

*Manual de scripts y automatización  
Python de IBM SPSS Modeler 16*

**IBM**

**Nota**

Antes de utilizar esta información y el producto al que da soporte, lea la información del apartado "Avisos" en la página 259.

**Información del producto**

Esta edición se aplica a la versión 16, release 0, modificación 0 de IBM SPSS Modeler y a todos los releases y las modificaciones subsiguientes hasta que se indique lo contrario en nuevas ediciones.

# Contenido

## Capítulo 1. Scripts . . . . . 1

Conceptos básicos del uso de scripts . . . . .	1
Tipos de scripts . . . . .	1
Scripts de ruta . . . . .	2
Scripts autónomos . . . . .	3
Scripts de Supernodo . . . . .	3
Creación de bucles y ejecución condicional en rutas . . . . .	4
Bucles en rutas . . . . .	5
Ejecución condicional en rutas . . . . .	8
Ejecución e interrupción de scripts . . . . .	9
Buscar y reemplazar . . . . .	10

## Capítulo 2. Language de scripts . . . . . 13

Conceptos básicos del lenguaje de scripts . . . . .	13
Python y Jython . . . . .	13
Creación de scripts Python . . . . .	14
Operaciones . . . . .	14
Listas . . . . .	14
Cadenas . . . . .	15
Comentarios . . . . .	16
Sintaxis de las sentencias . . . . .	17
Identificadores . . . . .	17
Bloques de código . . . . .	17
Pasar argumentos a un script . . . . .	18
Ejemplos . . . . .	18
Métodos matemáticos . . . . .	19
Utilización de caracteres no ASCII . . . . .	20
Programación orientada a objetos . . . . .	21
Definición de una clase . . . . .	22
Creación de una instancia de clase . . . . .	22
Añadir atributos a una instancia de clase . . . . .	22
Definición de atributos de clase y métodos . . . . .	23
Variables ocultas . . . . .	23
Herencia . . . . .	23

## Capítulo 3. Scripts de IBM SPSS Modeler . . . . . 25

Tipos de scripts . . . . .	25
Rutas, rutas de supernodo y diagramas . . . . .	25
Rutas . . . . .	25
Rutas de Supernodo . . . . .	25
Diagramas . . . . .	25
Ejecución de una ruta . . . . .	25
El contexto de los scripts . . . . .	26
Referencia a nodos existentes . . . . .	27
Buscar nodos . . . . .	27
Definición de propiedades . . . . .	28
Creación de nodos y modificación de rutas . . . . .	29
Creación de nodos . . . . .	29
Enlazar y desenlazar nodos . . . . .	29
Importar, sustituir y eliminar nodos . . . . .	31
Atravesar los nodos de una ruta . . . . .	31
Obtener información sobre los nodos . . . . .	32

## Capítulo 4. API de scripts. . . . . 35

Introducción a la API de scripts . . . . .	35
Ejemplo: Buscar nodos utilizando un filtro personalizado . . . . .	35
Metadatos: información sobre datos . . . . .	35
Acceso a objetos generados . . . . .	38
Tratamiento de errores. . . . .	39
Parámetros de ruta, sesión y Supernodo . . . . .	40
Valores globales . . . . .	44
Trabajar con varias rutas: Scripts autónomos . . . . .	45

## Capítulo 5. Sugerencias sobre scripts 47

Modificación de la ejecución de una ruta . . . . .	47
Trabajo con modelos . . . . .	47
Generación de una contraseña codificada . . . . .	47
Comprobación por script . . . . .	48
Scripts desde la línea de comandos . . . . .	48
Especificación de rutas de archivos . . . . .	48
Compatibilidad con versiones anteriores. . . . .	48

## Capítulo 6. Argumentos de la línea de comandos . . . . . 51

Invocación del software . . . . .	51
Uso de argumentos en la línea de comandos . . . . .	51
Argumentos del sistema . . . . .	52
Argumentos de parámetros . . . . .	53
Argumentos de conexión con el servidor IBM SPSS Collaboration and Deployment Services Repository Argumentos de conexión . . . . .	54
Combinación de varios argumentos . . . . .	55

## Capítulo 7. Referencia de propiedades 57

Conceptos básicos de referencia de propiedades . . . . .	57
Abreviaturas . . . . .	57
Ejemplos de propiedades de nodos y rutas . . . . .	57
Conceptos básicos de las propiedades de nodos . . . . .	58
Propiedades de nodos comunes . . . . .	58

## Capítulo 8. Propiedades de ruta . . . . . 59

## Capítulo 9. Propiedades de nodos de origen . . . . . 63

Propiedades comunes de nodos de origen . . . . .	63
Propiedades del nodo asimport. . . . .	64
Propiedades del nodo cognosimport . . . . .	65
Propiedades del nodo database. . . . .	66
Propiedades del nodo datacollectionimport. . . . .	67
Propiedades del nodo excelimport. . . . .	70
Propiedades del nodo evimport . . . . .	70
Propiedades del nodo fixedfile . . . . .	71
Propiedades del nodo sasimport . . . . .	73
Propiedades del nodo simgen . . . . .	73
Propiedades del nodo statisticsimport . . . . .	76
Propiedades del nodo userinput . . . . .	76

Propiedades del nodo variablefile . . . . .	77
Propiedades del nodo xmlimport . . . . .	80

## Capítulo 10. Propiedades de nodos de operaciones con registros . . . . . 81

Propiedades del nodo append . . . . .	81
Propiedades del nodo aggregate . . . . .	81
Propiedades del nodo balance . . . . .	82
Propiedades del nodo derive_stb . . . . .	82
Propiedades del nodo distinct . . . . .	84
Propiedades del nodo merge . . . . .	84
Propiedades del nodo rfmaggregate . . . . .	85
Propiedades del nodo Rprocess . . . . .	87
Propiedades del nodo sample . . . . .	87
Propiedades del nodo select . . . . .	88
Propiedades del nodo sort . . . . .	89
Propiedades del nodo Generación de análisis TS (streamingts) . . . . .	89

## Capítulo 11. Propiedades de nodos de operaciones con campos . . . . . 93

Propiedades del nodo anonymize . . . . .	93
Propiedades del nodo autodataprep . . . . .	93
Propiedades del nodo binning . . . . .	96
Propiedades del nodo derive . . . . .	99
Propiedades del nodo ensemble . . . . .	100
Propiedades del nodo filler . . . . .	101
Propiedades del nodo filter . . . . .	101
Propiedades del nodo history . . . . .	102
Propiedades del nodo partition . . . . .	102
Propiedades del nodo reclassify . . . . .	103
Propiedades del nodo reorder . . . . .	104
Propiedades del nodo restructure . . . . .	104
Propiedades del nodo rfmanalysis . . . . .	105
Propiedades del nodo settoflag . . . . .	106
Propiedades del nodo statistictransform . . . . .	106
Propiedades del nodo timeintervals . . . . .	107
Propiedades del nodo transpose . . . . .	111
Propiedades del nodo type . . . . .	111

## Capítulo 12. Propiedades de nodos Gráfico . . . . . 117

Propiedades comunes de nodos Gráfico . . . . .	117
Propiedades del nodo collection . . . . .	118
Propiedades del nodo distribution . . . . .	118
Propiedades del nodo evaluation . . . . .	119
Propiedades del nodo graphboard . . . . .	120
Propiedades del nodo histogram . . . . .	122
Propiedades del nodo multiplot . . . . .	123
Propiedades del nodo plot . . . . .	124
Propiedades del nodo timeplot . . . . .	126
Propiedades del nodo web . . . . .	127

## Capítulo 13. Propiedades de nodos de modelado . . . . . 129

Propiedades comunes de nodos de modelado . . . . .	129
Propiedades del nodo anomalydetection . . . . .	129
Propiedades del nodo apriori . . . . .	131
Propiedades del nodo autoclassifier . . . . .	132

Propiedades de ajustes de algoritmo . . . . .	133
Propiedades del nodo autocluster . . . . .	134
Propiedades del nodo autonumeric . . . . .	135
Propiedades del nodo bayesnet . . . . .	136
Propiedades del nodo buildr . . . . .	137
Propiedades del nodo c50 . . . . .	138
Propiedades del nodo carma . . . . .	139
Propiedades del nodo cart . . . . .	140
Propiedades del nodo chaid . . . . .	141
Propiedades del nodo coxreg . . . . .	143
Propiedades del nodo decisionlist . . . . .	145
Propiedades del nodo discriminant . . . . .	146
Propiedades del nodo factor . . . . .	147
Propiedades del nodo featureselection . . . . .	149
Propiedades del nodo genlin . . . . .	150
Propiedades del nodo glm . . . . .	153
Propiedades del nodo kmeans . . . . .	157
Propiedades del nodo knn . . . . .	158
Propiedades del nodo kohonen . . . . .	159
Propiedades del nodo linear . . . . .	160
Propiedades del nodo logreg . . . . .	161
Propiedades del nodo neuralnet . . . . .	164
Propiedades del nodo neuralnetwork . . . . .	166
Propiedades del nodo quest . . . . .	167
Propiedades del nodo regression . . . . .	169
Propiedades del nodo sequece . . . . .	170
Propiedades del nodo slrm . . . . .	171
Propiedades del nodo statisticsmodel . . . . .	172
Propiedades del nodo svm . . . . .	172
Propiedades del nodo timeseries . . . . .	173
Propiedades del nodo twostep . . . . .	175

## Capítulo 14. Propiedades de nodos de nugget de modelo . . . . . 177

Propiedades del nodo applyanomalydetection . . . . .	177
Propiedades del nodo applyapriori . . . . .	177
Propiedades del nodo applyautoclassifier . . . . .	178
Propiedades del nodo applyautocluster . . . . .	178
Propiedades del nodo applyautonumeric . . . . .	178
Propiedades del nodo applybayesnet . . . . .	179
Propiedades de nodo de applyc50 . . . . .	179
Propiedades del nodo applycarma . . . . .	179
Propiedades del nodo applycart . . . . .	179
Propiedades del nodo applychaid . . . . .	180
Propiedades del nodo applycoxreg . . . . .	180
Propiedades del nodo applydecisionlist . . . . .	181
Propiedades del nodo applydiscriminant . . . . .	181
Propiedades del nodo applyfactor . . . . .	181
Propiedades del nodo applyfeatureselection . . . . .	181
Propiedades del nodo applygeneralizedlinear . . . . .	182
Propiedades del nodo applyglm . . . . .	182
Propiedades del nodo applykmeans . . . . .	182
Propiedades del nodo applyknn . . . . .	183
Propiedades del nodo applykohonen . . . . .	183
Propiedades del nodo applylinear . . . . .	183
Propiedades del nodo applylogreg . . . . .	183
Propiedades del nodo applyneuralnet . . . . .	183
Propiedades del nodo applyneuralnetwork . . . . .	184
Propiedades del nodo applyquest . . . . .	184
Propiedades del nodo applyregression . . . . .	185
Propiedades del nodo applyr . . . . .	185

Propiedades del nodo applyselflearning . . . . .	185
Propiedades del nodo applysequence . . . . .	186
Propiedades del nodo applysvm . . . . .	186
Propiedades del nodo applytimeseries . . . . .	186
Propiedades del nodo applytwestep . . . . .	186

## Capítulo 15. Propiedades de nodos de modelado de bases de datos . . . . . 187

Propiedades de nodos de modelado de Microsoft	187
Propiedades de nodos de modelado de Microsoft . . . . .	187
Propiedades de nugget de modelo de Microsoft	189
Propiedades de nodos de modelado de Oracle . . . . .	191
Propiedades de nodos de modelado de Oracle	191
Propiedades de nugget de modelo de Oracle	196
Propiedades de nodos para modelado de IBM DB2	197
Propiedades de nodos de modelado de IBM DB2 . . . . .	197
Propiedades de nugget de modelo de IBM DB2	202
Propiedades de nodos de modelado de IBM Netezza Analytics . . . . .	203
Propiedades de nodos de modelado de Netezza	203
Propiedades de nugget de modelo de Netezza	213

## Capítulo 16. Propiedades de los nodos de resultados . . . . . 215

Propiedades del nodo analysis . . . . .	215
Propiedades del nodo dataaudit . . . . .	216
Propiedades del nodo matrix . . . . .	217
Propiedades del nodo means . . . . .	218
Propiedades del nodo report . . . . .	220
Propiedades del nodo Routput . . . . .	220
Propiedades del nodo setglobals . . . . .	221
Propiedades de nodo simeval . . . . .	221
Propiedades del nodo simfit . . . . .	222
Propiedades del nodo statistics . . . . .	223
Propiedades del nodo statisticsoutput . . . . .	224
Propiedades del nodo table . . . . .	224
Propiedades del nodo transform . . . . .	226

## Capítulo 17. Propiedades de nodos Exportar . . . . . 229

Propiedades de nodos Exportar comunes . . . . .	229
Propiedades del nodo asexport . . . . .	229
Propiedades del nodo cognosexport . . . . .	230
Propiedades del nodo databaseexport . . . . .	231
Propiedades del nodo datacollectionexport . . . . .	234
Propiedades del nodo excelexport . . . . .	234
Propiedades del nodo outputfile . . . . .	235
Propiedades del nodo sasexport . . . . .	236
Propiedades de nodo statisticsexport . . . . .	236

Propiedades del nodo xmlexport . . . . .	236
--	-----

## Capítulo 18. Propiedades de nodos de IBM SPSS Statistics . . . . . 239

Propiedades del nodo statisticsimport . . . . .	239
Propiedades del nodo statisticstransform . . . . .	239
Propiedades del nodo statisticsmodel . . . . .	240
Propiedades del nodo statisticsoutput . . . . .	240
Propiedades de nodo statisticsexport . . . . .	240

## Capítulo 19. Propiedades de Supernodos . . . . . 243

### Apéndice A. Referencia de nombres de nodo. . . . . 245

Nombres de nugget de modelo . . . . .	245
Evitar nombres duplicados del modelo . . . . .	247
Nombres de tipo de resultados . . . . .	247

### Apéndice B. Migración desde scripts de herencia a scripts Python. . . . . 249

Visión general de la migración de scripts de herencia . . . . .	249
Diferencias generales . . . . .	249
El contexto de los scripts . . . . .	249
Comparativa de comandos y funciones . . . . .	249
Literales y comentarios . . . . .	250
Operadores . . . . .	250
Comandos condicionales y de bucle . . . . .	251
Variables . . . . .	252
Tipos modelo, resultado y nodo . . . . .	252
Nombres de propiedades . . . . .	252
Referencias de nodos . . . . .	252
Obtener y establecer propiedades . . . . .	253
Edición de rutas . . . . .	253
Operaciones de nodo . . . . .	254
Bucle . . . . .	255
Ejecución de rutas . . . . .	255
Acceso a objetos mediante el sistema de archivos y el repositorio . . . . .	256
Operaciones de ruta . . . . .	257
Operaciones de modelo . . . . .	257
Operaciones de resultado de documento . . . . .	257
Otras diferencias entre scripts heredados y scripts Python . . . . .	258

### Avisos . . . . . 259

Marcas comerciales . . . . .	260
------------------------------	-----

### Índice. . . . . 263



---

# Capítulo 1. Scripts

---

## Conceptos básicos del uso de scripts

El procesamiento en IBM® SPSS Modeler es una herramienta potente para automatizar procesos en la interfaz de usuario. Los scripts pueden realizar los mismos tipos de acciones que se realizan con el ratón o el teclado y se utilizan para automatizar tareas que resultarían extremadamente repetitivas o llevarían mucho tiempo si se realizaran manualmente.

Puede utilizar los scripts para:

- Imponer un determinado orden en la ejecución de los nodos de una ruta y ejecutar nodos de forma condicional dependiendo de si se cumplen las condiciones de ejecución.
- Crear bucles para ejecutar nodos de forma repetida en una ruta.
- Especificar una secuencia automática de acciones que normalmente implican la interacción del usuario (por ejemplo, puede generar un modelo y comprobarlo a continuación).
- Configurar procesos complejos que requieren una interacción sustancial del usuario, como los procedimientos de validación cruzada que requieren una repetitiva generación y comprobación de modelo.
- Configurar procesos que manipulen rutas; por ejemplo, puede tomar una ruta de entrenamiento de modelo, ejecutarla y producir la ruta de comprobación del modelo automáticamente.

Este capítulo proporciona descripciones de alto nivel y ejemplos de scripts de nivel de ruta, scripts autónomos y scripts en Supernodos en la interfaz de IBM SPSS Modeler. Para obtener más información sobre el lenguaje, la sintaxis y los comandos de scripts, consulte los capítulos siguientes.<sup>1</sup>

*Note:* no puede importar y ejecutar scripts creados en IBM SPSS Statistics de IBM SPSS Modeler.

---

## Tipos de scripts

IBM SPSS Modeler utiliza tres tipos de scripts:

- Los **scripts de la ruta** se guardan como una propiedad de ruta y se guardan y se cargan con una ruta específica. Por ejemplo, puede escribir un script de ruta que automatice el proceso de entrenamiento y aplicación de un nugget de modelo. También puede especificar que cuando se ejecute una ruta particular, se ejecute el script, en lugar del contenido del lienzo de la ruta.
- Los **scripts autónomos** no están asociados a ninguna ruta en particular y se guardan en archivos de texto externos. Puede utilizar un script autónomo, por ejemplo, para manipular varias rutas a la vez.
- Los **scripts Supernodos** se guardan como una propiedad de ruta de supernodo. Los scripts Supernodos sólo están disponibles en supernodos terminales. Puede utilizar un script de supernodo para controlar la secuencia de ejecución del contenido del supernodo. En supernodos no terminales (origen o proceso), puede definir propiedades del supernodo o los nodos que contiene en su script de ruta directamente.

---

1. El lenguaje de script de IBM SPSS Modeler continúa disponible para ser utilizado con IBM SPSS Modeler 16. Para obtener más información, consulte el documento *Manual de scripts y automatización de IBM SPSS Modeler 16*. Consulte Apéndice B, "Migración desde scripts de herencia a scripts Python", en la página 249 para obtener información sobre la correlación de sus scripts de IBM SPSS Modeler de herencia existentes con los scripts Python.

---

## Scripts de ruta

Los scripts se pueden utilizar para personalizar operaciones dentro de una ruta particular y se guardan con esa ruta. Los scripts de la ruta se pueden utilizar para especificar un orden de ejecución particular para los nodos terminales de una ruta. El cuadro de diálogo del script de ruta se utiliza para editar el script que está guardado con la ruta actual.

Para acceder a la pestaña de scripts de ruta en el cuadro de diálogo Propiedades de ruta:

1. En el menú Herramientas, seleccione:  
**Propiedades de la ruta > Ejecución.**
2. Pulse en la pestaña **Ejecución** para trabajar con scripts en la ruta actual.
3. Seleccione el modo de ejecución: **Predeterminado (script opcional).**

Los iconos de la barra de herramientas de la parte superior del cuadro de diálogo del script de ruta le permiten realizar las siguientes operaciones:

- Importar el contenido de un script autónomo preexistente a la ventana.
- Guardar un script como archivo de texto.
- Imprimir un script.
- Añadir script predeterminado.
- Editar un script (deshacer, cortar, copiar, pegar y otras funciones de edición comunes).
- Ejecutar el script completo actual.
- Ejecutar líneas concretas de un script.
- Detener un script durante la ejecución. (Este icono sólo está habilitado cuando un script se está ejecutando).
- Comprobar la sintaxis del script y, si se detectan errores, abrirlos en el panel inferior del cuadro de diálogo para su revisión.

Además, puede especificar si este script se debe ejecutar o no cuando se ejecuta la ruta. Puede seleccionar **Ejecutar este script** para que se ejecute cada vez que se ejecute la ruta y se use el orden de ejecución especificado en el script. De este modo se proporciona una automatización a nivel de ruta para acelerar la generación del modelo. Sin embargo, la configuración predeterminada es omitir el script durante la ejecución de la ruta. Incluso si selecciona la opción **Omitir este script**, siempre puede ejecutar la ruta directamente desde este cuadro de diálogo.

También puede optar por cambiar el tipo de script de script Python a script de herencia.

El editor de scripts incluye las siguientes características que ayudan a crear scripts:

- Resaltado de sintaxis. Se resaltan las palabras claves, los valores literales (tales como cadenas y números) y los comentarios.
- Numeración de líneas.
- Coincidencia de bloques. Cuando se coloca el cursor al inicio de un bloque de programa, también se resalta el bloque final correspondiente.
- Finalización automática sugerida.

Los colores y los estilos de texto que utiliza el resaltado de sintaxis se pueden personalizar utilizando las preferencias de visualización de IBM SPSS Modeler. Puede acceder a las preferencias de visualización seleccionando **Herramientas > Opciones > Opciones de usuario** y pulsando la pestaña **Sintaxis**.

Se puede acceder a una lista de finalizaciones de sintaxis sugeridas seleccionando la **Sugerencia automática** en el menú de contexto o pulsando Ctrl más espacio. Utilice las teclas de cursor para



desplazarse hacia arriba y hacia abajo por la lista y, a continuación, pulse Intro para insertar el texto seleccionado. Pulse Esc para salir de la modalidad de sugerencia automática sin modificar el texto existente.

La pestaña **Depurar** muestra mensajes de depuración y puede utilizarse para evaluar el estado del script una vez ejecutado el script. La pestaña **Depurar** consta de un área de texto de sólo lectura y un campo de texto de entrada de una sola línea. El área de texto muestra el texto que se envía a la salida estándar, por ejemplo a través del comando print de Python, o un error estándar mediante los scripts, por ejemplo, a través del texto del mensaje de error. El campo de texto de entrada toma la entrada del usuario. Esta entrada se evalúa dentro del contexto del script que se ha ejecutado más recientemente en el diálogo (conocido como el *contexto de los scripts*). El área de texto contiene el mandato y la salida resultante, de modo que el usuario puede ver un rastro de los comandos. El campo de texto de entrada siempre contiene el indicador de comandos (>>> para scripts Python).

Un contexto de script nuevo se crea en las circunstancias siguientes:

- Se ejecuta un script utilizando el botón "Ejecutar este script" o el botón "Ejecutar líneas seleccionadas".
- Se modifica el lenguaje de script.

Si se crea un nuevo contexto de script, el área de texto se borra.

**Nota:** Si se ejecuta una ruta fuera del panel de scripts no se modificará el contexto del script contenido en el panel de scripts. Los valores de cualquier variable creada como parte de dicha ejecución no se podrán ver en el diálogo del script.

---

## Scripts autónomos

El cuadro de diálogo script autónomo se usa para crear o editar un script que se ha guardado como archivo de texto. En él se muestra el nombre del archivo y se proporcionan recursos para la carga, almacenamiento, importación y ejecución de scripts.

Para acceder al cuadro de diálogo del script autónomo:

En el menú principal, elija:

**Herramientas > Script autónomo**

Los scripts autónomos y los de ruta comparten las mismas opciones de comprobación de sintaxis de scripts y barra de herramientas. Consulte el tema "Scripts de ruta" en la página 2 para obtener más información.

---

## Scripts de Supernodo

Puede crear y guardar scripts con cualquier supernodo terminal utilizando el lenguaje de scripts de IBM SPSS Modeler. Estos scripts sólo están disponibles para supernodos terminales y se suelen utilizar cuando crea rutas de plantilla o para imponer un orden de ejecución especial del contenido del supernodo. Los scripts de supernodo también permiten ejecutar más de un script en una ruta.

Por ejemplo, supongamos que necesita especificar el orden de ejecución de una ruta compleja y su supernodo contiene varios nodos, incluyendo un nodo Val. globales, que se debe ejecutar antes de derivar un nuevo campo utilizado en un nodo Gráfico. En este caso, puede crear un script de supernodo que ejecute el nodo Val. globales en primer lugar. Los valores calculados por este nodo, como la media o la desviación estándar, se pueden usar posteriormente cuando se ejecute el nodo Gráfico.

En un script de Supernodo, puede especificar las propiedades del nodo de la misma manera que otros scripts. También puede cambiar y definir las propiedades de cualquier supernodo o sus nodos

encapsulados directamente desde un script de ruta. Consulte el tema Capítulo 19, “Propiedades de Supernodos”, en la página 243 para obtener más información. Este método funciona para supernodos de origen y proceso y supernodos terminales.

*Note:* como sólo los supernodos terminales pueden ejecutar sus propios scripts, la pestaña Scripts del cuadro de diálogo Supernodo sólo está disponible para supernodos terminales.

Para abrir el cuadro de diálogo de script de supernodo desde el lienzo principal:

Seleccione un supernodo terminal en el lienzo de rutas y, en el menú de supernodo, seleccione:

### Script de Supernodo...

Para abrir el cuadro de diálogo de script de supernodo desde el lienzo de supernodo aumentado:

Pulse con el botón derecho en el lienzo de supernodo y en el menú contextual seleccione:

### Script de Supernodo...

---

## Creación de bucles y ejecución condicional en rutas

A partir de la versión 16.0, SPSS Modeler permite crear scripts básicos desde dentro de una ruta seleccionando valores en varios cuadros de diálogo en lugar de tener que escribir instrucciones directamente en lenguaje de script. Los dos principales tipos de script que pueden crearse de este modo son los bucles sencillos y un modo de ejecutar nodos si se cumple una condición.

En una misma ruta pueden combinarse bucles y reglas de ejecución condicional. Por ejemplo, suponga que tiene datos relativos a ventas de vehículos de fabricantes de todo el mundo. Podría crearse un bucle para procesar los datos en una ruta, identificando los detalles por país del fabricante, y sacar los datos a distintas gráficas para mostrar detalles tales como volumen de ventas por modelo, niveles de emisión por fabricante y cilindrada, etc. Si solo le interesara analizar la información procedente de Europa, también podría añadir condiciones al bucle que impidieran la creación de gráficas de fabricantes procedentes de América y Asia.

**Nota:** Puesto que tanto un bucle como una ejecución condicional están basados en scripts de segundo plano, solo se aplican a una ruta entera cuando se ejecuta.

- **Bucles** Los bucles pueden utilizarse para automatizar tareas repetitivas. Por ejemplo, esto podría suponer añadir un determinado número de nodos a una ruta y modificar un parámetro del nodo cada vez. De forma opcional, podría controlarse la ejecución de una ruta o rama varias veces, como en los ejemplos siguientes:
  - Ejecutar la ruta un determinado número de veces y cambiar el origen cada vez.
  - Ejecutar la ruta un determinado número de veces cambiando el valor de una variable cada vez.
  - Ejecutar la ruta un determinado número de veces especificando un campo adicional en cada ejecución.
  - Construir un modelo un determinado número de veces y cambiar la configuración del modelo cada vez.
- **Ejecución condicional** Puede utilizarse para controlar cómo ejecutan los nodos en función de condiciones definidas previamente como, por ejemplo:
  - Dependiendo de si un determinado valor es verdadero o falso, se controla la ejecución de un nodo.
  - Definir si la iteración de nodos se ejecutará en paralelo o de forma secuencial.

Tanto bucles ejecuciones condicionales se configuran en la pestaña Ejecución dentro del cuadro de diálogo Propiedades de ruta. Los nodos que se utilicen en bucles o de forma condicional aparecerán con un símbolo adicional en el lienzo de rutas para indicar que forman parte de una ejecución por bucles o condicional.

Puede accederse a la pestaña Ejecución de tres maneras:

- Mediante los menús de la parte superior del cuadro de diálogo principal:
  1. En el menú Herramientas, seleccione:  
**Propiedades de la ruta > Ejecución.**
  2. Pulse en la pestaña Ejecución para trabajar con los scripts de la ruta actual.
- Dentro de una ruta:
  1. Pulse con el botón derecho en un nodo y seleccione **Bucles/Ejecución condicional.**
  2. Seleccione la opción de submenú que corresponda.
- En la barra de herramientas gráfica de la parte superior del cuadro de diálogo principal, pulse en el icono de propiedades de ruta.

Si es la primera vez que configura los detalles de un bucle o de una ejecución condicional, en la pestaña Ejecución seleccione el modo de ejecución **Ejecución de bucles/condicional** y después seleccione la subpestaña **Condicional** o **Bucles**.

## Bucles en rutas

Con la creación de bucles puede automatizar las tareas repetitivas en las rutas, por ejemplo:

- Ejecutar una ruta un número de veces determinado y cambiar el origen cada vez.
- Ejecutar una ruta un número de veces determinado y cambiar el valor de una variable cada vez.
- Ejecutar una ruta un número de veces determinado, especificando un campo adicional en cada ejecución.
- Crear un modelo un número de veces determinado y cambiar un valor de modelo cada vez.

Configurar las condiciones que deben cumplirse en la subpestaña **Bucle** de la pestaña Ejecución de la ruta. Para visualizar la subpestaña, seleccione el modo de ejecución **Ejecución en bucle/condicional**.

