

*Guide de génération de scripts Python
et d'automatisation IBM SPSS Modeler
16*

IBM

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 261.

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
Direction Qualité
17, avenue de l'Europe
92275 Bois-Colombes Cedex*

Cette version s'applique à IBM SPSS Modeler 16.0.0 et à toutes les publications et modifications ultérieures jusqu'à mention contraire dans les nouvelles versions.

Table des matières

Avis aux lecteurs canadiens vii

Chapitre 1. Scripts 1

Génération de scripts - Présentation	1
Types de script	1
Scripts de flux	2
Scripts autonomes	3
Scripts de super noeud	3
Exécution en boucle et conditionnelle dans les flux	4
Bouclage dans les flux	5
Exécution conditionnelle dans les flux	8
Exécution et interruption de scripts	9
Rechercher et remplacer	10

Chapitre 2. Langage de script 13

Présentation du langage de script	13
Python et Jython	13
Scripts Python	14
Opérations	14
Listes	14
Chaînes.	15
Remarques	17
Syntaxe des instructions	17
Identificateurs	17
Blocs de code.	17
Transmission d'arguments à un script.	18
Exemples	18
Méthodes mathématiques.	19
Utilisation des caractères non ASCII	20
Programmation orientée objet	21
Définition d'une classe.	22
Création d'une instance de classe	22
Ajout d'attributs à une instance de classe	22
Définition d'attributs de classe et de méthodes	23
Variables masquées.	23
Héritage	23

Chapitre 3. Génération de scripts dans IBM SPSS Modeler 25

Types de scripts	25
Flux, flux super noeud et diagrammes	25
Flux	25
Flux super noeud	25
Diagrammes	25
Exécution d'un flux.	25
Contexte de génération de scripts	26
Référence aux noeuds existants	27
Recherche de noeuds	27
Définition des propriétés	28
Création de noeuds et modification de flux.	28
Création de noeuds.	29
Création et suppression de liens entre les noeuds	29
Importation, remplacement et suppression de noeuds	30
Traversée des noeuds d'un flux.	31

Informations sur les noeuds 32

Chapitre 4. API de scriptage 35

Introduction à l'API de scriptage	35
Exemple : recherche de noeuds à l'aide d'un filtre personnalisé	35
Métadonnées : informations sur les données	35
Accès aux objets générés	38
Traitement des erreurs.	39
Paramètres de flux, de session et de super noeud.	40
Valeurs globales	44
Utilisation de plusieurs flux : scripts autonomes	45

Chapitre 5. Conseils pour la génération de scripts 47

Modification de l'exécution du flux	47
Utilisation de modèles.	47
Génération d'un mot de passe codé	47
Vérification du script	48
Génération de scripts à partir de la ligne de commande	48
Spécification de chemins de fichier	48
Compatibilité avec les versions précédentes	48

Chapitre 6. Arguments de ligne de commande 51

Appel du logiciel	51
Utilisation d'arguments de ligne de commande	51
Arguments système.	52
Arguments de paramètre	53
Arguments de connexion au serveur	54
IBM SPSS Collaboration and Deployment Services Repository Arguments de connexion	55
Combinaison de plusieurs arguments.	55

Chapitre 7. Référence sur les propriétés 57

Présentation des références sur les propriétés	57
Abréviations	57
Exemple de propriétés de noeud et de flux	57
Présentation des propriétés de noeud.	58
Propriétés communes des noeuds	58

Chapitre 8. Propriétés de flux 59

Chapitre 9. Propriétés des noeuds source 63

Propriétés communes aux noeuds source.	63
Propriétés du noeud asimport	65
Propriétés du noeud cognosimport	65
Propriétés du noeud SGBD	67
Propriétés du noeud datacollectionimport	68
Propriétés du noeud excelimport	70

Propriétés du noeud evimport	71
Propriétés du noeud fixedfile	71
Propriétés du noeud sasimport	74
Propriétés du noeud simgen	74
Propriétés du noeud statisticsimport	77
Propriétés du noeud userinput	77
Propriétés du noeud variablefile	78
Propriétés du noeud xmlimport	81

Chapitre 10. Propriétés des noeuds d'opérations sur les lignes 83

Propriétés du noeud Ajouter.	83
Propriétés du noeud Agréger	83
Propriétés du noeud Equilibrer	84
Propriétés du noeud derive_stb.	84
Propriétés du noeud Distinguer	86
Propriétés du noeud Fusionner	86
Propriétés du noeud rfmaggregate.	87
Propriétés du noeud Rprocess	89
Propriétés du noeud Echantillon	89
Propriétés du noeud Sélectionner	91
Propriétés du noeud Trier	91
Propriétés du noeud streamingts	91

Chapitre 11. Propriétés des noeuds d'opérations sur les champs 95

Propriétés du noeud Anonymiser	95
Propriétés du noeud autodataprep.	95
Propriétés du noeud Discrétiser	98
Propriétés du noeud Dériver	101
Propriétés du noeud Ensemble	102
Propriétés du noeud Remplacer	103
Propriétés du noeud Filtrer	103
Propriétés du noeud Historiser	104
Propriétés du noeud Partitionner	104
Propriétés du noeud Recoder	105
Propriétés du noeud Re-trier	106
Propriétés du noeud Restructurer.	106
Propriétés du noeud rfmanalysis	107
Propriétés du noeud settoflag	108
Propriétés du noeud statisticstransform.	108
Propriétés du noeud timeintervals	109
Propriétés du noeud Transposer	113
Propriétés du noeud type	113

Chapitre 12. Propriétés des noeuds Graphiques 119

Propriétés communes aux noeuds Graphiques	119
Propriétés du noeud Résumé	120
Propriétés du noeud distribution	121
Propriétés du noeud Evaluation	121
Propriétés du noeud Représentation Graphique	123
Propriétés du noeud Histogramme	125
Propriétés du noeud Courbes	126
Propriétés du noeud Tracé	127
Propriétés du noeud timeplot	129
Propriétés du noeud Relations.	130

Chapitre 13. Propriétés des noeuds de modélisation. 131

Propriétés communes des noeuds de modélisation	131
Propriétés du noeud anomalydetection	131
Propriétés du noeud apriori	133
Propriétés du noeud autoclassifier	134
Définition des propriétés de l'algorithme	135
Propriétés du noeud autocluster	136
Propriétés du noeud autonumeric	137
Propriétés du noeud bayesnet	138
Propriétés du noeud buildr.	139
Propriétés du noeud c50.	140
Propriétés du noeud Carma	141
Propriétés du noeud cart	142
Propriétés du noeud chaid	143
Propriétés du noeud coxreg.	145
Propriétés du noeud decisionlist	146
Propriétés du noeud discriminant	148
Propriétés du noeud ACP/Facteur	149
Propriétés du noeud featureselection	150
Propriétés du noeud genlin.	152
Propriétés du noeud MMLG	155
Propriétés du noeud kmeans	159
Propriétés du noeud knn	160
Propriétés du noeud Kohonen	161
Propriétés du noeud Linéaire	162
Propriétés du noeud logreg.	163
Propriétés du noeud neuralnet	166
Propriétés du noeud neuralnetwork	168
Propriétés du noeud quest	169
Propriétés du noeud Régression	170
Propriétés du noeud Séquence.	172
Propriétés du noeud MRAA	173
Propriétés du noeud statisticsmodel	173
Propriétés du noeud svm	174
Propriétés du noeud timeseries	174
Propriétés du noeud twostep	176

Chapitre 14. Propriétés du noeud de nugget de modèle 179

Propriétés du noeud applyanomalydetection	179
Propriétés du noeud applyapriori	179
Propriétés du noeud applyautoclassifier	180
Propriétés du noeud applyautocluster	180
Propriétés du noeud applyautonumeric.	180
Propriétés du noeud applybayesnet	181
Propriétés du noeud applyc50.	181
Propriétés du noeud applycarma	181
Propriétés du noeud applycart	181
Propriétés du noeud applychaid	182
Propriétés du noeud applycoxreg.	182
Propriétés du noeud applydecisionlist	183
Propriétés du noeud applydiscriminant.	183
Propriétés du noeud applyfactor	183
Propriétés du noeud applyfeatureselection.	183
Propriétés du noeud applygeneralizedlinear	184
Propriétés du noeud applyglm	184
Propriétés du noeud applykmeans	185
Propriétés du noeud applyknn	185
Propriétés du noeud applykohonen	185

Propriétés du noeud applylinear	185
Propriétés du noeud applylogreg	185
Propriétés du noeud applyneuralnet	186
Propriétés du noeud applyneuralnetwork	186
Propriétés du noeud applyquest	186
Propriétés du noeud applyregression	187
Propriétés du noeud applyr	187
Propriétés du noeud applyselflearning	187
Propriétés du noeud applysequence	188
Propriétés du noeud applysvm	188
Propriétés du noeud applytimeseries	188
Propriétés du noeud applytwestep	188

Chapitre 15. Propriétés du nœud de modélisation SGBD. 189

Propriétés du nœud pour la modélisation Microsoft	189
Propriétés des nœuds de modélisation Microsoft	189
Propriétés du nugget de modèle Microsoft	191
Propriétés du noeud pour la modélisation Oracle	193
Propriétés du noeud de modélisation Oracle	193
Propriétés du nugget de modèle Oracle	198
Propriétés de nœud pour la modélisation IBM DB2	199
Propriétés du noeud de modélisation IBM DB2	199
Propriétés du nugget de modèle IBM DB2.	204
Propriétés du noeud pour la modélisation IBM	
Netezza Analytics	205
Propriétés des nœuds de modélisation Netezza	205
Propriétés du nugget de modèle Netezza	214

Chapitre 16. Propriétés des noeuds de sortie 217

Propriétés du noeud Analyse	217
Propriétés du noeud dataaudit	218
Propriétés du noeud Matrice	219
Propriétés du noeud Moyennes	220
Propriétés du noeud Rapport	222
Propriétés du noeud Routput	222
Propriétés du noeud V. globales	223
Propriétés du noeud simeval	223
Propriétés du noeud simfit	224
Propriétés du noeud Statistiques	225
Propriétés du noeud statisticsoutput.	226
Propriétés du noeud Table	226
Propriétés du noeud Transformation.	228

Chapitre 17. Propriétés du nœud d'exportation 231

Propriétés communes des nœuds Exportation	231
Propriétés du noeud asexport	231
Propriétés du noeud cognosexport	232
Propriétés du noeud databaseexport.	233
Propriétés du noeud datacollectionexport	236
Propriétés du nœud excelexport	236

Propriétés du noeud outputfile	237
Propriétés du noeud sasexport	238
Propriétés du noeud statisticsexport	238
Propriétés du noeud xmlexport	238

Chapitre 18. Propriétés de noeuds IBM SPSS Statistics 241

Propriétés du noeud statisticsimport.	241
Propriétés du noeud statisticstransform.	241
Propriétés du noeud statisticsmodel	242
Propriétés du noeud statisticsoutput.	242
Propriétés du noeud statisticsexport	242

Chapitre 19. Propriétés du super noeud 245

Annexe A. Référence des noms de noeuds 247

Noms des nuggets de modèle	247
Pour éviter les noms de modèle en double	249
Nom des types de sortie.	249

Annexe B. Migration du scriptage existant au scriptage Python. 251

Présentation de la migration de script existant	251
Différences générales	251
Contexte de scriptage)	251
Commandes et fonctions	251
Littéraux et commentaires	252
Opérateurs	252
Commandes conditionnelles et de bouclage	253
Variables	254
Types de noeuds, de sorties et de modèles.	254
Noms de propriétés	254
Références de noeud	254
Extraction et définition de propriétés	255
Edition de flux	255
Opérations de noeud	256
Bouclage	257
exécution des flux	257
Accéder aux objets via le système de fichiers et le référentiel	258
Opérations de flux	259
Opérations de modèle	259
Opérations de sortie de document	259
Autres différences entre le scriptage existant et le scriptage Python	260

Remarques 261

Marques	263
-------------------	-----

Index 265

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.

OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
⌘ (Pos1)	⌘	Home
Fin	Fin	End
⇧ (PgAr)	⇧	PgUp
⇩ (PgAv)	⇩	PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
🔒 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

Chapitre 1. Scripts

Génération de scripts - Présentation

La génération de scripts IBM® SPSS Modeler est un outil performant pour automatiser les processus dans l'interface utilisateur. Les scripts permettent d'effectuer les mêmes opérations qu'avec la souris ou le clavier. Vous pouvez les utiliser pour automatiser les tâches dont l'exécution manuelle s'avère très répétitive et très longue.

Vous pouvez utiliser les scripts pour :

- Imposer un ordre spécifique d'exécution des noeuds dans un flux et exécuter sous condition des noeuds si des conditions d'exécution à remplir ont été définies.
- Créer des boucles afin d'exécuter de façon répétitive des noeuds dans un flux.
- Définir l'exécution automatique d'une séquence d'actions qui nécessite normalement l'intervention de l'utilisateur (création et test d'un modèle, par exemple).
- Mettre en place des processus complexes dans lesquels l'intervention de l'utilisateur est importante (procédures de validation croisée qui exigent la création et le test de nombreux modèles, par exemple).
- Mettre en place des processus de gestion des flux (exécution d'un flux d'apprentissage de modèle et production automatique du flux de test modèle correspondant, par exemple).

Ce chapitre offre des descriptions et des exemples de haut niveau de scripts de flux, de scripts autonomes et de scripts dans les super noeuds dans l'interface IBM SPSS Modeler. D'autres informations sur la syntaxe, les commandes et le langage de script vous sont fournies dans les chapitres qui suivent.¹

Remarque : vous ne pouvez pas importer et exécuter des scripts créés dans IBM SPSS Statistics avec IBM SPSS Modeler.

Types de script

IBM SPSS Modeler utilise trois types de script :

- **Les scripts de flux** sont stockés en tant que propriété de flux et sont, par conséquent, sauvegardés et chargés avec un flux spécifique. Par exemple, vous pouvez écrire un script de flux qui automatise le processus de formation et d'application d'un nugget de modèle. Vous pouvez également préciser que lorsqu'un script particulier s'exécute, ce script doit être exécuté à la place du contenu de l'espace de travail de flux.
- **Lesscripts autonomes** ne sont pas associés à un flux en particulier et sont sauvegardés dans des fichiers texte externes. Par exemple, vous pouvez utiliser un script autonome pour manipuler plusieurs flux ensemble.
- **Lesscripts de super noeud** sont stockés en tant que propriété de flux de super noeud. Les scripts de super noeud sont uniquement accessibles dans les super noeuds terminaux. Vous pouvez utiliser un script de super noeud pour contrôler la séquence d'exécution des contenus de super noeuds. Pour les super noeuds qui ne sont pas terminaux (source ou processus), vous pouvez définir les propriétés du super noeud ou des noeuds qu'il contient directement dans votre script de flux.

1. Le langage de script existant IBM SPSS Modeler peut encore être utilisé avec IBM SPSS Modeler 16. Pour plus d'informations, consultez le document *Guide de génération de scripts et d'automatisation IBM SPSS Modeler 16*. Voir Annexe B, «Migration du scriptage existant au scriptage Python», à la page 251 pour obtenir des conseils sur le mappage de vos scripts IBM SPSS Modeler existants vers des scripts Python.

Scripts de flux

Les scripts peuvent être utilisés pour personnaliser les opérations réalisées au sein d'un flux particulier, et ils peuvent être enregistrés avec ce flux. Les scripts de flux peuvent être utilisés pour spécifier un ordre d'exécution particulier des noeuds terminaux dans un flux. La boîte de dialogue du script de flux permet d'éditer le script qui est enregistré avec le flux en cours.

Pour accéder à l'onglet du script de flux dans la boîte de dialogue Propriétés du flux :

1. Dans le menu Outils, sélectionnez :
Propriétés du flux > Exécution
2. Cliquez sur l'onglet **Exécution** pour utiliser les scripts du flux en cours.
3. Sélectionnez le mode d'exécution : **Par défaut (script facultatif)**.

Les icônes de la barre d'outils se trouvant en haut de la boîte de dialogue du script de flux vous permettent d'effectuer les opérations suivantes :

- Importer les contenus d'un script autonome préexistant dans la fenêtre.
- Sauvegarder un script comme fichier texte.
- Imprimer un script.
- Ajouter le script par défaut.
- Editer un script (annuler, couper, copier, coller et autres fonctions d'édition courantes).
- Exécuter entièrement le script en cours.
- Exécuter les lignes sélectionnées d'un script.
- Arrêter un script pendant l'exécution. (Cette icône est activée uniquement lorsqu'un script est en cours d'exécution.)
- Vérifier la syntaxe d'un script et, en cas d'erreurs, les afficher dans le panneau inférieur de la boîte de dialogue.

De plus, vous pouvez spécifier si ce script doit ou non être exécuté lorsque le flux est exécuté. Vous pouvez sélectionner **Exécuter ce script** pour exécuter le script chaque fois que le flux est exécuté en respectant l'ordre d'exécution du script. Ce paramètre permet l'automatisation au niveau du flux et l'accélération de la vitesse de création des modèles. Cependant, le paramètre par défaut est d'ignorer ce script pendant l'exécution du flux. Même si vous sélectionnez l'option **Ignorer ce script**, vous pouvez quand même exécuter ce script directement depuis la boîte de dialogue.

Vous pouvez également décider de modifier le type de scriptage en passant du scriptage Python au scriptage existant.

L'éditeur de script inclut les fonctions suivantes qui facilitent la création de script :

- Mise en évidence de la syntaxe ; les mots-clés, les valeurs littérales (telles que les chaînes et les nombres) et les commentaires sont mis en évidence.
- Numérotation des lignes.
- Mise en correspondance des blocs ; lorsque le curseur est placé au début d'un bloc de programme, le bloc de fin correspondant est également mis en évidence.
- Suggestion de saisie automatique.

Vous pouvez personnaliser les couleurs et les styles de texte utilisés par le surligneur de syntaxe via les préférences d'affichage IBM SPSS Modeler. Pour accéder aux préférences d'affichage, sélectionnez **Outils > Options > Options utilisateur**, puis cliquez sur l'onglet **Syntaxe**.

Vous pouvez accéder à une liste de suggestions syntaxiques. Pour cela, sélectionnez **Suggestion automatique** dans le menu contextuel ou appuyez sur Ctrl + Espace. Utilisez les touches du curseur pour

vous déplacer vers le haut ou vers le bas dans la liste, puis appuyez sur Entrée pour insérer le texte sélectionné. Appuyez sur la touche Echap pour quitter le mode Suggestion automatique sans modifier le texte existant.

L'onglet **Débogage** affiche des messages de débogage et peut servir à évaluer l'état du script après l'exécution du script. L'onglet **Débogage** est composé d'une zone de texte en lecture seule et d'un champ de texte d'entrée de ligne unique. La zone de texte affiche le texte qui est envoyé soit à la sortie standard, par exemple via la commande Python `print`, soit à l'erreur standard par les scripts, par exemple via le texte de message d'erreur. Le champ de texte d'entrée utilise l'entrée de l'utilisateur. Cette entrée est ensuite évaluée dans le contexte du script le plus récemment exécuté dans la boîte de dialogue (appelé *contexte de génération de scripts*). La zone de texte contient la commande et le résultat de sortie pour que l'utilisateur ait une trace des commandes. Le champ de texte d'entrée contient toujours l'invite de commande (`>>>` pour le scriptage Python).

Un nouveau contexte de génération de scripts est créé dans les circonstances suivantes :

- Un script est exécuté à l'aide du bouton "Exécuter ce script" ou "Exécuter les lignes sélectionnées".
- Le langage de script est modifié.

Si un nouveau contexte de génération de scripts est créé, le contenu de la zone de texte est effacé.

Remarque : L'exécution d'un flux en dehors du panneau de script ne modifiera pas le contexte de script du panneau de script. Les valeurs des variables créées dans le cadre de cette exécution ne seront pas visibles dans la boîte de dialogue du script.

Scripts autonomes

Dans la boîte de dialogue Script autonome, vous pouvez créer ou éditer un script sauvegardé en tant que fichier texte. Cette boîte de dialogue indique le nom du fichier et fournit des fonctions pour charger, enregistrer, importer et exécuter des scripts.

Pour accéder à la boîte de dialogue du script autonome :

Dans le menu principal, sélectionnez :

Outils > Script autonome

Les scripts autonomes disposent de la même barre d'outils et des mêmes options de vérification de syntaxe de script que les scripts de flux. Pour plus d'informations, voir la rubrique «Scripts de flux», à la page 2.

Scripts de super noeud

Vous pouvez créer et sauvegarder des scripts dans tous les super noeuds terminaux avec le langage de script de IBM SPSS Modeler. Ces scripts sont uniquement accessibles pour les super noeuds terminaux et sont souvent utilisés lors de la création de modèles de flux ou pour imposer un ordre d'exécution particulier aux contenus des super noeuds. Les scripts de super noeud permettent également d'exécuter plusieurs scripts dans un même flux.

Par exemple, imaginons que vous avez besoin de préciser l'ordre d'exécution d'un flux complexe et que votre super noeud contient plusieurs noeuds qui comprennent un noeud v. globales qui doit être exécuté avant de calculer un nouveau champ utilisé dans un noeud Tracé. Dans ce cas, vous pouvez créer un script de super noeud qui exécute d'abord le noeud v. globales. Les valeurs calculées par ce noeud, telles que la moyenne ou l'écart-type, peuvent être utilisées lorsque le noeud Tracé est exécuté.

Dans le script de super noeud, vous pouvez spécifier les propriétés du noeud de la même façon que les autres scripts. Ou alors, vous pouvez modifier et définir les propriétés de n'importe quel super noeud ou de ses noeuds encapsulés directement depuis un script de flux. Pour plus d'informations, voir la rubrique Chapitre 19, «Propriétés du super noeud», à la page 245. Cette méthode fonctionne pour les super noeuds de source et de processus ainsi que pour les super noeuds terminaux.

Remarque : puisque seuls les super noeuds terminaux peuvent exécuter leurs propres scripts, l'onglet Scripts de la boîte de dialogue Super noeud est uniquement accessible pour les super noeuds terminaux.

Pour ouvrir la boîte de dialogue de script des super noeuds depuis l'espace de travail principal :

Sélectionnez un super noeud terminal dans l'espace de travail de flux et, dans le menu Super noeud, choisissez :

Script Super noeud...

Pour ouvrir la boîte de dialogue de script des super noeuds depuis l'espace de travail de super noeud en zoom avant :

Cliquez avec le bouton droit de la souris sur l'espace de travail de super noeud et, dans le menu contextuel, sélectionnez :

Script Super noeud...

Exécution en boucle et conditionnelle dans les flux

A compter de la version 16.0, SPSS Modeler vous permet de créer des scripts de base à partir d'un flux en sélectionnant des valeurs dans diverses boîtes de dialogue au lieu de devoir écrire des instructions directement dans le langage de script. Les deux principaux types de scripts que vous pouvez créer de cette façon sont les boucles simples et l'exécution de noeuds si une condition prédéfinie est remplie.

Vous pouvez combiner les règles d'exécution en boucle et les règles d'exécution conditionnelle dans un flux. Par exemple, vous pouvez avoir des données relatives aux ventes de voitures des fabricants du monde entier. Vous pouvez définir une boucle visant à traiter les données dans un flux, en identifiant les informations détaillées par pays de fabrication, et obtenir des résultats sous forme de graphiques montrant des informations détaillées telles que les volumes de ventes par modèle, les niveaux d'émissions par constructeur et par taille de moteur, etc. Si vous ne souhaitez analyser que les données européennes, vous pouvez aussi ajouter au bouclage des conditions empêchant la création de graphiques pour les constructeurs basées en Amérique et en Asie.

Remarque : Etant donné que l'exécution conditionnelle et l'exécution en boucle sont toutes deux basées sur des scripts d'arrière-plan, elles sont uniquement appliquées à un flux complet lors de son exécution.

- **Bouclage** Vous pouvez utiliser le bouclage (exécution en boucle) pour automatiser les tâches répétitives. Par exemple, cela peut signifier ajouter un nombre de noeuds donné à un flux et modifier un paramètre de noeud à chaque fois. Vous pouvez aussi contrôler l'exécution d'un flux ou créer des branches successives un certain nombre de fois, comme dans les exemples suivants :
 - Exécuter le flux un certain nombre de fois et modifier la source à chaque fois.
 - Exécuter le flux un certain nombre de fois en modifiant la valeur d'une variable à chaque fois.
 - Exécuter le flux un certain nombre de fois en entrant un champ supplémentaire à chaque exécution.
 - Générer un modèle un certain nombre de fois et modifier son paramétrage à chaque fois.
- **Exécution conditionnelle** Cette option vous permet de contrôler l'exécution de noeuds terminaux en fonction de conditions prédéfinies ; voici des exemples :
 - Vous pouvez définir si un noeud sera exécuté ou non en fonction de la valeur true ou false d'une valeur définie.

- Vous pouvez définir si l'exécution en boucle des noeuds sera effectuée en mode parallèle ou séquentiel.

Vous définissez l'exécution en boucle et l'exécution conditionnelle dans l'onglet Exécution de la boîte de dialogue Propriétés de flux. Les noeuds qui sont utilisés dans les exécutions en boucle ou conditionnelles sont affichés associés à un symbole supplémentaire dans le canevas de flux afin d'indiquer qu'ils prennent part à l'exécution en boucle ou conditionnelle.

Vous pouvez accéder à l'onglet Exécution de trois manières :

- En utilisant les menus situés en haut de la boîte de dialogue principale :
 1. Dans le menu Outils, sélectionnez :
Propriétés du flux > Exécution
 2. Cliquez sur l'onglet Exécution pour utiliser les scripts du flux en cours.
- A partir d'un flux :
 1. Cliquez avec le bouton droit de la souris sur un noeud et sélectionnez **Exécution en boucle/conditionnelle**.
 2. Sélectionnez l'option de sous-menu appropriée.
- A partir de la barre d'outils graphiques située en haut de la boîte de dialogue principale, cliquez sur l'icône des propriétés de flux.

Si c'est la première fois que vous définissez les détails d'une exécution en boucle ou conditionnelle, dans l'onglet Exécution, sélectionnez le mode d'exécution **Exécution en boucle/conditionnelle** puis sélectionnez le sous-onglet **Conditionnel** ou **Bouclage**.

Bouclage dans les flux

Le bouclage (ou exécution en boucle) vous permet d'automatiser les tâches répétitives dans les flux ; par exemple :

- Exécuter le flux un certain nombre de fois et modifier la source à chaque fois.
- Exécuter le flux un certain nombre de fois en modifiant la valeur d'une variable à chaque fois.
- Exécuter le flux un certain nombre de fois en entrant un champ supplémentaire à chaque exécution.
- Générer un modèle un certain nombre de fois et modifier son paramétrage à chaque fois.

Vous définissez les conditions à remplir dans le sous-onglet **Bouclage** de l'onglet d'exécution du flux. Pour afficher ce sous-onglet, sélectionnez le mode d'exécution **Exécution en boucle/conditionnelle**.

Les modalités d'exécution en boucle que vous définissez prendront effet lors de l'exécution du flux, si le mode d'exécution **Exécution en boucle/conditionnelle** a été défini. Vous pouvez facultativement générer le code de script des modalités d'exécution en boucle et le coller dans l'éditeur de script en cliquant sur **Coller...** dans l'angle inférieur droit du sous-onglet Bouclage. L'affichage de l'onglet principal Exécution change afin de présenter le mode d'exécution **Par défaut (script facultatif)** avec le script dans la partie supérieure de l'onglet. Cela signifie que vous pouvez définir une structure de bouclage en utilisant les diverses options de la boîte de dialogue de bouclage avant de générer un script que vous pouvez personnaliser ultérieurement dans l'éditeur de script. Notez que lorsque vous cliquez sur **Coller...**, les conditions conditionnelle que vous avez définies s'affichent aussi dans le script généré.

Pour définir une boucle :

1. Créez une clé d'itération pour définir la structure de bouclage principale à utiliser dans un flux. Pour plus d'informations, voir Créer une clé d'itération.
2. Lorsque cela est nécessaire, définissez une ou plusieurs variables d'itération. Pour plus d'informations, voir Créer une variable d'itération.
3. Les itérations et les variables que vous avez créées sont affichées dans le corps principal du sous-onglet. Par défaut, les itérations sont exécutées dans l'ordre dans lequel elles apparaissent. Pour

déplacer une itération vers le haut ou le bas de la liste, cliquez dessus pour la sélectionner puis utilisez les flèches vers le haut ou le bas situées dans la colonne de droite du sous-onglet pour modifier l'ordre.

Création d'une clé d'itération destinée au bouclage dans les flux

Vous utilisez une clé d'itération pour définir la principale structure de bouclage à utiliser dans un flux. Par exemple, si vous analysez des ventes de voitures, vous pouvez créer un paramètre de flux *Pays de fabrication* et l'utiliser comme clé d'itération. Lorsque le flux est exécuté, cette clé est définie successivement sur chaque valeur de pays lors de chaque itération. Utilisez la boîte de dialogue Définir la clé d'itération pour configurer la clé.

Pour ouvrir cette boîte de dialogue, sélectionnez le bouton **Clé d'itération...** dans l'angle inférieur gauche du sous-onglet Bouclage ou cliquez avec le bouton droit de la souris sur un noeud du flux et sélectionnez **Exécution en boucle/conditionnelle > Définir la clé d'itération (champs)** ou **Exécution en boucle/conditionnelle > Définir la clé d'itération (valeurs)**. Si vous ouvrez la boîte de dialogue à partir du flux, certains des champs peuvent avoir été renseignés automatiquement (par exemple, le nom du noeud).

Pour définir une clé d'itération, renseignez les champs suivants :

Itérer sur. Vous pouvez sélectionner l'une des options suivantes :

- **Paramètre de flux - Champs.** Utilisez cette option pour créer une boucle qui définit la valeur d'un paramètre de flux existant successivement sur chaque champ spécifié.
- **Paramètre de flux - Valeurs.** Utilisez cette option pour créer une boucle qui définit la valeur d'un paramètre de flux existant successivement sur chaque valeur spécifiée.
- **Propriété de noeud - Champs.** Utilisez cette option pour créer une boucle qui définit la valeur d'une propriété de noeud successivement sur chaque champ spécifié.
- **Propriété de noeud - Valeurs.** Utilisez cette option pour créer une boucle qui définit la valeur d'une propriété de noeud successivement sur chaque valeur spécifiée.

Élément à définir. Sélectionnez l'élément dont la valeur sera définie à chaque exécution de la boucle. Vous pouvez sélectionner l'une des options suivantes :

- **Paramètre.** Uniquement disponible si vous sélectionnez **Paramètre de flux - Champs** ou **Paramètres de flux - Valeurs**. Sélectionnez le paramètre requis dans la liste disponible.
- **Noeud.** Uniquement disponible si vous sélectionnez **Propriété de noeud - Champs** ou **Propriété de noeud - Valeurs**. Sélectionnez le noeud pour lequel vous souhaitez définir une boucle. Cliquez sur le bouton Parcourir pour ouvrir la boîte de dialogue de sélection de noeud et choisissez le noeud souhaité ; si la liste contient trop de noeuds, vous pouvez la filtrer afin de n'afficher que certains types de noeuds en sélectionnant l'une des catégories suivantes : Source, Processus, Graphique, Modélisation, Sortie, Exporter ou Appliquer les noeuds modèle.
- **Propriété.** Uniquement disponible si vous sélectionnez **Propriété de noeud - Champs** ou **Propriété de noeud - Valeurs**. Sélectionnez la propriété du noeud dans la liste disponible.

Champs à utiliser. Uniquement disponible si vous sélectionnez **Paramètre de flux - Champs** ou **Propriété de noeud - Champs**. Sélectionnez le ou les champs (dans un noeud) à utiliser pour fournir les valeurs d'itération. Vous pouvez sélectionner l'une des options suivantes :

- **Noeud.** Uniquement disponible si vous sélectionnez **Paramètre de flux - Champs**. Sélectionnez le noeud qui contient les détails pour lesquels vous souhaitez définir une boucle. Cliquez sur le bouton Parcourir pour ouvrir la boîte de dialogue de sélection de noeud et choisissez le noeud souhaité ; si la liste contient trop de noeuds, vous pouvez la filtrer afin de n'afficher que certains types de noeuds en sélectionnant l'une des catégories suivantes : Source, Processus, Graphique, Modélisation, Sortie, Exporter ou Appliquer les noeuds modèle.

- **Liste de champs.** Cliquez sur le bouton de listage dans la colonne de droite pour afficher la boîte de dialogue Sélectionner les champs, au sein de laquelle vous sélectionnez les champs du noeud qui fourniront les données d'itération. Pour plus d'informations, voir «Sélection de champs pour les itérations», à la page 8.

Valeur à utiliser. Uniquement disponible si vous sélectionnez **Paramètre de flux - Valeurs** ou **Propriété de noeud - Valeurs**. Sélectionnez la ou les valeurs du champ sélectionné à utiliser comme valeurs d'itération. Vous pouvez sélectionner l'une des options suivantes :

- **Noeud.** Uniquement disponible si vous sélectionnez **Paramètre de flux - Valeurs**. Sélectionnez le noeud qui contient les détails pour lesquels vous souhaitez définir une boucle. Cliquez sur le bouton Parcourir pour ouvrir la boîte de dialogue de sélection de noeud et choisissez le noeud souhaité ; si la liste contient trop de noeuds, vous pouvez la filtrer afin de n'afficher que certains types de noeuds en sélectionnant l'une des catégories suivantes : Source, Processus, Graphique, Modélisation, Sortie, Exporter ou Appliquer les noeuds modèle.
- **Liste de champs.** Sélectionnez le champ du noeud qui fournira les données d'itération.
- **Liste de valeurs.** Cliquez sur le bouton de listage dans la colonne de droite pour afficher la boîte de dialogue Sélectionner les valeurs, au sein de laquelle vous sélectionnez les valeurs du noeud qui fourniront les données d'itération.

Création d'une variable d'itération destinée au bouclage dans les flux

Vous pouvez utiliser les variables d'itération pour modifier les valeurs des paramètres ou propriétés de flux des noeuds sélectionnés dans un flux à chaque fois exécution d'une boucle. Par exemple, si la boucle de flux analyse les données de ventes de voitures et utilise *Pays de fabrication* comme clé d'itération, vous pouvez obtenir en sortie un graphique montrant les ventes par modèle et un autre graphique montrant les informations sur les émissions de gaz d'échappement. Dans cet exemple, vous pouvez créer des variables d'itération créant de nouveaux titres pour les graphiques résultants, tels que *Emissions des véhicules suédois* et *Ventes de voitures japonaises par modèle*. Utilisez la boîte de dialogue Définir la variable d'itération pour configurer les variables dont vous avez besoin.

Pour ouvrir cette boîte de dialogue, sélectionnez le bouton **Variable d'itération...** dans l'angle inférieur gauche du sous-onglet Bouclage ou cliquez avec le bouton droit de la souris sur un noeud du flux et sélectionnez **Exécution en boucle/conditionnelle > Définir la variable d'itération**.

Pour définir une variable d'itération, renseignez les champs suivants :

Modifier. Sélectionnez le type d'attribut à modifier. Vous avez le choix entre **Paramètre de flux** et **Propriété de noeud**.

- Si vous sélectionnez **Paramètre de flux**, choisissez le paramètre qui vous convient, puis, à l'aide de l'une des options suivantes (si elle est disponible dans le flux), définissez la valeur que devra avoir le paramètre avec chaque itération de la boucle :
 - **Variable globale.** Sélectionnez la variable globale que le paramètre de flux devra avoir comme valeur.
 - **Cellule de table de résultats.** Pour définir le paramètre de flux qui devra constituer la valeur d'une cellule de table de résultats, sélectionnez la table dans la liste et entrez les **Ligne** et **Colonne** à utiliser.
 - **Saisir manuellement.** Sélectionnez cette option si vous souhaitez entrer manuellement une valeur que ce paramètre devra prendre à chaque itération. Lorsque vous revenez au sous-onglet Bouclage, une nouvelle colonne est créée et vous permet de saisir le texte requis.
- Si vous sélectionnez **Propriété de noeud**, sélectionnez le noeud requis et l'une de ses propriétés, puis définissez la valeur à utiliser pour cette propriété. Définissez la nouvelle valeur de propriété à l'aide de l'une des options suivantes :
 - **Seul.** La valeur de propriété utilisera la valeur de la clé d'itération. Pour plus d'informations, voir «Création d'une clé d'itération destinée au bouclage dans les flux», à la page 6.

- **Comme préfixe à la racine.** Utilisez la valeur de la clé d'itération comme préfixe de ce que vous entrez dans le champ **Racine**.
- **Comme suffixe à la racine.** Utilisez la valeur de la clé d'itération comme suffixe de ce que vous entrez dans le champ **Racine**.

Si vous sélectionnez l'option de préfixe ou de suffixe, vous êtes invité à ajouter un texte supplémentaire dans le champ **Racine**. Par exemple, si la valeur de la clé d'itération est *Pays de fabrication* et que vous sélectionnez **Comme préfixe à la racine**, vous pouvez entrer - *ventes par modèle* dans ce champ.

Sélection de champs pour les itérations

Lorsque vous créez des itérations, vous pouvez sélectionner un ou plusieurs champs à l'aide de la boîte de dialogue Sélectionner les champs.

Trier par. Vous pouvez trier les champs disponibles en consultation en sélectionnant l'une des options suivantes :

- **Naturel.** Affichez les champs dans l'ordre dans lequel ils ont été transmis dans le flux de données au noeud en cours.
- **Nom.** Utilisez l'ordre alphabétique pour trier les champs à afficher.
- **Type.** Affichez les champs triés en fonction de leur niveau de mesure. Cette option est utile lorsque vous sélectionnez des champs avec un niveau de mesure particulier.

Sélectionnez les champs dans la liste un par un ou utilisez les méthodes Maj+clic et Ctrl+clic pour sélectionner plusieurs champs. Vous pouvez aussi utiliser les boutons situés sous la liste pour sélectionner des groupes de champs en fonction de leur niveau de mesure ou sélectionner ou désélectionner tous les champs de la table.

Notez que les champs disponibles en sélection sont filtrés pour n'afficher que ceux qui conviennent au paramètre de flux ou à la propriété de noeud que vous utilisez. Par exemple, si vous utilisez un paramètre de flux possédant un type de stockage Chaîne, seuls les champs possédant le même type de stockage sont affichés.

Exécution conditionnelle dans les flux

L'exécution conditionnelle vous permet de contrôler l'exécution des noeuds terminaux en définissant des conditions pour le contenu du flux ; voici des exemples :

- Vous pouvez définir si un noeud sera exécuté ou non en fonction de la valeur true ou false d'une valeur définie.
- Vous pouvez définir si l'exécution en boucle des noeuds sera effectuée en mode parallèle ou séquentiel.

Vous définissez les conditions à remplir dans le sous-onglet **Conditionnel** de l'onglet d'exécution du flux. Pour afficher ce sous-onglet, sélectionnez le mode d'exécution **Exécution en boucle/conditionnelle**.

Les modalités d'exécution conditionnelle que vous définissez prendront effet lors de l'exécution du flux, si le mode d'exécution **Exécution en boucle/conditionnelle** a été défini. Vous pouvez facultativement générer le code de script des modalités d'exécution conditionnelle et le coller dans l'éditeur de script en cliquant sur **Coller...** dans l'angle inférieur droit du sous-onglet Conditionnel. L'affichage de l'onglet principal Exécution change afin de présenter le mode d'exécution **Par défaut (script facultatif)** avec le script dans la partie supérieure de l'onglet. Cela signifie que vous pouvez définir les conditions en utilisant les diverses options de la boîte de dialogue de bouclage avant de générer un script que vous pouvez personnaliser ultérieurement dans l'éditeur de script. Notez que lorsque vous cliquez sur **Coller...**, les conditions de bouclage que vous avez définies s'affichent aussi dans le script généré.

Pour définir une condition :

1. Dans la colonne de droite du sous-onglet Conditionnel, cliquez sur le bouton d'ajout d'une instruction d'exécution  pour ouvrir la boîte de dialogue Instruction d'exécution conditionnelle. Cette boîte de dialogue vous permet de spécifier la condition qui doit être remplie pour que le noeud soit exécuté.
2. Dans la boîte de dialogue Instruction d'exécution conditionnelle, indiquez les informations suivantes :
 - a. **Noeud.** Sélectionnez le noeud pour lequel vous souhaitez définir une exécution conditionnelle. Cliquez sur le bouton Parcourir pour ouvrir la boîte de dialogue de sélection de noeud et choisissez le noeud souhaité ; si la liste contient trop de noeuds, vous pouvez la filtrer afin d'afficher les noeuds en fonction de l'une des catégories suivantes : Exporter, Graphique, Modélisation ou Noeud sortie.
 - b. **Condition basée sur.** Indiquez la condition qui doit être remplie pour que le noeud soit exécuté. Vous avez le choix entre quatre options : **Paramètre de flux**, **Variable globale**, **Cellule de table de résultats** ou **Toujours vraie**. Les détails que vous entrez dans la moitié inférieure de la boîte de dialogue sont définis par la condition que vous choisissez.
 - **Paramètre de flux.** Sélectionnez le paramètre dans la liste disponible, puis sélectionnez l'**opérateur** de ce paramètre ; par exemple, l'opérateur peut être Supérieur à, Egal, Inférieur à, Entre, etc. Vous entrez ensuite la **valeur** ou les valeurs minimale et maximale, selon l'opérateur.
 - **Variable globale.** Sélectionnez la variable dans la liste disponible ; il peut s'agir par exemple de Moyenne, Somme, Valeur minimale, Valeur maximale ou Ecart type. Sélectionnez ensuite l'**opérateur** et les valeurs requises.
 - **Cellule de table de résultats.** Sélectionnez le noeud table dans la liste disponible, puis sélectionnez la **ligne** et la **colonne** dans la table. Sélectionnez ensuite l'**opérateur** et les valeurs requises.
 - **Toujours vraie.** Sélectionnez cette option si le noeud doit toujours être exécuté. Si vous sélectionnez cette option, il n'y a aucun autre paramètre à sélectionner.
3. Répétez les étapes 1 et 2 aussi souvent que nécessaire jusqu'à ce que vous ayez configuré toutes les conditions souhaitées. Le noeud que vous avez sélectionné et la condition à remplir pour que le noeud soit exécuté sont affichés dans le corps principal du sous-onglet dans les colonnes **Exécuter un noeud** et **Si cette condition est vraie**.
4. Par défaut, les noeuds et conditions sont exécutés dans l'ordre dans lequel ils apparaissent. Pour déplacer un noeud et une condition vers le haut ou le bas de la liste, cliquez dessus pour le sélectionner puis utilisez les flèches vers le haut ou le bas situées dans la colonne de droite du sous-onglet pour modifier l'ordre.

En outre, vous pouvez définir les options suivantes au bas du sous-onglet Conditionnel :

- **Evaluer tout dans l'ordre.** Sélectionnez cette option pour évaluer les conditions dans l'ordre dans lequel elles apparaissent dans le sous-onglet. Les noeuds pour lesquels les conditions ont la valeur "True" sont tous exécutés une fois toutes les conditions évaluées.
- **Exécuter un à un.** Uniquement disponible si **Evaluer tout dans l'ordre** est sélectionné. Lorsque cette option est sélectionnée, si une condition est évaluée comme ayant la valeur "True", le noeud associé à cette condition est exécuté avant que la condition suivante ne soit évaluée.
- **Evaluer jusqu'au premier résultat.** Lorsque vous sélectionnez cette option, seul le premier noeud qui renvoie une évaluation "True" à partir des conditions spécifiées s'exécute.

Exécution et interruption de scripts

Plusieurs manières d'exécuter les scripts sont disponibles. Par exemple, dans la boîte de dialogue du script de flux ou du script autonome, le bouton « Exécuter ce script » permet d'exécuter l'intégralité du script :



Figure 1. Bouton Exécuter ce script

Le bouton « Exécuter les lignes sélectionnées » exécute une ligne unique ou un bloc de lignes adjacentes que vous avez sélectionnées dans le script :



Figure 2. Bouton Exécuter les lignes sélectionnées

Pour exécuter un script, utilisez l'une des méthodes suivantes :

- Cliquer sur le bouton « Exécuter ce script » ou sur le bouton « Exécuter les lignes sélectionnées » dans la boîte de dialogue du script de flux ou du script autonome.
- Exécutez un flux dans lequel la méthode d'exécution par défaut est **Exécuter ce script**.
- Utilisez l'indicateur -execute au démarrage en mode interactif. Pour plus d'informations, voir la rubrique «Utilisation d'arguments de ligne de commande», à la page 51.

Remarque : un script associé à un super noeud est exécuté en même temps que ce dernier si vous avez sélectionné **Exécuter ce script** dans la boîte de dialogue de script du super noeud.

Interruption de l'exécution d'un script

Dans la boîte de dialogue du script de flux, le bouton rouge d'arrêt est activé pendant l'exécution du script. Ce bouton permet d'abandonner l'exécution du script et de tout flux actuel.

Rechercher et remplacer

La boîte de dialogue Rechercher/Remplacer est disponible aux emplacements où vous modifiez un script ou le texte d'une expression (par exemple, l'éditeur de script) ou encore lors de la définition d'un modèle dans le noeud Rapport. Lorsque vous éditez du texte dans l'un de ces champs, appuyez sur Ctrl+F pour accéder à la boîte de dialogue, en vous assurant que le curseur est centré sur une zone de texte. Si vous travaillez dans un noeud Remplacer, par exemple, vous pouvez accéder à la boîte de dialogue depuis toute zone de texte de l'onglet Paramètres, ou depuis le champ de texte du générateur d'expression.

1. Lorsque le curseur se trouve sur une zone de texte, appuyez sur Ctrl+F pour accéder à la boîte de dialogue Rechercher/Remplacer.
2. Entrez le texte que vous souhaitez rechercher ou faites un choix dans la liste déroulante des éléments récemment consultés.
3. Saisissez éventuellement un texte de remplacement.
4. Cliquez sur **Suivant** pour lancer la recherche.
5. Cliquez sur **Remplacer** pour remplacer la sélection courante, ou **Remplacer tout** pour mettre à jour certaines instances ou les instances sélectionnées.
6. La boîte de dialogue se ferme après chaque opération. Appuyez sur F3 depuis toute zone de texte pour répéter la dernière opération de recherche ou sur Ctrl+F pour accéder de nouveau à la boîte de dialogue.

Options de recherche

Respecter la casse. Spécifie si l'opération de recherche est sensible à la casse ; par exemple, si *myvar* correspond à *myVar*. Le texte de remplacement est toujours inséré exactement tel qu'il a été saisi, quel que soit le réglage de ce paramètre.

Mot entier. Spécifie si l'opération de recherche doit porter sur le texte inséré dans des mots. Si cette option est sélectionnée, par exemple, une recherche portant sur *spider* ne produira pas la réponse *spiderman* ou *spider-man*.

Caractères génériques. Spécifie si la syntaxe des caractères génériques est utilisée (voir la section suivante). Lorsqu'elle est sélectionnée, l'option **Mot entier** est désactivée et sa valeur est ignorée.

Texte sélectionné. Contrôle la portée de la recherche lorsque vous utilisez l'option **Remplacer tout**.

Syntaxe des caractères génériques

Les caractères génériques vous permettent de rechercher des caractères spéciaux tels que les tabulations ou les sauts de ligne, des classes ou des intervalles de caractères telles que *a* à *d*, toute valeur numérique ou non et les limites telles que le début ou la fin d'une ligne. Un modèle d'expression régulière décrit la structure de la chaîne que l'expression tentera de rechercher dans une chaîne d'entrée. Voici les types de syntaxe d'expression régulière pris en charge :

Tableau 1. Correspondances de caractères

Caractères	Correspondances
x	Le caractère x
\\	Le caractère barre oblique inversée
\0n	Le caractère présentant la valeur octale 0n (0 <= n <= 7)
\0nn	Le caractère présentant la valeur octale 0nn (0 <= n <= 7)
\0mnn	Le caractère présentant la valeur octale 0mnn (0 <= m <= 3, 0 <= n <= 7)
\xhh	Le caractère présentant la valeur hexadécimale 0xhh
\uhhhh	Le caractère présentant la valeur hexadécimale 0xhhhh
\t	Le caractère tabulation ('\u0009')
\n	Le caractère saut de ligne (retour à la ligne) ('\u000A')
\r	Le caractère retour chariot ('\u000D')
\f	Le caractère alimentation de formulaire ('\u000C')
\a	Le caractère alerte (sonnerie) ('\u0007')
\e	Le caractère d'échappement ('\u001B')
\cx	Le caractère de contrôle correspondant à x

Tableau 2. Classes de caractères correspondantes

Classes de caractères	Correspondances
[abc]	a, b ou c (classe simple)
[^abc]	Tout caractère excepté a, b ou c (soustraction)
[a-zA-Z]	Caractères de a à z ou de A à Z compris (intervalle)
[a-d[m-p]]	Caractères de a à d ou de m à p (union). Cette option peut aussi être spécifiée comme [a-dm-p]
[a-z&&[def]]	Caractères de a à z ainsi que d, e ou f (intersection)
[a-z&&[^bc]]	Caractères de a à z à l'exception de b et c (soustraction). Cette option peut aussi être spécifiée comme [ad-z]
[a-z&&[^m-p]]	Caractères de a à z, mais pas de m à p (soustraction). Cette option peut aussi être spécifiée comme [a-lq-z]

Tableau 3. Classes de caractères prédéfinies

Classes de caractères prédéfinies	Correspondances
.	Tout caractère (peut correspondre ou non aux terminaisons de ligne)
\d	Tout chiffre : [0-9]
\D	Un caractère non numérique : [^0-9]
\s	Un espace blanc : [\t\n\r\f]
\S	Un espace non blanc : [^\s]
\w	Un mot : [a-zA-Z_0-9]
\W	Un caractère autre qu'un mot : [^\w]

Tableau 4. Correspondances de limites

Correspondances de limites	Correspondances
^	Le début d'une ligne
\$	La fin d'une ligne
\b	Une limite de mot
\B	Une limite autre que celle d'un mot
\A	Le début de la saisie
\Z	La fin de la saisie à l'exception de la terminaison finale éventuelle
\z	La fin de la saisie

Pour plus d'informations sur l'utilisation des expressions régulières et pour obtenir quelques exemples, voir <http://www.ibm.com/developerworks/java/tutorials/j-introjava2/section9.html>.

Exemples

Le code suivant recherche et correspond aux trois nombres en début de chaîne :

```
^[0-9]{3}
```

Le code suivant recherche et correspond aux trois nombres en fin de chaîne :

```
[0-9]{3}$
```

Chapitre 2. Langage de script

Présentation du langage de script

La fonction de scriptage (génération de scripts) pour IBM SPSS Modeler vous permet de créer des scripts qui fonctionnent sur l'interface utilisateur SPSS Modeler, manipulent des objets de sortie et exécutent la syntaxe de commande. Vous pouvez exécuter des scripts directement depuis SPSS Modeler.

Les scripts d'IBM SPSS Modeler sont écrits dans le langage de script Python. L'implémentation Java de Python qui est utilisée par IBM SPSS Modeler est appelée Jython. Le langage de script est composé des fonctions suivantes :

- un format pour le référencement des noeuds, des flux, des projets, des sorties, ainsi que d'autres objets IBM SPSS Modeler ;
- Un ensemble d'instructions ou de commandes de script qui permettent de manipuler ces objets.
- Un langage d'expression de script pour le paramétrage des valeurs des variables, paramètres et autres objets.
- Une prise en charge des commentaires, des lignes incomplètes et des blocs de texte littéral.

Les sections suivantes décrivent le langage de script Python, l'implémentation Jython de Python et la syntaxe de base pour s'initier au scriptage dans IBM SPSS Modeler. Des informations sur les propriétés et commandes spécifiques vous sont fournies dans les sections suivantes.

Python et Jython

Jython est une implémentation du langage de script Python, qui est écrit dans le langage Java et intégré à la plateforme Java. Python est un puissant langage de script orienté objet. Jython est utile car il offre les fonctions de productivité d'un langage de script abouti et, contrairement à Python, s'exécute dans tout environnement prenant en charge une machine virtuelle Java (JVM). Ainsi, les bibliothèques Java de la machine virtuelle Java peuvent être utilisées lorsque vous écrivez des programmes. Avec Jython, vous pouvez tirer profit de cette différence et utiliser la syntaxe ainsi que la plupart des fonctions du langage Python.

En tant que langage de script, Python (et son implémentation Jython) est facile à comprendre, permet un codage efficace et ne requiert qu'une structure minimale pour créer un programme d'exécution. Le code peut être saisi de manière interactive, c'est-à-dire une ligne à la fois. Python est un langage de script interprété ; il n'existe aucune étape de précompilation, comme c'est le cas dans Java. Les programmes Python sont simplement des fichiers textes interprétés lors de leur entrée (après l'analyse syntaxique recherchant les erreurs de syntaxe). Les expressions simples, telles que les valeurs définies, ainsi que les actions plus complexes, telles que les définitions de fonctions, sont immédiatement exécutées et utilisables. Les modifications apportées au code peuvent être testées rapidement. Toutefois, l'interprétation de script présente certains inconvénients. Par exemple, l'utilisation d'une variable non définie n'est pas une erreur de compilateur. Ainsi, elle est détectée uniquement si (et lorsque) l'instruction dans laquelle est utilisée la variable est exécutée. Dans ce cas, le programme peut être modifié et exécuté pour déboguer l'erreur.

Python perçoit tout, y compris l'ensemble du code et des données, comme un objet. Par conséquent, vous pouvez manipuler ces objets avec des lignes de code. Certains types, tels que les nombres et les chaînes, sont plus aisément considérés comme des valeurs, et non comme des objets ; Python assure cette prise en charge. Une seule valeur nulle est prise en charge. Cette valeur nulle porte le nom réservé None (aucun).

Pour une introduction plus approfondie des scripts Python et Jython et pour obtenir des exemples de scripts, voir www.ibm.com/developerworks/java/tutorials/j-jython1 et www.ibm.com/developerworks/java/tutorials/j-jython2.

Scripts Python

Ce guide relatif au langage de script Python est une introduction aux composants les plus susceptibles d'être utilisés lors de la génération de scripts (scriptage) dans IBM SPSS Modeler, notamment les concepts et les notions de base de la programmation. Ce guide vous apportera les connaissances nécessaires pour commencer à développer vos propres scripts Python à utiliser dans IBM SPSS Modeler.

Opérations

L'affectation s'effectue en utilisant le signe égal (=). Par exemple, pour affecter la valeur "3" à une variable appelée "x", utilisez l'instruction suivante :

```
x = 3
```

Le signe égal permet également d'affecter des données de type chaîne à une variable. Par exemple, pour affecter la valeur "a string value" à la variable "y", utilisez l'instruction suivante :

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

Le tableau suivant répertorie certaines des opérations numériques et de comparaison courantes ainsi que leurs descriptions.

Tableau 5. Opérations numériques et de comparaison courantes

Opération	Description
$x < y$	x est-il inférieur à y ?
$x > y$	x est-il supérieur à y ?
$x \leq y$	x est-il inférieur ou égal à y ?
$x \geq y$	x est-il supérieur ou égal à y ?
$x == y$	x est-il égal à y ?
$x != y$	x est-il différent de y ?
$x \lt;> y$	x est-il différent de y ?
$x + y$	Ajoute y à x
$x - y$	Soustrait y de x
$x * y$	Multiplie x par y
x / y	Divise x par y
$x ** y$	Elève x à la puissance y

Listes

Les listes sont des séquences d'éléments. Une liste peut contenir tout nombre d'éléments, et les éléments de la liste peuvent correspondre à tout type d'objet. Les listes peuvent également être considérées comme des tableaux. Le nombre d'éléments dans une liste peut augmenter ou diminuer au fur et à mesure que des éléments sont ajoutés, supprimés ou remplacés.

Exemples

[] Une liste vide.
[1] Une liste avec un seul élément : un entier.

```
["Mike", 10, "Don", 20]
```

Une liste avec quatre éléments : deux éléments chaînes et deux éléments entiers.

```
[[], [7], [8, 9]]
```

Une liste de listes. Chaque sous-liste est soit une liste vide soit une liste d'éléments entiers.

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

Une liste d'entiers. Cet exemple illustre l'utilisation de variables et d'expressions.

Vous pouvez affecter une liste à une variable, par exemple :

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

Vous pouvez ensuite accéder à des éléments spécifiques de la liste, par exemple :

```
mylist[0]
```

La sortie suivante est générée :

```
one
```

Le nombre entre crochets ([]) est appelé *index* et renvoie à un élément particulier de la liste. Les éléments d'une liste sont indexés à partir de 0.

Vous pouvez également sélectionner une plage d'éléments d'une liste ; on appelle cela le *tranchage* (slicing). Par exemple, `x[1:3]` sélectionne les deuxième et troisième éléments de `x`. L'index de fin est celui après la sélection.

Chaînes

Une *chaîne* est une séquence de caractères non modifiable considérée comme une valeur. Les chaînes prennent en charge l'ensemble des fonctions et opérateurs de séquence non modifiable qui donnent lieu à une nouvelle chaîne. Par exemple, `"abcdef"[1:4]` génère la sortie `"bcd"`.

En Python, les caractères sont représentés par des chaînes de longueur un.

Les littéraux de chaîne sont définis par l'utilisation de guillemets simples ou triples. Les chaînes définies à l'aide de guillemets simples ne peuvent pas s'étendre sur plusieurs lignes, alors que les chaînes définies à l'aide de guillemets triples le peuvent. Une chaîne peut figurer entre guillemets simples (') ou guillemets doubles ("). Un type de guillemet peut contenir l'autre type de guillemet sans caractère d'échappement, ou le même guillemet avec caractère d'échappement, c'est-à-dire précédé de la barre oblique inversée (\).

Exemples

```
"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"
```

Plusieurs chaînes séparées par un blanc sont automatiquement concaténées par l'analyseur Python. Cela facilite la saisie des chaînes longues et permet de mélanger les types de guillemets dans une même chaîne. Par exemple :

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

La sortie suivante est générée :

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

Les chaînes prennent en charge plusieurs méthodes utiles. Certaines de ces méthodes sont présentées dans le tableau suivant.

Tableau 6. Méthodes de chaîne

Méthode	Utilisation
s.capitalize()	Majuscule initiale pour s
s.count(ss {,start {,end}})	Compte les occurrences de ss dans s[start:end]
s.startswith(str {, start {, end}}) s.endswith(str {, start {, end}})	Test pour savoir si s commence par str Test pour savoir si s se termine par str
s.expandtabs({size})	Remplace les tabulations par des espaces, la taille (size) par défaut est 8
s.find(str {, start {, end}}) s.rfind(str {, start {, end}})	Recherche le premier index de str dans s ; s'il est introuvable, le résultat est -1. rfind recherche de droite à gauche.
s.index(str {, start {, end}}) s.rindex(str {, start {, end}})	Recherche le premier index de str dans s ; s'il est introuvable, renvoie l'erreur ValueError. rindex recherche de droite à gauche.
s.isalnum	Test pour savoir si la chaîne est alphanumérique
s.isalpha	Test pour savoir si la chaîne est alphabétique
s.isnum	Test pour savoir si la chaîne est numérique
s.isupper	Test pour savoir si la chaîne est tout en majuscules
s.islower	Test pour savoir si la chaîne est tout en minuscules
s.isspace	Test pour savoir si la chaîne n'est constituée que de blancs
s.istitle	Test pour savoir si la chaîne est une séquence de chaînes alphanumériques avec majuscule initiale
s.lower() s.upper() s.swapcase() s.title()	Convertit tout en minuscules Convertit tout en majuscules Convertit tout dans la casse opposée Convertit tout en casse titre
s.join(seq)	Joint les chaînes dans seq avec s comme séparateur
s.splitlines({keep})	Fractionne s en lignes, si keep a la valeur true, conserve les nouvelles lignes
s.split({sep {, max}})	Fractionne s en "mots" avec sep (par défaut, sep est un blanc), le nombre de fois indiqué par max
s.ljust(width) s.rjust(width) s.center(width) s.zfill(width)	Justifie à gauche la chaîne dans un champ de largeur width Justifie à droite la chaîne dans un champ de largeur width Centre la chaîne dans un champ de largeur width Remplit avec 0.
s.lstrip() s.rstrip() s.strip()	Supprime l'espace blanc de début Supprime l'espace blanc de fin Supprime les espaces blancs de début et de fin
s.translate(str {,delc})	Convertit s à l'aide de la table, après avoir supprimé les caractères de delc. str doit être une chaîne de longueur == 256.
s.replace(old, new {, max})	Remplace toutes les occurrences (ou le nombre d'occurrences indiqué par max) de la chaîne old par la chaîne new

Remarques

Les remarques sont des commentaires qui commencent par le signe dièse (#). L'ensemble du texte qui suit le signe dièse sur la même ligne est considéré comme appartenant à la remarque et est ignoré. Une remarque peut commencer dans n'importe quelle colonne. L'exemple suivant illustre l'utilisation des remarques :

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

Syntaxe des instructions

La syntaxe des instructions pour Python est très simple. En général, chaque ligne source correspond à une instruction unique. A l'exception des instructions expression et assignment, chaque instruction est introduite par un nom de mot-clé, tel que `if` ou `for`. Des lignes vierges ou des lignes de remarques peuvent être insérées partout entre les instructions du code. Si une ligne comporte plusieurs instructions, les instructions doivent être séparées par un point-virgule (;).

Les très longues instructions peuvent s'étendre sur plusieurs lignes. Dans ce cas, l'instruction qui doit se poursuivre sur la ligne suivante doit se terminer par une barre oblique inversée (\), par exemple :

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

Lorsqu'une structure figure entre parenthèses (()), crochets ([]) ou accolades ({}), l'instruction peut se poursuivre sur une nouvelle ligne après toute virgule, sans nécessiter l'insertion d'une barre oblique inversée, par exemple :

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

Identificateurs

Les identificateurs sont utilisés pour nommer des variables, des fonctions, des classes et des mots-clés. Les identificateurs peuvent être de n'importe quelle longueur, mais doivent commencer soit par un caractère alphabétique en majuscule ou en minuscule, soit par le caractère de soulignement (_). Les noms commençant par un trait de soulignement sont généralement réservés aux noms internes ou privés. Au delà du premier caractère, l'identificateur peut contenir n'importe quel nombre et n'importe quelle combinaison de caractères alphabétiques, nombres de 0 à 9, et le caractère de soulignement.

Il existe des mots réservés dans Python qui ne peuvent pas être utilisés pour nommer les variables, les fonctions ou les classes. Ils appartiennent aux catégories suivantes :

- **Préfixes d'instruction** : `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `exec`, `finally`, `for`, `from`, `global`, `if`, `import`, `pass`, `print`, `raise`, `return`, `try` et `while`
- **Préfixes de paramètre** : `as`, `import` et `in`
- **Opérateurs** : `and`, `in`, `is`, `lambda`, `not` et `or`

L'utilisation d'un mot-clé incorrect entraîne généralement une erreur `SyntaxError`.

