicli per automatizzare attività ripetitive. Ad esempio, questo potrebbe significare l'aggiunta di un dato numero di nodi a un flusso e la modifica di un parametro del nodo ogni volta. In alternativa, è possibile controllare l'esecuzione di un flusso o ramo ancora una volta per un dato numero di volte, come nei seguenti esempi:
 - Eseguire il flusso un dato numero di volte e modificare l'origine ogni volta.
 - Eseguire il flusso un dato numero di volte modificando il valore di una variabile ogni volta.
 - Eseguire il flusso un dato numero di volte immettendo un campo aggiuntivo ad ogni esecuzione.
 - Costruire un modello un dato numero di volte e modificare le impostazioni del modello ogni volta.
- **Esecuzione Condizionale** È possibile utilizzarla per controllare come i nodi terminali vengono eseguiti, in base alle condizioni che si predefiniscono, gli esempi possono includere i seguenti:
 - In base a se un dato valore è vero o falso, controlla se un nodo verrà eseguito.
 - Definisce se un'esecuzione di cicli di nodi verrà eseguita in parallelo o sequenziale.

Sia l'esecuzione di cicli che l'esecuzione condizionale vengono configurate sulla scheda Esecuzione all'interno della finestra di dialogo Proprietà del flusso. I nodi che vengono utilizzati nei requisiti

condizionali o di cicli vengono mostrati con un simbolo aggiuntivo a loro allegato sull'area di disegno del flusso per indicare che stanno prendendo parte nell'esecuzione di cicli e nell'esecuzione condizionale.

È possibile accedere alla scheda Esecuzione in uno dei 3 modi:

- Utilizzando i menu nella parte superiore della finestra di dialogo principale:
 1. Dal menu Strumenti, scegliere:
Proprietà flusso > Esecuzione
 2. Fare clic sulla scheda Esecuzione per utilizzare gli script per il flusso corrente.
- Dall'interno di un flusso:
 1. Fare clic col tasto destro su un nodo e scegliere **Esecuzione Cicli/Condizionale**.
 2. Selezionare l'opzione pertinente del sottomenu.
- Dalla barra degli strumenti del grafico nella parte superiore della finestra di dialogo principale, fare clic sull'icona proprietà del flusso.

Se questa è la prima volta che si configurano i dettagli o dell'esecuzione di cicli o dell'esecuzione condizionale, nella scheda Esecuzione selezionare la modalità di esecuzione **Esecuzione Cicli/Condizionale** e poi selezionare o la sottoscheda **Condizionale** o quella **Cicli**.

Esecuzione di cicli nei flussi

Con l'esecuzione di cicli è possibile automatizzare le attività ripetitive nei flussi; alcuni esempio potrebbero essere i seguenti:

- Eseguire il flusso un dato numero di volte e modificare l'origine ogni volta.
- Eseguire il flusso un dato numero di volte modificando il valore di una variabile ogni volta.
- Eseguire il flusso un dato numero di volte immettendo un campo aggiuntivo ad ogni esecuzione.
- Costruire un modello un dato numero di volte e modificare le impostazioni del modello ogni volta.

Le condizioni da soddisfare vengono impostate nella sottoscheda **Esecuzione di cicli** della scheda Esecuzione del flusso. Per visualizzare la sottoscheda, selezionare la modalità di esecuzione **Esecuzione di cicli/Esecuzione Condizionale**.

Ogni requisito dell'esecuzione di cicli che viene definita avrà effetto quando il flusso viene eseguito, se è stata impostata la modalità di esecuzione **Esecuzione di cicli/Esecuzione Condizionale**. Se lo si desidera, è possibile generare il codice dello script per i requisiti di esecuzione dei cicli ed incollarlo nell'editor dello script facendo clic su **Incolla...** nell'angolo in basso a destra della sottoscheda Esecuzione di Cicli; la scheda principale Esecuzione visualizza le modifiche da mostrare nella modalità di esecuzione **Default (script facoltativo)** con lo script nella parte in alto della scheda. Questo significa che è possibile definire una struttura di esecuzione dei cicli utilizzando le varie opzioni delle finestre di dialogo di esecuzione dei cicli prima di generare uno script che è possibile personalizzare ulteriormente nell'editor dello script. Si noti che quando si fa clic su **Incolla...** ogni requisito di esecuzione condizionale che è stato definito, verrà visualizzato nello script generato.

Importante: Le variabili nei cicli impostate in un flusso SPSS Modeler potrebbero essere sovrascritte se si esegue il flusso in un lavoro IBM SPSS Collaboration and Deployment Services. Ciò si verifica perché la voce dell'editor lavori IBM SPSS Collaboration and Deployment Services sostituisce la voce SPSS Modeler. Ad esempio, se si imposta una variabile nel ciclo nel flusso per creare un diverso nome del file di output per ciascun ciclo, i file vengono denominati correttamente in SPSS Modeler ma vengono sovrascritti dalla voce prefissata immessa nella scheda Risultato di IBM SPSS Collaboration and Deployment Services Deployment Manager.

Impostazione di un ciclo

1. Creare una chiave di iterazione per definire una struttura principale dell'esecuzione dei cicli che devono essere eseguiti nel flusso. Consultare Creare una chiave di iterazione per ulteriori informazioni.
2. Quando necessario, definire una o più variabili di iterazione. Consultare Creare una variabile di iterazione per ulteriori informazioni.
3. Le iterazioni, e qualsiasi variabile creata, vengono mostrate nel corpo principale della sottoscheda. Per default, le iterazioni vengono eseguite nell'ordine in cui appaiono; per spostare una iterazione su o giù nell'elenco, selezionarla con un clic e quindi utilizzare le frecce su o giù nella colonna a destra della sottoscheda, per modificarne l'ordine.

Creazione di una chiave di iterazione per l'esecuzione di cicli nei flussi

Si utilizza una chiave di iterazione per definire la struttura principale dell'esecuzione dei cicli che devono essere eseguiti nel flusso. Ad esempio, se si sta analizzando la vendita delle automobili, si potrebbe creare un parametro di flusso *Paese di produzione* e utilizzarlo come chiave di iterazione; quando il flusso viene eseguito, questa chiave è impostata su ogni valore dei diversi paesi nei propri dati durante ogni iterazione. Utilizzare la finestra di dialogo Definisci chiave di iterazione per impostare la chiave.

Per aprire la finestra di dialogo, selezionare o il pulsante **Chiave di iterazione...** nell'angolo in basso a sinistra della sottoscheda di Esecuzione dei cicli o fare clic sul pulsante destro del mouse su qualsiasi nodo nel flusso e selezionare o **Esecuzione cicli/Esecuzione condizionale > Definisci Chiave di iterazione (Campi)** o **Esecuzione cicli/Esecuzione condizionale > Definisci Chiave di iterazione (Valori)**. Se si apre la finestra di dialogo del flusso, alcuni dei campi possono essere completati automaticamente per l'utente, come ad esempio il nome del nodo.

Per impostare una chiave iterazione, completare i seguenti campi:

Agisce sui. È possibile selezionare una delle seguenti opzioni:

- **Parametro di flusso - Campi.** Utilizzare questa opzione per creare un ciclo che imposti il valore di un parametro di flusso esistente in ogni campo specificato mano a mano.
- **Parametro di flusso - Valori.** Utilizzare questa opzione per creare un ciclo che imposti il valore di un parametro di flusso esistente in ogni valore specificato mano a mano.
- **Proprietà del nodo - Campi.** Utilizzare questa opzione per creare un ciclo che imposti il valore della proprietà del nodo in ogni campo specificato mano a mano.
- **Proprietà del nodo - Valori.** Utilizzare questa opzione per creare un ciclo che imposti il valore della proprietà del nodo in ogni valore specificato mano a mano.

Cosa impostare. Scegliere l'elemento che avrà il valore impostato ogni volta che il ciclo viene eseguito. È possibile selezionare una delle seguenti opzioni:

- **Parametro.** Disponibile solo se si seleziona **Parametro di flusso- Campi** o **Parametro di flusso - Valori**. Selezionare il parametro richiesto dall'elenco disponibile.
- **Nodo.** Disponibile solo se si seleziona **Proprietà del nodo- Campi** o **Proprietà del nodo - Valori**. Selezionare il nodo per cui si desidera impostare un ciclo. Fare clic sul pulsante sfoglia per aprire la finestra di dialogo Seleziona Nodo e scegliere il nodo che si desidera; se vi sono troppi nodi elencati, è possibile filtrare la visualizzazione per mostrare solo certi tipi di nodi selezionando una delle seguenti categorie: Nodi Origine, Processo, Grafico, Modello, Output, Esporta o Modelli applicati.
- **Proprietà.** Disponibile solo se si seleziona **Proprietà del nodo- Campi** o **Proprietà del nodo - Valori**. Selezionare la proprietà del nodo dall'elenco disponibile.

Campi da utilizzare. Disponibile solo se si seleziona **Parametro di flusso- Campi** o **Proprietà del nodo - Campi**. Scegliere il campo, o i campi, all'interno di un nodo da utilizzare per fornire i valori di iterazione. È possibile selezionare una delle seguenti opzioni:

- **Nodo.** Disponibile solo se si seleziona **Parametro di flusso – Campi**. Selezionare il nodo che contiene i dettagli per i quali si desidera impostare un ciclo. Fare clic sul pulsante sfoglia per aprire la finestra di dialogo Seleziona Nodo e scegliere il nodo che si desidera; se vi sono troppi nodi elencati, è possibile filtrare la visualizzazione per mostrare solo certi tipi di nodi selezionando una delle seguenti categorie: Nodi Origine, Processo, Grafico, Modello, Output, Esporta o Modelli applicati.
- **Elenco campi.** Fare clic sul pulsante elenco nella colonna destra per visualizzare la finestra di dialogo Seleziona Campi, all'interno della quale è possibile selezionare i campi nel nodo per fornire i dati di iterazione. Consultare “Selezione campi per le iterazioni” a pagina 10 per ulteriori informazioni.

Valori da utilizzare. Disponibile solo se si seleziona **Parametro di flusso – Valori** o **Proprietà del nodo – Valori**. Scegliere il valore, o i valori, all'interno del campo selezionato da utilizzare come valori di iterazione. È possibile selezionare una delle seguenti opzioni:

- **Nodo.** Disponibile solo se si seleziona **Parametro di flusso – Valori**. Selezionare il nodo che contiene i dettagli per i quali si desidera impostare un ciclo. Fare clic sul pulsante sfoglia per aprire la finestra di dialogo Seleziona Nodo e scegliere il nodo che si desidera; se vi sono troppi nodi elencati, è possibile filtrare la visualizzazione per mostrare solo certi tipi di nodi selezionando una delle seguenti categorie: Nodi Origine, Processo, Grafico, Modello, Output, Esporta o Modelli applicati.
- **Elenco campi.** Selezionare il campo nel nodo per fornire i dati di iterazione.
- **Elenco valori.** Fare clic sul pulsante elenco nella colonna destra per visualizzare la finestra di dialogo Seleziona Valori, all'interno della quale è possibile selezionare i valori nel campo per fornire i dati di iterazione.

Creazione di una variabile di iterazione per l'esecuzione di cicli nei flussi

È possibile utilizzare le variabili di iterazione per modificare i valori dei parametri del flusso o le proprietà dei nodi selezionati all'interno di un flusso ogni volta che viene eseguito un ciclo. Ad esempio, se il ciclo del flusso sta analizzando i dati di vendita delle automobili e sta utilizzando *Paese di produzione* come chiave di iterazione, si potrebbe avere un grafico di output che mostra le vendite per modello e un altro grafico di output che mostra le informazioni sulle emissioni di gas di scarico. In questi casi, è possibile creare variabili di iterazione che creano nuovi titoli per i grafici risultanti, per esempio *Emissioni veicoli svedesi* e *Vendite automobili giapponesi per modello*. Utilizzare la finestra di dialogo Definisci variabile di iterazione per impostare una qualsiasi variabile che si desidera richiedere.

Per aprire la finestra di dialogo, selezionare il pulsante **Aggiungi variabile...** nell'angolo in basso alla sinistra della scheda secondaria Esecuzione di cicli, o fare clic con il tasto destro del mouse su qualsiasi nodo nel flusso e selezionare: **Esecuzione di cicli/Esecuzione condizionale > Definisci variabile di iterazione**.

Per impostare una variabile di iterazione, completare i seguenti campi:

Modifica. Selezionare il tipo di attributo che si desidera modificare. È possibile scegliere o tra **Parametro di flusso** o **Proprietà del nodo**.

- Se si seleziona **Parametro di flusso**, scegliere il parametro richiesto e quindi, utilizzando una delle seguenti opzioni, se disponibili nel proprio flusso, definire quale valore di quel parametro deve essere impostato con ogni iterazione del ciclo:
 - **Variabile globale.** Selezionare la variabile globale che il parametro di flusso deve impostare.
 - **Cella tabella di output.** Per impostare un parametro di flusso come valore nella cella della tabella di output, selezionare la tabella dall'elenco e immettere la **Riga** e la **Colonna** da utilizzare.
 - **Immettere manualmente.** Selezionare questa opzione se si desidera immettere manualmente un valore per questo parametro da prendere in ogni iterazione. Quando si torna alla sottoscheda esecuzione di cicli viene creata una nuova colonna in cui si inserisce il testo richiesto.
- Se si seleziona **Proprietà del nodo**, scegliere il nodo richiesto e una delle relative proprietà e quindi impostare il valore che si desidera utilizzare per tale proprietà. Impostare il nuovo valore della proprietà utilizzando una delle seguenti opzioni:

- **Singolo.** Il valore della proprietà utilizzerà il valore della chiave di iterazione. Consultare “Creazione di una chiave di iterazione per l'esecuzione di cicli nei flussi” a pagina 8 per ulteriori informazioni.
- **Come prefisso per Ramo.** Utilizza il valore della chiave di iterazione come prefisso di quello che è stato immesso nel campo **Ramo**.
- **Come suffisso per Ramo.** Utilizza il valore della chiave di iterazione come suffisso di quello che è stato immesso nel campo **Ramo**.

Se si seleziona l'opzione prefisso o suffisso viene richiesto di aggiungere il testo aggiuntivo al campo **Ramo**. Per esempio, se il valore della chiave di iterazione è *Paese di produzione* e si seleziona **Come prefisso per Ramo**, è possibile immettere *- vendite per modello* in questo campo.

Selezione campi per le iterazioni

Quando si creano le iterazioni è possibile selezionare uno o più campi utilizzando la finestra di dialogo **Seleziona Campi**.

Ordina per Per visualizzare i campi disponibili in un determinato ordine, sono disponibili le seguenti opzioni:

- **Naturale** Visualizza l'ordine dei campi secondo la modalità di passaggio a valle nel flusso di dati nel nodo corrente.
- **Nome** Utilizza l'ordine alfabetico per ordinare i campi per la visualizzazione.
- **Tipo** Visualizza i campi ordinati in base al relativo livello di misurazione. Questa opzione è utile quando si selezionano campi con un determinato livello di misurazione.

Selezionare i campi dall'elenco uno per volta, oppure utilizzare i metodi Maiusc-clic e Ctrl-clic per selezionare più campi contemporaneamente. È anche possibile utilizzare i pulsanti nella parte inferiore dell'elenco per selezionare gruppi di campi in base al livello di misurazione, oppure per selezionare o deselezionare tutti i campi nella tabella.

Si noti che i campi disponibili per essere selezionati sono filtrati in modo da mostrare solo i campi che sono appropriati per i parametri del flusso o le proprietà del nodo che si sta utilizzando. Per esempio, se si sta utilizzando un parametro di flusso che ha Stringa come tipo di archiviazione, vengono mostrati solo i campi che hanno come tipo di archiviazione Stringa.

Esecuzione condizionale nei flussi


Con l'esecuzione condizionale è possibile controllare il modo in cui i nodi terminali vengono eseguiti, in base ai contenuti del flusso corrispondenti alle condizioni che si desidera definire; esempi possono includere:

- In base a se un dato valore è vero o falso, controlla se un nodo verrà eseguito.
- Definisce se un'esecuzione di cicli di nodi verrà eseguita in parallelo o sequenziale.

Si impostano le condizioni che devono essere soddisfatte nella sottoscheda **Condizionale** della scheda **Esecuzione del flusso**. Per visualizzare la sottoscheda, selezionare la modalità di esecuzione **Esecuzione di cicli/Esecuzione Condizionale**.

Qualsiasi requisito dell'esecuzione condizionale che si definisce avrà effetto quando si eseguirà il flusso, se è stata impostata la modalità di esecuzione **Esecuzione di Cicli/Condizionale**. Facoltativamente, è possibile generare il codice dello script per i propri requisiti di esecuzione condizionale e incollarlo nell'editor dello script facendo clic su **Incolla...** nell'angolo destro in basso della sottoscheda **Condizionale**; la scheda principale **Esecuzione** visualizza le modifiche da mostrare nella modalità di esecuzione **Default (script facoltativo)** con lo script nella parte in alto della scheda. Questo significa che è possibile definire le condizioni utilizzando le varie finestre di dialogo delle opzioni per l'esecuzione dei cicli prima di generare uno script che è possibile personalizzare ulteriormente nell'editor dello script. Si noti che quando si fa clic su **Incolla...** qualunque requisito che è stato definito per l'esecuzione dei cicli sarà visualizzato nello script generato.

Per impostare una condizione:

1. Nella colonna a destra della scheda secondaria Condizionale, fare clic sul pulsante Aggiungi nuova condizione  per aprire la finestra di dialogo Aggiungi istruzione di esecuzione condizionale. In questa finestra di dialogo viene specificata la condizione che deve essere soddisfatta per far sì che il nodo venga eseguito.
2. Nella finestra di dialogo Aggiungi istruzione di esecuzione condizionale, specificare quanto segue:
 - a. **Nodo.** Selezionare il nodo per cui si desidera impostare un'esecuzione condizionale. Fare clic sul pulsante sfoglia per aprire la finestra di dialogo Seleziona Nodo e scegliere il nodo che si desidera; se vi sono troppi nodi elencati, è possibile filtrare la visualizzazione per mostrare i nodi da una delle seguenti categorie: Nodo Esporta, Grafico, Modello o Output.
 - b. **Condizione basata su.** Specificare la condizione che deve essere soddisfatta per il nodo da eseguire. È possibile scegliere tra quattro opzioni: **Parametri flusso**, **Variabile globale**, **Cella tabella di output** oppure **Sempre vero**. I dettagli immessi nella metà inferiore della finestra di dialogo vengono controllati dalle condizioni scelte.
 - **Parametri di flusso.** Selezionare il parametro dall'elenco disponibile e quindi scegliere **Operatore** per quel parametro; per esempio, l'operatore potrebbe essere Maggiore di, Uguale, Minore, Tra e così via. Quindi immettere il **Valore**, o i valori minimo e massimo, in base all'operatore.
 - **Variabile globale.** Selezionare la variabile dall'elenco disponibile; per esempio, potrebbe essere: Media, Somma, Valore minimo, Valore massimo oppure Deviazione standard. Quindi selezionare il campo **Operatore** ed i valori richiesti.
 - **Cella tabella di output.** Selezionare il nodo tabella dall'elenco disponibile e quindi scegliere **Riga** e **Colonna** nella tabella. Quindi selezionare il campo **Operatore** ed i valori richiesti.
 - **Sempre vero.** Selezionare questa opzione se il nodo deve essere sempre eseguito. Se si seleziona questa opzione, non ci sono ulteriori parametri da selezionare.
3. Ripetere i passi 1 e 2 il numero di volte necessario all'impostazione di tutte le condizioni richieste. Il nodo selezionato e la condizione da rispettare prima che il nodo venga eseguito, sono mostrati nel corpo principale della sottoscheda rispettivamente nelle colonne **Nodo di esecuzione** e **Se questa condizione è vera**.
4. Per default, i nodi e le condizioni vengono eseguite nell'ordine di visualizzazione; per spostare un nodo o una condizione su o giù nell'elenco, selezionarlo con un clic e quindi utilizzare le frecce su o giù nella colonna a destra della sottoscheda per modificare l'ordine.

Inoltre, è possibile impostare le seguenti opzioni nella parte inferiore della sottoscheda Condizionale:

- **Valuta tutti in ordine.** Selezionare questa opzione per valutare ogni condizione nell'ordine in cui sono visualizzate nella sottoscheda. I nodi per i quali le condizioni vengono verificate essere "Vero" saranno tutti eseguiti una volta che tutte le condizioni sono state valutate.
- **Esegui uno alla volta.** Disponibile solo se è stato selezionato **Valuta tutti in ordine**. Selezionando questa opzione significa che, se una condizione viene valutata come "Vera", il nodo associato con quella condizione viene eseguito prima che la condizione successiva venga valutata.
- **Valuta fino al primo risultato.** Selezionando questa opzione significa che sarà eseguito solo il primo nodo che ritorna una valutazione "Vero" dalle condizioni specificate.

Esecuzione e interruzione degli script

Sono disponibili diversi sistemi per l'esecuzione degli script. Per esempio, nello script del flusso o nella finestra di dialogo dello script autonomo, il pulsante "Esegui questo script" esegue lo script completo:



Figura 1. Pulsante Esegui questo script

Il pulsante "Esegui solo righe selezionate" esegue una sola riga o un blocco di righe adiacenti selezionate nello script:



Figura 2. Pulsante Esegui solo righe selezionate

Per eseguire gli script è possibile utilizzare i metodi seguenti:

- Fare clic sul pulsante "Esegui questo script" o "Esegui solo righe selezionate" all'interno dello script di un flusso o nella finestra di dialogo dello script autonomo.
- Eseguire un flusso nel quale il metodo di esecuzione predefinito impostato è **Esegui questo script**.
- Utilizzare il flag `-execute` all'avvio in modalità interattiva. Per ulteriori informazioni, consultare l'argomento "Utilizzo degli argomenti della riga di comando" a pagina 61.

Nota: lo script del Supernodo viene eseguito insieme al Supernodo se nella finestra di dialogo Script Supernodo è stata selezionata l'opzione **Esegui questo script**.

Interruzione dell'esecuzione degli script

Nella finestra di dialogo dello script di un flusso, il pulsante rosso Interrompi viene attivato durante l'esecuzione dello script. Questo pulsante consente di interrompere l'esecuzione dello script e di qualsiasi stream corrente.

Trova e sostituisci

La finestra di dialogo Trova/Sostituisci è disponibile ogni volta che è possibile modificare il testo di script o di espressioni, compreso l'editor di script, il generatore di espressioni CLEM e quando si definisce un modello nel nodo Report. Quando si modifica un testo in una di queste aree, premere `Ctrl+F` per accedere alla finestra di dialogo, assicurandosi che il cursore sia posizionato in un'area di testo. In un nodo Riempimento, per esempio, è possibile accedere alla finestra di dialogo da qualsiasi area di testo della scheda Impostazioni oppure dal campo testo nel Generatore di espressioni.

1. Con il cursore posizionato in un'area di testo, premere `Ctrl+F` per accedere alla finestra di dialogo Trova/Sostituisci.
2. Immettere il testo da cercare oppure sceglierne uno dall'elenco a discesa degli elementi cercati di recente.
3. Se necessario, immettere il testo sostitutivo.
4. Fare clic su **Trova successivo** per avviare la ricerca.
5. Fare clic su **Sostituisci** per sostituire la selezione corrente oppure scegliere **Sostituisci tutto** per aggiornare tutte le istanze o quelle selezionate.
6. Al termine di ogni operazione, la finestra di dialogo si chiude. Premere `F3` da qualsiasi area di testo per ripetere l'ultima operazione di ricerca oppure premere `Ctrl+F` per accedere nuovamente alla finestra di dialogo.

Opzioni di ricerca

Caratteri maiuscoli/minuscoli. Specifica se l'operazione di ricerca fa distinzione tra caratteri maiuscoli/minuscoli, per esempio se *miavar* corrisponde a *miaVar*. Il testo sostitutivo viene sempre inserito esattamente come viene digitato, indipendentemente da questa impostazione.

Solo parole intere. Specifica se l'operazione di ricerca cerca le occorrenze che sono parole intere. Se questa opzione è selezionata, la ricerca di *palla* non consentirà di trovare per esempio *pallavolo* o *Palladio*.

Espressioni regolari. Specifica se è utilizzata la sintassi delle espressioni regolari (vedere la sezione seguente). Quando questa opzione è selezionata, l'opzione **Solo parole intere** è disattivata e il relativo valore viene ignorato.

Solo testo selezionato. Controlla l'ambito della ricerca quando si utilizza l'opzione **Sostituisci tutto**.

Sintassi delle espressioni regolari

Le espressioni regolari consentono di cercare caratteri speciali, quali tabulazioni o caratteri di nuova riga, classi o intervalli di caratteri quali *a - d*, cifre e caratteri diversi da cifre, nonché limiti, per esempio l'inizio o la fine di una riga. Sono supportati i seguenti tipi di espressioni.

Tabella 1. Corrispondenze di caratteri.

| Caratteri | Corrispondenze |
|-----------|--|
| x | Il carattere x |
| \\ | Il carattere barra rovesciata |
| \0n | Il carattere con valore ottale 0n (0 <= n <= 7) |
| \0nn | Il carattere con valore ottale 0nn (0 <= n <= 7) |
| \0mnn | Il carattere con valore ottale 0mnn (0 <= m <= 3, 0 <= n <= 7) |
| \xhh | Il carattere con valore esadecimale 0xhh |
| \uhhhh | Il carattere con valore esadecimale 0xhhhh |
| \t | Il carattere di tabulazione ('\u0009') |
| \n | Il carattere di nuova riga (avanzamento riga) ('\u000A') |
| \r | Il carattere di ritorno a capo ('\u000D') |
| \f | Il carattere di avanzamento carta ('\u000C') |
| \a | Il carattere di avviso (campanello) ('\u0007') |
| \e | Il carattere di escape ('\u001B') |
| \cx | Il carattere di controllo corrispondente a x |

Tabella 2. Corrispondenze di classi di caratteri.

| Classi di caratteri | Corrispondenze |
|---------------------|--|
| [abc] | a, b o c (classe semplice) |
| [^abc] | Qualsiasi carattere, eccetto a, b o c (sottrazione) |
| [a-zA-Z] | a-z oppure A-Z, incluse (intervallo) |
| [a-d[m-p]] | a-d oppure m-p (unione). In alternativa è possibile specificare [a-dm-p]. |
| [a-z&&[def]] | a-z + d, e oppure f (intersezione) |
| [a-z&&[^bc]] | a-z, eccetto b e c (sottrazione). In alternativa è possibile specificare [ad-z]. |
| [a-z&&[^m-p]] | a-z, eccetto m-p (sottrazione). In alternativa è possibile specificare [a-lq-z]. |

Tabella 3. Classi di caratteri predefinite.

| Classi di caratteri predefinite | Corrispondenze |
|---------------------------------|---|
| . | Qualsiasi carattere (può corrispondere o meno a terminazioni di riga) |
| \d | Qualsiasi cifra: [0-9] |
| \D | Un carattere diverso da una cifra: [^0-9] |
| \s | Un spazio vuoto: [\n\x0B \t \f\r] |
| \S | Uno spazio non vuoto: [^\s] |
| \w | Un carattere alfanumerico: [a-zA-Z_0-9] |
| \W | Un carattere diverso da alfanumerico: [^\w] |

Tabella 4. Corrispondenze di limiti.

| Corrispondenze di limiti | Corrispondenze |
|--------------------------|--|
| ^ | L'inizio di una riga |
| \$ | La fine di una riga |
| \b | Un limite di parola |
| \B | Un limite diverso da un limite di parola |
| \A | L'inizio dell'input |
| \Z | La fine dell'input ma per la terminazione finale, se disponibile |
| \z | La fine dell'input |

Capitolo 2. Linguaggio di script

Panoramica sul linguaggio di script

La funzione di script per IBM SPSS Modeler consente di creare script che eseguono operazioni sull'interfaccia utente di SPSS Modeler, modificano gli oggetti di output ed eseguono sintassi dei comandi. È possibile eseguire gli script direttamente dall'interno di SPSS Modeler.

Gli script in IBM SPSS Modeler sono scritti nel linguaggio di script Python. L'implementazione di Python basata su Java utilizzata da IBM SPSS Modeler è denominata Jython. Il linguaggio di script dispone delle seguenti funzioni:

- Un formato per i riferimenti a nodi, flussi, progetti, output e altri oggetti IBM SPSS Modeler.
- Un insieme di istruzioni o comandi di script che può essere utilizzato per la manipolazione di questi oggetti.
- Un linguaggio di espressioni script per l'impostazione dei valori di variabili, parametri e altri oggetti.
- Supporto per commenti, continuazioni e blocchi di testo letterale.

Le sezioni riportate di seguito descrivono il linguaggio di script Python, l'implementazione Jython di Python e la sintassi di base per iniziare ad utilizzare gli script all'interno di IBM SPSS Modeler. Le sezioni che seguono contengono informazioni su proprietà e comandi specifici.

Python e Jython

Jython è un'implementazione del linguaggio di script Python, scritto nel linguaggio Java ed integrato con la piattaforma Java. Python è un potente linguaggio di script orientato agli oggetti. Jython è utile perché fornisce le funzioni di produttività di un solido linguaggio di script e, a differenza di Python, viene eseguito in un ambiente che supporta una JVM (Java virtual machine). Ciò significa che le librerie Java sulla JVM sono disponibili per l'utilizzo durante la scrittura di programmi. Con Jython, è possibile sfruttare questa differenza ed utilizzare la sintassi e la maggior parte delle funzioni del linguaggio Python

Come linguaggio di script, Python (e la relativa implementazione Jython) è semplice da apprendere ed efficace da codificare e dispone della struttura minima richiesta per creare un programma in esecuzione. Il codice può essere immesso in modo interattivo, vale a dire una riga alla volta. Python è un linguaggio di script interpretato; non è disponibile alcun passo di precompilazione, come in Java. I programmi Python sono semplicemente file di testo interpretati al momento dell'input (dopo l'analisi degli errori di sintassi). Le espressioni semplici, come i valori definiti, e le azioni più complesse, come le definizioni di funzioni, sono immediatamente eseguite e disponibili per l'utilizzo. Le modifiche apportate al codice possono essere verificate rapidamente. Tuttavia, l'interpretazione degli script presenta alcuni svantaggi. Ad esempio, l'utilizzo di una variabile non definita non è un errore del compilatore, per cui viene rilevato solo se (e quando) viene eseguita l'istruzione in cui viene utilizzata la variabile. In questo caso, il programma può essere modificato ed eseguito per il debug dell'errore.

Python considera tutti gli elementi, inclusi tutti i dati e tutto il codice, come un oggetto. Pertanto, è possibile modificare tali oggetti con righe di codice. Alcuni tipi di selezione, come numeri e stringhe, vengono considerati come valori e non come oggetti; questa modalità è supportata in Python. È supportato un valore null. Tale valore null ha il nome riservato None.

Per un'introduzione più approfondita agli script Python e Jython e per alcuni script di esempio, consultare <http://www.ibm.com/developerworks/java/tutorials/j-jython1/j-jython1.html> e <http://www.ibm.com/developerworks/java/tutorials/j-jython2/j-jython2.html>.

Script Python

Questa guida al linguaggio di script Python rappresenta un'introduzione ai componenti che hanno maggiori probabilità di essere utilizzati durante la creazione di script in IBM SPSS Modeler, inclusi concetti ed elementi di base della programmazione. Ciò fornirà una serie di informazioni sufficienti per iniziare a sviluppare i propri script Python da utilizzare all'interno di IBM SPSS Modeler.

Operazioni

L'assegnazione viene eseguita utilizzando il simbolo di uguaglianza (=). Ad esempio, per assegnare il valore "3" ad una variabile denominata "x", viene utilizzata la seguente istruzione:

```
x = 3
```

Il simbolo di uguaglianza viene utilizzato anche per assegnare dati di tipo stringa ad una variabile. Ad esempio, per assegnare il valore "a string value" alla variabile "y", viene utilizzata la seguente istruzione:

```
y = "a string value"
```

La tabella riportata di seguito elenca alcune delle operazioni numeriche e di confronto utilizzate frequentemente e le relative descrizioni.

Tabella 5. Operazioni numeriche e di confronto comuni

| Operazione | Descrizione |
|------------|----------------------------|
| $x < y$ | x è minore di y? |
| $x > y$ | x è maggiore di y? |
| $x \leq y$ | x è minore o uguale a y? |
| $x \geq y$ | x è maggiore o uguale a y? |
| $x == y$ | x è uguale a y? |
| $x != y$ | x non è uguale a y? |
| $x <> y$ | x non è uguale a y? |
| $x + y$ | Aggiungi y a x |
| $x - y$ | Sottrai y da x |
| $x * y$ | Moltiplica x per y |
| x / y | Dividi x per y |
| $x ** y$ | Eleva x alla potenza y |

Elenchi

Gli elenchi sono sequenze di elementi. Un elenco può contenere qualsiasi numero di elementi e gli elementi dell'elenco possono essere oggetti di qualsiasi tipo. È possibile pensare agli elenchi anche come ad array. Il numero di elementi in un elenco può aumentare o diminuire man mano che gli elementi vengono aggiunti, rimossi o sostituiti.

Esempi

| | |
|--------------------------------------|--|
| <code>[]</code> | Qualsiasi elenco vuoto. |
| <code>[1]</code> | Un elenco con un elemento singolo, un intero. |
| <code>["Mike", 10, "Don", 20]</code> | Un elenco con quattro elementi, due elementi stringa e due elementi interi. |
| <code>[[], [7], [8, 9]]</code> | Un elenco di elenchi. Ciascun elenco secondario è un elenco vuoto oppure un elenco di elementi interi. |

```
x = 7; y = 2; z = 3;
[1, x, y, x + y]
```

Un elenco di interi. Questo esempio illustra l'utilizzo di variabili ed espressioni.

È possibile assegnare un elenco ad una variabile; ad esempio:

```
mylist1 = ["one", "two", "three"]
```

È quindi possibile accedere ad elementi specifici dell'elenco, ad esempio:

```
mylist[0]
```

Il risultato sarà l'output riportato di seguito:

```
one
```

Il numero nelle parentesi quadre ([]) è conosciuto come *indice* e fa riferimento ad un particolare elemento dell'elenco. Gli elementi di un elenco sono indicizzati a partire da 0.

È anche possibile selezionare un intervallo di elementi di un elenco; questa operazione è definita *sezionamento*. Ad esempio, `x[1:3]` seleziona il secondo e terzo elemento di `x`. L'indice finale è un'unità dopo la selezione.

Stringhe

Una *stringa* è una sequenza immutabile di caratteri considerata come un valore. Le stringhe supportano tutti gli operatori e le funzioni di sequenza che risultano in una nuova stringa. Ad esempio, `"abcdef"[1:4]` ha come risultato l'output `"bcd"`.

In Python, i caratteri sono rappresentati da stringhe di lunghezza uno.

I literal di stringa sono definiti mediante l'utilizzo di tripli o singoli apici. Le stringhe definite utilizzando apici singoli non possono essere suddivise su più righe, al contrario delle stringhe definite utilizzando tripli apici. Una stringa può essere racchiusa tra apici singoli (') o doppi ("). Un carattere di quotatura può contenere l'altro carattere di quotatura senza carattere di escape o il carattere di quotatura con carattere di escape, preceduto dal carattere barra retroversa (\).

Esempi

```
"This is a string"
'This is also a string'
"It's a string"
'This book is called "Python Scripting and Automation Guide".'
"This is an escape quote (\") in a quoted string"
```

Più stringhe separate da spazi vengono automaticamente concatenate dal parser Python. In questo modo è più semplice immettere stringhe estese e combinare tipi di apici in una singola stringa, come ad esempio:

```
"This string uses ' and " 'that string uses ".'
```

Si ottiene il seguente risultato:

```
This string uses ' and that string uses ".
```

Le stringhe supportano diversi metodi utili. Alcuni di tali metodi sono indicati nella tabella riportata di seguito.

Tabella 6. Metodi stringa

| Metodo | Utilizzo |
|-----------------------------|------------------------------|
| <code>s.capitalize()</code> | Lettera maiuscola iniziale s |

Tabella 6. Metodi stringa (Continua)

| Metodo | Utilizzo |
|--|---|
| s.count(ss {,start {,end}}) | Conta le ricorrenze di ss in s[start:end] |
| s.startswith(str {, start {, end}}) s.endswith(str {, start {, end}}) | Verifica se s inizia con str Verifica se s termina con str |
| s.expandtabs({size}) | Sostituisce le tabulazioni con gli spazi; il valore size predefinito è 8 |
| s.find(str {, start {, end}}) s.rfind(str {, start {, end}}) | Individua il primo indice di str in s; se non trovato, il risultato è -1. rfind esegue la ricerca da destra a sinistra. |
| s.index(str {, start {, end}}) s.rindex(str {, start {, end}}) | Trova il primo indice di str in s; se non trovato, genera ValueError. rindex esegue la ricerca da destra a sinistra. |
| s.isalnum | Verifica se la stringa è alfanumerica |
| s.isalpha | Verifica se la stringa è alfabetica |
| s.isnum | Verifica se la stringa è numerica |
| s.isupper | Verifica se la stringa contiene tutte lettere maiuscole. |
| s.islower | Verifica se la stringa contiene tutte lettere minuscole |
| s.isspace | Verifica se la stringa contiene tutti spazi vuoti. |
| s.istitle | Verifica se la stringa è una sequenza di stringhe alfanumeriche con lettera maiuscola iniziale |
| s.lower() s.upper() s.swapcase() s.title() | Converte in lettere minuscole Converte in lettere maiuscole Converte nel caso opposto Converte in caratteri del titolo |
| s.join(seq) | Unisce le stringhe in seq con s come separatore |
| s.splitlines({keep}) | Suddivide s in righe, se keep è true, conserva la nuove righe |
| s.split({sep {, max}}) | Suddivide s in "parole" utilizzando sep (il valore predefinito sep è uno spazio) per un massimo di max volte |
| s.ljust(width) s.rjust(width) s.center(width) s.zfill(width) | Giustifica la stringa a sinistra in un campo di larghezza width Giustifica la stringa a destra in un campo di larghezza width Giustifica la stringa al centro in un campo di larghezza width Per il riempimento, viene utilizzato 0. |
| s.lstrip() s.rstrip() s.strip() | Rimuove lo spazio vuoto iniziale Rimuove le spazio vuoto finale Rimuove lo spazio vuoto iniziale e finale |
| s.translate(str {,delc}) | Converte s utilizzando la tabella, dopo aver rimosso tutti i caratteri in delc. str deve essere una stringa con lunghezza == 256. |
| s.replace(old, new {, max}) | Sostituisce tutte o le max ricorrenze della stringa old con la stringa new |

Contrassegni

I contrassegni sono commenti introdotti dal carattere cancelletto (#). Tutto il testo che segue il carattere cancelletto sulla stessa riga viene considerato come parte del contrassegno e viene ignorato. Un contrassegno può iniziare in qualsiasi colonna. L'esempio riportato di seguito illustra l'utilizzo dei contrassegni:

```
#The HelloWorld application is one of the most simple
print 'Hello World' # print the Hello World line
```

Sintassi delle istruzioni

La sintassi delle istruzioni in Python è molto semplice. In generale, ciascuna riga di origine è una singola istruzione. Ad eccezione delle istruzioni `expression` e `assignment`, ciascuna istruzione è introdotta da una parola chiave, come `if` o `for`. È possibile inserire righe vuote o di contrassegno un qualsiasi punto tra le istruzioni nel codice. Se una riga contiene più di una istruzione, ciascuna istruzione deve essere separata mediante un punto e virgola (;).

Le istruzioni molto lunghe possono continuare su più righe. In questo caso, l'istruzione che deve continuare alla riga successiva deve terminare con una barra retroversa (\), ad esempio:

```
x = "A loooooooooooooooooooooooooong string" + \
    "another loooooooooooooooooooooooooong string"
```

Quando una struttura è racchiusa tra parentesi tonde (()), quadre ([]) o graffe ({}), l'istruzione può continuare su una riga successiva dopo qualsiasi virgola, senza che sia necessario inserire una barra retroversa; ad esempio:

```
x = (1, 2, 3, "hello",
    "goodbye", 4, 5, 6)
```

Identificativi

Gli identificativi vengono utilizzati per assegnare nomi a variabili, funzioni, classi e parole chiave. Gli identificativi possono avere qualsiasi lunghezza, ma devono iniziare con un carattere alfabetico maiuscolo o minuscolo o con il carattere di sottolineatura (_). I nomi che iniziano con il carattere di sottolineatura sono generalmente riservati a nomi interni o privati. Dopo il primo carattere, l'identificativo può contenere qualsiasi numero e combinazione di caratteri alfabetici, numeri da 0 a 9 ed il carattere di sottolineatura.

In Python, alcune parole riservate non possono essere utilizzate per assegnare nomi a variabili, funzioni o classi. Tali parole chiave sono suddivise nelle seguenti categorie:

- **Parole che introducono istruzioni:** `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `exec`, `finally`, `for`, `from`, `global`, `if`, `import`, `pass`, `print`, `raise`, `return`, `try` e `while`
- **Parole che introducono parametri:** `as`, `import` e `in`
- **Operatori:** `and`, `in`, `is`, `lambda`, `not` e `or`

Generalmente, un utilizzo non appropriato delle parole chiave determina un errore `SyntaxError`.

Blocchi di codice

I blocchi di codice sono gruppi di istruzioni utilizzati in punti in cui sono previste istruzioni singole. I blocchi di codice possono seguire tutte le istruzioni riportate di seguito: `if`, `elif`, `else`, `for`, `while`, `try`, `except`, `def` e `class`. Tali istruzioni introducono il blocco di codice con il carattere due punti (:), ad esempio:

```
if x == 1:
    y = 2
    z = 3
elif:
    y = 4
    z = 5
```

Per delimitare i blocchi di codice, viene utilizzato il rientro (invece delle parentesi graffe utilizzate in Java). Tutte le righe in un blocco devono essere rientrate alla stessa posizione. Questo perché una modifica del rientro indica la fine di un blocco di codice. Generalmente, vengono utilizzati rientri di quattro spazi per ciascun livello. Per il rientro delle righe, si consiglia di utilizzare gli spazi invece delle tabulazioni. Gli spazi e le tabulazioni non devono essere utilizzati contemporaneamente. Le righe nel blocco più esterno di un modulo devono iniziare alla colonna uno; in caso contrario, si verifica un errore `SyntaxError`.

Le istruzioni che costituiscono un blocco di codice (e seguono i due punti) possono essere anche su una riga singola, separate da punto e virgola, ad esempio:

```
if x == 1: y = 2; z = 3;
```

Passaggio di argomenti ad uno script

Il passaggio di argomenti ad uno script è utile perché significa che è possibile utilizzare più volte uno script senza modifiche. Gli argomenti passati sulla riga comandi vengono passati come valori nell'elenco `sys.argv`. È possibile ottenere il numero di valori passati utilizzando il comando `len(sys.argv)`. Ad esempio:

```
import sys
print "test1"
print sys.argv[0]
print sys.argv[1]
print len(sys.argv)
```

In questo esempio, il comando `import` importa l'intera classe `sys`, in modo che sia possibile utilizzare i metodi esistenti per tale classe, come, ad esempio, `argv`.

Lo script in questo esempio può essere richiamato utilizzando la riga riportata di seguito:

```
/u/mjloos/test1 mike don
```

Il risultato è il seguente output:

```
/u/mjloos/test1 mike don
test1
mike
don
3
```

Esempi

La parola chiave `print` stampa gli argomenti immediatamente successivi. Se l'istruzione è seguita da una virgola, nell'output non viene inclusa una nuova riga. Ad esempio:

```
print "This demonstrates the use of a",
print " comma at the end of a print statement."
```

Il risultato sarà l'output riportato di seguito:

```
This demonstrates the use of a comma at the end of a print statement.
```

L'istruzione `for` viene utilizzata per l'iterazione attraverso un blocco di codice. Ad esempio:

```
mylist1 = ["one", "two", "three"]
for lv in mylist1:
    print lv
    continue
```

In questo esempio, tre stringhe vengono assegnate all'elenco `mylist1`. Gli elementi dell'elenco vengono quindi stampati, con un elemento di ciascuna riga. Il risultato sarà l'output riportato di seguito:

```
one
two
three
```

In questo esempio, l'iteratore `lv` prende il valore di ciascun elemento nell'elenco `mylist1` man mano che il loop `for` implementa il blocco di codice per ciascun elemento. Un iteratore può essere qualsiasi identificativo valido di qualsiasi lunghezza.

L'istruzione `if` è un'istruzione condizionale. Valuta la condizione e restituisce `true` o `false`, in base al risultato della valutazione. Ad esempio:

```
mylist1 = ["one", "two", "three"]
for lv in mylist1:
    if lv == "two"
        print "The value of lv is ", lv
    else
        print "The value of lv is not two, but ", lv
    continue
```

In questo esempio, viene valutato il valore dell'iteratore `lv`. Se il valore di `lv` è `two`, viene restituita una stringa differente da quella restituita se il valore di `lv` è diverso da `two`. Si ottiene il seguente risultato:

```
The value of lv is not two, but one
The value of lv is two
The value of lv is not two, but three
```

Metodi matematici

Dal modulo `math`, è possibile accedere a utili metodi matematici. Alcuni di tali metodi sono indicati nella tabella riportata di seguito. Se non diversamente specificato, tutti i valori vengono restituiti come `float`.

Tabella 7. Metodi matematici

| Metodo | Utilizzo |
|----------------------------------|---|
| <code>math.ceil(x)</code> | Restituisce il limite superiore di <code>x</code> come <code>float</code> , vale a dire il più piccolo intero maggiore o uguale a <code>x</code> |
| <code>math.copysign(x, y)</code> | Restituisce <code>x</code> con il segno di <code>y</code> . <code>copysign(1, -0.0)</code> restituisce <code>-1</code> |
| <code>math.fabs(x)</code> | Restituisce il valore assoluto di <code>x</code> |
| <code>math.factorial(x)</code> | Restituisce <code>x</code> fattoriale. Se <code>x</code> è negativo o non è un intero, viene generato un errore <code>ValueError</code> . |
| <code>math.floor(x)</code> | Restituisce il limite inferiore di <code>x</code> come <code>float</code> , vale a dire l'intero più grande minore o uguale a <code>x</code> |
| <code>math.frexp(x)</code> | Restituisce la mantissa (<code>m</code>) e l'esponente (<code>e</code>) di <code>x</code> come coppia (<code>m, e</code>). <code>m</code> è un <code>float</code> ed <code>e</code> è un intero, in modo che <code>x == m * 2**e</code> esattamente. Se <code>x</code> è zero, restituisce <code>(0.0, 0)</code> , in caso contrario <code>0.5 <= abs(m) < 1</code> . |
| <code>math.fsum(iterable)</code> | Restituisce una somma <code>float</code> accurata dei valori in <code>iterable</code> |
| <code>math.isinf(x)</code> | Verifica se il <code>float</code> <code>x</code> è infinito positivo o negativo |
| <code>math.isnan(x)</code> | Verifica se il <code>float</code> <code>x</code> è <code>NaN</code> (not a number - non un numero) |
| <code>math.ldexp(x, i)</code> | Restituisce <code>x * (2**i)</code> . Questa è la funzione inversa della funzione <code>frexp</code> . |
| <code>math.modf(x)</code> | Restituisce le parti intera e frazionaria di <code>x</code> . Entrambi i risultati hanno il segno di <code>x</code> e sono <code>float</code> . |
| <code>math.trunc(x)</code> | Restituisce il valore <code>Real</code> <code>x</code> , troncato ad un <code>Integral</code> . |
| <code>math.exp(x)</code> | Restituisce <code>e**x</code> |
| <code>math.log(x[, base])</code> | Restituisce il logaritmo di <code>x</code> al valore fornito <code>base</code> . Se <code>base</code> non è specificato, viene restituito il logaritmo naturale di <code>x</code> . |

Tabella 7. Metodi matematici (Continua)

| Metodo | Utilizzo |
|-----------------------------|---|
| <code>math.log1p(x)</code> | Restituisce il logaritmo naturale di $1+x$ (base e) |
| <code>math.log10(x)</code> | Restituisce il logaritmo in base 10 di x |
| <code>math.pow(x, y)</code> | Restituisce x elevato alla potenza y . <code>pow(1.0, x)</code> e <code>pow(x, 0.0)</code> restituisce sempre 1, anche quando x è zero o NaN. |
| <code>math.sqrt(x)</code> | Restituisce la radice quadrata di x |

Oltre alle funzioni matematiche, sono presenti anche alcuni utili metodi trigonometrici. Tali metodi sono illustrati nella tabella riportata di seguito.

Tabella 8. Metodi trigonometrici

| Metodo | Utilizzo |
|-------------------------------|---|
| <code>math.acos(x)</code> | Restituisce l'arcocoseno di x in radianti |
| <code>math.asin(x)</code> | Restituisce l'arcoseno di x in radianti |
| <code>math.atan(x)</code> | Restituisce l'arcotangente di x in radianti |
| <code>math.atan2(y, x)</code> | Restituisce <code>atan(y / x)</code> in radianti. |
| <code>math.cos(x)</code> | Restituisce il coseno di x in radianti. |
| <code>math.hypot(x, y)</code> | Restituisce la norma Euclidea <code>sqrt(x*x + y*y)</code> . Questa è la lunghezza del vettore dall'origine al punto (x, y) . |
| <code>math.sin(x)</code> | Restituisce il seno di x in radianti |
| <code>math.tan(x)</code> | Restituisce la tangente di x in radianti |
| <code>math.degrees(x)</code> | Converte l'angolo x da radianti a gradi |
| <code>math.radians(x)</code> | Converte l'angolo x da gradi a radianti |
| <code>math.acosh(x)</code> | Restituisce il coseno iperbolico inverso di x |
| <code>math.asinh(x)</code> | Restituisce il seno iperbolico inverso di x |
| <code>math.atanh(x)</code> | Restituisce la tangente iperbolica inversa di x |
| <code>math.cosh(x)</code> | Restituisce il coseno iperbolico di x |
| <code>math.sinh(x)</code> | Restituisce il seno iperbolico di x |
| <code>math.tanh(x)</code> | Restituisce la tangente iperbolica di x |

Sono disponibili anche due costanti matematiche. Il valore di `math.pi` è la costante matematica π . Il valore di `math.e` è la costante matematica e .

Utilizzo di caratteri Non-ASCII

Per utilizzare caratteri non-ASCII, Python richiede una codifica esplicita e una decodifica di stringhe in Unicode. In IBM SPSS Modeler, gli script Python si assume che siano codificati in UTF-8, che è la codifica standard Unicode che supporta caratteri non-ASCII. Il seguente script verrà compilato poiché il compilatore Python è stato impostato a UTF-8 da SPSS Modeler.

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", "テストノード", 96, 64)
```

Tuttavia, il nodo risultante avrà un'etichetta errata.



ãfã, 'ãf^ãf ãf'ãf%

Figura 3. Etichetta del nodo contenente caratteri non-ASCII, visualizzata in modo errato

L'etichetta è errata perché la stessa stringa letterale è stata convertita in una stringa ASCII da Python.

Python consente alle stringhe letterali Unicode di essere specificate aggiungendo un carattere u come prefisso prima della stringa letterale:

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", u"テストノード", 96, 64)
```

Ciò creerà una stringa Unicode e l'etichetta verrà visualizzata correttamente.



テストノード

Figura 4. Etichetta del nodo contenente caratteri non-ASCII, visualizzata correttamente

L'utilizzo di Python e Unicode è un argomento vasto che va oltre l'ambito di questo documento. Sono disponibili molti libri e risorse online che comprendono questo argomento in grande dettaglio.

Programmazione orientata agli oggetti

La programmazione orientata agli oggetti è basata sul concetto di creazione di un modello del problema di destinazione nei propri programmi. La programmazione orientata agli oggetti riduce gli errori di programmazione e favorisce il riutilizzo del codice. Python è un linguaggio orientato agli oggetti. Gli oggetti definiti in Python hanno le seguenti caratteristiche:

- **Identità.** Ciascun oggetto deve essere distinto e verificabile. A questo scopo, sono disponibili i test `is` e `is not`.
- **Stato.** Ciascun oggetto deve essere in grado di memorizzare lo stato. A questo scopo, sono disponibili gli attributi, come i campi e le variabili dell'istanza.
- **Comportamento.** Ciascun oggetto deve essere in grado di modificare il proprio stato. A questo scopo, sono disponibili alcuni metodi.

Python include le seguenti funzioni per il supporto della programmazione orientata agli oggetti:

- **Creazione di oggetti basati su classi.** Le classi sono modelli per la creazione degli oggetti. Gli oggetti sono strutture di dati con un comportamento associato.
- **Ereditarietà con polimorfismo.** Python supporta l'ereditarietà singola e multipla. Tutti i metodi dell'istanza Python sono polimorfici e possono essere sovrascritti dalle classi secondarie.
- **Incapsulamento con dati nascosti.** Python consente di nascondere gli attributi. Quando nascosti, è possibile accedere agli attributi dall'esterno della classe solo attraverso i metodi della classe. Le classi implementano i metodi per modificare i dati.

Definizione di una classe

All'interno di una classe Python, è possibile definire variabili e metodi. A differenza di Java, in Python è possibile definire qualsiasi numero di classi pubbliche per file di origine (o *modulo*). Quindi, un modulo in Python può essere considerato simile ad un package in Java.

In Python, le classi sono definite utilizzando l'istruzione `class`. Di seguito è riportato il formato dell'istruzione `class`:

```
class name (superclasses): statement
```

o

```
class name (superclasses):  
    assignment  
    .  
    .  
    function  
    .  
    .
```

Quando viene definita una classe, è possibile fornire zero o più istruzioni *assignment*. Tali istruzioni creano attributi della classe condivisi da tutte le istanze della classe. È anche possibile fornire zero o più definizioni *function*. Tali definizioni di funzione creano metodi. L'elenco delle superclassi è facoltativo.

Il nome della classe deve essere univoco nello stesso ambito, vale a dire all'interno di un modulo, una funzione o una classe. È possibile definire più variabili per fare riferimento alla stessa classe.

Creazione di un'istanza della classe

Le classi vengono utilizzate per conservare gli attributi della classe (o condivisi) oppure per creare istanze della classe. Per creare un'istanza di una classe, è possibile richiamare la classe come se fosse una funzione. Ad esempio, considerare la classe riportata di seguito:

```
class MyClass:  
    pass
```

In questo caso, viene utilizzata l'istruzione `pass` perché è necessaria un'istruzione per completare la classe, ma non è richiesta alcuna azione in modo programmatico.

L'istruzione riportata di seguito crea un'istanza della classe `MyClass`:

```
x = MyClass()
```

Aggiunta di attributi ad un'istanza della classe

A differenza di Java, in Python i client possono aggiungere attributi ad un'istanza di una classe. Viene modificata solo quella particolare istanza. Ad esempio, per aggiungere attributi ad un'istanza `x`, impostare nuovi valori su tale istanza:

```
x.attr1 = 1  
x.attr2 = 2  
    .  
    .  
x.attrN = n
```

Definizione dei metodi e degli attributi della classe

Qualsiasi variabile collegata in una classe è un *attributo della classe*. Qualsiasi funzione definita all'interno di una classe è un *metodo*. I metodi ricevono un'istanza della classe, denominata per convenzione `self`, come primo argomento. Ad esempio, per definire alcuni metodi ed attributi della classe, è possibile utilizzare il codice riportato di seguito:

```

class MyClass
    attr1 = 10          #class attributes
    attr2 = "hello"

    def method1(self):
        print MyClass.attr1  #reference the class attribute

    def method2(self):
        print MyClass.attr2  #reference the class attribute

    def method3(self, text):
        self.text = text     #instance attribute
        print text, self.text #print my argument and my attribute

    method4 = method3      #make an alias for method3

```

All'interno di una classe, è necessario qualificare tutti i riferimenti agli attributi della classe con il nome della classe; ad esempio, `MyClass.attr1`. Tutti i riferimenti agli attributi dell'istanza devono essere qualificati con la variabile `self`; ad esempio, `self.text`. All'esterno della classe, è necessario qualificare tutti i riferimenti agli attributi della classe con il nome della classe (ad esempio, `MyClass.attr1`) oppure con un'istanza della classe (ad esempio, `x.attr1`, dove `x` è un'istanza della classe). All'esterno della classe, tutti i riferimenti alle variabili dell'istanza devono essere qualificati con un'istanza della classe; ad esempio, `x.text`.

Variabili nascoste

È possibile nascondere i dati creando variabili *Private*. Alle variabili Private può accedere solo la classe stessa. Se vengono dichiarati nomi nel formato `__xxx` o `__xxx_yyy`, vale a dire con due caratteri di sottolineatura iniziali, il programma di analisi Python aggiunge automaticamente il nome della classe al nome dichiarato, creando variabili nascoste; ad esempio:

```

class MyClass:
    __attr = 10    #private class attribute

    def method1(self):
        pass

    def method2(self, p1, p2):
        pass

    def __privateMethod(self, text):
        self.__text = text    #private attribute

```

A differenza di Java, in Python tutti i riferimenti alle variabili dell'istanza devono essere qualificati con `self`; non è previsto l'utilizzo implicito di `this`.

Ereditarietà

La possibilità di ereditare dalle classi è fondamentale per la programmazione orientata ad oggetti. Python supporta l'ereditarietà singola e multipla. Il termine *ereditarietà singola* indica che può esistere una sola superclasse. Il termine *ereditarietà multipla* indica che può essere più di una superclasse.

L'ereditarietà viene implementata inserendo altre classi come sottoclassi. Qualsiasi numero di classi Python possono essere superclassi. Nell'implementazione Python di Python, è possibile ereditare direttamente o indirettamente solo da una classe Java. Non è necessario fornire una superclasse.

Qualsiasi attributo o metodo in una superclasse è anche in qualsiasi sottoclasse e può essere utilizzato dalla classe stessa o da qualsiasi client, purché l'attributo o il metodo non sia nascosto. Qualsiasi istanza di una sottoclasse può essere utilizzata in qualsiasi punto in cui può essere utilizzata un'istanza di una superclasse; questo è un esempio di *polimorfismo*. Tali funzioni abilitano il riutilizzo e la semplicità di estensione.

Esempio

```
class Class1: pass    #no inheritance
class Class2: pass
class Class3(Class1): pass    #single inheritance
class Class4(Class3, Class2): pass    #multiple inheritance
```

Capitolo 3. Script in IBM SPSS Modeler

Tipi di script

In IBM SPSS Modeler sono disponibili tre tipi di script:

- Gli *script del flusso* sono utilizzati per controllare l'esecuzione di un singolo flusso e sono archiviati all'interno del flusso.
- Gli *script del Supernodo* vengono utilizzati per controllare il funzionamento dei supernodi.
- Gli *script autonomi o di sessione* possono essere utilizzati per coordinare l'esecuzione attraverso un numero di flussi differenti.

Sono disponibili diversi metodi che possono essere utilizzati negli script in IBM SPSS Modeler con cui è possibile accedere ad una vasta gamma di funzionalità di SPSS Modeler. Tali metodi sono utilizzati anche in Capitolo 4, "API di script", a pagina 37 per creare funzioni più avanzate.

Flussi, flussi SuperNodo e diagrammi

Il più delle volte il termine *flusso* significa la stessa cosa, indipendentemente se si tratta di un flusso caricato da un file o utilizzato all'interno di un SuperNodo. Generalmente indica un insieme di nodi che sono connessi insieme e possono essere eseguiti. Negli script, comunque, non tutte le operazioni sono supportate in tutti i posti, vale a dire che un autore di uno script dovrebbe essere consapevole di quale variante di flusso si sta utilizzando.

Flussi

Un flusso è il tipo di documento principale di IBM SPSS Modeler. Può essere salvato, caricato, modificato ed eseguito. I flussi possono anche avere parametri, valori globali, uno script ed altre informazioni ad essi associati.

Flussi SuperNodo

Un *flusso supernodo* è un tipo di flusso utilizzato all'interno di un supernodo. Come un flusso normale, contiene i nodi che sono collegati tra di loro. I flussi di supernodo hanno una serie di differenze rispetto ad un normale flusso:

- I parametri ed ogni script sono associati con il supernodo proprietario del flusso del supernodo, piuttosto che con il flusso del supernodo stesso.
- I flussi di supernodo hanno dei nodi connettori aggiuntivi di input ed output, a seconda del tipo di supernodo. Questi nodi connettori sono utilizzati per passare le informazioni in entrata ed in uscita al flusso del supernodo e vengono automaticamente creati quando viene creato il supernodo stesso.

Diagrammi

Il termine *diagramma* comprende le funzioni che sono supportate sia dai flussi normali che dai flussi SuperNodo, come aggiungere e rimuovere nodi e modificare le connessioni tra i nodi.

Esecuzione di un flusso

L'esempio riportato di seguito esegue tutti i nodi eseguibili nel flusso e rappresenta il tipo più semplice di script del flusso:

```
modeler.script.stream().runAll(None)
```

L'esempio riportato di seguito, inoltre, esegue tutti i nodi eseguibili nel flusso:

```
stream = modeler.script.stream()
stream.runAll(None)
```

In questo esempio, il flusso è memorizzato in una variabile denominata `stream`. L'archiviazione del flusso in una variabile è utile perché uno script viene generalmente utilizzato per modificare il flusso o i nodi all'interno di un flusso. La creazione di una variabile che memorizza il flusso ha come risultato uno script più breve.

Contesto di script

Il modulo `modeler.script` fornisce il contesto in cui viene eseguito uno script. Il modulo viene importato automaticamente in uno script SPSS Modeler al runtime. Il modulo definisce quattro funzioni che forniscono ad uno script l'accesso al proprio ambiente di esecuzione:

- La funzione `session()` restituisce la sessione per lo script. La sessione definisce informazioni come la locale ed il backend the SPSS Modeler (un processo locale o un SPSS Modeler Server di rete) utilizzati per l'esecuzione dei flussi.
- La funzione `stream()` può essere utilizzata con script di supernodi e flussi. Questa funzione restituisce il flusso proprietario dello script del flusso o dello script del supernodo eseguito.
- La funzione `diagram()` può essere utilizzata con gli script del supernodo. Questa funzione restituisce il diagramma all'interno del supernodo. Per gli altri tipi di script, questa funzione restituisce lo stesso risultato della funzione `stream()`.
- La funzione `supernode()` può essere utilizzata con gli script del supernodo. Questa funzione restituisce il supernodo proprietario dello script che viene eseguito.

Le quattro funzioni ed i relativi output sono riepilogati nella tabella riportata di seguito.

Tabella 9. Riepilogo delle funzioni di `modeler.script`

| Tipo di script | <code>session()</code> | <code>stream()</code> | <code>diagram()</code> | <code>supernode()</code> |
|----------------|--------------------------|--|---------------------------------|--------------------------|
| Autonomo | Restituisce una sessione | Restituisce il flusso gestito corrente nel momento in cui è stato richiamato lo script (ad esempio, il flusso passato mediante l'opzione <code>-stream</code> della modalità batch) oppure <code>None</code> . | Come <code>stream()</code> | Non applicabile |
| Flusso | Restituisce una sessione | Restituisce un flusso | Come <code>stream()</code> | Non applicabile |
| Supernodo | Restituisce una sessione | Restituisce un flusso | Restituisce un flusso Supernodo | Restituisce un supernodo |

Il modulo `modeler.script`, inoltre, definisce un modo per terminare lo script con un codice di uscita. La funzione `exit(exit-code)` arresta l'esecuzione dello script e restituisce il codice di uscita intero fornito.

Uno dei metodi definiti per un flusso è `runAll(List)`. Questo metodo esegue tutti i nodi eseguibili. Qualsiasi modello o output generato mediante l'esecuzione dei nodi viene aggiunto all'elenco fornito.

È comune per un'esecuzione di flusso generare output come modelli, grafici ed altro output. Per catturare tale output, uno script può fornire una variabile inizializzata in un un elenco, ad esempio:

```
stream = modeler.script.stream()
results = []
stream.runAll(results)
```

Una volta completata l'esecuzione, è possibile accedere a tutti gli oggetti generati dall'esecuzione dall'elenco `results`.

Riferimento a nodi esistenti

Spesso un flusso già dispone di alcuni parametri che è necessario modificare prima che il flusso venga eseguito. La modifica di tali parametri implica le attività riportate di seguito:

1. Individuazione dei nodi nel relativo flusso.
2. Modifica delle impostazioni del nodo o del flusso (o di entrambi).

Ricerca di nodi

I flussi forniscono diversi modi per ricercare un nodo esistente. Tali metodi sono riepilogati nella tabella riportata di seguito.

Tabella 10. Metodi per la ricerca di un nodo esistente

| Metodo | Tipo di restituzione | Descrizione |
|---|----------------------|--|
| <code>s.findAll(type, label)</code> | Raccolta | Restituisce un elenco di tutti i nodi con il tipo e l'etichetta specificati. Il tipo o l'etichetta possono essere <code>None</code> ; in questo caso, viene utilizzato l'altro parametro. |
| <code>s.findAll(filter, recursive)</code> | Raccolta | Restituisce una raccolta di tutti i nodi accettati dal filtro specificato. Se l'indicatore <code>recursive</code> è <code>True</code> , viene eseguita la ricerca anche nei supernodi all'interno del flusso specificato. |
| <code>s.findById(id)</code> | Nodo | Restituisce il nodo con l'ID fornito oppure <code>None</code> se non esiste alcun nodo di questo tipo. La ricerca è limitata al flusso corrente. |
| <code>s.findByType(type, label)</code> | Nodo | Restituisce il nodo con il tipo e/o l'etichetta forniti. Il tipo o il nome può essere <code>None</code> ; in questo caso viene utilizzato l'altro parametro. Se si verifica una corrispondenza per più nodi, viene scelto e restituito un nodo arbitrario. Se non si verifica alcuna corrispondenza, il valore di restituzione è <code>None</code> . |
| <code>s.findDownstream(fromNodes)</code> | Raccolta | Ricerca dall'elenco di nodi fornito e restituisce l'insieme di nodi downstream dei nodi forniti. L'elenco restituito include i nodi forniti originariamente. |
| <code>s.findUpstream(fromNodes)</code> | Raccolta | Ricerca dall'elenco di nodi fornito e restituisce l'insieme dei nodi upstream dei nodi forniti. L'elenco restituito include i nodi forniti originariamente. |

Ad esempio, se un flusso contiene un singolo nodo `Filter` per cui lo script richiede l'accesso, è possibile trovare il nodo `Filter` utilizzando lo script riportato di seguito:

```

stream = modeler.script.stream()
node = stream.findByType("filter", None)
...

```

In alternativa, per trovare il nodo è possibile utilizzare l'ID del nodo (visualizzato nella scheda Annotazioni della finestra di dialogo del nodo), se noto; ad esempio:

```

stream = modeler.script.stream()
node = stream.findById("id32FJT71G2") # the filter node ID
...

```

Impostazione delle proprietà

I nodi, i flussi, i modelli e gli output dispongono di proprietà a cui è possibile accedere e che, nella maggior parte dei casi, è possibile impostare. Generalmente, le proprietà sono utilizzate per modificare il funzionamento o l'aspetto dell'oggetto. I metodi disponibili per l'accesso e l'impostazione delle proprietà dell'oggetto sono riepilogati nella tabella riportata di seguito.

Tabella 11. Metodi per l'accesso e l'impostazione delle proprietà dell'oggetto

| Metodo | Tipo di restituzione | Descrizione |
|--|----------------------|--|
| <code>p.getPropertyValue(propertyName)</code> | Oggetto | Restituisce il valore della proprietà indicata o None se non esiste alcuna proprietà di questo tipo. |
| <code>p.setPropertyValue(propertyName, value)</code> | Non applicabile | Imposta il valore della proprietà indicata. |
| <code>p.setPropertyValues(properties)</code> | Non applicabile | Imposta i valori delle proprietà indicate. Ciascuna voce nella mappa delle proprietà è composta da una chiave che rappresenta il nome della proprietà e dal valore che deve essere assegnato a tale proprietà. |
| <code>p.getKeyedPropertyValue(propertyName, keyName)</code> | Oggetto | Restituisce il valore della proprietà denominata e la chiave associata o None se non esistono una proprietà o una chiave di questo tipo. |
| <code>p.setKeyedPropertyValue(propertyName, keyName, value)</code> | Non applicabile | Imposta il valore delle proprietà indicata e della chiave. |

Ad esempio, se si desidera impostare il valore di un nodo Variable File all'inizio di un flusso, è possibile utilizzare il seguente script:

```

stream = modeler.script.stream()
node = stream.findByType("variablefile", None)
node.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")
...

```

In alternativa, è possibile che si desideri filtrare un campo da un nodo Filter. In questo caso, il valore è anche associato con chiave al nome del campo; ad esempio:

```

stream = modeler.script.stream()
# Locate the filter node ...
node = stream.findByType("filter", None)
# ... and filter out the "Na" field
node.setKeyedPropertyValue("include", "Na", False)

```

Creazione di nodi e modifica dei flussi

In alcune situazioni, è possibile che si desideri aggiungere nuovi nodi ai flussi esistenti. L'aggiunta di nodi ai flussi esistenti generalmente implica le seguenti attività:

1. Creazione dei nodi.
2. Collegamento dei nodi al flusso esistente.

Creazione di nodi

I flussi forniscono diversi modi per creare i nodi. Tali metodi sono riepilogati nella tabella riportata di seguito.

Tabella 12. Metodi per la creazione di nodi

| Metodo | Tipo di restituzione | Descrizione |
|--|----------------------|--|
| <code>s.create(nodeType, name)</code> | Nodo | Crea un nodo del tipo specificato e lo aggiunge al flusso specificato. |
| <code>s.createAt(nodeType, name, x, y)</code> | Nodo | Crea un nodo del tipo specificato e lo aggiunge al flusso specificato nel percorso specificato. Se $x < 0$ oppure $y < 0$, il percorso non viene impostato. |
| <code>s.createModelApplier(modelOutput, name)</code> | Nodo | Crea un nodo applicatore del modello derivato dall'oggetto di output del modello fornito. |

Ad esempio, per creare un nuovo nodo Type in un flusso, è possibile utilizzare lo script riportato di seguito:

```
stream = modeler.script.stream()
# Create a new type node
node = stream.create("type", "My Type")
```

Collegamento e scollegamento di nodi

Quando un nuovo nodo viene creato all'interno di un flusso, deve essere connesso in una sequenza di nodi prima di poter essere utilizzato. I flussi forniscono diversi metodi per collegare e scollegare i nodi. Tali metodi sono riepilogati nella tabella riportata di seguito.

Tabella 13. Metodi per il collegamento e lo scollegamento dei nodi

| Metodo | Tipo di restituzione | Descrizione |
|--|----------------------|--|
| <code>s.link(source, target)</code> | Non applicabile | Crea un nuovo collegamento tra i nodi di origine e di destinazione. |
| <code>s.link(source, targets)</code> | Non applicabile | Crea nuovi collegamenti tra il nodo di origine e ciascun nodo di destinazione nell'elenco fornito. |
| <code>s.linkBetween(inserted, source, target)</code> | Non applicabile | Connette un nodo tra due altre istanze del nodo (i nodi di origine e destinazione) ed imposta la posizione del nodo inserito tra di essi. Qualsiasi collegamento diretto tra i nodi di origine e destinazione viene rimosso prima. |
| <code>s.linkPath(path)</code> | Non applicabile | Crea un nuovo percorso tra le istanze del nodo. Il primo nodo viene collegato al secondo, il secondo viene collegato al terzo e così via. |

Tabella 13. Metodi per il collegamento e lo scollegamento dei nodi (Continua)

| Metodo | Tipo di restituzione | Descrizione |
|-------------------------------|----------------------|--|
| s.unlink(source, target) | Non applicabile | Rimuove qualsiasi collegamento diretto tra i nodi di origine e di destinazione. |
| s.unlink(source, targets) | Non applicabile | Rimuove i collegamenti diretti tra il nodo di origine e ciascun oggetto nell'elenco delle destinazioni. |
| s.unlinkPath(path) | Non applicabile | Rimuove qualsiasi percorso esistente tra le istanze del nodo. |
| s.disconnect(node) | Non applicabile | Rimuove qualsiasi collegamento tra il nodo fornito e qualsiasi altro nodo nel flusso specificato. |
| s.isValidLink(source, target) | booleano | Restituisce True se è valido creare un collegamento tra l'origine specificata ed i nodi di destinazione. Questo metodo verifica che entrambi gli oggetti appartengano al flusso specificato, che il nodo di origine possa fornire un collegamento e che il nodo di destinazione possa ricevere un collegamento, e che la creazione di un collegamento di questo tipo non causi circolarità nel flusso. |

Lo script di esempio riportato di seguito esegue queste cinque attività:

1. Crea un nodo di input Variable File, un nodo Filter ed un nodo di output Table.
2. Connette i nodi tra loro.
3. Imposta il nome del file sul nodo di input Variable File.
4. Filtra il campo "Drug" dall'output risultante.
5. Esegue il nodo Table.

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", "My File Input ", 96, 64)
filternode = stream.createAt("filter", "Filter", 192, 64)
tablenode = stream.createAt("table", "Table", 288, 64)
stream.link(filenode, filternode)
stream.link(filternode, tablenode)
filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
filternode.setKeyedPropertyValue("include", "Drug", False)
results = []
tablenode.run(results)
```

Importazione, sostituzione ed eliminazione di nodi

Oltre alla creazione ed alla connessione dei nodi, è spesso necessario sostituire ed eliminare nodi dal flusso. I metodi disponibili per l'importazione, la sostituzione e l'eliminazione dei nodi sono riepilogati nella seguente tabella.

Tabella 14. Metodi per l'importazione, la sostituzione e l'eliminazione dei nodi

| Metodo | Tipo di restituzione | Descrizione |
|---|----------------------|---|
| s.replace(originalNode, replacementNode, discardOriginal) | Non applicabile | Sostituisce il nodo specificato dal flusso specificato. Il nodo originale ed il nodo sostitutivo devono essere di proprietà del flusso specificato. |

Tabella 14. Metodi per l'importazione, la sostituzione e l'eliminazione dei nodi (Continua)

| Metodo | Tipo di restituzione | Descrizione |
|--|----------------------|---|
| <code>s.insert(source, nodes, newIDs)</code> | Elenco | Inserisce copie dei nodi nell'elenco fornito. Si suppone che tutti i nodi nell'elenco fornito siano contenuti nel flusso specificato. L'indicatore <code>newIDs</code> indica se è necessario generare nuovi ID per ciascun nodo o se è necessario copiare ed utilizzare l'ID esistente. Si suppone che tutti i nodi in un flusso dispongano di un ID univoco, per cui questo indicatore deve essere impostato su <code>True</code> se il flusso di origine è uguale al flusso specificato. Il metodo restituisce l'elenco dei nodi appena inseriti, in cui l'ordine dei nodi è indefinito (l'ordinamento non è necessariamente uguale all'ordine dei nodi nell'elenco di input). |
| <code>s.delete(node)</code> | Non applicabile | Elimina il nodo specificato dal flusso specificato. Il nodo deve essere di proprietà del flusso specificato. |
| <code>s.deleteAll(nodes)</code> | Non applicabile | Elimina tutti i nodi specificati dal flusso specificato. Tutti i nodi nella raccolta devono appartenere al flusso specificato. |
| <code>s.clear()</code> | Non applicabile | Elimina tutti i nodi dal flusso specificato. |

Attraversamento dei nodi in un flusso

Un requisito comune è quello di identificare i nodi upstream o downstream di un particolare nodo. Il flusso fornisce diversi metodi che è possibile utilizzare per identificare tali nodi. Tali metodi sono riepilogati nella tabella riportata di seguito.

Tabella 15. Metodi per identificare i nodi upstream e downstream

| Metodo | Tipo di restituzione | Descrizione |
|---|----------------------|--|
| <code>s.iterator()</code> | Iteratore | Restituisce un iteratore sugli oggetti del nodo contenuti nel flusso specificato. Se il flusso viene modificato tra le chiamate della funzione <code>next()</code> , il funzionamento dell'iteratore non è definito. |
| <code>s.predecessorAt(node, index)</code> | Nodo | Restituisce il predecessore immediato specificato del nodo fornito oppure <code>None</code> se l'indice non è compreso nei limiti. |
| <code>s.predecessorCount(node)</code> | <i>int</i> | Restituisce il numero di predecessori immediati del nodo fornito. |
| <code>s.predecessors(node)</code> | Elenco | Restituisce i predecessori immediati del nodo fornito. |
| <code>s.successorAt(node, index)</code> | Nodo | Restituisce il successore immediato specificato del nodo fornito oppure <code>None</code> se l'indice non è compreso nei limiti. |

Tabella 15. Metodi per identificare i nodi upstream e downstream (Continua)

| Metodo | Tipo di restituzione | Descrizione |
|-------------------------------------|----------------------|---|
| <code>s.successorCount(node)</code> | <i>int</i> | Restituisce il numero di successori immediati del nodo fornito. |
| <code>s.successors(node)</code> | Elenco | Restituisce i successori immediati del nodo fornito. |

Cancellazione o rimozione di elementi

Gli script legacy supportano diversi utilizzi del comando `clear`, ad esempio:

- `clear outputs` Per eliminare tutti gli elementi dell'output dalla tavolozza dei manager.
- `clear generated palette` Per eliminare tutti i nugget del modello dalla tavolozza dei modelli.
- `clear stream` Per rimuovere il contenuto di un flusso.

Gli script Python supportano un insieme simile di funzioni; il comando `removeAll()` viene utilizzato per cancellare i manager Flussi, Output e Modelli. Ad esempio:

- Per cancellare il manager Flussi:

```
session = modeler.script.session()
session.getStreamManager.removeAll()
```
- Per cancellare il manager Output:

```
session = modeler.script.session()
session.getDocumentOutputManager().removeAll()
```
- Per cancellare il manager Modelli:

```
session = modeler.script.session()
session.getModelOutputManager().removeAll()
```

Acquisizione delle informazioni relative ai nodi

I nodi possono essere suddivisi in diverse categorie, come, ad esempio, nodi di importazione ed esportazione dei dati, nodi di creazione dei modelli ed altri tipi di nodi. Ciascun nodo fornisce una serie di metodi che è possibile utilizzare per individuare informazioni relative al nodo.

I metodi che è possibile utilizzare per ottenere l'ID, il nome e l'etichetta di un nodo sono riepilogati nella tabella riportata di seguito.

Tabella 16. Metodi per ottenere l'ID, il nome e l'etichetta di un nodo

| Metodo | Tipo di restituzione | Descrizione |
|---------------------------|----------------------|--|
| <code>n.getLabel()</code> | <i>stringa</i> | Restituisce l'etichetta di visualizzazione del nodo specificato. L'etichetta è il valore della proprietà <code>custom_name</code> solo se tale proprietà è una stringa non vuota e la proprietà <code>use_custom_name</code> non è impostata; in caso contrario, l'etichetta è il valore di <code>getName()</code> . |

Tabella 16. Metodi per ottenere l'ID, il nome e l'etichetta di un nodo (Continua)

| Metodo | Tipo di restituzione | Descrizione |
|-------------------|----------------------|--|
| n.setLabel(label) | Non applicabile | Imposta l'etichetta di visualizzazione del nodo specificato. Se la nuova etichetta è una stringa non vuota viene assegnata alla proprietà custom_name, e False viene assegnato alla proprietà use_custom_name in modo che l'etichetta specificata ha la precedenza; altrimenti una stringa vuota viene assegnata alla proprietà custom_name e True viene assegnato alla proprietà use_custom_name. |
| n.getName() | stringa | Restituisce il nome del nodo specificato. |
| n.getID() | stringa | Restituisce l'ID del nodo specificato. Viene creato un nuovo ID ogni volta che viene creato un nuovo nodo. L'ID viene reso permanente con il nodo quando viene salvato come parte di un flusso, in modo che quando il flusso viene aperto, gli ID del nodo vengono conservati. Tuttavia, se un nodo salvato viene inserito in un flusso, il nodo inserito viene considerato come un nuovo oggetto e ad esso verrà assegnato un nuovo ID. |

I metodi che è possibile utilizzare per ottenere altre informazioni relative ad un nodo sono riepilogati nella seguente tabella.

Tabella 17. Metodi per ottenere informazioni relative ad un nodo

| Metodo | Tipo di restituzione | Descrizione |
|-----------------------|----------------------|---|
| n.getTypeName() | stringa | Restituisce il nome di script di questo nodo. È lo stesso nome che può essere utilizzato per creare una nuova istanza di questo nodo. |
| n.isInitial() | Booleana | Restituisce True se si tratta di un nodo <i>iniziale</i> , vale a dire un nodo che si verifica all'inizio di un flusso. |
| n.isInline() | Booleana | Restituisce True se questo è un nodo <i>in linea</i> , vale a dire un nodo presente a metà del flusso. |
| n.isTerminal() | Booleana | Restituisce True se questo è un nodo <i>terminale</i> , vale a dire un nodo presente alla fine di un flusso. |
| n.getXPosition() | int | Restituisce l'offset della posizione x del nodo nel flusso. |
| n.getYPosition() | int | Restituisce l'offset della posizione y del nodo nel flusso. |
| n.setXYPosition(x, y) | Non applicabile | Imposta la posizione del nodo nel flusso. |

Tabella 17. Metodi per ottenere informazioni relative ad un nodo (Continua)

| Metodo | Tipo di restituzione | Descrizione |
|--------------------------------------|----------------------|--|
| n.setPositionBetween(source, target) | Non applicabile | Imposta la posizione del nodo nel flusso, in modo che sia posizionato tra i nodi forniti. |
| n.isCacheEnabled() | Booleana | Restituisce True se la cache è abilitata; in caso contrario, restituisce False. |
| n.setCacheEnabled(val) | Non applicabile | Abilita o disabilita la cache per questo oggetto. Se la cache è piena e la memorizzazione nella cache viene disabilitata, la cache viene svuotata. |
| n.isCacheFull() | Booleana | Restituisce True se la cache è piena; in caso contrario, restituisce False. |
| n.flushCache() | Non applicabile | Svuota la cache di questo nodo. Non ha alcun effetto se la cache non è abilitata o non è piena. |

Capitolo 4. API di script

Introduzione all'API di script

L'API di script fornisce l'accesso ad una vasta gamma di funzionalità di SPSS Modeler. Tutti i metodi descritti fanno parte dell'API ed è possibile eseguirvi l'accesso in modo implicito all'interno dello script senza ulteriori importazioni. Tuttavia, se si desidera fare riferimento alle classi API, è necessario importare l'API in modo esplicito con la seguente istruzione:

```
import modeler.api
```

Tale istruzione di importazione è richiesta da molti degli esempi di API di script.

Una guida completa relativa alle classi, ai metodi ed ai parametri disponibili nell'API di script è disponibile nel documento *IBM SPSS Modeler 17 Python Scripting API Reference Guide*.

Esempio: ricerca di nodi utilizzando un filtro personalizzato

La sezione "Ricerca di nodi" a pagina 29 conteneva un esempio di ricerca di un nodo in un flusso in cui veniva utilizzato il nome del tipo del nodo come criterio di ricerca. In alcune situazioni, è richiesta una ricerca più generica ed è possibile utilizzare la classe `NodeFilter` ed il metodo `findAll()` del flusso. Questo tipo di ricerca implica le due fasi riportate di seguito:

1. Creazione di una nuova classe che estende `NodeFilter` ed implementa una versione personalizzata del metodo `accept()`.
2. Richiamo del metodo `findAll()` del flusso con un'istanza di questa nuova classe. In questo modo vengono restituiti tutti i nodi che soddisfano i criteri definiti nel metodo `accept()`.

L'esempio riportato di seguito illustra come ricercare i nodi in un flusso con la cache del nodo abilitata. L'elenco dei nodi restituito può essere utilizzato per svuotare o disabilitare le cache di tali nodi.

```
import modeler.api

class CacheFilter(modeler.api.NodeFilter):
    """A node filter for nodes with caching enabled"""
    def accept(this, node):
        return node.isCacheEnabled()

cachingnodes = modeler.script.stream().findAll(CacheFilter(), False)
```

Metadati: informazioni sui dati

Poiché i nodi sono collegati insieme in un flusso, sono disponibili tutte le informazioni relative alle colonne o ai campi che sono disponibili per ogni singolo nodo. Per esempio, nella interfaccia del Modeler, questo permette di selezionare quali campi utilizzare per ordinare o aggregare. Queste informazioni sono chiamate modello dati.

Gli script possono accedere al modello dati esaminando i campi entrano o escono da un nodo. Per alcuni nodi, i modelli dei dati in input ed in output coincidono, per esempio un nodo `Ordina` semplicemente riordina i record ma non cambierà il modello dati. Alcuni, come il nodo `Ricava`, possono aggiungere nuovi campi. Altri, come il nodo `Filtro` possono rinominare o rimuovere i campi.

Nell'esempio seguente, lo script prende il flusso standard `IBM SPSS Modeler druglearn.str` e, per ogni campo, costruisce un modello con uno dei campi in input eliminato. Questo viene realizzato così:

1. Accesso al modello dati output dal nodo `Tipo`.
2. Esecuzione di cicli per ogni campo nel modello dati output.

3. Modificare il nodo Filtro per ogni campo di input.
4. Cambiare il nome del modello che si sta costruendo.
5. Eseguire il nodo di costruzione del modello.

Nota: Prima di eseguire lo script nel flusso `druglean.str`, ricordarsi di impostare il linguaggio dello script su Python (il flusso è stato creato in una versione precedente di IBM SPSS Modeler e quindi il linguaggio di script del flusso è impostato su Legacy).