Los requisitos de bucle que defina entrarán en vigor cuando se ejecute la ruta, si se ha establecido la modalidad de ejecución **Ejecución en bucle/condicional**. De forma opcional, puede generar el código de script para los requisitos de bucle y pegarlo en el editor de scripts pulsando **Pegar...** en el ángulo inferior derecho de la subpestaña Bucle y la visualización de la pestaña Ejecución principal cambiará para mostrar la modalidad de ejecución **Predeterminada (script opcional)** con el script en la parte superior de la pestaña. Esto significa que puede definir bucles utilizando las diferentes opciones del cuadro de diálogo de bucle antes de generar un script que puede personalizar adicionalmente en el editor de scripts. Tenga en cuenta que cuando pulsa **Pegar...** los requisitos de bucle que ha definido también se mostrarán en el script generado.

Para configurar un bucle:

1. Cree una clave de iteración para definir la estructura principal del bucle principal que se creará en una ruta. Consulte el tema Crear una clave de iteración para obtener más información.
2. Si es necesario, defina una o varias variables de iteración. Consulte el tema Crear una variable de iteración para obtener más información.
3. Las iteraciones y las variables que cree se muestran en el cuerpo principal de la subpestaña. De forma predeterminada, las iteraciones se ejecutan en el orden en que aparecen. Para subir o bajar una iteración en la lista, pulse la iteración para seleccionarla y, a continuación, utilice la flecha arriba o la flecha abajo de la columna de la derecha de la subpestaña para cambiar el orden.

## Creación de una clave de iteración para bucles de rutas

Utilice una clave de iteración para definir la estructura principal del bucle principal que se creará en una ruta. Por ejemplo, si está analizando las ventas de automóviles, puede crear un parámetro de ruta *País de fabricación* y utilizarlo como la clave de la iteración. Cuando se ejecute la ruta, esta clave se establece en cada valor de país diferente de sus datos durante cada iteración. Utilice el cuadro de diálogo Definir clave de iteración para configurar la clave.

Para abrir el cuadro de diálogo, seleccione el botón **Clave de iteración...** en el ángulo inferior izquierdo de la subpestaña Bucle o pulse con el derecho cualquier nodo de la ruta y seleccione **Ejecución en bucle/condicional > Definir clave de iteración (campos)** o **Ejecución en bucle/condicional > Definir clave de iteración (valores)**. Si abre el cuadro de diálogo desde la ruta, algunos campos se completan automáticamente, tales como el nombre del nodo.

Para configurar una clave de iteración, complete los campos siguientes:

**Iterar en.** Puede seleccionar entre una de las opciones siguientes:

- **Parámetro de ruta - Campos.** Utilice esta opción para crear un bucle que establezca el valor de un parámetro de ruta existente en cada campo especificado de forma ordenada.
- **Parámetro de ruta - Valores.** Utilice esta opción para crear un bucle que establezca el valor de un parámetro de ruta existente en cada valor especificado de forma ordenada.
- **Propiedad del nodo - Campos.** Utilice esta opción para crear un bucle que establezca el valor de una propiedad de nodo en cada campo especificado de forma ordenada.
- **Propiedad del nodo - Valores.** Utilice esta opción para crear un bucle que establezca el valor de una propiedad de nodo en cada valor especificado de forma ordenada.

**Qué se ha de establecer.** Elija el elemento cuyo valor se establecerá cada vez que se ejecute el bucle. Puede seleccionar entre una de las opciones siguientes:

- **Parámetro.** Solo está disponible si se selecciona **Parámetro de ruta - Campos** o **Parámetro de ruta - Valores**. Seleccione el parámetro necesario en la lista disponible.
- **Nodo.** Solo está disponible si se selecciona **Propiedad del nodo - Campos** o **Propiedad del nodo - Valores**. Seleccione el nodo para el que desee configurar un bucle. Pulse el botón Examinar para abrir el diálogo Seleccionar nodo y elija el nodo que desee. Si hay demasiados nodos en la lista, puede filtrar la visualización para que únicamente se muestren los nodos de una de las siguientes categorías: Origen, Proceso, Gráfico, Modelado, Resultados o Aplicar modelo.
- **Propiedad.** Solo está disponible si se selecciona **Propiedad del nodo - Campos** o **Propiedad del nodo - Valores**. Seleccione la propiedad del nodo en la lista disponible.

**Campos de uso.** Solo está disponible si se selecciona **Parámetro de ruta - Campos** o **Propiedad del nodo - Campos**. Seleccione el campo o los campos de un nodo que se utilizarán para proporcionar los valores de iteración. Puede seleccionar entre una de las opciones siguientes:

- **Nodo.** Solo está disponible si se selecciona **Parámetro de ruta - Campos**. Seleccione el nodo que contiene los detalles para los que desea configurar un bucle. Pulse el botón Examinar para abrir el diálogo Seleccionar nodo y elija el nodo que desee. Si hay demasiados nodos en la lista, puede filtrar la visualización para que únicamente se muestren los nodos de una de las siguientes categorías: Origen, Proceso, Gráfico, Modelado, Resultados o Aplicar modelo.
- **Lista de campos.** Pulse el botón de lista de la columna derecha para visualizar el cuadro de diálogo Seleccionar campos, donde puede seleccionar los campos del nodo para proporcionar los datos de iteración. Para obtener más información, consulte “Selección de campos en iteraciones” en la página 8.

**Valores de uso.** Solo está disponible si se selecciona **Parámetro de ruta - Valores** o **Propiedad del nodo - Valores**. En el campo seleccionado, seleccione el o los valores que se utilizarán como valores de iteración. Puede seleccionar entre una de las opciones siguientes:

- **Nodo.** Solo está disponible si se selecciona **Parámetro de ruta - Valores**. Seleccione el nodo que contiene los detalles para los que desea configurar un bucle. Pulse el botón Examinar para abrir el diálogo Seleccionar nodo y elija el nodo que desee. Si hay demasiados nodos en la lista, puede filtrar la visualización para que únicamente se muestren los nodos de una de las siguientes categorías: Origen, Proceso, Gráfico, Modelado, Resultados o Aplicar modelo.
- **Lista de campos.** Seleccione el campo del nodo para proporcionar los datos de iteración.
- **Lista de valores.** Pulse el botón de lista de la columna derecha para visualizar el cuadro de diálogo Seleccionar valores, donde puede seleccionar los valores del campo para proporcionar los datos de iteración.

## Creación de una variable de iteración para bucles de rutas

Puede utilizar variables de iteración para cambiar los valores de los parámetros o las propiedades de ruta de los nodos seleccionados en una ruta, cada vez que se ejecute un bucle. Por ejemplo, si el bucle de ruta está analizando los datos de ventas de automóviles y utiliza *País de fabricación* como clave de iteración, puede tener un gráfico de resultados que muestre las ventas por modelo y otro gráfico de resultados que muestre información sobre emisiones contaminantes. En estos casos puede crear variables de iteración que creen nuevos títulos para los gráficos resultantes, tales como *Emisiones de vehículos suecos* y *Ventas de automóviles japoneses por modelo*. Utilice el cuadro de diálogo Definir variable de iteración para configurar las variables que necesite.

Para abrir el cuadro de diálogo, seleccione el botón **Variable de iteración...** en el ángulo superior izquierdo de la subpestaña Bucle, o pulse con el botón derecho cualquier nodo de la ruta y seleccione **Ejecución en bucle/condicional > Definir variable de iteración**.

Para configurar una variable de iteración, complete los campos siguientes:

**Cambiar.** Seleccione el tipo de atributo que desea enmendar. Puede elegir **Parámetro de ruta** o **Propiedad del nodo**.

- Si selecciona **Parámetro de ruta**, elija el parámetro necesario y, a continuación, utilizando una de las opciones siguientes, si están disponibles en su ruta, defina cómo se debe establecer el valor de dicho parámetro con cada iteración del bucle:
  - **Variable global.** Seleccione la variable global en la que se debe establecer el parámetro de ruta.
  - **Casilla de resultados de tabla.** Para que un parámetro de ruta sea el valor de una casilla de resultados de tabla, seleccione la tabla en la lista y especifique la **Fila** y la **Columna** que se han de utilizar.
  - **Especificar manualmente.** Seleccione esta opción si desea especificar manualmente el valor que tomará este parámetro en cada iteración. Cuando regrese a la subpestaña Bucle se habrá creado una columna nueva en la que puede especificar el texto necesario.
- Si selecciona **Propiedad del nodo**, elija el nodo necesario y una de sus propiedades, a continuación, establezca el valor que desea que se utilice para dicha propiedad. Establezca el nuevo valor de la propiedad utilizando una de las opciones siguientes:
  - **Solo.** El valor de la propiedad utilizará el valor de la clave de iteración. Para obtener más información, consulte “Creación de una clave de iteración para bucles de rutas” en la página 6.
  - **Como prefijo de tallo.** Utiliza el valor de la clave de iteración como un prefijo para lo que especifique en el campo **Tallo**.
  - **Como sufijo de tallo.** Utiliza el valor de la clave de iteración como un sufijo para lo que especifique en el campo **Tallo**.

Si selecciona la opción de prefijo o de sufijo se le solicitará que añada el texto adicional en el campo **Tallo**. Por ejemplo, si el valor de la clave de iteración es *País de fabricación* y selecciona **Como prefijo de tallo**, puede entrar - *ventas por modelo* en este campo.

## Selección de campos en iteraciones

Cuando se crean iteraciones pueden seleccionarse uno o más campos mediante el cuadro de diálogo Seleccionar campos.

**Ordenar por.** Puede ordenar campos disponibles para su visualización eligiendo una de las siguientes opciones:

- **Natural.** El orden de los campos es aquél en que pasaron desde la parte anterior de la ruta de datos al nodo actual.
- **Nombre.** Ordena los campos siguiendo un orden alfabético para su visualización.
- **Tipo.** Ordena los campos en función de su nivel de medición. Esta opción es útil cuando se seleccionan campos con un nivel de medición en particular.

Seleccione los campos de la lista de uno en uno o mantenga pulsada la tecla Mayús o Ctrl mientras selecciona otros campos para seleccionar varios campos. También puede utilizar los botones que se muestran bajo la lista para seleccionar grupos de campos en función de su nivel de medición o seleccionar y anular la selección de todos los campos de la tabla.

Tenga en cuenta que los campos disponibles para su selección se filtran para mostrar solo los campos que son adecuados para el parámetro de ruta o la propiedad de nodo que se están utilizando. Por ejemplo, si se está utilizando un parámetro de ruta que tiene un tipo de almacenamiento de cadena, solo se mostrarán los campos que tengan un tipo de almacenamiento de cadena.

## Ejecución condicional en rutas

Con la ejecución condicional puede controlar cómo se ejecutan los nodos terminales, en función de las condiciones de coincidencia de contenido de ruta que defina. Ejemplos de ello pueden ser los siguientes:

- En función de si un valor determinado es true o false, controlar si se ejecutará un nodo.
- Definir si un bucle de nodos se ejecutará en paralelo o secuencialmente.

Configurar las condiciones que deben cumplirse en la subpestaña **Condición** de la pestaña Ejecución de la ruta. Para visualizar la subpestaña, seleccione el modo de ejecución **Ejecución en bucle/condicional**.

Los requisitos de ejecución condicional que defina entrarán en vigor cuando se ejecute la ruta, si se ha establecido la modalidad de ejecución **Ejecución en bucle/condicional**. De forma opcional, puede generar el código de script para los requisitos de ejecución condicional y pegarlo en el editor de scripts pulsando **Pegar...** en el ángulo inferior derecho de la subpestaña Condición; la visualización de la pestaña Ejecución principal cambiará para mostrar la modalidad de ejecución **Predeterminada (script opcional)** con el script en la parte superior de la pestaña. Esto significa que puede definir condiciones utilizando las diferentes opciones del cuadro de diálogo de bucle antes de generar un script que puede personalizar adicionalmente en el editor de scripts. Tenga en cuenta que cuando pulsa **Pegar...** los requisitos de bucle que ha definido también se mostrarán en el script generado.

Para configurar una condición:

1. En la columna de la derecha de la subpestaña Condición, pulse el botón Añadir sentencia de



ejecución para abrir el cuadro de diálogo Sentencia de ejecución condicional. En este diálogo especifica la condición que se debe cumplir para que se ejecute el nodo.

2. En el cuadro de diálogo Sentencia de ejecución condicional, especifique lo siguiente:
  - a. **Nodo.** Seleccione el nodo para el que desee configurar una ejecución condicional. Pulse el botón Examinar para abrir el diálogo Seleccionar nodo y elija el nodo que desee. Si hay demasiados nodos en la lista, puede filtrar la visualización para que únicamente se muestren los nodos de una de las siguientes categorías: Exportar, Gráfico, Modelado o Resultados.
  - b. **Condición basada en.** Especifique la condición que se debe cumplir para que se ejecute el nodo. Puede elegir una de estas cuatro opciones: **Parámetro de ruta**, **Variable global**, **Casilla de**

**resultados de tabla** o **Siempre verdadero**. Los detalles que especifique en la mitad inferior del cuadro de diálogo están controlados por la condición que elija.

- **Parámetro de ruta.** Seleccione el parámetro de la lista disponible y, a continuación, seleccione el **Operador** para ese parámetro; por ejemplo, el operador puede ser Más, Igual, Menor que, Entre, etc. A continuación especifique el **Valor**, o los valores mínimos o máximos, dependiendo del operador seleccionado.
  - **Variable global.** Seleccione la variable de la lista disponible; por ejemplo, esto podría incluir: Media, Suma, Valor mínimo, Valor máximo o Desviación estándar. A continuación, seleccione **Operador** y los valores necesarios.
  - **Casilla de resultados de tabla.** Seleccione el nodo de tabla de la lista disponible y, a continuación, seleccione la **Fila** y la **Columna** en la tabla. A continuación, seleccione **Operador** y los valores necesarios.
  - **Siempre verdadero.** Seleccione esta opción si siempre se ha de ejecutar el nodo. Si selecciona esta opción, no hay parámetros adicionales que seleccionar.
3. Repita los pasos 1 y 2 tantas veces como sea necesario hasta que haya configurado todas las condiciones que requiere. El nodo que ha seleccionado y la condición que se debe cumplir antes de que se ejecute el nodo se muestran en el cuerpo principal de la subpestaña de las columnas **Ejecutar nodo** y **Si esta condición es verdadera** respectivamente.
  4. De forma predeterminada, los nodos y las condiciones se ejecutan en el orden en que aparecen. Para subir o bajar un nodo y condición en la lista, pulse el nodo para seleccionarlo y, a continuación, utilice la flecha arriba o la flecha abajo en la columna de la derecha de la subpestaña para cambiar el orden.

Además, puede establecer las siguientes opciones en la parte inferior de la subpestaña Condicional:

- **Evaluar todo en orden.** Seleccione esta opción para evaluar cada condición en el orden en que se muestra en la subpestaña. Los nodos para los que se han encontrado condiciones que son "True" se ejecutarán una vez evaluadas todas las condiciones.
- **Ejecutar uno por uno.** Sólo está disponible si se selecciona **Evaluar todo en orden**. Si se selecciona significa que si la condición se evalúa como "True", el nodo asociado con esa condición se ejecutará antes de que se evalúe la siguiente condición.
- **Evaluar hasta primer acierto.** Si se selecciona, significa que solo se ejecutará el primer nodo cuya evaluación de las condiciones devuelva el valor "True".

---

## Ejecución e interrupción de scripts

Existen diversas formas de ejecutar scripts. Por ejemplo, en el script de ruta o en el cuadro de diálogo del script, el botón "Ejecutar este script" ejecuta el script completo:



Figura 1. Botón Ejecutar este script

El botón "Ejecutar líneas seleccionadas" ejecuta una única línea, o un bloque de líneas adyacentes, que ha seleccionado en el script:



Figura 2. Botón Ejecutar líneas seleccionadas

Un script se puede ejecutar mediante cualquiera de los siguientes métodos:

- Pulse en el botón "Ejecutar este script" o "Ejecutar líneas seleccionadas" dentro de un script de ruta o un cuadro de diálogo de script.

- Ejecutando una ruta donde **Ejecutar este script** esté establecido como el método de ejecución predeterminado.
- Utilizando la marca `-execute` al inicio en modo interactivo. Consulte el tema “Uso de argumentos en la línea de comandos” en la página 51 para obtener más información.

*Note:* cuando se ejecuta el Supernodo, se ejecuta un script de Supernodo siempre que se haya seleccionado **Ejecutar este script** en el cuadro de diálogo del script de Supernodo.

#### Interrupción de la ejecución del script

En el cuadro de diálogo de scripts de ruta, se activará el botón rojo de detención durante la ejecución de scripts. Pulsando este botón, puede abandonar la ejecución del script y de cualquier ruta actual.

---

## Buscar y reemplazar

El cuadro de diálogo Buscar/reemplazar está disponible en lugares donde se edita texto de script o de expresión, incluido el editor de scripts, o cuando se define una plantilla en el nodo Informe. Al editar texto en cualquiera de estas áreas, pulse `Ctrl+F` para acceder al cuadro de diálogo, asegurándose de que el cursor está centrado en un área de texto. Por ejemplo, si trabaja en un nodo Rellenar, puede acceder al cuadro de diálogo desde cualquiera de las áreas de texto de la pestaña Configuración o desde el campo de texto del generador de expresiones.

1. Con el cursor en un área de texto, pulse `Ctrl+F` para acceder al cuadro de diálogo Buscar/reemplazar.
2. Introduzca el texto que desee buscar o selecciónelo de la lista desplegable de elementos buscados recientemente.
3. Introduzca el texto de reemplazo, si lo tiene.
4. Pulse en **Buscar siguiente** para iniciar la búsqueda.
5. Pulse en **Reemplazar** para reemplazar la sección actual o en **Reemplazar todos** para actualizar todas las instancias o sólo las seleccionadas.
6. El cuadro de diálogo se cierra después de cada operación. Pulse `F3` desde cualquier área de texto para repetir la operación de búsqueda más reciente o pulse `Ctrl+F` para volver a acceder al cuadro de diálogo.

#### Opciones de búsqueda

**Coincidir mayúsculas y minúsculas.** Especifica si la operación de búsqueda hace distinción entre mayúsculas y minúsculas; por ejemplo, si *mivar* es igual que *miVar*. El texto de reemplazo siempre se introduce exactamente como se ha introducido, independientemente de este ajuste.

**Sólo palabras completas.** Especifica si la operación de búsqueda tiene en cuenta el texto incluido dentro de las palabras. Por ejemplo, si se selecciona, la búsqueda de *fuego* no será igual que *cortafuegos* ni que *corta-fuegos*.

**Expresiones regulares.** Especifica si se utiliza la sintaxis de expresiones regulares (consulte la sección siguiente). Si está seleccionado, la opción **Sólo palabras completas** está desactivada y su valor se ignora.

**Sólo texto seleccionado.** Controla el ámbito de la búsqueda al utilizar la opción **Reemplazar todos**.

#### Sintaxis de expresiones regulares

Las expresiones regulares le permiten buscar caracteres especiales como caracteres de tabulador o de nueva línea, clases o rangos de caracteres como de la *a* a la *d*, cualquier dígito o no dígito y límites como el principio o el final de una línea. Una patrón de expresiones regulares describe la estructura de la cadena que intentará encontrar la expresión en una cadena de entrada. Se admiten los siguientes tipos de construcciones de expresiones regulares.



Tabla 1. Coincidencias de caracteres

Caracteres	Coincidencias
x	El carácter x
\\	El carácter de barra inclinada invertida
\0n	El carácter con valor octal 0n (0 <= n <= 7)
\0nn	El carácter con valor octal 0nn (0 <= n <= 7)
\0mnn	El carácter con valor octal 0mnn (0 <= m <= 3, 0 <= n <= 7)
\xhh	El carácter con valor hexadecimal 0xhh
\uhhhh	El carácter con valor hexadecimal 0xhhhh
\t	El carácter de tabulador ('\u0009')
\n	El carácter de nueva línea (avance de línea) ('\u000A')
\r	El carácter de retorno de carro ('\u000D')
\f	El carácter de avance de página ('\u000C')
\a	El carácter de alerta (campana) ('\u0007')
\e	El carácter de escape ('\u001B')
\cx	El carácter de control correspondiente a x

Tabla 2. Coincidencia de clases de caracteres

Clases de caracteres	Coincidencias
[abc]	a, b o c (clase simple)
[^abc]	Cualquier carácter excepto a, b o c (resta)
[a-zA-Z]	De la a a la z o de la A a la Z, ambas inclusive (rango)
[a-d[m-p]]	De la a a la d o de la m a la p (unión). Esto también puede especificarse como [a-dm-p].
[a-z&&[def]]	De la a a la z y d, e o f (intersección)
[a-z&&[^bc]]	De la a a la z, excepto b y c (resta). Esto también puede especificarse como [ad-z].
[a-z&&[^m-p]]	De la a a la z y no de la m a la p (resta). Esto también puede especificarse como [a-lq-z].

Tabla 3. Clases de caracteres predefinidas

Clases de caracteres predefinidas	Coincidencias
.	Cualquier carácter (puede o no coincidir con los terminadores de línea)
\d	Cualquier dígito: [0-9]
\D	Un carácter que no es un dígito: [^0-9]
\s	Un carácter de espacio en blanco : [ \t\n\x0B\f\r]
\S	Un carácter de espacio que no está en blanco : [^\s]
\w	Un carácter de palabra: [a-zA-Z_0-9]
\W	Un carácter que no sea de palabra : [^\w]

Tabla 4. Coincidencias de límite

Reconocedores de límite	Coincidencias
^	El comienzo de una línea

Tabla 4. Coincidencias de límite (continuación)

Reconocedores de límite	Coincidencias
\$	El final de una línea
\b	Un límite alfabético
\B	Un límite no alfabético
\A	El comienzo de la entrada
\Z	El final de la entrada pero para el terminador final, si lo hay
\z	El final de la entrada

Para obtener más información sobre el uso de expresiones regulares y algunos ejemplos, consulte <http://www.ibm.com/developerworks/java/tutorials/j-introjava2/section9.html>.

### Ejemplos

El código siguiente busca y hace coincidir los tres números iniciales de una serie:

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

El código siguiente busca y hace coincidir los tres números finales de una serie:

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

---

## Capítulo 2. Language de scripts

---

### Conceptos básicos del lenguaje de scripts

El recurso de scripts para IBM SPSS Modeler le permite crear scripts que funcionen en la interfaz de usuario de SPSS Modeler, manipular los objetos de salida y ejecutar la sintaxis de comandos. Puede ejecutar scripts directamente desde SPSS Modeler.

Los scripts de IBM SPSS Modeler están escritos en el lenguaje de script Python. La implementación Java de Python que IBM SPSS Modeler se denomina Jython. El lenguaje de script consta de las siguientes características:

- Un formato para hacer referencia a nodos, rutas, proyectos, resultados y otros objetos de IBM SPSS Modeler.
- Un conjunto de instrucciones o comandos de scripts que se puede utilizar para manipular tales objetos.
- Un lenguaje de expresión de script para establecer los valores de las variables, los parámetros y otros objetos.
- Compatibilidad con comentarios, continuaciones y bloques de texto literal.

Las secciones siguientes describen el lenguaje de scripts Python, la implementación de Jython por parte de Python y la sintaxis básica para empezar a crear scripts en IBM SPSS Modeler. Las secciones siguientes recogen información sobre comandos y propiedades específicas.

---

### Python y Jython

Jython es una implementación del lenguaje de scripts Python, escrito en el lenguaje Java e integrado con la plataforma Java. Python es un potente lenguaje de script orientado a objetos. Jython es útil porque proporciona las características de productividad de un lenguaje de script maduro y, a diferencia de Python, se ejecuta en cualquier entorno que soporte una máquina virtual Java (JVM). Esto significa que las bibliotecas Java de la máquina virtual Java están disponibles para utilizarlas cuando se escriben programas. Con Jython, puede beneficiarse de esta diferencia y utilizar la sintaxis y la mayoría de las características del lenguaje Python.

Como lenguaje de script, Python (y su implementación Jython) es fácil de aprender y ofrece una codificación eficaz con la estructura mínima necesaria para crear un programa de ejecución. El código se puede entrar de forma interactiva, es decir, una línea cada vez. Python es un lenguaje de script interpretado; no hay ningún paso de precompilación, tal como existe en Java. Los programas de Python simplemente son archivos de texto que se interpretan a medida que se entran (después de analizar los errores de sintaxis). Las expresiones simples, tales como los valores definidos, y también las acciones más complejas, tales como las definiciones de función, se ejecutan y están disponibles para su uso de forma inmediata. Los cambios realizados en el código se pueden probar fácilmente. Sin embargo, la interpretación del script tiene algunas desventajas. Por ejemplo, utilizar una variable no definida no es un error del compilador, por lo tanto solo se detecta si (y cuando) se ejecuta la sentencia en la que se utiliza la variable. En este caso, se puede editar y ejecutar el programa para depurar el error.

Python lo ve todo como un objeto, incluidos todos los datos y el código. Por lo tanto, puede manipular estos objetos con líneas de código. Algunos tipos de selección, tales como los números y cadenas, se consideran valores y no objetos, lo cual resulta más práctico, y Python da soporte a todo ello. Se da soporte a un valor nulo. Este valor nulo tiene el nombre reservado de None.

Para obtener una introducción más detallada del lenguaje de script Python y Jython y algunos scripts de ejemplo, consulte el tema [www.ibm.com/developerworks/java/tutorials/j-jython1](http://www.ibm.com/developerworks/java/tutorials/j-jython1) y el tema [www.ibm.com/developerworks/java/tutorials/j-jython2](http://www.ibm.com/developerworks/java/tutorials/j-jython2).

---

## Creación de scripts Python

Esta guía del lenguaje de script de Python es una introducción a los componentes que tienen más probabilidad de ser utilizados cuando se ejecutan scripts en IBM SPSS Modeler, incluidos conceptos y principios básicos de programación. Le proporcionará los conocimientos suficientes para comenzar a desarrollar sus propios scripts Python y utilizarlos en IBM SPSS Modeler.

### Operaciones

a asignación se realiza mediante un signo de igual (=). Por ejemplo, para asignar el valor "3" a una variable llamada "x" debe utilizar la siguiente sentencia:

```
x = 3
```

El signo igual también se utiliza para asignar datos de tipo de cadena a una variable. Por ejemplo, para asignar el valor "a string value" a la variable "y" utilice la sentencia siguiente:

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

La tabla siguiente enumera algunas de las operaciones numéricas y de comparación utilizadas con más frecuencia y sus descripciones.

Tabla 5. Operaciones numéricas y de comparación comunes

Operación	Descripción
$x < y$	¿Es x menor que y?
$x > y$	¿Es x mayor que y?
$x \leq y$	¿Es x menor que o igual a y?
$x \geq y$	¿Es x mayor que o igual a y?
$x == y$	¿Es x igual a y?
$x != y$	¿Es x no igual a y?
$x \lt;> y$	¿Es x no igual a y?
$x + y$	Sumar y a x
$x - y$	Restar y de x
$x * y$	Multiplicar x por y
$x / y$	Dividir x por y
$x ** y$	Elevar x a la potencia de y

### Listas

Las listas son secuencias de elementos. Una lista puede contener cualquier número de elementos, y los elementos de la lista pueden ser cualquier tipo de objeto. Las listas también se pueden considerar como matrices. El número de elementos de una lista puede aumentar o disminuir a medida que se añaden, eliminan o sustituyen elementos.

#### Ejemplos

<code>[]</code>	Cualquier lista vacía.
<code>[1]</code>	Una lista con un solo elemento, un entero.
<code>["Mike", 10, "Don", 20]</code>	Una lista con cuatro elementos, dos elementos de cadena y dos elementos de entero.
<code>[[], [7], [8, 9]]</code>	Una lista de listas. Cada sublista es una lista vacía o una lista de elementos de enteros.

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

Una lista de enteros. Este ejemplo muestra el uso de variables y expresiones.

Puede asignar una lista a una variable, por ejemplo:

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

A continuación, puede acceder a los elementos específicos de la lista, por ejemplo:

```
mylist[0]
```

Esto genera el resultado siguiente:

```
one
```

El número entre corchetes ([]) se considera un *index* y hace referencia a un elemento concreto de la lista. Los elementos de una lista se indexan a partir de 0.

También puede seleccionar un rango de elementos de una lista; esto se denomina *porciones*. Por ejemplo, `x[1:3]` selecciona el segundo y el tercer elemento de `x`. El índice final es uno más allá de la selección.

## Cadenas

Una *cadena* es una secuencia inmutable de caracteres que se trata como un valor. Las cadenas dan soporte a todas las funciones de secuencias inmutables y operadores que generan como resultado una nueva serie. Por ejemplo, `"abcdef"[1:4]` da como resultado la salida `"bcd"`.