Blocs de code

Les blocs de code sont des groupes d'instructions utilisés où des instructions uniques sont attendues. Les blocs de code peuvent respecter les instructions suivantes : `if`, `elif`, `else`, `for`, `while`, `try`, `except`, `def` et `class`. Ces instructions introduisent le bloc de code avec le caractère deux-points (:), par exemple :

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

L'indentation (mise en retrait) est utilisée pour délimiter les blocs de code (plutôt que les accolades utilisées dans Java). Toutes les lignes d'un bloc de code doivent être mises en retrait à la même position. En effet, un changement d'indentation indique la fin d'un bloc de code. Il est courant d'effectuer une mise en retrait de quatre espaces par niveau. Il est recommandé d'utiliser des espaces plutôt que des tabulations pour mettre en retrait les lignes. Vous ne devez pas mélanger les espaces et les tabulations. Les lignes du bloc le plus vers l'extérieur d'un module doivent commencer à la colonne un ; sinon, une erreur `SyntaxError` survient.

Les instructions qui constituent un bloc de code (et suivent le caractère deux-points) peuvent également figurer sur une seule ligne et être séparées par des points-virgules, par exemple :

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

Transmission d'arguments à un script

La transmission d'arguments à un script est utile car cela signifie qu'un script peut être utilisé de manière répétée sans modification. Les arguments transmis sur la ligne de commande sont transmis en tant que valeurs dans la liste `sys.argv`. Vous pouvez obtenir le nombre de valeurs transmises à l'aide de la commande `len(sys.argv)`. Par exemple :

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

Dans cet exemple, la commande `import` importe toute la classe `sys` de manière à ce que les méthodes qui existent pour cette classe, telles que `argv`, puissent être utilisées.

Le script de cet exemple peut être appelé à l'aide de la ligne suivante :

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

La sortie suivante est générée :

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

Exemples

Le mot-clé `print` imprime les arguments qui le suivent immédiatement. Si l'instruction est suivie d'une virgule, aucune nouvelle ligne n'est incluse dans la sortie. Par exemple :

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

La sortie suivante est générée :

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

L'instruction `for` est utilisée pour itérer un bloc de code. Par exemple :

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

Dans cet exemple, trois chaînes sont affectées à la liste `mylist1`. Les éléments de la liste sont ensuite imprimés, avec un élément de chaque ligne. La sortie suivante est générée :

```
one
two
three
```

Dans cet exemple, l'itérateur `lv` prend la valeur de chaque élément dans la liste `mylist1` tour à tour tandis que la boucle `for` implémente le bloc de code pour chaque élément. Un itérateur peut être n'importe quel identificateur valide, de n'importe quelle longueur.

L'instruction `if` est une instruction conditionnelle. Elle évalue la condition et renvoie `true` ou `false`, selon le résultat de l'évaluation. Par exemple :

```
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
```

Dans cet exemple, la valeur de l'itérateur `lv` est évaluée. Si la valeur de `lv` est `two`, la chaîne renvoyée est différente de la chaîne renvoyée si la valeur de `lv` n'est pas `two`. La sortie suivante est générée :

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

Méthodes mathématiques

A partir du module `math`, vous avez accès à des méthodes mathématiques utiles. Certaines de ces méthodes sont présentées dans le tableau suivant. Sauf indication contraire, toutes les valeurs sont renvoyées en tant que variables flottantes.

Tableau 7. Méthodes mathématiques

Méthode	Utilisation
<code>math.ceil(x)</code>	Renvoie la partie entière par excès de <code>x</code> en tant que variable flottante, c'est-à-dire le plus petit entier supérieur ou égal à <code>x</code> .
<code>math.copysign(x, y)</code>	Renvoie <code>x</code> avec le signe de <code>y</code> . <code>copysign(1, -0.0)</code> renvoie <code>-1</code> .
<code>math.fabs(x)</code>	Renvoie la valeur absolue de <code>x</code> .
<code>math.factorial(x)</code>	Renvoie la factorielle de <code>x</code> . Si <code>x</code> est négatif ou n'est pas un entier, une erreur <code>ValueError</code> est renvoyée.
<code>math.floor(x)</code>	Renvoie la partie entière par défaut de <code>x</code> en tant que variable flottante, c'est-à-dire le plus grand entier inférieur ou égal à <code>x</code> .
<code>math.frexp(x)</code>	Renvoie la mantisse (<code>m</code>) et l'exposant (<code>e</code>) de <code>x</code> en tant que paire (<code>m</code> , <code>e</code>). <code>m</code> est une variable flottante et <code>e</code> est un entier, tel que <code>x == m * 2**e</code> exactement. Si <code>x</code> est égal à zéro, renvoie <code>(0.0, 0)</code> , sinon <code>0.5 <= abs(m) < 1</code> .
<code>math.fsum(iterable)</code>	Renvoie une somme de valeurs à virgule flottante précise dans <code>iterable</code> .
<code>math.isinf(x)</code>	Vérifie si la variable flottante <code>x</code> est un infinitif positif ou négatif.
<code>math.isnan(x)</code>	Vérifie si la variable flottante <code>x</code> n'est pas un nombre, à savoir <code>NaN</code> (not a number).
<code>math.ldexp(x, i)</code>	Renvoie <code>x * (2**i)</code> . Il s'agit pour l'essentiel de l'inverse de la fonction <code>frexp</code> .
<code>math.modf(x)</code>	Renvoie les parties fractionnelle et entière de <code>x</code> . Les deux résultats portent le signe de <code>x</code> et sont des variables flottantes.

Tableau 7. Méthodes mathématiques (suite)

Méthode	Utilisation
<code>math.trunc(x)</code>	Renvoie la valeur réelle (Real) x , qui a été tronquée en intégrale (Integral).
<code>math.exp(x)</code>	Renvoie e^{**x} .
<code>math.log(x[, base])</code>	Renvoie le logarithme de x à la valeur donnée de base. Si base n'est pas spécifié, le logarithme naturel de x est renvoyé.
<code>math.log1p(x)</code>	Renvoie le logarithme naturel de $1+x$ (base e).
<code>math.log10(x)</code>	Renvoie le logarithme de base 10 de x .
<code>math.pow(x, y)</code>	Renvoie x élevé à la puissance y . <code>pow(1.0, x)</code> et <code>pow(x, 0.0)</code> renvoient toujours 1, même lorsque x est égal à zéro ou n'est pas un nombre (NaN).
<code>math.sqrt(x)</code>	Renvoie la racine carrée de x .

Outre les fonctions mathématiques, il existe certaines méthodes trigonométriques utiles. Ces méthodes sont présentées dans le tableau suivant.

Tableau 8. Méthodes trigonométriques

Méthode	Utilisation
<code>math.acos(x)</code>	Renvoie le cosinus inverse de x en radians.
<code>math.asin(x)</code>	Renvoie le sinus inverse de x en radians.
<code>math.atan(x)</code>	Renvoie la tangente inverse de x en radians.
<code>math.atan2(y, x)</code>	Renvoie <code>atan(y / x)</code> en radians.
<code>math.cos(x)</code>	Renvoie le cosinus de x en radians.
<code>math.hypot(x, y)</code>	Renvoie la norme euclidienne $\sqrt{x^2 + y^2}$. Il s'agit de la longueur du vecteur de l'origine jusqu'au point (x, y) .
<code>math.sin(x)</code>	Renvoie le sinus de x en radians.
<code>math.tan(x)</code>	Renvoie la tangente de x en radians.
<code>math.degrees(x)</code>	Convertit l'angle x de radians en degrés.
<code>math.radians(x)</code>	Convertit l'angle x de degrés en radians.
<code>math.acosh(x)</code>	Renvoie le cosinus hyperbolique inverse de x .
<code>math.asinh(x)</code>	Renvoie le sinus hyperbolique inverse de x .
<code>math.atanh(x)</code>	Renvoie la tangente hyperbolique inverse de x .
<code>math.cosh(x)</code>	Renvoie le cosinus hyperbolique de x .
<code>math.sinh(x)</code>	Renvoie le sinus hyperbolique de x .
<code>math.tanh(x)</code>	Renvoie la tangente hyperbolique de x .

Il existe également deux constantes mathématiques. La valeur de `math.pi` est la constante mathématique π . La valeur de `math.e` est la constante mathématique e .

Utilisation des caractères non ASCII

Pour pouvoir utiliser des caractères non ASCII, Python requiert un codage et un décodage explicite des chaînes en Unicode. Dans IBM SPSS Modeler, les scripts Python sont supposés être codés en UTF-8, ce qui est un codage Unicode standard prenant en charge les caractères non ASCII. Le script suivant peut

être compilé car le compilateur Python a été configuré en UTF-8 par SPSS Modeler.

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

Le noeud résultant aura un libellé incorrect.



Figure 3. Libellé de noeud contenant des caractères non ASCII affiché incorrectement

Le libellé est incorrect car le littéral chaîne a lui-même été converti en une chaîne ASCII par Python.

Python permet de spécifier des littéraux chaîne Unicode en ajoutant un préfixe u devant le littéral chaîne :

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

Une chaîne Unicode est créée et le libellé s'affiche correctement.



Figure 4. Libellé de noeud contenant des caractères non ASCII affiché correctement

L'utilisation de Python et de l'Unicode est un vaste sujet qui dépasse la portée du présent document. De nombreux ouvrages et de nombreuses ressources en lignes traitant en détail de ce sujet sont à votre disposition.

Programmation orientée objet

La programmation orientée objet repose sur la notion de création d'un modèle du problème cible dans vos programmes. La programmation orientée objet réduit les erreurs de programmation et favorise la réutilisation du code. Python est un langage orienté objet. Les objets définis dans Python sont dotés des caractéristiques suivantes :

- **Identité** : Chaque objet doit être distinct et vous devez pouvoir le tester. A cet effet, il existe des tests `is` et `is not`.
- **Etat** : Chaque objet doit pouvoir stocker un état. Des attributs, tels que les champs et les variables d'instance, existent à cet effet.
- **Comportement** : Chaque objet doit pouvoir manipuler son état. Il existe des méthodes pour cela.

Python inclut les fonctions suivantes pour prendre en charge la programmation orientée objet :

- **Création d'objet basée sur les classes** : Les classes sont des modèles pour la création d'objets. Les objets sont des structures de données avec le comportement associé.
- **Héritage avec polymorphisme** : Python prend en charge l'héritage unique et l'héritage multiple. Toutes les méthodes d'instance Python sont polymorphes et peuvent être remplacées par des sous-classes.
- **Encapsulation avec masquage des données** : Python permet de masquer les attributs. Lorsque les attributs sont masqués, ils sont accessibles depuis l'extérieur de la classe uniquement via des méthodes de la classe. Les classes implémentent des méthodes pour modifier les données.

Définition d'une classe

Dans une classe Python, les variables et les méthodes peuvent être définies. A la différence de Java, avec Python, vous pouvez définir n'importe quel nombre de classes publiques par fichier source (ou *module*). Par conséquent, on peut considérer un module dans Python comme semblable à un package dans Java.

Dans Python, les classes sont définies à l'aide de l'instruction `class`. L'instruction `class` présente la forme suivante :

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

ou

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

Lorsque vous définissez une classe, vous avez la possibilité de fournir plusieurs instructions d'*affectation* ou aucune. Ces dernières créent des attributs de classe partagés par toutes les instances de la classe. Vous pouvez également fournir plusieurs définitions de *fonction* ou aucune. Ces définitions de fonction créent des méthodes. La liste de superclasses est facultative.

Le nom de classe doit être unique dans un même périmètre, c'est-à-dire au sein d'un module, d'une fonction ou d'une classe. Vous pouvez définir plusieurs variables pour faire référence à la même classe.

Création d'une instance de classe

Les classes permettent de stocker des attributs de classe (ou partagés) ou de créer des instances de classe. Pour créer une instance de classe, vous appelez la classe comme s'il s'agissait d'une fonction. Prenons l'exemple de la classe suivante :

```
class MyClass:  
    pass
```

Dans ce cas, l'instruction `pass` est utilisée car une instruction est requise pour terminer la classe, mais aucune action à l'aide d'un programme n'est nécessaire.

L'instruction suivante crée une instance de la classe `MyClass` :

```
x = MyClass()
```

Ajout d'attributs à une instance de classe

A la différence de Java, avec Python, les clients peuvent ajouter des attributs à l'instance d'une classe. Une seule instance est modifiée. Par exemple, pour ajouter des attributs à une instance `x`, définissez de nouvelles valeurs sur cette instance :

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

Définition d'attributs de classe et de méthodes

Toute variable liée à une classe est un *attribut de classe*. Toute fonction définie dans une classe est une *méthode*. Les méthodes reçoivent une instance de la classe, appelée de manière conventionnelle `self`, comme premier argument. Par exemple, pour définir certains attributs de classe et méthodes, vous pouvez saisir le code suivant :

```
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
```

Au sein d'une classe, vous devez qualifier toutes les références à des attributs de classe avec le nom de classe ; par exemple, `MyClass.attr1`. Toutes les références à des attributs d'instance doivent être qualifiées avec la variable `self` ; par exemple, `self.text`. En dehors de la classe, vous devez qualifier toutes les références à des attributs de classe avec le nom de classe (par exemple `MyClass.attr1`) ou avec une instance de la classe (par exemple `x.attr1`, où `x` est une instance de la classe). En dehors de la classe, vous devez qualifier toutes les références à des variables d'instance avec une instance de la classe ; par exemple, `x.text`.

Variables masquées

Vous pouvez masquer des données en créant des variables *Private* (privées). Les variables privées sont accessibles uniquement par la classe elle-même. Si vous déclarez des noms de la forme `__xxx` ou `__xxx_yyy`, c'est-à-dire précédés de deux traits de soulignement, l'analyseur Python ajoutera automatiquement le nom de classe au nom déclaré, créant des variables masquées, par exemple :

```
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 la différence de Java, avec Python, toutes les références à des variables d'instance doivent être qualifiées avec `self` ; il n'y a pas d'utilisation implicite de `this`.

Héritage

La capacité à hériter de classes est fondamentale à la programmation orientée objet. Python prend en charge aussi bien l'héritage unique que l'héritage multiple. *L'héritage unique* signifie qu'il ne peut exister qu'une seule superclasse. *L'héritage multiple* signifie qu'il peut exister plusieurs superclasses.

L'héritage est mis en oeuvre par la sous-classification d'autres classes. Un nombre quelconque de classes Python peuvent être des superclasses. Dans l'implémentation Jython de Python, il n'est possible d'hériter directement ou indirectement que d'une seule classe Java. Il n'est pas nécessaire de fournir une superclasse.

Tout attribut ou toute méthode d'une superclasse se trouve également dans une sous-classe et peut être utilisé par la classe elle-même, ou par un client dans la mesure où l'attribut ou la méthode n'est pas masqué. Il est possible d'utiliser une instance de sous-classe partout où une instance de superclasse peut être utilisée ; c'est un exemple de *polymorphisme*. Ces fonctionnalités permettent la réutilisation et facilitent l'extension.

Exemple

```
class Class1: pass    #no inheritance

class Class2: pass

class Class3(Class1): pass    #single inheritance

class Class4(Class3, Class2): pass    #multiple inheritance
```

Chapitre 3. Génération de scripts dans IBM SPSS Modeler

Types de scripts

Dans IBM SPSS Modeler, il existe trois types de scripts :

- Les *scripts de flux* sont utilisés pour contrôler l'exécution d'un flux unique et sont stockés dans le flux.
- Les *scripts de super noeud* sont utilisés pour contrôler le comportement des super noeuds.
- Les *scripts autonomes ou de session* peuvent être utilisés pour coordonner l'exécution sur plusieurs flux différents.

Diverses méthodes peuvent être utilisées dans les scripts dans IBM SPSS Modeler et permettent d'accéder à un large éventail de fonctionnalités SPSS Modeler. Ces méthodes sont également utilisées dans Chapitre 4, «API de scriptage», à la page 35 pour créer des fonctions plus avancées.

Flux, flux super noeud et diagrammes

La plupart du temps, le terme *flux* signifie la même chose, qu'il s'agisse d'un flux chargé à partir d'un fichier ou utilisé dans un super noeud. Il signifie généralement une collection de noeuds connectés ensemble et pouvant être exécutés. Toutefois, dans le cadre du scriptage, toutes les opérations ne sont pas forcément prises en charge dans tous les emplacements. Cela signifie qu'un auteur de script doit connaître les variantes de script qu'il utilise.

Flux

Un flux est le principal type de document IBM SPSS Modeler. Il peut être enregistré, chargé, édité et exécuté. Les flux peuvent aussi être associés à des paramètres, des valeurs globales, un script et d'autres d'informations.

Flux super noeud

Le *flux super noeud* est le type de flux utilisé dans un super noeud. Tout comme un flux normal, il contient des noeuds qui sont liés ensemble. Les flux super noeud présentent cependant des différences par rapport à un flux normal :

- Les paramètres et les éventuels scripts sont associés au super noeud propriétaire du flux super noeud, et non au flux super noeud lui-même.
- Les flux super noeud possèdent des noeuds de connecteurs d'entrée et de sortie supplémentaires, selon le type de super noeud. Ces noeuds de connecteur sont utilisés pour faire circuler les informations en entrée et en sortie du flux super noeud et sont créés automatiquement lors de la création de ce dernier.

Diagrammes

Le terme *diagramme* couvre les fonctions qui sont prises en charge par les flux normaux et les flux super noeud, telles que l'ajout et la suppression de noeuds ou la modification des connexions entre noeuds.

Exécution d'un flux

L'exemple suivant exécute tous les noeuds exécutables dans le flux, et constitue le type de script de flux le plus simple :

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

L'exemple suivant exécute également tous les noeuds exécutables dans le flux :

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

Dans cet exemple, le flux est stocké dans une variable appelée `stream` (flux). Le stockage du flux dans une variable est utile car un script sert généralement à modifier soit le flux soit les noeuds au sein d'un flux. La création d'une variable qui stocke le flux permet d'obtenir un script plus concis.

Contexte de génération de scripts

Le module `modeler.script` fournit le contexte dans lequel un script est exécuté. Le module est automatiquement importé dans un script SPSS Modeler au moment de l'exécution. Le module définit quatre fonctions qui permettent au script d'accéder à son environnement d'exécution :

- La fonction `session()` renvoie la session pour le script. La session définit des informations telles que les paramètres régionaux et le backend SPSS Modeler (soit un processus local soit un SPSS Modeler Server en réseau) utilisé pour exécuter les flux.
- La fonction `stream()` peut être utilisée avec les scripts de flux et de super noeud. Cette fonction renvoie le flux propriétaire du script de flux ou de super noeud en cours d'exécution.
- La fonction `diagram()` peut être utilisée avec les scripts de super noeud. Cette fonction renvoie le diagramme au sein du super noeud. Pour les autres types de script, cette fonction renvoie les mêmes éléments que la fonction `stream()`.
- La fonction `supernode()` peut être utilisée avec les scripts de super noeud. Cette fonction renvoie le super noeud propriétaire du script en cours d'exécution.

Les quatre fonctions et leurs sorties sont résumées dans le tableau suivant.

Tableau 9. Récapitulatif des fonctions `modeler.script`

Type de script	<code>session()</code>	<code>stream()</code>	<code>diagram()</code>	<code>supernode()</code>
Autonome	Renvoie une session.	Renvoie le flux en cours de gestion au moment de l'appel du script (par exemple, le flux transmis via l'option <code>-stream</code> du mode de traitement par lots), ou <code>None</code> .	Identique à <code>stream()</code>	Sans objet
Flux	Renvoie une session.	Renvoie un flux.	Identique à <code>stream()</code>	Sans objet
Super noeud	Renvoie une session.	Renvoie un flux.	Renvoie un flux de super noeud.	Renvoie un super noeud.

Le module `modeler.script` définit également une méthode d'arrêt du script avec un code d'exit. La fonction `exit(exit-code)` arrête l'exécution du script et renvoie le code d'exit entier fourni.

L'une des méthodes définies pour un flux est `runAll(List)`. Cette méthode exécute tous les noeuds exécutables. Tous les modèles ou sorties générés par l'exécution des noeuds sont ajoutés à la liste fournie.

Il est courant pour une exécution de flux de générer des sorties telles que des modèles, des graphiques et autres résultats. Pour capturer cette sortie, un script peut fournir une variable initialisée en liste, par exemple :

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

Lorsque l'exécution est terminée, tous les objets générés par l'exécution sont accessibles à partir de la liste `results`.

Référence aux noeuds existants

Un flux est souvent préconfiguré avec certains paramètres qui doivent être modifiés avant l'exécution du flux. La modification de ces paramètres implique les tâches suivantes :

1. Localisation des noeuds dans le flux approprié.
2. Modification des paramètres de noeud et/ou de flux.

Recherche de noeuds

Les flux permettent de localiser un noeud existant de différentes manières. Ces méthodes sont résumées dans le tableau suivant.

Tableau 10. Méthodes de localisation d'un noeud existant

Méthode	Type de retour	Description
<code>s.findAll(type, label)</code>	Collection	Renvoie une liste de tous les noeuds avec le type et le libellé spécifiés. Le type ou le libellé peut avoir la valeur None, auquel cas l'autre paramètre est utilisé.
<code>s.findAll(filter, recursive)</code>	Collection	Renvoie l'ensemble de tous les noeuds acceptés par le filtre spécifié. Si l'indicateur récursif est True, tous les super noeuds appartenant au flux spécifié font également l'objet de la recherche.
<code>s.findById(id)</code>	Noeud	Renvoie le noeud avec l'ID fourni ou None si ce noeud n'existe pas. La recherche se limite au flux en cours.
<code>s.findByType(type, label)</code>	Noeud	Renvoie le noeud avec le type et/ou le libellé fourni. Le type ou le nom peut avoir la valeur None, auquel cas l'autre paramètre est utilisé. Si plusieurs noeuds génèrent une correspondance, un noeud arbitraire est choisi et renvoyé. Si aucun noeud ne génère une correspondance, la valeur renvoyée est None.
<code>s.findDownstream(fromNodes)</code>	Collection	Recherche dans la liste de noeuds fournie et renvoie l'ensemble de noeuds en aval des noeuds fournis. La liste renvoyée inclut les noeuds initialement fournis.
<code>s.findUpstream(fromNodes)</code>	Collection	Recherche dans la liste de noeuds fournie et renvoie l'ensemble de noeuds en amont des noeuds fournis. La liste renvoyée inclut les noeuds initialement fournis.

Par exemple, si un flux contient un seul noeud Filtrer auquel doit accéder le script, le noeud Filtrer peut être localisé à l'aide du script suivant :

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

Sinon, si l'ID du noeud (tel qu'il apparaît dans l'onglet Annotations de la boîte de dialogue du noeud) est connu, il peut être utilisé pour rechercher le noeud, par exemple :

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

Définition des propriétés

Les noeuds, les flux, les modèles et les sorties sont tous dotés de propriétés qui sont accessibles et qui peuvent, dans la plupart des cas, être définies. Les propriétés sont généralement utilisées pour modifier le comportement ou l'apparence de l'objet. Les méthodes disponibles pour accéder aux propriétés d'objet et pour les définir sont résumées dans le tableau suivant.

Tableau 11. Méthodes permettant d'accéder aux propriétés d'objet et de les définir

Méthode	Type de retour	Description
<code>p.getPropertyValue(propertyName)</code>	Objet	Renvoie la valeur de la propriété nommée, ou None s'il n'existe aucune propriété de ce genre.
<code>p.setPropertyValue(propertyName, value)</code>	Sans objet	Définit la valeur de la propriété nommée.
<code>p.setPropertyValues(properties)</code>	Sans objet	Définit les valeurs des propriétés nommées. Chaque entrée dans la carte de propriétés est composée d'une clé qui représente le nom de la propriété et la valeur qui doit être affectée à cette propriété.
<code>p.getKeyedPropertyValue(propertyName, keyName)</code>	Objet	Renvoie la valeur de la propriété nommée et la clé associée, ou None s'il n'existe aucune propriété ou clé de ce genre.
<code>p.setKeyedPropertyValue(propertyName, keyName, value)</code>	Sans objet	Définit la valeur de la propriété nommée et de la clé.

Par exemple, si vous souhaitez définir la valeur d'un noeud Délimité au début d'un flux, vous pouvez utiliser le script suivant :

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

Vous pouvez aussi souhaiter filtrer un champ à partir d'un noeud Filtrer. Dans ce cas, la valeur est également saisie sur le nom du champ, par exemple :

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

Création de noeuds et modification de flux

Dans certains cas, vous souhaitez peut-être ajouter de nouveaux noeuds à des flux existants. L'ajout de noeuds à des flux existants implique généralement les tâches suivantes :

1. Création de noeuds.
2. Création de liens entre les noeuds dans le flux existant.

Création de noeuds

Les flux permettent de créer des noeuds de différentes manières. Ces méthodes sont résumées dans le tableau suivant.

Tableau 12. Méthodes de création de noeuds

Méthode	Type de retour	Description
<code>s.create(nodeType, name)</code>	Noeud	Crée un noeud du type spécifié et l'ajoute au flux spécifié.
<code>s.createAt(nodeType, name, x, y)</code>	Noeud	Crée un noeud du type spécifié et l'ajoute au flux spécifié à l'emplacement spécifié. Si $x < 0$ ou $y < 0$, l'emplacement n'est pas défini.
<code>s.createModelApplier(modelOutput, name)</code>	Noeud	Crée un noeud application de modèle dérivé de l'objet de sortie de modèle fourni.

Par exemple, pour créer un nouveau noeud type dans un flux, vous pouvez utiliser le script suivant :

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

Création et suppression de liens entre les noeuds

Lorsqu'un nouveau noeud est créé au sein d'un flux, il doit être connecté dans une séquence de noeuds pour pouvoir être utilisé. Les flux fournissent plusieurs méthodes permettant de créer et de supprimer des liens entre les noeuds. Ces méthodes sont résumées dans le tableau suivant.

Tableau 13. Méthodes de création et de suppression de liens entre les noeuds

Méthode	Type de retour	Description
<code>s.link(source, target)</code>	Sans objet	Crée un nouveau lien entre les noeuds source et cible.
<code>s.link(source, targets)</code>	Sans objet	Crée de nouveaux liens entre le noeud source et chaque noeud cible dans la liste fournie.
<code>s.linkBetween(inserted, source, target)</code>	Sans objet	Connecte un noeud entre deux autres instances de noeud (les noeuds source et cible) et définit la position du noeud inséré pour qu'il soit entre eux. Tout lien direct entre les noeuds source et cible est d'abord supprimé.
<code>s.linkPath(path)</code>	Sans objet	Crée un nouveau chemin d'accès entre les instances de noeud. Le premier noeud est lié au deuxième, le deuxième est lié au troisième, et ainsi de suite.
<code>s.unlink(source, target)</code>	Sans objet	Supprime tout lien direct entre les noeuds source et cible.
<code>s.unlink(source, targets)</code>	Sans objet	Supprime tout lien direct entre le noeud source et chaque objet dans la liste des cibles.
<code>s.unlinkPath(path)</code>	Sans objet	Supprime tout chemin d'accès existant entre des instances de noeud.

Tableau 13. Méthodes de création et de suppression de liens entre les noeuds (suite)

Méthode	Type de retour	Description
<code>s.disconnect(node)</code>	Sans objet	Supprime tout lien entre le noeud fourni et les autres noeuds du flux spécifié.
<code>s.isValidLink(source, target)</code>	<i>booléen</i>	Renvoie True s'il est possible (valide) de créer un lien entre les noeuds source et cible spécifiés. Cette méthode vérifie que les deux objets appartiennent au flux spécifié, que le noeud source peut fournir un lien, que le noeud cible peut recevoir un lien, et que la création d'un tel lien ne provoquera pas de circularité dans le flux.

L'exemple de script ci-dessous effectue les cinq tâches suivantes :

1. Crée un noeud d'entrée Délimité, un noeud Filtrer et un noeud de sortie Table.
2. Connecte les noeuds.
3. Définit le nom de fichier sur le noeud d'entrée Délimité.
4. Filtre le champ "Drug" dans le résultat.
5. Exécute le noeud 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)
```

Importation, remplacement et suppression de noeuds

Outre la création et la connexion de noeuds, il est souvent nécessaire de remplacer et de supprimer des noeuds du flux. Les méthodes disponibles pour importer, remplacer et supprimer des noeuds sont résumées dans le tableau suivant.

Tableau 14. Méthodes d'importation, de remplacement et de suppression de noeuds

Méthode	Type de retour	Description
<code>s.replace(originalNode, replacementNode, discardOriginal)</code>	Sans objet	Remplace le noeud spécifié dans le flux spécifié. Le noeud d'origine et le noeud de remplacement doivent tous deux appartenir au flux spécifié.

Tableau 14. Méthodes d'importation, de remplacement et de suppression de noeuds (suite)

Méthode	Type de retour	Description
s.insert(source, nodes, newIDs)	Liste	Insère des copies des noeuds dans la liste fournie. On suppose que tous les noeuds de la liste fournie sont inclus dans le flux spécifié. L'indicateur newIDs indique si de nouveaux ID doivent être générés pour chaque noeud ou si l'ID existant doit être copié et utilisé. Il est admis que tous les noeuds d'un flux ont un ID unique ; cet indicateur doit donc être défini sur True si le flux source est identique au flux spécifié. La méthode renvoie la liste des noeuds récemment insérés, où l'ordre des noeuds est non défini (autrement dit, l'ordre n'est pas nécessairement le même que celui des noeuds dans la liste d'entrée).
s.delete(node)	Sans objet	Supprime du flux spécifié le noeud spécifié. Le noeud doit appartenir au flux spécifié.
s.deleteAll(nodes)	Sans objet	Supprime du flux spécifié tous les noeuds spécifiés. Tous les noeuds de la collection doivent appartenir au flux spécifié.
s.clear()	Sans objet	Supprime tous les noeuds du flux spécifié.

Traversée des noeuds d'un flux

L'une des conditions requises courantes est d'identifier les noeuds qui se trouvent en amont ou en aval d'un noeud particulier. Le flux fournit plusieurs méthodes qui peuvent être utilisées pour identifier ces noeuds. Ces méthodes sont résumées dans le tableau suivant.

Tableau 15. Méthodes d'identification des noeuds en amont et en aval

Méthode	Type de retour	Description
s.iterator()	Itérateur	Renvoie un itérateur sur les objets de noeud contenus dans le flux spécifié. Si le flux est modifié entre les appels de la fonction next(), le comportement de l'itérateur est non défini.
s.predecessorAt(node, index)	Noeud	Renvoie le prédécesseur immédiat spécifié du noeud fourni ou None si l'index est en dehors des limites.
s.predecessorCount(node)	Entier (int)	Renvoie le nombre de prédécesseurs immédiats du noeud fourni.
s.predecessors(node)	Liste	Renvoie les prédécesseurs immédiats du noeud fourni.
s.successorAt(node, index)	Noeud	Renvoie le successeur immédiat spécifié du noeud fourni ou None si l'index est en dehors des limites.

Tableau 15. Méthodes d'identification des noeuds en amont et en aval (suite)

Méthode	Type de retour	Description
s.successorCount(node)	Entier (int)	Renvoie le nombre de successeurs immédiats du noeud fourni.
s.successors(node)	Liste	Renvoie les successeurs immédiats du noeud fourni.

Informations sur les noeuds

Les noeuds appartiennent à différentes catégories telles que les noeuds d'importation et d'exportation de données, les noeuds de génération de modèle et d'autres types de noeuds. Pour chaque noeud, plusieurs méthodes peuvent être utilisées pour trouver des informations sur le noeud.

Les méthodes disponibles pour obtenir l'ID, le nom et le libellé d'un noeud sont résumées dans le tableau suivant.

Tableau 16. Méthodes permettant d'obtenir l'ID, le nom et le libellé d'un noeud

Méthode	Type de retour	Description
n.getLabel()	chaîne	Renvoie le libellé d'affichage du noeud spécifié. Le libellé correspond à la valeur de la propriété custom_name uniquement si cette propriété est une chaîne non vide et si la propriété use_custom_name n'est pas définie ; sinon, le libellé correspond à la valeur de getName().
n.setLabel(label)	Sans objet	Définit le libellé d'affichage du noeud spécifié. Si le nouveau libellé est une chaîne non vide, il est affecté à la propriété custom_name et False est affecté à la propriété use_custom_name afin que le libellé indiqué ait la priorité ; sinon, une chaîne vide est affectée à la propriété custom_name et True est affecté à la propriété use_custom_name.
n.getName()	chaîne	Renvoie le nom du noeud spécifié.
n.getID()	chaîne	Renvoie l'ID du noeud spécifié. Un nouvel ID est créé chaque fois qu'un nouveau noeud est créé. L'ID est conservé avec le noeud lorsqu'il est sauvegardé dans le cadre d'un flux, de sorte que lorsque le flux est ouvert, les ID de noeud sont préservés. Toutefois, si un noeud sauvegardé est inséré dans un flux, le noeud inséré est considéré comme un nouvel objet et un nouvel ID lui sera attribué.

Les méthodes disponibles pour obtenir d'autres informations sur un noeud sont résumées dans le tableau suivant.

Tableau 17. Méthodes permettant d'obtenir des informations sur un noeud

Méthode	Type de retour	Description
n.getTypeName()	chaîne	Renvoie le nom de génération de script de ce noeud. Il s'agit du même nom qui peut être utilisé pour créer une nouvelle instance de ce noeud.
n.isInitial()	Booléen	Renvoie True s'il s'agit d'un noeud <i>initial</i> , c'est-à-dire un noeud qui apparaît au début d'un flux.
n.isInline()	Booléen	Renvoie True s'il s'agit d'un noeud <i>en ligne</i> , c'est-à-dire un noeud qui apparaît au milieu d'un flux.
n.isTerminal()	Booléen	Renvoie True s'il s'agit d'un noeud <i>terminal</i> , c'est-à-dire un noeud qui apparaît à la fin d'un flux.
n.getXPosition()	Entier (int)	Renvoie le décalage de la position x du noeud dans le flux.
n.getYPosition()	Entier (int)	Renvoie le décalage de la position y du noeud dans le flux.
n.setXYPosition(x, y)	Sans objet	Définit la position du noeud dans le flux.
n.setPositionBetween(source, target)	Sans objet	Définit la position du noeud dans le flux de manière à le positionner entre les noeuds fournis.
n.isCacheEnabled()	Booléen	Renvoie True si le cache est activé ; sinon, renvoie False.
n.setCacheEnabled(val)	Sans objet	Active ou désactive le cache pour cet objet. Si le cache est saturé et que la mise en cache se trouve désactivée, le cache est vidé.
n.isCacheFull()	Booléen	Renvoie True si le cache est saturé ; sinon, renvoie False.
n.flushCache()	Sans objet	Vide le cache de ce noeud. N'a aucune incidence si le cache est désactivé ou n'est pas saturé.

Chapitre 4. API de scriptage

Introduction à l'API de scriptage

L'API de scriptage (génération de scripts) donne accès à un large éventail de fonctionnalités SPSS Modeler. Toutes les méthodes décrites jusqu'ici font partie de l'API et sont accessibles de manière implicite au sein du script, sans autre importation. Toutefois, si vous souhaitez faire référence aux classes de l'API, vous devez importer explicitement l'API avec l'instruction suivante :

```
import modeler.api
```

Cette instruction d'importation est requise par de nombreux exemples de l'API de scriptage.

Exemple : recherche de noeuds à l'aide d'un filtre personnalisé

La section «Recherche de noeuds», à la page 27 inclut un exemple de recherche de noeud dans un flux et utilise le nom de type du noeud comme critère de recherche. Dans certains cas, une recherche plus générique est requise et peut être implémentée en utilisant la classe `NodeFilter` et la méthode `findAll()` du flux. Ce type de recherche implique les deux étapes suivantes :

1. Création d'une nouvelle classe qui étend `NodeFilter` et qui implémente une version personnalisée de la méthode `accept()`.
2. Appel de la méthode `findAll()` du flux avec une instance de cette nouvelle classe. Tous les noeuds correspondant aux critères définis dans la méthode `accept()` sont ainsi renvoyés.

L'exemple suivant indique comment rechercher dans un flux des noeuds pour lesquels le cache de noeud est activé. La liste de noeuds renvoyée peut être utilisée pour vider ou désactiver les caches de ces noeuds.

```
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)
```

Métadonnées : informations sur les données

Etant donné que les noeuds sont connectés entre eux dans un flux, des informations sur les colonnes ou les champs disponibles à chaque noeud sont également disponibles. Par exemple, dans l'interface utilisateur Modeler, vous pouvez ainsi sélectionner les champs en fonction desquels effectuer un tri ou une agrégation. Ces informations sont appelées modèle de données.

Les scripts peuvent également accéder au modèle de données en consultant les champs entrant ou sortant dans un noeud. Pour certains noeuds, les modèles de données de sortie et d'entrée sont les mêmes (par exemple, un noeud Trier réorganise simplement les enregistrements mais ne modifie pas le modèle de données). Certains noeuds, comme le noeud Dériver, ajoutent de nouveaux champs. D'autres, tels que le noeud Filtrer, peuvent renommer ou supprimer des champs.

Dans l'exemple suivant, le script utilise le flux standard IBM SPSS Modeler `druglearn.str` et, pour chaque champ, génère un modèle avec l'un des champs d'entrée supprimés. Il effectue cette opération en :

1. accédant au modèle de données de sortie à partir du noeud Type ;
2. effectuant une boucle dans chaque champ du modèle de données de sortie ;

3. modifiant le noeud Filtrer pour chaque champ d'entrée ;
4. modifiant le nom du modèle généré ;
5. exécutant le noeud génération du modèle.

Remarque : Avant d'exécuter le script dans le flux `druglean.str`, pensez à définir le langage de scriptage Python (le flux a été créé dans une version antérieure de IBM SPSS Modeler ; le langage de scriptage du flux est donc défini par Legacy (existant)).

```
import modeler.api

stream = modeler.script.stream()
filternode = stream.findByType("filter", None)
typenode = stream.findByType("type", None)
c50node = stream.findByType("c50", None)
# Toujours utiliser un nom de modèle personnalisé
c50node.setPropertyValue("use_model_name", True)

lastRemoved = None
fields = typenode.getOutputDataModel()
for field in fields:
    # S'il s'agit du champ cible, l'ignorer
    if field.getModelingRole() == modeler.api.ModelingRole.OUT:
        continue

    # Activer de nouveau le champ le plus récemment supprimé
    if lastRemoved != None:
        filternode.setKeyedPropertyValue("include", lastRemoved, True)

    # Supprimer le champ
    lastRemoved = field.getColumnName()
    filternode.setKeyedPropertyValue("include", lastRemoved, False)

    # Définir le nom du nouveau modèle, puis exécuter le modèle généré
    c50node.setPropertyValue("model_name", "Exclude " + lastRemoved)
    c50node.run([])
```

L'objet Modèle de données offre un certain nombre de méthodes pour accéder aux informations relatives aux champs ou aux colonnes du modèle de données. Ces méthodes sont résumées dans le tableau suivant.

Tableau 18. Méthodes de l'objet Modèle de données pour accéder aux informations sur les champs ou les colonnes

Méthode	Type de retour	Description
<code>d.getColumnCount()</code>	<i>Entier (int)</i>	Renvoie le nombre de colonnes du modèle de données.
<code>d.columnIterator()</code>	Itérateur	Renvoie un itérateur qui renvoie chaque colonne dans l'ordre d'insertion "naturel". L'itérateur renvoie les instances de colonne.
<code>d.nameIterator()</code>	Itérateur	Renvoie un itérateur qui renvoie le nom de chaque colonne dans l'ordre d'insertion "naturel".
<code>d.contains(nom)</code>	<i>Booléen</i>	Renvoie True s'il existe une colonne avec le nom indiqué dans ce modèle de données et False dans le cas contraire.
<code>d.getColumn(nom)</code>	Colonne	Renvoie la colonne portant le nom indiqué.

Tableau 18. Méthodes de l'objet Modèle de données pour accéder aux informations sur les champs ou les colonnes (suite)

Méthode	Type de retour	Description
d.getColumnGroup(nom)	Groupe de colonnes	Renvoie le groupe de colonne désigné ou None si ce groupe n'existe pas.
d.getColumnGroupCount()	Entier (int)	Renvoie le nombre de groupes de colonnes du modèle de données.
d.columnGroupIterator()	Itérateur	Renvoie un itérateur qui renvoie chaque groupe de colonnes l'un après l'autre.
d.toArray()	Colonne[]	Renvoie le modèle de données sous forme de tableau de colonnes. Les colonnes sont classées dans leur ordre d'insertion "naturel".

Chaque champ (objet de colonne) inclut un certain nombre de méthodes pour accéder aux informations relatives à la colonne. Le tableau ci-après présente une sélection de ces méthodes.

Tableau 19. Méthodes de l'objet de colonne pour accéder aux informations sur la colonne

Méthode	Type de retour	Description
c.columnName()	chaîne	Renvoie le nom de la colonne.
c.columnLabel()	chaîne	Renvoie le libellé de la colonne ou une chaîne vide si aucun libellé n'est associé à la colonne.
c.measureType()	Type de mesure (MeasureType)	Renvoie le type de mesure de la colonne.
c.storageType()	Type de stockage (StorageType)	Renvoie le type de stockage de la colonne.
c.isMeasureDiscrete()	Booléen	Renvoie True si la colonne est discrète. Les colonnes qui sont un ensemble ou un indicateur sont considérées comme discrètes.
c.isModelOutputColumn()	Booléen	Renvoie True si la colonne est une colonne de sortie de modèle.
c.isStorageDatetime()	Booléen	Renvoie True si le stockage de la colonne est une valeur d'heure, de date ou d'horodatage.
c.isStorageNumeric()	Booléen	Renvoie True si le stockage de la colonne est un entier ou un nombre réel.
c.isValidValue(valeur)	Booléen	Renvoie True si la valeur spécifiée est valide pour ce stockage et valid lorsque les valeurs de colonne valides sont affichées.
c.getModelingRole()	Rôle de modélisation (ModelingRole)	Renvoie le rôle de modélisation de la colonne.
c.getSetValues()	Objet[]	Renvoie un tableau de valeurs valides pour la colonne ou None si les valeurs ne sont pas affichées ou si la colonne n'est pas un ensemble.

Tableau 19. Méthodes de l'objet de colonne pour accéder aux informations sur la colonne (suite)

Méthode	Type de retour	Description
c.getValueLabel(valeur)	chaîne	Renvoie le libellé de la valeur dans la colonne.
c.getFalseFlag()	Objet	Renvoie l'indicateur "false" pour la colonne ou None si la valeur n'est pas affichée ou si la colonne n'est pas un indicateur.
c.getTrueFlag()	Objet	Renvoie l'indicateur "true" pour la colonne ou None si la valeur n'est pas affichée ou si la colonne n'est pas un indicateur.
c.getLowerBound()	Objet	Renvoie la valeur limite inférieure des valeurs de la colonne ou None si la valeur n'est pas affichée ou si la colonne n'est pas continue.
c.getUpperBound()	Objet	Renvoie la valeur limite supérieure des valeurs de la colonne ou None si la valeur n'est pas affichée ou si la colonne n'est pas continue.

Notez que la plupart des méthodes qui accèdent aux informations relatives à une colonne possèdent des méthodes équivalentes définies dans l'objet de modèle de données lui-même. Par exemple, les deux instructions suivantes sont équivalentes :

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

Accès aux objets générés

L'exécution d'un flux implique généralement la génération d'objets de sortie supplémentaires. Ces objets supplémentaires peuvent être un nouveau modèle ou une partie de sortie fournissant des informations à utiliser lors des exécutions ultérieures.

Dans l'exemple ci-dessous, le flux `druglearn.str` est de nouveau utilisé comme point de départ pour le flux. Dans cet exemple, tous les noeuds du flux sont exécutés et les résultats sont stockés dans une liste. Ensuite, le script parcourt les résultats en boucle, toute sortie du modèle résultant de l'exécution est enregistrée en tant que fichier de modèle IBM SPSS Modeler (.gm), et le modèle est exporté en 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 d'exécution de tâche permet d'exécuter facilement diverses tâches courantes. Les méthodes disponibles dans cette classe sont résumées dans le tableau suivant.

Tableau 20. Méthodes de la classe d'exécution de tâche pour effectuer des tâches courantes

Méthode	Type de retour	Description
t.createStream(name, autoConnect, autoManage)	Flux	Crée et renvoie un nouveau flux. Notez que le code chargé de créer des flux de manière privée, sans qu'ils soient visibles pour l'utilisateur, doit définir l'indicateur autoManage sur False.
t.exportDocumentToFile(documentOutput, filename, fileFormat)	Sans objet	Exporte la description du flux vers un fichier en utilisant le format de fichier spécifié.
t.exportModelToFile(modelOutput, filename, fileFormat)	Sans objet	Exporte le modèle vers un fichier en utilisant le format de fichier spécifié.
t.exportStreamToFile(stream, filename, fileFormat)	Sans objet	Exporte le flux vers un fichier en utilisant le format de fichier spécifié.
t.insertNodeFromFile(filename, diagram)	Noeud	Lit et renvoie un noeud à partir du fichier spécifié, en l'insérant dans le diagramme fourni. Vous pouvez l'utiliser pour lire des objets de noeud et de super noeud.
t.openDocumentFromFile(filename, autoManage)	DocumentOutput	Lit et renvoie un document à partir du fichier spécifié.
t.openModelFromFile(filename, autoManage)	ModelOutput	Lit et renvoie un modèle à partir du fichier spécifié.
t.openStreamFromFile(filename, autoManage)	Flux	Lit et renvoie un flux à partir du fichier spécifié.
t.saveDocumentToFile(documentOutput, filename)	Sans objet	Enregistre le document à l'emplacement de fichier spécifié.
t.saveModelToFile(modelOutput, filename)	Sans objet	Enregistre le modèle à l'emplacement de fichier spécifié.
t.saveStreamToFile(stream, filename)	Sans objet	Enregistre le flux à l'emplacement de fichier spécifié.

Traitement des erreurs

Le langage Python fournit un traitement des erreurs via le bloc de code try...except. Celui-ci peut être utilisé dans les scripts pour intercepter les exceptions et gérer les problèmes qui, sans cela, entraîneraient l'arrêt du script.

Dans l'exemple de script ci-dessous, on tente d'extraire un modèle d'un IBM SPSS Collaboration and Deployment Services Repository. Cette opération peut provoquer une exception ; par exemple, il se peut

que les données de connexion au référentiel n'aient pas été correctement configurées ou que le chemin de référentiel soit incorrect. Dans le script, cela peut entraîner l'émission d'une exception `ModelerException` (toutes les exceptions générées par IBM SPSS Modeler sont dérivées de `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()
```

Remarque : certaines opérations de script peuvent provoquer l'émission d'exceptions Java standard ; celles-ci ne sont pas dérivées de `ModelerException`. Pour intercepter ces exceptions, un bloc `except` supplémentaire peut être utilisé pour intercepter toutes les exceptions Java, par exemple :

```
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()
```

Paramètres de flux, de session et de super noeud

Les paramètres sont utiles pour transmettre des valeurs au moment de l'exécution, plutôt que de les coder en dur directement dans un script. Les paramètres et leurs valeurs sont définis de la même manière que pour les flux, à savoir comme des entrées dans la table de paramètres d'un flux ou super noeud, ou comme des paramètres sur la ligne de commande. Les classes de flux (`Stream`) et de super noeud (`SuperNode`) implémentent un ensemble de fonctions définies par l'objet `ParameterProvider`, comme indiqué dans le tableau suivant. La session fournit un appel `getParameters()` qui renvoie un objet définissant ces fonctions.

Tableau 21. Fonctions définies par l'objet `ParameterProvider`

Méthode	Type de retour	Description
<code>p.parameterIterator()</code>	Itérateur	Renvoie un itérateur des noms de paramètres pour cet objet.
<code>p.getParameterDefinition(parameterName)</code>	<code>ParameterDefinition</code>	Renvoie la définition de paramètre pour le paramètre avec le nom spécifié, ou <code>None</code> s'il n'existe aucun paramètre de ce genre dans ce fournisseur. Le résultat peut être un instantané de la définition au moment de l'appel de la méthode et ne doit pas forcément refléter les modifications apportées ultérieurement au paramètre via ce fournisseur.
<code>p.getParameterLabel(parameterName)</code>	chaîne	Renvoie le libellé du paramètre nommé, ou <code>None</code> s'il n'existe aucun paramètre de ce genre.

Tableau 21. Fonctions définies par l'objet `ParameterProvider` (suite)

Méthode	Type de retour	Description
<code>p.setParameterLabel(parameterName, label)</code>	Sans objet	Définit le libellé du paramètre nommé.
<code>p.getParameterStorage(parameterName)</code>	<code>ParameterStorage</code>	Renvoie le stockage du paramètre nommé, ou <code>None</code> s'il n'existe aucun paramètre de ce genre.
<code>p.setParameterStorage(parameterName, storage)</code>	Sans objet	Définit le stockage du paramètre nommé.
<code>p.getParameterType(parameterName)</code>	<code>ParameterType</code>	Renvoie le type du paramètre nommé, ou <code>None</code> s'il n'existe aucun paramètre de ce genre.
<code>p.setParameterType(parameterName, type)</code>	Sans objet	Définit le type du paramètre nommé.
<code>p.getParameterValue(parameterName)</code>	Objet	Renvoie la valeur du paramètre nommé, ou <code>None</code> s'il n'existe aucun paramètre de ce genre.
<code>p.setParameterValue(parameterName, value)</code>	Sans objet	Définit la valeur du paramètre nommé.

Dans l'exemple suivant, le script agrège certaines données Telco pour rechercher la région dont le revenu moyen (average income) est le plus faible. Un paramètre de flux est ensuite défini avec cette région. Ce paramètre de flux est par la suite utilisé dans un noeud Sélectionner (Select) afin d'exclure cette région des données, avant qu'un modèle d'attrition soit généré sur le reste.

Cet exemple est fictif car le script génère le noeud Sélectionner (Select) lui-même et aurait donc pu générer directement la valeur correcte dans l'expression du noeud Sélectionner (Select). Toutefois, les flux sont généralement préconfigurés ; la définition des paramètres de cette manière permet donc de donner un exemple utile.

La première partie de l'exemple de script crée le paramètre de flux qui contiendra la région dont le revenu moyen est le plus faible. Le script crée également les noeuds dans la branche d'agrégation et la branche de génération de modèle, et les connecte.

```
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)

```

L'exemple de script crée le flux suivant.

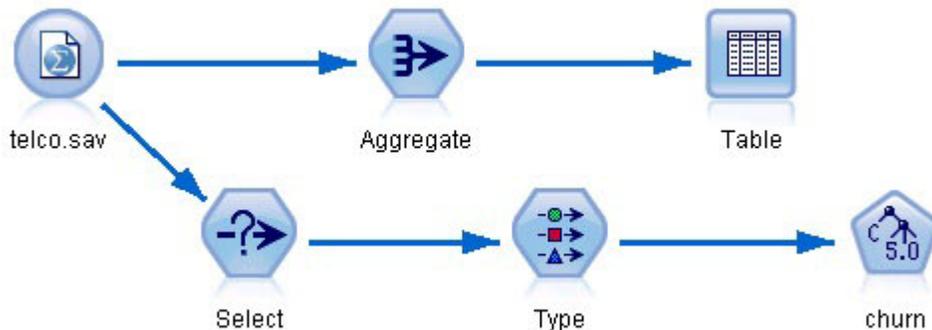


Figure 5. Flux résultant de l'exemple de script

La partie suivante de l'exemple de script exécute le noeud Table à la fin de la branche d'agrégation.

```

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

```

La partie suivante de l'exemple de script accède à la sortie de table générée par l'exécution du noeud Table. Ensuite, le script parcourt les lignes de la table, à la recherche de la région présentant le revenu moyen le plus faible.

```

# 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 partie suivante du script utilise la région présentant le revenu moyen le plus faible pour définir le paramètre de flux "LowestRegion" créé précédemment. Le script exécute ensuite le générateur de modèle en excluant la région spécifiée des données d'apprentissage.

```

# 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([])

```

L'exemple de script complet est présenté ci-dessous.

```

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 = 10000000.0
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([])

```

Valeurs globales

Les valeurs globales sont utilisées pour calculer diverses statistiques récapitulatives pour des champs spécifiés. Ces valeurs récapitulatives sont accessibles partout dans le flux. Les valeurs globales sont semblables aux paramètres de flux en ce sens qu'elles sont accessibles par nom via le flux. Elles diffèrent des paramètres de flux dans la mesure où les valeurs associées sont mises à jour automatiquement lorsqu'un noeud V. globales (Valeurs globales) est exécuté, plutôt que d'être affectées par le scriptage ou depuis la ligne de commande. Il est possible d'accéder aux valeurs globales pour un flux en appelant la méthode `getGlobalValues()` du flux.

L'objet `GlobalValues` définit les fonctions indiquées dans le tableau suivant.

Tableau 22. Fonctions définies par l'objet `GlobalValues`

Méthode	Type de retour	Description
<code>g.fieldNameIterator()</code>	Itérateur	Renvoie un itérateur pour chaque nom de champ avec au moins une valeur globale.
<code>g.getValue(type, fieldName)</code>	Objet	Renvoie la valeur globale pour le type et le nom de champ spécifiés, ou <code>None</code> si aucune valeur n'est trouvée. La valeur renvoyée est généralement un nombre, mais il se peut que des fonctionnalités futures renvoient des types de valeur différents.
<code>g.getValues(fieldName)</code>	Carte	Renvoie une carte contenant les entrées connues pour le nom de champ spécifié, ou <code>None</code> s'il n'existe aucune entrée pour le champ.

`GlobalValues.Type` définit le type des statistiques récapitulatives disponibles. Les statistiques récapitulatives suivantes sont disponibles :

- `MAX` : valeur maximum du champ.
- `MEAN` : valeur moyenne du champ.
- `MIN` : valeur minimum du champ.
- `STDDEV` : écart type du champ.
- `SUM` : somme des valeurs dans le champ.

Par exemple, le script suivant accède à la valeur moyenne du champ "income", qui est calculée par un noeud V. globales (Valeurs globales) :

```

import modeler.api

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

```

Utilisation de plusieurs flux : scripts autonomes

Pour utiliser plusieurs flux, il convient d'utiliser un script autonome. Le script autonome peut être édité et exécuté dans l'interface utilisateur IBM SPSS Modeler ou transmis comme paramètre de ligne de commande en mode de traitement par lots.

Le script autonome suivant ouvre deux flux. L'un de ces flux génère un modèle, et le deuxième trace la distribution des valeurs prédites.

```
# Change to the appropriate location for your system
demosDir = "C:/Program Files/IBM/SPSS/Modeler/16/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()
```

Chapitre 5. Conseils pour la génération de scripts

Cette section présente les différents conseils et techniques pour l'utilisation des scripts : modification de l'exécution du flux et utilisation d'un mot de passe codé dans un script.

Modification de l'exécution du flux

Lorsqu'un flux est exécuté, ses noeuds terminaux sont exécutés dans un ordre optimisé pour la situation par défaut. Dans certains cas, vous pouvez choisir un ordre d'exécution différent. Pour modifier l'ordre d'exécution d'un flux, réalisez les opérations suivantes dans l'onglet Exécution de la boîte de dialogue Propriétés du flux :

1. Commencez par un script vide.
2. Cliquez sur le bouton **Ajouter le script par défaut** de la barre d'outils pour ajouter le script de flux par défaut.
3. Modifiez à votre convenance l'ordre des instructions qu'il contient.

Utilisation de modèles

Si la fonction de remplacement automatique de modèle est activée dans IBM SPSS Modeler et qu'un noeud de création de modèles est exécuté via l'interface utilisateur IBM SPSS Modeler, un nugget de modèle existant lié au noeud de création de modèles est remplacé par le nouveau nugget de modèle. Si le noeud de création de modèles est exécuté via un script, le nugget de modèle existant lié n'est pas remplacé. Pour remplacer le nugget de modèle existant, vous devez explicitement spécifier le remplacement du nugget dans votre script.

Génération d'un mot de passe codé

Dans certains cas, vous pouvez être amené à ajouter un mot de passe à un script, par exemple, pour accéder à une source de données protégée par un mot de passe. Vous pouvez utiliser les mots de passe codés dans :

- Les propriétés des noeuds source et de sortie SGBD
- Les arguments de ligne de commande permettant la connexion au serveur
- Les propriétés de connexion à la base de données stockées dans un fichier *.par* (fichier de paramètres généré depuis l'onglet Publier d'un noeud Export)

L'interface utilisateur dispose d'un outil qui permet de générer des mots de passe codés à partir de l'algorithme Blowfish (pour plus de détails, consultez le site <http://www.schneier.com/blowfish.html>). Une fois le mot de passe codé, vous pouvez le copier et le stocker dans des fichiers de script et des arguments de ligne de commande. La propriété de noeud *epassword*, utilisée pour *database* et *databaseexport*, stocke le mot de passe codé.

1. Pour générer un mot de passe codé, choisissez l'option suivante dans le menu Outils :
Coder le mot de passe...
2. Entrez un mot de passe dans la zone de texte Mot de passe.
3. Cliquez sur **Coder** pour générer le codage aléatoire de votre mot de passe.
4. Cliquez sur le bouton Copier pour copier le mot de passe codé dans le Presse-papiers.
5. Collez le mot de passe dans le script ou le paramètre souhaité.

Vérification du script

Vous pouvez procéder à une vérification rapide de la syntaxe de tous les types de script en cliquant sur le bouton de vérification rouge dans la barre d'outils de la boîte de dialogue Script autonome.



Figure 6. Icônes de la barre d'outils du script de flux

La vérification du script vous prévient de toute erreur se produisant dans votre code et vous propose des recommandations d'amélioration. Pour visualiser la ligne comportant des erreurs, cliquez sur le commentaire dans la partie inférieure de la boîte de dialogue. L'erreur est surlignée en rouge.

Génération de scripts à partir de la ligne de commande

La génération de scripts vous permet d'exécuter des opérations généralement effectuées dans l'interface utilisateur. Il vous suffit d'indiquer et d'exécuter un flux autonome dans la ligne de commande lors du lancement de IBM SPSS Modeler.

Par exemple :

```
client -script scores.py -execute
```

L'indicateur `-script` charge le script spécifié et l'indicateur `-execute` exécute toutes les commandes du fichier de script.

Spécification de chemins de fichier

Lorsque vous spécifiez des chemins d'accès à des répertoires et des fichiers, vous pouvez utiliser une barre oblique (/) ou une double barre oblique inversée (\\) comme séparateur ; par exemple :

```
c:/demos/druglearn.str
```

ou

```
c:\\demos\\druglearn.str
```

Compatibilité avec les versions précédentes

Les scripts existants créés dans les versions antérieures d'IBM SPSS Modeler doivent en général fonctionner tels quels dans la nouvelle version. Pour que les scripts fonctionnent, il convient de sélectionner **Existant** dans l'onglet du script de flux dans la boîte de dialogue Propriétés du flux ou dans la boîte de dialogue Script autonome. Les nuggets de modèle peuvent désormais être insérés automatiquement dans le flux (réglage par défaut), et peuvent soit remplacer soit compléter un nugget existant de ce type dans le flux. Ceci dépend des réglages des options **Ajout de modèle à un flux** et **Remplacer le modèle précédent (Outils > Options > Options utilisateur > Notifications)**. Vous pouvez, par exemple, être amené à modifier le script d'une version précédente dans laquelle le remplacement de nugget est géré en supprimant le nugget existant et en insérant le nouveau.

Il se peut que des scripts créés dans la version actuelle ne fonctionnent pas dans les versions antérieures.

Les scripts Python créés dans la version actuelle ne fonctionneront pas dans les versions antérieures.

Si un script créé dans une version antérieure utilise une commande qui a été remplacée depuis (ou qui est obsolète), l'ancienne forme reste prise en charge mais un message d'avertissement apparaît. Par

exemple, l'ancien mot-clé `generated` a été remplacé par `model`, et `clear generated` par `clear generated palette`. Les scripts qui utilisent les anciennes formes continuent de s'exécuter mais un avertissement apparaît.

Chapitre 6. Arguments de ligne de commande

Appel du logiciel

Vous pouvez utiliser la ligne de commande de votre système d'exploitation pour lancer IBM SPSS Modeler comme suit :

1. Dans le cas d'un ordinateur sur lequel est installé IBM SPSS Modeler, ouvrez une fenêtre DOS ou une invite de commande.
2. Pour lancer l'interface IBM SPSS Modeler en mode interactif, tapez la commande `modelerclient` suivie des arguments souhaités, par exemple :

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

Les arguments disponibles (indicateurs) vous permettent de vous connecter à un serveur, de charger des flux, d'exécuter des scripts ou d'indiquer les autres paramètres nécessaires.

Utilisation d'arguments de ligne de commande

Vous pouvez ajouter des arguments de ligne de commande (également appelés **indicateurs**) à la commande `modelerclient` initiale pour modifier l'appel de IBM SPSS Modeler.

Plusieurs types d'arguments de ligne de commande sont disponibles et décrits plus loin dans cette section.

Tableau 23. Types d'arguments de ligne de commande.

Type d'argument	Description
Arguments système	Pour plus d'informations, voir la rubrique «Arguments système», à la page 52.
Arguments de paramètre	Pour plus d'informations, voir la rubrique «Arguments de paramètre», à la page 53.
Arguments de connexion au serveur	Pour plus d'informations, voir la rubrique «Arguments de connexion au serveur», à la page 54.
Arguments de connexion au IBM SPSS Collaboration and Deployment Services Repository	Pour plus d'informations, voir la rubrique «IBM SPSS Collaboration and Deployment Services Repository Arguments de connexion», à la page 55.

Vous pouvez par exemple utiliser les indicateurs `-server`, `-stream` et `-execute` pour vous connecter à un serveur, puis charger et exécuter un flux, comme indiqué ci-dessous :

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

Notez que lors d'une exécution en parallèle avec l'installation d'un client local, les arguments de connexion au serveur ne sont pas obligatoires.

Il est possible de placer entre guillemets doubles les valeurs de paramètre qui contiennent des espaces. Par exemple :

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

Vous pouvez également exécuter des scripts IBM SPSS Modeler de la même manière, en utilisant l'indicateur `-script`.

Débogage d'arguments de ligne de commande

Pour déboguer une ligne de commande, utilisez la commande `modelerclient` afin de lancer IBM SPSS Modeler avec les arguments souhaités. Vous avez ainsi la possibilité de vérifier que les commandes s'exécutent comme souhaité. Vous pouvez également vérifier les valeurs de paramètre transmises depuis la ligne de commande dans la boîte de dialogue Paramètres de session (menu Outils, Définir les paramètres de session).

Arguments système

Le tableau suivant décrit les arguments système disponibles pour l'appel de la ligne de commande de l'interface utilisateur.