```
import modeler.api

stream = modeler.script.stream()
filternode = stream.findByType("filter", None)
typenode = stream.findByType("type", None)
c50node = stream.findByType("c50", None)
# Always use a custom model name
c50node.setPropertyValue("use_model_name", True)

lastRemoved = None
fields = typenode.getOutputDataModel()
for field in fields:
    # If this is the target field then ignore it
    if field.getModelingRole() == modeler.api.ModelingRole.OUT:
        continue

    # Re-enable the field that was most recently removed
    if lastRemoved != None:
        filternode.setKeyedPropertyValue("include", lastRemoved, True)

    # Remove the field
    lastRemoved = field.getColumnName()
    filternode.setKeyedPropertyValue("include", lastRemoved, False)

    # Set the name of the new model then run the build
    c50node.setPropertyValue("model_name", "Exclude " + lastRemoved)
    c50node.run([])
```

L'oggetto `DataModel` fornisce diversi metodi per accedere alle informazioni dei campi o delle colonne all'interno di un certo modello dati. Tali metodi sono riepilogati nella tabella riportata di seguito.

Tabella 18. Metodi dell'oggetto `DataModel` per accedere alle informazioni relative ai campi o colonne

| Metodo | Tipo di restituzione | Descrizione |
|-------------------------------------|--------------------------|--|
| <code>d.getColumnCount()</code> | <i>int</i> | Restituisce il numero di colonne nel modello dati. |
| <code>d.columnIterator()</code> | Iteratore | Restituisce un iteratore che restituisce ogni colonna nell'ordine di inserimento "naturale". L'iteratore restituisce istanze di <code>Colonna</code> . |
| <code>d.nameIterator()</code> | Iteratore | Restituisce un iteratore che restituisce il nome di ogni colonna nell'ordine di inserimento "naturale". |
| <code>d.contains(name)</code> | <i>Booleana</i> | Restituisce <code>True</code> se esiste una colonna con il nome fornito in questo <code>DataModel</code> , <code>False</code> altrimenti. |
| <code>d.getColumn(name)</code> | Colonna | Restituisce la colonna con il nome specificato. |
| <code>d.getColumnGroup(name)</code> | <code>ColumnGroup</code> | Restituisce il gruppo di colonne specificato oppure <code>None</code> se il gruppo colonna non esiste. |

Tabella 18. Metodi dell'oggetto *DataModel* per accedere alle informazioni relative ai campi o colonne (Continua)

| Metodo | Tipo di restituzione | Descrizione |
|--------------------------------------|-----------------------|---|
| <code>d.getColumnGroupCount()</code> | <i>int</i> | Restituisce il numero di gruppi colonna in questo modello dati. |
| <code>d.columnGroupIterator()</code> | Iteratore | Restituisce un iteratore che restituisce ogni gruppo colonna. |
| <code>d.toArray()</code> | <code>Column[]</code> | Restituisce il modello dati come un array di colonne. Le colonne vengono ordinate secondo il loro ordine "naturale" di inserimento. |

Ogni campo (oggetto colonna) include un numero di metodi per accedere alle informazioni relative alla colonna. La tabella seguente mostra una selezione di questi.

Tabella 19. Metodi dell'oggetto *Colonna* per accedere alle informazioni relative ad una colonna

| Metodo | Tipo di restituzione | Descrizione |
|--------------------------------------|---------------------------|---|
| <code>c.columnName()</code> | <i>stringa</i> | Restituisce il nome della colonna. |
| <code>c.columnLabel()</code> | <i>stringa</i> | Restituisce l'etichetta della colonna oppure una stringa vuota se non c'è nessuna etichetta associata con la colonna. |
| <code>c.measureType()</code> | <code>MeasureType</code> | Restituisce il tipo misura per la colonna. |
| <code>c.storageType()</code> | <code>StorageType</code> | Restituisce il tipo archiviazione per la colonna. |
| <code>c.isMeasureDiscrete()</code> | <i>Booleana</i> | Restituisce True se la colonna ha un valore discreto. Colonne che sono un insieme o un indicatore sono considerate discrete. |
| <code>c.isModelOutputColumn()</code> | <i>Booleana</i> | Restituisce True se la colonna è una colonna di modello di output. |
| <code>c.isStorageDatetime()</code> | <i>Booleana</i> | Restituisce True se l'archiviazione della colonna è un valore di tipo ora, data o timestamp. |
| <code>c.isStorageNumeric()</code> | <i>Booleana</i> | Restituisce True se l'archiviazione della colonna è un intero o un numero reale. |
| <code>c.isValidValue(value)</code> | <i>Booleana</i> | Restituisce True se il valore specificato è valido per questa archiviazione e valid quando i valori validi della colonna sono noti. |
| <code>c.getModelingRole()</code> | <code>ModelingRole</code> | Restituisce il ruolo di modellazione per la colonna. |
| <code>c.getSetValues()</code> | <code>Object[]</code> | Restituisce un array di valori validi per la colonna o None se i valori non sono noti o la colonna non è un insieme. |
| <code>c.getValueLabel(value)</code> | <i>stringa</i> | Restituisce l'etichetta per il valore nella colonna, oppure una stringa vuota se non c'è alcuna etichetta associata con il valore. |

Tabella 19. Metodi dell'oggetto Colonna per accedere alle informazioni relative ad una colonna (Continua)

| Metodo | Tipo di restituzione | Descrizione |
|-------------------|----------------------|---|
| c.getFalseFlag() | Oggetto | Restituisce il valore di indicatore "false" per la colonna o None se il valore non è noto oppure se la colonna non è un indicatore. |
| c.getTrueFlag() | Oggetto | Restituisce il valore di indicatore "true" per la colonna o None se il valore non è noto oppure se la colonna non è un indicatore. |
| c.getLowerBound() | Oggetto | Restituisce il valore del limite inferiore per i valori nella colonna o None se il valore non è noto oppure se la colonna non è continua. |
| c.getUpperBound() | Oggetto | Restituisce il valore del limite superiore per i valori nella colonna o None se il valore non è noto oppure se la colonna non è continua. |

Si noti che la maggior parte dei metodi che accedono alle informazioni relative ad una colonna hanno metodi equivalenti definiti sull'oggetto DataModel stesso. Per esempio le istruzioni seguenti sono equivalenti:

```
dataModel.getColumn("someName").getModelingRole()
dataModel.getModelingRole("someName")
```

Accesso agli oggetti generati

Generalmente, l'esecuzione di un flusso implica la creazione di ulteriori oggetti di output. Tali oggetti aggiuntivi possono essere un nuovo modello oppure una parte di output che fornisce informazioni da utilizzare nelle esecuzioni successive.

Nell'esempio riportato di seguito, il flusso druglearn.str viene utilizzato nuovamente come punto di partenza per il flusso. In questo esempio, tutti i nodi nel flusso vengono eseguiti ed i risultati vengono archiviati in un elenco. Lo script, quindi, esegue un loop all'interno dei risultati e tutti gli output del modello risultanti dall'esecuzione vengono salvati come un file di modello IBM SPSS Modeler (.gm) ed il modello viene esportato in PMML.

```
import modeler.api

stream = modeler.script.stream()

# Set this to an existing folder on your system.
# Include a trailing directory separator
modelFolder = "C:/temp/models/"

# Execute the stream
models = []
stream.runAll(models)

# Save any models that were created
taskrunner = modeler.script.session().getTaskRunner()
for model in models:
    # If the stream execution built other outputs then ignore them
    if not(isinstance(model, modeler.api.ModelOutput)):
        continue

    label = model.getLabel()
    algorithm = model.getModelDetail().getAlgorithmName()
```

```

# save each model...
modelFile = modelFolder + label + algorithm + ".gm"
taskrunner.saveModelToFile(model, modelFile)

# ...and export each model PMML...
modelFile = modelFolder + label + algorithm + ".xml"
taskrunner.exportModelToFile(model, modelFile, modeler.api.FileFormat.XML)

```

La classe task runner fornisce un modo per l'esecuzione di varie attività comuni. I metodi disponibili in tale classe sono riepilogati nella tabella riportata di seguito.

Tabella 20. Metodi della classe task runner per l'esecuzione di attività comuni

| Metodo | Tipo di restituzione | Descrizione |
|--|----------------------|--|
| t.createStream(name, autoConnect, autoManage) | Flusso | Crea e restituisce un nuovo flusso. Notare che il codice che deve creare i flussi privatamente senza renderli visibili all'utente deve impostare l'indicatore autoManage su False. |
| t.exportDocumentToFile(documentOutput, filename, fileFormat) | Non applicabile | Esporta la descrizione del flusso in un file utilizzando il formato di file specificato. |
| t.exportModelToFile(modelOutput, filename, fileFormat) | Non applicabile | Esporta il modello in un file utilizzando il formato di file specificato. |
| t.exportStreamToFile(stream, filename, fileFormat) | Non applicabile | Esporta il flusso in un file utilizzando il formato file specificato. |
| t.insertNodeFromFile(filename, diagram) | Nodo | Legge e restituisce un nodo dal file specificato, inserendolo nel diagramma fornito. Notare che può essere utilizzato per leggere gli oggetti nodo e supernodo. |
| t.openDocumentFromFile(filename, autoManage) | DocumentOutput | Legge e restituisce un documento dal file specificato. |
| t.openModelFromFile(filename, autoManage) | ModelOutput | Legge e restituisce un modello dal file specificato. |
| t.openStreamFromFile(filename, autoManage) | Flusso | Legge e restituisce un flusso dal file specificato. |
| t.saveDocumentToFile(documentOutput, filename) | Non applicabile | Salva il documento nel percorso di file specificato. |
| t.saveModelToFile(modelOutput, filename) | Non applicabile | Salva il modello nel percorso di file specificato. |
| t.saveStreamToFile(stream, filename) | Non applicabile | Salva il flusso nel percorso di file specificato. |

Gestione degli errori

Il linguaggio Python fornisce la gestione degli errori mediante il blocco di codice try...except. Tale blocco può essere utilizzato all'interno degli script per racchiudere eccezioni e gestire problemi che, in caso contrario, potrebbero causare la fine dello script.

Nello script di esempio riportato di seguito, viene eseguito un tentativo di richiamo di un modello da un IBM SPSS Collaboration and Deployment Services Repository. Questa operazione può causare la generazione di un'eccezione, ad esempio, le credenziali di accesso al repository potrebbero non essere

state impostate correttamente oppure il percorso del repository non è corretto. Nello script, ciò può causare un'eccezione `ModelerException` (tutte le eccezioni generate da IBM SPSS Modeler sono derivate da `modeler.api.ModelerException`).

```
import modeler.api

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # print goes to the Modeler UI script panel Debug tab
    print "Everything OK"
except modeler.api.ModelerException, e:
    print "An error occurred:", e.getMessage()
```

Nota: Alcune operazioni di script potrebbero causare la generazione di eccezioni Java standard; tali eccezioni non derivano da `ModelerException`. Per rilevare tali eccezioni, è possibile utilizzare un ulteriore blocco per rilevare tutte le eccezioni Java, ad esempio:

```
import modeler.api

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # print goes to the Modeler UI script panel Debug tab
    print "Everything OK"
except modeler.api.ModelerException, e:
    print "An error occurred:", e.getMessage()
except java.lang.Exception, e:
    print "A Java exception occurred:", e.getMessage()
```

Parametri stream, sessione e Supernodo

I parametri rappresentano un utile modo per il passaggio dei valori al runtime, invece della codifica diretta in uno script. I parametri ed i relativi valori vengono definiti nello stesso modo dei flussi, vale a dire come voci nella tabella dei parametri di un flusso o supernodo oppure come parametri della riga comandi. Le classi `Stream` e `SuperNode` implementano una serie di funzioni definite dall'oggetto `ParameterProvider`, come illustrato nella tabella riportata di seguito. La sessione fornisce una chiamata `getParameters()` che restituisce un oggetto che definisce tali funzioni.

Tabella 21. Funzioni definite dall'oggetto `ParameterProvider`

| Metodo | Tipo di restituzione | Descrizione |
|--|----------------------------------|--|
| <code>p.parameterIterator()</code> | Iteratore | Restituisce un iteratore di nomi di parametri per questo oggetto. |
| <code>p.getParameterDefinition(parameterName)</code> | <code>ParameterDefinition</code> | Restituisce la definizione del parametro per il parametro con il nome specificato oppure <code>None</code> se in questo provider non esiste alcun parametro di questo tipo. Il risultato può essere un'istantanea della definizione nel momento in cui il metodo è stato richiamato e non deve riflettere tutte le modifiche successive apportate al parametro mediante questo provider. |
| <code>p.getParameterLabel(parameterName)</code> | <i>stringa</i> | Restituisce l'etichetta del parametro indicato oppure <code>None</code> se non esiste alcun parametro di questo tipo. |

Tabella 21. Funzioni definite dall'oggetto *ParameterProvider* (Continua)

| Metodo | Tipo di restituzione | Descrizione |
|---|----------------------|--|
| p.setParameterLabel(parameterName, label) | Non applicabile | Imposta l'etichetta del parametro indicato. |
| p.getParameterStorage(parameterName) | ParameterStorage | Restituisce l'archiviazione del parametro indicato oppure None se non esiste alcun parametro di questo tipo. |
| p.setParameterStorage(parameterName, storage) | Non applicabile | Imposta l'archiviazione del parametro indicato. |
| p.getParameterType(parameterName) | Tipo Parametro | Restituisce il tipo del parametro indicato oppure None se non esiste alcun parametro di questo tipo. |
| p.setParameterType(parameterName, type) | Non applicabile | Imposta il tipo del parametro indicato. |
| p.getParameterValue(parameterName) | Oggetto | Restituisce il valore del parametro indicato oppure None se non esiste alcun parametro di questo tipo. |
| p.setParameterValue(parameterName, value) | Non applicabile | Imposta il valore del parametro indicato. |

Nell'esempio riportato di seguito, lo script aggrega alcuni dati Telco per individuare la regione con i dati di reddito medio più basso. Con questa regione viene quindi impostato un parametro stream. Tale parametro stream viene quindi utilizzato in un nodo Select per escludere tale regione dai dati, prima che sulla parte rimanente venga creato un modello churn.

L'esempio è artificiale perché lo script genera il nodo Select da solo e, pertanto, potrebbe aver generato il valore corretto direttamente nell'espressione del nodo Select. Tuttavia, i flussi sono generalmente pregenerati, per cui l'impostazione dei parametri in questo modo rappresenta un esempio utile.

La prima parte dello script di esempio crea il parametro stream che conterrà la regione con il reddito medio più basso. Lo script crea anche i nodi nel ramo di aggregazione e nel ramo di creazione del modello e li collega tra loro.

```
import modeler.api

stream = modeler.script.stream()

# Initialize a stream parameter
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# First create the aggregation branch to compute the average income per region
statisticsimportnode = stream.createAt("statisticsimport", "SPSS File", 114, 142)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/telco.sav")
statisticsimportnode.setPropertyValue("use_field_format_for_storage", True)

aggregatenode = modeler.script.stream().createAt("aggregate", "Aggregate", 294, 142)
aggregatenode.setPropertyValue("keys", ["region"])
aggregatenode.setKeyedPropertyValue("aggregates", "income", ["Mean"])

tablenode = modeler.script.stream().createAt("table", "Table", 462, 142)

stream.link(statisticsimportnode, aggregatenode)
stream.link(aggregatenode, tablenode)

selectnode = stream.createAt("select", "Select", 210, 232)
selectnode.setPropertyValue("mode", "Discard")
# Reference the stream parameter in the selection
```

```

selectnode.setPropertyValue("condition", "'region' = '$P-LowestRegion'")

typenode = stream.createAt("type", "Type", 366, 232)
typenode.setKeyedPropertyValue("direction", "churn", "Target")

c50node = stream.createAt("c50", "C5.0", 534, 232)

stream.link(statisticsimportnode, selectnode)
stream.link(selectnode, typenode)
stream.link(typenode, c50node)

```

Lo script di esempio crea il seguente flusso.

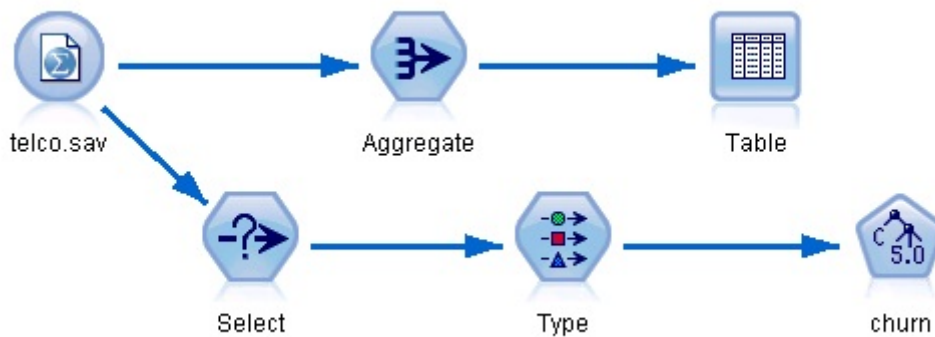


Figura 5. Flusso che risulta dallo script di esempio

La seguente parte dello script di esempio esegue il nodo Table alla fine del ramo di aggregazione.

```

# First execute the table node
results = []
tablenode.run(results)

```

La seguente parte dello script di esempio accede all'output della tabella generato dall'esecuzione del nodo Table. Lo script, quindi, esegue un'iterazione attraverso le righe nella tabella, ricercando la regione con il reddito medio più basso.

```

# Running the table node should produce a single table as output
table = results[0]

# table output contains a RowSet so we can access values as rows and columns
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# From the way the aggregate node is defined, the first column
# contains the region and the second contains the average income
row = 0
rowcount = rowset.getRowCount()
while row < rowcount:
    if rowset.getValueAt(row, 1) < min_income:
        min_income = rowset.getValueAt(row, 1)
        min_region = rowset.getValueAt(row, 0)
    row += 1

```

La seguente parte dello script utilizza la regione con il reddito medio più basso per impostare il parametro del flusso "LowestRegion" precedentemente creato. Lo script, quindi, esegue il builder del modello con la regione specificata esclusa dai dati di addestramento.

```

# Check that a value was assigned
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)

```

```

else:
    stream.setParameterValue("LowestRegion", -1)

# Finally run the model builder with the selection criteria
c50node.run([])

Lo script di esempio completo è riportato di seguito.
import modeler.api

stream = modeler.script.stream()

# Create a stream parameter
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# First create the aggregation branch to compute the average income per region
statisticsimportnode = stream.createAt("statisticsimport", "SPSS File", 114, 142)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/telco.sav")
statisticsimportnode.setPropertyValue("use_field_format_for_storage", True)

aggregatenode = modeler.script.stream().createAt("aggregate", "Aggregate", 294, 142)
aggregatenode.setPropertyValue("keys", ["region"])
aggregatenode.setKeyedPropertyValue("aggregates", "income", ["Mean"])

tablenode = modeler.script.stream().createAt("table", "Table", 462, 142)

stream.link(statisticsimportnode, aggregatenode)
stream.link(aggregatenode, tablenode)

selectnode = stream.createAt("select", "Select", 210, 232)
selectnode.setPropertyValue("mode", "Discard")
# Reference the stream parameter in the selection
selectnode.setPropertyValue("condition", "'region' = '$P-LowestRegion'")

typenode = stream.createAt("type", "Type", 366, 232)
typenode.setKeyedPropertyValue("direction", "churn", "Target")

c50node = stream.createAt("c50", "C5.0", 534, 232)

stream.link(statisticsimportnode, selectnode)
stream.link(selectnode, typenode)
stream.link(typenode, c50node)

# First execute the table node
results = []
tablenode.run(results)

# Running the table node should produce a single table as output
table = results[0]

# table output contains a RowSet so we can access values as rows and columns
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# From the way the aggregate node is defined, the first column
# contains the region and the second contains the average income
row = 0
rowcount = rowset.getRowCount()
while row < rowcount:
    if rowset.getValueAt(row, 1) < min_income:
        min_income = rowset.getValueAt(row, 1)
        min_region = rowset.getValueAt(row, 0)
    row += 1

# Check that a value was assigned

```

```

if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

# Finally run the model builder with the selection criteria
c50node.run([])

```

Valori globali

I valori globali vengono utilizzati per calcolare statistiche di riepilogo per campi specificati. È possibile accedere a tali valori di riepilogo in qualsiasi punto all'interno del flusso. I valori globali sono simili ai parametri del flusso perché ad essi si accede in base al nome attraverso il flusso. Sono differenti dai parametri del flusso perché i valori associati vengono aggiornati automaticamente quando viene eseguito un nodo Calcola globali, invece di essere assegnati mediante script o dalla riga comandi. È possibile accedere ai valori globali per un flusso richiamando il metodo `getGlobalValues()` del flusso.

L'oggetto `GlobalValues` definisce le funzioni indicate nella tabella riportata di seguito.

Tabella 22. Funzioni definite dall'oggetto `GlobalValues`

| Metodo | Tipo di restituzione | Descrizione |
|--|----------------------|--|
| <code>g.fieldNameIterator()</code> | Iteratore | Restituisce un iteratore per ciascun nome campo con almeno un valore globale. |
| <code>g.getValue(type, fieldName)</code> | Oggetto | Restituisce il valore globale per il nome del campo ed il tipo specificati oppure <code>None</code> se non è possibile individuare alcun valore. Generalmente, il valore restituito previsto è un numero, sebbene funzionalità future potrebbero restituire tipi di valori differenti. |
| <code>g.getValues(fieldName)</code> | Mappa | Restituisce una mappa che contiene le voci note per il nome del campo specificato oppure <code>None</code> se non esistono voci per il campo. |

`GlobalValues.Type` definisce il tipo di statistiche di riepilogo disponibili. Sono disponibili le statistiche di riepilogo riportate di seguito:

- MAX: il valore massimo del campo.
- MEAN: il valore medio del campo.
- MIN: il valore minimo del campo.
- STDDEV: la deviazione standard del campo.
- SUM: la somma dei valori nel campo.

Ad esempio, lo script riportato di seguito accede al valore medio del campo "income", calcolato da un nodo Calcola globali:

```

import modeler.api

globals = modeler.script.stream().getGlobalValues()
mean_income = globals.getValue(modeler.api.GlobalValues.Type.MEAN, "income")

```

Utilizzo di più flussi: script autonomi

Per utilizzare più flussi, è necessario utilizzare uno script autonomo. Lo script autonomo può essere modificato ed eseguito all'interno dell'interfaccia utente di IBM SPSS Modeler oppure passato come parametro della riga comandi in modalità batch.

Lo script autonomo riportato di seguito apre due flussi. Uno di tali flussi crea un modello, mentre l'altro flusso traccia la distribuzione dei valori previsti.

```
# Change to the appropriate location for your system
demosDir = "C:/Program Files/IBM/SPSS/Modeler/17/DEMOS/streams/"

session = modeler.script.session()
tasks = session.getTaskRunner()

# Open the model build stream, locate the C5.0 node and run it
buildstream = tasks.openStreamFromFile(demosDir + "druglearn.str", True)
c50node = buildstream.findByType("c50", None)
results = []
c50node.run(results)

# Now open the plot stream, find the Na_to_K derive and the histogram
plotstream = tasks.openStreamFromFile(demosDir + "drugplot.str", True)
derivenode = plotstream.findByType("derive", None)
histogramnode = plotstream.findByType("histogram", None)

# Create a model applier node, insert it between the derive and histogram nodes
# then run the histogram
applyc50 = plotstream.createModelApplier(results[0], results[0].getName())
applyc50.setPositionBetween(derivenode, histogramnode)
plotstream.linkBetween(applyc50, derivenode, histogramnode)
histogramnode.setPropertyValue("color_field", "$C-Drug")
histogramnode.run([])

# Finally, tidy up the streams
buildstream.close()
plotstream.close()
```

Capitolo 5. Suggerimenti per gli script

Questa sezione contiene una panoramica dei suggerimenti e delle tecniche di utilizzo degli script, incluse tecniche relative a modifica dell'esecuzione del flusso, utilizzo di una password codificata in uno script e accesso a oggetti in IBM SPSS Collaboration and Deployment Services Repository.

Modifica dell'esecuzione del flusso

Durante l'esecuzione di un flusso, i relativi nodi terminali vengono eseguiti in base all'ordine ottimale per la situazione di default. In alcuni casi, è tuttavia preferibile utilizzare un ordine di esecuzione diverso. Per modificare l'ordine di esecuzione di un flusso, completare i passi riportati di seguito dalla scheda Esecuzione della finestra di dialogo delle proprietà del flusso:

1. Iniziare con uno script vuoto.
2. Fare clic sul pulsante **Accoda lo script di default** nella barra degli strumenti per aggiungere lo script del flusso di default.
3. Modificare l'ordine delle istruzioni contenute nello script del flusso di default in base all'ordine in cui si desidera che vengano eseguite le istruzioni.

Esecuzione di cicli sui nodi

È possibile eseguire un ciclo for per eseguire un ciclo su tutti i nodi in un flusso. Ad esempio, i due esempi di script riportati di seguito eseguono un ciclo su tutti i nodi e cambiano da minuscolo a maiuscolo i nomi dei campi nei nodi Filtro.

Tali script possono essere utilizzati in qualsiasi flusso che dispone di un nodo Filtro, anche se non viene effettivamente filtrato alcun campo. È sufficiente aggiungere un nodo Filtro che passi tutti i campi per far sì che tutti i nomi dei campi diventino maiuscoli.

```
# Alternative 1: using the data model nameIterator() function
stream = modeler.script.stream()
for node in stream.iterator():
    if (node.getTypeName() == "filter"):
        # nameIterator() returns the field names
        for field in node.getInputDataModel().nameIterator():
            newname = field.upper()
            node.setKeyedPropertyValue("new_name", field, newname)

# Alternative 2: using the data model iterator() function
stream = modeler.script.stream()
for node in stream.iterator():
    if (node.getTypeName() == "filter"):
        # iterator() returns the field objects so we need
        # to call getColumnName() to get the name
        for field in node.getInputDataModel().iterator():
            newname = field.getColumnName().upper()
            node.setKeyedPropertyValue("new_name", field.getColumnName(), newname)
```

Lo script esegue un ciclo su tutti i nodi nel flusso corrente e verifica se ciascun nodo è un nodo Filtro. In questo caso, lo script esegue un ciclo su ciascun campo nel nodo ed utilizza la funzione `field.upper()` o `field.getColumnName().upper()` per modificare il nome in maiuscolo.

Accesso a oggetti nel IBM SPSS Collaboration and Deployment Services Repository

Se si dispone di una licenza per IBM SPSS Collaboration and Deployment Services Repository, è possibile archiviare, recuperare, bloccare e sbloccare oggetti dal repository tramite comandi script. Il repository consente di gestire il ciclo di vita dei modelli di data mining e gli oggetti predittivi correlati nel contesto delle applicazioni, degli strumenti e delle soluzioni aziendali esistenti.

Connessione a IBM SPSS Collaboration and Deployment Services Repository

Per accedere al repository, è innanzitutto necessario impostare una connessione valida al repository stesso, tramite il menu Strumenti dell'interfaccia utente di IBM SPSS Modeler o mediante la riga di comando. (Per ulteriori informazioni, consultare l'argomento "Argomenti per la connessione a IBM SPSS Collaboration and Deployment Services Repository" a pagina 65.)

Archiviazione e recupero di oggetti

All'interno di uno script, i comandi retrieve e store consentono di accedere a vari oggetti, inclusi stream, modelli, output, nodi e progetti. La sintassi è la seguente:

```
store object as REPOSITORY_PATH {label LABEL}
store object as URI [#1.label]

retrieve object REPOSITORY_PATH {label LABEL | version VERSION}
retrieve object URI [(#m.marker | #1.label)]
```

REPOSITORY_PATH fornisce la posizione dell'oggetto nel repository. Il percorso deve essere racchiuso tra virgolette ed è necessario utilizzare le barre (/) come delimitatori. Il percorso non fa distinzione tra caratteri maiuscoli/minuscoli.

```
store stream as "/folder_1/folder_2/mystream.str"
store model Drug as "/myfolder/drugmodel"
store model Drug as "/myfolder/drugmodel.gm" label "final"
store node DRUGIn as "/samples/drugIntypenode"
store project as "/CRISPDM/DrugExample.cpj"
store output "Data Audit of [6 fields]" as "/my folder/My Audit"
```

È possibile aggiungere un'estensione al nome dell'oggetto, quale *.str* o *.gm*; questa operazione non è obbligatoria a condizione che il nome sia coerente. Per esempio, se un modello è stato archiviato senza estensione, deve essere recuperato con lo stesso nome.

```
store model "/myfolder/drugmodel"
retrieve model "/myfolder/drugmodel"
```

non:

```
store model "/myfolder/drugmodel.gm"
retrieve model "/myfolder/drugmodel.gm" version "0:2005-10-12 14:15:41.281"
```

Si noti che quando si recuperano oggetti, se non si specifica una versione o un'etichetta, viene sempre restituita la versione più recente dell'oggetto. Quando si recupera un oggetto nodo, il nodo viene automaticamente inserito nello stream corrente. Quando si recupera un oggetto stream, è necessario utilizzare uno script autonomo. Non è possibile recuperare un oggetto stream da uno script dello stream.

Blocco e sblocco degli oggetti

Da uno script, è possibile bloccare un oggetto per impedire qualsiasi aggiornamento delle sue versioni esistenti o la creazione di nuove versioni. È inoltre possibile sbloccare un oggetto che è stato bloccato.

La sintassi per bloccare e sbloccare un oggetto è:


```
lock REPOSITORY_PATH
lock URI
```

```
unlock REPOSITORY_PATH
unlock URI
```

Come con l'archiviazione e il recupero di oggetti, REPOSITORY_PATH fornisce la posizione dell'oggetto nel repository. Il percorso deve essere racchiuso tra virgolette ed è necessario utilizzare le barre (/) come delimitatori. Il percorso non fa distinzione tra caratteri maiuscoli/minuscoli.

```
lock "/myfolder/Stream1.str"
```

```
unlock "/myfolder/Stream1.str"
```

In alternativa, è possibile utilizzare un URI (Uniform Resource Identifier) anziché un percorso di repository per indicare la posizione dell'oggetto. L'URI deve includere il prefisso `spsscr:` e deve essere racchiuso tra virgolette. Come delimitatori di percorso sono consentite solo le barre (/) e gli spazi devono essere codificati, ovvero utilizzare `%20` anziché lo spazio nel percorso. L'URI non fa distinzione tra caratteri maiuscoli/minuscoli. Di seguito sono riportati alcuni esempi:

```
lock "spsscr:///myfolder/Stream1.str"
```

```
unlock "spsscr:///myfolder/Stream1.str"
```

Tenere presente che il blocco dell'oggetto si applica a tutte le versioni di un oggetto. Non è possibile, infatti, bloccare o sbloccare versioni singole.

Generazione di una password codificata

In alcuni casi, è possibile includere una password in uno script, per esempio se si desidera accedere a un'origine dati protetta da password. Le password codificate possono essere utilizzate in:

- Proprietà dei nodi per nodi origine Database e nodi output
- Argomenti della riga di comando per l'accesso al server
- Proprietà di connessione al database archiviate in un file *.par* (file del parametro generato dalla scheda Pubblica di un nodo di esportazione)

Tramite l'interfaccia utente, è disponibile uno strumento per generare password codificate in base all'algoritmo Blowfish (per ulteriori informazioni, vedere <http://www.schneier.com/blowfish.html>). Dopo la codifica, è possibile copiare e archiviare la password in file script e argomenti della riga di comando. La proprietà del nodo `epassword` utilizzata per i nodi `datanode` e `databaseexportnode` consente di memorizzare la password codificata.

1. Per generare una password codificata, dal menu Strumenti scegliere:

Codifica password...

2. Specificare una password nella casella di testo Password.
3. Fare clic su **Codifica** per generare una codifica casuale per la password.
4. Fare clic sul pulsante Copia per copiare la password codificata negli Appunti.
5. Incollare la password nello script o nel parametro desiderato.

Controllo degli script

Per controllare in modo rapido la sintassi di tutti i tipi di script, fare clic sul pulsante di verifica di colore rosso disponibile nella barra degli strumenti della finestra di dialogo Script autonomo.



Figura 6. Icone della barra degli strumenti Script del flusso

Durante il controllo degli script verranno segnalati gli eventuali errori del codice e forniti suggerimenti per la risoluzione. Per visualizzare la riga contenente gli errori, fare clic sul feedback visualizzato nella metà inferiore della finestra di dialogo. L'errore verrà evidenziato in rosso.

Script dalla riga di comando

Gli script consentono di eseguire operazioni che in genere vengono eseguite nell'interfaccia utente. È sufficiente specificare ed eseguire gli script locali dalla riga di comando quando si avvia IBM SPSS Modeler. Ad esempio:

```
client -script scores.txt -execute
```

Il flag `-script` carica lo script specificato, mentre il flag `-execute` esegue tutti i comandi contenuti nel file di script.

Compatibilità con le versioni precedenti

Questa versione di IBM SPSS Modeler supporta il funzionamento degli script creati nelle versioni precedenti. Tuttavia, ora è possibile inserire automaticamente i nugget del modello nel flusso (questa è l'impostazione di default) nonché sostituire o integrare un nugget esistente dello stesso tipo all'interno del flusso. Il fatto che questo accada o meno dipende dalle impostazioni delle opzioni **Aggiungi modello al flusso** e **Sostituisci modello precedente (Strumenti > Opzioni > Opzioni utente > Notifiche)**. Per esempio, può essere necessario modificare uno script di una versione precedente in cui la sostituzione di un nugget avviene eliminando il nugget esistenti e inserendone uno nuovo.

Gli script creati in questa versione potrebbero non funzionare in versioni precedenti.

Se uno script creato in una versione precedente utilizza un comando che è stato nel frattempo sostituito o che non è più supportato, il formato sarà supportato, ma verrà visualizzato un messaggio di avviso. Per esempio, la vecchia parola chiave `generated` è stata sostituita da `model` e `clear generated` è stato sostituito da `clear generated palette`. Gli script che utilizzano i vecchi formati verranno eseguiti, tuttavia verrà visualizzato un messaggio di avviso.

Accesso ai risultati dell'esecuzione del flusso

Molti nodi IBM SPSS Modeler producono oggetti di output come modelli, grafici e dati in formato tabellare. Molti di tali output contengono valori utili che possono essere utilizzati dagli script per guidare la successiva esecuzione. Tali valori vengono raggruppati in contenitori del contenuto (indicati semplicemente come contenitori) a cui è possibile accedere utilizzando tag o ID che identificano ciascun contenitore. La modalità di accesso a tali valori dipende dal formato o "modello di contenuto" utilizzato da tale contenitore.

Ad esempio, molti output del modello predittivo utilizzano una variante di XML denominata PMML per rappresentare le informazioni relative al modello, come, ad esempio, i campi utilizzati da una struttura ad albero delle decisioni in ciascuna suddivisione oppure il modo in cui i neuroni sono connessi in una rete neurale e con quale intensità. Gli output del modello che utilizzano PMML forniscono un modello di contenuto XML che può essere utilizzato per accedere a tali informazioni. Ad esempio:

```
stream = modeler.script.stream()
# Assume the stream contains a single C5.0 model builder node
# and that the datasource, predictors and targets have already been
# set up
modelbuilder = stream.findByType("c50", None)
```

```

results = []
modelbuilder.run(results)
modeloutput = results[0]

# Now that we have the C5.0 model output object, access the
# relevant content model
cm = modeloutput.getContentModel("PMML")

# The PMML content model is a generic XML-based content model that
# uses XPath syntax. Use that to find the names of the data fields.
# The call returns a list of strings match the XPath values
dataFieldNames = cm.getStringValues("/PMML/DataDictionary/DataField", "name")

```

IBM SPSS Modeler supporta i seguenti modelli di contenuto negli script:

- Il **modello di contenuto tabella** fornisce l'accesso a dati in formato tabellare semplici rappresentati come righe e colonne
- Il **modello di contenuto XML** fornisce l'accesso al contenuto archiviato in formato XML
- Il **modello di contenuto JSON** fornisce l'accesso al contenuto archiviato in formato JSON
- Il **modello di contenuto delle statistiche di colonne** fornisce l'accesso a statistiche di riepilogo relative ad un campo specifico
- Il **modello di contenuto delle statistiche di colonne pair-wise** fornisce l'accesso a statistiche di riepilogo tra due campi o valori tra due campi separati

Modello di contenuto tabella

Il modello di contenuto tabella fornisce un modello semplice per l'accesso ai dati di righe e colonne semplici. I valori in una particolare colonna devono avere tutti lo stesso tipo di archiviazione (ad esempio, stringhe o interi).

API

Tabella 23. API

| Restituzione | Metodo | Descrizione |
|--------------|---|---|
| int | getRowCount() | Restituisce il numero di righe in questa tabella. |
| int | getColumnCount() | Restituisce il numero di colonne in questa tabella. |
| String | getColumnName(int columnIndex) | Restituisce il nome della colonna all'indice della colonna specificato. L'indice della colonna parte da 0. |
| StorageType | getStorageType(int columnIndex) | Restituisce il tipo di archiviazione della colonna all'indice specificato. L'indice della colonna parte da 0. |
| Oggetto | getValueAt(int rowIndex, int columnIndex) | Restituisce il valore all'indice della riga e della colonna specificato. Gli indici di riga e colonna partono da 0. |
| void | reset() | Svuota la memoria interna associata a questo modello di contenuto. |

Nodi ed output

Questa tabella elenca i nodi che creano output che includono questo tipo di modello di contenuto.

Tabella 24. Nodi ed output

| Nome nodo | Nome output | ID contenitore |
|-----------|-------------|----------------|
| table | table | "table" |

Script di esempio

```
stream = modeler.script.stream()
from modeler.api import StorageType

# Set up the variable file import node
varfilenode = stream.createAt("variablefile", "DRUG Data", 96, 96)
varfilenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")

# Next create the aggregate node and connect it to the variable file node
aggregatenode = stream.createAt("aggregate", "Aggregate", 192, 96)
stream.link(varfilenode, aggregatenode)

# Configure the aggregate node
aggregatenode.setPropertyValue("keys", ["Drug"])
aggregatenode.setKeyedPropertyValue("aggregates", "Age", ["Min", "Max"])
aggregatenode.setKeyedPropertyValue("aggregates", "Na", ["Mean", "SDev"])

# Then create the table output node and connect it to the aggregate node
tablenode = stream.createAt("table", "Table", 288, 96)
stream.link(aggregatenode, tablenode)

# Execute the table node and capture the resulting table output object
results = []
tablenode.run(results)
tableoutput = results[0]

# Access the table output's content model
tablecontent = tableoutput.getContentModel("table")

# For each column, print column name, type and the first row
# of values from the table content
col = 0
while col < tablecontent.getColumnCount():
    print tablecontent.getColumnName(col), \
          tablecontent.getStorageType(col), \
          tablecontent.getValueAt(0, col)
    col = col + 1
```

L'output nella scheda Debug dello script sarà simile a quello riportato di seguito:

```
Age_Min Integer 15
Age_Max Integer 74
Na_Mean Real 0.730851098901
Na_SDev Real 0.116669731242
Drug String drugY
Record_Count Integer 91
```

Modello di contenuto XML

Il modello di contenuto XML fornisce l'accesso al contenuto basato su XML.

Il modello di contenuto XML supporta la possibilità di accedere ai componenti in base alle espressioni XPath. Le espressioni XPath sono stringhe che definiscono gli elementi o gli attributi richiesti dal chiamante. Il modello di contenuto XML nasconde i dettagli relativi alla costruzione di diversi oggetti ed alla compilazione di espressioni generalmente richiesti dal supporto XPath. In questo modo, è più semplice eseguire chiamate dagli script Python.

Il modello di contenuto XML include una funzione che restituisce il documento XML come stringa. In questo modo, gli utenti di script Python possono utilizzare la propria libreria Python preferita per analizzare il codice XML.

API

Tabella 25. API

| Restituzione | Metodo | Descrizione |
|---|--|--|
| String | <code>getXMLAsString()</code> | Restituisce il codice XML come stringa. |
| numero | <code>getNumericValue(String xpath)</code> | Restituisce il risultato della valutazione del percorso con un tipo di restituzione numerico (ad esempio, conteggia il numero di elementi che corrispondono all'espressione del percorso). |
| booleano | <code>getBooleanValue(String xpath)</code> | Restituisce il risultato booleano della valutazione dell'espressione del percorso specificata. |
| String | <code>getStringValue(String xpath, String attribute)</code> | Restituisce il valore dell'attributo o il valore del nodo XML che corrisponde al percorso specificato. |
| Elenco di stringhe | <code>getStringValues(String xpath, String attribute)</code> | Restituisce un elenco di tutti i valori di attributo o valori del nodo XML che corrispondono al percorso specificato. |
| Elenco di elenchi di stringhe | <code>getValuesList(String xpath, <List of strings> attributes, boolean includeValue)</code> | Restituisce un elenco di tutti i valori di attributo che corrispondono al percorso specificato insieme al valore del nodo XML se richiesto. |
| Tabella hash (key:string, value:list of string) | <code>getValuesMap(String xpath, String keyAttribute, <List of strings> attributes, boolean includeValue)</code> | Restituisce una tabella hash che utilizza l'attributo chiave o il valore del nodo XML come chiave e l'elenco dei valori dell'attributo specificato come valori della tabella. |
| booleano | <code>isNamespaceAware()</code> | Indica se i parser XML devono riconoscere gli spazi dei nomi. Il valore di default è <code>False</code> . |
| void | <code>setNamespaceAware(boolean value)</code> | Imposta se i parser XML devono riconoscere gli spazi dei nomi. Inoltre, richiama <code>reset()</code> per garantire che le modifiche siano utilizzate dalle chiamate successive. |
| void | <code>reset()</code> | Svuota la memoria interna associata a questo modello di contenuto (ad esempio, un oggetto DOM memorizzato nella cache). |

Nodi ed output

Questa tabella elenca i nodi che creano output che includono questo tipo di modello di contenuto.

Tabella 26. Nodi ed output

| Nome nodo | Nome output | ID contenitore |
|--------------------------------------|------------------------------------|----------------|
| Maggior parte dei builder di modello | Maggior parte dei modelli generati | "PMML" |
| "autodataprep" | n/d | "PMML" |

Script di esempio

Il codice di script Python per accedere al contenuto potrebbe essere simile a quello riportato di seguito:

```
results = []
modelbuilder.run(results)
modeloutput = results[0]
cm = modeloutput.getContentModel("PMML")

dataFieldNames = cm.getStringValues("/PMML/DataDictionary/DataField", "name")
predictedNames = cm.getStringValues("//MiningSchema/MiningField[@usageType='predicted']", "name")
```

Modello di contenuto JSON

Il modello di contenuto JSON viene utilizzato per fornire il supporto per il contenuto in formato JSON. Fornisce un'API di base per consentire ai chiamanti di estrarre i valori, basandosi sull'ipotesi che questi ultimi conoscano i valori a cui accedere.

API

Tabella 27. API

| Restituzione | Metodo | Descrizione |
|---------------------------------------|--|--|
| String | getJSONAsString() | Restituisce il contenuto JSON come stringa. |
| Oggetto | getObjectAt(<List of object> path, JSONArtifact artifact) throws Exception | Restituisce l'oggetto nel percorso specificato. La risorsa root fornita può essere null; in questo caso, viene utilizzata la root del contenuto. Il valore restituito può essere una stringa a valore letterale, un intero, un numero reale o un valore booleano oppure una risorsa (un oggetto JSON o un array JSON). |
| Hash table (key:object, value:object) | getChildValuesAt(<List of object> path, JSONArtifact artifact) throws Exception | Restituisce i valori figlio del percorso specificato se il percorso indica un oggetto JSON o altrimenti null. Le chiavi nella tabella sono stringhe mentre il valore associato può essere una stringa a valore letterale, un intero, un numero reale o un valore booleano oppure una risorsa JSON (un oggetto JSON o un array JSON). |

Tabella 27. API (Continua)

| Restituzione | Metodo | Descrizione |
|-------------------|---|---|
| Elenco di oggetti | <code>getChildrenAt(<List of object> path path, JSONArtifact artifact)</code> throws Exception | Restituisce l'elenco di oggetti nel percorso specificato se il percorso indica un array JSON o altrimenti null. I valori restituiti possono essere una stringa a valore letterale, un intero, un numero reale o un valore booleano oppure una risorsa JSON (un oggetto JSON o un array JSON). |
| void | <code>reset()</code> | Svuota la memoria interna associata a questo modello di contenuto (ad esempio, un oggetto DOM memorizzato nella cache). |

Script di esempio

Se è presente un nodo builder di output che crea un output basato sul formato JSON, è possibile utilizzare il codice riportato di seguito per accedere alle informazioni relative ad un insieme di libri:

```
results = []
outputbuilder.run(results)
output = results[0]
cm = output.getContentModel("jsonContent")

bookTitle = cm.getObjectAt(["books", "ISIN123456", "title"], None)

# Alternatively, get the book object and use it as the root
# for subsequent entries
book = cm.getObjectAt(["books", "ISIN123456"], None)
bookTitle = cm.getObjectAt(["title"], book)

# Get all child values for a specific book
bookInfo = cm.getChildValuesAt(["books", "ISIN123456"], None)

# Get the third book entry. Assumes the top-level "books" value
# contains a JSON array which can be indexed
bookInfo = cm.getObjectAt(["books", 2], None)

# Get a list of all child entries
allBooks = cm.getChildrenAt(["books"], None)
```

Modello di contenuto delle statistiche di colonne e modello di contenuto delle statistiche pairwise

Il modello di contenuto delle statistiche di colonne fornisce l'accesso alle statistiche che possono essere calcolate per ciascun campo (statistiche univariate). Il modello di contenuto delle statistiche pairwise fornisce l'accesso alle statistiche che possono essere calcolate tra coppie di campi o valori in un campo.

Le misure delle statistiche possibili sono:

- Count
- UniqueCount
- ValidCount
- Mean
- Sum
- Min

- Max
- Range
- Variance
- StandardDeviation
- StandardErrorOfMean
- Skewness
- SkewnessStandardError
- Kurtosis
- KurtosisStandardError
- Median
- Mode
- Pearson
- Covarianza
- TTest
- FTest

Alcuni valori sono appropriati solo da statistiche di colonne singole mentre altri sono appropriati solo per le statistiche pairwise.

Di seguito sono riportati i nodi che producono tali valori:

- Il **nodo Statistiche** produce le statistiche di colonna e può produrre statistiche pairwise quando sono specificati i campi di correlazione
- Il **nodo Esplora** produce statistiche di colonne ed è in grado di produrre statistiche pairwise quando viene specificato un campo di sovrapposizione.
- Il **nodo Medie** produce statistiche pairwise quando viene eseguito il confronto tra coppie di campi o vengono confrontati i valori di un campo con i riepiloghi di un altro campo.

I modelli di contenuto e le statistiche disponibili dipendono dalle capacità del nodo e dalle impostazioni all'interno del nodo.

API ColumnStatsContentModel

Tabella 28. API ColumnStatsContentModel.

| Restituzione | Metodo | Descrizione |
|---------------------|--|--|
| List<StatisticType> | getAvailableStatistics() | Restituisce le statistiche disponibili in questo modello. Non tutti i campi dispongono necessariamente di valori per tutte le statistiche. |
| List<String> | getAvailableColumns() | Restituisce i nomi delle colonne per cui sono state calcolate le statistiche. |
| Number | getStatistic(String column, StatisticType statistic) | Restituisce i valori statistici associati alla colonna. |
| void | reset() | Svuota la memoria interna associata a questo modello di contenuto. |

API PairwiseStatsContentModel

Tabella 29. API PairwiseStatsContentModel.

| Restituzione | Metodo | Descrizione |
|---------------------|--|--|
| List<StatisticType> | getAvailableStatistics() | Restituisce le statistiche disponibili in questo modello. Non tutti i campi dispongono necessariamente di valori per tutte le statistiche. |
| List<String> | getAvailablePrimaryColumns() | Restituisce i nomi delle colonne primarie per cui sono state calcolate le statistiche. |
| List<Object> | getAvailablePrimaryValues() | Restituisce i valori della colonna primaria per cui sono state calcolate le statistiche. |
| List<String> | getAvailableSecondaryColumns() | Restituisce i nomi delle colonne secondarie per cui sono state calcolate le statistiche. |
| Number | getStatistic(String primaryColumn, String secondaryColumn, StatisticType statistic) | Restituisce i valori statistici associati alle colonne. |
| Number | getStatistic(String primaryColumn, Object primaryValue, String secondaryColumn, StatisticType statistic) | Restituisce i valori statistici associati al valore della colonna primaria ed alla colonna secondaria. |
| void | reset() | Svuota la memoria interna associata a questo modello di contenuto. |

Nodi ed output

Questa tabella elenca i nodi che creano output che includono questo tipo di modello di contenuto.

Tabella 30. Nodi ed output.

| Nome nodo | Nome output | ID contenitore | Note |
|------------------------------------|--------------|----------------------|---|
| "means" (nodo Medie) | "means" | "columnStatistics" | |
| "means" (nodo Medie) | "means" | "pairwiseStatistics" | |
| "dataaudit" (nodo Esplora) | "means" | "columnStatistics" | |
| "statistics" (nodo Statistiche) | "statistics" | "columnStatistics" | Generato solo quando vengono esaminati campi specifici. |
| "statistics" (nodo Statistiche) | "statistics" | "pairwiseStatistics" | Generato solo quando i campi sono correlati. |

Script di esempio

```
from modeler.api import StatisticType
stream = modeler.script.stream()

# Set up the input data
varfile = stream.createAt("variablefile", "File", 96, 96)
varfile.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")
```

```

# Now create the statistics node. This can produce both
# column statistics and pairwise statistics
statisticsnode = stream.createAt("statistics", "Stats", 192, 96)
statisticsnode.setPropertyValue("examine", ["Age", "Na", "K"])
statisticsnode.setPropertyValue("correlate", ["Age", "Na", "K"])
stream.link(varfile, statisticsnode)

results = []
statisticsnode.run(results)
statsoutput = results[0]
statscm = statsoutput.getContentModel("columnStatistics")
if (statscm != None):
    cols = statscm.getAvailableColumns()
    stats = statscm.getAvailableStatistics()
    print "Column stats:", cols[0], str(stats[0]), " = ", statscm.getStatistic(cols[0], stats[0])

statscm = statsoutput.getContentModel("pairwiseStatistics")
if (statscm != None):
    pcols = statscm.getAvailablePrimaryColumns()
    scols = statscm.getAvailableSecondaryColumns()
    stats = statscm.getAvailableStatistics()
    corr = statscm.getStatistic(pcols[0], scols[0], StatisticType.Pearson)
    print "Pairwise stats:", pcols[0], scols[0], " Pearson = ", corr

```

Capitolo 6. Argomenti della riga di comando

Modalità di richiamo del software

È possibile utilizzare la riga di comando del sistema operativo per avviare IBM SPSS Modeler:

1. Sul computer in cui è installato IBM SPSS Modeler, aprire una finestra DOS (prompt dei comandi).
2. Per avviare l'interfaccia di IBM SPSS Modeler in modalità interattiva, digitare il comando `modelerclient` seguito dagli argomenti richiesti, ad esempio:

```
modelerclient -stream report.str -execute
```

Gli argomenti disponibili (flag) consentono di connettersi a un server, caricare stream, eseguire script o specificare altri parametri.

Utilizzo degli argomenti della riga di comando

È possibile aggiungere alcuni argomenti della riga di comando (denominati anche *flag*) al comando `modelerclient` iniziale per modificare il modo in cui IBM SPSS Modeler viene richiamato.

Diversi tipi di argomenti della riga di comando non sono disponibili e vengono descritti in seguito in questa sezione.

Tabella 31. Tipi di argomenti della riga di comando.

| Tipo di argomento | Dove descritto |
|--|--|
| Argomenti di sistema | Per ulteriori informazioni, consultare l'argomento "Argomenti di sistema" a pagina 62. |
| Argomenti dei parametri | Per ulteriori informazioni, consultare l'argomento "Argomenti dei parametri" a pagina 63. |
| Argomenti per la connessione del server | Per ulteriori informazioni, consultare l'argomento "Argomenti per la connessione del server" a pagina 64. |
| Argomenti di connessione a IBM SPSS Collaboration and Deployment Services Repository | Per ulteriori informazioni, consultare l'argomento "Argomenti per la connessione a IBM SPSS Collaboration and Deployment Services Repository" a pagina 65. |
| Argomenti di connessione a IBM SPSS Analytic Server | Per ulteriori informazioni, consultare l'argomento "Argomenti per la connessione a IBM SPSS Analytic Server" a pagina 65. |

Per esempio, è possibile utilizzare gli argomenti della riga di comando `-server`, `-stream` ed `-execute` per connettersi a un server e caricare ed eseguire un flusso, come indicato di seguito:

```
modelerclient -server -hostname myserver -port 80 -username dminer  
-password 1234 -stream mystream.str -execute
```

Si noti che in caso di esecuzione dalla riga di comando con Clementine Client installato localmente, gli argomenti di connessione al server non sono necessari.

È possibile racchiudere tra virgolette i valori di parametri che contengono spazi, ad esempio:

```
modelerclient -stream mystream.str -Pusername="Joe User" -execute
```

Questa soluzione consente anche di eseguire stati e script di IBM SPSS Modeler, utilizzando rispettivamente i flag `-state` e `-script`.

Nota: Se in un comando viene utilizzato un parametro strutturato, è necessario inserire le virgolette una barra rovesciata. In questo modo, le virgolette non vengono rimosse durante l'interpretazione della stringa.

Debug degli argomenti della riga di comando

Per eseguire il debug di una riga di comando, utilizzare il comando `modelerclient` per avviare IBM SPSS Modeler con gli argomenti desiderati. Ciò consente di verificare che i comandi vengano eseguiti come previsto. È possibile confermare i valori di qualsiasi parametro passato dalla riga di comando nella finestra di dialogo Parametri di sessione (menu Strumenti, Imposta parametri di sessione).

Argomenti di sistema

Nella tabella seguente sono illustrati gli argomenti di sistema disponibili per il richiamo dell'interfaccia utente dalla riga di comando.

Tabella 32. Argomenti di sistema

| Argomento | Comportamento/descrizione |
|-------------------------|--|
| @ <Filecomando> | Il carattere '@' seguito da un nome di file specifica un elenco di comandi. Quando <code>modelerclient</code> incontra un argomento che inizia con @, utilizza i comandi del file esattamente come se fossero stati specificati nella riga di comando. Per ulteriori informazioni, consultare l'argomento "Combinazione di più argomenti" a pagina 66. |
| -directory <dir> | Consente di impostare la directory di default. Nella modalità locale tale directory viene utilizzata sia per i dati che per l'output. Esempio: <code>-directory c:/</code> o <code>-directory c:\\</code> |
| -server_directory <dir> | Consente di impostare la directory del server di default per i dati. Tale directory, specificata mediante il flag <code>-directory</code> , viene utilizzata per l'output. |
| -execute | Dopo l'avvio, consente di eseguire eventuali flussi, stati o script caricati all'avvio. Se oltre a un flusso o stato viene caricato anche uno script, verrà eseguito solamente lo script. |
| -stream <stream> | Consente di caricare all'avvio il flusso specificato. È possibile specificare più flussi, ma l'ultimo flusso specificato verrà impostato come flusso corrente. |
| -script <script> | Consente di caricare all'avvio lo script autonomo specificato. Come illustrato di seguito, tale script può essere specificato insieme a un flusso o a uno stato, ma è possibile caricare un solo script all'avvio. |
| -model <modello> | Consente di caricare all'avvio il modello generato specificato (formato di file <code>.gm</code>). |
| -state <stato> | Consente di caricare all'avvio lo stato salvato specificato. |
| -project <progetto> | Consente di caricare il progetto specificato. È possibile caricare un solo progetto all'avvio. |
| -output <output> | Consente di caricare l'oggetto di output salvato (file di formato <code>COU</code>) all'avvio. |
| -help | Consente di visualizzare un elenco di argomenti della riga di comando. Quando si specifica questa opzione, tutti gli altri argomenti vengono ignorati e viene visualizzata la finestra della Guida in linea. |
| -P <nome>=<valore> | Utilizzato per impostare un parametro di avvio. Può essere utilizzato anche per impostare le proprietà dei nodi (parametri di slot). |

Nota: è possibile impostare le directory di default anche nell'interfaccia utente. Per accedere alle opzioni, scegliere **Imposta directory** o **Imposta directory server** dal menu File.

Caricamento di più file

Dalla riga di comando è possibile caricare più flussi, stati e output all'avvio ripetendo l'argomento rilevante per ogni oggetto caricato. Per esempio, per caricare ed eseguire due stream denominati *report.str* e *train.str*, è necessario utilizzare il seguente comando:

```
modelerclient -stream report.str -stream train.str -execute
```

Caricamento di oggetti da IBM SPSS Collaboration and Deployment Services Repository

Poiché è possibile caricare determinati oggetti da un file o da IBM SPSS Collaboration and Deployment Services Repository (se concesso in licenza), il prefisso `spsscr:` che precede il nome file e, facoltativamente, `file:` (per oggetti su disco) indica IBM SPSS Modeler dove cercare l'oggetto. Il prefisso viene utilizzato con i seguenti flag:

- `-stream`
- `-script`
- `-output`
- `-model`
- `-project`

Il prefisso utilizzato per creare un URI che specifica l'ubicazione dell'oggetto, ad esempio `-stream "spsscr:///folder_1/scoring_stream.str"`. La presenza del prefisso `spsscr:` richiede che nello stesso comando sia stata specificata una connessione valida a IBM SPSS Collaboration and Deployment Services Repository. Pertanto il comando completo si presenterà come segue:

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080  
-spsscr_username myusername -spsscr_password mypassword  
-stream "spsscr:///cartella_1/punteggio_stream.str".
```

Si noti che dalla riga di comando è *necessario* utilizzare un URI. Non è infatti supportato il semplice `REPOSITORY_PATH` che viene utilizzato solo all'interno di script. Per ulteriori informazioni sugli URI per gli oggetti nel IBM SPSS Collaboration and Deployment Services Repository, consultare l'argomento "Accesso a oggetti nel IBM SPSS Collaboration and Deployment Services Repository" a pagina 50.

Argomenti dei parametri

I parametri possono essere utilizzati come flag durante l'esecuzione della riga di comando di IBM SPSS Modeler. Negli argomenti della riga di comando il flag `-P` consente di specificare un parametro nel formato `-P <nome>=<valore>`.

I parametri possono essere dei seguenti tipi:

- **Parametri semplici** o parametri utilizzati direttamente nelle espressioni CLEM.
- **Parametri di slot**, detti anche **proprietà dei nodi**. Questi parametri vengono utilizzati per modificare le impostazioni dei nodi nel flusso. Per ulteriori informazioni, consultare l'argomento "Panoramica sulle proprietà dei nodi" a pagina 69.
- **Parametri della riga di comando** che consentono di modificare il richiamo di IBM SPSS Modeler.

Per esempio, è possibile specificare nomi utente e password per le sorgenti dei dati sotto forma di flag della riga di comando, come nel seguente esempio:

```
modelerclient -stream response.str -P:databasenode.datasource="{\"ORA 10gR2\", user1, mypsw,  
true}"
```

Il formato è lo stesso del parametro `datasource` della proprietà del nodo `databasenode`. Per ulteriori informazioni, consultare: "Proprietà `databasenode`" a pagina 81.

Nota: Se il nodo è denominato, è necessario racchiudere il nome del nodo tra virgolette e utilizzare la barra rovesciata come carattere di escape per le virgolette. Ad esempio, se il nodo dell'origine dati nell'esempio precedente è denominato *Source_ABC*, la voce è simile a quella riportata di seguito:

```
modelerclient -stream response.str -P:databasenode.\"Source_ABC\".datasource=\"{\"ORA 10gR2\",
user1, mypsw, true}\"
```

È inoltre necessario utilizzare una barra rovesciata prima delle virgolette che identificano un parametro strutturato, come riportato nel seguente esempio di origine dati TML:

```
clomb -server -hostname 9.115.21.169 -port 28053 -username administrator
-execute -stream C:\Share\TML_Script.str -P:tmlimport.pm_host="http://9.115.21.163:9510/pmhub/pm"
-P:tmlimport.tml_connection={\"SData\", \"\", \"admin\", \"apple\"}
-P:tmlimport.selected_view={\"SalesPriorCube\", \"salesmargin%\"}
```

Argomenti per la connessione del server

Il flag `-server` indica che è necessario che IBM SPSS Modeler si connetta a un server pubblico e i flag `-hostname`, `-use_ssl`, `-port`, `-username`, `-password` e `-domain` vengono utilizzati per fornire a IBM SPSS Modeler i parametri necessari per la connessione al server pubblico. Se non viene specificato un argomento `-server`, viene utilizzato il server di default o locale.

Esempi

Per connettersi a un server pubblico:

```
modelerclient -server -hostname myserver -port 80 -username dminer
-password 1234 -stream mystream.str -execute
```

Per connettersi a un cluster di server:

```
modelerclient -server -cluster "QA Machines" \
-spsscr_hostname pes_host -spsscr_port 8080 \
-spsscr_username asmith -spsscr_epassword xyz
```

Si noti che la connessione a un cluster di server richiede il plug-in Coordinator of Processes attraverso IBM SPSS Collaboration and Deployment Services, quindi l'argomento `-cluster` deve essere utilizzato in combinazione con le opzioni di connessione al repository (`spsscr_*`). Per ulteriori informazioni, consultare l'argomento "Argomenti per la connessione a IBM SPSS Collaboration and Deployment Services Repository" a pagina 65.

Tabella 33. Argomenti per la connessione del server.

| Argomento | Comportamento/descrizione |
|---|--|
| <code>-server</code> | Esegue IBM SPSS Modeler in modalità server, effettuando la connessione a un server pubblico tramite i flag <code>-hostname</code> , <code>-port</code> , <code>-username</code> , <code>-password</code> e <code>-domain</code> . |
| <code>-hostname <nome></code> | Nome host del server. Disponibile solo nella modalità server. |
| <code>-use_ssl</code> | Specifica che la connessione deve utilizzare il protocollo SSL (Secure Socket Layer). Flag facoltativo; l'impostazione predefinita <i>non</i> prevede l'uso di SSL. |
| <code>-port <numero></code> | Numero di porta del server specificato. Disponibile solo nella modalità server. |
| <code>-cluster <nome></code> | Specifica una connessione a un cluster di server invece che a un server denominato. Argomento alternativo a <code>hostname</code> , <code>port</code> e <code>use_ssl</code> . Il nome è il nome del cluster o un URI univoco che identifica il cluster in IBM SPSS Collaboration and Deployment Services Repository. Il cluster di server viene gestito da Coordinator of Processes attraverso IBM SPSS Collaboration and Deployment Services. Per ulteriori informazioni, consultare l'argomento "Argomenti per la connessione a IBM SPSS Collaboration and Deployment Services Repository" a pagina 65. |
| <code>-username <nome></code> | Nome utente utilizzato per l'accesso al server. Disponibile solo nella modalità server. |
| <code>-password <password></code> | Password utilizzata per l'accesso al server. Disponibile solo nella modalità server. <i>Nota:</i> se l'argomento <code>-password</code> non viene specificato, verrà richiesta l'immissione di una password. |

Tabella 33. Argomenti per la connessione del server (Continua).

| Argomento | Comportamento/descrizione |
|---|---|
| -epassword <stringapasswordcodificata> | Password codificata utilizzata per l'accesso al server. Disponibile solo nella modalità server. <i>Nota:</i> per creare una password codificata, utilizzare il menu Strumenti dell'applicazione IBM SPSS Modeler. |
| -domain <nome> | Domaino utilizzato per l'accesso al server. Disponibile solo nella modalità server. |
| -P <nome>=<valore> | Utilizzato per impostare un parametro di avvio. Può essere utilizzato anche per impostare le proprietà dei nodi (parametri di slot). |

Argomenti per la connessione a IBM SPSS Collaboration and Deployment Services Repository

Se si desidera archiviare o recuperare oggetti da IBM SPSS Collaboration and Deployment Services tramite la riga di comando, è necessario specificare una connessione valida a IBM SPSS Collaboration and Deployment Services Repository. Ad esempio:

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080
-spsscr_username myusername -spsscr_password mypassword
-stream "spsscr:///cartella_1/punteggio_stream.str".
```

Nella tabella riportata di seguito sono elencati gli argomenti da utilizzare per impostare la connessione.

Tabella 34. Argomenti per la connessione a IBM SPSS Collaboration and Deployment Services Repository

| Argomento | Comportamento/descrizione |
|---|--|
| -spsscr_hostname <nome host o indirizzo IP> | Nome host o indirizzo IP del server su cui è installato IBM SPSS Collaboration and Deployment Services Repository. |
| -spsscr_port <numero> | Numero di porta su cui IBM SPSS Collaboration and Deployment Services Repository accetta connessioni, generalmente è la porta 8080 di default. |
| -spsscr_use_ssl | Specifica che la connessione deve utilizzare il protocollo SSL (Secure Socket Layer). Flag facoltativo; l'impostazione predefinita <i>non</i> prevede l'uso di SSL. |
| -spsscr_username <nome> | Nome utente utilizzato per l'accesso a IBM SPSS Collaboration and Deployment Services Repository. |
| -spsscr_password <password> | Password utilizzata per l'accesso a IBM SPSS Collaboration and Deployment Services Repository. |
| -spsscr_epassword <password codificata> | Password codificata utilizzata per l'accesso a IBM SPSS Collaboration and Deployment Services Repository. |
| -spsscr_domain <nome> | Domaino utilizzato per l'accesso a IBM SPSS Collaboration and Deployment Services Repository. Indicatore facoltativo; utilizzarlo solo se si effettua l'accesso con LDAP o Active Directory. |

Argomenti per la connessione a IBM SPSS Analytic Server

Se si desidera archiviare o recuperare oggetti da IBM SPSS Analytic Server attraverso la riga di comando, è necessario specificare una connessione valida a IBM SPSS Analytic Server.

Nota: Il percorso di Analytic Server viene ottenuto da SPSS Modeler Server e non può essere modificato sul client.

Nella tabella riportata di seguito sono elencati gli argomenti da utilizzare per impostare la connessione.

Tabella 35. Argomenti di connessione a IBM SPSS Analytic Server

| Argomento | Comportamento/descrizione |
|-----------------------------|---|
| -analytic_server_username | Il nome utente con cui eseguire l'accesso a IBM SPSS Analytic Server. |
| -analytic_server_password | La password con cui eseguire l'accesso a IBM SPSS Analytic Server. |
| -analytic_server_epassword | La password codificata con cui eseguire l'accesso a IBM SPSS Analytic Server. |
| -analytic_server_credential | Le credenziali utilizzate per l'accesso a IBM SPSS Analytic Server. |

Combinazione di più argomenti

È possibile combinare più argomenti in un unico file dei comandi, che potrà essere specificato all'avvio utilizzando il simbolo @ seguito dal nome del file. In questo modo è possibile abbreviare il richiamo dalla riga di comando e superare eventuali limitazioni di lunghezza dei comandi previste dal sistema operativo. Per esempio, il seguente comando di avvio utilizza gli argomenti specificati nel file indicato da <NomeFilecomando>.

```
modelerclient @<NomeFilecomando>
```

Se è necessario specificare degli spazi, racchiudere il nome del file e il percorso tra virgolette, per esempio:

```
modelerclient @ "C:\Program Files\IBM\SPSS\Modeler\nn\scripts\my_command_file.txt"
```

Il file dei comandi può contenere tutti gli argomenti che in precedenza venivano specificati singolarmente all'avvio, con un argomento per riga. Ad esempio:

```
-stream report.str
-Porder.full_filename=APR_orders.dat
-Preport.filename=APR_report.txt
-execute
```

Quando si scrivono o si richiamano file dei comandi è importante attenersi alle seguenti indicazioni:

- Specificare un solo comando per riga.
- Non incorporare un argomento @fileComando in un file dei comandi.

Capitolo 7. Guida alle proprietà

Panoramica sui riferimenti alle proprietà

È possibile specificare numerose proprietà per nodi, stream, Supernodi e progetti. Alcune proprietà sono comuni a tutti i nodi, per esempio name, annotation e ToolTip, altre invece sono specifiche di alcuni tipi di nodi. Altre proprietà fanno riferimento a operazioni di alto livello dei flussi, quali la memorizzazione nella cache o il funzionamento dei Supernodi. È possibile accedere alle proprietà tramite l'interfaccia utente standard (per esempio, tramite la finestra di dialogo per la modifica delle opzioni di un nodo) e utilizzarle in molti modi.

- È possibile modificare le proprietà tramite gli script, come illustrato in questa sezione. Per ulteriori informazioni, vedere "Sintassi delle proprietà".
- È possibile utilizzare le proprietà dei nodi nei parametri dei Supernodi.
- Le proprietà dei nodi possono inoltre essere specificate come parte di un'opzione della riga di comando (mediante il flag -P) all'avvio di IBM SPSS Modeler.

Per gli script di IBM SPSS Modeler, le proprietà dei nodi e dei flussi vengono spesso denominate **parametri di slot**. In questa guida, verranno invece definite come proprietà dei nodi o dei flussi.

Per ulteriori informazioni sul linguaggio di script, vedere Linguaggio di script.

Sintassi delle proprietà

È possibile impostare le proprietà utilizzando la sintassi riportata di seguito

```
OBJECT.setPropertyValue(PROPERTY, VALUE)
```

oppure:

```
OBJECT.setKeyedPropertyValue(PROPERTY, KEY, VALUE)
```

È possibile richiamare il valore delle proprietà utilizzando la seguente sintassi:

```
VARIABLE = OBJECT.getPropertyValue(PROPERTY)
```

oppure:

```
VARIABLE = OBJECT.getKeyedPropertyValue(PROPERTY, KEY)
```

dove OBJECT è un nodo o un output, PROPERTY è il nome della proprietà del nodo a cui fa riferimento l'espressione e KEY è il valore della chiave per le proprietà basate su chiavi. Ad esempio, la seguente sintassi viene utilizzata per trovare il nodo filtro e quindi impostare il valore predefinito per includere tutti i campi e filtrare il campo Age dai dati downstream:

```
filternode = modeler.script.stream().findByType("filter", None)
filternode.setPropertyValue("default_include", True)
filternode.setKeyedPropertyValue("include", "Age", False)
```

Tutti i nodi utilizzati in IBM SPSS Modeler possono essere individuati utilizzando la funzione `findByType(TYPE, LABEL)` del flusso. È necessario specificare almeno uno tra TYPE o LABEL.

Proprietà strutturate

Le proprietà strutturate vengono utilizzate negli script per semplificare l'analisi ed essenzialmente per due motivi:

- Per strutturare i nomi delle proprietà dei nodi complessi, quali i nodi tipo, filtro o bilanciamento.
- Per rendere disponibile un formato per la specifica di più proprietà contemporaneamente.

Strutturazione per interfacce complesse

Gli script per i nodi con tabelle e altre interfacce complesse quali i nodi tipo, filtro e bilanciamento devono avere una struttura particolare per poter eseguire l'analisi correttamente. A queste proprietà deve essere assegnato un nome più complesso di quello di un singolo identificativo; questo nome è denominato la chiave. Per esempio, all'interno di un nodo Filtro ogni campo disponibile (nel lato a monte) è attivato o disattivato. Per fare riferimento a queste informazioni, il nodo Filtro archivia un'informazione per campo (se ogni campo è true o false). Questa proprietà può avere (o alla quale può essere assegnato) il valore True o False. Si supponga che un nodo Filtro denominato `mi` includa un campo (nel lato upstream) denominato `Età`. Per disattivarlo, impostare la proprietà `include`, con la chiave `Età`, sul valore `False`, come indicato di seguito:

```
mynode.setKeyedPropertyValue("include", "Age", False)
```

Strutturazione per l'impostazione di proprietà multiple

Per molti nodi è possibile assegnare più di una proprietà di nodo o di flusso contemporaneamente. Questo tipo di operazione è definita **comando multiset** o **blocco di impostazioni**.

A volte le proprietà strutturate possono essere molto complesse. Di seguito è riportato un esempio:

```
sortnode.setPropertyValue("keys", [{"K", "Descending"}, {"Age", "Ascending"}, {"Na", "Descending"}])
```

Un altro vantaggio delle proprietà strutturate consiste nella possibilità di impostare numerose proprietà in un nodo prima che questo diventi stabile. Per default, un comando multiset definisce tutte le proprietà del blocco prima di eseguire qualsiasi operazione basata sull'impostazione di una singola proprietà. Per esempio, se si definisce un nodo `Testo` fisso e si utilizzano due passaggi per impostare le proprietà dei campi, verranno generati degli errori perché il nodo non è coerente fino a quando entrambe le impostazioni non sono valide. La definizione delle proprietà come un multiset elude il problema, perché entrambe le proprietà vengono impostate prima dell'aggiornamento del modello di dati.

Abbreviazioni

Per le proprietà dei nodi, nella sintassi vengono utilizzate abbreviazioni standard il cui apprendimento può risultare utile per la creazione degli script.

Tabella 36. Abbreviazioni standard utilizzate nella sintassi

| Abbreviazione | Significato |
|---------------|---|
| abs | In valore assoluto |
| len | Lunghezza |
| min | Minimo |
| max | Massimo |
| correl | Correlazione |
| covar | Covarianza |
| num | Numero o numerico |
| pct | Per cento o percentuale |
| transp | Trasparenza |
| xval | Convalida incrociata |
| var | Varianza o variabile (nei nodi origine) |

Esempi di proprietà dei nodi e dei flussi

IBM SPSS Modeler consente di utilizzare le proprietà di nodi e stream in vari modi. Nella maggior parte dei casi vengono utilizzate come parte di uno script, sia di uno **script autonomo** che consente di automatizzare più flussi o più operazioni, sia di uno **script del flusso** che consente di automatizzare i

processi all'interno di un singolo flusso. È possibile specificare i parametri dei nodi anche utilizzando le proprietà dei nodi all'interno del Supernodo. A livello di base, le proprietà possono inoltre essere utilizzate come opzioni della riga di comando per l'avvio di IBM SPSS Modeler. L'utilizzo dell'argomento `-p` nella chiamata alla riga di comando consente di modificare un'impostazione del flusso mediante una proprietà del flusso.

Tabella 37. Esempi di proprietà dei nodi e dei flussi

| Proprietà | Significato |
|----------------------------------|---|
| <code>s.max_size</code> | Fa riferimento alla proprietà <code>max_size</code> del nodo denominato <code>s</code> . |
| <code>s:samplene.max_size</code> | Fa riferimento alla proprietà <code>max_size</code> del nodo denominato <code>s</code> , che deve essere un nodo Campione. |
| <code>:samplene.max_size</code> | Fa riferimento alla proprietà <code>max_size</code> del nodo Campione dello stream corrente (deve essere presente un solo nodo Campione). |
| <code>s:samplene.max_size</code> | Fa riferimento alla proprietà <code>max_size</code> del nodo denominato <code>s</code> , che deve essere un nodo Campione. |
| <code>t.direction.Età</code> | Fa riferimento al ruolo del campo <i>Età</i> nel nodo Tipo <code>t</code> . |
| <code>:.max_size</code> | *** NON VALIDO *** È necessario specificare il nome o il tipo del nodo. |

L'esempio `s:sample.max_size` indica che non è necessario scrivere per esteso i tipi di nodo.

L'esempio `t.direction.Age` indica che anche i nomi di alcuni slot possono essere strutturati, se gli attributi di un nodo sono più complessi delle singole configurazioni con valori singoli. Questi slot sono definiti proprietà **strutturate** o **complesse**.

Panoramica sulle proprietà dei nodi

Per ogni tipo di nodo è disponibile un insieme specifico di proprietà valide. Questo tipo può essere generale, numero, indicatore o stringa, in cui le impostazioni relative alla proprietà vengono forzate sul tipo corretto. Se questo non è possibile, viene generato un errore. In alternativa, è possibile che il riferimento alla proprietà specifichi l'intervallo di valori validi, per esempio `Discard`, `PairAndDiscard` e `IncludeAsText`, nel qual caso verrà generato un errore se si utilizza un qualsiasi altro valore. Le proprietà flag dovrebbero essere lette o impostate utilizzando i valori `vero` e `falso`. (Per l'impostazione di questi valori è inoltre possibile utilizzare variazioni quali `Off`, `OFF`, `off`, `No`, `NO`, `no`, `n`, `N`, `f`, `F`, `false`, `False`, `FALSE` o `0` vengono riconosciute anche durante l'impostazione dei valori ma, in alcuni casi, potrebbero causare errori durante la lettura dei valori delle proprietà. Tutti gli altri valori vengono considerati come `vero`. L'utilizzo coerente di `vero` e `falso` consente di evitare confusioni. Nelle tabelle di riferimento di questa guida, le proprietà strutturate vengono indicate come tali nella colonna *Descrizione proprietà* e vengono inoltre forniti i relativi formati di utilizzo.

Proprietà comuni dei nodi

Esistono numerose proprietà comuni a tutti i nodi di IBM SPSS Modeler, inclusi i Supernodi.

Tabella 38. Proprietà comuni dei nodi.

| Nome proprietà | Data type | Descrizione proprietà |
|------------------------------|-------------------|---|
| <code>use_custom_name</code> | <i>indicatore</i> | |
| <code>name</code> | <i>stringa</i> | La proprietà di sola lettura che legge il nome (automatico o personalizzato) per un nodo nell'area. |

Tabella 38. Proprietà comuni dei nodi (Continua).

| Nome proprietà | Data type | Descrizione proprietà |
|----------------|---|---|
| custom_name | stringa | Specifica un nome personalizzato per il nodo. |
| tooltip | stringa | |
| annotation | stringa | |
| keywords | stringa | Slot strutturato che specifica un elenco di parole chiave associate all'oggetto, per esempio, ["Keyword1" "Keyword2"] |
| cache_enabled | indicatore | |
| node_type | source_supernode process_supernode terminal_supernode tutti i nomi di nodi, come specificato script | Proprietà di sola lettura utilizzata per fare riferimento a un nodo in base al tipo. Per esempio, invece di fare riferimento a un nodo utilizzando solo il nome, quale real_income, è anche possibile specificarne il tipo, per esempio userInputnode o filternode. |

Le proprietà specifiche dei Supernodi vengono illustrate separatamente, analogamente a tutti gli altri nodi. Per ulteriori informazioni, consultare l'argomento Capitolo 19, "Proprietà dei Supernodi", a pagina 315.

Capitolo 8. Proprietà del flusso

Gli script consentono di controllare numerose proprietà dei flussi. Per indicare le proprietà del flusso, è necessario impostare il metodo di esecuzione per utilizzare gli script:

```
stream = modeler.script.stream()
stream.setPropertyValue("execute_method", "Script")
```

Esempio

La proprietà del nodo viene utilizzata per fare riferimento ai nodi nel flusso corrente. Un esempio è costituito dallo script del flusso seguente:

```
stream = modeler.script.stream()
annotation = stream.getPropertyValue("annotation")

annotation = annotation + "\n\nThis stream is called \"" + stream.getLabel() + "\" and
contains the following nodes:\n"

for node in stream.iterator():
    annotation = annotation + "\n" + node.getTypeName() + " node called \"" + node.getLabel()
    + "\""

stream.setPropertyValue("annotation", annotation)
```

Nell'esempio riportato sopra, la proprietà del nodo viene utilizzata per creare un elenco di tutti i nodi dello stream e per scrivere tale elenco nelle annotazioni dello stream. L'annotazione generata si presenta nel modo seguente:

Il flusso è denominato "druglearn" e contiene i nodi seguenti:

```
type node called "Define Types"
derive node called "Na_to_K"
variablefile node called "DRUG1n"
neuralnetwork node called "Drug"
c50 node called "Drug"
filter node called "Discard Fields"
```

Nella tabella seguente vengono illustrate le proprietà dei flussi.

Tabella 39. Proprietà dei flussi.

| Nome proprietà | Data type | Descrizione proprietà |
|----------------|------------------|-----------------------|
| execute_method | Normal Script | |

Tabella 39. Proprietà dei flussi (Continua).

| Nome proprietà | Data type | Descrizione proprietà |
|---------------------------|---|-----------------------|
| date_format | "DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY | |
| date_baseline | numero | |
| date_2digit_baseline | numero | |
| time_format | "HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM." "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S" | |
| time_rollover | indicatore | |
| import_datetime_as_string | indicatore | |
| decimal_places | numero | |
| decimal_symbol | Default Period Comma | |
| angles_in_radians | indicatore | |
| use_max_set_size | indicatore | |
| max_set_size | numero | |
| ruleset_evaluation | Voting FirstHit | |

Tabella 39. Proprietà dei flussi (Continua).

| Nome proprietà | Data type | Descrizione proprietà |
|---|--------------------------|---|
| refresh_source_nodes | indicatore | Consente di aggiornare automaticamente i nodi origine all'esecuzione del flusso. |
| script | stringa | |
| annotation | stringa | |
| name | stringa | Nota: questa proprietà è di sola lettura. Se si desidera modificare il nome di uno stream, salvarlo con un nome diverso. |
| parametri | | Utilizzare questa proprietà per aggiornare i parametri dei flussi dall'interno di uno script autonomo. |
| nodes | | Vedere le informazioni dettagliate riportate di seguito. |
| encoding | SystemDefault "UTF-8" | |
| stream_rewriting | booleano | |
| stream_rewriting_maximise_sql | booleano | |
| stream_rewriting_optimise_clem_esecuzione | booleano | |
| stream_rewriting_optimise_syntax_esecuzione | booleano | |
| enable_parallelism | booleano | |
| sql_generation | booleano | |
| database_caching | booleano | |
| sql_logging | booleano | |
| sql_generation_logging | booleano | |
| sql_log_native | booleano | |
| sql_log_prettyprint | booleano | |
| record_count_suppress_input | booleano | |
| record_count_feedback_interval | numero intero | |
| use_stream_auto_create_node_impostazioni | booleano | Se true, vengono utilizzate le impostazioni specifiche del flusso; in caso contrario, vengono utilizzate le preferenze utente. |
| create_model_applier_for_new_modelli | booleano | Se true, quando un builder del modello crea un nuovo modello e non dispone di collegamenti di aggiornamento attivi, viene aggiunto un nuovo applicatore del modello. Nota: Se si utilizza IBM SPSS Modeler Batch versione 15, è necessario aggiungere in modo esplicito l'applicatore del modello all'interno del proprio script. |

Tabella 39. Proprietà dei flussi (Continua).

| Nome proprietà | Data type | Descrizione proprietà |
|-----------------------------------|--|---|
| create_model_applier_update_links | createEnabled createDisabled doNotCreate | Definisce il tipo di collegamento creato quando un nodo applicatore del modello viene aggiunto automaticamente. |
| create_source_node_from_builders | booleano | Se true, quando un builder di origine crea un nuovo output di origine e non dispone di collegamenti di aggiornamento attivi, viene aggiunto un nuovo nodo di origine. |
| create_source_node_update_links | createEnabled createDisabled doNotCreate | Definisce il tipo di collegamento creato quando un nodo di origine viene aggiunto automaticamente. |
| has_coordinate_system | booleano | Se true, viene applicato un sistema di coordinate al flusso intero. |
| coordinate_system | stringa | Il nome del sistema di coordinate proiettate selezionato. |

Capitolo 9. Proprietà dei nodi origine

Proprietà comuni dei nodi origine

Di seguito vengono elencate le proprietà comuni a tutti i nodi origine, corredate da informazioni sui nodi specifici negli argomenti che seguono.

Esempio 1

```
varfilenode = modeler.script.stream().create("variablefile", "Var. File")
varfilenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
varfilenode.setKeyedPropertyValue("check", "Age", "None")
varfilenode.setKeyedPropertyValue("values", "Age", [1, 100])
varfilenode.setKeyedPropertyValue("type", "Age", "Range")
varfilenode.setKeyedPropertyValue("direction", "Age", "Input")
```

Esempio 2

In questo script si suppone che il file di dati specificato contenga un campo denominato Region che rappresenta una stringa a più righe.