En Python, los caracteres se representan mediante cadenas de caracteres de longitud uno.

Los literales de cadenas se definen mediante comillas simples o triples. Las cadenas definidas mediante comillas simples no pueden abarcar líneas, mientras que las series definidas mediante comillas triples sí que pueden. Una cadena puede estar entre comillas simples (') o entre comillas dobles ("). Un carácter entrecomillado puede contener el otro carácter entrecomillado o el carácter entrecomillado de escape, que es el carácter de barra invertida (\).

Ejemplos

```
"Esta es una cadena"
'Esta también es una cadena'
"Es una cadena"
'Este manual se titula "Guía de Python Scripting and Automation".'
"Estas son comillas de escape (\") en una cadena entrecomillada"
```

El analizador de Python automáticamente concatena varias cadenas separadas por un espacio en blanco. Esto facilita la entrada de cadenas largas y la combinación de tipos de comillas en una sola cadena, por ejemplo:

```
"Esta cadena utiliza ' y " 'esta cadena utiliza ".'
```

Esto da lugar a la salida siguiente:

```
Esta cadena utiliza ' y esa cadena utiliza ".
```

Las cadenas dan soporte a varios métodos útiles. Algunos de estos métodos se proporcionan en la tabla siguiente.

Tabla 6. Métodos de cadena

Método	Uso
<code>s.capitalize()</code>	Mayúscula inicial <code>s</code>
<code>s.count(ss {,start {,end}})</code>	Recuento de apariciones de <code>ss</code> en <code>s[start:end]</code>

Tabla 6. Métodos de cadena (continuación)

Método	Uso
s.startswith(str {, start {, end}}) s.endswith(str {, start {, end}})	Probar si s comienza por str Probar si s acaba en str
s.expandtabs({size})	Sustituir tabulaciones por espacios, el valor predeterminado de size es 8
s.find(str {, start {, end}}) s.rfind(str {, start {, end}})	Busca el primer índice de str en s; si no se encuentra, el resultado es -1. rfind busca de derecha a izquierda.
s.index(str {, start {, end}}) s.rindex(str {, start {, end}})	Busca el primer índice de str en s; si no se encuentra: se genera ValueError. rindex busca de derecha a izquierda.
s.isalnum	Probar si la cadena es alfanumérica
s.isalpha	Probar si la cadena es alfabética
s.isnum	Probar si la cadena es numérica
s.isupper	Probar si la cadena está toda en mayúsculas
s.islower	Probar si la cadena está toda en minúsculas
s.isspace	Probar si la cadena está toda en espacios en blanco
s.istitle	Probar si la cadena es una secuencia de cadenas alfanuméricas con mayúscula inicial
s.lower() s.upper() s.swapcase() s.title()	Convertir todo a minúsculas Convertir todo a mayúsculas Convertir de mayúsculas a minúsculas o viceversa Convertir todo a mayúsculas o minúsculas del título
s.join(seq)	Unir las cadenas de seq con s como separador
s.splitlines({keep})	Dividir s en líneas, si keep es true, mantener las nuevas líneas
s.split({sep {, max}})	Dividir s en "palabras" utilizando sep (el valor predeterminado de sep es un espacio en blanco) para un máximo de max veces
s.ljust(width) s.rjust(width) s.center(width) s.zfill(width)	Justificar la cadena a la izquierda con un ancho de campo de width Justificar la cadena a la derecha con un ancho de campo de width Justificar la cadena al centro con un ancho de campo de width Rellenar con 0.
s.lstrip() s.rstrip() s.strip()	Eliminar espacios en blanco iniciales Eliminar espacios en blanco de cola Eliminar espacios en blanco iniciales y de cola
s.translate(str {, delc})	Traducir s utilizando la tabla, después eliminar cualquier carácter de delc. str debe ser una cadena con una longitud de == 256.
s.replace(old, new {, max})	Sustituye todas las apariciones de max de la cadena old por la cadena new

## Comentarios

Los comentarios se introducen con el signo de almohadilla (#) o hash. Todo el texto que sigue al signo de almohadilla en la misma línea se considera parte del comentario y se omite. Un comentario puede comenzar en cualquier columna. El ejemplo siguiente muestra el uso de los comentarios:

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

## Sintaxis de las sentencias

La sintaxis de las sentencias para Python es muy sencilla. En general, cada línea de origen es una sola sentencia. A excepción de las sentencias `expression` y `assignment`, cada sentencia se introduce mediante un nombre de palabra clave, tal como `if` o `for`. Las líneas en blanco o las líneas de comentarios se pueden insertar en cualquier lugar entre cualquier sentencia del código. Si existe más de una sentencia en una línea, cada sentencia debe estar separada por un signo de punto y coma (;).

Las sentencias muy largas pueden continuar en más de una línea. En este caso, la sentencia que ha de continuar en la línea siguiente debe acabar con una barra invertida (\), por ejemplo:

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

Cuando una estructura está encerrada entre paréntesis (), corchetes [] o llaves {}, la sentencia puede continuar en una línea nueva después de cualquier coma, sin tener que insertar una barra invertida, por ejemplo:

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

## Identificadores

Los identificadores se utilizan para el nombre de las variables, funciones, clases y palabras clave. Los identificadores pueden tener cualquier longitud, pero debe empezar con un carácter alfabético en mayúsculas o minúsculas o el carácter de subrayado (\_). Los nombres que empiezan con un carácter de subrayado están generalmente reservados para los nombres internos o privados. Después del primer carácter, el identificador puede contener cualquier número y combinación de caracteres alfabéticos, los números del 0-9, y el carácter de subrayado.

Existen algunas palabras reservadas en Python que no se pueden utilizar para el nombre de variables, funciones o clases. Estas palabras entran en las siguientes categorías:

- **Introducciones de sentencias:** `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `exec`, `finally`, `for`, `from`, `global`, `if`, `import`, `pass`, `print`, `raise`, `return`, `try` y `while`
- **Introducciones de parámetros:** `as`, `import` y `in`
- **Operadores:** `and`, `in`, `is`, `lambda`, `not` y `or`

El uso incorrecto de palabras claves suele generar `SyntaxError`.

## Bloques de código

Bloques de código son grupos de sentencias que se utilizan donde se esperan sentencias individuales. Los bloques de código pueden seguir a cualquiera de las sentencias siguientes: `if`, `elif`, `else`, `for`, `while`, `try`, `except`, `def` y `class`. Estas sentencias introducen el bloque de código con el carácter de dos puntos (:), por ejemplo:

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

Se utiliza la indentación para delimitar los bloques de código (en lugar de las llaves que se utilizan en Java). Todas las líneas de un bloque han de indentarse en la misma posición. Esto es debido a que un cambio en la indentación indica el final de un bloque de código. Normalmente la indentación es de cuatro espacios por nivel. Se recomienda utilizar espacios para la indentación, en lugar de tabulaciones. No se deben combinar espacios y tabulaciones. Las líneas del bloque de un módulo situado más al extremo deben comenzar en la columna uno, de lo contrario, se genera el error `SyntaxError`.

Las sentencias que componen un bloque de código (y siguen el signo de dos puntos) también deben estar en una sola línea, separadas por signos de punto y coma, por ejemplo:

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

## Pasar argumentos a un script

Pasar argumentos a un script puede resultar útil para poder utilizar un script reiteradamente sin modificarlo. Los argumentos se pasan en la línea de comandos como valores de la lista `sys.argv`. El número de valores que se pasan se puede obtener mediante el comando `len(sys.argv)`. Por ejemplo:

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

En este ejemplo, el comando `import` importa toda la clase `sys`, por lo que se pueden utilizar los métodos existentes para esta clase, tales como `argv`.

El script de este ejemplo se puede invocar utilizando la línea siguiente:

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

Esto genera el resultado siguiente:

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

## Ejemplos

La palabra clave `print` imprime los argumentos situados inmediatamente después de la misma. Si la sentencia va seguida de una coma, no se incluye una línea nueva en los resultados. Por ejemplo:

```
print "Esto muestra el uso de una",
print " coma al final de una sentencia de impresión."
```

Esto genera el resultado siguiente:

```
Esto muestra el uso de una coma al final de una sentencia de impresión.
```

La sentencia `for` se utiliza para la iteración por un bloque de código. Por ejemplo:

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

En este ejemplo, se asignan tres cadenas a la lista `mylist1`. Los elementos de la lista se imprimen a continuación, con un elemento de cada línea. Esto genera el resultado siguiente:

```
one
two
three
```

En este ejemplo, el iterador `lv` toma el valor de cada elemento de la lista `mylist1` por orden, mientras el bucle `for` implementa el bloque de código de cada elemento. Un iterador puede ser cualquier identificador válido de cualquier longitud.

La sentencia `if` es una sentencia condicional. Evalúa la condición y devuelve `true` o `false`, en función del resultado de la evaluación. Por ejemplo:



```

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

```

En este ejemplo, se evalúa el valor del iterador lv. Si el valor de lv es two se devuelve una cadena diferente a la cadena que se devuelve si el valor de lv no es two. Esto da lugar a la salida siguiente:

```

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

```

## Métodos matemáticos

Desde el módulo matemáticas puede acceder a métodos matemáticos útiles. Algunos de estos métodos se proporcionan en la tabla siguiente. A menos que se especifique lo contrario, todos los valores se devuelven como valores flotantes.

Tabla 7. Métodos matemáticos

Método	Uso
math.ceil(x)	Devuelve el punto más alto de x como un valor flotante, que es el entero más pequeño mayor o igual a x
math.copysign(x, y)	Devuelve x con el signo de y. copysign(1, -0.0) devuelve -1
math.fabs(x)	Devuelve el valor absoluto de x
math.factorial(x)	Devuelve el factor de x. Si x es negativo o no es un entero, se genera ValueError.
math.floor(x)	Devuelve el punto más bajo de x como un valor flotante, que es el entero más alto menor o igual a x
math.frexp(x)	Devuelve la mantisa (m) y el exponente (e) de x como el par (m, e). m es un valor flotante y e es un entero, tal como $x == m * 2^{**}e$ exactamente. Si x es cero, devuelve (0,0, 0), de lo contrario $0,5 <= abs(m) < 1$ .
math.fsum(iterable)	Devuelve una suma de coma flotante precisa de los valores de iterable
math.isinf(x)	Comprueba si el valor flotante x es positivo o negativo infinito
math.isnan(x)	Comprueba si el valor flotante x es NaN (no es un número)
math.ldexp(x, i)	Devuelve $x * (2^{**}i)$ . Esencialmente es la función inversa de frexp.
math.modf(x)	Devuelve las partes de fracción y entero de x. Los dos resultados llevan el signo de x y son flotantes.
math.trunc(x)	Devuelve el valor Real de x, que se ha truncado en un Integral.
math.exp(x)	Devuelve $e^{**}x$
math.log(x[, base])	Devuelve el logaritmo de x para el valor dado de base. Si no se especifica base, se devuelve el logaritmo natural de x.
math.log1p(x)	Devuelve el logaritmo natural de 1+x (base e)
math.log10(x)	Devuelve el logaritmo de base-10 de x

Tabla 7. Métodos matemáticos (continuación)

Método	Uso
<code>math.pow(x, y)</code>	Devuelve x elevado a la potencia de y. <code>pow(1.0, x)</code> y <code>pow(x, 0.0)</code> siempre devuelve 1, incluso si x es cero o NaN.
<code>math.sqrt(x)</code>	Devuelve la raíz cuadrada de x

Además de las funciones matemáticas, hay algunos métodos trigonométricos útiles. Estos métodos se muestran en la siguiente tabla.

Tabla 8. Métodos trigonométricos

Método	Uso
<code>math.acos(x)</code>	Devuelve el arco coseno de x en radianes
<code>math.asin(x)</code>	Devuelve el arcoseno de x en radianes
<code>math.atan(x)</code>	Devuelve el arco tangente de x en radianes
<code>math.atan2(y, x)</code>	Devuelve <code>atan(y / x)</code> en radianes.
<code>math.cos(x)</code>	Devuelve el coseno de x en radianes.
<code>math.hypot(x, y)</code>	Devuelve la norma euclidiana de <code>sqrt(x*x + y*y)</code> . Esta es la longitud del vector desde el origen al punto (x, y).
<code>math.sin(x)</code>	Devuelve el seno de x en radianes
<code>math.tan(x)</code>	Devuelve la tangente de x en radianes
<code>math.degrees(x)</code>	Convierte el ángulo x de radianes a grados
<code>math.radians(x)</code>	Convierte el ángulo x de grados a radianes
<code>math.acosh(x)</code>	Devuelve el coseno hiperbólico inverso de x
<code>math.asinh(x)</code>	Devuelve el seno hiperbólico inverso de x
<code>math.atanh(x)</code>	Devuelve la tangente hiperbólica inversa de x
<code>math.cosh(x)</code>	Devuelve el coseno hiperbólico de x
<code>math.sinh(x)</code>	Devuelve el seno hiperbólico de x
<code>math.tanh(x)</code>	Devuelve la tangente hiperbólica de x

También hay constantes matemáticas. El valor de `math.pi` es la constante matemática pi. El valor de `math.e` es la constante matemática e.

## Utilización de caracteres no ASCII

Para utilizar caracteres no ASCII, Python requiere la codificación y decodificación explícitas de las cadenas en Unicode. En IBM SPSS Modeler, se presupone que los scripts Python están codificados UTF-8, la cual es una codificación Unicode estándar que da soporte a caracteres no ASCII. El script siguiente se compilará porque SPSS Modeler ha establecido el compilador Python en UTF-8.

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

Sin embargo, el nodo resultante tendrá una etiqueta incorrecta.



Figura 3. Etiqueta del nodo que contiene caracteres no ASCII, visualiza incorrectamente

La etiqueta es incorrecta porque Python ha convertido el propio literal de serie en una cadena ASCII.

Python permite que los literales de cadenas Unicode se especifiquen añadiendo un prefijo con el carácter u antes del literal de cadena:

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

Esto crear una cadena Unicode y la etiqueta aparecerá correctamente.



Figura 4. Etiqueta de nodo que contiene caracteres no ASCII, visualizados correctamente

La utilización de Python y Unicode es un tema de gran volumen que no entra dentro del ámbito de este documento. Existen muchas publicaciones y recursos en línea disponibles que describen detalladamente este tema.

---

## Programación orientada a objetos

La programación orientada a objetos se basa en el concepto de crear un modelo del problema de destino en sus programas. La programación orientada a objetos disminuye los errores y promociona la reutilización del código. Python es un lenguaje orientado a objetos. Los objetos definidos en Python tienen las características siguientes:

- **Identidad.** Cada objeto debe ser distinguido y ello debe poder demostrarse mediante pruebas. Las pruebas `is` e `is not` existen para este fin.
- **Estado** Cada objeto debe ser capaz de almacenar el estado. Para este fin, existen atributos, tales como variables de instancias y campos.
- **Comportamiento.** Cada objeto debe ser capaz de manipular su estado. Para este fin existen métodos.

Python incluye las características siguientes para dar soporte a la programación orientada a objetos:

- **Creación de objetos basada en clases.** Las clases son plantillas para la creación de objetos. Los objetos son estructuras de datos con el comportamiento asociado.
- **Herencia con polimorfismo.** Python da soporte a la herencia individual y múltiple. Todos los métodos de instancias de Python son polimórficos y se pueden alterar temporalmente mediante subclases.

- **Encapsulación con ocultación de datos.** Python permite ocultar los atributos. Cuando se ocultan los atributos, se puede acceder a los mismos desde fuera de la clase únicamente mediante los métodos de la clase. Las clases implementan métodos para modificar los datos.

## Definición de una clase

En una clase Python, se pueden definir tanto variables como métodos. A diferencia de Java, en Python puede definir cualquier número de clases públicas por archivo de origen (o *module*). Por lo tanto, un módulo en Python puede considerarse similar a un paquete en Java.

En Python, las clases se definen utilizando la sentencia `class`. La sentencia `class` tiene el formato siguiente:

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

o

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

Cuando define una clase tiene la opción de proporcionar cero o más sentencias *assignment*. Estos crean atributos de clase que comparten todas las instancias de la clase. Puede proporcionar cero o más definiciones de *function*. Estas definiciones de función crean métodos. La lista de superclases es opcional.

El nombre de clase debe ser exclusivo en el mismo ámbito, esto es, dentro de un módulo, función o clase. Puede definir varias variables para que hagan referencia a la misma clase.

## Creación de una instancia de clase

Las clases se utilizan para contener (o compartir) los atributos de clase o para crear instancias de clase. Para crear una instancia de una clase, debe llamar a la clase como si fuera una función. Por ejemplo, considere las clases siguientes:

```
class MyClass:
    pass
```

Aquí, se utiliza la sentencia `pass` por que se requiere una sentencia para completar la clase, pero no se requiere ninguna acción de programación.

La sentencia siguiente crea una instancia de la clase `MyClass`:

```
x = MyClass()
```

## Añadir atributos a una instancia de clase

A diferencia de Java, en Python los clientes pueden añadir atributos a una instancia de una clase. Solo se cambia la instancia. Por ejemplo, para añadir atributos a una instancia de `x`, establezca valores nuevos en dicha instancia:

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

## Definición de atributos de clase y métodos

Cualquier variable enlazada a una clase es un *atributo de clase*. Cualquier función definida en una clase es un *método*. Los métodos reciben como primer argumento una instancia de la clase, que convencionalmente se denomina `self`. Por ejemplo, para definir algunos atributos de clase y métodos, puede entrar el siguiente código:

```
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
```

Dentro de una clase, debe cualificar todas las referencias a los atributos de clase con el nombre de clase; por ejemplo, `MyClass.attr1`. Todas las referencias a los atributos de la instancia deben cualificarse con la variable `self`, por ejemplo, `self.text`. Fuera de la clase, debe cualificar todas las referencias a los atributos de clase con el nombre de clase (por ejemplo, `MyClass.attr1`) o con una instancia de la clase (por ejemplo, `x.attr1`, donde `x` es una instancia de la clase). Fuera de la clase, todas las referencias a las variables de la instancia deben cualificarse con una instancia de la clase, por ejemplo, `x.text`.

## Variables ocultas

Los datos se pueden ocultar creando variables *privadas*. Solo la propia clase puede acceder a las variables privadas. Si declara nombres con el formato `__xxx` o `__xxx_yyy`, estos es, con dos signos de subrayado antes de los nombres, el analizador Python automáticamente añadirá el nombre de clase al nombre declarado y creará las variables ocultas, por ejemplo:

```
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 diferencia de Java, en Python todas las referencias a variables de instancia deben estar calificadas con `self`; no existe un uso implícito de `this`.

## Herencia

La posibilidad de herencia de las clases es fundamental en la programación orientada a objetos. Python da soporte a la herencia individual y múltiple. *Herencia individual* significa que solo puede haber una superclase. *Herencia múltiple* significa que puede haber más de una superclase.

La herencia se implementa generando subclases de otras clases. Cualquier número de clases Python pueden ser superclases. En la implementación de Jython en Python, solo se puede heredar directa o indirectamente de una clase Java. No es necesario suministrar una superclase.

Cualquier atributo o método de una superclase también está en cualquier subclase y lo puede utilizar la propia clase o cualquier cliente, siempre que el atributo o método no esté oculto. Se puede utilizar cualquier instancia de una subclase; esto se denomina *polimorfismo*. Estas características permiten la reutilización y facilitan la extensión.

Ejemplo

```
class Class1: pass    #no inheritance

class Class2: pass

class Class3(Class1): pass    #single inheritance

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

---

## Capítulo 3. Scripts de IBM SPSS Modeler

---

### Tipos de scripts

En IBM SPSS Modeler existen tres tipos de scripts:

- Los *scripts de ruta* se utilizan para controlar la ejecución de una sola ruta y se almacenan dentro de la ruta.
- Los *scripts Supernodo* se utilizan para controlar el comportamiento de los supernodos.
- Los *scripts autónomos o de sesión* se pueden utilizar para coordinar la ejecución entre un número de rutas diferentes.

Existen diferentes métodos disponibles que puede utilizar en scripts en IBM SPSS Modeler lo que le permite acceder a una amplia gama de funciones de SPSS Modeler. Estos métodos se utilizan también en Capítulo 4, “API de scripts”, en la página 35 para crear funciones más avanzadas.

---

### Rutas, rutas de supernodo y diagramas

La mayoría de las veces, el término *ruta* significa lo mismo independientemente de que se trate de una ruta cargada de un archivo o utilizada dentro de un supernodo. En general significa una colección de nodos conectados entre sí que puede ejecutarse. Sin embargo, en la creación de scripts no todas las operaciones se soportan en todos los sitios, lo que significa que el autor de un script deberá tener en cuenta qué variante de ruta está utilizando.

### Rutas

Una ruta es el principal tipo de documento de IBM SPSS Modeler. Se puede guardar, cargar, editar y ejecutar. Las rutas también pueden tener parámetros, valores globales, un script y otra información asociada a ellos.

### Rutas de Supernodo

Una *ruta de Supernodo* es el tipo de ruta que se utiliza en un Supernodo. Al igual que una ruta normal, contiene nodos enlazados entre sí. Las rutas de Supernodo tienen una serie de diferencias respecto de una ruta normal.

- Los parámetros y scripts están asociados al Supernodo propietario de la ruta de Supernodo en lugar de a la propia ruta de Supernodo.
- Las rutas de Supernodo tienen nodos de conector de entrada y salida adicionales dependiendo del tipo de Supernodo. Estos nodos de conector se utilizan en los flujos de información entrantes y salientes de la ruta de Supernodo y se crean automáticamente cuando se crea el Supernodo.

### Diagramas

El término *diagrama* abarca las funciones soportadas en rutas normales y en rutas de supernodo como, por ejemplo, la adición y eliminación de nodos y la modificación de conexiones entre nodos.

---

### Ejecución de una ruta

El ejemplo siguiente ejecuta todos los nodos ejecutables en la ruta y es el tipo de script de ruta más sencillo:

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

El ejemplo siguiente también se ejecuta todos los nodos ejecutables de la ruta:

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

En este ejemplo, la ruta se almacena en una variable denominada `stream`. Almacenar la ruta en una variable resulta útil ya que un script se utiliza generalmente para modificar la ruta o los nodos contenidos en una ruta. Si se crea una variable que almacena los resultados de la ruta, el script resultará más conciso.

## El contexto de los scripts

El módulo `modeler.script` proporciona el contexto en el que se ejecuta un script. El módulo se importa automáticamente a un script de SPSS Modeler durante la ejecución. El módulo define cuatro funciones que proporcionan un script con acceso a su entorno de ejecución:

- La función `session()` devuelve la sesión para el script. La sesión define información, tal como el entorno local y el proceso de fondo de SPSS Modeler (ya sea un proceso local o un proceso de SPSS Modeler Server conectado a la red) que se está utilizando para ejecutar rutas.
- La función `stream()` se puede utilizar con la ruta y los scripts Supernodo. Esta función devuelve la ruta que es propietaria del script de ruta o el script Supernodo que se está ejecutando.
- La función `diagram()` se puede utilizar con los scripts Supernodo. Esta función devuelve el diagrama dentro del Supernodo. Para otros tipos de script, esta función devuelve el mismo que la función `stream()`.
- La función `supernode()` se puede utilizar con los scripts Supernodo. Esta función devuelve el Supernodo propietario del script que se está ejecutando.

En la tabla siguiente se resumen las cuatro funciones y sus resultados.

Tabla 9. Resumen de las funciones de `modeler.script`

Tipo de script	<code>session()</code>	<code>stream()</code>	<code>diagram()</code>	<code>supernode()</code>
Autónomo	Devuelve una sesión	Devuelve la ruta gestionada actual en el momento en que se invoca el script (por ejemplo, la ruta se pasa con la opción <code>-stream</code> de modalidad de proceso por lotes) o <code>None</code> .	Igual que para <code>stream()</code>	No es aplicable
Ruta	Devuelve una sesión	Devuelve una ruta	Igual que para <code>stream()</code>	No es aplicable
Supernodo	Devuelve una sesión	Devuelve una ruta	Devuelve una ruta Supernodo	Devuelve un Supernodo

El módulo `modeler.script` también define un modo de finalizar el script con un código de salida. La función `exit(exit-code)` detiene la ejecución del script y devuelve el código de salida de entero suministrado.

Uno de los métodos que se define para una ruta es `runAll(List)`. Este método ejecuta todos los nodos ejecutables. Los modelos o resultados que se generan mediante la ejecución de los nodos se añaden a la lista suministrada.

Es común que la ejecución de ruta genere resultados, tales como modelos, gráficos y otros. Para capturar este resultado, un script puede proporcionar una variable que se inicializa en una lista, por ejemplo:



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

Cuando se completa la ejecución, se puede acceder a todos los objetos generados por la ejecución en la lista `results`.

---

## Referencia a nodos existentes

Una ruta suele estar construida previamente con algunos parámetros que se deben modificar antes de ejecutar la ruta. Para modificar estos parámetros se han de realizar las tareas siguientes:

1. Localizar los nodos en la ruta relevante.
2. Cambiar los valores de los nodos o de la ruta (o de ambas cosas).

## Buscar nodos

Las rutas proporcionan varios modos de localizar un nodo existente. Estos métodos se resumen en la siguiente tabla.

*Tabla 10. Métodos para localizar un nodo existente*

Método	Tipo devuelto	Descripción
<code>s.findAll(type, label)</code>	Colección	Devuelve una lista de todos los nodos con el tipo y la etiqueta. El tipo o la etiqueta pueden ser <code>None</code> , en cuyo caso se utiliza el otro parámetro.
<code>s.findAll(filter, recursive)</code>	Colección	Devuelve una colección de todos los nodos que están aceptados por el filtro especificado. Si el distintivo recursivo es <code>True</code> , también se buscan los supernodos contenidos en la ruta especificada.
<code>s.findById(id)</code>	Nodo	Devuelve el nodo con el ID proporcionado o <code>None</code> si no existe dicho nodo. La búsqueda se limita a la ruta actual.
<code>s.findByName(type, label)</code>	Nodo	Devuelve el nodo con el tipo, etiqueta o ambas cosas. El tipo o el nombre pueden ser <code>None</code> , en cuyo caso se utiliza el otro parámetro. Si varios nodos dan como resultado una coincidencia, se elige uno arbitrario y se devuelve. Si ningún nodo da como resultado una coincidencia, se devuelve el valor <code>None</code> .
<code>s.findDownstream(fromNodes)</code>	Colección	Busca en la lista de nodos suministrada y devuelve el conjunto de nodos en sentido descendente de los nodos suministrados. La lista devuelta incluye los nodos proporcionados originalmente.

Tabla 10. Métodos para localizar un nodo existente (continuación)

Método	Tipo devuelto	Descripción
<code>s.findUpstream(fromNodes)</code>	Colección	Busca en la lista de nodos suministrada y devuelve el conjunto de nodos en sentido ascendente de los nodos suministrados. La lista devuelta incluye los nodos proporcionados originalmente.

Por ejemplo, si una ruta contiene un nodo Filtro único que el script necesita para acceso, el nodo Filtro se puede encontrar utilizando el siguiente script:

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

Como alternativa, si se conoce el ID del nodo (tal como se muestra en la pestaña Anotaciones del cuadro de diálogo del nodo) se puede utilizar el ID para buscar el nodo, por ejemplo:

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

## Definición de propiedades

Los nodos, rutas, modelos y resultados tienen propiedades a las que se puede acceder y que, en la mayor parte de los casos, se pueden establecer. Las propiedades suelen utilizarse para modificar el aspecto o el comportamiento del objeto. En la tabla siguiente se resumen los métodos disponibles para establecer y acceder a las propiedades de los objetos.

Tabla 11. Métodos para establecer y acceder a las propiedades de los objetos

Método	Tipo devuelto	Descripción
<code>p.getPropertyValue(propertyName)</code>	Objeto	Devuelve el valor de la propiedad con nombre, o None si no existe tal propiedad.
<code>p.setPropertyValue(propertyName, value)</code>	No es aplicable	Establece el valor de la propiedad con nombre.
<code>p.setPropertyValues(properties)</code>	No es aplicable	Establece los valores de la propiedad con nombre. Cada entrada de la correlación de propiedades consta de una clave que representa el nombre de la propiedad y del valor que debe asignarse a la propiedad.
<code>p.getKeyedPropertyValue(propertyName, keyName)</code>	Objeto	Devuelve el valor de la propiedad con nombre, o None si no existe dicha propiedad o clave.
<code>p.setKeyedPropertyValue(propertyName, keyName, value)</code>	No es aplicable	Establece el valor de la propiedad con nombre y de la clave.