Tableau 24. Arguments système

Argument	Comportement/description
@ <commandFile>	Le caractère @ suivi d'un nom de fichier indique une liste de commandes. Lorsque <code>modelerclient</code> détecte un argument commençant par le caractère @, il agit sur les commandes du fichier correspondant comme si elles s'étaient trouvées dans la ligne de commande. Pour plus d'informations, voir la rubrique «Combinaison de plusieurs arguments», à la page 55.
-directory <dir>	Définit le répertoire de travail par défaut. En mode local, ce répertoire est utilisé pour les données et la sortie. Exemple : <code>-directory c:/</code> ou <code>-directory c:\</code>
-server_directory <dir>	Définit le répertoire du serveur par défaut pour les données. Le répertoire de travail, défini à l'aide de l'indicateur <code>-directory</code> est utilisé pour la sortie.
-execute	Après le démarrage, exécute tout flux, état ou script chargé au démarrage. Si un script est chargé en plus d'un flux ou d'un état, seul le script est exécuté.
-stream <flux>	Au démarrage, charge le flux spécifié. Vous pouvez spécifier plusieurs flux, mais le dernier flux sera défini comme flux en cours.
-script <script>	Charge, au démarrage, le script autonome spécifié. Vous pouvez le spécifier en plus d'un flux ou d'un état comme décrit ci-dessous, mais un seul script peut être chargé au démarrage. Si le suffixe du fichier de script est <code>.py</code> , on suppose que le fichier est un script Python ; sinon, on suppose qu'il s'agit d'un script existant.
-model <modèle>	Au démarrage, charge le modèle généré (fichier au format <code>.gm</code>) indiqué.
-state <état>	Au démarrage, charge l'état enregistré indiqué.
-project <projet>	Charge le projet spécifié. Vous ne pouvez charger qu'un seul projet au démarrage.
-output <sortie>	Au démarrage, charge l'objet de sortie enregistré (fichier au format <code>.cou</code>).
-help	Affiche la liste des arguments de ligne de commande. Lorsque cette option est précisée, tous les autres arguments sont ignorés et l'écran Aide apparaît.
-P <name>=<value>	Sert à définir un paramètre de démarrage. Peut également être utilisé pour définir les propriétés de noeud (paramètres de propriété).
-scriptlang <python legacy>	Peut être utilisé pour indiquer le langage de script associé à l'option <code>-script</code> , quel que soit le suffixe du fichier de script. Exemple <code>client -scriptlang python -script scores.txt -execute</code> Exécute le fichier de script fourni avec Python même si le suffixe du fichier n'était pas <code>.py</code> .

Remarque : vous pouvez également définir les répertoires par défaut dans l'interface utilisateur. Pour accéder aux options, dans le menu Fichier, sélectionnez **Définir le répertoire de travail** ou **Définir le répertoire du serveur**.

Chargement de plusieurs fichiers

A partir de la ligne de commande, vous pouvez charger plusieurs flux, états et sorties au démarrage en répétant l'argument applicable pour chaque objet chargé. Par exemple, pour charger et exécuter deux flux nommés *report.str* et *train.str*, utilisez la commande suivante :

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

Chargement d'objets du IBM SPSS Collaboration and Deployment Services Repository

Etant donné que vous pouvez charger certains objets à partir d'un fichier ou du IBM SPSS Collaboration and Deployment Services Repository (si vous disposez de la licence correspondante), le préfixe de nom de fichier `spsscr:` et éventuellement `fichier:` (pour les objets sur disque) indique à IBM SPSS Modeler l'emplacement de l'objet. Le préfixe fonctionne avec les indicateurs suivants :

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

Vous utilisez le préfixe pour créer un URI qui indique l'emplacement de l'objet. Par exemple : `-stream "spsscr:///folder_1/scoring_stream.str"`. Lorsque le préfixe `spsscr:` est présent, vous devez définir dans la même commande une connexion valide au IBM SPSS Collaboration and Deployment Services Repository. Ainsi, par exemple, la commande complète serait semblable à ce qui suit :

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080  
-spsscr_username myusername -spsscr_password mypassword  
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

A partir de la ligne de commande, vous *devez* utiliser un URI. L'élément simple `REPOSITORY_PATH` n'est pas pris en charge. (Il fonctionne uniquement au sein des scripts.)

Arguments de paramètre

Les paramètres peuvent être utilisés en tant qu'indicateurs pendant l'exécution de la ligne de commande de IBM SPSS Modeler. Dans les arguments de ligne de commande, l'indicateur `-P` sert à indiquer un paramètre du type `-P <nom>=<valeur>`.

Il peut s'agir des paramètres suivants :

- **Paramètres simples**
- Des **paramètres de propriété**, également appelés **propriétés de noeud**. Ces paramètres servent à modifier les paramètres des noeuds du flux. Pour plus d'informations, voir la rubrique «Présentation des propriétés de noeud», à la page 58.
- Des **paramètres de ligne de commande**, utilisés pour modifier l'appel de IBM SPSS Modeler.

Par exemple, vous pouvez fournir les noms d'utilisateur et les mots de passe d'une source de données sous la forme d'un indicateur de ligne de commande, comme l'illustre l'exemple suivant :

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

Le format est identique à celui du paramètre `datasources` de la propriété de noeud `database`. Pour plus d'informations, voir la rubrique «Propriétés du noeud SGBD», à la page 67.

Arguments de connexion au serveur

L'indicateur `-server` indique à IBM SPSS Modeler de se connecter à un serveur public et les indicateurs `-hostname`, `-use_ssl`, `-port`, `-username`, `-password`, et `-domain` sont utilisés pour indiquer à IBM SPSS Modeler comment se connecter au serveur public. Si aucun argument `-server` n'est spécifié, le serveur par défaut ou le serveur local est utilisé.

Exemples

Pour vous connecter à un serveur public :

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

Pour vous connecter à un cluster de serveurs :

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

Veillez noter que la connexion à un groupe de serveur nécessite le coordinateur de processus avec IBM SPSS Collaboration and Deployment Services. Par conséquent, l'argument `-cluster` doit être utilisé conjointement avec les options de connexion du référentiel (`spsscr_*`). Pour plus d'informations, voir la rubrique «IBM SPSS Collaboration and Deployment Services Repository Arguments de connexion», à la page 55.

Tableau 25. Arguments de connexion au serveur.

Argument	Comportement/description
<code>-server</code>	Exécute IBM SPSS Modeler en mode serveur, en se connectant à un serveur public à l'aide des indicateurs <code>-hostname</code> , <code>-port</code> , <code>-username</code> , <code>-password</code> et <code>-domain</code> .
<code>-hostname <name></code>	Nom d'hôte du serveur. Disponible en mode serveur uniquement.
<code>-use_ssl</code>	Indique que la connexion doit utiliser le protocole SSL (Secure Socket Layer). Cet indicateur est facultatif ; par défaut, le protocole SSL n'est <i>pas</i> utilisé.
<code>-port <number></code>	Numéro de port du serveur spécifié. Disponible en mode serveur uniquement.
<code>-cluster <name></code>	Spécifie une connexion à un cluster de serveurs plutôt qu'à un serveur nommé ; cet argument est une alternative aux arguments <code>hostname</code> , <code>port</code> et <code>use_ssl</code> . Le nom est le nom de groupe ou une URI unique qui identifie le groupe dans IBM SPSS Collaboration and Deployment Services Repository. Le cluster de serveurs est géré par le coordinateur de processus dans IBM SPSS Collaboration and Deployment Services. Pour plus d'informations, voir la rubrique «IBM SPSS Collaboration and Deployment Services Repository Arguments de connexion», à la page 55.
<code>-username <name></code>	Nom d'utilisateur utilisé pour la connexion au serveur. Disponible en mode serveur uniquement.
<code>-password <password></code>	Mot de passe utilisé pour la connexion au serveur. Disponible en mode serveur uniquement. <i>Remarque</i> : si l'argument <code>-password</code> n'est pas utilisé, le système vous invite à entrer un mot de passe.
<code>-epassword <encodedpasswordstring></code>	Mot de passe codé utilisé pour la connexion au serveur. Disponible en mode serveur uniquement. <i>Remarque</i> : vous pouvez créer un mot de passe codé à partir du menu Outils de l'application IBM SPSS Modeler.
<code>-domain <name></code>	Domaine utilisé pour la connexion au serveur. Disponible en mode serveur uniquement.
<code>-P <name>=<value></code>	Sert à définir un paramètre de démarrage. Peut également être utilisé pour définir les propriétés de noeud (paramètres de propriété).

IBM SPSS Collaboration and Deployment Services Repository

Arguments de connexion

Remarque : une licence distincte est requise pour accéder à un référentiel IBM SPSS Collaboration and Deployment Services. Pour plus d'informations, voir <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Pour pouvoir stocker ou extraire des objets du IBM SPSS Collaboration and Deployment Services via la ligne de commande, vous devez indiquer une connexion valide au IBM SPSS Collaboration and Deployment Services Repository. Par exemple :

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080  
-spsscr_username myusername -spsscr_password mypassword  
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

Le tableau suivant répertorie les arguments qu'il convient d'utiliser pour paramétrer la connexion.

Tableau 26. Arguments de connexion au IBM SPSS Collaboration and Deployment Services Repository

Argument	Comportement/description
-spsscr_hostname <nom d'hôte ou adresse IP>	Nom d'hôte ou adresse IP du serveur sur lequel est installé IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_port <numéro>	Numéro de port sur lequel IBM SPSS Collaboration and Deployment Services Repository accepte les connexions (8080 par défaut).
-spsscr_use_ssl	Indique que la connexion doit utiliser le protocole SSL (Secure Socket Layer). Cet indicateur est facultatif ; par défaut, le protocole SSL n'est pas utilisé.
-spsscr_username <nom>	Nom d'utilisateur utilisé pour la connexion au IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_password <mot de passe>	Mot de passe utilisé pour la connexion au IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_epassword <mot de passe codé>	Mot de passe codé utilisé pour la connexion au IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_domain <nom>	Domaine utilisé pour la connexion au IBM SPSS Collaboration and Deployment Services Repository. Cet indicateur est facultatif ; ne l'utilisez que si vous vous connectez via LDAP ou Active Directory.

Combinaison de plusieurs arguments

Utilisez le symbole @ suivi du nom du fichier pour combiner plusieurs arguments dans un fichier de commande unique défini lors de l'appel de la commande. Vous pouvez ainsi raccourcir l'appel via la ligne de commande et remédier aux éventuelles limites appliquées à la longueur des commandes par les systèmes d'exploitation. Par exemple, la commande de démarrage suivante utilise les arguments spécifiés dans le fichier référencé par <commandFileName>.

```
modelerclient @<commandFileName>
```

Si vous devez utiliser des espaces, placez le nom et le chemin d'accès du fichier entre guillemets, comme indiqué ci-dessous :

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

Le fichier de commande peut contenir tous les arguments qui étaient auparavant spécifiés séparément au démarrage, avec un argument par ligne. Par exemple :

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

Lorsque vous rédigez des fichiers de commande et que vous y faites référence, veillez à respecter les contraintes suivantes :

- N'inscrivez qu'une commande par ligne.
- N'intégrez pas un argument @CommandFile au sein d'un fichier de commande.

Chapitre 7. Référence sur les propriétés

Présentation des références sur les propriétés

Vous pouvez indiquer un certain nombre de propriétés pour les noeuds, les flux, les super noeuds et les projets. Certaines propriétés sont communes à tous les noeuds, telles que le nom, l'annotation et l'info-bulle, alors que d'autres sont propres à certains types de noeud. D'autres propriétés font référence aux opérations de flux de haut niveau, comme la mise en cache ou le comportement du super noeud. Vous pouvez accéder aux propriétés via l'interface utilisateur standard (par exemple, lorsque vous ouvrez une boîte de dialogue pour modifier les options d'un noeud) et les utiliser de plusieurs manières.

- Vous pouvez modifier les propriétés via des scripts, comme l'explique cette section. Pour plus d'informations, reportez-vous à «Définition des propriétés», à la page 28.
- Les propriétés de noeud peuvent être utilisées dans les paramètres du super noeud.
- Les propriétés de noeud peuvent également être utilisées dans le cadre d'une option de ligne de commande (en utilisant l'indicateur -P) lors du démarrage de IBM SPSS Modeler.

Dans le cadre de la génération de scripts dans IBM SPSS Modeler, les propriétés de noeud et de flux sont souvent appelées **paramètres de propriété**. Dans cette aide, elles sont appelées propriétés de noeud ou de flux.

Pour plus d'informations sur le langage de script, voir Chapitre 2, «Langage de script», à la page 13.

Abréviations

Les abréviations standard sont utilisées tout au long de la syntaxe des propriétés de noeud. La connaissance des abréviations est utile lors de la rédaction de scripts.

Tableau 27. Abréviations standard utilisées tout au long de la syntaxe.

Abréviation	Signification
abs	Absolute value
len	Longueur
min	Minimum
max	Maximum
correl	Corrélation
covar	Covariance :
num	Nombre ou numérique
pct (pourcentage)	Percent (pour cent) ou (percentage) pourcentage
transp	Transparence
xval	Cross-validation (validation croisée)
var	Variance ou variable (dans les noeuds source)

Exemple de propriétés de noeud et de flux

Dans IBM SPSS Modeler, les propriétés de noeud et de flux peuvent être utilisées de différentes façons. Ces propriétés sont le plus souvent utilisées dans le cadre d'un script, que ce soit un **script autonome**, pour automatiser plusieurs flux ou opérations, ou un **script de flux**, pour automatiser des processus au sein d'un flux unique. Vous pouvez également indiquer des paramètres de noeud à l'aide des propriétés de noeud du super noeud. Au niveau le plus basique, les propriétés peuvent aussi être utilisées sous

forme d'une option de ligne de commande pour démarrer IBM SPSS Modeler. Si vous utilisez le commutateur -p dans le cadre de l'appel d'une ligne de commande, vous pouvez modifier un paramètre dans le flux à l'aide d'une propriété du flux.

Pour consulter d'autres exemples de scripts, voir les rubriques «Paramètres de flux, de session et de super noeud», à la page 40 et «Arguments système», à la page 52.

Présentation des propriétés de noeud

Pour chaque type de noeud, un certain nombre de propriétés sont autorisées. Chaque propriété comporte un type. Ce type peut être « général » (nombre, indicateur ou chaîne), auquel cas le type approprié est attribué automatiquement aux paramètres de la propriété. Si le type correct ne peut pas être défini, une erreur est renvoyée. Un paramètre de propriété peut également spécifier un intervalle de valeurs autorisées, telles que Supprimer, ApparierEtSupprimer et InclureCommeTexte. Dans ce cas, une erreur est renvoyée si une autre valeur est utilisée. Les propriétés indicateurs doivent être lues ou définies à l'aide des valeurs True (vrai) et False (faux). Dans les tableaux de référence du présent document, les propriétés structurées sont indiquées telles quelles dans la colonne *Description de la propriété* et sont accompagnées de la syntaxe à utiliser.

Propriétés communes des noeuds

Dans IBM SPSS Modeler, un certain nombre de propriétés sont communes à tous les noeuds (y compris les super noeuds).

Tableau 28. Propriétés communes des noeuds.

Nom de la propriété	Le type de données	Description de la propriété
use_custom_name	booléen	
name	chaîne	Propriété en lecture seule qui lit le nom (automatique ou personnalisé) d'un noeud de l'espace de travail.
custom_name	chaîne	Indique le nom personnalisé du noeud.
tooltip	chaîne	
annotation	chaîne	
keywords	chaîne	Propriété structurée indiquant la liste de mots-clés associés à l'objet
cache_enabled	booléen	
node_type	source_supernode process_supernode terminal_supernode tous les noms de noeud indiqués pour la génération de scripts	Propriété en lecture seule utilisée pour faire référence à un noeud par type. Par exemple, au lieu de faire référence à un noeud uniquement par son nom, tel que revenu_réel, vous pouvez également indiquer son type (par exemple, utilisateur ou filtre).

Les propriétés propres aux super noeuds sont abordées séparément, comme pour tous les autres noeuds. Pour plus d'informations, voir la rubrique Chapitre 19, «Propriétés du super noeud», à la page 245.

Chapitre 8. Propriétés de flux

La génération de scripts permet de contrôler différentes propriétés de flux.

Le script peut accéder au flux en cours à l'aide de la fonction `stream()` dans le module `modeler.script`, par exemple :

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

Pour référencer les propriétés de flux, vous devez utiliser une variable de flux spéciale et placer le symbole `^` devant le flux.

La propriété `nodes` est utilisée pour faire référence aux noeuds du flux en cours.

Les propriétés de flux sont décrites dans le tableau suivant.

Tableau 29. Propriétés du flux.

Nom de la propriété	Le type de données	Description de la propriété
<code>execute_method</code>	Normal Script	
<code>date_format</code>	"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.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	
<code>date_baseline</code>	<i>nombre</i>	
<code>date_2digit_baseline</code>	<i>nombre</i>	

Tableau 29. Propriétés du flux (suite).

Nom de la propriété	Le type de données	Description de la propriété
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	booléen	
import_datetime_as_string	booléen	
decimal_places	nombre	
decimal_symbol	Default Period Comma	
angles_in_radians	booléen	
use_max_set_size	booléen	
max_set_size	nombre	
ruleset_evaluation	Voting FirstHit	
refresh_source_nodes	booléen	Permet de rafraîchir les noeuds source automatiquement dès l'exécution du flux.
script	chaîne	
script_language	Python Existant	Définit le langage de script pour le script de flux.
annotation	chaîne	
encoding	SystemDefault "UTF-8"	
stream_rewriting	booléen	
stream_rewriting_maximise_sql	booléen	
stream_rewriting_optimise_clem_exécution	booléen	
stream_rewriting_optimise_syntax_exécution	booléen	
enable_parallelism	booléen	
sql_generation	booléen	
database_caching	booléen	
sql_logging	booléen	
sql_generation_logging	booléen	
sql_log_native	booléen	
sql_log_prettyprint	booléen	

Tableau 29. Propriétés du flux (suite).

Nom de la propriété	Le type de données	Description de la propriété
record_count_suppress_input	<i>booléen</i>	
record_count_feedback_interval	<i>entier</i>	
use_stream_auto_create_node_settings	<i>booléen</i>	Si la valeur est définie sur true (vrai), les paramètres spécifiques au flux sont utilisés ; sinon, les préférences utilisateur sont utilisées.
create_model_applier_for_new_models	<i>booléen</i>	Si la valeur est définie sur true (vrai), lorsqu'un générateur de modèle crée un nouveau modèle et qu'il ne contient aucun lien de mise à jour actif, un nouvel applicateur de modèle est ajouté.
create_model_applier_update_links	createEnabled createDisabled doNotCreate	Définit le type de lien créé lorsqu'un noeud applicateur de modèle est automatiquement ajouté.
create_source_node_from_builders	<i>booléen</i>	Si la valeur est définie sur true (vrai), lorsqu'un générateur de source crée une nouvelle sortie source et qu'elle ne contient aucun lien de mise à jour actif, un nouveau noeud source est ajouté.
create_source_node_update_links	createEnabled createDisabled doNotCreate	Définit le type de lien créé lorsqu'un noeud source est automatiquement ajouté.

Chapitre 9. Propriétés des nœuds source

Propriétés communes aux nœuds source

Les propriétés communes à tous les nœuds source sont répertoriées ci-dessous, avec des informations sur certains nœuds dans les rubriques suivantes.

Tableau 30. Propriétés communes aux nœuds source.

Nom de la propriété	Le type de données	Description de la propriété
direction	Input Target Both None Partitionner Split Frequency RecordID	Propriété saisie (rôles des champs). <i>Remarque</i> : les valeurs In et Out sont désormais obsolètes. Leur prise en charge pourrait être supprimée dans une version ultérieure.
type	Range Indicateur Set Sans type Discrete Ordered Set Default	Type de champ. Si vous paramétrez cette propriété sur <i>Par défaut</i> , toute définition de propriété de valeurs est effacée. Si la valeur <i>value_mode</i> est paramétrée sur <i>Spécifier</i> , elle est redéfinie sur <i>Lire</i> . Si la propriété <i>value_mode</i> est paramétrée sur <i>Transférer</i> ou <i>Lire</i> , la définition du type n'a aucune incidence sur elle.
storage	Inconnu String Integer Real Time Date Timestamp	Propriété saisie en lecture seule pour le type de stockage de champ.
check	None Nullify Coerce Discard Warn Abort	Propriété saisie (vérification du type et de l'intervalle des champs).
values	[<i>valeur valeur</i>]	Pour un champ continu (intervalle), la première valeur représente la valeur minimale et la dernière valeur, la valeur maximale. Pour des champs nominaux (ensemble), spécifiez toutes les valeurs. Pour les champs indicateurs, la première valeur représente <i>false</i> (faux) et la dernière valeur, <i>true</i> (vrai). La définition automatique de cette propriété paramètre la propriété <i>value_mode</i> sur <i>Specify</i> .
value_mode	Read Pass Lire + Current Specify	Détermine la façon dont les valeurs sont déterminées pour un champ lors du passage suivant des données. <i>Remarque</i> : vous ne pouvez pas paramétrer directement cette propriété sur <i>Spécifier</i> . Pour utiliser des valeurs spécifiques, paramétrez la propriété <i>values</i> .

Tableau 30. Propriétés communes aux noeuds source (suite).

Nom de la propriété	Le type de données	Description de la propriété
default_value_mode	Read Pass	Indique la méthode par défaut de définition des valeurs de tous les champs. Ce paramètre peut être ignoré pour certains champs à l'aide de la propriété value_mode.
extend_values	booléen	S'applique lorsque la propriété value_mode est paramétrée sur <i>Read</i> . Paramétrez cette valeur sur <i>T</i> pour ajouter des valeurs qui viennent d'être lues aux valeurs existantes du champ. Paramétrez cette valeur sur <i>F</i> pour supprimer des valeurs existantes en faveur des valeurs qui viennent d'être lues.
value_labels	chaîne	Permet d'utiliser un libellé de valeur. Vous devez d'abord indiquer les valeurs.
enable_missing	booléen	Lorsque cette propriété est paramétrée sur <i>T</i> , elle active le suivi des valeurs manquantes du champ.
missing_values	[valeur valeur ...]	Spécifie les valeurs de données qui indiquent les données manquantes.
range_missing	booléen	Lorsque cette propriété est définie sur <i>T</i> , spécifie si un intervalle de valeurs manquantes (vides) est défini pour un champ.
missing_lower	chaîne	Lorsque le paramètre range_missing a pour valeur true (vrai), indique la limite inférieure de l'intervalle des valeurs manquantes.
missing_upper	chaîne	Lorsque le paramètre range_missing a pour valeur true, indique la limite supérieure de l'intervalle des valeurs manquantes.
null_missing	booléen	Lorsque cette propriété est paramétrée sur <i>T</i> , les valeurs nulles (valeurs non définies affichées sous la forme \$null\$ dans le logiciel) sont considérées comme des valeurs manquantes.
whitespace_missing	booléen	Lorsque cette propriété est paramétrée sur <i>T</i> , les valeurs composées uniquement d'espaces blancs (espaces, tabulations et caractères de nouvelle ligne) sont considérées comme des valeurs manquantes.
description	chaîne	Permet d'indiquer un libellé de champ ou une description.
default_include	booléen	Propriété saisie permettant d'indiquer si le comportement par défaut consiste à transmettre ou à filtrer les champs :
include	booléen	Propriété saisie utilisée pour déterminer pour chaque champ s'il doit être inclus ou filtré :
new_name	chaîne	

Propriétés du noeud asimport

La source Analytic Server vous permet d'exécuter un flux sur le système de fichiers HDFS (Hadoop Distributed File System).

Tableau 31. Propriétés du noeud asimport.

Propriétés du noeud asimport	Type de données	Description de la propriété
data_source	chaîne	Nom de la source de données.
host	chaîne	Nom de l'hôte Analytic Server.
port	entier	Port sur lequel Analytic Server est en mode écoute.
tenant	chaîne	Dans un environnement à service partagé, nom du titulaire auquel vous appartenez. Dans un environnement à service exclusif, prend par défaut la valeur ibm .
set_credentials	booléen	Si l'authentification utilisateur sur Analytic Server est la même que sur le serveur SPSS Modeler, donnez à cette propriété la valeur false . Sinon, donnez-lui la valeur true .
user_name	chaîne	Nom d'utilisateur utilisé pour la connexion à Analytic Server. Uniquement nécessaire si set_credentials a la valeur true.
password	chaîne	Mot de passe de connexion à Analytic Server. Uniquement nécessaire si set_credentials a la valeur true.

Propriétés du noeud cognosimport



Le noeud source IBM Cognos BI importe des données depuis les bases de données Cognos BI.

Tableau 32. Propriétés du noeud cognosimport.

Propriétés du noeud cognosimport	Le type de données	Description de la propriété
mode	Data Report	Spécifie s'il faut importer les données Cognos BI (par défaut) ou les rapports.

Tableau 32. Propriétés du noeud cognosimport (suite).

Propriétés du noeud cognosimport	Le type de données	Description de la propriété
cognos_connection	<i>{ "chaîne", booléen, "chaîne", "chaîne", "chaîne" }</i>	Une propriété de liste contenant les détails de la connexion du serveur Cognos. Le format est le suivant : <i>{ "URL_Serveur_Cognos", mode_connexion, "espace de nommage", "nom utilisateur", "mot de passe" }</i> où : <i>URL_Serveur_Cognos</i> est l'URL du serveur Cognos vers lequel vous effectuez l'exportation <i>mode_connexion</i> indique si la connexion anonyme est utilisée et est soit true soit false ; si la valeur est true, les champs suivants doivent être définis sur "" <i>espace de nommage</i> spécifie le fournisseur de sécurité pour l'authentification utilisé pour se connecter au serveur. <i>nom utilisateur</i> et <i>mot de passe</i> sont ceux utilisés pour la connexion au serveur Cognos
cognos_package_name	<i>chaîne</i>	Le chemin d'accès et le nom du package Cognos depuis lequel vous importez les objets de données, par exemple : <i>/Public Folders/GOSALES</i> Remarque : seules les barres obliques sont valides.
cognos_items	<i>{ "champ", "champ", ... "champ" }</i>	Le nom d'un ou plusieurs objets de données à importer. Le format de <i>champ</i> est <i>[namespace].[query_subject].[query_item]</i>
cognos_filters	<i>champ</i>	Le nom d'un ou plusieurs filtres à appliquer avant d'importer les données.
cognos_data_parameters	<i>liste</i>	Valeurs des paramètres d'invite des données. Les paires nom/valeur sont entre accolades et les paires multiples sont séparées par des virgules et la chaîne entière est entre crochets. Format : <i>[{"param1", "value"},...,{"paramN", "value"}]</i>
cognos_report_directory	<i>champ</i>	Le chemin Cognos d'un dossier ou d'un package depuis lequel importer les rapports, par exemple : <i>/Public Folders/GOSALES</i> Remarque : seules les barres obliques sont valides.
cognos_report_name	<i>champ</i>	Le chemin et le nom dans l'emplacement d'un rapport à importer.
cognos_report_parameters	<i>liste</i>	Valeurs des paramètres de rapport. Les paires nom/valeur sont entre accolades et les paires multiples sont séparées par des virgules et la chaîne entière est entre crochets. Format : <i>[{"param1", "value"},...,{"paramN", "value"}]</i>

Propriétés du noeud SGBD



Le noeud SGBD peut être utilisé pour importer des données provenant de nombreux autres logiciels utilisant la connectivité ODBC (Open Database Connectivity), tels que Microsoft SQL Server, DB2, Oracle, etc.

Tableau 33. Propriétés du noeud SGBD.

Propriétés du noeud SGBD	Le type de données	Description de la propriété
mode	Table Query	Spécifiez <i>Table</i> pour établir la connexion à une table de base de données à l'aide des commandes de la boîte de dialogue ou indiquez <i>Requête</i> pour interroger la base de données sélectionnée en utilisant SQL.
datasource	chaîne	Nom de la base de données (reportez-vous à la remarque ci-dessous).
username	chaîne	Détails de la connexion à la base de données (reportez-vous aussi à la remarque ci-dessous).
password	chaîne	
epassword	chaîne	Indique un mot de passe codé à la place du codage en dur d'un mot de passe dans un script. Pour plus d'informations, voir la rubrique «Génération d'un mot de passe codé», à la page 47. Cette propriété est en lecture seule au cours de l'exécution.
tablename	chaîne	Nom de la table à laquelle vous souhaitez accéder.
strip_spaces	None Left Right Both	Options permettant de supprimer des espaces situés en début et en fin de chaîne.
use_quotes	AsNeeded Always Never	Indiquez si les noms des tables et des colonnes sont placés entre guillemets lors de l'envoi des requêtes à la base de données (par exemple, s'ils contiennent des espaces ou des signes de ponctuation).
query	chaîne	Indique le code SQL de la requête à soumettre.

Remarque : si le nom de la base de données (dans la propriété de datasource) comporte des espaces, au lieu des propriétés individuelles de datasource, username et password, il est alors préférable d'utiliser une propriété de datasource unique au format suivant :

Tableau 34. Propriétés du noeud SGBD - datasource spécifique.

Propriétés du noeud SGBD	Le type de données	Description de la propriété
datasource	chaîne	Format : [database_name,username,password[,true false]] Le dernier paramètre sert aux mots de passe codés. S'il est défini sur true, le mot de passe est décodé avant usage.

Utilisez aussi ce format si vous modifiez la source de données ; cependant, si vous souhaitez simplement changer le nom d'utilisateur ou le mot de passe, vous pouvez utiliser les propriétés username ou password.

Propriétés du noeud datacollectionimport



Le noeud d'importation des données IBM SPSS Data Collection importe des données d'enquête en fonction du modèle IBM SPSS Data Collection Data Model utilisé par les produits IBM Corp. dédiés aux études de marché. Pour pouvoir utiliser ce noeud, vous devez avoir installé avant la bibliothèque de données IBM SPSS Data Collection.

Figure 7. Noeud d'importation des données Dimensions

Tableau 35. Propriétés du noeud datacollectionimport.

Propriétés du noeud datacollectionimport	Le type de données	Description de la propriété
metadata_name	chaîne	Nom du MDSC. La valeur spéciale DimensionsMDD indique que le document de métadonnées IBM SPSS Data Collection standard doit être utilisé. Autres valeurs possibles : mrADODsc mrI2dDsc mrLogDsc mrQdiDrsDsc mrQvDsc mrSampleReportingMDSC mrSavDsc mrSCDsc mrScriptMDSC La valeur spéciale none (aucun) indique qu'il n'existe aucun MDSC.
metadata_file	chaîne	Nom du fichier de stockage des métadonnées.
casedata_name	chaîne	Nom du CDSC. Valeurs possibles : mrADODsc mrI2dDsc mrLogDsc mrPunchDSC mrQdiDrsDsc mrQvDsc mrRdbDsc2 mrSavDsc mrScDSC mrXm1Dsc La valeur spéciale none (aucun) indique qu'il n'existe aucun CDSC.

Tableau 35. Propriétés du noeud `datacollectionimport` (suite).

Propriétés du noeud <code>datacollectionimport</code>	Le type de données	Description de la propriété
<code>casedata_source_type</code>	Inconnu File Folder UDL DSN	Indique le type de source du CDSC.
<code>casedata_file</code>	<i>chaîne</i>	Quand la propriété <code>casedata_source_type</code> est paramétrée sur <i>Fichier</i> , cette propriété indique le fichier contenant les données d'observation.
<code>casedata_folder</code>	<i>chaîne</i>	Quand la propriété <code>casedata_source_type</code> est paramétrée sur <i>Dossier</i> , cette propriété indique le dossier contenant les données d'observation.
<code>casedata_udl_string</code>	<i>chaîne</i>	Quand la propriété <code>casedata_source_type</code> est paramétrée sur <i>UDL</i> , cette propriété indique la chaîne de connexion OLE-DB correspondant à la source de données contenant les données d'observation.
<code>casedata_dsn_string</code>	<i>chaîne</i>	Quand la propriété <code>casedata_source_type</code> est paramétrée sur <i>DSN</i> , cette propriété indique la chaîne de connexion ODBC correspondant à la source de données.
<code>casedata_project</code>	<i>chaîne</i>	Lorsque vous lisez des données d'observation provenant d'une base de données IBM SPSS Data Collection, vous pouvez fournir le nom du projet. Pour tous les autres types de données d'observation, ce paramètre doit rester vide.
<code>version_import_mode</code>	All Latest Specify	Définit le mode de traitement des versions.
<code>specific_version</code>	<i>chaîne</i>	Quand la propriété <code>version_import_mode</code> est paramétrée sur <i>Spécifier</i> , cette propriété définit la version des données d'observation à importer.
<code>use_language</code>	<i>chaîne</i>	Définit si les libellés d'une langue spécifique doivent être utilisées.
<code>language</code>	<i>chaîne</i>	Si la propriété <code>use_language</code> est paramétrée sur <i>True</i> (vrai), cette propriété définit le code langue à utiliser pour l'importation. Il doit s'agir de l'un des codes disponibles dans les données d'observation.
<code>use_context</code>	<i>chaîne</i>	Définit si un contexte spécifique doit être importé. Les contextes sont utilisés pour varier la description associée aux réponses.
<code>context</code>	<i>chaîne</i>	Si la propriété <code>use_context</code> est paramétrée sur <i>True</i> (vrai), cette propriété définit le contexte à importer. Il doit s'agir de l'un des contextes disponibles dans les données d'observation.
<code>use_label_type</code>	<i>chaîne</i>	Définit si un type de libellé spécifique doit être importé.

Tableau 35. Propriétés du noeud datacollectionimport (suite).

Propriétés du noeud datacollectionimport	Le type de données	Description de la propriété
label_type	chaîne	Si la propriété use_label_type est paramétrée sur True (vrai), cette propriété définit le type de libellé à importer. Il doit s'agir de l'un des types de libellé disponibles dans les données d'observation.
user_id	chaîne	Pour les bases de données exigeant une connexion explicite, indiquez l'ID et le mot de passe utilisateur nécessaires pour accéder à la source de données.
password	chaîne	
import_system_variables	Commun None All	Spécifie les variables système qui sont importées.
import_codes_variables	booléen	
import_sourcefile_variables	booléen	
import_multi_response	MultipleFlags Single	

Propriétés du noeud excelimport



Le noeud Import Excel permet d'importer des données issues de n'importe quelle version de Microsoft Excel. Aucune source de données ODBC n'est requise.

Tableau 36. Propriétés du noeud excelimport.

Propriétés du noeud excelimport	Le type de données	Description de la propriété
excel_file_type	Excel2003 Excel2007	
full_filename	chaîne	Nom du fichier complet, y compris son chemin d'accès.
use_named_range	Booléen	Si vous voulez utiliser un intervalle nommé. Si cette propriété est définie sur True (vrai), la propriété named_range est utilisée pour indiquer l'intervalle à lire. Les autres paramètres de feuille de calcul et d'intervalle de données sont ignorés.
named_range	chaîne	
worksheet_mode	Index Name	Indique si la feuille de calcul est définie par un index ou par un nom.
worksheet_index	entier	Index de la feuille de calcul à lire : 0 pour la première, 1 pour la deuxième, etc.
worksheet_name	chaîne	Nom de la feuille de calcul à lire.
data_range_mode	FirstNonBlank ExplicitRange	Indique le mode de détermination de l'intervalle.

Tableau 36. Propriétés du noeud excelimport (suite).

Propriétés du noeud excelimport	Le type de données	Description de la propriété
blank_rows	StopReading ReturnBlankRows	Quand la propriété data_range_mode est paramétrée sur <i>PremièreLigneRenseignée</i> , cette propriété indique comment traiter les lignes non renseignées.
explicit_range_start	chaîne	Quand la propriété data_range_mode est paramétrée sur <i>IntervalleExplicite</i> , cette propriété indique le point de départ de l'intervalle à lire.
explicit_range_end	chaîne	
noms_champ_lecture	Booléen	Indique si la première ligne de l'intervalle spécifié doit être utilisée pour les noms de champ (de colonne).

Propriétés du nœud evimport



Le nœud Enterprise View crée une connexion à un IBM SPSS Collaboration and Deployment Services Repository, vous permettant de lire des données Enterprise View dans un flux et de regrouper un modèle dans un scénario accessible depuis le référentiel par d'autres utilisateurs.

Tableau 37. Propriétés du noeud evimport.

Propriétés du noeud evimport	Le type de données	Description de la propriété
connection	liste	Propriété structurée : liste de paramètres constituant une connexion Enterprise View.
tablename	chaîne	Nom d'un tableau dans la Vue d'application.

Propriétés du noeud fixedfile



Le nœud Fixe permet d'importer les données de fichiers texte de longueur fixe, c'est-à-dire les fichiers dont les champs ne sont pas délimités, mais commencent au même endroit et sont de longueur fixe. Les données générées automatiquement ou héritées sont souvent stockées au format de longueur fixe.

Tableau 38. Propriétés du noeud fixedfile.

Propriétés du noeud fixedfile	Le type de données	Description de la propriété
record_len	nombre	Indique le nombre de caractères dans chaque enregistrement.
line_oriented	booléen	Ignore le caractère de nouvelle ligne figurant à la fin de chaque enregistrement.
decimal_symbol	Default Comma Period	Type de séparateur décimal utilisé dans votre source de données.
skip_header	nombre	Indique le nombre de lignes à ignorer au début du premier enregistrement. Utile pour ignorer les en-têtes de colonne.

Tableau 38. Propriétés du noeud fixedfile (suite).

Propriétés du noeud fixedfile	Le type de données	Description de la propriété
auto_recognize_datetime	booléen	Spécifie si les dates ou les heures sont automatiquement identifiées dans les données source.
lines_to_scan	nombre	
fields	liste	Propriétés structurées.
full_filename	chaîne	Nom complet (répertoire compris) du fichier à lire.
strip_spaces	None Left Right Both	Les espaces situés en début et en fin de chaîne sont supprimés lors de l'importation.
invalid_char_mode	Discard Replace	Supprime les caractères non valides (null, 0 ou tout caractère inexistant dans le codage en cours) de l'entrée de données ou remplace les caractères non valides par le symbole représentant un caractère.
invalid_char_replacement	chaîne	
use_custom_values	booléen	
custom_storage	Inconnu String Integer Real Time Date Timestamp	

Tableau 38. Propriétés du noeud fixedfile (suite).

Propriétés du noeud fixedfile	Le type de données	Description de la propriété
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "JJ-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	Cette propriété est applicable uniquement si un stockage personnalisé est indiqué.
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"	Cette propriété est applicable uniquement si un stockage personnalisé est indiqué.
custom_decimal_symbol	<i>champ</i>	Applicable uniquement si un stockage personnalisé est indiqué.
encoding	StreamDefault SystemDefault "UTF-8"	Indique la méthode de codage de texte.

Propriétés du noeud sasimport



Le noeud SAS permet d'importer des données SAS dans IBM SPSS Modeler.

Tableau 39. Propriétés du noeud sasimport.

Propriétés du noeud sasimport	Le type de données	Description de la propriété
format	Windows UNIX Transport SAS7 SAS8 SAS9	Format du fichier à importer.
full_filename	chaîne	Nom de fichier complet que vous entrez, y compris son chemin d'accès.
member_name	chaîne	Indique le membre à importer du fichier Transport SAS spécifié.
read_formats	booléen	Lit les formats de données (tels que les libellés de variable) du fichier de format spécifié.
full_format_filename	chaîne	
import_names	NamesAndLabels LabelsasNames	Spécifie la méthode de mappage des noms de variable et des libellés lors de l'importation.

Propriétés du noeud simgen



Le noeud Génération de simulation permet de générer facilement des données simulées, à partir de zéro en utilisant des distributions statistiques spécifiées par l'utilisateur ou automatiquement en utilisant les distributions issues de l'exécution d'un noeud Ajustement de simulation sur des données historiques existantes. Cela peut être utile si vous souhaitez évaluer le résultat d'un modèle prédictif en cas d'incertitude dans les entrées du modèle.

Tableau 40. Propriétés du noeud simgen.

Propriétés du noeud simgen	Type de données	Description de la propriété
fields	Propriété structurée	
correlations	Propriété structurée	
max_cases	entier	La valeur minimum est de 1000 ; la valeur maximum est de 2 147 483 647.
create_iteration_field	booléen	
iteration_field_name	chaîne	
replicate_results	booléen	
random_seed	entier	
overwrite_when_refitting	booléen	
parameter_xml	chaîne	Renvoie le paramètre Xml sous forme de chaîne.

Tableau 40. Propriétés du noeud *simgen* (suite).

Propriétés du noeud <i>simgen</i>	Type de données	Description de la propriété
distribution	Bernoulli Bêta Binomial Catégorielle Exponential Fixe Gamma Lognormal NegativeBinomialFailures NegativeBinomialTrials Normal Poisson Plage Triangulaire Uniforme Weibull	
bernoulli_prob	<i>nombre</i>	$0 \leq \text{bernoulli_prob} \leq 1$
beta_shape1	<i>nombre</i>	Doit être ≥ 0
beta_shape2	<i>nombre</i>	Doit être ≥ 0
beta_min	<i>nombre</i>	Facultatif. Doit être inférieur à <i>beta_max</i> .
beta_max	<i>nombre</i>	Facultatif. Doit être supérieur à <i>beta_min</i> .
binomial_n	<i>entier</i>	Doit être > 0
binomial_prob	<i>nombre</i>	$0 \leq \text{binomial_prob} \leq 1$
binomial_min	<i>nombre</i>	Facultatif. Doit être inférieur à <i>binomial_max</i> .
binomial_max	<i>nombre</i>	Facultatif. Doit être supérieur à <i>binomial_min</i> .
exponential_scale	<i>nombre</i>	Doit être > 0
exponential_min	<i>nombre</i>	Facultatif. Doit être inférieur à <i>exponential_max</i> .
exponential_max	<i>nombre</i>	Facultatif. Doit être supérieur à <i>exponential_min</i> .
fixed_value	<i>chaîne</i>	
gamma_shape	<i>nombre</i>	Doit être ≥ 0
gamma_scale	<i>nombre</i>	Doit être ≥ 0
gamma_min	<i>nombre</i>	Facultatif. Doit être inférieur à <i>gamma_max</i> .
gamma_max	<i>nombre</i>	Facultatif. Doit être supérieur à <i>gamma_min</i> .
lognormal_shape1	<i>nombre</i>	Doit être ≥ 0
lognormal_shape2	<i>nombre</i>	Doit être ≥ 0
lognormal_min	<i>nombre</i>	Facultatif. Doit être inférieur à <i>lognormal_max</i> .
lognormal_max	<i>nombre</i>	Facultatif. Doit être supérieur à <i>lognormal_min</i> .
negative_bin_failures_threshold	<i>nombre</i>	Doit être ≥ 0

Tableau 40. Propriétés du noeud simgen (suite).

Propriétés du noeud simgen	Type de données	Description de la propriété
negative_bin_failures_prob	nombre	$0 \leq \text{negative_bin_failures_prob} \leq 1$
negative_bin_failures_min	nombre	Facultatif. Doit être inférieur à negative_bin_failures_max.
negative_bin_failures_max	nombre	Facultatif. Doit être supérieur à negative_bin_failures_min.
negative_bin_trials_threshold	nombre	Doit être ≥ 0
negative_bin_trials_prob	nombre	$0 \leq \text{negative_bin_trials_prob} \leq 1$
negative_bin_trials_min	nombre	Facultatif. Doit être inférieur à negative_bin_trials_max.
negative_bin_trials_max	nombre	Facultatif. Doit être inférieur à negative_bin_trials_min.
normal_mean	nombre	
normal_sd	nombre	Doit être > 0
normal_min	nombre	Facultatif. Doit être inférieur à normal_max.
normal_max	nombre	Facultatif. Doit être supérieur à normal_min.
poisson_mean	nombre	Doit être ≥ 0
poisson_min	nombre	Facultatif. Doit être inférieur à poisson_max.
poisson_max	nombre	Facultatif. Doit être supérieur à poisson_min.
triangular_mode	nombre	$\text{triangular_min} \leq \text{triangular_mode} \leq \text{triangular_max}$
triangular_min	nombre	Doit être inférieur à triangular_mode.
triangular_max	nombre	Doit être supérieur à triangular_mode.
uniform_min	nombre	Doit être inférieur à uniform_max.
uniform_max	nombre	Doit être supérieur à uniform_min.
weibull_rate	nombre	Doit être ≥ 0
weibull_scale	nombre	Doit être ≥ 0
weibull_location	nombre	Doit être ≥ 0
weibull_min	nombre	Facultatif. Doit être inférieur à weibull_max.
weibull_max	nombre	Facultatif. Doit être supérieur à weibull_min.

La corrélation peut correspondre à n'importe quel nombre compris entre +1 et -1. Vous pouvez indiquer autant ou aussi peu de corrélations que vous le souhaitez. Toutes les corrélations non spécifiées sont définies sur zéro. Si des champs sont inconnus, la valeur de corrélation doit être définie sur la matrice (ou table) de corrélation et apparaît sous forme de texte rouge. Lorsque des champs sont inconnus, il est impossible d'exécuter le noeud.

Propriétés du noeud statisticsimport



Le noeud Fichier IBM SPSS Statistics lit les données du format de fichier *.sav* utilisé par IBM SPSS Statistics ainsi que des fichiers cache enregistrés dans IBM SPSS Modeler, qui utilisent le même format.

Les propriétés de ce noeud sont décrites dans «Propriétés du noeud statisticsimport», à la page 241.

Propriétés du noeud userinput



Le noeud Utilisateur représente une façon simple de créer des données synthétiques (à partir de zéro ou en modifiant des données existantes). Ceci est utile, par exemple, si vous souhaitez créer un jeu de données de test pour la modélisation.

Tableau 41. Propriétés du noeud userinput.

Propriétés du noeud userinput	Le type de données	Description de la propriété
data		Les données de chaque champ peuvent avoir des longueurs différentes, mais doivent correspondre au stockage du champ. ce champ est créé lorsque vous définissez les valeurs d'un champ absent. De plus, lorsque vous définissez les valeurs d'un champ sur une chaîne vide (" "), le champ indiqué est supprimé. Remarque : Les valeurs entrées pour cette propriété doivent être des chaînes et non des nombres. Par exemple, les nombres 1, 2, 3 et 4 doivent être entrés sous la forme "1 2 3 4".
names		Propriété structurée définissant ou renvoyant une liste de noms de champ générés par le noeud.
custom_storage	Inconnu String Integer Real Time Date Timestamp	Propriété saisie définissant ou renvoyant le stockage d'un champ.
data_mode	Combined Ordered	Si la propriété est définie sur Combined, les enregistrements sont générés pour chaque combinaison de valeurs définies et de valeurs min/max. Le nombre d'enregistrements générés est égal au produit du nombre de valeurs dans chaque champ. Si la propriété est définie sur Ordered, une valeur est copiée à partir des colonnes de chaque enregistrement pour générer une ligne de données. Le nombre d'enregistrements générés est égal au plus grand nombre de valeurs associées à un champ. Tous les champs comportant moins de valeurs de données seront remplis à l'aide de valeurs nulles.

Tableau 41. Propriétés du noeud userinput (suite).

Propriétés du noeud userinput	Le type de données	Description de la propriété
values		Cette propriété a été remplacée par data et ne doit plus être utilisée.

Propriétés du noeud variablefile



Le noeud Délimité lit les données de fichiers texte de longueur variable, c'est-à-dire les fichiers dont les enregistrements contiennent un nombre fixe de champs et un nombre variable de caractères. Ce noeud est également utile pour les fichiers contenant des textes d'en-tête de longueur fixe et certains types d'annotation.

Tableau 42. Propriétés du noeud variablefile.

Propriétés du noeud variablefile	Le type de données	Description de la propriété
skip_header	nombre	Indique le nombre de caractères à ignorer au début du premier enregistrement.
num_fields_auto	booléen	Détermine automatiquement le nombre de champs dans chaque enregistrement. Les enregistrements doivent être terminés par un caractère de retour à la ligne.
num_fields	nombre	Indiquez manuellement le nombre de champs dans chaque enregistrement.
delimit_space	booléen	Indique le caractère utilisé pour délimiter les champs dans le fichier.
delimit_tab	booléen	
delimit_new_line	booléen	
delimit_non_printing	booléen	
delimit_comma	booléen	Si la virgule sert à la fois de délimiteur de champs et de séparateur décimal pour les flux, paramétrez delimit_other sur true (vrai), puis indiquez la virgule comme délimiteur à l'aide de la propriété other.
delimit_other	booléen	Vous permet d'indiquer un délimiteur personnalisé à l'aide de la propriété other.
other	chaîne	Indique le délimiteur utilisé quand la propriété delimit_other est paramétrée sur true (vrai).
decimal_symbol	Default Comma Period	Indique le séparateur décimal utilisé dans la source de données.
multi_blank	booléen	Considère plusieurs délimiteurs non renseignés non adjacents comme un seul délimiteur.
noms_champ_lecture	booléen	Considère la première ligne du fichier de données comme des libellés de colonne.
strip_spaces	None Left Right Both	Les espaces situés en début et en fin de chaîne sont supprimés lors de l'importation.

Tableau 42. Propriétés du noeud variablefile (suite).

Propriétés du noeud variablefile	Le type de données	Description de la propriété
invalid_char_mode	Discard Replace	Supprime les caractères non valides (null, 0 ou tout caractère inexistant dans le codage en cours) de l'entrée de données ou remplace les caractères non valides par le symbole représentant un caractère.
invalid_char_replacement	chaîne	
break_case_by_newline	booléen	Indique que le délimiteur de ligne est le caractère de retour à la ligne.
lines_to_scan	nombre	Indique le nombre de lignes à analyser pour les types de données spécifiés.
auto_recognize_datetime	booléen	Spécifie si les dates ou les heures sont automatiquement identifiées dans les données source.
quotes_1	Discard PairAndDiscard IncludeAsText	Indique la manière dont les guillemets simples sont traités lors de l'importation.
quotes_2	Discard PairAndDiscard IncludeAsText	Indique la manière dont les guillemets doubles sont traités lors de l'importation.
full_filename	chaîne	Nom complet (répertoire compris) du fichier à lire.
use_custom_values	booléen	
custom_storage	Inconnu String Integer Real Time Date Timestamp	

Tableau 42. Propriétés du noeud variablefile (suite).

Propriétés du noeud variablefile	Le type de données	Description de la propriété
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "JJ-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	Applicable uniquement si un stockage personnalisé est indiqué.
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"	Applicable uniquement si un stockage personnalisé est indiqué.
custom_decimal_symbol	<i>champ</i>	Applicable uniquement si un stockage personnalisé est indiqué.
encoding	StreamDefault SystemDefault "UTF-8"	Indique la méthode de codage de texte.

Propriétés du noeud xmlimport



Le noeud source XML importe des données au format XML dans le flux. Vous pouvez importer un fichier ou tous les fichiers dans un répertoire. Vous pouvez aussi spécifier un fichier de schéma à partir duquel lire la structure XML.

Tableau 43. Propriétés du noeud xmlimport.

Propriétés du noeud xml import	Le type de données	Description de la propriété
read	single directory	Lit un seul fichier de données (par défaut), ou tous les fichiers XML d'un répertoire.
recurse	booléen	Spécifie s'il faut lire des fichiers XML supplémentaires dans tous les sous-répertoires du répertoire spécifié.
full_filename	chaîne	(requis) Chemin complet et nom de fichier du fichier XML à importer (si read = single).
directory_name	chaîne	(requis) Chemin complet et nom du répertoire à partir duquel importer les fichiers XML (si read = directory).
full_schema_filename	chaîne	Chemin complet et nom du fichier XSD ou DTD à partir duquel la structure XML est lue. Si vous omettez ce paramètre, la structure est lue à partir du fichier source XML.
records	chaîne	Expression XPath (par exemple. /auteur/nom) pour définir la limite de l'enregistrement. À chaque fois que cet élément est rencontré dans le fichier source, un nouvel enregistrement est créé.
mode	read specify	Lire toutes les données (par défaut), ou spécifier les éléments à lire.
fields		Liste des éléments (éléments et attributs) à importer. Chaque élément de la liste est une expression XPath.

Chapitre 10. Propriétés des noeuds d'opérations sur les lignes

Propriétés du noeud Ajouter



Le noeud Ajouter réalise la concaténation d'ensembles d'enregistrements. Il permet de combiner des jeux de données dont les structures sont similaires, mais les données différentes.

Tableau 44. Propriétés du noeud Ajouter.

Propriétés du noeud Ajouter	Le type de données	Description de la propriété
match_by	Position Name	Vous pouvez ajouter des jeux de données sur la base de la position des champs dans la source de données principale, ou du nom des champs dans les jeux de données d'entrée.
match_case	booléen	Active la distinction des majuscules/minuscules lors de la mise en correspondance des noms de champ.
include_fields_from	Principal All	
create_tag_field	booléen	
tag_field_name	chaîne	

Propriétés du noeud Agréger



Le noeud Agréger remplace une séquence d'enregistrements d'entrée par des enregistrements de sortie abrégés et agrégés.

Tableau 45. Propriétés du noeud Agréger.

Propriétés du noeud Agréger	Le type de données	Description de la propriété
keys	<i>[champ champ ... champ]</i>	Affiche les champs qui peuvent être utilisés comme clés pour l'agrégation. Par exemple, si vous avez choisi les champs-clés Sexe et Région, chaque combinaison unique de M et F avec les régions N et S (quatre combinaisons uniques) est associée à un enregistrement agrégé.
contiguous	booléen	Sélectionnez cette option si vous savez que tous les enregistrements présentant les mêmes valeurs clés sont regroupés dans l'entrée (par exemple, si l'entrée est triée dans les champs clés). Ainsi, vous améliorez les performances.
aggregates		Propriété structurée affichant les champs numériques dont les valeurs seront agrégées, ainsi que les modes d'agrégation sélectionnés.

Tableau 45. Propriétés du noeud Agréger (suite).

Propriétés du noeud Agréger	Le type de données	Description de la propriété
extension	chaîne	Indique un préfixe ou un suffixe pour les champs agrégés dupliqués (exemple ci-dessous).
add_as	Suffix Prefix	
inc_record_count	booléen	Crée un champ supplémentaire qui indique le nombre d'enregistrements d'entrée agrégés pour constituer chaque enregistrement agrégé.
count_field	chaîne	Indique le nom du champ de nombre d'enregistrements.

Propriétés du noeud Equilibrer



Le noeud Equilibrer corrige les déséquilibres survenant dans un jeu de données, de manière à respecter une condition précise. La règle d'équilibrage ajuste la proportion d'enregistrements présentant une condition True (vrai) par rapport au facteur indiqué.

Tableau 46. Propriétés du noeud Equilibrer.

Propriétés du noeud Equilibrer	Le type de données	Description de la propriété
directives		Propriété structurée permettant d'équilibrer la proportion de valeurs de champ en fonction du nombre spécifié (reportez-vous à l'exemple ci-dessous).
training_data_only	booléen	Spécifie que seules des données d'apprentissage devraient être équilibrées. Si aucun champ de partitionnement n'est présent dans le flux, cette option n'est pas prise en compte.

La propriété de noeud directives utilise le format suivant :

{ { chaîne de nombres } \ { chaîne de nombres } \ ... { chaîne de nombres } }.

Remarque : si des chaînes (utilisant des guillemets doubles) sont intégrées à l'expression, elles doivent être précédées du caractère d'échappement " \ ". Le caractère " \ " est également le caractère de continuation de ligne, qui vous permet d'aligner les arguments pour plus de clarté.

Propriétés du noeud derive_stb



Le noeud Cases-Espace-Heure dérive la valeur Cases-Espace-Heure des champs de latitude, de longitude et d'horodatage. Vous pouvez également identifier les valeurs Cases-Espace-Heure fréquentes comme des arrêts ou des ralentissements.

Tableau 47. Propriétés du noeud derive_stb.

Propriétés du noeud derive_stb	Type de données	Description de la propriété
mode	Enregistrements individuels : blocages	
latitude_field	champ	

Tableau 47. Propriétés du noeud *derive_stb* (suite).

Propriétés du noeud <i>derive_stb</i>	Type de données	Description de la propriété
<i>longitude_field</i>	<i>champ</i>	
<i>timestamp_field</i>	<i>champ</i>	
<i>hangout_density</i>	<i>densité</i>	Densité unique. Pour obtenir les valeurs de densité valides, voir <i>densities</i> .
<i>densities</i>	[<i>densité,densité,..., densité</i>]	Chaque densité est une chaîne ; par exemple STB_GH8_1DAY. Remarque : Il existe des limites de validité des densités. Pour geohash, les valeurs doivent être comprises entre GH1 et GH15. Pour la partie temporelle, les valeurs suivantes peuvent être utilisées : JAMAIS 1AN 1MOIS 1JOUR 12HEURES 8HEURES 6HEURES 4HEURES 3HEURES 2HEURES 1HEURE 30MINS 15MINS 10MINS 5MINS 2MINS 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SEC
<i>id_field</i>	<i>champ</i>	
<i>qualifying_duration</i>	1JOUR 12HEURES 8HEURES 6HEURES 4HEURES 3HEURES 2HEURES 1HEURE 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS	Doit être une chaîne.
<i>min_events</i>	<i>entier</i>	La valeur entière minimale valide est 2.
<i>qualifying_pct</i>	<i>entier</i>	Doit être compris entre 1 et 100.

Tableau 47. Propriétés du noeud *derive_stb* (suite).

Propriétés du noeud <i>derive_stb</i>	Type de données	Description de la propriété
add_extension_as	Préfixe Suffixe	
name_extension	chaîne	

Propriétés du noeud Distinguer



Le noeud Distinguer supprime les enregistrements en double, soit en incluant le premier enregistrement dans le flux de données, soit en le supprimant et en incluant ses doublons dans le flux de données.

Tableau 48. Propriétés du noeud *Distinguer*.

Propriétés du noeud <i>Distinguer</i>	Le type de données	Description de la propriété
mode	Include Discard	Vous pouvez enlever le premier enregistrement distinct du flux de données, ou l'isoler et transmettre les éventuels enregistrements dupliqués au flux de données.
grouping_fields	[<i>champ champ champ</i>]	Affiche les champs utilisés pour détecter les enregistrements identiques. Remarque : cette propriété est obsolète à partir de IBM SPSS Modeler 16.
composite_value	Propriété structurée	
composite_values	Propriété structurée	
inc_record_count	booléen	Crée un champ supplémentaire qui indique le nombre d'enregistrements d'entrée agrégés pour constituer chaque enregistrement agrégé.
count_field	chaîne	Indique le nom du champ de nombre d'enregistrements.
sort_keys	Propriété structurée.	Remarque : cette propriété est obsolète à partir de IBM SPSS Modeler 16.
default_ascending	booléen	
low_distinct_key_count	booléen	Spécifie que vous avez seulement un petit nombre d'enregistrements et/ou un petit nombre de valeurs uniques du/des champ(s)-clé(s).
keys_pre_sorted	booléen	Spécifie que tous les enregistrements avec les mêmes valeurs-clés sont regroupés dans l'entrée.
disable_sql_generation	booléen	

Propriétés du noeud Fusionner



Le noeud Fusionner permet de créer, à partir de plusieurs enregistrements d'entrée, un seul enregistrement de sortie contenant tout ou partie des champs d'entrée. Il sert notamment à fusionner des données provenant de différentes sources, telles que les données client internes et les données démographiques acquises.

Tableau 49. Propriétés du noeud Fusionner.

Propriétés du noeud Fusionner	Le type de données	Description de la propriété
method	Order Keys condition	Indique si les enregistrements font l'objet d'une fusion dans l'ordre dans lequel ils sont répertoriés dans les fichiers de données, si des champs-clés sont utilisés pour fusionner les enregistrements comportant des valeurs identiques dans ces champs-clés, ou si des enregistrements feront l'objet d'une fusion si une condition particulière est satisfaite.
condition	chaîne	Si method est défini sur Condition, spécifie la condition d'inclusion ou d'exclusion des enregistrements.
key_fields	[<i>champ champ champ</i>]	
common_keys	booléen	
join	Interne FullOuter PartialOuter Anti	
outer_join_tag.n	booléen	Dans cette propriété, <i>n</i> est le nom de la balise tel qu'il apparaît dans la boîte de dialogue de sélection du jeu de données. Remarque : vous pouvez indiquer plusieurs noms de balise, puisque n'importe quel nombre de jeux de données peut générer des enregistrements incomplets.
single_large_input	booléen	Indique si une fonction d'optimisation est utilisée lorsqu'une entrée est plus volumineuse que les autres.
single_large_input_tag	chaîne	Indique le nom de la balise tel qu'il apparaît dans la boîte de dialogue de sélection du jeu de données volumineux. L'utilisation de cette propriété diffère légèrement de celle de la propriété outer_join_tag (booléen/chaîne) car un seul jeu de données d'entrée peut être spécifié.
use_existing_sort_keys	booléen	Indique si les entrées sont déjà triées en fonction d'un ou de plusieurs champs-clés.
existing_sort_keys	[[<i>chaîne Ascending</i>] \ { <i>chaîne Descending</i> }]	Indique les champs déjà triés ainsi que le sens du tri.

Propriétés du noeud rfmaggregate



Le noeud agrégé Recency, Frequency, Monetary (RFM) vous permet de prendre les données de l'historique des transactions d'un client, d'en éliminer les éventuelles données inutilisées et de combiner le reste des données de transaction sur une seule ligne qui indique la date de la dernière consultation, le nombre de transactions réalisées et la valeur monétaire totale de ces transactions.

Tableau 50. Propriétés du noeud *rfmaggregate*.

Propriétés du noeud <i>rfmaggregate</i>	Le type de données	Description de la propriété
<i>relative_to</i>	Fixed Today	Spécifie la date à partir de laquelle la récence des transactions sera calculée.
<i>reference_date</i>	<i>date</i>	Disponible uniquement si Fixed est choisi dans <i>relative_to</i> .
<i>contiguous</i>	<i>booléen</i>	Si vos données sont pré-triées de façon à ce que tous les enregistrements avec le même ID apparaissent ensemble dans le flux de données, la sélection de cette option accélère le traitement.
<i>id_field</i>	<i>champ</i>	Spécifie le champ à utiliser pour identifier le client et ses transactions.
<i>date_field</i>	<i>champ</i>	Spécifie le champ de date à utiliser pour calculer la récence.
<i>value_field</i>	<i>champ</i>	Spécifie le champ à utiliser pour calculer la valeur monétaire.
<i>extension</i>	<i>chaîne</i>	Indique un préfixe ou un suffixe pour les champs agrégés en double.
<i>add_as</i>	Suffix Prefix	Spécifie si l'extension doit être ajoutée comme suffixe ou comme préfixe.
<i>discard_low_value_records</i>	<i>booléen</i>	Active l'utilisation du paramètre <i>discard_records_below</i> .
<i>discard_records_below</i>	<i>nombre</i>	Spécifie une valeur minimale sous laquelle les éventuels détails de la transaction ne seront pas utilisés pour calculer les totaux RFM. Les unités de valeur font référence au champ de <i>value</i> sélectionné.
<i>only_recent_transactions</i>	<i>booléen</i>	Active l'utilisation des paramètres <i>specify_transaction_date</i> ou <i>transaction_within_last</i> .
<i>specify_transaction_date</i>	<i>booléen</i>	
<i>transaction_date_after</i>	<i>date</i>	Disponible uniquement si <i>specify_transaction_date</i> est sélectionné. Spécifie la date de transaction après laquelle les enregistrements seront inclus dans votre analyse.
<i>transaction_within_last</i>	<i>nombre</i>	Disponible uniquement si <i>transaction_within_last</i> est sélectionné. Spécifie le nombre et le type de périodes (jours, semaines, mois ou années) provenant de la valeur de calcul de récence par rapport à la date après laquelle les enregistrements seront inclus dans votre analyse.
<i>transaction_scale</i>	Days Weeks Mois Years	Disponible uniquement si <i>transaction_within_last</i> est sélectionné. Spécifie le nombre et le type de périodes (jours, semaines, mois ou années) provenant de la valeur de calcul de récence par rapport à la date après laquelle les enregistrements seront inclus dans votre analyse.
<i>save_r2</i>	<i>booléen</i>	Affiche la date de l'avant-dernière transaction pour chaque client.