```
from modeler.api import StorageType
from modeler.api import MeasureType

# Create a Variable File node that reads the data set containing
# the "Region" field
varfilenode = modeler.script.stream().create("variablefile", "My Geo Data")
varfilenode.setPropertyValue("full_filename", "C:/mydata/mygeodata.csv")
varfilenode.setPropertyValue("treat_square_brackets_as_lists", True)

# Override the storage type to be a list...
varfilenode.setKeyedPropertyValue("custom_storage_type", "Region", StorageType.LIST)
# ...and specify the type if values in the list and the list depth
varfilenode.setKeyedPropertyValue("custom_list_storage_type", "Region", StorageType.INTEGER)
varfilenode.setKeyedPropertyValue("custom_list_depth", "Region", 2)

# Now change the measurement to identify the field as a geospatial value...
varfilenode.setKeyedPropertyValue("measure_type", "Region", MeasureType.GEOSPATIAL)
# ...and finally specify the necessary information about the specific
# type of geospatial object
varfilenode.setKeyedPropertyValue("geo_type", "Region", "MultiLineString")
varfilenode.setKeyedPropertyValue("geo_coordinates", "Region", "2D")
varfilenode.setKeyedPropertyValue("has_coordinate_system", "Region", True)
varfilenode.setKeyedPropertyValue("coordinate_system", "Region", "ETRS_1989_EPSG_Arctic_zone_5-47")
```

Tabella 40. Proprietà comuni dei nodi origine.

| Nome proprietà | Data type | Descrizione proprietà |
|----------------|--|--|
| direction | Input Target Both None Partition Split Frequency RecordID | Proprietà basata su chiavi per i ruoli del campo. Formato di utilizzo: NODE.direction.FIELDNAME Nota: i valori In e Out sono obsoleti. Nelle versioni future potrebbero non essere più supportati. |

Tabella 40. Proprietà comuni dei nodi origine (Continua).

| Nome proprietà | Data type | Descrizione proprietà |
|--------------------|---|---|
| type | Range Flag Set Typeless Discrete Insieme ordinato Default | Tipo di campo. L'impostazione di questa proprietà su <i>Default</i> cancella qualsiasi impostazione di proprietà <i>values</i> e se <i>value_mode</i> è impostata su <i>Specifica</i> , verrà reimpostata su <i>Leggi</i> . Se <i>value_mode</i> è già impostata su <i>Passa</i> o <i>Leggi</i> , non verrà influenzata dall'impostazione <i>type</i> . Formato di utilizzo: NODE.type.FIELDNAME |
| storage | Unknown String Integer Real Time Date Timestamp | Proprietà basata su chiavi in sola lettura per il tipo di archiviazione del campo. Formato di utilizzo: NODE.storage.FIELDNAME |
| check | None Annulla Coerce Discard Warn Abort | Proprietà basata su chiavi per il controllo del tipo di campo e dell'intervallo. Formato di utilizzo: NODE.check.FIELDNAME |
| values | [valore valore] | Per un campo continuo (intervallo), il primo valore corrisponde al minimo e l'ultimo valore al massimo. Per i campi nominali (insieme), specificare tutti i valori. Nel caso dei campi flag, il primo valore rappresenta <i>falso</i> e l'ultimo valore rappresenta <i>vero</i> . L'impostazione automatica di questa proprietà consente di impostare la proprietà <i>value_mode</i> su <i>Specify</i> . L'archiviazione è determinata in base al primo valore nell'elenco, ad esempio, se il primo valore è una <i>stringa</i> , l'archiviazione viene impostata su <i>String</i> . Formato di utilizzo: NODE.values.FIELDNAME |
| value_mode | Read Pass Leggi+ Current Specify | Determina la modalità di impostazione dei valori per un campo nel passaggio di dati successivo. Formato di utilizzo: NODE.value_mode.FIELDNAME Si noti che non è possibile impostare questa proprietà direttamente su <i>Specify</i> . Per utilizzare valori specifici, impostare la proprietà <i>values</i> . |
| default_value_mode | Read Pass | Specifica il metodo di default per l'impostazione dei valori di tutti i campi. Formato di utilizzo: NODE.default_value_mode È possibile sovrascrivere questa impostazione per campi specifici utilizzando la proprietà <i>value_mode</i> . |

Tabella 40. Proprietà comuni dei nodi origine (Continua).

| Nome proprietà | Data type | Descrizione proprietà |
|--------------------|---------------------|--|
| extend_values | <i>indicatore</i> | Viene applicato quando <i>value_mode</i> è impostata su <i>Read</i> . Per aggiungere valori appena letti a eventuali valori esistenti per il campo, impostare su <i>T</i> . Per scartare i valori esistenti e sostituirli con i valori appena letti, impostare su <i>F</i> . Formato di utilizzo: NODE.extend_values.FIELDNAME |
| value_labels | <i>stringa</i> | Utilizzata per specificare l'etichetta di un valore. Tenere presente che i valori devono essere specificati prima. |
| enable_missing | <i>indicatore</i> | Se impostato su <i>V</i> , attiva la registrazione dei valori mancanti per il campo. Formato di utilizzo: NODE.enable_missing.FIELDNAME |
| missing_values | [valore valore ...] | Specifica i valori dei dati che indicano dati mancanti. Formato di utilizzo: NODE.missing_values.FIELDNAME |
| range_missing | <i>indicatore</i> | Quando questa proprietà è impostata su <i>T</i> , specifica se viene definito un intervallo di valori mancanti (vuoti) per un campo. Formato di utilizzo: NODE.range_missing.FIELDNAME |
| missing_lower | <i>stringa</i> | Se <i>range_missing</i> è impostata su vero, specifica il limite inferiore dell'intervallo di valori mancanti. Formato di utilizzo: NODE.missing_lower.FIELDNAME |
| missing_upper | <i>stringa</i> | Se <i>range_missing</i> è impostata su vero, specifica il limite superiore dell'intervallo di valori mancanti. Formato di utilizzo: NODE.missing_upper.FIELDNAME |
| null_missing | <i>indicatore</i> | Se questa proprietà è impostata su <i>T</i> , i valori null (valori non definiti, visualizzati come \$null\$ nel software) vengono considerati valori mancanti. Formato di utilizzo: NODE.null_missing.FIELDNAME |
| whitespace_missing | <i>indicatore</i> | Quando questa proprietà è impostata su <i>T</i> , i valori contenenti solo uno spazio vuoto (spazi, tabulazioni e nuove righe) vengono considerati valori mancanti. Formato di utilizzo: NODE.whitespace_missing.FIELDNAME |
| descrizione | <i>stringa</i> | Utilizzata per specificare una descrizione o etichetta di campo. |

Tabella 40. Proprietà comuni dei nodi origine (Continua).

| Nome proprietà | Data type | Descrizione proprietà |
|-----------------------|---|---|
| default_include | <i>indicatore</i> | Proprietà basata su chiavi utilizzata per specificare se il comportamento di default determina il passaggio o il filtro di campi: NODE.default_include Esempio: set mynode:filternode.default_include = false |
| include | <i>indicatore</i> | Proprietà basata su chiavi utilizzata per determinare se i singoli campi vengono inclusi o filtrati: NODE.include.FIELDNAME. |
| new_name | <i>stringa</i> | |
| measure_type | Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL | Questa proprietà basata su chiavi è simile a type in quanto può essere utilizzata per definire la misurazione associata al campo. La differenza consiste nel fatto che, negli script Python, alla funzione setter può essere passato anche uno dei valori MeasureType mentre getter viene restituito sempre sui valori MeasureType. |
| collection_measure | Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS | Per i campi di raccolta (elenchi con profondità uguale a 0), questa proprietà basata su chiavi definisce il tipo di misurazione associato ai valori sottostanti. |
| geo_type | Point MultiPoint LineString MultiLineString Polygon MultiPolygon | Per i campi geospaziali, questa proprietà basata su chiavi definisce il tipo di oggetto geospaziale rappresentato da questo campo. Questo valore deve essere coerente con la profondità di elenco dei valori. |
| has_coordinate_system | <i>booleano</i> | Per i campi geospaziali, questa proprietà definisce se questo campo dispone di un sistema di coordinate |
| coordinate_system | <i>stringa</i> | Per i campi geospaziali, questa proprietà basata su chiavi definisce il sistema di coordinate per questo campo. |

Tabella 40. Proprietà comuni dei nodi origine (Continua).

| Nome proprietà | Data type | Descrizione proprietà |
|--------------------------|--|--|
| custom_storage_type | Unknown / MeasureType.UNKNOWN String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP List / MeasureType.LIST | Questa proprietà basata su chiavi è simile a custom_storage in quanto può essere utilizzata per definire l'archiviazione di sostituzione per il campo. La differenza consiste nel fatto che, negli script Python, alla funzione setter può essere passato anche uno dei valori StorageType mentre getter viene restituito sempre sui valori StorageType. |
| custom_list_storage_type | String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP | Per i campi di elenco, questa proprietà basata su chiavi specifica il tipo di archiviazione dei valori sottostanti. |
| custom_list_depth | <i>integer</i> | Per i campi di elenco, questa proprietà basata su chiavi specifica la profondità del campo |

Proprietà asimport

L'origine di Analytic Server consente di eseguire un flusso su HDFS (Hadoop Distributed File System).

Esempio

```
node = stream.create("asimport", "My node")
node.setPropertyValue("data_source", "Drug1n")
```

Tabella 41. Proprietà asimport.

| Proprietà asimport | Data type | Descrizione proprietà |
|--------------------|----------------|----------------------------|
| data_source | <i>stringa</i> | Il nome dell'origine dati. |

Proprietà del nodo cognosimport



Il nodo origine IBM Cognos BI importa i dati dai database Cognos BI.

Esempio

```
node = stream.create("cognosimport", "My node")
node.setPropertyValue("cognos_connection", ["http://mycogsrv1:9300/p2pd/servlet/dispatch",
True, "", "", ""])
node.setPropertyValue("cognos_package_name", "/Public Folders/GOSALES")
node.setPropertyValue("cognos_items", ["[GreatOutdoors].[BRANCH].[BRANCH_CODE]", "[GreatOutdoors].[BRANCH].[COUNTRY_CODE]"])
```

Tabella 42. Proprietà del nodo cognosimport.

| Proprietà del nodo cognosimport | Data type | Descrizione proprietà |
|---------------------------------|---|---|
| mode | Data Report | Specifica se importare i dati (default) o i report Cognos BI. |
| cognos_connection | <code>["string",flag,"string", "string" ,"string"]</code> | <p>Una proprietà elenco contenente i dettagli di connessione per il server Cognos. Il formato è: ["Cognos_server_URL", login_mode, "namespace", "username", "password"]</p> <p>dove: Cognos_server_URL è l'URL del server Cognos contenente la sorgente. login_mode indica se viene utilizzato un accesso anonimo e può essere true o false; se impostato su true, i seguenti campi devono essere impostati su "". namespace specifica il provider di protezione per l'autenticazione utilizzato per accedere al server. username e password sono i dati utilizzati per accedere al server Cognos.</p> <p>Invece di login_mode, sono disponibili le seguenti modalità:</p> <ul style="list-style-type: none"> • anonymousMode. Ad esempio: ['Cognos_server_url', 'anonymousMode', "namespace", "username", "password"] • credentialMode. Ad esempio: ['Cognos_server_url', 'credentialMode', "namespace", "username", "password"] • storedCredentialMode. Ad esempio: ['Cognos_server_url', 'storedCredentialMode', "stored_credential_name"] <p>Dove stored_credential_name rappresenta il nome delle credenziali Cognos nel repository.</p> |
| nome_package_cognos | stringa | <p>Il percorso e il nome del package Cognos da cui importare gli oggetti di dati, per esempio: /Public Folders/GOSALES Nota: sono valide solo le barre (/).</p> |
| cognos_items | <code>["campo","campo", ... ,"campo"]</code> | <p>Il nome di uno o più oggetti di dati da importare. Il formato di <i>campo</i> è <code>[spaziodeinomi].[oggetto_query].[elemento_query]</code></p> |
| cognos_filters | campo | <p>Il nome di uno o più filtri da applicare prima di importare i dati.</p> |

Tabella 42. Proprietà del nodo cognosimport (Continua).

| Proprietà del nodo cognosimport | Data type | Descrizione proprietà |
|---------------------------------|-----------|---|
| cognos_data_parameters | elenco | Valori per i parametri di prompt per i dati. Le coppie nome-valore sono racchiuse tra parentesi quadrate, più coppie sono separate tra loro da virgole e l'intera stringa è racchiusa tra parentesi quadre. Formato: [["param1", "value"],...["paramN", "value"]] |
| cognos_report_directory | campo | Il percorso Cognos di una cartella o un package da cui si importano i report, per esempio: /Public Folders/GOSALES Nota: sono valide solo le barre (/). |
| cognos_report_name | campo | Il percorso ed il nome nella posizione del report da importare. |
| cognos_report_parameters | elenco | Valori per i parametri di report. Le coppie nome-valore sono racchiuse tra parentesi quadrate, più coppie sono separate tra loro da virgole e l'intera stringa è racchiusa tra parentesi quadre. Formato: [["param1", "value"],...["paramN", "value"]] |

Proprietà databasenode



Il nodo Database può essere utilizzato per importare dati da numerosi altri pacchetti che utilizzano ODBC (Open Database Connectivity), tra cui Microsoft SQL Server, DB2, Oracle e altri.

Esempio

```
import modeler.api
stream = modeler.script.stream()
nnode = stream.create("database", "My node")
node.setPropertyValue("mode", "Table")
node.setPropertyValue("query", "SELECT * FROM drug1n")
node.setPropertyValue("datasource", "Drug1n_db")
node.setPropertyValue("username", "spss")
node.setPropertyValue("password", "spss")
node.setPropertyValue("tablename", ".Drug1n")
```

Tabella 43. Proprietà databasenode.

| proprietà databasenode | Data type | Descrizione proprietà |
|------------------------|----------------|--|
| mode | Table Query | Specificare <i>Table</i> per connettersi a una tabella di database tramite i controlli della finestra di dialogo, oppure <i>Query</i> per eseguire una query del database selezionato tramite SQL. |
| datasource | stringa | Nome database (vedere anche la nota riportata sotto). |

Tabella 43. Proprietà databasenode (Continua).

| proprietà databasenode | Data type | Descrizione proprietà |
|------------------------|-------------------------------|---|
| username | stringa | Dettagli connessione database (vedere anche la nota riportata sotto). |
| password | stringa | |
| credential | stringa | Nome della credenziale memorizzata in IBM SPSS Collaboration and Deployment Services. Può essere utilizzato in sostituzione delle proprietà username e password. Il nome utente e la password della credenziale devono corrispondere al nome utente e alla password richiesti per accedere al database. |
| use_credential | | Impostare su True o False. |
| epassword | stringa | Specifica una password codificata come alternativa all'hardcoding di una password in uno script. Per ulteriori informazioni, consultare l'argomento "Generazione di una password codificata" a pagina 51. Proprietà di sola lettura durante l'esecuzione. |
| tablename | stringa | Nome della tabella a cui si desidera accedere. |
| strip_spaces | None Left Right Both | Opzioni per scartare gli spazi iniziali e finali nelle stringhe. |
| use_quotes | AsNeeded Always Never | Specifica se si desidera che i nomi delle tabelle e delle colonne vengano racchiusi tra virgolette quando le query vengono inviate al database (per esempio nel caso in cui tali nomi contengano spazi o punteggiatura). |
| query | stringa | Specifica il codice SQL per la query che si desidera effettuare. |

Nota: Se il nome del database (nella proprietà datasource) contiene uno o più spazi, punti o caratteri di sottolineatura, è possibile utilizzare il formato "barra rovesciata virgolette" per considerarlo come stringa. Ad esempio: "{\db2v9.7.6_linux\}" o: "{\TDATA 131\}". Inoltre, racchiudere sempre i valori stringa datasource tra virgolette e parentesi graffe, come nell'esempio riportato di seguito: "{\SQL Server\",spssuser,abcd1234,false}".

Nota: Se il nome del database (nella proprietà datasource) contiene degli spazi, invece delle singole proprietà per datasource, username e password, è possibile utilizzare una singola proprietà datasource nel seguente formato:

Tabella 44. Proprietà databasenode specifiche dell'origine dati.

| proprietà databasenode | Data type | Descrizione proprietà |
|------------------------|-----------|---|
| datasource | stringa | <p>Formato: [database_name,username,password[,true false]]</p> <p>L'ultimo parametro è destinato all'uso con le password crittografate. Se è impostato su true, prima dell'uso la password verrà decrittografata.</p> |

Utilizzare questo formato anche se si sta modificando la sorgente dati; tuttavia, se si desidera soltanto modificare il nome utente o la password, è possibile utilizzare le proprietà username o password.

Proprietà datacollectionimportnode



Il nodo IBM SPSS Data Collection Importazione dati importa dati di indagine basati sul Modello dati di IBM SPSS Data Collection utilizzato dai prodotti di ricerca di mercato IBM Corp.. Per utilizzare questo nodo, è necessario che sia installata IBM SPSS Data Collection Data Library.

Figura 7. Nodo Importazione dati di Dimensions

Esempio

```
node = stream.create("datacollectionimport", "My node")
node.setPropertyValue("metadata_name", "mrQvDsc")
node.setPropertyValue("metadata_file", "C:/Program Files/IBM/SPSS/DataCollection/DDL/Data/
Quanvert/Museum/museum.pkd")
node.setPropertyValue("casedata_name", "mrQvDsc")
node.setPropertyValue("casedata_source_type", "File")
node.setPropertyValue("casedata_file", "C:/Program Files/IBM/SPSS/DataCollection/DDL/Data/
Quanvert/Museum/museum.pkd")
node.setPropertyValue("import_system_variables", "Common")
node.setPropertyValue("import_multi_response", "MultipleFlags")
```

Tabella 45. proprietà datacollectionimportnode.

| Proprietà datacollectionimportnode | Data type | Descrizione proprietà |
|------------------------------------|---|--|
| metadata_name | stringa | Nome dell'MDSC. Il valore speciale DimensionsMDD indica che deve essere utilizzato il documento metadati di IBM SPSS Data Collection standard. Gli altri valori possibili sono: mrADODsc mrI2dDsc mrLogDsc mrQdiDrsDsc mrQvDsc mrSampleReportingMDSC mrSavDsc mrSCDsc mrScriptMDSC Il valore speciale none indica che non è presente alcun MDSC. |
| metadata_file | stringa | Nome del file nel quale sono archiviati i metadati. |
| casedata_name | stringa | Nome del CDSC. I valori possibili sono: mrADODsc mrI2dDsc mrLogDsc mrPunchDSC mrQdiDrsDsc mrQvDsc mrRdbDsc2 mrSavDsc mrScDSC mrXmlDsc Il valore speciale none indica che non è presente alcun CDSC. |
| casedata_source_type | Unknown File Folder UDL DSN | Indica il tipo di sorgente del CDSC. |
| casedata_file | stringa | Se casedata_source_type è <i>File</i> , specifica il file contenente i dati del caso. |
| casedata_folder | stringa | Se casedata_source_type è <i>Folder</i> , specifica la cartella contenente i dati del caso. |
| casedata_udl_string | stringa | Se casedata_source_type è <i>UDL</i> , specifica la stringa di connessione OLD-DB della sorgente dati contenente i dati del caso. |
| casedata_dsn_string | stringa | Se casedata_source_type è <i>DSN</i> , specifica la stringa di connessione ODBC della sorgente dati. |
| casedata_project | stringa | Durante la lettura dei dati del caso da un database di IBM SPSS Data Collection, è possibile immettere il nome del progetto. Per tutti gli altri tipi di dati del caso, questa impostazione deve essere lasciata vuota. |

Tabella 45. proprietà `datacollectionimportnode` (Continua).

| Proprietà <code>datacollectionimportnode</code> | Data type | Descrizione proprietà |
|---|--------------------------|---|
| <code>version_import_mode</code> | All Latest Specify | Definisce quante versioni devono essere gestite. |
| <code>specific_version</code> | <i>stringa</i> | Se <code>version_import_mode</code> è <i>Specify</i> , definisce la versione dei dati del caso da importare. |
| <code>use_language</code> | <i>stringa</i> | Definisce se debbano essere utilizzate delle etichette di una lingua specifica. |
| <code>language</code> | <i>stringa</i> | Se <code>use_language</code> è impostata su vero, specifica il codice della lingua da utilizzare nell'importazione. Il codice della lingua deve essere uno di quelli disponibili nei dati del caso. |
| <code>use_context</code> | <i>stringa</i> | Definisce se debba essere importato un contesto specifico. I contesti vengono utilizzati per variare la descrizione associata alle risposte. |
| <code>context</code> | <i>stringa</i> | Se <code>use_context</code> è impostato su vero, definisce il contesto da importare. Il contesto deve essere uno di quelli disponibili nei dati del caso. |
| <code>use_label_type</code> | <i>stringa</i> | Definisce se debba essere importato un tipo specifico di etichetta. |
| <code>label_type</code> | <i>stringa</i> | Se <code>use_label_type</code> è impostato su vero, definisce il tipo di etichetta da importare. Il tipo di etichetta deve essere uno di quelli disponibili nei dati del caso. |
| <code>user_id</code> | <i>stringa</i> | Per i database che richiedono un login esplicito, è possibile fornire un ID utente e una password per accedere alla sorgente dati. |
| <code>password</code> | <i>stringa</i> | |
| <code>import_system_variables</code> | Common None All | Specifica quali variabili di sistema vengono importate. |
| <code>import_codes_variables</code> | <i>indicatore</i> | |
| <code>import_sourcefile_variables</code> | <i>indicatore</i> | |
| <code>import_multi_response</code> | MultipleFlags Single | |

Proprietà `excelimportnode`



Il nodo Importazione da Excel importa i dati da Microsoft Excel nel formato file `.xlsx`. Non è richiesta alcuna sorgente dati ODBC.

Esempi

```
#Per utilizzare un intervallo denominato:
node = stream.create("excelimport", "My node")
node.setPropertyValue("excel_file_type", "Excel2007")
```

```
node.setPropertyValue("full_filename", "C:/drug.xlsx")
node.setPropertyValue("use_named_range", True)
node.setPropertyValue("named_range", "DRUG")
node.setPropertyValue("read_field_names", True)
```

```
#Per utilizzare un intervallo esplicito:
node = stream.create("excelimport", "My node")
node.setPropertyValue("excel_file_type", "Excel2007")
node.setPropertyValue("full_filename", "C:/drug.xlsx")
node.setPropertyValue("worksheet_mode", "Name")
node.setPropertyValue("worksheet_name", "Drug")
node.setPropertyValue("explicit_range_start", "A1")
node.setPropertyValue("explicit_range_end", "F300")
```

Tabella 46. proprietà excelimportnode.

| proprietà excelimportnode | Data type | Descrizione proprietà |
|---------------------------|--------------------------------|---|
| excel_file_type | Excel2007 | |
| full_filename | stringa | Il nome del file completo compreso il percorso. |
| use_named_range | Booleano | Indica se utilizzare o meno un intervallo denominato. Se vero, la proprietà named_range viene utilizzata per specificare l'intervallo da leggere, mentre le altre impostazioni relative al foglio di lavoro e all'intervallo dati vengono ignorate. |
| named_range | stringa | |
| worksheet_mode | Index Nome | Specifica se il foglio di lavoro è definito in base all'indice o al nome. |
| worksheet_index | numero intero | Indice dei fogli di lavoro da leggere che inizia con 0 per il primo foglio di lavoro, 1 per il secondo e così via. |
| worksheet_name | stringa | Nome del foglio di lavoro da leggere. |
| data_range_mode | FirstNonBlank ExplicitRange | Specifica come viene determinato l'intervallo. |
| blank_rows | StopReading ReturnBlankRows | Se data_range_mode è FirstNonBlank, specifica come vanno gestite le righe vuote. |
| explicit_range_start | stringa | Se data_range_mode è ExplicitRange, specifica il punto di partenza dell'intervallo da leggere. |
| explicit_range_end | stringa | |
| read_field_names | Booleano | Specifica se la prima riga dell'intervallo specificato deve essere utilizzata come nome di campo (colonna). |

Proprietà evimportnode



Il nodo Enterprise View crea una connessione a un IBM SPSS Collaboration and Deployment Services Repository, consentendo di leggere i dati di Enterprise View in un flusso e di assemblare un modello in uno scenario accessibile da altri utenti tramite il repository.

Nota: Il nodo Enterprise View è stato sostituito in SPSS Modeler 16.0 dal nodo Vista dati. Per i flussi salvati nelle release precedenti, il nodo Enterprise View è ancora supportato. Tuttavia, durante l'aggiornamento o la creazione di nuovi flussi, si consiglia di utilizzare il nodo Vista dati.

Esempio

```
node = stream.create("evimport", "My node")
node.setPropertyValue("connection", ["Training data", "/Application views/Marketing", "LATEST",
"Analytic", "/Data Providers/Marketing"])
node.setPropertyValue("tablename", "cust1")
```

Tabella 47. proprietà evimportnode.

| proprietà evimportnode | Data type | Descrizione proprietà |
|------------------------|-----------|---|
| connection | elenco | Proprietà strutturata--elenco dei parametri che costituiscono una connessione Enterprise View. Formato di utilizzo: evimportnode.connection = [description,app_view_path, app_view_version_label, environment,DPD_path] |
| tablename | stringa | Il nome di una tabella nella Visualizzazione applicazione. |

Proprietà fixedfilenode



Il nodo Testo fisso importa dati da file di testo a campi fissi, ovvero file i cui campi non vengono delimitati ma iniziano nella stessa posizione e hanno una lunghezza fissa. Nel formato a campi fissi vengono in genere archiviati dati di versioni precedenti o generati dalla macchina.

Esempio

```
node = stream.create("fixedfile", "My node")
node.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node.setPropertyValue("record_len", 32)
node.setPropertyValue("skip_header", 1)
node.setPropertyValue("fields", [{"Age", 1, 3}, {"Sex", 5, 7}, {"BP", 9, 10}, {"Cholesterol", 12, 22}, {"Na", 24, 25}, {"K", 27, 27}, {"Drug", 29, 32}])
node.setPropertyValue("decimal_symbol", "Period")
node.setPropertyValue("lines_to_scan", 30)
```

Tabella 48. proprietà fixedfilenode.

| proprietà fixedfilenode | Data type | Descrizione proprietà |
|-------------------------|----------------------------|---|
| record_len | numero | Specifica il numero di caratteri in ogni record. |
| line_oriented | indicatore | Ignora il carattere di nuova riga alla fine di ogni record. |
| decimal_symbol | Default Comma Period | Tipo di separatore decimale utilizzato nella sorgente dati. |
| skip_header | numero | Specifica il numero di righe che si desidera ignorare all'inizio del primo record. Utile per ignorare le intestazioni di colonna. |
| auto_recognize_datetime | indicatore | Specifica se data e ora vengono identificate automaticamente nei dati di origine. |
| lines_to_scan | numero | |
| campi | elenco | Proprietà strutturata. |

Tabella 48. proprietà fixedfilenode (Continua).

| proprietà fixedfilenode | Data type | Descrizione proprietà |
|--------------------------|---|--|
| full_filename | stringa | Nome completo del file da leggere, inclusa la directory. |
| strip_spaces | None Left Right Both | Scarta gli spazi iniziali e finali nelle stringhe durante l'importazione. |
| invalid_char_mode | Discard Replace | Rimuove i caratteri non validi (null, 0 o qualsiasi carattere non esistente nella codifica corrente) dall'input dei dati o sostituisce i caratteri non validi con il simbolo a un carattere specificato. |
| invalid_char_replacement | stringa | |
| use_custom_values | indicatore | |
| custom_storage | Unknown String Integer Real Time Date Timestamp | |
| custom_date_format | "DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "GG-MM-AA" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY | Questa proprietà è applicabile solo se è stata specificata un'archiviazione personalizzata. |

Tabella 48. proprietà *fixedfilenode* (Continua).

| proprietà <i>fixedfilenode</i> | Data type | Descrizione proprietà |
|--------------------------------|---|---|
| custom_time_format | "HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM." "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S" | Questa proprietà è applicabile solo se è stata specificata un'archiviazione personalizzata. |
| custom_decimal_symbol | <i>campo</i> | Valida solo se è stata specificata un'archiviazione personalizzata. |
| encoding | StreamDefault SystemDefault "UTF-8" | Specifica il metodo di codifica del testo. |

Proprietà del nodo *gsdata_import*



Utilizzare il nodo origine geospaziale per inserire i dati spaziali o della mappa nella propria sessione di data mining.

Tabella 49. proprietà del nodo *gsdata_import*

| Proprietà del nodo <i>gsdata_import</i> | Data type | Descrizione proprietà |
|---|----------------|---|
| full_filename | <i>stringa</i> | Immettere il percorso del file .shp che si desidera caricare. |
| map_service_URL | <i>stringa</i> | Immettere l'URL del servizio mappa a cui connettersi. |
| map_name | <i>stringa</i> | Solo se si utilizza map_service_URL; questo contiene la struttura della cartella di livello superiore del servizio mappa. |

Proprietà *sasimportnode*



Il nodo File SAS importa dati SAS in IBM SPSS Modeler.

Esempio

```
node = stream.create("sasimport", "My node")
node.setPropertyValue("format", "Windows")
node.setPropertyValue("full_filename", "C:/data/retail.sas7bdat")
```

```

node.setPropertyValue("member_name", "Test")
node.setPropertyValue("read_formats", False)
node.setPropertyValue("full_format_filename", "Test")
node.setPropertyValue("import_names", True)

```

Tabella 50. proprietà sasimportnode.

| proprietà sasimportnode | Data type | Descrizione proprietà |
|-------------------------|--|--|
| format | Windows UNIX Transport SAS7 SAS8 SAS9 | Formato del file di importazione. |
| full_filename | stringa | Il nome del file completo che è stato specificato e il relativo percorso. |
| member_name | stringa | Specifica il membro da importare dal file di trasporto SAS specificato. |
| read_formats | indicatore | Legge i formati dei dati (quali etichette di variabile) dal file del formato specificato. |
| full_format_filename | stringa | |
| import_names | NamesAndLabels LabelsasNames | Specifica il metodo per la mappatura di nomi ed etichette di variabili durante l'importazione. |

Proprietà simgennode



Il nodo Genera simulazione fornisce un modo semplice per generare dati simulati — partendo da zero utilizzando distribuzioni statistiche specificate dall'utente oppure automaticamente utilizzando le distribuzioni ottenute dall'esecuzione di un nodo Adattamento simulazione su dati cronologici esistenti. Ciò è utile quando si decide di valutare il risultato di un modello predittivo in presenza di incertezza negli input del modello.

Tabella 51. Proprietà simgennode.

| Proprietà simgennode | Data type | Descrizione proprietà |
|------------------------|-----------------------|--|
| fields | Proprietà strutturata | Vedere l'esempio |
| correlations | Proprietà strutturata | Vedere l'esempio |
| keep_min_max_setting | booleano | |
| refit_correlations | booleano | |
| max_cases | integer | Il valore minimo è 1000, il valore massimo è 2.147.483.647 |
| create_iteration_field | booleano | |
| iteration_field_name | stringa | |
| replicate_results | booleano | |
| random_seed | integer | |
| parameter_xml | stringa | Restituisce il codice Xml del parametro come stringa |

Esempio di fields

Questo è un parametro di slot strutturato con la seguente sintassi:

```
simgennode.setPropertyValue("fields", [
  [field1, storage, locked, [distribution1], min, max],
  [field2, storage, locked, [distribution2], min, max],
  [field3, storage, locked, [distribution3], min, max]
])
```

distribution è una dichiarazione del nome della distribuzione seguito da un elenco contenente le coppie di nomi di attributo e valori. Ciascuna distribuzione è definita nel seguente modo:

```
[distributionname, [[par1], [par2], [par3]]]
```

```
simgennode = modeler.script.stream().createAt("simgen", u"Sim Gen", 726, 322)
simgennode.setPropertyValue("fields", [[["Age", "integer", False, ["Uniform", [{"min", "1"}, {"max", "2"}]]], ["", ""]])
```

Ad esempio, per creare un nodo che genera un solo campo con una distribuzione Binomial è possibile utilizzare il seguente script:

```
simgen_node1 = modeler.script.stream().createAt("simgen", u"Sim Gen", 200, 200)
simgen_node1.setPropertyValue("fields", [[["Education", "Real", False, ["Binomial", [{"n", 32}, {"prob", 0.7}]]], ["", ""]])
```

La distribuzione Binomial utilizza due parametri: n e prob. Poiché Binomial non supporta i valori minimo e massimo, questi vengono forniti da una stringa vuota.

Nota: Se non è possibile impostare direttamente distribution, lo si usa insieme alla proprietà fields.

I seguenti esempi mostrano tutti i tipi di distribuzione possibili. Tenere presente che la soglia viene inserita come thresh in NegativeBinomialFailures e NegativeBinomialTrial.

```
stream = modeler.script.stream()

simgennode = stream.createAt("simgen", u"Sim Gen", 200, 200)

beta_dist = ["Field1", "Real", False, ["Beta", [{"shape1", "1"}, {"shape2", "2"}]], "", ""]
binomial_dist = ["Field2", "Real", False, ["Binomial", [{"n", "1"}, {"prob", "1"}]], "", ""]
categorical_dist = ["Field3", "String", False, ["Categorical", [{"A", "0.3"}, {"B", "0.5"}, {"C", "0.2"}]], "", ""]
dice_dist = ["Field4", "Real", False, ["Dice", [{"1", "0.5"}, {"2", "0.5"}]], "", ""]
exponential_dist = ["Field5", "Real", False, ["Exponential", [{"scale", "1"}]], "", ""]
fixed_dist = ["Field6", "Real", False, ["Fixed", [{"value", "1"}]], "", ""]
gamma_dist = ["Field7", "Real", False, ["Gamma", [{"scale", "1"}, {"shape", "1"}]], "", ""]
lognormal_dist = ["Field8", "Real", False, ["Lognormal", [{"a", "1"}, {"b", "1"}]], "", ""]
negbinomialfailures_dist = ["Field9", "Real", False, ["NegativeBinomialFailures", [{"prob", "0.5"}, {"thresh", "1"}]], "", ""]
negbinomialtrial_dist = ["Field10", "Real", False, ["NegativeBinomialTrials", [{"prob", "0.2"}, {"thresh", "1"}]], "", ""]
normal_dist = ["Field11", "Real", False, ["Normal", [{"mean", "1"}, {"stddev", "2"}]], "", ""]
poisson_dist = ["Field12", "Real", False, ["Poisson", [{"mean", "1"}]], "", ""]
range_dist = ["Field13", "Real", False, ["Range", [{"BEGIN", "1,3"}, {"END", "2,4"}, {"PROB", "[0.5],[0.5]"}]], "", ""]
triangular_dist = ["Field14", "Real", False, ["Triangular", [{"min", "0"}, {"max", "1"}, {"mode", "1"}]], "", ""]
uniform_dist = ["Field15", "Real", False, ["Uniform", [{"min", "1"}, {"max", "2"}]], "", ""]
weibull_dist = ["Field16", "Real", False, ["Weibull", [{"a", "0"}, {"b", "1"}, {"c", "1"}]], "", ""]

simgennode.setPropertyValue("fields", [
  beta_dist, \
  binomial_dist, \
  categorical_dist, \
  dice_dist, \
  exponential_dist, \
  fixed_dist, \
  gamma_dist, \
  lognormal_dist, \
  negbinomialfailures_dist, \
  negbinomialtrial_dist, \
  normal_dist, \
  poisson_dist, \
  range_dist, \
  triangular_dist, \
  uniform_dist, \
  weibull_dist
])
```

Esempio di correlations

Questo è un parametro di slot strutturato con la seguente sintassi:

```
simgennode.setPropertyValue("correlations", [  
    [field1, field2, correlation],  
    [field1, field3, correlation],  
    [field2, field3, correlation]  
])
```

La correlazione può essere qualsiasi numero compreso tra +1 e -1. È possibile specificare tutte le correlazioni desiderate. Tutte le correlazioni non specificate vengono impostate su zero. Se alcuni campi sono sconosciuti, il valore di correlazione deve essere impostato sulla matrice (o tabella) di correlazione e viene visualizzato in rosso. Quando sono presenti campi sconosciuti, non è possibile eseguire il nodo.

Proprietà statisticsimportnode



Il nodo File IBM SPSS Statistics legge i dati dal formato di file *.sav* utilizzato da IBM SPSS Statistics, nonché da file della cache salvati in IBM SPSS Modeler, che utilizzano lo stesso formato.

Le proprietà di questo nodo sono descritte in “Proprietà statisticsimportnode” a pagina 311.

Proprietà del nodo tm1import



Il nodo origine IBM Cognos TM1 importa i dati dai database Cognos TM1.

Tabella 52. proprietà del nodo tm1import.

| Proprietà del nodo tm1import | Data type | Descrizione proprietà |
|------------------------------|---|---|
| pm_host | <i>stringa</i> | Nota: Solo per le versioni 16.0 e 17.0 Il nome host. Ad esempio: TM1_import.setPropertyValue("pm_host", 'http://9.191.86.82:9510/pmhub/pm') |
| tm1_connection | <i>["campo", "campo", ... ,"campo"]</i> | Nota: Solo per le versioni 16.0 e 17.0 Una proprietà elenco che contiene i dettagli di connessione per il server TM1. Il formato è: ["TM1_Server_Name", "tm1_username", "tm1_ password"] Ad esempio: TM1_import.setPropertyValue("tm1_connection", ['Planning Sample', "admin", "apple"]) |
| admin_host | <i>stringa</i> | Nota: Solo a partire dalla versione 17.1 L'URL per il nome host dell'API REST. |
| server_name | <i>stringa</i> | Nota: Solo a partire dalla versione 17.1 Il nome del server TM1 selezionato da admin_host. |

Tabella 52. proprietà del nodo tm1import (Continua).

| Proprietà del nodo tm1import | Data type | Descrizione proprietà |
|------------------------------|-------------------|--|
| selected_view | ["campo" "campo"] | Una proprietà elenco che contiene i dettagli del cubo TM1 selezionato ed il nome della vista cubo da cui i dati verranno importati in SPSS. Ad esempio: TM1_import.setPropertyValue("selected_view", ['plan_BudgetPlan', 'Goal Input']) |

Proprietà userinputnode



Il nodo Input utente consente di ottenere in modo semplice dati sintetici creandoli ex-novo oppure modificando dati esistenti. È utile, per esempio, quando si desidera creare un insieme di dati di test per la modellazione.

Esempio

```
node = stream.create("userinput", "My node")
node.setPropertyValue("names", ["test1", "test2"])
node.setKeyedPropertyValue("data", "test1", "2, 4, 8")
node.setKeyedPropertyValue("custom_storage", "test1", "Integer")
node.setPropertyValue("data_mode", "Ordered")
```

Tabella 53. proprietà userinputnode.

| proprietà userinputnode | Data type | Descrizione proprietà |
|-------------------------|---|---|
| data | | |
| names | | Slot strutturato che imposta o restituisce un elenco di nomi di campi generati dal nodo. |
| custom_storage | Unknown String Integer Real Time Date Timestamp | Slot basato su chiavi che imposta o restituisce l'archiviazione per un campo. |
| data_mode | Combined Ordered | Se è specificato Combined, i record vengono generati per ogni combinazione di valori di insieme e valori minimi e massimi. Il numero di record generati è uguale al prodotto del numero di valori in ogni campo. Se è specificato Ordered, viene preso un solo valore da ogni colonna per ogni record allo scopo di generare una riga di dati. Il numero di record generati è uguale al numero di valori più grande associato a un campo. Tutti i campi con valori di dati inferiori verranno riempiti con valori null. |
| values | | Nota: Questa proprietà è obsoleta, non viene più utilizzata ed è stata sostituita dalla proprietà userinputnode.data. |

Proprietà variabilefilenode



Il nodo Testo variabile legge dati da file di testo a campi liberi, ovvero file i cui record contengono un numero costante di campi e un numero variabile di caratteri. Questo nodo può essere utilizzato per file con testo di intestazione a lunghezza fissa e alcuni tipi di annotazioni.

Esempio

```
node = stream.create("variablefile", "My node")
node.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node.setPropertyValue("read_field_names", True)
node.setPropertyValue("delimit_other", True)
node.setPropertyValue("other", ",")
node.setPropertyValue("quotes_1", "Discard")
node.setPropertyValue("decimal_symbol", "Comma")
node.setPropertyValue("invalid_char_mode", "Replace")
node.setPropertyValue("invalid_char_replacement", "|")
node.setKeyedPropertyValue("use_custom_values", "Age", True)
node.setKeyedPropertyValue("direction", "Age", "Input")
node.setKeyedPropertyValue("type", "Age", "Range")
node.setKeyedPropertyValue("values", "Age", [1, 100])
```

Tabella 54. proprietà variabilefilenode.

| proprietà variabilefilenode | Data type | Descrizione proprietà |
|-----------------------------|----------------------------|--|
| skip_header | numero | Specifica il numero di caratteri che si desidera ignorare all'inizio del primo record. |
| num_fields_auto | indicatore | Determina automaticamente il numero di campi in ogni record. I record devono terminare con un carattere di nuova riga. |
| num_fields | numero | Specifica manualmente il numero di campi in ogni record. |
| delimit_space | indicatore | Specifica il carattere utilizzato per delimitare i delimitatori di campo nel file. |
| delimit_tab | indicatore | |
| delimit_new_line | indicatore | |
| delimit_non_printing | indicatore | |
| delimit_comma | indicatore | Nei casi in cui la virgola è sia il delimitatore di campo che il separatore decimale dei flussi, impostare <code>delimit_other</code> su <code>true</code> e specificare una virgola come delimitatore utilizzando la proprietà <code>other</code> . |
| delimit_other | indicatore | Consente di specificare un delimitatore personalizzato utilizzando la proprietà <code>other</code> . |
| other | stringa | Specifica il delimitatore utilizzato quando <code>delimit_other</code> è <code>vero</code> . |
| decimal_symbol | Default Comma Period | Specifica il separatore decimale utilizzato nella sorgente dati. |
| multi_blank | indicatore | Considera più caratteri di delimitazione vuoti adiacenti come un delimitatore singolo. |
| read_field_names | indicatore | Considera la prima riga del file di dati come etichette per la colonna. |

Tabella 54. proprietà variablefilenode (Continua).

| proprietà variablefilenode | Data type | Descrizione proprietà |
|----------------------------|---|--|
| strip_spaces | None Left Right Both | Scarta gli spazi iniziali e finali nelle stringhe durante l'importazione. |
| invalid_char_mode | Discard Replace | Rimuove i caratteri non validi (null, 0 o qualsiasi carattere non esistente nella codifica corrente) dall'input dei dati o sostituisce i caratteri non validi con il simbolo a un carattere specificato. |
| invalid_char_replacement | <i>stringa</i> | |
| break_case_by_newline | <i>indicatore</i> | Specifica che il delimitatore di riga è un carattere di nuova riga. |
| lines_to_scan | <i>numero</i> | Indica il numero di righe da esaminare per i tipi di dati specificati. |
| auto_recognize_datetime | <i>indicatore</i> | Specifica se data e ora vengono identificate automaticamente nei dati di origine. |
| quotes_1 | Discard PairAndDiscard IncludeAsText | Specifica il trattamento delle virgolette singole durante l'importazione. |
| quotes_2 | Discard PairAndDiscard IncludeAsText | Specifica il trattamento delle virgolette durante l'importazione. |
| full_filename | <i>stringa</i> | Nome completo del file da leggere, inclusa la directory. |
| use_custom_values | <i>indicatore</i> | |
| custom_storage | Unknown String Integer Real Time Date Timestamp | |

Tabella 54. proprietà variablefilenode (Continua).

| proprietà variablefilenode | Data type | Descrizione proprietà |
|----------------------------|---|---|
| custom_date_format | "DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "GG-MM-AA" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY | Valida solo se è stata specificata un'archiviazione personalizzata. |
| custom_time_format | "HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM." "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S" | Valida solo se è stata specificata un'archiviazione personalizzata. |
| custom_decimal_symbol | campo | Valida solo se è stata specificata un'archiviazione personalizzata. |
| encoding | StreamDefault SystemDefault "UTF-8" | Specifica il metodo di codifica del testo. |

Proprietà xmlimportnode



Il nodo origine XML importa i dati in formato XML nel flusso. È possibile importare un singolo file oppure tutti i file in una directory. Come opzione, è possibile specificare un file schema da cui leggere la struttura XML.

Esempio

```
node = stream.create("xmlimport", "My node")
node.setPropertyValue("full_filename", "c:/import/ebooks.xml")
node.setPropertyValue("records", "/author/name")
```

Tabella 55. proprietà xmlimportnode.

| proprietà xmlimportnode | Data type | Descrizione proprietà |
|-------------------------|---------------------|--|
| read | single directory | Legge un singolo file di dati (default), oppure tutti i file XML in una directory. |
| recurse | <i>indicatore</i> | Specifica se leggere anche i file XML da tutte le sottodirectory della directory specificata. |
| full_filename | <i>stringa</i> | (obbligatorio) Percorso e nome file completi del file XML da importare (se read = single). |
| directory_name | <i>stringa</i> | (obbligatorio) Percorso completo e nome della directory dalla quale importare i file XML (se read = directory). |
| full_schema_filename | <i>stringa</i> | Percorso e nome file completi del file XSD o DTD dal quale leggere la struttura XML. Se si omette questo parametro, la struttura viene letta dal file di origine XML. |
| records | <i>stringa</i> | Espressione XPath (ad esempio, /autore/nome) che indica i limiti del record. Ogni volta che si incontra questo elemento nel file di origine, viene creato un nuovo record. |
| mode | read specify | Leggere tutti i dati (default), oppure specificare gli elementi da leggere. |
| fields | | Elenco di voci (elementi e attributi) da importare. Ogni voce dell'elenco è un'espressione XPath. |

Proprietà dataviewimport



Il nodo Vista dati importa i dati della Vista dati in IBM SPSS Modeler.

Esempio

```
stream = modeler.script.stream()

dnode = stream.createAt("dataviewimport", "Data View", 96, 96)
dnode.setPropertyValue("analytic_data_source",
["", "/folder/adv", "LATEST"])
dnode.setPropertyValue("table_name", ["", "com.ibm.spss.Table"])
dnode.setPropertyValue("data_access_plan",
```

```

["", "DataAccessPlan"])
dnode.setPropertyValue("optional_attributes",
["", "NewDerivedAttribute"])
dnode.setPropertyValue("include_xml", True)
dnode.setPropertyValue("include_xml_field", "xml_data")

```

Tabella 56. Proprietà dataviewimport

| Proprietà dataviewimport | Data type | Descrizione proprietà |
|--------------------------|-----------|--|
| analytic_data_source | stringa | L'oggetto Vista dati analitici archiviato in IBM SPSS Collaboration and Deployment Services. Il nome del percorso e l'etichetta della versione per la versione da utilizzare. ["Object ID", "Full path", "Version"] |
| table_name | stringa | La tabella della vista dati utilizzata nella Vista dati analitici. Il nome della tabella essere qualificato con il package. È possibile ottenere il package esportando il BOM dal client IBM SPSS Collaboration and Deployment Services Deployment Manager e ricercando nel file default.bom nell'archivio zip esportato. Il nome del package deve essere sempre uguale, a meno che il BOM non sia stato importato da IBM Operational Decision Management (iLOG). ["Object ID", "Name"] |
| data_access_plan | stringa | Il piano di accesso dati utilizzato per fornire i dati per la Vista dati analitici. ["Object ID", "Name"] |
| optional_attributes | stringa | Un elenco degli attributi derivati da includere. [["ID1", "Name1"], ["ID2", "Name2"]] |
| include_xml | booleano | True se deve essere incluso un campo con dati di istanza XOM. A meno che non vengano utilizzati nodi IBM Analytical Decision Management iLOG, l'impostazione consigliata è false. L'attivazione di questa proprietà può aggiungere un'elevata quantità di elaborazione supplementare. |
| include_xml_field | stringa | Il nome del campo da aggiungere quando include_xml è impostato su true. |

Capitolo 10. Proprietà dei nodi Operazioni su record

Proprietà appendnode



Il nodo Accodamento concatena insieme di record. Può essere utilizzato per combinare insieme di dati con strutture simili ma dati diversi.

Esempio

```
node = stream.create("append", "My node")
node.setPropertyValue("match_by", "Name")
node.setPropertyValue("match_case", True)
node.setPropertyValue("include_fields_from", "All")
node.setPropertyValue("create_tag_field", True)
node.setPropertyValue("tag_field_name", "Append_Flag")
```

Tabella 57. proprietà appendnode.

| Proprietà appendnode | Tipo di dati | Descrizione proprietà |
|----------------------|-------------------|--|
| match_by | Posizione Nome | È possibile accodare insieme di dati in base alla posizione dei campi nella sorgente dati principale o al nome dei campi nei dataset di input. |
| match_case | <i>indicatore</i> | Attiva la distinzione tra caratteri maiuscoli/minuscoli quando si esegue la corrispondenza tra i nomi dei campi. |
| include_fields_from | Main All | |
| create_tag_field | <i>indicatore</i> | |
| tag_field_name | <i>stringa</i> | |

Proprietà aggregatenode



Il nodo Aggregazione sostituisce una sequenza di record di input con record di output aggregati di riepilogo.

Esempio

```
node = stream.create("aggregate", "My node")
# dbnode is a configured database import node
stream.link(dbnode, node)
node.setPropertyValue("contiguous", True)
node.setPropertyValue("keys", ["Drug"])
node.setKeyedPropertyValue("aggregates", "Age", ["Sum", "Mean"])
node.setPropertyValue("inc_record_count", True)
node.setPropertyValue("count_field", "index")
node.setPropertyValue("extension", "Aggregated_")
node.setPropertyValue("add_as", "Prefix")
```

Tabella 58. Proprietà aggregatenode.

| Proprietà aggregatenode | Data type | Descrizione proprietà |
|-------------------------|-------------------|---|
| keys | <i>elenco</i> | Elenca i campi che possono essere utilizzati come chiavi per l'aggregazione. Per esempio, se Sesso e Regione sono i campi chiave disponibili, verrà generato un record aggregato per ogni combinazione univoca di M e F con le aree N e S (ovvero quattro combinazioni univoche). |
| contiguous | <i>indicatore</i> | Selezionare questa opzione se si sa che tutti i record con gli stessi valori chiave sono raggruppati insieme nell'input (per esempio, se l'input è ordinato in base ai campi chiave). In questo modo si migliorano le prestazioni. |
| aggregates | | Proprietà strutturata che elenca i campi numerici i cui valori verranno aggregati e le modalità di aggregazione selezionate. |
| aggregate_exprs | | Una proprietà le cui chiavi vengono utilizzate dal nome campo derivato con l'espressione aggregato per eseguirne il calcolo. Ad esempio: aggregatenode.setKeyedPropertyValue("aggregate_exprs", "Na_MAX", "MAX('Na')") |
| extension | <i>stringa</i> | Specifica un prefisso o suffisso per campi aggregati duplicati (vedere l'esempio seguente). |
| add_as | Suffix Prefix | |
| inc_record_count | <i>indicatore</i> | Crea un campo aggiuntivo che specifica quanti record di input sono stati aggregati per formare ogni record aggregato. |
| count_field | <i>stringa</i> | Specifica il nome del campo conteggio record. |
| allow_approximation | <i>Booleana</i> | Consente l'approssimazione delle statistiche di ordinamento quando viene eseguita l'aggregazione in Analytic Server |
| bin_count | <i>integer</i> | Specifica il numero di bin da utilizzare nell'approssimazione |

Proprietà balancenode



Il nodo bilanciamento corregge squilibri in un insieme di dati in modo che soddisfi una determinata condizione. La direttiva di bilanciamento regola la proporzione di record in cui una condizione è vera in base al fattore specificato.

Esempio

```
node = stream.create("balance", "My node")
node.setPropertyValue("training_data_only", True)
node.setPropertyValue("directives", [[1.3, "Age > 60"], [1.5, "Na > 0.5"]])
```

Tabella 59. proprietà balancenode.

| Proprietà balancenode | Data type | Descrizione proprietà |
|-----------------------|------------|--|
| directives | | Proprietà strutturata per il bilanciamento della proporzione dei valori del campo in base al numero specificato (vedere l'esempio seguente). |
| training_data_only | indicatore | Specifica che devono essere bilanciati solo i dati di addestramento. Se nel flusso non è presente alcun campo partizione, tale opzione viene ignorata. |

La proprietà di questo nodo utilizza il formato:

[[numero, stringa] \ [numero, stringa] \ ... [numero, stringa]].

Nota: se nell'espressione sono integrate delle stringhe (che utilizzano le virgolette), tali stringhe devono essere precedute dal carattere di escape " \ ". Il carattere " \ " è anche il carattere di continuazione della riga, che può essere utilizzato per allineare gli argomenti in modo da migliorare la leggibilità.

Proprietà derive_stbnode



Il nodo STB (Space-Time-Boxes) determina le STB dai campi latitudine, longitudine e data/ora. È possibile anche identificare frequenti STB (Space-Time-Boxes) come hangout.

Esempio

```
node = modeler.script.stream().createAt("derive_stb", "My node", 96, 96)
```

```
# Individual Records mode
node.setPropertyValue("mode", "IndividualRecords")
node.setPropertyValue("latitude_field", "Latitude")
node.setPropertyValue("longitude_field", "Longitude")
node.setPropertyValue("timestamp_field", "OccurredAt")
node.setPropertyValue("densities", ["STB_GH7_1HOURL", "STB_GH7_30MINS"])
node.setPropertyValue("add_extension_as", "Prefix")
node.setPropertyValue("name_extension", "stb_")
```

```
# Hangouts mode
node.setPropertyValue("mode", "Hangouts")
node.setPropertyValue("hangout_density", "STB_GH7_30MINS")
node.setPropertyValue("id_field", "Event")
node.setPropertyValue("qualifying_duration", "30MINUTES")
node.setPropertyValue("min_events", 4)
node.setPropertyValue("qualifying_pct", 65)
```

Tabella 60. Proprietà del nodo STB (Space-Time-Boxes)

| Proprietà derive_stbnode | Data type | Descrizione proprietà |
|--------------------------|-------------------------------|-----------------------|
| mode | IndividualRecords Hangouts | |
| latitude_field | campo | |
| longitude_field | campo | |
| timestamp_field | campo | |

Tabella 60. Proprietà del nodo STB (Space-Time-Boxes) (Continua)

| Proprietà derive_stbnode | Data type | Descrizione proprietà |
|--------------------------|--|--|
| hangout_density | densità | Una singola densità. Vedere densities per valori validi di densità. |
| densities | [densità,densità,..., densità] | Ogni densità è una stringa, ad esempio STB_GH8_1DAY. Nota: Vi sono limiti ai quali le densità sono valide. Per la geohash, possono essere utilizzati valori da GH1 a GH15. Per la parte temporale, possono essere utilizzati i seguenti valori: EVER 1YEAR 1MONTH 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2HOURS 1HOUR 30MINS 15MINS 10MINS 5MINS 2MINS 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SEC |
| id_field | campo | |
| qualifying_duration | 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2Hours 1HOUR 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS | Deve essere una stringa. |
| min_events | integer | Il valore del numero intero minimo valido è 2. |
| qualifying_pct | integer | Deve essere in un intervallo tra 1 e 100. |
| add_extension_as | Prefix Suffix | |
| name_extension | stringa | |

Proprietà distinctnode



Il nodo Elimina duplicati rimuove record duplicati passando il primo record distinto nel flusso di dati oppure scartando il primo record e passando nel flusso tutti i duplicati.

Esempio

```
node = stream.create("distinct", "My node")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("fields", ["Age" "Sex"])
node.setPropertyValue("keys_pre_sorted", True)
```

Tabella 61. proprietà distinctnode.

| Proprietà distinctnode | Data type | Descrizione proprietà |
|------------------------|--------------------|---|
| mode | Include Discard | È possibile includere il primo record distinto nel flusso di dati oppure scartare il primo record distinto e passare invece tutti i record duplicati al flusso di dati. |
| grouping_fields | <i>elenco</i> | Elenca i campi utilizzati per stabilire se i record sono identici. Nota: Questa proprietà è obsoleta a partire da IBM SPSS Modeler 16 in poi. |
| composite_value | Slot strutturato | Vedere l'esempio seguente. |
| composite_values | Slot strutturato | Vedere l'esempio seguente. |
| inc_record_count | <i>indicatore</i> | Crea un campo aggiuntivo che specifica quanti record di input sono stati aggregati per formare ogni record aggregato. |
| count_field | <i>stringa</i> | Specifica il nome del campo conteggio record. |
| sort_keys | Slot strutturato. | Nota: Questa proprietà è obsoleta a partire da IBM SPSS Modeler 16 in poi. |
| default_ascending | <i>indicatore</i> | |
| low_distinct_key_count | <i>indicatore</i> | Indica che i record e/o i valori univoci dei campi chiave sono in numero ridotto. |
| keys_pre_sorted | <i>indicatore</i> | Specifica che tutti i record con gli stessi valori chiave sono raggruppati insieme nell'input. |
| disable_sql_generation | <i>indicatore</i> | |

Esempio per la proprietà composite_value

Di seguito è riportato il formato generale della proprietà composite_value:

```
node.setKeyedPropertyValue("composite_value", FIELD, FILLOPTION)
```

FILLOPTION ha il formato [FillType, Option1, Option2, ...].

Esempi:

```
node.setKeyedPropertyValue("composite_value", "Age", ["First"])
node.setKeyedPropertyValue("composite_value", "Age", ["last"])
node.setKeyedPropertyValue("composite_value", "Age", ["Total"])
node.setKeyedPropertyValue("composite_value", "Age", ["Average"])
node.setKeyedPropertyValue("composite_value", "Age", ["Min"])
node.setKeyedPropertyValue("composite_value", "Age", ["Max"])
```

```
node.setKeyedPropertyValue("composite_value", "Date", ["Earliest"])
node.setKeyedPropertyValue("composite_value", "Date", ["Latest"])
node.setKeyedPropertyValue("composite_value", "Code", ["FirstAlpha"])
node.setKeyedPropertyValue("composite_value", "Code", ["LastAlpha"])
```

Le opzioni personalizzate richiedono più di un argomento; tali argomenti vengono aggiunti come elenco, ad esempio:

```
node.setKeyedPropertyValue("composite_value", "Name", ["MostFrequent", "FirstRecord"])
node.setKeyedPropertyValue("composite_value", "Date", ["LeastFrequent", "LastRecord"])
node.setKeyedPropertyValue("composite_value", "Pending", ["IncludesValue", "T", "F"])
node.setKeyedPropertyValue("composite_value", "Marital", ["FirstMatch", "Married", "Divorced", "Separated"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "Space"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "Comma"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "UnderScore"])
```

Esempio per la proprietà `composite_values`

Di seguito è riportato il formato generale della proprietà `composite_values`:

```
node.setPropertyValue("composite_values", [
    [FIELD1, [FILLOPTION1]],
    [FIELD2, [FILLOPTION2]],
    .
    .
])
```

Esempio:

```
node.setPropertyValue("composite_values", [
    ["Age", ["First"]],
    ["Name", ["MostFrequent", "First"]],
    ["Pending", ["IncludesValue", "T"]],
    ["Marital", ["FirstMatch", "Married", "Divorced", "Separated"]],
    ["Code", ["Concatenate", "Comma"]]
])
```

Proprietà `mergenode`



Il nodo Unione prende più record di input e crea un singolo record di output contenente tutti o alcuni campi di input. È utile per unire dati da sorgenti diverse, per esempio dati interni sui clienti e dati demografici acquistati.

Esempio

```
node = stream.create("merge", "My node")
# assume customerdata and salesdata are configured database import nodes
stream.link(customerdata, node)
stream.link(salesdata, node)
node.setPropertyValue("method", "Keys")
node.setPropertyValue("key_fields", ["id"])
node.setPropertyValue("common_keys", True)
node.setPropertyValue("join", "PartialOuter")
node.setKeyedPropertyValue("outer_join_tag", "2", True)
node.setKeyedPropertyValue("outer_join_tag", "4", True)
node.setPropertyValue("single_large_input", True)
node.setPropertyValue("single_large_input_tag", "2")
node.setPropertyValue("use_existing_sort_keys", True)
node.setPropertyValue("existing_sort_keys", [["id", "Ascending"]])
```

Tabella 62. proprietà mergenode.

| Proprietà mergenode | Data type | Descrizione proprietà |
|------------------------|--|--|
| method | Order Keys Condition Rankedcondition | Specifica se i record vengono uniti nell'ordine secondo cui sono elencati nei file di dati o se verranno utilizzati uno o più campi chiave per unire i record con lo stesso valore nei campi chiave, se i record verranno uniti nel caso venga soddisfatta una specifica condizione, oppure se è necessario unire ciascuna coppia di righe nel dataset primario ed in tutti i dataset secondari; utilizzare l'espressione di classificazione per ordinare le corrispondenze multiple dal valore più basso a quello più alto. |
| condition | stringa | Se method è impostato su Condition, specifica la condizione per includere o scartare i record. |
| key_fields | elenco | |
| common_keys | indicatore | |
| join | Inner FullOuter PartialOuter Anti | |
| outer_join_tag.n | indicatore | In questa proprietà, <i>n</i> è il nome del tag come viene visualizzato nella finestra di dialogo Seleziona insieme di dati. Si noti che è possibile specificare più nomi di tag, in quanto qualsiasi numero di insiemi di dati può contribuire con record incompleti. |
| single_large_input | indicatore | Specifica se verrà utilizzata l'ottimizzazione per avere un input relativamente grande rispetto agli altri input. |
| single_large_input_tag | stringa | Specifica il nome del tag come viene visualizzato nella finestra di dialogo Seleziona insieme di dati grande. Si noti che l'utilizzo di questa proprietà differisce leggermente rispetto alla proprietà outer_join_tag (flag e stringa) perché è possibile specificare solo un dataset di input. |
| use_existing_sort_keys | indicatore | Specifica se gli input sono già ordinati in base a uno o più campi chiave. |
| existing_sort_keys | [['stringa', 'Ascending'] \ \ ['stringa', 'Descending']] | Specifica i campi già ordinati e la direzione nella quale sono ordinati. |
| primary_dataset | stringa | Se method è Rankedcondition, selezionare il dataset principale nell'unione. Questo può essere considerato il lato sinistro di un'unione esterna. |
| add_tag_duplicate | Booleana | Se method è Rankedcondition, e questo è impostato su Y, se il dataset risultante dall'unione contiene più campi con lo stesso nome provenienti da origini dati diverse, all'inizio delle intestazioni delle colonna del campo vengono aggiunti i tag relativi provenienti dalle origini dati. |
| merge_condition | stringa | |
| ranking_expression | stringa | |

Tabella 62. proprietà mergenode (Continua).

| Proprietà mergenode | Data type | Descrizione proprietà |
|---------------------|----------------|---|
| Num_matches | <i>integer</i> | Il numero di corrispondenze da restituire, in base a merge_condition e ranking_expression. Minimo 1, massimo 100. |

Proprietà rfmaggreatenode



Il nodo Aggregazione RFM (Recency, Frequency, Monetary, Passato recente, Frequenza, Monetario) consente di prendere in considerazione i dati storici delle transazioni dei clienti, eliminare i dati non utilizzati e combinare tutti i dati delle transazioni rimanenti in un'unica riga che indica quanto tempo è trascorso dall'ultima transazione, il numero di transazioni effettuate e il valore monetario totale delle transazioni.

Esempio

```
node = stream.create("rfmaggregate", "My node")
node.setPropertyValue("relative_to", "Fixed")
node.setPropertyValue("reference_date", "2007-10-12")
node.setPropertyValue("id_field", "CardID")
node.setPropertyValue("date_field", "Date")
node.setPropertyValue("value_field", "Amount")
node.setPropertyValue("only_recent_transactions", True)
node.setPropertyValue("transaction_date_after", "2000-10-01")
```

Tabella 63. proprietà rfmaggreatenode.

| Proprietà rfmaggreatenode | Data type | Descrizione proprietà |
|---------------------------|-------------------|--|
| relative_to | Fixed Today | Specifica la data a partire dalla quale verrà calcolato il passato recente delle transazioni. |
| reference_date | <i>date</i> | Disponibile solo se per relative_to viene scelto Fissa. |
| contiguous | <i>indicatore</i> | Se i dati sono preordinati in modo che tutti i record con lo stesso ID appaiano insieme nel flusso di dati, selezionare questa opzione per accelerare l'elaborazione. |
| id_field | <i>campo</i> | Specifica il campo da utilizzare per identificare il cliente e le relative transazioni. |
| date_field | <i>campo</i> | Specifica il campo data da utilizzare per calcolare il passato recente. |
| value_field | <i>campo</i> | Specifica il campo da utilizzare per calcolare il valore monetario. |
| extension | <i>stringa</i> | Specifica un prefisso o suffisso per campi aggregati duplicati. |
| add_as | Suffix Prefix | Specifica se extension viene aggiunta come suffisso o prefisso. |
| discard_low_value_records | <i>indicatore</i> | Attiva l'utilizzo dell'impostazione discard_records_below. |
| discard_records_below | <i>numero</i> | Specifica un valore minimo al di sotto del quale non vengono utilizzati i dettagli delle transazioni nel calcolo dei totali RFM. Le unità di valore sono relative al campo valore selezionato. |

Tabella 63. proprietà rfmaggreatenode (Continua).

| Proprietà rfmaggreatenode | Data type | Descrizione proprietà |
|---------------------------|--------------------------------|--|
| only_recent_transactions | <i>indicatore</i> | Attiva l'utilizzo dell'impostazione specify_transaction_date o transaction_within_last. |
| specify_transaction_date | <i>indicatore</i> | |
| transaction_date_after | <i>date</i> | Disponibile solo se è selezionata specify_transaction_date. Specificare la data della transazione dopo la quale i record verranno inclusi nell'analisi. |
| transaction_within_last | <i>numero</i> | Disponibile solo se è selezionata transaction_within_last. Specifica il numero e il tipo di periodi (giorni, settimane, mesi o anni) dalla data di Calcola passato recente relativo a dopo la quale i record saranno inclusi nell'analisi. |
| transaction_scale | Days Weeks Mesi Years | Disponibile solo se è selezionata transaction_within_last. Specifica il numero e il tipo di periodi (giorni, settimane, mesi o anni) dalla data di Calcola passato recente relativo a dopo la quale i record saranno inclusi nell'analisi. |
| save_r2 | <i>indicatore</i> | Visualizza la data della seconda transazione più recente per ogni cliente. |
| save_r3 | <i>indicatore</i> | Disponibile solo se è selezionata save_r2. Visualizza la data della terza transazione più recente per ogni cliente. |

Proprietà Rprocessnode



Il nodo Trasformazioni R consente di estrarre i dati da un flusso IBM(r) SPSS(r) Modeler e di modificarli utilizzando il proprio script R personalizzato. Una volta modificati, i dati vengono restituiti al flusso.

Esempio

```
node = stream.create("rprocess", "My node")
node.setPropertyValue("custom_name", "my_node")
node.setPropertyValue("syntax", "" "day<-as.Date(modelerData$dob, format="%Y-%m-%d")
next_day<-day + 1
modelerData<-cbind(modelerData,next_day)
var1<-c(fieldName="Next day",fieldLabel="",fieldStorage="date",fieldMeasure="",fieldFormat="",
fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""")
node.setPropertyValue("convert_datetime", "POSIXct")
```

Tabella 64. Proprietà Rprocessnode.

| Proprietà Rprocessnode | Data type | Descrizione proprietà |
|------------------------|------------------------------------|-----------------------|
| syntax | <i>stringa</i> | |
| convert_flags | StringsAndDoubles LogicalValues | |
| convert_datetime | <i>indicatore</i> | |
| convert_datetime_class | POSIXct POSIXlt | |

Tabella 64. Proprietà Rprocessnode (Continua).

| Proprietà Rprocessnode | Data type | Descrizione proprietà |
|------------------------|------------|---|
| convert_missing | indicatore | |
| use_batch_size | indicatore | Attiva l'utilizzo dell'elaborazione batch |
| batch_size | integer | Specifica il numero di record di dati da includere in ciascun batch |

Proprietà del nodo Campione



Il nodo Campione seleziona un sottoinsieme di record. Sono supportati vari tipi di campioni, inclusi campioni stratificati, raggruppati e non casuali (strutturati). Il campionamento può essere utile per migliorare le prestazioni e per selezionare gruppi di record correlati o transazioni per un'analisi.

Esempio

```
/* Create two Sample nodes to extract
   different samples from the same data */
```

```
node = stream.create("sample", "My node")
node.setPropertyValue("method", "Simple")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("sample_type", "First")
node.setPropertyValue("first_n", 500)
```

```
node = stream.create("sample", "My node")
node.setPropertyValue("method", "Complex")
node.setPropertyValue("stratify_by", ["Sex", "Cholesterol"])
node.setPropertyValue("sample_units", "Proportions")
node.setPropertyValue("sample_size_proportions", "Custom")
node.setPropertyValue("sizes_proportions", [["M", "High", "Default"], ["M", "Normal", "Default"],
["F", "High", 0.3], ["F", "Normal", 0.3]])
```

Tabella 65. proprietà samplenode.

| Proprietà samplenode | Data type | Descrizione proprietà |
|----------------------|------------------------------|---|
| method | Simple Complex | |
| mode | Include Discard | Include o scarta i record che soddisfano la condizione specificata. |
| sample_type | First OneInN RandomPct | Specifica il metodo di campionamento. |
| first_n | numero intero | I record fino al punto di interruzione specificato verranno inclusi o scartati. |
| one_in_n | numero | Include o scarta ogni <i>n</i> record. |
| rand_pct | numero | Specifica la percentuale di record da includere o scartare. |
| use_max_size | indicatore | Attiva l'utilizzo dell'impostazione maximum_size. |

Tabella 65. proprietà *samplenode* (Continua).

| Proprietà <i>samplenode</i> | Data type | Descrizione proprietà |
|--------------------------------------|--|--|
| <code>maximum_size</code> | <i>numero intero</i> | Specifica la dimensione massima del campione da includere nel flusso di dati o da scartare. Questa opzione è ridondante e risulta pertanto disattivata se vengono specificati <code>Primi</code> e <code>Includi</code> . |
| <code>set_random_seed</code> | <i>indicatore</i> | Attiva l'utilizzo dell'impostazione del seme random. |
| <code>random_seed</code> | <i>numero intero</i> | Specifica il valore utilizzato come seme random. |
| <code>complex_sample_type</code> | Random Systematic | |
| <code>sample_units</code> | Proportions Counts | |
| <code>sample_size_proportions</code> | Fixed Custom Variable | |
| <code>sample_size_counts</code> | Fixed Custom Variable | |
| <code>fixed_proportions</code> | <i>numero</i> | |
| <code>fixed_counts</code> | <i>numero intero</i> | |
| <code>variable_proportions</code> | <i>campo</i> | |
| <code>variable_counts</code> | <i>campo</i> | |
| <code>use_min_stratum_size</code> | <i>indicatore</i> | |
| <code>minimum_stratum_size</code> | <i>numero intero</i> | Questa opzione è valida solo quando con <code>sample_units=Proportions</code> viene acquisito un campione Complesso. |
| <code>use_max_stratum_size</code> | <i>indicatore</i> | |
| <code>maximum_stratum_size</code> | <i>numero intero</i> | Questa opzione è valida solo quando con <code>sample_units=Proportions</code> viene acquisito un campione Complesso. |
| <code>clusters</code> | <i>campo</i> | |
| <code>stratify_by</code> | <i>[campo1 ... campoN]</i> | |
| <code>specify_input_weight</code> | <i>indicatore</i> | |
| <code>input_weight</code> | <i>campo</i> | |
| <code>new_output_weight</code> | <i>stringa</i> | |
| <code>sizes_proportions</code> | <i>[[string valore stringa][string valore stringa]...]</i> | Se <code>sample_units=proportions</code> e <code>sample_size_proportions=Custom</code> , specifica un valore per ogni possibile combinazione di valori di campi di stratificazione. |
| <code>default_proportion</code> | <i>numero</i> | |
| <code>sizes_counts</code> | <i>[[string valore stringa][string valore stringa]...]</i> | Specifica un valore per ogni possibile combinazione di valori di campi di stratificazione. L'utilizzo è simile a quello della proprietà <code>sizes_proportions</code> , con la differenza che viene specificato un numero intero anziché una proporzione. |
| <code>default_count</code> | <i>numero</i> | |

Proprietà selectnode



Il nodo Seleziona consente di selezionare o scartare un sottoinsieme di record dal flusso dei dati basato su una condizione specifica. Per esempio, è possibile selezionare i record relativi a una determinata area vendite.

Esempio

```
node = stream.create("select", "My node")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("condition", "Age < 18")
```

Tabella 66. proprietà selectnode.

| Proprietà selectnode | Data type | Descrizione proprietà |
|----------------------|--------------------|--|
| mode | Include Discard | Indica se includere o scartare i record selezionati. |
| condition | stringa | Condizione per includere o scartare i record. |

Proprietà sortnode



Il nodo Ordina ordina record in ordine crescente o decrescente in base ai valori di uno o più campi.

Esempio

```
node = stream.create("sort", "My node")
node.setPropertyValue("keys", [["Age", "Ascending"], ["Sex", "Descending"]])
node.setPropertyValue("default_ascending", False)
node.setPropertyValue("use_existing_keys", True)
node.setPropertyValue("existing_keys", [["Age", "Ascending"]])
```

Tabella 67. proprietà sortnode.

| Proprietà sortnode | Data type | Descrizione proprietà |
|--------------------|------------|--|
| keys | elenco | Specifica i campi in base ai quali si desidera eseguire l'ordinamento. Se non viene specificata una direzione, viene utilizzata quella di default. |
| default_ascending | indicatore | Specifica il criterio di ordinamento di default. |
| use_existing_keys | indicatore | Specifica se l'ordinamento è ottimizzato utilizzando il criterio di ordinamento precedente per i campi già ordinati. |
| existing_keys | | Specifica i campi già ordinati e la direzione nella quale sono ordinati. Utilizza lo stesso formato della proprietà keys. |

Proprietà streamingts



Il nodo Streaming TS crea e calcola il punteggio dei modelli delle serie temporali in un'unica fase, senza dover utilizzare un nodo Intervalli di tempo.

Esempio

```
node = stream.create("streamings", "My node")
node.setPropertyValue("deployment_force_rebuild", True)
node.setPropertyValue("deployment_rebuild_mode", "Count")
node.setPropertyValue("deployment_rebuild_count", 3)
node.setPropertyValue("deployment_rebuild_pct", 11)
node.setPropertyValue("deployment_rebuild_field", "Year")
```

Tabella 68. Proprietà streamings.

| Proprietà streamings | Data type | Descrizione proprietà |
|----------------------------------|---|--|
| custom_fields | <i>indicatore</i> | Se custom_fields=false, vengono usate le impostazioni dal nodo di tipo upstream. Se custom_fields=true, targets e inputs devono essere specificati. |
| targets | [campo1...campoN] | |
| inputs | [campo1...campoN] | |
| method | ExpertModeler Exsmooth Arima | |
| calculate_conf | <i>indicatore</i> | |
| conf_limit_pct | <i>reale</i> | |
| use_time_intervals_node | <i>indicatore</i> | Se use_time_intervals_node=true, vengono utilizzate le impostazioni del nodo Intervalli di tempo upstream. Se use_time_intervals_node=false, è necessario specificare interval_offset_position, interval_offset e interval_type. |
| interval_offset_position | LastObservation LastRecord | LastObservation si riferisce alla ultima osservazione valida . LastRecord si riferisce a ripresa conteggio dall'ultimo record . |
| interval_offset | <i>number</i> | |
| interval_type | Periods Years Quarters Months WeeksNonPeriodic DaysNonPeriodic HoursNonPeriodic MinutesNonPeriodic SecondsNonPeriodic | |
| events | <i>campi</i> | |
| expert_modeler_method | AllModels Exsmooth Arima | |
| consider_seasonal | <i>indicatore</i> | |
| detect_outliers | <i>indicatore</i> | |
| expert_outlier_additive | <i>indicatore</i> | |
| expert_outlier_level_shift | <i>indicatore</i> | |
| expert_outlier_innovational | <i>indicatore</i> | |
| expert_outlier_transient | <i>indicatore</i> | |
| expert_outlier_seasonal_additive | <i>indicatore</i> | |

Tabella 68. Proprietà *streamingts* (Continua).

| Proprietà <i>streamingts</i> | Data type | Descrizione proprietà |
|---|---|--|
| <code>expert_outlier_local_trend</code> | <i>indicatore</i> | |
| <code>expert_outlier_additive_patch</code> | <i>indicatore</i> | |
| <code>exsmooth_model_type</code> | Simple HoltLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative | |
| <code>exsmooth_transformation_type</code> | None SquareRoot NaturalLog | |
| <code>arima_p</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali |
| <code>arima_d</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali |
| <code>arima_q</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali |
| <code>arima_sp</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali |
| <code>arima_sd</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali |
| <code>arima_sq</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali |
| <code>arima_transformation_type</code> | None SquareRoot NaturalLog | Stessa proprietà del nodo modelli serie temporali |
| <code>arima_include_constant</code> | <i>indicatore</i> | Stessa proprietà del nodo modelli serie temporali |
| <code>tf_arima_p.fieldname</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali. Per le funzioni di trasferimento. |
| <code>tf_arima_d.fieldname</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali. Per le funzioni di trasferimento. |
| <code>tf_arima_q.fieldname</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali. Per le funzioni di trasferimento. |
| <code>tf_arima_sp.fieldname</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali. Per le funzioni di trasferimento. |
| <code>tf_arima_sd.fieldname</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali. Per le funzioni di trasferimento. |
| <code>tf_arima_sq.fieldname</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali. Per le funzioni di trasferimento. |
| <code>tf_arima_delay.fieldname</code> | <i>integer</i> | Stessa proprietà del nodo modelli serie temporali. Per le funzioni di trasferimento. |
| <code>tf_arima_transformation_type.fieldname</code> | None SquareRoot NaturalLog | |
| <code>arima_detect_outlier_mode</code> | None Automatic | |
| <code>arima_outlier_additive</code> | <i>indicatore</i> | |

Tabella 68. Proprietà streamings (Continua).

| Proprietà streamings | Data type | Descrizione proprietà |
|---------------------------------|----------------------|-----------------------|
| arima_outlier_level_shift | <i>indicatore</i> | |
| arima_outlier_innovational | <i>indicatore</i> | |
| arima_outlier_transient | <i>indicatore</i> | |
| arima_outlier_seasonal_additive | <i>indicatore</i> | |
| arima_outlier_local_trend | <i>indicatore</i> | |
| arima_outlier_additive_patch | <i>indicatore</i> | |
| deployment_force_rebuild | <i>indicatore</i> | |
| deployment_rebuild_mode | Count Percent | |
| deployment_rebuild_count | <i>number</i> | |
| deployment_rebuild_pct | <i>number</i> | |
| deployment_rebuild_field | <i><field></i> | |

Capitolo 11. Proprietà dei nodi Operazioni su campi

Proprietà anonymizenode



Il nodo Anonimizza consente di mascherare i nomi o i valori dei campi, quando si utilizzano dati da includere in un modello a valle del nodo, permettendo di nascondere i dati originali. Questa funzionalità può essere utile se si desidera consentire ad altri utenti di creare modelli utilizzando dati riservati, quali nomi di clienti o altri dettagli.

Esempio

```
stream = modeler.script.stream()
varfilenode = stream.createAt("variablefile", "File", 96, 96)
varfilenode.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")
node = stream.createAt("anonymize", "My node", 192, 96)
# Anonymize node requires the input fields while setting the values
stream.link(varfilenode, node)
node.setKeyedPropertyValue("enable_anonymize", "Age", True)
node.setKeyedPropertyValue("transformation", "Age", "Random")
node.setKeyedPropertyValue("set_random_seed", "Age", True)
node.setKeyedPropertyValue("random_seed", "Age", 123)
node.setKeyedPropertyValue("enable_anonymize", "Drug", True)
node.setKeyedPropertyValue("use_prefix", "Drug", True)
node.setKeyedPropertyValue("prefix", "Drug", "myprefix")
```

Tabella 69. proprietà anonymizenode

| Proprietà anonymizenode | Data type | Descrizione proprietà |
|-------------------------|----------------------|---|
| enable_anonymize | <i>indicatore</i> | Se impostata su True, attiva l'anonimizzazione dei valori dei campi (equivale alla selezione di Si per tale campo nella colonna Anonimizza valori). |
| use_prefix | <i>indicatore</i> | Quando è impostata su True, viene utilizzato un prefisso personalizzato, se ne è stato definito uno. È valida per i campi che saranno anonimizzati con il metodo hash ed equivale alla selezione del pulsante di scelta Personalizzato nella finestra di dialogo Sostituisci valori di quel campo. |
| prefix | <i>stringa</i> | Equivale alla digitazione di un prefisso nella casella di testo della finestra di dialogo Sostituisci valori. Se non sono stati specificati altri valori, il prefisso di default è il valore di default. |
| transformation | Random Fixed | Determina se i parametri di trasformazione di un campo anonimizzato con il metodo Trasformazioni saranno casuali o fissi. |
| set_random_seed | <i>indicatore</i> | Quando è impostata su True, viene utilizzato il valore di seed specificato (se transformation è impostato su Random). |
| random_seed | <i>numero intero</i> | Quando set_random_seed è impostata su True, questo è il seed del numero casuale. |
| scale | <i>numero</i> | Quando transformation è impostata su Fixed, questo valore viene utilizzato per "scale by". Il valore di scala massimo in genere è 10, ma può essere ridotto per evitare l'overflow. |
| translate | <i>numero</i> | Quando transformation è impostata su Fixed, questo valore viene utilizzato per "translate". Il valore di translate massimo in genere è 1000, ma può essere ridotto per evitare l'overflow. |

Proprietà autodatapreinode



Il nodo Preparazione automatica dati (ADP) può analizzare i dati e individuare le correzioni, escludere i campi problematici o probabilmente inutili e derivare all'occorrenza nuovi attributi, migliorando le performance grazie allo screening intelligente e alle tecniche di campionamento. Il nodo si può utilizzare in modo completamente automatico, permettendogli di scegliere e di applicare le correzioni, oppure visualizzando in anteprima le modifiche prima dell'applicazione e accettandole, respingendole o modificandole a seconda dei casi.

Esempio

```
node = stream.create("autodataprep", "My node")
node.setPropertyValue("objective", "Balanced")
node.setPropertyValue("excluded_fields", "Filter")
node.setPropertyValue("prepare_dates_and_times", True)
node.setPropertyValue("compute_time_until_date", True)
node.setPropertyValue("reference_date", "Today")
node.setPropertyValue("units_for_date_durations", "Automatic")
```

Tabella 70. proprietà autodatapreinode

| Proprietà autodatapreinode | Data type | Descrizione proprietà |
|----------------------------|---|--|
| objective | Balanced Speed Accuracy Custom | |
| custom_fields | <i>indicatore</i> | Se vera, consente di specificare i campi obiettivo, di input e di altro tipo per il nodo corrente. Se falsa, vengono utilizzate le impostazioni correnti di un nodo tipologia a monte. |
| obiettivo | <i>campo</i> | Specifica un singolo campo obiettivo. |
| inputs | [<i>campo1 ... campoN</i>] | I campi di input o predittore utilizzati dal modello. |
| use_frequency | <i>indicatore</i> | |
| frequency_field | <i>campo</i> | |
| use_weight | <i>indicatore</i> | |
| weight_field | <i>campo</i> | |
| excluded_fields | Filter None | |
| if_fields_do_not_match | StopExecution ClearAnalysis | |
| prepare_dates_and_times | <i>indicatore</i> | Controllo dell'accesso a tutti i campi data e ora |
| compute_time_until_date | <i>indicatore</i> | |
| reference_date | Today Fixed | |
| fixed_date | <i>data</i> | |
| units_for_date_durations | Automatic Fixed | |
| fixed_date_units | Years Mesi Days | |

Tabella 70. proprietà autodatapreinode (Continua)

| Proprietà autodatapreinode | Data type | Descrizione proprietà |
|-----------------------------------|-----------------------------|-----------------------|
| compute_time_until_time | <i>indicatore</i> | |
| reference_time | CurrentTime Fixed | |
| fixed_time | <i>ora</i> | |
| units_for_time_durations | Automatic Fixed | |
| fixed_date_units | Hours Minutes Seconds | |
| extract_year_from_date | <i>indicatore</i> | |
| extract_month_from_date | <i>indicatore</i> | |
| extract_day_from_date | <i>indicatore</i> | |
| extract_hour_from_time | <i>indicatore</i> | |
| extract_minute_from_time | <i>indicatore</i> | |
| extract_second_from_time | <i>indicatore</i> | |
| exclude_low_quality_inputs | <i>indicatore</i> | |
| exclude_too_many_missing | <i>indicatore</i> | |
| maximum_percentage_missing | <i>numero</i> | |
| exclude_too_many_categories | <i>indicatore</i> | |
| maximum_number_categories | <i>numero</i> | |
| exclude_if_large_category | <i>indicatore</i> | |
| maximum_percentage_category | <i>numero</i> | |
| prepare_inputs_and_target | <i>indicatore</i> | |
| adjust_type_inputs | <i>indicatore</i> | |
| adjust_type_target | <i>indicatore</i> | |
| reorder_nominal_inputs | <i>indicatore</i> | |
| reorder_nominal_target | <i>indicatore</i> | |
| replace_outliers_inputs | <i>indicatore</i> | |
| replace_outliers_target | <i>indicatore</i> | |
| replace_missing_continuous_inputs | <i>indicatore</i> | |
| replace_missing_continuous_target | <i>indicatore</i> | |
| replace_missing_nominal_inputs | <i>indicatore</i> | |
| replace_missing_nominal_target | <i>indicatore</i> | |
| replace_missing_ordinal_inputs | <i>indicatore</i> | |
| replace_missing_ordinal_target | <i>indicatore</i> | |
| maximum_values_for_ordinal | <i>numero</i> | |
| minimum_values_for_continuous | <i>numero</i> | |
| outlier_cutoff_value | <i>numero</i> | |
| outlier_method | Replace Delete | |
| rescale_continuous_inputs | <i>indicatore</i> | |

Tabella 70. proprietà autodatapreppnode (Continua)

| Proprietà autodatapreppnode | Data type | Descrizione proprietà |
|-----------------------------------|------------------|-----------------------|
| rescaling_method | MinMax ZScore | |
| min_max_minimum | numero | |
| min_max_maximum | numero | |
| z_score_final_mean | numero | |
| z_score_final_sd | numero | |
| rescale_continuous_target | indicatore | |
| target_final_mean | numero | |
| target_final_sd | numero | |
| transform_select_input_fields | indicatore | |
| maximize_association_with_target | indicatore | |
| p_value_for_merging | numero | |
| merge_ordinal_features | indicatore | |
| merge_nominal_features | indicatore | |
| minimum_cases_in_category | numero | |
| bin_continuous_fields | indicatore | |
| p_value_for_binning | numero | |
| perform_feature_selection | indicatore | |
| p_value_for_selection | numero | |
| perform_feature_construction | indicatore | |
| transformed_target_name_extension | stringa | |
| transformed_inputs_name_extension | stringa | |
| constructed_features_root_name | stringa | |
| years_duration_name_extension | stringa | |
| months_duration_name_extension | stringa | |
| days_duration_name_extension | stringa | |
| hours_duration_name_extension | stringa | |
| minutes_duration_name_extension | stringa | |
| seconds_duration_name_extension | stringa | |
| year_cyclical_name_extension | stringa | |
| month_cyclical_name_extension | stringa | |
| day_cyclical_name_extension | stringa | |
| hour_cyclical_name_extension | stringa | |
| minute_cyclical_name_extension | stringa | |
| second_cyclical_name_extension | stringa | |

Proprietà astimeintervalsnode



Il nodo Intervalli di tempo originale non è compatibile con Analytic Server (AS). Il nodo Intervalli di tempo AS (nuovo in SPSS Modeler release 17.0) contiene un sottoinsieme delle funzioni del nodo Intervalli di tempo esistente che può essere utilizzato con Analytic Server.

Utilizzare il nodo Intervalli di tempo AS per specificare gli intervalli e derivare un nuovo campo ora per effettuare stime o previsioni. È supportata una gamma completa di intervalli temporali, dai secondi agli anni.

Tabella 71. Proprietà astimeintervalsnode

| Proprietà astimeintervalsnode | Data type | Descrizione proprietà |
|-------------------------------|-----------------------------------|--|
| time_field | <i>campo</i> | Prevede sono un solo campo continuo. Tale campo viene utilizzato dal nodo come chiave di aggregazione per la conversione dell'intervallo. Se viene utilizzato un campo intero, viene considerato come un indice temporale. |
| dimensions | <i>[campo1 campo2 ... campon]</i> | Questi campi vengono utilizzati per creare singole serie temporali basate sui valori del campo. |
| fields_to_aggregate | <i>[campo1 campo2 ... campon]</i> | Questi campi vengono aggregati come parte della modifica del campo periodo di tempo. Tutti i campi non inclusi in questa selezione vengono filtrati dai dati che escono dal nodo. |

Proprietà binningnode



Il nodo Discretizza crea automaticamente nuovi campi nominali (insieme) basati sui valori di uno o più campi continui (intervallo numerico) esistenti. Per esempio, è possibile trasformare un campo continuo relativo al reddito in campo categoriale contenente gruppi di reddito come deviazioni dalla media. Dopo aver creato bin per il nuovo campo, è possibile generare un nodo Ricava basato sui punti di divisione.

Esempio