Por ejemplo, si desea establecer el valor de un nodo Archivo variable al comienzo de una ruta, puede utilizar el siguiente script:

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

Como alternativa, puede que desee filtrar un campo desde un nodo Filtrar. En este caso, el valor es con clave en el nombre de campo, por ejemplo:

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

## Creación de nodos y modificación de rutas

En algunas situaciones, es posible que desee añadir nuevos nodos a rutas existentes. Para añadir nodos a rutas existentes suele ser necesario realizar las tareas siguientes:

1. Crear los nodos.
2. Enlazar los nodos con el flujo de ruta existente.

### Creación de nodos

Las rutas proporcionan varios modos de crear nodos. Estos métodos se resumen en la siguiente tabla.

Tabla 12. Métodos para crear nodos

Método	Tipo devuelto	Descripción
<code>s.create(nodeType, name)</code>	Nodo	Crea un nodo del tipo especificado y lo añade a la ruta especificada.
<code>s.createAt(nodeType, name, x, y)</code>	Nodo	Crea un nodo del tipo especificado y lo añade a la ruta especificada en la ubicación especificada. Si $x < 0$ o $y < 0$ , no se establece la ubicación.
<code>s.createModelApplier(modelOutput, name)</code>	Nodo	Crea un nodo aplicador de modelos que se deriva del objeto de resultados del modelo proporcionado.

Por ejemplo, para crear un tipo de nodo nuevo en una ruta puede utilizar el siguiente script:

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

### Enlazar y desenlazar nodos

Cuando un nodo nuevo se crea dentro de una ruta, debe estar conectado a una ruta de nodos para poder utilizarlo. Las rutas proporciona varios métodos para enlazar y desenlazar nodos. Estos métodos se resumen en la siguiente tabla.

Tabla 13. Métodos para enlazar y desenlazar nodos

Método	Tipo devuelto	Descripción
<code>s.link(source, target)</code>	No es aplicable	Crea un nuevo enlace entre los nodos de origen y destino.
<code>s.link(source, targets)</code>	No es aplicable	Crea nuevos enlaces entre el nodo de origen y cada nodo de destino de la lista suministrada.

Tabla 13. Métodos para enlazar y desenlazar nodos (continuación)

Método	Tipo devuelto	Descripción
<code>s.linkBetween(inserted, source, target)</code>	No es aplicable	Conecta un nodo entre dos instancias de otro nodo (los nodos de origen y de destino) y establece la posición del nodo insertado de modo que quede entre ellos. Cualquier enlace directo entre los nodos de origen y de destino se elimina en primer lugar.
<code>s.linkPath(path)</code>	No es aplicable	Creación de una nueva ruta entre instancias de nodo. El primer nodo se enlaza con el segundo, el segundo nodo se enlaza con el tercero y así sucesivamente.
<code>s.unlink(source, target)</code>	No es aplicable	Elimina cualquier enlace directo entre los nodos de origen y de destino.
<code>s.unlink(source, targets)</code>	No es aplicable	Elimina los enlaces directos entre el nodo de origen y cada objeto de la lista de destinos.
<code>s.unlinkPath(path)</code>	No es aplicable	Elimina cualquier ruta que existe entre las instancias del nodo.
<code>s.disconnect(node)</code>	No es aplicable	Elimina los enlaces entre el nodo suministrado y todos los demás nodos de la ruta especificada.
<code>s.isValidLink(source, target)</code>	<i>booleano</i>	Devuelve True si es válido crear un enlace entre los nodos de origen y de destino especificados. Este método comprueba que ambos objetos pertenezcan a la ruta especificada, que el nodo de origen puede proporcionar un enlace y que el nodo de destino puede recibir un enlace, y que la creación de un enlace de este tipo no creará un circularidad en la ruta.

El script de ejemplo siguiente realiza estas cinco tareas:

1. Crea un nodo de entrada Archivo de variables, un nodo Filtro y un nodo de salida Tabla.
2. Conecta los nodos entre sí.
3. Establece el nombre de archivo del nodo de entrada Archivo de variables.
4. Filtra el campo "Drug" de la salida resultante.
5. Ejecute el nodo Tabla.

```
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)
```

## Importar, sustituir y eliminar nodos

Además de crear y conectar nodos, a menudo es necesario sustituir y suprimir nodos de la corriente. Los métodos que están disponibles para importar, sustituir y suprimir nodos se resumen en la tabla siguiente.

Tabla 14. Métodos para importar, sustituir y suprimir nodos

Método	Tipo devuelto	Descripción
<code>s.replace(originalNode, replacementNode, discardOriginal)</code>	No es aplicable	Sustituye el nodo especificado de la ruta actual. Tanto el nodo original como el nodo de sustitución deben ser propiedad de la ruta especificada.
<code>s.insert(source, nodes, newIDs)</code>	Lista	Inserta copias de los nodos en la lista suministrada. Se presupone que todos los nodos de la lista suministrada se encuentran dentro de la ruta especificada. El distintivo <code>newIDs</code> indica si se deben generar nuevos ID para cada nodo o si se debe copiar y utilizar el ID existente. Se presupone que todos los nodos de una ruta tienen ID exclusivos, por lo tanto este distintivo se debe establecer en <code>True</code> si la ruta de origen es la misma que la ruta especificada. El método devuelve la lista de nodos recién insertados, en la que el orden de los nodos está sin definir (es decir, el orden no es necesariamente el mismo que el orden de los nodos de la lista de entrada).
<code>s.delete(node)</code>	No es aplicable	Elimina el nodo especificado de la ruta especificada. El nodo debe ser propiedad de la ruta especificada.
<code>s.deleteAll(nodes)</code>	No es aplicable	Elimina todos los nodos especificados de la ruta especificada. Todos los nodos de la colección debe pertenecer a la ruta especificada.
<code>s.clear()</code>	No es aplicable	Elimina todos los nodos de la ruta especificada.

## Atravesar los nodos de una ruta

Un requisito común es identificar los nodos que están en un punto de la ruta anterior o posterior a un determinado nodo. La ruta proporciona una serie de métodos que pueden utilizarse para identificar estos nodos. Estos métodos se resumen en la siguiente tabla.

Tabla 15. Métodos para identificar los nodos en sentido ascendente y descendente

Método	Tipo devuelto	Descripción
<code>s.iterator()</code>	Iterador	Devuelve un iterador de los objetos de nodo que están contenidos en la ruta especificada. Si la ruta se modifica entre llamadas a la función <code>next()</code> , el comportamiento del iterador no está definido.

Tabla 15. Métodos para identificar los nodos en sentido ascendente y descendente (continuación)

Método	Tipo devuelto	Descripción
s.predecessorAt(node, index)	Nodo	Devuelve el predecesor inmediato especificado del nodo suministrado o None si el índice está fuera de los límites.
s.predecessorCount(node)	int	Devuelve el número de predecesores inmediatos del nodo suministrado.
s.predecessors(node)	Lista	Devuelve los predecesores inmediatos del nodo suministrado.
s.successorAt(node, index)	Nodo	Devuelve el sucesor inmediato especificado del nodo suministrado o None si el índice está fuera de los límites.
s.successorCount(node)	int	Devuelve el número de sucesores inmediatos del nodo suministrado.
s.successors(node)	Lista	Devuelve los sucesores inmediatos del nodo suministrado.

## Obtener información sobre los nodos

Los nodos entran en diferentes categorías, tales como nodos de importación y exportación de datos, nodos de construcción de modelos y otros tipos de nodos. Cada nodo proporciona una serie de métodos que pueden utilizarse para obtener información sobre el nodo.

Los métodos que se pueden utilizar para obtener el ID, el nombre y la etiqueta de un nodo se resumen en la tabla siguiente.

Tabla 16. Métodos para obtener el ID, el nombre y la etiqueta de un nodo

Método	Tipo devuelto	Descripción
n.getLabel()	cadena	Devuelve la etiqueta de visualización del nodo especificado. La etiqueta es el valor de la propiedad custom_name sólo si la propiedad es una serie no vacía y la propiedad use_custom_name no está establecida; de lo contrario, la etiqueta es el valor de getName().
m.setLabel(label)	No es aplicable	Establece la etiqueta de visualización del nodo especificado. Si la nueva etiqueta es una cadena no vacía se asigna a la propiedad custom_name, y False se asigna a la propiedad use_custom_name, de tal modo que la etiqueta especificada tiene prioridad; de lo contrario, se asigna una serie vacía a la propiedad custom_name y se asigna True a la propiedad use_custom_name.
n.getName()	cadena	Devuelve el nombre del nodo especificado.

Tabla 16. Métodos para obtener el ID, el nombre y la etiqueta de un nodo (continuación)

Método	Tipo devuelto	Descripción
n.getID()	<i>cadena</i>	Devuelve el ID del nodo especificado. Se crea un ID nuevo cada vez que se crea un nuevo nodo. El ID se conserva con el nodo cuando se guarda como parte de una ruta de modo que, cuando se abre la ruta, los ID de nodo se conservan. Sin embargo, si un nodo guardado se inserta en una ruta, el nodo insertado se considera un nuevo objeto y se le asigna un nuevo ID.

Los métodos que se pueden utilizar para obtener información acerca de un nodo se resumen en la tabla siguiente.

Tabla 17. Métodos para obtener información acerca de un nodo

Método	Tipo devuelto	Descripción
n.getTypeName()	<i>cadena</i>	Devuelve el nombre de script de este nodo. Este es el mismo nombre que puede utilizarse para crear una nueva instancia de este nodo.
n.isInitial()	<i>Booleana</i>	Devuelve True si es un nodo <i>initial</i> , esto es, un nodo que aparece al inicio de una ruta.
n.isInline()	<i>Booleana</i>	Devuelve True si es un nodo <i>in-line</i> , esto es, un nodo que aparece a mitad de una ruta.
n.isTerminal()	<i>Booleana</i>	Devuelve True si es un nodo <i>terminal</i> , esto es, un nodo que aparece al final de una ruta.
n.getXPosition()	<i>int</i>	Devuelve el desplazamiento de la posición x del nodo en la ruta.
n.getYPosition()	<i>int</i>	Devuelve el desplazamiento de la posición y del nodo en la ruta.
n.setXYPosition(x, y)	No es aplicable	Devuelve la posición del nodo en la ruta.
n.setPositionBetween(source, target)	No es aplicable	Establece la posición del nodo en la ruta, de modo que esté posicionado entre los nodos suministrados.
n.isCacheEnabled()	<i>Booleana</i>	Devuelve True si la memoria caché está habilitada; devuelve False de lo contrario.
n.setCacheEnabled(val)	No es aplicable	Habilita o inhabilita la memoria caché para este objeto. Si la memoria caché está llena y la memoria caché pasa a estar inhabilitada, la memoria caché se vacía.
n.isCacheFull()	<i>Booleana</i>	Devuelve True si la memoria caché está llena; devuelve False de lo contrario.

Tabla 17. Métodos para obtener información acerca de un nodo (continuación)

<b>Método</b>	<b>Tipo devuelto</b>	<b>Descripción</b>
n.flushCache()	No es aplicable	Vacía la memoria caché de este nodo. No tiene efecto si la memoria caché no está activada o no está llena.



---

## Capítulo 4. API de scripts

---

### Introducción a la API de scripts

La API de scripts proporciona acceso a una amplia gama de funciones de SPSS Modeler. Todos los métodos descritos hasta ahora forman parte de la API y se puede acceder a los mismos de forma implícita en el script sin importaciones adicionales. Sin embargo, si desea hacer referencia a las clases de la API, debe importar la API explícitamente con la sentencia siguiente:

```
import modeler.api
```

Esta sentencia import es necesaria para muchos de los ejemplos de la API de scripts.

---

### Ejemplo: Buscar nodos utilizando un filtro personalizado

La sección “Buscar nodos” en la página 27 se incluye un ejemplo de cómo buscar un nodo en una ruta utilizando el nombre de tipo del nodo como criterio de búsqueda. En algunas situaciones, se requiere una búsqueda más genérica y ésta se puede implementar utilizando la clase `NodeFilter` y el método `findAll()` de la ruta. Este tipo de búsqueda requiere los pasos siguientes:

1. Crear una clase nueva que amplíe `NodeFilter` e implemente una versión personalizada del método `accept()`.
2. Llamar al método `findAll()` de la ruta con una instancia de esta clase nueva. Esto devuelve todos los nodos que cumplen el criterio definido en el método `accept()`.

El ejemplo siguiente muestra cómo buscar nodos de una ruta que tienen habilitada la memoria caché de nodo. La lista de nodos devuelta se puede utilizar para vaciar o inhabilitar las memorias caché de estos nodos.

```
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)
```

---

### Metadatos: información sobre datos

Puesto que en una ruta los nodos se conectan entre sí, está disponible la información relativa a las columnas o los campos disponibles en cada nodo. Por ejemplo, en la interfaz de usuario de Modeler esto permite seleccionar por qué campos hay que ordenar o agregar. Esta información se llama modelo de datos.

Los scripts también pueden acceder al modelo de datos inspeccionando los campos que entran en un nodo o que salen de él. En algunos nodos coinciden los modelos de datos de entrada y salida; por ejemplo, un nodo Ordenar se limita a cambiar el orden de los registros, sin alterar el modelo de datos. Otros, como el nodo Derivar, pueden añadir nuevos campos. Otros, como el nodo Filtrar, pueden renombrar o eliminar campos.

En el ejemplo siguiente, el script toma la ruta estándar IBM SPSS Modeler `druglearn.str` y, para cada campo, construye un modelo descartando uno de los campos de entrada. Lo hace de la siguiente manera:

1. Accede al modelo de datos de salida del nodo Tipo.
2. Itera cada campo del modelo de datos de salida.

3. Modifica el nodo Filtro de cada campo de entrada.
4. Cambia el nombre del modelo que se construye.
5. Ejecuta el nodo de construcción de modelos.

**Nota:** Antes de ejecutar el script en la ruta `druglean.str`, no olvide establecer el lenguaje de script a Python (la ruta se creó en una versión anterior de IBM SPSS Modeler, de modo que el lenguaje de script de la ruta está establecido a Herencia).

```
import modeler.api

stream = modeler.script.stream()
filternode = stream.findByType("filter", None)
typenode = stream.findByType("type", None)
c50node = stream.findByType("c50", None)
# Usar siempre un nombre de modelo personalizado
c50node.setPropertyValue("use_model_name", True)

lastRemoved = None
fields = typenode.getOutputDataModel()
for field in fields:
    # Si este es el campo de destino, se hace caso omiso del mismo
    if field.getModelingRole() == modeler.api.ModelingRole.OUT:
        continue

    # Se rehabilita el campo eliminado más recientemente
    if lastRemoved != None:
        filternode.setKeyedPropertyValue("include", lastRemoved, True)

    # Se elimina el campo
    lastRemoved = field.getColumnName()
    filternode.setKeyedPropertyValue("include", lastRemoved, False)

    # Se establece el nombre del nuevo modelo y se ejecuta la construcción
    c50node.setPropertyValue("model_name", "Exclude " + lastRemoved)
    c50node.run([])
```

El objeto `DataModel` (modelo de datos) proporciona una serie de métodos de acceso a la información relativa a los campos y columnas del modelo de datos. Estos métodos se resumen en la siguiente tabla.

*Tabla 18. Métodos del objeto `DataModel` de acceso a la información relativa a campos o columnas*

Método	Tipo devuelto	Descripción
<code>d.getColumnCount()</code>	<i>int</i>	Devuelve el número de columnas del modelo de datos.
<code>d.columnIterator()</code>	Iterator	Devuelve un iterador que devuelve cada columna en el orden "natural" de inserción. El iterador devuelve instancias de <code>Column</code> .
<code>d.nameIterator()</code>	Iterator	Devuelve un iterador que devuelve el nombre de cada columna en el orden "natural" de inserción.
<code>d.contains(nombre)</code>	<i>Boolean</i>	Devuelve <code>True</code> si en este <code>DataModel</code> existe una columna con el nombre proporcionado y <code>False</code> en caso contrario.
<code>d.getColumn(nombre)</code>	<code>Column</code>	Devuelve la columna cuyo nombre es el especificado.
<code>d.getColumnGroup(nombre)</code>	<code>ColumnGroup</code>	Devuelve el grupo de columnas nombrado o <code>None</code> si no existe dicho grupo de columnas.

Tabla 18. Métodos del objeto *DataModel* de acceso a la información relativa a campos o columnas (continuación)

Método	Tipo devuelto	Descripción
d.getColumnGroupCount()	<i>int</i>	Devuelve el número de grupos de columnas de este modelo de datos.
d.columnGroupIterator()	Iterator	Devuelve un iterador que devuelve a su vez cada grupo de columnas.
d.toArray()	Columnn[]	Devuelve el modelo de datos como un vector de columnas. Las columnas van ordenadas según su orden "natural" de inserción.

Cada campo (objeto *Column*) incluye una serie de métodos de acceso a la información de la columna. La tabla que se muestra a continuación muestra una selección de los mismos.

Tabla 19. Métodos del objeto *Column* de acceso a la información de una columna

Método	Tipo devuelto	Descripción
c.getColumnName()	<i>string</i>	Devuelve el nombre de la columna.
c.getColumnLabel()	<i>string</i>	Devuelve la etiqueta de la columna o una cadena vacía si no hay ninguna etiqueta asociada a la columna.
c.getMeasureType()	MeasureType	Devuelve el tipo de medición de la columna.
c.getStorageType()	StorageType	Devuelve el tipo de almacenamiento de la columna.
c.isMeasureDiscrete()	<i>Boolean</i>	Devuelve True si la columna es discreta. Se consideran discretas las columnas que son un conjunto o un distintivo.
c.isModelOutputColumn()	<i>Boolean</i>	Devuelve True si la columna es una columna de resultado del modelo.
c.isStorageDatetime()	<i>Boolean</i>	Devuelve True si el almacenamiento de la columna es un valor de hora, fecha o indicación de fecha y hora.
c.isStorageNumeric()	<i>Boolean</i>	Devuelve True si el almacenamiento de la columna es un entero o un número real.
c.isValidValue(value)	<i>Boolean</i>	Devuelve True si el valor especificado es válido para este almacenamiento y valid cuando se conocen los valores de columna válidos.
c.getModelingRole()	ModelingRole	Devuelve el rol de modelado de la columna.
c.getSetValues()	Object[]	Devuelve un vector de valores válidos para la columna o None si se desconocen los valores o si la columna no es un conjunto.
c.getValueLabel(valor)	<i>string</i>	Devuelve la etiqueta del valor de la columna o una cadena vacía si no hay ninguna etiqueta asociada al valor.

Tabla 19. Métodos del objeto Column de acceso a la información de una columna (continuación)

Método	Tipo devuelto	Descripción
c.getFalseFlag()	Object	Devuelve el valor indicador de "falso" de la columna o None si se desconoce el valor o la columna no es un indicador.
c.getTrueFlag()	Object	Devuelve el valor indicador de "verdadero" de la columna o None si se desconoce el valor o la columna no es un indicador.
c.getLowerBound()	Object	Devuelve el valor del límite inferior de los valores de la columna o None si se desconoce el valor o si la columna no es continua.
c.getUpperBound()	Object	Devuelve el valor del límite superior de los valores de la columna o None si se desconoce el valor o si la columna no es continua.

Observe que la mayoría de los métodos de acceso a la información de una columna tienen métodos equivalentes definidos en el propio objeto DataModel. Por ejemplo, las dos sentencias siguientes son equivalentes:

```
dataModel.getColumn("unNombre").getModelRole()
dataModel.getModelRole("unNombre")
```

## Acceso a objetos generados

Ejecución de una corriente normalmente implica producir objetos de salida adicionales. Estos objetos adicionales pueden ser un nuevo modelo o un fragmento de la salida que proporciona información para utilizarla en las ejecuciones posteriores.

En el ejemplo siguiente, la ruta `druglearn.str` se utiliza de nuevo como punto de partida para la ruta. En este ejemplo, se ejecutan todos los nodos de la ruta y los resultados se almacenan en una lista. A continuación, el script crea un bucle por los resultados y se guarda cualquier salida del modo resultante de la ejecución como un archivo de modelo de IBM SPSS Modeler (.gm) y se exporta como 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 clase `taskrunner` proporciona un modo práctico de ejecutar diferentes tareas comunes. Los métodos que están disponibles en esta clase se resumen en la tabla siguiente.

*Tabla 20. Métodos de la clase `taskrunner` para realizar tareas comunes*

Método	Tipo devuelto	Descripción
<code>t.createStream(name, autoConnect, autoManage)</code>	Ruta	Crea y devuelve una nueva ruta. Tenga en cuenta que el código que debe crear las rutas de forma privada sin que las vea el usuario debe establecer el distintivo <code>autoManage</code> en <code>False</code> .
<code>t.exportDocumentToFile(documentOutput, filename, fileFormat)</code>	No es aplicable	Exporta la descripción de la ruta a un archivo utilizando el formato de archivo especificado.
<code>t.exportModelToFile(modelOutput, filename, fileFormat)</code>	No es aplicable	Exporta el modelo a un archivo utilizando el formato de archivo especificado.
<code>t.exportStreamToFile(stream, filename, fileFormat)</code>	No es aplicable	Exporta la ruta a un archivo utilizando el formato de archivo especificado.
<code>t.insertNodeFromFile(filename, diagram)</code>	Nodo	Lee un nodo en el archivo especificado y lo devuelve insertándolo en el diagrama suministrado. Tenga en cuenta que lo pueden utilizar los objetos <code>Nodo</code> y <code>Supermodo</code> .
<code>t.openDocumentFromFile(filename, autoManage)</code>	<code>DocumentOutput</code>	Lee un nodo en el archivo especificado y lo devuelve.
<code>t.openModelFromFile(filename, autoManage)</code>	<code>ModelOutput</code>	Lee un modelo en el archivo especificado y lo devuelve.
<code>t.openStreamFromFile(filename, autoManage)</code>	Ruta	Lee una ruta en el archivo especificado y la devuelve.
<code>t.saveDocumentToFile(documentOutput, filename)</code>	No es aplicable	Guarda el documento en la ubicación de archivo especificada.
<code>t.saveModelToFile(modelOutput, filename)</code>	No es aplicable	Guarda el modelo en la ubicación de archivo especificada.
<code>t.saveStreamToFile(stream, filename)</code>	No es aplicable	Guarda la ruta en la ubicación de archivo especificada.

## Tratamiento de errores

El lenguaje Python proporciona manejo de errores mediante el bloque de código `try...except`. Se puede utilizar en los scripts para capturar excepciones y manejar los problemas que podrían ocasionar la finalización del script.

En el script de ejemplo siguiente, se realiza un intento para recuperar un modelo desde IBM SPSS Collaboration and Deployment Services Repository. Esta operación puede hacer que se genere una

excepción, por ejemplo, es posible que las credenciales de inicio de sesión en el repositorio no se hayan configurado correctamente o que la ruta del repositorio sea errónea. En el script, esto puede generar una excepción `ModelerException` (todas las excepciones que genera IBM SPSS Modeler se derivan de `modeler.api.ModelerException`).

```
import modeler.api

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

**Nota:** Algunas operaciones de scripts pueden generar excepciones Java estándar, estas excepciones no se derivan de `ModelerException`. Para capturar estas excepciones, se puede utilizar un bloque `except` adicional que capture todas las excepciones Java, por ejemplo:

```
import modeler.api

session = modeler.script.session()
try:
    tepo = 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()
```

## Parámetros de ruta, sesión y Supernodo

Los parámetros proporcionan una forma útil de pasar valores en el momento de la ejecución, en lugar de codificarlos directamente en un script. Los parámetros y sus valores se definen de la misma forma que las corrientes, es decir, como entradas de la tabla de los parámetros de una corriente o Supernodo o como parámetros de la línea de comandos. Las clases de Corriente y Supernodo implementan un conjunto de funciones definidas por el objeto `ParameterProvider`, como se muestra en la tabla siguiente. La sesión proporciona una llamada `getParameters()` que devuelve un objeto que define dichas funciones.

Tabla 21. Funciones definidas por el objeto `ParameterProvider`

Método	Tipo devuelto	Descripción
<code>p.parameterIterator()</code>	Iterador	Devuelve un iterador de nombres de parámetro para este objeto.
<code>p.getParameterDefinition(parameterName)</code>	<code>ParameterDefinition</code>	Devuelve la definición de parámetro para el parámetro con el nombre especificado, o <code>None</code> si no existe tal parámetro en este proveedor. El resultado puede ser una instantánea de la definición en el momento en que el método se ha llamado y que no necesariamente refleja las modificaciones posteriores realizadas en el parámetro a través de este proveedor.
<code>p.getParameterLabel(parameterName)</code>	<i>cadena</i>	Devuelve la etiqueta del parámetro con nombre, o <code>None</code> si no existe tal parámetro.

Tabla 21. Funciones definidas por el objeto *ParameterProvider* (continuación)

Método	Tipo devuelto	Descripción
p.setParameterLabel(parameterName, label)	No es aplicable	Establece la etiqueta del parámetro con nombre.
p.getParameterStorage(parameterName)	ParameterStorage	Devuelve el almacenamiento del parámetro con nombre, o None si no existe tal parámetro.
p.setParameterStorage(parameterName, storage)	No es aplicable	Establece el almacenamiento del parámetro con nombre.
p.getParameterType(parameterName)	ParameterType	Devuelve el tipo del parámetro con nombre, o None si no existe tal parámetro.
p.setParameterType(parameterName, type)	No es aplicable	Establece el tipo del parámetro con nombre.
p.getParameterValue(parameterName)	Objeto	Devuelve el valor del parámetro con nombre, o None si no existe tal parámetro.
p.setParameterValue(parameterName, value)	No es aplicable	Establece el valor del parámetro con nombre.

En el ejemplo siguiente, el script agrega algunos datos Telco para averiguar qué región tiene los datos de promedio de ingresos más bajos. A continuación, se establece un parámetro de ruta con esta región. Este parámetro de ruta se utiliza en un nodo Seleccionar para excluir dicha región de los datos, antes de que se cree un modelo de abandono en el resto.

El ejemplo es artificial porque el script genera el propio nodo Seleccionar y, por lo tanto, podría haber generado el valor correcto directamente en la expresión del nodo Seleccionar. Sin embargo, las rutas se suelen construir previamente, de modo que establecer los parámetros de este modo proporciona un ejemplo útil.

La primera parte del script de ejemplo crea el parámetro de ruta que contendrá la región con el promedio de ingresos más bajo. El script también crea los nodos de la rama de agregación y la rama de creación de modelos y los conecta.

```
import modeler.api

stream = modeler.script.stream()

# Inicializar un parámetro de ruta
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# Crear primero la rama de agregación para calcular el promedio de ingresos por región
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")
# Hacer referencia al parámetro de ruta en la selección
```

```

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)

```

El script de ejemplo crea la ruta siguiente.

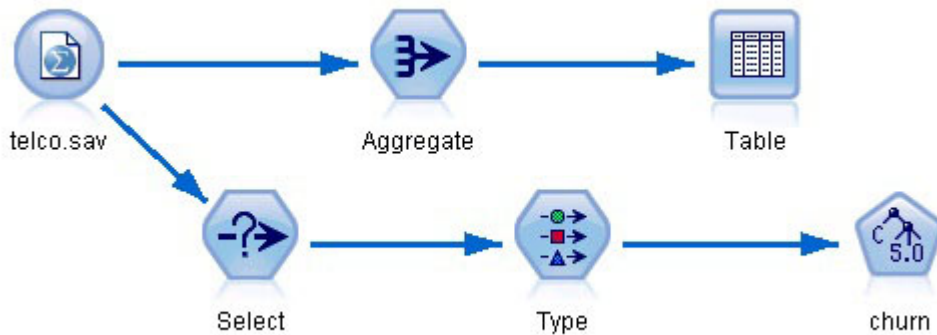


Figura 5. Ruta resultante del script de ejemplo

La parte siguiente del script de ejemplo ejecuta el nodo Tabla al final de la rama de agregación.

```

# Ejecutar primero el nodo Tabla
results = []
tablenode.run(results)

```

La parte siguiente del script de ejemplo accede a la salida de la tabla que ha generado la ejecución del nodo Tabla. A continuación, el script itera por las filas de la tabla, buscando la región con el promedio de ingresos más bajo.