Tableau 50. Propriétés du noeud *rfmaggregate* (suite).

Propriétés du noeud <i>rfmaggregate</i>	Le type de données	Description de la propriété
save_r3	<i>booléen</i>	Disponible uniquement si save_r2 est sélectionné. Affiche la date de la troisième transaction la plus récente pour chaque client.

Propriétés du noeud Rprocess



Le noeud de processus R vous permet d'utiliser les données d'un flux IBM(r) SPSS(r) Modeler et de modifier ces données à l'aide de votre propre script R personnalisé. Une fois que les données sont modifiées, elles sont renvoyées au flux.

Tableau 51. Propriétés du noeud *Rprocess*.

Propriétés du noeud <i>Rprocess</i>	Type de données	Description de la propriété
syntax	<i>chaîne</i>	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	<i>booléen</i>	
convert_datetime_class	POSIXct POSIXlt	
convert_missing	<i>booléen</i>	

Propriétés du noeud Echantillon



Le noeud Echantillonner sélectionne un sous-ensemble d'enregistrements. Divers types d'échantillons sont pris en charge, notamment les échantillons stratifiés, en cluster et non aléatoires (structurés). L'échantillonnage peut être utile pour améliorer les performances et pour sélectionner des groupes d'enregistrements associés ou des transactions pour analyse.

Tableau 52. Propriétés du noeud *Echantillon*.

Propriétés du noeud <i>Echantillon</i>	Le type de données	Description de la propriété
method	Simple Complexe	
mode	Include Discard	Enlève ou isole les enregistrements qui correspondent à la condition indiquée.
sample_type	First OneInN RandomPct	Indique la méthode d'échantillonnage.
first_n	<i>entier</i>	Les enregistrements jusqu'au point de césure indiqué sont inclus ou exclus.
one_in_n	<i>nombre</i>	Inclut ou exclut un enregistrement tous les <i>n</i> enregistrements.
rand_pct	<i>nombre</i>	Indique le pourcentage d'enregistrements à inclure ou à exclure.

Tableau 52. Propriétés du noeud Echantillon (suite).

Propriétés du noeud Echantillon	Le type de données	Description de la propriété
use_max_size	booléen	Active l'utilisation du paramètre maximum_size.
maximum_size	entier	Indiquez la taille maximale des échantillons à inclure dans le flux de données ou à exclure du flux. Cette option est redondante et par conséquent désactivée si les options First et Include sont sélectionnées.
set_random_seed	booléen	Active l'utilisation du paramètre de valeur de départ aléatoire.
random_seed	entier	Indique la valeur utilisée en tant que valeur de départ aléatoire.
complex_sample_type	Random Systematic	
sample_units	Proportions Counts	
sample_size_proportions	Fixed Custom Variable	
sample_size_counts	Fixed Custom Variable	
fixed_proportions	nombre	
fixed_counts	entier	
variable_proportions	champ	
variable_counts	champ	
use_min_stratum_size	booléen	
minimum_stratum_size	entier	Cette option s'applique uniquement lorsqu'un échantillon Complexe est utilisé avec Sample units=Proportions.
use_max_stratum_size	booléen	
maximum_stratum_size	entier	Cette option s'applique uniquement lorsqu'un échantillon Complexe est utilisé avec Sample units=Proportions.
clusters	champ	
stratify_by	[champ1 ... champN]	
specify_input_weight	booléen	
input_weight	champ	
new_output_weight	chaîne	
sizes_proportions	[[string valeur de chaîne]{string valeur de chaîne}...]	Si sample_units=proportions et sample_size_proportions=Custom, spécifie une valeur pour chaque combinaison possible de valeurs pour les champs de stratification.
default_proportion	nombre	
sizes_counts	[[string valeur de chaîne]{string valeur de chaîne}...]	Spécifie une valeur pour chaque combinaison de valeurs possible pour les champs de stratification. L'utilisation est similaire à celle de sizes_proportions, mais en spécifiant un entier au lieu d'une proportion.

Tableau 52. Propriétés du noeud Echantillon (suite).

Propriétés du noeud Echantillon	Le type de données	Description de la propriété
default_count	nombre	

Propriétés du noeud Sélectionner



Le noeud Sélectionner permet de sélectionner ou d'exclure des sous-ensembles d'enregistrements d'un flux de données sur la base d'une condition spécifique. Par exemple, vous pouvez sélectionner les enregistrements qui appartiennent à un secteur de ventes particulier.

Tableau 53. Propriétés du noeud Sélectionner.

Propriétés du noeud Sélectionner	Le type de données	Description de la propriété
mode	Include Discard	Indique si les enregistrements sélectionnés sont à enlever ou à isoler.
condition	chaîne	Condition d'inclusion ou d'exclusion des enregistrements.

Propriétés du noeud Trier



Le noeud Trier trie les enregistrements par ordre croissant ou décroissant, en fonction de la valeur d'un ou de plusieurs champs.

Tableau 54. Propriétés du noeud Trier.

Propriétés du noeud Trier	Le type de données	Description de la propriété
keys	[[chaîne Ascending] \ {chaîne Descending}]	Spécifie les champs servant de base au tri. Si aucune direction n'est indiquée, la valeur utilisée est celle par défaut.
default_ascending	booléen	Indique l'ordre de tri par défaut.
use_existing_keys	booléen	Indique si le tri est optimisé à l'aide de l'ordre de tri précédent des champs déjà triés.
existing_keys		Indique les champs déjà triés ainsi que le sens du tri. Utilise le même format que la propriété keys.

Propriétés du noeud streamingts



Le noeud Flux TS (Streaming TS) crée et évalue des modèles de séries chronologiques en une seule étape, sans avoir besoin d'un noeud Intervalles de temps.

Tableau 55. Propriétés du noeud streamings.

Propriétés du noeud streamings	Type de données	Description de la propriété
custom_fields	booléen	Si custom_fields=false, on utilise les paramètres provenant d'un noeud Type en amont. Si custom_fields=true, vous devez spécifier targets et inputs.
targets	[champ1...champN]	
inputs	[champ1...champN]	
method	ExpertModeler Exsmooth Arima	
calculate_conf	booléen	
conf_limit_pct	réel	
use_time_intervals_node	booléen	Si use_time_intervals_node=true, on utilise les paramètres provenant d'un noeud Intervalle de temps en amont. Sinon, vous devez spécifier interval_offset_position, interval_offset et interval_type.
interval_offset_position	LastObservation LastRecord	LastObservation fait référence à dernière observation valide . LastRecord fait référence à Compter depuis le dernier enregistrement .
interval_offset	nombre	
interval_type	Periods Years Quarters Months WeeksNonPeriodic DaysNonPeriodic HoursNonPeriodic MinutesNonPeriodic SecondsNonPeriodic	
events	champs	
expert_modeler_method	AllModels Exsmooth Arima	
consider_seasonal	booléen	
detect_outliers	booléen	
expert_outlier_additive	booléen	
expert_outlier_level_shift	booléen	
expert_outlier_innovational	booléen	
expert_outlier_transient	booléen	
expert_outlier_seasonal_additive	booléen	
expert_outlier_local_trend	booléen	
expert_outlier_additive_patch	booléen	
exsmooth_model_type	Simple HoltLinearTrend BrownLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	

Tableau 55. Propriétés du noeud `streamingts` (suite).

Propriétés du noeud <code>streamingts</code>	Type de données	Description de la propriété
<code>exsmooth_transformation_type</code>	None SquareRoot NaturalLog	
<code>arma_p</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles
<code>arma_d</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles
<code>arma_q</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles
<code>arma_sp</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles
<code>arma_sd</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles
<code>arma_sq</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles
<code>arma_transformation_type</code>	None SquareRoot NaturalLog	Même propriété que pour le noeud de modélisation des séries temporelles
<code>arma_include_constant</code>	<i>booléen</i>	Même propriété que pour le noeud de modélisation des séries temporelles
<code>tf_arma_p.nomchamp</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles. Pour les fonctions de transfert.
<code>tf_arma_d.nomchamp</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles. Pour les fonctions de transfert.
<code>tf_arma_q.nomchamp</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles. Pour les fonctions de transfert.
<code>tf_arma_sp.nomchamp</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles. Pour les fonctions de transfert.
<code>tf_arma_sd.nomchamp</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles. Pour les fonctions de transfert.
<code>tf_arma_sq.nomchamp</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles. Pour les fonctions de transfert.
<code>tf_arma_delay.nomchamp</code>	<i>entier</i>	Même propriété que pour le noeud de modélisation des séries temporelles. Pour les fonctions de transfert.
<code>tf_arma_transformation_type.nomchamp</code>	None SquareRoot NaturalLog	
<code>arma_detect_outlier_mode</code>	None Automatic	
<code>arma_outlier_additive</code>	<i>booléen</i>	
<code>arma_outlier_level_shift</code>	<i>booléen</i>	
<code>arma_outlier_innovational</code>	<i>booléen</i>	

Tableau 55. Propriétés du noeud *streamingts* (suite).

Propriétés du noeud <i>streamingts</i>	Type de données	Description de la propriété
<i>arma_outlier_transient</i>	<i>booléen</i>	
<i>arma_outlier_seasonal_additive</i>	<i>booléen</i>	
<i>arma_outlier_local_trend</i>	<i>booléen</i>	
<i>arma_outlier_additive_patch</i>	<i>booléen</i>	
<i>deployment_force_rebuild</i>	<i>booléen</i>	
<i>deployment_rebuild_mode</i>	Count Percent	
<i>deployment_rebuild_count</i>	<i>nombre</i>	
<i>deployment_rebuild_pct</i>	<i>nombre</i>	
<i>deployment_rebuild_field</i>	< <i>champ</i> >	

Chapitre 11. Propriétés des nœuds d'opérations sur les champs

Propriétés du nœud Anonymiser



Le nœud Anonymiser transforme la façon dont les noms et les valeurs des champs sont représentés en aval, masquant ainsi les données d'origine. Cela peut s'avérer utile si vous souhaitez permettre à d'autres utilisateurs de générer des modèles utilisant des données confidentielles, par exemple des noms de clients ou autre.

Tableau 56. Propriétés du nœud Anonymiser.

Propriétés du nœud Anonymiser	Le type de données	Description de la propriété
enable_anonymize	<i>booléen</i>	Lorsque cette propriété est définie sur T, elle active l'anonymisation des valeurs de champ (revient à sélectionner Oui pour ce champ dans la colonne Anonymiser des valeurs).
use_prefix	<i>booléen</i>	Lorsque cette propriété est définie sur T, un préfixe personnalisé sera utilisé, s'il en existe un. S'applique aux champs qui seront anonymisés grâce à la méthode de hachage et revient à sélectionner la case d'option Personnalisé dans la boîte de dialogue Remplacer les valeurs pour ce champ.
prefix	<i>chaîne</i>	Revient à taper un préfixe dans la zone de texte de la boîte de dialogue Remplacer les valeurs. Le préfixe par défaut est la valeur par défaut si rien d'autre n'a été indiqué.
transformation	Random Fixed	Détermine si les paramètres de transformation pour un champ anonymisé par la méthode de transformation seront aléatoires ou fixes.
set_random_seed	<i>booléen</i>	Lorsque cette propriété est définie sur T, la valeur de départ indiquée sera utilisée (si la propriété transformation est également définie sur Aléatoire).
random_seed	<i>entier</i>	Lorsque set_random_seed est défini sur T, il s'agit de la valeur de départ pour le nombre aléatoire.
scale	<i>nombre</i>	Lorsque la propriété transformation est définie sur Fixe, cette valeur sert pour la mise à l'échelle. La valeur d'échelle maximale est normalement 10, mais elle peut être diminuée pour éviter tout dépassement.
translate	<i>nombre</i>	Lorsque la propriété transformation est définie sur Fixe, cette valeur sert à la « traduction ». La valeur maximale de conversion est normalement 1 000, mais elle peut être diminuée pour éviter tout dépassement.

Propriétés du nœud autodataprep



Le nœud de préparation automatisée de données (ADP) peut analyser vos données, identifier des corrections et filtrer des champs qui sont problématiques ou qui sont peu susceptibles d'être utiles. Il peut aussi créer de nouveaux attributs le cas échéant et améliorer la performance au moyen de techniques de filtrage et d'échantillonnage intelligentes. Vous pouvez utiliser le nœud de manière totalement automatisée, en laissant le nœud choisir et appliquer les corrections, ou vous pouvez prévisualiser les modifications avant qu'elles ne soient mises en place et les accepter, les rejeter ou les modifier selon les besoins.

Tableau 57. Propriétés du noeud autodataprep.

Propriétés du noeud autodataprep	Le type de données	Description de la propriété
objective	Balanced Vitesse Accuracy Custom	
custom_fields	booléen	Si la valeur est définie sur true (vrai), elle permet de spécifier les champs cible, entrée et d'autres champs pour le noeud actuel. Si elle est définie sur false (faux), les paramètres actuels provenant d'un noeud type en amont sont utilisés.
target	champ	Spécifie un champ cible unique.
inputs	[champ1 ... champN]	Champs d'entrée ou prédicteur utilisés par le modèle.
use_frequency	booléen	
frequency_field	champ	
use_weight	booléen	
weight_field	champ	
excluded_fields	Filter None	
if_fields_do_not_match	StopExecution ClearAnalysis	
prepare_dates_and_times	booléen	Contrôle l'accès à tous les champs date et heure
compute_time_until_date	booléen	
reference_date	Today Fixed	
fixed_date	date	
units_for_date_durations	Automatique Fixed	
fixed_date_units	Years Mois Days	
compute_time_until_time	booléen	
reference_time	CurrentTime Fixed	
fixed_time	time	
units_for_time_durations	Automatique Fixed	
fixed_date_units	Hours Minutes Secondes	
extract_year_from_date	booléen	
extract_month_from_date	booléen	
extract_day_from_date	booléen	
extract_hour_from_time	booléen	

Tableau 57. Propriétés du noeud autodataprep (suite).

Propriétés du noeud autodataprep	Le type de données	Description de la propriété
extract_minute_from_time	booléen	
extract_second_from_time	booléen	
exclude_low_quality_inputs	booléen	
exclude_too_many_missing	booléen	
maximum_percentage_missing	nombre	
exclude_too_many_categories	booléen	
maximum_number_categories	nombre	
exclude_if_large_category	booléen	
maximum_percentage_category	nombre	
prepare_inputs_and_target	booléen	
adjust_type_inputs	booléen	
adjust_type_target	booléen	
reorder_nominal_inputs	booléen	
reorder_nominal_target	booléen	
replace_outliers_inputs	booléen	
replace_outliers_target	booléen	
replace_missing_continuous_inputs	booléen	
replace_missing_continuous_target	booléen	
replace_missing_nominal_inputs	booléen	
replace_missing_nominal_target	booléen	
replace_missing_ordinal_inputs	booléen	
replace_missing_ordinal_target	booléen	
maximum_values_for_ordinal	nombre	
minimum_values_for_continuous	nombre	
outlier_cutoff_value	nombre	
outlier_method	Replace Delete	
rescale_continuous_inputs	booléen	
rescaling_method	MinMax ZScore	
min_max_minimum	nombre	
min_max_maximum	nombre	
z_score_final_mean	nombre	
z_score_final_sd	nombre	
rescale_continuous_target	booléen	
target_final_mean	nombre	
target_final_sd	nombre	
transform_select_input_fields	booléen	
maximize_association_with_target	booléen	
p_value_for_merging	nombre	

Tableau 57. Propriétés du noeud autodataprep (suite).

Propriétés du noeud autodataprep	Le type de données	Description de la propriété
merge_ordinal_features	booléen	
merge_nominal_features	booléen	
minimum_cases_in_category	nombre	
bin_continuous_fields	booléen	
p_value_for_binning	nombre	
perform_feature_selection	booléen	
p_value_for_selection	nombre	
perform_feature_construction	booléen	
transformed_target_name_extension	chaîne	
transformed_inputs_name_extension	chaîne	
constructed_features_root_name	chaîne	
years_duration_name_extension	chaîne	
months_duration_name_extension	chaîne	
days_duration_name_extension	chaîne	
hours_duration_name_extension	chaîne	
minutes_duration_name_extension	chaîne	
seconds_duration_name_extension	chaîne	
year_cyclical_name_extension	chaîne	
month_cyclical_name_extension	chaîne	
day_cyclical_name_extension	chaîne	
hour_cyclical_name_extension	chaîne	
minute_cyclical_name_extension	chaîne	
second_cyclical_name_extension	chaîne	

Propriétés du noeud Discrétiser



Le noeud Discrétiser crée automatiquement des champs nominaux (ensemble) sur la base des valeurs d'un ou de plusieurs champs continus (intervalle numérique) existants. Par exemple, vous pouvez transformer un champ continu de revenus en un nouveau champ catégoriel contenant des groupes de revenus comme écarts par rapport à la moyenne. Une fois les intervalles du nouveau champ créés, vous pouvez générer un noeud Dériver à partir des points de césure.

Tableau 58. Propriétés du noeud Discrétiser.

Propriétés du noeud Discrétiser	Le type de données	Description de la propriété
fields	[champ1 champ2 ... champn]	Champs continus (intervalle numérique) en attente de transformation. Vous pouvez créer des intervalles pour plusieurs champs simultanément.
method	FixedWidth EqualCount Rank SDev Optimal	Méthode utilisée pour déterminer les points de césure des nouveaux intervalles de champ (catégories).

Tableau 58. Propriétés du noeud Discrétiser (suite).

Propriétés du noeud Discrétiser	Le type de données	Description de la propriété
recalculate_bins	Always IfNecessary	Indique si les intervalles sont recalculés et les données placées dans l'intervalle approprié à chaque exécution du noeud ou si les données sont uniquement ajoutées aux intervalles existants et aux nouveaux intervalles éventuellement ajoutés.
fixed_width_name_extension	chaîne	L'extension par défaut est <i>_BIN</i> .
fixed_width_add_as	Suffix Prefix	Indique si l'extension est ajoutée à la fin (suffixe) ou au début (préfixe) du nom du champ. L'extension par défaut est <i>income_BIN</i> .
fixed_bin_method	Width Count	
fixed_bin_count	entier	Désigne l'entier déterminant le nombre d'intervalles de largeur fixe (catégories) des nouveaux champs.
fixed_bin_width	réel	Valeur (entier ou réel) permettant de calculer la largeur de l'intervalle.
equal_count_name_extension	chaîne	L'extension par défaut est <i>_TILE</i> .
equal_count_add_as	Suffix Prefix	Indique une extension, suffixe ou préfixe, utilisée pour le nom de champ généré à l'aide de centiles standard. L'extension par défaut est <i>_TILE</i> plus <i>N</i> , <i>N</i> étant le numéro du quantile.
tile4	booléen	Génère quatre intervalles de quintile, chacun contenant 25 % des observations.
tile5	booléen	Génère cinq intervalles de quintiles.
tile10	booléen	Génère 10 intervalles de déciles.
tile20	booléen	Génère 20 intervalles de vingtiles.
tile100	booléen	Génère 100 intervalles de centiles.
use_custom_tile	booléen	
custom_tile_name_extension	chaîne	L'extension par défaut est <i>_TILEN</i> .
custom_tile_add_as	Suffix Prefix	
custom_tile	entier	
equal_count_method	RecordCount ValueSum	La méthode RecordCount vise à affecter un nombre égal d'enregistrements à chaque intervalle, tandis que la méthode ValueSum affecte les enregistrements de sorte que la somme des valeurs de chaque intervalle soit égale.
tied_values_method	Next Current Random	Spécifie l'intervalle dans lequel les données de valeur ex æquo doivent être placées.
rank_order	Ascending Descending	Cette propriété peut avoir la valeur Ascending (la valeur la plus faible est marquée 1) ou Descending (la valeur la plus élevée est marquée 1).

Tableau 58. Propriétés du noeud Discrétiser (suite).

Propriétés du noeud Discrétiser	Le type de données	Description de la propriété
rank_add_as	Suffix Prefix	Cette option s'applique au rang, au rang fractionnaire et au rang de pourcentage.
rang	booléen	
rank_name_extension	chaîne	L'extension par défaut est <i>_RANK</i> .
rank_fractional	booléen	Permet de classer les observations dans lesquelles la valeur du nouveau champ équivaut au rang divisé par la somme des pondérations des observations non manquantes. Les rangs fractionnaires sont compris dans l'intervalle 0–1.
rank_fractional_name_extension	chaîne	L'extension par défaut est <i>_F_RANK</i> .
rank_pct	booléen	Chaque rang est divisé par le nombre d'enregistrements avec valeurs valides et multiplié par 100. Les rangs fractionnaires de pourcentage sont compris dans l'intervalle 1–100.
rank_pct_name_extension	chaîne	L'extension par défaut est <i>_P_RANK</i> .
sdev_name_extension	chaîne	
sdev_add_as	Suffix Prefix	
sdev_count	One Two Three	
optimal_name_extension	chaîne	L'extension par défaut est <i>_OPTIMAL</i> .
optimal_add_as	Suffix Prefix	
optimal_supervisor_field	champ	Champ choisi comme champ de superviseur et auquel les champs sélectionnés pour la création d'intervalles sont associés.
optimal_merge_bins	booléen	Indique que tous les intervalles présentant peu d'observations seront ajoutés à un intervalle voisin plus grand.
optimal_small_bin_threshold	entier	
optimal_pre_bin	booléen	Indique que la pré-crédation d'intervalles du jeu de données va avoir lieu.
optimal_max_bins	entier	Définit une limite supérieure afin d'éviter de créer un nombre d'intervalles trop important.
optimal_lower_end_point	Inclusive Exclusive	
optimal_first_bin	Unbounded Bounded	
optimal_last_bin	Unbounded Bounded	

Propriétés du noeud Dériver



Le noeud Dériver modifie les valeurs de données ou crée des nouveaux champs à partir d'un ou de plusieurs champs existants. Il crée des champs de type formule, indicateur, ensemble, nominal, statistiques, comptage et conditionnel.

Tableau 59. Propriétés du noeud Dériver.

Propriétés du noeud Dériver	Le type de données	Description de la propriété
new_name	chaîne	Nom du nouveau champ.
mode	Single Multiple	Spécifie un ou plusieurs champs.
fields	[<i>champ champ champ</i>]	Utilisé en mode Multiple uniquement pour sélectionner plusieurs champs.
name_extension	chaîne	Indique l'extension des nouveaux noms de champ.
add_as	Suffix Prefix	Ajoute l'extension du nom de champ en tant que préfixe (au début) ou en tant que suffixe (à la fin).
result_type	Formula Indicateur Set Etat Count Conditional	Les six types de nouveau champ que vous pouvez créer.
formula_expr	chaîne	Expression de calcul de la valeur du nouveau champ dans un noeud Dériver
flag_expr	chaîne	
flag_true	chaîne	
flag_false	chaîne	
set_default	chaîne	
set_value_cond	chaîne	Propriété structurée (définition de la condition associée à une valeur donnée).
state_on_val	chaîne	Indique la valeur du nouveau champ si la condition Activé est vérifiée.
state_off_val	chaîne	Indique la valeur du nouveau champ si la condition Désactivé est vérifiée.
state_on_expression	chaîne	
state_off_expression	chaîne	
state_initial	On Off	Affecte à chaque enregistrement du nouveau champ la valeur initiale On ou Off. Cette valeur peut changer au fur et à mesure que les conditions sont respectées.
count_initial_val	chaîne	
count_inc_condition	chaîne	
count_inc_expression	chaîne	
count_reset_condition	chaîne	
cond_if_cond	chaîne	

Tableau 59. Propriétés du noeud Dériver (suite).

Propriétés du noeud Dériver	Le type de données	Description de la propriété
cond_then_expr	chaîne	
cond_else_expr	chaîne	

Propriétés du noeud Ensemble



Le noeud Ensemble combine deux ou plusieurs nuggets de modèles pour obtenir des prévisions plus précises que celles acquises à partir d'un modèle quelconque.

Tableau 60. Propriétés du noeud Ensemble.

Propriétés du noeud Ensemble	Le type de données	Description de la propriété
ensemble_target_field	champ	Indique le champ cible pour tous les modèles utilisés dans l'ensemble.
filter_individual_model_output	booléen	Indique si les résultats de scoring de modèles individuels doivent être supprimés.
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity	Indique la méthode utilisée pour déterminer le score de l'ensemble. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ indicateur.
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	Indique la méthode utilisée pour déterminer le score de l'ensemble. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ nominal.
flag_voting_tie_selection	Random HighestConfidence RawPropensity AdjustedPropensity	Si une méthode de vote est sélectionnée, indique la manière dont les ex æquo sont résolus. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ indicateur.
set_voting_tie_selection	Random HighestConfidence	Si une méthode de vote est sélectionnée, indique la manière dont les ex æquo sont résolus. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ nominal.
calculate_standard_error	booléen	Si le champ cible est continu, un calcul d'erreur standard est exécuté par défaut pour calculer la différence entre les valeurs mesurées ou estimées et les valeurs réelles, et pour montrer la correspondance proche de ces évaluations.

Propriétés du noeud Remplacer



Le noeud Remplacer permet de remplacer les valeurs de champ et de modifier le type de stockage. Vous pouvez décider de remplacer les valeurs reposant sur une condition CLEM, telle que @BLANK(@FIELD). Vous pouvez également choisir de remplacer tous les blancs ou toutes les valeurs nulles par une valeur précise. Un noeud Remplacer est souvent associé à un noeud type pour remplacer les valeurs manquantes.

Tableau 61. Propriétés du noeud Remplacer.

Propriétés du noeud Remplacer	Le type de données	Description de la propriété
fields	[<i>champ champ champ</i>]	Champs du jeu de données dont les valeurs sont examinées et remplacées.
replace_mode	Always Conditional Blanc Valeur nulle BlankAndNull	Vous pouvez remplacer toutes les valeurs, les valeurs vides ou les valeurs nulles, ou effectuer le remplacement en fonction d'une condition définie.
condition	<i>chaîne</i>	
replace_with	<i>chaîne</i>	

Propriétés du noeud Filtrer



Le noeud Filtrer filtre (supprime) les champs, les renomme et les mappe entre un noeud source et un autre.

Utilisation de la propriété default_include. Le paramétrage de la valeur de la propriété default_include n'implique pas l'inclusion ou l'exclusion automatique de tous les champs ; il détermine simplement la valeur par défaut pour la sélection actuelle. Ceci équivaut, d'un point de vue fonctionnel, à cliquer sur le bouton **Inclure les champs par défaut** dans la boîte de dialogue du noeud Filtrer.

Tableau 62. Propriétés du noeud Filtrer.

Propriétés du noeud Filtrer	Le type de données	Description de la propriété
default_include	<i>booléen</i>	Propriété saisie permettant d'indiquer si le comportement par défaut consiste à transmettre ou à filtrer les champs. Le paramétrage de cette propriété n'implique pas l'inclusion ou l'exclusion automatique de tous les champs ; il détermine simplement si les champs sélectionnés sont inclus ou exclus par défaut.
include	<i>booléen</i>	Propriété saisie (inclusion et suppression de champs).
new_name	<i>chaîne</i>	

Propriétés du noeud Historiser



Le noeud Historiser crée des champs contenant des données provenant de champs d'enregistrements antérieurs. Les noeuds Historiser sont souvent utilisés pour les données séquentielles, telles que les séries temporelles. Avant d'utiliser un noeud Historiser, vous pouvez trier les données à l'aide d'un noeud Trier.

Tableau 63. Propriétés du noeud Historiser.

Propriétés du noeud Historiser	Le type de données	Description de la propriété
fields	[<i>champ champ champ</i>]	Champs pour lesquels vous souhaitez un historique.
offset	<i>nombre</i>	Indique le dernier enregistrement (avant l'enregistrement actuel) à partir duquel extraire les valeurs de champ historiques.
span	<i>nombre</i>	Indique le nombre d'enregistrements précédents desquels extraire des valeurs.
unavailable	Discard Leave Fill	Propriété destinée à la gestion des enregistrements qui n'ont aucune valeur historique ; fait généralement référence aux premiers enregistrements (situés au début du jeu de données) et pour lesquels il n'existe aucun enregistrement précédent à utiliser en tant qu'historique.
fill_with	Chaîne Nombre	Indique la valeur ou la chaîne à utiliser pour les enregistrements dans lesquels aucune valeur d'historique n'est disponible.

Propriétés du noeud Partitionner



Le noeud Partitionner génère un champ de partition qui répartit les données dans des sous-ensembles distincts pour les étapes d'apprentissage, de test et de validation de la création d'un modèle.

Tableau 64. Propriétés du noeud Partitionner.

Propriétés du noeud Partitionner	Le type de données	Description de la propriété
new_name	<i>chaîne</i>	Nom du champ de partition généré par le noeud.
create_validation	<i>booléen</i>	Indique si une partition de validation doit être créée.
training_size	<i>entier</i>	Pourcentage des enregistrements (0 à 100) à allouer à la partition d'apprentissage.
testing_size	<i>entier</i>	Pourcentage des enregistrements (0 à 100) à allouer à la partition de test.
validation_size	<i>entier</i>	Pourcentage des enregistrements (0 à 100) à allouer à la partition de validation. Ignoré si aucune partition de validation n'est créée.
training_label	<i>chaîne</i>	Libellé de la partition d'apprentissage.
testing_label	<i>chaîne</i>	Libellé de la partition de test.

Tableau 64. Propriétés du noeud Partitionner (suite).

Propriétés du noeud Partitionner	Le type de données	Description de la propriété
validation_label	chaîne	Libellé de la partition de validation. Ignoré si aucune partition de validation n'est créée.
value_mode	System SystemAndLabel Label	Indique les valeurs utilisées pour représenter chaque partition dans les données. Par exemple, l'échantillon d'apprentissage peut être représenté par l'entier système 1, le libellé Training ou une combinaison des deux : 1_Training.
set_random_seed	booléen	Indique si vous devez utiliser une valeur de départ aléatoire définie par l'utilisateur.
random_seed	entier	Valeur de départ aléatoire définie par l'utilisateur. Pour utiliser cette valeur, set_random_seed doit avoir la valeur True (vrai).
enable_sql_generation	booléen	Spécifie s'il faut utiliser la répercussion SQL pour affecter des enregistrements à des partitions.
unique_field		Spécifie le champ d'entrée utilisé pour vérifier que des enregistrements sont attribués à des partitions de manière aléatoire mais répétitive. Pour utiliser cette valeur, enable_sql_generation doit avoir la valeur True (vrai).

Propriétés du noeud Recoder



Le noeud Recoder permet de transformer un ensemble de valeurs catégorielles en un autre. La recodification est utile pour réduire des catégories ou regrouper des données à analyser.

Tableau 65. Propriétés du noeud Recoder.

Propriétés du noeud Recoder	Le type de données	Description de la propriété
mode	Single Multiple	La propriété Single recodifie les catégories d'un champ. La propriété Multiple active les options permettant de transformer plusieurs champs simultanément.
replace_field	booléen	
field	chaîne	Utilisé en mode Simple uniquement.
new_name	chaîne	Utilisé en mode Simple uniquement.
fields	[champ1 champ2 ... champn]	Utilisé en mode Multiple uniquement.
name_extension	chaîne	Utilisé en mode Multiple uniquement.
add_as	Suffix Prefix	Utilisé en mode Multiple uniquement.
reclassify	chaîne	Propriété structurée (valeur des champs).
use_default	booléen	Utilisez la valeur par défaut.
default	chaîne	Spécifiez une valeur par défaut.

Tableau 65. Propriétés du noeud Recoder (suite).

Propriétés du noeud Recoder	Le type de données	Description de la propriété
pick_list	[chaîne chaîne ... chaîne]	Permet à l'utilisateur d'importer la liste des nouvelles valeurs connues pour remplir la liste déroulante de la table.

Propriétés du noeud Re-trier



Le noeud Re-trier définit l'ordre naturel utilisé pour afficher les champs situés en aval. Cet ordre a une incidence sur l'affichage des champs en différents endroits : tableaux, listes et sélecteur de champs. Cette opération est utile lorsque vous utilisez des jeux de données volumineux pour rendre plus visibles les champs intéressants.

Tableau 66. Propriétés du noeud Re-trier.

Propriétés du noeud Re-trier	Le type de données	Description de la propriété
mode	Custom Auto	Vous pouvez trier les valeurs automatiquement ou indiquer un ordre personnalisé.
sort_by	Name Type Storage	
ascending	booléen	
start_fields	[champ1 champ2 ... champn]	Les nouveaux champs sont insérés après ces champs.
end_fields	[champ1 champ2 ... champn]	Les nouveaux champs sont insérés avant ces champs.

Propriétés du noeud Restructurer



Le noeud Restructurer convertit un champ nominal ou un champ indicateur en un groupe de champs renseignés à partir des valeurs d'un autre champ. Par exemple, si l'on considère un champ nommé *type de paiement*, qui comporte les valeurs *crédit*, *liquide* et *débit*, trois champs sont alors créés (*crédit*, *liquide*, *débit*), chacun contenant la valeur du paiement réel effectué.

Tableau 67. Propriétés du noeud Restructurer.

Propriétés du noeud Restructurer	Le type de données	Description de la propriété
fields_from	[catégorie catégorie catégorie] all	
include_field_name	booléen	Indique si le nom de champ doit être utilisé dans le nom de champ restructuré.
value_mode	OtherFields Indicateurs	Indique le mode de spécification des valeurs pour les champs restructurés. Avec OtherFields, vous devez indiquer les champs à utiliser (voir ci-dessous). Avec Flags, les valeurs sont des indicateurs numériques.
value_fields	[champ champ champ]	Obligatoire si la propriété value_mode a la valeur OtherFields. Indique les champs à utiliser en tant que champs de valeur.

Propriétés du noeud rfmanalysis



Le noeud Analyse RFM (Récence, Effectif, Monétaire) permet de déterminer de façon quantitative les clients susceptibles d'être les meilleurs par l'étude de leur dernier achat (récence), l'effectif de leurs achats (effectif), et la somme dépensée lors de toutes les transactions (monétaire).

Tableau 68. Propriétés du noeud rfmanalysis.

Propriétés du noeud rfmanalysis	Le type de données	Description de la propriété
recency	<i>champ</i>	Indiquez le champ de récence. Il peut s'agir d'une date, d'un horodatage ou d'un simple nombre.
frequency	<i>champ</i>	Indiquez le champ de fréquence.
monetary	<i>champ</i>	Spécifiez le champ monétaire.
recency_bins	<i>entier</i>	Indiquez le nombre d'intervalles de récence à générer.
recency_weight	<i>nombre</i>	Indiquez la pondération à appliquer aux données de récence. La valeur par défaut est 100.
frequency_bins	<i>entier</i>	Indiquez le nombre d'intervalles de fréquence à générer.
frequency_weight	<i>nombre</i>	Indiquez la pondération à appliquer aux données de fréquence. La valeur par défaut est 10.
monetary_bins	<i>entier</i>	Indiquez le nombre d'intervalles monétaires à générer.
monetary_weight	<i>nombre</i>	Indiquez la pondération à appliquer aux données monétaires. La valeur par défaut est 1.
tied_values_method	Next Current	Spécifiez l'intervalle dans lequel les données de valeur ex æquo doivent être placées.
recalculate_bins	Always IfNecessary	
add_outliers	<i>booléen</i>	Disponible uniquement si recalculate_bins est défini sur IfNecessary. Si cette valeur est définie, les enregistrements situés sous l'intervalle inférieur seront ajoutés à celui-ci et les enregistrements situés au-dessus de l'intervalle supérieur seront ajoutés à ce dernier.
binned_field	Recency Frequency Monetary	
recency_thresholds	<i>valeur valeur</i>	Disponible uniquement si recalculate_bins est défini sur Always. Spécifiez les seuils supérieur et inférieur pour les intervalles de récence. Le seuil supérieur d'un intervalle est aussi le seuil inférieur de l'intervalle suivant ; par exemple, [10 30 60] définirait deux intervalles, le premier présentant des seuils supérieur et inférieur de 10 et 30, les seuils du deuxième intervalle étant de 30 et 60.

Tableau 68. Propriétés du noeud *rfanalysis* (suite).

Propriétés du noeud <i>rfanalysis</i>	Le type de données	Description de la propriété
<i>frequency_thresholds</i>	<i>valeur valeur</i>	Disponible uniquement si <i>recalculate_bins</i> est défini sur Always.
<i>monetary_thresholds</i>	<i>valeur valeur</i>	Disponible uniquement si <i>recalculate_bins</i> est défini sur Always.

Propriétés du noeud *settoflag*



Le noeud Binariser calcule plusieurs champs indicateurs en fonction des valeurs catégorielles définies pour un ou plusieurs champs nominaux.

Tableau 69. Propriétés du noeud *settoflag*.

Propriétés du noeud <i>settoflag</i>	Le type de données	Description de la propriété
<i>fields_from</i>	[<i>catégorie catégorie catégorie</i>] all	
<i>true_value</i>	<i>chaîne</i>	Indiquez la valeur true (vrai) utilisée par le noeud lors de la définition d'un indicateur. La valeur par défaut est T.
<i>false_value</i>	<i>chaîne</i>	Indique la valeur false (faux) utilisée par le noeud lors de la définition d'un indicateur. La valeur par défaut est F.
<i>use_extension</i>	<i>booléen</i>	Utilisez une extension en tant que suffixe ou préfixe pour le nouveau champ indicateur.
<i>extension</i>	<i>chaîne</i>	
<i>add_as</i>	Suffix Prefix	Indique si l'extension est ajoutée en tant que suffixe ou préfixe.
<i>aggregate</i>	<i>booléen</i>	Regroupe des enregistrements sur la base des champs-clés. Tous les champs indicateurs d'un groupe sont activés si un enregistrement est paramétré sur true (vrai).
<i>keys</i>	[<i>champ champ champ</i>]	Champs-clés.

Propriétés du noeud *statistictransform*



Le noeud Transformation exécute une sélection de commandes de syntaxe IBM SPSS Statistics en fonction des sources de données dans IBM SPSS Modeler. Ce noeud requiert une copie avec licence de IBM SPSS Statistics.

Les propriétés de ce noeud sont décrites dans «Propriétés du noeud *statistictransform*», à la page 241.

Propriétés du noeud timeintervals



Le noeud Intervalle de temps définit des intervalles et crée, si nécessaire, des libellés pour la modélisation des séries temporelles. Si les valeurs ne sont pas espacées de manière égale, ce noeud peut les remplir ou les agréger, selon les besoins, pour générer un intervalle uniforme entre les enregistrements.

Tableau 70. Propriétés du noeud timeintervals.

Propriétés du noeud timeintervals	Le type de données	Description de la propriété
interval_type	None Periods CyclicPeriods Years Quarters Mois DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
mode	Label Create	Indique si vous souhaitez étiqueter les enregistrements de manière consécutive ou créer la série sur la base d'un champ de date, d'horodatage ou d'heure spécifié.
field	<i>champ</i>	Lorsque la série est créée à partir des données, indique le champ qui détermine la date ou l'heure de chaque enregistrement.
period_start	<i>entier</i>	Indique l'intervalle de début pour les périodes ou les périodes cycliques.
cycle_start	<i>entier</i>	Cycle de début pour les périodes cycliques.
year_start	<i>entier</i>	Pour les types d'intervalle concernés, année qui inclut le premier intervalle.
quarter_start	<i>entier</i>	Pour les types d'intervalle concernés, trimestre qui inclut le premier intervalle.
month_start	Janvier Février Mars Avril Mai Juin Juillet Août Septembre Octobre Novembre Décembre	
day_start	<i>entier</i>	
hour_start	<i>entier</i>	
minute_start	<i>entier</i>	
second_start	<i>entier</i>	

Tableau 70. Propriétés du noeud `timeintervals` (suite).

Propriétés du noeud <code>timeintervals</code>	Le type de données	Description de la propriété
<code>periods_per_cycle</code>	<i>entier</i>	Pour les périodes cycliques, nombre de périodes au sein de chaque cycle.
<code>fiscal_year_begins</code>	Janvier Février Mars Avril Mai Juin Juillet Août Septembre Octobre Novembre Décembre	Pour les intervalles trimestriels, indique le mois au cours duquel l'exercice commence.
<code>week_begins_on</code>	Sunday Monday Tuesday Wednesday Thursday Friday Saturday Sunday	Pour les intervalles périodiques (jours par semaine, heures par jour, minutes par jour et secondes par jour), indique le jour au cours duquel la semaine commence.
<code>day_begins_hour</code>	<i>entier</i>	Pour les intervalles périodiques (heures par jour, minutes par jour, secondes par jour), indique l'heure au cours de laquelle le jour commence. Il est possible d'utiliser cette propriété en combinaison avec <code>day_begins_minute</code> et <code>day_begins_second</code> pour indiquer une heure exacte telle que <code>8:05:01</code> . Reportez-vous à l'exemple de syntaxe ci-dessous.
<code>day_begins_minute</code>	<i>entier</i>	Pour les intervalles périodiques (heures par jour, minutes par jour, secondes par jour), indique la minute au cours de laquelle le jour commence (par exemple, <code>5</code> dans <code>8:05</code>).
<code>day_begins_second</code>	<i>entier</i>	Pour les intervalles périodiques (heures par jour, minutes par jour, secondes par jour), indique la seconde au cours de laquelle le jour commence (par exemple, <code>17</code> dans <code>8:05:17</code>).
<code>days_per_week</code>	<i>entier</i>	Pour les intervalles périodiques (jours par semaine, heures par jour, minutes par jour et secondes par jour), indique le nombre de jours par semaine.
<code>hours_per_day</code>	<i>entier</i>	Pour les intervalles périodiques (heures par jour, minutes par jour et secondes par jour), indique le nombre d'heures par jour.

Tableau 70. Propriétés du noeud `timeintervals` (suite).

Propriétés du noeud <code>timeintervals</code>	Le type de données	Description de la propriété
<code>interval_increment</code>	1 2 3 4 5 6 10 15 20 30	Pour les minutes par jour et les secondes par jour, indique le nombre de minutes ou de secondes qui fait l'objet de l'incrément pour chaque enregistrement.
<code>field_name_extension</code>	<i>chaîne</i>	
<code>field_name_extension_as_prefix</code>	<i>booléen</i>	
<code>date_format</code>	"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.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	
<code>time_format</code>	"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"	

Tableau 70. Propriétés du noeud *timeintervals* (suite).

Propriétés du noeud <i>timeintervals</i>	Le type de données	Description de la propriété
<i>aggregate</i>	Mean Sum Mode Min Max First Last TrueIfAnyTrue	Indique la méthode d'agrégation pour un champ.
<i>pad</i>	Blanc MeanOfRecentPoints Vrai False	Indique la méthode de remplissage pour un champ.
<i>agg_mode</i>	All Specify	Indique si l'agrégation ou le remplissage de tous les champs doit avoir lieu à l'aide des fonctions par défaut (si nécessaire) ou si les champs et les fonctions à utiliser doivent être spécifiés.
<i>agg_range_default</i>	Mean Sum Mode Min Max	Indique la fonction par défaut à utiliser lors de l'agrégation de champs continus.
<i>agg_set_default</i>	Mode First Last	Indique la fonction par défaut à utiliser lors de l'agrégation de champs nominaux.
<i>agg_flag_default</i>	TrueIfAnyTrue Mode First Last	
<i>pad_range_default</i>	Blanc MeanOfRecentPoints	Indique la fonction par défaut à utiliser lors du remplissage de champs continus.
<i>pad_set_default</i>	Blanc MostRecentValue	
<i>pad_flag_default</i>	Blanc Vrai False	
<i>max_records_to_create</i>	<i>entier</i>	Indique le nombre maximal d'enregistrements à créer lors du remplissage de la série.
<i>estimation_from_beginning</i>	<i>booléen</i>	
<i>estimation_to_end</i>	<i>booléen</i>	
<i>estimation_start_offset</i>	<i>entier</i>	
<i>estimation_num_holdouts</i>	<i>entier</i>	
<i>create_future_records</i>	<i>booléen</i>	
<i>num_future_records</i>	<i>entier</i>	
<i>create_future_field</i>	<i>booléen</i>	
<i>future_field_name</i>	<i>chaîne</i>	

Propriétés du noeud Transposer



Le noeud Transposer fait passer les données des lignes vers les colonnes (et réciproquement) de sorte que les enregistrements deviennent des champs et les champs des enregistrements.

Tableau 71. Propriétés du noeud Transposer.

Propriétés du noeud Transposer	Le type de données	Description de la propriété
transposed_names	Prefix Read	Les nouveaux noms de champ peuvent être générés automatiquement à partir d'un préfixe défini ou ils peuvent être lus à partir d'un champ existant dans les données.
prefix	<i>chaîne</i>	
num_new_fields	<i>entier</i>	Lorsqu'un préfixe est utilisé, indique le nombre maximal de champs à créer.
read_from_field	<i>champ</i>	Champ à partir duquel les noms sont lus. Il doit s'agir d'un champ instancié ; sinon, une erreur apparaît lorsque le noeud est exécuté.
max_num_fields	<i>entier</i>	Lorsque des noms sont lus à partir d'un champ, indique une limite supérieure afin d'éviter de créer un nombre de champs trop important.
transpose_type	Numérique Chaîne Custom	Par défaut, seuls les champs continus (intervalle numérique) sont transposés, mais vous pouvez choisir un sous-ensemble personnalisé de champs numériques ou choisir de transposer tous les champs de type chaîne.
transpose_fields	[<i>champ champ champ</i>]	Indique les champs à transposer lorsque l'option Custom est utilisée.
id_field_name	<i>champ</i>	

Propriétés du noeud type



Le noeud type définit les propriétés et métadonnées de champ. Par exemple, vous pouvez indiquer un niveau de mesure (continu, nominal, ordinal ou indicateur) pour chaque champ, définir des options pour la gestion des valeurs manquantes et des valeurs système nulles, spécifier le rôle d'un champ en vue de la modélisation, définir des libellés de champ et de valeur, et indiquer les valeurs d'un champ.

Dans certains cas, le noeud type doit être complètement instancié de façon à ce que d'autres noeuds fonctionnent correctement, notamment la propriété `fields from` du noeud `Binariser`. Vous pouvez simplement connecter un noeud `Table` et l'exécuter pour instancier les champs.

Tableau 72. Propriétés du noeud type.

Propriétés du noeud type	Le type de données	Description de la propriété
direction	Input Target Both None Partition Split Frequency RecordID	Propriété saisie (rôles des champs). <i>Remarque</i> : les valeurs In et Out sont désormais obsolètes. Leur prise en charge pourrait être supprimée dans une version ultérieure.
type	Range Flag Set Sans type Discrete OrderedSet Default	Niveau de mesure du champ (auparavant appelé "type" du champ). Si vous paramétrez le type sur Default, toute définition de paramètre de values est effacée. Si la valeur value_mode est paramétrée sur Specify, elle est redéfinie sur Read. Si la propriété value_mode est paramétrée sur Pass ou Read, la définition du type n'a aucune incidence sur value_mode. <i>Remarque</i> : Les types de données utilisés en interne diffèrent de ceux visibles dans le noeud de type. La correspondance est comme suit : Range -> Continuous Set - > Nominal OrderedSet -> Ordinal Discrete- > Categorical
storage	Inconnu String Integer Real Time Date Timestamp	Propriété saisie en lecture seule pour le type de stockage de champ.
check	None Nullify Coerce Discard Warn Abort	Propriété saisie (vérification du type et de l'intervalle des champs).
values	[<i>valeur valeur</i>]	Pour les champs continus , la première valeur représente la valeur minimale et la dernière valeur, la valeur maximale. Pour des champs nominaux, spécifiez toutes les valeurs. Pour les champs indicateurs, la première valeur représente <i>false</i> (faux) et la dernière valeur, <i>true</i> (vrai). La définition automatique de cette propriété paramètre la propriété value_mode sur Specify.

Tableau 72. Propriétés du noeud type (suite).

Propriétés du noeud type	Le type de données	Description de la propriété
value_mode	Read Pass Read+ Current Specify	Détermine le mode de définition des valeurs. Remarque : vous ne pouvez pas paramétrer directement cette propriété sur Spécifier. Pour utiliser des valeurs spécifiques, paramétrez la propriété values.
extend_values	booléen	S'applique lorsque la propriété value_mode est paramétrée sur Read. Paramétrez cette valeur sur T pour ajouter des valeurs qui viennent d'être lues aux valeurs existantes du champ. Paramétrez cette valeur sur F pour supprimer des valeurs existantes en faveur des valeurs qui viennent d'être lues.
enable_missing	booléen	Lorsque cette propriété est paramétrée sur T, elle active le suivi des valeurs manquantes du champ.
missing_values	[valeur valeur ...]	Spécifie les valeurs de données qui indiquent les données manquantes.
range_missing	booléen	Indique si un intervalle de valeurs manquantes (non renseignées) a été défini pour un champ.
missing_lower	chaîne	Lorsque le paramètre range_missing a pour valeur true (vrai), indique la limite inférieure de l'intervalle des valeurs manquantes.
missing_upper	chaîne	Lorsque le paramètre range_missing a pour valeur true, indique la limite supérieure de l'intervalle des valeurs manquantes.
null_missing	booléen	Lorsque cette propriété est paramétrée sur T, les valeurs nulles (valeurs non définies affichées sous la forme \$null\$ dans le logiciel) sont considérées comme des valeurs manquantes.
whitespace_missing	booléen	Lorsque cette propriété est paramétrée sur T, les valeurs composées uniquement de blancs (espaces, tabulations et caractères de nouvelle ligne) sont considérées comme des valeurs manquantes.
description	chaîne	Définit la description d'un champ.
value_labels	[[Valeur ChaîneLibellé] { Valeur ChaîneLibellé} ...]	Permet d'attribuer des libellés aux paires de valeurs.
display_places	entier	Définit le nombre de décimales du champ lorsqu'il est affiché (s'applique uniquement aux champs dont le stockage est REAL (REEL)). La valeur -1 utilise le flux par défaut.
export_places	entier	Définit le nombre de décimales du champ lorsqu'il est exporté (s'applique uniquement aux champs dont le stockage est REAL (REEL)). La valeur -1 utilise le flux par défaut.
decimal_separator	DEFAULT PERIOD COMMA	Définit le séparateur décimal du champ (s'applique uniquement aux champs dont le stockage est REAL (REEL)).

Tableau 72. Propriétés du noeud type (suite).

Propriétés du noeud type	Le type de données	Description de la propriété
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.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	Définit le format de date du champ (s'applique uniquement aux champs dont le stockage est DATE ou 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"	Définit le format d'heure du champ (s'applique uniquement aux champs dont le stockage est TIME ou TIMESTAMP).
number_format	DEFAULT Standard SCIENTIFIC CURRENCY	Définit le format d'affichage des nombres du champ.
standard_places	<i>entier</i>	Définit le nombre de décimales du champ lorsqu'il est affiché au format standard. La valeur -1 utilise le flux par défaut. La propriété <i>display_places</i> existante modifie également ce nombre, mais a été remplacée.
scientific_places	<i>entier</i>	Définit le nombre de décimales du champ lorsqu'il est affiché au format scientifique. La valeur -1 utilise le flux par défaut.
currency_places	<i>entier</i>	Définit le nombre de décimales du champ lorsqu'il est affiché au format monétaire. La valeur -1 utilise le flux par défaut.

Tableau 72. Propriétés du noeud type (suite).

Propriétés du noeud type	Le type de données	Description de la propriété
grouping_symbol	DEFAULT NONE LOCALE PERIOD COMMA SPACE	Définit le symbole de regroupement du champ.
column_width	<i>entier</i>	Définit la largeur des colonnes du champ. La valeur -1 définit la largeur des colonnes sur Auto.
justify	AUTO CENTER LEFT RIGHT	Définit la justification des colonnes du champ.

Chapitre 12. Propriétés des noeuds Graphiques

Propriétés communes aux noeuds Graphiques

Cette section décrit les propriétés disponibles pour les noeuds Graphiques, notamment les propriétés communes, ainsi que celles propres à chaque type de noeud.

Tableau 73. Propriétés communes aux noeuds Graphiques.

Propriétés communes aux noeuds Graphiques	Le type de données	Description de la propriété
title	chaîne	Spécifie le titre. Exemple : « Ceci est un titre ».
caption	chaîne	Spécifie la légende. Exemple : « Ceci est une légende ».
output_mode	Screen File	Indique si la sortie d'un noeud Graphique doit être affichée ou écrite dans un fichier.
output_format	BMP JPEG PNG HTML output (.cou)	Indique le type de sortie. Le type exact de sortie autorisée varie pour chaque noeud.
full_filename	chaîne	Indique le chemin et le nom de fichier cible de la sortie générée à partir du noeud Graphique.
use_graph_size	booléen	Vérifie si le graphique est dimensionné de manière explicite, à l'aide des propriétés de largeur et de hauteur définies ci-dessous. Ne concerne que les graphiques affichés à l'écran. Non disponible pour le noeud distribution.
graph_width	nombre	Lorsque use_graph_size prend la valeur True (vrai), définit la largeur du graphique en pixels.
graph_height	nombre	Lorsque use_graph_size prend la valeur True (vrai), définit la hauteur du graphique en pixels.

Remarques

Désactivation des champs facultatifs. Vous pouvez désactiver les champs facultatifs (par exemple, un champ de superposition pour les tracés) en paramétrant la valeur de la propriété sur " " (chaîne vide).

Indication des couleurs. Vous pouvez indiquer les couleurs des titres, légendes, arrière-plans et libellés à l'aide des chaînes hexadécimales commençant par le symbole dièse (#).

Les deux premiers caractères indiquent le contenu rouge, les deux caractères du milieu indiquent le contenu vert, et les deux derniers caractères indiquent le contenu bleu. Chaque caractère peut correspondre à une valeur comprise entre 0-9 ou A-F. Si elles sont associées, ces valeurs peuvent indiquer une couleur RVB (rouge, vert, bleu).

Remarque : lorsque vous spécifiez des couleurs en RVB, vous pouvez utiliser le sélecteur de champs de l'interface utilisateur pour déterminer le code couleur correct. Déplacez la souris au-dessus de la couleur pour activer l'info-bulle contenant les informations souhaitées.

Propriétés du noeud Résumé



Le noeud Résumé fournit la proportion de valeurs d'un champ numérique par rapport aux valeurs d'un autre champ. (Il génère des graphiques semblables aux histogrammes.) Il est utile pour illustrer une variable ou un champ dont les valeurs changent avec le temps. Grâce à la représentation graphique en 3D, vous pouvez en outre inclure un axe symbolique affichant les proportions par catégorie.

Tableau 74. Propriétés du noeud Résumé.

Propriétés du noeud Résumé	Le type de données	Description de la propriété
over_field	champ	
over_label_auto	booléen	
over_label	chaîne	
collect_field	champ	
collect_label_auto	booléen	
collect_label	chaîne	
three_D	booléen	
by_field	champ	
by_label_auto	booléen	
by_label	chaîne	
operation	Sum Mean Min Max SDev	
color_field	chaîne	
panel_field	chaîne	
animation_field	chaîne	
range_mode	Automatic UserDefined	
range_min	nombre	
range_max	nombre	
bins	ByNumber ByWidth	
num_bins	nombre	
bin_width	nombre	
use_grid	booléen	
graph_background	couleur	Les couleurs de graphique standard sont décrites au début de cette section.
page_background	couleur	Les couleurs de graphique standard sont décrites au début de cette section.

Propriétés du noeud distribution



Le noeud distribution fournit l'occurrence des valeurs symboliques (catégorielles), comme un type de prêt hypothécaire ou le sexe d'un individu. Ce noeud est souvent utilisé pour montrer les déséquilibres des données, déséquilibres que vous pouvez rectifier à l'aide d'un noeud Equilibrer avant la création d'un modèle.

Tableau 75. Propriétés du noeud distribution.

Propriétés du noeud distribution	Le type de données	Description de la propriété
plot	SelectedFields Indicateurs	
x_field	champ	
color_field	champ	Champ de superposition.
normalize	booléen	
sort_mode	ByOccurrence Alphabetic	
use_proportional_scale	booléen	

Propriétés du noeud Evaluation



Le noeud Evaluation permet d'évaluer et de comparer des modèles prédictifs. Le graphique d'évaluation montre l'aptitude des modèles à prédire des résultats spécifiques. Il trie les enregistrements en fonction de la valeur prédite et de la confiance dans cette prévision. Il scinde les enregistrements en groupes de taille égale (**quantiles**), puis reporte la valeur du critère traité pour chaque quantile, du plus élevé au plus faible. Les divers modèles apparaissent sous forme de lignes dans le graphique.

Tableau 76. Propriétés du noeud Evaluation.

Propriétés du noeud Evaluation	Le type de données	Description de la propriété
chart_type	Gains Réponse Lift Profit ROI ROC	
inc_baseline	booléen	
field_detection_method	Metadata Name	
use_fixed_cost	booléen	
cost_value	nombre	
cost_field	chaîne	
use_fixed_revenue	booléen	
revenue_value	nombre	
revenue_field	chaîne	
use_fixed_weight	booléen	
weight_value	nombre	
weight_field	champ	

Tableau 76. Propriétés du noeud Evaluation (suite).

Propriétés du noeud Evaluation	Le type de données	Description de la propriété
n_tile	Quartiles Quintiles Deciles Vingtiles Percentiles 1000-tiles	
cumulative	<i>indicateur</i>	
style	Line Point	
point_type	Rectangle Point Triangle Hexagone Plus Pentagone Etoile NoeudPapillon TraitHorizontal TraitVertical CroixDeFer Fabrique Maison Cathédrale Bulbe TriangleConcave GlobeAplati OeilDeChat OreillerAQuatreCôtés RectangleArrondi Eventail	
export_data	<i>booléen</i>	
data_filename	<i>chaîne</i>	
delimiter	<i>chaîne</i>	
new_line	<i>booléen</i>	
inc_field_names	<i>booléen</i>	
inc_best_line	<i>booléen</i>	
inc_business_rule	<i>booléen</i>	
business_rule_condition	<i>chaîne</i>	
plot_score_fields	<i>booléen</i>	
score_fields	<i>[champ1 ... champN]</i>	
target_field	<i>champ</i>	
use_hit_condition	<i>booléen</i>	
hit_condition	<i>chaîne</i>	
use_score_expression	<i>booléen</i>	
score_expression	<i>chaîne</i>	
caption_auto	<i>booléen</i>	

Propriétés du noeud Représentation Graphique



Le noeud Représentation Graphique offre de nombreux types de graphiques différents dans un seul noeud. Ce noeud permet de choisir les champs de données que vous souhaitez explorer puis de sélectionner un graphique parmi ceux disponibles pour les données sélectionnées. Le noeud filtre automatiquement tous les types de graphiques ne fonctionnant pas avec les sélections de champs.

Remarque : si vous définissez une propriété non valide pour le type de graphique (par exemple, si vous spécifiez `y_field` pour un histogramme), cette propriété est ignorée.

Tableau 77. Propriétés du noeud Représentation Graphique.

Propriétés du noeud Représentation Graphique	Le type de données	Description de la propriété
graph_type	2DDotplot 3DArea 3DBar 3DDensity 3DHistogram 3DPie 3DScatterplot Aire ArrowMap Barre BarCounts BarCountsMap BarMap BinnedScatter Boxplot Bubble ChoroplethMeans ChoroplethMedians ChoroplethSums ChoroplethValues ChoroplethCounts CoordinateMap CoordinateChoroplethMeans CoordinateChoroplethMedians CoordinateChoroplethSums CoordinateChoroplethValues CoordinateChoroplethCounts Dotplot Heatmap HexBinScatter Histogram Line LineChartMap LineOverlayMap Parallel Path Graphique circulaire PieCountMap PieCounts PieMap PointOverlayMap PolygonOverlayMap Ribbon Graphique en nuage de points SPLOM Surface	Identifie le type de graphique.
x_field	<i>champ</i>	Spécifie un libellé personnalisé pour l'axe <i>x</i> . Disponible uniquement pour les libellés.
y_field	<i>champ</i>	Spécifie un libellé personnalisé pour l'axe <i>y</i> . Disponible uniquement pour les libellés.
z_field	<i>champ</i>	Utilisé dans certains graphiques en 3D

Tableau 77. Propriétés du noeud Représentation Graphique (suite).

Propriétés du noeud Représentation Graphique	Le type de données	Description de la propriété
color_field	champ	Utilisé dans les cartes de zones de chaleurs.
size_field	champ	Utilisé dans les graphiques à bulles.
categories_field	champ	
values_field	champ	
rows_field	champ	
columns_field	champ	
fields	champ	
start_longitude_field	champ	Utilisé avec les flèches sur une carte de référence.
end_longitude_field	champ	
start_latitude_field	champ	
end_latitude_field	champ	
data_key_field	champ	Utilisé dans diverses cartes.
panelrow_field	chaîne	
panelcol_field	chaîne	
animation_field	chaîne	
longitude_field	champ	Utilisé avec les coordonnées sur les cartes.
latitude_field	champ	
map_color_field	champ	

Propriétés du noeud Histogramme



Le noeud Histogramme montre l'occurrence des valeurs des champs numériques. Il est souvent utilisé pour explorer les données avant toute création de modèles ou manipulation. Semblable au noeud distribution, le noeud Histogramme sert souvent à montrer les déséquilibres des données.

Tableau 78. Propriétés du noeud Histogramme.

Propriétés du noeud Histogramme	Le type de données	Description de la propriété
field	champ	
color_field	champ	
panel_field	champ	
animation_field	champ	
range_mode	Automatic UserDefined	
range_min	nombre	
range_max	nombre	
bins	ByNumber ByWidth	
num_bins	nombre	

Tableau 78. Propriétés du noeud Histogramme (suite).

Propriétés du noeud Histogramme	Le type de données	Description de la propriété
bin_width	nombre	
normalize	booléen	
separate_bands	booléen	
x_label_auto	booléen	
x_label	chaîne	
y_label_auto	booléen	
y_label	chaîne	
use_grid	booléen	
graph_background	couleur	Les couleurs de graphique standard sont décrites au début de cette section.
page_background	couleur	Les couleurs de graphique standard sont décrites au début de cette section.
normal_curve	booléen	Indique si la courbe de proportion normale doit apparaître sur la sortie.