```
node = stream.create("binning", "My node")
node.setPropertyValue("fields", ["Na", "K"])
node.setPropertyValue("method", "Rank")
node.setPropertyValue("fixed_width_name_extension", "_binned")
node.setPropertyValue("fixed_width_add_as", "Suffix")
node.setPropertyValue("fixed_bin_method", "Count")
node.setPropertyValue("fixed_bin_count", 10)
node.setPropertyValue("fixed_bin_width", 3.5)
node.setPropertyValue("tile10", True)
```

Tabella 72. proprietà binningnode

| Proprietà binningnode | Data type | Descrizione proprietà |
|-----------------------|-----------------------------------|---|
| campi | <i>[campo1 campo2 ... campon]</i> | Campi continui (intervalli numerici) in attesa di trasformazione. È possibile eseguire la discretizzazione di più campi contemporaneamente. |

Tabella 72. proprietà binningnode (Continua)

| Proprietà binningnode | Data type | Descrizione proprietà |
|----------------------------|---|---|
| method | FixedWidth EqualCount Rank SDev Optimal | Metodo utilizzato per determinare i punti di divisione per i nuovi bin di campo (categorie). |
| rcalculate_bins | Always IfNecessary | Specifica se i bin vengono ricalcolati e i dati collocati nel bin corrispondente ogni volta che viene eseguito il nodo o se i dati vengono semplicemente inseriti nei bin esistenti e negli eventuali nuovi bin aggiunti. |
| fixed_width_name_extension | string | The default extension is <i>_BIN</i> . |
| fixed_width_add_as | Suffix Prefix | Specifica se l'estensione viene aggiunta alla fine (suffisso) del nome del campo oppure all'inizio (prefisso). L'estensione di default è <i>income_BIN</i> . |
| fixed_bin_method | Width Count | |
| fixed_bin_count | numero intero | Specifica un numero intero utilizzato per determinare il numero di bin a larghezza fissa (categorie) per i nuovi campi. |
| fixed_bin_width | real | Valore (numero intero o reale) utilizzato per calcolare la larghezza del bin. |
| equal_count_name_extension | stringa | L'estensione di default è <i>_TILE</i> . |
| equal_count_add_as | Suffix Prefix | Specifica un'estensione, un suffisso o un prefisso, utilizzata per il nome del campo generato con p-tili standard. L'estensione di default è <i>_TILE</i> preceduta da <i>N</i> , dove <i>N</i> è il numero percentile. |
| tile4 | indicatore | Genera quattro bin quantile, ognuno contenente il 25% dei casi. |
| tile5 | indicatore | Genera cinque bin quintile. |
| tile10 | indicatore | Genera 10 bin decile. |
| tile20 | indicatore | Genera 20 bin ventile. |
| tile100 | indicatore | Genera 100 bin percentile. |
| use_custom_tile | indicatore | |
| custom_tile_name_extension | stringa | L'estensione di default è <i>_TILEN</i> . |
| custom_tile_add_as | Suffix Prefix | |
| custom_tile | numero intero | |
| equal_count_method | RecordCount ValueSum | Il metodo RecordCount cerca di assegnare un numero uguale di record a ciascun bin, mentre ValueSum assegna i record in modo che la somma dei valori in ogni bin sia uguale. |
| tied_values_method | Next Current Random | Specifica quali dati relativi ai valori pari merito dei bin devono essere inseriti. |

Tabella 72. proprietà binningnode (Continua)

| Proprietà binningnode | Data type | Descrizione proprietà |
|--------------------------------|-------------------------|--|
| rank_order | Ascending Descending | Questa proprietà include Ascending (il valore più basso viene indicato con 1) o Descending (il valore più alto viene indicato con 1). |
| rank_add_as | Suffix Prefix | Questa opzione è applicabile a rango, rango frazionario e percentuale rango. |
| rango | <i>indicatore</i> | |
| rank_name_extension | <i>stringa</i> | L'estensione di default è <i>_RANK</i> . |
| rank_fractional | <i>indicatore</i> | Opzioni dei ranghi in cui il valore del nuovo campo equivale al rango diviso per la somma dei pesi dei casi non mancanti. I ranghi frazionari sono compresi nell'intervallo tra 0 e 1. |
| rank_fractional_name_extension | <i>stringa</i> | L'estensione di default è <i>_F_RANK</i> . |
| rank_pct | <i>indicatore</i> | Ogni rango è diviso in base al numero di record con valori validi e moltiplicato per 100. I ranghi frazionari in percentuale sono compresi nell'intervallo tra 1 e 100. |
| rank_pct_name_extension | <i>stringa</i> | L'estensione di default è <i>_P_RANK</i> . |
| sdev_name_extension | <i>stringa</i> | |
| sdev_add_as | Suffix Prefix | |
| sdev_count | One Two Three | |
| optimal_name_extension | <i>stringa</i> | L'estensione di default è <i>_OPTIMAL</i> . |
| optimal_add_as | Suffix Prefix | |
| optimal_supervisor_field | <i>campo</i> | Campo scelto come campo supervisore a cui sono correlati i campi selezionati per la discretizzazione. |
| optimal_merge_bins | <i>indicatore</i> | Specifica che tutti i bin con conteggi di casi ridotti vengono aggiunti a bin più grandi adiacenti. |
| optimal_small_bin_threshold | <i>numero intero</i> | |
| optimal_pre_bin | <i>indicatore</i> | Indica che deve essere eseguita la discretizzazione preventiva del dataset. |
| optimal_max_bins | <i>numero intero</i> | Specifica un limite superiore per evitare di creare un numero eccessivamente elevato di bin. |
| optimal_lower_end_point | Inclusive Exclusive | |
| optimal_first_bin | Unbounded Bounded | |
| optimal_last_bin | Unbounded Bounded | |

Proprietà derivenode



Il nodo Ricava modifica valori di dati o crea nuovi campi da uno o più campi esistenti. Crea campi di tipo Formula, Flag, Nominale, Stato, Conteggio e Condizionale.

Esempio 1

```
# Create and configure a Flag Derive field node
node = stream.create("derive", "My node")
node.setPropertyValue("new_name", "DrugX_Flag")
node.setPropertyValue("result_type", "Flag")
node.setPropertyValue("flag_true", "1")
node.setPropertyValue("flag_false", "0")
node.setPropertyValue("flag_expr", "'Drug' == \"drugX\"")

# Create and configure a Conditional Derive field node
node = stream.create("derive", "My node")
node.setPropertyValue("result_type", "Conditional")
node.setPropertyValue("cond_if_cond", "@OFFSET(\"Age\", 1) = \"Age\"")
node.setPropertyValue("cond_then_expr", "@OFFSET(\"Age\", 1) = \"Age\" >> @INDEX")
node.setPropertyValue("cond_else_expr", "\"Age\"")
```

Esempio 2

In questo script, si suppone siano disponibili due colonne numeriche denominate XPos e YPos che rappresentano le coordinate X e Y di un punto (ad esempio, il punto in cui si è verificato un evento). Lo script crea un nodo Ricava che calcola una colonna geospaziale dalle coordinate X e Y che rappresentano quel punto in uno specifico sistema di coordinate:

```
stream = modeler.script.stream()
# Other stream configuration code
node = stream.createAt("derive", "Location", 192, 96)
node.setPropertyValue("new_name", "Location")
node.setPropertyValue("formula_expr", "['XPos', 'YPos']")
node.setPropertyValue("formula_type", "Geospatial")
# Now we have set the general measurement type, define the
# specifics of the geospatial object
node.setPropertyValue("geo_type", "Point")
node.setPropertyValue("has_coordinate_system", True)
node.setPropertyValue("coordinate_system", "ETRS_1989_EPSG_Arctic_zone_5-47")
```

Tabella 73. Proprietà derivenode

| Proprietà derivenode | Data type | Descrizione proprietà |
|----------------------|--------------------|--|
| new_name | stringa | Nome del nuovo campo. |
| mode | Single Multiple | Specifica campi singoli o multipli. |
| fields | elenco | Utilizzata nella modalità Multiple solo per selezionare più campi. |
| name_extension | stringa | Specifica l'estensione del nome del nuovo campo. |
| add_as | Suffisso Prefix | Aggiunge l'estensione come prefisso (all'inizio) o come suffisso (alla fine) del nome del campo. |

Tabella 73. Proprietà derivenode (Continua)

| Proprietà derivenode | Data type | Descrizione proprietà |
|-----------------------|---|--|
| result_type | Formula Flag Set State Count Conditional | Sei tipi di nuovi campi che è possibile creare. |
| formula_expr | stringa | Espressione per il calcolo del nuovo valore del campo in qualsiasi nodo Ricava. |
| flag_expr | stringa | |
| flag_true | stringa | |
| flag_false | stringa | |
| set_default | stringa | |
| set_value_cond | stringa | Strutturata per fornire la condizione associata a un valore specificato. |
| state_on_val | stringa | Specifica il valore per il nuovo campo quando viene soddisfatta la condizione Attivato. |
| state_off_val | stringa | Specifica il valore per il nuovo campo quando viene soddisfatta la condizione Disattivato. |
| state_on_expression | stringa | |
| state_off_expression | stringa | |
| state_initial | On Off | Assegna ad ogni record del nuovo campo un valore iniziale di On o Off. Questo valore può cambiare quando viene soddisfatta ciascuna condizione. |
| count_initial_val | stringa | |
| count_inc_condition | stringa | |
| count_inc_expression | stringa | |
| count_reset_condition | stringa | |
| cond_if_cond | stringa | |
| cond_then_expr | stringa | |
| cond_else_expr | stringa | |
| formula_measure_type | Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL | Questa proprietà può essere utilizzata per definire la misurazione associata al campo derivato. La funzione setter può essere passata come stringa o come uno dei valori MeasureType. La funzione getter viene sempre restituita sui valori MeasureType. |
| collection_measure | Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS | Per i campi di raccolta (elenchi con profondità uguale a 0), questa proprietà definisce il tipo di misurazione associato ai valori sottostanti. |

Tabella 73. Proprietà *derivenode* (Continua)

| Proprietà <i>derivenode</i> | Data type | Descrizione proprietà |
|-----------------------------|---|---|
| geo_type | Point MultiPoint LineString MultiLineString Polygon MultiPolygon | Per i campi geospaziali, questa proprietà definisce il tipo di oggetto geospaziale rappresentato da questo campo. Questo valore deve essere coerente con la profondità di elenco dei valori |
| has_coordinate_system | <i>booleano</i> | Per i campi geospaziali, questa proprietà definisce se questo campo dispone di un sistema di coordinate |
| coordinate_system | <i>stringa</i> | Per i campi geospaziali, questa proprietà definisce il sistema di coordinate per questo campo |

Proprietà *ensemblenode*



Il nodo dell'insieme combina due o più nugget del modello al fine di ottenere previsioni più precise di quelle ricavabili dai singoli modelli.

Esempio

```
# Create and configure an Ensemble node
# Use this node with the models in demos\streams\pm_binaryclassifier.str
node = stream.create("ensemble", "My node")
node.setPropertyValue("ensemble_target_field", "response")
node.setPropertyValue("filter_individual_model_output", False)
node.setPropertyValue("flag_ensemble_method", "ConfidenceWeightedVoting")
node.setPropertyValue("flag_voting_tie_selection", "HighestConfidence")
```

Tabella 74. proprietà *ensemblenode*.

| Proprietà <i>ensemblenode</i> | Data type | Descrizione proprietà |
|--------------------------------|---|--|
| ensemble_target_field | <i>campo</i> | Specifica il campo obiettivo per tutti i modelli utilizzati nell'insieme. |
| filter_individual_model_output | <i>indicatore</i> | Specifica se i risultati del calcolo del punteggio dei singoli modelli devono essere esclusi. |
| flag_ensemble_method | Voting ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity | Specifica il metodo utilizzato per determinare il punteggio dell'insieme. Questa impostazione è valida solamente se l'obiettivo selezionato è un campo flag. |
| set_ensemble_method | Voting ConfidenceWeightedVoting HighestConfidence | Specifica il metodo utilizzato per determinare il punteggio dell'insieme. Questa impostazione è valida solamente se l'obiettivo selezionato è un campo nominale. |

Tabella 74. proprietà *ensemblenode* (Continua).

| Proprietà <i>ensemblenode</i> | Data type | Descrizione proprietà |
|-------------------------------|--|---|
| flag_voting_tie_selection | Random HighestConfidence RawPropensity AdjustedPropensity | Se è selezionato un metodo del confronto, specifica le modalità di risoluzione delle situazioni di pari merito. Questa impostazione è valida solamente se l'obiettivo selezionato è un campo flag. |
| set_voting_tie_selection | Random HighestConfidence | Se è selezionato un metodo del confronto, specifica le modalità di risoluzione delle situazioni di pari merito. Questa impostazione è valida solamente se l'obiettivo selezionato è un campo nominale. |
| calculate_standard_error | <i>indicatore</i> | Se il campo obiettivo è continuo viene eseguito per default il calcolo dell'errore standard per calcolare la differenza fra i valori misurati o stimati e i valori veri e per evidenziare il grado di corrispondenza di tali stime. |

Proprietà *fillernode*



Il nodo Riempimento sostituisce valori di campo e modifica l'archiviazione. È possibile scegliere di sostituire i valori in base a una condizione CLEM, per esempio @BLANK(@FIELD). In alternativa, si può scegliere di sostituire tutti i valori null o vuoti con un valore specifico. Il nodo Riempimento è utilizzato spesso in combinazione con il nodo Tipo per sostituire valori mancanti.

Esempio

```
node = stream.create("filler", "My node")
node.setPropertyValue("fields", ["Age"])
node.setPropertyValue("replace_mode", "Always")
node.setPropertyValue("condition", "(\"Age\" > 60) and (\"Sex\" = \"M\")")
node.setPropertyValue("replace_with", "\"old man\"")
```

Tabella 75. proprietà *fillernode*

| Proprietà <i>fillernode</i> | Data type | Descrizione proprietà |
|-----------------------------|--|---|
| fields | <i>elenco</i> | Campi dell'insieme di dati i cui valori saranno esaminati e sostituiti. |
| replace_mode | Always Conditional Blank Null BlankAndNull | È possibile sostituire tutti i valori, i valori vuoti, i valori null oppure sostituire i valori basati su una condizione specifica. |
| condition | <i>stringa</i> | |
| replace_with | <i>stringa</i> | |

Proprietà filternode



Il nodo Filtro filtra (ignora) campi, rinomina campi e mappa campi tra i nodi origine.

Esempio

```
node = stream.create("filter", "My node")
node.setPropertyValue("default_include", True)
node.setKeyedPropertyValue("new_name", "Drug", "Chemical")
node.setKeyedPropertyValue("include", "Drug", False)
```

Utilizzo della proprietà default_include. Si noti che l'impostazione del valore della proprietà default_include non include o esclude automaticamente tutti i campi, ma determina semplicemente l'impostazione di default della selezione corrente. Dal punto di vista funzionale, equivale a fare clic sul pulsante **Include i campi per default** nella finestra di dialogo Nodo Filtro. Per esempio, si supponga di eseguire lo script seguente:

```
node = modeler.script.stream().create("filter", "Filter")
node.setPropertyValue("default_include", False)
# Include these two fields in the list
for f in ["Age", "Sex"]:
    node.setKeyedPropertyValue("include", f, True)
```

Il nodo passerà i campi *Età* e *Sesso* e scarterà tutti gli altri. Si supponga ora di eseguire di nuovo lo stesso script ma indicando due campi diversi:

```
node = modeler.script.stream().create("filter", "Filter")
node.setPropertyValue("default_include", False)
# Include these two fields in the list
for f in ["BP", "Na"]:
    node.setKeyedPropertyValue("include", f, True)
```

Verranno aggiunti altri due campi al filtro, per un totale di quattro campi passati (*Età*, *Sesso*, *Pressione*, *Na*). In altre parole, il fatto di reimpostare il valore di default_include su Falso non reimposta automaticamente tutti i campi.

In alternativa, se si cambia default_include in Vero utilizzando uno script o dalla finestra di dialogo Nodo Filtro, si inverte il comportamento in modo che i quattro campi sopraindicati vengano scartati anziché inclusi. In caso di dubbio, potrebbe essere utile sperimentare con i controlli della finestra di dialogo Nodo Filtro per capire questa interazione.

Tabella 76. proprietà filternode

| Proprietà filternode | Data type | Descrizione proprietà |
|----------------------|------------|--|
| default_include | indicatore | Proprietà basata su chiavi utilizzata per specificare se il comportamento di default determina il passaggio o il filtro di campi: Si noti che l'impostazione di questa proprietà non include o esclude automaticamente tutti i campi, ma determina semplicemente se i campi selezionati sono inclusi o esclusi per default. Per commenti aggiuntivi, vedere l'esempio seguente. |
| include | indicatore | Proprietà basata su chiavi utilizzata per l'inclusione e la rimozione. |

Tabella 76. proprietà *filternode* (Continua)

| Proprietà <i>filternode</i> | Data type | Descrizione proprietà |
|-----------------------------|----------------|-----------------------|
| <code>new_name</code> | <i>stringa</i> | |

Proprietà *historynode*



Il nodo Cronologia crea nuovi campi contenenti dati dei campi di record precedenti. I nodi Cronologia sono utilizzati in genere per dati sequenziali, per esempio per dati di serie temporali. Prima di utilizzare un nodo Cronologia, può essere utile ordinare i dati con un nodo Ordina.

Esempio

```
node = stream.create("history", "My node")
node.setPropertyValue("fields", ["Drug"])
node.setPropertyValue("offset", 1)
node.setPropertyValue("span", 3)
node.setPropertyValue("unavailable", "Discard")
node.setPropertyValue("fill_with", "undef")
```

Tabella 77. proprietà *historynode*

| Proprietà <i>historynode</i> | Data type | Descrizione proprietà |
|------------------------------|--------------------------|---|
| <code>fields</code> | <i>elenco</i> | Campi di cui si desidera creare una cronologia. |
| <code>offset</code> | <i>numero</i> | Specifica il record che precede quello corrente dal quale si desidera estrarre i valori di campo cronologici. |
| <code>span</code> | <i>numero</i> | Specifica il numero di record precedenti a partire dal quale si desidera estrarre i valori. |
| <code>unavailable</code> | Discard Leave Fill | Per la gestione di record senza valori di cronologia, in genere riferiti ai primi record (all'inizio dell'insieme di dati), per i quali non esistono record precedenti da utilizzare come cronologia. |
| <code>fill_with</code> | String Number | Specifica il valore o la stringa da utilizzare per i record in cui non sia disponibile un valore cronologico. |

Proprietà *partitionnode*



Il nodo Partizione genera un campo partizione che suddivide i dati in sottoinsiemi separati per le fasi di addestramento, verifica e convalida della creazione del modello.

Esempio

```
node = stream.create("partition", "My node")
node.setPropertyValue("create_validation", True)
node.setPropertyValue("training_size", 33)
node.setPropertyValue("testing_size", 33)
```

```

node.setPropertyValue("validation_size", 33)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 123)
node.setPropertyValue("value_mode", "System")

```

Tabella 78. proprietà *partitionnode*

| Proprietà <i>partitionnode</i> | Data type | Descrizione proprietà |
|--------------------------------|-----------------------------------|--|
| <i>new_name</i> | <i>stringa</i> | Nome del campo partizione generato dal nodo. |
| <i>create_validation</i> | <i>indicatore</i> | Specifica se deve essere creata una partizione di convalida. |
| <i>training_size</i> | <i>numero intero</i> | Percentuale di record (0–100) da allocare alla partizione di addestramento. |
| <i>testing_size</i> | <i>numero intero</i> | Percentuale di record (0-100) da allocare alla partizione di test. |
| <i>validation_size</i> | <i>numero intero</i> | Percentuale di record (0-100) da allocare alla partizione di convalida. Viene ignorata se non viene creata alcuna partizione di convalida. |
| <i>training_label</i> | <i>stringa</i> | Etichetta per la partizione di convalida. |
| <i>testing_label</i> | <i>stringa</i> | Etichetta per la partizione di test. |
| <i>validation_label</i> | <i>stringa</i> | Etichetta per la partizione di convalida. Viene ignorata se non viene creata alcuna partizione di convalida. |
| <i>value_mode</i> | System SystemAndLabel Label | Specifica i valori utilizzati per rappresentare ogni partizione nei dati. Per esempio, il campione di addestramento può essere rappresentato dal valore intero di sistema 1, dall'etichetta Training o da una combinazione dei due 1_Training. |
| <i>set_random_seed</i> | <i>Booleano</i> | Specifica se è necessario utilizzare un seme random definito dall'utente. |
| <i>random_seed</i> | <i>numero intero</i> | Valore del seme random definito dall'utente. Per l'utilizzo di questo valore è necessario che <i>set_random_seed</i> sia impostata su True. |
| <i>enable_sql_generation</i> | <i>Booleano</i> | Specifica se utilizzare il push back SQL per assegnare i record alle partizioni. |
| <i>unique_field</i> | | Specifica il campo di input utilizzato per garantire che i record vengano assegnati alle partizioni in modo casuale ma ripetibile. Per l'utilizzo di questo valore è necessario che <i>enable_sql_generation</i> sia impostata su True. |

Proprietà del nodo Ricodifica



Il nodo Ricodifica trasforma un insieme di valori categoriali in un altro. L'operazione di ricodifica consente di comprimere categorie o raggruppare dati per l'analisi.

Esempio

```

node = stream.create("reclassify", "My node")
node.setPropertyValue("mode", "Multiple")
node.setPropertyValue("replace_field", True)
node.setPropertyValue("field", "Drug")

```

```

node.setPropertyValue("new_name", "Chemical")
node.setPropertyValue("fields", ["Drug", "BP"])
node.setPropertyValue("name_extension", "reclassified")
node.setPropertyValue("add_as", "Prefix")
node.setKeyedPropertyValue("reclassify", "drugA", True)
node.setPropertyValue("use_default", True)
node.setPropertyValue("default", "BrandX")
node.setPropertyValue("pick_list", ["BrandX", "Placebo", "Generic"])

```

Tabella 79. proprietà del nodo Ricodifica

| Proprietà reclassifynode | Data type | Descrizione proprietà |
|--------------------------|----------------------------------|---|
| mode | Single Multiple | La modalità Single ricodifica le categorie per un campo. La modalità Multiple attiva le opzioni che consentono la trasformazione di più campi contemporaneamente. |
| replace_field | indicatore | |
| campo | stringa | Utilizzata solo in modalità Singola. |
| new_name | stringa | Utilizzata solo in modalità Singola. |
| fields | [campo1 campo2 ... campon] | Utilizzata solo in modalità Multipla. |
| name_extension | stringa | Utilizzata solo in modalità Multipla. |
| add_as | Suffisso Prefix | Utilizzata solo in modalità Multipla. |
| ricodifica | stringa | Proprietà strutturata per i valori dei campi. |
| use_default | indicatore | Utilizza il valore di default. |
| default | stringa | Specificare un valore di default. |
| pick_list | [stringa stringa ... stringa] | Consente all'utente di importare un elenco di nuovi valori noti per popolare l'elenco a discesa nella tabella. |

Proprietà reordernode



Il nodo Ordina campi definisce l'ordine naturale utilizzato per visualizzare i campi a valle. Tale ordine incide sulla visualizzazione dei campi in vari contesti, quali tabelle, elenchi e Selettore di campo. Questa operazione risulta utile se si desidera rendere più visibili i campi interessanti in insiemi di dati di grandi dimensioni.

Esempio

```

node = stream.create("reorder", "My node")
node.setPropertyValue("mode", "Custom")
node.setPropertyValue("sort_by", "Storage")
node.setPropertyValue("ascending", False)
node.setPropertyValue("start_fields", ["Age", "Cholesterol"])
node.setPropertyValue("end_fields", ["Drug"])

```

Tabella 80. proprietà reordernode

| Proprietà reordernode | Data type | Descrizione proprietà |
|-----------------------|----------------|--|
| mode | Custom Auto | È possibile ordinare i valori automaticamente oppure specificare un ordine personalizzato. |

Tabella 80. proprietà reordernode (Continua)

| Proprietà reordernode | Data type | Descrizione proprietà |
|-----------------------|---------------------------------------|---|
| sort_by | Name Type Storage | |
| ascending | <i>indicatore</i> | |
| start_fields | <i>[campo1 campo2 ... campon]</i> | Dopo questi campi vengono inseriti altri nuovi campi. |
| end_fields | <i>[campo1 campo2 ... campon]</i> | Prima di questi campi vengono inseriti altri nuovi campi. |

Proprietà reprojectnode



In SPSS Modeler, gli elementi come le funzioni spaziali del Builder di espressioni, il nodo STP (Spatio-Temporal Prediction) ed il nodo Visualizzazione mappa utilizzano il sistema di coordinate proiettate. Utilizzare il nodo Riproiezione per modificare il sistema di coordinate dei dati importati che utilizzano un sistema di coordinate geografiche.

Tabella 81. Proprietà reprojectnode

| Proprietà reprojectnode | Data type | Descrizione proprietà |
|-------------------------|---------------------------------------|---|
| reproject_fields | <i>[campo1 campo2 ... campon]</i> | Elenca tutti i campi riproiettati. |
| reproject_type | Streamdefault Specify | Definisce come vengono riproiettati i campi. |
| coordinate_system | <i>stringa</i> | Il nome del sistema di coordinate da applicare ai campi. Esempio: set reprojectnode.coordinate_system = "WGS_1984_World_Mercator" |

Proprietà restructurenode



Il nodo Riorganizza converte un campo nominale o flag in un gruppo di campi in cui è possibile inserire i valori di un altro campo. Per esempio, dato un campo denominato *tipo di pagamento*, con valori di *credito*, *contanti* e *debito*, verrebbero creati tre nuovi campi (*credito*, *contanti*, *debito*), ognuno dei quali può contenere il valore del pagamento effettuato.

Esempio

```
node = stream.create("restructure", "My node")
node.setKeyedPropertyValue("fields_from", "Drug", ["drugA", "drugX"])
node.setPropertyValue("include_field_name", True)
node.setPropertyValue("value_mode", "OtherFields")
node.setPropertyValue("value_fields", ["Age", "BP"])
```

Tabella 82. proprietà restructurenode

| Proprietà restructurenode | Data type | Descrizione proprietà |
|---------------------------|--|--|
| fields_from | <i>[categoria categoria categoria] all</i> | |
| include_field_name | <i>indicatore</i> | Indica se utilizzare il nome del campo nel nome del campo riorganizzato. |

Tabella 82. proprietà restructurenode (Continua)

| Proprietà restructurenode | Data type | Descrizione proprietà |
|---------------------------|----------------------|--|
| value_mode | OtherFields Flags | Indica la modalità per specificare i valori dei campi riorganizzati. Con OtherFields, è necessario specificare quali campi utilizzare (vedere sezione seguente). Con Flags, i valori sono flag numerici. |
| value_fields | elenco | Necessario se value_mode è OtherFields. Specifica quali campi utilizzare come campi valore. |

Proprietà rfmanalysisnode



Il nodo Analisi RFM (Recency, Frequency, Monetary, Passato recente, Frequenza, Monetario) consente di determinare in modo quantitativo i clienti potenzialmente migliori verificando quanto tempo è trascorso dal loro ultimo acquisto (passato recente), con quale frequenza hanno effettuato acquisti (frequenza) e quanto hanno speso per tutte le transazioni (monetario).

Esempio

```
node = stream.create("rfmanalysis", "My node")
node.setPropertyValue("recency", "Recency")
node.setPropertyValue("frequency", "Frequency")
node.setPropertyValue("monetary", "Monetary")
node.setPropertyValue("tied_values_method", "Next")
node.setPropertyValue("recalculate_bins", "IfNecessary")
node.setPropertyValue("recency_thresholds", [1, 500, 800, 1500, 2000, 2500])
```

Tabella 83. proprietà rfmanalysisnode

| Proprietà rfmanalysisnode | Data type | Descrizione proprietà |
|---------------------------|-----------------|---|
| attualità | campo | Specifica il campo Passato recente, il cui valore può essere una data, un timestamp o un semplice numero. |
| frequency | campo | Specifica il campo Frequenza. |
| monetary | campo | Specifica il campo Monetario. |
| recency_bins | numero intero | Specifica il numero di bin di passato recente da generare. |
| recency_weight | numero | Specifica la ponderazione da applicare ai dati di passato recente. Il valore di default è 100. |
| frequency_bins | numero intero | Specifica il numero di bin di frequenza da generare. |
| frequency_weight | numero | Specifica la ponderazione da applicare ai dati di frequenza. Il valore di default è 10. |
| monetary_bins | numero intero | Specifica il numero di bin monetari da generare. |
| monetary_weight | numero | Specifica la ponderazione da applicare ai dati monetari. L'impostazione di default è 1. |
| tied_values_method | Next Current | Specifica quali dati relativi ai valori pari merito dei bin devono essere inseriti. |

Tabella 83. proprietà *rfanalysisnode* (Continua)

| Proprietà <i>rfanalysisnode</i> | Data type | Descrizione proprietà |
|---------------------------------|----------------------------------|---|
| recalculate_bins | Always IfNecessary | |
| add_outliers | <i>indicatore</i> | Disponibile solo se <i>recalculate_bins</i> è impostata su <i>IfNecessary</i> . Se la proprietà è impostata, i record di valore inferiore a quello del bin più basso vengono aggiunti a tale bin e quelli di valore superiore a quello del bin più alto vengono aggiunti a tale bin. |
| binned_field | Recency Frequency Monetary | |
| recency_thresholds | <i>valore valore</i> | Disponibile solo se <i>recalculate_bins</i> è impostata su <i>Always</i> . Specifica la soglia superiore e inferiore per i bin di passato recente. La soglia superiore di un bin viene utilizzata come soglia inferiore del bin successivo, esempio, [10 30 60] definirebbe due bin, il primo con soglia superiore e inferiore rispettivamente di 10 e 30 e il secondo con soglia superiore e inferiore rispettivamente di 30 e 60. |
| frequency_thresholds | <i>valore valore</i> | Disponibile solo se <i>recalculate_bins</i> è impostata su <i>Always</i> . |
| monetary_thresholds | <i>valore valore</i> | Disponibile solo se <i>recalculate_bins</i> è impostata su <i>Always</i> . |

Proprietà *settoflagnode*



Il nodo Crea flag crea campi flag in base ai valori categoriali di uno o più campi nominali.

Esempio

```
node = stream.create("settoflag", "My node")
node.setKeyedPropertyValue("fields_from", "Drug", ["drugA", "drugX"])
node.setPropertyValue("true_value", "1")
node.setPropertyValue("false_value", "0")
node.setPropertyValue("use_extension", True)
node.setPropertyValue("extension", "Drug_Flag")
node.setPropertyValue("add_as", "Suffix")
node.setPropertyValue("aggregate", True)
node.setPropertyValue("keys", ["Cholesterol"])
```

Tabella 84. proprietà *settoflagnode*

| Proprietà <i>settoflagnode</i> | Data type | Descrizione proprietà |
|--------------------------------|---|--|
| fields_from | [<i>categoria categoria categoria</i>] all | |
| true_value | <i>stringa</i> | Specifica il valore vero utilizzato dal nodo durante l'impostazione di un flag. L'impostazione di default è T. |

Tabella 84. proprietà settoflagnode (Continua)

| Proprietà settoflagnode | Data type | Descrizione proprietà |
|-------------------------|------------------|--|
| false_value | stringa | Specifica il valore falso utilizzato dal nodo durante l'impostazione di un flag. L'impostazione di default è F. |
| use_extension | indicatore | Utilizzare un'estensione come suffisso o prefisso al nuovo campo flag. |
| extension | stringa | |
| add_as | Suffix Prefix | Specifica se l'estensione viene aggiunta come suffisso o prefisso. |
| aggregate | indicatore | Raggruppa i record in base ai campi chiave. Tutti i campi flag di un gruppo vengono attivati se un record è impostato su vero. |
| keys | elenco | Campi chiave. |

Proprietà statisticstransformnode



Il nodo Trasformazioni Statistics esegue una selezione di comandi di sintassi IBM SPSS Statistics rispetto alle sorgenti dati in IBM SPSS Modeler. Questo nodo richiede una copia di IBM SPSS Statistics con regolare licenza.

Le proprietà di questo nodo sono descritte in “Proprietà statisticstransformnode” a pagina 311.

Proprietà timeintervalsnode



Il nodo Intervalli di tempo specifica intervalli e, se necessario, crea etichette per la modellazione di dati di serie temporali. Se i valori non sono spazati in modo uniforme, il nodo può riempire o aggregare i valori in base alle proprie esigenze per generare un intervallo uniforme tra i record.

Esempio

```
node = stream.create("timeintervals", "My node")
node.setPropertyValue("interval_type", "SecondsPerDay")
node.setPropertyValue("days_per_week", 4)
node.setPropertyValue("week_begins_on", "Tuesday")
node.setPropertyValue("hours_per_day", 10)
node.setPropertyValue("day_begins_hour", 7)
node.setPropertyValue("day_begins_minute", 5)
node.setPropertyValue("day_begins_second", 17)
node.setPropertyValue("mode", "Label")
node.setPropertyValue("year_start", 2005)
node.setPropertyValue("month_start", "January")
node.setPropertyValue("day_start", 4)
node.setKeyedPropertyValue("pad", "AGE", "MeanOfRecentPoints")
node.setPropertyValue("agg_mode", "Specify")
node.setPropertyValue("agg_set_default", "Last")
```

Tabella 85. proprietà *timeintervalnode*.

| Proprietà <i>timeintervalnode</i> | Data type | Descrizione proprietà |
|-----------------------------------|---|---|
| interval_type | None Periodi CyclicPeriods Years Trimestri Mesi DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic | |
| mode | Label Create | Specifica se si desidera identificare i record consecutivamente o creare le serie in base a un campo data, timestamp o ora specifico. |
| campo | <i>campo</i> | Quando si creano le serie a partire dai dati, specifica il campo che indica la data o l'ora di ciascun record. |
| period_start | <i>numero intero</i> | Specifica l'intervallo iniziale dei periodi o dei periodi ciclici. |
| cycle_start | <i>numero intero</i> | Ciclo iniziale dei periodi ciclici. |
| year_start | <i>numero intero</i> | Per i tipi di intervalli, ove applicabile, anno nel quale cade il primo intervallo. |
| quarter_start | <i>numero intero</i> | Per i tipi di intervalli, ove applicabile, trimestre nel quale cade il primo intervallo. |
| month_start | gennaio febbraio marzo aprile maggio giugno luglio agosto settembre ottobre novembre Dicembre | |
| day_start | <i>numero intero</i> | |
| hour_start | <i>numero intero</i> | |
| minute_start | <i>numero intero</i> | |
| second_start | <i>numero intero</i> | |
| periods_per_cycle | <i>numero intero</i> | Per i periodi ciclici, numero entro ciascun ciclo. |

Tabella 85. proprietà *timeintervalnode* (Continua).

| Proprietà <i>timeintervalnode</i> | Data type | Descrizione proprietà |
|-----------------------------------|--|--|
| <i>fiscal_year_begins</i> | gennaio febbraio marzo aprile maggio giugno luglio agosto settembre ottobre novembre Dicembre | Per gli intervalli trimestrali, specifica il mese nel quale inizia l'anno fiscale. |
| <i>week_begins_on</i> | Sunday Monday Tuesday Wednesday Thursday Friday Saturday Sunday | Per gli intervalli periodici (giorni alla settimana, ore al giorno, minuti al giorno e secondi al giorno), specifica il giorno in cui inizia la settimana. |
| <i>day_begins_hour</i> | <i>numero intero</i> | Per gli intervalli periodici (ore al giorno, minuti al giorno e secondi al giorno), specifica l'ora in cui inizia il giorno. Può essere utilizzata in combinazione con <i>day_begins_minute</i> e <i>day_begins_second</i> per specificare un orario esatto, per esempio 8:05:01. Vedere l'esempio di utilizzo seguente. |
| <i>day_begins_minute</i> | <i>numero intero</i> | Per gli intervalli periodici (ore al giorno, minuti al giorno e secondi al giorno), specifica il minuto in cui inizia il giorno (per esempio, il 5 in 8:05). |
| <i>day_begins_second</i> | <i>numero intero</i> | Per gli intervalli periodici (ore al giorno, minuti al giorno e secondi al giorno), specifica il secondo in cui inizia il giorno (per esempio, il 17 in 8:05:17). |
| <i>days_per_week</i> | <i>numero intero</i> | Per gli intervalli periodici (giorni alla settimana, ore al giorno, minuti al giorno e secondi al giorno), specifica il numero di giorni per settimana. |
| <i>hours_per_day</i> | <i>numero intero</i> | Per gli intervalli periodici (ore al giorno, minuti al giorno e secondi al giorno), specifica il numero di ore del giorno. |
| <i>interval_increment</i> | 1 2 3 4 5 6 10 15 20 30 | Per i minuti al giorno e i secondi al giorno, specifica il numero di minuti o secondi da incrementare per ogni record. |
| <i>field_name_extension</i> | <i>stringa</i> | |

Tabella 85. proprietà timeintervalnode (Continua).

| Proprietà timeintervalnode | Data type | Descrizione proprietà |
|--------------------------------|---|---|
| field_name_extension_as_prefix | indicatore | |
| date_format | "DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY | |
| time_format | "HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM." "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S" | |
| aggregate | Mean Sum Mode Min Max First Last TrueIfAnyTrue | Specifica il metodo di aggregazione per un campo. |
| pad | Blank MeanOfRecentPoints True False | Specifica il metodo di padding per un campo. |

Tabella 85. proprietà *timeintervalsnode* (Continua).

| Proprietà <i>timeintervalsnode</i> | Data type | Descrizione proprietà |
|--|--|--|
| <code>agg_mode</code> | All Specify | Specifica se aggregare o riempire tutti i campi con le funzioni di default quando necessario oppure specifica i campi e le funzioni da utilizzare. |
| <code>agg_range_default</code> | Mean Sum Mode Min Max | Specifica la funzione di default da utilizzare durante l'aggregazione dei campi continui. |
| <code>agg_set_default</code> | Mode First Last | Specifica la funzione di default da utilizzare durante l'aggregazione dei campi nominali. |
| <code>agg_flag_default</code> | TrueIfAnyTrue Mode First Last | |
| <code>pad_range_default</code> | Blank MeanOfRecentPoints | Specifica la funzione di default da utilizzare durante il padding dei campi continui. |
| <code>pad_set_default</code> | Blank MostRecentValue | |
| <code>pad_flag_default</code> | Blank True False | |
| <code>max_records_to_create</code> | <i>numero intero</i> | Specifica il numero massimo di record da creare durante il riempimento delle serie. |
| <code>estimation_from_beginning</code> | <i>indicatore</i> | |
| <code>estimation_to_end</code> | <i>indicatore</i> | |
| <code>estimation_start_offset</code> | <i>numero intero</i> | |
| <code>estimation_num_holdouts</code> | <i>numero intero</i> | |
| <code>create_future_records</code> | <i>indicatore</i> | |
| <code>num_future_records</code> | <i>numero intero</i> | |
| <code>create_future_field</code> | <i>indicatore</i> | |
| <code>future_field_name</code> | <i>stringa</i> | |

Proprietà *transposenode*



Il nodo Trasponi scambia i dati delle righe e delle colonne in modo da trasporre i campi in record e i record in campi.

Esempio

```
node = stream.create("transpose", "My node")
node.setPropertyValue("transposed_names", "Read")
node.setPropertyValue("read_from_field", "TimeLabel")
node.setPropertyValue("max_num_fields", "1000")
node.setPropertyValue("id_field_name", "ID")
```

Tabella 86. proprietà transposenode

| Proprietà transposenode | Data type | Descrizione proprietà |
|-------------------------|-----------------------------|--|
| transposed_names | Prefix Read | È possibile generare automaticamente i nomi dei nuovi campi in base a un prefisso specificato, oppure leggerli da un campo esistente nei dati. |
| prefix | stringa | |
| num_new_fields | numero intero | Quando si utilizza un prefisso, specifica il numero massimo di nuovi campi da creare. |
| read_from_field | campo | Campo dal quale vengono letti i nomi. Deve essere un campo istanziato, altrimenti si verificherà un errore al momento dell'esecuzione del nodo. |
| max_num_fields | numero intero | Quando si leggono i nomi da un campo, specifica un limite superiore per evitare di creare un numero eccessivamente elevato di campi. |
| transpose_type | Numeric String Custom | Per default vengono trasposti solo i campi continui (intervalli numerici), ma è possibile scegliere un sottoinsieme personalizzato di campi numerici oppure trasporre tutti i campi stringa. |
| transpose_fields | elenco | Specifica i campi da trasporre quando viene utilizzata l'opzione Personalizzato. |
| id_field_name | campo | |

Proprietà typenode



Il nodo Tipo specifica proprietà e metadati di campo. Per esempio, è possibile specificare un livello di misurazione (continuo, nominale, ordinale o flag) per ogni campo, impostare opzioni relative alla gestione dei valori mancanti e dei valori null di sistema, impostare il ruolo di un campo per la modellazione, specificare le etichette di campo e valore e specificare i valori per un campo.

Esempio

```
node = stream.createAt("type", "My node", 50, 50)
node.setKeyedPropertyValue("check", "Cholesterol", "Coerce")
node.setKeyedPropertyValue("direction", "Drug", "Input")
node.setKeyedPropertyValue("type", "K", "Range")
node.setKeyedPropertyValue("values", "Drug", ["drugA", "drugB", "drugC", "drugD", "drugX",
"drugY", "drugZ"])
node.setKeyedPropertyValue("null_missing", "BP", False)
node.setKeyedPropertyValue("whitespace_missing", "BP", False)
node.setKeyedPropertyValue("description", "BP", "Blood Pressure")
node.setKeyedPropertyValue("value_labels", "BP", [["HIGH", "High Blood Pressure"],
["NORMAL", "normal blood pressure"]])
```

Si noti che in alcuni casi potrebbe essere necessario istanziare il nodo Tipo per consentire il corretto funzionamento di altri nodi, quali la proprietà `fields_from` del nodo Crea flag. È possibile collegare semplicemente un nodo Tabella ed eseguirlo per istanziare i campi:

```
tablenode = stream.createAt("table", "Table node", 150, 50)
stream.link(node, tablenode)
tablenode.run(None)
stream.delete(tablenode)
```


Tabella 87. proprietà typenode.

| Proprietà typenode | Data type | Descrizione proprietà |
|--------------------|--|---|
| direction | Input Target Both None Partition Split Frequency RecordID | Proprietà basata su chiavi per i ruoli del campo. Nota: i valori In e Out sono obsoleti. Nelle versioni future potrebbero non essere più supportati. |
| type | Range Flag Set Typeless Discrete OrderedSet Default | Livello di misurazione del campo (precedentemente definito "tipo" di campo). Se si imposta il type su Default, eventualmente Se value_mode è impostato su Pass o Read, l'impostazione di type non influirà su value_mode. Nota: I tipi di dati utilizzati internamente sono diversi da quelli visibili nel nodo tipologia. La corrispondenza è riportata di seguito: Range -> Continuous Set -> Nominal OrderedSet -> Ordinal Discrete- > Categorical |
| storage | Unknown String Integer Real Time Date Timestamp | Proprietà basata su chiavi in sola lettura per il tipo di archiviazione del campo. |
| check | None Annulla Coerce Discard Warn Abort | Proprietà basata su chiavi per il controllo del tipo di campo e dell'intervallo. |
| values | [valore valore] | Per un campo continuo, il primo valore corrisponde al minimo e l'ultimo valore al massimo. Per i campi nominali, specificare tutti i valori. Nel caso dei campi flag, il primo valore rappresenta <i>falso</i> e l'ultimo valore rappresenta <i>vero</i> . L'impostazione automatica di questa proprietà consente di impostare la proprietà value_mode su Specify. |
| value_mode | Read Pass Leggi+ Current Specify | Determina la modalità di impostazione dei valori. Si noti che non è possibile impostare questa proprietà direttamente su Specify. Per utilizzare valori specifici, impostare la proprietà values. |
| extend_values | indicatore | Viene applicato quando value_mode è impostata su Read. Per aggiungere valori appena letti a eventuali valori esistenti per il campo, impostare su T. Per scartare i valori esistenti e sostituirli con i valori appena letti, impostare su F. |
| enable_missing | indicatore | Se impostato su V, attiva la registrazione dei valori mancanti per il campo. |

Tabella 87. proprietà typenode (Continua).

| Proprietà typenode | Data type | Descrizione proprietà |
|--------------------|--|--|
| missing_values | [valore valore ...] | Specifica i valori dei dati che indicano dati mancanti. |
| range_missing | indicatore | Specifica se viene definito un intervallo di valori mancanti (vuoti) per un campo. |
| missing_lower | stringa | Se range_missing è impostata su vero, specifica il limite inferiore dell'intervallo di valori mancanti. |
| missing_upper | stringa | Se range_missing è impostata su vero, specifica il limite superiore dell'intervallo di valori mancanti. |
| null_missing | indicatore | Se impostata su T, i valori null (valori non definiti, visualizzati come \$null\$ nel software) vengono considerati valori mancanti. |
| whitespace_missing | indicatore | Se impostata su T, i valori contenenti solo uno spazio vuoto (spazi, tabulazioni e nuove righe) vengono considerati valori mancanti. |
| descrizione | stringa | Specifica la descrizione di un campo. |
| value_labels | [[Valore EtichettaStringa] [Valore EtichettaStringa] ...] | Utilizzata per specificare etichette per coppie di valori. |
| display_places | integer | Imposta il numero di decimali del campo per la visualizzazione (valida solo per campi con archiviazione di tipo REAL). Se viene specificato il valore -1, verrà utilizzata l'impostazione di default del flusso. |
| export_places | integer | Imposta il numero di decimali del campo per l'esportazione (valida solo per campi con archiviazione di tipo REAL). Se viene specificato il valore -1, verrà utilizzata l'impostazione di default del flusso. |
| decimal_separator | DEFAULT PERIOD COMMA | Imposta il separatore decimale per il campo (valida solo per i campi con archiviazione di tipo REAL). |

Tabella 87. proprietà typenode (Continua).

| Proprietà typenode | Data type | Descrizione proprietà |
|--------------------|---|--|
| date_format | "DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY | Imposta il formato di data per il campo (valida solo per campi con archiviazione di tipo DATE o TIMESTAMP). |
| time_format | "HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM." "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S" | Imposta il formato di ora per il campo (valida solo per campi con archiviazione di tipo TIME o TIMESTAMP). |
| number_format | DEFAULT STANDARD SCIENTIFIC CURRENCY | Imposta il formato di visualizzazione dei numeri per il campo. |
| standard_places | <i>integer</i> | Imposta il numero di decimali del campo per la visualizzazione in formato standard. Se viene specificato il valore -1, verrà utilizzata l'impostazione di default del flusso. Si noti che lo slot display_places esistente consente di ottenere lo stesso risultato, tuttavia tale configurazione è obsoleta in questa versione. |
| scientific_places | <i>integer</i> | Imposta il numero di decimali del campo per la visualizzazione in formato scientifico. Se viene specificato il valore -1, verrà utilizzata l'impostazione di default del flusso. |

Tabella 87. proprietà typenode (Continua).

| Proprietà typenode | Data type | Descrizione proprietà |
|-----------------------|---|--|
| currency_places | <i>integer</i> | Imposta il numero di decimali del campo per la visualizzazione in formato valuta. Se viene specificato il valore -1, verrà utilizzata l'impostazione di default del flusso. |
| grouping_symbol | DEFAULT NONE LOCALE PERIOD COMMA SPACE | Imposta il simbolo di raggruppamento per il campo. |
| column_width | <i>integer</i> | Imposta la larghezza delle colonne per il campo. Se viene specificato il valore -1, la larghezza delle colonne verrà impostata su Auto. |
| justify | AUTO CENTER LEFT RIGHT | Imposta la giustificazione delle colonne per il campo. |
| measure_type | Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL | Questa proprietà basata su chiavi è simile a type in quanto può essere utilizzata per definire la misurazione associata al campo. La differenza consiste nel fatto che, negli script Python, alla funzione setter può essere passato anche uno dei valori MeasureType mentre getter viene restituito sempre sui valori MeasureType. |
| collection_measure | Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS | Per i campi di raccolta (elenchi con profondità uguale a 0), questa proprietà basata su chiavi definisce il tipo di misurazione associato ai valori sottostanti. |
| geo_type | Point MultiPoint LineString MultiLineString Polygon MultiPolygon | Per i campi geospaziali, questa proprietà basata su chiavi definisce il tipo di oggetto geospaziale rappresentato da questo campo. Questo valore deve essere coerente con la profondità di elenco dei valori. |
| has_coordinate_system | <i>booleano</i> | Per i campi geospaziali, questa proprietà definisce se questo campo dispone di un sistema di coordinate |
| coordinate_system | <i>stringa</i> | Per i campi geospaziali, questa proprietà basata su chiavi definisce il sistema di coordinate per questo campo. |
| custom_storage_type | Unknown / MeasureType.UNKNOWN String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP List / MeasureType.LIST | Questa proprietà basata su chiavi è simile a custom_storage in quanto può essere utilizzata per definire l'archiviazione di sostituzione per il campo. La differenza consiste nel fatto che, negli script Python, alla funzione setter può essere passato anche uno dei valori StorageType mentre getter viene restituito sempre sui valori StorageType. |

Tabella 87. proprietà typenode (Continua).

| Proprietà typenode | Data type | Descrizione proprietà |
|--------------------------|--|---|
| custom_list_storage_type | String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP | Per i campi di elenco, questa proprietà basata su chiavi specifica il tipo di archiviazione dei valori sottostanti. |
| custom_list_depth | <i>integer</i> | Per i campi di elenco, questa proprietà basata su chiavi specifica la profondità del campo |

Capitolo 12. Proprietà dei nodi Grafici

Proprietà comuni dei nodi Grafici

In questa sezione vengono illustrate le proprietà disponibili per i nodi Grafici, incluse le proprietà comuni e quelle specifiche per ogni tipo di nodo.

Tabella 88. Proprietà comuni dei nodi Grafici

| Proprietà comuni dei nodi Grafici | Data type | Descrizione proprietà |
|-----------------------------------|---|---|
| title | stringa | Specifica il titolo. Esempio: "Questo è un titolo". |
| caption | stringa | Specifica la didascalia. Esempio: "Questa è una didascalia". |
| output_mode | Screen File | Specifica se l'output del nodo Grafico viene visualizzato o scritto su un file. |
| output_format | BMP JPEG PNG HTML output (.cou) | Specifica il tipo di output. Il tipo di output consentito varia da nodo a nodo. |
| full_filename | stringa | Specifica il percorso di destinazione e il nome file per l'output generato dal nodo Grafico. |
| use_graph_size | indicatore | Controlla se le dimensioni del grafico vengono indicate esplicitamente, utilizzando le proprietà di larghezza e altezza seguenti. Influisce solo sui grafici per i quali viene visualizzato l'output su schermo. Non disponibile per il nodo distribuzione. |
| graph_width | numero | Quando use_graph_size è Vero, imposta la larghezza del grafico in pixel. |
| graph_height | numero | Quando use_graph_size è Vero, imposta l'altezza del grafico in pixel. |

Disattivazione dei campi facoltativi

È possibile disattivare i campi facoltativi, per esempio un campo di sovrapposizione per i plot, impostando il valore della proprietà su " " (stringa vuota), come illustrato nell'esempio seguente.

```
plotnode.setPropertyValue("color_field", "")
```

Specifica dei colori

Il colore dei titoli, delle didascalie, degli sfondi e delle etichette può essere specificato utilizzando le stringhe esadecimali che iniziano con il simbolo del cancelletto (#). Per esempio, per impostare uno sfondo di colore azzurro per il grafico è possibile utilizzare l'istruzione seguente:

```
mygraphnode.setPropertyValue("graph_background", "#87CEEB")
```

Le prime due cifre, 87, specificano il contenuto rosso; le due cifre intermedie, CE, specificano il contenuto verde e le ultime due cifre, EB, specificano il contenuto blu. Ogni cifra può avere un valore compreso nell'intervallo 0-9 o A-F. Utilizzando la combinazione di questi valori è possibile specificare un colore RGB (red-green-blue).

Nota: quando si specificano colori RGB, è possibile utilizzare il selettore di campo disponibile nell'interfaccia utente per definire il codice di colore corretto. È sufficiente posizionare il puntatore del mouse sul colore per visualizzare una descrizione contenente le informazioni desiderate.

Proprietà collectionnode



Il nodo Raccolta mostra la distribuzione dei valori di un campo numerico in relazione ai valori di un altro, ovvero crea grafici simili a istogrammi. È utile per illustrare una variabile o un campo i cui valori vengono modificati nel tempo. La grafica 3-D consente inoltre di includere un asse simbolico che visualizza le distribuzioni per categoria.

Esempio

```
node = stream.create("collection", "My node")
# "Plot" tab
node.setPropertyValue("three_D", True)
node.setPropertyValue("collect_field", "Drug")
node.setPropertyValue("over_field", "Age")
node.setPropertyValue("by_field", "BP")
node.setPropertyValue("operation", "Sum")
# "Overlay" section
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "")
# "Options" tab
node.setPropertyValue("range_mode", "Automatic")
node.setPropertyValue("range_min", 1)
node.setPropertyValue("range_max", 100)
node.setPropertyValue("bins", "ByNumber")
node.setPropertyValue("num_bins", 10)
node.setPropertyValue("bin_width", 5)
```

Tabella 89. proprietà collectionnode

| Proprietà collectionnode | Data type | Descrizione proprietà |
|--------------------------|-----------------------------------|-----------------------|
| over_field | campo | |
| over_label_auto | indicatore | |
| over_label | stringa | |
| collect_field | campo | |
| collect_label_auto | indicatore | |
| collect_label | stringa | |
| three_D | indicatore | |
| by_field | campo | |
| by_label_auto | indicatore | |
| by_label | stringa | |
| operation | Sum Mean Min Max SDev | |
| color_field | stringa | |
| panel_field | stringa | |
| animation_field | stringa | |

Tabella 89. proprietà collectionnode (Continua)

| Proprietà collectionnode | Data type | Descrizione proprietà |
|--------------------------|--------------------------|--|
| range_mode | Automatic UserDefined | |
| range_min | numero | |
| range_max | numero | |
| bins | ByNumber ByWidth | |
| num_bins | numero | |
| bin_width | numero | |
| use_grid | indicatore | |
| graph_background | colore | I colori standard dei grafici sono descritti all'inizio di questa sezione. |
| page_background | colore | I colori standard dei grafici sono descritti all'inizio di questa sezione. |

Proprietà distributionnode



Il nodo distribuzione mostra l'occorrenza di valori simbolici (categoriali), per esempio tipo o genere di ipoteca. In genere è possibile utilizzare un nodo distribuzione per mostrare squilibri nei dati, che possono essere successivamente corretti con un nodo bilanciamento prima di creare un modello.

Esempio

```
node = stream.create("distribution", "My node")
# "Plot" tab
node.setPropertyValue("plot", "Flags")
node.setPropertyValue("x_field", "Age")
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("normalize", True)
node.setPropertyValue("sort_mode", "ByOccurrence")
node.setPropertyValue("use_proportional_scale", True)
```

Tabella 90. proprietà distributionnode

| Proprietà distributionnode | Data type | Descrizione proprietà |
|----------------------------|----------------------------|-----------------------|
| plot | SelectedFields Flags | |
| x_field | campo | |
| color_field | campo | Campo sovrapposto. |
| normalize | indicatore | |
| sort_mode | ByOccurrence Alphabetic | |
| use_proportional_scale | indicatore | |

Proprietà evaluationnode



Il nodo Valutazione facilita la valutazione e il confronto di modelli predittivi. Il grafico di valutazione mostra il comportamento dei modelli nella previsione di particolari risultati. Ordina i record in base al valore previsto e alla confidenza della previsione, quindi li suddivide in gruppi di uguale dimensione (**quantili**) e infine rappresenta il valore del criterio di business per ciascun quantile, dal più alto al più basso. I modelli multipli sono mostrati nel grafico come linee separate.

Esempio

```
node = stream.create("evaluation", "My node")
# "Plot" tab
node.setPropertyValue("chart_type", "Gains")
node.setPropertyValue("cumulative", False)
node.setPropertyValue("field_detection_method", "Name")
node.setPropertyValue("inc_baseline", True)
node.setPropertyValue("n_tile", "Deciles")
node.setPropertyValue("style", "Point")
node.setPropertyValue("point_type", "Dot")
node.setPropertyValue("use_fixed_cost", True)
node.setPropertyValue("cost_value", 5.0)
node.setPropertyValue("cost_field", "Na")
node.setPropertyValue("use_fixed_revenue", True)
node.setPropertyValue("revenue_value", 30.0)
node.setPropertyValue("revenue_field", "Age")
node.setPropertyValue("use_fixed_weight", True)
node.setPropertyValue("weight_value", 2.0)
node.setPropertyValue("weight_field", "K")
```

Tabella 91. proprietà evaluationnode.

| Proprietà evaluationnode | Data type | Descrizione proprietà |
|--------------------------|---|-----------------------|
| chart_type | Gains Response Lift Profit ROI ROC | |
| inc_baseline | <i>indicatore</i> | |
| field_detection_method | Metadata Name | |
| use_fixed_cost | <i>indicatore</i> | |
| cost_value | <i>numero</i> | |
| cost_field | <i>stringa</i> | |
| use_fixed_revenue | <i>indicatore</i> | |
| revenue_value | <i>numero</i> | |
| revenue_field | <i>stringa</i> | |
| use_fixed_weight | <i>indicatore</i> | |
| weight_value | <i>numero</i> | |
| weight_field | <i>campo</i> | |

Tabella 91. proprietà evaluationnode (Continua).

| Proprietà evaluationnode | Data type | Descrizione proprietà |
|--------------------------|---|-----------------------|
| n_tile | Quartiles Quintles Deciles Vingtiles Percentiles 1000-tiles | |
| cumulative | <i>indicatore</i> | |
| style | Line Point | |
| point_type | Rettangolo Punto Triangolo Esagono Segno più Pentagono Stella Farfallino Trattino orizzontale Trattino verticale Croce Fabbrica Casa Cattedrale Cupola Triangolo concavo Geoide Occhio di gatto Cuscino quadrato Rettangolo arrotondato Fan | |
| export_data | <i>indicatore</i> | |
| data_filename | <i>stringa</i> | |
| delimiter | <i>stringa</i> | |
| new_line | <i>indicatore</i> | |
| inc_field_names | <i>indicatore</i> | |
| inc_best_line | <i>indicatore</i> | |
| inc_business_rule | <i>indicatore</i> | |
| business_rule_condition | <i>stringa</i> | |
| plot_score_fields | <i>indicatore</i> | |
| score_fields | <i>[campo1 ... campoN]</i> | |
| target_field | <i>campo</i> | |
| use_hit_condition | <i>indicatore</i> | |
| hit_condition | <i>stringa</i> | |
| use_score_expression | <i>indicatore</i> | |
| score_expression | <i>stringa</i> | |
| caption_auto | <i>indicatore</i> | |

Proprietà graphboardnode



Il nodo Lavagna grafica offre numerosi tipi di grafici diversi in un unico nodo. Con questo nodo è possibile scegliere i campi di dati da esplorare e selezionare quindi un grafico fra quelli disponibili per i dati selezionati. Il nodo esclude automaticamente tutti i tipi di grafici non adatti ai campi selezionati.

Nota: se si imposta una proprietà non valida per il tipo di grafico, per esempio specificando `y_field` per un istogramma, tale proprietà viene ignorata.

Nota: Nell'interfaccia utente, nella scheda Dettagliato di molti tipi di grafici differenti, è presente un campo **Riepilogo**; tale campo non è attualmente supportato dagli script.

Esempio

```
node = stream.create("graphboard", "My node")
node.setPropertyValue("graph_type", "Line")
node.setPropertyValue("x_field", "K")
node.setPropertyValue("y_field", "Na")
```

Tabella 92. proprietà graphboardnode

| Proprietà graphboard | Data type | Descrizione proprietà |
|----------------------|---|--|
| graph_type | 2DDotplot 3DArea 3DBar 3DDensity 3DHistogram 3DPie 3DScatterplot Area ArrowMap Barre BarCounts BarCountsMap BarMap BinnedScatter Grafico a scatole Bubble ChoroplethMeans ChoroplethMedians ChoroplethSums ChoroplethValues ChoroplethCounts CoordinateMap CoordinateChoroplethMeans CoordinateChoroplethMedians CoordinateChoroplethSums CoordinateChoroplethValues CoordinateChoroplethCounts Dotplot Heatmap HexBinScatter Histogram Line LineChartMap LineOverlayMap Parallelo Percorso Pie PieCountMap PieCounts PieMap PointOverlayMap PolygonOverlayMap Nastro Grafico a dispersione SPL0M Superficie | Identifica il tipo di grafico. |
| x_field | campo | Specifica un'etichetta personalizzata per l'asse x. Disponibile solo per le etichette. |
| y_field | campo | Specifica un'etichetta personalizzata per l'asse y. Disponibile solo per le etichette. |
| z_field | campo | Utilizzata in alcuni grafici 3D. |
| color_field | campo | Utilizzata nelle mappe termiche. |
| size_field | campo | Utilizzata nei grafici a bolle. |
| categories_field | campo | |

Tabella 92. proprietà graphboardnode (Continua)

| Proprietà graphboard | Data type | Descrizione proprietà |
|-----------------------|-----------|---|
| values_field | campo | |
| rows_field | campo | |
| columns_field | campo | |
| fields | campo | |
| start_longitude_field | campo | Utilizzata con le frecce su una mappa di riferimento. |
| end_longitude_field | campo | |
| start_latitude_field | campo | |
| end_latitude_field | campo | |
| data_key_field | campo | Utilizzata in varie mappe. |
| panelrow_field | stringa | |
| panelcol_field | stringa | |
| animation_field | stringa | |
| longitude_field | campo | Utilizzata con le coordinate sulle mappe. |
| latitude_field | campo | |
| map_color_field | campo | |

Proprietà histogramnode



Il nodo Istogramma mostra l'occorrenza dei valori per i campi numerici. Viene spesso utilizzato per analizzare i dati prima delle manipolazioni e della creazione del modello. Come il nodo distribuzione, anche il nodo Istogramma viene frequentemente utilizzato per rivelare squilibri nei dati.

Esempio

```
node = stream.create("histogram", "My node")
# "Plot" tab
node.setPropertyValue("field", "Drug")
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "")
# "Options" tab
node.setPropertyValue("range_mode", "Automatic")
node.setPropertyValue("range_min", 1.0)
node.setPropertyValue("range_max", 100.0)
node.setPropertyValue("num_bins", 10)
node.setPropertyValue("bin_width", 10)
node.setPropertyValue("normalize", True)
node.setPropertyValue("separate_bands", False)
```

Tabella 93. proprietà histogramnode

| Proprietà histogramnode | Data type | Descrizione proprietà |
|-------------------------|-----------|-----------------------|
| campo | campo | |
| color_field | campo | |
| panel_field | campo | |
| animation_field | campo | |

Tabella 93. proprietà histogramnode (Continua)

| Proprietà histogramnode | Data type | Descrizione proprietà |
|-------------------------|--------------------------|---|
| range_mode | Automatic UserDefined | |
| range_min | numero | |
| range_max | numero | |
| bins | ByNumber ByWidth | |
| num_bins | numero | |
| bin_width | numero | |
| normalize | indicatore | |
| separate_bands | indicatore | |
| x_label_auto | indicatore | |
| x_label | stringa | |
| y_label_auto | indicatore | |
| y_label | stringa | |
| use_grid | indicatore | |
| graph_background | colore | I colori standard dei grafici sono descritti all'inizio di questa sezione. |
| page_background | colore | I colori standard dei grafici sono descritti all'inizio di questa sezione. |
| normal_curve | indicatore | Indica se la curva di distribuzione normale deve essere visualizzata nell'output. |

Proprietà multiplotnode



Un nodo Multiplot crea un grafico che consente di visualizzare più campi Y in un singolo campo X. I campi Y sono rappresentati come linee colorate e ognuno di essi equivale a un nodo Plot con lo Stile impostato su **Linea** e la Modalità X impostata su **Ordina**. I multiplot sono utili quando si desidera esplorare la fluttuazione di numerose variabili nel tempo.

Esempio

```
node = stream.create("multiplot", "My node")
# "Plot" tab
node.setPropertyValue("x_field", "Age")
node.setPropertyValue("y_fields", ["Drug", "BP"])
node.setPropertyValue("panel_field", "Sex")
# "Overlay" section
node.setPropertyValue("animation_field", "")
node.setPropertyValue("tooltip", "test")
node.setPropertyValue("normalize", True)
node.setPropertyValue("use_overlay_expr", False)
node.setPropertyValue("overlay_expression", "test")
node.setPropertyValue("records_limit", 500)
node.setPropertyValue("if_over_limit", "PlotSample")
```

Tabella 94. proprietà multiplotnode

| Proprietà multiplotnode | Data type | Descrizione proprietà |
|-------------------------|-----------|-----------------------|
| x_field | campo | |

Tabella 94. proprietà multiplotnode (Continua)

| Proprietà multiplotnode | Data type | Descrizione proprietà |
|-------------------------|-----------------------------------|--|
| y_fields | elenco | |
| panel_field | campo | |
| animation_field | campo | |
| normalize | indicatore | |
| use_overlay_expr | indicatore | |
| overlay_expression | stringa | |
| records_limit | numero | |
| if_over_limit | PlotBins PlotSample PlotAll | |
| x_label_auto | indicatore | |
| x_label | stringa | |
| y_label_auto | indicatore | |
| y_label | stringa | |
| use_grid | indicatore | |
| graph_background | colore | I colori standard dei grafici sono descritti all'inizio di questa sezione. |
| page_background | colore | I colori standard dei grafici sono descritti all'inizio di questa sezione. |

Proprietà plotnode



Il nodo Plot mostra la relazione tra campi numerici. È possibile creare un grafico utilizzando punti (un grafico a dispersione) oppure linee.

Esempio

```
node = stream.create("plot", "My node")
# "Plot" tab
node.setPropertyValue("three_D", True)
node.setPropertyValue("x_field", "BP")
node.setPropertyValue("y_field", "Cholesterol")
node.setPropertyValue("z_field", "Drug")
# "Overlay" section
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("size_field", "Age")
node.setPropertyValue("shape_field", "")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "BP")
node.setPropertyValue("transp_field", "")
node.setPropertyValue("style", "Point")
# "Output" tab
node.setPropertyValue("output_mode", "File")
node.setPropertyValue("output_format", "JPEG")
node.setPropertyValue("full_filename", "C:/temp/graph_output/plot_output.jpeg")
```


Tabella 95. proprietà plotnode.

| Proprietà plotnode | Data type | Descrizione proprietà |
|--------------------|---|--|
| x_field | campo | Specifica un'etichetta personalizzata per l'asse x. Disponibile solo per le etichette. |
| y_field | campo | Specifica un'etichetta personalizzata per l'asse y. Disponibile solo per le etichette. |
| three_D | indicatore | Specifica un'etichetta personalizzata per l'asse y. Disponibile solo per le etichette nei grafici 3-D. |
| z_field | campo | |
| color_field | campo | Campo sovrapposto. |
| size_field | campo | |
| shape_field | campo | |
| panel_field | campo | Specifica un campo nominale o flag da utilizzare per creare un grafico distinto per ciascuna categoria. I grafici vengono rappresentati in pannelli in un'unica finestra di output. |
| animation_field | campo | Specifica un campo nominale o flag per rappresentare le categorie per i valori dei dati creando una serie di grafici visualizzati in sequenza tramite l'animazione. |
| transp_field | campo | Specifica un campo per rappresentare le categorie per i valori dei dati utilizzando un livello di trasparenza diverso per ciascuna categoria. Non disponibile per i grafici a linee. |
| overlay_type | None Smoother Function | Specifica se viene visualizzata una funzione sovrapposta o un livellamento di LOESS. |
| overlay_expression | stringa | Specifica l'espressione utilizzata quando overlay_type è impostata su Function. |
| style | Point Line | |
| point_type | Rettangolo Punto Triangolo Esagono Segno più Pentagono Stella Farfallino Trattino orizzontale Trattino verticale Croce Fabbrica Casa Cattedrale Cupola Triangolo concavo Geoide Occhio di gatto Cuscino quadrato Rettangolo arrotondato Fan | |
| x_mode | Sort Overlay AsRead | |

Tabella 95. proprietà plotnode (Continua).

| Proprietà plotnode | Data type | Descrizione proprietà |
|--------------------|-----------------------------------|--|
| x_range_mode | Automatic UserDefined | |
| x_range_min | numero | |
| x_range_max | numero | |
| y_range_mode | Automatic UserDefined | |
| y_range_min | numero | |
| y_range_max | numero | |
| z_range_mode | Automatic UserDefined | |
| z_range_min | numero | |
| z_range_max | numero | |
| jitter | indicatore | |
| records_limit | numero | |
| if_over_limit | PlotBins PlotSample PlotAll | |
| x_label_auto | indicatore | |
| x_label | stringa | |
| y_label_auto | indicatore | |
| y_label | stringa | |
| z_label_auto | indicatore | |
| z_label | stringa | |
| use_grid | indicatore | |
| graph_background | colore | I colori standard dei grafici sono descritti all'inizio di questa sezione. |
| page_background | colore | I colori standard dei grafici sono descritti all'inizio di questa sezione. |
| use_overlay_expr | indicatore | Reso obsoleto a favore di overlay_type. |

Proprietà timeplotnode



Il nodo del grafico temporale visualizza uno o più insiemi di dati di serie temporali. In genere, si utilizza prima un nodo Intervalli di tempo per creare un campo *EtichettaTempo*, che viene utilizzato per attribuire un'etichetta all'asse *x*.

Esempio

```
node = stream.create("timeplot", "My node")
node.setPropertyValue("y_fields", ["sales", "men", "women"])
node.setPropertyValue("panel", True)
node.setPropertyValue("normalize", True)
node.setPropertyValue("line", True)
node.setPropertyValue("smoother", True)
```

```

node.setPropertyValue("use_records_limit", True)
node.setPropertyValue("records_limit", 2000)
# Appearance settings
node.setPropertyValue("symbol_size", 2.0)

```

Tabella 96. proprietà *timeplotnode*.

| Proprietà <i>timeplotnode</i> | Data type | Descrizione proprietà |
|-------------------------------|---|--|
| plot_series | Series Models | |
| use_custom_x_field | <i>indicatore</i> | |
| x_field | <i>campo</i> | |
| y_fields | <i>elenco</i> | |
| panel | <i>indicatore</i> | |
| normalize | <i>indicatore</i> | |
| line | <i>indicatore</i> | |
| points | <i>indicatore</i> | |
| point_type | Rettangolo Punto Triangolo Esagono Segno più Pentagono Stella Farfallino Trattino orizzontale Trattino verticale Croce Fabbrica Casa Cattedrale Cupola Triangolo concavo Geoide Occhio di gatto Cuscino quadrato Rettangolo arrotondato Fan | |
| smoother | <i>indicatore</i> | È possibile aggiungere livellamenti al plot solo se si imposta <i>panel</i> su True. |
| use_records_limit | <i>indicatore</i> | |
| records_limit | <i>numero intero</i> | |
| symbol_size | <i>numero</i> | Specifica la dimensione di un simbolo. |
| panel_layout | Horizontal Vertical | |

Proprietà *webnode*



Il nodo Web illustra l'intensità della relazione tra valori di due o più campi simbolici (categoriali). Il grafico utilizza linee di spessore diverso per indicare l'intensità della connessione. Un nodo Web può essere utilizzato, per esempio, per analizzare la relazione tra l'acquisto di vari oggetti in un sito di e-commerce.

Esempio

```

node = stream.create("web", "My node")
# "Plot" tab
node.setPropertyValue("use_directed_web", True)
node.setPropertyValue("to_field", "Drug")
node.setPropertyValue("fields", ["BP", "Cholesterol", "Sex", "Drug"])
node.setPropertyValue("from_fields", ["BP", "Cholesterol", "Sex"])
node.setPropertyValue("true_flags_only", False)
node.setPropertyValue("line_values", "Absolute")
node.setPropertyValue("strong_links_heavier", True)
# "Options" tab
node.setPropertyValue("max_num_links", 300)
node.setPropertyValue("links_above", 10)
node.setPropertyValue("num_links", "ShowAll")
node.setPropertyValue("discard_links_min", True)
node.setPropertyValue("links_min_records", 5)
node.setPropertyValue("discard_links_max", True)
node.setPropertyValue("weak_below", 10)
node.setPropertyValue("strong_above", 19)
node.setPropertyValue("link_size_continuous", True)
node.setPropertyValue("web_display", "Circular")

```

Tabella 97. proprietà webnode

| Proprietà webnode | Data type | Descrizione proprietà |
|----------------------|---|--|
| use_directed_web | indicatore | |
| fields | elenco | |
| to_field | campo | |
| from_fields | elenco | |
| true_flags_only | indicatore | |
| line_values | Absolute OverallPct PctLarger PctSmaller | |
| strong_links_heavier | indicatore | |
| num_links | ShowMaximum ShowLinksAbove ShowAll | |
| max_num_links | numero | |
| links_above | numero | |
| discard_links_min | indicatore | |
| links_min_records | numero | |
| discard_links_max | indicatore | |
| links_max_records | numero | |
| weak_below | numero | |
| strong_above | numero | |
| link_size_continuous | indicatore | |
| web_display | Circular Network Directed Grid | |
| graph_background | colore | I colori standard dei grafici sono descritti all'inizio di questa sezione. |

Tabella 97. proprietà webnode (Continua)

| Proprietà webnode | Data type | Descrizione proprietà |
|-------------------|---------------|--|
| symbol_size | <i>numero</i> | Specifica la dimensione di un simbolo. |

Capitolo 13. Proprietà dei nodi Modelli

Proprietà comuni nodi modellazione

Le seguenti proprietà sono comuni ad alcuni o a tutti i nodi Modelli. Le eventuali eccezioni sono segnalate, ove necessario, nella documentazione relativa ai singoli nodi Modelli.

Tabella 98. Proprietà comuni nodo modellazione

| Proprietà | Valori | Descrizione proprietà |
|---------------------------|---|---|
| custom_fields | <i>indicatore</i> | Se vera, consente di specificare i campi obiettivo, di input e di altro tipo per il nodo corrente. Se falsa, vengono utilizzate le impostazioni correnti di un nodo Tipo a monte. |
| obiettivo o targets | <i>campo</i> o [<i>campo1 ... campoN</i>] | Specifica un unico campo obiettivo o più campi obiettivo, a seconda del tipo di modello. |
| inputs | [<i>campo1 ... campoN</i>] | I campi di input o predittore utilizzati dal modello. |
| partition | <i>campo</i> | |
| use_partitioned_data | <i>indicatore</i> | Se è definito un campo partizione, questa opzione garantisce che per la creazione del modello verranno utilizzati solo i dati della partizione di addestramento. |
| use_split_data | <i>indicatore</i> | |
| splits | [<i>campo1 ... campoN</i>] | Specifica il campo o i campi da usare per la creazione di modelli suddivisi. Efficace solo se <i>use_split_data</i> è impostato su True. |
| use_frequency | <i>indicatore</i> | I campi peso e frequenza vengono utilizzati da determinati modelli, come riportato per ogni tipo di modello. |
| frequency_field | <i>campo</i> | |
| use_weight | <i>indicatore</i> | |
| weight_field | <i>campo</i> | |
| use_model_name | <i>indicatore</i> | |
| model_name | <i>stringa</i> | Nome personalizzato per il nuovo modello. |
| mode | Simple Expert | |

Proprietà anomalydetectionnode



Il nodo Rilevamento anomalie identifica casi insoliti, o valori anomali, non conformi a schemi di dati "normali". Con questo nodo è possibile identificare valori anomali anche se questi non rientrano in schemi precedentemente conosciuti e anche se l'utente non sa esattamente ciò che sta cercando.

Esempio

```

node = stream.create("anomalydetection", "My node")
node.setPropertyValue("anomaly_method", "PerRecords")
node.setPropertyValue("percent_records", 95)
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("peer_group_num_auto", True)
node.setPropertyValue("min_num_peer_groups", 3)
node.setPropertyValue("max_num_peer_groups", 10)

```

Tabella 99. proprietà anomalydetectionnode

| Proprietà anomalydetectionnode | Valori | Descrizione proprietà |
|--------------------------------|--|--|
| inputs | [campo1 ... campoN] | I modelli Rilevamento anomalie effettuano lo screening dei record in base ai campi di input specificati. Non utilizzano un campo obiettivo, né i campi peso e frequenza. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| mode | Expert Simple | |
| anomaly_method | IndexLevel PerRecords NumRecords | Specifica il metodo utilizzato per determinare il valore di interruzione per contrassegnare i record come anomali. |
| index_level | numero | Specifica il valore di interruzione minimo per contrassegnare le anomalie. |
| percent_records | numero | Imposta la soglia per contrassegnare i record in base alla percentuale di record nei dati di addestramento. |
| num_records | numero | Imposta la soglia per contrassegnare i record in base al numero di record nei dati di addestramento. |
| num_fields | numero intero | Numero di campi da segnalare per ciascun record anomalo. |
| impute_missing_values | indicatore | |
| adjustment_coeff | numero | Valore utilizzato per bilanciare il peso relativo attribuito ai campi continui e categoriali nel calcolo della distanza. |
| peer_group_num_auto | indicatore | Calcola automaticamente il numero dei gruppi di peer. |
| min_num_peer_groups | numero intero | Specifica il numero minimo di gruppi di peer utilizzati quando peer_group_num_auto è impostata su True. |
| max_num_per_groups | numero intero | Specifica il numero massimo di gruppi di peer. |
| num_peer_groups | numero intero | Specifica il numero di gruppi di peer utilizzati quando peer_group_num_auto è impostata su False. |
| noise_level | numero | Determina come trattare i valori anomali durante il raggruppamento tramite cluster. Specificare un valore compreso tra 0 e 0.5. |

Tabella 99. proprietà anomalydetectionnode (Continua)

| Proprietà anomalydetectionnode | Valori | Descrizione proprietà |
|--------------------------------|--------|--|
| noise_ratio | numero | Specifica la parte di memoria allocata per il componente da utilizzare per la memorizzazione del rumore. Specificare un valore compreso tra 0 e 0.5. |

Proprietà apriorinode



Il nodo Apriori estrae un insieme di regole dai dati, estrapolando le regole con il più alto contenuto di informazioni. Apriori offre cinque diversi metodi per la selezione delle regole e utilizza uno schema di indicizzazione sofisticato per elaborare in modo efficiente insiemi di dati di grandi dimensioni. In caso di problemi complessi, l'addestramento di Apriori è in genere più rapido. Apriori non ha un limite arbitrario per quanto riguarda il numero di regole che possono essere mantenute e può gestire regole con un massimo di 32 precondizioni. Apriori richiede che tutti i campi di input e output siano categoriali ma garantisce prestazioni migliori perché è ottimizzato per questo tipo di dati.

Esempio

```
node = stream.create("apriori", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("partition", "Test")
# For non-transactional
node.setPropertyValue("use_transactional_data", False)
node.setPropertyValue("consequents", ["Age"])
node.setPropertyValue("antecedents", ["BP", "Cholesterol", "Drug"])
# For transactional
node.setPropertyValue("use_transactional_data", True)
node.setPropertyValue("id_field", "Age")
node.setPropertyValue("contiguous", True)
node.setPropertyValue("content_field", "Drug")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Apriori_bp_choles_drug")
node.setPropertyValue("min_supp", 7.0)
node.setPropertyValue("min_conf", 30.0)
node.setPropertyValue("max_antecedents", 7)
node.setPropertyValue("true_flags", False)
node.setPropertyValue("optimize", "Memory")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("evaluation", "ConfidenceRatio")
node.setPropertyValue("lower_bound", 7)
```

Tabella 100. proprietà apriorinode

| Proprietà apriorinode | Valori | Descrizione proprietà |
|-----------------------|---------------------|--|
| consequents | campo | I modelli Apriori utilizzano antecedenti e conseguenti al posto dei campi obiettivo e di input standard. I campi peso e frequenza non sono utilizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| antecedents | [campo1 ... campoN] | |
| min_supp | numero | |

Tabella 100. proprietà apriorinode (Continua)

| Proprietà apriorinode | Valori | Descrizione proprietà |
|------------------------|--|--|
| min_conf | numero | |
| max_antecedents | numero | |
| true_flags | indicatore | |
| optimize | Speed Memory | |
| use_transactional_data | indicatore | |
| contiguous | indicatore | |
| id_field | stringa | |
| content_field | stringa | |
| mode | Simple Expert | |
| evaluation | RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare | |
| lower_bound | numero | |
| optimize | Speed Memory | Utilizzare per specificare se ottimizzare la velocità o la memoria durante la creazione del modello. |

Proprietà associationrulesnode



Il Nodo Regole di associazione è simile al Nodo Apriori; tuttavia, a differenza del nodo Apriori, il Nodo Regole di associazione è in grado di elaborare i dati dell'elenco. Inoltre, il Nodo Regole di associazione può essere utilizzato con IBM SPSS Analytic Server per elaborare dati di quantità elevata e sfruttare una più rapida elaborazione parallela.

Tabella 101. Proprietà associationrulesnode

| Proprietà associationrulesnode | Data type | Descrizione proprietà |
|--------------------------------|-------------------|---|
| predictions | campo | I campi possono essere visualizzati in questo elenco solo come predittori di una regola |
| conditions | [campo1...campoN] | I campi possono essere visualizzati in questo elenco solo come condizione di una regola |
| max_rule_conditions | integer | Il numero massimo di condizioni che possono essere incluse in una regola singola. Minimo 1, massimo 9. |
| max_rule_predictions | integer | Il numero massimo di previsioni che possono essere incluse in una regola singola. Minimo 1, massimo 5. |
| max_num_rules | integer | Il numero massimo di regole che possono essere considerate come parte della creazione della regola. Minimo 1, massimo 10.000. |

Tabella 101. Proprietà *associationrulesnode* (Continua)

| Proprietà <i>associationrulesnode</i> | Data type | Descrizione proprietà |
|---------------------------------------|--|---|
| <code>rule_criterion_top_n</code> | Confidence Rulesupport Lift Conditionsupport Distribuibilità | Il criterio della regola che determina il valore in base al quale vengono scelte le prime "N" regole nel modello. |
| <code>true_flags</code> | <i>Booleana</i> | Impostando Y si determina che durante la creazione della regola vengono considerati solo i valori true per i campi indicatore. |
| <code>rule_criterion</code> | <i>Booleana</i> | Impostando Y si determina che durante la creazione del modello vengono considerati solo i valori di creazione della regola per l'esclusione delle regole. |
| <code>min_confidence</code> | <i>number</i> | Da 0,1 a 100 - il valore percentuale per il livello di confidenza minimo richiesto per una regola prodotta da un modello. Se il modello produce una regola con un livello di confidenza inferiore al valore specificato, la regola viene eliminata. |
| <code>min_rule_support</code> | <i>number</i> | Da 0,1 a 100 - il valore percentuale per il supporto della regola minimo richiesto per una regola prodotta da un modello. Se il modello produce una regola con un livello di supporto della regola inferiore al valore specificato, la regola viene eliminata. |
| <code>min_condition_support</code> | <i>number</i> | Da 0,1 a 100 - il valore percentuale per il supporto della condizione minimo richiesto per una regola prodotta da un modello. Se il modello produce una regola con un livello di supporto della condizione inferiore al valore specificato, la regola viene eliminata. |
| <code>min_lift</code> | <i>integer</i> | Da 1 a 10 - rappresenta il minimo guadagno cumulativo richiesto per una regola generata dal modello. Se il modello produce una regola con un livello di guadagno cumulativo inferiore al valore specificato, la regola viene eliminata. |
| <code>exclude_rules</code> | <i>Booleana</i> | Utilizzata per selezionare un elenco di campi correlati da cui non si desidera che il modello crei regole. Esempio: <code>set :gsarsnode.exclude_rules = [[field1,field2, field3]],[field4, field5]]</code> - dove ogni elenco di campi separati da [] rappresenta una riga nella tabella. |
| <code>num_bins</code> | <i>integer</i> | Impostare il numero di bin automatici a cui i campi continui vengono discretizzati. Minimo 2, massimo 10. |
| <code>max_list_length</code> | <i>integer</i> | Si applica a tutti i campi dell'elenco di cui è sconosciuta la lunghezza massima. Gli elementi dell'elenco vengono inseriti nella creazione del modello fino al raggiungimento del numero qui specificato; eventuali ulteriori elementi vengono eliminati. Minimo 1, massimo 100. |
| <code>output_confidence</code> | <i>Booleana</i> | |

Tabella 101. Proprietà *associationrulesnode* (Continua)

| Proprietà <i>associationrulesnode</i> | Data type | Descrizione proprietà |
|---------------------------------------|--|---|
| <code>output_rule_support</code> | <i>Booleana</i> | |
| <code>output_lift</code> | <i>Booleana</i> | |
| <code>output_condition_support</code> | <i>Booleana</i> | |
| <code>output_deployability</code> | <i>Booleana</i> | |
| <code>rules_to_display</code> | upto all | Il numero massimo di regole da visualizzare nelle tabelle di output. |
| <code>display_upto</code> | <i>integer</i> | Se si imposta <code>upto</code> in <code>rules_to_display</code> , impostare il numero di regole da visualizzare nelle tabelle di output. Minimo 1. |
| <code>field_transformations</code> | <i>Booleana</i> | |
| <code>records_summary</code> | <i>Booleana</i> | |
| <code>rule_statistics</code> | <i>Booleana</i> | |
| <code>most_frequent_values</code> | <i>Booleana</i> | |
| <code>most_frequent_fields</code> | <i>Booleana</i> | |
| <code>word_cloud</code> | <i>Booleana</i> | |
| <code>word_cloud_sort</code> | Confidence Rulesupport Lift Conditionsupport Distribuibilità | |
| <code>word_cloud_display</code> | <i>integer</i> | Minimo 1, massimo 20 |
| <code>max_predictions</code> | <i>integer</i> | Il numero massimo di regole che possono essere applicate a ciascun input nel punteggio. |
| <code>criterion</code> | Confidence Rulesupport Lift Conditionsupport Distribuibilità | Selezionare la misura che determina l'efficacia delle regole. |
| <code>allow_repeats</code> | <i>Booleana</i> | Determina se nel punteggio vengono incluse le regole con la stessa previsione. |
| <code>check_input</code> | NoPredictions Predictions NoCheck | |

Proprietà *autoclassifiernode*



Il nodo Classificatore automatico crea e confronta svariati tipi di modelli per risultati binari (sì o no, abbandono oppure no e così via), consentendo di scegliere l'approccio migliore per una determinata analisi. Sono supportati numerosi algoritmi di modellazione ed è possibile selezionare i metodi da utilizzare, le opzioni specifiche per ognuno di essi e i criteri per confrontare i risultati. Il nodo genera un insieme di modelli basato sulle opzioni specificate e classifica i candidati migliori in base ai criteri indicati.

Esempio

```
node = stream.create("autoclassifier", "My node")
node.setPropertyValue("ranking_measure", "Accuracy")
node.setPropertyValue("ranking_dataset", "Training")
```