```

# Ejecutar el nodo tabla para generar una sola tabla como salida
table = results[0]

```

```

# la salida de la tabla contiene un RowSet que permite acceder a valores como filas y columnas
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

```

```

# Del modo que se ha definido el nodo, la primera columna
# contiene la región y la segunda contiene el promedio de ingresos
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 parte siguiente del script utiliza la región con el promedio de ingresos más bajo para establecer el parámetro de ruta "LowestRegion" creado anteriormente. El script ejecuta el constructor de modelos excluyendo la región especificada de los datos de formación.



```

# Comprobar que se ha asignado un valor
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

# Finalmente, ejecutar el constructor de modelos con el criterio de selección
c50node.run([])

A continuación, se muestra el script de ejemplo.
import modeler.api

stream = modeler.script.stream()

# Crear un parámetro de ruta
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# Crear primero la rama de agregación para calcular el promedio de ingresos por región
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")
# Hacer referencia al parámetro de ruta en la selección
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)

# Ejecutar primero el nodo Tabla
results = []
tablenode.run(results)

# Ejecutar el nodo tabla para generar una sola tabla como salida
table = results[0]

# la salida de la tabla contiene un RowSet que permite acceder a valores como filas y columnas
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# Del modo que se ha definido el nodo, la primera columna
# contiene la región y la segunda contiene el promedio de ingresos
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

# Comprobar que se ha asignado un valor
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

# Finalmente, ejecutar el constructor de modelos con el criterio de selección
c50node.run([])

```

## Valores globales

Los valores globales se utilizan para calcular diferentes estadísticas de resumen para los campos especificados. Se puede acceder a estos valores de resumen desde cualquier lugar de la ruta. Los valores globales son similares a los parámetros de ruta, ya que se puede acceder a los mismos por nombre a través de la ruta. Se diferencian de los parámetros de ruta en que los valores asociados se actualizan automáticamente cuando se ejecuta uno nodo Establecer globales, en lugar de asignarlos mediante script o desde la línea de comandos. Se accede a los valores globales de una ruta invocando el método `getGlobalValues()` de la ruta.

El objeto `GlobalValues` define las funciones que se muestran en la tabla siguiente.

*Tabla 22. Funciones definidas por el objeto GlobalValues*

Método	Tipo devuelto	Descripción
<code>g.fieldNameIterator()</code>	Iterador	Devuelve un iterador para cada nombre de campo con al menos un valor global.
<code>g.getValue(type, fieldName)</code>	Objeto	Devuelve el valor global para el tipo especificado y nombre de campo, o <code>None</code> si no se puede localizar ningún valor. Generalmente se espera que el valor devuelto sea un número, aunque en las funciones futuras se pueden devolver tipos de valores diferentes.
<code>g.getValues(fieldName)</code>	Mapa	Devuelve un mapa que contiene las entradas conocidas para el nombre de campo especificado o <code>None</code> si no hay entradas para el campo.

`GlobalValues.Type` define el tipo de estadísticas de resumen disponibles. Están disponibles las siguientes estadísticas de resumen:

- MAX: el valor máximo del campo.
- MEAN: el valor medio del campo.
- MIN: el valor mínimo del campo.
- STDDEV: la desviación estándar del campo.
- SUM: la suma de los valores del campo.

Por ejemplo, el script siguiente accede el valor medio del campo "income" que calcula un nodo Val. globales:

```

import modeler.api

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

```

---

## Trabajar con varias rutas: Scripts autónomos

Para trabajar con varias rutas se debe utilizar un script autónomo. El script autónomo se puede editar y ejecutar dentro de la interfaz de usuario de IBM SPSS Modeler o se puede pasar como un parámetro de línea de comandos en modalidad de proceso por lotes.

El siguiente script autónomo abre dos rutas. Una de estas rutas crea un modelo y la segunda ruta traza la distribución de los valores predichos.

```
# 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)
derivnode = 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(derivnode, histogramnode)
plotstream.linkBetween(applyc50, derivnode, histogramnode)
histogramnode.setPropertyValue("color_field", "$C-Drug")
histogramnode.run([])

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



---

## Capítulo 5. Sugerencias sobre scripts

Esta sección proporciona una visión general de las técnicas y sugerencias para utilizar scripts, incluida la modificación de la ejecución de la ruta, la utilización de una contraseña codificada en un script.

---

### Modificación de la ejecución de una ruta

Cuando se ejecuta una ruta, sus nodos terminales se ejecutan en un orden optimizado para la situación predeterminada. En algunos casos, es posible que prefiera un orden de ejecución diferente. Para modificar el orden de ejecución de una ruta, lleve a cabo los siguientes pasos desde la pestaña Ejecución del cuadro de diálogo Propiedades de ruta:

1. Comience con un script vacío.
2. Pulse en el botón **Añadir script predeterminado** de la barra de herramientas y añada el script de ruta predeterminado.
3. Cambie el orden de las instrucciones del script de ruta predeterminado por el orden en que desee que se ejecuten las instrucciones.

---

### Trabajo con modelos

Si está habilitada la sustitución automática de modelo en IBM SPSS Modeler y se ejecuta un nodo generador de modelos a través de la interfaz de usuario de IBM SPSS Modeler, se sustituye un nugget de modelo existente enlazado al generador de modelos por el nuevo nugget de modelo. Si se ejecuta el nodo generador de modelos mediante un script, el nugget de modelo enlazado existente no se sustituye. Para sustituir el nugget de modelo existente debe especificar explícitamente la sustitución del nugget en el script.

---

### Generación de una contraseña codificada

En algunos casos, puede que necesite incluir una contraseña en un script. Por ejemplo, es posible que desee acceder a un origen de datos protegido con contraseña. Las contraseñas codificadas pueden utilizarse en:

- Propiedades de los nodos para un origen de base de datos y nodos de resultado
- Argumentos de línea de comando para conectarse al servidor
- Propiedades de conexión con la base de datos almacenadas en un archivo *.par* (archivo de parámetro generado desde la pestaña Publicar de un nodo de exportación)

En la interfaz de usuario se proporciona una herramienta para generar contraseñas codificadas en función del algoritmo Blowfish (si desea obtener más información, consulte <http://www.schneier.com/blowfish.html>). Una vez codificada, puede copiar y almacenar la contraseña en archivos de script y argumentos de líneas de comando. La propiedad de nodo *epassword* utilizada para *database* y *databaseexport* almacena la contraseña codificada.

1. Para generar una contraseña codificada, en el menú Herramientas seleccione:  
**Codificar contraseña...**
2. Especifique una contraseña en el cuadro de texto Contraseña.
3. Pulse en **Codificar** para generar una codificación aleatoria de la contraseña.
4. Pulse en el botón Copiar para copiar la contraseña codificada al Portapapeles.
5. Pegue la contraseña en el script o parámetro deseado.

---

## Comprobación por script

Puede comprobar rápidamente la sintaxis de todos los tipos de scripts pulsando en el botón de comprobación de la barra de herramientas del cuadro de diálogo Script de ruta.



Figura 6. Iconos de barra de herramientas del script de ruta

En la comprobación por script se avisa de cualquier error que se detecte en el código y se sugieren recomendaciones de mejora. Para ver la línea con errores, pulse en los comentarios, en la mitad inferior del cuadro de diálogo. Los errores se señalan en rojo.

---

## Scripts desde la línea de comandos

El uso scripts permite ejecutar operaciones típicamente desarrolladas en la interfaz de usuario. Simplemente especifique y ejecute una ruta independiente en la línea de comandos cuando ejecute IBM SPSS Modeler.

Por ejemplo:

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

La marca `-script` carga el script especificado, mientras que la marca `-execute` ejecuta todos los comandos del archivo de script.

---

## Especificación de rutas de archivos

Al especificar rutas a directorios y archivos, puede utilizar una sola barra inclinada (/) o una doble barra invertida (\\) como separador de directorios; por ejemplo:

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

o

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

---

## Compatibilidad con versiones anteriores

Los scripts de herencia creados en versiones anteriores de IBM SPSS Modeler deberían funcionar normalmente sin cambios en la versión actual. Para que los scripts funcionen, debe seleccionarse **De herencia** en la pestaña del script de ruta del cuadro de diálogo Propiedades de ruta o del cuadro de diálogo Script autónomo. Los nuggets de modelos podrán ahora insertarse en la ruta automáticamente (es el comportamiento predeterminado) y podrán sustituir o complementar un nugget existente del tipo en la ruta. El que esto ocurra depende de la configuración de las opciones **Añadir modelo a ruta** y **Sustituir modelo anterior** (**Herramientas > Opciones > Opciones de usuario > Notificaciones**). Por ejemplo, es posible que tenga que modificar un script de una versión anterior en el que la sustitución del nugget se trate borrando el nugget existente e insertando uno nuevo.

Es posible que los scripts creados en esta versión no funcionen en versiones anteriores.

Los scripts Python creados en esta versión no funcionan en versiones anteriores.

Si un script creó una liberación antigua utiliza un comando que se ha sustituido desde entonces (o desaprobadado), la forma antigua se seguirá admitiendo, pero aparecerá un mensaje de advertencia. Por

ejemplo, la palabra clave antigua `generated` se ha sustituido por `model`, y `clear generated` se ha sustituido por `clear generated palette`. Los scripts que utilizan las formas antiguas se seguirán ejecutando, pero se mostrará una advertencia.





---

## Capítulo 6. Argumentos de la línea de comandos

---

### Invocación del software

Puede utilizar la línea de comandos del sistema operativo para iniciar IBM SPSS Modeler de la siguiente manera:

1. En un ordenador en el que se haya instalado IBM SPSS Modeler, abra una ventana de DOS o del indicador de comandos.
2. Para iniciar la interfaz de IBM SPSS Modeler en modo interactivo, escriba el comando `clementine` seguido de los argumentos necesarios; por ejemplo:

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

Los argumentos disponibles (modificadores) permiten conectar con un servidor, cargar rutas, ejecutar scripts o especificar otros parámetros, según sea necesario.

---

### Uso de argumentos en la línea de comandos

Se pueden añadir argumentos de línea de comandos (también denominados **marcas**) al comando inicial `modelerclient` o para alterar la invocación de IBM SPSS Modeler.

Hay varios tipos de argumentos de línea de comandos disponibles que se describen más adelante en esta sección.

*Tabla 23. Tipos de argumentos de línea de comandos.*

Tipo de argumento	Dónde se describe
Argumentos del sistema	Consulte el tema "Argumentos del sistema" en la página 52 para obtener más información.
Argumentos de parámetros	Consulte el tema "Argumentos de parámetros" en la página 53 para obtener más información.
Argumentos de conexión del servidor	Consulte el tema "Argumentos de conexión con el servidor" en la página 54 para obtener más información.
Argumentos de conexión de IBM SPSS Collaboration and Deployment Services Repository	Consulte el tema "IBM SPSS Collaboration and Deployment Services Repository Argumentos de conexión" en la página 55 para obtener más información.

Por ejemplo, se pueden utilizar las marcas `-server`, `-stream` y `-execute` para conectar con un servidor y, a continuación, cargar y ejecutar una ruta, de la siguiente forma:

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

Tenga en cuenta que al ejecutarse en una instalación cliente local, no se necesitan argumentos de conexión con el servidor.

Los valores de parámetros que contienen espacios se pueden poner entre comillas dobles, por ejemplo:

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

También puede ejecutar scripts de IBM SPSS Modeler de esta forma, utilizando el distintivo `-script`.

Depuración de argumentos de la línea de comandos

Para depurar una línea de comandos, utilice el comando `modelerclient` para iniciar IBM SPSS Modeler con los argumentos deseados. Esto permite comprobar que los comandos se ejecutarán como se espera. También puede confirmar los valores de cualquier parámetro pasado desde la línea de comandos en el cuadro de diálogo Parámetros de sesión (menú Herramientas, Definir parámetros de sesión).

## Argumentos del sistema

En la siguiente tabla se describen los argumentos del sistema disponibles para la invocación de la línea de comandos de la interfaz de usuario.

Tabla 24. Argumentos del sistema

Argumento	Comportamiento/Descripción
@ <archivo de comandos>	El carácter @ seguido de un nombre de archivo especifica una lista de comandos. Cuando <code>modelerclient</code> encuentra un argumento que comienza por @, opera en los comandos de este archivo como si hubieran estado en la línea de comandos. Consulte el tema “Combinación de varios argumentos” en la página 55 para obtener más información.
-directory <dir>	Define el directorio de trabajo predeterminado. En el modo local, este directorio se utiliza tanto para datos como para resultados. Ejemplo: <code>-directory c:/</code> o <code>-directory c:\\</code>
-server_directory <dir>	Define el directorio de servidor predeterminado para datos. El directorio de trabajo, especificado con la marca <code>-directory</code> , se utiliza para resultados.
-execute	Después del inicio, ejecuta cualquier ruta, estado o script que se haya cargado en el inicio. Si se carga un script además de una ruta o un estado, el script se ejecutará solo.
-stream <ruta>	Carga en el inicio la ruta especificada. Se pueden especificar varias rutas, pero la última se definirá como la actual.
-script <script>	Carga en el inicio el script autónomo especificado. Se puede especificar además de una ruta o un estado, tal y como se describe a continuación, pero sólo se puede cargar un único script en el inicio. Si el sufijo de archivo de script es <code>.py</code> , se presupone que el archivo es un script Python, de lo contrario, se presupone que es un script existente.
-model <modelo>	En el inicio, carga el modelo generado (archivo de formato <code>.gm</code> ) especificado.
-state <estado>	Carga en el inicio el estado especificado guardado.
-project <proyecto>	Carga el proyecto especificado. Sólo se puede cargar un único proyecto en el inicio.
-output <resultado>	Carga en el inicio el objeto de resultados guardado (archivo de formato <code>.cou</code> ).
-help	Muestra una lista de argumentos de la línea de comandos. Cuando se especifica esta opción, todos los demás argumentos se ignoran y se muestra la pantalla Ayuda.
-P <nombre>=<valor>	Se utiliza para definir un parámetro de inicio. También se puede utilizar para definir propiedades de nodos (parámetros de intervalo).
-scriptlang <python   legacy>	Puede utilizarse para especificar el lenguaje de script asociado con la opción <code>-script</code> , independientemente del sufijo de archivo de script.  Ejemplo <code>client -scriptlang python -script scores.txt -execute</code>  Esto ejecuta el archivo de script proporcionado utilizando Python, aunque el sufijo de archivo no era <code>.py</code> .

*Note:* los directorios predeterminados también se pueden definir en la interfaz de usuario. Para acceder a las opciones en el menú Archivo, seleccione **Definir directorio** o **Definir directorio de servidor**.

Carga de varios archivos

Desde la línea de comandos puede cargar varias rutas, estados y resultados en el inicio repitiendo el argumento relevante para cada objeto cargado. Por ejemplo, para cargar y ejecutar dos rutas llamadas *report.str* y *train.str*, utilizaría el siguiente comando:

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

### Carga de objetos desde IBM SPSS Collaboration and Deployment Services Repository

Dado que puede cargar determinados objetos de un archivo o desde IBM SPSS Collaboration and Deployment Services Repository (si dispone de licencia), el prefijo de nombre de archivo `spsscr:` y, si lo desea, `file:` (para objetos en disco) indica a IBM SPSS Modeler donde buscar el objeto. El prefijo funciona con las siguientes marcas:

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

Puede utilizar el prefijo para crear un URI que especifique la ubicación del objeto, por ejemplo, `-stream "spsscr:///folder_1/scoring_stream.str"`. La presencia del prefijo `spsscr:` requiere que se especifique una conexión válida a IBM SPSS Collaboration and Deployment Services Repository en el mismo comando. Así, por ejemplo, el comando completo sería:

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

Recuerde que en la línea de comandos *debe* utilizar un URI. El `REPOSITORY_PATH` más simple no se admite. (Sólo funciona en scripts.)

## Argumentos de parámetros

Los parámetros se pueden utilizar como marcas durante la ejecución de la línea de comandos de IBM SPSS Modeler. En los argumentos de la línea de comandos, la marca `-P` se utiliza para denotar un parámetro del tipo `-P <nombre>=<valor>`.

Los parámetros pueden ser:

- **Parámetros simples**
- **Parámetros de intervalo**, también denominados **propiedades de nodos**. Estos parámetros se utilizan para modificar la configuración de los nodos en la ruta. Consulte el tema “Conceptos básicos de las propiedades de nodos” en la página 58 para obtener más información.
- **Parámetros de la línea de comandos**, que son parámetros utilizados para alterar la invocación de IBM SPSS Modeler.

Por ejemplo, puede proporcionar contraseñas y nombres de usuario de orígenes de datos como una marca de la línea de comandos, por ejemplo:

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

El formato es el mismo que el del parámetro `datasource` de la propiedad de nodo `database`. Consulte el tema “Propiedades del nodo `database`” en la página 66 para obtener más información.

## Argumentos de conexión con el servidor

La marca `-server` indica a IBM SPSS Modeler que debe conectar con un servidor público, y las marcas `-hostname`, `-use_ssl`, `-port`, `-username`, `-password` y `-domain` se utilizan para indicar a IBM SPSS Modeler cómo conectar con el servidor público. Si no se especifica ningún argumento `-server`, se utilizará el servidor predeterminado o local.

### Ejemplos

Para conectarse con un servidor público:

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

Para conectarse con un clúster de servidores:

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

Tenga en cuenta que para conectarse a un clúster de servidores necesita Coordinator of Processes a través de IBM SPSS Collaboration and Deployment Services, de modo que debe utilizar el argumento `-cluster` junto con las opciones de conexión de repositorio (`spsscr_*`). Consulte el tema “IBM SPSS Collaboration and Deployment Services Repository Argumentos de conexión” en la página 55 para obtener más información.

Tabla 25. Argumentos de conexión del servidor.

Argumento	Comportamiento/Descripción
<code>-server</code>	Ejecuta IBM SPSS Modeler en el modo servidor, conectando con un servidor público utilizando las marcas <code>-hostname</code> , <code>-port</code> , <code>-username</code> , <code>-password</code> y <code>-domain</code> .
<code>-hostname &lt;nombre&gt;</code>	Nombre de host del equipo servidor. Disponible en el modo servidor solamente.
<code>-use_ssl</code>	Especifica que la conexión debería utilizar SSL (secure socket layer). La marca es opcional, el parámetro predeterminado <i>no</i> utiliza SSL.
<code>-port &lt;número&gt;</code>	Número de puerto del servidor especificado. Disponible en el modo servidor solamente.
<code>-cluster &lt;nombre&gt;</code>	Especifica una conexión a un clúster de servidores en lugar de un servidor especificado; este argumento es una alternativa a los argumentos <code>hostname</code> , <code>port</code> y <code>use_ssl</code> . El nombre es el del clúster o un URI exclusivo que identifica el clúster en IBM SPSS Collaboration and Deployment Services Repository. Coordinator of Processes gestiona el clúster de servidores a través de IBM SPSS Collaboration and Deployment Services. Consulte el tema “IBM SPSS Collaboration and Deployment Services Repository Argumentos de conexión” en la página 55 para obtener más información.
<code>-username &lt;nombre&gt;</code>	Nombre de usuario con el que iniciar sesión en el servidor. Disponible en el modo servidor solamente.
<code>-password &lt;contraseña&gt;</code>	Contraseña con la que iniciar sesión en el servidor. Disponible en el modo servidor solamente. <i>Note:</i> si no se utiliza el argumento <code>-password</code> , se le solicitará una contraseña.
<code>-epassword &lt;cadena de contraseña codificada&gt;</code>	Contraseña codificada con la que iniciar sesión en el servidor. Disponible en el modo servidor solamente. <i>Note:</i> Las contraseñas codificadas se pueden generar desde el menú Herramientas de la aplicación IBM SPSS Modeler.
<code>-domain &lt;nombre&gt;</code>	Dominio utilizado para iniciar sesión en el servidor. Disponible en el modo servidor solamente.
<code>-P &lt;nombre&gt;=&lt;valor&gt;</code>	Se utiliza para definir un parámetro de inicio. También se puede utilizar para definir propiedades de nodos (parámetros de intervalo).

# IBM SPSS Collaboration and Deployment Services Repository

## Argumentos de conexión

*Note:* Es necesario disponer de una licencia independiente para acceder a un repositorio IBM SPSS Collaboration and Deployment Services. Si desea obtener más información, consulte <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Si desea almacenar o recuperar objetos de IBM SPSS Collaboration and Deployment Services a través de la línea de comandos, debe especificar una conexión válida con IBM SPSS Collaboration and Deployment Services Repository. Por ejemplo:

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

La siguiente tabla enumera los argumentos que pueden utilizarse para configurar la conexión.

*Tabla 26. Argumentos de conexión de IBM SPSS Collaboration and Deployment Services Repository*

Argumento	Comportamiento/Descripción
-spsscr_hostname <nombre del host o dirección IP>	El nombre del host o la dirección IP del servidor en que se ha instalado IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_port <número>	Número de puerto en el que el IBM SPSS Collaboration and Deployment Services Repository acepta las conexiones (normalmente 8080 de forma predeterminada).
-spsscr_use_ssl	Especifica que la conexión debería utilizar SSL (secure socket layer). La marca es opcional, el parámetro predeterminado <i>no</i> utiliza SSL.
-spsscr_username <nombre>	Nombre de usuario con el que iniciar sesión en el IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_password <contraseña>	Contraseña con la que iniciar sesión en el IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_epassword <contraseña codificada>	Contraseña cifrada con la que iniciar sesión en el IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_domain <nombre>	Dominio utilizado para iniciar sesión en el IBM SPSS Collaboration and Deployment Services Repository. Esta marca es opcional, no la utilice a menos que inicie sesión mediante LDAP o Active Directory.

## Combinación de varios argumentos

Es posible combinar varios argumentos en un único archivo de comandos especificado en la invocación utilizando el símbolo @ seguido del nombre de archivo. De este modo podrá acortar la invocación de la línea de comandos y superar cualquier limitación del sistema operativo en la longitud del comando. Por ejemplo, el siguiente comando de inicio utiliza todos los argumentos especificados en el archivo de referencia <nombre de archivo de comandos>.