Propriétés du noeud Courbes



Le noeud Courbes génère un graphique qui affiche plusieurs champs Y pour un seul champ X. Les champs Y sont représentés par des lignes colorées. Chacun équivaut à un noeud Tracé dont le style est défini sur **Ligne** et le mode X sur **Trier**. Les graphiques Courbes sont utiles lorsque vous souhaitez étudier la fluctuation de plusieurs variables au fil du temps.

Tableau 79. Propriétés du noeud Courbes.

Propriétés du noeud Courbes	Le type de données	Description de la propriété
x_field	champ	
y_fields	[champ champ champ]	
panel_field	champ	
animation_field	champ	
normalize	booléen	
use_overlay_expr	booléen	
overlay_expression	chaîne	
records_limit	nombre	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	booléen	
x_label	chaîne	
y_label_auto	booléen	
y_label	chaîne	
use_grid	booléen	
graph_background	couleur	Les couleurs de graphique standard sont décrites au début de cette section.

Tableau 79. Propriétés du noeud Courbes (suite).

Propriétés du noeud Courbes	Le type de données	Description de la propriété
page_background	couleur	Les couleurs de graphique standard sont décrites au début de cette section.

Propriétés du noeud Tracé



Le noeud Tracé montre les relations existant entre les champs numériques. Vous pouvez créer un tracé à l'aide de points (nuage de points) ou de courbes.

Tableau 80. Propriétés du noeud Tracé.

Propriétés du noeud Tracé	Le type de données	Description de la propriété
x_field	champ	Spécifie un libellé personnalisé pour l'axe x. Disponible uniquement pour les libellés.
y_field	champ	Spécifie un libellé personnalisé pour l'axe y. Disponible uniquement pour les libellés.
three_D	booléen	Spécifie un libellé personnalisé pour l'axe y. Disponible uniquement pour les libellés des graphiques en 3D.
z_field	champ	
color_field	champ	Champ de superposition.
size_field	champ	
shape_field	champ	
panel_field	champ	Sélectionne un champ nominal ou un champ indicateur pour créer un graphique distinct pour chaque catégorie. Les graphiques apparaissent tous sous forme de panneaux dans une fenêtre de sortie.
animation_field	champ	Spécifie un champ nominal ou un champ indicateur pour mettre en évidence les catégories des valeurs de données en créant une série de graphiques animés apparaissant les uns après les autres.
transp_field	champ	Désigne un champ destiné à mettre en évidence les catégories des valeurs de données en utilisant un degré de transparence différent pour chaque catégorie. Non disponible pour les graphiques de répartition.
overlay_type	None Smoother Function	Indique si une fonction de superposition ou un lissage LOESS est affiché.
overlay_expression	chaîne	Définit l'expression utilisée lorsque la valeur Function est affectée à overlay_type.
style	Point Line	

Tableau 80. Propriétés du noeud Tracé (suite).

Propriétés du noeud Tracé	Le type de données	Description de la propriété
point_type	Rectangle Point Triangle Hexagone Plus Pentagone Etoile NoeudPapillon TraitHorizontal TraitVertical CroixDeFer Fabrique Maison Cathédrale Bulbe TriangleConcave GlobeAplati OeilDeChat OreillerAQuatreCôtés RectangleArrondi Eventail	
x_mode	Sort Overlay AsRead	
x_range_mode	Automatique UserDefined	
x_range_min	<i>nombre</i>	
x_range_max	<i>nombre</i>	
y_range_mode	Automatique UserDefined	
y_range_min	<i>nombre</i>	
y_range_max	<i>nombre</i>	
z_range_mode	Automatique UserDefined	
z_range_min	<i>nombre</i>	
z_range_max	<i>nombre</i>	
jitter	<i>booléen</i>	
records_limit	<i>nombre</i>	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	<i>booléen</i>	
x_label	<i>chaîne</i>	
y_label_auto	<i>booléen</i>	
y_label	<i>chaîne</i>	
z_label_auto	<i>booléen</i>	
z_label	<i>chaîne</i>	
use_grid	<i>booléen</i>	
graph_background	<i>couleur</i>	Les couleurs de graphique standard sont décrites au début de cette section.

Tableau 80. Propriétés du noeud Tracé (suite).

Propriétés du noeud Tracé	Le type de données	Description de la propriété
page_background	couleur	Les couleurs de graphique standard sont décrites au début de cette section.
use_overlay_expr	booléen	Désormais obsolète et remplacé par overlay_type.

Propriétés du noeud timeplot



Le noeud Tracé horaire affiche un ou plusieurs jeux de données temporelles. En règle générale, vous utilisez un noeud Intervalles de temps, en premier lieu, pour créer un champ *TimeLabel* qui servira de libellé à l'axe *x*.

Tableau 81. Propriétés du noeud timeplot.

Propriétés du noeud timeplot	Le type de données	Description de la propriété
plot_series	Série Models	
use_custom_x_field	booléen	
x_field	champ	
y_fields	[champ champ champ]	
panel	booléen	
normalize	booléen	
line	booléen	
points	booléen	
point_type	Rectangle Point Triangle Hexagone Plus Pentagone Etoile NoeudPapillon TraitHorizontal TraitVertical CroixDeFer Fabrique Maison Cathédrale Bulbe TriangleConcave GlobeAplati OeilDeChat OreillerAQuatreCôtés RectangleArrondi Eventail	
smoother	booléen	Pour pouvoir ajouter des fonctions de lissage au graphique, paramétrez panel sur True (vrai).
use_records_limit	booléen	
records_limit	entier	
symbol_size	nombre	Définit une taille de symbole.

Tableau 81. Propriétés du noeud timeplot (suite).

Propriétés du noeud timeplot	Le type de données	Description de la propriété
panel_layout	Horizontal Vertical	

Propriétés du noeud Relations



Le noeud Relations illustre la force de la relation existant entre les valeurs de plusieurs champs symboliques (catégoriels). Le graphique utilise des lignes d'épaisseur différente pour représenter les forces de connexion. Par exemple, vous pouvez utiliser un noeud Relations pour explorer la relation avec l'achat d'un ensemble d'articles sur un site de commerce électronique.

Tableau 82. Propriétés du noeud Relations.

Propriétés du noeud Relations	Le type de données	Description de la propriété
use_directed_web	booléen	
fields	[champ champ champ]	
to_field	champ	
from_fields	[champ champ champ]	
true_flags_only	booléen	
line_values	Absolute OverallPct PctLarger PctSmaller	
strong_links_heavier	booléen	
num_links	ShowMaximum ShowLinksAbove ShowAll	
max_num_links	nombre	
links_above	nombre	
discard_links_min	booléen	
links_min_records	nombre	
discard_links_max	booléen	
links_max_records	nombre	
weak_below	nombre	
strong_above	nombre	
link_size_continuous	booléen	
web_display	Circular Réseau Directed Grid	
graph_background	couleur	Les couleurs de graphique standard sont décrites au début de cette section.
symbol_size	nombre	Définit une taille de symbole.

Chapitre 13. Propriétés des noeuds de modélisation

Propriétés communes des noeuds de modélisation

Les propriétés suivantes sont communes à certains ou à tous les noeuds de modélisation. Les éventuelles exceptions sont notées dans la documentation des noeuds de modélisation individuels, le cas échéant.

Tableau 83. Propriétés communes des noeuds de modélisation.

Propriété	Valeurs	Description de la propriété
custom_fields	booléen	Si la valeur est définie sur true (vrai), elle permet de spécifier les champs cible, entrée et d'autres champs pour le noeud actuel. Si elle est définie sur false (faux), les paramètres actuels provenant d'un noeud typeen amont sont utilisés.
target ou targets	champ ou [champ1 ... champN]	Spécifie un ou plusieurs champs cibles selon le type de modèle.
inputs	[champ1 ... champN]	Champs d'entrée ou prédicteur utilisés par le modèle.
partition	champ	
use_partitioned_data	booléen	Si un champ de partition est défini, cette option assure que seules les données provenant de la partition d'apprentissage sont utilisées pour générer le modèle.
use_split_data	booléen	
splits	[champ1 ... champN]	Indique le champ ou les champs à utiliser pour la modélisation découpée. Activé uniquement si use_split_data est défini sur True.
use_frequency	booléen	Les champs de pondération et de fréquence sont utilisés par des modèles spécifiques comme indiqué pour chaque type de modèle.
frequency_field	champ	
use_weight	booléen	
weight_field	champ	
use_model_name	booléen	
model_name	chaîne	Nom personnalisé du nouveau modèle.
mode	Simple Expert	

Propriétés du noeud anomalydetection



Le noeud Détection des anomalies identifie les observations inhabituelles, ou valeurs éloignées, qui ne se conforment pas aux motifs de données "normales". Il vous permet d'identifier les valeurs éloignées même si celles-ci ne correspondent pas aux motifs connus précédemment et même si vous ne savez pas exactement ce que vous recherchez.

Tableau 84. Propriétés du noeud anomalydetection.

Propriétés du noeud anomalydetection	Valeurs	Description de la propriété
inputs	[champ1 ... champN]	Les modèles Détection des anomalies filtrent les enregistrements en fonction des champs d'entrée spécifiés. Ils n'utilisent pas de champ cible. Les champs de pondération et de fréquence également ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
mode	Expert Simple	
anomaly_method	IndexLevel PerRecords NumRecords	Indique la méthode utilisée pour déterminer la valeur de césure de signalement des enregistrements jugés comme irréguliers.
index_level	nombre	Indique la valeur de césure limite pour le signalement des anomalies.
percent_records	nombre	Définit le seuil de signalement des enregistrements, basé sur le pourcentage d'enregistrements dans les données d'apprentissage.
num_records	nombre	Définit le seuil de signalement des enregistrements, basé sur le nombre d'enregistrements dans les données d'apprentissage.
num_fields	entier	Nombre de champs à signaler pour chaque enregistrement irrégulier.
impute_missing_values	booléen	
adjustment_coeff	nombre	Valeur utilisée pour équilibrer la pondération relative attribuée aux champs continus et aux champs catégoriels pour les calculs de distance.
peer_group_num_auto	booléen	Calcule automatiquement le nombre de groupes d'homologues.
min_num_peer_groups	entier	Indique le nombre minimal de groupes d'homologues utilisés quand la propriété peer_group_num_auto est définie sur True (vrai).
max_num_per_groups	entier	Indique le nombre maximal de groupes d'homologues.
num_peer_groups	entier	Indique le nombre de groupes d'homologues utilisés quand la propriété peer_group_num_auto est définie sur False (faux).
noise_level	nombre	Détermine le mode de traitement des valeurs éloignées au cours de la classification non supervisée. Indiquez une valeur comprise entre 0 et 0,5.

Tableau 84. Propriétés du noeud anomalydetection (suite).

Propriétés du noeud anomalydetection	Valeurs	Description de la propriété
noise_ratio	nombre	Indique la quantité de mémoire allouée au composant qui doit être utilisée pour la mise en mémoire tampon du bruit. Indiquez une valeur comprise entre 0 et 0,5.

Propriétés du noeud apriori



Le noeud Apriori extrait des données un ensemble de règles et retient les règles contenant la plus grande quantité d'informations. Le noeud Apriori fournit cinq méthodes de sélection de règles et utilise un modèle d'indexation sophistiqué pour traiter efficacement les volumes de données importants. Pour les problèmes importants, l'apprentissage du noeud Apriori est généralement plus rapide ; il n'existe aucune limite quant au nombre de règles pouvant être conservées et il peut prendre en charge des règles faisant l'objet de 32 pré-conditions. Le noeud Apriori exige que les champs d'entrée et de sortie soient tous catégoriels, mais fournit de meilleures performances car il est optimisé de ce type de données.

Tableau 85. Propriétés du noeud apriori.

Propriétés du noeud apriori	Valeurs	Description de la propriété
consequents	champ	Les modèles Apriori utilisent les Conséquences et les Antécédents à la place des champs cible et entrée standard. Les champs de pondération et de fréquence ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
antecedents	[champ1 ... champN]	
min_supp	nombre	
min_conf	nombre	
max_antecedents	nombre	
true_flags	booléen	
optimize	Vitesse Memory	
use_transactional_data	booléen	
contiguous	booléen	
id_field	chaîne	
content_field	chaîne	
mode	Simple Expert	
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	
lower_bound	nombre	
optimize	Vitesse Memory	Permet d'indiquer si la création du modèle doit être optimisée en vitesse ou en mémoire.

Propriétés du noeud autoclassifier



Le noeud Discriminant automatique crée et compare les résultats binaires de plusieurs modèles différents (oui ou non, avec ou sans attrition, etc.), ce qui vous permet de choisir la meilleure approche pour une analyse donnée. Plusieurs algorithmes de modélisation sont pris en charge. Vous pouvez alors sélectionner les méthodes que vous souhaitez utiliser, les options spécifiques pour chacune d'elles et le critère de comparaison des résultats. Le noeud génère un ensemble de modèles basé sur les options spécifiées et classe les meilleurs candidats en fonction des critères indiqués.

Tableau 86. Propriétés du noeud autoclassifier.

Propriétés du noeud autoclassifier	Valeurs	Description de la propriété
target	champ	Pour les cibles indicateur, le noeud Discriminant automatique requiert un seul champ cible et un ou plusieurs champs d'entrée. Il est également possible de spécifier des champs de pondération et de fréquence. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
ranking_measure	Accuracy Area_under_curve Profit Lift Num_variables	
ranking_dataset	Training Test	
number_of_models	entier	Nombre de modèles à inclure dans le nugget de modèle. Indiquez un entier compris entre 1 et 100.
calculate_variable_importance	booléen	
enable_accuracy_limit	booléen	
accuracy_limit	entier	Entier compris entre 0 et 100.
enable_area_under_curve_limit	booléen	
area_under_curve_limit	nombre	Nombre réel compris entre 0,0 et 1,0.
enable_profit_limit	booléen	
profit_limit	nombre	Entier supérieur à 0.
enable_lift_limit	booléen	
lift_limit	nombre	Nombre réel supérieur à 1.0.
enable_number_of_variables_limit	booléen	
number_of_variables_limit	nombre	Entier supérieur à 0.
use_fixed_cost	booléen	
fixed_cost	nombre	Nombre réel supérieur à 0.0.
variable_cost	champ	
use_fixed_revenue	booléen	
fixed_revenue	nombre	Nombre réel supérieur à 0.0.
variable_revenue	champ	
use_fixed_weight	booléen	

Tableau 86. Propriétés du noeud autoclassifier (suite).

Propriétés du noeud autoclassifier	Valeurs	Description de la propriété
fixed_weight	nombre	Nombre réel supérieur à 0,0.
variable_weight	champ	
lift_percentile	nombre	Entier compris entre 0 et 100.
enable_model_build_time_limit	booléen	
model_build_time_limit	nombre	Entier défini sur le nombre de minutes devant être utilisé pour limiter le temps nécessaire à la création de chaque modèle.
enable_stop_after_time_limit	booléen	
stop_after_time_limit	nombre	Nombre réel défini sur le nombre d'heures servant à limiter le temps global passé pour une exécution de Discriminant automatique.
enable_stop_after_valid_model_produced	booléen	
use_costs	booléen	
<algorithm>	booléen	Active ou désactive l'utilisation d'un algorithme particulier.
<algorithm>.<property>	chaîne	Définit une valeur de propriété pour un algorithme spécifique. Pour plus d'informations, voir la rubrique «Définition des propriétés de l'algorithme».

Définition des propriétés de l'algorithme

Les noms d'algorithme pour le noeud Discriminant automatique sont cart, chaid, quest, c50, logreg, decisionlist, bayesnet, discriminant, svm et knn.

Les noms d'algorithme pour le noeud Numérisation automatique sont cart, chaid, neuralnetwork, genlin, svm, regression, linear et knn.

Les noms d'algorithme pour le noeud Cluster automatique sont twostep, k moyenne et kohonen.

Les noms de propriété sont standard (reportez-vous aux informations relatives à chaque noeud d'algorithme).

Les propriétés d'algorithme qui contiennent des points ou d'autres signes de ponctuation doivent être placées entre guillemets simples (' ').

Plusieurs valeurs peuvent également être affectées à une propriété.

Remarques :

- Utilisez des minuscules lors de la définition des valeurs true (vrai) et false (faux) (plutôt que False).
- Lorsque certaines options d'algorithme ne sont pas disponibles dans le noeud Discriminant automatique ou qu'une seule valeur peut être indiquée plutôt qu'un intervalle de valeurs, les mêmes limites s'appliquent à la génération de scripts que lorsque l'on accède au noeud de la manière standard.

Propriétés du noeud autocluster



Le noeud Cluster automatique évalue et compare les modèles de classification identifiant des groupes d'enregistrements ayant des caractéristiques similaires. Le noeud fonctionne de la même manière que les autres noeuds de modélisation automatiques, vous permettant de tester plusieurs combinaisons d'options en une seule modélisation. Les modèles peuvent être comparés à l'aide de mesures de bases permettant d'essayer de filtrer et de classer l'utilité des modèles de classification et de fournir une mesure en fonction de l'importance de champs particuliers.

Tableau 87. Propriétés du noeud autocluster.

Propriétés du noeud autocluster	Valeurs	Description de la propriété
evaluation	champ	Remarque : noeud de cluster automatique uniquement. Identifie le champ pour lequel une valeur d'importance sera calculée. Il peut aussi être utilisé pour identifier la façon dont un cluster différencie la valeur de ce champ et, par conséquent, la façon dont le modèle pourra prédire ce champ.
ranking_measure	Silhouette Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Importance	
ranking_dataset	Training Test	
summary_limit	entier	Nombre de modèles devant figurer dans le rapport. Indiquez un entier compris entre 1 et 100.
enable_silhouette_limit	booléen	
silhouette_limit	entier	Entier compris entre 0 et 100.
enable_number_less_limit	booléen	
number_less_limit	nombre	Nombre réel compris entre 0,0 et 1,0.
enable_number_greater_limit	booléen	
number_greater_limit	nombre	Entier supérieur à 0.
enable_smallest_cluster_limit	booléen	
smallest_cluster_units	Pourcentage Counts	
smallest_cluster_limit_percentage	nombre	
smallest_cluster_limit_count	entier	Entier supérieur à 0.
enable_largest_cluster_limit	booléen	
largest_cluster_units	Pourcentage Counts	
largest_cluster_limit_percentage	nombre	
largest_cluster_limit_count	entier	
enable_smallest_largest_limit	booléen	
smallest_largest_limit	nombre	
enable_importance_limit	booléen	

Tableau 87. Propriétés du noeud autocluster (suite).

Propriétés du noeud autocluster	Valeurs	Description de la propriété
importance_limit_condition	Greater_than Less_than	
importance_limit_greater_than	nombre	Entier compris entre 0 et 100.
importance_limit_less_than	nombre	Entier compris entre 0 et 100.
<algorithm>	booléen	Active ou désactive l'utilisation d'un algorithme particulier.
<algorithm>.<property>	chaîne	Définit une valeur de propriété pour un algorithme spécifique. Pour plus d'informations, voir la rubrique «Définition des propriétés de l'algorithme», à la page 135.

Propriétés du noeud autonumeric



Le noeud Numérisation automatique évalue et compare des modèles pour des résultats d'intervalle numérique continus par le biais de différentes méthodes. Le noeud fonctionne de la même manière que le noeud Discriminant automatique, vous permettant ainsi de choisir les algorithmes à utiliser et à tester avec différentes combinaisons d'options en un seul passage de modélisation. Les algorithmes pris en charge comprennent les réseaux de neurones, l'arborescence C&R Tree, CHAID, la régression linéaire, la régression linéaire généralisée et support vector machines (SVM). Les modèles peuvent être comparés selon la corrélation, l'erreur relative ou le nombre de variables utilisées.

Tableau 88. Propriétés du noeud autonumeric.

Propriétés du noeud autonumeric	Valeurs	Description de la propriété
custom_fields	booléen	Si cette valeur est définie sur True (vrai), les paramètres de champ personnalisés seront utilisés à la place des paramètres du noeud type.
target	champ	Le noeud Numérisation automatique requiert un seul champ cible et un ou plusieurs champs d'entrée. Il est également possible de spécifier des champs de pondération et de fréquence. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
inputs	[champ1 ... champ2]	
partition	champ	
use_frequency	booléen	
frequency_field	champ	
use_weight	booléen	
weight_field	champ	
use_partitioned_data	booléen	Si un champ de partition est défini, seules les données d'apprentissage sont utilisées pour la création du modèle.
ranking_measure	Correlation NumberOfFields	

Tableau 88. Propriétés du noeud autonumeric (suite).

Propriétés du noeud autonumeric	Valeurs	Description de la propriété
ranking_dataset	Test Training	
number_of_models	entier	Nombre de modèles à inclure dans le nugget de modèle. Indiquez un entier compris entre 1 et 100.
calculate_variable_importance	booléen	
enable_correlation_limit	booléen	
correlation_limit	entier	
enable_number_of_fields_limit	booléen	
number_of_fields_limit	entier	
enable_relative_error_limit	booléen	
relative_error_limit	entier	
enable_model_build_time_limit	booléen	
model_build_time_limit	entier	
enable_stop_after_time_limit	booléen	
stop_after_time_limit	entier	
stop_if_valid_model	booléen	
<algorithm>	booléen	Active ou désactive l'utilisation d'un algorithme particulier.
<algorithm>.<property>	chaîne	Définit une valeur de propriété pour un algorithme spécifique. Pour plus d'informations, voir la rubrique «Définition des propriétés de l'algorithme», à la page 135.

Propriétés du noeud bayesnet



Le noeud Réseau Bayésien permet de créer un modèle de probabilité en combinant les preuves observées et enregistrées avec les connaissances réelles pour établir la probabilité des occurrences. Le noeud est axé sur le Tree Augmented Naïve Bayes (TAN) et sur les réseaux Couverture de Markov qui servent principalement à la classification.

Tableau 89. Propriétés du noeud bayesnet.

Propriétés du noeud bayesnet	Valeurs	Description de la propriété
inputs	[champ1 ... champN]	Les modèles de réseau Bayésien utilisent un seul champ cible et un ou plusieurs champs d'entrée. Les champs continus sont automatiquement mis en intervalles. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
continue_training_existing_model	booléen	
structure_type	TAN MarkovBlanket	Sélectionne la structure à utiliser lors de la création du réseau Bayésien.
use_feature_selection	booléen	

Tableau 89. Propriétés du noeud bayesnet (suite).

Propriétés du noeud bayesnet	Valeurs	Description de la propriété
parameter_learning_method	Likelihood Bayes	Spécifie la méthode utilisée pour estimer les tableaux de probabilités conditionnelles entre les noeuds où les valeurs des parent sont connues.
mode	Expert Simple	
missing_values	booléen	
all_probabilities	booléen	
independence	Likelihood Pearson	Spécifie la méthode utilisée pour déterminer si des observations par paire sur deux variables sont indépendantes l'une de l'autre.
significance_level	nombre	Spécifie la valeur de césure permettant de déterminer l'indépendance.
maximal_conditioning_set	nombre	Définit le nombre maximum de variables de conditionnement à utiliser pour les essais d'indépendance.
inputs_always_selected	[champ1 ... champN]	Spécifie les champs du jeu de données qui doivent toujours être utilisés lors de la création du réseau Bayésien. <i>Remarque</i> : le champ cible est toujours sélectionné.
maximum_number_inputs	nombre	Spécifie le nombre maximum de champs d'entrée à utiliser pour créer le réseau Bayésien.
calculate_variable_importance	booléen	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés du noeud buildr



Le noeud de création R vous permet d'entrer un script R personnalisé pour procéder à la création de modèle et au scoring de modèle déployés dans IBM SPSS Modeler.

Tableau 90. Propriétés buildr.

Propriétés du noeud buildr	Valeurs	Description de la propriété
build_syntax	chaîne	Syntaxe de script R pour la création de modèle.
score_syntax	chaîne	Syntaxe de script R pour le scoring de modèle.
convert_flags	StringsAndDoubles LogicalValues	Option permettant de convertir des champs indicateurs.

Tableau 90. Propriétés buildr (suite).

Propriétés du noeud buildr	Valeurs	Description de la propriété
convert_datetime	booléen	Option permettant de convertir des variables au format de date ou date-heure en format de date/heure R.
convert_datetime_class	POSIXct POSIXlt	Options permettant d'indiquer dans quel format sont converties les variables au format de date ou date-heure.
convert_missing	booléen	Option permettant de convertir des valeurs manquantes en valeur R NA.
output_html	booléen	Option permettant d'afficher les graphiques sur un onglet du nugget de modèle R.
output_text	booléen	Option permettant d'écrire la sortie texte de console R sur un onglet du nugget de modèle R.

Propriétés du noeud c50



Le noeud C5.0 crée un arbre décision ou un ensemble de règles. Le fonctionnement de ce modèle repose sur un découpage de l'échantillon basé sur le champ qui fournit le gain d'informations le plus important à chaque niveau. Le champ cible doit être catégoriel. Les divisions multiples en plus de deux sous-groupes sont autorisées.

Tableau 91. Propriétés du noeud c50.

Propriétés du noeud c50	Valeurs	Description de la propriété
target	champ	Les modèles C50 utilisent un seul champ cible et un ou plusieurs champs d'entrée. Un champ poids peut aussi être spécifié. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
output_type	DecisionTree RuleSet	
group_symbolics	booléen	
use_boost	booléen	
boost_num_trials	nombre	
use_xval	booléen	
xval_num_folds	nombre	
mode	Simple Expert	
favor	Accuracy Generality	Privilégiez l'exactitude ou la généralité.
expected_noise	nombre	
min_child_records	nombre	
pruning_severity	nombre	
use_costs	booléen	
costs	structurée	Il s'agit d'une propriété structurée.
use_winning	booléen	

Tableau 91. Propriétés du noeud c50 (suite).

Propriétés du noeud c50	Valeurs	Description de la propriété
use_global_pruning	booléen	Activé (True(vrai)) par défaut.
calculate_variable_importance	booléen	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés du noeud Carma



Le modèle CARMA extrait un ensemble de règles des données sans que vous ayez à définir les champs d'entrée ou les champs cible. Au contraire du noeud Apriori le noeud CARMA offre des paramètres de création pour la prise en charge de la règle (à la fois pour les antécédents et les conséquences), plutôt qu'une simple prise en charge d'antécédents. Cela signifie que les règles générées peuvent être utilisées dans un grand nombre d'applications, par exemple pour rechercher une liste des produits ou des services (antécédents) dont la conséquence correspond à l'élément que vous souhaitez promouvoir à l'occasion de cette période de congés.

Tableau 92. Propriétés du noeud Carma.

Propriétés du noeud carma	Valeurs	Description de la propriété
inputs	[champ1 ... champn]	Les modèles CARMA utilisent une liste de champs d'entrée, mais pas de cible. Les champs de pondération et de fréquence ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
id_field	champ	Champ utilisé en tant que champ d'ID pour la création du modèle.
contiguous	booléen	Permet d'indiquer si les ID du champ ID sont contigus.
use_transactional_data	booléen	
content_field	champ	
min_supp	nombre (pourcentage)	Est lié à la prise en charge de la règle plutôt qu'à celle des antécédents. La valeur par défaut est 20 %.
min_conf	nombre (pourcentage)	La valeur par défaut est 20 %.
max_size	nombre	La valeur par défaut est 10.
mode	Simple Expert	La valeur par défaut est Simple.
exclude_multiple	booléen	Exclut les règles comportant plusieurs conséquences. La valeur par défaut est False.
use_pruning	booléen	La valeur par défaut est False.
pruning_value	nombre	La valeur par défaut est 500.
vary_support	booléen	
estimated_transactions	entier	

Tableau 92. Propriétés du noeud Carma (suite).

Propriétés du noeud carma	Valeurs	Description de la propriété
rules_without_antecedents	booléen	

Propriétés du noeud cart



Le noeud Arbre Classification et Régression, (C&R), génère un arbre décision qui vous permet de prévoir ou de classer les observations futures. La méthode utilise la technique de partition récursive afin de diviser les données d'apprentissage en segments en réduisant l'index d'impureté à chaque étape, un noeud de l'arbre étant considéré comme "pur" si 100 % de ses observations appartiennent à une catégorie spécifique du champ cible. Les champs cible et les champs d'entrée peuvent être des champs d'intervalle numériques ou des champs catégoriels numériques (nominal, ordinal ou indicateur). Toutes les divisions sont binaires (deux sous-groupes uniquement).

Tableau 93. Propriétés du noeud cart.

Propriétés du noeud cart	Valeurs	Description de la propriété
target	champ	Les modèles d'arbre C&R requièrent un seul champ cible et un ou plusieurs champs d'entrée. Un champ de fréquence peut aussi être spécifié. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
continue_training_existing_model	booléen	
objective	Standard Boosting Bagging psm	psm est destinée aux jeux de données très volumineux et nécessite une connexion à un serveur.
model_output_type	Single InteractiveBuilder	
use_tree_directives	booléen	
tree_directives	chaîne	Définissez les directives de développement de l'arbre. Vous pouvez placer les directives entre guillemets triples pour ne pas tenir compte des caractères d'insertion de ligne ou des guillemets doubles. Notez que les directives sont parfois très sensibles aux modifications, même mineures, apportées aux données ou aux options de modélisation. Elles peuvent ne pas s'étendre aux autres jeux de données.
use_max_depth	Default Custom	
max_depth	entier	Profondeur maximale de l'arbre, comprise entre 0 et 1000. Utilisée uniquement si use_max_depth = Custom
prune_tree	booléen	Elagage de l'arbre pour éviter le surajustement.
use_std_err	booléen	Utiliser la différence maximale pour le risque (dans Erreurs standard).
std_err_multiplier	nombre	Différence maximale.

Tableau 93. Propriétés du noeud cart (suite).

Propriétés du noeud cart	Valeurs	Description de la propriété
max_surrogates	nombre	Nombre maximal de substitutions.
use_percentage	booléen	
min_parent_records_pc	nombre	
min_child_records_pc	nombre	
min_parent_records_abs	nombre	
min_child_records_abs	nombre	
use_costs	booléen	
costs	structurée	Propriétés structurées.
priors	Data Equal Custom	
custom_priors	structurée	Propriétés structurées.
adjust_priors	booléen	
trails	nombre	Nombre des modèles de composant pour le boosting ou le bagging.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Règles de combinaison par défaut pour les cibles catégorielles.
range_ensemble_method	Mean Median	Règles de combinaison par défaut pour les cibles continues.
large_boost	booléen	Appliquer le boosting aux jeux de données très volumineux.
min_impurity	nombre	
impurity_measure	Gini Twoing Ordered	
train_pct	nombre	Ensemble de prévention de surajustement.
set_random_seed	booléen	Dupliquer l'option des résultats.
seed	nombre	
calculate_variable_importance	booléen	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés du noeud chaid



Le noeud CHAID génère des arbres décision à l'aide des statistiques du khi-deux pour identifier les séparations optimales. Contrairement aux noeuds C&R Tree et QUEST, CHAID peut générer des arbres non binaires, ce qui implique que certaines divisions possèdent plusieurs branches. Les champs cibles et les champs d'entrée peuvent être d'intervalle numérique (continu) ou catégoriques. La méthode Exhaustive CHAID correspond à une modification du CHAID qui examine plus en détail toutes les divisions possibles, mais dont les calculs sont plus longs.

Tableau 94. Propriétés du noeud chaid.

Propriétés du noeud chaid	Valeurs	Description de la propriété
target	champ	Les modèles CHAID requièrent un seul champ cible et un ou plusieurs champs d'entrée. Un champ de fréquence peut aussi être spécifié. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
continue_training_existing_model	booléen	
objective	Standard Boosting Bagging psm	psm est destinée aux jeux de données très volumineux et nécessite une connexion à un serveur.
model_output_type	Single InteractiveBuilder	
use_tree_directives	booléen	
tree_directives	chaîne	
method	Chaid ExhaustiveChaid	
use_max_depth	Default Custom	
max_depth	entier	Profondeur maximale de l'arbre, comprise entre 0 et 1000. Utilisée uniquement si use_max_depth = Custom
use_percentage	booléen	
min_parent_records_pc	nombre	
min_child_records_pc	nombre	
min_parent_records_abs	nombre	
min_child_records_abs	nombre	
use_costs	booléen	
costs	structurée	Propriétés structurées.
trails	nombre	Nombre des modèles de composant pour le boosting ou le bagging.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Règles de combinaison par défaut pour les cibles catégorielles.
range_ensemble_method	Mean Median	Règles de combinaison par défaut pour les cibles continues.
large_boost	booléen	Appliquer le boosting aux jeux de données très volumineux.
split_alpha	nombre	Niveau de signification pour la division.
merge_alpha	nombre	Niveau de signification pour la fusion.
bonferroni_adjustment	booléen	Ajuster les valeurs de signification à l'aide de la méthode Bonferroni.
split_merged_categories	booléen	Autoriser une nouvelle division des catégories fusionnées.
chi_square	Pearson LR	Méthode utilisée pour calculer les statistiques du khi-carré : Pearson ou Likelihood Ratio

Tableau 94. Propriétés du noeud chaid (suite).

Propriétés du noeud chaid	Valeurs	Description de la propriété
epsilon	<i>nombre</i>	Modification minimale dans les prévisions de fréquence de cellule..
max_iterations	<i>nombre</i>	Itérations maximales pour convergence.
set_random_seed	<i>entier</i>	
seed	<i>nombre</i>	
calculate_variable_importance	<i>booléen</i>	
calculate_raw_propensities	<i>booléen</i>	
calculate_adjusted_propensities	<i>booléen</i>	
adjusted_propensity_partition	Test Validation	
maximum_number_of_models	<i>entier</i>	

Propriétés du noeud coxreg



Le noeud de régression de Cox vous permet de créer un modèle de survie pour les données de durée jusqu'à l'événement en présence d'enregistrements censurés. Ce modèle produit une fonction de survie qui prédit la probabilité que l'événement en question se soit produit à un moment (t) pour des valeurs données des variables d'entrée.

Tableau 95. Propriétés du noeud coxreg.

Propriétés du noeud coxreg	Valeurs	Description de la propriété
survival_time	<i>champ</i>	Les modèles de régression de Cox requièrent un champ unique contenant les durées de survie.
target	<i>champ</i>	Les modèles de régression de Cox requièrent un seul champ cible et un ou plusieurs champs d'entrée. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
method	Enter Stepwise BackwardsStepwise	
groups	<i>champ</i>	
model_type	MainEffects Custom	
custom_terms	["BP*Sex" "BP*Age"]	
mode	Expert Simple	
max_iterations	<i>nombre</i>	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	

Tableau 95. Propriétés du noeud coxreg (suite).

Propriétés du noeud coxreg	Valeurs	Description de la propriété
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>nombre</i>	
probability_removal	<i>nombre</i>	
output_display	EachStep LastStep	
ci_enable	<i>booléen</i>	
ci_value	90 95 99	
Corrélation	<i>booléen</i>	
display_baseline	<i>booléen</i>	
survival	<i>booléen</i>	
hazard	<i>booléen</i>	
log_minus_log	<i>booléen</i>	
one_minus_survival	<i>booléen</i>	
separate_line	<i>champ</i>	
value	<i>nombre ou chaîne</i>	Si aucune valeur n'est spécifiée pour un champ, l'option par défaut, « Moyenne » sera utilisée pour ce champ.

Propriétés du noeud decisionlist



Le noeud Liste de décision identifie les sous-groupes, ou les segments, qui présentent une probabilité plus élevée ou plus faible d'un résultat binaire donné par rapport à la population globale. Vous pouvez, par exemple, rechercher les clients qui ont une faible probabilité d'attrition ou ceux qui ont une plus forte probabilité de répondre favorablement à une campagne. Vous pouvez incorporer vos connaissances métier dans le modèle en ajoutant vos propres segments personnalisés et en prévisualisant des modèles alternatifs côte à côte de façon à comparer les résultats. Les modèles Liste de décision se composent d'une liste de règles dans laquelle chaque règle présente une condition et un résultat. Les règles sont appliquées dans l'ordre et la première règle correspondante détermine le résultat.

Tableau 96. Propriétés du noeud decisionlist.

Propriétés du noeud decisionlist	Valeurs	Description de la propriété
target	champ	Les modèles Liste de décision utilisent un seul champ cible et un ou plusieurs champs d'entrée. Un champ de fréquence peut aussi être spécifié. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
model_output_type	Model InteractiveBuilder	
search_direction	Up Down	Est lié à la recherche de segments ; où Up est l'équivalent de Forte probabilité et Down est l'équivalent de Faible probabilité.
target_value	chaîne	Si cette propriété n'est pas définie, la valeur True (vrai) est sélectionnée pour les champs indicateurs.
max_rules	entier	Nombre maximal de segments, sauf le reste.
taille_groupe_min	entier	Taille minimale de segment.
min_group_size_pct	nombre	Taille minimale de segment, en pourcentage.
confidence_level	nombre	Seuil minimal dont dispose un champ d'entrée pour améliorer (augmenter) la probabilité d'une réponse, pour que son ajout à une définition de segment soit utile.
max_segments_per_rule	entier	
mode	Simple Expert	
bin_method	EqualWidth EqualCount	
bin_count	nombre	
max_models_per_cycle	entier	Largeur de recherche des listes.
max_rules_per_cycle	entier	Largeur de recherche des règles de segment.
segment_growth	nombre	
include_missing	booléen	
final_results_only	booléen	
reuse_fields	booléen	Permet aux attributs (champs d'entrée qui apparaissent dans les règles) d'être réutilisés.
max_alternatives	entier	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés du noeud discriminant



L'analyse discriminante crée des hypothèses plus strictes que la régression logistique mais peut constituer une alternative ou un complément précieux à une analyse de régression logistique lorsque ces hypothèses sont réunies.

Tableau 97. Propriétés du noeud discriminant.

Propriétés du noeud discriminant	Valeurs	Description de la propriété
target	champ	Les modèles Discriminant requièrent un seul champ cible et un ou plusieurs champs d'entrée. Les champs de pondération et de fréquence ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
method	Enter Stepwise	
mode	Simple Expert	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups SeparateGroups	
means	booléen	Options Statistiques dans la boîte de dialogue Sortie avancée.
univariate_anovas	booléen	
box_m	booléen	
within_group_covariance	booléen	
within_groups_correlation	booléen	
separate_groups_covariance	booléen	
total_covariance	booléen	
fishers	booléen	
unstandardized	booléen	
casewise_results	booléen	Options Classification dans la boîte de dialogue Sortie avancée.
limit_to_first	nombre	La valeur par défaut est 10.
summary_table	booléen	
leave_one_classification	booléen	
combined_groups	booléen	
separate_groups_covariance	booléen	Option Matrices, Covariance par groupes distincts
territorial_map	booléen	
combined_groups	booléen	Option Tracé Groupes regroupés.
separate_groups	booléen	Option Tracé Groupes distincts
summary_of_steps	booléen	
F_pairwise	booléen	

Tableau 97. Propriétés du noeud discriminant (suite).

Propriétés du noeud discriminant	Valeurs	Description de la propriété
stepwise_method	WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	nombre	
criteria	UseValue UseProbability	
F_value_entry	nombre	La valeur par défaut est 3,84.
F_value_removal	nombre	La valeur par défaut est 2,71.
probability_entry	nombre	La valeur par défaut est 0.05.
probability_removal	nombre	La valeur par défaut est 0.10.
calculate_variable_importance	booléen	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés du noeud ACP/Facteur



Le noeud ACP/Analyse factorielle propose des techniques de factorisation puissantes qui vous permettent de réduire la complexité de vos données. L'analyse en composantes principales (ACP) recherche les combinaisons linéaires des champs d'entrée qui permettent de capturer au mieux la variance dans l'ensemble de champs, où les composantes sont orthogonales (perpendiculaires) les unes par rapport aux autres. L'analyse factorielle a pour but d'identifier les facteurs sous-jacents qui expliquent la tendance des corrélations dans un ensemble de champs observés. Quelle que soit l'approche choisie, le but consiste à trouver un nombre limité de champs dérivés récapitulant les informations contenues dans l'ensemble de champs d'origine.

Tableau 98. Propriétés du noeud ACP/Facteur.

Propriétés du noeud ACP/Facteur	Valeurs	Description de la propriété
inputs	[champ1 ... champN]	Les modèles APC/Facteur utilisent une liste de champs d'entrée, mais pas de cible. Les champs de pondération et de fréquence ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
method	PC ULS GLS ML PAF Alpha Image	
mode	Simple Expert	

Tableau 98. Propriétés du noeud ACP/Facteur (suite).

Propriétés du noeud ACP/Facteur	Valeurs	Description de la propriété
max_iterations	nombre	
complete_records	booléen	
matrix	Correlation Covariance	
extract_factors	ByEigenvalues ByFactors	
min_eigenvalue	nombre	
max_factor	nombre	
rotation	None Varimax DirectOblimin Equamax Quartimax Promax	
delta	nombre	Si vous sélectionnez DirectOblimin comme type de données de rotation, vous pouvez indiquer une valeur de delta. Si vous ne le faites pas, la valeur par défaut pour delta est utilisée.
kappa	nombre	Si vous sélectionnez Promax comme type de données de rotation, vous pouvez indiquer une valeur de kappa. Si vous ne le faites pas, la valeur par défaut pour kappa est utilisée.
sort_values	booléen	
hide_values	booléen	
hide_below	nombre	

Propriétés du noeud featureselection



Le noeud Sélection de fonction filtre les champs d'entrée en vue de leur suppression, en fonction d'un ensemble de critères donné (tel que le pourcentage de valeurs manquantes) ; il classe ensuite les entrées restantes selon leur importance par rapport à la cible indiquée. Si l'on prend, par exemple, un jeu de données comportant des centaines d'entrées potentielles, quelles sont celles susceptibles d'être les plus utiles dans la modélisation des résultats de patients ?

Tableau 99. Propriétés du noeud featureselection.

Propriétés du noeud featureselection	Valeurs	Description de la propriété
target	champ	Les modèles Sélection de fonction classent les prédicteurs par rapport à la cible spécifiée. Les champs de pondération et de fréquence ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.

Tableau 99. Propriétés du noeud featureselection (suite).

Propriétés du noeud featureselection	Valeurs	Description de la propriété
screen_single_category	booléen	Si cette propriété est définie sur True (vrai), elle filtre les champs qui comportent, par rapport au nombre total d'enregistrements, trop d'enregistrements relatifs à une même catégorie.
max_single_category	nombre	Indique le seuil utilisé quand la propriété screen_single_category est définie sur True (vrai).
screen_missing_values	booléen	Si cette propriété est définie sur True (vrai), elle filtre les champs qui comportent un nombre trop important de valeurs manquantes, exprimé en pourcentage du nombre total d'enregistrements.
max_missing_values	nombre	
screen_num_categories	booléen	Si cette propriété est définie sur True (vrai), elle filtre les champs comportant trop de catégories par rapport au nombre total d'enregistrements.
max_num_categories	nombre	
screen_std_dev	booléen	Si cette propriété est définie sur True (vrai), elle filtre les champs dont l'écart-type est inférieur ou égal à la valeur minimale indiquée.
min_std_dev	nombre	
screen_coeff_of_var	booléen	Si cette propriété est définie sur True (vrai), elle filtre les champs dont le coefficient de la variance est inférieur ou égal à la valeur minimale spécifiée.
min_coeff_of_var	nombre	
criteria	Pearson Likelihood CramersV Lambda	Lors du classement des prédicteurs indépendants par rapport à une cible catégorielle, indique la mesure sur laquelle est basée la valeur d'importance.
unimportant_below	nombre	Indique les valeurs p du seuil, utilisées pour classer les variables comme étant importantes, marginales ou non significatives. Accepte des valeurs de 0,0 à 1,0.
important_above	nombre	Accepte des valeurs de 0,0 à 1,0.
unimportant_label	chaîne	Indique le libellé du classement non significatif.
marginal_label	chaîne	
important_label	chaîne	
selection_mode	ImportanceLevel ImportanceValue TopN	

Tableau 99. Propriétés du noeud featureselection (suite).

Propriétés du noeud featureselection	Valeurs	Description de la propriété
select_important	booléen	Quand la propriété selection_mode est définie sur ImportanceLevel, indique si les champs importants doivent être sélectionnés.
select_marginal	booléen	Quand la propriété selection_mode est définie sur ImportanceLevel, indique si les champs marginaux doivent être sélectionnés.
select_unimportant	booléen	Quand la propriété selection_mode est définie sur ImportanceLevel, indique si les champs non significatifs doivent être sélectionnés.
importance_value	nombre	Quand la propriété selection_mode est définie sur ImportanceValue, indique la valeur de césure à utiliser. Accepte des valeurs de 0 à 100.
top_n	entier	Quand la propriété selection_mode est définie sur TopN, indique la valeur de césure à utiliser. Accepte des valeurs de 0 à 1000.

Propriétés du noeud genlin



La procédure Modèles linéaires généralisés développe le modèle linéaire général de sorte que la variable dépendante soit linéairement reliée aux facteurs et covariables via une fonction de lien précise. En outre, le modèle permet à la variable dépendante de suivre une distribution non normale. Il couvre les fonctionnalités d'un grand nombre de modèles statistiques, notamment le modèle de régression linéaire, le modèle de régression logistique, le modèle log-linéaire pour les données d'effectif et le modèle de survie avec censure par intervalle.

Tableau 100. Propriétés du noeud genlin.

Propriétés du noeud genlin	Valeurs	Description de la propriété
target	champ	Les modèles linéaires généralisés requièrent un seul champ cible, qui doit être un champ nominal ou indicateur, et un ou plusieurs champs d'entrée. Un champ poids peut aussi être spécifié. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
use_weight	booléen	
weight_field	champ	Le champ peut uniquement être de type continu.
target_represents_trials	booléen	
trials_type	Variable FixedValue	
trials_field	champ	Le champ est de type Continu, Indicateur ou Ordinal.
trials_number	nombre	La valeur par défaut est 10.

Tableau 100. Propriétés du noeud `genlin` (suite).

Propriétés du noeud <code>genlin</code>	Valeurs	Description de la propriété
<code>model_type</code>	MainEffects MainAndAllTwoWayEffects	
<code>offset_type</code>	Variable FixedValue	
<code>offset_field</code>	<i>champ</i>	Le champ peut uniquement être de type continu.
<code>offset_value</code>	<i>nombre</i>	Doit être un nombre réel.
<code>base_category</code>	Last First	
<code>include_intercept</code>	<i>booléen</i>	
<code>mode</code>	Simple Expert	
Distribution	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIALE	IGAUSS: Gaussienne inverse. NEGBIN : Binomiale négative.
<code>negbin_para_type</code>	Specify Estimate	
<code>negbin_parameter</code>	<i>nombre</i>	La valeur par défaut est 1. Doit contenir un nombre réel non négatif.
<code>tweedie_parameter</code>	<i>nombre</i>	
<code>link_function</code>	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPower PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG : Log-log complémentaire. LOGC : complément log. NEGBIN : Binomiale négative. NLOGLOG : Log-log négatif. CUMCAUCHIT : Cauchit cumulé. CUMCLOGLOG : Log-log complémentaire cumulé. CUMLOGIT : Logit cumulé. CUMNLOGLOG : Log-log négatif cumulé. CUMPROBIT : Probit cumulé.
<code>power</code>	<i>nombre</i>	La valeur doit être un nombre réel autre que zéro.
<code>method</code>	Hybrid Fisher NewtonRaphson	
<code>max_fisher_iterations</code>	<i>nombre</i>	La valeur par défaut est 1 ; seuls les entiers positifs sont autorisés.

Tableau 100. Propriétés du noeud genlin (suite).

Propriétés du noeud genlin	Valeurs	Description de la propriété
scale_method	MaxLikelihoodEstimate Deviance PearsonChiSquare FixedValue	
scale_value	<i>nombre</i>	La valeur par défaut est 1 ; elle doit être supérieure à 0.
covariance_matrix	ModelEstimator RobustEstimator	
max_iterations	<i>nombre</i>	La valeur par défaut est 100 ; entiers non négatifs uniquement.
max_step_halving	<i>nombre</i>	La valeur par défaut est 5 ; entiers positifs uniquement.
check_separation	<i>booléen</i>	
start_iteration	<i>nombre</i>	La valeur par défaut est 20 ; seuls les entiers positifs sont autorisés.
estimates_change	<i>booléen</i>	
estimates_change_min	<i>nombre</i>	La valeur par défaut est 1E-006 ; seuls les nombres positifs sont autorisés.
estimates_change_type	Absolute Relative	
loglikelihood_change	<i>booléen</i>	
loglikelihood_change_min	<i>nombre</i>	Seuls les nombres positifs sont autorisés.
loglikelihood_change_type	Absolute Relative	
hessian_convergence	<i>booléen</i>	
hessian_convergence_min	<i>nombre</i>	Seuls les nombres positifs sont autorisés.
hessian_convergence_type	Absolute Relative	
case_summary	<i>booléen</i>	
contrast_matrices	<i>booléen</i>	
descriptive_statistics	<i>booléen</i>	
estimable_functions	<i>booléen</i>	
model_info	<i>booléen</i>	
iteration_history	<i>booléen</i>	
goodness_of_fit	<i>booléen</i>	
print_interval	<i>nombre</i>	La valeur par défaut est 1 ; il doit s'agir d'un entier positif.
model_summary	<i>booléen</i>	
lagrange_multiplieur	<i>booléen</i>	
parameter_estimates	<i>booléen</i>	
include_exponential	<i>booléen</i>	
covariance_estimates	<i>booléen</i>	
correlation_estimates	<i>booléen</i>	

Tableau 100. Propriétés du noeud *genlin* (suite).

Propriétés du noeud <i>genlin</i>	Valeurs	Description de la propriété
<i>analysis_type</i>	TypeI TypeIII TypeIAndTypeIII	
<i>statistics</i>	Wald LR	
<i>citype</i>	Wald Profile	
<i>tolerancelevel</i>	<i>nombre</i>	La valeur par défaut est 0,0001.
<i>confidence_interval</i>	<i>nombre</i>	La valeur par défaut est 95.
<i>loglikelihood_function</i>	Complet Kernel	
<i>singularity_tolerance</i>	1E-007 1E-008 1E-009 1E-010 1E-011 1E-012	
<i>value_order</i>	Ascending Descending DataOrder	
<i>calculate_variable_importance</i>	<i>booléen</i>	
<i>calculate_raw_propensities</i>	<i>booléen</i>	
<i>calculate_adjusted_propensities</i>	<i>booléen</i>	
<i>adjusted_propensity_partition</i>	Test Validation	

Propriétés du noeud MMLG



Un modèle mixte linéaire généralisé (MMLG) élargit le modèle linéaire de sorte que la cible puisse avoir une distribution non normale, qu'elle soit liée linéairement aux facteurs et covariables via une fonction de lien spécifiée, et que les observations puissent être corrélées. Les modèles mixtes linéaires généralisés couvrent une large variété de modèles, depuis les modèles de régression linéaire simple aux modèles multi-niveaux complexes destinés aux données longitudinales non normales.

Tableau 101. Propriétés du noeud *MMLG*.

Propriétés du noeud <i>MMLG</i>	Valeurs	Description de la propriété
<i>residual_subject_spec</i>	<i>structurée</i>	La combinaison des valeurs des champs catégoriels spécifiés qui définissent de manière unique les sujets dans le jeu de données
<i>repeated_measures</i>	<i>structurée</i>	Champs à utiliser pour identifier les observations répétées.
<i>residual_group_spec</i>	[<i>champ1 ... champN</i>]	Champs qui définissent les ensembles indépendants des paramètres de covariance d'effets répétés.

Tableau 101. Propriétés du noeud MMLG (suite).

Propriétés du noeud MMLG	Valeurs	Description de la propriété
residual_covariance_type	Diagonale AR1 ARMA11 COMPOUND_SYMMETRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS	Spécifie la structure de la covariance des résidus.
custom_target	booléen	Indique s'il faut utiliser la cible définie dans le noeud en amont (false) ou la cible personnalisée spécifiée par target_field (true).
target_field	champ	Champ à utiliser comme cible si custom_target a la valeur true.
use_trials	booléen	Indique si un champ supplémentaire ou une valeur spécifiant le nombre d'essais doit être utilisé lorsque la réponse cible est un nombre d'événements se produisant dans un ensemble d'essais. La valeur par défaut est false (faux).
use_field_or_value	Field Valeur	Indique si un champ (par défaut) ou une valeur est utilisé pour spécifier le nombre d'essais.
trials_field	champ	Champ à utiliser pour spécifier le nombre d'essais.
trials_value	entier	Valeur à utiliser pour spécifier le nombre d'essais. Si spécifiée, la valeur minimum est 1.
use_custom_target_reference	booléen	Indique si la catégorie de référence personnalisée doit être utilisée pour une cible catégorielle. La valeur par défaut est false (faux).
target_reference_value	chaîne	Catégorie de référence à utiliser si use_custom_target_reference a la valeur true.
dist_link_combination	Nominal Logit GammaLog BinomialLogit PoissonLog BinomialProbit NegbinLog BinomialLogC Custom	Modèles standard pour la distribution des valeurs de la cible. Choisissez Custom pour spécifier une distribution à partir de la liste fournie par target_distribution.
target_distribution	Normal Binomial Multinomial Gamma Inverse NegativeBinomial Poisson	Distribution des valeurs de la cible quand dist_link_combination a la valeur Custom.

Tableau 101. Propriétés du noeud MMLG (suite).

Propriétés du noeud MMLG	Valeurs	Description de la propriété
link_function_type	IDENTITY LOGC LOG CLOGLOG LOGIT NLOGLOG PROBIT POWER CAUCHIT	Fonction de lien pour associer les valeurs cibles aux prédicteurs. Si target_distribution a la valeur Binomial, vous pouvez utiliser n'importe laquelle des fonctions de lien listées. Si target_distribution a la valeur Multinomial, vous pouvez utiliser CLOGLOG, CAUCHIT, LOGIT, NLOGLOG ou PROBIT. Si target_distribution a une valeur autre que Binomial ou Multinomial, vous pouvez utiliser IDENTITY, LOG ou POWER.
link_function_param	<i>nombre</i>	Valeur de paramètre de fonction de lien à utiliser. S'applique uniquement si normal_link_function ou link_function_type a la valeur POWER.
use_predefined_inputs	<i>booléen</i>	Indique si les champs d'effets fixes doivent être ceux définis en amont comme champs d'entrée (true) ou ceux de fixed_effects_list (false). La valeur par défaut est false (faux).
fixed_effects_list	<i>structurée</i>	Si use_predefined_inputs est false, spécifie les champs d'entrée à utiliser comme champs d'effets fixes.
use_intercept	<i>booléen</i>	Si true (par défaut), inclut la constante dans le modèle.
random_effects_list	<i>structurée</i>	Liste des champs à spécifier comme effets aléatoires.
regression_weight_field	<i>champ</i>	Champ à utiliser comme champ de pondération d'analyse.
use_offset	None offset_value offset_field	Indique comment le décalage est spécifié. La valeur None signifie qu'aucun décalage n'est spécifié.
offset_value	<i>nombre</i>	Valeur à utiliser pour le décalage si use_offset est défini sur offset_value.
offset_field	<i>champ</i>	Champ à utiliser pour la valeur de décalage si use_offset est défini sur offset_field.
target_category_order	Ascending Descending Data	Ordre de tri des cibles catégorielles. La valeur Data spécifie qu'il faut utiliser l'ordre de tri trouvé dans les données. La valeur par défaut est Ascending.
inputs_category_order	Ascending Descending Data	Ordre de tri pour les prédicteurs indépendants. La valeur Data spécifie qu'il faut utiliser l'ordre de tri trouvé dans les données. La valeur par défaut est Ascending.
max_iterations	<i>entier</i>	Nombre maximum d'itérations que l'algorithme effectuera. Un entier positif ; la valeur par défaut est 100.

Tableau 101. Propriétés du noeud MMLG (suite).

Propriétés du noeud MMLG	Valeurs	Description de la propriété
confidence_level	entier	Niveau de confiance utilisé pour calculer les estimations d'intervalle des coefficients de modèle. Un entier positif ; le nombre maximum est 100, la valeur par défaut est 95.
degrees_of_freedom_method	Fixed Varied	Spécifie comment les degrés de liberté sont calculés pour le test de signification.
test_fixed_effects_coeffecients	Model Robust	Méthode de calcul de la matrice de covariance des estimations de paramètre.
use_p_converge	booléen	Option pour la convergence de paramètres.
p_converge	nombre	Vide ou valeur positive.
p_converge_type	Absolute Relative	
use_l_converge	booléen	Option pour la convergence du log de vraisemblance.
l_converge	nombre	Vide ou valeur positive.
l_converge_type	Absolute Relative	
use_h_converge	booléen	Option pour la convergence hessienne.
h_converge	nombre	Vide ou valeur positive.
h_converge_type	Absolute Relative	
max_fisher_steps	entier	
singularity_tolerance	nombre	
use_model_name	booléen	Indique s'il faut spécifier un nom personnalisé pour le modèle (true) ou utiliser le nom généré par le système (false). La valeur par défaut est false.
model_name	chaîne	Si use_model_name a la valeur true (vrai), indique le nom de modèle à utiliser.
confidence	onProbability onIncrease	Base de calcul de la valeur de confiance de scoring : probabilité prédite la plus élevée ou différence entre les probabilités prédites les plus élevées et les deuxièmes plus élevées.
score_category_probabilities	booléen	Si true, produit des probabilités prédites pour les cibles catégorielles. La valeur par défaut est false (faux).
max_categories	entier	Si score_category_probabilities a la valeur true, spécifie le nombre maximum de catégories à enregistrer.
score_propensity	booléen	Si true, produit des scores de propension pour les champs cibles indicateurs qui indiquent la probabilité du résultat « true » pour le champ.
emeans	structure	Pour chaque champ catégoriel de la liste des effets fixes, spécifie s'il faut produire des moyennes marginales estimées.

Tableau 101. Propriétés du noeud MMLG (suite).

Propriétés du noeud MMLG	Valeurs	Description de la propriété
covariance_list	structure	Pour chaque champ continu de la liste des effets fixes, spécifie s'il faut utiliser la moyenne ou une valeur personnalisée lors du calcul des moyennes marginales estimées.
mean_scale	Original Transformation	Spécifie s'il faut calculer les moyennes marginales estimées en fonction de l'échelle originale de la cible (par défaut) ou de la transformation de la fonction de lien.
comparison_adjustment_method	LSD SEQBONFERRONI SEQSIDAK	Méthode d'ajustement à utiliser lors des tests d'hypothèse avec contrastes multiples.

Propriétés du noeud kmeans



Le noeud k moyenne classe le jeu de données dans différents groupes (ou clusters). La méthode définit un nombre de clusters fixe, affecte à plusieurs reprises des enregistrements à des clusters et ajuste les centres de cluster, jusqu'à ce que le modèle ne puisse plus être amélioré. Au lieu de tenter de prédire un résultat, le modèle *k*-means utilise un processus connu sous le nom d'apprentissage non supervisé pour découvrir des tendances dans l'ensemble de champs d'entrée.

Tableau 102. Propriétés du noeud kmeans.

Propriétés du noeud kmeans	Valeurs	Description de la propriété
inputs	[champ1 ... champN]	Les modèles k moyenne procèdent à une analyse des clusters sur un ensemble de champs d'entrée, mais n'utilisent pas de champ cible. Les champs de pondération et de fréquence ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
num_clusters	nombre	
gen_distance	booléen	
cluster_label	String Nombre	
label_prefix	chaîne	
mode	Simple Expert	
stop_on	Default Custom	
max_iterations	nombre	
tolerance	nombre	
encoding_value	nombre	
optimize	Vitesse Memory	Permet d'indiquer si la création du modèle doit être optimisée en vitesse ou en mémoire.

Propriétés du noeud knn



Le noeud k -Voisin le plus proche (KNN) associe une nouvelle observation à la catégorie ou à la valeur des objets k les plus proches dans l'espace du prédicteur, où k est un entier. Les observations semblables sont proches l'une de l'autre et les observations dissemblables sont éloignées l'une de l'autre.

Tableau 103. Propriétés du noeud knn.

Propriétés du noeud knn	Valeurs	Description de la propriété
analyse	PredictTarget IdentifyNeighbors	
objective	Solde Vitesse Accuracy Custom	
normalize_ranges	booléen	
use_case_labels	booléen	Cochez la case pour activer l'option suivante.
case_labels_field	champ	
identify_focal_cases	booléen	Cochez la case pour activer l'option suivante.
focal_cases_field	champ	
automatic_k_selection	booléen	
fixed_k	entier	Activé uniquement si automatic_k_selectio est False
minimum_k	entier	Activé uniquement si automatic_k_selectio est True
maximum_k	entier	
distance_computation	Euclidean CityBlock	
weight_by_importance	booléen	
range_predictions	Mean Median	
perform_feature_selection	booléen	
forced_entry_inputs	[champ1 ... champN]	
stop_on_error_ratio	booléen	
number_to_select	entier	
minimum_change	nombre	
validation_fold_assign_by_field	booléen	
number_of_folds	entier	Activé uniquement si validation_fold_assign_by_field est False
set_random_seed	booléen	
random_seed	nombre	
folds_field	champ	Activé uniquement si validation_fold_assign_by_field est True
all_probabilities	booléen	
save_distances	booléen	

Tableau 103. Propriétés du noeud knn (suite).

Propriétés du noeud knn	Valeurs	Description de la propriété
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés du noeud Kohonen



Le noeud Kohonen génère un type de réseau de neurones qui peut être utilisé pour classer les données en groupes distincts. Lorsque l'apprentissage du réseau est terminé, les enregistrements similaires doivent être regroupés dans la connexion de sortie, tandis que les enregistrements différents sont à l'opposé. Vous pouvez étudier le nombre d'observations capturées par chaque unité du nugget de modèle afin d'identifier les unités fortes. Vous pouvez ainsi vous faire une idée du nombre de clusters approprié.

Tableau 104. Propriétés du noeud Kohonen.

Propriétés du noeud Kohonen	Valeurs	Description de la propriété
inputs	[champ1 ... champN]	Les modèles Kohonen utilisent une liste de champs d'entrée, mais pas de cible. Les champs de fréquences et de pondération ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
continue	booléen	
show_feedback	booléen	
stop_on	Default Heure	
time	nombre	
optimize	Vitesse Memory	Permet d'indiquer si la création du modèle doit être optimisée en vitesse ou en mémoire.
cluster_label	booléen	
mode	Simple Expert	
width	nombre	
length	nombre	
decay_style	Linear Exponential	
phase1_neighborhood	nombre	
phase1_eta	nombre	
phase1_cycles	nombre	
phase2_neighborhood	nombre	
phase2_eta	nombre	
phase2_cycles	nombre	

Propriétés du noeud Linéaire



Les modèles de régression linéaire prédisent une cible continue en fonction de relations linéaires entre la cible et un ou plusieurs prédicteurs.