```

node.setPropertyValue("enable_accuracy_limit", True)
node.setPropertyValue("accuracy_limit", 0.9)
node.setPropertyValue("calculate_variable_importance", True)
node.setPropertyValue("use_costs", True)
node.setPropertyValue("svm", False)

```

Tabella 102. proprietà autoclassifiernode.

| Proprietà autoclassifiernode | Valori | Descrizione proprietà |
|----------------------------------|---|---|
| target | campo | Per i campi obiettivo, il nodo Classificatore automatico richiede un solo campo obiettivo e uno o più campi di input. È inoltre possibile specificare i campi frequenza e peso. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| ranking_measure | Accuracy Area_under_curve Profit Lift Num_variables | |
| ranking_dataset | Training Test | |
| number_of_models | numero intero | Numero dei modelli da includere nel nugget del modello. Specificare un numero intero compreso fra 1 e 100. |
| calculate_variable_importance | indicatore | |
| enable_accuracy_limit | indicatore | |
| accuracy_limit | numero intero | Numero intero compreso fra 0 e 100. |
| enable_area_under_curve_limit | indicatore | |
| area_under_curve_limit | numero | Numero reale compreso tra 0.0 e 1.0. |
| enable_profit_limit | indicatore | |
| profit_limit | numero | Numero intero maggiore di 0. |
| enable_lift_limit | indicatore | |
| lift_limit | numero | Numero reale maggiore di 1.0. |
| enable_number_of_variables_limit | indicatore | |
| number_of_variables_limit | numero | Numero intero maggiore di 0. |
| use_fixed_cost | indicatore | |
| fixed_cost | numero | Numero reale maggiore di 0.0. |
| variable_cost | campo | |
| use_fixed_revenue | indicatore | |
| fixed_revenue | numero | Numero reale maggiore di 0.0. |
| variable_revenue | campo | |
| use_fixed_weight | indicatore | |
| fixed_weight | numero | Numero reale maggiore di 0,0. |
| variable_weight | campo | |
| lift_percentile | numero | Numero intero compreso fra 0 e 100. |
| enable_model_build_time_limit | indicatore | |

Tabella 102. proprietà autotrasferite (Continua).

| Proprietà autotrasferite | Valori | Descrizione proprietà |
|--|------------|--|
| model_build_time_limit | numero | Numero intero impostato sul numero di minuti per limitare il tempo impiegato per creare ogni singolo modello. |
| enable_stop_after_time_limit | indicatore | |
| stop_after_time_limit | numero | Numero reale impostato sul numero di ore per limitare il tempo complessivo impiegato per l'esecuzione di un classificatore automatico. |
| enable_stop_after_valid_model_produced | indicatore | |
| use_costs | indicatore | |
| <algorithm> | indicatore | Abilita o disabilita l'utilizzo di un algoritmo specifico. |
| <algorithm>.<property> | stringa | Imposta il valore di una proprietà di un algoritmo specifico. Per ulteriori informazioni, consultare l'argomento "Impostazione delle proprietà degli algoritmi". |

Impostazione delle proprietà degli algoritmi

Per i nodi Classificatore automatico, Numerico automatico e Cluster automatico, le proprietà degli algoritmi specifici utilizzati dal nodo si possono impostare utilizzando il formato generico:

```
autonode.setKeyedPropertyValue(<algorithm>, <property>, <value>)
```

Ad esempio:

```
node.setKeyedPropertyValue("neuralnetwork", "method", "MultilayerPerceptron")
```

I nomi degli algoritmi per il nodo Classificatore automatico sono cart, chaid, quest, c50, logreg, decisionlist, bayesnet, discriminant, svm e knn.

I nomi degli algoritmi per il nodo Numerico automatico sono cart, chaid, neuralnetwork, genlin, svm, regression, linear e knn.

I nomi degli algoritmi per il nodo Cluster automatico sono twostep, Medie K e kohonen.

I nomi delle proprietà sono standard, come documentato per i nodi dei singoli algoritmi.

Le proprietà degli algoritmi che contengono punti o altri tipi di punteggiatura devono essere racchiuse tra virgolette singole, per esempio:

```
node.setKeyedPropertyValue("logreg", "tolerance", "1.0E-5")
```

Come proprietà è possibile assegnare anche valori multipli, per esempio:

```
node.setKeyedPropertyValue("decisionlist", "search_direction", ["Up", "Down"])
```

Per attivare o disattivare l'utilizzo di un algoritmo specifico:

```
node.setPropertyValue("chaid", True)
```

Nota: Nei casi in cui determinate opzioni di algoritmi non siano disponibili nel nodo Classificatore automatico o quando è possibile specificare un solo valore anziché un intervallo di valori, per gli script si applicano gli stessi limiti validi per l'accesso al nodo con la normale procedura.

Proprietà autoclusternode



Il nodo Cluster automatico stima e confronta i modelli di cluster che identificano gruppi di record con caratteristiche simili. Il nodo funziona in modo analogo ad altri nodi Modelli automatici e consente di sperimentare varie combinazioni di opzioni in un singolo passaggio di modellazione. I modelli si possono confrontare utilizzando misure di base con cui tentare di filtrare e classificare l'utilità dei modelli di cluster e fornire una misura in base all'importanza di determinati campi.

Esempio

```
node = stream.create("autocluster", "My node")
node.setPropertyValue("ranking_measure", "Silhouette")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_silhouette_limit", True)
node.setPropertyValue("silhouette_limit", 5)
```

Tabella 103. proprietà autoclusternode

| Proprietà autoclusternode | Valori | Descrizione proprietà |
|-----------------------------------|--|--|
| evaluation | <i>campo</i> | Nota: solo nodo Cluster automatico. Identifica il campo per cui verrà calcolato un valore di importanza. In alternativa, può essere utilizzato per identificare il modo in cui il cluster distingue il valore di questo campo e, pertanto, la validità della previsione di questo campo da parte del modello. |
| ranking_measure | Silhouette Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Importance | |
| ranking_dataset | Training Test | |
| summary_limit | <i>numero intero</i> | Numero dei modelli da elencare nel report. Specificare un numero intero compreso fra 1 e 100. |
| enable_silhouette_limit | <i>indicatore</i> | |
| silhouette_limit | <i>numero intero</i> | Numero intero compreso fra 0 e 100. |
| enable_number_less_limit | <i>indicatore</i> | |
| number_less_limit | <i>numero</i> | Numero reale compreso tra 0.0 e 1.0. |
| enable_number_greater_limit | <i>indicatore</i> | |
| number_greater_limit | <i>numero</i> | Numero intero maggiore di 0. |
| enable_smallest_cluster_limit | <i>indicatore</i> | |
| smallest_cluster_units | Percentage Counts | |
| smallest_cluster_limit_percentage | <i>numero</i> | |
| smallest_cluster_limit_count | <i>numero intero</i> | Numero intero maggiore di 0. |

Tabella 103. proprietà autoclusternode (Continua)

| Proprietà autoclusternode | Valori | Descrizione proprietà |
|----------------------------------|---------------------------|---|
| enable_largest_cluster_limit | indicatore | |
| largest_cluster_units | Percentage Counts | |
| largest_cluster_limit_percentage | numero | |
| largest_cluster_limit_count | numero intero | |
| enable_smallest_largest_limit | indicatore | |
| smallest_largest_limit | numero | |
| enable_importance_limit | indicatore | |
| importance_limit_condition | Greater_than Less_than | |
| importance_limit_greater_than | numero | Numero intero compreso fra 0 e 100. |
| importance_limit_less_than | numero | Numero intero compreso fra 0 e 100. |
| <algorithm> | indicatore | Abilita o disabilita l'utilizzo di un algoritmo specifico. |
| <algorithm>.<property> | stringa | Imposta il valore di una proprietà di un algoritmo specifico. Per ulteriori informazioni, consultare l'argomento "Impostazione delle proprietà degli algoritmi" a pagina 168. |

Proprietà autonumericnode



Il nodo Numerico automatico stima e confronta i modelli per i risultati di intervalli numerici continui utilizzando svariati metodi. Il nodo funziona in modo analogo al nodo Classificatore automatico e consente di scegliere gli algoritmi da utilizzare e di sperimentare più combinazioni di opzioni in un singolo passaggio di modellazione. Gli algoritmi supportati includono reti neurali, C&R Tree, CHAID, regressione lineare, regressione lineare generalizzata e SVM (Support Vector Machine). I modelli si possono confrontare in base a correlazione, errore relativo o numero di variabili utilizzato.

Esempio

```
node = stream.create("autonumeric", "My node")
node.setPropertyValue("ranking_measure", "Correlation")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_correlation_limit", True)
node.setPropertyValue("correlation_limit", 0.8)
node.setPropertyValue("calculate_variable_importance", True)
node.setPropertyValue("neuralnetwork", True)
node.setPropertyValue("chaid", False)
```

Tabella 104. proprietà autonumericnode

| Proprietà autonumericnode | Valori | Descrizione proprietà |
|---------------------------|------------|--|
| custom_fields | indicatore | Se True (vero), saranno utilizzate le impostazioni personalizzate dei campi anziché le impostazioni del nodo Tipo. |

Tabella 104. proprietà autonumericnode (Continua)

| Proprietà autonumericnode | Valori | Descrizione proprietà |
|-------------------------------|-------------------------------|--|
| obiettivo | <i>campo</i> | Il nodo Numerico automatico richiede un solo campo obiettivo e uno o più campi di input. È inoltre possibile specificare i campi frequenza e peso. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| inputs | <i>[campo1 ... campo2]</i> | |
| partition | <i>campo</i> | |
| use_frequency | <i>indicatore</i> | |
| frequency_field | <i>campo</i> | |
| use_weight | <i>indicatore</i> | |
| weight_field | <i>campo</i> | |
| use_partitioned_data | <i>indicatore</i> | Se è definito un campo partizione, per la creazione del modello verranno utilizzati solo i dati di addestramento. |
| ranking_measure | Correlation NumberOfFields | |
| ranking_dataset | Test Training | |
| number_of_models | <i>numero intero</i> | Numero dei modelli da includere nel nugget del modello. Specificare un numero intero compreso fra 1 e 100. |
| calculate_variable_importance | <i>indicatore</i> | |
| enable_correlation_limit | <i>indicatore</i> | |
| correlation_limit | <i>numero intero</i> | |
| enable_number_of_fields_limit | <i>indicatore</i> | |
| number_of_fields_limit | <i>numero intero</i> | |
| enable_relative_error_limit | <i>indicatore</i> | |
| relative_error_limit | <i>numero intero</i> | |
| enable_model_build_time_limit | <i>indicatore</i> | |
| model_build_time_limit | <i>numero intero</i> | |
| enable_stop_after_time_limit | <i>indicatore</i> | |
| stop_after_time_limit | <i>numero intero</i> | |
| stop_if_valid_model | <i>indicatore</i> | |
| <algorithm> | <i>indicatore</i> | Abilita o disabilita l'utilizzo di un algoritmo specifico. |
| <algorithm>.<property> | <i>stringa</i> | Imposta il valore di una proprietà di un algoritmo specifico. Per ulteriori informazioni, consultare l'argomento "Impostazione delle proprietà degli algoritmi" a pagina 168. |

Proprietà bayesnetnode



Il nodo Rete bayesiana consente di generare un modello di probabilità combinando elementi osservati e registrati con conoscenze del mondo reale per stabilire la probabilità di occorrenze. Il nodo si concentra sulle reti TAN (Tree Augmented Naïve Bayes) e coperta di Markov, che sono prevalentemente utilizzate a scopo di classificazione.

Esempio

```
node = stream.create("bayesnet", "My node")
node.setPropertyValue("continue_training_existing_model", True)
node.setPropertyValue("structure_type", "MarkovBlanket")
node.setPropertyValue("use_feature_selection", True)
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("independence", "Pearson")
```

Tabella 105. Proprietà bayesnetnode

| Proprietà bayesnetnode | Valori | Descrizione proprietà |
|----------------------------------|-----------------------|--|
| inputs | [campo1 ... campoN] | I modelli di rete bayesiana utilizzano un solo campo obiettivo e uno o più campi di input. I campi continui vengono automaticamente discretizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| continue_training_existing_model | indicatore | |
| structure_type | TAN MarkovBlanket | Selezionare la struttura da utilizzare per la creazione della rete bayesiana. |
| use_feature_selection | indicatore | |
| parameter_learning_method | Likelihood Bayes | Specifica il metodo utilizzato per stimare le tabelle di probabilità condizionale tra i nodi in cui i valori dei nodi padre sono noti. |
| mode | Expert Simple | |
| missing_values | indicatore | |
| all_probabilities | indicatore | |
| independence | Likelihood Pearson | Specifica il metodo utilizzato per determinare se le osservazioni accoppiate su due variabili sono indipendenti tra loro. |
| significance_level | numero | Specifica il valore di interruzione per determinare l'indipendenza. |
| maximal_conditioning_set | numero | Imposta il numero massimo di variabili di condizionamento da utilizzare per il test dell'indipendenza. |
| inputs_always_selected | [campo1 ... campoN] | Specifica quali campi dell'insieme di dati devono sempre essere utilizzati nella creazione della rete bayesiana. Nota: il campo obiettivo è sempre selezionato. |

Tabella 105. Proprietà bayesnetnode (Continua)

| Proprietà bayesnetnode | Valori | Descrizione proprietà |
|---------------------------------|--------------------|---|
| maximum_number_inputs | numero | Specifica il numero massimo di campi di input da utilizzare nella creazione della rete bayesiana. |
| calculate_variable_importance | indicatore | |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |
| adjusted_propensity_partition | Test Validation | |

Proprietà buildr



Il nodo Creazione R consente di immettere uno script R personalizzato per eseguire la creazione del modello e il calcolo del punteggio del modello distribuito in IBM SPSS Modeler.

Esempio

```
node = stream.create("buildr", "My node")
node.setPropertyValue("score_syntax", "")
result<-predict(modelerModel,newdata=modelerData)
modelerData<-cbind(modelerData,result)
var1<-c(fieldName="NaPrediction",fieldLabel="",fieldStorage="real",fieldMeasure="",
fieldFormat="",fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""
```

Tabella 106. Proprietà buildr.

| Proprietà buildr | Valori | Descrizione proprietà |
|------------------------|------------------------------------|--|
| build_syntax | stringa | Sintassi degli script R per la creazione del modello. |
| score_syntax | stringa | Sintassi degli script R per il calcolo del punteggio del modello. |
| convert_flags | StringsAndDoubles LogicalValues | Opzione per la conversione dei campi indicatore. |
| convert_datetime | indicatore | L'opzione per convertire le variabili con formati di date o data/ora in formati data/ora R. |
| convert_datetime_class | POSIXct POSIXlt | Le opzioni per specificare in quale formato vengono convertite le variabili con formati data o data/ora. |
| convert_missing | indicatore | Opzione per convertire i valore mancanti nel valore R NA. |
| output_html | indicatore | Opzione per visualizzare grafici su una scheda nel nugget del modello R. |
| output_text | indicatore | Opzione per scrivere l'output di testo della console R in una scheda nel nugget del modello R. |

Proprietà c50node



Il nodo C5.0 crea una struttura ad albero delle decisioni o un insieme di regole. Il modello suddivide il campione in base al campo che fornisce il massimo guadagno di informazioni a ogni livello. Il campo obiettivo deve essere categoriale. Sono consentite suddivisioni multiple in più di due sottogruppi.

Esempio

```
node = stream.create("c50", "My node")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "C5_Drug")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("output_type", "DecisionTree")
node.setPropertyValue("use_xval", True)
node.setPropertyValue("xval_num_folds", 3)
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("favor", "Generality")
node.setPropertyValue("min_child_records", 3)
# "Costs" tab
node.setPropertyValue("use_costs", True)
node.setPropertyValue("costs", [["drugA", "drugX", 2]])
```

Tabella 107. proprietà c50node

| Proprietà c50node | Valori | Descrizione proprietà |
|-------------------------------|-------------------------|---|
| obiettivo | <i>campo</i> | I modelli C50 utilizzano un solo campo obiettivo e uno o più campi di input. È anche possibile specificare un campo peso. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| output_type | DecisionTree RuleSet | |
| group_symbolics | <i>indicatore</i> | |
| use_boost | <i>indicatore</i> | |
| boost_num_trials | <i>numero</i> | |
| use_xval | <i>indicatore</i> | |
| xval_num_folds | <i>numero</i> | |
| mode | Simple Expert | |
| favor | Accuracy Generality | Precisione o generalità della preferenza. |
| expected_noise | <i>numero</i> | |
| min_child_records | <i>numero</i> | |
| pruning_severity | <i>numero</i> | |
| use_costs | <i>indicatore</i> | |
| costs | <i>strutturato</i> | Si tratta di una proprietà strutturata. |
| use_winnowing | <i>indicatore</i> | |
| use_global_pruning | <i>indicatore</i> | Impostato su (True) per default. |
| calculate_variable_importance | <i>indicatore</i> | |

Tabella 107. proprietà c50node (Continua)

| Proprietà c50node | Valori | Descrizione proprietà |
|---------------------------------|--------------------|-----------------------|
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |
| adjusted_propensity_partition | Test Validation | |

Proprietà carmanode



Il modello CARMA estrae un insieme di regole dai dati senza che venga richiesto all'utente di specificare i campi di input o obiettivo. A differenza di Apriori, il nodo CARMA fornisce le impostazioni di creazione per il supporto delle regole (sia per l'antecedente che per il conseguente) anziché solo per il supporto antecedente. Pertanto, le regole generate possono essere utilizzate per una gamma più vasta di applicazioni, ad esempio per trovare un elenco di prodotti o di servizi (antecedenti) il cui conseguente è rappresentato dall'articolo che si desidera promuovere per le festività correnti.

Esempio

```
node = stream.create("carma", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("use_transactional_data", True)
node.setPropertyValue("inputs", ["BP", "Cholesterol", "Drug"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "age_bp_drug")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("min_supp", 10.0)
node.setPropertyValue("min_conf", 30.0)
node.setPropertyValue("max_size", 5)
# Expert Options
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("use_pruning", True)
node.setPropertyValue("pruning_value", 300)
node.setPropertyValue("vary_support", True)
node.setPropertyValue("estimated_transactions", 30)
node.setPropertyValue("rules_without_antecedents", True)
```

Tabella 108. proprietà carmanode

| Proprietà carmanode | Valori | Descrizione proprietà |
|------------------------|---------------------|--|
| inputs | [campo1 ... campon] | I modelli CARMA utilizzano un elenco di campi di input, ma nessun campo obiettivo. I campi peso e frequenza non sono utilizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| id_field | campo | Campo utilizzato come campo ID per la creazione del modello. |
| contiguous | indicatore | Utilizzata per specificare se gli ID del campo ID sono contigui. |
| use_transactional_data | indicatore | |
| content_field | campo | |

Tabella 108. proprietà carmanode (Continua)

| Proprietà carmanode | Valori | Descrizione proprietà |
|---------------------------|----------------------|---|
| min_supp | numero (percentuale) | Si riferisce al supporto della regola più che al supporto antecedente. Il valore predefinito è 20%. |
| min_conf | numero (percentuale) | Il valore predefinito è 20%. |
| max_size | numero | Il valore di default è 10. |
| mode | Simple Expert | L'impostazione di default è Simple. |
| exclude_multiple | indicatore | Esclude le regole con più conseguenti. L'impostazione predefinita è False. |
| use_pruning | indicatore | L'impostazione predefinita è False. |
| pruning_value | numero | Il valore predefinito è 500. |
| vary_support | indicatore | |
| estimated_transactions | numero intero | |
| rules_without_antecedents | indicatore | |

Proprietà cartnode



Il nodo Struttura ad albero di classificazione e regressione (C&R) genera una struttura ad albero delle decisioni che consente di prevedere o classificare osservazioni future. Il metodo utilizza partizionamento ricorsivo per suddividere i record di addestramento in segmenti, riducendo l'impurità ad ogni passaggio. Un nodo della struttura ad albero è considerato "puro" quando il 100% dei casi nel nodo fa parte di una categoria specifica del campo obiettivo. I campi obiettivo e di input possono essere intervalli numerici o categoriali (nominali, ordinali o flag); tutte le suddivisioni sono binarie (solo due sottogruppi).

Esempio

```
node = stream.createAt("cart", "My node", 200, 100)
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "BP", "Cholesterol"])
# "Build Options" tab, "Objective" panel
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("tree_directives", """Grow Node Index 0 Children 1 2
Grow Node Index 2 Children 3 4""")
# "Build Options" tab, "Basics" panel
node.setPropertyValue("prune_tree", False)
node.setPropertyValue("use_std_err_rule", True)
node.setPropertyValue("std_err_multiplier", 3.0)
node.setPropertyValue("max_surrogates", 7)
# "Build Options" tab, "Stopping Rules" panel
node.setPropertyValue("use_percentage", True)
node.setPropertyValue("min_parent_records_pc", 5)
node.setPropertyValue("min_child_records_pc", 3)
# "Build Options" tab, "Advanced" panel
node.setPropertyValue("min_impurity", 0.0003)
node.setPropertyValue("impurity_measure", "Twoing")
```

```
# "Model Options" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Cart_Drug")
```

Tabella 109. proprietà cartnode

| Proprietà cartnode | Valori | Descrizione proprietà |
|----------------------------------|--|--|
| obiettivo | <i>campo</i> | I modelli C&R Tree richiedono un solo campo obiettivo e uno o più campi di input. È inoltre possibile specificare un campo frequenza. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| continue_training_existing_model | <i>indicatore</i> | |
| objective | Standard Boosting Bagging psm | psm viene utilizzato per insiemi di dati di grandi dimensioni e richiede una connessione Server. |
| model_output_type | Single InteractiveBuilder | |
| use_tree_directives | <i>indicatore</i> | |
| tree_directives | <i>stringa</i> | Specifica le direttive per l'ingrandimento di strutture ad albero. È possibile racchiudere le direttive tra virgolette triple per evitare il carattere escape in nuove righe o virgolette. Si noti che le direttive possono essere molto sensibili a piccole modifiche nelle opzioni di modellazione e nei dati e possono non essere generalizzate ad altri insiemi di dati. |
| use_max_depth | Default Custom | |
| max_depth | <i>numero intero</i> | Profondità massima della struttura ad albero, da 0 a 1000. Valore utilizzato solo se use_max_depth = Custom. |
| prune_tree | <i>indicatore</i> | Taglia struttura ad albero per evitare sovradattamento. |
| use_std_err | <i>indicatore</i> | Utilizza differenza massima di rischio (in errori standard). |
| std_err_multiplier | <i>numero</i> | Differenza massima. |
| max_surrogates | <i>numero</i> | Numero massimo surrogati. |
| use_percentage | <i>indicatore</i> | |
| min_parent_records_pc | <i>numero</i> | |
| min_child_records_pc | <i>numero</i> | |
| min_parent_records_abs | <i>numero</i> | |
| min_child_records_abs | <i>numero</i> | |
| use_costs | <i>indicatore</i> | |
| costs | <i>strutturato</i> | Proprietà strutturata. |

Tabella 109. proprietà cartnode (Continua)

| Proprietà cartnode | Valori | Descrizione proprietà |
|---------------------------------|--|--|
| priors | Data Equal Custom | |
| custom_priors | <i>strutturato</i> | Proprietà strutturata. |
| adjust_priors | <i>indicatore</i> | |
| trails | <i>numero</i> | Numero di modelli di componenti per boosting o bagging. |
| set_ensemble_method | Voting HighestProbability HighestMeanProbability | Regola di combinazione di default per obiettivi categoriali. |
| range_ensemble_method | Mean Median | Regola di combinazione di default per target continui. |
| large_boost | <i>indicatore</i> | Applica il boosting a insiemi di dati di grandi dimensioni. |
| min_impurity | <i>numero</i> | |
| impurity_measure | Gini Twoing Ordered | |
| train_pct | <i>numero</i> | Insieme di prevenzione del sovradattamento. |
| set_random_seed | <i>indicatore</i> | Opzione Replica risultati. |
| seed | <i>numero</i> | |
| calculate_variable_importance | <i>indicatore</i> | |
| calculate_raw_propensities | <i>indicatore</i> | |
| calculate_adjusted_propensities | <i>indicatore</i> | |
| adjusted_propensity_partition | Test Validation | |

Proprietà chaidnode



Il nodo CHAID genera una struttura ad albero delle decisioni utilizzando statistiche chi-quadrato per identificare suddivisioni ottimali. A differenza dei nodi C&R Tree e QUEST, il nodo CHAID può generare strutture ad albero non binarie e pertanto alcune suddivisioni possono avere più di due rami. I campi obiettivo e di input possono essere intervallo numerico (continui) o categoriali. Un CHAID completo è una modificazione di CHAID che esegue operazioni avanzate per l'analisi di tutte le suddivisioni possibili, ma richiede tempi di elaborazione maggiori.

Esempio

```

filenode = stream.createAt("variablefile", "My node", 100, 100)
filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node = stream.createAt("chaid", "My node", 200, 100)
stream.link(filenode, node)

node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
node.setPropertyValue("use_model_name", True)

```



```

node.setPropertyValue("model_name", "CHAID")
node.setPropertyValue("method", "Chaid")
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("tree_directives", "Test")
node.setPropertyValue("split_alpha", 0.03)
node.setPropertyValue("merge_alpha", 0.04)
node.setPropertyValue("chi_square", "Pearson")
node.setPropertyValue("use_percentage", False)
node.setPropertyValue("min_parent_records_abs", 40)
node.setPropertyValue("min_child_records_abs", 30)
node.setPropertyValue("epsilon", 0.003)
node.setPropertyValue("max_iterations", 75)
node.setPropertyValue("split_merged_categories", True)
node.setPropertyValue("bonferroni_adjustment", True)

```

Tabella 110. proprietà chaidnode

| Proprietà chaidnode | Valori | Descrizione proprietà |
|----------------------------------|--|--|
| obiettivo | <i>campo</i> | I modelli CHAID richiedono un solo campo obiettivo e uno o più campi di input. È inoltre possibile specificare un campo frequenza. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| continue_training_existing_model | <i>indicatore</i> | |
| objective | Standard Boosting Bagging psm | psm viene utilizzato per insiemi di dati di grandi dimensioni e richiede una connessione Server. |
| model_output_type | Single InteractiveBuilder | |
| use_tree_directives | <i>indicatore</i> | |
| tree_directives | <i>stringa</i> | |
| method | Chaid ExhaustiveChaid | |
| use_max_depth | Default Custom | |
| max_depth | <i>numero intero</i> | Profondità massima della struttura ad albero, da 0 a 1000. Valore utilizzato solo se use_max_depth = Custom. |
| use_percentage | <i>indicatore</i> | |
| min_parent_records_pc | <i>numero</i> | |
| min_child_records_pc | <i>numero</i> | |
| min_parent_records_abs | <i>numero</i> | |
| min_child_records_abs | <i>numero</i> | |
| use_costs | <i>indicatore</i> | |
| costs | <i>strutturato</i> | Proprietà strutturata. |
| trails | <i>numero</i> | Numero di modelli di componenti per boosting o bagging. |
| set_ensemble_method | Voting HighestProbability HighestMeanProbability | Regola di combinazione di default per obiettivi categoriali. |

Tabella 110. proprietà chaidnode (Continua)

| Proprietà chaidnode | Valori | Descrizione proprietà |
|---------------------------------|--------------------|---|
| range_ensemble_method | Mean Median | Regola di combinazione di default per target continui. |
| large_boost | indicatore | Applica il boosting a insiemi di dati di grandi dimensioni. |
| split_alpha | numero | Livello di significatività per suddivisione. |
| merge_alpha | numero | Livello di significatività per unione. |
| bonferroni_adjustment | indicatore | Adegua valori di significatività tramite il metodo di Bonferroni. |
| split_merged_categories | indicatore | Consenti risuddivisione di categorie unite. |
| chi_square | Pearson LR | Metodo utilizzato per calcolare la statistica chi-quadrato: Pearson o Rapporto di verosimiglianza |
| epsilon | numero | Modifica minima nelle frequenze di cella previste. |
| max_iterations | numero | Numero massimo di iterazioni per la convergenza. |
| set_random_seed | numero intero | |
| seed | numero | |
| calculate_variable_importance | indicatore | |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |
| adjusted_propensity_partition | Test Validation | |
| maximum_number_of_models | numero intero | |

Proprietà coxregnode



Il nodo Regressione di Cox consente di generare un modello di sopravvivenza per i dati della relazione tempo-evento in presenza di record censurati. Il modello produce una funzione di sopravvivenza che prevede la probabilità che l'evento di interesse si sia verificato a una determinata ora (t) per i valori dati delle variabili di input.

Esempio

```
node = stream.create("coxreg", "My node")
node.setPropertyValue("survival_time", "tenure")
node.setPropertyValue("method", "BackwardsStepwise")
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("removal_criterion", "Conditional")
node.setPropertyValue("survival", True)
```

Tabella 111. proprietà coxregnode

| Proprietà coxregnode | Valori | Descrizione proprietà |
|----------------------|--------|---|
| survival_time | campo | I modelli di regressione di Cox richiedono un solo campo contenente i tempi di sopravvivenza. |

Tabella 111. proprietà coxregnode (Continua)

| Proprietà coxregnode | Valori | Descrizione proprietà |
|----------------------|---|--|
| obiettivo | <i>campo</i> | I modelli di regressione di Cox richiedono un solo campo obiettivo e uno o più campi di input. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| method | Enter Stepwise BackwardsStepwise | |
| groups | <i>campo</i> | |
| model_type | MainEffects Custom | |
| custom_terms | ["BP*Sex" "BP*Age"] | |
| mode | Expert Simple | |
| max_iterations | <i>numero</i> | |
| p_converge | 1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0 | |
| p_converge | 1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0 | |
| l_converge | 1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0 | |
| removal_criterion | LR Wald Conditional | |
| probability_entry | <i>numero</i> | |
| probability_removal | <i>numero</i> | |
| output_display | EachStep LastStep | |
| ci_enable | <i>indicatore</i> | |
| ci_value | 90 95 99 | |
| correlation | <i>indicatore</i> | |
| display_baseline | <i>indicatore</i> | |
| survival | <i>indicatore</i> | |
| hazard | <i>indicatore</i> | |

Tabella 111. proprietà coxregnode (Continua)

| Proprietà coxregnode | Valori | Descrizione proprietà |
|----------------------|-------------------------|--|
| log_minus_log | <i>indicatore</i> | |
| one_minus_survival | <i>indicatore</i> | |
| separate_line | <i>campo</i> | |
| valore | <i>numero o stringa</i> | Se per un campo non viene specificato alcun valore sarà utilizzata l'opzione di default "Media". |

Proprietà decisionlistnode



Il nodo Elenco di decisioni identifica i sottogruppi o i segmenti che mostrano una probabilità maggiore o minore che si verifichi un determinato risultato binario rispetto alla popolazione globale. Per esempio, è possibile che si cerchino i clienti non a rischio di abbandono o quelli che più probabilmente rispondano in modo favorevole a una campagna. È possibile incorporare le proprie conoscenze di business nel modello aggiungendo propri segmenti personalizzati e visualizzando in anteprima modelli alternativi uno accanto all'altro per confrontarne i risultati. I modelli Elenco di decisioni consistono in un elenco di regole in cui ogni regola ha una condizione e un risultato. Le regole vengono applicate in ordine e la prima regola corrispondente determina il risultato.

Esempio

```
node = stream.create("decisionlist", "My node")
node.setPropertyValue("search_direction", "Down")
node.setPropertyValue("target_value", 1)
node.setPropertyValue("max_rules", 4)
node.setPropertyValue("min_group_size_pct", 15)
```

Tabella 112. Proprietà decisionlistnode

| Proprietà decisionlistnode | Valori | Descrizione proprietà |
|----------------------------|-------------------------------|--|
| obiettivo | <i>campo</i> | I modelli Elenco di decisioni utilizzano un solo campo obiettivo e uno o più campi di input. È inoltre possibile specificare un campo frequenza. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| model_output_type | Modello InteractiveBuilder | |
| search_direction | Up Down | Si riferisce alla ricerca di segmenti, dove Up è l'equivalente di Probabilità elevata e Down è l'equivalente di Probabilità bassa. |
| target_value | <i>stringa</i> | Se non specificata, presuppone il valore vero per i flag. |
| max_rules | <i>numero intero</i> | Il numero massimo di segmenti escluso il resto. |
| min_group_size | <i>numero intero</i> | Dimensione minima del segmento. |
| min_group_size_pct | <i>numero</i> | Dimensioni minime del segmento espresse come percentuale. |

Tabella 112. Proprietà decisionlistnode (Continua)

| Proprietà decisionlistnode | Valori | Descrizione proprietà |
|---------------------------------|--------------------------|---|
| confidence_level | numero | Soglia minima di cui un campo di input deve migliorare la probabilità di risposta (guadagno cumulativo) perché valga la pena aggiungerlo alla definizione di un segmento. |
| max_segments_per_rule | numero intero | |
| mode | Simple Expert | |
| bin_method | EqualWidth EqualCount | |
| bin_count | numero | |
| max_models_per_cycle | numero intero | Larghezza di ricerca per gli elenchi. |
| max_rules_per_cycle | numero intero | Larghezza di ricerca per le regole di segmento. |
| segment_growth | numero | |
| include_missing | indicatore | |
| final_results_only | indicatore | |
| reuse_fields | indicatore | Consente il riutilizzo degli attributi (campi di input che compaiono nelle regole). |
| max_alternatives | numero intero | |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |
| adjusted_propensity_partition | Test Validation | |

Proprietà discriminantnode



L'analisi discriminante prevede presupposti più rigidi rispetto alla regressione logistica, ma può essere una valida alternativa o un complemento dell'analisi di regressione logistica quando vengono soddisfatti tali presupposti.

Esempio

```
node = stream.create("discriminant", "My node")
node.setPropertyValue("target", "custcat")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("method", "Stepwise")
```

Tabella 113. proprietà discriminantnode

| Proprietà discriminantnode | Valori | Descrizione proprietà |
|----------------------------|--------|--|
| obiettivo | campo | I modelli Discriminante richiedono un solo campo obiettivo e uno o più campi di input. I campi peso e frequenza non sono utilizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |

Tabella 113. proprietà discriminantnode (Continua)

| Proprietà discriminantnode | Valori | Descrizione proprietà |
|----------------------------|---|--|
| method | Enter Stepwise | |
| mode | Simple Expert | |
| prior_probabilities | AllEqual ComputeFromSizes | |
| covariance_matrix | WithinGroups SeparateGroups | |
| means | <i>indicatore</i> | Opzioni relative alle statistiche nella finestra di dialogo Output avanzato. |
| univariate_anovas | <i>indicatore</i> | |
| box_m | <i>indicatore</i> | |
| within_group_covariance | <i>indicatore</i> | |
| within_groups_correlation | <i>indicatore</i> | |
| separate_groups_covariance | <i>indicatore</i> | |
| total_covariance | <i>indicatore</i> | |
| fishers | <i>indicatore</i> | |
| unstandardized | <i>indicatore</i> | |
| casewise_results | <i>indicatore</i> | Opzioni relative alla classificazione nella finestra di dialogo Output avanzato. |
| limit_to_first | <i>numero</i> | Il valore di default è 10. |
| summary_table | <i>indicatore</i> | |
| leave_one_classification | <i>indicatore</i> | |
| combined_groups | <i>indicatore</i> | |
| separate_groups_covariance | <i>indicatore</i> | Opzione delle matrici Covarianza per gruppi separati . |
| territorial_map | <i>indicatore</i> | |
| combined_groups | <i>indicatore</i> | Opzione del nodo Plot Gruppi combinati . |
| separate_groups | <i>indicatore</i> | Opzione del nodo Plot Gruppi separati . |
| summary_of_steps | <i>indicatore</i> | |
| F_pairwise | <i>indicatore</i> | |
| stepwise_method | WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV | |
| V_to_enter | <i>numero</i> | |
| criteria | UseValue UseProbability | |
| F_value_entry | <i>numero</i> | Il valore di default è 3.84. |
| F_value_removal | <i>numero</i> | Il valore di default è 2,71. |
| probability_entry | <i>numero</i> | Il valore di default è 0.05. |
| probability_removal | <i>numero</i> | Il valore di default è 0.10. |

Tabella 113. proprietà discriminantnode (Continua)

| Proprietà discriminantnode | Valori | Descrizione proprietà |
|---------------------------------|--------------------|-----------------------|
| calculate_variable_importance | indicatore | |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |
| adjusted_propensity_partition | Test Validation | |

Proprietà factornode



Il nodo fattoriale/PCA offre potenti tecniche di riduzione dei dati che consentono di diminuirne la complessità. L'analisi dei componenti principali (PCA, Principal Components Analysis) trova le combinazioni lineari dei campi di input che catturano meglio la varianza nell'intero insieme di campi, dove i componenti sono ortogonali (perpendicolari) l'uno rispetto all'altro. L'analisi fattoriale tenta di identificare i concetti sottostanti, o fattori, che spiegano lo schema delle correlazioni all'interno dell'insieme di campi osservati. Entrambi gli approcci mirano a trovare un numero ridotto di campi derivati che riassumono in modo efficace le informazioni presenti nell'insieme originale di campi.

Esempio

```
node = stream.create("factor", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["BP", "Na", "K"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Factor_Age")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("method", "GLS")
# Expert options
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("complete_records", True)
node.setPropertyValue("matrix", "Covariance")
node.setPropertyValue("max_iterations", 30)
node.setPropertyValue("extract_factors", "ByFactors")
node.setPropertyValue("min_eigenvalue", 3.0)
node.setPropertyValue("max_factor", 7)
node.setPropertyValue("sort_values", True)
node.setPropertyValue("hide_values", True)
node.setPropertyValue("hide_below", 0.7)
# "Rotation" section
node.setPropertyValue("rotation", "DirectOblimin")
node.setPropertyValue("delta", 0.3)
node.setPropertyValue("kappa", 7.0)
```

Tabella 114. proprietà factornode

| Proprietà factornode | Valori | Descrizione proprietà |
|----------------------|---------------------|---|
| inputs | [campo1 ... campoN] | I modelli fattoriali/PCA utilizzano un elenco di campi di input, ma nessun campo obiettivo. I campi peso e frequenza non sono utilizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |

Tabella 114. proprietà factornode (Continua)

| Proprietà factornode | Valori | Descrizione proprietà |
|----------------------|--|---|
| method | PC ULS GLS ML PAF Alpha Image | |
| mode | Simple Expert | |
| max_iterations | <i>numero</i> | |
| complete_records | <i>indicatore</i> | |
| matrix | Correlation Covarianza | |
| extract_factors | ByEigenvalues ByFactors | |
| min_eigenvalue | <i>numero</i> | |
| max_factor | <i>numero</i> | |
| rotation | None Varimax DirectOblimin Equamax Quartimax Promax | |
| delta | <i>numero</i> | Se si seleziona DirectOblimin come tipo di dati di rotazione, è possibile specificare un valore per delta. Se non si specifica un valore, per delta verrà utilizzato il valore di default. |
| kappa | <i>numero</i> | Se si seleziona Promax come tipo di dati di rotazione, è possibile specificare un valore per kappa. Se non si specifica un valore, per kappa verrà utilizzato il valore di default. |
| sort_values | <i>indicatore</i> | |
| hide_values | <i>indicatore</i> | |
| hide_below | <i>numero</i> | |

Proprietà featureselectionnode



Il nodo Selezione funzioni effettua lo screening dei campi di input, rimuovendoli in base a un insieme di criteri quali la percentuale di valori mancanti. Classifica quindi gli input restanti in ordine di importanza rispetto a un determinato obiettivo. Per esempio, dato un insieme di dati con centinaia di input potenziali, quali sono quelli con la maggiore probabilità di essere utili nella modellazione di risultati clinici?

Esempio


```

node = stream.create("featureselection", "My node")
node.setPropertyValue("screen_single_category", True)
node.setPropertyValue("max_single_category", 95)
node.setPropertyValue("screen_missing_values", True)
node.setPropertyValue("max_missing_values", 80)
node.setPropertyValue("criteria", "Likelihood")
node.setPropertyValue("unimportant_below", 0.8)
node.setPropertyValue("important_above", 0.9)
node.setPropertyValue("important_label", "Check Me Out!")
node.setPropertyValue("selection_mode", "TopN")
node.setPropertyValue("top_n", 15)

```

Per un esempio più dettagliato di creazione e applicazione di un modello di selezione funzioni, vedere in in.

Tabella 115. proprietà featureselectionnode

| Proprietà featureselectionnode | Valori | Descrizione proprietà |
|--------------------------------|---|---|
| obiettivo | <i>campo</i> | I modelli di selezione funzioni classificano i predittori rispetto all'obiettivo specificato. I campi peso e frequenza non sono utilizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| screen_single_category | <i>indicatore</i> | Se True, esegue lo screening dei campi che hanno troppi record che rientrano nella stessa categoria rispetto al numero totale di record. |
| max_single_category | <i>numero</i> | Specifica la soglia utilizzata quando screen_single_category è True. |
| screen_missing_values | <i>indicatore</i> | Se True, esegue lo screening dei campi con troppi valori mancanti, espressi come percentuale del numero totale di record. |
| max_missing_values | <i>numero</i> | |
| screen_num_categories | <i>indicatore</i> | Se True, esegue lo screening dei campi con troppe categorie rispetto al numero totale di record. |
| max_num_categories | <i>numero</i> | |
| screen_std_dev | <i>indicatore</i> | Se True, esegue lo screening dei campi con una deviazione standard inferiore o uguale al minimo specificato. |
| min_std_dev | <i>numero</i> | |
| screen_coeff_of_var | <i>indicatore</i> | Se True, esegue lo screening dei campi con un coefficiente di varianza inferiore o uguale al minimo specificato. |
| min_coeff_of_var | <i>numero</i> | |
| criteria | Pearson Likelihood CramersV Lambda | Quando si classificano i predittori categoriali rispetto a un obiettivo categoriale, specifica la misura sulla quale si basa il valore di importanza. |
| unimportant_below | <i>numero</i> | Specifica i valori p di soglia utilizzati per classificare variabili quali importante, marginale o non importante. Accetta i valori compresi fra 0.0 e 1.0. |

Tabella 115. proprietà featureselectionnode (Continua)

| Proprietà featureselectionnode | Valori | Descrizione proprietà |
|--------------------------------|--|---|
| important_above | numero | Accetta i valori compresi fra 0.0 e 1.0. |
| unimportant_label | stringa | Specifica l'etichetta per la classificazione non importante. |
| marginal_label | stringa | |
| important_label | stringa | |
| selection_mode | ImportanceLevel ImportanceValue TopN | |
| select_important | indicatore | Quando selection_mode è impostata su ImportanceLevel, specifica se selezionare i campi importanti. |
| select_marginal | indicatore | Quando selection_mode è impostata su ImportanceLevel, specifica se selezionare i campi marginali. |
| select_unimportant | indicatore | Quando selection_mode è impostata su ImportanceLevel, specifica se selezionare i campi non importanti. |
| importance_value | numero | Quando selection_mode è impostata su ImportanceValue, specifica il valore di interruzione da utilizzare. Accetta i valori compresi tra 0 e 100. |
| top_n | numero intero | Quando selection_mode è impostata su TopN, specifica il valore di interruzione da utilizzare. Accetta i valori compresi tra 0 e 1000. |

Proprietà genlinnode



Il modello Lineare generalizzato amplia il modello lineare generale in modo che la variabile dipendente venga linearmente correlata ai fattori e alle covariate tramite una funzione di collegamento specifica. Inoltre, il modello consente alla variabile dipendente di avere una distribuzione non normale. Copre la funzionalità di un grande numero di modelli statistici, inclusi modelli di regressione lineare, modelli di regressione logistica, modelli loglineari per dati dei conteggi e modelli di sopravvivenza censurati per intervallo.

Esempio

```
node = stream.create("genlin", "My node")
node.setPropertyValue("model_type", "MainAndAllTwoWayEffects")
node.setPropertyValue("offset_type", "Variable")
node.setPropertyValue("offset_field", "Claimant")
```

Tabella 116. proprietà genlinnode

| Proprietà genlinnode | Valori | Descrizione proprietà |
|----------------------|--------|---|
| obiettivo | campo | I modelli lineari generalizzati richiedono un solo campo obiettivo (che deve essere nominale o flag) e uno o più campi di input. È anche possibile specificare un campo peso. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |

Tabella 116. proprietà genl innode (Continua)

| Proprietà genl innode | Valori | Descrizione proprietà |
|--------------------------|---|--|
| use_weight | indicatore | |
| weight_field | campo | Il tipo di campo è solo continuo. |
| target_represents_trials | indicatore | |
| trials_type | Variable FixedValue | |
| trials_field | campo | Il tipo di campo è continuo, flag o ordinale. |
| trials_number | numero | Il valore di default è 10. |
| model_type | MainEffects MainAndAllTwoWayEffects | |
| offset_type | Variable FixedValue | |
| offset_field | campo | Il tipo di campo è solo continuo. |
| offset_value | numero | Deve essere un numero reale. |
| base_category | Last First | |
| include_intercept | indicatore | |
| mode | Simple Expert | |
| distribution | BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL | IGAUSS: gaussiana inversa. NEGBIN: binomiale negativa. |
| negbin_para_type | Specify Estimate | |
| negbin_parameter | numero | Il valore di default è 1. Deve contenere un numero reale non negativo. |
| tweedie_parameter | numero | |
| link_function | IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPOWER PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT | CLOGLOG: doppia logaritmica complementare. LOGC: log-complemento. NEGBIN: binomiale negativa. NLOGLOG: doppia logaritmica negativa. CUMCAUCHIT: cauchit cumulativa. CUMCLOGLOG: log-log complementare cumulativa. CUMLOGIT: logit cumulativa. CUMNLOGLOG: log-log negativa cumulativa. CUMPROBIT: probit cumulativa. |

Tabella 116. proprietà genl innode (Continua)

| Proprietà genl innode | Valori | Descrizione proprietà |
|---------------------------|---|---|
| power | numero | Il valore deve essere un numero reale diverso da zero. |
| method | Hybrid Fisher NewtonRaphson | |
| max_fisher_iterations | numero | Il valore di default è 1; sono consentiti solo numeri interi positivi. |
| scale_method | MaxLikelihoodEstimate Deviance PearsonChiSquare FixedValue | |
| scale_value | numero | Il valore di default è 1; deve essere maggiore di 0. |
| covariance_matrix | ModelEstimator RobustEstimator | |
| max_iterations | numero | Il valore di default è 100; solo numeri interi non negativi. |
| max_step_halving | numero | Il valore di default è 5; solo numeri interi positivi. |
| check_separation | indicatore | |
| start_iteration | numero | Il valore di default è 20; sono consentiti solo numeri interi positivi. |
| estimates_change | indicatore | |
| estimates_change_min | numero | Il valore di default è 1E-006; sono consentiti solo numeri positivi. |
| estimates_change_type | Absolute Relative | |
| loglikelihood_change | indicatore | |
| loglikelihood_change_min | numero | Sono consentiti solo numeri positivi. |
| loglikelihood_change_type | Absolute Relative | |
| hessian_convergence | indicatore | |
| hessian_convergence_min | numero | Sono consentiti solo numeri positivi. |
| hessian_convergence_type | Absolute Relative | |
| case_summary | indicatore | |
| contrast_matrices | indicatore | |
| descriptive_statistics | indicatore | |
| estimable_functions | indicatore | |
| model_info | indicatore | |
| iteration_history | indicatore | |
| goodness_of_fit | indicatore | |
| print_interval | numero | Il valore di default è 1; deve essere un numero intero positivo. |
| model_summary | indicatore | |

Tabella 116. proprietà genlnode (Continua)

| Proprietà genlnode | Valori | Descrizione proprietà |
|---------------------------------|--|--------------------------------|
| lagrange_multiplier | indicatore | |
| parameter_estimates | indicatore | |
| include_exponential | indicatore | |
| covariance_estimates | indicatore | |
| correlation_estimates | indicatore | |
| analysis_type | TypeI TypeIII TypeIAndTypeIII | |
| statistics | Wald LR | |
| citype | Wald Profile | |
| tolerancelevel | numero | Il valore di default è 0.0001. |
| confidence_interval | numero | Il valore di default è 95. |
| loglikelihood_function | Full Kernel | |
| singularity_tolerance | 1E-007 1E-008 1E-009 1E-010 1E-011 1E-012 | |
| value_order | Crescente Decrescente DataOrder | |
| calculate_variable_importance | indicatore | |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |
| adjusted_propensity_partition | Test Validation | |

Proprietà glmmnode



Un modello misto lineare generalizzato (GLMM) estende il modello lineare in modo che l'obiettivo possa avere una distribuzione non normale, sia linearmente correlato ai fattori e alle covariate tramite una funzione di collegamento specifica e in modo che le osservazioni possano essere correlate. I modelli misti lineari generalizzati includono un'ampia gamma di modelli, dalla regressione lineare semplice ai modelli multilivello complessi per i dati longitudinali non normali.

Tabella 117. proprietà glmmnode.

| Proprietà glmmnode | Valori | Descrizione proprietà |
|-----------------------|-------------|---|
| residual_subject_spec | strutturata | La combinazione di valori dei campi categoriali specificati che definisce in modo univoco i soggetti all'interno dell'insieme di dati |

Tabella 117. proprietà glmmnode (Continua).

| Proprietà glmmnode | Valori | Descrizione proprietà |
|-----------------------------|---|---|
| repeated_measures | <i>strutturata</i> | I campi utilizzati per identificare le osservazioni ripetute. |
| residual_group_spec | [<i>campo1 ... campoN</i>] | I campi che definiscono insiemi indipendenti di parametri di covarianza a effetti ripetuti. |
| residual_covariance_type | Diagonale AR1 ARMA11 COMPOUND_SYMMETRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS | Specifica la struttura di covarianza per i residui. |
| custom_target | <i>indicatore</i> | Indica se utilizzare la destinazione definita nel nodo upstream (false) o la destinazione personalizzata specificata da <i>target_field</i> (true). |
| target_field | <i>campo</i> | Il campo da utilizzare come destinazione se <i>custom_target</i> è true. |
| use_trials | <i>indicatore</i> | Indica se un campo o valore aggiuntivo che specifica il numero di prove deve essere utilizzato quando la risposta obiettivo rappresenta un numero di eventi che si verificano in un insieme di prove. Il valore di default è false. |
| use_field_or_value | Campo Value | Indica se il campo (default) o valore viene utilizzato per specificare il numero di prove. |
| trials_field | <i>campo</i> | Campo da utilizzare per specificare il numero di prove. |
| trials_value | <i>numero intero</i> | Valore da utilizzare per specificare il numero di prove. Se specificato, il valore minimo è 1. |
| use_custom_target_reference | <i>indicatore</i> | Indica se la categoria di riferimento personalizzata deve essere utilizzata per un target di categoria. Il valore di default è false. |
| target_reference_value | <i>stringa</i> | La categoria di riferimento da utilizzare se <i>use_custom_target_reference</i> è true. |
| dist_link_combination | Nominale Logit GammaLog BinomialLogit PoissonLog BinomialProbit NegbinLog BinomialLogC Custom | I modelli comuni per la distribuzione dei valori dell'obiettivo. Scegliere Custom per specificare una distribuzione dall'elenco fornito da <i>target_distribution</i> . |

Tabella 117. proprietà glmmnode (Continua).

| Proprietà glmmnode | Valori | Descrizione proprietà |
|-------------------------|--|--|
| target_distribution | Normal Binomial Multinomial Gamma Inverso NegativeBinomial Poisson | Distribuzione dei valori per l'obiettivo quando dist_link_combination è Custom. |
| link_function_type | Identità LogC Log CLOGLOG Logit NLOGLOG PROBIT POWER CAUCHIT | Funzione di collegamento per correlare i valori obiettivo. Se target_distribution è Binomial è possibile utilizzare una qualsiasi delle funzioni di collegamento elencate. Se target_distribution è Multinomial è possibile utilizzare CLOGLOG, CAUCHIT, LOGIT, NLOGLOG oppure PROBIT. Se target_distribution è diverso da Binomial o Multinomial è possibile utilizzare IDENTITY, LOG oppure POWER. |
| link_function_param | numero | Il valore del parametro della funzione di collegamento da utilizzare. Applicabile solo se normal_link_function o link_function_type è POWER. |
| use_predefined_inputs | indicatore | Indica se i campi a effetto fisso devono essere quelli definiti a monte come campi di input (true) o quelli di fixed_effects_list (false). Il valore di default è false. |
| fixed_effects_list | strutturata | Se use_predefined_inputs è false, specifica i campi di input da utilizzare come campi a effetto fisso. |
| use_intercept | indicatore | Se true (default), include l'intercettazione nel modello. |
| random_effects_list | strutturata | Elenco dei campi da specificare come effetti random. |
| regression_weight_field | campo | Campo da utilizzare come campo del peso dell'analisi. |
| use_offset | None offset_value offset_field | Indica il modo in cui viene specificato l'offset. Il valore None indica che non viene utilizzato nessun offset. |
| offset_value | numero | Il valore da utilizzare per l'offset se use_offset è impostato su offset_value. |
| offset_field | campo | Il campo da utilizzare per il valore offset se use_offset è impostato su offset_field. |
| target_category_order | Crescente Descending Data | Criterio di ordinamento per i target di categoria. Il valore Data specifica l'utilizzo del criterio di ordinamento trovato nei dati. L'impostazione di default è Ascending. |
| inputs_category_order | Crescente Descending Data | Criterio di ordinamento per i predittori di categoria. Il valore Data specifica l'utilizzo del criterio di ordinamento trovato nei dati. L'impostazione di default è Ascending. |

Tabella 117. proprietà glmmnode (Continua).

| Proprietà glmmnode | Valori | Descrizione proprietà |
|---------------------------------|-----------------------------|--|
| max_iterations | numero intero | Numero massimo di iterazioni che l'algoritmo eseguirà. Un numero intero non negativo; l'impostazione di default è 100. |
| confidence_level | numero intero | Livello di confidenza utilizzato per calcolare le stime di intervallo dei coefficienti del modello. Un numero intero non negativo; il massimo è 100, l'impostazione di default è 95. |
| degrees_of_freedom_method | Fixed Varied | Specifica la modalità di calcolo dei gradi di libertà per i test di significatività. |
| test_fixed_effects_coefficients | Modello Robust | Il metodo per il calcolo della matrice di covarianza delle stime dei parametri. |
| use_p_converge | indicatore | Opzione per la convergenza dei parametri. |
| p_converge | numero | Vuoto, o qualsiasi valore positivo. |
| p_converge_type | Assoluti Relative | |
| use_l_converge | indicatore | Opzione per la convergenza di verosimiglianza logaritmica. |
| l_converge | numero | Vuoto, o qualsiasi valore positivo. |
| l_converge_type | Assoluti Relative | |
| use_h_converge | indicatore | Opzione per la convergenza hessiana. |
| h_converge | numero | Vuoto, o qualsiasi valore positivo. |
| h_converge_type | Assoluti Relative | |
| max_fisher_steps | numero intero | |
| singularity_tolerance | numero | |
| use_model_name | indicatore | Indica se specificare un nome personalizzato per il modello (true) o utilizzare il nome generato dal sistema (false). Il valore di default è false. |
| model_name | stringa | Se use_model_name è true, specifica il nome del modello da utilizzare. |
| confidence | onProbability onIncrease | Base per il calcolo del valore di confidenza del punteggio: probabilità prevista più alta o differenza tra le probabilità più alte e la seconda massima prevista. |
| score_category_probabilities | indicatore | Se true, produce le probabilità previste per i target di categoria. Il valore di default è false. |
| max_categories | numero intero | Se score_category_probabilities è true, specifica il numero massimo di categorie da salvare. |
| score_propensity | indicatore | Se true, produce punteggi di propensione per i campi obiettivo di tipo indicatore che indicano la probabilità del risultato "true" per il campo. |

Tabella 117. proprietà glmmnode (Continua).

| Proprietà glmmnode | Valori | Descrizione proprietà |
|------------------------------|----------------------------------|---|
| emeans | struttura | Per ogni campo relativo alla categoria dall'elenco a effetti fissi, specifica se produrre le medie marginali stimate. |
| covariance_list | struttura | Per ogni campo continuo dall'elenco effetti fissi, specifica se utilizzare la media o un valore personalizzato quando si calcola le medie marginali stimate. |
| mean_scale | Original Trasformata | Specifica se calcolare le medie marginali stimate in base alla scala originale dell'obiettivo (default) o in base alla trasformazione della funzione di collegamento. |
| comparison_adjustment_method | LSD SEQBONFERRONI SEQSIDAK | Metodo di regolazione da utilizzare quando si esegue il test sull'ipotesi con più contrasti. |

Proprietà gle



Il nodo GLE è disponibile solo se si è connessi a IBM SPSS Analytic Server. Un modello GLE estende il modello lineare così che l'obiettivo possa avere una distribuzione non normale, possa essere correlato in modo lineare ai fattori e covariate mediante una specifica funzione di collegamento e le osservazioni possano essere correlate. I modelli misti lineari generalizzati includono un'ampia gamma di modelli, dalla regressione lineare semplice ai modelli multilivello complessi per i dati longitudinali non normali.

Tabella 118. Proprietà gle

| Proprietà gle | Valori | Descrizione proprietà |
|-----------------------------|----------------|---|
| custom_target | indicatore | Indica se utilizzare la destinazione definita nel nodo upstream (false) o la destinazione personalizzata specificata da target_field (true). |
| target_field | campo | Il campo da utilizzare come destinazione se custom_target è true. |
| use_trials | indicatore | Indica se un campo o valore aggiuntivo che specifica il numero di prove deve essere utilizzato quando la risposta obiettivo rappresenta un numero di eventi che si verificano in un insieme di prove. Il valore di default è false. |
| use_trials_field_or_value | Campo Value | Indica se il campo (default) o valore viene utilizzato per specificare il numero di prove. |
| trials_field | campo | Campo da utilizzare per specificare il numero di prove. |
| trials_value | integer | Valore da utilizzare per specificare il numero di prove. Se specificato, il valore minimo è 1. |
| use_custom_target_reference | indicatore | Indica se la categoria di riferimento personalizzata deve essere utilizzata per un target di categoria. Il valore di default è false. |
| target_reference_value | stringa | La categoria di riferimento da utilizzare se use_custom_target_reference è true. |

Tabella 118. Proprietà gle (Continua)

| Proprietà gle | Valori | Descrizione proprietà |
|-----------------------|--|---|
| dist_link_combination | NormalIdentity GammaLog PoissonLog NegbinLog TweedieIdentity NominalLogit BinomialLogit BinomialProbit BinomialLogC CUSTOM | I modelli comuni per la distribuzione dei valori dell'obiettivo. Scegliere CUSTOM per specificare una distribuzione dall'elenco fornito da target_distribution. |
| target_distribution | Normal Binomial Multinomial Gamma INVERSE_GAUSS NEG_BINOMIAL Poisson TWEEDIE UNKNOWN | Distribuzione dei valori per l'obiettivo quando dist_link_combination è Custom. |
| link_function_type | UNKNOWN IDENTITY LOG LOGIT PROBIT COMPL_LOG_LOG POWER LOG_COMPL NEG_LOG_LOG ODDS_POWER NEG_BINOMIAL GEN_LOGIT CUMUL_LOGIT CUMUL_PROBIT CUMUL_COMPL_LOG_LOG CUMUL_NEG_LOG_LOG CUMUL_CAUCHIT | Funzione di collegamento per correlare i valori obiettivo per i predittori. Se target_distribution è Binomial è possibile utilizzare: UNKNOWN IDENTITY LOG LOGIT PROBIT COMPL_LOG_LOG POWER LOG_COMPL NEG_LOG_LOG ODDS_POWER Se target_distribution è NEG_BINOMIAL è possibile utilizzare: NEG_BINOMIAL. Se target_distribution è UNKNOWN è possibile utilizzare: GEN_LOGIT CUMUL_LOGIT CUMUL_PROBIT CUMUL_COMPL_LOG_LOG CUMUL_NEG_LOG_LOG CUMUL_CAUCHIT |
| link_function_param | <i>numero</i> | Valore del parametro Tweedie da utilizzare. Applicabile solo se normal_link_function o link_function_type è POWER. |
| tweedie_param | <i>numero</i> | Il valore del parametro della funzione di collegamento da utilizzare. Applicabile solo se dist_link_combination è impostata su TweedieIdentity o link_function_type è TWEEDIE. |
| use_predefined_inputs | <i>indicatore</i> | Indica se i campi a effetto del modello devono essere quelli definiti a monte come campi di input (true) o quelli di fixed_effects_list (false). |

Tabella 118. Proprietà gle (Continua)

| Proprietà gle | Valori | Descrizione proprietà |
|---------------------------------|---|--|
| model_effects_list | <i>strutturato</i> | Se <i>use_predefined_inputs</i> è false, specifica i campi di input da utilizzare come campi a effetto del modello. |
| use_intercept | <i>indicatore</i> | Se true (default), include l'intercettazione nel modello. |
| regression_weight_field | <i>campo</i> | Campo da utilizzare come campo del peso dell'analisi. |
| use_offset | None Value Variable | Indica il modo in cui viene specificato l'offset. Il valore None indica che non viene utilizzato nessun offset. |
| offset_value | <i>numero</i> | Il valore da utilizzare per l'offset se <i>use_offset</i> è impostato su <i>offset_value</i> . |
| offset_field | <i>campo</i> | Il campo da utilizzare per il valore offset se <i>use_offset</i> è impostato su <i>offset_field</i> . |
| target_category_order | Crescente Decrescente | Criterio di ordinamento per i target di categoria. L'impostazione di default è Ascending. |
| inputs_category_order | Crescente Decrescente | Criterio di ordinamento per i predittori di categoria. L'impostazione di default è Ascending. |
| max_iterations | <i>integer</i> | Numero massimo di iterazioni che l'algoritmo eseguirà. Un numero intero non negativo; l'impostazione di default è 100. |
| confidence_level | <i>numero</i> | Livello di confidenza utilizzato per calcolare le stime di intervallo dei coefficienti del modello. Un numero intero non negativo; il massimo è 100, l'impostazione di default è 95. |
| test_fixed_effects_coefficients | Modello Robust | Il metodo per il calcolo della matrice di covarianza delle stime dei parametri. |
| detect_outliers | <i>indicatore</i> | Quando impostata su true, l'algoritmo trova i valori anomali di influenza per tutte le distribuzioni tranne che per la distribuzione multinomiale. |
| conduct_trend_analysis | <i>indicatore</i> | Quando impostata su true, l'algoritmo effettua l'analisi sulla tendenza per il grafico a dispersione. |
| estimation_method | FISHER_SCORING NEWTON_RAPHSON HYBRID | Specifica l'algoritmo di stima della massima verosomiglianza. |
| max_fisher_iterations | <i>integer</i> | Se si utilizza FISHER_SCORING estimation_method, il numero massimo di iterazioni. Minimo 0, massimo 20. |
| scale_parameter_method | MLE FIXED DEVIANCE PEARSON_CHISQUARE | Specifica il metodo da utilizzare per la stima del parametro di scala. |
| scale_value | <i>numero</i> | Disponibile solo se <i>scale_parameter_method</i> è impostata su Fixed. |
| negative_binomial_method | MLE FIXED | Specifica il metodo da utilizzare per la stima del parametro ausiliario binomiale negativo. |
| negative_binomial_value | <i>numero</i> | Disponibile solo se <i>negative_binomial_method</i> è impostato su Fixed. |

Tabella 118. Proprietà gle (Continua)

| Proprietà gle | Valori | Descrizione proprietà |
|-----------------------------|---|--|
| use_p_converge | <i>indicatore</i> | Opzione per la convergenza dei parametri. |
| p_converge | <i>numero</i> | Vuoto, o qualsiasi valore positivo. |
| p_converge_type | <i>indicatore</i> | True = Absolute, False = Relative |
| use_l_converge | <i>indicatore</i> | Opzione per la convergenza di verosimiglianza logaritmica. |
| l_converge | <i>numero</i> | Vuoto, o qualsiasi valore positivo. |
| l_converge_type | <i>indicatore</i> | True = Absolute, False = Relative |
| use_h_converge | <i>indicatore</i> | Opzione per la convergenza hessiana. |
| h_converge | <i>numero</i> | Vuoto, o qualsiasi valore positivo. |
| h_converge_type | <i>indicatore</i> | True = Absolute, False = Relative |
| max_iterations | <i>integer</i> | Numero massimo di iterazioni che l'algoritmo eseguirà. Un numero intero non negativo; l'impostazione di default è 100. |
| sing_tolerance | <i>integer</i> | |
| use_model_selection | <i>indicatore</i> | Abilita i controlli della soglia del parametro e del metodo di selezione del modello. |
| method | LASSO ELASTIC_NET FORWARD_STEPWISE RIDGE | Determina il metodo di selezione del modello o se si utilizza Ridge il metodo di regolarizzazione utilizzato. |
| detect_two_way_interactions | <i>indicatore</i> | Quando True il modello automaticamente rileverà le interazioni a due vie tra i campi di input. Questo controllo dovrebbe essere abilitato solo se il modello è solo di effetti principali (cioè in cui l'utente non ha creato altri effetti di ordine superiore) e se il Metodo selezionato è Forward Stepwise, Lasso o Elastic Net. |
| automatic_penalty_params | <i>indicatore</i> | Disponibile solo se il method di selezione del modello è Lasso o Elastic Net. Utilizzare questa funzione per immettere i parametri di penalità associati ai metodi di selezione della variabile Lasso o Elastic Net. Se True, vengono utilizzati i valori predefiniti. Se False, i parametri di penalità vengono abilitati e possono essere immessi valori personalizzati. |
| lasso_penalty_param | <i>numero</i> | Disponibile solo se il method di selezione del modello è Lasso o Elastic Net e automatic_penalty_params è False. Specifica il valore del parametro di penalità per Lasso. |
| elastic_net_penalty_param1 | <i>numero</i> | Disponibile solo se il method di selezione del modello è Lasso o Elastic Net e automatic_penalty_params è False. Specifica il valore del parametro di penalità per il parametro 1 di Elastic Net. |

Tabella 118. Proprietà gle (Continua)

| Proprietà gle | Valori | Descrizione proprietà |
|----------------------------|-------------------|--|
| elastic_net_penalty_param2 | <i>numero</i> | Disponibile solo se il method di selezione del modello è Lasso o Elastic Net e automatic_penalty_params è False. Specifica il valore del parametro di penalità per il parametro 2 di Elastic Net. |
| probability_entry | <i>numero</i> | Disponibile solo se il method selezionato è Forward Stepwise. Specifica il livello di significatività del criterio statistico f per inclusione dell'effetto. |
| probability_removal | <i>numero</i> | Disponibile solo se il method selezionato è Forward Stepwise. Specifica il livello di significatività del criterio statistico f per la rimozione dell'effetto. |
| use_max_effects | <i>indicatore</i> | Disponibile solo se il method selezionato è Forward Stepwise. Abilita il controllo max_effects. Quando False il numero predefinito di effetti inclusi dovrebbe essere uguale al numero totale di effetti forniti al modello, meno l'intercettazione. |
| max_effects | <i>integer</i> | Specificare il numero massimo di effetti quando si utilizza il metodo di creazione Stepwise in avanti. |
| use_max_steps | <i>indicatore</i> | Abilita il controllo max_steps. Quando False il numero predefinito di fasi dovrebbe essere uguale a tre volte il numero di effetti forniti al modello, escluso l'intercettazione. |
| max_steps | <i>integer</i> | Specifica il numero massimo di fasi da compiere quando si utilizza il method di creazione Forward Stepwise. |
| use_model_name | <i>indicatore</i> | Indica se specificare un nome personalizzato per il modello (true) o utilizzare il nome generato dal sistema (false). Il valore di default è false. |
| model_name | <i>stringa</i> | Se use_model_name è true, specifica il nome del modello da utilizzare. |
| usePI | <i>indicatore</i> | Se true, l'importanza predittore viene calcolata. |

Proprietà kmeansnode



Il nodo Medie K raggruppa l'insieme di dati in gruppi distinti (o cluster). Il metodo definisce un numero fisso di cluster, esegue un'assegnazione iterativa dei record ai cluster e modifica i centri di cluster finché un'ulteriore ridefinizione non consente più un miglioramento del modello. Invece di tentare di prevedere un risultato, il nodo K-medie utilizza un processo denominato apprendimento non supervisionato per scoprire gli schemi nell'insieme di campi di input.

Esempio

```

node = stream.create("kmeans", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["Cholesterol", "BP", "Drug", "Na", "K", "Age"])
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Kmeans_allinputs")
node.setPropertyValue("num_clusters", 9)
node.setPropertyValue("gen_distance", True)
node.setPropertyValue("cluster_label", "Number")
node.setPropertyValue("label_prefix", "Kmeans_")
node.setPropertyValue("optimize", "Speed")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("stop_on", "Custom")
node.setPropertyValue("max_iterations", 10)
node.setPropertyValue("tolerance", 3.0)
node.setPropertyValue("encoding_value", 0.3)

```

Tabella 119. proprietà kmeansnode

| Proprietà kmeansnode | Valori | Descrizione proprietà |
|----------------------|---------------------|--|
| inputs | [campo1 ... campoN] | I modelli Medie K eseguono l'analisi dei cluster su un insieme di campi di input ma non utilizzano un campo obiettivo. I campi peso e frequenza non sono utilizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| num_clusters | numero | |
| gen_distance | indicatore | |
| cluster_label | String Number | |
| label_prefix | stringa | |
| mode | Simple Expert | |
| stop_on | Default Custom | |
| max_iterations | numero | |
| tolerance | numero | |
| encoding_value | numero | |
| optimize | Speed Memory | Utilizzare per specificare se ottimizzare la velocità o la memoria durante la creazione del modello. |

Proprietà knnnode



Il nodo Elemento vicino più prossimo K (KNN) associa un nuovo caso alla categoria o valore degli oggetti K più vicini ad esso nello spazio predittore, dove K è un numero intero. I casi simili sono vicini gli uni agli altri, mentre i casi dissimili sono distanti gli uni dagli altri.

Esempio

```

node = stream.create("knn", "My node")
# Objectives tab
node.setPropertyValue("objective", "Custom")
# Settings tab - Neighbors panel
node.setPropertyValue("automatic_k_selection", False)
node.setPropertyValue("fixed_k", 2)
node.setPropertyValue("weight_by_importance", True)
# Settings tab - Analyze panel
node.setPropertyValue("save_distances", True)

```

Tabella 120. proprietà knnnode

| Proprietà knnnode | Valori | Descrizione proprietà |
|---------------------------------|--|--|
| analysis | PredictTarget IdentifyNeighbors | |
| objective | Balance Speed Accuracy Custom | |
| normalize_ranges | <i>indicatore</i> | |
| use_case_labels | <i>indicatore</i> | Selezionare la casella per abilitare l'opzione successiva. |
| case_labels_field | <i>campo</i> | |
| identify_focal_cases | <i>indicatore</i> | Selezionare la casella per abilitare l'opzione successiva. |
| focal_cases_field | <i>campo</i> | |
| automatic_k_selection | <i>indicatore</i> | |
| fixed_k | <i>numero intero</i> | Attiva solo se automatic_k_selection è False. |
| minimum_k | <i>numero intero</i> | Attiva solo se automatic_k_selection è True. |
| maximum_k | <i>numero intero</i> | |
| distance_computation | Euclidean CityBlock | |
| weight_by_importance | <i>indicatore</i> | |
| range_predictions | Mean Median | |
| perform_feature_selection | <i>indicatore</i> | |
| forced_entry_inputs | [<i>campo1 ... campoN</i>] | |
| stop_on_error_ratio | <i>indicatore</i> | |
| number_to_select | <i>numero intero</i> | |
| minimum_change | <i>numero</i> | |
| validation_fold_assign_by_field | <i>indicatore</i> | |
| number_of_folds | <i>numero intero</i> | Attiva solo se validation_fold_assign_by_field è False |
| set_random_seed | <i>indicatore</i> | |
| random_seed | <i>numero</i> | |
| folds_field | <i>campo</i> | Attiva solo se validation_fold_assign_by_field è True |
| all_probabilities | <i>indicatore</i> | |

Tabella 120. proprietà knnnode (Continua)

| Proprietà knnnode | Valori | Descrizione proprietà |
|---------------------------------|--------------------|-----------------------|
| save_distances | indicatore | |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |
| adjusted_propensity_partition | Test Validation | |

Proprietà kohonennode



Il nodo Kohonen genera un tipo di rete neurale che può essere utilizzato per raggruppare l'insieme di dati in gruppi distinti. Al termine dell'apprendimento della rete, i record analoghi dovranno essere vicini nella mappa di output, mentre i record diversi saranno a notevole distanza. Per identificare le unità forti, è possibile controllare il numero di osservazioni catturate da ciascuna unità nel nugget del modello. In questo modo è possibile avere un'idea del numero appropriato di cluster.

Esempio

```
node = stream.create("kohonen", "My node")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Symbolic Cluster")
node.setPropertyValue("stop_on", "Time")
node.setPropertyValue("time", 1)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 12345)
node.setPropertyValue("optimize", "Speed")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("width", 3)
node.setPropertyValue("length", 3)
node.setPropertyValue("decay_style", "Exponential")
node.setPropertyValue("phase1_neighborhood", 3)
node.setPropertyValue("phase1_eta", 0.5)
node.setPropertyValue("phase1_cycles", 10)
node.setPropertyValue("phase2_neighborhood", 1)
node.setPropertyValue("phase2_eta", 0.2)
node.setPropertyValue("phase2_cycles", 75)
```

Tabella 121. proprietà kohonennode

| Proprietà kohonennode | Valori | Descrizione proprietà |
|-----------------------|---------------------|--|
| inputs | [campo1 ... campoN] | I modelli Kohonen utilizzano un elenco di campi di input, ma nessun campo obiettivo. I campi frequenza e peso non sono utilizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| continue | indicatore | |
| show_feedback | indicatore | |
| stop_on | Default Time | |
| time | numero | |

Tabella 121. proprietà kohonennode (Continua)

| Proprietà kohonennode | Valori | Descrizione proprietà |
|-----------------------|-----------------------|--|
| optimize | Speed Memory | Utilizzare per specificare se ottimizzare la velocità o la memoria durante la creazione del modello. |
| cluster_label | <i>indicatore</i> | |
| mode | Simple Expert | |
| width | <i>numero</i> | |
| length | <i>numero</i> | |
| decay_style | Linear Exponential | |
| phase1_neighborhood | <i>numero</i> | |
| phase1_eta | <i>numero</i> | |
| phase1_cycles | <i>numero</i> | |
| phase2_neighborhood | <i>numero</i> | |
| phase2_eta | <i>numero</i> | |
| phase2_cycles | <i>numero</i> | |

Proprietà linearnode



I modelli di regressione lineare prevedono un target continuo basato sulle relazioni lineari tra l'obiettivo e uno o più predittori.

Esempio

```
node = stream.create("linear", "My node")
# Build Options tab - Objectives panel
node.setPropertyValue("objective", "Standard")
# Build Options tab - Model Selection panel
node.setPropertyValue("model_selection", "BestSubsets")
node.setPropertyValue("criteria_best_subsets", "ASE")
# Build Options tab - Ensembles panel
node.setPropertyValue("combining_rule_categorical", "HighestMeanProbability")
```

Tabella 122. Proprietà linearnode.

| Proprietà linearnode | Valori | Descrizione proprietà |
|----------------------------------|--|--|
| obiettivo | <i>campo</i> | Specifica un singolo campo obiettivo. |
| inputs | [<i>campo1 ... campoN</i>] | I campi predittore utilizzati dal modello. |
| continue_training_existing_model | <i>indicatore</i> | |
| objective | Standard Bagging Boosting psm | psm viene utilizzato per insiemi di dati di grandi dimensioni e richiede una connessione Server. |
| use_auto_data_preparation | <i>indicatore</i> | |

Tabella 122. Proprietà linearnode (Continua).

| Proprietà linearnode | Valori | Descrizione proprietà |
|---------------------------|---|-----------------------|
| confidence_level | numero | |
| model_selection | ForwardStepwise BestSubsets None | |
| criteria_forward_stepwise | AICC Fstatistics AdjustedRSquare ASE | |
| probability_entry | numero | |
| probability_removal | numero | |
| use_max_effects | indicatore | |
| max_effects | numero | |
| use_max_steps | indicatore | |
| max_steps | numero | |
| criteria_best_subsets | AICC AdjustedRSquare ASE | |
| combining_rule_continuous | Mean Median | |
| component_models_n | numero | |
| use_random_seed | indicatore | |
| random_seed | numero | |
| use_custom_model_name | indicatore | |
| custom_model_name | stringa | |
| use_custom_name | indicatore | |
| custom_name | stringa | |
| tooltip | stringa | |
| keywords | stringa | |
| annotation | stringa | |

Proprietà linearnode



I modelli di regressione lineare prevedono un target continuo basato sulle relazioni lineari tra l'obiettivo e uno o più predittori.

Tabella 123. Proprietà linearnode

| Proprietà linearnode | Valori | Descrizione proprietà |
|----------------------|---------------------|--|
| obiettivo | campo | Specifica un singolo campo obiettivo. |
| inputs | [campo1 ... campoN] | I campi predittore utilizzati dal modello. |
| weight_field | campo | Campo di analisi utilizzato dal modello. |

Tabella 123. Proprietà *linearasnode* (Continua)

| Proprietà <i>linearasnode</i> | Valori | Descrizione proprietà |
|--------------------------------------|---|--|
| <i>custom_fields</i> | <i>indicatore</i> | Il valore predefinito è TRUE. |
| <i>intercept</i> | <i>indicatore</i> | Il valore predefinito è TRUE. |
| <i>detect_2way_interaction</i> | <i>indicatore</i> | Indica se considerare o meno un'interazione a due vie. Il valore predefinito è TRUE. |
| <i>cin</i> | <i>number</i> | L'intervallo di confidenza utilizzato per calcolare stime dei coefficienti del modello. Specificare un valore maggiore di 0 e minore di 100. Il valore predefinito è 95. |
| <i>factor_order</i> | ascending descending | Il criterio di ordinamento per i predittori di categoria. Il valore predefinito è ascending. |
| <i>var_select_method</i> | ForwardStepwise BestSubsets none | Il metodo di selezione del modello da utilizzare. Il valore predefinito è ForwardStepwise. |
| <i>criteria_for_forward_stepwise</i> | AICC Fstatistics AdjustedRSquare ASE | La statistica utilizzata per determinare se un effetto deve essere aggiunto o eliminato dal modello. Il valore predefinito è AdjustedRSquare. |
| <i>pin</i> | <i>number</i> | L'effetto con il valore P più piccolo e minore di quello specificato nella soglia <i>pin</i> viene aggiunto al modello. Il valore predefinito è 0.05. |
| <i>pout</i> | <i>number</i> | Tutti gli effetti presenti nel modello che hanno un valore p superiore alla soglia <i>pout</i> specificata vengono eliminati. Il valore predefinito è 0.10. |
| <i>use_custom_max_effects</i> | <i>indicatore</i> | Indica se utilizzare il numero massimo di effetti nel modello finale. Il valore predefinito è FALSE. |
| <i>max_effects</i> | <i>number</i> | Numero massimo di effetti da utilizzare nel modello finale. Il valore predefinito è 1. |
| <i>use_custom_max_steps</i> | <i>indicatore</i> | Indica se utilizzare il numero massimo di fasi. Il valore predefinito è FALSE. |
| <i>max_steps</i> | <i>number</i> | Il numero massimo di fasi prima che l'algoritmo stepwise venga arrestato. Il valore predefinito è 1. |
| <i>criteria_for_best_subsets</i> | AICC AdjustedRSquare ASE | La modalità del criterio da utilizzare. Il valore predefinito è AdjustedRSquare. |

Proprietà *logregnode*



La regressione logistica, una tecnica statistica che consente di classificare i record in base ai valori dei campi di input, è analoga alla regressione lineare ma, al posto di un intervallo numerico, prende un campo obiettivo categoriale.

Esempio multinomiale

```

node = stream.create("logreg", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["BP", "Cholesterol", "Age"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Log_reg Drug")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Stepwise")
node.setPropertyValue("logistic_procedure", "Multinomial")
node.setPropertyValue("multinomial_base_category", "BP")
node.setPropertyValue("model_type", "FullFactorial")
node.setPropertyValue("custom_terms", [["BP", "Sex"], ["Age"], ["Na", "K"]])
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("scale", "Pearson")
node.setPropertyValue("scale_value", 3.0)
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("tolerance", "1.0E-7")
# "Convergence..." section
node.setPropertyValue("max_iterations", 50)
node.setPropertyValue("max_steps", 3)
node.setPropertyValue("l_converge", "1.0E-3")
node.setPropertyValue("p_converge", "1.0E-7")
node.setPropertyValue("delta", 0.03)
# "Output..." section
node.setPropertyValue("summary", True)
node.setPropertyValue("likelihood_ratio", True)
node.setPropertyValue("asymptotic_correlation", True)
node.setPropertyValue("goodness_fit", True)
node.setPropertyValue("iteration_history", True)
node.setPropertyValue("history_steps", 3)
node.setPropertyValue("parameters", True)
node.setPropertyValue("confidence_interval", 90)
node.setPropertyValue("asymptotic_covariance", True)
node.setPropertyValue("classification_table", True)
# "Stepping" options
node.setPropertyValue("min_terms", 7)
node.setPropertyValue("use_max_terms", True)
node.setPropertyValue("max_terms", 10)
node.setPropertyValue("probability_entry", 3)
node.setPropertyValue("probability_removal", 5)
node.setPropertyValue("requirements", "Containment")

```

Esempio binomiale

```

node = stream.create("logreg", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Cholesterol")
node.setPropertyValue("inputs", ["BP", "Drug", "Age"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Log_reg Cholesterol")
node.setPropertyValue("multinomial_base_category", "BP")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("binomial_method", "Forwards")

```

```

node.setPropertyValue("logistic_procedure", "Binomial")
node.setPropertyValue("binomial_categorical_input", "Sex")
node.setKeyedPropertyValue("binomial_input_contrast", "Sex", "Simple")
node.setKeyedPropertyValue("binomial_input_category", "Sex", "Last")
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("scale", "Pearson")
node.setPropertyValue("scale_value", 3.0)
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("tolerance", "1.0E-7")
# "Convergence..." section
node.setPropertyValue("max_iterations", 50)
node.setPropertyValue("l_converge", "1.0E-3")
node.setPropertyValue("p_converge", "1.0E-7")
# "Output..." section
node.setPropertyValue("binomial_output_display", "at_each_step")
node.setPropertyValue("binomial_goodness_of_fit", True)
node.setPropertyValue("binomial_iteration_history", True)
node.setPropertyValue("binomial_parameters", True)
node.setPropertyValue("binomial_ci_enable", True)
node.setPropertyValue("binomial_ci", 85)
# "Stepping" options
node.setPropertyValue("binomial_removal_criterion", "LR")
node.setPropertyValue("binomial_probability_removal", 0.2)

```

Tabella 124. proprietà logregnode.

| Proprietà logregnode | Valori | Descrizione proprietà |
|----------------------|---|---|
| obiettivo | <i>campo</i> | I modelli di regressione logistica richiedono un solo campo obiettivo e uno o più campi di input. I campi frequenza e peso non sono utilizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| logistic_procedure | Binomial Multinomial | |
| include_constant | <i>indicatore</i> | |
| mode | Simple Expert | |
| method | Enter Stepwise Forwards Backwards BackwardsStepwise | |
| binomial_method | Enter Forwards Backwards | |

Tabella 124. proprietà logregnode (Continua).

| Proprietà logregnode | Valori | Descrizione proprietà |
|------------------------------|---|--|
| model_type | MainEffects FullFactorial Custom | Se FullFactorial è specificato come tipo di modello, i criteri di controllo non verranno eseguiti, anche se sono specificati. Verrà invece utilizzato il metodo Enter. Se il tipo di modello è impostato su Custom, ma non sono stati specificati campi personalizzati, verrà creato un modello effetti principali. |
| custom_terms | [[BP Sex]][BP][Age]] | |
| multinomial_base_category | stringa | Specifica come viene determinata la categoria di riferimento. |
| binomial_categorical_input | stringa | |
| binomial_input_contrast | Indicator Simple Difference Helmert Repeated Polynomial Deviation | Proprietà basata su chiavi per input categoriali che indica come viene determinato il confronto. |
| binomial_input_category | First Last | Proprietà basata su chiavi per input categoriali che indica come viene determinata la categoria di riferimento. |
| scale | None UserDefined Pearson Deviance | |
| scale_value | numero | |
| all_probabilities | indicatore | |
| tolerance | 1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10 | |
| min_terms | numero | |
| use_max_terms | indicatore | |
| max_terms | numero | |
| entry_criterion | Punteggio LR | |
| removal_criterion | LR Wald | |
| probability_entry | numero | |
| probability_removal | numero | |
| binomial_probability_entry | numero | |
| binomial_probability_removal | numero | |

Tabella 124. proprietà logregnode (Continua).

| Proprietà logregnode | Valori | Descrizione proprietà |
|--------------------------------|--|-----------------------|
| requisiti | HierarchyDiscrete HierarchyAll Containment None | |
| max_iterations | numero | |
| max_steps | numero | |
| p_converge | 1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0 | |
| l_converge | 1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0 | |
| delta | numero | |
| iteration_history | indicatore | |
| history_steps | numero | |
| summary | indicatore | |
| likelihood_ratio | indicatore | |
| asymptotic_correlation | indicatore | |
| goodness_fit | indicatore | |
| parametri | indicatore | |
| confidence_interval | numero | |
| asymptotic_covariance | indicatore | |
| classification_table | indicatore | |
| stepwise_summary | indicatore | |
| info_criteria | indicatore | |
| monotonicity_measures | indicatore | |
| binomial_output_display | at_each_step at_last_step | |
| binomial_goodness_of_fit | indicatore | |
| binomial_parameters | indicatore | |
| binomial_iteration_history | indicatore | |
| binomial_classification_plots | indicatore | |
| binomial_ci_enable | indicatore | |
| binomial_ci | numero | |
| binomial_residual | valori anomali All | |
| binomial_residual_enable | indicatore | |
| binomial_outlier_threshold | numero | |
| binomial_classification_cutoff | numero | |

Tabella 124. proprietà logregnode (Continua).

| Proprietà logregnode | Valori | Descrizione proprietà |
|-------------------------------|---------------------------|-----------------------|
| binomial_removal_criterion | LR Wald Conditional | |
| calculate_variable_importance | <i>indicatore</i> | |
| calculate_raw_propensities | <i>indicatore</i> | |

Proprietà lsvmnode



Il nodo LSVM (Linear Support Vector Machine) consente di classificare i dati in uno di due gruppi senza sovradattamento. Il nodo LSVM è lineare e particolarmente indicato con dataset di grandi dimensioni, come quelli con un numero elevato di record. Il nodo LSVM viene eseguito quando connesso a IBM SPSS Analytic Server.

Tabella 125. Proprietà lsvmnode

| Proprietà lsvmnode | Valori | Descrizione proprietà |
|------------------------|--------------------------|---|
| intercept | <i>indicatore</i> | Include l'intercettazione nel modello. Il valore predefinito è True. |
| target_order | Crescente Decrescente | Specifica il criterio di ordinamento per il target di categoria. Ignorato per i target continui. L'impostazione di default è Ascending. |
| precision | <i>number</i> | Utilizzata solo se il livello di misurazione del campo obiettivo è Continuous. Specifica il parametro correlato alla sensibilità della perdita di per la regressione. Il valore minimo è 0 e non è previsto un valore massimo. Il valore predefinito è 0.1. |
| exclude_missing_values | <i>indicatore</i> | Quando è impostata su True, un record viene escluso se manca un singolo valore qualsiasi. Il valore predefinito è False. |
| penalty_function | L1 L2 | Specifica il tipo di funzione di penalità utilizzata. Il valore predefinito è L2. |
| lambda | <i>number</i> | Parametro di penalità (regolarizzazione). |

Tabella 125. Proprietà lsvmnode (Continua)

| Proprietà lsvmnode | Valori | Descrizione proprietà |
|-------------------------------|------------|--|
| calculate_variable_importance | indicatore | Per i modelli che producono un'appropriata misurazione dell'importanza, questa opzione visualizza un grafico che indica l'importanza relativa di ogni predittore nella stima del modello. Considerare che per calcolare l'importanza di una variabile alcuni modelli richiedono molto tempo, in particolare quando si lavorano dataset di grandi dimensioni e di conseguenza per alcuni modelli è disattivo per impostazione predefinita. L'importanza di una variabile non è disponibile per i modelli Elenco di decisioni. |

Proprietà neuralnetnode

Attenzione: in questa release è disponibile una nuova versione del modello Rete neurale con funzionalità avanzate, descritta nella sezione che segue (*neuralnetwork*). Sebbene sia ancora possibile creare e calcolare il punteggio di un modello con la versione precedente, si consiglia di aggiornare gli script in modo da utilizzare la nuova versione. I dettagli della versione precedente sono riportati a scopo informativo.

Esempio