```
modelerclient @<commandFileName>
```

Ponga el nombre del archivo y la ruta del archivo de comandos entre comillas si hay que incluir espacios, de la siguiente forma:

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

El archivo de comandos puede contener todos los argumentos especificados previamente a nivel individual en el inicio. Por ejemplo:

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

Cuando escriba y referencie archivos de comandos, asegúrese de cumplir estas restricciones:

- Utilice sólo un comando por línea.
- No incruste un argumento @archivo de comandos en un archivo de comandos.

---

## Capítulo 7. Referencia de propiedades

---

### Conceptos básicos de referencia de propiedades

Puede especificar varias propiedades diferentes para los nodos, rutas, supernodos y proyectos. Algunas propiedades son comunes a todos los nodos, como el nombre, la anotación y la información sobre herramientas, mientras que otras son específicas para determinados tipos de nodos. Otras propiedades hacen referencia a operaciones de rutas de alto nivel, como el comportamiento del Supernodo o el almacenamiento en caché. Se puede acceder a las propiedades a través de la interfaz de usuario estándar (por ejemplo, al abrir un cuadro de diálogo para editar opciones para un nodo) y se pueden utilizar también de varias otras formas.

- Las propiedades se pueden modificar a través de los scripts, como se describe en esta sección. Si desea obtener más información, consulte “Definición de propiedades” en la página 28.
- Las propiedades de los nodos se pueden utilizar en los parámetros de Supernodo.
- Asimismo, las propiedades de los nodos se pueden utilizar como parte de una opción de línea de comandos (mediante la marca -P) al iniciar IBM SPSS Modeler.

En el contexto de los scripts de IBM SPSS Modeler, las propiedades de nodos y rutas se suelen llamar **parámetros de intervalo**. En este manual, se denominan propiedades de nodos y rutas.

Si desea obtener más información sobre el lenguaje de scripts, consulte Capítulo 2, “Language de scripts”, en la página 13.

### Abreviaturas

Las abreviaturas estándar se utilizan en la sintaxis para las propiedades de nodos. El aprendizaje de las abreviaturas le ayudará en la creación de scripts.

*Tabla 27. Abreviaturas estándar utilizadas en toda la sintaxis.*

Abreviatura	Significado
abs	Valor absoluto
lon	Longitud
min	Mínimo.
max	Máximo.
correl	Correlation
covar	Covariance
núm	Número o numérico
pct	Porcentaje
transp	Transparencia
xval	Validación cruzada
var	Varianza o variable (en nodos de origen)

### Ejemplos de propiedades de nodos y rutas

Las propiedades de nodos y rutas se pueden utilizar de varias formas con IBM SPSS Modeler. Normalmente se utilizan como parte de un script, bien un **script autónomo**, utilizado para automatizar rutas u operaciones o un **script de ruta**, utilizado para automatizar procesos en una sola ruta. Los parámetros de nodo se pueden especificar también utilizando las propiedades para los nodos del Supernodo. En el nivel más básico, las propiedades se pueden utilizar también como una opción de línea

de comandos para iniciar IBM SPSS Modeler. Si utiliza el argumento -p como parte de la invocación de la línea de comandos, podrá utilizar una propiedad de ruta para cambiar una configuración de la ruta.

Consulte los temas “Parámetros de ruta, sesión y Supernodo” en la página 40 y “Argumentos del sistema” en la página 52 para obtener más ejemplos de scripts.

## Conceptos básicos de las propiedades de nodos

Cada tipo de nodo tiene su propio conjunto de propiedades legales y cada propiedad tiene un tipo. Este tipo puede ser un tipo general, número, marca o cadena, en cuyo caso, las configuraciones de la propiedad se forzarán en el tipo correcto. Surgirá un error en caso de que no se puedan forzar. También se puede dar el caso de que la referencia de la propiedad pueda especificar el rango de valores legales como Discard, PairAndDiscard e IncludeAsText, en cuyo caso se producirá un error si se utiliza otro valor. Las propiedades de marcas se deben leer o definir mediante los valores True y False. En las tablas de referencia de este manual, las propiedades estructuradas se indican como tales en la columna *Descripción de la propiedad* y se proporcionan los formatos de uso.

## Propiedades de nodos comunes

Existen varias propiedades que son comunes a todos los nodos (incluidos los Supernodos) en IBM SPSS Modeler.

Tabla 28. Propiedades comunes de nodos.

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
use_custom_name	booleano	
name	cadena	Propiedad de sólo lectura que lee el nombre (automático o personalizado) para un nodo del lienzo.
custom_name	cadena	Especifica un nombre personalizado para el nodo.
tooltip	cadena	
annotation	cadena	
keywords	cadena	Intervalo estructurado que especifica una lista de palabras clave con el objeto
cache_enabled	booleano	
tipo_nodo	source_supernode process_supernode terminal_supernode todos los nombres de nodo tal y como se especifican para la creación de scripts	Propiedad de sólo lectura utilizada para hacer referencia a un nodo por tipo. Por ejemplo, en lugar de hacer referencia al nodo sólo por el nombre, como real_income, puede también especificar el tipo, como userinput o filter.

Las propiedades específicas del Supernodo se tratan aparte como con los demás nodos. Consulte el tema Capítulo 19, “Propiedades de Supernodos”, en la página 243 para obtener más información.



## Capítulo 8. Propiedades de ruta

Los scripts pueden controlar una serie de propiedades de la ruta.

El script puede acceder a la corriente actual utilizando la función `stream()` en el módulo `modeler.script`, por ejemplo:

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

Para hacer referencia a las propiedades de la ruta, debe utilizar una variable de ruta especial, indicada por el símbolo `^` antes de la ruta.

La propiedad `nodes` se utiliza para hacer referencia a los nodos de la ruta actual.

Las propiedades de la ruta se describen en la tabla siguiente.

Tabla 29. Propiedades de ruta.

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
<code>execute_method</code>	Normal Script	
<code>date_format</code>	"DDMAA" "MMDDYY" "AAMMDD" "YYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-AAAA" "DD-MES-YY" "DD-MES-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.AAAA" "MM.DD.YY" "MM.DD.YYYY" "DD.MES.YY" "DD.MES.YYYY" "DD/MM/YY" "DD/MM/AAAA" "MM/DD/YY" "MM/DD/YYYY" "DD/MES/YY" "DD/MES/YYYY" MON YYYY q Q YYYY ww WK YYYY	
<code>date_baseline</code>	número	
<code>date_2digit_baseline</code>	número	

Tabla 29. Propiedades de ruta (continuación).

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
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	booleano	
import_datetime_as_string	booleano	
decimal_places	número	
decimal_symbol	Predeterminado Period Comma	
angles_in_radians	booleano	
use_max_set_size	booleano	
max_set_size	número	
ruleset_evaluation	Voting FirstHit	
refresh_source_nodes	booleano	Se utiliza para actualizar los nodos de origen de forma automática al realizar la ejecución de la ruta.
script	cadena	
script_language	Python Legacy	Establece el lenguaje de script para el script de ruta.
annotation	cadena	
codificación	SystemDefault "UTF-8"	
stream_rewriting	booleano	
stream_rewriting_maximise_sql	booleano	
stream_rewriting_optimise_clem_ejecución	booleano	
stream_rewriting_optimise_syntax_ejecución	booleano	
enable_parallelism	booleano	
sql_generation	booleano	
database_caching	booleano	
sql_logging	booleano	
sql_generation_logging	booleano	
sql_log_native	booleano	
sql_log_prettyprint	booleano	

Tabla 29. Propiedades de ruta (continuación).

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
record_count_suppress_input	<i>booleano</i>	
record_count_feedback_interval	<i>entero</i>	
use_stream_auto_create_node_configuración	<i>booleano</i>	Si es true, se utilizan los valores específicos de la ruta, de lo contrario, se utilizan las preferencias de usuario.
create_model_applier_for_new_modelos	<i>booleano</i>	Si es true, cuando un constructor de modelos crea un modelo nuevo y no tiene enlaces de actualización activos, se añade un nuevo aplicador de modelos.
create_model_applier_update_links	createEnabled createDisabled doNotCreate	Define el tipo de enlace creado cuando se añade automáticamente un nodo aplicador de modelos.
create_source_node_from_builders	<i>booleano</i>	Si es true, cuando un constructor de modelos crea un resultado de origen nuevo y no tiene enlaces de actualización activos, se añade un nuevo nodo de origen.
create_source_node_update_links	createEnabled createDisabled doNotCreate	Define el tipo de enlace creado cuando se añade automáticamente un nodo de origen.



## Capítulo 9. Propiedades de nodos de origen

### Propiedades comunes de nodos de origen

Las propiedades comunes a todos los nodos de origen se enumeran a continuación, con información sobre nodos específicos en los temas siguientes.

Tabla 30. Propiedades comunes de nodos de origen.

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
dirección	Input Destino Both None Partition Split Frequency RecordID	Propiedad con clave para los roles de los campos. <i>Note:</i> Los valores In y Out ahora se han desaprobado. Deberán de ser compatibles en versiones posteriores.
type	Range Flag Set Sin tipo Discrete Conjunto ordenado Predeterminado	Tipo de campo. Si se establece esta propiedad como <i>Default</i> , se borrará cualquier configuración del parámetro values y si value_mode tiene el valor <i>Specify</i> , se restablecerá a <i>Read</i> . Si value_mode se establece en <i>Pass</i> o <i>Read</i> , la configuración de type no le afectará.
storage	Desconocido Cadena Entero Real Hora Fecha Marca de tiempo	Propiedad con clave de solamente lectura para el tipo de almacenamiento de campos.
check	Ninguno Nullify Coerce Discard Warn Abort	Propiedad con clave para la comprobación del rango y el tipo de campo.
values	[value value]	Para un campo continuo (rango), el primer valor es el mínimo y el último valor es el máximo. Para campos nominales (conjunto), especifique todos los valores. Para los campos marca, el primer valor representa <i>falso</i> y el último, <i>verdadero</i> . La configuración de esta propiedad establece de forma automática la propiedad value_mode en <i>Specify</i> .
value_mode	Leer Pasar Leer+ Actual Especifica	Determina la forma en que se han establecido los valores para un campo en la siguiente lectura de datos. Tenga en cuenta que no puede establecer esta propiedad directamente en <i>Specify</i> . Para utilizar valores específicos, establezca la propiedad values.

Tabla 30. Propiedades comunes de nodos de origen (continuación).

Nombre de la propiedad	Tipo de datos	Descripción de la propiedad
default_value_mode	Leer Pasar	Especifica el método predeterminado para configurar los valores de todos los campos. Esta configuración puede anularse para determinados campos mediante la propiedad value_mode.
extend_values	booleano	Se aplica cuando value_mode se establece en Read. Establézcala en T para añadir nuevos valores de lectura a los valores existentes del campo. Establézcala en F para descartar los valores existentes y favorecer a los nuevos valores de lectura.
value_labels	cadena	Se utiliza para especificar una etiqueta de valor. Tenga en cuenta que estos valores se deben especificar primero.
enable_missing	booleano	Cuando está definida como T, activa el seguimiento de los valores perdidos para el campo.
missing_values	[value value ...]	Especifica los valores de datos que denotan los datos perdidos.
range_missing	booleano	Cuando esta propiedad se establece como T, especifica si se define un rango de valores perdidos (en blanco) para un campo.
missing_lower	cadena	Si range_missing es verdadero, especifica el límite inferior del rango de valores perdidos.
missing_upper	cadena	Si range_missing es verdadero, especifica el límite superior del rango de valores perdidos.
null_missing	booleano	Cuando esta propiedad se establece en T, los valores nulos (valores no definidos que se muestran como \$null\$ en el software) se consideran valores perdidos.
whitespace_missing	booleano	Cuando esta propiedad está definida como T, los valores que solamente contienen un espacio en blanco (espacios, tabulaciones y líneas nuevas) se consideran valores perdidos.
description	cadena	Se utiliza para especificar la descripción o etiqueta de un campo.
default_include	booleano	Propiedad con clave para especificar si el comportamiento predeterminado es para pasar o filtrar los campos:
include	booleano	Propiedad con clave que se utiliza para determinar si los campos individuales se han incluido o se han filtrado:
new_name	cadena	

## Propiedades del nodo asimport

El origen de Analytic Server permite ejecutar una ruta en un sistema de archivos distribuido de Hadoop (HDFS en sus siglas inglesas).

Tabla 31. propiedades del nodo asimport.

Propiedades del nodo asimport	Tipo de datos	Descripción de la propiedad
data_source	cadena	Nombre del origen de datos.
host	cadena	El nombre del host de Analytic Server.

Tabla 31. propiedades del nodo asimport (continuación).

Propiedades del nodo asimport	Tipo de datos	Descripción de la propiedad
port	entero	El puerto por el que escucha Analytic Server.
tenant	cadena	En un entorno de varios consumidores, el nombre del consumidor al que se pertenece. En un entorno de un único consumidor, toma el valor predeterminado de <b>ibm</b> .
set_credentials	booleano	Si la autenticación de usuario en Analytic Server es la misma que en el servidor SPSS Modeler, asigne a esta propiedad el valor <b>false</b> . En caso contrario, asígnele <b>true</b> .
user_name	cadena	El nombre de usuario de inicio de sesión de Analytic Server. Solo es necesario si set_credentials es true.
password	cadena	La contraseña de inicio de sesión de Analytic Server. Solo es necesario si set_credentials es true.

## Propiedades del nodo cognosimport



El nodo de origen de IBM Cognos BI importa datos desde las bases de datos de Cognos BI.

Tabla 32. Propiedades del nodo cognosimport.

Propiedades del nodo cognosimport	Tipo de datos	Descripción de la propiedad
mode	Datos Report	Especifica si se importarán los datos de Cognos BI (valor predeterminado) o informes.
cognos_connection	{ "cadena", booleano, "cadena", "cadena", "cadena" }	Una propiedad de la lista que contiene los detalles de conexión para el servidor de Cognos. El formato es: {"Cognos_server_URL", login_mode, "namespace", "username", "password"} donde: Cognos_server_URL es la URL del servidor de Cognos que contiene el origen login_mode indica si se utiliza el inicio de sesión anónimo, y es true o false; si se establece como true, los campos siguientes deben establecerse en " namespace especifica el proveedor de autenticación de seguridad utilizado para registrarse en el servidor username y password son los utilizados para registrarse en el servidor de Cognos

Tabla 32. Propiedades del nodo cognosimport (continuación).

Propiedades del nodo cognosimport	Tipo de datos	Descripción de la propiedad
cognos_package_name	<i>cadena</i>	La ruta y el nombre del paquete de Cognos desde el que está importando objetos de datos, por ejemplo: /Public Folders/GOSALES Nota: solo es válida la barra inclinada.
cognos_items	{ <i>"campo", "campo", ...</i> , " <i>campo</i> "}	El nombre de uno o más objetos de datos que van a importarse. El formato de <i>campo</i> es [ <i>espacio de nombre</i> ].[ <i>asunto de consulta</i> ].[ <i>elemento de consulta</i> ]
cognos_filters	<i>field</i>	El nombre de uno o más filtros que van a aplicarse antes de importar datos.
cognos_data_parameters	<i>lista</i>	Valores de parámetros de solicitud de datos. Los pares de nombre y valor se encierran entre llaves y si hay varios pares se separan con comas y toda la cadena se encierra entre corchetes. Formato: [{" <i>param1</i> ", " <i>value</i> "},...,{" <i>paramN</i> ", " <i>value</i> "}]
cognos_report_directory	<i>field</i>	La ruta de Cognos de una carpeta o paquete de la que importar informes, por ejemplo: /Public Folders/GOSALES Nota: solo es válida la barra inclinada.
cognos_report_name	<i>field</i>	La ruta y nombre dentro de la ubicación del informe de un informe que se ha de importar.
cognos_report_parameters	<i>lista</i>	Valores de parámetros de informe. Los pares nombre-valor están entre llaves, los pares múltiples están separados por comas y toda la cadena está entre corchetes. Formato: [{" <i>param1</i> ", " <i>value</i> "},...,{" <i>paramN</i> ", " <i>value</i> "}]

## Propiedades del nodo database



El nodo Base de datos se puede utilizar para importar datos desde otros paquetes mediante ODBC (del inglés, Open Database Connectivity), incluidos Microsoft SQL Server, DB2, Oracle, etc.

Tabla 33. Propiedades del nodo database.

Propiedades del nodo database	Tipo de datos	Descripción de la propiedad
mode	Tabla Query	Especifique <i>Table</i> para conectarse a una tabla de base de datos mediante los controles del cuadro de diálogo, o <i>Query</i> para realizar una consulta a la base de datos seleccionada mediante SQL.
datasource	<i>cadena</i>	Nombre de la base de datos (consulte la siguiente nota).
username	<i>cadena</i>	Detalles de conexión de la base de datos (consulte la siguiente nota).
password	<i>cadena</i>	



Tabla 33. Propiedades del nodo database (continuación).

Propiedades del nodo database	Tipo de datos	Descripción de la propiedad
epassword	cadena	Especifica una contraseña codificada como una alternativa a codificar una contraseña en un script.  Consulte el tema "Generación de una contraseña codificada" en la página 47 para obtener más información. Esta propiedad es de sólo lectura durante la ejecución.
tablename	cadena	Nombre de la tabla a la que se desea tener acceso.
strip_spaces	Ninguno Left Right Both	Opciones para descartar los espacios iniciales y finales en las cadenas.
use_quotes	AsNeeded Always Nunca	Especifica si los nombres de columna y tabla aparecen entre comillas cuando las consultas se envían a la base de datos (en el caso, por ejemplo, de que contengan espacios o signos de puntuación).
query	cadena	Especifica el código SQL para la consulta que desea enviar.

*Note:* Si el nombre de la base de datos (en la propiedad datasource) contiene espacios, entonces en vez de las propiedades individuales para datasource, username y password, utilice un único origen de datos en el siguiente formato:

Tabla 34. Propiedades del nodo database - específicas de datasource.

Propiedades del nodo database	Tipo de datos	Descripción de la propiedad
datasource	cadena	Formato: [database_name,username,password[,true   false]]  El último parámetro se usa con contraseñas cifradas. Si se define como true, la contraseña se cifrará antes de usarse.

Utilice este formato también si está cambiando el origen de datos, sin embargo, si tan sólo desea cambiar el nombre de usuario o contraseña, puede usar las propiedades username o password.

## Propiedades del nodo datacollectionimport



El nodo de importación de datos IBM SPSS Data Collection importa datos de encuesta basados en el modelo de datos de IBM SPSS Data Collection que utilizan los productos de investigación de mercados de IBM Corp.. Se debe instalar la biblioteca de datos de IBM SPSS Data Collection para utilizar este nodo.

Figura 7. Nodo de importación de datos Dimensiones

Tabla 35. Propiedades del nodo *datacollectionimport*.

Propiedades del nodo <i>datacollectionimport</i>	Tipo de datos	Descripción de la propiedad
<i>metadata_name</i>	<i>cadena</i>	El nombre del MDSC. El valor especial <i>DimensionsMDD</i> indica que se debería utilizar el documento de metadatos de IBM SPSS Data Collection estándar. Otro posibles valores podrían ser: <i>mrADODsc</i> <i>mrI2dDsc</i> <i>mrLogDsc</i> <i>mrQdiDrsDsc</i> <i>mrQvDsc</i> <i>mrSampleReportingMDSC</i> <i>mrSavDsc</i> <i>mrSCDsc</i> <i>mrScriptMDSC</i>  El valor especial <i>none</i> indica que no existe ningún MDSC.
<i>metadata_file</i>	<i>cadena</i>	Nombre del archivo en el que se almacenan los metadatos.
<i>casedata_name</i>	<i>cadena</i>	El nombre del CDSC. Entre los posibles valores se encuentran: <i>mrADODsc</i> <i>mrI2dDsc</i> <i>mrLogDsc</i> <i>mrPunchDSC</i> <i>mrQdiDrsDsc</i> <i>mrQvDsc</i> <i>mrRdbDsc2</i> <i>mrSavDsc</i> <i>mrScDSC</i> <i>mrXmIDsc</i>  El valor especial <i>none</i> indica que no existe ningún CDSC.
<i>casedata_source_type</i>	Desconocido File Folder UDL DSN	Indica el tipo de origen del CDSC.
<i>casedata_file</i>	<i>cadena</i>	Cuando <i>casedata_source_type</i> es <i>File</i> , especifica el archivo que contiene los datos de casos.
<i>casedata_folder</i>	<i>cadena</i>	Cuando <i>casedata_source_type</i> es <i>Folder</i> , especifica la carpeta que contiene los datos de casos.
<i>casedata_udl_string</i>	<i>cadena</i>	Cuando <i>casedata_source_type</i> es <i>UDL</i> , especifica la cadena de conexión OLD-DB del origen de datos que contiene los datos de casos.
<i>casedata_dsn_string</i>	<i>cadena</i>	Cuando <i>casedata_source_type</i> es <i>DSN</i> , especifica la cadena de conexión ODBC del origen de datos.

Tabla 35. Propiedades del nodo *datacollectionimport* (continuación).

Propiedades del nodo <i>datacollectionimport</i>	Tipo de datos	Descripción de la propiedad
<i>casedata_project</i>	<i>cadena</i>	Al leer datos de casos de una base de datos de IBM SPSS Data Collection, puede escribir el nombre del proyecto. Para el resto de tipos de datos de casos, esta configuración se deberá dejar en blanco.
<i>version_import_mode</i>	All Latest Específica	Define el modo en que deben tratarse las versiones.
<i>specific_version</i>	<i>cadena</i>	Cuando <i>version_import_mode</i> es <i>Specify</i> , define la versión de los datos de casos que se van a importar.
<i>use_language</i>	<i>cadena</i>	Determina si deben usarse las etiquetas de un idioma concreto.
<i>language</i>	<i>cadena</i>	Si <i>use_language</i> es verdadero, define el código de idioma que se va a usar en la importación. Este código de idioma debe incluirse entre aquellos disponibles en los datos de casos.
<i>use_context</i>	<i>cadena</i>	Determina si se debe importar un contexto específico. Los contextos se utilizan para modificar la descripción asociada con las respuestas.
<i>context</i>	<i>cadena</i>	Cuando <i>use_context</i> es verdadero, define el contexto de la importación. Este contexto debe encontrarse entre aquellos disponibles en los datos de casos.
<i>use_label_type</i>	<i>cadena</i>	Determina si se debe importar un tipo de etiqueta específico.
<i>label_type</i>	<i>cadena</i>	Cuando <i>use_label_type</i> es verdadero, define el tipo de etiqueta de la importación. Este tipo de etiqueta debe encontrarse entre aquellos disponibles en los datos de casos.
<i>user_id</i>	<i>cadena</i>	En el caso de las bases de datos que requieren un inicio de sesión explícito, puede proporcionar un ID de usuario y una contraseña para acceder al origen de datos.
<i>password</i>	<i>cadena</i>	
<i>import_system_variables</i>	Común None All	Especifica las variables del sistema que se importan.
<i>import_codes_variables</i>	<i>booleano</i>	
<i>import_sourcefile_variables</i>	<i>booleano</i>	
<i>import_multi_response</i>	MultipleFlags Single	

## Propiedades del nodo excelimport



El nodo de importación Excel importa datos de cualquier versión de Microsoft Excel. No es necesario un origen de datos ODBC.

Tabla 36. Propiedades del nodo excelimport.

Propiedades del nodo excelimport	Tipo de datos	Descripción de la propiedad
excel_file_type	Excel2003 Excel2007	
full_filename	<i>cadena</i>	El nombre completo del archivo, incluyendo la ruta.
use_named_range	<i>Booleana</i>	Si usar un rango con nombre. Si es verdadero, la propiedad named_range se utiliza para especificar el rango de lectura y se ignoran el resto de configuraciones de rango de datos y hojas de trabajo.
named_range	<i>cadena</i>	
worksheet_mode	Índice Name	Determina si la hoja de trabajo se define por el índice o por el nombre.
worksheet_index	<i>entero</i>	Índice de la hoja de trabajo que se va a leer, siendo 0 la primera hoja de trabajo, 1 la segunda, etc.
worksheet_name	<i>cadena</i>	Nombre de la hoja de trabajo que se va a leer.
data_range_mode	FirstNonBlank ExplicitRange	Especifica cómo debe establecerse el rango.
blank_rows	StopReading ReturnBlankRows	Cuando data_range_mode es <i>FirstNonBlank</i> , especifica cómo deben tratarse las filas en blanco.
explicit_range_start	<i>cadena</i>	Cuando data_range_mode es <i>ExplicitRange</i> , especifica el punto de partida del rango de lectura.
explicit_range_end	<i>cadena</i>	
read_field_names	<i>Booleana</i>	Determina si la primera fila del rango concreto debería usarse como nombres de campo (columna).

## Propiedades del nodo evimport



El nodo Enterprise View crea una conexión a IBM SPSS Collaboration and Deployment Services Repository, lo que permite leer los datos de Enterprise View en una ruta y empaquetar un modelo en un escenario al que otros usuarios pueden acceder desde el repositorio.

Tabla 37. Propiedades del nodo evimport.

Propiedades del nodo evimport	Tipo de datos	Descripción de la propiedad
connection	<i>lista</i>	Propiedad estructura: lista de parámetros que componen una conexión de Enterprise View.

Tabla 37. Propiedades del nodo *evimport* (continuación).

Propiedades del nodo <i>evimport</i>	Tipo de datos	Descripción de la propiedad
tablename	<i>cadena</i>	Nombre de una tabla de la vista de aplicación.

## Propiedades del nodo *fixedfile*



El nodo Archivo fijo importa datos desde archivos de texto de campo fijo; esto es, archivos cuyos campos no están delimitados pero empiezan en la misma posición y tienen una longitud fija. Los datos heredados o generados por la máquina se suelen almacenar en formato de campo fijo.

Tabla 38. Propiedades del nodo *fixedfile*.

Propiedades del nodo <i>fixedfile</i>	Tipo de datos	Descripción de la propiedad
record_len	<i>número</i>	Especifica el número de caracteres de cada registro.
line_oriented	<i>booleano</i>	Omite el carácter de nueva línea al final de cada registro.
decimal_symbol	Predeterminado Comma Period	Tipo de separador decimal utilizado en el origen de datos.
skip_header	<i>número</i>	Especifica el número de líneas que se ignorarán al principio del primer registro. Esto resulta útil para ignorar las cabeceras de columna.
auto_recognize_datetime	<i>booleano</i>	Especifica si las fechas o las horas se identifican automáticamente en los datos de origen.
lines_to_scan	<i>número</i>	
fields	<i>lista</i>	Propiedad estructurada.
full_filename	<i>cadena</i>	Nombre completo del archivo que se va a leer, incluido el directorio.
strip_spaces	Ninguno Left Right Both	Descarta los espacios iniciales y finales en las cadenas de importación.
invalid_char_mode	Discard Replace	Elimina los caracteres no válidos (nulo, 0 o cualquier carácter que no exista en la codificación actual) de la entrada de datos o sustituye los caracteres no válidos con el símbolo especificado de un carácter.
invalid_char_replacement	<i>cadena</i>	
use_custom_values	<i>booleano</i>	
custom_storage	Desconocido Cadena Entero Real Hora Fecha Marca de tiempo	

Tabla 38. Propiedades del nodo fixedfile (continuación).

Propiedades del nodo fixedfile	Tipo de datos	Descripción de la propiedad
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MES-YY" "DD-MES-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MES.YY" "DD.MES.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MES/YY" "DD/MES/YYYY" MON YYYY q Q YYYY ww WK YYYY	Aplicable solamente si ha especificado un almacenamiento personalizado.
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"	Aplicable solamente si ha especificado un almacenamiento personalizado.
custom_decimal_symbol	<i>field</i>	Aplicable solamente si ha especificado un almacenamiento personalizado.
codificación	StreamDefault SystemDefault "UTF-8"	Especifica el método de codificación de textos.

## Propiedades del nodo sasimport



El nodo importar SAS importa datos SAS a IBM SPSS Modeler.

Tabla 39. Propiedades de nodo sas.

Propiedades del nodo sasimport	Tipo de datos	Descripción de la propiedad
format	Windows UNIX Transport SAS7 SAS8 SAS9	Formato del archivo que se va a importar.
full_filename	<i>cadena</i>	El nombre completo del archivo introducido, incluyendo su ruta.
member_name	<i>cadena</i>	Especifique el miembro para importar desde el archivo de transporte SAS especificado.
read_formats	<i>booleano</i>	Lee formatos de datos (como etiquetas de variables) del archivo de formatos especificado.
full_format_filename	<i>cadena</i>	
import_names	NamesAndLabels LabelsasNames	Especifica el método para la correlación de nombres y etiquetas de variables en la importación.

## Propiedades del nodo simgen



El nodo de generación de simulación proporciona una manera fácil de generar datos, ya sea desde cero utilizando las distribuciones o estadísticas especificada por el usuario o automáticamente utilizando las distribuciones obtenidas de la ejecución de un nodo de simulación de ajuste sobre datos históricos. Esto es útil cuando desea evaluar el resultado de un modelo predictivo en caso de dudas sobre las entradas del modelo.

Tabla 40. Propiedades del nodo simgen.

Propiedades del nodo simgen	Tipo de datos	Descripción de la propiedad
campos	Propiedad estructurada	
correlaciones	Propiedad estructurada	
max_cases	<i>entero</i>	El valor mínimo es 1000, el valor máximo es 2,147,483,647.
create_iteration_field	<i>booleano</i>	
iteration_field_name	<i>cadena</i>	
replicate_results	<i>booleano</i>	
random_seed	<i>entero</i>	
overwrite_when_refitting	<i>booleano</i>	
parameter_xml	<i>cadena</i>	Devuelve el XML del parámetro como una cadena.

Tabla 40. Propiedades del nodo simgen (continuación).

Propiedades del nodo simgen	Tipo de datos	Descripción de la propiedad
distribution	Bernoulli Beta Binomial Categorical Exponential Fixed Gamma Lognormal NegativeBinomialFailures NegativeBinomialTrials Normal Poisson Range Triangular Uniform Weibull	
bernoulli_prob	número	$0 \leq \text{bernoulli\_prob} \leq 1$
beta_shape1	número	Debe ser $\geq 0$
beta_shape2	número	Debe ser $\geq 0$
beta_min	número	Opcional. Debe ser menor que beta_max.
beta_max	número	Opcional. Debe ser mayor que beta_min.
binomial_n	entero	Debe ser $> 0$
binomial_prob	número	$0 \leq \text{binomial\_prob} \leq 1$
binomial_min	número	Opcional. Debe ser menor que binomial_max.
binomial_max	número	Opcional. Debe ser mayor que binomial_min.
exponential_scale	número	Debe ser $> 0$
exponential_min	número	Opcional. Debe ser menor que exponential_max.
exponential_max	número	Opcional. Debe ser mayor que exponential_min.
fixed_value	cadena	
gamma_shape	número	Debe ser $\geq 0$
gamma_scale	número	Debe ser $\geq 0$
gamma_min	número	Opcional. Debe ser menor que gamma_max.
gamma_max	número	Opcional. Debe ser mayor que gamma_min.
lognormal_shape1	número	Debe ser $\geq 0$
lognormal_shape2	número	Debe ser $\geq 0$
lognormal_min	número	Opcional. Debe ser menor que lognormal_max.
lognormal_max	número	Opcional. Debe ser mayor que lognormal_min.
negative_bin_failures_threshold	número	Debe ser $\geq 0$



Tabla 40. Propiedades del nodo simgen (continuación).

Propiedades del nodo simgen	Tipo de datos	Descripción de la propiedad