Tableau 105. Propriétés du noeud Linéaire.

Propriétés du noeud Linéaire	Valeurs	Description de la propriété
target	<i>champ</i>	Spécifie un champ cible unique.
inputs	[<i>champ1 ... champN</i>]	Champs prédicteurs utilisés par le modèle.
continue_training_existing_model	<i>booléen</i>	
objective	Standard Bagging Boosting psm	psm est destinée aux jeux de données très volumineux et nécessite une connexion à un serveur.
use_auto_data_preparation	<i>booléen</i>	
confidence_level	<i>nombre</i>	
model_selection	ForwardStepwise BestSubsets None	
criteria_forward_stepwise	AICC Fstatistics AdjustedRSquared ASE	
probability_entry	<i>nombre</i>	
probability_removal	<i>nombre</i>	
use_max_effects	<i>booléen</i>	
max_effects	<i>nombre</i>	
use_max_steps	<i>booléen</i>	
max_steps	<i>nombre</i>	
criteria_best_subsets	AICC AdjustedRSquared ASE	
combining_rule_continuous	Mean Median	
component_models_n	<i>nombre</i>	
use_random_seed	<i>booléen</i>	
random_seed	<i>nombre</i>	
use_custom_model_name	<i>booléen</i>	
custom_model_name	<i>chaîne</i>	
use_custom_name	<i>booléen</i>	
custom_name	<i>chaîne</i>	
tooltip	<i>chaîne</i>	
keywords	<i>chaîne</i>	

Tableau 105. Propriétés du noeud Linéaire (suite).

Propriétés du noeud Linéaire	Valeurs	Description de la propriété
annotation	chaîne	

Propriétés du noeud logreg



La régression logistique est une technique statistique de classification des enregistrements sur la base des valeurs des champs d'entrée. Excepté le fait qu'elle utilise un champ cible catégoriel et non pas numérique, cette régression est similaire à la régression linéaire.

Tableau 106. Propriétés du noeud logreg.

Propriétés du noeud logreg	Valeurs	Description de la propriété
target	champ	Les modèles de régression logistique requièrent un seul champ cible et un ou plusieurs champs d'entrée. Les champs de fréquences et de pondération ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
logistic_procedure	Binomial Multinomial	
include_constant	booléen	
mode	Simple Expert	
method	Enter Stepwise Forwards Backwards BackwardsStepwise	
binomial_method	Enter Forwards Backwards	
model_type	MainEffects FullFactorial Custom	Lorsque le type de modèle indiqué est FullFactorial, les méthodes d'analyse pas à pas ne sont pas exécutées, même si elles sont spécifiées. La méthode utilisée sera la méthode Enter. Si le type de modèle est paramétré sur Custom, mais qu'aucun champ personnalisé n'est indiqué, un modèle Effets principaux est créé.
custom_terms	{BP Sex}{BP}{Age}	
multinomial_base_category	chaîne	Indique le mode de détermination de la catégorie de référence.
binomial_categorical_input	chaîne	

Tableau 106. Propriétés du noeud logreg (suite).

Propriétés du noeud logreg	Valeurs	Description de la propriété
binomial_input_contrast	Indicator Simple Difference Helmert Repeated Polynomial Deviation	Propriété saisie pour l'entrée catégorielle qui indique la façon dont le contraste est déterminé.
binomial_input_category	First Last	Propriété saisie pour l'entrée catégorielle qui indique la façon dont la catégorie de référence est déterminée.
scale	None UserDefined Pearson Deviance	
scale_value	<i>nombre</i>	
all_probabilities	<i>booléen</i>	
tolerance	1.0E-5 1.0E-6 1.0E-7 1.0E-8 1,0E-9 1.0E-10	
min_terms	<i>nombre</i>	
use_max_terms	<i>booléen</i>	
max_terms	<i>nombre</i>	
entry_criterion	Score LR	
removal_criterion	LR Wald	
probability_entry	<i>nombre</i>	
probability_removal	<i>nombre</i>	
binomial_probability_entry	<i>nombre</i>	
binomial_probability_removal	<i>nombre</i>	
requirements	HierarchyDiscrete HierarchyAll Containment None	
max_iterations	<i>nombre</i>	
max_steps	<i>nombre</i>	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	

Tableau 106. Propriétés du noeud logreg (suite).

Propriétés du noeud logreg	Valeurs	Description de la propriété
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
delta	nombre	
iteration_history	booléen	
history_steps	nombre	
summary	booléen	
likelihood_ratio	booléen	
asymptotic_correlation	booléen	
goodness_fit	booléen	
parameters	booléen	
confidence_interval	nombre	
asymptotic_covariance	booléen	
classification_table	booléen	
stepwise_summary	booléen	
info_criteria	booléen	
monotonicity_measures	booléen	
binomial_output_display	at_each_step at_last_step	
binomial_goodness_of_fit	booléen	
binomial_parameters	booléen	
binomial_iteration_history	booléen	
binomial_classification_plots	booléen	
binomial_ci_enable	booléen	
binomial_ci	nombre	
binomial_residual	outliers all	
binomial_residual_enable	booléen	
binomial_outlier_threshold	nombre	
binomial_classification_cutoff	nombre	
binomial_removal_criterion	LR Wald Conditional	
calculate_variable_importance	booléen	
calculate_raw_propensities	booléen	

Propriétés du noeud neuralnet

Attention : une nouvelle version du noeud de modélisation Réseau de neurones, avec des caractéristiques améliorées, est disponible dans cette version et est décrite dans la section suivante (*neuralnetwork*). Bien que vous soyez encore en mesure de créer et d'évaluer un modèle avec la version précédente, nous vous conseillons de procéder à la mise à jour de vos scripts afin d'utiliser la nouvelle version. Les détails de la version précédentes sont conservées ici pour référence.

Tableau 107. Propriétés du noeud neuralnet.

Propriétés du noeud neuralnet	Valeurs	Description de la propriété
targets	[<i>champ1 ... champN</i>]	Le noeud Réseau de neurones attend un ou plusieurs champs cible et un ou plusieurs champs d'entrée. Les champs de fréquence et de pondération sont ignorés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
method	Quick Dynamic Multiple Prune ExhaustivePrune RBFN	
prevent_overtrain	<i>booléen</i>	
train_pct	<i>nombre</i>	
set_random_seed	<i>booléen</i>	
random_seed	<i>nombre</i>	
mode	Simple Expert	
stop_on	Default Accuracy Cycles Heure	Mode d'arrêt.
accuracy	<i>nombre</i>	Exactitude d'arrêt.
cycles	<i>nombre</i>	Cycles d'apprentissage.
time	<i>nombre</i>	Durée d'apprentissage (minutes).
continue	<i>booléen</i>	
show_feedback	<i>booléen</i>	
binary_encode	<i>booléen</i>	
use_last_model	<i>booléen</i>	
gen_logfile	<i>booléen</i>	
logfile_name	<i>chaîne</i>	
alpha	<i>nombre</i>	
initial_eta	<i>nombre</i>	
high_eta	<i>nombre</i>	
low_eta	<i>nombre</i>	
eta_decay_cycles	<i>nombre</i>	

Tableau 107. Propriétés du noeud neuralnet (suite).

Propriétés du noeud neuralnet	Valeurs	Description de la propriété
hid_layers	One Two Three	
hl_units_one	<i>nombre</i>	
hl_units_two	<i>nombre</i>	
hl_units_three	<i>nombre</i>	
persistence	<i>nombre</i>	
m_topologies	<i>chaîne</i>	
m_non_pyramids	<i>booléen</i>	
m_persistence	<i>nombre</i>	
p_hid_layers	One Two Three	
p_hl_units_one	<i>nombre</i>	
p_hl_units_two	<i>nombre</i>	
p_hl_units_three	<i>nombre</i>	
p_persistence	<i>nombre</i>	
p_hid_rate	<i>nombre</i>	
p_hid_pers	<i>nombre</i>	
p_inp_rate	<i>nombre</i>	
p_inp_pers	<i>nombre</i>	
p_overall_pers	<i>nombre</i>	
r_persistence	<i>nombre</i>	
r_num_clusters	<i>nombre</i>	
r_eta_auto	<i>booléen</i>	
r_alpha	<i>nombre</i>	
r_eta	<i>nombre</i>	
optimize	Vitesse Memory	Permet d'indiquer si la création du modèle doit être optimisée en vitesse ou en mémoire.
calculate_variable_importance	<i>booléen</i>	Remarque : la propriété <code>sensitivity_analysis</code> utilisée dans les versions précédentes est remplacée par cette propriété. L'ancienne propriété est toujours prise en charge, mais il est recommandé d'utiliser <code>calculate_variable_importance</code> .
calculate_raw_propensities	<i>booléen</i>	
calculate_adjusted_propensities	<i>booléen</i>	
adjusted_propensity_partition	Test Validation	

Propriétés du noeud neuralnetwork

Le noeud R. neurones est un modèle simplifié de la manière dont le cerveau humain traite les informations. Le fonctionnement de ce modèle repose sur la simulation d'un grand nombre d'unités de traitement simples interconnectées, qui sont en quelque sorte des versions abstraites de nos neurones. Les réseaux de neurones sont de puissants estimateurs de fonctions qui ne requièrent qu'une connaissance limitée en matière de statistiques ou de mathématiques.

Tableau 108. Propriétés du noeud neuralnetwork.

Propriétés du noeud neuralnetwork	Valeurs	Description de la propriété
targets	[champ1 ... champN]	Spécifie des champs cible.
inputs	[champ1 ... champN]	Champs prédicteurs utilisés par le modèle.
splits	[champ1 ... champN]	Indique le champ ou les champs à utiliser pour la modélisation découpée.
use_partition	booléen	Si un champ de partition est défini, cette option assure que seules les données provenant de la partition d'apprentissage sont utilisées pour générer le modèle.
continue	booléen	Poursuivre le modèle d'apprentissage existant.
objective	Standard Bagging Boosting psm	psm est destinée aux jeux de données très volumineux et nécessite une connexion à un serveur.
method	MultilayerPerceptron RadialBasisFunction	
use_custom_layers	booléen	
first_layer_units	nombre	
second_layer_units	nombre	
use_max_time	booléen	
max_time	nombre	
use_max_cycles	booléen	
max_cycles	nombre	
use_min_accuracy	booléen	
min_accuracy	nombre	
combining_rule_categorical	Voting HighestProbability HighestMeanProbability	
combining_rule_continuous	Mean Median	
component_models_n	nombre	
overfit_prevention_pct	nombre	
use_random_seed	booléen	
random_seed	nombre	

Tableau 108. Propriétés du noeud neuralnetwork (suite).

Propriétés du noeud neuralnetwork	Valeurs	Description de la propriété
missing_values	listwiseDeletion missingValueImputation	
use_custom_model_name	booléen	
custom_model_name	chaîne	
confidence	onProbability onIncrease	
score_category_probabilities	booléen	
max_categories	nombre	
score_propensity	booléen	
use_custom_name	booléen	
custom_name	chaîne	
tooltip	chaîne	
keywords	chaîne	
annotation	chaîne	

Propriétés du noeud quest



Le noeud QUEST est une méthode de classification binaire permettant de créer des arbres décision, développée pour réduire le temps de traitement nécessaire aux analyses C&R Tree importantes, tout en limitant la tendance, observée parmi les méthodes d'arbre de classification, à favoriser les entrées autorisant un nombre supérieur de divisions. Les champs d'entrée peuvent être des intervalles numériques (continues) mais les champs cible doivent être catégoriels. Toutes les divisions sont binaires.

Tableau 109. Propriétés du noeud quest.

Propriétés du noeud quest	Valeurs	Description de la propriété
target	champ	Les modèles QUEST requièrent un seul champ cible et un ou plusieurs champs d'entrée. Un champ de fréquence peut aussi être spécifié. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
continue_training_existing_model	booléen	
objective	Standard Boosting Bagging psm	psm est destinée aux jeux de données très volumineux et nécessite une connexion à un serveur.
model_output_type	Single InteractiveBuilder	
use_tree_directives	booléen	
tree_directives	chaîne	
use_max_depth	Default Custom	
max_depth	entier	Profondeur maximale de l'arbre, comprise entre 0 et 1000. Utilisée uniquement si use_max_depth = Custom

Tableau 109. Propriétés du noeud quest (suite).

Propriétés du noeud quest	Valeurs	Description de la propriété
prune_tree	booléen	Elagage de l'arbre pour éviter le surajustement.
use_std_err	booléen	Utiliser la différence maximale pour le risque (dans Erreurs standard).
std_err_multiplier	nombre	Différence maximale.
max_surrogates	nombre	Nombre maximal de substitutions.
use_percentage	booléen	
min_parent_records_pc	nombre	
min_child_records_pc	nombre	
min_parent_records_abs	nombre	
min_child_records_abs	nombre	
use_costs	booléen	
costs	structurée	Propriétés structurées.
priors	Data Equal Custom	
custom_priors	structurée	Propriétés structurées.
adjust_priors	booléen	
trails	nombre	Nombre des modèles de composant pour le boosting ou le bagging.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Règles de combinaison par défaut pour les cibles catégorielles.
range_ensemble_method	Mean Median	Règles de combinaison par défaut pour les cibles continues.
large_boost	booléen	Appliquer le boosting aux jeux de données très volumineux.
split_alpha	nombre	Niveau de signification pour la division.
train_pct	nombre	Ensemble de prévention de surajustement.
set_random_seed	booléen	Dupliquer l'option des résultats.
seed	nombre	
calculate_variable_importance	booléen	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	
adjusted_propensity_partition	Test Validation	

Propriétés du noeud Régression



La régression linéaire est une technique statistique couramment utilisée dans le domaine de la synthèse de données et de la prévision. Cette technique établit une ligne droite ou une surface afin de réduire les écarts entre les valeurs de sortie prévues et observées.

Remarque : le noeud Régression sera remplacé par le noeud Linéaire dans une prochaine version. Nous recommandons d'ores et déjà d'utiliser les modèles linéaires pour la régression linéaire.

Tableau 110. Propriétés du noeud Régression.

Propriétés du noeud Régression	Valeurs	Description de la propriété
target	champ	Les modèles Régression requièrent un seul champ cible et un ou plusieurs champs d'entrée. Un champ poids peut aussi être spécifié. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
method	Enter Stepwise Backwards Forwards	
include_constant	booléen	
use_weight	booléen	
weight_field	champ	
mode	Simple Expert	
complete_records	booléen	
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	Pour les arguments, utilisez des guillemets doubles.
stepping_method	useP useF	useP : utiliser la probabilité de F useF : utiliser la valeur F
probability_entry	nombre	
probability_removal	nombre	
F_value_entry	nombre	
F_value_removal	nombre	
selection_criteria	booléen	
confidence_interval	booléen	
covariance_matrix	booléen	
collinearity_diagnostics	booléen	
regression_coefficients	booléen	
exclude_fields	booléen	
durbin_watson	booléen	
model_fit	booléen	
r_squared_change	booléen	
p_correlations	booléen	

Tableau 110. Propriétés du noeud Régression (suite).

Propriétés du noeud Régression	Valeurs	Description de la propriété
descriptives	booléen	
calculate_variable_importance	booléen	

Propriétés du noeud Séquence



Le noeud Séquence recherche des règles d'association dans des données dotées d'une dimension temporelle. Une séquence est une liste de jeux d'éléments ayant tendance à survenir dans un ordre prévisible. Par exemple, un client qui achète un rasoir et une lotion après-rasage achètera vraisemblablement de la crème à raser. Le noeud Séquence est basé sur l'algorithme de règles d'association CARMA, qui utilise une méthode efficace de double lecture pour rechercher des séquences.

Tableau 111. Propriétés du noeud Séquence.

Propriétés du noeud Séquence	Valeurs	Description de la propriété
id_field	champ	Pour créer un modèle Séquence, vous devez renseigner un champ ID, un champ de temps facultatif et au moins un champ d'analyse. Les champs de pondération et de fréquence ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
time_field	champ	
use_time_field	booléen	
content_fields	[champ1 ... champn]	
contiguous	booléen	
min_supp	nombre	
min_conf	nombre	
max_size	nombre	
max_predictions	nombre	
mode	Simple Expert	
use_max_duration	booléen	
max_duration	nombre	
use_gaps	booléen	
min_item_gap	nombre	
max_item_gap	nombre	
use_pruning	booléen	
pruning_value	nombre	
set_mem_sequences	booléen	
mem_sequences	entier	

Propriétés du noeud MRAA



Le noeud Modèle de réponse en auto-apprentissage (SLRM) vous permet de créer un modèle dans lequel une nouvelle observation unique, ou un petit nombre de nouvelles observations, peuvent être utilisés pour réestimer un modèle sans qu'un recyclage de toutes les données soit nécessaire.

Tableau 112. Propriétés du noeud MRAA.

Propriétés du noeud MRAA	Valeurs	Description de la propriété
target	<i>champ</i>	Le champ cible doit être nominal ou indicateur. Un champ de fréquence peut aussi être spécifié. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
target_response	<i>champ</i>	Type doit être indicateur.
continue_training_existing_model	<i>booléen</i>	
target_field_values	<i>booléen</i>	Use all : Utiliser toutes les valeurs source. Specify : Sélectionner les valeurs nécessaires.
target_field_values_specify	<i>[champ1 ... champN]</i>	
include_model_assessment	<i>booléen</i>	
model_assessment_random_seed	<i>nombre</i>	Doit être un nombre réel.
model_assessment_sample_size	<i>nombre</i>	Doit être un nombre réel.
model_assessment_iterations	<i>nombre</i>	Nombre d'itérations.
display_model_evaluation	<i>booléen</i>	
max_predictions	<i>nombre</i>	
randomization	<i>nombre</i>	
scoring_random_seed	<i>nombre</i>	
sort	Ascending Descending	Indique si les premières offres affichées sont celles dont les scores sont les plus élevés ou les moins élevés.
model_reliability	<i>booléen</i>	
calculate_variable_importance	<i>booléen</i>	

Propriétés du noeud statisticsmodel



Le noeud Modèle Statistics vous permet d'analyser et de travailler avec vos données en exécutant des procédures IBM SPSS Statistics qui produisent un PMML. Ce noeud requiert une copie avec licence de IBM SPSS Statistics.

Les propriétés de ce noeud sont décrites dans «Propriétés du noeud statisticsmodel», à la page 242.

Propriétés du noeud svm



Le noeud Support Vector Machine (SVM) vous permet de classer les données dans l'un de deux groupes sans surajustement. SVM fonctionne bien avec les grands jeux de données, comme ceux qui disposent d'un très grand nombre de champs d'entrée.

Tableau 113. Propriétés du noeud svm.

Propriétés du noeud svm	Valeurs	Description de la propriété
all_probabilities	<i>booléen</i>	
stopping_criteria	1.0E-1 1.0E-2 1.0E-3 (valeur par défaut) 1.0E-4 1.0E-5 1.0E-6	Détermine le moment de l'arrêt de l'algorithme d'optimisation.
regularization	<i>nombre</i>	Egalement appelé Paramètre C
precision	<i>nombre</i>	Option uniquement utilisée si le niveau de mesure du champ cible est Continuus.
kernel	RBF(valeur par défaut) Polynomial Sigmoid Linear	Type de fonction de noyau utilisée pour la transformation.
rbf_gamma	<i>nombre</i>	Utilisé uniquement si le kernel est RBF.
gamma	<i>nombre</i>	Utilisé uniquement si le kernel est Polynomial ou Sigmoid.
bias	<i>nombre</i>	
degree	<i>nombre</i>	Utilisé uniquement si le kernel est Polynomial.
calculate_variable_importance	<i>booléen</i>	
calculate_raw_propensities	<i>booléen</i>	
calculate_adjusted_propensities	<i>booléen</i>	
adjusted_propensity_partition	Test Validation	

Propriétés du noeud timeseries



Le noeud Séries temporelles estime les modèles de lissage exponentiel, d'ARIMA (Autoregressive Integrated Moving Average) univariable et d'ARIMA multivariable (ou fonction de transfert) pour les données de séries temporelles et génère des prévisions d'une performance future. Un noeud Séries temporelles doit toujours être précédé d'un noeud Intervalles de temps.

Tableau 114. Propriétés du noeud timeseries.

Propriétés du noeud timeseries	Valeurs	Description de la propriété
targets	champ	Le noeud Séries temporelles prévoit une ou plusieurs cibles, utilisant éventuellement un ou plusieurs champs d'entrée en tant que prédictors. Les champs de fréquences et de pondération ne sont pas utilisés. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
continue	booléen	
method	ExpertModeler Exsmooth Arima Reuse	
expert_modeler_method	booléen	
consider_seasonal	booléen	
detect_outliers	booléen	
expert_outlier_additive	booléen	
expert_outlier_level_shift	booléen	
expert_outlier_innovational	booléen	
expert_outlier_level_shift	booléen	
expert_outlier_transient	booléen	
expert_outlier_seasonal_additive	booléen	
expert_outlier_local_trend	booléen	
expert_outlier_additive_patch	booléen	
exsmooth_model_type	Simple HoltLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima_p	entier	
arima_d	entier	
arima_q	entier	
arima_sp	entier	
arima_sd	entier	
arima_sq	entier	
arima_transformation_type	None SquareRoot NaturalLog	

Tableau 114. Propriétés du noeud timeseries (suite).

Propriétés du noeud timeseries	Valeurs	Description de la propriété
arima_include_constant	booléen	
tf_arima_p. <i>nomchamp</i>	entier	Pour les fonctions de transfert.
tf_arima_d. <i>nomchamp</i>	entier	Pour les fonctions de transfert.
tf_arima_q. <i>nomchamp</i>	entier	Pour les fonctions de transfert.
tf_arima_sp. <i>nomchamp</i>	entier	Pour les fonctions de transfert.
tf_arima_sd. <i>nomchamp</i>	entier	Pour les fonctions de transfert.
tf_arima_sq. <i>nomchamp</i>	entier	Pour les fonctions de transfert.
tf_arima_delay. <i>nomchamp</i>	entier	Pour les fonctions de transfert.
tf_arima_transformation_type. <i>nomchamp</i>	None SquareRoot NaturalLog	Pour les fonctions de transfert.
arima_detect_outlier_mode	None Automatique	
arima_outlier_additive	booléen	
arima_outlier_level_shift	booléen	
arima_outlier_innovational	booléen	
arima_outlier_transient	booléen	
arima_outlier_seasonal_additive	booléen	
arima_outlier_local_trend	booléen	
arima_outlier_additive_patch	booléen	
conf_limit_pct	réel	
max_lags	entier	
events	champs	
scoring_model_only	booléen	Utilisation pour les modèles comportant un grand nombre (des dizaines de milliers) de séries temporelles.

Propriétés du noeud twostep



Le noeud TwoStep utilise une méthode de classification non supervisée en deux étapes. La première étape consiste en une exploration des données visant à compresser les données d'entrée brutes en sous-clusters plus faciles à manipuler. Au cours de la seconde étape, l'utilisation d'une méthode de classification hiérarchique permet de fusionner progressivement les sous-clusters en clusters de plus en plus importants. La technique TwoStep a l'avantage d'évaluer automatiquement le nombre de clusters optimal pour les données d'apprentissage. Il peut prendre en charge de manière efficace des types de champ mixtes et des jeux de données volumineux.

Tableau 115. Propriétés du noeud twostep.

Propriétés du noeud twostep	Valeurs	Description de la propriété
inputs	[<i>champ1 ... champN</i>]	Les modèles TwoStep utilisent une liste de champs d'entrée, mais pas de cible. Les champs de pondération et de fréquence ne sont pas reconnus. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds de modélisation», à la page 131.
standardize	<i>booléen</i>	
exclude_outliers	<i>booléen</i>	
percentage	<i>nombre</i>	
cluster_num_auto	<i>booléen</i>	
min_num_clusters	<i>nombre</i>	
max_num_clusters	<i>nombre</i>	
num_clusters	<i>nombre</i>	
cluster_label	Chaîne Nombre	
label_prefix	<i>chaîne</i>	
distance_measure	Euclidean Loglikelihood	
clustering_criterion	AIC BIC	

Chapitre 14. Propriétés du noeud de nugget de modèle

Les noeuds de nugget de modèle partagent les mêmes propriétés que les autres noeuds. Pour plus d'informations, voir la rubrique «Propriétés communes des noeuds», à la page 58.

Propriétés du noeud `applyanomalydetection`

Les noeuds de modélisation Détection des anomalies peuvent être utilisés pour générer un nugget de modèle Détection des anomalies. Le nom de génération de script de ce nugget de modèle est `applyanomalydetection`. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud `anomalydetection`», à la page 131.

Tableau 116. Propriétés du noeud `applyanomalydetection`.

Propriétés du noeud <code>applyanomalydetection</code>	Valeurs	Description de la propriété
<code>anomaly_score_method</code>	FlagAndScore FlagOnly ScoreOnly	Détermine les sorties créées pour le scoring.
<code>num_fields</code>	<i>entier</i>	Champs à signaler.
<code>discard_records</code>	<i>booléen</i>	Indique si les enregistrements sont supprimés de la sortie.
<code>discard_anomalous_records</code>	<i>booléen</i>	Indique s'il faut supprimer les enregistrements irréguliers ou <i>réguliers</i> . Cette propriété est désactivée par défaut, ce qui signifie que les enregistrements <i>réguliers</i> sont supprimés. Si elle est on, les enregistrements irréguliers sont supprimés. Cette propriété n'est activée que si la propriété <code>discard_records</code> l'est également.

Propriétés du noeud `applyapriori`

Les noeuds de modélisation Apriori peuvent être utilisés pour générer un nugget de modèle Apriori. Le nom de génération de script de ce nugget de modèle est `applyapriori`. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud `apriori`», à la page 133.

Tableau 117. Propriétés du noeud `applyapriori`.

Propriétés du noeud <code>applyapriori</code>	Valeurs	Description de la propriété
<code>max_predictions</code>	<i>nombre (entier)</i>	
<code>ignore_unmatched</code>	<i>booléen</i>	
<code>allow_repeats</code>	<i>booléen</i>	
<code>check_basket</code>	NoPredictions Predictions NoCheck	
<code>criterion</code>	Confidence Support RuleSupport Lift Deployability	

Propriétés du noeud applyautoclassifier

Les noeuds de modélisation de Discriminant automatique peuvent être utilisés pour générer un nugget de modèle de Discriminant automatique. Le nom de génération de script de ce nugget de modèle est *applyautoclassifier*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud autoclassifier», à la page 134.

Tableau 118. Propriétés du noeud applyautoclassifier.

Propriétés du noeud applyautoclassifier	Valeurs	Description de la propriété
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence AverageRawPropensity	Indique la méthode utilisée pour déterminer le score de l'ensemble. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ indicateur.
flag_voting_tie_selection	Random HighestConfidence RawPropensity	Si une méthode de vote est sélectionnée, indique la manière dont les ex æquo sont résolus. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ indicateur.
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	Indique la méthode utilisée pour déterminer le score de l'ensemble. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ d'ensemble.
set_voting_tie_selection	Random HighestConfidence	Si une méthode de vote est sélectionnée, indique la manière dont les ex æquo sont résolus. Ce paramètre s'applique uniquement si la cible sélectionnée est un champ nominal.

Propriétés du noeud applyautocluster

Les noeuds de modélisation de cluster automatique peuvent être utilisés pour générer un nugget de modèle de cluster automatique. Le nom de génération de script de ce nugget de modèle est *applyautocluster*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud autocluster», à la page 136.

Propriétés du noeud applyautonumeric

Les noeuds de modélisation de numérisation automatique peuvent être utilisés pour générer un nugget de modèle de numérisation automatique. Le nom de génération de script de ce nugget de modèle est *applyautonumeric*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud autonumeric», à la page 137.

Tableau 119. Propriétés du noeud applyautonumeric.

Propriétés du noeud applyautonumeric	Valeurs	Description de la propriété
calculate_standard_error	booléen	

Propriétés du noeud applybayesnet

Les noeuds de modélisation Réseau Bayésien peuvent être utilisés pour générer un nugget de modèle Réseau Bayésien. Le nom de génération de script de ce nugget de modèle est *applybayesnet*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud bayesnet», à la page 138.

Tableau 120. Propriétés du noeud applybayesnet.

Propriétés du noeud applybayesnet	Valeurs	Description de la propriété
all_probabilities	booléen	
raw_propensity	booléen	
adjusted_propensity	booléen	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	

Propriétés du noeud applyc50

Les noeuds de modélisation C5.0 peuvent être utilisés pour générer un nugget de modèle C5.0. Le nom de génération de script de ce nugget de modèle est *applyc50*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud c50», à la page 140.

Tableau 121. Propriétés du noeud applyc50.

Propriétés du noeud applyc50	Valeurs	Description de la propriété
sql_generate	Never NoMissingValues	Permet de définir les options de génération SQL au cours de l'exécution des ensembles de règles.
calculate_conf	booléen	Disponible lorsque la génération SQL est activée, cette propriété inclut des calculs de confiance dans l'arbre généré.
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	

Propriétés du noeud applycarma

Les noeuds de modélisation CARMA peuvent être utilisés pour générer un nugget de modèle CARMA. Le nom de génération de script de ce nugget de modèle est *applycarma*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud Carma», à la page 141.

Propriétés du noeud applycart

Des noeuds de modélisation d'arbre C&R peuvent être utilisés pour générer un nugget de modèle d'arbre C&R. Le nom de génération de script de ce nugget de modèle est *applycart*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud cart», à la page 142.

Tableau 122. Propriétés du noeud applycart.

Propriétés du noeud applycart	Valeurs	Description de la propriété
sql_generate	Never MissingValues NoMissingValues	Permet de définir les options de génération SQL au cours de l'exécution des ensembles de règles.
calculate_conf	booléen	Disponible lorsque la génération SQL est activée, cette propriété inclut des calculs de confiance dans l'arbre généré.
display_rule_id	booléen	Ajoute un champ à la sortie de scoring, indiquant l'ID du noeud terminal auquel chaque enregistrement est affecté.
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	

Propriétés du noeud applychaid

Les noeuds de modélisation CHAID peuvent être utilisés pour générer un nugget de modèle CHAID. Le nom de génération de script de ce nugget de modèle est *applychaid*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud chaid», à la page 143.

Tableau 123. Propriétés du noeud applychaid.

Propriétés du noeud applychaid	Valeurs	Description de la propriété
sql_generate	Never MissingValues	
calculate_conf	booléen	
display_rule_id	booléen	Ajoute un champ à la sortie de scoring, indiquant l'ID du noeud terminal auquel chaque enregistrement est affecté.
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	

Propriétés du noeud applycoxreg

Les noeuds de modélisation de Cox peuvent être utilisés pour générer un nugget de modèle de Cox. Le nom de génération de script de ce nugget de modèle est *applycoxreg*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud coxreg», à la page 145.

Tableau 124. Propriétés du noeud applycoxreg.

Propriétés du noeud applycoxreg	Valeurs	Description de la propriété
future_time_as	Intervals Fields	
time_interval	nombre	
num_future_times	entier	
time_field	champ	
past_survival_time	champ	
all_probabilities	booléen	
cumulative_hazard	booléen	

Propriétés du noeud applydecisionlist

Les noeuds de modélisation Liste de décision peuvent être utilisés pour générer un nugget de modèle Liste de décision. Le nom de génération de script de ce nugget de modèle est *applydecisionlist*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud decisionlist», à la page 146.

Tableau 125. Propriétés du noeud applydecisionlist.

Propriétés du noeud applydecisionlist	Valeurs	Description de la propriété
enable_sql_generation	booléen	Lorsque la valeur est true (vrai), IBM SPSS Modéler essaie de répercuter le modèle Liste de décision dans SQL.
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	

Propriétés du noeud applydiscriminant

Les noeuds de modélisation Discriminant peuvent être utilisés pour générer un nugget de modèle Discriminant. Le nom de génération de script de ce nugget de modèle est *applydiscriminant*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud discriminant», à la page 148.

Tableau 126. Propriétés du noeud applydiscriminant.

Propriétés du noeud applydiscriminant	Valeurs	Description de la propriété
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	

Propriétés du noeud applyfactor

Les noeuds de modélisation APC/Facteur peuvent être utilisés pour générer un nugget de modèle APC/Facteur. Le nom de génération de script de ce nugget de modèle est *applyfactor*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud ACP/Facteur», à la page 149.

Propriétés du noeud applyfeatureselection

Les noeuds de modélisation Sélection de fonction peuvent être utilisés pour générer un nugget de modèle Sélection de fonction. Le nom de génération de script de ce nugget de modèle est *applyfeatureselection*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud featureselection», à la page 150.

Tableau 127. Propriétés du noeud applyfeatureselection.

Propriétés du noeud applyfeatureselection	Valeurs	Description de la propriété
selected_ranked_fields		Indique les champs classés sélectionnés dans le navigateur de modèle.

Tableau 127. Propriétés du noeud *applyfeatureselection* (suite).

Propriétés du noeud <i>applyfeatureselection</i>	Valeurs	Description de la propriété
<code>selected_screened_fields</code>		Indique les champs filtrés sélectionnés dans le navigateur de modèle.

Propriétés du noeud *applygeneralizedlinear*

Les noeuds de modélisation linéaire généralisée (*genlin*) peuvent être utilisés pour générer un nugget de modèle linéaire généralisé. Le nom de génération de script de ce nugget de modèle est *applygeneralizedlinear*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud *genlin*», à la page 152.

Tableau 128. Propriétés du noeud *applygeneralizedlinear*.

Propriétés du noeud <i>applygeneralizedlinear</i>	Valeurs	Description de la propriété
<code>calculate_raw_propensities</code>	<i>booléen</i>	
<code>calculate_adjusted_propensities</code>	<i>booléen</i>	

Propriétés du noeud *applyglm*

Les noeuds de modélisation MMLG peuvent être utilisés pour générer un nugget de modèle MMLG. Le nom de génération de script de ce nugget de modèle est *applyglm*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud MMLG», à la page 155.

Tableau 129. Propriétés du noeud *applyglm*.

Propriétés du noeud <i>applyglm</i>	Valeurs	Description de la propriété
<code>confidence</code>	<code>onProbability</code> <code>onIncrease</code>	Base de calcul de la valeur de confiance de scoring : probabilité prédite la plus élevée ou différence entre les probabilités prédites les plus élevées et les deuxièmes plus élevées.
<code>score_category_probabilities</code>	<i>booléen</i>	Si elle est définie sur <code>True</code> , produit des probabilités prédites pour les cibles catégorielles. Un champ est créé pour chaque modalité. La valeur par défaut est <code>False</code> .
<code>max_categories</code>	<i>entier</i>	Nombre maximal de catégories pour lesquelles prédire des probabilités. Utilisée uniquement si <code>score_category_probabilities</code> a la valeur <code>True</code> .
<code>score_propensity</code>	<i>booléen</i>	Si elle est définie sur <code>True</code> , produit des scores de propension brute (probabilité du résultat " <code>True</code> ") pour les modèles avec des cibles indicateur. Si les partitions sont activées, produit également des scores de propension ajustée basés sur la partition de test. La valeur par défaut est <code>False</code> .

Propriétés du noeud applykmeans

Les noeuds de modélisation k moyenne peuvent être utilisés pour générer un nugget de modèle k moyenne. Le nom de génération de script de ce nugget de modèle est *applykmeans*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud kmeans», à la page 159.

Propriétés du noeud applyknn

Les noeuds de modélisation KNN peuvent être utilisés pour générer un nugget de modèle KNN. Le nom de génération de script de ce nugget de modèle est *applyknn*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud knn», à la page 160.

Tableau 130. Propriétés du noeud applyknn.

Propriétés du noeud applyknn	Valeurs	Description de la propriété
all_probabilities	booléen	
save_distances	booléen	

Propriétés du noeud applykohonen

Les noeuds de modélisation Kohonen peuvent être utilisés pour générer un nugget de modèle Kohonen. Le nom de génération de script de ce nugget de modèle est *applykohonen*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud Kohonen», à la page 161.

Propriétés du noeud applylinear

Les noeuds de modélisation linéaire peuvent être utilisés pour générer un nugget de modèle linéaire. Le nom de génération de script de ce nugget de modèle est *applylinear*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud Linéaire», à la page 162.

Tableau 131. Propriétés du noeud applylinear.

Propriétés du noeud applylinear	Valeurs	Description de la propriété
use_custom_name	booléen	
custom_name	chaîne	
enable_sql_generation	booléen	

Propriétés du noeud applylogreg

Les noeuds de modélisation Régression logistique peuvent être utilisés pour générer un nugget de modèle Régression logistique. Le nom de génération de script de ce nugget de modèle est *applylogreg*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud logreg», à la page 163.

Tableau 132. Propriétés du noeud applylogreg.

Propriétés du noeud applylogreg	Valeurs	Description de la propriété
calculate_raw_propensities	booléen	
calculate_conf	booléen	
enable_sql_generation	booléen	

Propriétés du noeud applyneuralnet

Les noeuds de modélisation Réseau de neurones peuvent être utilisés pour générer un nugget de modèle Réseau de neurones. Le nom de génération de script de ce nugget de modèle est *applyneuralnet*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud neuralnet», à la page 166.

Attention : une nouvelle version du nugget Réseau de neurones, avec des fonctions améliorées, est disponible dans cette version et est décrite dans la section suivante (*applyneuralnetwork*). Bien que la version précédente soit encore disponible, nous vous conseillons de procéder à la mise à jour de vos scripts afin d'utiliser la nouvelle version. Les détails de la version précédente sont préservés ici à titre de référence, mais la prise en charge sera supprimée dans une version ultérieure.

Tableau 133. Propriétés du noeud applyneuralnet.

Propriétés du noeud applyneuralnet	Valeurs	Description de la propriété
calculate_conf	booléen	Disponible lorsque la génération SQL est activée, cette propriété inclut des calculs de confiance dans l'arbre généré.
enable_sql_generation	booléen	
nn_score_method	Difference SoftMax	
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	

Propriétés du noeud applyneuralnetwork

Les noeuds de modélisation Réseau de neurones peuvent être utilisés pour générer un nugget de modèle Réseau de neurones. Le nom de génération de script de ce nugget de modèle est *applyneuralnetwork*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud neuralnetwork», à la page 168.

Tableau 134. Propriétés du noeud applyneuralnetwork.

Propriétés du noeud applyneuralnetwork	Valeurs	Description de la propriété
use_custom_name	booléen	
custom_name	chaîne	
confidence	onProbability onIncrease	
score_category_probabilities	booléen	
max_categories	nombre	
score_propensity	booléen	

Propriétés du noeud applyquest

Les noeuds de modélisation QUEST peuvent être utilisés pour générer un nugget de modèle QUEST. Le nom de génération de script de ce nugget de modèle est *applyquest*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud quest», à la page 169.

Tableau 135. Propriétés du noeud *applyquest*.

Propriétés du noeud <i>applyquest</i>	Valeurs	Description de la propriété
sql_generate	Never MissingValues NoMissingValues	
calculate_conf	booléen	
display_rule_id	booléen	Ajoute un champ à la sortie de scoring, indiquant l'ID du noeud terminal auquel chaque enregistrement est affecté.
calculate_raw_propensities	booléen	
calculate_adjusted_propensities	booléen	

Propriétés du noeud *applyregression*

Les noeuds de modélisation Régression linéaire peuvent être utilisés pour générer un nugget de modèle Régression linéaire. Le nom de génération de script de ce nugget de modèle est *applyregression*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud Régression», à la page 170.

Propriétés du noeud *applyr*

Les noeuds de création R peuvent être utilisés pour générer un nugget de modèle R. Le nom de génération de script de ce nugget de modèle est *applyr*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud *buildr*», à la page 139.

Tableau 136. Propriétés du noeud *applyr*

Propriétés du noeud <i>applyr</i>	Valeurs	Description de la propriété
score_syntax	chaîne	Syntaxe de script R pour le scoring de modèle.
convert_flags	StringsAndDoubles LogicalValues	Option permettant de convertir des champs indicateurs.
convert_datetime	booléen	Option permettant de convertir des variables au format de date ou date-heure en format de date/heure R.
convert_datetime_class	POSIXct POSIXlt	Options permettant d'indiquer dans quel format sont converties les variables au format de date ou date-heure.
convert_missing	booléen	Option permettant de convertir des valeurs manquantes en valeur R NA.

Propriétés du noeud *applyselflearning*

Les noeuds de modélisation Réponse en auto-apprentissage (SLRM) peuvent être utilisés pour générer un nugget de modèle SLRM. Le nom de génération de script de ce nugget de modèle est *applyselflearning*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud *MRAA*», à la page 173.

Tableau 137. Propriétés du noeud *applyselflearning*.

Propriétés du noeud <i>applyselflearning</i>	Valeurs	Description de la propriété
<code>max_predictions</code>	<i>nombre</i>	
<code>randomization</code>	<i>nombre</i>	
<code>scoring_random_seed</code>	<i>nombre</i>	
<code>sort</code>	ascending descending	Indique si les premières offres affichées sont celles dont les scores sont les plus élevés ou les moins élevés.
<code>model_reliability</code>	<i>booléen</i>	Option Prendre en compte la fiabilité du modèle dans l'onglet Paramètres.

Propriétés du noeud *applysequence*

Les noeuds de modélisation Séquence peuvent être utilisés pour générer un nugget de modèle Séquence. Le nom de génération de script de ce nugget de modèle est *applysequence*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud Séquence», à la page 172.

Propriétés du noeud *appliesvm*

Les noeuds de modélisation SVM peuvent être utilisés pour générer un nugget de modèle SVM. Le nom de génération de script de ce nugget de modèle est *appliesvm*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud svm», à la page 174.

Tableau 138. Propriétés du noeud *appliesvm*.

Propriétés du noeud <i>appliesvm</i>	Valeurs	Description de la propriété
<code>all_probabilities</code>	<i>booléen</i>	
<code>calculate_raw_propensities</code>	<i>booléen</i>	
<code>calculate_adjusted_propensities</code>	<i>booléen</i>	

Propriétés du noeud *applytimeseries*

Les noeuds de modélisation Séries temporelles peuvent être utilisés pour générer un nugget de modèle Séries temporelles. Le nom de génération de script de ce nugget de modèle est *applytimeseries*. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud timeseries», à la page 174.

Tableau 139. Propriétés du noeud *applytimeseries*.

Propriétés du noeud <i>applytimeseries</i>	Valeurs	Description de la propriété
<code>calculate_conf</code>	<i>booléen</i>	
<code>calculate_residuals</code>	<i>booléen</i>	

Propriétés du noeud *applytwostep*

Les noeuds de modélisation TwoStep peuvent être utilisés pour générer un nugget de modèle TwoStep. Le nom de génération de script de ce nugget de modèle est *applytwostep*. Aucune autre propriété n'existe pour ce nugget de modèle. Pour plus d'informations à propos de la génération de script pour le noeud de modélisation lui-même, voir la rubrique «Propriétés du noeud twostep», à la page 176.

Chapitre 15. Propriétés du nœud de modélisation SGBD

IBM SPSS Modeler prend en charge l'intégration des outils d'exploration de données et de modélisation disponibles auprès des fournisseurs de base de données, notamment Microsoft SQL Server Analysis Services, Oracle Data Mining, IBM DB2 InfoSphere Warehouse et IBM Netezza Analytics. Vous pouvez créer et déterminer le score des modèles à l'aide d'algorithmes natifs de base de données, qui proviennent tous de l'application IBM SPSS Modeler. Les modèles de base de données peuvent également être créés et manipulés via des scripts à l'aide des propriétés décrites dans cette section.

Propriétés du nœud pour la modélisation Microsoft

Propriétés des nœuds de modélisation Microsoft

Propriétés communes

Les propriétés suivantes sont communes aux nœuds de modélisation de base de données Microsoft.

Tableau 140. Propriétés communes des nœuds Microsoft.

Propriétés communes des nœuds Microsoft	Valeurs	Description de la propriété
analysis_database_name	chaîne	Nom de la base de données Analysis Services.
analysis_server_name	chaîne	Nom de l'hôte Analysis Services.
use_transactional_data	booléen	Spécifie si les données d'entrée sont au format tabulaire ou transactionnel.
inputs	[champ champ champ]	Champs d'entrée pour les données tabulaires.
target	champ	Champ prédit (ne s'applique pas aux nœuds Classification non supervisée MS ou Classification de séquences).
unique_field	champ	Champ-clé.
msas_parameters	structurée	Paramètres d'algorithme. Pour plus d'informations, voir la rubrique «Paramètres d'algorithme», à la page 190.
with_drillthrough	booléen	Option Avec extraction

Arbre de décision MS

Aucune propriété particulière n'est définie pour les nœuds du type `mstree`. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Classification non supervisée MS

Aucune propriété particulière n'est définie pour les nœuds du type `mscluster`. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Règles d'association MS

Les propriétés particulières suivantes sont disponibles pour les nœuds du type `msassoc` :

Tableau 141. Propriétés du noeud *msassoc*.

Propriétés du noeud <i>msassoc</i>	Valeurs	Description de la propriété
<i>id_field</i>	<i>champ</i>	Identifie chaque transaction dans les données.
<i>trans_inputs</i>	[<i>champ champ champ</i>]	Champs d'entrée pour les données transactionnelles.
<i>transactional_target</i>	<i>champ</i>	Champ variable indépendante (données transactionnelles).

MS Naive Bayes

Aucune propriété particulière n'est définie pour les noeuds du type *msbayes*. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Régression linéaire MS

Aucune propriété particulière n'est définie pour les noeuds du type *msregression*. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Réseau neuronal MS

Aucune propriété particulière n'est définie pour les noeuds du type *msneuralnetwork*. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Régression logistique MS

Aucune propriété particulière n'est définie pour les noeuds du type *mslogistic*. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Séries temporelles MS

Aucune propriété particulière n'est définie pour les noeuds du type *mstimeseries*. Reportez-vous aux propriétés Microsoft communes au début de cette section.

Mise en cluster de séquences MS

Les propriétés particulières suivantes sont disponibles pour les noeuds du type *mssequencecluster* :

Tableau 142. Propriétés du noeud *mssequencecluster*.

Propriétés du noeud <i>mssequencecluster</i>	Valeurs	Description de la propriété
<i>id_field</i>	<i>champ</i>	Identifie chaque transaction dans les données.
<i>input_fields</i>	[<i>champ champ champ</i>]	Champs d'entrée pour les données transactionnelles.
<i>sequence_field</i>	<i>champ</i>	Identificateur de séquence.
<i>target_field</i>	<i>champ</i>	Champ variable indépendante (données tabulaires).

Paramètres d'algorithme

Chaque type de modèle de base de données Microsoft a des paramètres spécifiques pouvant être définis à l'aide de la propriété *msas_parameters*.

Ces paramètres sont issus de SQL Server. Pour visualiser les paramètres pertinents de chaque noeud, procédez comme suit :

1. Placez un noeud source de base de données dans l'espace de travail.

2. Ouvrez le noeud source de base de données.
3. Sélectionnez une source valide dans la liste déroulante **Source de données**.
4. Sélectionnez une table valide dans la liste **Nom de la table**.
5. Cliquez sur **OK** pour fermer le noeud source de base de données.
6. Reliez le noeud de modélisation de base de données Microsoft dont vous voulez répertorier les propriétés.
7. Ouvrez le noeud de modélisation de base de données.
8. Cliquez sur l'onglet **Expert**.

Les propriétés `msas_parameters` disponibles pour ce noeud apparaissent.

Propriétés du nugget de modèle Microsoft

Les propriétés suivantes s'appliquent aux nuggets de modèle créés à l'aide des noeuds de modélisation de la base de données Microsoft.

Arbre de décision MS

Tableau 143. Propriétés de l'arbre de décision MS.

Propriétés du noeud <code>appliedmstree</code>	Valeurs	Description
<code>analysis_database_name</code>	<i>chaîne</i>	Ce noeud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
<code>analysis_server_name</code>	<i>chaîne</i>	Nom de l'hôte du serveur Analysis.
<code>datasource</code>	<i>chaîne</i>	Nom de la source de données (DSN) ODBC du serveur SQL.
<code>sql_generate</code>	<i>booléen</i>	Active la génération SQL.

Régression linéaire MS

Tableau 144. Propriétés de la régression linéaire MS.

Propriétés du noeud <code>appliedmsregression</code>	Valeurs	Description
<code>analysis_database_name</code>	<i>chaîne</i>	Ce noeud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
<code>analysis_server_name</code>	<i>chaîne</i>	Nom de l'hôte du serveur Analysis.

Réseau neuronal MS

Tableau 145. Propriétés du réseau neuronal MS.

Propriétés du noeud <code>appliedmsneuralnetwork</code>	Valeurs	Description
<code>analysis_database_name</code>	<i>chaîne</i>	Ce noeud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
<code>analysis_server_name</code>	<i>chaîne</i>	Nom de l'hôte du serveur Analysis.

Régression logistique MS

Tableau 146. Propriétés de la régression logistique MS.

Propriétés du noeud aplymslogistic	Valeurs	Description
analysis_database_name	chaîne	Ce noeud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
analysis_server_name	chaîne	Nom de l'hôte du serveur Analysis.

Séries temporelles MS

Tableau 147. Propriétés des séries temporelles MS.

Propriétés du noeud aplymstimeseries	Valeurs	Description
analysis_database_name	chaîne	Ce noeud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
analysis_server_name	chaîne	Nom de l'hôte du serveur Analysis.
start_from	new_prediction historical_prediction	Spécifie s'il faut faire des prévisions futures ou des prévisions historiques.
new_step	nombre	Définit la période de commencement des prévisions futures.
historical_step	nombre	Définit la période de commencement des prévisions historiques.
end_step	nombre	Définit la période de fin des prévisions.

Mise en cluster de séquences MS

Tableau 148. Propriétés de mise en cluster de séquences MS.

Propriétés du noeud aplymssequencecluster	Valeurs	Description
analysis_database_name	chaîne	Ce noeud peut être directement évalué dans un flux. Cette propriété est utilisée pour identifier le nom de la base de données Analysis Services.
analysis_server_name	chaîne	Nom de l'hôte du serveur Analysis.

Propriétés du noeud pour la modélisation Oracle

Propriétés du noeud de modélisation Oracle

Les propriétés suivantes sont communes aux noeuds de modélisation de base de données Oracle.

Tableau 149. Propriétés communes aux noeuds Oracle.

Propriétés communes des noeuds Oracle	Valeurs	Description de la propriété
target	champ	
inputs	Liste des champs	
partition	champ	Champ utilisé pour partitionner les données en échantillons distincts pour les étapes d'apprentissage, de test et de validation de la création d'un modèle.
datasource		
username		
password		
epassword		
use_model_name	booléen	
model_name	chaîne	Nom personnalisé du nouveau modèle.
use_partitioned_data	booléen	Si un champ de partition est défini, cette option assure que seules les données provenant de la partition d'apprentissage sont utilisées pour générer le modèle.
unique_field	champ	
auto_data_prep	booléen	Active ou désactive la fonction de préparation de données automatique d'Oracle (Bases de données 11g uniquement).
costs	structurée	Propriétés structurées.
mode	Simple Expert	Certaines propriétés sont ignorées si ce paramètre est défini sur Simple, comme indiqué dans les propriétés de noeud individuelles.
use_prediction_probability	booléen	
prediction_probability	chaîne	
use_prediction_set	booléen	

Oracle Naive Bayes

Les propriétés suivantes sont disponibles pour les noeuds du type oranb.

Tableau 150. Propriétés du noeud oranb.

Propriétés du noeud oranb	Valeurs	Description de la propriété
singleton_threshold	nombre	0,0-1,0.*
pairwise_threshold	nombre	0,0-1,0.*
priors	Data Equal Custom	
custom_priors	structurée	Propriétés structurées.

* Propriété ignorée si mode est défini sur Simple.

Oracle Adaptive Bayes

Les propriétés suivantes sont disponibles pour les noeuds du type oraabn.

Tableau 151. Propriétés du noeud oraabn.

Propriétés du noeud oraabn	Valeurs	Description de la propriété
model_type	SingleFeature MultiFeature NaiveBayes	
use_execution_time_limit	booléen	*
execution_time_limit	entier	La valeur doit être supérieure à 0.*
max_naive_bayes_predictors	entier	La valeur doit être supérieure à 0.*
max_predictors	entier	La valeur doit être supérieure à 0.*
priors	Data Equal Custom	
custom_priors	structurée	Propriétés structurées.

* Propriété ignorée si mode est défini sur Simple.

Oracle Support Vector Machines

Les propriétés suivantes sont disponibles pour les noeuds du type orasvm.

Tableau 152. Propriétés du noeud orasvm.

Propriétés du noeud orasvm	Valeurs	Description de la propriété
active_learning	Enable Disable	
kernel_function	Linear Gaussian System	
normalization_method	zscore minmax none	
kernel_cache_size	entier	Noyau gaussien uniquement. La valeur doit être supérieure à 0.*
convergence_tolerance	nombre	La valeur doit être supérieure à 0.*
use_standard_deviation	booléen	Noyau gaussien uniquement.*
standard_deviation	nombre	La valeur doit être supérieure à 0.*
use_epsilon	booléen	Modèles de régression uniquement.*
epsilon	nombre	La valeur doit être supérieure à 0.*
use_complexity_factor	booléen	*
complexity_factor	nombre	*
use_outlier_rate	booléen	Variante à classe unique seulement.*
outlier_rate	nombre	Variante à classe unique seulement. 0,0-1,0.*

Tableau 152. Propriétés du noeud orasvm (suite).

Propriétés du noeud orasvm	Valeurs	Description de la propriété
weights	Data Equal Custom	
custom_weights	<i>structurée</i>	Propriétés structurées.

* Propriété ignorée si mode est défini sur Simple.

Modèles linéaires généralisés d'Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type oraglm.

Tableau 153. Propriétés du noeud oraglm.

Propriétés du noeud oraglm	Valeurs	Description de la propriété
normalization_method	zscore minmax none	
missing_value_handling	ReplaceWithMean UseCompleteRecords	
use_row_weights	<i>booléen</i>	*
row_weights_field	<i>champ</i>	*
save_row_diagnostics	<i>booléen</i>	*
row_diagnostics_table	<i>chaîne</i>	*
coefficient_confidence	<i>nombre</i>	*
use_reference_category	<i>booléen</i>	*
reference_category	<i>chaîne</i>	*
ridge_regression	Auto Off On	*
parameter_value	<i>nombre</i>	*
vif_for_ridge	<i>booléen</i>	*

* Propriété ignorée si mode est défini sur Simple.

Arbre décision Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type oradecisiontree.

Tableau 154. Propriétés du noeud oradecisiontree.

Propriétés du noeud oradecisiontree	Valeurs	Description de la propriété
use_costs	<i>booléen</i>	
impurity_metric	Entropy Gini	
term_max_depth	<i>entier</i>	2–20.*
term_minpct_node	<i>nombre</i>	0,0–10,0.*
term_minpct_split	<i>nombre</i>	0,0–20,0.*

Tableau 154. Propriétés du noeud oradecisiontree (suite).

Propriétés du noeud oradecisiontree	Valeurs	Description de la propriété
term_minrec_node	entier	La valeur doit être supérieure à 0.*
term_minrec_split	entier	La valeur doit être supérieure à 0.*
display_rule_ids	booléen	*

* Propriété ignorée si mode est défini sur Simple.

O-Cluster Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type oraoccluster.

Tableau 155. Propriétés du noeud oraoccluster.

Propriétés du noeud oraoccluster	Valeurs	Description de la propriété
max_num_clusters	entier	La valeur doit être supérieure à 0.
max_buffer	entier	La valeur doit être supérieure à 0.*
sensitivity	nombre	0,0–1,0.*

* Propriété ignorée si mode est défini sur Simple.

KMeans Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type orakmeans.

Tableau 156. Propriétés du noeud orakmeans.

Propriétés du noeud orakmeans	Valeurs	Description de la propriété
num_clusters	entier	La valeur doit être supérieure à 0.
normalization_method	zscore minmax none	
distance_function	Euclidean Cosine	
iterations	entier	0–20.*
conv_tolerance	nombre	0,0–0,5.*
split_criterion	Variance Size	La valeur par défaut est Variance.*
num_bins	entier	La valeur doit être supérieure à 0.*
block_growth	entier	1–5.*
min_pct_attr_support	nombre	0,0–1,0.*

* Propriété ignorée si mode est défini sur Simple.

NMF Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type oranmf.

Tableau 157. Propriétés du noeud oranmf.

Propriétés du noeud oranmf	Valeurs	Description de la propriété
normalization_method	minmax none	
use_num_features	booléen	*
num_features	entier	0-1. La valeur par défaut est estimée à partir des données par l'algorithme.*
random_seed	nombre	*
num_iterations	entier	0-500.*
conv_tolerance	nombre	0,0-0,5.*
display_all_features	booléen	*

* Propriété ignorée si mode est défini sur Simple.

Apriori Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type oraapriori.

Tableau 158. Propriétés du noeud oraapriori.

Propriétés du noeud oraapriori	Valeurs	Description de la propriété
content_field	champ	
id_field	champ	
max_rule_length	entier	2-20.
min_confidence	nombre	0,0-1,0.
min_support	nombre	0,0-1,0.
use_transactional_data	booléen	

Description de longueur minimale d'Oracle (MDL)

Aucune propriété particulière n'est définie pour les noeuds du type oramd1. Reportez-vous aux propriétés Oracle courantes au début de cette section.

Importance de l'attribut d'Oracle (IA)

Les propriétés suivantes sont disponibles pour les noeuds du type oraai.

Tableau 159. Propriétés du noeud oraai.

Propriétés du noeud oraai	Valeurs	Description de la propriété
custom_fields	booléen	Si la valeur est définie sur true (vrai), elle permet de spécifier les champs cible, entrée et d'autres champs pour le noeud actuel. Si elle est définie sur false (faux), les paramètres actuels provenant d'un noeud typeen amont sont utilisés.
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	booléen	Quand la propriété selection_mode est définie sur ImportanceLevel, indique si les champs importants doivent être sélectionnés.

Tableau 159. Propriétés du noeud oraii (suite).

Propriétés du noeud oraii	Valeurs	Description de la propriété
important_label	chaîne	Indique le libellé du classement « significatif ».
select_marginal	booléen	Quand la propriété selection_mode est définie sur ImportanceLevel, indique si les champs marginaux doivent être sélectionnés.
marginal_label	chaîne	Indique le libellé du classement « marginal ».
important_above	nombre	0,0–1,0.
select_unimportant	booléen	Quand la propriété selection_mode est définie sur ImportanceLevel, indique si les champs non significatifs doivent être sélectionnés.
unimportant_label	chaîne	Indique le libellé du classement « non significatif ».
unimportant_below	nombre	0,0–1,0.
importance_value	nombre	Quand la propriété selection_mode est définie sur ImportanceValue, indique la valeur de césure à utiliser. Accepte des valeurs de 0 à 100.
top_n	nombre	Quand la propriété selection_mode est définie sur TopN, indique la valeur de césure à utiliser. Accepte des valeurs de 0 à 1000.

Propriétés du nugget de modèle Oracle

Les propriétés suivantes s'appliquent aux nuggets de modèle créés à l'aide des modèles Oracle.

Oracle Naive Bayes

Aucune propriété particulière n'est définie pour les noeuds du type applyoranb.

Oracle Adaptive Bayes

Aucune propriété particulière n'est définie pour les noeuds du type applyoraabn.

Oracle Support Vector Machines

Aucune propriété particulière n'est définie pour les noeuds du type applyorasvm.

Arbre décision Oracle

Les propriétés suivantes sont disponibles pour les noeuds du type applyoradecisiontree.

Tableau 160. Propriétés du noeud applyoradecisiontree.

Propriétés du noeud applyoradecisiontree	Valeurs	Description de la propriété
use_costs	booléen	
display_rule_ids	booléen	

O-Cluster Oracle

Aucune propriété particulière n'est définie pour les noeuds du type applyoraocluster.

KMeans Oracle

Aucune propriété particulière n'est définie pour les noeuds du type applyorakmeans.

NMF Oracle

La propriété suivante est disponible pour les noeuds du type applyoranmf :

Tableau 161. Propriétés du noeud applyoranmf.

Propriétés du noeud applyoranmf	Valeurs	Description de la propriété
display_all_features	booléen	

Apriori Oracle

Ce nugget de modèle ne peut pas être appliqué dans la génération de script.

MDL Oracle

Ce nugget de modèle ne peut pas être appliqué dans la génération de script.

Propriétés de nœud pour la modélisation IBM DB2

Propriétés du nœud de modélisation IBM DB2

Les propriétés suivantes sont communes aux noeuds de modélisation de base de données IBM InfoSphere Warehouse (ISW).

Tableau 162. Propriétés communes du noeud ISW.

Propriétés communes du noeud ISW	Valeurs	Description de la propriété
inputs	Liste des champs	
datasource		
username		
password		
epassword		
enable_power_options	booléen	
power_options_max_memory	entier	La valeur doit être supérieure à 32.
power_options_cmdline	chaîne	
mining_data_custom_sql	chaîne	
logical_data_custom_sql	chaîne	
mining_settings_custom_sql		

Arbre décision ISW

Les propriétés suivantes sont disponibles pour les noeuds du type db2imtree.

Tableau 163. Propriétés du noeud db2imtree.

Propriétés du noeud db2imtree	Valeurs	Description de la propriété
target	champ	
perform_test_run	booléen	
use_max_tree_depth	booléen	

Tableau 163. Propriétés du noeud db2imtree (suite).

Propriétés du noeud db2imtree	Valeurs	Description de la propriété
max_tree_depth	entier	Valeur supérieure à 0.
use_maximum_purity	booléen	
maximum_purity	nombre	Nombre compris entre 0 et 100.
use_minimum_internal_cases	booléen	
minimum_internal_cases	entier	Valeur supérieure à 1.
use_costs	booléen	
costs	structurée	Propriétés structurées.

Association ISW

Les propriétés suivantes sont disponibles pour les noeuds du type Association IM DB2.

Tableau 164. Propriétés du noeud db2imassoc.

Propriétés du noeud db2imassoc	Valeurs	Description de la propriété
use_transactional_data	booléen	
id_field	champ	
content_field	champ	
data_table_layout	basic limited_length	
max_rule_size	entier	La valeur doit être supérieure à 2.
min_rule_support	nombre	0–100%
min_rule_confidence	nombre	0–100%
use_item_constraints	booléen	
item_constraints_type	Include Exclude	
use_taxonomy	booléen	
taxonomy_table_name	chaîne	Nom de la table DB2 dans laquelle stocker les détails de la taxonomie.
taxonomy_child_column_name	chaîne	Nom de la colonne enfant de la table de taxonomie. La colonne enfant comporte les noms des éléments ou des catégories.
taxonomy_parent_column_name	chaîne	Nom de la colonne parent de la table de taxonomie. La colonne parent contient les noms des catégories.
load_taxonomy_to_table	booléen	Permet de savoir si les informations sur la taxonomie stockées dans IBM SPSS Modeler doivent être envoyées à la table de taxonomie lors de la création du modèle. La table de taxonomie est supprimée si elle existe déjà. Les informations sur la taxonomie sont stockées avec le noeud de création du modèle et peuvent être modifiées via les boutons Modifier les catégories et Modifier la taxonomie .