```
node = stream.create("neuralnet", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("targets", ["Drug"])
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
# "Model" tab
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Dynamic")
node.setPropertyValue("train_pct", 30)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 12345)
node.setPropertyValue("stop_on", "Time")
node.setPropertyValue("accuracy", 95)
node.setPropertyValue("cycles", 200)
node.setPropertyValue("time", 3)
node.setPropertyValue("optimize", "Speed")
# "Multiple Method Expert Options" section
node.setPropertyValue("m_topologies", "5 30 5; 2 20 3, 1 10 1")
node.setPropertyValue("m_non_pyramids", False)
node.setPropertyValue("m_persistence", 100)
```

Tabella 126. proprietà neuralnetnode

| Proprietà neuralnetnode | Valori | Descrizione proprietà |
|-------------------------|---------------------|--|
| targets | [campo1 ... campoN] | Il nodo Rete neurale richiede uno o più campi obiettivo e uno o più campi di input. I campi frequenza e peso vengono ignorati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |

Tabella 126. proprietà neuralnetnode (Continua)

| Proprietà neuralnetnode | Valori | Descrizione proprietà |
|-------------------------|--|----------------------------------|
| method | Quick Dynamic Multiple Prune ExhaustivePrune RBFN | |
| prevent_overtrain | <i>indicatore</i> | |
| train_pct | <i>numero</i> | |
| set_random_seed | <i>indicatore</i> | |
| random_seed | <i>numero</i> | |
| mode | Simple Expert | |
| stop_on | Default Accuracy Cycles Time | Modalità di arresto. |
| accuracy | <i>numero</i> | Precisione di arresto. |
| cycles | <i>numero</i> | Cicli di apprendimento. |
| time | <i>numero</i> | Tempo di addestramento (minuti). |
| continue | <i>indicatore</i> | |
| show_feedback | <i>indicatore</i> | |
| binary_encode | <i>indicatore</i> | |
| use_last_model | <i>indicatore</i> | |
| gen_logfile | <i>indicatore</i> | |
| logfile_name | <i>stringa</i> | |
| alpha | <i>numero</i> | |
| initial_eta | <i>numero</i> | |
| high_eta | <i>numero</i> | |
| low_eta | <i>numero</i> | |
| eta_decay_cycles | <i>numero</i> | |
| hid_layers | One Two Three | |
| h1_units_one | <i>numero</i> | |
| h1_units_two | <i>numero</i> | |
| h1_units_three | <i>numero</i> | |
| persistence | <i>numero</i> | |
| m_topologies | <i>stringa</i> | |
| m_non_pyramids | <i>indicatore</i> | |
| m_persistence | <i>numero</i> | |
| p_hid_layers | One Two Three | |
| p_h1_units_one | <i>numero</i> | |

Tabella 126. proprietà neuralnetnode (Continua)

| Proprietà neuralnetnode | Valori | Descrizione proprietà |
|---------------------------------|--------------------|---|
| p_hl_units_two | numero | |
| p_hl_units_three | numero | |
| p_persistence | numero | |
| p_hid_rate | numero | |
| p_hid_pers | numero | |
| p_inp_rate | numero | |
| p_inp_pers | numero | |
| p_overall_pers | numero | |
| r_persistence | numero | |
| r_num_clusters | numero | |
| r_eta_auto | indicatore | |
| r_alpha | numero | |
| r_eta | numero | |
| optimize | Speed Memory | Utilizzare per specificare se ottimizzare la velocità o la memoria durante la creazione del modello. |
| calculate_variable_importance | indicatore | Nota: la proprietà sensitivity_analysis utilizzata nelle versioni precedenti è obsoleta ed è stata sostituita da questa proprietà. La vecchia proprietà è ancora supportata, ma si consiglia di utilizzare calculate_variable_importance. |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |
| adjusted_propensity_partition | Test Validation | |

Proprietà neuralnetworknode



Il nodo Rete neurale utilizza un modello semplificato del modo in cui il cervello umano elabora le informazioni. Funziona simulando un elevato numero di semplici unità di elaborazione interconnesse che assomigliano a versioni astratte di neuroni. Le reti neurali sono potenti strumenti di valutazione delle funzioni generali e richiedono una conoscenza statistica o matematica minima per l'addestramento o l'applicazione.

Esempio

```
node = stream.create("neuralnetwork", "My node")
# Build Options tab - Objectives panel
node.setPropertyValue("objective", "Standard")
# Build Options tab - Ensembles panel
node.setPropertyValue("combining_rule_categorical", "HighestMeanProbability")
```

Tabella 127. proprietà neuralnetworknode

| Proprietà neuralnetworknode | Valori | Descrizione proprietà |
|-----------------------------|---------------------|------------------------------|
| targets | [campo1 ... campoN] | Specifica i campi obiettivo. |

Tabella 127. proprietà neuralnetworknode (Continua)

| Proprietà neuralnetworknode | Valori | Descrizione proprietà |
|------------------------------|--|--|
| inputs | [campo1 ... campoN] | I campi predittore utilizzati dal modello. |
| splits | [campo1 ... campoN] | Specifica il campo o i campi da usare per la creazione di modelli suddivisi. |
| use_partition | indicatore | Se è definito un campo partizione, questa opzione garantisce che per la creazione del modello verranno utilizzati solo i dati della partizione di addestramento. |
| continue | indicatore | Addestramento continuo modello esistente. |
| objective | Standard Bagging Boosting psm | psm viene utilizzato per insiemi di dati di grandi dimensioni e richiede una connessione Server. |
| method | MultilayerPerceptron RadialBasisFunction | |
| use_custom_layers | indicatore | |
| first_layer_units | numero | |
| second_layer_units | numero | |
| use_max_time | indicatore | |
| max_time | numero | |
| use_max_cycles | indicatore | |
| max_cycles | numero | |
| use_min_accuracy | indicatore | |
| min_accuracy | numero | |
| combining_rule_categorical | Voting HighestProbability HighestMeanProbability | |
| combining_rule_continuous | Mean Median | |
| component_models_n | numero | |
| overfit_prevention_pct | numero | |
| use_random_seed | indicatore | |
| random_seed | numero | |
| missing_values | listwiseDeletion missingValueImputation | |
| use_model_name | booleano | |
| model_name | stringa | |
| confidence | onProbability onIncrease | |
| score_category_probabilities | indicatore | |
| max_categories | numero | |
| score_propensity | indicatore | |
| use_custom_name | indicatore | |

Tabella 127. proprietà neuralnetworknode (Continua)

| Proprietà neuralnetworknode | Valori | Descrizione proprietà |
|-----------------------------|---------|-----------------------|
| custom_name | stringa | |
| tooltip | stringa | |
| keywords | stringa | |
| annotation | stringa | |

Proprietà questnode



Il nodo QUEST offre un metodo di classificazione binario per la creazione di strutture ad albero delle decisioni, progettato per ridurre i tempi di elaborazione necessari per le analisi C&R Tree più complesse, riducendo inoltre la tendenza dei metodi per le strutture ad albero di classificazione a favorire gli input che consentono un numero maggiore di suddivisioni. I campi di input possono essere intervalli numerici (continui), ma il campo obiettivo deve essere categoriale. Tutte le suddivisioni sono binarie.

Esempio

```
node = stream.create("quest", "My node")
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("max_surrogates", 5)
node.setPropertyValue("split_alpha", 0.03)
node.setPropertyValue("use_percentage", False)
node.setPropertyValue("min_parent_records_abs", 40)
node.setPropertyValue("min_child_records_abs", 30)
node.setPropertyValue("prune_tree", True)
node.setPropertyValue("use_std_err", True)
node.setPropertyValue("std_err_multiplier", 3)
```

Tabella 128. proprietà questnode

| Proprietà questnode | Valori | Descrizione proprietà |
|----------------------------------|--|--|
| obiettivo | campo | I modelli QUEST richiedono un solo campo obiettivo e uno o più campi di input. È inoltre possibile specificare un campo frequenza. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| continue_training_existing_model | indicatore | |
| objective | Standard Boosting Bagging psm | psm viene utilizzato per insiemi di dati di grandi dimensioni e richiede una connessione Server. |
| model_output_type | Single InteractiveBuilder | |
| use_tree_directives | indicatore | |
| tree_directives | stringa | |
| use_max_depth | Default Custom | |

Tabella 128. proprietà questnode (Continua)

| Proprietà questnode | Valori | Descrizione proprietà |
|---------------------------------|--|--|
| max_depth | numero intero | Profondità massima della struttura ad albero, da 0 a 1000. Valore utilizzato solo se use_max_depth = Custom. |
| prune_tree | indicatore | Taglia struttura ad albero per evitare sovradattamento. |
| use_std_err | indicatore | Utilizza differenza massima di rischio (in errori standard). |
| std_err_multiplier | numero | Differenza massima. |
| max_surrogates | numero | Numero massimo surrogati. |
| use_percentage | indicatore | |
| min_parent_records_pc | numero | |
| min_child_records_pc | numero | |
| min_parent_records_abs | numero | |
| min_child_records_abs | numero | |
| use_costs | indicatore | |
| costs | strutturato | Proprietà strutturata. |
| priors | Data Equal Custom | |
| custom_priors | strutturato | Proprietà strutturata. |
| adjust_priors | indicatore | |
| trails | numero | Numero di modelli di componenti per boosting o bagging. |
| set_ensemble_method | Voting HighestProbability HighestMeanProbability | Regola di combinazione di default per obiettivi categoriali. |
| range_ensemble_method | Mean Median | Regola di combinazione di default per target continui. |
| large_boost | indicatore | Applica il boosting a insiemi di dati di grandi dimensioni. |
| split_alpha | numero | Livello di significatività per suddivisione. |
| train_pct | numero | Insieme di prevenzione del sovradattamento. |
| set_random_seed | indicatore | Opzione Replica risultati. |
| seed | numero | |
| calculate_variable_importance | indicatore | |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |
| adjusted_propensity_partition | Test Validation | |

Proprietà randomtrees



Il nodo Random Trees è disponibile solo se si dispone di una connessione a IBM SPSS Analytic Server. Tale nodo è simile al nodo C&RT esistente; tuttavia, il nodo Random Trees è progettato per elaborare una grande quantità di dati e visualizza il modello risultante nel visualizzatore di output aggiunto in SPSS Modeler versione 17. Il nodo struttura ad albero Random Trees genera una struttura ad albero delle decisioni che può essere utilizzata per prevedere o classificare osservazioni future. Il metodo utilizza l'esecuzione ricorsiva di partizioni per suddividere i record di addestramento in segmenti riducendo l'impurità ad ogni passaggio. Un nodo della struttura ad albero è considerato *puro* quando il 100% dei casi nel nodo fa parte di una categoria specifica del campo obiettivo. I campi obiettivo e di input possono essere intervalli numerici o categoriali (nominali, ordinali o flag); tutte le suddivisioni sono binarie (solo due sottogruppi).

Tabella 129. proprietà randomtrees

| Proprietà randomtrees | Valori | Descrizione proprietà |
|------------------------|-------------------|---|
| target | <i>campo</i> | Nel nodo Random Trees, i modelli richiedono un singolo obiettivo ed uno o più campi di input. È inoltre possibile specificare un campo frequenza. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| number_of_models | <i>integer</i> | Determina il numero di modelli da creare come parte della modellazione dell'insieme. |
| sample_size | <i>number</i> | Ridurre questo valore per migliorare le prestazioni durante l'elaborazione di dataset di grandi dimensioni. |
| handle_imbalanced_data | <i>indicatore</i> | Se l'obiettivo del modello è un particolare risultato flag ed il rapporto tra il risultato desiderato ed un risultato non desiderato è molto piccolo, i dati sono sbilanciati ed il campionamento bootstrap eseguito dal modello può avere effetti sulla precisione del modello. Abilitare la gestione dei dati sbilanciati in modo che il modello catturerà una proporzione maggiore del risultato desiderato e potrà generare un modello più forte. |
| use_weighted_sampling | <i>indicatore</i> | Quando è impostata su <i>False</i> , le variabili per ciascun nodo vengono selezionate casualmente con la stessa probabilità. Quando è impostata su <i>True</i> , le variabili vengono ponderate e selezionate di conseguenza. |
| use_stepwise | <i>indicatore</i> | Determina se viene utilizzata la creazione del modello stepwise. |
| max_node_number | <i>integer</i> | Il numero massimo di nodi consentiti nelle singole strutture ad albero. Se il numero viene superato alla suddivisione successiva, l'accrescimento della struttura ad albero viene arrestato. |
| max_depth | <i>integer</i> | Profondità massima della struttura ad albero prima dell'arresto dell'accrescimento. |

Tabella 129. proprietà randomtrees (Continua)

| Proprietà randomtrees | Valori | Descrizione proprietà |
|------------------------|------------------------------------|---|
| min_child_node_size | <i>integer</i> | Determina il numero minimo di record consentiti in un nodo figlio dopo la suddivisione del nodo padre. Se un nodo figlio contiene un numero di record inferiore a quello specificato in questo punto, il nodo padre non viene suddiviso |
| use_costs | <i>indicatore</i> | |
| costs | <i>strutturato</i> | Proprietà strutturata. Il formato è un elenco di 3 valori: il valore effettivo, il valore previsto ed il costo nel caso di previsione errata. Ad esempio: tree.setPropertyValue("costs", [{"drugA", "drugB", 3.0}, {"drugX", "drugY", 4.0}]) |
| default_cost_increase | none linear square custom | Nota: abilitata solo per obiettivi ordinali. Impostare i valori predefiniti nelle matrici costi. |
| max_pct_missing | <i>integer</i> | Se la percentuale di valori mancanti in un input è maggior del valore specificato in questo punto, l'input viene escluso. Minimo 0, massimo 100. |
| exclude_single_cat_pct | <i>integer</i> | Se un valore di categoria rappresenta una percentuale di record più alta rispetto a quanto specificato in questo punto, l'intero campo viene escluso dalla creazione del modello. Minimo 1, massimo 99. |
| max_category_number | <i>integer</i> | Se il numero di categorie in un campo supera questo valore, il campo viene escluso dalla creazione del modello. Minimo 2. |
| min_field_variation | <i>number</i> | Se il coefficiente di variazione di un campo continuo è più piccolo di questo valore, il campo viene escluso dalla creazione del modello. |
| num_bins | <i>integer</i> | Utilizzata solo se i dati sono costituiti da input continui. Impostare il numero di bin di frequenza da utilizzare per gli input; le opzioni sono: 2, 4, 5, 10, 20, 25, 50 o 100. |

Proprietà regressionnode



La regressione lineare è una tecnica statistica molto comune per riassumere i dati ed eseguire previsioni individuando un'area o una linea retta in grado di ridurre le discrepanze tra i valori di output previsti e quelli osservati.

Nota: il nodo Regressione verrà sostituito dal nodo Lineare nella prossima versione. Da questo momento si consiglia di utilizzare i modelli lineari per la regressione lineare.

Esempio


```

node = stream.create("regression", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Age")
node.setPropertyValue("inputs", ["Na", "K"])
node.setPropertyValue("partition", "Test")
node.setPropertyValue("use_weight", True)
node.setPropertyValue("weight_field", "Drug")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Regression Age")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Stepwise")
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("complete_records", False)
node.setPropertyValue("tolerance", "1.0E-3")
# "Stepping..." section
node.setPropertyValue("stepping_method", "Probability")
node.setPropertyValue("probability_entry", 0.77)
node.setPropertyValue("probability_removal", 0.88)
node.setPropertyValue("F_value_entry", 7.0)
node.setPropertyValue("F_value_removal", 8.0)
# "Output..." section
node.setPropertyValue("model_fit", True)
node.setPropertyValue("r_squared_change", True)
node.setPropertyValue("selection_criteria", True)
node.setPropertyValue("descriptives", True)
node.setPropertyValue("p_correlations", True)
node.setPropertyValue("collinearity_diagnostics", True)
node.setPropertyValue("confidence_interval", True)
node.setPropertyValue("covariance_matrix", True)
node.setPropertyValue("durbin_watson", True)

```

Tabella 130. proprietà regressionnode

| Proprietà regressionnode | Valori | Descrizione proprietà |
|--------------------------|--|--|
| obiettivo | <i>campo</i> | I modelli di regressione richiedono un solo campo obiettivo e uno o più campi di input. È anche possibile specificare un campo peso. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| method | Enter Stepwise Backwards Forwards | |
| include_constant | <i>indicatore</i> | |
| use_weight | <i>indicatore</i> | |
| weight_field | <i>campo</i> | |
| mode | Simple Expert | |
| complete_records | <i>indicatore</i> | |

Tabella 130. proprietà regressionnode (Continua)

| Proprietà regressionnode | Valori | Descrizione proprietà |
|-------------------------------|---|---|
| tolerance | 1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10 1.0E-11 1.0E-12 | Utilizzare le virgolette per gli argomenti. |
| stepping_method | useP useF | useP : usa probabilità di F useF: usa valore F |
| probability_entry | <i>numero</i> | |
| probability_removal | <i>numero</i> | |
| F_value_entry | <i>numero</i> | |
| F_value_removal | <i>numero</i> | |
| selection_criteria | <i>indicatore</i> | |
| confidence_interval | <i>indicatore</i> | |
| covariance_matrix | <i>indicatore</i> | |
| collinearity_diagnostics | <i>indicatore</i> | |
| regression_coefficients | <i>indicatore</i> | |
| exclude_fields | <i>indicatore</i> | |
| durbin_watson | <i>indicatore</i> | |
| model_fit | <i>indicatore</i> | |
| r_squared_change | <i>indicatore</i> | |
| p_correlations | <i>indicatore</i> | |
| descriptives | <i>indicatore</i> | |
| calculate_variable_importance | <i>indicatore</i> | |

Proprietà sequencenode



Il nodo Sequenza consente di scoprire le regole di associazione nei dati sequenziali o basati su valori temporali. Per sequenza si intende un elenco di serie di elementi che tendono a ricorrere secondo un ordine prevedibile. Ad esempio, un cliente che acquista un rasoio e la lozione dopobarba potrebbe in seguito acquistare la schiuma da barba. Il nodo Sequenza si basa sull'algoritmo delle regole di associazione CARMA, che utilizza un metodo efficiente in due passaggi per trovare le sequenze.

Esempio

```
node = stream.create("sequence", "My node")
# "Fields" tab
node.setPropertyValue("id_field", "Age")
node.setPropertyValue("contiguous", True)
node.setPropertyValue("use_time_field", True)
node.setPropertyValue("time_field", "Date1")
```

```

node.setPropertyValue("content_fields", ["Drug", "BP"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Sequence_test")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("min_supp", 15.0)
node.setPropertyValue("min_conf", 14.0)
node.setPropertyValue("max_size", 7)
node.setPropertyValue("max_predictions", 5)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("use_max_duration", True)
node.setPropertyValue("max_duration", 3.0)
node.setPropertyValue("use_pruning", True)
node.setPropertyValue("pruning_value", 4.0)
node.setPropertyValue("set_mem_sequences", True)
node.setPropertyValue("mem_sequences", 5.0)
node.setPropertyValue("use_gaps", True)
node.setPropertyValue("min_item_gap", 20.0)
node.setPropertyValue("max_item_gap", 30.0)

```

Tabella 131. proprietà sequencenode

| Proprietà sequencenode | Valori | Descrizione proprietà |
|------------------------|---------------------|---|
| id_field | campo | Per creare un modello Sequenza, è necessario specificare un campo ID, un campo ora facoltativo e uno o più campi contenuto. I campi peso e frequenza non sono utilizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| time_field | campo | |
| use_time_field | indicatore | |
| content_fields | [campo1 ... campon] | |
| contiguous | indicatore | |
| min_supp | numero | |
| min_conf | numero | |
| max_size | numero | |
| max_predictions | numero | |
| mode | Simple Expert | |
| use_max_duration | indicatore | |
| max_duration | numero | |
| use_gaps | indicatore | |
| min_item_gap | numero | |
| max_item_gap | numero | |
| use_pruning | indicatore | |
| pruning_value | numero | |
| set_mem_sequences | indicatore | |
| mem_sequences | numero intero | |

Proprietà slrmnode



Il nodo Modello risposta autoapprendimento consente di creare un modello in cui è possibile utilizzare un unico nuovo caso oppure un numero limitato di nuovi casi per eseguire una nuova stima del modello senza doverlo riaddestrare con tutti i dati.

Esempio

```
node = stream.create("slrm", "My node")
node.setPropertyValue("target", "Offer")
node.setPropertyValue("target_response", "Response")
node.setPropertyValue("inputs", ["Cust_ID", "Age", "Ave_Bal"])
```

Tabella 132. proprietà slrmnode

| Proprietà slrmnode | Valori | Descrizione proprietà |
|----------------------------------|----------------------------|---|
| obiettivo | <i>campo</i> | Il campo obiettivo deve essere nominale o flag. È inoltre possibile specificare un campo frequenza. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| target_response | <i>campo</i> | Il tipo deve essere Flag. |
| continue_training_existing_model | <i>indicatore</i> | |
| target_field_values | <i>indicatore</i> | Utilizza tutto: utilizza tutti i valori dalla sorgente. Specifica: è necessario selezionare valori. |
| target_field_values_specify | <i>[campo1 ... campoN]</i> | |
| include_model_assessment | <i>indicatore</i> | |
| model_assessment_random_seed | <i>numero</i> | Deve essere un numero reale. |
| model_assessment_sample_size | <i>numero</i> | Deve essere un numero reale. |
| model_assessment_iterations | <i>numero</i> | Numero di iterazioni. |
| display_model_evaluation | <i>indicatore</i> | |
| max_predictions | <i>numero</i> | |
| randomization | <i>numero</i> | |
| scoring_random_seed | <i>numero</i> | |
| sort | Crescente Decrescente | Specifica se verranno visualizzate per prime le offerte con i punteggi più alti o più bassi. |
| model_reliability | <i>indicatore</i> | |
| calculate_variable_importance | <i>indicatore</i> | |

Proprietà statisticsmodelnode



Il nodo Modello Statistics consente di analizzare e operare con i dati eseguendo le procedure IBM SPSS Statistics che generano PMML. Questo nodo richiede una copia di IBM SPSS Statistics con regolare licenza.

Le proprietà di questo nodo sono descritte in “Proprietà statisticsmodelnode” a pagina 312.

Proprietà stpnode



Il nodo di previsione spazio temporale (STP, Spatio-Temporal Prediction) utilizza dati contenenti informazioni sull'ubicazione, campi di input per la previsione (predittori), e un campo di destinazione. Ciascuna ubicazione ha numerose righe nei dati che rappresentano i valori di ciascun predittore in ogni momento di misurazione. Una volta analizzati i dati, essi possono essere utilizzati per prevedere i valori di destinazione in qualsiasi ubicazione all'interno dei dati di forma utilizzati nell'analisi.

Tabella 133. Proprietà stpnode

| Proprietà stpnode | Data type | Descrizione proprietà |
|-----------------------------------|--|--|
| Scheda Campi | | |
| target | <i>campo</i> | Questo è il campo obiettivo. |
| location | <i>campo</i> | Il campo ubicazione del modello. Sono consentiti solo campi geospaziali. |
| location_label | <i>campo</i> | Il campo categoriale da utilizzare nell'output per apporre un'etichetta alle ubicazioni scelte in location |
| time_field | <i>campo</i> | Il campo ora per il modello. Sono consentiti solo campi con una misurazione continua, mentre il tipo di archiviazione deve essere time, date, timestamp o integer. |
| inputs | <i>[campo1 ... campoN]</i> | Un elenco di campi di input. |
| Scheda Intervalli di tempo | | |
| interval_type_timestamp | Years Trimestri Mesi Weeks Days Hours Minutes Seconds | |
| interval_type_date | Years Trimestri Mesi Weeks Days | |
| interval_type_time | Hours Minutes Seconds | Riduce il numero di giorni a settimana presi in considerazione durante la creazione dell'indice ora utilizzato da STP per il calcolo |
| interval_type_integer | Periods (solo campi indice Time, archiviazione Integer) | L'intervallo in cui il dataset verrà convertito. La selezione disponibile è dipendente dal tipo di archiviazione del campo scelto come time_field per il modello. |
| period_start | <i>integer</i> | |

Tabella 133. Proprietà *stpnode* (Continua)

| Proprietà <i>stpnode</i> | Data type | Descrizione proprietà |
|---------------------------|--|---|
| <i>start_month</i> | Gennaio Febbraio Marzo Aprile Maggio Giugno Luglio Agosto Settembre Ottobre Novembre Dicembre | Il mese da cui il modello inizierà ad eseguire l'indicizzazione (ad esempio, se impostato su Marzo ma il primo record del dataset è Gennaio, il modello ignora i primi due record ed inizia l'indicizzazione da Marzo). |
| <i>week_begins_on</i> | Sunday Monday Tuesday Wednesday Thursday Friday Saturday | Il punto iniziale dell'indice ora creato da STP dai dati |
| <i>days_per_week</i> | <i>integer</i> | Minimo 1, massimo 7, con incrementi di 1 |
| <i>hours_per_day</i> | <i>integer</i> | Il numero di ore di cui il modello tiene conto in un giorno. Se questo valore è impostato su 10, il modello inizierà l'indicizzazione all'ora indicata da <i>day_begins_at</i> e continuerà l'indicizzazione per 10 ore, quindi passerà al valore successivo che corrisponde al valore <i>day_begins_at</i> e così via. |
| <i>day_begins_at</i> | 00:00 01:00 02:00 03:00 ... 23:00 | Imposta il valore dell'ora a partire dal quale il modello inizia l'indicizzazione. |
| <i>interval_increment</i> | 1 2 3 4 5 6 10 12 15 20 30 | Questa impostazione di incremento è relativa a minuti o secondi. Determina il punto in cui il modello crea gli indici dai dati. Pertanto, con un incremento uguale a 30 e tipo di intervallo secondi, il modello creerà un indice dai dati ogni 30 secondi. |

Tabella 133. Proprietà *stpnode* (Continua)

| Proprietà <i>stpnode</i> | Data type | Descrizione proprietà |
|-------------------------------------|---|---|
| <code>data_matches_interval</code> | <i>Booleana</i> | <p>Se impostato su N, la conversione dei dati nel tipo <code>interval_type</code> regolare si verifica prima che il modello venga creato.</p> <p>Se i propri dati sono già nel formato corretto, e <code>interval_type</code> e tutte le impostazioni associate corrispondono ai propri dati, impostare questo valore su Y per impedire la conversione o l'aggregazione dei propri dati.</p> <p>L'impostazione di questo valore su Y disabilita tutti i controlli Aggregazione.</p> |
| <code>agg_range_default</code> | Sum Mean Min Max Median 1stQuartile 3rdQuartile | <p>Determina il metodo di aggregazione predefinito utilizzato per i campi continui. Tutti i campi continui non inclusi in modo specifico nell'aggregazione personalizzata verranno aggregati utilizzando il metodo specificato in questo punto.</p> |
| <code>custom_agg</code> | <pre>[[field, aggregation method],[..]]</pre> <p>Demo: <pre>[['x5' 'FirstQuartile']['x4' 'Sum']]</pre></p> | <p>Proprietà strutturata:</p> <p>Parametro script: <code>custom_agg</code></p> <p>Ad esempio: <pre>set :stpnode.custom_agg = [[field1 function] [field2 function]]</pre></p> <p>Dove <code>function</code> è la funzione di aggregazione da utilizzare con tale campo.</p> |
| Scheda Di base | | |
| <code>include_intercept</code> | <i>indicatore</i> | |
| <code>max_autoregressive_lag</code> | <i>integer</i> | <p>Valore minimo 1, valore massimo 5, con incrementi di 1. Questo è il numero di record precedenti richiesti per una previsione. Quindi, ad esempio, se impostato su 5, per creare una nuova previsione vengono utilizzati i 5 record precedenti. Il numero di record specificati in questo punto dai dati di creazione viene incorporato nel modello e, pertanto, l'utente non deve fornire nuovamente i dati durante il calcolo del punteggio del modello.</p> |
| <code>estimation_method</code> | Parametric Nonparametric | <p>Il metodo per la modellazione della matrice di conversione spaziale</p> |

Tabella 133. Proprietà *stpnode* (Continua)

| Proprietà <i>stpnode</i> | Data type | Descrizione proprietà |
|-------------------------------|---|---|
| parametric_model | Gaussian Exponential PoweredExponential | Parametro ordine per il modello di covarianza spaziale Parametric |
| exponential_power | number | Livello di potenza per il modello PoweredExponential. Valore minimo 1, valore massimo 2. |
| Scheda Avanzate | | |
| max_missing_values | integer | La percentuale massima di record con valori mancanti consentita nel modello. |
| significance | number | Il livello di significatività per il test delle ipotesi nella creazione del modello. Specifica il valore di significatività per tutti i test nella valutazione del modello STP, inclusi due test Bontà di adattamento, test Effect F e Coefficient t. |
| Scheda Output | | |
| model_specifications | indicatore | |
| temporal_summary | indicatore | |
| location_summary | indicatore | Determina se la tabella Riepilogo ubicazione è inclusa nell'output del modello. |
| model_quality | indicatore | |
| test_mean_structure | indicatore | |
| mean_structure_coefficients | indicatore | |
| autoregressive_coefficients | indicatore | |
| test_decay_space | indicatore | |
| parametric_spatial_covariance | indicatore | |
| correlations_heat_map | indicatore | |
| correlations_map | indicatore | |
| location_clusters | indicatore | |
| similarity_threshold | number | La soglia alla quale i cluster di output vengono considerati abbastanza simili per essere uniti in un singolo cluster. |
| max_number_clusters | integer | Il limite superiore per il numero di cluster che è possibile includere nell'output del modello. |
| Scheda Opzioni modello | | |
| use_model_name | indicatore | |
| model_name | stringa | |
| uncertainty_factor | number | Valore minimo 0, valore massimo 100. Determina l'incremento dell'incertezza (errore) applicata alle previsioni nel futuro. Rappresenta il limite superiore ed inferiore per le previsioni. |

Proprietà svmnode



Il nodo SVM (Support Vector Machine) consente di classificare i dati in uno di due gruppi senza sovradattamento. Il nodo SVM è particolarmente indicato per l'utilizzo con insiemi di dati di grandi dimensioni, cioè quelli con un elevato numero di campi di input.

Esempio

```
node = stream.create("svm", "My node")
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("kernel", "Polynomial")
node.setPropertyValue("gamma", 1.5)
```

Tabella 134. proprietà svmnode.

| Proprietà svmnode | Valori | Descrizione proprietà |
|---------------------------------|--|--|
| all_probabilities | <i>indicatore</i> | |
| stopping_criteria | 1.0E-1 1.0E-2 1.0E-3 (default) 1.0E-4 1.0E-5 1.0E-6 | Determina quando interrompere l'algoritmo di ottimizzazione. |
| regularization | <i>numero</i> | Nota anche come parametro C. |
| precision | <i>numero</i> | Utilizzata solo se il livello di misurazione del campo obiettivo è Continuous. |
| kernel | RBF(default) Polynomial Sigmoid Linear | Tipo di funzione Kernel utilizzata per la trasformazione. |
| rbf_gamma | <i>numero</i> | Utilizzata solo se kernel è RBF. |
| gamma | <i>numero</i> | Utilizzata solo se kernel è Polynomial o Sigmoid. |
| bias | <i>numero</i> | |
| degree | <i>numero</i> | Utilizzata solo se kernel è Polynomial. |
| calculate_variable_importance | <i>indicatore</i> | |
| calculate_raw_propensities | <i>indicatore</i> | |
| calculate_adjusted_propensities | <i>indicatore</i> | |
| adjusted_propensity_partition | Test Validation | |

Proprietà tcmnode



La modellazione causale temporale tenta di rilevare le relazioni causali principali nei dati di serie temporali. Nella modellazione causale temporale, vengono specificati un insieme di serie di destinazione e un insieme di immissioni candidati per tali destinazioni. La procedura quindi crea un modello di serie temporale autoregressivo per ciascuna destinazione ed include solo gli input che hanno una relazione causale più significativa con la destinazione.

Tabella 135. Proprietà tcmnode

| Proprietà tcmnode | Valori | Descrizione proprietà |
|-----------------------|--|-----------------------|
| custom_fields | Booleana | |
| dimensionlist | [dimension1 ... dimensionN] | |
| data_struct | Multiple Single | |
| metric_fields | campi | |
| both_target_and_input | [f1 ... fN] | |
| targets | [f1 ... fN] | |
| candidate_inputs | [f1 ... fN] | |
| forced_inputs | [f1 ... fN] | |
| use_timestamp | Timestamp Period | |
| input_interval | None Unknown Year Quarter Month Week Day Hour Hour_nonperiod Minute Minute_nonperiod Second Second_nonperiod | |
| period_field | stringa | |
| period_start_value | integer | |
| num_days_per_week | integer | |
| start_day_of_week | Sunday Monday Tuesday Wednesday Thursday Friday Saturday | |
| num_hours_per_day | integer | |
| start_hour_of_day | integer | |
| timestamp_increments | integer | |
| cyclic_increments | integer | |
| cyclic_periods | elenco | |

Tabella 135. Proprietà *tcmnode* (Continua)

| Proprietà <i>tcmnode</i> | Valori | Descrizione proprietà |
|----------------------------|--|-----------------------|
| output_interval | None Year Quarter Month Week Day Hour Minute Second | |
| is_same_interval | Stesso Notsame | |
| cross_hour | <i>Booleana</i> | |
| aggregate_and_distribute | <i>elenco</i> | |
| aggregate_default | Mean Sum Mode Min Max | |
| distribute_default | Mean Sum | |
| group_default | Mean Sum Mode Min Max | |
| missing_imput | Linear_interp Series_mean K_mean K_meridian Linear_trend None | |
| k_mean_param | <i>integer</i> | |
| k_median_param | <i>integer</i> | |
| missing_value_threshold | <i>integer</i> | |
| conf_level | <i>integer</i> | |
| max_num_predictor | <i>integer</i> | |
| max_lag | <i>integer</i> | |
| epsilon | <i>number</i> | |
| threshold | <i>integer</i> | |
| is_re_est | <i>Booleana</i> | |
| num_targets | <i>integer</i> | |
| percent_targets | <i>integer</i> | |
| fields_display | <i>elenco</i> | |
| series_display | <i>elenco</i> | |
| network_graph_for_target | <i>Booleana</i> | |
| sign_level_for_target | <i>number</i> | |
| fit_and_outlier_for_target | <i>Booleana</i> | |

Tabella 135. Proprietà tcmnode (Continua)

| Proprietà tcmnode | Valori | Descrizione proprietà |
|---------------------------------|------------------------------|-----------------------|
| sum_and_para_for_target | Booleana | |
| impact_diag_for_target | Booleana | |
| impact_diag_type_for_target | Effect Cause Both | |
| impact_diag_level_for_target | integer | |
| series_plot_for_target | Booleana | |
| res_plot_for_target | Booleana | |
| top_input_for_target | Booleana | |
| forecast_table_for_target | Booleana | |
| same_as_for_target | Booleana | |
| network_graph_for_series | Booleana | |
| sign_level_for_series | number | |
| fit_and_outlier_for_series | Booleana | |
| sum_and_para_for_series | Booleana | |
| impact_diagram_for_series | Booleana | |
| impact_diagram_type_for_series | Effect Cause Both | |
| impact_diagram_level_for_series | integer | |
| series_plot_for_series | Booleana | |
| residual_plot_for_series | Booleana | |
| forecast_table_for_series | Booleana | |
| outlier_root_cause_analysis | Booleana | |
| causal_levels | integer | |
| outlier_table | Interactive Pivot Both | |
| rmsp_error | Booleana | |
| bic | Booleana | |
| r_square | Booleana | |
| outliers_over_time | Booleana | |
| series_transormation | Booleana | |
| use_estimation_period | Booleana | |
| estimation_period | Times Observation | |
| observations | elenco | |
| observations_type | Latest Earliest | |
| observations_num | integer | |
| observations_exclude | integer | |
| extend_records_into_future | Booleana | |

Tabella 135. Proprietà *tcmnode* (Continua)

| Proprietà <i>tcmnode</i> | Valori | Descrizione proprietà |
|--------------------------|----------------------------|-----------------------|
| forecastperiods | <i>integer</i> | |
| max_num_distinct_values | <i>integer</i> | |
| display_targets | FIXEDNUMBER PERCENTAGE | |
| goodness_fit_measure | ROOTMEAN BIC RSQUARE | |
| top_input_for_series | <i>Booleana</i> | |
| aic | <i>Booleana</i> | |
| rmse | <i>Booleana</i> | |

Proprietà *timeseriesnode*



Il nodo Serie temporali stima i modelli di livellamento esponenziale, i modelli ARIMA (Autoregressive Integrated Moving Average, autoregressivi integrati a media mobile) univariati e ARIMA (o a funzione di trasferimento) multivariati per i dati di serie temporali e genera previsioni di prestazioni future. Il nodo Serie temporali deve sempre essere preceduto da un nodo Intervalli di tempo.

Esempio

```
node = stream.create("timeseries", "My node")
node.setPropertyValue("method", "Exsmooth")
node.setPropertyValue("exsmooth_model_type", "HoltsLinearTrend")
node.setPropertyValue("exsmooth_transformation_type", "None")
```

Tabella 136. proprietà *timeseriesnode*

| Proprietà <i>timeseriesnode</i> | Valori | Descrizione proprietà |
|---------------------------------|---|---|
| targets | <i>campo</i> | Il nodo Serie temporali prevede uno o più obiettivi, utilizzando in via facoltativa uno o più campi di input come predittori. I campi frequenza e peso non sono utilizzati. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| continue | <i>indicatore</i> | |
| method | ExpertModeler Exsmooth Arima Reuse | |
| expert_modeler_method | <i>indicatore</i> | |
| consider_seasonal | <i>indicatore</i> | |
| detect_outliers | <i>indicatore</i> | |
| expert_outlier_additive | <i>indicatore</i> | |
| expert_outlier_level_shift | <i>indicatore</i> | |

Tabella 136. proprietà timeseriesnode (Continua)

| Proprietà timeseriesnode | Valori | Descrizione proprietà |
|--|---|-----------------------------------|
| expert_outlier_innovational | <i>indicatore</i> | |
| expert_outlier_level_shift | <i>indicatore</i> | |
| expert_outlier_transient | <i>indicatore</i> | |
| expert_outlier_seasonal_additive | <i>indicatore</i> | |
| expert_outlier_local_trend | <i>indicatore</i> | |
| expert_outlier_additive_patch | <i>indicatore</i> | |
| exsmooth_model_type | Simple HoltLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative | |
| exsmooth_transformation_type | None SquareRoot NaturalLog | |
| arima_p | <i>numero intero</i> | |
| arima_d | <i>numero intero</i> | |
| arima_q | <i>numero intero</i> | |
| arima_sp | <i>numero intero</i> | |
| arima_sd | <i>numero intero</i> | |
| arima_sq | <i>numero intero</i> | |
| arima_transformation_type | None SquareRoot NaturalLog | |
| arima_include_constant | <i>indicatore</i> | |
| tf_arima_p. <i>nomecampo</i> | <i>numero intero</i> | Per le funzioni di trasferimento. |
| tf_arima_d. <i>nomecampo</i> | <i>numero intero</i> | Per le funzioni di trasferimento. |
| tf_arima_q. <i>nomecampo</i> | <i>numero intero</i> | Per le funzioni di trasferimento. |
| tf_arima_sp. <i>nomecampo</i> | <i>numero intero</i> | Per le funzioni di trasferimento. |
| tf_arima_sd. <i>nomecampo</i> | <i>numero intero</i> | Per le funzioni di trasferimento. |
| tf_arima_sq. <i>nomecampo</i> | <i>numero intero</i> | Per le funzioni di trasferimento. |
| tf_arima_delay. <i>nomecampo</i> | <i>numero intero</i> | Per le funzioni di trasferimento. |
| tf_arima_transformation_type. <i>nomecampo</i> | None SquareRoot NaturalLog | Per le funzioni di trasferimento. |
| arima_detect_outlier_mode | None Automatic | |
| arima_outlier_additive | <i>indicatore</i> | |

Tabella 136. proprietà timeseriesnode (Continua)

| Proprietà timeseriesnode | Valori | Descrizione proprietà |
|---------------------------------|----------------------|---|
| arima_outlier_level_shift | <i>indicatore</i> | |
| arima_outlier_innovational | <i>indicatore</i> | |
| arima_outlier_transient | <i>indicatore</i> | |
| arima_outlier_seasonal_additive | <i>indicatore</i> | |
| arima_outlier_local_trend | <i>indicatore</i> | |
| arima_outlier_additive_patch | <i>indicatore</i> | |
| conf_limit_pct | <i>reale</i> | |
| max_lags | <i>numero intero</i> | |
| events | <i>campi</i> | |
| scoring_model_only | <i>indicatore</i> | Utilizzato per i modelli con grandi quantità (decine di migliaia) di serie temporali. |

Proprietà treeas



Il nodo Tree-AS è disponibile solo se si dispone di una connessione a IBM SPSS Analytic Server. Questo nodo è simile a quello CHAID; tuttavia il nodo Tree-AS è progettato per elaborare una grande mole di dati per creare una singola struttura ad albero e visualizza il modello risultante in un visualizzatore di output aggiunto in SPSS Modeler versione 17. Il nodo genera una struttura ad albero delle decisioni utilizzando statistiche chi-quadrato (CHAID) per identificare le suddivisioni ottimali. Tale utilizzo di CHAID può generare strutture ad albero non binarie; ciò significa che alcune suddivisioni dispongono di più di due rami. I campi obiettivo e di input possono essere intervallo numerico (continui) o categoriali. Un CHAID completo è una modificazione di CHAID che esegue operazioni avanzate per l'analisi di tutte le suddivisioni possibili, ma richiede tempi di elaborazione maggiori.

Tabella 137. Proprietà treeas

| Proprietà treeas | Valori | Descrizione proprietà |
|------------------|---------------------------|--|
| target | <i>campo</i> | Nel nodo Tree-AS, i modelli CHAID richiedono un solo campo obiettivo e uno o più campi di input. È inoltre possibile specificare un campo frequenza. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| method | chaid exhaustive_chaid | |
| max_depth | <i>integer</i> | Profondità massima della struttura ad albero, da 0 a 20. Il valore predefinito è 5. |
| num_bins | <i>integer</i> | Utilizzata solo se i dati sono costituiti da input continui. Impostare il numero di bin di frequenza da utilizzare per gli input; le opzioni sono: 2, 4, 5, 10, 20, 25, 50 o 100. |

Tabella 137. Proprietà treeas (Continua)

| Proprietà treeas | Valori | Descrizione proprietà |
|----------------------------|--------------------------------|--|
| record_threshold | <i>integer</i> | Il numero di record a cui il modello passerà dall'utilizzo dei valori P all'utilizzo delle dimensioni degli effetti durante la creazione della struttura ad albero. Il valore predefinito è 1.000.000; aumentare o diminuire questo valore con incrementi di 10.000. |
| split_alpha | <i>number</i> | Livello di significatività per suddivisione. Il valore deve essere compreso tra 0,01 e 0,99. |
| merge_alpha | <i>number</i> | Livello di significatività per unione. Il valore deve essere compreso tra 0,01 e 0,99. |
| bonferroni_adjustment | <i>indicatore</i> | Adegua valori di significatività tramite il metodo di Bonferroni. |
| effect_size_threshold_cont | <i>number</i> | Impostare la soglia della dimensione degli effetti durante la suddivisione dei nodi e l'unione delle categorie quando si utilizza un obiettivo continuo. Il valore deve essere compreso tra 0,01 e 0,99. |
| effect_size_threshold_cat | <i>number</i> | Impostare la soglia della dimensione degli effetti durante la suddivisione dei nodi e l'unione delle categorie quando si utilizza un obiettivo categorico. Il valore deve essere compreso tra 0,01 e 0,99. |
| split_merged_categories | <i>indicatore</i> | Consenti risuddivisione di categorie unite. |
| grouping_sig_level | <i>number</i> | Utilizzata per determinare il modo in cui i gruppi di nodi sono costituiti o come vengono identificati i nodi inusuali. |
| chi_square | pearson likelihood_ratio | Metodo utilizzato per calcolare la statistica chi-quadrato: Pearson o Rapporto di verosimiglianza |
| minimum_record_use | use_percentage use_absolute | |
| min_parent_records_pc | <i>number</i> | Il valore predefinito è 2. Valore minimo 1, valore massimo 100, con incrementi di 1. Il valore del ramo principale deve essere superiore al valore del ramo secondario. |
| min_child_records_pc | <i>number</i> | Il valore predefinito è 1. Valore minimo 1, valore massimo 100, con incrementi di 1. |
| min_parent_records_abs | <i>number</i> | Il valore predefinito è 100. Valore minimo 1, valore massimo 100, con incrementi di 1. Il valore del ramo principale deve essere superiore al valore del ramo secondario. |
| min_child_records_abs | <i>number</i> | Il valore predefinito è 50. Valore minimo 1, massimo 100, con incrementi di 1. |
| epsilon | <i>number</i> | Modifica minima nelle frequenze di cella previste. |
| max_iterations | <i>number</i> | Numero massimo di iterazioni per la convergenza. |
| use_costs | <i>indicatore</i> | |

Tabella 137. Proprietà treeas (Continua)

| Proprietà treeas | Valori | Descrizione proprietà |
|-----------------------|------------------------------------|---|
| costs | strutturato | Proprietà strutturata. Il formato è un elenco di 3 valori: il valore effettivo, il valore previsto ed il costo nel caso di previsione errata. Ad esempio: tree.setPropertyValue("costs", [{"drugA", "drugB", 3.0}, {"drugX", "drugY", 4.0}]) |
| default_cost_increase | none linear square custom | Nota: abilitata solo per obiettivi ordinali. Impostare i valori predefiniti nelle matrice costi. |
| calculate_conf | indicatore | |
| display_rule_id | indicatore | Aggiunge un campo all'output del calcolo del punteggio che indica l'ID del nodo terminale al quale è assegnato ogni record. |

Proprietà twostepnode



Il nodo TwoStep è un metodo di raggruppamento tramite cluster in due fasi. La prima fase esegue un singolo passaggio nei dati per comprimere i dati di input non elaborati in un insieme gestibile di cluster secondari. Nella seconda fase viene utilizzato un metodo di raggruppamento tramite cluster gerarchico per unire progressivamente i cluster secondari in cluster sempre più grandi. Il nodo TwoStep offre il vantaggio di stimare automaticamente il numero ottimale di cluster per i dati di addestramento. Può gestire in modo efficiente tipi di campo misti e insiemi di dati di grandi dimensioni.

Esempio

```
node = stream.create("twostep", "My node")
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["Age", "K", "Na", "BP"])
node.setPropertyValue("partition", "Test")
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "TwoStep_Drug")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("exclude_outliers", True)
node.setPropertyValue("cluster_label", "String")
node.setPropertyValue("label_prefix", "TwoStep_")
node.setPropertyValue("cluster_num_auto", False)
node.setPropertyValue("max_num_clusters", 9)
node.setPropertyValue("min_num_clusters", 3)
node.setPropertyValue("num_clusters", 7)
```

Tabella 138. proprietà twostepnode

| Proprietà twostepnode | Valori | Descrizione proprietà |
|-----------------------|---------------------|---|
| inputs | [campo1 ... campoN] | I modelli TwoStep utilizzano un elenco di campi di input, ma nessun campo obiettivo. I campi peso e frequenza non vengono riconosciuti. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni nodi modellazione" a pagina 161. |
| standardize | indicatore | |
| exclude_outliers | indicatore | |

Tabella 138. proprietà twostepnode (Continua)

| Proprietà twostepnode | Valori | Descrizione proprietà |
|-----------------------|----------------------------|-----------------------|
| percentage | numero | |
| cluster_num_auto | indicatore | |
| min_num_clusters | numero | |
| max_num_clusters | numero | |
| num_clusters | numero | |
| cluster_label | String Number | |
| label_prefix | stringa | |
| distance_measure | Euclidean Loglikelihood | |
| clustering_criterion | AIC BIC | |

Proprietà twostepAS



TwoStep Cluster è uno strumento esplorativo progettato per rivelare i raggruppamenti naturali (o cluster) all'interno di un dataset, che altrimenti non sarebbero evidenti. L'algoritmo impiegato da questa procedura ha diverse funzioni interessanti che lo differenziano dalle tecniche tradizionali di clustering, quali la gestione delle variabili continue e categoriali, la selezione automatica del numero di cluster e la scalabilità.

Tabella 139. Proprietà twostepAS

| Proprietà twostepAS | Valori | Descrizione proprietà |
|------------------------------|--|---|
| inputs | [f1 ... fN] | I modelli TwoStepAS utilizzano un elenco di campi di input, ma nessun campo obiettivo. I campi peso e frequenza non vengono riconosciuti. |
| use_predefined_roles | Booleana | Valore predefinito=True |
| use_custom_field_assignments | Booleana | Valore predefinito=False |
| cluster_num_auto | Booleana | Valore predefinito=True |
| min_num_clusters | integer | Valore predefinito=2 |
| max_num_clusters | integer | Valore predefinito=15 |
| num_clusters | integer | Valore predefinito=5 |
| clustering_criterion | AIC BIC | |
| automatic_clustering_method | use_clustering_criterion_setting Distance_jump Minimo Massimo | |
| feature_importance_method | use_clustering_criterion_setting effect_size | |
| use_random_seed | Booleana | |

Tabella 139. Proprietà twostepAS (Continua)

| Proprietà twostepAS | Valori | Descrizione proprietà |
|--|----------------------------|-------------------------|
| random_seed | integer | |
| distance_measure | Euclidean Loglikelihood | |
| include_outlier_clusters | Booleana | Valore predefinito=True |
| num_cases_in_feature_tree_leaf_is_less_than | integer | Valore predefinito=10 |
| top_perc_outliers | integer | Valore predefinito=5 |
| initial_dist_change_threshold | integer | Valore predefinito=0 |
| leaf_node_maximum_branches | integer | Valore predefinito=8 |
| non_leaf_node_maximum_branches | integer | Valore predefinito=8 |
| max_tree_depth | integer | Valore predefinito=3 |
| adjustment_weight_on_measurement_level | integer | Valore predefinito=6 |
| memory_allocation_mb | number | Valore predefinito=512 |
| delayed_split | Booleana | Valore predefinito=True |
| fields_to_standardize | [f1 ... fN] | |
| adaptive_feature_selection | Booleana | Valore predefinito=True |
| featureMisPercent | integer | Valore predefinito=70 |
| coefRange | number | Valore predefinito=0.05 |
| percCasesSingleCategory | integer | Valore predefinito=95 |
| numCases | integer | Valore predefinito=24 |
| include_model_specifications | Booleana | Valore predefinito=True |
| include_record_summary | Booleana | Valore predefinito=True |
| include_field_transformations | Booleana | Valore predefinito=True |
| excluded_inputs | Booleana | Valore predefinito=True |
| evaluate_model_quality | Booleana | Valore predefinito=True |
| show_feature_importance_bar_chart | Booleana | Valore predefinito=True |
| show_feature_importance_word_cloud | Booleana | Valore predefinito=True |
| show_outlier_clusters interactive_table_and_chart | Booleana | Valore predefinito=True |
| show_outlier_clusters_pivot_table | Booleana | Valore predefinito=True |
| across_cluster_feature_importance | Booleana | Valore predefinito=True |
| across_cluster_profiles_pivot_table | Booleana | Valore predefinito=True |
| withinprofiles | Booleana | Valore predefinito=True |
| cluster_distances | Booleana | Valore predefinito=True |
| cluster_label | String Number | |
| label_prefix | String | |

Capitolo 14. Proprietà dei nodi dei nugget del modello

I nodi dei nugget del modello condividono le stesse proprietà comuni agli altri nodi. Per ulteriori informazioni, consultare l'argomento "Proprietà comuni dei nodi" a pagina 69.

Proprietà `applyanomalydetectionnode`

I nodi Modelli Rilevamento anomalie si possono utilizzare per generare un nugget del modello Rilevamento anomalie. Il nome di script di questo nugget del modello è `applyanomalydetectionnode`. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere "Proprietà `anomalydetectionnode`" a pagina 161

Tabella 140. proprietà `applyanomalydetectionnode`.

| Proprietà <code>applyanomalydetectionnode</code> | Valori | Descrizione proprietà |
|--|---------------------------------------|--|
| <code>anomaly_score_method</code> | FlagAndScore FlagOnly ScoreOnly | Determina quali output sono creati per il calcolo del punteggio. |
| <code>num_fields</code> | <i>numero intero</i> | Campi da inserire nel report. |
| <code>discard_records</code> | <i>indicatore</i> | Indica se i record sono scartati o meno dall'output. |
| <code>discard_anomalous_records</code> | <i>indicatore</i> | Indica se scartare i record anomali o <i>non</i> anomali. L'impostazione di default è <code>off</code> , ad indicare che i record <i>non</i> anomali vengono scartati. Altrimenti, se l'impostazione è su <code>on</code> , verranno scartati i record anomali. Questa proprietà è attivata solo se è attivata la proprietà <code>discard_records</code> . |

Proprietà `applyapriorinode`

I nodi Modelli Apriori si possono utilizzare per generare un nugget del modello Apriori. Il nome di script di questo nugget del modello è `applyapriorinode`. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere "Proprietà `apriorinode`" a pagina 163

Tabella 141. proprietà `applyapriorinode`.

| Proprietà <code>applyapriorinode</code> | Valori | Descrizione proprietà |
|---|---|-----------------------|
| <code>max_predictions</code> | <i>numero (intero)</i> | |
| <code>ignore_unmatched</code> | <i>indicatore</i> | |
| <code>allow_repeats</code> | <i>indicatore</i> | |
| <code>check_basket</code> | NoPredictions Predictions NoCheck | |
| <code>criterion</code> | Confidence Support RuleSupport Lift Deployability | |

Proprietà applyassociationrulesnode

È possibile utilizzare il nodo modellazione Regole di associazione per generare un nugget del modello Regole di associazione. Il nome di script di questo nugget del modello è *applyassociationrulesnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà associationrulesnode” a pagina 164.

Tabella 142. Proprietà applyassociationrulesnode

| Proprietà applyassociationrulesnode | Data type | Descrizione proprietà |
|-------------------------------------|--|---|
| max_predictions | integer | Il numero massimo di regole che possono essere applicate a ciascun input nel punteggio. |
| criterion | Confidence Rulesupport Lift Conditionsupport Distribuibilità | Selezionare la misura che determina l'efficacia delle regole. |
| allow_repeats | Booleana | Determina se nel punteggio vengono incluse le regole con la stessa previsione. |
| check_input | NoPredictions Predictions NoCheck | |

Proprietà applyautoclassifiernode

I nodi Modelli Classificatore automatico si possono utilizzare per generare un nugget del modello Classificatore automatico. Il nome di script di questo nugget del modello è *applyautoclassifiernode*. Per ulteriori informazioni sugli script per il nodo Modelli, vedere “Proprietà autoclassifiernode” a pagina 166

Tabella 143. proprietà applyautoclassifiernode.

| Proprietà applyautoclassifiernode | Valori | Descrizione proprietà |
|-----------------------------------|--|--|
| flag_ensemble_method | Voting ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence AverageRawPropensity | Specifica il metodo utilizzato per determinare il punteggio dell'insieme. Questa impostazione è valida solamente se l'obiettivo selezionato è un campo flag. |
| flag_voting_tie_selection | Random HighestConfidence RawPropensity | Se è selezionato un metodo del confronto, specifica le modalità di risoluzione delle situazioni di pari merito. Questa impostazione è valida solamente se l'obiettivo selezionato è un campo flag. |
| set_ensemble_method | Voting ConfidenceWeightedVoting HighestConfidence | Specifica il metodo utilizzato per determinare il punteggio dell'insieme. Questa impostazione è valida solamente se l'obiettivo selezionato è un campo insieme. |
| set_voting_tie_selection | Random HighestConfidence | Se è selezionato un metodo del confronto, specifica le modalità di risoluzione delle situazioni di pari merito. Questa impostazione è valida solamente se l'obiettivo selezionato è un campo nominale. |

Proprietà applyautoclusternode

I nodi Modelli Cluster automatico si possono utilizzare per generare un nugget del modello Cluster automatico. Il nome di script di questo nugget del modello è *applyautoclusternode*. Per questo nugget del modello non esistono altre proprietà. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà autoclusternode” a pagina 169

Proprietà applyautonumericnode

I nodi Modelli Numerico automatico si possono utilizzare per generare un nugget del modello Numerico automatico. Il nome di script di questo nugget del modello è *applyautonumericnode*. Per ulteriori informazioni sugli script per il nodo Modelli, vedere “Proprietà autonumericnode” a pagina 170

Tabella 144. proprietà applyautonumericnode.

| Proprietà applyautonumericnode | Valori | Descrizione proprietà |
|--------------------------------|------------|-----------------------|
| calculate_standard_error | indicatore | |

Proprietà applybayesnetnode

I nodi Modelli Rete bayesiana si possono utilizzare per generare un nugget del modello Rete bayesiana. Il nome di script di questo nugget del modello è *applybayesnetnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà bayesnetnode” a pagina 172.

Tabella 145. proprietà applybayesnetnode.

| Proprietà applybayesnetnode | Valori | Descrizione proprietà |
|---------------------------------|------------|-----------------------|
| all_probabilities | indicatore | |
| raw_propensity | indicatore | |
| adjusted_propensity | indicatore | |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |

Proprietà applyc50node

I nodi Modelli C5.0 si possono utilizzare per generare un nugget del modello C5.0. Il nome di script di questo nugget del modello è *applyc50node*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà c50node” a pagina 174.

Tabella 146. proprietà applyc50node.

| Proprietà applyc50node | Valori | Descrizione proprietà |
|---------------------------------|--------------------------|--|
| sql_generate | Never NoMissingValues | Consente di impostare le opzioni di generazione SQL durante l'esecuzione dell'insieme di regole. |
| calculate_conf | indicatore | Disponibile quando è attivata la generazione SQL, questa proprietà include i calcoli di confidenza nella struttura ad albero generata. |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |

Proprietà applycarmanode

I nodi Modelli CARMA si possono utilizzare per generare un nugget del modello CARMA. Il nome di script di questo nugget del modello è *applycarmanode*. Per questo nugget del modello non esistono altre proprietà. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà carmanode” a pagina 175.

Proprietà applycartnode

I nodi Modelli C&R possono essere utilizzati per generare un nugget del modello C&R Tree. Il nome di script di questo nugget del modello è *applycartnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà cartnode” a pagina 176.

Tabella 147. proprietà applycartnode.

| Proprietà applycartnode | Valori | Descrizione proprietà |
|---------------------------------|---|--|
| sql_generate | Never MissingValues NoMissingValues | Consente di impostare le opzioni di generazione SQL durante l'esecuzione dell'insieme di regole. |
| calculate_conf | <i>indicatore</i> | Disponibile quando è attivata la generazione SQL, questa proprietà include i calcoli di confidenza nella struttura ad albero generata. |
| display_rule_id | <i>indicatore</i> | Aggiunge un campo all'output del calcolo del punteggio che indica l'ID del nodo terminale al quale è assegnato ogni record. |
| calculate_raw_propensities | <i>indicatore</i> | |
| calculate_adjusted_propensities | <i>indicatore</i> | |

Proprietà applychaidnode

I nodi Modelli CHAID si possono utilizzare per generare un nugget del modello CHAID. Il nome di script di questo nugget del modello è *applychaidnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà chaidnode” a pagina 178.

Tabella 148. proprietà applychaidnode.

| Proprietà applychaidnode | Valori | Descrizione proprietà |
|---------------------------------|------------------------|---|
| sql_generate | Never MissingValues | |
| calculate_conf | <i>indicatore</i> | |
| display_rule_id | <i>indicatore</i> | Aggiunge un campo all'output del calcolo del punteggio che indica l'ID del nodo terminale al quale è assegnato ogni record. |
| calculate_raw_propensities | <i>indicatore</i> | |
| calculate_adjusted_propensities | <i>indicatore</i> | |

Proprietà applycoxregnode

I nodi Modelli Cox si possono utilizzare per generare un nugget del modello Cox. Il nome di script di questo nugget del modello è *applycoxregnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà coxregnode” a pagina 180.

Tabella 149. proprietà applycoxregnode.

| Proprietà applycoxregnode | Valori | Descrizione proprietà |
|---------------------------|--------------------|-----------------------|
| future_time_as | Intervals Campi | |
| time_interval | numero | |
| num_future_times | numero intero | |
| time_field | campo | |
| past_survival_time | campo | |
| all_probabilities | indicatore | |
| cumulative_hazard | indicatore | |

Proprietà applydecisionlistnode

I nodi Modelli Elenco di decisioni si possono utilizzare per generare un nugget del modello Elenco di decisioni. Il nome di script di questo nugget del modello è *applydecisionlistnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà decisionlistnode” a pagina 182.

Tabella 150. proprietà applydecisionlistnode.

| Proprietà applydecisionlistnode | Valori | Descrizione proprietà |
|---------------------------------|------------|--|
| enable_sql_generation | indicatore | Se questa proprietà è vera, IBM SPSS Modeler cerca di rinviare il modello Elenco di decisioni a SQL. |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |

Proprietà applydiscriminantnode

I nodi Modelli Discriminante si possono utilizzare per generare un nugget del modello Discriminante. Il nome di script di questo nugget del modello è *applydiscriminantnode*. Per ulteriori informazioni sugli script del nodo Modelli, vedere “Proprietà discriminantnode” a pagina 183.

Tabella 151. proprietà applydiscriminantnode.

| Proprietà applydiscriminantnode | Valori | Descrizione proprietà |
|---------------------------------|------------|-----------------------|
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |

Proprietà applyfactornode

I nodi modelli fattoriale/PCA si possono utilizzare per generare un nugget del modello fattoriale/PCA. Il nome di script di questo nugget del modello è *applyfactornode*. Per questo nugget del modello non esistono altre proprietà. Per ulteriori informazioni, sugli script del nodo Modelli specifico, vedere “Proprietà factornode” a pagina 185.

Proprietà applyfeatureselectionnode

I nodi Modelli Selezione funzioni si possono utilizzare per generare un nugget del modello Selezione funzioni. Il nome di script di questo nugget del modello è *applyfeatureselectionnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà featureselectionnode” a pagina 186.

Tabella 152. proprietà applyfeatureselectionnode.

| Proprietà applyfeatureselectionnode | Valori | Descrizione proprietà |
|-------------------------------------|--------|--|
| selected_ranked_fields | | Specifica quali campi classificati sono selezionati nel browser dei modelli. |
| selected_screened_fields | | Specifica quali campi sottoposti a screening sono selezionati nel browser dei modelli. |

Proprietà applygeneralizedlinearnode

I nodi Modelli lineari generalizzati (GenLin) si possono utilizzare per generare un nugget del modello Lineare generalizzato. Il nome di script di questo nugget del modello è *applygeneralizedlinearnode*. Per ulteriori informazioni sugli script del nodo Modelli, vedere “Proprietà genlinnode” a pagina 188.

Tabella 153. proprietà applygeneralizedlinearnode.

| Proprietà applygeneralizedlinearnode | Valori | Descrizione proprietà |
|--------------------------------------|-------------------|-----------------------|
| calculate_raw_propensities | <i>indicatore</i> | |
| calculate_adjusted_propensities | <i>indicatore</i> | |

Proprietà applyglmnode

I nodi Modelli GLMM si possono utilizzare per generare un nugget del modello GLMM. Il nome di script di questo nugget del modello è *applyglmnode*. Per ulteriori informazioni sugli script del nodo Modelli, vedere “Proprietà glmnode” a pagina 191.

Tabella 154. Proprietà applyglmnode.

| Proprietà applyglmnode | Valori | Descrizione proprietà |
|------------------------------|-----------------------------|---|
| confidence | onProbability onIncrease | Base per il calcolo del valore di confidenza del punteggio: probabilità prevista più alta o differenza tra le probabilità più alte e la seconda massima prevista. |
| score_category_probabilities | <i>indicatore</i> | Se impostato su True, produce le probabilità previste per i target di categoria. Viene creato un campo per ogni categoria. Il valore di default è False. |
| max_categories | <i>numero intero</i> | Numero massimo di categorie per cui prevedere le probabilità. Utilizzata solo se score_category_probabilities è True. |
| score_propensity | <i>indicatore</i> | Se impostato su True, produce punteggi di propensione grezza (verosimiglianza del risultato "True") per i modelli con obiettivi flag. Se le partizioni sono attive, producono anche punteggi di propensione regolata basati sulla partizione di test. Il valore di default è False. |

proprietà applygle

Il nodo Modellazione GLE può essere utilizzato per generare un nugget del modello GLE. Il nome dello script di questo nugget del modello è *applygle*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà gle” a pagina 195.

Tabella 155. Proprietà applygle

| Proprietà applygle | Valori | Descrizione proprietà |
|-----------------------|---------------|--|
| enable_sql_generation | udf native | Utilizzato per impostare le opzioni di generazione SQL durante l'esecuzione del flusso. Scegliere di eseguire il pushback verso il database e il calcolo del punteggio utilizzando un adattatore per calcolo punteggio SPSS Modeler Server (se connessi a un database con un adattatore per calcolo punteggio installato), oppure eseguire il calcolo all'interno di SPSS Modeler. |

Proprietà applykmeansnode

I nodi Modelli Medie K si possono utilizzare per generare un nugget del modello Medie K. Il nome di script di questo nugget del modello è *applykmeansnode*. Per questo nugget del modello non esistono altre proprietà. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà kmeansnode” a pagina 199.

Proprietà applyknnnode

I nodi Modelli KNN possono essere utilizzati per generare un nugget del modello KNN. Il nome di script di questo nugget del modello è *applyknnnode*. Per ulteriori informazioni sugli script del nodo Modelli, vedere “Proprietà knnnode” a pagina 200.

Tabella 156. proprietà applyknnnode.

| Proprietà applyknnnode | Valori | Descrizione proprietà |
|------------------------|------------|-----------------------|
| all_probabilities | indicatore | |
| save_distances | indicatore | |

Proprietà applykohonennode

I nodi Modelli Kohonen si possono utilizzare per generare un nugget del modello Kohonen. Il nome di script di questo nugget del modello è *applykohonennode*. Per questo nugget del modello non esistono altre proprietà. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà c50node” a pagina 174.

Proprietà applylinearnode

I nodi Modelli lineari si possono utilizzare per generare un nugget del modello lineari. Il nome di script di questo nugget del modello è *applylinearnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà linearnode” a pagina 203.

Tabella 157. Proprietà *applylinearnode*.

| Proprietà linear | Valori | Descrizione proprietà |
|-----------------------|------------|-----------------------|
| use_custom_name | indicatore | |
| custom_name | stringa | |
| enable_sql_generation | indicatore | |

Proprietà applylinearasnode

I nodi Modelli Linear-AS si possono utilizzare per generare un nugget del modello Linear-AS. Il nome di script di questo nugget del modello è *applylinearasnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà linearasnode” a pagina 204.

Tabella 158. Proprietà *applylinearasnode*

| Proprietà applylinearasnode | Valori | Descrizione proprietà |
|-----------------------------|---------------|------------------------------|
| enable_sql_generation | udf native | Il valore predefinito è udf. |

Proprietà applylogregnode

I nodi Modelli Regressione logistica si possono utilizzare per generare un nugget del modello Regressione logistica. Il nome di script di questo nugget del modello è *applylogregnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà logregnode” a pagina 205.

Tabella 159. proprietà *applylogregnode*.

| Proprietà applylogregnode | Valori | Descrizione proprietà |
|----------------------------|------------|-----------------------|
| calculate_raw_propensities | indicatore | |
| calculate_conf | indicatore | |
| enable_sql_generation | indicatore | |

Proprietà applylsvmnode

I nodi modello LSVM possono essere utilizzati per generare un nugget del modello LSVM. Il nome di script di questo nugget del modello è *applylsvmnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà lsvmnode” a pagina 210. Il nodo modelli LSVM viene eseguito quando viene eseguita la connessione a IBM SPSS Analytic Server.

Tabella 160. Proprietà *applylsvmnode*

| Proprietà applylsvmnode | Valori | Descrizione proprietà |
|----------------------------|---------------|--|
| calculate_raw_propensities | indicatore | Specifica se calcolare i punteggi di propensione grezza. |
| enable_sql_generation | udf native | Specifica se calcolare il punteggio utilizzando l'Adattatore per calcolo del punteggio (se installato) o nel processo oppure se calcolare il punteggio all'esterno del database. |

Proprietà applyneuralnetnode

I nodi Modelli Rete neurale si possono utilizzare per generare un nugget del modello Rete neurale. Il nome di script di questo nugget del modello è *applyneuralnetnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà neuralnetnode” a pagina 211.

Attenzione: in questa release è disponibile una nuova versione del nugget Rete neurale con funzionalità avanzate, descritta nella sezione che segue (*applyneuralnetwork*). Sebbene la versione precedente sia ancora disponibile, si consiglia di aggiornare gli script in modo da utilizzare la nuova versione. I dettagli della versione precedente vengono mantenuti in questa sezione per riferimento, ma nelle versioni future non sarà più supportata.

Tabella 161. proprietà applyneuralnetnode.

| Proprietà applyneuralnetnode | Valori | Descrizione proprietà |
|---------------------------------|-----------------------|--|
| calculate_conf | indicatore | Disponibile quando è attivata la generazione SQL, questa proprietà include i calcoli di confidenza nella struttura ad albero generata. |
| enable_sql_generation | indicatore | |
| nn_score_method | Difference SoftMax | |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |

Proprietà applyneuralnetworknode

I nodi Modelli Rete neurale si possono utilizzare per generare un nugget del modello Rete neurale. Il nome di script di questo nugget del modello è *applyneuralnetworknode*. Per ulteriori informazioni sugli script del nodo Modelli specifico vedere in .

Tabella 162. proprietà applyneuralnetworknode

| Proprietà applyneuralnetworknode | Valori | Descrizione proprietà |
|----------------------------------|-----------------------------|-----------------------|
| use_custom_name | indicatore | |
| custom_name | stringa | |
| confidence | onProbability onIncrease | |
| score_category_probabilities | indicatore | |
| max_categories | numero | |
| score_propensity | indicatore | |

Proprietà applyquestnode

I nodi Modelli QUEST si possono utilizzare per generare un nugget del modello QUEST. Il nome di script di questo nugget del modello è *applyquestnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà questnode” a pagina 215.

Tabella 163. proprietà applyquestnode.

| Proprietà applyquestnode | Valori | Descrizione proprietà |
|--------------------------|---|-----------------------|
| sql_generate | Never MissingValues NoMissingValues | |

Tabella 163. proprietà applyquestnode (Continua).

| Proprietà applyquestnode | Valori | Descrizione proprietà |
|---------------------------------|------------|---|
| calculate_conf | indicatore | |
| display_rule_id | indicatore | Aggiunge un campo all'output del calcolo del punteggio che indica l'ID del nodo terminale al quale è assegnato ogni record. |
| calculate_raw_propensities | indicatore | |
| calculate_adjusted_propensities | indicatore | |

Proprietà applyr

I nodi di creazione R possono essere utilizzati per generare un nugget del modello R. Il nome di script di questo nugget del modello è *applyr*. Per ulteriori informazioni sugli script del nodo Modelli, vedere "Proprietà buildr" a pagina 173.

Tabella 164. proprietà di applyr

| Proprietà applyr | Valori | Descrizione proprietà |
|------------------------|------------------------------------|--|
| score_syntax | stringa | Sintassi degli script R per il calcolo del punteggio del modello. |
| convert_flags | StringsAndDoubles LogicalValues | Opzione per la conversione dei campi indicatore. |
| convert_datetime | indicatore | L'opzione per convertire le variabili con formati di date o data/ora in formati data/ora R. |
| convert_datetime_class | POSIXct POSIXlt | Le opzioni per specificare in quale formato vengono convertite le variabili con formati data o data/ora. |
| convert_missing | indicatore | Opzione per convertire il valore mancanti nel valore R NA. |
| use_batch_size | indicatore | Attiva l'utilizzo dell'elaborazione batch |
| batch_size | integer | Specifica il numero di record di dati da includere in ciascun batch |

Proprietà applyrandomtrees

Il nodo modelli Random Trees può essere utilizzato per generare un nugget del modello Random Trees. Il nome di script di questo nugget del modello è *applyrandomtrees*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere "Proprietà randomtrees" a pagina 217.

Tabella 165. proprietà applyrandomtrees

| Proprietà applyrandomtrees | Valori | Descrizione proprietà |
|----------------------------|------------|--|
| calculate_conf | indicatore | Questa proprietà include i calcoli di confidenza nella struttura ad albero generata. |

Tabella 165. proprietà *applyrandomtrees* (Continua)

| Proprietà <i>applyrandomtrees</i> | Valori | Descrizione proprietà |
|-----------------------------------|---------------|--|
| enable_sql_generation | udf native | Utilizzato per impostare le opzioni di generazione SQL durante l'esecuzione del flusso. Scegliere di eseguire il pushback verso il database e il calcolo del punteggio utilizzando un adattatore per calcolo punteggio SPSS Modeler Server (se connessi a un database con un adattatore per calcolo punteggio installato), oppure eseguire il calcolo all'interno di SPSS Modeler. |

Proprietà *applyregressionnode*

I nodi Modelli Regressione lineare si possono utilizzare per generare un nugget del modello Regressione lineare. Il nome di script di questo nugget del modello è *applyregressionnode*. Per questo nugget del modello non esistono altre proprietà. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere "Proprietà *regressionnode*" a pagina 218.

Proprietà *applyselflearningnode*

I nodi Modelli SLRM (Risposta autoapprendimento) si possono utilizzare per generare un nugget del modello SLRM. Il nome di script di questo nugget del modello è *applyselflearningnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere "Proprietà *slrmnode*" a pagina 222.

Tabella 166. proprietà *applyselflearningnode*.

| Proprietà <i>applyselflearningnode</i> | Valori | Descrizione proprietà |
|--|-------------------------|--|
| max_predictions | numero | |
| randomization | numero | |
| scoring_random_seed | numero | |
| sort | ascending descending | Specifica se verranno visualizzate per prime le offerte con i punteggi più alti o più bassi. |
| model_reliability | indicatore | Tiene conto dell'opzione di affidabilità del modello inclusa nella scheda Impostazioni. |

Proprietà *applysequencenode*

I nodi Modelli Sequenza si possono utilizzare per generare un nugget del modello Sequenza. Il nome di script di questo nugget del modello è *applysequencenode*. Per questo nugget del modello non esistono altre proprietà. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere "Proprietà *sequencenode*" a pagina 220.

Proprietà *appliesvmnode*

I nodi Modelli SVM si possono utilizzare per generare un nugget del modello SVM. Il nome di script di questo nugget del modello è *appliesvmnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere "Proprietà *svmnode*" a pagina 227.

Tabella 167. proprietà *appliesvmnode*.

| Proprietà <i>appliesvmnode</i> | Valori | Descrizione proprietà |
|--------------------------------|------------|-----------------------|
| all_probabilities | indicatore | |

Tabella 167. proprietà *applysvmnode* (Continua).

| Proprietà <i>applysvmnode</i> | Valori | Descrizione proprietà |
|---------------------------------|-------------------|-----------------------|
| calculate_raw_propensities | <i>indicatore</i> | |
| calculate_adjusted_propensities | <i>indicatore</i> | |

Proprietà *applystnode*

È possibile utilizzare il nodo di modellazione STP per generare un nugget del modello associato, che visualizzi l'output del modello nel visualizzatore output. Il nome di script di questo nugget del modello è *applystnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere "Proprietà *stnode*" a pagina 223.

Tabella 168. proprietà *applystnode*

| Proprietà <i>applystnode</i> | Data type | Descrizione proprietà |
|------------------------------|-----------------|------------------------|
| uncertainty_factor | <i>Booleana</i> | Minimo 0, massimo 100. |

Proprietà *applytcmnode*

I nodi di modeling TCM (Temporal Causal Modeling) possono essere utilizzati per generare un nugget del modello TCM. Il nome di script di questo nugget del modello è *applytcmnode*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere "Proprietà *tcmnode*" a pagina 228.

Tabella 169. Proprietà *applytcmnode*

| Proprietà <i>applytcmnode</i> | Valori | Descrizione proprietà |
|-------------------------------|-----------------|-----------------------|
| ext_future | <i>booleano</i> | |
| ext_future_num | <i>integer</i> | |
| noise_res | <i>booleano</i> | |
| conf_limits | <i>booleano</i> | |
| target_fields | <i>elenco</i> | |
| target_series | <i>elenco</i> | |

Proprietà *applytimeseriesnode*

I nodi Modelli Serie temporali si possono utilizzare per generare un nugget del modello Serie temporali. Il nome di script di questo nugget del modello è *applytimeseriesnode*. Per ulteriori informazioni sugli script del nodo Modelli, vedere "Proprietà *timeseriesnode*" a pagina 231.

Tabella 170. proprietà *applytimeseriesnode*.

| Proprietà <i>applytimeseriesnode</i> | Valori | Descrizione proprietà |
|--------------------------------------|-------------------|-----------------------|
| calculate_conf | <i>indicatore</i> | |
| calculate_residuals | <i>indicatore</i> | |

Proprietà applytreeas

I nodi di modeling Tree-AS si possono utilizzare per generare un nugget del modello Tree-AS. Il nome di script di questo nugget del modello è *applytreeas*. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà treeas” a pagina 233.

Tabella 171. Proprietà applytreeas

| Proprietà applytreeas | Valori | Descrizione proprietà |
|-----------------------|-------------------|--|
| calculate_conf | <i>indicatore</i> | Questa proprietà include i calcoli di confidenza nella struttura ad albero generata. |
| display_rule_id | <i>indicatore</i> | Aggiunge un campo all'output del calcolo del punteggio che indica l'ID del nodo terminale al quale è assegnato ogni record. |
| enable_sql_generation | udf native | Utilizzato per impostare le opzioni di generazione SQL durante l'esecuzione del flusso. Scegliere di eseguire il pushback verso il database e il calcolo del punteggio utilizzando un adattatore per calcolo punteggio SPSS Modeler Server (se connessi a un database con un adattatore per calcolo punteggio installato), oppure eseguire il calcolo all'interno di SPSS Modeler. |

Proprietà applytwostepnode

I nodi Modelli TwoStep si possono utilizzare per generare un nugget del modello TwoStep. Il nome di script di questo nugget del modello è *applytwostepnode*. Per questo nugget del modello non esistono altre proprietà. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà twostepnode” a pagina 235.

Proprietà applytwostepAS

I nodi modelli TwoStep AS possono essere utilizzati per generare un nugget del modello TwoStep AS. Il nome di script di questo nugget del modello è *applytwostepAS*. Per questo nugget del modello non esistono altre proprietà. Per ulteriori informazioni sugli script del nodo Modelli specifico, vedere “Proprietà twostepAS” a pagina 236.

Capitolo 15. Proprietà dei nodi Modelli database

IBM SPSS Modeler supporta l'integrazione con gli strumenti di modellazione e di data mining offerti dai fornitori di database, quali Microsoft SQL Server Analysis Services, Oracle Data Mining, IBM DB2 InfoSphere Warehouse e IBM Netezza Analytics. Operando all'interno dell'applicazione IBM SPSS Modeler, è possibile creare modelli e calcolarne il punteggio mediante algoritmi nativi del database. I modelli di database possono essere creati e manipolati anche tramite script utilizzando le proprietà descritte in questa sezione.

Ad esempio, il seguente estratto di script illustra la creazione di un modello Strutture ad albero delle decisioni Microsoft utilizzando l'interfaccia script di IBM SPSS Modeler:

```
stream = modeler.script.stream()
msbuilder = stream.createAt("mstreenode", "MSBuilder", 200, 200)

msbuilder.setPropertyValue("analysis_server_name", 'localhost')
msbuilder.setPropertyValue("analysis_database_name", 'TESTDB')
msbuilder.setPropertyValue("mode", 'Expert')
msbuilder.setPropertyValue("datasource", 'LocalServer')
msbuilder.setPropertyValue("target", 'Drug')
msbuilder.setPropertyValue("inputs", ['Age', 'Sex'])
msbuilder.setPropertyValue("unique_field", 'IDX')
msbuilder.setPropertyValue("custom_fields", True)
msbuilder.setPropertyValue("model_name", 'MSDRUG')

typenode = stream.findByType("type", None)
stream.link(typenode, msbuilder)
results = []
msbuilder.run(results)
msapplier = stream.createModelApplierAt(results[0], "Drug", 200, 300)
tablenode = stream.createAt("table", "Results", 300, 300)
stream.linkBetween(msapplier, typenode, tablenode)
msapplier.setPropertyValue("sql_generate", True)
tablenode.run([])
```

Proprietà dei nodi Modelli Microsoft

Proprietà dei nodi Modelli Microsoft

Proprietà comuni

Le seguenti proprietà sono comuni ai nodi Modelli database Microsoft.

Tabella 172. Proprietà comuni dei nodi Microsoft

| Proprietà comuni dei nodi Microsoft | Valori | Descrizione proprietà |
|-------------------------------------|-------------------|--|
| analysis_database_name | <i>stringa</i> | Nome del database di Analysis Services. |
| analysis_server_name | <i>stringa</i> | Nome dell'host di Analysis Services. |
| use_transactional_data | <i>indicatore</i> | Specifica se i dati di input sono in formato tabulare o transazionale. |
| inputs | <i>elenco</i> | Campi di input per dati in formato tabellare. |
| obiettivo | <i>campo</i> | Campo predittore (non applicabile al nodo Raggruppamento cluster MS o Cluster di sequenze MS). |

Tabella 172. Proprietà comuni dei nodi Microsoft (Continua)

| Proprietà comuni dei nodi Microsoft | Valori | Descrizione proprietà |
|-------------------------------------|-------------|---|
| unique_field | campo | Campo chiave. |
| msas_parameters | strutturato | Parametri degli algoritmi. Per ulteriori informazioni, consultare l'argomento "Parametri degli algoritmi" a pagina 255. |
| with_drillthrough | indicatore | Opzione Con funzione drill-through. |

Struttura ad albero delle decisioni MS

Per i nodi di tipo `mstreenode` non sono definite proprietà specifiche. Vedere le proprietà comuni di Microsoft all'inizio di questa sezione.

Raggruppamento cluster MS

Per i nodi di tipo `msclusternode` non sono definite proprietà specifiche. Vedere le proprietà comuni di Microsoft all'inizio di questa sezione.

Regole di associazione MS

Per i nodi di tipo `msassocnode` sono disponibili le seguenti proprietà specifiche:

Tabella 173. proprietà `msassocnode`

| Proprietà <code>msassocnode</code> | Valori | Descrizione proprietà |
|------------------------------------|--------|---|
| id_field | campo | Identifica le singole transazioni nei dati. |
| trans_inputs | elenco | Campi di input per dati transazionali. |
| transactional_target | campo | Campo predittore (dati transazionali). |

Naive Bayes MS

Per i nodi di tipo `msbayesnode` non sono definite proprietà specifiche. Vedere le proprietà comuni di Microsoft all'inizio di questa sezione.

Regressione lineare MS

Per i nodi di tipo `msregressionnode` non sono definite proprietà specifiche. Vedere le proprietà comuni di Microsoft all'inizio di questa sezione.

Rete neurale MS

Per i nodi di tipo `msneuralnetworknode` non sono definite proprietà specifiche. Vedere le proprietà comuni di Microsoft all'inizio di questa sezione.

Regressione logistica MS

Per i nodi di tipo `mslogisticnode` non sono definite proprietà specifiche. Vedere le proprietà comuni di Microsoft all'inizio di questa sezione.

Serie temporali MS

Per i nodi di tipo `mstimeseriesnode` non sono definite proprietà specifiche. Vedere le proprietà comuni di Microsoft all'inizio di questa sezione.

MS Sequence Clustering

Per i nodi di tipo `mssequenceclusternode` sono disponibili le seguenti proprietà specifiche:

Tabella 174. proprietà `mssequenceclusternode`

| Proprietà <code>mssequenceclusternode</code> | Valori | Descrizione proprietà |
|--|---------------|---|
| <code>id_field</code> | <i>campo</i> | Identifica le singole transazioni nei dati. |
| <code>input_fields</code> | <i>elenco</i> | Campi di input per dati transazionali. |
| <code>sequence_field</code> | <i>campo</i> | Identificativo sequenza. |
| <code>target_field</code> | <i>campo</i> | Campo predittore (dati in formato tabellare). |

Parametri degli algoritmi

Ogni tipo di modello di database Microsoft possiede parametri specifici che è possibile impostare mediante la proprietà `msas_parameters`, per esempio:

```
stream = modeler.script.stream()
msregressionnode = stream.findByType("msregression", None)
msregressionnode.setPropertyValue("msas_parameters", [["MAXIMUM_INPUT_ATTRIBUTES", 255],
["MAXIMUM_OUTPUT_ATTRIBUTES", 255]])
```

Tali parametri sono derivati da SQL Server. Per visualizzare i parametri relativi ai singoli nodi:

1. Collocare un nodo origine del database nell'area.
2. Aprire il nodo origine del database.
3. Selezionare un'origine valida dall'elenco a discesa **Sorgente dati**.
4. Selezionare una tabella valida dall'elenco **Nome tabella**.
5. Fare clic su **OK** per chiudere il nodo origine del database.
6. Collegare il nodo Modelli database Microsoft di cui si desiderano elencare le proprietà.
7. Aprire il nodo Modelli database.
8. Selezionare la scheda **Livello avanzato**.

Vengono visualizzate le proprietà `msas_parameters` disponibili per quel nodo.

Proprietà dei nugget del modello Microsoft

Le seguenti proprietà sono relative ai nugget del modello creati mediante i nodi Modelli database Microsoft.

Struttura ad albero delle decisioni MS

Tabella 175. Proprietà della struttura ad albero delle decisioni MS.

| Proprietà <code>appliedstreenode</code> | Valori | Descrizione |
|---|----------------|--|
| <code>analysis_database_name</code> | <i>stringa</i> | Il calcolo del punteggio di questo nodo può essere eseguito direttamente in un flusso. Questa proprietà consente di identificare il nome del database di Analysis Services. |
| <code>analysis_server_name</code> | <i>stringa</i> | Nome dell'host di Analysis Server. |
| <code>datasource</code> | <i>stringa</i> | Nome del DSN (nome sorgente dati, Data Source Name) ODBC SQL Server. |

Tabella 175. Proprietà della struttura ad albero delle decisioni MS (Continua).

| Proprietà applymstreenode | Valori | Descrizione |
|----------------------------------|------------|----------------------------|
| sql_generate | indicatore | Attiva la generazione SQL. |

Regressione lineare MS

Tabella 176. Proprietà della regressione lineare MS.

| Proprietà applymsregressionnode | Valori | Descrizione |
|--|---------|--|
| analysis_database_name | stringa | Il calcolo del punteggio di questo nodo può essere eseguito direttamente in un flusso. Questa proprietà consente di identificare il nome del database di Analysis Services. |
| analysis_server_name | stringa | Nome dell'host di Analysis Server. |

Rete neurale MS

Tabella 177. Proprietà della rete neurale MS.

| Proprietà applymsneuralnetworknode | Valori | Descrizione |
|---|---------|--|
| analysis_database_name | stringa | Il calcolo del punteggio di questo nodo può essere eseguito direttamente in un flusso. Questa proprietà consente di identificare il nome del database di Analysis Services. |
| analysis_server_name | stringa | Nome dell'host di Analysis Server. |

Regressione logistica MS

Tabella 178. Proprietà della regressione logistica MS.

| Proprietà applymslogisticnode | Valori | Descrizione |
|--------------------------------------|---------|--|
| analysis_database_name | stringa | Il calcolo del punteggio di questo nodo può essere eseguito direttamente in un flusso. Questa proprietà consente di identificare il nome del database di Analysis Services. |
| analysis_server_name | stringa | Nome dell'host di Analysis Server. |

Serie temporali MS

Tabella 179. Proprietà delle serie temporali MS.

| proprietà applymstimeseriesnode | Valori | Descrizione |
|--|---|--|
| analysis_database_name | stringa | Il calcolo del punteggio di questo nodo può essere eseguito direttamente in un flusso. Questa proprietà consente di identificare il nome del database di Analysis Services. |
| analysis_server_name | stringa | Nome dell'host di Analysis Server. |
| start_from | new_prediction historical_prediction | Specifica se effettuare previsioni future o storiche. |

Tabella 179. Proprietà delle serie temporali MS (Continua).

| proprietà <code>appliedtimeseriesnode</code> | Valori | Descrizione |
|--|---------------|--|
| <code>new_step</code> | <i>numero</i> | Definisce il periodo di tempo iniziale per le previsioni future. |
| <code>historical_step</code> | <i>numero</i> | Definisce il periodo di tempo iniziale per le previsioni storiche. |
| <code>end_step</code> | <i>numero</i> | Definisce il periodo di tempo finale per le previsioni. |

MS Sequence Clustering

Tabella 180. Proprietà del cluster di sequenze MS.

| proprietà <code>appliedsequenceclusternode</code> | Valori | Descrizione |
|---|----------------|--|
| <code>analysis_database_name</code> | <i>stringa</i> | Il calcolo del punteggio di questo nodo può essere eseguito direttamente in un flusso. Questa proprietà consente di identificare il nome del database di Analysis Services. |
| <code>analysis_server_name</code> | <i>stringa</i> | Nome dell'host di Analysis Server. |

Proprietà dei nodi Modelli Oracle

Proprietà dei nodi Modelli Oracle

Le seguenti proprietà sono comuni ai nodi Modelli database Oracle.

Tabella 181. Proprietà comuni dei nodi Oracle.

| Proprietà comuni dei nodi Oracle | Valori | Descrizione proprietà |
|-----------------------------------|------------------------|--|
| <code>obiettivo</code> | <i>campo</i> | |
| <code>inputs</code> | <i>Elenco di campi</i> | |
| <code>partition</code> | <i>campo</i> | Campo utilizzato per partizionare i dati in campioni distinti per le fasi di addestramento, di test e di convalida della creazione del modello. |
| <code>datasource</code> | | |
| <code>username</code> | | |
| <code>password</code> | | |
| <code>epassword</code> | | |
| <code>use_model_name</code> | <i>indicatore</i> | |
| <code>model_name</code> | <i>stringa</i> | Nome personalizzato per il nuovo modello. |
| <code>use_partitioned_data</code> | <i>indicatore</i> | Se è definito un campo partizione, questa opzione garantisce che per la creazione del modello verranno utilizzati solo i dati della partizione di addestramento. |
| <code>unique_field</code> | <i>campo</i> | |
| <code>auto_data_prep</code> | <i>indicatore</i> | Attiva o disattiva la funzione di preparazione dei dati automatici di Oracle (solo database 11g). |

Tabella 181. Proprietà comuni dei nodi Oracle (Continua).

| Proprietà comuni dei nodi Oracle | Valori | Descrizione proprietà |
|----------------------------------|------------------|---|
| costs | strutturata | Proprietà strutturata nel formato: [[drugA drugB 1.5] [drugA drugC 2.1]], dove gli argomenti racchiusi tra [] rappresentano i costi previsti effettivi. |
| mode | Simple Expert | Consente di ignorare determinate proprietà se impostate su Simple, come illustrato nelle proprietà dei singoli nodi. |
| use_prediction_probability | indicatore | |
| prediction_probability | stringa | |
| use_prediction_set | indicatore | |

Naive Bayes Oracle

Per i nodi di tipo oranbnode sono disponibili le seguenti proprietà.

Tabella 182. Proprietà oranbnode.

| Proprietà oranbnode | Valori | Descrizione proprietà |
|---------------------------------------|-------------------------|--|
| singleton_threshold | numero | 0.0–1.0.* |
| pairwise_threshold | numero | 0.0–1.0.* |
| distribuzioni di probabilità a priori | Data Equal Custom | |
| custom_priors | strutturata | Proprietà strutturata nel formato: set :oranbnode.custom_priors = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]] |

* Proprietà ignorata se mode è impostata su Simple.

Bayes adattivo Oracle

Per i nodi di tipo oraabnnode sono disponibili le seguenti proprietà.

Tabella 183. Proprietà oraabnnode.

| Proprietà oraabnnode | Valori | Descrizione proprietà |
|---------------------------------------|---|---|
| model_type | SingleFeature MultiFeature NaiveBayes | |
| use_execution_time_limit | indicatore | * |
| execution_time_limit | numero intero | Il valore deve essere maggiore di 0.* |
| max_naive_bayes_predictors | numero intero | Il valore deve essere maggiore di 0.* |
| max_predictors | numero intero | Il valore deve essere maggiore di 0.* |
| distribuzioni di probabilità a priori | Data Equal Custom | |
| custom_priors | strutturata | Proprietà strutturata nel formato: set :oraabnnode.custom_priors = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]] |

* Proprietà ignorata se mode è impostata su Simple.

SVM Oracle

Per i nodi di tipo orasvmnode sono disponibili le seguenti proprietà.

Tabella 184. proprietà orasvmnode.

| Proprietà orasvmnode | Valori | Descrizione proprietà |
|------------------------|------------------------------|--|
| active_learning | Enable Disable | |
| kernel_function | Linear Gaussian System | |
| normalization_method | zscore minmax none | |
| kernel_cache_size | numero intero | Solo kernel gaussiano. Il valore deve essere maggiore di 0.* |
| convergence_tolerance | numero | Il valore deve essere maggiore di 0.* |
| use_standard_deviation | indicatore | Solo kernel gaussiano.* |
| standard_deviation | numero | Il valore deve essere maggiore di 0.* |
| use_epsilon | indicatore | Solo modelli di regressione.* |
| epsilon | numero | Il valore deve essere maggiore di 0.* |
| use_complexity_factor | indicatore | * |
| complexity_factor | numero | * |
| use_outlier_rate | indicatore | Solo variante a una classe.* |
| outlier_rate | numero | Solo variante a una classe. 0.0-1.0.* |
| weights | Data Equal Custom | |
| custom_weights | strutturata | Proprietà strutturata nel formato: set :orasvmnode.custom_weights = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]] |

* Proprietà ignorata se mode è impostata su Simple.

Modelli lineari generalizzati Oracle

Le seguenti proprietà sono disponibili per i nodi di tipo oraglmnode.

Tabella 185. proprietà oraglmnode.

| Proprietà oraglmnode | Valori | Descrizione proprietà |
|------------------------|---------------------------------------|-----------------------|
| normalization_method | zscore minmax none | |
| missing_value_handling | ReplaceWithMean UseCompleteRecords | |
| use_row_weights | indicatore | * |
| row_weights_field | campo | * |

Tabella 185. proprietà oraglmnode (Continua).

| Proprietà oraglmnode | Valori | Descrizione proprietà |
|------------------------|-------------------|-----------------------|
| save_row_diagnostics | indicatore | * |
| row_diagnostics_table | stringa | * |
| coefficient_confidence | numero | * |
| use_reference_category | indicatore | * |
| reference_category | stringa | * |
| ridge_regression | Auto Off On | * |
| parameter_value | numero | * |
| vif_for_ridge | indicatore | * |

* Proprietà ignorata se mode è impostata su Simple.

struttura ad albero delle decisioni Oracle

Le seguenti proprietà sono disponibili per i nodi di tipo oradecisiontreenode.

Tabella 186. proprietà oradecisiontreenode.

| Proprietà oradecisiontreenode | Valori | Descrizione proprietà |
|-------------------------------|-----------------|---------------------------------------|
| use_costs | indicatore | |
| impurity_metric | Entropy Gini | |
| term_max_depth | numero intero | 2–20.* |
| term_minpct_node | numero | 0.0–10.0.* |
| term_minpct_split | numero | 0.0–20.0.* |
| term_minrec_node | numero intero | Il valore deve essere maggiore di 0.* |
| term_minrec_split | numero intero | Il valore deve essere maggiore di 0.* |
| display_rule_ids | indicatore | * |

* Proprietà ignorata se mode è impostata su Simple.

O-Cluster Oracle

Le seguenti proprietà sono disponibili per i nodi di tipo oraoclusternode.

Tabella 187. proprietà oraoclusternode.

| Proprietà oraoclusternode | Valori | Descrizione proprietà |
|---------------------------|---------------|---------------------------------------|
| max_num_clusters | numero intero | Il valore deve essere maggiore di 0. |
| max_buffer | numero intero | Il valore deve essere maggiore di 0.* |
| sensitivity | numero | 0.0–1.0.* |

* Proprietà ignorata se mode è impostata su Simple.

Medie K Oracle

Le seguenti proprietà sono disponibili per i nodi di tipo orakmeansnode.

Tabella 188. proprietà orakmeansnode.

| Proprietà orakmeansnode | Valori | Descrizione proprietà |
|-------------------------|--------------------------|--|
| num_clusters | numero intero | Il valore deve essere maggiore di 0. |
| normalization_method | zscore minmax none | |
| distance_function | Euclidean Cosine | |
| iterazioni | numero intero | 0–20.* |
| conv_tolerance | numero | 0.0–0.5.* |
| split_criterion | Variance Size | L'impostazione di default è Variance.* |
| num_bins | numero intero | Il valore deve essere maggiore di 0.* |
| block_growth | numero intero | 1–5.* |
| min_pct_attr_support | numero | 0.0–1.0.* |

* Proprietà ignorata se mode è impostata su Simple.

NMF Oracle

Le seguenti proprietà sono disponibili per i nodi di tipo oranmfnode.

Tabella 189. proprietà oranmfnode.

| Proprietà oranmfnode | Valori | Descrizione proprietà |
|----------------------|----------------|---|
| normalization_method | minmax none | |
| use_num_features | indicatore | * |
| num_features | numero intero | 0–1. Il valore di default viene stimato dall' algoritmo in base ai dati.* |
| random_seed | numero | * |
| num_iterations | numero intero | 0–500.* |
| conv_tolerance | numero | 0.0–0.5.* |
| display_all_features | indicatore | * |

* Proprietà ignorata se mode è impostata su Simple.

Apriori Oracle

Le seguenti proprietà sono disponibili per i nodi di tipo oraapriorinode.

Tabella 190. proprietà oraapriorinode.

| Proprietà oraapriorinode | Valori | Descrizione proprietà |
|--------------------------|--------|-----------------------|
| content_field | campo | |
| id_field | campo | |

Tabella 190. proprietà oraapriorinode (Continua).

| Proprietà oraapriorinode | Valori | Descrizione proprietà |
|--------------------------|---------------|-----------------------|
| max_rule_length | numero intero | 2–20. |
| min_confidence | numero | 0.0–1.0. |
| min_support | numero | 0.0–1.0. |
| use_transactional_data | indicatore | |

Oracle MDL (Lunghezza descrizione minima)

Per i nodi di tipo oramdlnode non sono definite proprietà specifiche. Vedere le proprietà comuni di Oracle all'inizio di questa sezione.

Importanza attributo Oracle (AI)

Le seguenti proprietà sono disponibili per i nodi di tipo oraainode.

Tabella 191. proprietà oraainode.

| proprietà oraainode | Valori | Descrizione proprietà |
|---------------------|--|---|
| custom_fields | indicatore | Se vera, consente di specificare i campi obiettivo, di input e di altro tipo per il nodo corrente. Se falsa, vengono utilizzate le impostazioni correnti di un nodo Tipo a monte. |
| selection_mode | ImportanceLevel ImportanceValue TopN | |
| select_important | indicatore | Quando selection_mode è impostata su ImportanceLevel, specifica se selezionare i campi importanti. |
| important_label | stringa | Specifica l'etichetta per la classificazione "importante". |
| select_marginal | indicatore | Quando selection_mode è impostata su ImportanceLevel, specifica se selezionare i campi marginali. |
| marginal_label | stringa | Specifica l'etichetta per la classificazione "marginale". |
| important_above | numero | 0.0–1.0. |
| select_unimportant | indicatore | Quando selection_mode è impostata su ImportanceLevel, specifica se selezionare i campi non importanti. |
| unimportant_label | stringa | Specifica l'etichetta per la classificazione "non importante". |
| unimportant_below | numero | 0.0–1.0. |
| importance_value | numero | Quando selection_mode è impostata su ImportanceValue, specifica il valore di interruzione da utilizzare. Accetta i valori compresi tra 0 e 100. |
| top_n | numero | Quando selection_mode è impostata su TopN, specifica il valore di interruzione da utilizzare. Accetta i valori compresi tra 0 e 1000. |

Proprietà dei nugget del modello Oracle

Le seguenti proprietà sono relative ai nugget del modello creati mediante i modelli Oracle.

Naive Bayes Oracle

Per i nodi di tipo `applyoranbnode` non sono definite proprietà specifiche.

Bayes adattivo Oracle

Per i nodi di tipo `applyoraabnode` non sono definite proprietà specifiche.

SVM Oracle

Non vi sono proprietà specifiche definite per i nodi di tipo `applyorasvmnode`.

struttura ad albero delle decisioni Oracle

Le seguenti proprietà sono disponibili per i nodi di tipo `applyoradecisiontreenode`.

Tabella 192. proprietà `applyoradecisiontreenode`

| Proprietà <code>applyoradecisiontreenode</code> | Valori | Descrizione proprietà |
|---|-------------------|-----------------------|
| <code>use_costs</code> | <i>indicatore</i> | |
| <code>display_rule_ids</code> | <i>indicatore</i> | |

O-Cluster Oracle

Non vi sono proprietà specifiche definite per i nodi di tipo `applyoraoclusternode`.

Medie K Oracle

Non vi sono proprietà specifiche definite per i nodi di tipo `applyorakmeansnode`.

NMF Oracle

Per i nodi di tipo `applyoranmfnode` è disponibile la seguente proprietà:

Tabella 193. proprietà `applyoranmfnode`

| proprietà <code>applyoranmfnode</code> | Valori | Descrizione proprietà |
|--|-------------------|-----------------------|
| <code>display_all_features</code> | <i>indicatore</i> | |

Apriori Oracle

Questo nugget del modello non può essere applicato negli script.

MDL Oracle

Questo nugget del modello non può essere applicato negli script.

Proprietà dei nodi Modelli IBM DB2

Proprietà dei nodi Modelli IBM DB2

Le seguenti proprietà sono comuni ai nodi Modelli database IBM InfoSphere Warehouse (ISW).

Tabella 194. Proprietà comuni dei nodi ISW.

| Proprietà comuni dei nodi ISW | Valori | Descrizione proprietà |
|-------------------------------|------------------------|---------------------------------------|
| inputs | <i>Elenco di campi</i> | |
| datasource | | |
| username | | |
| password | | |
| epassword | | |
| enable_power_options | <i>indicatore</i> | |
| power_options_max_memory | <i>numero intero</i> | Il valore deve essere maggiore di 32. |
| power_options_cmdline | <i>stringa</i> | |
| mining_data_custom_sql | <i>stringa</i> | |
| logical_data_custom_sql | <i>stringa</i> | |
| mining_settings_custom_sql | | |

Struttura ad albero delle decisioni ISW

Per i nodi di tipo db2imtreenode sono disponibili le seguenti proprietà.

Tabella 195. proprietà db2imtreenode.

| Proprietà db2imtreenode | Valori | Descrizione proprietà |
|----------------------------|----------------------|---|
| obiettivo | <i>campo</i> | |
| perform_test_run | <i>indicatore</i> | |
| use_max_tree_depth | <i>indicatore</i> | |
| max_tree_depth | <i>numero intero</i> | Valore maggiore di 0. |
| use_maximum_purity | <i>indicatore</i> | |
| maximum_purity | <i>numero</i> | Numero compreso tra 0 e 100. |
| use_minimum_internal_cases | <i>indicatore</i> | |
| minimum_internal_cases | <i>numero intero</i> | Valore maggiore di 1. |
| use_costs | <i>indicatore</i> | |
| costs | <i>strutturata</i> | Proprietà strutturata nel formato: [[drugA drugB 1.5] [drugA drugC 2.1]], dove gli argomenti racchiusi tra [] rappresentano i costi previsti effettivi. |

Associazione ISW

Le seguenti proprietà sono disponibili per i nodi di tipo db2imassocnode.

Tabella 196. proprietà db2imassocnode.

| Proprietà db2imassocnode | Valori | Descrizione proprietà |
|--------------------------|-------------------|-----------------------|
| use_transactional_data | <i>indicatore</i> | |

Tabella 196. proprietà db2imassocnode (Continua).

| Proprietà db2imassocnode | Valori | Descrizione proprietà |
|-----------------------------|-------------------------|--|
| id_field | campo | |
| content_field | campo | |
| data_table_layout | basic limited_length | |
| max_rule_size | numero intero | Il valore deve essere maggiore di 2. |
| min_rule_support | numero | 0-100% |
| min_rule_confidence | numero | 0-100% |
| use_item_constraints | indicatore | |
| item_constraints_type | Include Exclude | |
| use_taxonomy | indicatore | |
| taxonomy_table_name | stringa | Nome della tabella DB2 in cui archiviare i dettagli relativi alla tassonomia. |
| taxonomy_child_column_name | stringa | Nome della colonna figlio nella tabella di tassonomia. Tale colonna contiene i nomi di elemento o i nomi di categoria. |
| taxonomy_parent_column_name | stringa | Nome della colonna padre nella tabella di tassonomia. Tale colonna contiene i nomi di categoria. |
| load_taxonomy_to_table | indicatore | Stabilisce se le informazioni di tassonomia memorizzate in IBM SPSS Modeler devono essere caricate nella tabella di tassonomia nella fase di creazione del modello. Si noti che se già esiste una tabella di tassonomia, tale tabella verrà eliminata. Le informazioni relative alla tassonomia vengono archiviate con il nodo di creazione modello e si possono modificare mediante i pulsanti Modifica categorie e Modifica tassonomia . |

Sequenza ISW

Le seguenti proprietà sono disponibili per i nodi di tipo db2imsequencenode.

Tabella 197. proprietà db2imsequencenode.

| proprietà db2imsequencenode | Valori | Descrizione proprietà |
|-----------------------------|--------------------|---|
| id_field | campo | |
| group_field | campo | |
| content_field | campo | |
| max_rule_size | numero intero | Il valore deve essere maggiore di 2. |
| min_rule_support | numero | 0-100% |
| min_rule_confidence | numero | 0-100% |
| use_item_constraints | indicatore | |
| item_constraints_type | Include Exclude | |
| use_taxonomy | indicatore | |
| taxonomy_table_name | stringa | Nome della tabella DB2 in cui archiviare i dettagli relativi alla tassonomia. |

Tabella 197. proprietà db2imsequencenode (Continua).

| proprietà db2imsequencenode | Valori | Descrizione proprietà |
|-----------------------------|------------|--|
| taxonomy_child_column_name | stringa | Nome della colonna figlio nella tabella di tassonomia. Tale colonna contiene i nomi di elemento o i nomi di categoria. |
| taxonomy_parent_column_name | stringa | Nome della colonna padre nella tabella di tassonomia. Tale colonna contiene i nomi di categoria. |
| load_taxonomy_to_table | indicatore | Stabilisce se le informazioni di tassonomia memorizzate in IBM SPSS Modeler devono essere caricate nella tabella di tassonomia nella fase di creazione del modello. Si noti che se già esiste una tabella di tassonomia, tale tabella verrà eliminata. Le informazioni relative alla tassonomia vengono archiviate con il nodo di creazione modello e si possono modificare mediante i pulsanti Modifica categorie e Modifica tassonomia . |

Regressione ISW

Le seguenti proprietà sono disponibili per i nodi di tipo db2imregnode.

Tabella 198. proprietà db2imregnode.

| Proprietà db2imregnode | Valori | Descrizione proprietà |
|-----------------------------------|--|---|
| obiettivo | campo | |
| regression_method | transform linear polynomial rbf | Vedere la tabella successiva per le proprietà che sono valide solo se regression_method è impostato su rbf. |
| perform_test_run | campo | |
| limit_rsquared_value | indicatore | |
| max_rsquared_value | numero | Valore compreso tra 0,0 e 1,0. |
| use_execution_time_limit | indicatore | |
| execution_time_limit_mins | numero intero | Valore maggiore di 0. |
| use_max_degree_polynomial | indicatore | |
| max_degree_polynomial | numero intero | |
| use_intercept | indicatore | |
| use_auto_feature_selection_method | indicatore | |
| auto_feature_selection_method | normal adjusted | |
| use_min_significance_level | indicatore | |
| min_significance_level | numero | |
| use_min_significance_level | indicatore | |

Le seguenti proprietà sono valide solo se regression_method è impostato su rbf.

Tabella 199. Le proprietà db2imregnode se regression_method è impostato su rbf-services.

| Proprietà db2imregnode | Valori | Descrizione proprietà |
|------------------------|------------|--|
| use_output_sample_size | indicatore | Se vera, imposta automaticamente il valore di default. |

Tabella 199. Le proprietà db2imregnode se regression_method è impostato su rbf-services (Continua).

| | | |
|-----------------------|---------------|--|
| output_sample_size | numero intero | L'impostazione di default è 2. Il minimo è 1. |
| use_input_sample_size | indicatore | Se vera, imposta automaticamente il valore di default. |
| input_sample_size | numero intero | L'impostazione di default è 2. Il minimo è 1. |
| use_max_num_centers | indicatore | Se vera, imposta automaticamente il valore di default. |
| max_num_centers | numero intero | L'impostazione di default è 20. Il minimo è 1. |
| use_min_region_size | indicatore | Se vera, imposta automaticamente il valore di default. |
| min_region_size | numero intero | L'impostazione di default è 15. Il minimo è 1. |
| use_max_data_passes | indicatore | Se vera, imposta automaticamente il valore di default. |
| max_data_passes | numero intero | L'impostazione di default è 5. Il minimo è 2. |
| use_min_data_passes | indicatore | Se vera, imposta automaticamente il valore di default. |
| min_data_passes | numero intero | L'impostazione di default è 5. Il minimo è 2. |

Raggruppamento tramite cluster ISW

Le seguenti proprietà sono disponibili per i nodi di tipo db2imclusternode.

Tabella 200. proprietà db2imclusternode.

| Proprietà db2imclusternode | Valori | Descrizione proprietà |
|----------------------------|---------------------------------|--|
| cluster_method | demographic kohonen birch | |
| kohonen_num_rows | numero intero | |
| kohonen_num_columns | numero intero | |
| kohonen_passes | numero intero | |
| use_num_passes_limit | indicatore | |
| use_num_clusters_limit | indicatore | |
| max_num_clusters | numero intero | Valore maggiore di 1. |
| birch_dist_measure | log_likelihood euclidean | L'impostazione di default è log_likelihood. |
| birch_num_cfleaves | numero intero | L'impostazione di default è 1000. |
| birch_num_refine_passes | numero intero | L'impostazione di default è 3; il valore minimo è 1. |
| use_execution_time_limit | indicatore | |
| execution_time_limit_mins | numero intero | Valore maggiore di 0. |
| min_data_percentage | numero | 0-100% |

Tabella 200. proprietà db2imclusternode (Continua).

| Proprietà db2imclusternode | Valori | Descrizione proprietà |
|----------------------------|-------------------|--------------------------------|
| use_similarity_threshold | <i>indicatore</i> | |
| similarity_threshold | <i>numero</i> | Valore compreso tra 0,0 e 1,0. |

Naive Bayes ISW

Le seguenti proprietà sono disponibili per i nodi di tipo db2imnbsnode.

Tabella 201. proprietà db2imnbnode.

| Proprietà db2imnbnode | Valori | Descrizione proprietà |
|-----------------------|--------------------|--|
| perform_test_run | <i>indicatore</i> | |
| probability_threshold | <i>numero</i> | L'impostazione di default è 0.001. Il valore minimo è 0; il valore massimo è 1,000 |
| use_costs | <i>indicatore</i> | |
| costs | <i>strutturata</i> | Proprietà strutturata nel formato: [[drugA drugB 1.5] [drugA drugC 2.1]], dove gli argomenti racchiusi tra [] rappresentano i costi previsti effettivi. |

Regressione logistica ISW

Le seguenti proprietà sono disponibili per i nodi di tipo db2imlognode.

Tabella 202. proprietà db2imlognode.

| Proprietà db2imlognode | Valori | Descrizione proprietà |
|------------------------|--------------------|--|
| perform_test_run | <i>indicatore</i> | |
| use_costs | <i>indicatore</i> | |
| costs | <i>strutturata</i> | Proprietà strutturata nel formato: [[drugA drugB 1.5] [drugA drugC 2.1]], dove gli argomenti racchiusi tra [] rappresentano i costi previsti effettivi. |

Serie temporali ISW

Nota: per questo nodo non viene utilizzato il parametro dei campi di input. Se questo parametro viene rilevato nello script, un messaggio di avviso segnala che il nodo dispone dei campi in entrata *time* e *targets*, ma non dispone di campi di input.

Le seguenti proprietà sono disponibili per i nodi di tipo db2imtimeseriesnode.

Tabella 203. proprietà db2imtimeseriesnode.

| Proprietà db2imtimeseriesnode | Valori | Descrizione proprietà |
|-------------------------------|------------------------|---------------------------------------|
| time | <i>campo</i> | Sono consentiti integer, time o date. |
| targets | <i>Elenco di campi</i> | |

Tabella 203. proprietà db2imtimeseriesnode (Continua).

| Proprietà db2imtimeseriesnode | Valori | Descrizione proprietà |
|-------------------------------|--|---|
| forecasting_algorithm | arima exponential_ smoothing seasonal_trend_ decomposition | |
| forecasting_end_time | auto integer date time | |
| use_records_all | booleano | Se falsa, è necessario impostare use_records_start e use_records_end. |
| use_records_start | numero intero / ora / data | Dipende dal tipo di campo time |
| use_records_end | numero intero / ora / data | Dipende dal tipo di campo time |
| interpolation_method | none linear exponential_splines cubic_splines | |

Proprietà dei nugget del modello IBM DB2

Le seguenti proprietà sono relative ai nugget del modello creati mediante i modelli IBM DB2 ISW.

Struttura ad albero delle decisioni ISW

Per i nodi di tipo applydb2imtreeode non sono definite proprietà specifiche.

Associazione ISW

Questo nugget del modello non può essere applicato negli script.

Sequenza ISW

Questo nugget del modello non può essere applicato negli script.

Regressione ISW

Per i nodi di tipo applydb2imregnode non sono definite proprietà specifiche.

Raggruppamento tramite cluster ISW

Per i nodi di tipo applydb2imclusternode non sono definite proprietà specifiche.

Naive Bayes ISW

Per i nodi di tipo applydb2imbnnode non sono definite proprietà specifiche.

Regressione logistica ISW

Per i nodi di tipo applydb2imlognode non sono definite proprietà specifiche.

Serie temporali ISW

Questo nugget del modello non può essere applicato negli script.

Proprietà dei nodi Modelli IBM Netezza Analytics

Proprietà dei nodi Modelli Netezza

Le seguenti proprietà sono comuni ai nodi Modelli database IBM Netezza.

Tabella 204. Proprietà comuni dei nodi Netezza.

| Proprietà comuni dei nodi Netezza | Valori | Descrizione proprietà |
|-----------------------------------|----------------------------|---|
| custom_fields | <i>indicatore</i> | Se vera, consente di specificare i campi obiettivo, di input e di altro tipo per il nodo corrente. Se falsa, vengono utilizzate le impostazioni correnti di un nodo Tipo a monte. |
| inputs | <i>[campo1 ... campoN]</i> | I campi di input o predittore utilizzati dal modello. |
| obiettivo | <i>campo</i> | Campo obiettivo (continuo o categoriale). |
| record_id | <i>campo</i> | Campo da utilizzare come identificatore univoco del record. |
| use_upstream_connection | <i>indicatore</i> | Se vera (default) usa i dettagli di connessione specificati in un nodo a monte. Non utilizzato se viene specificato <code>move_data_to_connection</code> . |
| move_data_connection | <i>indicatore</i> | Se vera, sposta i dati nel database specificato da connessione. Non utilizzato se viene specificato <code>use_upstream_connection</code> . |
| connection | <i>strutturata</i> | La stringa di connessione per il database Netezza in cui è archiviato il modello. Proprietà strutturata nel formato: <code>['odbc' '<dsn>' '<username>' '<psw>' '<catname>' '<conn_attribs>' [true false]]</code> dove: <dsn> è il nome dell'origine dati <username> e <psw> sono il nome utente e la password del database <catname> è il nome del catalogo <conn_attribs> sono gli attributi di connessione true false indica se la password è necessaria. |
| table_name | <i>stringa</i> | Nome della tabella di database in cui sarà archiviato il modello. |
| use_model_name | <i>indicatore</i> | Se vera, utilizza il nome specificato da <code>model_name</code> come nome del modello, in caso contrario il nome del modello viene creato dal sistema. |
| model_name | <i>stringa</i> | Nome personalizzato per il nuovo modello. |
| include_input_fields | <i>indicatore</i> | Se vera, passa tutti i campi di input a valle, in caso contrario passa solo <code>record_id</code> e i campi generati dal modello. |

Struttura ad albero delle decisioni di Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo `netezzadectreenode`.

Tabella 205. proprietà netezzadectreenode.

| Proprietà netezzadectreenode | Valori | Descrizione proprietà |
|------------------------------|---|---|
| impurity_measure | Entropy Gini | La misurazione dell'impurità utilizzata per valutare il punto migliore in cui suddividere la struttura ad albero. |
| max_tree_depth | <i>numero intero</i> | Numero massimo di livelli di cui una struttura ad albero può crescere. Il valore di default è 62 (il massimo possibile). |
| min_improvement_splits | <i>numero</i> | Miglioramento minimo in impurità perché si verifichi una suddivisione. L'impostazione di default è 0.01. |
| min_instances_split | <i>numero intero</i> | Numero minimo di record ancora da suddividere perché sia possibile effettuare una suddivisione. Il valore di default è 2 (il minimo possibile). |
| weights | <i>strutturata</i> | Ponderazioni relative per le classi. Proprietà strutturata nel formato: set :netezza_dectree.weights = [[drugA 0.3][drugB 0.6]] L'impostazione di default è il peso 1 per tutte le classi. |
| pruning_measure | Acc wAcc | L'impostazione di default è Acc (accuracy). In alternativa wAcc (precisione ponderata) tiene conto dei pesi delle classi durante l'applicazione del taglio. |
| prune_tree_options | allTrainingData partitionTrainingData useOtherTable | L'impostazione di default da utilizzare è allTrainingData per valutare la precisione del modello. Utilizzare partitionTrainingData per specificare una percentuale dei dati di addestramento da utilizzare o useOtherTable per utilizzare un insieme di dati di addestramento di una tabella di database specifica. |
| perc_training_data | <i>numero</i> | Se prune_tree_options è impostato su partitionTrainingData, specifica la percentuale di dati da utilizzare per l'addestramento. |
| prune_seed | <i>numero intero</i> | Seme random da utilizzare per replicare i risultati delle analisi quando prune_tree_options è impostato su partitionTrainingData; il valore di default è 1. |
| pruning_table | <i>stringa</i> | Nome della tabella di un insieme di dati di taglio separato per la stima della precisione del modello. |
| compute_probabilities | <i>indicatore</i> | Se vera, produce un campo livello di confidenza (probabilità), nonché un campo di previsione. |

Medie K Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo `netezzakmeansnode`.

Tabella 206. proprietà `netezzakmeansnode`.

| Proprietà <code>netezzakmeansnode</code> | Valori | Descrizione proprietà |
|--|---|---|
| <code>distance_measure</code> | Euclidean Manhattan Canberra maximum | Metodo da utilizzare per misurare la distanza fra punti dati. |
| <code>num_clusters</code> | <i>numero intero</i> | Numero di cluster da creare; l'impostazione di default è 3. |
| <code>max_iterations</code> | <i>numero intero</i> | Numero di iterazioni dell'algoritmo dopo cui interrompere l'addestramento del modello; l'impostazione di default è 5. |
| <code>rand_seed</code> | <i>numero intero</i> | Seme random da utilizzare per replicare i risultati delle analisi; l'impostazione di default è 12345. |

Rete di Bayes Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo `netezzabayesnode`.

Tabella 207. proprietà `netezzabayesnode`.

| Proprietà <code>netezzabayesnode</code> | Valori | Descrizione proprietà |
|---|-----------------------------------|---|
| <code>base_index</code> | <i>numero intero</i> | Identificatore numerico assegnato al primo campo di input per la gestione interna; l'impostazione di default è 777. |
| <code>sample_size</code> | <i>numero intero</i> | Dimensione del campione da prendere se il numero di attributi è molto elevato; l'impostazione di default è 10.000. |
| <code>display_additional_information</code> | <i>indicatore</i> | Se vera, visualizza ulteriori informazioni sull'avanzamento in una finestra di dialogo. |
| <code>type_of_prediction</code> | best neighbors nn-neighbors | Tipo di algoritmo di previsione da utilizzare: ottima (elemento adiacente con maggiore correlazione), elementi adiacenti (previsione ponderata degli elementi adiacenti) o elementi adiacenti NN (elementi adiacenti non null). |

Naive Bayes Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo `netezzanaiveyesnode`.

Tabella 208. proprietà `netezzanaiveyesnode`.

| Proprietà <code>netezzanaiveyesnode</code> | Valori | Descrizione proprietà |
|--|-------------------|---|
| <code>compute_probabilities</code> | <i>indicatore</i> | Se vera, produce un campo livello di confidenza (probabilità), nonché un campo di previsione. |
| <code>use_m_estimation</code> | <i>indicatore</i> | Se vera, utilizza la tecnica della stima m per evitare le probabilità zero durante la stima. |

KNN Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo `netezzaknnnode`.

Tabella 209. proprietà netezzaknnnode.

| Proprietà netezzaknnnode | Valori | Descrizione proprietà |
|--------------------------|---|---|
| weights | strutturata | Proprietà strutturata utilizzata per assegnare i pesi alle singole classi. Esempio: set :netezzaknnnode.weights = [[drugA 0.3][drugB 0.6]] |
| distance_measure | Euclidean Manhattan Canberra Maximum | Metodo da utilizzare per misurare la distanza fra punti dati. |
| num_nearest_neighbors | numero intero | Numero di elementi adiacenti più vicini per un caso particolare; l'impostazione di default è 3. |
| standardize_measurements | indicatore | Se vera, standardizza le misurazioni per i campi di input continui prima di calcolare i valori delle distanze. |
| use_coresets | indicatore | Se vera, utilizza il campionamento degli insiemi centrali per velocizzare il calcolo per insiemi di dati di grandi dimensioni. |

Raggruppamento cluster divisivo Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo netezzadivclusternode.

Tabella 210. proprietà netezzadivclusternode.

| Proprietà netezzadivclusternode | Valori | Descrizione proprietà |
|---------------------------------|---|---|
| distance_measure | Euclidean Manhattan Canberra Maximum | Metodo da utilizzare per misurare la distanza fra punti dati. |
| max_iterations | numero intero | Numero massimo di iterazioni dell'algoritmo da eseguire prima di interrompere l'addestramento del modello; l'impostazione di default è 5. |
| max_tree_depth | numero intero | Numero massimo di livelli in cui possono essere suddivisi gli insiemi di dati; l'impostazione di default è 3. |
| rand_seed | numero intero | Seme random, utilizzato per replicare le analisi; l'impostazione di default è 12345. |
| min_instances_split | numero intero | Numero minimo di record che possono essere suddivisi; l'impostazione di default è 5. |
| level | numero intero | Livello di gerarchia a cui deve essere calcolato il punteggio dei record; l'impostazione di default è -1. |

PCA Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo netezzapcanode.

Tabella 211. proprietà netezzapcanode.

| Proprietà netezzapcanode | Valori | Descrizione proprietà |
|--------------------------|------------|--|
| center_data | indicatore | Se vera (default), esegue la centratura dei dati (nota anche come "sottrazione delle medie") prima dell'analisi. |

Tabella 211. proprietà netezzapcanode (Continua).

| Proprietà netezzapcanode | Valori | Descrizione proprietà |
|--------------------------|----------------------|---|
| perform_data_scaling | <i>indicatore</i> | Se vera, esegue il ridimensionamento dei dati prima dell'analisi. Questa operazione può ridurre l'arbitrarietà dell'analisi quando vengono misurate diverse variabili in diverse unità. |
| force_eigenolve | <i>indicatore</i> | Se vera, utilizza un metodo meno accurato ma più veloce per trovare le componenti principali. |
| pc_number | <i>numero intero</i> | Numero di componenti principali a cui deve essere ridotto l'insieme di dati; l'impostazione di default è 1. |

Struttura ad albero di regressione Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo netezzaregtreenode.

Tabella 212. proprietà netezzaregtreenode.

| Proprietà netezzaregtreenode | Valori | Descrizione proprietà |
|------------------------------|---|---|
| max_tree_depth | <i>numero intero</i> | Numero massimo di livelli a cui può espandersi la struttura ad albero al di sotto del nodo root; l'impostazione di default è 10. |
| split_evaluation_measure | Variance | Misurazione dell'impurità delle classi, utilizzata per valutare il punto migliore in cui suddividere la struttura ad albero; l'impostazione di default (e al momento l'unica opzione) è Variance. |
| min_improvement_splits | <i>numero</i> | Quantità minima di riduzione dell'impurità prima che venga creata una nuova suddivisione nella struttura ad albero. |
| min_instances_split | <i>numero intero</i> | Numero minimo di record che possono essere suddivisi. |
| pruning_measure | mse r2 pearson spearman | Metodo da utilizzare per il taglio. |
| prune_tree_options | allTrainingData partitionTrainingData useOtherTable | L'impostazione di default da utilizzare è allTrainingData per valutare la precisione del modello. Utilizzare partitionTrainingData è per specificare una percentuale dei dati di addestramento da utilizzare o useOtherTable per utilizzare un insieme di dati di addestramento di una tabella di database specifica. |
| perc_training_data | <i>numero</i> | Se prune_tree_options è impostato su PercTrainingData, specifica la percentuale di dati da utilizzare per l'addestramento. |

Tabella 212. proprietà *netezzaregtreenode* (Continua).

| Proprietà <i>netezzaregtreenode</i> | Valori | Descrizione proprietà |
|-------------------------------------|----------------------|--|
| <i>prune_seed</i> | <i>numero intero</i> | Seme random da utilizzare per replicare i risultati delle analisi quando <i>prune_tree_options</i> è impostato su <i>PercTrainingData</i> ; l'impostazione di default è 1. |
| <i>pruning_table</i> | <i>stringa</i> | Nome della tabella di un insieme di dati di taglio separato per la stima della precisione del modello. |
| <i>compute_probabilities</i> | <i>indicatore</i> | Se vera, specifica che le varianze delle classi assegnate devono essere incluse nell'output. |

Regressione lineare Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo' *netezzalineressionnode*.

Tabella 213. proprietà *netezzalineressionnode*.

| Proprietà <i>netezzalineressionnode</i> | Valori | Descrizione proprietà |
|---|-------------------|---|
| <i>use_svd</i> | <i>indicatore</i> | Se vera, utilizza la matrice Decomposizione ai valori singolari, anziché la matrice originale, per una maggiore velocità e precisione numerica. |
| <i>include_intercept</i> | <i>indicatore</i> | Se vera (default), aumenta la precisione generale della soluzione. |
| <i>calculate_model_diagnostics</i> | <i>indicatore</i> | Se vera, calcola la diagnostica del modello. |

Serie temporali Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo *netezzatimeseriesnode*.

Tabella 214. proprietà *netezzatimeseriesnode*.

| Proprietà <i>netezzatimeseriesnode</i> | Valori | Descrizione proprietà |
|--|--------------|--|
| <i>time_points</i> | <i>campo</i> | Il campo di input contenente i valori di data o ora per le serie temporali. |
| <i>time_series_ids</i> | <i>campo</i> | Campo di input contenente gli ID delle serie temporali; utilizzarlo se l'input contiene più di una serie temporali. |
| <i>model_table</i> | <i>campo</i> | Nome della tabella di database in cui sarà archiviato il modello di serie temporali Netezza. |
| <i>description_table</i> | <i>campo</i> | Nome della tabella di input che contiene i nomi e le descrizioni delle serie temporali. |
| <i>seasonal_adjustment_table</i> | <i>campo</i> | Nome della tabella di output in cui i valori corretti per stagionalità calcolati dagli algoritmi di decomposizione a tendenza stagionale o a livellamento esponenziale verranno memorizzati. |

Tabella 214. proprietà netezzatimeseriesnode (Continua).

| Proprietà netezzatimeseriesnode | Valori | Descrizione proprietà |
|---------------------------------|---|---|
| algorithm_name | SpectralAnalysis o spectral ExponentialSmoothing o esmoothing ARIMA SeasonalTrendDecomposition o std | Un algoritmo da utilizzare per la modellazione di serie temporali. |
| trend_name | N A DA M DM | Tipo di tendenza per il livellamento esponenziale: N - nessuno A - additivo DA - additivo smorzato M - moltiplicativo DM - moltiplicativo smorzato |
| seasonality_type | N A M | Tipo di stagionalità per il livellamento esponenziale: N - nessuno A - additivo M - moltiplicativo |
| interpolation_method | linear cubicspline exponentialspline | Metodo di interpolazione da utilizzare. |
| timerange_setting | SD SP | Impostazione dell'intervallo di tempo da utilizzare: SD - determinato dal sistema (utilizza la gamma completa dei dati di serie temporali) SP - specificato dall'utente tramite earliest_time e latest_time |
| earliest_time | <i>integer</i> | Valori di inizio e fine, se timerange_setting è SP. |
| latest_time | <i>data</i> <i>ora</i> <i>timestamp</i> | Il formato deve seguire il valore time_points. Ad esempio, se il campo time_points contiene una data, questo valore deve essere una data. Esempio: set NZ_DT1.timerange_setting = 'SP' set NZ_DT1.earliest_time = '1921-01-01' set NZ_DT1.latest_time = '2121-01-01' |

Tabella 214. proprietà netezzatimeseriesnode (Continua).

| Proprietà netezzatimeseriesnode | Valori | Descrizione proprietà |
|---------------------------------|----------------------|--|
| arima_setting | SD SP | <p>Impostazione per l'algoritmo ARIMA (utilizzato solo se algorithm_name è impostato su ARIMA):</p> <p>SD - determinato dal sistema SP - specificato dall'utente</p> <p>Se arima_setting = SP, utilizzare i seguenti parametri per impostare i valori stagionali e non stagionali. Esempio (non solo stagionali):</p> <pre>set NZ_DT1.algorithm_name = 'arima' set NZ_DT1.arima_setting = 'SP' set NZ_DT1.p_symbol = 'lesseq' set NZ_DT1.p = '4' set NZ_DT1.d_symbol = 'lesseq' set NZ_DT1.d = '2' set NZ_DT1.q_symbol = 'lesseq' set NZ_DT1.q = '4'</pre> |
| p_symbol | less eq lesseq | <p>ARIMA - operatore per i parametri p, d, q, sp, sd, e sq:</p> <p>less - minore di eq - uguale a lesseq - minore di o uguale a</p> |
| d_symbol | | |
| q_symbol | | |
| sp_symbol | | |
| sd_symbol | | |
| sq_symbol | | |
| p | numero intero | ARIMA - gradi non stagionali di autocorrelazione. |
| q | numero intero | ARIMA - valore di derivazione non stagionale. |
| d | numero intero | ARIMA - numero non stagionale di ordini di media mobile nel modello. |
| sp | numero intero | ARIMA - gradi stagionali di autocorrelazione. |
| sq | numero intero | ARIMA - valore di derivazione stagionale. |
| sd | numero intero | ARIMA - numero stagionale di ordini di media mobile nel modello. |
| advanced_setting | SD SP | <p>Determina il modo in cui le impostazioni avanzate devono essere gestite:</p> <p>SD - determinato dal sistema SP - specificato dall'utente tramite period, units_period e forecast_setting.</p> <p>Esempio:</p> <pre>set NZ_DT1.advanced_setting = 'SP' set NZ_DT1.period = 5 set NZ_DT1.units_period = 'd'</pre> |

Tabella 214. proprietà *netezzatimeseriesnode* (Continua).

| Proprietà <i>netezzatimeseriesnode</i> | Valori | Descrizione proprietà |
|--|---|---|
| period | <i>numero intero</i> | Lunghezza del ciclo stagionale, specificato insieme a <i>units_period</i> . Non valido per l'analisi spettrale. |
| units_period | ms s min h d wk q y | Le unità in cui viene espresso <i>period</i> : ms - millisecondi s - secondi min - minuti h - ore d - giorni wk - settimane q - trimestri y - anni Ad esempio, per una serie temporale settimanale utilizzare 1 per <i>period</i> e wk per <i>units_period</i> . |
| forecast_setting | forecasthorizon forecasttimes | Specifica il modo in cui le previsioni devono essere eseguite. |
| forecast_horizon | <i>integer</i> <i>data</i> <i>ora</i> <i>timestamp</i> | Se <i>forecast_setting</i> = <i>forecasthorizon</i> , specifica il valore del punto finale per la previsione. Il formato deve seguire il valore <i>time_points</i> . Ad esempio, se il campo <i>time_points</i> contiene una data, questo valore deve essere una data. |
| forecast_times | <i>integer</i> <i>data</i> <i>ora</i> <i>timestamp</i> | Se <i>forecast_setting</i> = <i>forecasttimes</i> , specifica i valori da utilizzare per effettuare le previsioni. Il formato deve seguire il valore <i>time_points</i> . Ad esempio, se il campo <i>time_points</i> contiene una data, questo valore deve essere una data. |
| include_history | <i>indicatore</i> | Indica se i valori storici devono essere inclusi nell'output. |
| include_interpolated_values | <i>indicatore</i> | Indica se i valori interpolari devono essere inclusi nell'output. Non valido se <i>include_history</i> è false. |

Lineare generalizzato Netezza

Le seguenti proprietà sono disponibili per i nodi di tipo *netezzaglmnode*.

Tabella 215. proprietà netezzaglmnode.

| Proprietà netezzaglmnode | Valori | Descrizione proprietà |
|--------------------------|---|---|
| dist_family | bernoulli gaussian poisson negativebinomial wald gamma | Tipo di distribuzione; l'impostazione di default è bernoulli. |
| dist_params | numero | Il valore del parametro di distribuzione da utilizzare. Applicabile solo se distribution è Negativebinomial. |
| trials | numero intero | Applicabile solo se distribution è Binomial. la risposta obiettivo rappresenta il numero di eventi di un insieme di prove, il campo obiettivo contiene il numero di eventi e il campo trials contiene il numero di prove. |
| model_table | campo | Nome della tabella di database in cui sarà archiviato il modello lineare generalizzato Netezza. |
| maxit | numero intero | Numero massimo di iterazioni che possono essere eseguite dall'algoritmo; il valore di default è 20. |
| eps | numero | Il valore di errore massimo (in notazione scientifica) raggiunto il quale l'algoritmo deve interrompere la ricerca del modello più adatto. Il valore di default è -3, vale a dire 1E-3 oppure 0,001. |
| tol | numero | Il valore (in notazione scientifica) sotto il quale gli errori vengono trattati come se avessero valore zero. Il valore di default è -7, vale a dire che i valori inferiori a 1E-7 (o 0,0000001) sono considerati insignificanti. |

Tabella 215. proprietà netezzaglmode (Continua).

| Proprietà netezzaglmode | Valori | Descrizione proprietà |
|-------------------------|--|---|
| link_func | identità inverse invnegative invsquare sqrt power oddspower log clog loglog cloglog logit probit gaussit cauchit canbinom cangeom cannegbinom | Funzione di collegamento da utilizzare; il valore di default è logit. |
| link_params | numero | Il valore del parametro della funzione di collegamento da utilizzare. Applicabile solo se link_function è power o oddspower. |
| interaction | [[[<i>nomicol1</i>],[<i>livelli1</i>]], [[<i>nomicol2</i>],[<i>livelli2</i>]], ...,[<i>nomicolN</i>],[<i>livelliN</i>]], | Specifica le interazioni tra i campi. <i>nomicol</i> è un elenco di campi di input e <i>livello</i> è sempre 0 per ogni campo. Esempio: [[["K", "BP", "Sex", "K"], [0,0,0,0]], [["Age", "Na"], [0,0]]] |
| intercept | indicatore | Se true, include l'intercettazione nel modello. |

Proprietà dei nugget del modello Netezza

Le seguenti proprietà sono comuni ai nugget del modello di database Netezza.

Tabella 216. Proprietà comuni dei nugget del modello Netezza

| Proprietà comuni dei nugget del modello Netezza | Valori | Descrizione proprietà |
|---|---------|---|
| connection | stringa | La stringa di connessione per il database Netezza in cui è archiviato il modello. |
| table_name | stringa | Nome della tabella di database in cui è archiviato il modello. |

Altre proprietà dei nugget del modello sono identiche a quelle del nodo Modelli corrispondente.

Di seguito sono indicati i nomi script dei nugget del modello.

Tabella 217. I nomi degli script dei nugget del modello Netezza

| Nugget del modello | Nome script |
|-------------------------------------|---------------------------|
| Struttura ad albero delle decisioni | appliednetezzadectreenode |

Tabella 217. I nomi degli script dei nugget del modello Netezza (Continua)

| Nugget del modello | Nome script |
|---------------------------------------|---------------------------------|
| Medie K | applynetezzakmeansnode |
| Rete di Bayes | applynetezزابayesnode |
| Naive Bayes | applynetezzanaivebayesnode |
| KNN | applynetezzaknnnode |
| Raggruppamento cluster divisivo | applynetez zadivclusternode |
| PCA | applynetezzapcanode |
| Struttura ad albero di regressione | applynetez zaregtreenode |
| Regressione lineare | applynetez zalineregressionnode |
| Serie temporali | applynetez zatimeseriesnode |
| Lineare generalizzato | applynetez zaglmnode |

Capitolo 16. Proprietà dei nodi Output

Le proprietà del nodo Output sono leggermente diverse da quelle di altri tipi di nodi. Anziché fare riferimento all'opzione di un nodo specifico, le proprietà dei nodi Output consentono di memorizzare un riferimento all'oggetto di output. Ciò risulta utile per recuperare un valore da una tabella e impostarlo come un parametro del flusso.

In questa sezione vengono illustrate le proprietà degli script disponibili per i nodi Output.

Proprietà analysisnode



Il nodo Analisi valuta la capacità dei modelli predittivi di generare previsioni accurate. I nodi Analisi eseguono diversi confronti tra i valori previsti e i valori effettivi per uno o più nugget del modello. Possono inoltre confrontare i modelli predittivi fra loro.

Esempio

```
node = stream.create("analysis", "My node")
# "Analysis" tab
node.setPropertyValue("coincidence", True)
node.setPropertyValue("performance", True)
node.setPropertyValue("confidence", True)
node.setPropertyValue("threshold", 75)
node.setPropertyValue("improve_accuracy", 3)
node.setPropertyValue("inc_user_measure", True)
# "Define User Measure..."
node.setPropertyValue("user_if", "@TARGET = @PREDICTED")
node.setPropertyValue("user_then", "101")
node.setPropertyValue("user_else", "1")
node.setPropertyValue("user_compute", ["Mean", "Sum"])
node.setPropertyValue("by_fields", ["Drug"])
# "Output" tab
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/output/analysis_out.html")
```

Tabella 218. *analysisnode* properties.

| analysisnode properties | Data type | Descrizione proprietà |
|-------------------------|--|--|
| output_mode | Screen File | Utilizzata per specificare la posizione di destinazione per l'output generato dal nodo Output. |
| use_output_name | <i>indicatore</i> | Specifica se viene utilizzato un nome di output personalizzato. |
| output_name | <i>stringa</i> | Se use_output_name è impostata su true, specifica il nome da utilizzare. |
| output_format | Text (.txt) HTML (.html) Output (.cou) | Utilizzata per specificare il tipo di output. |
| by_fields | <i>elenco</i> | |
| full_filename | <i>stringa</i> | Se si tratta di output su disco, di dati o HTML, rappresenta il nome del file di output. |

Tabella 218. *analysisnode properties (Continua).*

| analysisnode properties | Data type | Descrizione proprietà |
|--------------------------------|----------------------------|------------------------------|
| coincidence | <i>indicatore</i> | |
| performance | <i>indicatore</i> | |
| evaluation_binary | <i>indicatore</i> | |
| confidence | <i>indicatore</i> | |
| Soglia | <i>numero</i> | |
| improve_accuracy | <i>numero</i> | |
| inc_user_measure | <i>indicatore</i> | |
| user_if | <i>expr</i> | |
| user_then | <i>expr</i> | |
| user_else | <i>expr</i> | |
| user_compute | [Mean Sum Min Max SDev] | |

Proprietà dataauditnode



Il nodo Esplora offre una prima panoramica completa dei dati, incluse statistiche riassuntive, istogrammi e distribuzione per ciascun campo, nonché informazioni su valori anomali, mancanti ed estremi. I risultati vengono visualizzati in una matrice di semplice lettura che può essere ordinata e utilizzata per generare grafici a schermo intero e nodi di preparazione dei dati.

Esempio

```

filenode = stream.createAt("variablefile", "File", 100, 100)
filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node = stream.createAt("dataaudit", "My node", 196, 100)
stream.link(filenode, node)
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("fields", ["Age", "Na", "K"])
node.setPropertyValue("display_graphs", True)
node.setPropertyValue("basic_stats", True)
node.setPropertyValue("advanced_stats", True)
node.setPropertyValue("median_stats", False)
node.setPropertyValue("calculate", ["Count", "Breakdown"])
node.setPropertyValue("outlier_detection_method", "std")
node.setPropertyValue("outlier_detection_std_outlier", 1.0)
node.setPropertyValue("outlier_detection_std_extreme", 3.0)
node.setPropertyValue("output_mode", "Screen")

```

Tabella 219. *proprietà dataauditnode.*

| Proprietà dataauditnode | Data type | Descrizione proprietà |
|--------------------------------|------------------------------|--|
| custom_fields | <i>indicatore</i> | |
| fields | [<i>campo1 ... campoN</i>] | |
| overlay | <i>campo</i> | |
| display_graphs | <i>indicatore</i> | Utilizzato per attivare o disattivare la visualizzazione di grafici nella matrice di output. |
| basic_stats | <i>indicatore</i> | |

Tabella 219. proprietà dataauditnode (Continua).

| Proprietà dataauditnode | Data type | Descrizione proprietà |
|-------------------------------|---|--|
| advanced_stats | indicatore | |
| median_stats | indicatore | |
| calculate | Count Breakdown | Utilizzato per calcolare valori mancanti. Selezionare uno, entrambi o nessun metodo di calcolo. |
| outlier_detection_method | std iqr | Utilizzato per specificare il metodo di rilevamento dei valori anomali ed estremi. |
| outlier_detection_std_outlier | numero | Se outlier_detection_method è std, specifica il numero da utilizzare per definire i valori anormali. |
| outlier_detection_std_extreme | numero | Se outlier_detection_method è std, specifica il numero da utilizzare per definire i valori estremi. |
| outlier_detection_iqr_outlier | numero | Se outlier_detection_method è iqr, specifica il numero da utilizzare per definire i valori anormali. |
| outlier_detection_iqr_extreme | numero | Se outlier_detection_method è iqr, specifica il numero da utilizzare per definire i valori estremi. |
| use_output_name | indicatore | Specifica se viene utilizzato un nome di output personalizzato. |
| output_name | stringa | Se use_output_name è impostata su true, specifica il nome da utilizzare. |
| output_mode | Screen File | Utilizzata per specificare la posizione di destinazione per l'output generato dal nodo Output. |
| output_format | Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou) | Utilizzata per specificare il tipo di output. |
| paginate_output | indicatore | Quando output_format è HTML, l'output viene separato in pagine. |
| lines_per_page | numero | Se utilizzato con paginate_output, specifica le righe per pagina di output. |
| full_filename | stringa | |

Proprietà matrixnode



Il nodo Matrice crea una tabella che mostra le relazioni tra i campi. In genere viene utilizzato per mostrare le relazioni tra due campi simbolici, ma è possibile avvalersene anche per mostrare le relazioni tra campi flag o numerici.

Esempio

```

node = stream.create("matrix", "My node")
# "Settings" tab
node.setPropertyValue("fields", "Numerics")
node.setPropertyValue("row", "K")
node.setPropertyValue("column", "Na")
node.setPropertyValue("cell_contents", "Function")
node.setPropertyValue("function_field", "Age")
node.setPropertyValue("function", "Sum")
# "Appearance" tab
node.setPropertyValue("sort_mode", "Ascending")
node.setPropertyValue("highlight_top", 1)
node.setPropertyValue("highlight_bottom", 5)
node.setPropertyValue("display", ["Counts", "Expected", "Residuals"])
node.setPropertyValue("include_totals", True)
# "Output" tab
node.setPropertyValue("full_filename", "C:/output/matrix_output.html")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("paginate_output", True)
node.setPropertyValue("lines_per_page", 50)

```

Tabella 220. proprietà *matrixnode*.

| Proprietà <i>matrixnode</i> | Data type | Descrizione proprietà |
|-----------------------------|--|---|
| fields | Selected Flags Numerics | |
| row | <i>campo</i> | |
| column | <i>campo</i> | |
| include_missing_values | <i>indicatore</i> | Specifica se i valori mancanti definiti dall'utente (vuoti) e i valori mancanti di sistema (null) sono inclusi nell'output delle righe e delle colonne. |
| cell_contents | CrossTabs Function | |
| function_field | <i>stringa</i> | |
| function | Sum Mean Min Max SDev | |
| sort_mode | Unsorted Crescente Descending | |
| highlight_top | <i>numero</i> | Se diversa da zero, è vera. |
| highlight_bottom | <i>numero</i> | Se diversa da zero, è vera. |
| display | [Counts Expected Residuals RowPct ColumnPct TotalPct] | |
| include_totals | <i>indicatore</i> | |

Tabella 220. proprietà matrixnode (Continua).

| Proprietà matrixnode | Data type | Descrizione proprietà |
|----------------------|---|--|
| use_output_name | indicatore | Specifica se viene utilizzato un nome di output personalizzato. |
| output_name | stringa | Se use_output_name è impostata su true, specifica il nome da utilizzare. |
| output_mode | Screen File | Utilizzata per specificare la posizione di destinazione per l'output generato dal nodo Output. |
| output_format | Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou) | Utilizzata per specificare il tipo di output. I formati Formatted e Delimited possono utilizzare il modificatore transposed, che traspone righe e colonne nella tabella. |
| paginate_output | indicatore | Quando output_format è HTML, l'output viene separato in pagine. |
| lines_per_page | numero | Se utilizzato con paginate_output, specifica le righe per pagina di output. |
| full_filename | stringa | |

Proprietà meansnode



Il nodo Medie confronta le medie tra gruppi indipendenti o coppie di campi correlati per verificare se esiste una differenza significativa. Per esempio, è possibile confrontare le entrate medie prima e dopo il lancio di una promozione, oppure confrontare le entrate determinate da clienti che non hanno ricevuto la promozione con quelli che l'hanno ricevuta.

Esempio

```
node = stream.create("means", "My node")
node.setPropertyValue("means_mode", "BetweenFields")
node.setPropertyValue("paired_fields", [["OPEN_BAL", "CURR_BAL"]])
node.setPropertyValue("label_correlations", True)
node.setPropertyValue("output_view", "Advanced")
node.setPropertyValue("output_mode", "File")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/output/means_output.html")
```

Tabella 221. proprietà meansnode.

| Proprietà meansnode | Data type | Descrizione proprietà |
|---------------------|---|--|
| means_mode | BetweenGroups BetweenFields | Specifica il tipo di medie statistiche da eseguire sui dati. |
| test_fields | [campo1 ... campon] | Specifica il campo di verifica quando means_mode è impostato su BetweenGroups. |
| grouping_field | campo | Specifica il campo di raggruppamento. |
| paired_fields | [[campo1 campo2] [campo3 campo4] ...] | Specifica le coppie di campi da utilizzare quando means_mode è impostato su BetweenFields. |

Tabella 221. proprietà meansnode (Continua).

| Proprietà meansnode | Data type | Descrizione proprietà |
|--------------------------|---|---|
| label_correlations | <i>indicatore</i> | Specifica se le etichette di correlazione devono essere visualizzate nell'output. Questa impostazione si applica solo quando means_mode è impostato su BetweenFields. |
| correlation_mode | Probability Absolute | Specifica se etichettare le correlazioni per probabilità o valore assoluto. |
| weak_label | <i>stringa</i> | |
| medium_label | <i>stringa</i> | |
| strong_label | <i>stringa</i> | |
| weak_below_probability | <i>numero</i> | Quando correlation_mode è impostato su Probabilità, specifica il valore di interruzione per correlazioni deboli. Deve essere un valore compreso tra 0 e 1, ad esempio 0.90. |
| strong_above_probability | <i>numero</i> | Valore di interruzione per correlazioni forti. |
| weak_below_absolute | <i>numero</i> | Quando correlation_mode è impostato su Absolute, specifica il valore di interruzione per correlazioni deboli. Deve essere un valore compreso tra 0 e 1, ad esempio 0.90. |
| strong_above_absolute | <i>numero</i> | Valore di interruzione per correlazioni forti. |
| unimportant_label | <i>stringa</i> | |
| marginal_label | <i>stringa</i> | |
| important_label | <i>stringa</i> | |
| unimportant_below | <i>numero</i> | Valore di interruzione per importanza di campo bassa. Deve essere un valore compreso tra 0 e 1, ad esempio 0.90. |
| important_above | <i>numero</i> | |
| use_output_name | <i>indicatore</i> | Specifica se viene utilizzato un nome di output personalizzato. |
| output_name | <i>stringa</i> | Nome da utilizzare. |
| output_mode | Screen File | Specifica la posizione di destinazione per l'output generato dal nodo Output. |
| output_format | Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou) | Specifica il tipo di output. |
| full_filename | <i>stringa</i> | |

Tabella 221. proprietà meansnode (Continua).

| Proprietà meansnode | Data type | Descrizione proprietà |
|---------------------|--------------------|--|
| output_view | Simple Advanced | Specifica se l'output deve presentare la visualizzazione di base o avanzata. |

Proprietà reportnode



Il nodo Report crea report formattati che contengono sia testo fisso sia dati e altre espressioni derivate dai dati. Il formato del report viene specificato utilizzando modelli di testo per definire il testo fisso e costruzioni di output dei dati. È possibile fornire una formattazione personalizzata del testo utilizzando tag HTML nel modello e impostando apposite opzioni nella scheda Output. È possibile includere valori di dati e altro output condizionale utilizzando espressioni CLEM nel modello.

Esempio

```
node = stream.create("report", "My node")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/report_output.html")
node.setPropertyValue("lines_per_page", 50)
node.setPropertyValue("title", "Report node created by a script")
node.setPropertyValue("highlights", False)
```

Tabella 222. proprietà reportnode.

| Proprietà reportnode | Data type | Descrizione proprietà |
|----------------------|--|--|
| output_mode | Screen File | Utilizzata per specificare la posizione di destinazione per l'output generato dal nodo Output. |
| output_format | HTML (.html) Text (.txt) Output (.cou) | Utilizzata per specificare il tipo di output del file. |
| format | Auto Personalizza | Utilizzata per decidere se l'output viene formattato automaticamente o utilizzando l'HTML incluso nel modello. Per utilizzare la formattazione HTML nel modello, specificare Custom. |
| use_output_name | <i>indicatore</i> | Specifica se viene utilizzato un nome di output personalizzato. |
| output_name | <i>stringa</i> | Se use_output_name è impostata su true, specifica il nome da utilizzare. |
| text | <i>stringa</i> | |
| full_filename | <i>stringa</i> | |
| highlights | <i>indicatore</i> | |
| title | <i>stringa</i> | |
| lines_per_page | <i>numero</i> | |

Proprietà routputnode



Il nodo Output R consente di analizzare i dati ed i risultati del calcolo del punteggio del modello utilizzando il proprio script R personalizzato. L'output dell'analisi può essere grafico o di testo. L'output viene aggiunto alla scheda **Output** del riquadro dei manager; in alternativa, è possibile reindirizzare l'output in un file.

Tabella 223. Proprietà routputnode

| Proprietà routputnode | Data type | Descrizione proprietà |
|------------------------|------------------------------------|-----------------------|
| syntax | stringa | |
| convert_flags | StringsAndDoubles LogicalValues | |
| convert_datetime | indicatore | |
| convert_datetime_class | POSIXct POSIXlt | |
| convert_missing | indicatore | |
| output_name | Automatico Custom | |
| custom_name | stringa | |
| output_to | Screen File | |
| output_type | Graph Text | |
| full_filename | stringa | |
| graph_file_type | HTML COU | |
| text_file_type | HTML TEXT COU | |

Proprietà setglobalsnode



Il nodo Calcola globali analizza i dati e calcola i valori di riepilogo che possono essere utilizzati nelle espressioni CLEM. Per esempio, è possibile utilizzare questo nodo per calcolare le statistiche di un campo denominato *età* e utilizzare quindi la media globale dell'*età* nelle espressioni CLEM inserendo la funzione @GLOBAL_MEAN(età).

Esempio

```
node = stream.create("setglobals", "My node")
node.setKeyedPropertyValue("globals", "Na", ["Max", "Sum", "Mean"])
node.setKeyedPropertyValue("globals", "K", ["Max", "Sum", "Mean"])
node.setKeyedPropertyValue("globals", "Age", ["Max", "Sum", "Mean", "SDev"])
node.setPropertyValue("clear_first", False)
node.setPropertyValue("show_preview", True)
```


Tabella 224. proprietà setglobalsnode.

| Proprietà setglobalsnode | Data type | Descrizione proprietà |
|--------------------------|-------------------------|--|
| globals | [Sum Mean Min Max SDev] | Proprietà strutturata nella quale per fare riferimento ai campi da impostare è necessario utilizzare la sintassi seguente: node.setKeyedPropertyValue("globals", "Age", ["Max", "Sum", "Mean", "SDev"]) |
| clear_first | indicatore | |
| show_preview | indicatore | |

Proprietà simevalnode



Il nodo Valutazione della simulazione valuta un campo obiettivo previsto specificato e visualizza le informazioni di distribuzione e correlazione relative al campo obiettivo.

Tabella 225. Proprietà simevalnode.

| Proprietà simevalnode | Data type | Descrizione proprietà |
|------------------------|-----------------------------------|-----------------------|
| target | campo | |
| iteration | campo | |
| presorted_by_iteration | booleano | |
| max_iterations | number | |
| tornado_fields | [campo1...campoN] | |
| plot_pdf | booleano | |
| plot_cdf | booleano | |
| show_ref_mean | booleano | |
| show_ref_median | booleano | |
| show_ref_sigma | booleano | |
| num_ref_sigma | number | |
| show_ref_pct | booleano | |
| ref_pct_bottom | number | |
| ref_pct_top | number | |
| show_ref_custom | booleano | |
| ref_custom_values | [numero1...numeroN] | |
| category_values | Category Probabilities Both | |
| category_groups | Categories Iterations | |
| create_pct_table | booleano | |
| pct_table | Quartili Intervals Custom | |

Tabella 225. Proprietà simevalnode (Continua).

| Proprietà simevalnode | Data type | Descrizione proprietà |
|-----------------------|---------------------|-----------------------|
| pct_intervals_num | number | |
| pct_custom_values | [numero1...numeroN] | |

Proprietà simfitnode



Il nodo Adattamento simulazione esamina la distribuzione statistica dei dati in ciascun campo e genera (o aggiorna) un nodo Genera simulazione, con la migliore distribuzione di adattamento assegnata a ciascun campo. Il nodo Genera simulazione può essere quindi utilizzato per generare dati simulati.

Tabella 226. Proprietà simfitnode.

| Proprietà simfitnode | Data type | Descrizione proprietà |
|---------------------------|--------------------------------------|---|
| build | Node XMLExport Both | |
| use_source_node_name | booleano | |
| source_node_name | stringa | Il nome personalizzato del nodo di origine generato o aggiornato. |
| use_cases | All LimitFirstN | |
| use_case_limit | integer | |
| fit_criterion | AndersonDarling KolmogorovSmirnov | |
| num_bins | integer | |
| parameter_xml_filename | stringa | |
| generate_parameter_import | booleano | |

Proprietà statisticsnode



Il nodo Statistiche fornisce informazioni riassuntive di base su campi numerici. Calcola statistiche riassuntive per singoli campi e per correlazioni tra campi.

Esempio

```
node = stream.create("statistics", "My node")
# "Settings" tab
node.setPropertyValue("examine", ["Age", "BP", "Drug"])
node.setPropertyValue("statistics", ["Mean", "Sum", "SDev"])
node.setPropertyValue("correlate", ["BP", "Drug"])
# "Correlation Labels..." section
node.setPropertyValue("label_correlations", True)
node.setPropertyValue("weak_below_absolute", 0.25)
node.setPropertyValue("weak_label", "lower quartile")
node.setPropertyValue("strong_above_absolute", 0.75)
node.setPropertyValue("medium_label", "middle quartiles")
```

```

node.setPropertyValue("strong_label", "upper quartile")
# "Output" tab
node.setPropertyValue("full_filename", "c:/output/statistics_output.html")
node.setPropertyValue("output_format", "HTML")

```

Tabella 227. proprietà statisticsnode.

| Proprietà statisticsnode | Data type | Descrizione proprietà |
|--------------------------|---|---|
| use_output_name | <i>indicatore</i> | Specifica se viene utilizzato un nome di output personalizzato. |
| output_name | <i>stringa</i> | Se use_output_name è impostata su true, specifica il nome da utilizzare. |
| output_mode | Screen File | Utilizzata per specificare la posizione di destinazione per l'output generato dal nodo Output. |
| output_format | Text (.txt) HTML (.html) Output (.cou) | Utilizzata per specificare il tipo di output. |
| full_filename | <i>stringa</i> | |
| examine | <i>elenco</i> | |
| correlate | <i>elenco</i> | |
| statistics | [Count Mean Sum Min Max Range Variance SDev SErr Median Mode] | |
| correlation_mode | Probability Absolute | Specifica se etichettare le correlazioni per probabilità o valore assoluto. |
| label_correlations | <i>indicatore</i> | |
| weak_label | <i>stringa</i> | |
| medium_label | <i>stringa</i> | |
| strong_label | <i>stringa</i> | |
| weak_below_probability | <i>numero</i> | Quando correlation_mode è impostato su Probabilità, specifica il valore di interruzione per correlazioni deboli. Deve essere un valore compreso tra 0 e 1, ad esempio 0.90. |
| strong_above_probability | <i>numero</i> | Valore di interruzione per correlazioni forti. |
| weak_below_absolute | <i>numero</i> | Quando correlation_mode è impostato su Absolute, specifica il valore di interruzione per correlazioni deboli. Questo valore deve essere compreso 0 e 1, ad esempio 0.90. |
| strong_above_absolute | <i>numero</i> | Valore di interruzione per correlazioni forti. |

Proprietà statisticsoutputnode



Il nodo Output Statistics consente di chiamare una procedura IBM SPSS Statistics per analizzare i dati di IBM SPSS Modeler. È disponibile una vasta gamma di procedure analitiche di IBM SPSS Statistics. Questo nodo richiede una copia di IBM SPSS Statistics con regolare licenza.

Le proprietà di questo nodo sono descritte in “Proprietà statisticsoutputnode” a pagina 312.

Proprietà tablenode



Il nodo Tabella visualizza i dati in formato tabella, che è inoltre possibile scrivere su un file. Questa funzione è utile tutte le volte che si desidera controllare i valori dei dati o esportarli in un formato di facile lettura.

Esempio

```
node = stream.create("table", "My node")
node.setPropertyValue("highlight_expr", "Age > 30")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("transpose_data", True)
node.setPropertyValue("full_filename", "C:/output/table_output.htm")
node.setPropertyValue("paginate_output", True)
node.setPropertyValue("lines_per_page", 50)
```

Tabella 228. proprietà tablenode.

| Proprietà tablenode | Data type | Descrizione proprietà |
|---------------------|---|---|
| full_filename | stringa | Se si tratta di output su disco, di dati o HTML, rappresenta il nome del file di output. |
| use_output_name | indicatore | Specifica se viene utilizzato un nome di output personalizzato. |
| output_name | stringa | Se use_output_name è impostata su true, specifica il nome da utilizzare. |
| output_mode | Screen File | Utilizzata per specificare la posizione di destinazione per l'output generato dal nodo Output. |
| output_format | Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou) | Utilizzata per specificare il tipo di output. |
| transpose_data | indicatore | Traspone i dati prima dell'esportazione in modo che le righe rappresentino i campi e le colonne rappresentino i record. |
| paginate_output | indicatore | Quando output_format è HTML, l'output viene separato in pagine. |
| lines_per_page | numero | Se utilizzato con paginate_output, specifica le righe per pagina di output. |
| highlight_expr | stringa | |

Tabella 228. proprietà tablenode (Continua).

| Proprietà tablenode | Data type | Descrizione proprietà |
|---------------------|---|---|
| output | stringa | Proprietà di sola lettura che restituisce un riferimento all'ultima tabella creata dal nodo. |
| value_labels | [[Value LabelString] [Value LabelString] ...] | Utilizzata per specificare etichette per coppie di valori. |
| display_places | numero intero | Imposta il numero di decimali del campo per la visualizzazione (valida solo per campi con archiviazione di tipo Reale). Se viene specificato il valore -1, verrà utilizzata l'impostazione di default del flusso. |
| export_places | numero intero | Imposta il numero di decimali del campo per l'esportazione (valida solo per campi con archiviazione di tipo Reale). Se viene specificato il valore -1, verrà utilizzata l'impostazione di default del flusso. |
| decimal_separator | DEFAULT PERIOD COMMA | Imposta il separatore decimale per il campo (valido solo per campi con archiviazione di tipo Reale). |
| date_format | "DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY | Imposta il formato di data per il campo (valida solo per campi con archiviazione di tipoDATE o TIMESTAMP). |

Tabella 228. proprietà tablenode (Continua).

| Proprietà tablenode | Data type | Descrizione proprietà |
|---------------------|---|---|
| time_format | "HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM." "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S" | Imposta il formato di ora per il campo (valida solo per campi con archiviazione di tipo TIME o TIMESTAMP). |
| column_width | numero intero | Imposta la larghezza delle colonne per il campo. Se viene specificato il valore -1, la larghezza delle colonne verrà impostata su Auto. |
| justify | AUTO CENTER LEFT RIGHT | Imposta la giustificazione delle colonne per il campo. |

Proprietà transformnode



Il nodo Trasformazioni consente di selezionare e visualizzare in anteprima i risultati di trasformazioni prima di applicarli ai campi selezionati.

Esempio

```
node = stream.create("transform", "My node")
node.setPropertyValue("fields", ["AGE", "INCOME"])
node.setPropertyValue("formula", "Select")
node.setPropertyValue("formula_log_n", True)
node.setPropertyValue("formula_log_n_offset", 1)
```

Tabella 229. proprietà transformnode.

| Proprietà transformnode | Data type | Descrizione proprietà |
|-------------------------|---------------------|--|
| fields | [campo1... campon] | I campi da utilizzare nella trasformazione. |
| formula | All Select | Indica se devono essere calcolate tutte le trasformazioni o solo quelle selezionate. |
| formula_inverse | indicatore | Indica se deve essere utilizzata la trasformazione inversa. |
| formula_inverse_offset | number | Indica l'offset tra i dati da utilizzare per la formula. Se non è specificata dall'utente, è impostata su 0 per default. |

Tabella 229. proprietà transformnode (Continua).

| Proprietà transformnode | Data type | Descrizione proprietà |
|-------------------------|-------------------------------|--|
| formula_log_n | indicatore | Indica se deve essere utilizzata la trasformazione \log_n . |
| formula_log_n_offset | number | |
| formula_log_10 | indicatore | Indica se deve essere utilizzata la trasformazione \log_{10} . |
| formula_log_10_offset | number | |
| formula_exponential | indicatore | Indica se deve essere utilizzata la trasformazione esponenziale (e^x). |
| formula_square_root | indicatore | Indica se deve essere utilizzata la trasformazione radice quadrata. |
| use_output_name | indicatore | Specifica se viene utilizzato un nome di output personalizzato. |
| output_name | stringa | Se use_output_name è impostata su vero, specifica il nome da utilizzare. |
| output_mode | Screen File | Utilizzata per specificare la posizione di destinazione per l'output generato dal nodo Output. |
| output_format | HTML (.html) Output (.cou) | Utilizzata per specificare il tipo di output. |
| paginate_output | indicatore | Quando output_format è HTML, l'output viene separato in pagine. |
| lines_per_page | numero | Se utilizzato con paginate_output, specifica le righe per pagina di output. |
| full_filename | stringa | Indica il nome di file da utilizzare per l'output su file. |

Capitolo 17. Proprietà dei nodi di esportazione

Proprietà comuni dei nodi di esportazione

Le seguenti proprietà sono valide per tutti i nodi di esportazione:

Tabella 230. Proprietà comuni dei nodi di esportazione

| Proprietà | Valori | Descrizione proprietà |
|------------------------|------------------------|--|
| publish_path | stringa | Specificare il nome di base da utilizzare per i file immagine e dei parametri pubblicati. |
| publish_metadata | indicatore | Specifica se viene generato un file di metadati che descrive gli input e gli output dell'immagine e dei rispettivi modelli di dati. |
| publish_use_parameters | indicatore | Specifica se i parametri del flusso sono contenuti nel file *.par. |
| publish_parameters | elenco di stringhe | Specifica i parametri da includere. |
| execute_mode | export_data publish | Specifica se il nodo viene eseguito senza pubblicare il flusso o se il flusso viene pubblicato automaticamente quando si esegue il nodo. |

Proprietà asexport

L'esportazione di Analytic Server consente di eseguire un flusso su HDFS (Hadoop Distributed File System).

Esempio

```
node = stream.create("asexport", "My node")
node.setPropertyValue("data_source", "DrugIn")
node.setPropertyValue("export_mode", "overwrite")
```

Tabella 231. Proprietà asexport.

| Proprietà asexport | Data type | Descrizione proprietà |
|--------------------|-----------|---|
| data_source | stringa | Il nome dell'origine dati. |
| export_mode | stringa | Specifica se accodare i dati esportati all' origine dati esistente o sovrascrivere i dati esportati all'origine dati esistente. |

Proprietà del nodo di esportazione Cognos



Il nodo di esportazione IBM Cognos BI esporta i dati in un formato leggibile dai database Cognos BI.

Per questo nodo è necessario definire una connessione Cognos e una connessione ODBC.

Connessione Cognos

Le proprietà della connessione Cognos sono le seguenti.

Tabella 232. proprietà cognosexportnode

| proprietà cognosexportnode | Data type | Descrizione proprietà |
|----------------------------|---|--|
| cognos_connection | <code>["string", "flag", "string", "string", "string"]</code> | <p>Una proprietà elenco contenente i dettagli di connessione per il server Cognos. Il formato è: <code>["Cognos_server_URL", login_mode, "namespace", "username", "password"]</code></p> <p>dove: Cognos_server_URL è l'URL del server Cognos che contiene la sorgente. login_mode indica se viene utilizzato l'accesso anonimo e può essere true o false; se impostato su true, i seguenti campi devono essere impostati su "". namespace specifica il provider di protezione per l'autenticazione utilizzato per accedere al server. username e password sono i valori utilizzati per accedere al server Cognos.</p> <p>Invece di login_mode, sono disponibili anche le seguenti modalità:</p> <ul style="list-style-type: none"> anonymousMode. Ad esempio: <code>['Cognos_server_url', 'anonymousMode', "namespace", "username", "password"]</code> credentialMode. Ad esempio: <code>['Cognos_server_url', 'credentialMode', "namespace", "username", "password"]</code> storedCredentialMode. Ad esempio: <code>['Cognos_server_url', 'storedCredentialMode', "stored_credential_name"]</code> <p>Dove stored_credential_name è il nome di una credenziale Cognos nel repository.</p> |
| cognos_package_name | stringa | Il percorso e il nome del package Cognos in cui esportare i dati, per esempio: /Public Folders/MyPackage |
| cognos_datasource | stringa | |
| cognos_export_mode | Publish ExportFile | |
| cognos_filename | stringa | |

Connessione ODBC

Le proprietà della connessione ODBC sono identiche a quelle riportate per databaseexportnode nella sezione che segue, con la differenza che la proprietà datasource non è valida.

Proprietà databaseexportnode



Il nodo di esportazione del database scrive dati in una sorgente dati relazionale compatibile con ODBC. Per scrivere in una sorgente dati ODBC, è necessario utilizzare una sorgente dati esistente e disporre dell'autorizzazione in scrittura per tale sorgente.

Esempio

```
...
Assumes a datasource named "MyDatasource" has been configured
...
stream = modeler.script.stream()
db_exportnode = stream.createAt("databaseexport", "DB Export", 200, 200)
applynn = stream.findByType("applyneuralnetwork", None)
stream.link(applynn, db_exportnode)

# Export tab
db_exportnode.setPropertyValue("username", "user")
db_exportnode.setPropertyValue("datasource", "MyDatasource")
db_exportnode.setPropertyValue("password", "password")
db_exportnode.setPropertyValue("table_name", "predictions")
db_exportnode.setPropertyValue("write_mode", "Create")
db_exportnode.setPropertyValue("generate_import", True)
db_exportnode.setPropertyValue("drop_existing_table", True)
db_exportnode.setPropertyValue("delete_existing_rows", True)
db_exportnode.setPropertyValue("default_string_size", 32)

# Schema dialog
db_exportnode.setKeyedPropertyValue("type", "region", "VARCHAR(10)")
db_exportnode.setKeyedPropertyValue("export_db_primarykey", "id", True)
db_exportnode.setPropertyValue("use_custom_create_table_command", True)
db_exportnode.setPropertyValue("custom_create_table_command", "My SQL Code")