negative_bin_failures_prob	número	$0 \leq \text{negative\_bin\_failures\_prob} \leq 1$
negative_bin_failures_min	número	Opcional. Debe ser menor que negative_bin_failures_max.
negative_bin_failures_max	número	Opcional. Debe ser mayor que negative_bin_failures_min.
negative_bin_trials_threshold	número	Debe ser $\geq 0$
negative_bin_trials_prob	número	$0 \leq \text{negative\_bin\_trials\_prob} \leq 1$
negative_bin_trials_min	número	Opcional. Debe ser menor que negative_bin_trials_max.
negative_bin_trials_max	número	Opcional. Debe ser menor que negative_bin_trials_min.
normal_mean	número	
normal_sd	número	Debe ser $> 0$
normal_min	número	Opcional. Debe ser menor que normal_max.
normal_max	número	Opcional. Debe ser mayor que normal_min.
poisson_mean	número	Debe ser $\geq 0$
poisson_min	número	Opcional. Debe ser menor que poisson_max.
poisson_max	número	Opcional. Debe ser mayor que poisson_min.
triangular_mode	número	$\text{triangular\_min} \leq \text{triangular\_mode} \leq \text{triangular\_max}$
triangular_min	número	Debe ser menor que triangular_mode.
triangular_max	número	Debe ser mayor que triangular_mode.
uniform_min	número	Debe ser menor que uniform_max.
uniform_max	número	Debe ser mayor que uniform_min.
weibull_rate	número	Debe ser $\geq 0$
weibull_scale	número	Debe ser $\geq 0$
weibull_location	número	Debe ser $\geq 0$
weibull_min	número	Opcional. Debe ser menor que weibull_max.
weibull_max	número	Opcional. Debe ser mayor que weibull_min.

La correlación puede ser cualquier número entre +1 y -1. Puede especificar tantas correlaciones como desee. Las correlaciones no especificadas se establecen en cero. Si alguno de los campos se desconocen, el valor de la correlación debe establecerse en la matriz de correlación (o tabla) y se muestra en texto rojo. Cuando hay campos desconocidos, no es posible ejecutar el nodo.

## Propiedades del nodo statisticsimport



El nodo IBM SPSS StatisticsArchivo lee los datos desde un formato de archivo *.sav* que utiliza IBM SPSS Statistics y archivos caché guardados en IBM SPSS Modeler, que también puede utilizar el mismo formato.

Las propiedades de este nodo están descritas en “Propiedades del nodo statisticsimport” en la página 239.

## Propiedades del nodo userinput



El nodo Datos de usuario proporciona una manera fácil de crear datos sintéticos, ya sea partiendo de cero o modificando los datos existentes. Esto resulta útil, por ejemplo, cuando desee crear un conjunto de datos de comprobación para el modelado.

Tabla 41. Propiedades del nodo userinput.

Propiedades del nodo userinput	Tipo de datos	Descripción de la propiedad
data		Los datos de cada campo pueden tener longitudes diferentes pero deben ser coherentes con el almacenamiento del campo. La configuración de los valores para un campo que no está presente, supone la creación de dicho campo. También se pueden establecer los valores para un campo en una cadena vacía (" ") para eliminar el campo especificado. <b>Nota:</b> Los valores especificados para esta propiedad deben ser cadenas, no números. Por ejemplo, los números 1, 2, 3 y 4 deben especificarse como "1 2 3 4".
names		Intervalo estructurado que establece o devuelve una lista de nombres de campos generados por el nodo.
custom_storage	Desconocido Cadena Entero Real Hora Fecha Marca de tiempo	Intervalo con clave que establece o devuelve el almacenamiento para un campo.
data_mode	Combined Ordered	Si se especifica Combined, los registros se generarán para cada combinación de valores del conjunto y valores mínimos y máximos. El número de registros generados será igual al producto del número de valores de cada campo. Si se especifica Ordered, se tomará un valor de cada columna para cada registro con el fin de generar una fila de datos. El número de registros generados será igual al número más grande de valores asociados a un campo. Los campos que tengan menos valores de datos se rellenarán con valores nulos.

Tabla 41. Propiedades del nodo `userinput` (continuación).

Propiedades del nodo <code>userinput</code>	Tipo de datos	Descripción de la propiedad
<code>values</code>		Esta propiedad está en desuso a favor de <code>data</code> y no debe volver a utilizarse.

## Propiedades del nodo `variablefile`



El nodo Archivo variable lee datos desde los archivos de texto de campo libre, esto es, campos cuyos registros contienen un número constante de campos pero un número variado de caracteres. Este nodo resulta también útil para los archivos con texto de cabecera de longitud fija y determinados tipos de anotaciones.

Tabla 42. Propiedades del nodo `variablefile`.

Propiedades del nodo <code>variablefile</code>	Tipo de datos	Descripción de la propiedad
<code>skip_header</code>	<i>número</i>	Especifica el número de caracteres que se ignorarán al principio del primer registro.
<code>num_fields_auto</code>	<i>booleano</i>	Determina el número de campos de cada registro de forma automática. Los registros deben terminar con un carácter de nueva línea.
<code>num_fields</code>	<i>número</i>	Especifica manualmente el número de campos de cada registro.
<code>delimit_space</code>	<i>booleano</i>	Especifica el carácter utilizado para delimitar los límites de los campos del archivo.
<code>delimit_tab</code>	<i>booleano</i>	
<code>delimit_new_line</code>	<i>booleano</i>	
<code>delimit_non_printing</code>	<i>booleano</i>	
<code>delimit_comma</code>	<i>booleano</i>	En aquellos casos en los que la coma sea el delimitador del campo y el separador decimal para rutas, establezca <code>delimit_other</code> en <i>True</i> y especifique una coma como delimitador mediante la propiedad <code>other</code> .
<code>delimit_other</code>	<i>booleano</i>	Permite especificar un delimitador personalizado mediante la propiedad <code>other</code> .
<code>other</code>	<i>cadena</i>	Especifica el delimitador utilizado cuando <code>delimit_other</code> es <i>True</i> .
<code>decimal_symbol</code>	Predeterminado Comma Period	Especifica el separador decimal utilizado en el origen de datos.
<code>multi_blank</code>	<i>booleano</i>	Trata varios caracteres delimitadores vacíos adyacentes como un único delimitador.
<code>read_field_names</code>	<i>booleano</i>	Trata la primera fila del archivo de datos como etiquetas para la columna.
<code>strip_spaces</code>	Ninguno Left Right Both	Descarta los espacios iniciales y finales en las cadenas de importación.

Tabla 42. Propiedades del nodo variablefile (continuación).

Propiedades del nodo variablefile	Tipo de datos	Descripción de la propiedad
invalid_char_mode	Discard Replace	Elimina los caracteres no válidos (nulo, 0 o cualquier carácter que no exista en la codificación actual) de la entrada de datos o sustituye los caracteres no válidos con el símbolo especificado de un carácter.
invalid_char_replacement	<i>cadena</i>	
break_case_by_newline	<i>booleano</i>	Especifica que el delimitador de línea es el carácter de nueva línea.
lines_to_scan	<i>número</i>	Especifica cuántas líneas se van a explorar para los tipos de datos especificados.
auto_recognize_datetime	<i>booleano</i>	Especifica si las fechas o las horas se identifican automáticamente en los datos de origen.
quotes_1	Discard PairAndDiscard IncludeAsText	Especifica cómo se tratarán las comillas simples en la importación.
quotes_2	Discard PairAndDiscard IncludeAsText	Especifica cómo se tratarán las comillas dobles en la importación.
full_filename	<i>cadena</i>	Nombre completo del archivo que se va a leer, incluido el directorio.
use_custom_values	<i>booleano</i>	
custom_storage	Desconocido Cadena Entero Real Hora Fecha Marca de tiempo	

Tabla 42. Propiedades del nodo `variablefile` (continuación).

Propiedades del nodo <code>variablefile</code>	Tipo de datos	Descripción de la propiedad
<code>custom_date_format</code>	"DDMMYY" "MMDDYY" "YYMMDD" "YYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MES-YY" "DD-MES-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MES.YY" "DD.MES.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MES/YY" "DD/MES/YYYY" MON YYYY q Q YYYY ww WK YYYY	Aplicable solamente si ha especificado un almacenamiento personalizado.
<code>custom_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"	Aplicable solamente si ha especificado un almacenamiento personalizado.
<code>custom_decimal_symbol</code>	<i>field</i>	Aplicable solamente si ha especificado un almacenamiento personalizado.
codificación	StreamDefault SystemDefault "UTF-8"	Especifica el método de codificación de textos.

## Propiedades del nodo xmlimport



El nodo de origen XML importa datos en formato XML en la ruta. Puede importar un único archivo o todos los archivos en un directorio. Puede especificar opcionalmente un archivo de esquema para leer la estructura XML.

Tabla 43. Propiedades de nodo xmlimport.

Propiedades del nodo xmlimport	Tipo de datos	Descripción de la propiedad
read	single directory	Lee un único archivo de datos (valor predeterminado) o todos los archivos XML de un directorio.
recurse	booleano	Especifica si leer además archivos XML de todos los subdirectorios del directorio especificado.
full_filename	cadena	(obligatorio) Ruta completa y nombre de archivo del archivo XML a importar (si read = single).
directory_name	cadena	(obligatorio) Ruta completa y nombre del directorio desde el que importar los archivos XML (si read = directory).
full_schema_filename	cadena	Ruta completa y nombre de archivo del archivo XSD o DTD desde el que leer la estructura XML. Si omite este parámetro, se leerá la estructura desde el archivo de origen XML.
records	cadena	Expresión XPath (p.ej. /author/name) para definir el límite del registro. Cada vez que este elemento se encuentra en el archivo de origen se crea un nuevo registro.
mode	read specify	Lee todos los datos (valor predeterminado) o especifica qué elementos leer.
fields		Lista de elementos (elementos y atributos) para importar. Cada elemento de la lista es una expresión XPath.

---

## Capítulo 10. Propiedades de nodos de operaciones con registros

---

### Propiedades del nodo append



El nodo Añadir concatena conjuntos de registros. Es útil para combinar conjuntos de datos con estructuras parecidas, pero con datos diferentes.

Tabla 44. Propiedades del nodo append.

Propiedades del nodo append	Tipo de datos	Descripción de la propiedad
match_by	Position Name	Se pueden añadir conjuntos de datos basándose en la posición que tienen los campos en el origen de datos principal o el nombre de los campos en los conjuntos de datos de entrada.
match_case	booleano	Activa la coincidencia de mayúsculas y minúsculas al hacer coincidir nombres de campos.
include_fields_from	Main All	
create_tag_field	booleano	
tag_field_name	cadena	

---

### Propiedades del nodo aggregate



El nodo Agregar reemplaza una secuencia de registros de entrada con registros de salida agregados y resumidos.

Tabla 45. Propiedades del nodo aggregate.

Propiedades del nodo aggregate	Tipo de datos	Descripción de la propiedad
keys	<i>[field field ... field]</i>	Enumera los campos que se pueden usar como claves en la agregación. Por ejemplo, si Sexo y Región son los campos clave, cada combinación exclusiva de V y M con las regiones N y S (cuatro combinaciones exclusivas) tendrá un registro agregado.
contiguous	booleano	Seleccione esta opción si sabe que todos los registros con los mismos valores clave se agrupan en la entrada (por ejemplo, si la entrada se clasifica en los campos clave). Con ello puede mejorar el rendimiento.
aggregates		Enumera los campos numéricos cuyos valores se añadirán, así como los modos de agregación elegidos.

Tabla 45. Propiedades del nodo aggregate (continuación).

Propiedades del nodo aggregate	Tipo de datos	Descripción de la propiedad
extension	cadena	Especifica un prefijo o sufijo para campos agregados duplicados (consulte un ejemplo a continuación).
add_as	Suffix Prefix	
inc_record_count	booleano	Crea un campo adicional que indica la cantidad de registros de entrada agregados para conformar cada registro agregado.
count_field	cadena	Especifica el nombre del campo de recuento de registros.

## Propiedades del nodo balance



El nodo Equilibrar corrige los desequilibrios de un conjunto de datos para que cumpla una condición determinada. La directiva de equilibrado ajusta la proporción de registros si una condición es verdadera por el factor determinado.

Tabla 46. Propiedades del nodo balance.

Propiedades del nodo balance	Tipo de datos	Descripción de la propiedad
directives		Propiedad estructurada para equilibrar la proporción de los valores de campos basados en un número determinado (consulte el ejemplo a continuación).
training_data_only	booleano	Especifica que sólo se deben equilibrar los datos de entrenamiento. Si no se incluye ningún campo de partición en la ruta, se ignorará esta opción.

La propiedad del nodo `directives` utiliza este formato:

`[{ number string } \ { number string } \ ... { number string }]`.

*Note:* Si las cadenas (que utilizan comillas dobles) están incrustadas en la expresión, han de estar precedidas del carácter de escape " \ ". El carácter " \ " es también el carácter de continuación de línea, que le permite alinear los argumentos para la claridad.

## Propiedades del nodo derive\_stb



El nodo Cajas-espacio-tiempo deriva Cajas-espacio-tiempo de los campos latitud, longitud e indicación de fecha y hora. Las Cajas-espacio-tiempo también pueden identificarse como lugares comunes.

Tabla 47. Propiedades del nodo derive\_stb.

Propiedades del nodo derive_stb	Tipo de datos	Descripción de la propiedad
mode	IndividualRecords Hangouts	
latitude_field	campo	



Tabla 47. Propiedades del nodo *derive\_stb* (continuación).

Propiedades del nodo <i>derive_stb</i>	Tipo de datos	Descripción de la propiedad
<i>longitude_field</i>	<i>campo</i>	
<i>timestamp_field</i>	<i>campo</i>	
<i>hangout_density</i>	<i>densidad</i>	Una sola densidad. Consulte en <i>densities</i> los valores de densidad válidos.
<i>densities</i>	[ <i>densidad, densidad, ..., densidad</i> ]	Cada densidad es una cadena, por ejemplo, STB_GH8_1DAY. <b>Nota:</b> Existen límites para que las densidades sean válidas. En geohash, se pueden utilizar los valores de GH1 a GH15. Para la parte temporal, se pueden utilizar los valores siguientes: EVER 1YEAR 1MONTH 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2HOURS 1HOUR 30MINS 15MINS 10MINS 5MINS 2MINS 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SEC
<i>id_field</i>	<i>campo</i>	
<i>qualifying_duration</i>	1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2Hours 1HOUR 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS	Debe ser una cadena.
<i>min_events</i>	<i>entero</i>	El valor de entero válido mínimo es de 2.
<i>qualifying_pct</i>	<i>entero</i>	Debe estar en el rango de 1 a 100.

Tabla 47. Propiedades del nodo *derive\_stb* (continuación).

Propiedades del nodo <i>derive_stb</i>	Tipo de datos	Descripción de la propiedad
add_extension_as	Prefix Suffix	
name_extension	<i>cadena</i>	

## Propiedades del nodo *distinct*



El nodo Distinguir se puede usar para eliminar registros duplicados pasando el primero de los registros distintos a la ruta de datos o descartando el primer registro y pasando cualquier duplicado a la ruta de datos en su lugar.

Tabla 48. Propiedades del nodo *distinct*.

Propiedades del nodo <i>distinct</i>	Tipo de datos	Descripción de la propiedad
mode	Incluir Discard	Se puede incluir el primer nodo distinto en la ruta o descartar el primer nodo distinto y pasar en su lugar todos los registros duplicados a la ruta de datos.
grouping_fields	[ <i>field field field</i> ]	Enumera los campos utilizados para determinar si los registros son idénticos. <b>Nota:</b> Esta propiedad está en desuso desde IBM SPSS Modeler 16 y posterior.
composite_value	Intervalo estructurado	
composite_values	Intervalo estructurado	
inc_record_count	<i>booleano</i>	Crea un campo adicional que indica la cantidad de registros de entrada agregados para conformar cada registro agregado.
count_field	<i>cadena</i>	Especifica el nombre del campo de recuento de registros.
sort_keys	Intervalo estructurado	<b>Nota:</b> Esta propiedad está en desuso desde IBM SPSS Modeler 16 y posterior.
default_ascending	<i>booleano</i>	
low_distinct_key_count	<i>booleano</i>	Especifica que sólo tiene un pequeño número de registros y/o un pequeño número de valores exclusivos del campo(s) clave.
keys_pre_sorted	<i>booleano</i>	Especifica que todos los registros con los mismos valores clave se agrupan en la entrada.
disable_sql_generation	<i>booleano</i>	

## Propiedades del nodo *merge*



El nodo Fundir toma varios registros de entrada y crea un registro de salida único que contiene todos o algunos de los campos de entrada. Es útil para fusionar datos desde diferentes orígenes, como datos de clientes internos y datos demográficos adquiridos.

Tabla 49. Propiedades del nodo merge.

Propiedades del nodo merge	Tipo de datos	Descripción de la propiedad
método	Order Claves Condition	Determina si los registros se fusionan en el orden en el que aparecen enumerados en los archivos de datos, si se usan uno o varios campos clave para fusionar registros con el mismo valor en los campos clave o si los registros se fusionan si se cumple una condición determinada.
condition	<i>cadena</i>	Si method se establece en Condition, especifica la condición para incluir o descartar registros.
key_fields	[ <i>field field field</i> ]	
common_keys	<i>booleano</i>	
join	Interior FullOuter PartialOuter Anti	
outer_join_tag.n	<i>booleano</i>	En esta propiedad, <i>n</i> es el nombre de etiqueta tal y como recoge el cuadro de diálogo Seleccionar conjunto de datos. Tenga en cuenta que es posible que existan varios nombres de etiquetas especificados, ya que pueden ser varios los conjuntos de datos que aporten registros incompletos.
single_large_input	<i>booleano</i>	Determina si se va a usar la optimización para tener una entrada relativamente grande en comparación con el resto de entradas.
single_large_input_tag	<i>cadena</i>	Especifica el nombre de etiqueta tal y como se muestra en el cuadro de diálogo Seleccionar conjunto de datos grande. Tenga en cuenta que el uso de esta propiedad es ligeramente distinto que el de la propiedad outer_join_tag (booleano comparado con cadena), ya que solamente se puede especificar un único conjunto de datos de entrada.
use_existing_sort_keys	<i>booleano</i>	Determina si las entradas ya se han ordenado en función de uno o varios campos clave.
existing_sort_keys	[[ <i>string</i> Ascending] \ <i>string</i> Descending]]	Especifica los campos que ya están ordenados y la dirección en que dicho orden se ha establecido.

## Propiedades del nodo rfmaggregate



El nodo Adición de RFM (actualidad, frecuencia, monetario) permite tomar datos de transacciones históricas de clientes, deshacerse de los datos no utilizados y combinar todos los datos de transacciones restantes en una única fila que indica cuándo hizo negociaciones con los clientes por última vez, cuántas transacciones hicieron y el valor monetario total de dichas transacciones.

Tabla 50. Propiedades del nodo rfmaggregate.

Propiedades del nodo rfmaggregate	Tipo de datos	Descripción de la propiedad
relative_to	Fixed Today	Especifica la fecha a partir de la que se calculará la actualidad de las transacciones.
reference_date	date	Sólo está disponible si se selecciona Fixed en relative_to.
contiguous	booleano	Si los datos se han clasificado previamente de forma que todos los registros con el mismo ID aparecen en la ruta de datos, al seleccionar esta opción acelerará el procesamiento.
id_field	field	Especifica el campo que desea utilizar para identificar el cliente y sus transacciones.
date_field	field	Especifica el campo de fecha que se utilizará para calcular la actualidad.
value_field	field	Especifica el campo que se utilizará para calcular el valor monetario.
extension	cadena	Especifica un prefijo o sufijo para campos agregados duplicados.
add_as	Suffix Prefix	Especifica si la extensión se debe añadir como sufijo o prefijo.
discard_low_value_records	booleano	Permite utilizar la configuración discard_records_below.
discard_records_below	número	Especifique un valor mínimo por debajo del cual no se utilice ningún detalle de transacción al calcular los totales de RFM. Las unidades del valor se relacionan con el campo valor seleccionado.
only_recent_transactions	booleano	Permite utilizar de la configuración specify_transaction_date o transaction_within_last.
specify_transaction_date	booleano	
transaction_date_after	date	Sólo está disponible si selecciona specify_transaction_date. Especifica la fecha de transacción tras la que se incluirán los registros en su análisis.
transaction_within_last	número	Sólo está disponible si selecciona transaction_within_last. Especifica el número y tipo de períodos (días, semanas, meses o años) desde la fecha Calcular actualidad relativa a tras la cual se incluirán los registros en su análisis.
transaction_scale	Days Weeks Meses Años	Sólo está disponible si selecciona transaction_within_last. Especifica el número y tipo de períodos (días, semanas, meses o años) desde la fecha Calcular actualidad relativa a tras la cual se incluirán los registros en su análisis.
save_r2	booleano	Muestra la fecha de la segunda transacción más reciente para cada cliente.
save_r3	booleano	Sólo está disponible si selecciona save_r2. Muestra la fecha de la tercera transacción más reciente para cada cliente.

## Propiedades del nodo Rprocess



El nodo Rprocess le permite tomar los datos de una cadena de IBM(r) SPSS(r) Modeler y modificarlos utilizando su propio script R personalizado. Una vez modificados los datos, se devuelven a la cadena.

Tabla 51. Propiedades del nodo Rprocess.

Propiedades del nodo Rprocess	Tipo de datos	Descripción de la propiedad
syntax	cadena	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	booleano	
convert_datetime_class	POSIXct POSIXlt	
convert_missing	booleano	

## Propiedades del nodo sample



El nodo Muestrear selecciona un subconjunto de registros. Se admite una variedad de tipos de muestras, entre las que se incluyen las muestras estratificadas, agrupadas en clústeres y no aleatorias (estructuradas). El muestreo puede ser de gran utilidad para mejorar el rendimiento y para seleccionar grupos de registros o transacciones relacionadas para un análisis.

Tabla 52. Propiedades de nodo de ejemplo.

Propiedades del nodo sample	Tipo de datos	Descripción de la propiedad
método	Simple Complex	
mode	Incluir Discard	Incluye o descarta los registros que reúnan la condición especificada.
sample_type	Primero OneInN RandomPct	Especifica el método de muestreo.
first_n	entero	Se incluirán o descartarán los registros hasta el punto de corte especificado.
one_in_n	número	Incluye o descarta cada <i>n</i> registros.
rand_pct	número	Especifica el porcentaje de registros que incluir o descartar.
use_max_size	booleano	Activa el uso del parámetro maximum_size.
maximum_size	entero	Especifica la muestra más grande que se va a incluir o descartar de la ruta de datos. Esta opción es redundante, por lo que se desactiva cuando se especifican las opciones First e Include.
set_random_seed	booleano	Activa el uso del parámetro de semillas aleatorias.

Tabla 52. Propiedades de nodo de ejemplo (continuación).

Propiedades del nodo <b>sample</b>	Tipo de datos	Descripción de la propiedad
random_seed	entero	Especifica el valor utilizado como semilla aleatoria.
complex_sample_type	Random Systematic	
sample_units	Proportions Counts	
sample_size_proportions	Fixed Custom Variable	
sample_size_counts	Fixed Custom Variable	
fixed_proportions	número	
fixed_counts	entero	
variable_proportions	field	
variable_counts	field	
use_min_stratum_size	booleano	
minimum_stratum_size	entero	Esta opción sólo se aplica cuando se toma una muestra compleja con Sample units=Proportions.
use_max_stratum_size	booleano	
maximum_stratum_size	entero	Esta opción sólo se aplica cuando se toma una muestra compleja con Sample units=Proportions.
clusters	field	
stratify_by	[campo1 ... campoN]	
specify_input_weight	booleano	
input_weight	field	
new_output_weight	cadena	
sizes_proportions	[[string string value]{string string value}...]	Si sample_units=proportions y sample_size_proportions=Custom, especifica un valor para cada combinación posible de valores de campos de especificación.
default_proportion	número	
sizes_counts	[[string string value]{string string value}...]	Especifica un valor para cada combinación de valores posible de campos de estratificación. Se utiliza de forma similar a sizes_proportions pero especificando un entero en lugar de una proporción.
default_count	número	

## Propiedades del nodo **select**



El nodo Seleccionar selecciona o descarta un subconjunto de registros de la ruta de datos en función de una condición específica. Por ejemplo, podría seleccionar los registros que pertenezcan a una región de ventas determinada.

Tabla 53. Propiedades del nodo select.

Propiedades del nodo select	Tipo de datos	Descripción de la propiedad
mode	Incluir Discard	Especifica si incluir o descartar los registros seleccionados.
condition	cadena	Condición para incluir o descartar registros.

## Propiedades del nodo sort



Los nodos Ordenar organizan registros en orden ascendente o descendente atendiendo a los valores de uno o varios campos.

Tabla 54. Propiedades del nodo sort.

Propiedades del nodo sort	Tipo de datos	Descripción de la propiedad
keys	[[cadena Ascending] \ {cadena Descending}]	Especifica los campos que desea ordenar. Si no se especifica ninguna dirección, se utilizará la predeterminada.
default_ascending	booleano	Especifica el orden de clasificación predeterminado.
use_existing_keys	booleano	Determina si la clasificación se optimiza usando el orden de clasificación anterior de los campos que ya están ordenados.
existing_keys		Especifica los campos que ya están ordenados y la dirección en que dicho orden se ha establecido. Utiliza el mismo formato que las propiedades keys.

## Propiedades del nodo Generación de análisis TS (streamingts)



El nodo Generación de análisis TS construye y puntúa modelos de series temporales en un paso, sin la necesidad de un nodo Intervalos de tiempo.

Tabla 55. propiedades del nodo generación de análisis ts.

Propiedades del nodo streamingts	Tipo de datos	Descripción de la propiedad
custom_fields	booleano	Si custom_fields=false, se utilizarán los valores de un nodo Tipo situado en un punto anterior de la ruta. Si custom_fields=true, deberán especificarse targets e inputs.
targets	[campo1...campoN]	
inputs	[campo1...campoN]	
method	ExpertModeler Exsmooth Arima	
calculate_conf	booleano	
conf_limit_pct	real	

Tabla 55. propiedades del nodo generación de análisis ts (continuación).

Propiedades del nodo streamingts	Tipo de datos	Descripción de la propiedad
use_time_intervals_node	booleano	Si use_time_intervals_node=true, se utilizarán los valores de un nodo Intervalos de tiempo situado en un punto anterior de la ruta. De lo contrario, deberán especificarse interval_offset_position, interval_offset, e interval_type.
interval_offset_position	LastObservation LastRecord	LastObservation se refiere a la <b>Última observación válida</b> . LastRecord se refiere a la <b>Cuenta hacia atrás a partir del último registro</b> .
interval_offset	número	
interval_type	Períodos Años Trimestres Meses WeeksNonPeriodic DaysNonPeriodic HoursNonPeriodic MinutesNonPeriodic SecondsNonPeriodic	
eventos	campos	
expert_modeler_method	AllModels Exsmooth Arima	
consider_seasonal	booleano	
detect_outliers	booleano	
expert_outlier_additive	booleano	
expert_outlier_level_shift	booleano	
expert_outlier_innovational	booleano	
expert_outlier_transient	booleano	
expert_outlier_seasonal_additive	booleano	
expert_outlier_local_trend	booleano	
expert_outlier_additive_patch	booleano	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	Ninguno SquareRoot NaturalLog	
arima_p	entero	Misma propiedad que el nodo de modelado Serie de tiempo
arima_d	entero	Misma propiedad que el nodo de modelado Serie de tiempo
arima_q	entero	Misma propiedad que el nodo de modelado Serie de tiempo



Tabla 55. propiedades del nodo generación de análisis ts (continuación).

Propiedades del nodo streamingts	Tipo de datos	Descripción de la propiedad
arima_sp	entero	Misma propiedad que el nodo de modelado Serie de tiempo
arima_sd	entero	Misma propiedad que el nodo de modelado Serie de tiempo
arima_sq	entero	Misma propiedad que el nodo de modelado Serie de tiempo
arima_transformation_type	Ninguno SquareRoot NaturalLog	Misma propiedad que el nodo de modelado Serie de tiempo
arima_include_constant	booleano	Misma propiedad que el nodo de modelado Serie de tiempo
tf_arima_p.nombrecampo	entero	Misma propiedad que el nodo de modelado Serie de tiempo. Para funciones de transferencia.
tf_arima_d.nombrecampo	entero	Misma propiedad que el nodo de modelado Serie de tiempo. Para funciones de transferencia.
tf_arima_q.nombrecampo	entero	Misma propiedad que el nodo de modelado Serie de tiempo. Para funciones de transferencia.
tf_arima_sp.nombrecampo	entero	Misma propiedad que el nodo de modelado Serie de tiempo. Para funciones de transferencia.
tf_arima_sd.nombrecampo	entero	Misma propiedad que el nodo de modelado Serie de tiempo. Para funciones de transferencia.
tf_arima_sq.nombrecampo	entero	Misma propiedad que el nodo de modelado Serie de tiempo. Para funciones de transferencia.
tf_arima_delay.nombrecampo	entero	Misma propiedad que el nodo de modelado Serie de tiempo. Para funciones de transferencia.
tf_arima_transformation_type.nombredecampo	Ninguno SquareRoot NaturalLog	
arima_detect_outlier_mode	Ninguno Automatic	
arima_outlier_additive	booleano	
arima_outlier_level_shift	booleano	
arima_outlier_innovational	booleano	
arima_outlier_transient	booleano	
arima_outlier_seasonal_additive	booleano	
arima_outlier_local_trend	booleano	
arima_outlier_additive_patch	booleano	
deployment_force_rebuild	booleano	
deployment_rebuild_mode	Recuento Porcentaje	
deployment_rebuild_count	número	
deployment_rebuild_pct	número	
deployment_rebuild_field	<campo>	



---

## Capítulo 11. Propiedades de nodos de operaciones con campos

---

### Propiedades del nodo anonymize



El nodo Anonimizar transforma la manera en que se representan los nombres y los valores de los campos a partir de ese punto de la ruta, lo que permite disfrazar los datos originales. Puede resultar útil si desea permitir que otros usuarios generen modelos utilizando datos confidenciales, como los nombres de los clientes u otros detalles.

Tabla 56. propiedades del nodo anonymize.

Propiedades del nodo anonymize	Tipo de datos	Descripción de la propiedad
enable_anonymize	<i>booleano</i>	Cuando se establece en T, activa la anonimización de los valores de los campos (equivale a seleccionar <b>Sí</b> para dicho campo en la columna Anonimizar valores).
use_prefix	<i>booleano</i>	Cuando se establece en T, se utilizará un prefijo personalizado, si es que se ha establecido uno. Se aplica a los campos que se anonimizarán mediante el método Hash y es equivalente a elegir el botón de radio <b>Personalizado</b> en el cuadro de diálogo Reemplazar valores correspondiente a dicho campo.