Séquence ISW

Les propriétés suivantes sont disponibles pour les noeuds du type db2imsequence.

Tableau 165. Propriétés du noeud db2imsequence.

Propriétés du noeud db2imsequence	Valeurs	Description de la propriété
id_field	champ	
group_field	champ	
content_field	champ	
max_rule_size	entier	La valeur doit être supérieure à 2.
min_rule_support	nombre	0–100%
min_rule_confidence	nombre	0–100%
use_item_constraints	booléen	
item_constraints_type	Include Exclude	
use_taxonomy	booléen	
taxonomy_table_name	chaîne	Nom de la table DB2 dans laquelle stocker les détails de la taxonomie.
taxonomy_child_column_name	chaîne	Nom de la colonne enfant de la table de taxonomie. La colonne enfant comporte les noms des éléments ou des catégories.
taxonomy_parent_column_name	chaîne	Nom de la colonne parent de la table de taxonomie. La colonne parent contient les noms des catégories.
load_taxonomy_to_table	booléen	Permet de savoir si les informations sur la taxonomie stockées dans IBM SPSS Modeler doivent être envoyées à la table de taxonomie lors de la création du modèle. La table de taxonomie est supprimée si elle existe déjà. Les informations sur la taxonomie sont stockées avec le noeud de création du modèle et peuvent être modifiées via les boutons Modifier les catégories et Modifier la taxonomie .

Régression ISW

Les propriétés suivantes sont disponibles pour les noeuds du type Régression IM DB2.

Tableau 166. Propriétés du noeud db2imreg.

Propriétés du noeud db2imreg	Valeurs	Description de la propriété
target	champ	
regression_method	transform linear polynomial rbf	Consultez le tableau suivant pour les propriétés s'appliquant uniquement si regression_method est définie sur rbf.
perform_test_run	champ	
limit_rsquared_value	booléen	
max_rsquared_value	nombre	Valeur comprise entre 0,0 et 1,0.
use_execution_time_limit	booléen	
execution_time_limit_mins	entier	Valeur supérieure à 0.
use_max_degree_polynomial	booléen	
max_degree_polynomial	entier	
use_intercept	booléen	
use_auto_feature_selection_method	booléen	

Tableau 166. Propriétés du noeud db2imreg (suite).

Propriétés du noeud db2imreg	Valeurs	Description de la propriété
auto_feature_selection_method	normal adjusted	
use_min_significance_level	booléen	
min_significance_level	nombre	
use_min_significance_level	booléen	

Les propriétés suivantes ne s'appliquent que si regression_method est définie sur rbf.

Tableau 167. Propriétés du noeud db2imreg si regression_method est définie sur rbf.

Propriétés du noeud db2imreg	Valeurs	Description de la propriété
use_output_sample_size	booléen	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
output_sample_size	entier	La valeur par défaut est 2. Le minimum est 1.
use_input_sample_size	booléen	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
input_sample_size	entier	La valeur par défaut est 2. Le minimum est 1.
use_max_num_centers	booléen	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
max_num_centers	entier	La valeur par défaut est 20. Le minimum est 1.
use_min_region_size	booléen	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
min_region_size	entier	La valeur par défaut est 15. Le minimum est 1.
use_max_data_passes	booléen	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
max_data_passes	entier	La valeur par défaut est 5. Le minimum est 2.
use_min_data_passes	booléen	Si vrai, définissez automatiquement la valeur sur la valeur par défaut.
min_data_passes	entier	La valeur par défaut est 5. Le minimum est 2.

Classification ISW

Les propriétés suivantes sont disponibles pour les noeuds du type Classification IM DB2.

Tableau 168. Propriétés du noeud db2imcluster.

Propriétés du noeud db2imcluster	Valeurs	Description de la propriété
cluster_method	demographic kohonen birch	
kohonen_num_rows	entier	
kohonen_num_columns	entier	
kohonen_passes	entier	
use_num_passes_limit	booléen	
use_num_clusters_limit	booléen	
max_num_clusters	entier	Valeur supérieure à 1.
birch_dist_measure	log_likelihood euclidean	La valeur par défaut est log_likelihood.
birch_num_cfleaves	entier	La valeur par défaut est 1000.
birch_num_refine_passes	entier	La valeur par défaut est 3, la valeur minimum est 1.
use_execution_time_limit	booléen	
execution_time_limit_mins	entier	Valeur supérieure à 0.
min_data_percentage	nombre	0-100%
use_similarity_threshold	booléen	
similarity_threshold	nombre	Valeur comprise entre 0,0 et 1,0.

Naive Bayes ISW

Les propriétés suivantes sont disponibles pour les noeuds du type db2imnbs.

Tableau 169. Propriétés du noeud db2imnb.

Propriétés du noeud db2imnb	Valeurs	Description de la propriété
perform_test_run	booléen	
probability_threshold	nombre	La valeur par défaut est 0.001. La valeur minimum est de 0 ; la valeur maximum est de 1,000.
use_costs	booléen	
costs	structurée	Propriétés structurées.

Régression logistique ISW

Les propriétés suivantes sont disponibles pour les noeuds du type db2imlog.

Tableau 170. Propriétés du noeud db2imlog.

Propriétés du noeud db2imlog	Valeurs	Description de la propriété
perform_test_run	booléen	
use_costs	booléen	
costs	structurée	Propriétés structurées.

Séries temporelles ISW

Remarque : le paramètre des champs d'entrée n'est pas utilisé pour ce noeud. Si le paramètre des champs d'entrée se trouve dans le script, un avertissement apparaît et signale que le noeud contient les champs entrants *heure* et *cibles* mais pas de champs d'entrée.

Les propriétés suivantes sont disponibles pour les noeuds du type `db2imtimeseries`.

Tableau 171. Propriétés du noeud `db2imtimeseries`.

Propriétés du noeud <code>db2imtimeseries</code>	Valeurs	Description de la propriété
<code>time</code>	<i>champ</i>	Entier, heure ou date autorisé.
<code>targets</code>	<i>liste des champs</i>	
<code>forecasting_algorithm</code>	arima exponential_smoothing seasonal_trend_decomposition	
<code>forecasting_end_time</code>	auto integer date time	
<code>use_records_all</code>	<i>booléen</i>	Si faux, <code>use_records_start</code> et <code>use_records_end</code> doivent être définis.
<code>use_records_start</code>	<i>entier / heure / date</i>	Dépend du type du champ Temps
<code>use_records_end</code>	<i>entier / heure / date</i>	Dépend du type du champ Temps
<code>interpolation_method</code>	none linear exponential_splines cubic_splines	

Propriétés du nugget de modèle IBM DB2

Les propriétés suivantes s'appliquent aux nuggets de modèle créés à l'aide des modèles IBM DB2 ISW.

Arbre décision ISW

Aucune propriété particulière n'est définie pour les noeuds du type `applydb2imtree`.

Association ISW

Ce nugget de modèle ne peut pas être appliqué dans la génération de script.

Séquence ISW

Ce nugget de modèle ne peut pas être appliqué dans la génération de script.

Régression ISW

Aucune propriété particulière n'est définie pour les noeuds du type `applydb2imreg`.

Classification ISW

Aucune propriété particulière n'est définie pour les noeuds du type `applydb2imcluster`.

Naive Bayes ISW

Aucune propriété particulière n'est définie pour les noeuds du type `applydb2imnb`.

Régression logistique ISW

Aucune propriété particulière n'est définie pour les noeuds du type `applydb2imlog`.

Séries temporelles ISW

Ce nugget de modèle ne peut pas être appliqué dans la génération de script.

Propriétés du noeud pour la modélisation IBM Netezza Analytics

Propriétés des noeuds de modélisation Netezza

Les propriétés suivantes sont communes aux noeuds de modélisation de base de données IBM Netezza.

Tableau 172. Propriétés communes aux noeuds Netezza.

Propriétés communes des noeuds Netezza	Valeurs	Description de la propriété
<code>custom_fields</code>	<i>booléen</i>	Si la valeur est définie sur <code>true</code> (vrai), elle permet de spécifier les champs cible, entrée et d'autres champs pour le noeud actuel. Si elle est définie sur <code>false</code> (faux), les paramètres actuels provenant d'un noeud <code>typeen</code> amont sont utilisés.
<code>inputs</code>	<i>[champ1 ... champN]</i>	Champs d'entrée ou prédicteur utilisés par le modèle.
<code>target</code>	<i>champ</i>	Champ cible (continu ou qualitatif).
<code>record_id</code>	<i>champ</i>	Champ à utiliser comme identifiant d'enregistrement unique.
<code>use_upstream_connection</code>	<i>booléen</i>	Si <code>vrai</code> (valeur par défaut), les détails de connexion spécifiés dans un noeud en amont. Non utilisé si <code>move_data_to_connection</code> est spécifié.
<code>move_data_connection</code>	<i>booléen</i>	Si <code>true</code> (vrai), déplace les données vers la base de données mentionnée par <code>connection</code> . Non utilisé si <code>use_upstream_connection</code> est spécifié.
<code>connection</code>	<i>structurée</i>	La chaîne de connexion à la base de données Netezza où le modèle est stocké. Propriété structurée utilisant la syntaxe : <code>['odbc' '<dsn>' '<username>' '<psw>' '<catname>' '<conn_attribs>' {true false}]</code> où : <dsn> est le nom de la source de données <username> et <psw> sont le nom d'utilisateur et le mot de passe de la base de données <catname> est le nom du catalogue <conn_attribs> sont les attributs de connexion <code>true</code> <code>false</code> indique si un mot de passe est requis.
<code>table_name</code>	<i>chaîne</i>	Le nom de la table de la base de données où le modèle est stocké.
<code>use_model_name</code>	<i>booléen</i>	Si <code>true</code> (vrai), utilise le nom spécifié par <code>model_name</code> comme nom du modèle ; sinon, le nom du modèle est créé par le système.
<code>model_name</code>	<i>chaîne</i>	Nom personnalisé du nouveau modèle.
<code>include_input_fields</code>	<i>booléen</i>	Si <code>true</code> (vrai), passe tous les champs d'entrée en aval ; sinon, passe uniquement <code>record_id</code> et les champs générés par le modèle.

Arbre décision Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type netezzadectree.

Tableau 173. Propriétés du noeud netezzadectree.

Propriétés du noeud netezzadectree	Valeurs	Description de la propriété
impurity_measure	Entropy Gini	La mesure d'impureté, utilisée pour évaluer le meilleur endroit où diviser l'arbre.
max_tree_depth	entier	Nombre de niveaux maximum que l'arbre peut atteindre. La valeur par défaut est 62 (le nombre maximum possible).
min_improvement_splits	nombre	Amélioration minimum de l'impureté pour que la division ait lieu. La valeur par défaut est de 0,01.
min_instances_split	entier	Nombre minimum d'enregistrements non divisés restants avant que la division n'ait lieu. La valeur par défaut est 2 (le nombre minimum possible).
weights	structurée	Pondérations relatives des classes. Propriétés structurées. La valeur par défaut est une pondération de 1 pour toutes les classes.
pruning_measure	Acc wAcc	La valeur par défaut est Acc (exactitude). wAcc (exactitude pondérée) tient compte des pondérations de classe tout en appliquant l'élagage.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	La valeur par défaut est d'utiliser allTrainingData pour estimer l'exactitude du modèle. Utilisez partitionTrainingData pour spécifier un pourcentage de données d'apprentissage à utiliser ou useOtherTable pour utiliser un jeu de données d'apprentissage d'une table de base de données spécifique.
perc_training_data	nombre	Si prune_tree_options est définie sur partitionTrainingData, spécifie le pourcentage de données à utiliser pour l'apprentissage.
prune_seed	entier	Valeur de départ aléatoire à utiliser pour dupliquer les résultats d'analyse quand prune_tree_options est définie sur partitionTrainingData ; la valeur par défaut est 1.
pruning_table	chaîne	Nom de la table d'un jeu de données d'élagage distinct pour estimer l'exactitude du modèle.

Tableau 173. Propriétés du noeud netezzadectree (suite).

Propriétés du noeud netezzadectree	Valeurs	Description de la propriété
compute_probabilities	booléen	Si vrai, produit un champ de niveau de confiance (probabilité) ainsi que le champ de prévision.

k moyenne Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type netezzakmeans.

Tableau 174. Propriétés du noeud netezzakmeans.

Propriétés du noeud netezzakmeans	Valeurs	Description de la propriété
distance_measure	Euclidean Manhattan Canberra maximum	Méthode à utiliser pour mesurer la distance entre les points de données.
num_clusters	entier	Le nombre de clusters à créer ; la valeur par défaut est 3.
max_iterations	entier	Le nombre d'itérations d'algorithme après lequel arrêter l'apprentissage du modèle ; la valeur par défaut est 5.
rand_seed	entier	Valeur de départ aléatoire à utiliser pour dupliquer les résultats d'analyse ; la valeur par défaut est 12345.

Bayes Net Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type netezzabayes.

Tableau 175. Propriétés du noeud netezzabayes.

Propriétés du noeud netezzabayes	Valeurs	Description de la propriété
base_index	entier	Identifiant numérique attribué au premier champ d'entrée pour la gestion interne ; la valeur par défaut est 777.
sample_size	entier	Taille de l'échantillon à prélever si le nombre d'attributs est très grand ; la valeur par défaut est 10 000.
display_additional_information	booléen	Si vrai, affiche les informations de progression supplémentaires dans une boîte de dialogue.
type_of_prediction	best neighbors nn-neighbors	Type d'algorithme de prévision à utiliser : best (meilleur ; voisin le plus corrélé), neighbors (voisins ; prévision pondérée des voisins) ou nn-neighbors (voisins nn ; voisins non nuls).

Naive Bayes Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type netezzanaiwebayes.

Tableau 176. Propriétés du noeud netezzanaiwebayes.

Propriétés du noeud netezzanaiwebayes	Valeurs	Description de la propriété
compute_probabilities	booléen	Si vrai, produit un champ de niveau de confiance (probabilité) ainsi que le champ de prévision.

Tableau 176. Propriétés du noeud netezzanaivebayes (suite).

Propriétés du noeud netezzanaivebayes	Valeurs	Description de la propriété
use_m_estimation	booléen	Si vrai, utilise la technique de m-estimation pour éviter les probabilités zéro pendant l'estimation.

KNN Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type netezzaknn.

Tableau 177. Propriétés du noeud netezzaknn.

Propriétés du noeud netezzaknn	Valeurs	Description de la propriété
weights	structurée	Propriété structurée utilisée pour attribuer des pondérations aux classes individuelles.
distance_measure	Euclidean Manhattan Canberra Maximum	Méthode à utiliser pour mesurer la distance entre les points de données.
num_nearest_neighbors	entier	Nombre de voisins les plus proches pour un cas particulier ; la valeur par défaut est 3.
standardize_measurements	booléen	Si vrai, standardise les mesures des champs d'entrée continus avant de calculer les valeurs de distance.
use_coresets	booléen	Si vrai, utilise l'échantillonnage d'ensembles principaux pour accélérer le calcul de grands jeux de données.

Classification par division Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type netezzacluster.

Tableau 178. Propriétés du noeud netezzacluster.

Propriétés du noeud netezzacluster	Valeurs	Description de la propriété
distance_measure	Euclidean Manhattan Canberra Maximum	Méthode à utiliser pour mesurer la distance entre les points de données.
max_iterations	entier	Nombre maximum d'itérations d'algorithme à effectuer avant l'arrêt de l'apprentissage du modèle ; la valeur par défaut est 5.
max_tree_depth	entier	Nombre maximum de niveaux dans lesquels les jeux de données peuvent être divisés ; la valeur par défaut est 3.
rand_seed	entier	Valeur de départ aléatoire utilisée pour dupliquer les analyses ; la valeur par défaut est 12345.
min_instances_split	entier	Nombre minimum d'enregistrements pouvant être divisés, la valeur par défaut est 5.
level	entier	Niveau hiérarchique selon lequel les enregistrements sont évalués ; la valeur par défaut est -1.

ACP Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type netezzapca.

Tableau 179. Propriétés du noeud netezzapca.

Propriétés du noeud netezzapca	Valeurs	Description de la propriété
center_data	booléen	Si vrai (par défaut), effectue un centrage des données (également nommé « soustraction moyenne ») avant l'analyse.
perform_data_scaling	booléen	Si vrai, effectue la mise à l'échelle des données avant l'analyse. Ceci peut rendre l'analyse moins arbitraire quand différentes variables sont mesurées dans différentes unités.
force_eigensolve	booléen	Si vrai, utilise une méthode moins précise mais plus rapide pour trouver les composantes principales.
pc_number	entier	Nombre de composantes principales auquel le jeu de données doit être réduit ; la valeur par défaut est 1.

Arbre de régression Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type netezzaregtree.

Tableau 180. Propriétés du noeud netezzaregtree.

Propriétés du noeud netezzaregtree	Valeurs	Description de la propriété
max_tree_depth	entier	Nombre maximum de niveaux auxquels l'arbre peut grandir sous le noeud racine ; la valeur par défaut est 10.
split_evaluation_measure	Variance	Mesure d'impureté de la classe, utilisée pour évaluer le meilleur endroit pour diviser l'arbre ; la valeur par défaut (et actuellement la seule option) est Variance.
min_improvement_splits	nombre	Quantité minimale pour réduire l'impureté avant de créer une nouvelle division dans l'arbre.
min_instances_split	entier	Le nombre minimal d'enregistrements pouvant être divisés.
pruning_measure	mse r2 pearson spearman	Méthode à utiliser pour l'élagage.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	La valeur par défaut est d'utiliser allTrainingData pour estimer l'exactitude du modèle. Utilisez partitionTrainingData pour spécifier un pourcentage de données d'apprentissage à utiliser ou useOtherTable pour utiliser un jeu de données d'apprentissage d'une table de base de données spécifique.
perc_training_data	nombre	Si prune_tree_options est définie sur PercTrainingData, spécifie le pourcentage de données à utiliser pour l'apprentissage.

Tableau 180. Propriétés du noeud netezzaregtree (suite).

Propriétés du noeud netezzaregtree	Valeurs	Description de la propriété
prune_seed	entier	Valeur de départ aléatoire à utiliser pour dupliquer les résultats d'analyse quand prune_tree_options est définie sur PercTrainingData ; la valeur par défaut est 1.
pruning_table	chaîne	Nom de la table d'un le jeu de données d'élagage distinct pour estimer l'exactitude du modèle.
compute_probabilities	booléen	Si vrai, spécifie que les variances des classes attribuées doivent être incluses dans la sortie.

Régression linéaire Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type netezzalineression.

Tableau 181. Propriétés du noeud netezzalineression.

Propriétés du noeud netezzalineression	Valeurs	Description de la propriété
use_svd	booléen	Si vrai, utilise la matrice de décomposition en valeurs singulières à la place de la matrice d'origine, pour améliorer la vitesse et l'exactitude numérique.
include_intercept	booléen	Si vrai (défaut), augmente l'exactitude générale de la solution.
calculate_model_diagnostics	booléen	Si vrai, calcule le diagnostic sur le modèle.

Séries temporelles Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type netezzatimeseries.

Tableau 182. Propriétés du noeud netezzatimeseries.

Propriétés du noeud netezzatimeseries	Valeurs	Description de la propriété
time_points	champ	Champ d'entrée contenant les valeurs de date et d'heure des séries temporelles.
time_series_ids	champ	Un champ d'entrée contenant des ID de séries temporelles à utiliser si l'entrée contient plusieurs séries temporelles.
model_table	champ	Nom de la table de la base de données dans laquelle le modèle de séries temporelles Netezza sera stocké.
description_table	champ	Nom de la table d'entrée qui contient les noms et descriptions des séries temporelles.

Tableau 182. Propriétés du nœud *netezzatimeseries* (suite).

Propriétés du nœud <i>netezzatimeseries</i>	Valeurs	Description de la propriété
<code>seasonal_adjustment_table</code>	<i>champ</i>	Nom de la table de sortie dans laquelle les valeurs ajustées de manière saisonnière calculées par des algorithmes de lissage exponentiel ou de décomposition des tendances saisonnières seront stockées.
<code>algorithm_name</code>	SpectralAnalysis ou spectral ExponentialSmoothing ou esmoothing ARIMA SeasonalTrendDecomposition ou std	Algorithme à utiliser pour la modélisation des séries temporelles.
<code>trend_name</code>	N A DA M DM	Type de tendance du lissage exponentiel : N - none (aucun) A - additive (additif) DA - damped additive (additif amorti) M - multiplicative (multiplicatif) DM - damped multiplicative (Multiplicatif amorti)
<code>seasonality_type</code>	N A M	Type de saisonnalité pour le lissage exponentiel : N - none (aucun) A - additive (additif) M - multiplicative (multiplicatif)
<code>interpolation_method</code>	linear cubicspline exponentialspline	Méthode d'interpolation à utiliser.
<code>timerange_setting</code>	ET SP	Paramètre pour l'intervalle de temps à utiliser : SD - system-determined (déterminé par le système) (utilise toute la plage de séries temporelles) SP - spécifié par l'utilisateur via <code>earliest_time</code> et <code>latest_time</code>
<code>earliest_time</code>	<i>Date</i>	Heures de début et de fin, si <code>timerange_setting</code> est SP. Format : <yyyy>-<mm>-<dd>
<code>latest_time</code>		
<code>arima_setting</code>	ET SP	Paramètre pour l'algorithme ARIMA (utilisé uniquement si <code>algorithm_name</code> est défini sur ARIMA) : SD - system-determined (déterminé par le système) SP - spécifié par l'utilisateur Si <code>arima_setting</code> = SP, utilisez les paramètres suivants pour définir les valeurs saisonnières et non saisonnières.

Tableau 182. Propriétés du noeud netezatimeseries (suite).

Propriétés du noeud netezatimeseries	Valeurs	Description de la propriété
p_symbol	less eq lesseq	ARIMA - opérateur pour les paramètres p, d, q, sp, sd et sq : less - inférieur à eq - égal à lesseq - inférieur ou égal à
d_symbol		
q_symbol		
sp_symbol		
sd_symbol		
sq_symbol		
p	entier	ARIMA - degrés d'autocorrélation non saisonniers.
q	entier	ARIMA - valeur de calcul non saisonnier.
d	entier	ARIMA - nombre d'ordres de moyenne mobile non saisonniers dans le modèle.
sp	entier	ARIMA - degrés d'autocorrélation saisonniers.
sq	entier	ARIMA - valeur de dérivation saisonnière.
sd	entier	ARIMA - nombre d'ordres de moyenne mobile saisonniers dans le modèle.
advanced_setting	ET SP	Détermine comment traiter les paramètres avancés : SD - system-determined (déterminé par le système) SP - spécifié par l'utilisateur via period, units_period et forecast_setting.
period	entier	Longueur du cycle saisonnier spécifié en association avec units_period. Ne s'applique pas à l'analyse spectrale.
units_period	ms s min h d wk q y	Unités dans lesquelles period est exprimé : ms - millisecondes s - secondes min - minutes h - heures d - jours wk - semaines q - trimestres y - années Par exemple, pour une série temporelle hebdomadaire, utilisez 1 pour period et wk pour units_period.
forecast_setting	forecasthorizon forecasttimes	Indique comment doivent être effectuées les prévisions.

Tableau 182. Propriétés du noeud *netezzatimeseries* (suite).

Propriétés du noeud <i>netezzatimeseries</i>	Valeurs	Description de la propriété
<code>forecast_horizon</code>	<i>chaîne</i>	Si <code>forecast_setting = forecasthorizon</code> , indique le point final des prévisions. Format : <yyyy>-<mm>-<dd>
<code>forecast_times</code>	[[<i>'date'</i>], <i>'date'</i>],..., <i>'date'</i>]]	Si <code>forecast_setting = forecasttimes</code> , indique les moments à utiliser pour les prévisions. Format : <yyyy>-<mm>-<dd>
<code>include_history</code>	<i>booléen</i>	Indique si les valeurs historiques doivent être incluses dans la sortie.
<code>include_interpolated_values</code>	<i>booléen</i>	Indique si les valeurs interpolées doivent être incluses dans la sortie. Ne s'applique pas si <code>include_history</code> est <i>false</i> .

Linéaire généralisé Netezza

Les propriétés suivantes sont disponibles pour les noeuds du type *netezzaglm*.

Tableau 183. Propriétés du noeud *netezzaglm*.

Propriétés du noeud <i>netezzaglm</i>	Valeurs	Description de la propriété
<code>dist_family</code>	bernoulli gaussian poisson negativebinomial wald gamma	Type de distribution ; le type par défaut est <i>bernoulli</i> .
<code>dist_params</code>	<i>nombre</i>	Valeur de paramètre de distribution à utiliser. S'applique uniquement si <code>distribution</code> est <i>Negativebinomial</i> .
<code>trials</code>	<i>entier</i>	S'applique uniquement si <code>distribution</code> est <i>Binomial</i> . Lorsque la réponse cible est un nombre d'événements se produisant dans un ensemble d'essais, le champ <code>target</code> contient le nombre d'événements et le champ <code>trials</code> contient le nombre d'essais.
<code>model_table</code>	<i>champ</i>	Nom de la table de la base de données dans laquelle le modèle linéaire généralisé Netezza sera stocké.
<code>maxit</code>	<i>entier</i>	Nombre maximum d'itérations que l'algorithme doit exécuter ; la valeur par défaut est de 20.

Tableau 183. Propriétés du noeud netezzaglm (suite).

Propriétés du noeud netezzaglm	Valeurs	Description de la propriété
eps	nombre	Valeur du nombre maximal d'erreurs (en notation scientifique) à laquelle l'algorithme doit s'arrêter de rechercher le modèle le plus adapté. La valeur par défaut est de -3, soit 1E-3 ou 0,001.
tol	nombre	Valeur (en notation scientifique) sous laquelle les erreurs sont traitées comme ayant une valeur de zéro. La valeur par défaut est de -7, ce qui signifie que les valeurs d'erreurs inférieures à 1E-7 (ou 0,0000001) sont considérées comme non significatives.
link_func	identity inverse invnegative invsquare sqrt power oddspower log clog loglog cloglog logit probit gaussit cauchit canbinom cangeom cannegbinom	Fonction de lien à utiliser ; la fonction par défaut est logit.
link_params	nombre	Valeur de paramètre de fonction de lien à utiliser. S'applique uniquement si link_function est power ou oddspower.
interaction	[[{colnames1},{levels1}],[{colnames2},{levels2}],...,[{colnamesN},{levelsN}],]	Spécifie des interactions entre les champs. colnames est une liste de champs d'entrée et level est toujours 0 pour chaque champ.
intercept	booléen	Si true, inclut la constante dans le modèle.

Propriétés du nugget de modèle Netezza

Les propriétés suivantes sont communes aux noeuds de modèle de base de données Netezza.

Tableau 184. Propriétés communes du nugget de modèle Netezza.

Propriétés communes du nugget de modèle Netezza	Valeurs	Description de la propriété
connection	chaîne	La chaîne de connexion à la base de données Netezza où le modèle est stocké.
table_name	chaîne	Le nom de la table de la base de données où le modèle est stocké.

Les autres propriétés du nugget de modèle sont les mêmes que celles du noeud de modélisation correspondant.

Les noms de script des nuggets de modèle sont les suivants.

Tableau 185. Noms de script des nuggets de modèle Netezza.

Nugget de modèle	Nom du script
Arbre de décision	applynetezadectree
k moyenne	applynetezzakmeans
Bayes Net	applynetezababes
Naive Bayes	applynetezanaivebayes
KNN	applynetezzaknn
Classification par division	applynetezadivcluster
ACP	applynetezzapca
Arbre de régression	applynetezzaregtree
Régression linéaire	applynetezzalinereregression
Séries temporelles	applynetezzatimeseries
Linéaire généralisé	applynetezzaglm

Chapitre 16. Propriétés des noeuds de sortie

Les propriétés de nœud de sortie diffèrent légèrement de celles des autres types de nœud. Au lieu de faire référence à une option de nœud particulière, les propriétés du nœud de sortie stockent une référence à l'objet de sortie. Cela peut être utile lorsque vous choisissez une valeur dans un tableau pour la définir en tant que paramètre de flux.

Cette section décrit les propriétés de génération de scripts disponibles pour les noeuds de sortie.

Propriétés du noeud Analyse



Le noeud Analyse évalue la capacité des modèles prédictifs à générer des prévisions précises. Les noeuds Analyse comparent les valeurs prédites et les valeurs réelles d'un ou de plusieurs nuggets de modèle. Ils peuvent également comparer entre eux les modèles prédictifs.

Tableau 186. Propriétés du noeud Analyse.

Propriétés du noeud Analyse	Le type de données	Description de la propriété
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
use_output_name	booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Si use_output_name a la valeur true (vrai), indique le nom à utiliser.
output_format	Text (.txt) HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie.
by_fields	[champ champ champ]	
full_filename	chaîne	Si vous indiquez Disque, Données ou HTML : nom du fichier de sortie utilisé.
coincidence	booléen	
performance	booléen	
evaluation_binary	booléen	
confidence	booléen	
seuil	nombre	
improve_accuracy	nombre	
inc_user_measure	booléen	
user_if	expression	
user_then	expression	
user_else	expression	
user_compute	[Mean Sum Min Max SDev]	

Propriétés du noeud dataaudit



Le noeud Audit données fournit un premier aperçu complet des données, notamment des statistiques récapitulatives, des histogrammes et distributions pour chaque champ, ainsi que des informations sur les valeurs éloignées, les valeurs manquantes et les valeurs extrêmes. Les résultats sont affichés dans une matrice facile à lire pouvant être triée et utilisée pour générer les noeuds de préparation des données et des graphiques grandeur nature.

Tableau 187. Propriétés du noeud dataaudit.

Propriétés du noeud dataaudit	Le type de données	Description de la propriété
custom_fields	booléen	
fields	[champ1 ... champN]	
overlay	champ	
display_graphs	booléen	Permet d'activer ou de désactiver l'affichage des graphiques dans la matrice de sortie.
basic_stats	booléen	
advanced_stats	booléen	
median_stats	booléen	
calculate	Count Breakdown	Utilisée pour calculer les valeurs manquantes. Sélectionnez une méthode de calcul, les deux ou bien aucune.
outlier_detection_method	std iqr	Utilisé pour indiquer la méthode de détection des valeurs éloignées et des valeurs extrêmes.
outlier_detection_std_outlier	nombre	Si outlier_detection_method est défini sur std, indique le nombre à utiliser pour définir les valeurs éloignées.
outlier_detection_std_extreme	nombre	Si outlier_detection_method est défini sur std, indique le nombre à utiliser pour définir les valeurs extrêmes.
outlier_detection_iqr_outlier	nombre	Si outlier_detection_method est défini sur iqr, indique le nombre à utiliser pour définir les valeurs éloignées.
outlier_detection_iqr_extreme	nombre	Si outlier_detection_method est défini sur iqr, indique le nombre à utiliser pour définir les valeurs extrêmes.
use_output_name	booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Si use_output_name a la valeur true (vrai), indique le nom à utiliser.
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.

Tableau 187. Propriétés du noeud dataaudit (suite).

Propriétés du noeud dataaudit	Le type de données	Description de la propriété
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie.
paginate_output	booléen	Lorsque output_format est défini sur HTML, cette propriété divise la sortie en pages.
lines_per_page	nombre	Lorsque cette propriété est utilisée avec paginate_output, elle indique le nombre de lignes par page de sortie.
full_filename	chaîne	

Propriétés du noeud Matrice



Le noeud Matrice permet de créer un tableau dans lequel les relations entre les champs sont indiquées. Il s'agit généralement de deux champs symboliques, mais il peut également s'agir de champs indicateurs ou numériques.

Tableau 188. Propriétés du noeud Matrice.

Propriétés du noeud Matrice	Le type de données	Description de la propriété
fields	sélectionnées Flags Numérique	
row	champ	
column	champ	
include_missing_values	booléen	Indique si les valeurs utilisateur (blancs) et système (valeurs nulles) manquantes sont incluses dans la sortie des lignes et des colonnes.
cell_contents	CrossTabs Function	
function_field	chaîne	
function	Sum Mean Min Max SDev	
sort_mode	Unsorted Ascending Descending	
highlight_top	nombre	Si non égal à zéro, alors True (vrai).
highlight_bottom	nombre	Si non égal à zéro, alors True (vrai).

Tableau 188. Propriétés du noeud Matrice (suite).

Propriétés du noeud Matrice	Le type de données	Description de la propriété
display	[Counts Expected Résiduels RowPct ColumnPct TotalPct]	
include_totals	<i>booléen</i>	
use_output_name	<i>booléen</i>	Indique si un nom de sortie personnalisé est utilisé.
output_name	<i>chaîne</i>	Si use_output_name a la valeur true (vrai), indique le nom à utiliser.
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie. Les formats Formatted et Delimited peuvent utiliser le modificateur transposed, qui transpose les lignes et les colonnes dans le tableau.
paginate_output	<i>booléen</i>	Lorsque output_format est défini sur HTML, cette propriété divise la sortie en pages.
lines_per_page	<i>nombre</i>	Lorsque cette propriété est utilisée avec paginate_output, elle indique le nombre de lignes par page de sortie.
full_filename	<i>chaîne</i>	

Propriétés du noeud Moyennes



Le noeud Moyennes compare les moyennes de groupes indépendants ou de paires de champs associés, afin de détecter toute différence sensible. Par exemple, vous pouvez comparer les revenus moyens avant et après l'application d'une augmentation, ou comparer les revenus des personnes ayant obtenu une augmentation avec ceux des personnes qui n'en ont pas eu.

Tableau 189. Propriétés du noeud Moyennes.

Propriétés du noeud Moyennes	Le type de données	Description de la propriété
means_mode	BetweenGroups BetweenFields	Indique le type de statistique de moyenne à exécuter sur les données.
test_fields	[field1 ... fieldn]	Indique le champ de test lorsque means_mode est paramétré sur BetweenGroups.
grouping_field	<i>champ</i>	Indique le champ de regroupement.
paired_fields	[{field1 field2} {field3 field4} ...]	Indique les paires de champs à utiliser lorsque means_mode est paramétré sur BetweenFields.

Tableau 189. Propriétés du noeud Moyennes (suite).

Propriétés du noeud Moyennes	Le type de données	Description de la propriété
label_correlations	<i>booléen</i>	Indique si les libellés de corrélation apparaissent dans la sortie. Ce paramètre ne s'applique que si means_mode est paramétré sur BetweenFields.
correlation_mode	Probabilité Absolue	Indique si les corrélations doivent être étiquetées par probabilité ou valeur absolue.
weak_label	<i>chaîne</i>	
medium_label	<i>chaîne</i>	
strong_label	<i>chaîne</i>	
weak_below_probability	<i>nombre</i>	Lorsque correlation_mode est paramétré sur Probability, cette propriété indique la valeur de césure utilisée pour les corrélations faibles. Il doit s'agir d'une valeur comprise entre 0 et 1 (par exemple, 0,90).
strong_above_probability	<i>nombre</i>	Valeur de césure utilisée pour les corrélations fortes.
weak_below_absolute	<i>nombre</i>	Lorsque correlation_mode est paramétré sur Absolute, cette propriété indique la valeur de césure utilisée pour les corrélations faibles. Il doit s'agir d'une valeur comprise entre 0 et 1 (par exemple, 0,90).
strong_above_absolute	<i>nombre</i>	Valeur de césure utilisée pour les corrélations fortes.
unimportant_label	<i>chaîne</i>	
marginal_label	<i>chaîne</i>	
important_label	<i>chaîne</i>	
unimportant_below	<i>nombre</i>	Valeur de césure utilisée pour une importance de champ faible. Il doit s'agir d'une valeur comprise entre 0 et 1 (par exemple, 0,90).
important_above	<i>nombre</i>	
use_output_name	<i>booléen</i>	Indique si un nom de sortie personnalisé est utilisé.
output_name	<i>chaîne</i>	Nom à utiliser.
output_mode	Screen File	Indique l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Indique le type de sortie.
full_filename	<i>chaîne</i>	

Tableau 189. Propriétés du noeud Moyennes (suite).

Propriétés du noeud Moyennes	Le type de données	Description de la propriété
output_view	Simple Advanced	Indique si la vue simple ou avancée est affichée dans la sortie.

Propriétés du noeud Rapport



Ce noeud permet de créer des rapports formatés contenant du texte fixe et des données, ainsi que des expressions calculées à partir de ces dernières. Le format du rapport est déterminé par des modèles texte définissant la structure du texte fixe et de la sortie de données. Vous pouvez définir un formatage de texte personnalisé en utilisant des balises HTML dans le modèle et en définissant des options dans l'onglet Sortie. Vous pouvez inclure des valeurs de données et d'autres sorties conditionnelles à l'aide des expressions CLEM du modèle.

Tableau 190. Propriétés du noeud Rapport.

Propriétés du noeud Rapport	Le type de données	Description de la propriété
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	HTML (.html) Text (.txt) Output (.cou)	Permet d'indiquer le type de sortie.
use_output_name	booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Si use_output_name a la valeur true (vrai), indique le nom à utiliser.
text	chaîne	
full_filename	chaîne	
highlights	booléen	
title	chaîne	
lines_per_page	nombre	

Propriétés du noeud Routput



Le noeud de sortie R vous permet d'analyser des données et les résultats du scoring de modèle à l'aide de votre propre script R personnalisé. La sortie de l'analyse peut être textuelle ou graphique. La sortie est ajoutée à l'onglet **Sortie** du panneau du gestionnaire ; autrement, la sortie peut être redirigée vers un fichier.

Tableau 191. Propriétés du noeud Routput.

Propriétés du noeud Routput	Type de données	Description de la propriété
syntax	chaîne	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	booléen	

Tableau 191. Propriétés du noeud Routput (suite).

Propriétés du noeud Routput	Type de données	Description de la propriété
convert_datetime_class	POSIXct POSIXlt	
convert_missing	booléen	
output_name	Automatique Custom	
custom_name	chaîne	
output_to	Screen File	
output_type	Graph Text	
full_filename	chaîne	
graph_file_type	HTML COU	
text_file_type	HTML TXT COU	

Propriétés du noeud V. globales



Le noeud V. globales (Valeurs globales) analyse les données et calcule des valeurs récapitulatives pouvant être utilisées dans des expressions CLEM. Par exemple, vous pouvez utiliser ce noeud pour calculer les statistiques d'un champ *âge*, puis utiliser la moyenne globale du champ *âge* dans des expressions CLEM en insérant la fonction @GLOBAL_MEAN(*âge*).

Tableau 192. Propriétés du noeud V. globales.

Propriétés du noeud V. globales	Le type de données	Description de la propriété
globals	[Sum Mean Min Max SDev]	Propriété structurée
clear_first	booléen	
show_preview	booléen	

Propriétés du noeud simeval



Le noeud Evaluation de simulation évalue un champ cible prévu et spécifié, et présente les informations de distribution et de corrélation concernant le champ cible.

Tableau 193. Propriétés du noeud simeval.

Propriétés du noeud simeval	Type de données	Description de la propriété
target	champ	
itération	champ	
presorted_by_iteration	booléen	
max_iterations	nombre	
tornado_fields	[champ1...champN]	

Tableau 193. Propriétés du noeud simeval (suite).

Propriétés du noeud simeval	Type de données	Description de la propriété
plot_pdf	booléen	
plot_cdf	booléen	
show_ref_mean	booléen	
show_ref_median	booléen	
show_ref_sigma	booléen	
num_ref_sigma	nombre	
show_ref_pct	booléen	
ref_pct_bottom	nombre	
ref_pct_top	nombre	
show_ref_custom	booléen	
ref_custom_values	[numéro1...numéroN]	
category_values	Catégorie Probabilities Both	
category_groups	Catégories Iterations	
create_pct_table	booléen	
pct_table	Quartiles Intervalles Custom	
pct_intervals_num	nombre	
pct_custom_values	[numéro1...numéroN]	

Propriétés du noeud simfit



Le noeud Ajustement de simulation examine la distribution statistique des données dans chaque champ et génère (ou met à jour) un noeud Génération de simulation, avec la distribution la plus appropriée affectée à chaque champ. Le noeud Génération de simulation peut ensuite être utilisé pour générer les données simulées.

Tableau 194. Propriétés du noeud simfit.

Propriétés du noeud simfit	Type de données	Description de la propriété
build (génération)	Node XMLExport Both	
use_source_node_name	booléen	
source_node_name	chaîne	Nom personnalisé du noeud source qui est soit en cours de génération soit en cours de mise à jour.
use_cases	Tous LimitFirstN	
use_case_limit	entier	
fit_criterion	AndersonDarling KolmogorovSmirnov	
num_bins	entier	

Tableau 194. Propriétés du noeud simfit (suite).

Propriétés du noeud simfit	Type de données	Description de la propriété
parameter_xml_filename	chaîne	
generate_parameter_import	booléen	

Propriétés du noeud Statistiques



Le noeud Statistiques fournit des informations récapitulatives de base sur les champs numériques. Il calcule les statistiques récapitulatives des champs individuels et des corrélations entre les champs.

Tableau 195. Propriétés du noeud Statistiques.

Propriétés du noeud Statistiques	Le type de données	Description de la propriété
use_output_name	booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Si use_output_name a la valeur true (vrai), indique le nom à utiliser.
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	Text (.txt) HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie.
full_filename	chaîne	
examine	[champ champ champ]	
correlate	[champ champ champ]	
statistics	[Count Mean Sum Min Max Range Variance SDev SErr Median Mode]	
correlation_mode	Probabilité Absolute	Indique si les corrélations doivent être étiquetées par probabilité ou valeur absolue.
label_correlations	booléen	
weak_label	chaîne	
medium_label	chaîne	
strong_label	chaîne	
weak_below_probability	nombre	Lorsque correlation_mode est paramétré sur Probabilité, cette propriété indique la valeur de césure utilisée pour les corrélations faibles. Il doit s'agir d'une valeur comprise entre 0 et 1 (par exemple, 0,90).
strong_above_probability	nombre	Valeur de césure utilisée pour les corrélations fortes.

Tableau 195. Propriétés du noeud Statistiques (suite).

Propriétés du noeud Statistiques	Le type de données	Description de la propriété
weak_below_absolute	nombre	Lorsque correlation_mode est paramétré sur Absolute, cette propriété indique la valeur de césure utilisée pour les corrélations faibles. Il doit s'agir d'une valeur comprise entre 0 et 1 (par exemple, 0,90).
strong_above_absolute	nombre	Valeur de césure utilisée pour les corrélations fortes.

Propriétés du noeud statisticsoutput



Le noeud Sortie Statistics vous permet d'appeler une procédure IBM SPSS Statistics pour analyser les données IBM SPSS Modeler. De nombreuses procédures d'analyses IBM SPSS Statistics sont disponibles. Ce noeud requiert une copie avec licence de IBM SPSS Statistics.

Les propriétés de ce noeud sont décrites dans «Propriétés du noeud statisticsoutput», à la page 242.

Propriétés du noeud Table



Le noeud Table affiche les données au format tabulaire (ces données peuvent également être écrites dans un fichier). Ainsi, vous pouvez passer en revue les valeurs de données ou les exporter dans un format facilement lisible.

Tableau 196. Propriétés du noeud Table.

Propriétés du noeud Table	Le type de données	Description de la propriété
full_filename	chaîne	Si vous indiquez Disque, Données ou HTML : nom du fichier de sortie utilisé.
use_output_name	booléen	Indique si un nom de sortie personnalisé est utilisé.
output_name	chaîne	Si use_output_name a la valeur true (vrai), indique le nom à utiliser.
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie.
transpose_data	booléen	Transpose les données avant l'exportation, afin que les lignes et les colonnes représentent respectivement les champs et les enregistrements.
paginate_output	booléen	Lorsque output_format est défini sur HTML, cette propriété divise la sortie en pages.

Tableau 196. Propriétés du noeud Table (suite).

Propriétés du noeud Table	Le type de données	Description de la propriété
lines_per_page	<i>nombre</i>	Lorsque cette propriété est utilisée avec <code>paginate_output</code> , elle indique le nombre de lignes par page de sortie.
highlight_expr	<i>chaîne</i>	
output	<i>chaîne</i>	Propriété en lecture seule qui comporte une référence à la dernière table créée par le nœud.
value_labels	<code>{{Valeur ChaîneLibellé}}</code> <code>{Valeur ChaîneLibellé} ...]</code>	Permet d'attribuer des libellés aux paires de valeurs.
display_places	<i>entier</i>	Définit le nombre de décimales du champ lorsqu'il est affiché (s'applique uniquement aux champs dont le stockage est Réel). La valeur -1 utilise le flux par défaut.
export_places	<i>entier</i>	Définit le nombre de décimales du champ lorsqu'il est exporté (s'applique uniquement aux champs dont le stockage est Réel). La valeur -1 utilise le flux par défaut.
decimal_separator	DEFAULT PERIOD COMMA	Définit le séparateur décimal du champ (s'applique uniquement aux champs dont le stockage est Réel).
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.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	Définit le format de date du champ (s'applique uniquement aux champs dont le stockage est DATE ou TIMESTAMP).

Tableau 196. Propriétés du noeud Table (suite).

Propriétés du noeud Table	Le type de données	Description de la propriété
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"	Définit le format d'heure du champ (s'applique uniquement aux champs dont le stockage est TIME ou TIMESTAMP).
column_width	entier	Définit la largeur des colonnes du champ. La valeur -1 définit la largeur des colonnes sur Auto.
justify	AUTO CENTER LEFT RIGHT	Définit la justification des colonnes du champ.

Propriétés du noeud Transformation



Le noeud Transformation vous permet de sélectionner et de prévisualiser les résultats des transformations avant de les appliquer aux champs sélectionnés.

Tableau 197. Propriétés du noeud Transformation.

Propriétés du noeud Transformation	Le type de données	Description de la propriété
fields	[champ1... champn]	Champs à utiliser dans la transformation.
formula	All sélectionner	Indique si toutes les transformations ou seules les transformations sélectionnées doivent être calculées.
formula_inverse	booléen	Indique si la transformation inverse doit être utilisée.
formula_inverse_offset	nombre	Indique un décalage de données à utiliser pour la formule. Élément paramétré sur 0 par défaut, sauf s'il est défini par l'utilisateur.
formula_log_n	booléen	Indique si la transformation par le logarithme _n doit être utilisée.
formula_log_n_offset	nombre	
formula_log_10	booléen	Indique si la transformation par le logarithme ₁₀ doit être utilisée.
formula_log_10_offset	nombre	

Tableau 197. Propriétés du noeud Transformation (suite).

Propriétés du noeud Transformation	Le type de données	Description de la propriété
formula_exponential	<i>booléen</i>	Indique si la transformation exponentielle (e ^x) doit être utilisée.
formula_square_root	<i>booléen</i>	Indique si la transformation par la racine carrée doit être utilisée.
use_output_name	<i>booléen</i>	Indique si un nom de sortie personnalisé est utilisé.
output_name	<i>chaîne</i>	Si use_output_name a la valeur true (vrai), indique le nom à utiliser.
output_mode	Screen File	Permet d'indiquer l'emplacement cible pour les sorties générées à partir du noeud de sortie.
output_format	HTML (.html) Output (.cou)	Permet d'indiquer le type de sortie.
paginate_output	<i>booléen</i>	Lorsque output_format est défini sur HTML, cette propriété divise la sortie en pages.
lines_per_page	<i>nombre</i>	Lorsque cette propriété est utilisée avec paginate_output, elle indique le nombre de lignes par page de sortie.
full_filename	<i>chaîne</i>	Indique le nom du fichier à utiliser pour la sortie de fichier.

Chapitre 17. Propriétés du nœud d'exportation

Propriétés communes des nœuds Exportation

Les propriétés suivantes sont communes à tous les nœuds d'exportation :

Tableau 198. Propriétés communes des nœuds d'exportation.

Propriété	Valeurs	Description de la propriété
publish_path	chaîne	Saisissez le nom racine à utiliser pour les fichiers d'images et de paramètres publiés.
publish_metadata	booléen	Spécifie si un fichier de métadonnées décrit les entrées et les sorties de l'image et de leurs modèles de données.
publish_use_parameters	booléen	Spécifie si des paramètres de flux sont inclus dans le fichier *.par.
publish_parameters	liste de chaîne	Spécifie les paramètres à inclure.
execute_mode	export_data publish	Spécifie si le nœud s'exécute sans publier le flux, aussi le flux est automatiquement publié lorsque le nœud est exécuté.

Propriétés du nœud asexport

L'exportation Analytic Server vous permet d'exécuter un flux sur le système de fichiers HDFS (Hadoop Distributed File System).

Tableau 199. Propriétés du nœud asexport.

Propriétés du nœud asexport	Type de données	Description de la propriété
data_source	chaîne	Nom de la source de données.
export_mode	chaîne	Indique s'il faut ajouter (append) les données exportées à la source de données existante ou s'il faut remplacer (overwrite) la source de données existante.
host	chaîne	Nom de l'hôte Analytic Server.
port	entier	Port sur lequel Analytic Server est en mode écoute.
tenant	chaîne	Dans un environnement à service partagé, nom du titulaire auquel vous appartenez. Dans un environnement à service exclusif, prend par défaut la valeur ibm .
set_credentials	booléen	Si l'authentification utilisateur sur Analytic Server est la même que sur le serveur SPSS Modeler, donnez à cette propriété la valeur false . Sinon, donnez-lui la valeur true .
user_name	chaîne	Nom d'utilisateur utilisé pour la connexion à Analytic Server. Uniquement nécessaire si set_credentials a la valeur true.

Tableau 199. Propriétés du noeud asexport (suite).

Propriétés du noeud asexport	Type de données	Description de la propriété
password	chaîne	Mot de passe de connexion à Analytic Server. Uniquement nécessaire si set_credentials a la valeur true.

Propriétés du noeud cognosexport



Le noeud IBM Cognos BI Export exporte des données dans un format qui peut être lu par les bases de données Cognos BI.

Remarque : pour ce noeud, vous devez définir une connexion Cognos et une connexion ODBC.

Connexion Cognos

Les propriétés de la connexion Cognos sont les suivantes.

Tableau 200. Propriétés du noeud cognosexport.

Propriétés du noeud cognosexport	Le type de données	Description de la propriété
cognos_connection	{"champ", "champ", ... ,"champ"}	Une propriété de liste contenant les détails de la connexion du serveur Cognos. Le format est le suivant : {"URL_Serveur_Cognos", mode_connexion, "espace de nommage", "nom utilisateur", "mot de passe"} où : URL_Serveur_Cognos est l'URL du serveur Cognos vers lequel vous effectuez l'exportation mode_connexion indique si la connexion anonyme est utilisée et est soit true soit false ; si la valeur est true, les champs suivants doivent être définis sur "" espace de nommage spécifie le fournisseur de sécurité pour l'authentification utilisé pour se connecter au serveur. nom utilisateur et mot de passe sont ceux utilisés pour la connexion au serveur Cognos
cognos_package_name	chaîne	Le chemin d'accès et le nom du package Cognos vers lequel vous exportez les données, par exemple : /Public Folders/MyPackage
cognos_datasource	chaîne	
cognos_export_mode	Publish ExportFile	
cognos_filename	chaîne	

connexion ODBC

Les propriétés de la connexion ODBC sont identiques à celles répertoriées pour databaseexport dans la prochaine section, à l'exception de la propriété datasource qui n'est pas valide.

Propriétés du noeud databaseexport



Le noeud exportation de base de données écrit des données dans une source de données relationnelles compatible ODBC. Pour que cette opération puisse être effectuée, la source de données ODBC doit exister et vous devez y avoir accès en écriture.

Tableau 201. Propriétés du noeud databaseexport.

Propriétés du noeud databaseexport	Le type de données	Description de la propriété
datasource	chaîne	
username	chaîne	
password	chaîne	
epassword	chaîne	Cette propriété est en lecture seule au cours de l'exécution. Pour générer un mot de passe codé, sélectionnez Outil pour mot de passe dans le menu Outils. Pour plus d'informations, voir la rubrique «Génération d'un mot de passe codé», à la page 47.
table_name	chaîne	
write_mode	Create Append Merge	
map	chaîne	Mappe un nom de champ de flux sur un nom de colonne de base de données (valide uniquement si write_mode est sur Merge). Pour une fusion, tous les champs doivent être mappés afin d'être exportés. Les noms de champ qui n'existent pas dans la base de données sont ajoutés en tant que nouvelles colonnes.
key_fields	[champ champ ... champ]	Spécifie le champ de flux utilisé pour la clé ; la propriété map indique à quoi cela correspond dans la base de données.
join	Database Add	
drop_existing_table	booléen	
delete_existing_rows	booléen	
default_string_size	entier	
type		Propriété structurée utilisée pour définir le type de schéma.
generate_import	booléen	

Tableau 201. Propriétés du noeud databaseexport (suite).

Propriétés du noeud databaseexport	Le type de données	Description de la propriété
use_custom_create_table_command	booléen	Utilisez la propriété <i>custom_create_table</i> pour modifier la commande SQL standard CREATE TABLE.
custom_create_table_command	chaîne	Indique la commande de chaîne à utiliser à la place de la commande SQL standard CREATE TABLE.
use_batch	booléen	Les propriétés suivantes sont les options avancées du chargement en masse dans la base de données. La valeur True (vrai) pour Use_batch désactive les validations ligne par ligne soumises à la base de données.
batch_size	nombre	Indique le nombre d'enregistrements à envoyer à la base de données avant validation dans la mémoire.
bulk_loading	Off ODBC External	Indique le type de chargement en masse. Les options supplémentaires relatives à ODBC et à External sont répertoriées ci-dessous.
not_logged	booléen	
odbc_binding	Ligne Column	Indiquez le lien par ligne ou par colonne pour le chargement en masse via ODBC.
loader_delimit_mode	Tabulation Space Other	Indiquez le type de délimiteur pour le chargement en masse via un programme externe. Sélectionnez Other avec la propriété loader_other_delimiter pour indiquer les délimiteurs, tels que la virgule (,).
loader_other_delimiter	chaîne	
specify_data_file	booléen	Un indicateur ayant la valeur True (vrai) active la propriété data_file ci-dessous, vous permettant d'indiquer le nom et le chemin du fichier où écrire lors du chargement en masse dans la base de données.
data_file	chaîne	
specify_loader_program	booléen	Un indicateur ayant la valeur True (vrai) active la propriété loader_program ci-dessous, vous permettant d'indiquer le nom et l'emplacement d'un programme ou d'un script de module de chargement externe.
loader_program	chaîne	

Tableau 201. Propriétés du noeud databaseexport (suite).

Propriétés du noeud databaseexport	Le type de données	Description de la propriété
gen_logfile	booléen	Un indicateur ayant la valeur True (vrai) active la propriété logfile_name ci-dessous, vous permettant d'indiquer le nom d'un fichier se trouvant sur le serveur pour générer un journal des erreurs.
logfile_name	chaîne	
check_table_size	booléen	Un indicateur ayant la valeur True (vrai) permet de vérifier les tables pour garantir que l'augmentation de la taille des tables de la base de données correspond au nombre de lignes exportées à partir de IBM SPSS Modeler.
loader_options	chaîne	Indiquez des arguments supplémentaires (comme -comment et -specialdir) pour le programme du module de chargement.
export_db_primarykey	booléen	Indique si un champ donné est une clé primaire.
use_custom_create_index_command	booléen	Si la valeur true (vrai) est utilisée, active le code SQL personnalisé pour tous les index.
custom_create_index_command	chaîne	Indique la commande SQL utilisée pour créer des index lorsque le code SQL personnalisé est activé. (Cette valeur peut être ignorée pour certains index comme indiqué ci-dessous.)
indexes.INDEXNAME.fields		Crée l'index indiqué si nécessaire et répertorie les noms de champ à inclure dans cet index.
indexes.INDEXNAME.use_custom_create_index_command	booléen	Permet d'activer/de désactiver le code SQL personnalisé pour un index spécifique.
indexes.INDEXNAME.custom_create_command		Indique le code SQL personnalisé utilisé pour l'index spécifié.
indexes.INDEXNAME.remove	booléen	Si la valeur true (vrai) est utilisée, supprime l'index spécifié de l'ensemble d'index.
table_space	chaîne	Spécifie l'espace Table qui sera créé.
use_partition	booléen	Spécifie que le champ de distribution par hachage sera utilisé.
partition_field	chaîne	Spécifie le contenu du champ de distribution par hachage.

Remarque : pour certaines bases de données, vous pouvez spécifier que les tables de base de données doivent être créées pour l'exportation avec compression (par exemple, l'équivalent de CREATE TABLE MYTABLE (...) COMPRESS YES; dans SQL). Les propriétés use_compression et compression_mode sont fournies pour prendre en charge cette fonctionnalité, comme suit.

Tableau 202. Propriétés du noeud databaseexport avec fonctions de compression.

Propriétés du noeud databaseexport	Le type de données	Description de la propriété
use_compression	booléen	Si défini sur true, crée des tables à exporter avec la compression.
compression_mode	Ligne Page	Définit le niveau de compression des bases de données SQL Server.
	Default Direct_Load_Operations All_Operations Basique OLTP Query_High Query_Low Archive_High Archive_Low	Définit le niveau de compression des bases de données Oracle. Veuillez noter que les valeurs OLTP, Query_High, Query_Low, Archive_High et Archive_Low requièrent Oracle 11gR2 au minimum.

Propriétés du noeud datacollectionexport



Le noeud Export IBM SPSS Data Collection génère des données au format utilisé par les logiciels d'étude de marché IBM SPSS Data Collection. Pour pouvoir utiliser ce noeud, vous devez avoir installé avant la bibliothèque de données IBM SPSS Data Collection.

Tableau 203. Propriétés du noeud datacollectionexport.

Propriétés du noeud datacollectionexport	Le type de données	Description de la propriété
metadata_file	chaîne	Nom du fichier de métadonnées à exporter.
merge_metadata	Overwrite MergeCurrent	
enable_system_variables	booléen	Spécifie si le fichier .mdd exporté devrait inclure les variables système IBM SPSS Data Collection.
casedata_file	chaîne	Le nom du fichier .sav vers lequel les données d'observation sont exportées.
generate_import	booléen	

Propriétés du noeud excelexport



Le noeud Export Excel génère une sortie de données au format Microsoft Excel (.xls). Si vous le souhaitez, vous pouvez choisir de lancer Excel automatiquement et d'ouvrir le fichier exporté lors de l'exécution du noeud.

Tableau 204. Propriétés du noeud excelexport.

Propriétés du noeud excelexport	Le type de données	Description de la propriété
full_filename	chaîne	
excel_file_type	Excel2003 Excel2007	

Tableau 204. Propriétés du noeud excelexport (suite).

Propriétés du noeud excelexport	Le type de données	Description de la propriété
export_mode	Create Append	
inc_field_names	booléen	Indique si les noms de champ doivent être inclus dans la première ligne de la feuille de calcul.
start_cell	chaîne	Spécifie la cellule de démarrage de l'export.
worksheet_name	chaîne	Nom de la feuille de calcul à écrire.
launch_application	booléen	Indique si Excel doit être appelé pour le fichier obtenu. Notez que le chemin de lancement d'Excel doit être spécifié dans la boîte de dialogue Programmes externes (menu Outils, Programmes externes).
generate_import	booléen	Indique si un noeud Import Excel doit être généré pour lire le fichier de données exporté.

Propriétés du noeud outputfile



Le noeud Export Fichier plat génère des données dans un fichier texte délimité. Elles peuvent ainsi être lues par d'autres logiciels d'analyse ou par des tableurs.

Tableau 205. Propriétés du noeud outputfile.

Propriétés du noeud outputfile	Le type de données	Description de la propriété
full_filename	chaîne	Nom du fichier de sortie.
write_mode	Overwrite Append	
inc_field_names	booléen	
use_newline_after_records	booléen	
delimit_mode	Comma Tabulation Space Other	
other_delimiter	char	
quote_mode	None Single Double Other	
other_quote	booléen	
generate_import	booléen	

Tableau 205. Propriétés du noeud outputfile (suite).

Propriétés du noeud outputfile	Le type de données	Description de la propriété
encoding	StreamDefault SystemDefault "UTF-8"	

Propriétés du noeud sasexport



Le noeud Export SAS permet d'obtenir des données de sortie au format SAS afin qu'elles puissent être lues par SAS ou par un logiciel compatible. Trois formats de fichier SAS sont disponibles : SAS pour Windows/OS2, SAS pour UNIX ou SAS Version 7/8.

Tableau 206. Propriétés du noeud sasexport.

Propriétés du noeud sasexport	Le type de données	Description de la propriété
format	Windows UNIX SAS7 SAS8	Champs étiquette de propriétés de variante.
full_filename	chaîne	
export_names	NamesAndLabels NamesAsLabels	Permet de mapper les noms de champ de IBM SPSS Modeler lors de l'exportation vers des noms de variable IBM SPSS Statistics ou SAS.
generate_import	booléen	

Propriétés du noeud statisticsexport



Le noeud Exporter Statistics génère des données au format IBM SPSS Statistics *.sav*. Les fichiers *.sav* peuvent être lus par IBM SPSS Statistics Base et d'autres produits . Ce format est également utilisé pour les fichiers cache IBM SPSS Modeler.

Les propriétés de ce noeud sont décrites dans «Propriétés du noeud statisticsexport», à la page 242.

Propriétés du noeud xmlexport



Le noeud Export XML génère une sortie de données dans un fichier au format XML. Vous pouvez également créer un noeud source XML pour lire de nouveau les données exportées dans le flux.

Tableau 207. Propriétés du noeud xmlexport.

Propriétés du noeud xmlexport	Le type de données	Description de la propriété
full_filename	chaîne	(requis) Chemin complet et nom de fichier du fichier d'exportation XML.

Tableau 207. Propriétés du nœud `xmlexport` (suite).

Propriétés du nœud <code>xmlexport</code>	Le type de données	Description de la propriété
<code>use_xml_schema</code>	<i>booléen</i>	Spécifiez s'il faut utiliser un schéma XML (fichier XSD ou DTD) pour contrôler la structure des données exportées.
<code>full_schema_filename</code>	<i>chaîne</i>	Chemin complet et nom de fichier et du fichier XSD ou DTD à utiliser. Requis si <code>use_xml_schema</code> est défini sur <code>true</code> (vrai).
<code>generate_import</code>	<i>booléen</i>	Génère un nœud source XML pour lire le fichier de données exporté dans le flux.
<code>records</code>	<i>chaîne</i>	Expression XPath qui indique la limite de l'enregistrement.
<code>map</code>	<i>chaîne</i>	Mappe le nom de fichier dans une structure XML.

Chapitre 18. Propriétés de noeuds IBM SPSS Statistics

Propriétés du noeud statisticsimport



Le noeud Statistics lit les données du format de fichier *.sav* utilisé par IBM SPSS Statistics ainsi que des fichiers cache enregistrés dans IBM SPSS Modeler, qui utilisent le même format.