# Indexes dialog
db_exportnode.setPropertyValue("use_custom_create_index_command", True)
db_exportnode.setPropertyValue("custom_create_index_command", "CREATE BITMAP INDEX <index-name>
ON <table-name> <(index-columns)>")
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", [{"fields", ["id", "region"]}]
```

Tabella 233. proprietà databaseexportnode.

| proprietà databaseexportnode | Data type | Descrizione proprietà |
|------------------------------|-----------|---|
| datasource | stringa | |
| username | stringa | |
| password | stringa | |
| epassword | stringa | Questo slot è di sola lettura durante l'esecuzione. Per generare una password codificata, utilizzare lo strumento Password disponibile dal menu Strumenti. Per ulteriori informazioni, consultare l'argomento "Generazione di una password codificata" a pagina 51. |
| table_name | stringa | |

Tabella 233. proprietà databaseexportnode (Continua).

| proprietà databaseexportnode | Data type | Descrizione proprietà |
|---------------------------------|---------------------------|---|
| write_mode | Create Append Merge | |
| map | stringa | Mappa il nome campo di un flusso al nome di una colonna di database (valido solo se write_mode è Merge). In caso di unione è necessario che tutti i campi siano mappati per poter essere esportati. I nomi di campi che non esistono nel database vengono aggiunti come nuove colonne. |
| key_fields | elenco | Specifica il campo del flusso utilizzato come chiave; la proprietà map mostra a che cosa corrisponde nel database. |
| join | Database Add | |
| drop_existing_table | indicatore | |
| delete_existing_rows | indicatore | |
| default_string_size | numero intero | |
| type | | Proprietà strutturata utilizzata per impostare il tipo di schema. |
| generate_import | indicatore | |
| use_custom_create_table_command | indicatore | Utilizzare lo slot <i>custom_create_table</i> per modificare il comando SQL standard CREATE TABLE. |
| custom_create_table_command | stringa | Specifica un comando stringa da utilizzare al posto del comando SQL standard CREATE TABLE. |
| use_batch | indicatore | Le seguenti proprietà sono opzioni avanzate per il caricamento di massa di database. Un valore vero (true) per Use_batch disattiva riga per riga i commit al database. |
| batch_size | numero | Specificare il numero di record da inviare al database prima del commit nella memoria. |
| bulk_loading | Off ODBC External | Specifica il tipo di caricamento di massa. Di seguito vengono elencate opzioni aggiuntive per ODBC ed External. |
| not_logged | indicatore | |
| odbc_binding | Row Column | Specificare l'associazione in base a righe o colonne per il caricamento di massa tramite ODBC. |

Tabella 233. proprietà databaseexportnode (Continua).

| proprietà databaseexportnode | Data type | Descrizione proprietà |
|---------------------------------|-----------------------|---|
| loader_delimit_mode | Tab Space Other | Specificare il tipo di delimitatore per il caricamento di massa tramite un programma esterno. Selezionare Other insieme alla proprietà loader_other_delimiter per specificare i delimitatori, quali la virgola (,). |
| loader_other_delimiter | stringa | |
| specify_data_file | indicatore | Un flag vero (true) attiva la proprietà data_file illustrata di seguito, che consente di specificare il nome e il percorso del file in cui scrivere durante il caricamento di massa nel database. |
| data_file | stringa | |
| specify_loader_program | indicatore | Un flag vero (true) attiva la proprietà loader_program illustrata di seguito, che consente di specificare il nome e la posizione di un programma o di uno script di caricamento esterno. |
| loader_program | stringa | |
| gen_logfile | indicatore | Un flag vero (true) attiva la proprietà logfile_name illustrata di seguito, che consente di specificare il nome di un file sul server per generare un registro errori. |
| logfile_name | stringa | |
| check_table_size | indicatore | Un flag vero (true) consente il controllo della tabella che assicura che l'incremento nelle dimensioni della tabella di database corrisponda al numero di righe esportate da IBM SPSS Modeler. |
| loader_options | stringa | Specificare argomenti aggiuntivi, quali -comment e -specialdir, al programma di caricamento. |
| export_db_primarykey | indicatore | Specifica se un determinato campo è una chiave primaria. |
| use_custom_create_index_command | indicatore | Se true, (vero), attiva l'SQL personalizzato per tutti gli indici. |
| custom_create_index_command | stringa | Specifica il comando SQL utilizzato per creare gli indici quando è attivato l'SQL personalizzato. (Questo valore può essere sovrascritto per indici specifici come indicato di seguito). |
| indexes.INDEXNAME.fields | | Crea l'indice specificato, se necessario, ed elenca i nomi dei campi da includere in tale indice. |

Tabella 233. proprietà databaseexportnode (Continua).

| proprietà databaseexportnode | Data type | Descrizione proprietà |
|--|------------|---|
| INDEXNAME "use_custom_create_index_command" | indicatore | Utilizzato per attivare o disattivare l'SQL personalizzato per un indice specifico. Consultare gli esempi dopo la tabella riportata di seguito. |
| INDEXNAME "custom_create_index_command" | stringa | Specifica l'SQL personalizzato utilizzato per l'indice specificato. Consultare gli esempi dopo la tabella riportata di seguito. |
| indexes.INDEXNAME.remove | indicatore | Se True, rimuove l'indice specificato dall'insieme di indici. |
| table_space | stringa | Specifica lo spazio di tabella che verrà creato. |
| use_partition | indicatore | Indica che verrà utilizzato il campo di hash distribuito. |
| partition_field | stringa | Specifica il contenuto del campo di hash distribuito. |

Nota: Per alcuni database, è possibile specificare che le tabelle del database vengano create per l'esportazione con la compressione (ad esempio, l'equivalente di CREATE TABLE MYTABLE (...) COMPRESS YES; in SQL). Le proprietà use_compression e compression_mode vengono fornite per supportare questa funzione, come segue.

Tabella 234. Proprietà databaseexportnode utilizzando le funzioni di compressione.

| proprietà databaseexportnode | Data type | Descrizione proprietà |
|------------------------------|--|---|
| use_compression | Booleano | Se impostata su True, questa opzione utilizza la compressione nella creazione delle tabelle per l'esportazione. |
| compression_mode | Row Page | Imposta il livello di compressione per i database SQL Server. |
| | Default Direct_Load_Operations All_Operations Basic OLTP Query_High Query_Low Archive_High Archive_Low | Imposta il livello di compressione per i database Oracle. Si noti che i valori OLTP, Query_High, Query_Low, Archive_High e Archive_Low richiedono come minimo Oracle 11gR2. |

Di seguito è riportato un esempio che mostra il modo in cui modificare il comando CREATE INDEX per un indice specifico:

```
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", ["use_custom_create_index_command", True])
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", ["custom_create_index_command", "CREATE BITMAP INDEX <index-name> ON <table-name> <(index-columns)>"])
```

In alternativa, questa operazione può essere eseguita mediante una tabella hash:

```
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", ["fields":["id", "region"], "use_custom_create_index_command":True, "custom_create_index_command":"CREATE INDEX <index-name> ON <table-name> <(index-columns)>"])
```

Proprietà datacollectionexportnode



Il nodo Esportazione di IBM SPSS Data Collection esegue l'output di dati nel formato utilizzato dal software di ricerche di mercato IBM SPSS Data Collection. Per utilizzare questo nodo, è necessario che sia installata IBM SPSS Data Collection Data Library.

Esempio

```
stream = modeler.script.stream()
datacollectionexportnode = stream.createAt("datacollectionexport", "Data Collection", 200, 200)
datacollectionexportnode.setPropertyValue("metadata_file", "c:\\museums.mdd")
datacollectionexportnode.setPropertyValue("merge_metadata", "Overwrite")
datacollectionexportnode.setPropertyValue("casedata_file", "c:\\museumdata.sav")
datacollectionexportnode.setPropertyValue("generate_import", True)
datacollectionexportnode.setPropertyValue("enable_system_variables", True)
```

Tabella 235. proprietà datacollectionexportnode

| Proprietà datacollectionexportnode | Data type | Descrizione proprietà |
|------------------------------------|---------------------------|--|
| metadata_file | stringa | Nome del file di metadati da esportare. |
| merge_metadata | Overwrite MergeCurrent | |
| enable_system_variables | indicatore | Specifica se il file <i>.mdd</i> esportato deve includere le variabili di sistema di IBM SPSS Data Collection. |
| casedata_file | stringa | Il nome del file <i>.sav</i> in cui vengono esportati i dati del caso. |
| generate_import | indicatore | |

Proprietà excelexportnode



Il nodo Esportazione da Excel esegue l'output di dati in formato file *.xlsx* Microsoft Excel. Se lo si desidera, è possibile scegliere di avviare Excel automaticamente e aprire il file esportato quando si esegue il nodo.

Esempio

```
stream = modeler.script.stream()
excelexportnode = stream.createAt("excelexport", "Excel", 200, 200)
excelexportnode.setPropertyValue("full_filename", "C:/output/myexport.xlsx")
excelexportnode.setPropertyValue("excel_file_type", "Excel2007")
excelexportnode.setPropertyValue("inc_field_names", True)
excelexportnode.setPropertyValue("inc_labels_as_cell_notes", False)
excelexportnode.setPropertyValue("launch_application", True)
excelexportnode.setPropertyValue("generate_import", True)
```

Tabella 236. proprietà excelexportnode

| proprietà excelexportnode | Data type | Descrizione proprietà |
|---------------------------|-----------|-----------------------|
| full_filename | stringa | |
| excel_file_type | Excel2007 | |

Tabella 236. proprietà *excelexportnode* (Continua)

| proprietà <i>excelexportnode</i> | Data type | Descrizione proprietà |
|----------------------------------|-------------------|---|
| export_mode | Create Append | |
| inc_field_names | <i>indicatore</i> | Specifica se i nomi dei campi devono essere inclusi nella prima riga del foglio di lavoro. |
| start_cell | <i>stringa</i> | Specifica la cella di partenza per l'esportazione. |
| worksheet_name | <i>stringa</i> | Nome del foglio di lavoro da scrivere. |
| launch_application | <i>indicatore</i> | Specifica se sul file risultante deve essere richiamato Excel. Si noti che il percorso per avviare Excel deve essere specificato nella finestra di dialogo Applicazioni di supporto (menu Strumenti, Applicazioni di supporto). |
| generate_import | <i>indicatore</i> | Specifica se deve essere generato un nodo Importazione da Excel che legga il file di dati esportato. |

Proprietà *outputfilenode*



Il nodo di esportazione File flat restituisce dati in un file di testo delimitato. È utile per esportare i dati che possono essere letti da altri software di analisi o fogli di calcolo.

Esempio

```
stream = modeler.script.stream()
outputfile = stream.createAt("outputfile", "File Output", 200, 200)
outputfile.setPropertyValue("full_filename", "c:/output/flatfile_output.txt")
outputfile.setPropertyValue("write_mode", "Append")
outputfile.setPropertyValue("inc_field_names", False)
outputfile.setPropertyValue("use_newline_after_records", False)
outputfile.setPropertyValue("delimit_mode", "Tab")
outputfile.setPropertyValue("other_delimiter", ",")
outputfile.setPropertyValue("quote_mode", "Double")
outputfile.setPropertyValue("other_quote", "*")
outputfile.setPropertyValue("decimal_symbol", "Period")
outputfile.setPropertyValue("generate_import", True)
```

Tabella 237. proprietà *outputfilenode*

| Proprietà <i>outputfilenode</i> | Data type | Descrizione proprietà |
|---------------------------------|---------------------|--------------------------|
| full_filename | <i>stringa</i> | Nome del file di output. |
| write_mode | Overwrite Append | |
| inc_field_names | <i>indicatore</i> | |
| use_newline_after_records | <i>indicatore</i> | |

Tabella 237. proprietà outputfilenode (Continua)

| Proprietà outputfilenode | Data type | Descrizione proprietà |
|--------------------------|---|-----------------------|
| delimit_mode | Comma Tab Space Other | |
| other_delimiter | <i>car</i> | |
| quote_mode | None Single Double Other | |
| other_quote | <i>indicatore</i> | |
| generate_import | <i>indicatore</i> | |
| encoding | StreamDefault SystemDefault "UTF-8" | |

Proprietà sasexportnode



Il nodo Esporta SAS restituisce nel formato SAS i dati che devono essere letti in SAS o in un package software compatibile con SAS. Sono disponibili tre formati di file SAS: SAS per Windows/OS2, SAS per UNIX o SAS Versione 7/8.

Esempio

```
stream = modeler.script.stream()
sasexportnode = stream.createAt("sasexport", "SAS Export", 200, 200)
sasexportnode.setPropertyValue("full_filename", "c:/output/SAS_output.sas7bdat")
sasexportnode.setPropertyValue("format", "SAS8")
sasexportnode.setPropertyValue("export_names", "NamesAndLabels")
sasexportnode.setPropertyValue("generate_import", True)
```

Tabella 238. proprietà sasexportnode

| proprietà sasexportnode | Data type | Descrizione proprietà |
|-------------------------|---------------------------------|---|
| format | Windows UNIX SAS7 SAS8 | Campi delle etichette delle proprietà Variant. |
| full_filename | <i>stringa</i> | |
| export_names | NamesAndLabels NamesAsLabels | Utilizzata per mappare nomi di campi IBM SPSS Modeler a nomi di variabili IBM SPSS Statistics o SAS durante l'esportazione. |
| generate_import | <i>indicatore</i> | |

Proprietà statisticsexportnode



Il nodo Esporta Statistics restituisce i dati in formato IBM SPSS Statistics *.sav* o *.zsav*. È possibile leggere i file *.sav* o *.zsav* utilizzando IBM SPSS Statistics Base ed altri prodotti. Questo formato viene inoltre utilizzato per i file cache di IBM SPSS Modeler.

Le proprietà di questo nodo sono descritte in “Proprietà statisticsexportnode” a pagina 313.

Proprietà del nodo tm1export



Il nodo di esportazione IBM Cognos TM1 esporta i dati in un formato che può essere letto dai database Cognos TM1.

Esempio di impostazione della proprietà di mapping

```
exportNode.setPropertyValue("spss_field_to_tm1_element_mapping",  
  [[["Dimension_1_1", "Dimension_1_1", False], ["Dimension_2_1", "Dimension_2_1", False],  
  ["Dimension_3_1", "Dimension_3_1", False], ["Periods", "Periods",  
  False]],["Measure_1", "Measure_1", False], ["Measure_2", "Measure_2", False],  
  ["Measure_3", "Measure_3", False]])
```

Tabella 239. Proprietà del nodo tm1export.

| Proprietà del nodo tm1export | Data type | Descrizione proprietà |
|------------------------------|---|---|
| pm_host | stringa | Nota: Solo per le versioni 16.0 e 17.0 Il nome host. Ad esempio: TM1_export.setPropertyValue("pm_host", 'http://9.191.86.82:9510/pmhub/pm') |
| tm1_connection | [<i>"campo"</i> , <i>"campo"</i> , ... <i>"campo"</i>] | Nota: Solo per le versioni 16.0 e 17.0 Una proprietà elenco che contiene i dettagli di connessione per il server TM1. Il formato è: ["TM1_Server_Name", "tm1_username", "tm1_password"] Ad esempio: TM1_export.setPropertyValue("tm1_connection", ['Planning Sample', "admin" "apple"]) |
| admin_host | stringa | Nota: Solo a partire dalla versione 17.1 L'URL per il nome host dell'API REST. |
| server_name | stringa | Nota: Solo a partire dalla versione 17.1 Il nome del server TM1 selezionato da admin_host. |
| selected_cube | campo | Il nome del cubo in cui si stanno esportando i dati. Ad esempio: TM1_export.setPropertyValue("selected_cube", "plan_BudgetPlan") |

Tabella 239. Proprietà del nodo *tm1export* (Continua).

| Proprietà del nodo <i>tm1export</i> | Data type | Descrizione proprietà |
|-------------------------------------|---------------|---|
| <i>spssfield_tm1element_mapping</i> | <i>elenco</i> | L'elemento <i>tm1</i> a cui eseguire il mapping deve essere parte della dimensione colonna per la vista cubo selezionata. Il formato è: <code>[["param1", "valore"], ..., ["paramN", "valore"]]</code> Ad esempio: <code>TM1_export.setPropertyValue("spssfield_tm1element_mapping", [["plan_version", "plan_version"], ["plan_department", "plan_department"]])</code> |

Proprietà *xmlexportnode*



Il nodo *Esporta XML* restituisce i dati in un file in formato XML. Se lo si desidera, è possibile creare un nodo origine XML per leggere nuovamente i dati esportati nel flusso.

Esempio

```
stream = modeler.script.stream()
xmlexportnode = stream.createAt("xmlexport", "XML Export", 200, 200)
xmlexportnode.setPropertyValue("full_filename", "c:/export/data.xml")
xmlexportnode.setPropertyValue("map", ["/catalog/book/genre", "genre"], ["/catalog/book/title", "title"]])
```

Tabella 240. proprietà *xmlexportnode*

| proprietà <i>xmlexportnode</i> | Data type | Descrizione proprietà |
|--------------------------------|-------------------|---|
| <i>full_filename</i> | <i>stringa</i> | (obbligatorio) Percorso e nome file completi del file di esportazione XML. |
| <i>use_xml_schema</i> | <i>indicatore</i> | Specifica se utilizzare uno schema XML (file XSD o DTD) per controllare la struttura dei dati esportati. |
| <i>full_schema_filename</i> | <i>stringa</i> | Percorso e nome file completi del file XSD o DTD da utilizzare. Necessario solo se <i>use_xml_schema</i> è impostato su <i>true</i> (vero). |
| <i>generate_import</i> | <i>indicatore</i> | Genera un nodo origine XML che rileggerà il file di dati esportato nel flusso. |
| <i>records</i> | <i>stringa</i> | Espressione XPath che indica i limiti dei record. |
| <i>map</i> | <i>stringa</i> | Mappa i nomi dei campi alla struttura XML. |

Capitolo 18. Proprietà dei nodi IBM SPSS Statistics

Proprietà statisticsimportnode



Il nodo File legge i dati dal file in formato *.sav* o *.zsav* utilizzato da IBM SPSS Statistics e dai file della cache salvati in IBM SPSS Modeler, che utilizza lo stesso formato.

Esempio

```
stream = modeler.script.stream()
statisticsimportnode = stream.createAt("statisticsimport", "SAV Import", 200, 200)
statisticsimportnode.setPropertyValue("full_filename", "C:/data/drug1n.sav")
statisticsimportnode.setPropertyValue("import_names", True)
statisticsimportnode.setPropertyValue("import_data", True)
```

Tabella 241. proprietà statisticsimportnode.

| Proprietà statisticsimportnode | Data type | Descrizione proprietà |
|--------------------------------|---------------------------------|---|
| full_filename | stringa | Il nome del file completo compreso il percorso. |
| password | stringa | La password. Il parametro password deve essere impostato prima del parametro file_encrypted. |
| file_encrypted | indicatore | Indica se il file è protetto o meno da password. |
| import_names | NamesAndLabels LabelsAsNames | Metodo per gestire nomi ed etichette di variabili. |
| import_data | DataAndLabels LabelsAsData | Metodo per gestire valori ed etichette. |
| use_field_format_for_storage | Booleano | Specifica se utilizzare le informazioni relative al formato dei campi di IBM SPSS Statistics durante le importazioni. |

Proprietà statistictransformnode



Il nodo Trasformazioni Statistics esegue una selezione di comandi di sintassi IBM SPSS Statistics rispetto alle sorgenti dati in IBM SPSS Modeler. Questo nodo richiede una copia di IBM SPSS Statistics con regolare licenza.

Esempio

```
stream = modeler.script.stream()
statistictransformnode = stream.createAt("statistictransform", "Transform", 200, 200)
statistictransformnode.setPropertyValue("syntax", "COMPUTE NewVar = Na + K.")
statistictransformnode.setKeyedPropertyValue("new_name", "NewVar", "Mixed Drugs")
statistictransformnode.setPropertyValue("check_before_saving", True)
```

Tabella 242. proprietà statisticstransformnode

| Proprietà statisticstransformnode | Data type | Descrizione proprietà |
|-----------------------------------|------------|---|
| syntax | stringa | |
| check_before_saving | indicatore | Convalida la sintassi inserita prima di salvare le voci. Visualizza un messaggio di errore se la sintassi non è valida. |
| default_include | indicatore | Per ulteriori informazioni, consultare l'argomento "Proprietà filternode" a pagina 126. |
| include | indicatore | Per ulteriori informazioni, consultare l'argomento "Proprietà filternode" a pagina 126. |
| new_name | stringa | Per ulteriori informazioni, consultare l'argomento "Proprietà filternode" a pagina 126. |

Proprietà statisticsmodelnode



Il nodo Modello Statistics consente di analizzare e operare con i dati eseguendo le procedure IBM SPSS Statistics che generano PMML. Questo nodo richiede una copia di IBM SPSS Statistics con regolare licenza.

Esempio

```
stream = modeler.script.stream()
statisticsmodelnode = stream.createAt("statisticsmodel", "Model", 200, 200)
statisticsmodelnode.setPropertyValue("syntax", "COMPUTE NewVar = Na + K.")
statisticsmodelnode.setKeyedPropertyValue("new_name", "NewVar", "Mixed Drugs")
```

| Proprietà statisticsmodelnode | Data type | Descrizione proprietà |
|-------------------------------|------------|---|
| syntax | stringa | |
| default_include | indicatore | Per ulteriori informazioni, consultare l'argomento "Proprietà filternode" a pagina 126. |
| include | indicatore | Per ulteriori informazioni, consultare l'argomento "Proprietà filternode" a pagina 126. |
| new_name | stringa | Per ulteriori informazioni, consultare l'argomento "Proprietà filternode" a pagina 126. |

Proprietà statisticsoutputnode



Il nodo Output Statistics consente di chiamare una procedura IBM SPSS Statistics per analizzare i dati di IBM SPSS Modeler. È disponibile una vasta gamma di procedure analitiche di IBM SPSS Statistics. Questo nodo richiede una copia di IBM SPSS Statistics con regolare licenza.

Esempio

```
stream = modeler.script.stream()
statisticsoutputnode = stream.createAt("statisticsoutput", "Output", 200, 200)
statisticsoutputnode.setPropertyValue("syntax", "SORT CASES BY Age(A) Sex(A) BP(A) Cholesterol(A)")
statisticsoutputnode.setPropertyValue("use_output_name", False)
statisticsoutputnode.setPropertyValue("output_mode", "File")
statisticsoutputnode.setPropertyValue("full_filename", "Cases by Age, Sex and Medical History")
statisticsoutputnode.setPropertyValue("file_type", "HTML")
```

Tabella 243. proprietà statisticsoutputnode

| Proprietà statisticsoutputnode | Data type | Descrizione proprietà |
|--------------------------------|--------------------|--|
| mode | Dialog Syntax | Selezionare l'opzione "Finestra di dialogo IBM SPSS Statistics" o Editor di sintassi |
| syntax | stringa | |
| use_output_name | indicatore | |
| output_name | stringa | |
| output_mode | Screen File | |
| full_filename | stringa | |
| file_type | HTML SPV SPW | |

Proprietà statisticsexportnode



Il nodo Esporta Statistics restituisce i dati in formato IBM SPSS Statistics *.sav* o *.zsav*. È possibile leggere i file *.sav* o *.zsav* utilizzando IBM SPSS Statistics Base ed altri prodotti. Questo formato viene inoltre utilizzato per i file cache di IBM SPSS Modeler.

Esempio

```
stream = modeler.script.stream()
statisticsexportnode = stream.createAt("statisticsexport", "Export", 200, 200)
statisticsexportnode.setPropertyValue("full_filename", "c:/output/SPSS_Statistics_out.sav")
statisticsexportnode.setPropertyValue("field_names", "Names")
statisticsexportnode.setPropertyValue("launch_application", True)
statisticsexportnode.setPropertyValue("generate_import", True)
```

Tabella 244. Proprietà statisticsexportnode.

| Proprietà statisticsexportnode | Data type | Descrizione proprietà |
|--------------------------------|-------------|--|
| full_filename | stringa | |
| file_type | sav zsav | Salva il file in formato <i>sav</i> o <i>zsav</i> . Ad esempio: statisticsexportnode.setPropertyValue("file_type", "sav") |
| encrypt_file | indicatore | Indica se il file è protetto o meno da password. |
| password | stringa | La password. |
| launch_application | indicatore | |

Tabella 244. Proprietà statisticsexportnode (Continua).

| Proprietà statisticsexportnode | Data type | Descrizione proprietà |
|-----------------------------------|---------------------------------|---|
| export_names | NamesAndLabels NamesAsLabels | Utilizzata per mappare nomi di campi IBM SPSS Modeler a nomi di variabili IBM SPSS Statistics o SAS durante l'esportazione. |
| generate_import | <i>indicatore</i> | |

Capitolo 19. Proprietà dei Supernodi

Nelle tabelle seguenti vengono illustrate le proprietà specifiche dei Supernodi. Si noti che le proprietà comuni dei nodi si applicano anche ai Supernodi.

Tabella 245. Proprietà supernodo terminale

| Nome proprietà | Tipo di proprietà/Elenco di valori | Descrizione proprietà |
|----------------|------------------------------------|-----------------------|
| execute_method | Script Normal | |
| script | stringa | |

Parametri dei Supernodi

Per creare o impostare i parametri dei Supernodi è possibile utilizzare gli script, con il seguente formato generale:

```
mySuperNode.setParameterValue("minvalue", 30)
```

È possibile richiamare il valore del parametro con:

```
value mySuperNode.getParameterValue("minvalue")
```

Ricerca di Supernodi esistenti

È possibile ricercare i Supernodi nei flussi utilizzando la funzione `findByType()`:

```
source_supernode = modeler.script.stream().findByType("source_super", None)
process_supernode = modeler.script.stream().findByType("process_super", None)
terminal_supernode = modeler.script.stream().findByType("terminal_super", None)
```

Impostazione delle proprietà dei nodi incapsulati

È possibile impostare le proprietà per nodi specifici incapsulati all'interno di un Supernodo accedendo al diagramma figlio all'interno del Supernodo. Si supponga per esempio di disporre di un Supernodo origine con un nodo Testo variabile incapsulato per la lettura dei dati. È possibile passare il nome del file da leggere (specificato utilizzando la proprietà `full_filename`) accedendo al diagramma figlio e ricercando il nodo pertinente nel modo riportato di seguito:

```
childDiagram = source_supernode.getChildDiagram()
varfilenode = childDiagram.findByType("variablefile", None)
varfilenode.setPropertyValue("full_filename", "c:/mydata.txt")
```

Creazione di Supernodi

Se si desidera creare ex-novo un Supernodo ed il relativo contenuto, è possibile procedere in modo simile creando il Supernodo, accedendo al diagramma figlio e creando i nodi desiderati. Inoltre, è necessario verificare che i nodi all'interno del diagramma del Supernodo siano collegati ai nodi connettore di input e/o di output. Ad esempio, se si desidera creare un Supernodo di elaborazione:

```
process_supernode = modeler.script.stream().createAt("process_super", "My SuperNode", 200, 200)
childDiagram = process_supernode.getChildDiagram()
filternode = childDiagram.createAt("filter", "My Filter", 100, 100)
childDiagram.linkFromInputConnector(filternode)
childDiagram.linkToOutputConnector(filternode)
```

Appendice A. Riferimento dei nomi del nodo

Questa sezione fornisce un riferimento per i nomi degli script dei nodi in IBM SPSS Modeler.

Nomi dei nugget del modello

Ai nugget del modello (detti anche modelli generati) può essere fatto riferimento per tipo, come avviene per gli oggetti nodo e output. Le seguenti tabelle elencano i nomi di riferimento degli oggetti modello.

Si noti che questi nomi vengono utilizzati specificamente per fare riferimento ai nugget del modello nella palette Modelli (nell'angolo superiore destro della finestra di IBM SPSS Modeler). Per fare riferimento ai nodi modello aggiunti a un flusso ai fini del calcolo del punteggio, viene utilizzato un insieme diverso di nomi preceduti dal prefisso `apply...`. Per ulteriori informazioni, consultare l'argomento Proprietà dei nodi dei nugget del modello.

Nota: in circostanze normali, si consiglia di fare riferimento ai modelli per nome e tipo, in modo da evitare confusione.

Tabella 246. Nomi dei nugget del modello (palette Modelli).

| Nome modello | Modello |
|------------------|---|
| anomalydetection | Anomalia |
| apriori | Apriori |
| autoclassifier | Classificatore automatico |
| autocluster | Cluster automatico |
| autonumeric | Numerico automatico |
| bayesnet | Rete bayesiana |
| c50 | C5.0 |
| carma | Carma |
| cart | C&R Tree |
| chaid | CHAID |
| coxreg | regressione di Cox |
| decisionlist | Elenco decisionale |
| discriminante | Discriminante |
| fattore | PCA/Fattore |
| featureselection | Selezione delle funzioni |
| genlin | Regressione lineare generalizzata |
| glm | GLMM |
| kmeans | Medie K |
| knn | <i>k</i> -elemento adiacente più vicino |
| kohonen | Kohonen |
| regressione | Lineare |
| logreg | Regressione logistica |
| neuralnetwork | Rete neurale |
| quest | QUEST |

Tabella 246. Nomi dei nugget del modello (palette Modelli) (Continua).

| Nome modello | Modello |
|-----------------|--|
| regressione | Regressione lineare |
| sequence | Sequenza |
| slrm | Modello di risposta di autoapprendimento |
| statisticsmodel | modello IBM SPSS Statistics |
| svm | Support vector machine |
| timeseries | Serie temporali |
| twostep | TwoStep |

Tabella 247. Nomi dei nugget del modello (palette Modelli in-database).

| Nome modello | Modello |
|----------------------|--|
| db2imcluster | Raggruppamento cluster IBM ISW |
| db2imlog | Regressione logistica IBM ISW |
| db2imnb | IBM ISW Naive Bayes |
| db2imreg | Regressione IBM ISW |
| db2imtree | Struttura ad albero delle decisioni IBM ISW |
| msassoc | Regole di associazione MS |
| msbayes | Naive Bayes MS |
| mscluster | Raggruppamento cluster MS |
| mslogistic | Regressione logistica MS |
| msneuralnetwork | Rete neurale MS |
| msregression | Regressione lineare MS |
| mssequencecluster | MS Sequence Clustering |
| mstimeseries | Serie temporali MS |
| mstree | Struttura ad albero delle decisioni MS |
| netezabayes | Rete di Bayes Netezza |
| netezadectree | Struttura ad albero delle decisioni di Netezza |
| netezadivcluster | Raggruppamento cluster divisivo Netezza |
| netezaglm | Lineare generalizzato Netezza |
| netezakmeans | Medie K Netezza |
| netezaknn | KNN Netezza |
| netezalineregression | Regressione lineare Netezza |
| netezanaivebayes | Naive Bayes Netezza |
| netezapca | PCA Netezza |
| netezaregtree | Struttura ad albero di regressione Netezza |
| netezatimeseries | Serie temporali Netezza |
| oraabn | Bayes adattivo Oracle |
| oraai | Oracle AI |
| oradecisiontree | struttura ad albero delle decisioni Oracle |
| oraglm | GLM Oracle |
| orakmeans | Oracle k-Medie |

Tabella 247. Nomi dei nugget del modello (palette Modelli in-database) (Continua).

| Nome modello | Modello |
|--------------|--------------------|
| oranb | Naive Bayes Oracle |
| oranmf | NMF Oracle |
| oracluster | O-Cluster Oracle |
| orasvm | SVM Oracle |

Per evitare nomi di modelli duplicati

Quando si utilizzano gli script per manipolare i modelli generati, ricordare che la duplicazione dei nomi dei modelli può comportare riferimenti ambigui. Per evitare questo problema, si consiglia di richiedere nomi univoci per i modelli generati durante lo script.

Per impostare le opzioni per i nomi di modelli duplicati:

1. Dai menu, scegliere:
Strumenti > Opzioni utente
2. Fare clic sulla scheda **Notifiche**.
3. Selezionare **Sostituisci modello precedente** per limitare la duplicazione dei nomi per i modelli generati.

Il comportamento dell'esecuzione dello script può variare tra SPSS Modeler e IBM SPSS Collaboration and Deployment Services nel caso in cui i riferimenti ai modelli siano ambigui. SPSS Modeler client contiene l'opzione "Sostituisci modello precedente", che sostituisce automaticamente i modelli con lo stesso nome (ad esempio, nel caso in cui uno script esegua iterazioni sulla base di un ciclo per generare ogni volta un modello diverso). Tuttavia, questa opzione non è disponibile quando il medesimo script viene eseguito in IBM SPSS Collaboration and Deployment Services. Per evitare questa situazione, si può rinominare il modello generato da ogni iterazione, evitando così riferimenti ambigui ai modelli, oppure eliminare il modello corrente (aggiungendo, ad esempio un'istruzione `clear generated palette`) prima della fine del ciclo.

Nomi dei tipi di output

La seguente tabella elenca tutti i tipi di oggetti output e i nodi che li creano. Per un elenco completo dei formati di esportazione disponibili per ciascun tipo di oggetto di output, vedere la descrizione delle proprietà per il nodo che crea il tipo di output, disponibile in Proprietà comuni dei nodi Grafici e Proprietà dei nodi Output.

Tabella 248. Tipi di oggetto di output e i nodi che li creano.

| Tipo di oggetto di output | Nodo |
|---------------------------|---------------|
| analysisoutput | Analisi |
| collectionoutput | Raccolta |
| dataauditoutput | Verifica dati |
| distributionoutput | Distribuzione |
| evaluationoutput | Valutazione |
| histogramoutput | Istogramma |
| matrixoutput | Matrice |
| meansoutput | Medie |
| multiplotoutput | Multiplot |
| plotoutput | Grafico |

Tabella 248. Tipi di oggetto di output e i nodi che li creano (Continua).

| Tipo di oggetto di output | Nodo |
|---------------------------|---|
| qualityoutput | Qualità |
| reportdocumentoutput | Questo tipo di oggetto non proviene da un nodo, ma si tratta dell'output creato da un report progetto |
| reportoutput | Report |
| statisticsprocedureoutput | Output di Statistics |
| statisticsoutput | Statistiche |
| tableoutput | Tabella |
| timeplotoutput | Grafico temporale |
| weboutput | Web |

Appendice B. Migrazione da script legacy a script Python

Panoramica sulla migrazione di script Legacy

Questa sezione fornisce un riepilogo delle differenze tra gli script Python e legacy in IBM SPSS Modeler e fornisce informazioni relative alla migrazione degli script legacy in script Python. In questa sezione, è riportato un elenco di comandi Legacy SPSS Modeler standard e dei comandi Python equivalenti.

Differenze generali

Gli script legacy devono gran parte della loro progettazione agli script di comandi del sistema operativo. Gli script legacy sono orientati alla linea di comando e sebbene ci siano alcune strutture a blocco, per esempio `if...then...else...endif` e `for...endfor`, i rientri generalmente non sono significativi.

Negli script Python, il rientro è significativo e le linee che appartengono allo stesso blocco logico devono avere lo stesso livello di rientro.

Nota: È necessario prestare attenzione quando si copia ed incolla il codice scritto in Python. Una linea che ha un rientro che utilizza le tabulazioni, nell'editor potrebbe sembrare uguale ad un'altra che utilizza un rientro con gli spazi. Comunque, lo script Python genererà un errore a causa del fatto che le linee non hanno gli stessi rientri.

Contesto di script

Il contesto di script definisce l'ambiente in cui viene eseguito lo script, per esempio il flusso o il supernodo che esegue lo script. Nello script legacy il contesto è implicito, che significa, per esempio, che si presuppone che tutti i riferimenti del nodo in uno script del flusso siano all'interno del flusso che esegue lo script.

Nello script Python, il contesto dello script viene fornito esplicitamente tramite il modulo `modeler.script`. Per esempio, uno script Python del flusso può accedere al flusso che esegue lo script con il seguente codice:

```
s = modeler.script.stream()
```

Le funzioni correlate del flusso possono quindi essere invocate attraverso l'oggetto restituito.

Comandi o funzioni

Gli script Legacy sono comandi orientati. Questo significa che ogni riga di script in genere viene avviata con il comando da eseguire seguito da parametri, ad esempio:

```
connect 'Type':typenode to :filternode  
rename :derivenode as "Compute Total"
```

Python utilizza funzioni che vengono di solito invocate attraverso un oggetto (un modulo, classe o oggetto) che definisce la funzione, ad esempio:

```
stream = modeler.script.stream()  
typenode = stream.findByType("type", "Type")  
filternode = stream.findByType("filter", None)  
stream.link(typenode, filternode)  
derive.setLabel("Compute Total")
```

Valori letterali e commenti

Alcuni comandi relativi a commenti e valori letterali comunemente utilizzati in IBM SPSS Modeler dispongono di comandi equivalenti negli script Python. Ciò può rendere più semplice la conversione degli script Legacy SPSS Modeler esistenti in script Python da utilizzare in IBM SPSS Modeler 17.

Tabella 249. Mapping degli script Legacy con gli script Python per valori letterali e commenti.

| Script Legacy | Script Python |
|---|---|
| Intero, ad esempio 4 | Stesso |
| Float, ad esempio 0.003 | Stesso |
| Stringhe racchiuse tra apici singoli, ad esempio 'Hello' | Stesso Nota: I valori letterali della stringa contenente caratteri non ASCII devono essere preceduti da una u per garantire che siano rappresentati come Unicode. |
| Stringhe racchiuse tra virgolette, ad esempio "Hello again" | Stesso Nota: I valori letterali della stringa contenente caratteri non ASCII devono essere preceduti da una u per garantire che siano rappresentati come Unicode. |
| Stringhe di lunghezza elevata, ad esempio """This is a string that spans multiple lines""" | Stesso |
| Elenchi, ad esempio [1 2 3] | [1, 2, 3] |
| Riferimento a variabile, ad esempio set x = 3 | x = 3 |
| Continuazione di riga (\), ad esempio set x = [1 2 \ 3 4] | x = [1, 2,\n3, 4] |
| Commento di blocco, ad esempio /* This is a long comment over a line. */ | """ This is a long comment over a line. """ |
| Commento di riga, ad esempio set x = 3 # make x 3 | x = 3 # make x 3 |
| undef | None |
| true | True |
| false | False |

Operatori

Alcuni comandi dell'operatore comunemente utilizzati in IBM SPSS Modeler dispongono di comandi equivalenti negli script Python. Ciò può rendere più semplice la conversione degli script Legacy SPSS Modeler esistenti in script Python da utilizzare in IBM SPSS Modeler 17.

Tabella 250. Mapping degli script Legacy con gli script Python per gli operatori.

| Script Legacy | Script Python |
|---|---|
| NUM1 + NUM2 LIST + ITEM LIST1 + LIST2 | NUM1 + NUM2 LIST.append(ITEM) LIST1.extend(LIST2) |
| NUM1 - NUM2 LIST - ITEM | NUM1 - NUM2 LIST.remove(ITEM) |
| NUM1 * NUM2 | NUM1 * NUM2 |
| NUM1 / NUM2 | NUM1 / NUM2 |

Tabella 250. Mapping degli script Legacy con gli script Python per gli operatori (Continua).

| Script Legacy | Script Python |
|------------------------------------|------------------------------------|
| = == | == |
| /= /== | != |
| X ** Y | X ** Y |
| X < Y X <= Y X > Y X >= Y | X < Y X <= Y X > Y X >= Y |
| X div Y X rem Y X mod Y | X // Y X % Y X % Y |
| and or not (EXPR) | and or not EXPR |

Istruzioni condizionali e cicli

Alcuni comandi condizionali e di ciclo comunemente utilizzati in IBM SPSS Modeler hanno comandi equivalenti nel linguaggio di script di Python. Ciò può rendere più semplice la conversione degli script Legacy SPSS Modeler esistenti in script Python da utilizzare in IBM SPSS Modeler 17.

Tabella 251. Mapping degli script Legacy con gli script Python per istruzioni condizionali e cicli.

| Script Legacy | Script Python |
|--|--|
| for VAR from INT1 to INT2 ... endfor | for VAR in range(INT1, INT2): ... o VAR = INT1 while VAR <= INT2: ... VAR += 1 |
| for VAR in LIST ... endfor | for VAR in LIST: ... |
| for VAR in_fields_to NODE ... endfor | for VAR in NODE.getInputDataModel(): ... |
| for VAR in_fields_at NODE ... endfor | for VAR in NODE.getOutputDataModel(): ... |
| if...then ... elseif...then ... else ... endif | if ...: ... elif ...: ... else: ... |
| with TYPE OBJECT ... endwith | Nessun equivalente |
| var VAR1 | La dichiarazione della variabile non è richiesta |

Variabili

Nello script legacy, le variabili vengono dichiarate prima di farvi riferimento, per esempio:

```
var mynode
set mynode = create typenode at 96 96
```

Nello script Python, le variabili vengono dichiarate prima di farvi riferimento, per esempio:

```
mynode = stream.createAt("type", "Type", 96, 96)
```

Nello script legacy, i riferimenti alle variabili devono essere esplicitamente rimossi utilizzando l'operatore `^`, per esempio:

```
var mynode
set mynode = create typenode at 96 96
set ^mynode.direction."Age" = Input
```

Come molti linguaggi di script, ciò non è necessario nello script Python, per esempio:

```
mynode = stream.createAt("type", "Type", 96, 96)
mynode.setKeyedPropertyValue("direction", "Age", "Input")
```

Tipi di nodo, output e modello

Negli script legacy, i diversi tipi di oggetto (nodo, output e modello) hanno tipicamente il tipo accodato al tipo di oggetto. Ad esempio, il nodo Ricava ha il tipo `derivenode`:

```
set feature_name_node = create divenode at 96 96
```

Le API IBM SPSS Modeler in Python non includono il suffisso `node`, per cui il nodo Ricava ha il tipo `derive`, ad esempio:

```
feature_name_node = stream.createAt("derive", "Feature", 96, 96)
```

La sola differenza nei nomi del tipo negli script legacy e Python è la mancanza del suffisso del tipo.

Nomi proprietà

I nomi delle proprietà sono gli stessi sia nello script legacy che in quello Python. Ad esempio, nel nodo Testo Variabile, la proprietà che definisce la posizione del file è `full_filename` in entrambi gli ambienti di script.

Riferimenti a nodi

Molti script legacy utilizzano una ricerca implicita per trovare ed accedere al nodo da modificare. Per esempio, i comandi seguenti ricercano nel flusso corrente un nodo Tipo con etichetta "Type", quindi impostano la direzione (o ruolo di modellazione) del campo "Age" ad input ed il campo "Drug" come target, che è il valore che deve essere previsto:

```
set 'Type':typenode.direction."Age" = Input
set 'Type':typenode.direction."Drug" = Target
```

Negli script Python, gli oggetti nodo devono essere individuati esplicitamente prima di richiamare la funzione per impostare il valore della proprietà, per esempio:

```
typenode = stream.findByType("type", "Type")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
typenode.setKeyedPropertyValue("direction", "Drug", "Target")
```

Nota: In questo caso, "Target" deve essere tra virgolette.

Gli script Python possono utilizzare in alternativa l'enumerazione `ModelingRole` nel package `modeler.api`.

Sebbene la versione di script Python può essere più dettagliata, si ottengono prestazioni a runtime migliori poiché la ricerca del nodo di solito viene eseguita solo una volta al giorno. Nell'esempio di script legacy, la ricerca di un nodo viene fatta per ogni comando.

È supportata anche la ricerca dei nodi tramite ID (l'ID del nodo è visibile nella scheda Annotazioni della finestra di dialogo del nodo). Per esempio, nello script legacy:

```
# id65EMPB9VL87 is the ID of a Type node
set @id65EMPB9VL87.direction."Age" = Input
```

Il seguente script mostra lo stesso esempio negli script Python:

```
typenode = stream.findByID("id65EMPB9VL87")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
```

Ottenimento ed impostazione di proprietà

Gli script Legacy utilizzano il comando set per assegnare un valore. Il termine successivo al comando set può essere una definizione di proprietà. Il seguente script mostra due possibili formati di script per l'impostazione delle proprietà:

```
set <node reference>.<property> = <value>
set <node reference>.<keyed-property>.<key> = <value>
```

Negli script Python, lo stesso risultato si ottiene utilizzando le funzioni setPropertyValue() e setKeyedPropertyValue(), ad esempio:

```
object.setPropertyValue(property, value)
object.setKeyedPropertyValue(keyed-property, key, value)
```

Negli script legacy, l'accesso ai valori delle proprietà può essere ottenuto utilizzando il comando get, ad esempio:

```
var n v
set n = get node :filternode
set v = ^n.name
```

Negli script Python, lo stesso risultato si ottiene utilizzando la funzione getPropertyValue(), ad esempio:

```
n = stream.findByType("filter", None)
v = n.getPropertyValue("name")
```

Modifica dei flussi

Negli script legacy, il comando create viene utilizzato per creare un nuovo nodo, ad esempio:

```
var agg select
set agg = create aggregatenode at 96 96
set select = create selectnode at 164 96
```

Negli script Python, i flussi hanno vari metodi per la creazione di nodi, ad esempio:

```
stream = modeler.script.stream()
agg = stream.createAt("aggregate", "Aggregate", 96, 96)
select = stream.createAt("select", "Select", 164, 96)
```

Negli script legacy, il comando connect viene utilizzato per creare collegamenti tra nodi, per esempio:

```
connect ^agg to ^select
```

Negli script Python, il metodo link viene utilizzato per creare collegamenti tra nodi, ad esempio:

```
stream.link(agg, select)
```

Negli script legacy, il comando disconnect viene utilizzato per rimuovere i collegamenti tra i nodi, ad esempio:

```
disconnect ^agg from ^select
```

Negli script Python, il metodo `unlink` viene utilizzato per rimuovere i collegamenti tra i nodi, ad esempio:

```
stream.unlink(agg, select)
```

Negli script legacy, il comando `position` viene utilizzato per posizionare i nodi sull'area di disegno del flusso o tra altri nodi, ad esempio:

```
position ^agg at 256 256  
position ^agg between ^myselect and ^mydistinct
```

Negli script Python, lo stesso risultato viene ottenuto utilizzando due metodi separati: `setXYPosition` e `setPositionBetween`. Ad esempio:

```
agg.setXYPosition(256, 256)  
agg.setPositionBetween(myselect, mydistinct)
```

Operazioni nodo

Alcuni comandi di operazione nodo che sono comunemente utilizzati in IBM SPSS Modeler hanno un comando equivalente nel linguaggio di script Python. Ciò può rendere più semplice la conversione degli script Legacy SPSS Modeler esistenti in script Python da utilizzare in IBM SPSS Modeler 17.

Tabella 252. Mapping degli script Legacy con gli script Python per le operazioni di nodo.

| Script Legacy | Script Python |
|--|---|
| create <i>nodespec</i> at x y | <code>stream.create(type, name)</code> <code>stream.createAt(type, name, x, y)</code> <code>stream.createBetween(type, name, preNode, postNode)</code> <code>stream.createModelApplier(model, name)</code> |
| connect <i>fromNode</i> to <i>toNode</i> | <code>stream.link(fromNode, toNode)</code> |
| delete <i>node</i> | <code>stream.delete(node)</code> |
| disable <i>node</i> | <code>stream.setEnabled(node, False)</code> |
| enable <i>node</i> | <code>stream.setEnabled(node, True)</code> |
| disconnect <i>fromNode</i> from <i>toNode</i> | <code>stream.unlink(fromNode, toNode)</code> <code>stream.disconnect(node)</code> |
| duplicate <i>node</i> | <code>node.duplicate()</code> |
| execute <i>node</i> | <code>stream.runSelected(nodes, results)</code> <code>stream.runAll(results)</code> |
| flush <i>node</i> | <code>node.flushCache()</code> |
| position <i>node</i> at x y | <code>node.setXYPosition(x, y)</code> |
| position <i>node</i> between <i>node1</i> and <i>node2</i> | <code>node.setPositionBetween(node1, node2)</code> |
| rename <i>node</i> as <i>name</i> | <code>node.setLabel(name)</code> |

Esecuzione di cicli

Negli script legacy, vi sono due opzioni principali di esecuzione di cicli che sono supportate:

- Esecuzione di cicli *Conteggiati*, dove una variabile indice si sposta tra due limiti interi.
- Esecuzione di cicli in *Sequenza* che ciclano attraverso una sequenza di valori, associando il valore corrente alla variabile dell'esecuzione di cicli.

Il seguente script è un esempio di esecuzione di cicli conteggiato negli script legacy:

```
for i from 1 to 10
  println ^i
endfor
```

Il seguente script è un esempio di esecuzione di cicli in sequenza negli script legacy:

```
var items
set items = [a b c d]

for i in items
  println ^i
endfor
```

Vi sono anche altri tipi di esecuzione di cicli che possono essere utilizzati:

- Iterazione tra i modelli nella tavolozza dei modelli oppure tra gli output nella tavolozza degli output.
- Iterazione tra i campi che entrano o escono da un nodo.

Gli script Python supportano anche diversi tipi di esecuzione di cicli. Il seguente script è un esempio di esecuzione di cicli conteggiati negli script Python:

```
i = 1
while i <= 10:
  print i
  i += 1
```

Il seguente script è un esempio di esecuzione di cicli in sequenza negli script Python:

```
items = ["a", "b", "c", "d"]
for i in items:
  print i
```

L'esecuzione di cicli in sequenza è molto flessibile e quando viene associata ai metodi API di IBM SPSS Modeler può supportare la maggioranza di casi di utilizzo di script legacy. Il seguente esempio mostra come utilizzare una esecuzione di cicli in sequenza negli script Python per scorrere attraverso i campi che provengono da un nodo:

```
node = modeler.script.stream().findByType("filter", None)
for column in node.getOutputDataModel().columnIterator():
  print column.getColumnName()
```

esecuzione di flussi

Durante l'esecuzione del flusso, il modello o gli oggetti di output che vengono generati vengono aggiunti ad uno dei gestori dell'oggetto. Negli script legacy, lo script deve individuare gli oggetti creati dal gestore dell'oggetto oppure accedere all'output generato più recentemente dal nodo che ha generato l'output stesso.

Il flusso di esecuzione in Python è diverso, in quel caso ogni oggetto modello o output che sono generati dall'esecuzione vengono restituiti in un elenco che viene inoltrato alla funzione di esecuzione. Questo rende più semplice l'accesso ai risultati di un flusso di esecuzione.

Gli script legacy supportano tre comandi di esecuzione del flusso:

- `execute_all` esegue tutti i nodi terminale eseguibili nel flusso.
- `execute_script` esegue lo script del flusso indipendentemente dall'impostazione di esecuzione dello script.
- `execute node` esegue il nodo specificato.

Gli script Python supportano un insieme analogo di funzioni:

- `stream.runAll(results-list)` esegue tutti i nodi terminali eseguibili nel flusso.

- `stream.runScript(results-list)` esegue lo script del flusso indipendentemente dall'impostazione dell'esecuzione dello script.
- `stream.runSelected(node-array, results-list)` esegue l'insieme di nodi specificato nell'ordine in cui sono stati forniti.
- `node.run(results-list)` esegue il nodo specificato.

In uno script, l'esecuzione di un flusso può essere interrotta utilizzando il comando `exit` seguito da un codice intero facoltativo, per esempio:

```
exit 1
```

In uno script Python, lo stesso risultato si ottiene con il seguente script:

```
modeler.script.exit(1)
```

Accesso ad oggetti attraverso il file system ed il repository

Negli script legacy, è possibile aprire un flusso esistente, un modello o un oggetto di output utilizzando il comando `open`, per esempio:

```
var s
set s = open stream "c:/my streams/modeling.str"
```

Negli script Python, esiste una classe `TaskRunner` che è accessibile dalla sessione e può essere utilizzata per effettuare una azione simile, per esempio:

```
taskrunner = modeler.script.session().getTaskRunner()
s = taskrunner.openStreamFromFile("c:/my streams/modeling.str", True)
```

Per salvare un oggetto negli script legacy, è possibile utilizzare il comando `save`, per esempio:

```
save stream s as "c:/my streams/new_modeling.str"
```

L'approccio equivalente negli script Python sarebbe quello di utilizzare la classe `TaskRunner`, per esempio:

```
taskrunner.saveStreamToFile(s, "c:/my streams/new_modeling.str")
```

Le operazioni basate su IBM SPSS Collaboration and Deployment Services Repository sono supportate negli script legacy attraverso i comandi `retrieve` e `store`, per esempio:

```
var s
set s = retrieve stream "/my repository folder/my_stream.str"
store stream ^s as "/my repository folder/my_stream_copy.str"
```

Negli script Python, la funzionalità equivalente potrebbe essere accessibile tramite l'oggetto `Repository` associato alla sessione, per esempio:

```
session = modeler.script.session()
repo = session.getRepository()
s = repo.retrieveStream("/my repository folder/my_stream.str", None, None, True)
repo.storeStream(s, "/my repository folder/my_stream_copy.str", None)
```

Nota: L'accesso al repository richiede che la sessione sia stata configurata con una connessione al repository valida.

Operazioni di flusso

Alcuni comandi di operazioni di flusso che sono comunemente utilizzati in IBM SPSS Modeler hanno un comando equivalente nel linguaggio di script di Python. Ciò può rendere più semplice la conversione degli script Legacy SPSS Modeler esistenti in script Python da utilizzare in IBM SPSS Modeler 17.

Tabella 253. Mapping degli script Legacy con gli script Python per le operazioni di flusso.

| Script Legacy | Script Python |
|---------------------------------------|--|
| create stream <i>DEFAULT_FILENAME</i> | <code>taskrunner.createStream(name, autoConnect, autoManage)</code> |
| close stream | <code>stream.close()</code> |
| clear stream | <code>stream.clear()</code> |
| get stream <i>stream</i> | Nessun equivalente |
| load stream <i>path</i> | Nessun equivalente |
| open stream <i>path</i> | <code>taskrunner.openStreamFromFile(path, autoManage)</code> |
| save <i>stream</i> as <i>path</i> | <code>taskrunner.saveStreamToFile(stream, path)</code> |
| retrieve stream <i>path</i> | <code>repository.retrieveStream(path, version, label, autoManage)</code> |
| store <i>stream</i> as <i>path</i> | <code>repository.storeStream(stream, path, label)</code> |

Operazioni del modello

Alcuni comandi di operazione di modello che sono comunemente utilizzati in IBM SPSS Modeler hanno un comando equivalente nel linguaggio di script Python. Ciò può rendere più semplice la conversione degli script Legacy SPSS Modeler esistenti in script Python da utilizzare in IBM SPSS Modeler 17.

Tabella 254. Mapping degli script Legacy con gli script Python per le operazioni di modello.

| Script Legacy | Script Python |
|-----------------------------------|---|
| open model <i>path</i> | <code>taskrunner.openModelFromFile(path, autoManage)</code> |
| save <i>model</i> as <i>path</i> | <code>taskrunner.saveModelToFile(model, path)</code> |
| retrieve model <i>path</i> | <code>repository.retrieveModel(path, version, label, autoManage)</code> |
| store <i>model</i> as <i>path</i> | <code>repository.storeModel(model, path, label)</code> |

Operazioni di output di documento

Alcuni comandi di operazioni di output di documenti che sono comunemente utilizzati in IBM SPSS Modeler hanno un comando equivalente nel linguaggio di script di Python. Ciò può rendere più semplice la conversione degli script Legacy SPSS Modeler esistenti in script Python da utilizzare in IBM SPSS Modeler 17.

Tabella 255. Mapping degli script legacy con gli script Python per le operazioni di output di documenti.

| Script Legacy | Script Python |
|------------------------------------|--|
| open output <i>path</i> | <code>taskrunner.openDocumentFromFile(path, autoManage)</code> |
| save <i>output</i> as <i>path</i> | <code>taskrunner.saveDocumentToFile(output, path)</code> |
| retrieve output <i>path</i> | <code>repository.retrieveDocument(path, version, label, autoManage)</code> |
| store <i>output</i> as <i>path</i> | <code>repository.storeDocument(output, path, label)</code> |

Altre differenze tra script legacy e script Python

Gli script Legacy forniscono supporto per la gestione di progetti IBM SPSS Modeler. Attualmente gli script Python non lo supportano.

Gli script Legacy forniscono supporto per il caricamento di oggetti *stato* (combinazioni di flussi e modelli). Gli oggetti Stato sono obsoleti da IBM SPSS Modeler 8.0. Gli script Python non supportano gli oggetti Stato.

Gli script Python offrono le seguenti funzioni aggiuntive che non sono disponibili negli script legacy:

- Definizioni di classe e funzione
- Gestione degli errori
- Supporto più sofisticato di input/output
- Moduli esterni e terze parti

Informazioni particolari

Queste informazioni sono state preparate per prodotti e servizi offerti in tutto il mondo.

È possibile che IBM non offra i prodotti, servizi o funzioni illustrati in questa documentazione. Consultare il rappresentante locale IBM per le informazioni sui prodotti e servizi attualmente disponibili nella propria zona. Qualsiasi riferimento a un prodotto, programma o servizio IBM non implica o intende dichiarare che può essere utilizzato solo quel prodotto, programma o servizio IBM. In sostituzione a quelli forniti da IBM è possibile utilizzare qualsiasi prodotto, programma o servizio funzionalmente equivalente che non comporti la violazione dei diritti di proprietà intellettuale IBM o altri diritti. Tuttavia, è responsabilità dell'utente valutare e verificare il funzionamento di qualsiasi prodotto, programma o servizio non IBM.

IBM potrebbe avere brevetti o domande di brevetti in corso relativi ad argomenti discussi nella presente pubblicazione. La consegna del presente documento non conferisce alcuna licenza rispetto a questi brevetti. Rivolgere per iscritto i quesiti sulle licenze a:

IBM Director of Licensing
IBM Europe
Schoenaicher Str.220
D-7030 Boeblingen
Deutschland

Per richieste di licenze relative ad informazioni double-byte (DBCS) contattare il Dipartimento di Proprietà Intellettuale IBM nel proprio paese o inviare richieste per iscritto a:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

Il seguente paragrafo non è valido nel Regno Unito o per tutti i paesi le cui leggi nazionali siano in contrasto con le disposizioni in esso contenute INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE QUESTA PUBBLICAZIONE "NELLO STATO IN CUI ESSA SI TROVA" SENZA ALCUNA GARANZIA ESPLICITA O IMPLICITA IVI INCLUSE EVENTUALI GARANZIE DI COMMERCIALIZZABILITÀ ED IDONEITÀ AD UNO SCOPO PARTICOLARE Alcuni stati non consentono limitazioni di garanzie espresse o implicite in determinate transazioni, pertanto quanto sopra potrebbe non essere applicabile.

Le presenti informazioni possono includere imprecisioni tecniche o errori tipografici. Le modifiche periodiche apportate alle informazioni contenute in questa pubblicazione verranno inserite nelle nuove edizioni della pubblicazione. IBM si riserva il diritto di apportare miglioramenti e/o modifiche al prodotto o programma descritto in questa pubblicazione in qualsiasi momento e senza preavviso.

Qualsiasi riferimento nelle presenti informazioni a siti Web non IBM viene fornito esclusivamente per facilitare la consultazione e non rappresenta in alcun modo un'approvazione o sostegno da parte nostra di tali siti Web. I materiali disponibili sui siti Web non fanno parte di questo prodotto IBM e l'utilizzo di questi è a discrezione dell'utente.

IBM può utilizzare o distribuire qualsiasi informazione fornita dall'utente nel modo che ritiene più idoneo senza incorrere in alcun obbligo nei confronti dell'utente stesso.

I licenziatari di questo programma che desiderano avere informazioni allo scopo di abilitare: (i) lo scambio di informazioni tra i programmi creati indipendentemente e gli altri programmi (incluso il presente) e (ii) il reciproco utilizzo di informazioni che sono state scambiate, dovrebbero contattare:

IBM Software Group
ATTN: Licensing
200 W. Madison St.
Chicago, IL; 60606
U.S.A.

Tali informazioni saranno fornite in conformità ai termini e alle condizioni in vigore e, in alcuni casi, dietro pagamento.

Il programma su licenza descritto in questa documentazione e tutto il materiale su licenza ad esso relativo vengono forniti da IBM nei termini del Customer Agreement IBM IBM International Program License Agreement o di eventuali accordi equivalenti intercorsi tra le parti.

Tutti i dati sulle prestazioni qui contenuti sono stati elaborati in ambiente controllato. Di conseguenza, i risultati ottenuti con sistemi operativi diversi possono variare in modo significativo. Alcune misurazioni potrebbero essere state effettuate su sistemi in corso di sviluppo e non c'è garanzia che tali misurazioni coincidano con quelle effettuate sui sistemi comunemente disponibili. Inoltre, alcune misurazioni potrebbero essere stime elaborate tramite l'estrapolazione. I risultati effettivi potrebbero variare. Gli utenti di questo documento devono verificare i dati relativi al proprio ambiente specifico.

Le informazioni relative a prodotti non IBM sono state ottenute dai fornitori di tali prodotti, da loro annunci pubblicati e da altre fonti disponibili al pubblico. IBM non ha testato quei prodotti e non può garantire l'accuratezza delle prestazioni, la compatibilità o qualsiasi altra dichiarazione relativa a prodotti non-IBM. Eventuali domande in merito alle funzionalità dei prodotti non IBM vanno indirizzate ai fornitori di tali prodotti.

Qualsiasi affermazione relativa agli obiettivi e alla direzione futura di IBM è soggetta a modifica o revoca senza preavviso e concerne esclusivamente gli scopi dell'azienda.

Le presenti informazioni includono esempi di dati e report utilizzati in operazioni di business quotidiane. Per fornire una descrizione il più possibile esaustiva, gli esempi includono nomi di persone, società, marchi e prodotti. Tutti questi nomi sono fittizi e ogni somiglianza a nomi e indirizzi utilizzati da aziende reali è puramente casuale.

Per chi visualizza queste informazioni a video: le fotografie e le illustrazioni a colori potrebbero non essere disponibili.

Marchi

IBM, il logo IBM e ibm.com sono marchi o marchi registrati di International Business Machines Corp., registrati in molte giurisdizioni nel mondo. Altri nomi di prodotti e servizi possono essere marchi di IBM o altre società. Un elenco aggiornato dei marchi IBM è disponibile su Web alla pagina "Copyright and trademark information" all'indirizzo www.ibm.com/legal/copytrade.shtml.

Intel, il logo Intel, Intel Inside, il logo Intel Inside, Intel Centrino, il logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium e Pentium sono marchi o marchi registrati di Intel Corporation o delle sue consociate negli Stati Uniti e in altri paesi.

Linux è un marchio registrato di Linus Torvalds negli Stati Uniti e/o negli altri paesi.

Microsoft, Windows, Windows NT e il logo Windows sono marchi di Microsoft Corporation negli Stati Uniti e/o negli altri paesi.

UNIX è un marchio registrato di The Open Group negli Stati Uniti e in altri paesi.

Java e tutti i marchi e logo basati su Java sono marchi o marchi registrati di Oracle e/o suoi affiliati.

Altri nomi di prodotti e servizi possono essere marchi commerciali di IBM o di altre aziende.

Indice analitico

A

accesso ai risultati dell'esecuzione del flusso 52, 57
 modello di contenuto JSON 56
 modello di contenuto tabella 53
 modello di contenuto XML 54
aggiunta di attributi 24
API di script
 accesso agli oggetti generati 40
 esempio 37
 gestione degli errori 41
 introduzione 37
 metadati 37
 parametri di sessione 42
 parametri flusso 42
 parametri Supernodo 42
 più flussi 47
 ricerca 37
 script autonomi 47
 valori globali 46
argomenti
 Connessione a IBM SPSS
 Collaboration and Deployment Services Repository 65
 connessione al repository IBM SPSS Analytic Server 65
 connessione al server 64
 file dei comandi 66
 sistema 62
attraversamento dei nodi 33

B

blocchi di codice 19

C

campi
 disattivazione negli script 145
caratteri non-ASCII 22
chiave di iterazione
 esecuzione di cicli negli script 8
cicli
 uso negli script 49
CLEM
 script 1
Cluster di sequenze MS
 proprietà script dei nodi 255
comando clear generated palette 52
comando for 49
comando multiset 67
contrassegni 19
controllo degli errori
 script 51
creazione di nodi 31, 32
creazione di una classe 24

D

definizione degli attributi 24
definizione dei metodi 24
definizione di una classe 24
derive_stbnode
 proprietà 101
diagrammi 27

E

elenchi 16
ereditarietà 25
esecuzione condizionale di flussi 6, 10
esecuzione degli script 11
esecuzione di cicli nei flussi 6, 7
esecuzione di flussi 27
esempi 20

F

flussi
 comando multiset 67
 esecuzione condizionale 6, 10
 esecuzione di cicli 6
 proprietà 71
 script 1, 27
flusso
 esecuzione 27
 esecuzione di cicli 7
 modifica 31
funzione lowertoupper 49
funzioni
 ciclo 323
 comandi condizionali 323
 commenti 322
 operatori 322
 operazioni del modello 329
 operazioni di flusso 329
 operazioni di output di documento 329
 operazioni nodo 326
 riferimenti a oggetti 322
 valori letterali 322
funzioni stringa 49

I

I modelli IBM SPSS Statistics
 proprietà script dei nodi 312
IBM SPSS Collaboration and Deployment Services Repository
 argomenti della riga di comando 65
 script 50
IBM SPSS Modeler
 esecuzione dalla riga di comando 61
identificatori 19
impostazione di proprietà 30
indicatori
 argomenti della riga di comando 61

indicatori (*Continua*)

 combinazione di più flag 66
 interruzione degli script 11
 istruzioni 19

J

Jython 15

M

Mdelli K-medie Oracle
 proprietà script dei nodi 263
metodi matematici 21
migrazione
 accesso agli oggetti 328
 cancellazione di manager flussi, output e modelli 34
 comandi 321
 contesto di script 321
 differenze generali 321
 esecuzione di cicli 326
 esecuzione di flussi 327
 file system 328
 funzioni 321
 impostazione delle proprietà 325
 modifica dei flussi 325
 nomi proprietà 324
 ottenimento di proprietà 325
 panoramica 321
 repository 328
 riferimenti a nodi 324
 sovrapposte 324
 tipi di modello 324
 tipi di nodo 324
 tipi di output 324
 varie 329
modellazione di database 253
modelli
 nomi di script 317, 319
modelli Apriori
 proprietà script dei nodi 163, 239
modelli Apriori Oracle
 proprietà script dei nodi 257, 263
modelli Bayes adattivi Oracle
 proprietà script dei nodi 257, 263
modelli C&R Tree
 proprietà script dei nodi 176, 242
modelli C5.0
 proprietà script dei nodi 174, 241
modelli CARMA
 proprietà script dei nodi 175, 242
modelli causali temporali
 proprietà script dei nodi 228
modelli CHAID
 proprietà script dei nodi 178, 242
Modelli Classificatore automatico
 proprietà script dei nodi 240
Modelli Cluster automatico
 proprietà script dei nodi 241

modelli dell'elemento adiacente più vicino
 proprietà script dei nodi 200
 modelli di associazione IBM ISW
 proprietà script dei nodi 264, 269
 Modelli di raggruppamento cluster divisivo Netezza
 proprietà script dei nodi 270, 280
 modelli di raggruppamento cluster IBM ISW
 proprietà script dei nodi 264, 269
 modelli di regressione di Cox
 proprietà script dei nodi 180, 243
 modelli di regressione IBM ISW
 proprietà script dei nodi 264, 269
 modelli di regressione lineare
 proprietà script dei nodi 218, 248, 249
 Modelli di regressione lineare Netezza
 proprietà script dei nodi 270, 280
 modelli di regressione logistica
 proprietà script dei nodi 205, 246
 modelli di regressione logistica IBM ISW
 proprietà script dei nodi 264, 269
 modelli di rete bayesiana
 proprietà script dei nodi 172
 modelli di selezione funzioni
 applicazione 4
 proprietà script dei nodi 186, 244
 script 4
 modelli di sequenza IBM ISW
 proprietà script dei nodi 264, 269
 modelli di serie storiche Netezza
 proprietà script dei nodi 270
 modelli di serie temporali
 proprietà script dei nodi 231, 250
 Modelli di serie temporali IBM ISW
 proprietà script dei nodi 264
 modelli di struttura ad albero delle decisioni IBM ISW
 proprietà script dei nodi 264, 269
 Modelli di struttura ad albero delle decisioni Netezza
 proprietà script dei nodi 270, 280
 Modelli di struttura ad albero di regressione Netezza
 proprietà script dei nodi 270, 280
 modelli discriminanti
 proprietà script dei nodi 183, 243
 modelli Elenco di decisioni
 proprietà script dei nodi 182, 243
 Modelli fattoriali/PCA
 proprietà script dei nodi 185, 243
 modelli generati
 nomi di script 317, 319
 Modelli GLE
 proprietà script dei nodi 195
 modelli GLMM
 proprietà script dei nodi 191, 244
 modelli IBM DB2
 proprietà script dei nodi 264
 Modelli K-medie Oracle
 proprietà script dei nodi 257
 modelli KNN
 proprietà script dei nodi 245
 Modelli KNN Netezza
 proprietà script dei nodi 270, 280
 modelli Kohonen
 proprietà script dei nodi 202, 245
 modelli linear-AS
 proprietà script dei nodi 204, 246
 modelli lineari
 proprietà script dei nodi 203, 246
 modelli lineari generalizzati
 proprietà script dei nodi 188, 244
 modelli lineari generalizzati Netezza
 proprietà script dei nodi 270
 modelli lineari generalizzati Oracle
 proprietà script dei nodi 257
 modelli LSVM
 proprietà script dei nodi 210
 modelli LSVM (linear support vector machine)
 proprietà script dei nodi 210, 246
 modelli MDL Oracle
 proprietà script dei nodi 257, 263
 modelli Medie K
 proprietà script dei nodi 199, 245
 modelli Medie K Netezza
 proprietà script dei nodi 270, 280
 modelli Microsoft
 proprietà script dei nodi 253, 255
 Modelli Naive Bayes IBM ISW
 proprietà script dei nodi 264, 269
 Modelli Naive Bayes Netezza
 proprietà script dei nodi 270, 280
 modelli Naive Bayes Oracle
 proprietà script dei nodi 257, 263
 Modelli Netezza
 proprietà script dei nodi 270
 modelli NMF Oracle
 proprietà script dei nodi 257, 263
 modelli numerici automatici
 proprietà script dei nodi 170
 Modelli Numerici automatici
 proprietà script dei nodi 241
 modelli Oracle
 proprietà script dei nodi 257
 Modelli Oracle AI
 proprietà script dei nodi 257
 Modelli PCA
 proprietà script dei nodi 185, 243
 Modelli PCA Netezza
 proprietà script dei nodi 270, 280
 modelli QUEST
 proprietà script dei nodi 215, 247
 modelli Random Trees
 proprietà script dei nodi 217, 248
 Modelli rete di Bayes Netezza
 proprietà script dei nodi 270, 280
 modelli Rete neurale
 proprietà script dei nodi 211, 247
 modelli Rilevamento anomalie
 proprietà script dei nodi 161, 239
 modelli Risposta autoapprendimento
 proprietà script dei nodi 222, 249
 modelli Sequenza
 proprietà script dei nodi 220, 249
 modelli SLRM
 proprietà script dei nodi 222, 249
 Modelli struttura ad albero delle decisioni Oracle
 proprietà script dei nodi 257, 263
 modelli support vector machine
 proprietà script dei nodi 249
 modelli SVM
 proprietà script dei nodi 227
 modelli SVM Oracle
 proprietà script dei nodi 257, 263
 modelli tcm
 proprietà script dei nodi 250
 modelli Tree-AS
 proprietà script dei nodi 233, 251
 modelli TwoStep
 proprietà script dei nodi 235, 251
 modelli TwoStep AS
 proprietà script dei nodi 236, 251
 modello di contenuto JSON 56
 modello di contenuto tabella 53
 modello di contenuto XML 54
 Modello GLE
 proprietà script dei nodi 245
 modifica dei flussi 31, 33

N

nodi
 collegamento di nodi 31
 eliminazione 32
 esecuzione di cicli sugli script 49
 importazione 32
 informazioni 34
 riferimento nomi 317
 scollamento di nodi 31
 sostituzione 32
 nodi di esportazione
 proprietà script dei nodi 299
 nodi di modellazione
 proprietà script dei nodi 161
 nodi Grafici
 proprietà script 145
 nodi origine
 proprietà 75
 nodi output
 proprietà script 283
 nodo Accodamento
 proprietà 99
 nodo Adattamento della simulazione
 proprietà 292
 nodo Aggregazione
 proprietà 99
 nodo Aggregazione RFM
 proprietà 106
 nodo Analisi
 proprietà 283
 nodo Analisi RFM
 proprietà 131
 nodo Anonimizza
 proprietà 115
 nodo bilanciamento
 proprietà 100
 nodo Calcola globali
 proprietà 290
 nodo Campione
 proprietà 108
 nodo Classificatore automatico
 proprietà script dei nodi 166
 Nodo Cluster automatico
 proprietà script dei nodi 169

- nodo Crea flag
 - proprietà 132
 - nodo Creazione R
 - proprietà script dei nodi 173
 - nodo Cronologia
 - proprietà 127
 - nodo Database
 - proprietà 81
 - nodo del grafico temporale
 - proprietà 156
 - nodo dell'insieme
 - proprietà 124
 - nodo Deriva
 - proprietà 122
 - Nodo di esportazione del database
 - proprietà 301
 - Nodo di esportazione IBM SPSS Data Collection
 - proprietà 305
 - Nodo di esportazione IBM Statistics
 - proprietà 313
 - Nodo di output IBM SPSS Statistics
 - proprietà 312
 - nodo di riproiezione
 - proprietà 130
 - nodo distribuzione
 - proprietà 147
 - nodo Elimina duplicati
 - proprietà 103
 - nodo Enterprise View
 - proprietà 86
 - nodo Esplora
 - proprietà 284
 - nodo Esporta SAS
 - proprietà 307
 - nodo Esporta XML
 - proprietà 309
 - nodo Esportazione da Excel
 - proprietà 305
 - nodo File flat
 - proprietà 306
 - nodo Filtro
 - proprietà 126
 - nodo Genera simulazione
 - proprietà 90
 - nodo Input utente
 - proprietà 93
 - nodo Intervalli di tempo
 - proprietà 133
 - nodo Intervalli di tempo AS
 - proprietà 119
 - nodo Istogramma
 - proprietà 152
 - nodo Lavagna grafica
 - proprietà 150
 - nodo Matrice
 - proprietà 285
 - nodo Medie
 - proprietà 287
 - nodo Multiplot
 - proprietà 153
 - nodo Ordina
 - proprietà 110
 - nodo origine Excel
 - proprietà 85
 - nodo origine geospaziale
 - proprietà 89
 - Nodo origine IBM Cognos BI
 - proprietà 79
 - nodo origine IBM Cognos TM1
 - proprietà 92
 - Nodo origine IBM SPSS Data Collection
 - proprietà 83
 - Nodo origine IBM SPSS Statistics
 - proprietà 311
 - nodo origine SAS
 - proprietà 89
 - Nodo origine Server analitici
 - proprietà 79
 - nodo origine Vista dati
 - proprietà 97
 - Nodo origine XML
 - proprietà 97
 - nodo Output R
 - proprietà 290
 - nodo Partizione
 - proprietà 127
 - nodo Plot
 - proprietà 154
 - nodo Raccolta
 - proprietà 119, 146
 - nodo Regole di associazione
 - proprietà 164
 - nodo Report
 - proprietà 289
 - nodo Ricodifica
 - proprietà 128
 - nodo Riempimento
 - proprietà 125
 - nodo Riordina
 - proprietà 129
 - nodo Riordina campi
 - proprietà 129
 - nodo Riorganizza
 - proprietà 130
 - nodo Seleziona
 - proprietà 110
 - nodo Statistiche
 - proprietà 292
 - Nodo STB (Space-Time-Boxes)
 - proprietà 101
 - nodo STP
 - proprietà 223
 - nodo STP (Spatio-Temporal Prediction)
 - proprietà 223
 - nodo Tabella
 - proprietà 294
 - nodo Testo fisso
 - proprietà 87
 - nodo Testo variabile
 - proprietà 94
 - nodo Tipo
 - proprietà 138
 - nodo Trasformazioni
 - proprietà 296
 - Nodo Trasformazioni IBM SPSS Statistics
 - proprietà 311
 - nodo Trasformazioni R
 - proprietà 107
 - nodo Trasponi
 - proprietà 137
 - nodo Unione
 - proprietà 104
 - nodo Valutazione
 - proprietà 148
 - nodo Valutazione simulazione
 - proprietà 291
 - nodo Web
 - proprietà 157
 - nodo Web diretto
 - proprietà 157
 - nomi di campo
 - modifica di caratteri maiuscoli/minuscoli 49
 - nugget
 - proprietà script dei nodi 239
 - nugget del modello
 - nomi di script 317, 319
 - proprietà script dei nodi 239
 - nugget del nodo STP
 - proprietà 250
 - nugget nodo regole di associazione
 - proprietà 240
- ## O
- O-Cluster Oracle
 - proprietà script dei nodi 257, 263
 - oggetti di output
 - nomi di script 319
 - oggetti modello
 - nomi di script 317, 319
 - operazioni 16
 - ordine di esecuzione
 - modifica con script 49
 - ordine di esecuzione del flusso
 - modifica con script 49
 - orientata agli oggetti 23
- ## P
- parametri 5, 67, 68, 71
 - script 16
 - Supernodi 315
 - parametri di slot 5, 67, 69
 - parola chiave generated 52
 - passaggio degli argomenti 20
 - password
 - aggiunta a script 51
 - codifica 64
 - password codificata
 - aggiunta a script 51
 - preparazione automatica dati
 - proprietà 116
 - proprietà
 - flusso 71
 - nodi Filtro 67
 - nodi Modelli database 253
 - script 67, 68, 69, 161, 239, 299
 - script comuni 69
 - Supernodi 315
 - Proprietà aggregatenode 99
 - proprietà analysisnode 283
 - proprietà anomalydetectionnode 161
 - proprietà anonymizenode 115
 - proprietà appendnode 99
 - proprietà
 - applyanomalydetectionnode 239
 - proprietà applyapriorinode 239

proprietà applyassociationrulesnode 240
 proprietà applyautoclassifiernode 240
 proprietà applyautoclusternode 241
 proprietà applyautonumericnode 241
 proprietà applybayesnetnode 241
 proprietà applyc50node 241
 proprietà applycarmanode 242
 proprietà applycartnode 242
 proprietà applychaidnode 242
 proprietà applycoxregnode 243
 proprietà applydb2imclusternode 269
 proprietà applydb2imlognode 269
 proprietà applydb2imnbnode 269
 proprietà applydb2imregnode 269
 proprietà applydb2imtreenode 269
 proprietà applydecisionlistnode 243
 proprietà applydiscriminantnode 243
 proprietà applyfactornode 243
 proprietà applyfeatureselectionnode 244
 proprietà
 applygeneralizedlinearnode 244
 proprietà applyggle 245
 proprietà applyglmnode 244
 proprietà applykmeansnode 245
 proprietà applyknnnode 245
 proprietà applykohonennode 245
 proprietà applylinearasnode 246
 proprietà applylinearnode 246
 proprietà applylogregnode 246
 proprietà applylsvmnnode 246
 proprietà applymslogisticnode 255
 proprietà
 applymsneuralnetworknode 255
 proprietà applymsregressionnode 255
 proprietà
 applymssequenceclusternode 255
 proprietà applymstimeseriesnode 255
 proprietà applymstreenode 255
 proprietà applynetezزابayesnode 280
 proprietà applynetezžadectreenode 280
 proprietà
 applynetezžadivclusternode 280
 proprietà applynetezzakmeansnode 280
 proprietà applynetezzaknnode 280
 proprietà
 applynetezزالineregressionnode 280
 proprietà
 applynetezzanaivebayesnode 280
 proprietà applynetezzapcanode 280
 proprietà applynetezzareggreenode 280
 proprietà applyneuralnetnode 247
 proprietà applyneuralnetworknode 247
 proprietà applyoraabnode 263
 proprietà applyoradecisiontreenode 263
 proprietà applyorakmeansnode 263
 proprietà applyoranbnode 263
 proprietà applyoranmfnode 263
 proprietà applyoraoclusternode 263
 proprietà applyorasvmnode 263
 proprietà applyquestnode 247
 proprietà applyrandomtrees 248
 proprietà applyregressionnode 249
 proprietà applyselflearningnode 249
 proprietà applysequencenode 249
 proprietà applystpnode 250
 proprietà applysvmnnode 249
 proprietà applytcmnode 250
 proprietà applytimeseriesnode 250
 proprietà applytreeas 251
 proprietà applytwestepAS 251
 proprietà applytwestepnode 251
 proprietà apriorinode 163
 proprietà asexport 299
 proprietà asimport 79
 proprietà associationrulesnode 164
 proprietà astimeintervalsnode 119
 proprietà autoclassifiernode 166
 proprietà autoclusternode 169
 proprietà autodataprepnode 116
 proprietà autonumericnode 170
 proprietà balancenode 100
 proprietà bayesnet 172
 proprietà binningnode 119
 proprietà buildr 173
 proprietà c50node 174
 proprietà carmanode 175
 proprietà cartnode 176
 proprietà chaidnode 178
 proprietà collectionnode 146
 proprietà coxregnode 180
 proprietà dataauditnode 284
 proprietà databaseexportnode 301
 Proprietà databasenode 81
 proprietà datacollectionexportnode 305
 proprietà datacollectionimportnode 83
 proprietà dataviewimport 97
 proprietà db2imassocnode 264
 proprietà db2imclusternode 264
 proprietà db2imlognode 264
 proprietà db2imnbnode 264
 proprietà db2imregnode 264
 proprietà db2imsequencenode 264
 proprietà db2imtimeseriesnode 264
 proprietà db2imtreenode 264
 proprietà decisionlist 182
 proprietà del nodo cognosimport 79
 proprietà del nodo gsdata_import 89
 proprietà del nodo Ricodifica 128
 Proprietà del nodo STB
 (Space-Time-Boxes) 101
 proprietà del nodo tm1import 92
 Proprietà derivenode 122
 proprietà di applyr 248
 proprietà directedwebnode 157
 proprietà discriminantnode 183
 proprietà distinctnode 103
 proprietà distributionnode 147
 proprietà ensemblenode 124
 proprietà evaluationnode 148
 proprietà evimportnode 86
 proprietà excelexportnode 305
 proprietà excelimportnode 85
 proprietà factornode 185
 proprietà featureselectionnode 4, 186
 proprietà fillernode 125
 proprietà filternode 126
 proprietà fixedfilenode 87
 proprietà flatfilenode 306
 proprietà genlinnode 188
 proprietà gle 195
 proprietà glmnode 191
 proprietà graphboardnode 150
 proprietà histogramnode 152
 proprietà historynode 127
 proprietà kmeansnode 199
 proprietà knnnode 200
 proprietà kohonennode 202
 proprietà linear-AS 204
 proprietà lineari 203
 proprietà logregnode 205
 proprietà lsvmnnode 210
 proprietà matrixnode 285
 proprietà meansnode 287
 proprietà mergenode 104
 proprietà msassocnode 253
 proprietà msbayesnode 253
 proprietà msclusternode 253
 proprietà mslogisticnode 253
 proprietà msneuralnetworknode 253
 proprietà msregressionnode 253
 proprietà msssequenceclusternode 253
 proprietà mstimeseriesnode 253
 proprietà mstreenode 253
 proprietà multiplotnode 153
 proprietà netezزابayesnode 270
 proprietà netezžadectreenode 270
 proprietà netezžadivclusternode 270
 proprietà netezzaglmnode 270
 proprietà netezzakmeansnode 270
 proprietà netezzaknnode 270
 proprietà netezزالineregressionnode 270
 proprietà netezzanaivebayesnode 270
 proprietà netezzapcanode 270
 proprietà netezzareggreenode 270
 proprietà netezzatimeseriesnode 270
 proprietà neuralnetnode 211
 proprietà neuralnetworknode 213
 proprietà numericpredictornode 170
 Proprietà oraabnode 257
 proprietà oraainode 257
 proprietà oraapriorinode 257
 proprietà oradecisiontreenode 257
 proprietà oraglmnode 257
 proprietà orakmeansnode 257
 proprietà oramdlnode 257
 Proprietà oranbnode 257
 proprietà oranmfnode 257
 proprietà oraoclusternode 257
 proprietà orasvmnode 257
 proprietà outputfilenode 306
 proprietà partitionnode 127
 proprietà plotnode 154
 proprietà questnode 215
 proprietà randomtrees 217
 proprietà regressionnode 218
 proprietà reordernode 129
 proprietà reportnode 289
 proprietà reprojectnode 130
 proprietà restructurenode 130
 proprietà rfmaggregatenode 106
 proprietà rfanalysisnode 131
 proprietà routputnode 290
 proprietà Rprocessnode 107
 proprietà samplenode 108
 proprietà sasexportnode 307
 proprietà sasimportnode 89
 proprietà script dei nodi 253
 nodi di esportazione 299
 nodi di modellazione 161
 nugget del modello 239
 proprietà selectnode 110

- proprietà sequencenode 220
- proprietà setglobalsnode 290
- proprietà settoflagnode 132
- proprietà simevalnode 291
- proprietà simfitnode 292
- proprietà simgenode 90
- proprietà slrmnode 222
- proprietà sortnode 110
- Proprietà statisticsexportnode 313
- proprietà statisticsimportnode 4, 311
- proprietà statisticsmodelnode 312
- proprietà statisticsnode 292
- proprietà statisticsoutputnode 312
- proprietà statisticstransformnode 311
- proprietà stpnode 223
- proprietà stream.nodes 49
- proprietà streamingts 110
- proprietà strutturate 67
- proprietà svmnode 227
- proprietà tablenode 294
- proprietà tcmnode 228
- proprietà timeintervalsnode 133
- proprietà timeplotnode 156
- proprietà timeseriesnode 231
- proprietà transformnode 296
- proprietà transposenode 137
- proprietà treeas 233
- proprietà twostepAS 236
- proprietà twostepnode 235
- proprietà typenode 4, 138
- proprietà userinputnode 93
- proprietà variablefilenode 94
- proprietà webnode 157
- proprietà xmlexportnode 309
- proprietà xmlimportnode 97
- Python 15
 - script 16

R

- Regressione lineare MS
 - proprietà script dei nodi 253, 255
- Regressione logistica MS
 - proprietà script dei nodi 253, 255
- repository IBM SPSS Analytic Server
 - argomenti della riga di comando 65
- Rete bayesiana, modelli
 - proprietà script dei nodi 241
- Rete neurale MS
 - proprietà script dei nodi 253, 255
- reti neurali
 - proprietà script dei nodi 213, 247
- retrieve, comando 50
- ricerca di nodi 29
- riferimento ai nodi 29
 - impostazione di proprietà 30
- ricerca di nodi 29
- riga di comando
 - elenco di argomenti 62, 64, 65
 - esecuzione di IBM SPSS Modeler 61
 - parametri 63
 - più argomenti 66
 - script 52
- riproiezione del sistema di coordinate
 - proprietà 130

S

- script
 - abbreviazioni utilizzate 68
 - chiave di iterazione 8
 - compatibilità con versioni
 - precedenti 52
 - contesto 28
 - controllo degli errori 51
 - dalla riga di comando 52
 - diagrammi 27
 - esecuzione 11
 - esecuzione condizionale 6, 10
 - esecuzione di cicli 6, 7
 - esecuzione di cicli visiva 6, 7
 - flussi 1
 - Flussi SuperNodo 27
 - flusso 27
 - importazione da file di testo 1
 - interfaccia utente 1, 4, 5
 - interruzione 11
 - modelli di selezione funzioni 4
 - nei Supernodi 5
 - nodi di output 283
 - nodi Grafici 145
 - ordine di esecuzione del flusso 49
 - panoramica 1, 15
 - proprietà comuni 69
 - salvataggio 1
 - script autonomi 1, 27
 - script del Supernodo 1, 27
 - script legacy 322, 323, 326, 329
 - script Python 322, 323, 326, 329
 - selezione campi 10
 - sintassi 16, 17, 19, 20, 21, 22, 23, 24, 25
 - variabile di iterazione 9
- script autonomi 1, 4, 27
- Serie temporali Microsoft
 - proprietà script dei nodi 255
- server
 - argomenti della riga di comando 64
- sicurezza
 - password codificata 51, 64
- sistema
 - argomenti della riga di comando 62
- sovrapposte
 - script 16
- store, comando 50
- Streaming del nodo serie temporale
 - proprietà 110
- stringhe 17
 - modifica di caratteri
 - maiuscoli/minuscoli 49
- Struttura ad albero delle decisioni MS
 - proprietà script dei nodi 253, 255
- Supernodi
 - flusso 27
 - impostazione delle proprietà 315
 - parametri 315
 - proprietà 315
 - script 1, 5, 6, 27, 315
- Supernodo 67
 - flusso 27

V

- variabile di iterazione
 - esecuzione di cicli negli script 9
- variabili nascoste 25



Stampato in Italia