Tableau 208. Propriétés du noeud statisticsimport.

Propriétés du noeud statisticsimport	Le type de données	Description de la propriété
full_filename	chaîne	Nom du fichier complet, y compris son chemin d'accès.
import_names	NamesAndLabels LabelsAsNames	Méthode de gestion des noms de variable et des libellés.
import_data	DataAndLabels LabelsAsData	Méthode de gestion des valeurs et des libellés.
use_field_format_for_storage	Booléen	Spécifie s'il convient d'utiliser les informations de format de champ IBM SPSS Statistics lors de l'importation.

Propriétés du noeud statisticstransform



Le noeud Transformation exécute une sélection de commandes de syntaxe IBM SPSS Statistics en fonction des sources de données dans IBM SPSS Modeler. Ce noeud requiert une copie avec licence de IBM SPSS Statistics.

Tableau 209. Propriétés du noeud statisticstransform.

Propriétés du noeud statisticstransform	Le type de données	Description de la propriété
syntax	chaîne	
check_before_saving	booléen	Valide la syntaxe saisie avant d'enregistrer les entrées. Affiche un message d'erreur si la syntaxe n'est pas valide.
default_include	booléen	Pour plus d'informations, voir la rubrique «Propriétés du noeud Filtrer», à la page 103.
include	booléen	Pour plus d'informations, voir la rubrique «Propriétés du noeud Filtrer», à la page 103.
new_name	chaîne	Pour plus d'informations, voir la rubrique «Propriétés du noeud Filtrer», à la page 103.

Propriétés du noeud statisticsmodel



Le noeud Modèle Statistics vous permet d'analyser et de travailler avec vos données en exécutant des procédures IBM SPSS Statistics qui produisent un PMML. Ce noeud requiert une copie avec licence de IBM SPSS Statistics.

Propriétés du noeud statisticsmodel	Le type de données	Description de la propriété
syntax	chaîne	
default_include	booléen	Pour plus d'informations, voir la rubrique «Propriétés du noeud Filtrer», à la page 103.
include	booléen	Pour plus d'informations, voir la rubrique «Propriétés du noeud Filtrer», à la page 103.
new_name	chaîne	Pour plus d'informations, voir la rubrique «Propriétés du noeud Filtrer», à la page 103.

Propriétés du noeud statisticsoutput



Le noeud Sortie Statistics vous permet d'appeler une procédure IBM SPSS Statistics pour analyser les données IBM SPSS Modeler. De nombreuses procédures d'analyses IBM SPSS Statistics sont disponibles. Ce noeud requiert une copie avec licence de IBM SPSS Statistics.

Tableau 210. Propriétés du noeud statisticsoutput.

Propriétés du noeud statisticsoutput	Le type de données	Description de la propriété
mode	Dialog Syntax	Sélectionne l'option « Boîte de dialogue IBM SPSS Statistics » ou l'Éditeur de syntaxe.
syntax	chaîne	
use_output_name	booléen	
output_name	chaîne	
output_mode	Screen File	
full_filename	chaîne	
type_fichier	HTML SPV SPW	

Propriétés du noeud statisticsexport



Le noeud Exporter Statistics génère des données au format IBM SPSS Statistics *.sav*. Les fichiers *.sav* peuvent être lus par IBM SPSS Statistics Base et d'autres produits. Ce format est également utilisé pour les fichiers cache IBM SPSS Modeler.

Tableau 211. Propriétés du noeud *statisticsexport*.

Propriétés du noeud <i>statisticsexport</i>	Le type de données	Description de la propriété
<code>full_filename</code>	<i>chaîne</i>	
<code>launch_application</code>	<i>booléen</i>	
<code>export_names</code>	NamesAndLabels NamesAsLabels	Permet de mapper les noms de champ de IBM SPSS Modeler lors de l'exportation vers des noms de variable IBM SPSS Statistics ou SAS.
<code>generate_import</code>	<i>booléen</i>	

Chapitre 19. Propriétés du super noeud

Les propriétés propres aux super noeuds sont décrites dans les tableaux suivants : Remarque : les propriétés de noeud communes s'appliquent également aux super noeuds.

Tableau 212. Propriétés de super noeud terminal.

Nom de la propriété	Type de propriété/Liste de valeurs	Description de la propriété
execute_method	Script Normal	
script	chaîne	
script_language	Python Existant	Définit le langage de script pour le script de super noeud.

Paramètres du super noeud

Vous pouvez utiliser des scripts pour créer ou définir les paramètres du super noeud en utilisant les mêmes fonctions que celles utilisées pour modifier les paramètres du flux. Pour plus d'informations, voir la rubrique «Paramètres de flux, de session et de super noeud», à la page 40.

Définition des propriétés des noeuds encapsulés

Pour définir les propriétés sur les noeuds au sein du super noeud, vous devez accéder au diagramme appartenant à ce super noeud, puis utiliser les diverses méthodes find (telles que findByName() et findById()) pour localiser les noeuds. Par exemple, dans un script de super noeud incluant un seul noeud type :

```
supernode = modeler.script.supernode()
diagram = supernode.getCompositeProcessorDiagram()
# Find the type node within the supernode internal diagram
typenode = diagram.findByName("type", None)
typenode.setKeyedProperty("direction", "Drug", "Input")
typenode.setKeyedProperty("direction", "Age", "Target")
```

Limites des scripts de super noeud. Les super noeuds ne peuvent pas manipuler d'autres flux, ni modifier le flux actuel.

Annexe A. Référence des noms de noeuds

Cette section fournit une référence pour les noms de scriptage des noeuds dans IBM SPSS Modeler.

Noms des nuggets de modèle

Les nuggets de modèle (également appelés modèles générés) peuvent être référencés par type, tout comme les objets de noeud et de sortie. Le tableau ci-dessous reprend les noms de référence d'objet de modèle.

Ces noms sont utilisés spécialement pour le référencement des nuggets de modèle figurant dans la palette Modèles (dans l'angle supérieur droit de la fenêtre IBM SPSS Modeler). Pour référencer les noeuds de modèle qui ont été ajoutés à un flux à des fins de scoring, le système utilise un ensemble différent de noms commençant par `apply...`. Pour plus d'informations, voir la rubrique Chapitre 14, «Propriétés du noeud de nugget de modèle», à la page 179.

Remarque : en règle générale, le référencement des modèles à la fois par nom *et* par type est recommandé car il permet d'éviter toute confusion.

Tableau 213. Noms des nuggets de modèle (Palette Modélisation).

Nom du modèle	Modèle
anomalydetection	Anomalie
apriori	Apriori
autoclassifier	Discriminant automatique
autocluster	Classification non supervisée automatique
autonumeric	Numérisation automatique
bayesnet	Réseau Bayésien
c50	C5.0
carma	Carma
cart	Arbre C&RT
chaid	CHAID
coxreg	Régression de Cox
decisionlist	Liste de décision
discriminant	Analyse discriminante
facteur	ACP/Facteur
featureselection	Sélection de fonction
genlin	Régression linéaire généralisée
glmm	GLMM
kmeans	k moyenne
knn	k-Voisin le plus proche
kohonen	Kohonen
linear	Linéaire
logreg	Régression logistique
neuralnetwork	Réseau de neurones

Tableau 213. Noms des nuggets de modèle (Palette Modélisation) (suite).

Nom du modèle	Modèle
quest	QUEST
régression	Régression linéaire
séquence	Séquence
slrm	Modèle de réponse en auto-apprentissage
statisticsmodel	Modèle IBM SPSS Statistics
svm	Support vector machine
timeseries	Séries temporelles
twostep	TwoStep

Tableau 214. Noms des nuggets de modèle (Palette Modélisation de base de données).

Nom du modèle	Modèle
db2imcluster	Classification ISW IBM
db2imlog	Régression logistique ISW IBM
db2imnb	Naive Bayes ISW IBM
db2imreg	Régression ISW IBM
db2imtree	Arbre de décision ISW IBM
msassoc	Règles d'association MS
msbayes	MS Naive Bayes
mscluster	Classification non supervisée MS
mslogistic	Régression logistique MS
msneuralnetwork	Réseau neuronal MS
msregression	Régression linéaire MS
mssequencecluster	Mise en cluster de séquences MS
mstimeseries	Séries temporelles MS
mstree	Arbre de décision MS
netezزابayes	Bayes Net Netezza
netezзадectree	Arbre décision Netezza
netezзадivcluster	Classification par division Netezza
netezzaglm	Linéaire généralisé Netezza
netezzakmeans	k moyenne Netezza
netezzaknn	KNN Netezza
netezzalinerégression	Régression linéaire Netezza
netezzanaivebayes	Naive Bayes Netezza
netezzapca	ACP Netezza
netezzaregtree	Arbre de régression Netezza
netezzatimeseries	Séries temporelles Netezza
oraabn	Oracle Adaptive Bayes
oraai	Oracle AI
oradecisiontree	Arbre décision Oracle
oraglm	Oracle GLM

Tableau 214. Noms des nuggets de modèle (Palette Modélisation de base de données) (suite).

Nom du modèle	Modèle
orakmeans	k moyenne Oracle
oranb	Oracle Naive Bayes
oranmf	NMF Oracle
oracluster	O-Cluster Oracle
orasvm	Oracle SVM

Pour éviter les noms de modèle en double

Lorsque vous utilisez des scripts pour manipuler les modèles générés, gardez à l'esprit que les noms de modèle en double peuvent donner lieu à des références ambiguës. Pour éviter ce problème, il s'avère judicieux, lors de la génération de scripts, d'utiliser des noms uniques pour les modèles générés.

Pour définir les options des noms de modèle en double, procédez comme suit :

1. A partir des menus, sélectionnez :
Outils > Options d'utilisateur
2. Cliquez sur l'onglet **Notifications**.
3. Sélectionnez **Remplacer le modèle précédent** afin de limiter les noms en double pour les modèles générés.

Le comportement de l'exécution du script peut varier entre SPSS Modeler et IBM SPSS Collaboration and Deployment Services lorsqu'il existe des références de modèle ambiguës. Le client SPSS Modeler inclut l'option « Remplacer le modèle précédent » qui remplace automatiquement les modèles qui ont le même nom (par exemple, lorsqu'un script parcourt une boucle pour produire un modèle différent à chaque fois). Cependant, cette option n'est pas disponible lorsque le même script est exécuté dans IBM SPSS Collaboration and Deployment Services. Vous pouvez éviter cette situation en renommant le modèle généré dans chaque itération pour éviter des références ambiguës aux modèles ou en effaçant le modèle actuel (par exemple, en ajoutant une instruction `clear generated palette`) avant la fin de la boucle.

Nom des types de sortie

Le tableau suivant répertorie tous les types d'objet de sortie et les noeuds qui les génèrent. Pour obtenir la liste complète des formats d'exportation disponibles pour chaque type d'objet de sortie, reportez-vous à la description des propriétés propres au noeud qui génère le type de sortie en question, dans «Propriétés communes aux noeuds Graphiques», à la page 119 et Chapitre 16, «Propriétés des noeuds de sortie», à la page 217.

Tableau 215. Types d'objet de sortie et noeuds qui les génèrent.

Type d'objet de sortie	Noeud
analysisoutput	Analyse
collectionoutput	Collecte
dataauditoutput	Audit données
distributionoutput	Proportion
evaluationoutput	Evaluation
histogramoutput	Histogramme
matrixoutput	Matrice
meansoutput	Moyennes
multiplotoutput	Courbes

Tableau 215. Types d'objet de sortie et noeuds qui les génèrent (suite).

Type d'objet de sortie	Noeud
plotoutput	Tracé
qualityoutput	Qualité
reportdocumentoutput	Ce type d'objet n'est pas issu d'un noeud ; il s'agit en fait de la sortie créée par un rapport de projet.
reportoutput	Rapport
statisticsprocedureoutput	Sortie Statistiques
statisticsoutput	Statistiques
tableoutput	Table
timeplotoutput	Tracé horaire
weboutput	Web

Annexe B. Migration du scriptage existant au scriptage Python

Présentation de la migration de script existant

Cette section présente un récapitulatif des différences entre le scriptage Python et le scriptage existant dans IBM SPSS Modeler. Elle fournit également des informations sur la façon de migrer vos scripts existants vers des scripts Python. Dans cette section, vous trouverez une liste des commandes existantes SPSS Modeler standard et les commandes Python équivalentes.

Différences générales

Le design du scriptage existant doit beaucoup aux scripts de commande OS. Le scriptage existant est orienté ligne et, bien qu'il y ait quelques structures de bloc (par exemple `if...then...else...endif` et `for...endfor`), l'indentation n'est généralement pas significative.

Dans le scriptage Python, l'indentation est significative et les lignes appartenant au même bloc logique doivent être mises en retrait au même niveau.

Remarque : Vous devez faire attention lorsque vous copiez et collez du code Python. Une ligne qui est mise en retrait à l'aide de tabulations peut avoir la même présentation dans l'éditeur qu'une ligne mise en retrait à l'aide d'espaces. Toutefois, le script Python générera une erreur car les lignes ne sont pas considérées comme ayant la même indentation.

Contexte de scriptage)

Le contexte de scriptage (ou génération de scripts) définit l'environnement dans lequel le script est exécuté (par exemple, le flux ou le super noeud qui exécute le script). Dans le scriptage existant, le contexte est implicite, ce qui signifie par exemple que toute référence de noeud dans un script de flux est supposé se trouver dans le flux qui exécute le script.

Dans le scriptage Python, le contexte de scriptage est fourni explicitement via le module `modeler.script`. Par exemple, un script de flux Python peut accéder au flux qui exécute le script via le code suivant :

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

Les fonctions liées au flux peuvent ensuite être invoquées via l'objet renvoyé.

Commandes et fonctions

Le scriptage existant est orienté commande. Cela signifie que chaque ligne de script commence généralement par la commande à exécuter, cette dernière étant suivie des paramètres ; par exemple :

```
connect 'Type':typenode to :filternode  
rename :derivenode as "Compute Total"
```

Python utilise des fonctions qui sont généralement invoquées via un objet (module, classe ou objet) définissant la fonction ; par exemple :

```
stream = modeler.script.stream()  
typenode = stream.findByName("type", "Type")  
filternode = stream.findByName("filter", None)  
stream.link(typenode, filternode)  
derive.setLabel("Compute Total")
```

Littéraux et commentaires

Certaines commandes de littéraux et commentaires couramment utilisées dans IBM SPSS Modeler présentent des commandes équivalentes dans le scriptage Python. Cela peut vous aider à convertir vos scripts SPSS Modeler existants en scripts Python afin de les utiliser dans IBM SPSS Modeler 16.

Tableau 216. Mappage du scriptage existant au scriptage Python pour les littéraux et commentaires.

Scriptage existant	Scriptage Python
Entier, par exemple 4	Identique
Valeur flottante, par exemple 0.003	Identique
Chaînes entre guillemets simples, par exemple 'Hello'	Identique Remarque : Les littéraux chaîne qui contiennent des caractères non-ASCII doivent être précédés d'un u afin d'être sûr qu'ils soient représentés en Unicode.
Chaînes entre guillemets doubles, par exemple "Hello again"	Identique Remarque : Les littéraux chaîne qui contiennent des caractères non-ASCII doivent être précédés d'un u afin d'être sûr qu'ils soient représentés en Unicode.
Chaînes longues, par exemple """This is a string that spans multiple lines"""	Identique
Listes, par exemple [1 2 3]	[1, 2, 3]
Référence de variable, par exemple set x = 3	x = 3
Continuation de ligne (\), par exemple set x = [1 2 \ 3 4]	x = [1, 2,\n3, 4]
Commentaire de bloc, par exemple /* This is a long comment over a line. */	/* This is a long comment over a line. */
Commentaire de ligne, par exemple set x = 3 # make x 3	x = 3 # make x 3
undef	None
true	True
false	False

Opérateurs

Certaines commandes d'opérateur couramment utilisées dans IBM SPSS Modeler présentent des commandes équivalentes dans le scriptage Python. Cela peut vous aider à convertir vos scripts SPSS Modeler existants en scripts Python afin de les utiliser dans IBM SPSS Modeler 16.

Tableau 217. Mappage du scriptage existant au scriptage Python pour les opérateurs.

Scriptage existant	Scriptage 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

Tableau 217. Mappage du scriptage existant au scriptage Python pour les opérateurs (suite).

Scriptage existant	Scriptage Python
NUM1 / NUM2	NUM1 / NUM2
= ==	==
/= /==	!=
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
et ou not(EXPR)	et ou not EXPR

Commandes conditionnelles et de bouclage

Certaines commandes conditionnelles et de bouclage couramment utilisées dans IBM SPSS Modeler présentent des commandes équivalentes dans le scriptage Python. Cela peut vous aider à convertir vos scripts SPSS Modeler existants en scripts Python afin de les utiliser dans IBM SPSS Modeler 16.

Tableau 218. Mappage du scriptage existant au scriptage Python pour les commandes conditionnelles et de bouclage.

Scriptage existant	Scriptage Python
for VAR from INT1 to INT2 ... endfor	for VAR in range(INT1, INT2): ... ou 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	Pas d'équivalent

Tableau 218. Mappage du scriptage existant au scriptage Python pour les commandes conditionnelles et de bouclage (suite).

Scriptage existant	Scriptage Python
var VAR1	La déclaration de variable n'est pas requise

Variabes

Dans le scriptage existant, les variables sont déclarées avant d'être référencées ; par exemple :

```
var mynode
set mynode = create typenode at 96 96
```

Dans le scriptage Python, les variables sont déclarées lors de leur premier référencement ; par exemple :

```
mynode = stream.createAt("type", "Type", 96, 96)
```

Dans le scriptage existant, les références aux variables doivent être explicitement supprimées via l'opérateur ^ ; par exemple :

```
var mynode
set mynode = create typenode at 96 96
set ^mynode.direction."Age" = Input
```

Comme dans la plupart des langages de scriptage, cette opération n'est pas nécessaire dans Python ; par exemple :

```
mynode = stream.createAt("type", "Type", 96, 96)
mynode.setKeyedPropertyValue("direction", "Age", "Input")
```

Types de noeuds, de sorties et de modèles

Dans le scriptage existant, les différents types d'objets (noeud, sortie et modèle) sont généralement ajoutés à l'objet concerné. Par exemple, le noeud Dériver possède le type `derivenode` :

```
set feature_name_node = create derivenode at 96 96
```

L'API IBM SPSS Modeler dans le langage Python n'inclut pas le suffixe `node` ; le noeud Dériver possède donc le type `derive` ; par exemple :

```
feature_name_node = stream.createAt("derive", "Feature", 96, 96)
```

La seule différence entre les noms de type dans le scriptage existant et le scriptage Python réside dans l'absence de suffixe de type.

Noms de propriétés

Les noms de propriétés sont identiques dans le scriptage existant et le scriptage Python. Par exemple, dans le noeud Délimité, la propriété qui définit l'emplacement du fichier est `full_filename` dans les deux environnements de scriptage.

Références de noeud

De nombreux scripts existants utilisent une recherche implicite pour rechercher le noeud à modifier et y accéder. Par exemple, les commandes suivantes recherchent dans le flux en cours un noeud `Type` ayant le libellé `"Type"`, puis définissent la direction (ou le rôle de modélisation) du champ `"Age"` sur `Input` et du champ `"Drug"` sur `Target` (qui constitue la valeur à prévoir) :

```
set 'Type':typenode.direction."Age" = Input
set 'Type':typenode.direction."Drug" = Target
```

Dans le scriptage Python, les objets de noeud doivent être localisés de façon explicite avant d'appeler la fonction en vue de définir la valeur de la propriété ; par exemple :

```
typenode = stream.findByType("type", "Type")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
typenode.setKeyedPropertyValue("direction", "Drug", "Target")
```

Remarque : Dans ce cas, "Target" doit être entre guillemets.

Les scripts Python peuvent aussi utiliser l'énumération `ModelingRole` dans le pack `modeler.api`.

Bien que la version du scriptage Python puisse être plus prolix, elle conduit à des meilleures performances d'exécution car la recherche du noeud n'est généralement effectuée qu'une seule fois. Dans l'exemple de scriptage existant, la recherche du noeud est effectuée pour chaque commande.

La recherche de noeuds par ID est également prise en charge (l'ID de noeud est visible dans l'onglet Annotations de la boîte de dialogue de noeud). Par exemple, dans le scriptage existant :

```
# id65EMPB9VL87 est l'ID d'un noeud Type
set @id65EMPB9VL87.direction."Age" = Input
```

Le script suivant présente le même exemple en Python :

```
typenode = stream.findByID("id65EMPB9VL87")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
```

Extraction et définition de propriétés

Le scriptage existant utilise la commande `set` pour affecter une valeur. Le terme suivant la commande `set` peut être une définition de propriété. Le script suivant présente deux formats de script possibles pour définir une propriété :

```
set <référence_noeud>.<propriété> = <valeur>
set <référence_noeud>.<propriété_saisie>.<clé> = <valeur>
```

Dans le scriptage Python, on obtient le même résultat en utilisant les fonctions `setPropertyValue()` et `setKeyedPropertyValue()` ; par exemple :

```
objet.setPropertyValue(propriété, valeur)
objet.setKeyedPropertyValue(propriété_saisie, clé, valeur)
```

Dans le scriptage existant, on peut accéder aux valeurs de propriété en utilisant la commande `get` ; par exemple :

```
var n v
set n = get node :filternode
set v = ^n.name
```

Dans le scriptage Python, on obtient le même résultat en utilisant la fonction `getPropertyValue()` ; par exemple :

```
n = stream.findByName("filter", None)
v = n.getPropertyValue("name")
```

Edition de flux

Dans le scriptage existant, la commande `create` est utilisée pour créer un noeud ; par exemple :

```
var agg select
set agg = create aggregatenode at 96 96
set select = create selectnode at 164 96
```

Dans le scriptage Python, les flux possèdent diverses méthodes de création de noeuds ; par exemple :

```
stream = modeler.script.stream()
agg = stream.createAt("aggregate", "Aggregate", 96, 96)
select = stream.createAt("select", "Select", 164, 96)
```

Dans le scriptage existant, la commande `connect` est utilisée pour créer des liens entre les noeuds ; par exemple :

```
connect ^agg to ^select
```

Dans le scriptage Python, c'est la méthode `link` qui est utilisée pour créer des liens entre les noeuds ; par exemple :

```
stream.link(agg, select)
```

Dans le scriptage existant, la commande `disconnect` est utilisée pour supprimer des liens entre les noeuds ; par exemple :

```
disconnect ^agg from ^select
```

Dans le scriptage Python, c'est la méthode `unlink` qui est utilisée pour supprimer des liens entre les noeuds ; par exemple :

```
stream.unlink(agg, select)
```

Dans le scriptage existant, la commande `position` est utilisée pour positionner les noeuds dans les canevas de flux ou entre d'autres noeuds ; par exemple :

```
position ^agg at 256 256
position ^agg between ^myselect and ^mydistinct
```

Dans le scriptage Python, on obtient le même résultat en utilisant deux méthodes distinctes : `setXYPosition` et `setPositionBetween`. Par exemple :

```
agg.setXYPosition(256, 256)
agg.setPositionBetween(myselect, mydistinct)
```

Opérations de noeud

Certaines commandes d'opération de noeud couramment utilisées dans IBM SPSS Modeler présentent des commandes équivalentes dans le scriptage Python. Cela peut vous aider à convertir vos scripts SPSS Modeler existants en scripts Python afin de les utiliser dans IBM SPSS Modeler 16.

Tableau 219. Mappage du scriptage existant au scriptage Python pour les opérations de noeud.

Scriptage existant	Scriptage Python
create <i>nodespec</i> at x y	<code>stream.create(type, nom)</code> <code>stream.createAt(type, nom, x, y)</code> <code>stream.createBetween(type, nom, preNode, postNode)</code> <code>stream.createModelApplier(modèle, nom)</code>
connect <i>noeud_source</i> to <i>noeud_cible</i>	<code>stream.link(noeud_source, noeud_cible)</code>
delete <i>noeud</i>	<code>stream.delete(noeud)</code>
disable <i>noeud</i>	<code>stream.setEnabled(noeud, False)</code>
enable <i>noeud</i>	<code>stream.setEnabled(noeud, True)</code>
disconnect <i>noeud_source</i> from <i>noeud_cible</i>	<code>stream.unlink(noeud_source, noeud_cible)</code> <code>stream.disconnect(noeud)</code>
duplicate <i>noeud</i>	<code>noeud.duplicate()</code>
execute <i>noeud</i>	<code>stream.runSelected(noeuds, résultats)</code> <code>stream.runAll(résultats)</code>
flush <i>noeud</i>	<code>noeud.flushCache()</code>
position <i>noeud</i> at x y	<code>noeud.setXYPosition(x, y)</code>

Tableau 219. Mappage du scriptage existant au scriptage Python pour les opérations de noeud (suite).

Scriptage existant	Scriptage Python
position <i>noeud</i> between <i>noeud1</i> and <i>noeud2</i>	<i>noeud.setPositionBetween(noeud1, noeud2)</i>
rename <i>noeud</i> as <i>nom</i>	<i>noeud.setLabel(nom)</i>

Bouclage

Dans le scriptage existant, il existe principalement deux options de bouclage prises en charge :

- Les boucles *comptabilisées* dans lesquelles une variable d'index varie entre deux limites entières.
- Les boucles *en séquence* qui forment une boucle via une séquence de valeurs, en liant la valeur en cours à la variable de boucle.

Le script suivant est un exemple de boucle comptabilisée dans le scriptage existant :

```
for i from 1 to 10
  println ^i
endfor
```

Le script suivant est un exemple de boucle en séquence dans le scriptage existant :

```
var items
set items = [a b c d]

for i in items
  println ^i
endfor
```

Il existe également d'autres types de boucles pouvant être utilisés :

- Itération par les modèles de la palette de modèles ou par les sorties de la palette de sorties.
- Itération par les champs entrant dans un noeud ou en sortant.

Le scriptage Python prend également en charge d'autres types de boucles. Le script suivant est un exemple de boucle comptabilisée dans le scriptage Python :

```
i = 1
while i <= 10:
  print i
  i += 1
```

Le script suivant est un exemple de boucle en séquence dans le scriptage Python :

```
items = ["a", "b", "c", "d"]
for i in items:
  print i
```

La boucle en séquence est très flexible et lorsqu'elle est combinée avec les méthodes API IBM SPSS Modeler, elle peut prendre en charge la majorité des cas d'utilisation du scriptage existant. L'exemple suivant montre comment utiliser une boucle en séquence dans le scriptage Python pour itérer via les champs sortant d'un noeud :

```
node = modeler.script.stream().findByType("filter", None)
for column in node.getOutputDataModel().columnIterator():
  print column.getColumnname()
```

exécution des flux

Durant l'exécution d'un flux, les objets de modèle ou de sortie générés sont ajoutés à l'un des gestionnaires d'objets. Dans le scriptage existant, le script doit soit localiser les objets générés dans le gestionnaire d'objets, soit accéder à la dernière sortie générée à partir du noeud ayant généré cette sortie.

L'exécution d'un flux est différente dans Python : tout objet de modèle ou de sortie généré par l'exécution est renvoyé dans une liste transmise à la fonction d'exécution. Cela simplifie l'accès aux résultats de l'exécution de flux.

Le scriptage existant prend en charge trois commandes d'exécution de flux :

- `execute_all` exécute tous les noeuds terminaux exécutables du flux.
- `execute_script` exécute le script de flux quelle que soit la configuration de l'exécution de script.
- `execute noeud` exécute le noeud spécifié.

Le scriptage Python prend en charge une série de fonctions similaire :

- `stream.runAll(liste_résultats)` exécute tous les noeuds terminaux exécutables du flux.
- `stream.runScript(liste_résultats)` exécute le script de flux quelle que soit la configuration de l'exécution de script.
- `stream.runSelected(grappe_de_noeuds, liste_résultats)` exécute l'ensemble de noeuds spécifié dans l'ordre dans lequel ils sont fournis.
- `stream.run(liste_résultats)` exécute le noeud spécifié.

Dans le scriptage existant, une exécution de flux peut être arrêtée via la commande `exit` avec un code entier facultatif ; par exemple :

```
exit 1
```

Dans le scriptage Python, on obtient le même résultat à l'aide du script suivant :

```
modeler.script.exit(1)
```

Accéder aux objets via le système de fichiers et le référentiel

Dans le scriptage existant, vous pouvez ouvrir un objet de flux, de modèle ou de sortie existant à l'aide de la commande `open` ; par exemple :

```
var s  
set s = open stream "c:/my streams/modeling.str"
```

Dans le scriptage Python, il existe une classe `TaskRunner` qui est accessible à partir de la session et qui peut être utilisée pour effectuer des tâches similaires ; par exemple :

```
taskrunner = modeler.script.session().getTaskRunner()  
s = taskrunner.openStreamFromFile("c:/my streams/modeling.str", True)
```

Pour enregistrer un objet dans le scriptage existant, vous pouvez utiliser la commande `save` ; par exemple :

```
save stream s as "c:/my streams/new_modeling.str"
```

L'approche de script Python équivalente consisterait à utiliser la classe `TaskRunner` ; par exemple :

```
taskrunner.saveStreamToFile(s, "c:/my streams/new_modeling.str")
```

Les opérations basées sur IBM SPSS Collaboration and Deployment Services Repository sont prises en charge dans le scriptage existant via les commandes `retrieve` et `store` ; par exemple :

```
var s  
set s = retrieve stream "/my repository folder/my_stream.str"  
store stream ^s as "/my repository folder/my_stream_copy.str"
```

Dans le scriptage Python, l'utilisateur accéderait à la fonctionnalité équivalente via l'objet Référentiel associé à la session ; par exemple :

```

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)

```

Remarque : L'accès via l'objet Référentiel requiert que la session ait été configurée avec une connexion de référentiel valide.

Opérations de flux

Certaines commandes d'opération de flux couramment utilisées dans IBM SPSS Modeler présentent des commandes équivalentes dans le scriptage Python. Cela peut vous aider à convertir vos scripts SPSS Modeler existants en scripts Python afin de les utiliser dans IBM SPSS Modeler 16.

Tableau 220. Mappage du scriptage existant au scriptage Python pour les opérations de flux.

Scriptage existant	Scriptage Python
create stream <i>NOMFICHIER_PAR_DEFAULT</i>	<i>taskrunner.createStream(name, autoConnect, autoManage)</i>
close stream	<i>stream.close()</i>
clear stream	<i>stream.clear()</i>
get stream <i>stream</i>	Pas d'équivalent
load stream <i>path</i>	Pas d'équivalent
open stream <i>path</i>	<i>taskrunner.openStreamFromFile(path, autoManage)</i>
save <i>stream</i> as <i>path</i>	<i>taskrunner.saveStreamToFile(stream, path)</i>
retrieive stream <i>path</i>	<i>repository.retrieveStream(path, version, label, autoManage)</i>
store <i>stream</i> as <i>path</i>	<i>repository.storeStream(stream, path, label)</i>

Opérations de modèle

Certaines commandes d'opération de modèle couramment utilisées dans IBM SPSS Modeler présentent des commandes équivalentes dans le scriptage Python. Cela peut vous aider à convertir vos scripts SPSS Modeler existants en scripts Python afin de les utiliser dans IBM SPSS Modeler 16.

Tableau 221. Mappage du scriptage existant au scriptage Python pour les opérations de modèle.

Scriptage existant	Scriptage Python
open model <i>chemin</i>	<i>taskrunner.openModelFromFile(path, autoManage)</i>
save <i>model</i> as <i>path</i>	<i>taskrunner.saveModelToFile(model, path)</i>
retrieve model <i>path</i>	<i>repository.retrieveModel(path, version, label, autoManage)</i>
store <i>model</i> as <i>path</i>	<i>repository.storeModel(model, path, label)</i>

Opérations de sortie de document

Certaines commandes d'opération de sortie de document couramment utilisées dans IBM SPSS Modeler présentent des commandes équivalentes dans le scriptage Python. Cela peut vous aider à convertir vos scripts SPSS Modeler existants en scripts Python afin de les utiliser dans IBM SPSS Modeler 16.

Tableau 222. Mappage du scriptage existant au scriptage Python pour les opérations de sortie de document.

Scriptage existant	Scriptage Python
open output <i>path</i>	<i>taskrunner.openDocumentFromFile(path, autoManage)</i>

Tableau 222. Mappage du scriptage existant au scriptage Python pour les opérations de sortie de document (suite).

Scriptage existant	Scriptage Python
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>

Autres différences entre le scriptage existant et le scriptage Python

Les scripts existants prennent en charge la manipulation des projets IBM SPSS Modeler. Les scripts Python ne prennent pas en charge cette fonction.

Le scriptage existant offre une prise en charge du chargement des objets d'état (combinaison de flux et de modèles). Ce type d'objet est devenu obsolète depuis la version 8.0 de IBM SPSS Modeler. Le scriptage Python ne prend pas en charge les objets d'état.

Le scriptage Python offre les fonctionnalités supplémentaires suivantes, qui ne sont pas disponibles dans le scriptage existant :

- Définitions de classe et de fonction
- Gestion des erreurs
- Support plus sophistiqué des entrées et sorties
- Modules externes et tiers

Remarques

Ces informations ont été développées pour les produits et services offerts dans le monde.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, programme ou service IBM n'implique pas que seul ce produit, programme ou service IBM puisse être utilisé. Tout produit, programme ou service fonctionnellement équivalent peut être utilisé s'il n'enfreint aucun droit de propriété intellectuelle d'IBM. Cependant l'utilisateur doit évaluer et vérifier l'utilisation d'un produit, programme ou service non IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. L'octroi de ce document n'équivaut aucunement à celui d'une licence pour ces brevets. Vous pouvez envoyer par écrit des questions concernant la licence à :

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada

Pour toute demande au sujet des licences concernant les jeux de caractères codés sur deux octets (DBCS), contactez le service Propriété intellectuelle IBM de votre pays ou adressez vos questions par écrit à :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japon

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certains états n'autorisent pas l'exclusion de garanties explicites ou implicites lors de certaines transactions, par conséquent, il est possible que cet énoncé ne vous concerne pas.

Ces informations peuvent contenir des erreurs techniques ou des erreurs typographiques. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Toute référence dans ces informations à des sites Web autres qu'IBM est fournie dans un but pratique uniquement et ne sert en aucun cas de recommandation pour ces sites Web. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation à votre égard, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Software Group
ATTN: Licensing
200 W. Madison St.
Chicago, IL; 60606
U.S.A.

Ces informations peuvent être disponibles, soumises à des conditions générales, et dans certains cas payantes.

Le programme sous licence décrit dans le présent document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Livret Contractuel IBM, des Conditions internationales d'utilisation des Logiciels IBM ou de tout autre contrat équivalent.

Toutes les données sur les performances contenues dans le présent document ont été obtenues dans un environnement contrôlé. Par conséquent, les résultats obtenus dans d'autres environnements d'exploitation peuvent varier de manière significative. Certaines mesures peuvent avoir été effectuées sur des systèmes en cours de développement et il est impossible de garantir que ces mesures seront les mêmes sur les systèmes commercialisés. De plus, certaines mesures peuvent avoir été estimées par extrapolation. Les résultats réels peuvent être différents. Les utilisateurs de ce document doivent vérifier les données applicables à leur environnement spécifique.

les informations concernant les produits autres qu'IBM ont été obtenues auprès des fabricants de ces produits, leurs annonces publiques ou d'autres sources publiques disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Aucune réclamation relative à des produits non IBM ne pourra être reçue par IBM. Les questions sur les capacités de produits autres qu'IBM doivent être adressées aux fabricants de ces produits.

Toutes les déclarations concernant la direction ou les intentions futures d'IBM peuvent être modifiées ou retirées sans avertissement préalable et représentent uniquement des buts et des objectifs.

Ces informations contiennent des exemples de données et de rapports utilisés au cours d'opérations quotidiennes standard. Pour les illustrer le mieux possible, ces exemples contiennent des noms d'individus, d'entreprises, de marques et de produits. Tous ces noms sont fictifs et toute ressemblance avec des noms et des adresses utilisés par une entreprise réelle ne serait que pure coïncidence.

Si vous consultez la version papier de ces informations, il est possible que certaines photographies et illustrations en couleurs n'apparaissent pas.

Marques

IBM, le logo IBM et ibm.com sont des marques d'International Business Machines dans de nombreux pays. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web «Copyright and trademark information» à l'adresse www.ibm.com/legal/copytrade.shtml.

Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, et Pentium sont des marques de Intel Corporation ou de ses filiales aux Etats-Unis et dans d'autres pays.

Linux est une marque déposée de Linus Torvalds aux Etats-Unis et/ou dans d'autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans d'autres pays.

UNIX est une marque déposée de The Open Group aux Etats-Unis et dans d'autres pays.

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques d'Oracle et/ou de ses sociétés affiliées.

Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés.

Index

A

- Affichage des modèles de séries temporelles ISW IBM
 - propriétés de génération de scripts de noeud 199
- ajout d'attributs 22
- API de scriptage
 - accès aux objets générés 38
 - exemple 35
 - Introduction 35
 - métadonnées 35
 - paramètres de flux 40
 - paramètres de session 40
 - paramètres du super noeud 40
 - plusieurs flux 45
 - recherche 35
 - scripts autonomes 45
 - traitement des erreurs 39
 - valeurs globales 44
- arbre de décision MS
 - propriétés de génération de scripts de noeud 189, 191
- arguments
 - connexion à IBM SPSS Collaboration and Deployment Services Repository 55
 - connexion au serveur 54
 - fichier de commande 55
 - système 52

B

- blocs de code 17
- bouclage dans les flux 4, 5

C

- caractères non ASCII 20
- chaînes 15
- champs
 - désactivation lors de la génération de scripts 119
- clé d'itération
 - bouclage dans les scripts 6
- commande clear generated palette 48
- création d'une classe 22
- création de noeuds 28, 29, 30

D

- définition d'attributs 23
- définition d'une classe 22
- définition de méthodes 23
- définition de propriétés 28
- détection d'erreurs
 - génération de scripts 48
- diagrammes 25

E

- exécution conditionnelle de flux 4, 8
- exécution de scripts 9
- exécution des flux 25
- exemples 18
- expressions régulières 10

F

- flux
 - bouclage 4, 5
 - commande de définition globale 57
 - exécution 25
 - exécution conditionnelle 4, 8
 - génération de scripts 1, 2, 25
 - modification 28
 - propriétés 59
- fonctions
 - bouclage 253
 - commandes conditionnelles 253
 - commentaires 252
 - littéraux 252
 - opérateurs 252
 - opérations de flux 259
 - opérations de modèle 259
 - opérations de noeud 256
 - opérations de sortie de document 259
 - références d'objet 252

G

- génération de scripts
 - à partir de la ligne de commande 48
- abréviations utilisées 57
- Aperçu 1
- bouclage visuel 4, 5
- chemins de fichier 48
- clé d'itération 6
- compatibilité avec les versions
 - antérieures 48
- contexte 26
- dans les super noeuds 3
- détection d'erreurs 48
- diagrammes 25
- exécution 9
- exécution conditionnelle 4, 8
- exécution de noeud de modélisation 47
- flux 1, 25
- flux super noeud 25
- interface utilisateur 3, 10
- interruption 9
- noeuds de sortie 217
- noeuds Graphiques 119
- ordre d'exécution de flux 47
- présentation 13
- propriétés communes 58
- remplacement de modèle 47
- scriptage existant 252, 253, 256, 259

- génération de scripts (*suite*)
 - scriptage Python 252, 253, 256, 259
 - scripts autonomes 1, 25
 - scripts de super noeud 1, 25
 - sélection de champs 8
 - syntaxe 14, 15, 17, 18, 19, 20, 21, 22, 23
 - variable d'itération 7

H

- héritage 23

I

- IBM SPSS Collaboration and Deployment Services Repository
 - arguments de ligne de commande 55
- IBM SPSS Modeler
 - démarrage à partir de la ligne de commande 51
- identificateurs 17
- indicateurs
 - arguments de ligne de commande 51
 - combinaison de plusieurs indicateurs 55
- instructions 17
- interruption de scripts 9

J

- Jython 13

L

- ligne de commande
 - arguments multiples 55
 - exécution d'IBM SPSS Modeler 51
 - génération de scripts 48
 - liste des arguments 52, 54, 55
 - paramètres 53
- listes 14

M

- méthodes mathématiques 19
- migration
 - accéder aux objets 258
 - bouclage 257
 - commandes 251
 - contexte de scriptage 251
 - définition de propriétés 255
 - différences générales 251
 - divers 260
 - édition de flux 255
 - exécution des flux 257
 - extraction de propriétés 255
 - fonctions 251
 - noms de propriétés 254

- migration (*suite*)
 - présentation 251
 - références de noeud 254
 - référentiel 258
 - système de fichiers 258
 - types de modèles 254
 - types de noeuds 254
 - types de sorties 254
 - variables 254
- mise en cluster de séquences MS
 - propriétés de génération de scripts de noeud 191
- Modèles
 - noms de scripts 247, 249
- modèles ACP
 - propriétés de génération de scripts de noeud 149, 183
- Modèles ACP/Analyse factorielle
 - propriétés de génération de scripts de noeud 149, 183
- Modèles ACP Netezza
 - propriétés de génération de scripts de noeud 205, 214
- modèles apriori
 - propriétés de génération de scripts de noeud 133, 179
- Modèles Apriori Oracle
 - propriétés de génération de scripts de noeud 193, 198
- Modèles Bayes Net Netezza
 - propriétés de génération de scripts de noeud 205, 214
- Modèles C5.0
 - propriétés de génération de scripts de noeud 140, 181
- Modèles CARMA
 - propriétés de génération de scripts de noeud 141, 181
- Modèles CHAID
 - propriétés de génération de scripts de noeud 143, 182
- modèles d'agrégation suivant le saut minimum
 - propriétés de génération de scripts de noeud 160
- Modèles d'arbre C&RT
 - propriétés de génération de scripts de noeud 142, 181
- Modèles d'arbre de décision ISW IBM
 - propriétés de génération de scripts de noeud 199, 204
- Modèles d'arbre de décision Netezza
 - propriétés de génération de scripts de noeud 205, 214
- Modèles d'arbre de régression Netezza
 - propriétés de génération de scripts de noeud 205, 214
- Modèles d'arbre décision Oracle
 - propriétés de génération de scripts de noeud 193, 198
- Modèles d'association ISW IBM
 - propriétés de génération de scripts de noeud 199, 204
- Modèles d'IA d'Oracle
 - propriétés de génération de scripts de noeud 193
- Modèles de classification ISW IBM
 - propriétés de génération de scripts de noeud 199, 204
- Modèles de classification par division Netezza
 - propriétés de génération de scripts de noeud 205, 214
- Modèles de cluster automatique
 - propriétés de génération de scripts de noeud 180
- modèles de détection des anomalies
 - propriétés de génération de scripts de noeud 131, 179
- Modèles de Discriminant automatique
 - propriétés de génération de scripts de noeud 180
- modèles de numérisation automatique
 - propriétés de génération de scripts de noeud 137
- Modèles de numérisation automatique
 - propriétés de génération de scripts de noeud 180
- Modèles de régression de Cox
 - propriétés de génération de scripts de noeud 145, 182
- Modèles de régression ISW IBM
 - propriétés de génération de scripts de noeud 199, 204
- modèles de régression linéaire
 - propriétés de génération de scripts de noeud 170, 187
- Modèles de régression linéaire Netezza
 - propriétés de génération de scripts de noeud 205, 214
- modèles de régression logistique
 - propriétés de génération de scripts de noeud 163, 185
- Modèles de régression logistique ISW IBM
 - propriétés de génération de scripts de noeud 199, 204
- Modèles de Réponse en auto-apprentissage
 - propriétés de génération de scripts de noeud 173, 187
- modèles de réseau Bayésien
 - propriétés de génération de scripts de noeud 138
- Modèles de réseau Bayésien
 - propriétés de génération de scripts de noeud 181
- modèles de réseau de neurones
 - propriétés de génération de scripts de noeud 166, 186
- Modèles de séquence ISW IBM
 - propriétés de génération de scripts de noeud 199, 204
- modèles de séquences
 - propriétés de génération de scripts de noeud 172, 188
- modèles de séries temporelles
 - propriétés de génération de scripts de noeud 174, 188
- Modèles de séries temporelles Netezza
 - propriétés de génération de scripts de noeud 205
- modèles discriminants
 - propriétés de génération de scripts de noeud 148, 183
- modèles générés
 - noms de scripts 247, 249
- Modèles IBM DB2
 - propriétés de génération de scripts de noeud 199
- Modèles IBM SPSS Statistics
 - propriétés de génération de scripts de noeud 242
- Modèles k moyenne
 - propriétés de génération de scripts de noeud 159, 185
- Modèles k moyenne Netezza
 - propriétés de génération de scripts de noeud 205, 214
- Modèles KMeans Oracle
 - propriétés de génération de scripts de noeud 193, 198
- modèles KNN
 - propriétés de génération de scripts de noeud 185
- Modèles KNN Netezza
 - propriétés de génération de scripts de noeud 205, 214
- modèles Kohonen
 - propriétés de génération de scripts de noeud 161
- Modèles Kohonen
 - propriétés de génération de scripts de noeud 185
- modèles linéaires
 - propriétés de génération de scripts de noeud 162, 185
- modèles linéaires généralisés
 - propriétés de génération de scripts de noeud 152, 184
- Modèles linéaires généralisés d'Oracle
 - propriétés de génération de scripts de noeud 193
- Modèles linéaires généralisés Netezza
 - propriétés de génération de scripts de noeud 205
- modèles Liste de décision
 - propriétés de génération de scripts de noeud 146, 183
- Modèles MDL Oracle
 - propriétés de génération de scripts de noeud 193, 198
- Modèles Microsoft
 - propriétés de génération de scripts de noeud 189, 191
- Modèles MMLG
 - propriétés de génération de scripts de noeud 155, 184
- Modèles Naive Bayes d'IBM ISW
 - propriétés de génération de scripts de noeud 199, 204
- Modèles Naive Bayes Netezza
 - propriétés de génération de scripts de noeud 205, 214
- Modèles Netezza
 - propriétés de génération de scripts de noeud 205

- Modèles NMF Oracle
 - propriétés de génération de scripts de nœud 193, 198
- modèles Oracle
 - propriétés de génération de scripts de nœud 193
- Modèles Oracle Adaptive Bayes
 - propriétés de génération de scripts de nœud 193, 198
- Modèles Oracle Naive Bayes
 - propriétés de génération de scripts de nœud 193, 198
- Modèles Oracle Support Vector Machines
 - propriétés de génération de scripts de nœud 193, 198
- Modèles QUEST
 - propriétés de génération de scripts de nœud 169, 186
- modèles sélection de fonction
 - propriétés de génération de scripts de nœud 150, 183
- Modèles SLRM
 - propriétés de génération de scripts de nœud 173, 187
- modèles support vector machine
 - propriétés de génération de scripts de nœud 188
- Modèles Support vector machine
 - propriétés de génération de scripts de nœud 174
- Modèles SVM
 - propriétés de génération de scripts de nœud 174
- modèles TwoStep
 - propriétés de génération de scripts de nœud 176, 188
- modélisation de bases de données 189
- modification de flux 28, 31
- mot-clé generated 48
- mots de passe
 - ajout aux scripts 47
 - codés 54
- mots de passe codés
 - ajout aux scripts 47

N

- nœud Analyse
 - propriétés 217
- Nœud Analyse RFM
 - propriétés 107
- Nœud Anonymiser
 - propriétés 95
- nœud Binariser
 - propriétés 108
- nœud Délimiter
 - propriétés 78
- nœud Discrétiser
 - propriétés 98
- Nœud Ensemble
 - propriétés 102
- nœud export Excel
 - propriétés 236
- nœud Export SAS
 - propriétés 238
- Nœud Export XML
 - propriétés 238
- nœud Fichier plat
 - propriétés 237
- nœud Fixe
 - propriétés 71
- nœud Historiser
 - propriétés 104
- nœud Partitionner
 - propriétés 104
- nœud Re-trier
 - propriétés 106
- nœud Restructurer
 - propriétés 106
- Nœud SGBD.
 - propriétés 67
- Nœud source SAS
 - propriétés 74
- nœud Transposer
 - propriétés 113
- nœuds d'exportation
 - propriétés de génération de scripts de nœud 231
- nœuds de sortie
 - propriétés de génération de scripts 217
- noeud Agréger
 - propriétés 83
- Noeud Agréger RFM
 - propriétés 87
- noeud Ajouter
 - propriétés 83
- noeud Ajustement de simulation
 - propriétés 224
- Noeud Audit données
 - propriétés 218
- Noeud cluster automatique
 - propriétés de génération de scripts de nœud 136
- noeud Courbes
 - propriétés 126
- noeud d'export IBM SPSS Data Collection
 - propriétés 236
- noeud d'export IBM SPSS Statistics
 - propriétés 242
- noeud de création R
 - propriétés de génération de scripts de nœud 139
- noeud de processus R
 - propriétés 89
- noeud de sortie IBM SPSS Statistics
 - propriétés 242
- noeud de sortie R
 - propriétés 222
- noeud Dérivée
 - propriétés 101
- Noeud Discriminant automatique
 - propriétés de génération de scripts de nœud 134
- noeud Distinguer
 - propriétés 86
- noeud distribution
 - propriétés 121
- noeud Echantillon
 - propriétés 89
- Noeud Enterprise View
 - propriétés 71
- noeud Equilibrer
 - propriétés 84
- noeud Evaluation
 - propriétés 121
- noeud Evaluation de simulation
 - propriétés 223
- noeud exportation de base de données
 - propriétés 233
- Noeud Filtrer
 - propriétés 103
- noeud Fusionner
 - propriétés 86
- noeud Génération de simulation
 - propriétés 74
- noeud Histogramme
 - propriétés 125
- noeud Intervalles de temps
 - propriétés 109
- noeud Matrice
 - propriétés 219
- noeud Moyennes
 - propriétés 220
- noeud Rapport
 - propriétés 222
- noeud Re-trier
 - propriétés 106
- noeud Recoder
 - propriétés 105
- noeud Relations
 - propriétés 130
- noeud Relations orientées
 - propriétés 130
- noeud Remplacer
 - propriétés 103
- Noeud Représentation Graphique
 - propriétés 123
- noeud Résumé
 - propriétés 120
- noeud Sélectionner
 - propriétés 91
- noeud source Analytic Server
 - propriétés 65
- Noeud source Excel
 - propriétés 70
- Noeud source IBM Cognos BI
 - propriétés 65
- noeud source IBM SPSS Data Collection
 - propriétés 68
- noeud source IBM SPSS Statistics
 - propriétés 241
- Noeud source XML
 - propriétés 81
- noeud Statistiques
 - propriétés 225
- noeud stb (cases-espace-temps)
 - propriétés 84
- noeud streaminggts (série temporelle de diffusion en flux)
 - propriétés 91
- noeud Table
 - propriétés 226
- noeud Tracé
 - propriétés 127
- noeud Tracé horaire
 - propriétés 129
- noeud Transformation
 - propriétés 228
- noeud Transformation IBM SPSS Statistics
 - propriétés 241

- noeud Trier
 - propriétés 91
- noeud type
 - propriétés 113
- Noeud Utilisateur
 - propriétés 77
- noeud V. globales
 - propriétés 223
- noeuds
 - création de liens entre les noeuds 29
 - importation 30
 - informations 32
 - référence des noms 247
 - remplacement 30
 - suppression 30
 - suppression de liens entre les noeuds 29
- noeuds de modélisation
 - propriétés de génération de scripts de noeud 131
- noeuds Graphiques
 - propriétés de génération de scripts 119
- noeuds source
 - propriétés 63
- nuggets
 - propriétés de génération de scripts de noeud 179
- nuggets de modèle
 - noms de scripts 247, 249
 - propriétés de génération de scripts de noeud 179

O

- O-Cluster Oracle
 - propriétés de génération de scripts de noeud 193, 198
- objets de modèle
 - noms de scripts 247, 249
- objets de sortie
 - noms de scripts 249
- opérations 14
- ordre d'exécution
 - changement à l'aide de scripts 47
- ordre d'exécution de flux
 - changement à l'aide de scripts 47
- orienté objet 21

P

- paramètres 3, 57, 59
 - génération de scripts 14
 - Super noeuds 245
- paramètres de propriété 3, 57, 58
- préparation automatique des données
 - propriétés 95
- propriétés
 - flux 59
 - génération de scripts 57, 58, 131, 179, 231
 - génération de scripts commune 58
 - noeuds de modélisation de base de données 189
 - Super noeuds 245
 - propriétés applyr 187

- propriétés buildr 139
- propriétés de flatfilenode 237
- propriétés de génération de scripts de noeud 189
 - noeuds d'exportation 231
 - noeuds de modélisation 131
 - nuggets de modèle 179
- propriétés du noeud ACP/Facteur 149
- propriétés du noeud Agréger 83
- propriétés du noeud Ajouter 83
- propriétés du noeud Analyse 217
- propriétés du noeud
 - anomalydetection 131
- propriétés du noeud Anonymiser 95
- propriétés du noeud
 - applyanomalydetection 179
- propriétés du noeud applyapriori 179
- propriétés du noeud
 - applyautoclassifier 180
- propriétés du noeud
 - applyautocluster 180
- propriétés du noeud
 - applyautonumeric 180
- propriétés du noeud applybayesnet 181
- propriétés du noeud applyc50 181
- propriétés du noeud applycarma 181
- propriétés du noeud applycart 181
- propriétés du noeud applychaid 182
- propriétés du noeud applycoxreg 182
- propriétés du noeud
 - applydb2imcluster 204
- propriétés du noeud applydb2imlog 204
- propriétés du noeud applydb2imnb 204
- propriétés du noeud applydb2imreg 204
- propriétés du noeud
 - applydb2imtree 204
- propriétés du noeud
 - applydecisionlist 183
- propriétés du noeud
 - applydiscriminant 183
- propriétés du noeud applyfactor 183
- propriétés du noeud
 - applyfeatureselection 183
- propriétés du noeud
 - applygeneralizedlinear 184
- propriétés du noeud applyglm 184
- propriétés du noeud applykmeans 185
- propriétés du noeud applyknn 185
- propriétés du noeud applykohonen 185
- propriétés du noeud applylinear 185
- propriétés du noeud applylogreg 185
- propriétés du noeud
 - applymslogistic 191
- propriétés du noeud
 - applymsneuralnetwork 191
- propriétés du noeud
 - applymsregression 191
- propriétés du noeud
 - applymssequencecluster 191
- propriétés du noeud
 - applymstimeseries 191
- propriétés du noeud applymstree 191
- propriétés du noeud
 - applynetzezbayes 214
- propriétés du noeud
 - applynetzezadectree 214

- propriétés du noeud
 - applynetzezzadivcluster 214
- propriétés du noeud
 - applynetzezzakmeans 214
- propriétés du noeud
 - applynetzezzaknn 214
- propriétés du noeud
 - applynetzezzalineregression 214
- propriétés du noeud
 - applynetzezzanaivebayes 214
- propriétés du noeud
 - applynetzezzapca 214
- propriétés du noeud
 - applynetzezzaregtree 214
- propriétés du noeud applyneuralnet 186
- propriétés du noeud
 - applyneuralnetwork 186
- propriétés du noeud applyoraabn 198
- propriétés du noeud
 - applyoradecisiontree 198
- propriétés du noeud
 - applyorakmeans 198
- propriétés du noeud applyoranb 198
- propriétés du noeud applyoranmf 198
- propriétés du noeud
 - applyoracluster 198
- propriétés du noeud applyorasvm 198
- propriétés du noeud applyquest 186
- propriétés du noeud
 - applyregression 187
- propriétés du noeud
 - applyselflearning 187
- propriétés du noeud applysequence 188
- propriétés du noeud applysvm 188
- propriétés du noeud
 - applytimeseries 188
- propriétés du noeud applytwestep 188
- propriétés du noeud apriori 133
- propriétés du noeud asexport 231
- propriétés du noeud asimport 65
- propriétés du noeud autoclassifier 134
- propriétés du noeud autocluster 136
- propriétés du noeud autodataprep 95
- propriétés du noeud autonumeric 137
- propriétés du noeud bayesnet 138
- propriétés du noeud c50 140
- propriétés du noeud Carma 141
- propriétés du noeud cart 142
- propriétés du noeud chaid 143
- propriétés du noeud cognosimport 65
- propriétés du noeud Courbes 126
- propriétés du noeud coxreg 145
- propriétés du noeud dataaudit 218
- propriétés du noeud databaseexport 233
- propriétés du noeud
 - datacollectionexport 236
- propriétés du noeud
 - datacollectionimport 68
- propriétés du noeud db2imassoc 199
- propriétés du noeud db2imcluster 199
- propriétés du noeud db2imlog 199
- propriétés du noeud db2imnb 199
- propriétés du noeud db2imreg 199
- propriétés du noeud db2imsequence 199
- propriétés du noeud
 - db2imtimeseries 199
- propriétés du noeud db2imtree 199

- propriétés du noeud decisionlist 146
- propriétés du noeud derive_stb 84
- propriétés du noeud Dériver 101
- propriétés du noeud directedweb 130
- propriétés du noeud Discrétiser 98
- propriétés du noeud discriminant 148
- propriétés du noeud Distinguer 86
- propriétés du noeud distribution 121
- propriétés du noeud Echantillon 89
- propriétés du noeud Ensemble 102
- propriétés du noeud Equilibrer 84
- propriétés du noeud Evaluation 121
- propriétés du noeud Export 71
- propriétés du noeud excelexport 236
- propriétés du noeud excelimport 70
- propriétés du noeud
 - featureselection 150
- propriétés du noeud Filtrer 103
- propriétés du noeud fixedfile 71
- propriétés du noeud Fusionner 86
- propriétés du noeud genlin 152
- propriétés du noeud Histogramme 125
- propriétés du noeud Historiser 104
- propriétés du noeud kmeans 159
- propriétés du noeud knn 160
- propriétés du noeud Kohonen 161
- propriétés du noeud Linéaire 162
- propriétés du noeud logreg 163
- propriétés du noeud Matrice 219
- propriétés du noeud MMLG 155
- propriétés du noeud Moyennes 220
- propriétés du noeud MRAA 173
- propriétés du noeud msassoc 189
- propriétés du noeud msbayes 189
- propriétés du noeud mscluster 189
- propriétés du noeud mslogistic 189
- propriétés du noeud
 - msneuralnetwork 189
- propriétés du noeud msregression 189
- propriétés du noeud
 - mssequencecluster 189
- propriétés du noeud mstimeseries 189
- propriétés du noeud mstree 189
- propriétés du noeud netezababes 205
- propriétés du noeud netezadectree 205
- propriétés du noeud
 - netezadivcluster 205
- propriétés du noeud netezzaglm 205
- propriétés du noeud netezzakmeans 205
- propriétés du noeud netezzaknn 205
- propriétés du noeud
 - netezalineregression 205
- propriétés du noeud
 - netezanaivebayes 205
- propriétés du noeud netezzapca 205
- propriétés du noeud netezzaregtree 205
- propriétés du noeud
 - netezatimeseries 205
- propriétés du noeud neuralnet 166
- propriétés du noeud neuralnetwork 168
- propriétés du noeud
 - numericpredictor 137
- propriétés du noeud oraabn 193
- propriétés du noeud oraai 193
- propriétés du noeud oraapriori 193
- propriétés du noeud oradecisiontree 193
- propriétés du noeud oraglm 193

- propriétés du noeud orakmeans 193
- propriétés du noeud oramdl 193
- propriétés du noeud oranb 193
- propriétés du noeud oranmf 193
- propriétés du noeud oraoccluster 193
- propriétés du noeud orasvm 193
- propriétés du noeud outputfile 237
- propriétés du noeud Partitionner 104
- propriétés du noeud quest 169
- propriétés du noeud Rapport 222
- propriétés du noeud Re-trier 106
- propriétés du noeud Recoder 105
- propriétés du noeud Régression 170
- propriétés du noeud Relations 130
- propriétés du noeud Remplacer 103
- propriétés du noeud Représentation
 - Graphique 123
- propriétés du noeud Restructurer 106
- propriétés du noeud Résumé 120
- propriétés du noeud rfmaggregate 87
- propriétés du noeud rfmanalysis 107
- propriétés du noeud Routput 222
- propriétés du noeud Rprocessnode 89
- propriétés du noeud sasexport 238
- propriétés du noeud sasimport 74
- propriétés du noeud Sélectionner 91
- propriétés du noeud Séquence 172
- propriétés du noeud settoflag 108
- propriétés du noeud SGBD 67
- propriétés du noeud simeval 223
- propriétés du noeud simfit 224
- propriétés du noeud simgen 74
- propriétés du noeud statisticsexport 242
- propriétés du noeud statisticsimport 241
- propriétés du noeud statisticsmodel 242
- propriétés du noeud statisticsoutput 242
- propriétés du noeud
 - statisticstransform 241
- propriétés du noeud Statistiques 225
- propriétés du noeud streamings 91
- propriétés du noeud svm 174
- propriétés du noeud Table 226
- propriétés du noeud timeintervals 109
- propriétés du noeud timeplot 129
- propriétés du noeud timeseries 174
- propriétés du noeud Tracé 127
- propriétés du noeud Transformation 228
- propriétés du noeud Transposer 113
- propriétés du noeud Trier 91
- propriétés du noeud twostep 176
- propriétés du noeud type 113
- propriétés du noeud userinput 77
- propriétés du noeud V. globales 223
- propriétés du noeud variablefile 78
- propriétés du noeud xmllexport 238
- propriétés du noeud xmlimport 81
- Python 13
 - génération de scripts 14

R

- recherche de noeuds 27
- rechercher et remplacer 10
- référence aux noeuds 27
 - définition de propriétés 28
 - recherche de noeuds 27

- Régression linéaire MS
 - propriétés de génération de scripts de noeud 189, 191
- Régression logistique MS
 - propriétés de génération de scripts de noeud 189, 191
- remarques 17
- réseau neuronal MS
 - propriétés de génération de scripts de noeud 189, 191
- réseaux de neurones
 - propriétés de génération de scripts de noeud 168, 186

S

- scriptage
 - interface utilisateur 2
- scripts
 - bouclage 4, 5
 - clé d'itération 6
 - enregistrement 2
 - exécution conditionnelle 4, 8
 - importation à partir de fichiers
 - texte 2
 - sélection de champs 8
 - variable d'itération 7
- scripts autonomes 1, 3, 25
- sécurité
 - mots de passe codés 47, 54
- Séries temporelles MS
 - propriétés de génération de scripts de noeud 191
- serveur
 - arguments de ligne de commande 54
- super noeud 57
 - flux 25
- super noeuds
 - flux 25
- Super noeuds
 - définition de propriétés dans 245
 - génération de scripts 245
 - paramètres 245
 - propriétés 245
 - scripts 1, 3, 25
- système
 - arguments de ligne de commande 52

T

- transmission d'arguments 18
- traversée des noeuds 31

V

- variable d'itération
 - bouclage dans les scripts 7
- Variables
 - génération de scripts 14
- variables masquées 23