prefix	<i>cadena</i>	Equivale a escribir un prefijo en el cuadro de texto del cuadro de diálogo Reemplazar valores. El prefijo predeterminado es el valor predeterminado, si no se ha especificado otra cosa.
transformation	Random Fixed	Determina si los parámetros de transformación que se aplican a un campo anonimizado mediante el método Transformar serán aleatorios o fijos.
set_random_seed	<i>booleano</i>	Cuando se establece en T, se utilizará el valor especificado para la semilla (si también se ha establecido transformation en Random).
random_seed	<i>entero</i>	Cuando set_random_seed se establece en T, este número es la semilla de los números aleatorios.
scale	<i>número</i>	Cuando transformation se establece en Fixed, este valor se utiliza para escalar los valores. El valor máximo de la escala suele ser 10, pero puede reducirse para evitar desbordamientos.
translate	<i>número</i>	Si transformation se establece en Fixed, este valor se utiliza para traducir. El valor máximo de traslación suele ser 1000, pero puede reducirse para evitar desbordamientos.

---

### Propiedades del nodo autodataprep



El nodo de preparación automática de datos (ADP) puede analizar sus datos e identificar los valores fijos, cribar los campos problemáticos o que no serán útiles y derivar nuevos atributos cuando sea necesario y mejorar el rendimiento mediante técnicas de cribado y muestreo inteligente. Puede utilizar el nodo de forma totalmente automática, permitiendo que el nodo seleccione y aplique valores fijos, o bien puede tener una vista previa de los cambios antes de que se apliquen y aceptarlos o rechazarlos.

Tabla 57. Propiedades del nodo autodataprep.

Propiedades del nodo autodataprep	Tipo de datos	Descripción de la propiedad
objective	Balanced Speed Accuracy Custom	
custom_fields	booleano	Si es verdadero, le permite especificar el objetivo, la entrada y otros campos del nodo actual. Si es falso, se utiliza la configuración actual de un nodo Tipo situado en un punto anterior de la ruta.
target	field	Especifica un campo de objetivo único.
inputs	[field1 ... fieldN]	Campos de entrada o predictor utilizados por el modelo.
use_frequency	booleano	
frequency_field	field	
use_weight	booleano	
weight_field	field	
excluded_fields	Filter None	
if_fields_do_not_match	StopExecution ClearAnalysis	
prepare_dates_and_times	booleano	Controla el acceso a todos los campos de fecha y hora
compute_time_until_date	booleano	
reference_date	Today Fixed	
fixed_date	date	
units_for_date_durations	Automatic Fixed	
fixed_date_units	Años Meses Days	
compute_time_until_time	booleano	
reference_time	CurrentTime Fixed	
fixed_time	time	
units_for_time_durations	Automatic Fixed	
fixed_date_units	Hours Minutes Seconds	
extract_year_from_date	booleano	
extract_month_from_date	booleano	
extract_day_from_date	booleano	
extract_hour_from_time	booleano	
extract_minute_from_time	booleano	
extract_second_from_time	booleano	

Tabla 57. Propiedades del nodo autodataprep (continuación).

Propiedades del nodo autodataprep	Tipo de datos	Descripción de la propiedad
exclude_low_quality_inputs	booleano	
exclude_too_many_missing	booleano	
maximum_percentage_missing	número	
exclude_too_many_categories	booleano	
maximum_number_categories	número	
exclude_if_large_category	booleano	
maximum_percentage_category	número	
prepare_inputs_and_target	booleano	
adjust_type_inputs	booleano	
adjust_type_target	booleano	
reorder_nominal_inputs	booleano	
reorder_nominal_target	booleano	
replace_outliers_inputs	booleano	
replace_outliers_target	booleano	
replace_missing_continuous_inputs	booleano	
replace_missing_continuous_target	booleano	
replace_missing_nominal_inputs	booleano	
replace_missing_nominal_target	booleano	
replace_missing_ordinal_inputs	booleano	
replace_missing_ordinal_target	booleano	
maximum_values_for_ordinal	número	
minimum_values_for_continuous	número	
outlier_cutoff_value	número	
outlier_method	Replace Delete	
rescale_continuous_inputs	booleano	
rescaling_method	MinMax ZScore	
min_max_minimum	número	
min_max_maximum	número	
z_score_final_mean	número	
z_score_final_sd	número	
rescale_continuous_target	booleano	
target_final_mean	número	
target_final_sd	número	
transform_select_input_fields	booleano	
maximize_association_with_target	booleano	
p_value_for_merging	número	
merge_ordinal_features	booleano	
merge_nominal_features	booleano	

Tabla 57. Propiedades del nodo autodataprep (continuación).

Propiedades del nodo autodataprep	Tipo de datos	Descripción de la propiedad
minimum_cases_in_category	número	
bin_continuous_fields	booleano	
p_value_for_binning	número	
perform_feature_selection	booleano	
p_value_for_selection	número	
perform_feature_construction	booleano	
transformed_target_name_extension	cadena	
transformed_inputs_name_extension	cadena	
constructed_features_root_name	cadena	
years_duration_name_extension	cadena	
months_duration_name_extension	cadena	
days_duration_name_extension	cadena	
hours_duration_name_extension	cadena	
minutes_duration_name_extension	cadena	
seconds_duration_name_extension	cadena	
year_cyclical_name_extension	cadena	
month_cyclical_name_extension	cadena	
day_cyclical_name_extension	cadena	
hour_cyclical_name_extension	cadena	
minute_cyclical_name_extension	cadena	
second_cyclical_name_extension	cadena	

## Propiedades del nodo binning



El nodo Intervalos crea automáticamente nuevos campos nominales (conjunto) en función de los valores de uno o más campos continuos (rango numérico) existentes. Por ejemplo, puede transformar un campo de ingresos continuo en un campo categórico nuevo que contenga grupos de ingresos como desviaciones desde la media. Una vez creados los intervalos para el campo nuevo, puede generar un nodo Derivar en función de los puntos de corte.

Tabla 58. Propiedades del nodo binning.

Propiedades del nodo binning	Tipo de datos	Descripción de la propiedad
fields	[campo1 campo2 ... campon]	Los campos continuos (rango numérico) pendientes de transformación. Se pueden crear intervalos de varios campos de forma simultánea.
método	FixedWidth EqualCount Rank SDev Optimal	Método utilizado para determinar los puntos de corte de los intervalos de campo nuevos (categorías).

Tabla 58. Propiedades del nodo binning (continuación).

Propiedades del nodo binning	Tipo de datos	Descripción de la propiedad
rcalculate_bins	Always IfNecessary	Especifica si se vuelven a calcular los intervalos y los datos se colocan en el intervalo adecuado cada vez que se ejecuta el nodo o si los datos sólo se añaden a los intervalos existentes y cualquier nuevo intervalo que se haya añadido.
fixed_width_name_extension	cadena	La extensión predeterminada es <i>_BIN</i> .
fixed_width_add_as	Suffix Prefix	Determina si la extensión se debe añadir al principio (prefijo) o al final (sufijo) del nombre de campo. La extensión predeterminada es <i>income_BIN</i> .
fixed_bin_method	Width Recuento	
fixed_bin_count	entero	Especifica un número entero para determinar el número de intervalos de anchura fija (categorías) para los nuevos campos.
fixed_bin_width	real	Valor (entero o real) para calcular el ancho del intervalo.
equal_count_name_extensión	cadena	La extensión predeterminada es <i>_TILE</i> .
equal_count_add_as	Suffix Prefix	Especifica una extensión, sufijo o prefijo, utilizada para el nombre de los campos generados con p-tiles estándar. La extensión predeterminada es <i>_TILE</i> más <i>N</i> , donde <i>N</i> es el número de cuantil.
tile4	booleano	Genera cuatro intervalos de cuantiles, cada uno con el 25% de los casos.
tile5	booleano	Genera cinco intervalos de quintiles.
tile10	booleano	Genera 10 intervalos de deciles.
tile20	booleano	Genera 20 intervalos de veintiles.
tile100	booleano	Genera 100 intervalos de percentiles.
use_custom_tile	booleano	
custom_tile_name_extension	cadena	La extensión predeterminada es <i>_TILEN</i> .
custom_tile_add_as	Suffix Prefix	
custom_tile	entero	
equal_count_method	RecordCount ValueSum	El método RecordCount trata de asignar el mismo número de registros a cada intervalo, mientras que ValueSum asigna registros de manera que la suma de los valores de cada intervalo sea la misma.
tied_values_method	Next Actual Random	Especifica en qué intervalo se van a insertar los datos de valor empatado.
rank_order	Ascending Descending	Esta propiedad incluye Ascending (el valor más bajo se marca con 1) o Descending (el valor más alto se marca con 1).

Tabla 58. Propiedades del nodo binning (continuación).

Propiedades del nodo binning	Tipo de datos	Descripción de la propiedad
rank_add_as	Suffix Prefix	Esta opción se aplica al rango, rango fraccional y rango como porcentaje.
rank	booleano	
rank_name_extension	cadena	La extensión predeterminada es <i>_RANK</i> .
rank_fractional	booleano	Establece rangos de casos en los que el valor del campo nuevo es igual al rango dividido por la suma de las ponderaciones de los casos que no están perdidos. Los rangos fraccionales están dentro del rango de 0–1.
rank_fractional_name_extension	cadena	La extensión predeterminada es <i>_F_RANK</i> .
rank_pct	booleano	Cada rango se divide por el número de registros con valores válidos y se multiplica por 100. Los rangos fraccionales de porcentaje están dentro del rango de 1–100.
rank_pct_name_extension	cadena	La extensión predeterminada es <i>_P_RANK</i> .
sdev_name_extension	cadena	
sdev_add_as	Suffix Prefix	
sdev_count	One Two Three	
optimal_name_extension	cadena	La extensión predeterminada es <i>_OPTIMAL</i> .
optimal_add_as	Suffix Prefix	
optimal_supervisor_field	field	Campo elegido como campo supervisor, con el que se relacionan los campos seleccionados para los intervalos.
optimal_merge_bins	booleano	Especifica que todos los intervalos con un número pequeño de casos se añadirán a un intervalo vecino de mayor tamaño.
optimal_small_bin_threshold	entero	
optimal_pre_bin	booleano	Indica si debe agruparse previamente en intervalos el conjunto de datos.
optimal_max_bins	entero	Especifica un límite superior con el fin de evitar que se genere un número desmesurado de intervalos.
optimal_lower_end_point	Inclusive Exclusive	
optimal_first_bin	Unbounded Bounded	
optimal_last_bin	Unbounded Bounded	



## Propiedades del nodo derive



El nodo Derivar modifica los valores de datos o crea campos nuevos desde uno o más campos existentes. Crea campos del tipo fórmula, marca, nominal, estado, recuento y condicional.

Tabla 59. Propiedades del nodo derivar.

Propiedades del nodo derive	Tipo de datos	Descripción de la propiedad
new_name	cadena	Nombre del campo nuevo.
mode	Single Multiple	Especifica si los campos son únicos o múltiples.
fields	[field field field]	Se utiliza en modo Múltiple solamente para seleccionar varios campos.
name_extension	cadena	Especifica la extensión de los nombres de los nuevos campos.
add_as	Suffix Prefix	Añade la extensión como un prefijo (al principio) o como un sufijo (al final) del nombre de los campos.
result_type	Formula Flag Set State Recuento Conditional	Los seis tipos de campos nuevos que se pueden crear.
formula_expr	cadena	Expresión para calcular un nuevo valor de campo en el nodo Derivar.
flag_expr	cadena	
flag_true	cadena	
flag_false	cadena	
set_default	cadena	
set_value_cond	cadena	Estructurada para proporcionar la condición asociada a un valor dado.
state_on_val	cadena	Especifica el valor del campo nuevo cuando se cumple la condición Activado.
state_off_val	cadena	Especifica el valor del campo nuevo cuando se cumple la condición Desactivado.
state_on_expression	cadena	
state_off_expression	cadena	
state_initial	On Off	Asigna a cada registro del nuevo campo un valor inicial activado (On) o desactivado (Off). Este valor puede cambiar a medida que se cumplan las condiciones.
count_initial_val	cadena	
count_inc_condition	cadena	
count_inc_expression	cadena	
count_reset_condition	cadena	
cond_if_cond	cadena	

Tabla 59. Propiedades del nodo derivar (continuación).

Propiedades del nodo derive	Tipo de datos	Descripción de la propiedad
cond_then_expr	cadena	
cond_else_expr	cadena	

## Propiedades del nodo ensemble



El nodo Conjunto combina dos o más nugget de modelo para obtener predicciones más precisas que pueden conseguirse de cualquier modelo.

Tabla 60. Propiedades del nodo ensemble.

Propiedades del nodo ensemble	Tipo de datos	Descripción de la propiedad
ensemble_target_field	field	Especifica el campo objetivo de todos los modelos utilizados en el conjunto.
filter_individual_model_output	booleano	Especifica si los resultados de puntuación de los modelos individuales se deben eliminar.
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity	Especifica el método utilizado para determinar la puntuación del conjunto. Este conjunto sólo se aplica si el objetivo seleccionado es un campo de marca.
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	Especifica el método utilizado para determinar la puntuación del conjunto. Este conjunto sólo se aplica si el objetivo seleccionado es un campo nominal.
flag_voting_tie_selection	Random HighestConfidence RawPropensity AdjustedPropensity	Si se selecciona un método de votación, especifica cómo se resolverán los empates. Este conjunto sólo se aplica si el objetivo seleccionado es un campo de marca.
set_voting_tie_selection	Random HighestConfidence	Si se selecciona un método de votación, especifica cómo se resolverán los empates. Este conjunto sólo se aplica si el objetivo seleccionado es un campo nominal.
calculate_standard_error	booleano	Si el campo objetivo es continuo, se ejecuta un error estándar de forma predeterminada para calcular la diferencia entre los valores medidos o estimados y los valores true; y para mostrar si las estimaciones coinciden.

## Propiedades del nodo filler



El nodo Rellenar sustituye valores de campos y cambia el almacenamiento. Puede sustituir los valores en función de una condición CLEM, como @BLANK(@FIELD). También puede sustituir todos los espacios vacíos o valores nulos por un valor específico. Un nodo Rellenar suelen utilizarse junto con un nodo Tipo para sustituir valores perdidos.

Tabla 61. Propiedades del nodo filler.

Propiedades del nodo filler	Tipo de datos	Descripción de la propiedad
fields	[field field field]	Campos del conjunto de datos cuyos valores se van a examinar y sustituir.
replace_mode	Always Conditional Vacío Null BlankAndNull	Se pueden sustituir todos los valores, valores vacíos o valores nulos, o bien reemplazar aquellos basados en una condición específica.
condition	cadena	
replace_with	cadena	

## Propiedades del nodo filter



El nodo Filtrar filtra (descarta) campos, vuelve a nombrar campos y correlaciona campos de nodo de origen a otro.

**Usando la propiedad default\_include.** Tenga en cuenta que, si establece el valor de la propiedad default\_include, no se incluirán o excluirán automáticamente todos los campos, sino que simplemente se determinará el valor predeterminado de los seleccionados actualmente. Esto equivale funcionalmente a pulsar en el botón **Incluir campos de forma predeterminada** del cuadro de diálogo del nodo Filtrar.

Tabla 62. Propiedades del nodo filter.

Propiedades del nodo filter	Tipo de datos	Descripción de la propiedad
default_include	booleano	Propiedad con clave para especificar si el comportamiento predeterminado es el de pasar o filtrar los campos. Tenga en cuenta que, si se establece esta propiedad, no se incluirán o excluirán automáticamente todos los campos, sino que simplemente se determinará si los campos seleccionados se incluyen o excluyen de forma predeterminada.
include	booleano	Propiedad con clave para incluir y eliminar el campo.
new_name	cadena	

## Propiedades del nodo history



El nodo Historial se utiliza para crear campos nuevos que contienen datos de los campos de registros anteriores. Los nodos Historial se suelen utilizar para los datos secuenciales, como los datos de series temporales. Antes de utilizar un nodo Historial, puede desear ordenar los datos utilizando un nodo Ordenar.

Tabla 63. Propiedades del nodo history.

Propiedades del nodo history	Tipo de datos	Descripción de la propiedad
fields	[field field field]	Campos para los que desea un historial.
offset	número	Especifica el último registro anterior al registro actual desde el que desea extraer valores del campo histórico.
span	número	Especifica el número de registros anteriores de los que desea extraer valores.
unavailable	Discard Leave Fill	Para tratar registros que no tienen valores de historial, suele hacer referencia a los primeros registros (en la parte superior del conjunto de datos), de los que no hay registros previos que utilizar como historial.
fill_with	String Number	Especifica un valor o cadena que utilizar en el caso de los registros en los que no existen valores de historial disponibles.

## Propiedades del nodo partition



El nodo Partición genera un campo de partición, que divide los datos en subconjuntos diferentes para las fases de entrenamiento, comprobación y validación en la generación del modelo.

Tabla 64. Propiedades del nodo partition.

Propiedades del nodo partition	Tipo de datos	Descripción de la propiedad
new_name	cadena	Nombre del campo de partición generado por el nodo.
create_validation	booleano	Especifica si se debe crear una partición de validación.
training_size	entero	Porcentaje de registros (0-100) que se van a asignar a la partición de entrenamiento.
resting_size	entero	Porcentaje de registros (0-100) que se van a asignar a la partición de comprobación.
validation_size	entero	Porcentaje de registros (0-100) que se van a asignar a la partición de entrenamiento. Se ignora si no se crea una partición de validación.
training_label	cadena	Etiqueta para la partición de entrenamiento.
testing_label	cadena	Etiqueta para la partición de comprobación.
validation_label	cadena	Etiqueta para la partición de validación. Se ignora si no se crea una partición de validación.

Tabla 64. Propiedades del nodo partition (continuación).

Propiedades del nodo partition	Tipo de datos	Descripción de la propiedad
value_mode	Sistema SystemAndLabel Label	Especifica los valores utilizados para representar cada partición en los datos. Por ejemplo, el entero del sistema 1, la etiqueta Entrenamiento o una combinación de los dos, 1_Entrenamiento, pueden representar la muestra de entrenamiento.
set_random_seed	booleano	Especifica si se debe utilizar una semilla aleatoria especificada por el usuario.
random_seed	entero	Valor de semilla aleatoria especificada por el usuario. Para que se pueda utilizar este valor, set_random_seed se debe establecer en True.
enable_sql_generation	booleano	Especifica si se utiliza la retrotracción SQL para asignar registros a particiones.
unique_field		Especifica el campo de entrada que se utiliza para garantizar que los registros se asignan a particiones de una forma aleatoria pero reproducible. Para que se pueda utilizar este valor, enable_sql_generation se debe establecer en True.

## Propiedades del nodo reclassify



El nodo Reclasificar transforma un conjunto de valores categóricos en otro. La reclasificación es útil para contraer categorías o reagrupar datos para su análisis.

Tabla 65. Propiedades del nodo reclassify.

Propiedades del nodo reclassify	Tipo de datos	Descripción de la propiedad
mode	Single Multiple	Single reclasifica las categorías de un campo. Multiple activa las opciones que permiten la transformación de varios campos al mismo tiempo.
replace_field	booleano	
campo	cadena	Sólo se utiliza en modo Single.
new_name	cadena	Sólo se utiliza en modo Single.
fields	[campo1 campo2 ... campon]	Sólo se utiliza en modo Multiple.
name_extension	cadena	Sólo se utiliza en modo Multiple.
add_as	Suffix Prefix	Sólo se utiliza en modo Multiple.
reclassify	cadena	Propiedad estructurada para valores de campos.
use_default	booleano	Utiliza el valor predeterminado.
default	cadena	Especifica un valor predeterminado.
pick_list	[string string ... string]	Permite al usuario importar una lista de valores nuevos conocidos para rellenar la lista desplegable de la tabla.

## Propiedades del nodo reorder



El nodo Reorg. campos define el orden natural utilizado para mostrar los campos en la parte posterior de la ruta. Este orden afecta a la visualización de los campos en diversas ubicaciones, como las tablas, las listas y el selector de campos. Esta operación resulta útil al trabajar con conjuntos de datos amplios que hacen más visibles los campos de interés.

Tabla 66. Propiedades del nodo reorder.

Propiedades del nodo reorder	Tipo de datos	Descripción de la propiedad
mode	Custom Auto	Se pueden ordenar los valores de forma automática o especificar un orden personalizado.
sort_by	Name Tipo Storage	
ascending	booleano	
start_fields	[field1 field2 ... fieldn]	Los campos nuevos se han insertado después de estos campos.
end_fields	[field1 field2 ... fieldn]	Los campos nuevos se han insertado antes de estos campos.

## Propiedades del nodo restructure



El nodo Reestructurar convierte un campo nominal o marca en un grupo de campos que se puede rellenar con los valores todavía de otro campo. Por ejemplo, para un campo determinado llamado *tipo de pago*, con valores de *crédito*, *efectivo*, y *débito*, se crearían tres campos nuevos (*crédito*, *efectivo*, *débito*), que contendría cada uno el valor del pago real realizado.

Tabla 67. Propiedades del nodo restructure.

Propiedades del nodo restructure	Tipo de datos	Descripción de la propiedad
fields_from	[category category category] all	
include_field_name	booleano	Indica si se debe usar el nombre del campo en el nombre de campo reestructurado.
value_mode	OtherFields Flags	Indica el modo de definir los valores para los campos reestructurados. Con OtherFields, debe especificar los campos que se van a usar (consulte a continuación). Con Flags, los valores son marcas numéricas.
value_fields	[field field field]	Es necesario en caso de que value_mode sea OtherFields. Especifica los campos que se van a usar como campos de valores.

## Propiedades del nodo rfmanalysis



El nodo Análisis de RFM (actualidad, frecuencia, monetario) permite determinar cuantitativamente qué clientes son los mejores examinando cuándo ha sido la compra más reciente de un cliente (actualidad), cuántas veces suele comprar (frecuencia) y cuánto gasta el cliente en todas las transacciones (valor monetario).

Tabla 68. Propiedades del nodo rfmanalysis.

Propiedades del nodo rfmanalysis	Tipo de datos	Descripción de la propiedad
recency	<i>field</i>	Especifica el campo de actualidad. Puede ser una fecha, marca de tiempo o un número simple.
frequency	<i>field</i>	Especifica el campo de frecuencia.
monetary	<i>field</i>	Especifica el campo de monetario.
recency_bins	<i>entero</i>	Especifica el número de intervalos de actividades recientes que se van a generar.
recency_weight	<i>número</i>	Especifica la ponderación que se aplicará a los datos de actividades recientes. El valor predeterminado es 100.
frequency_bins	<i>entero</i>	Especifica el número de intervalos de frecuencia que se van a generar.
frequency_weight	<i>número</i>	Especifica la ponderación que se aplicará a los datos de frecuencia. El valor por omisión es 10.
monetary_bins	<i>entero</i>	Especifica el número de intervalos de monetario que se van a generar.
monetary_weight	<i>número</i>	Especifica la ponderación que se aplicará a los datos de monetario. El valor predeterminado es 1.
tied_values_method	Next Actual	Especifica en qué intervalo se van a insertar los datos de valor empatado.
recalculate_bins	Always IfNecessary	
add_outliers	<i>booleano</i>	Sólo está disponible si recalculate_bins se define como IfNecessary. Si se selecciona, los registros por debajo del intervalo más inferior se añaden al intervalo inferior y los registros por encima, se añaden al intervalo superior.
binned_field	Recency Frequency Monetary	
recency_thresholds	<i>valor valor</i>	Sólo está disponible si recalculate_bins se define como Siempre. Especifica los umbrales superior e inferior de los intervalos de actividades recientes. El umbral superior de un intervalo se utiliza como el umbral inferior del siguiente, por ejemplo, [10 30 60] definiría dos intervalos, el primer intervalo con los umbrales superior e inferior de 10 y 30, con los umbrales del segundo intervalo de 30 y 60.

Tabla 68. Propiedades del nodo rfanalysis (continuación).

Propiedades del nodo rfanalysis	Tipo de datos	Descripción de la propiedad
frequency_thresholds	valor valor	Sólo está disponible si recalculate_bins se define como Siempre.
monetary_thresholds	valor valor	Sólo está disponible si recalculate_bins se define como Siempre.

## Propiedades del nodo settoflag



El nodo Marcas deriva varios campos de marcas en función de los valores categóricos definidos para uno o más campos nominales.

Tabla 69. Propiedades del nodo settoflag.

Propiedades del nodo settoflag	Tipo de datos	Descripción de la propiedad
fields_from	[category category category] all	
true_value	cadena	Especifica el valor para verdadero utilizado por el nodo al configurar una marca. El valor predeterminado es T (del inglés 'True').
false_value	cadena	Especifica el valor para falso utilizado por el nodo al configurar una marca. El valor predeterminado es F (del inglés 'False').
use_extension	booleano	Utiliza una extensión como sufijo o prefijo para el nuevo campo de marca.
extension	cadena	
add_as	Suffix Prefix	Especifica si la extensión es un sufijo o un prefijo.
aggregate	booleano	Agrupar registros en función de campos clave. Si algún registro se establece como verdadero, se activarán todos los campos de marca de un grupo.
keys	[field field field]	Campos clave.

## Propiedades del nodo statisticstransform



El nodo Transformación Statistics ejecuta una selección de comandos de sintaxis de IBM SPSS Statistics en los orígenes de datos de IBM SPSS Modeler. Este nodo requiere una copia de IBM SPSS Statistics con licencia.

Las propiedades de este nodo están descritas en "Propiedades del nodo statisticstransform" en la página 239.



## Propiedades del nodo timeintervals



El nodo Intervalos de tiempo especifica los intervalos y genera etiquetas (si es necesario) para modelar los datos de series temporales. Si los valores no están espaciados de manera uniforme, el nodo puede rellenar o agregar valores según sea necesario para crear un intervalo uniforme entre registros.

Tabla 70. Propiedades del nodo timeintervals.

Propiedades del nodo timeintervals	Tipo de datos	Descripción de la propiedad
interval_type	None Períodos CyclicPeriods Años Trimestres Meses DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
mode	Label Create	Determina si desea etiquetar los registros de manera consecutiva o crear la serie según una fecha, marca de tiempo o campo de tiempo concretos.
campo	<i>field</i>	Al crear la serie a partir de los datos, especifica el campo que informa de la fecha u hora de cada registro.
period_start	<i>entero</i>	Especifica el intervalo de inicio para períodos o períodos cíclicos.
cycle_start	<i>entero</i>	Ciclo de inicio de los períodos cíclicos.
year_start	<i>entero</i>	En el caso de los tipos de intervalo que procedan, el año en que el primer intervalo tiene lugar.
quarter_start	<i>entero</i>	En el caso de los tipos de intervalo que procedan, el trimestre en que el primer intervalo tiene lugar.
month_start	Enero Febrero Marzo Abril Mayo Junio Julio Agosto Septiembre Octubre Noviembre Diciembre	
day_start	<i>entero</i>	
hour_start	<i>entero</i>	
minute_start	<i>entero</i>	

Tabla 70. Propiedades del nodo *timeintervals* (continuación).

Propiedades del nodo <i>timeintervals</i>	Tipo de datos	Descripción de la propiedad
<i>second_start</i>	<i>entero</i>	
<i>periods_per_cycle</i>	<i>entero</i>	En el caso de los períodos cíclicos, número en cada ciclo.
<i>fiscal_year_begins</i>	Enero Febrero Marzo Abril Mayo Junio Julio Agosto Septiembre Octubre Noviembre Diciembre	En el caso de los intervalos trimestrales, especifica el mes en el que comienza el año fiscal.
<i>week_begins_on</i>	Sunday Monday Tuesday Wednesday Thursday Friday Saturday Sunday	En el caso de los intervalos periódicos (días a la semana, horas al día, minutos al día y segundos al día), especifica el día en el que comienza la semana.
<i>day_begins_hour</i>	<i>entero</i>	En el caso de los intervalos periódicos (horas al día, minutos al día y segundos al día), especifica la hora a la que comienza el día. Se puede usar junto con <i>day_begins_minute</i> y <i>day_begins_second</i> para determinar una hora exacta, como 8:05:01. Vea el ejemplo de uso que se muestra a continuación.
<i>day_begins_minute</i>	<i>entero</i>	En el caso de los intervalos periódicos (horas al día, minutos al día y segundos al día), especifica el minuto en el que comienza el día (por ejemplo, 5 en 8:05).
<i>day_begins_second</i>	<i>entero</i>	En el caso de los intervalos periódicos (horas al día, minutos al día y segundos al día), especifica el segundo en el que comienza el día (por ejemplo, 17 en 8:05:17).
<i>days_per_week</i>	<i>entero</i>	En el caso de los intervalos periódicos (días a la semana, horas al día, minutos al día y segundos al día), especifica el número de días a la semana.
<i>hours_per_day</i>	<i>entero</i>	En el caso de los intervalos periódicos (horas al día, minutos al día y segundos al día), especifica el número de horas al día.

Tabla 70. Propiedades del nodo *timeintervals* (continuación).

Propiedades del nodo <i>timeintervals</i>	Tipo de datos	Descripción de la propiedad
<i>interval_increment</i>	1 2 3 4 5 6 10 15 20 30	En el caso de los minutos al día y de los segundos al día, especifica el número de minutos o segundos que se va a aumentar en cada registro.
<i>field_name_extension</i>	<i>cadena</i>	
<i>field_name_extension_as_prefix</i>	<i>booleano</i>	
<i>date_format</i>	"DDMMAA" "MDDYY" "AAMMDD" "YYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-AAAA" "DD-MES-YY" "DD-MES-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.AAAA" "MM.DD.YY" "MM.DD.YYYY" "DD.MES.YY" "DD.MES.YYYY" "DD/MM/YY" "DD/MM/AAAA" "MM/DD/YY" "MM/DD/YYYY" "DD/MES/YY" "DD/MES/YYYY" MON YYYY q Q YYYY ww WK YYYY	
<i>time_format</i>	"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"	

Tabla 70. Propiedades del nodo *timeintervals* (continuación).

Propiedades del nodo <i>timeintervals</i>	Tipo de datos	Descripción de la propiedad
<i>aggregate</i>	Media Sum Mode Mín Máx Primero Last TrueIfAnyTrue	Especifica el método de agregación para un campo.
<i>pad</i>	Vacío MeanOfRecentPoints True False	Especifica el método de relleno para un campo.
<i>agg_mode</i>	All Especifica	Determina si se agregan o rellenan todos los campos con las funciones predeterminadas según sea necesario o bien si se especifican los campos y las funciones que deben usarse.
<i>agg_range_default</i>	Media Sum Mode Mín Máx	Especifica la función predeterminada que se va a usar al agregar campos continuos.
<i>agg_set_default</i>	Mode Primero Last	Especifica la función predeterminada que se va a usar al agregar campos nominales.
<i>agg_flag_default</i>	TrueIfAnyTrue Mode Primero Last	
<i>pad_range_default</i>	Vacío MeanOfRecentPoints	Especifica la función predeterminada que se va a usar al agregar campos continuos.
<i>pad_set_default</i>	Vacío MostRecentValue	
<i>pad_flag_default</i>	Vacío True False	
<i>max_records_to_create</i>	<i>entero</i>	Especifica el número máximo de registros que se van a crear al rellenar la serie.
<i>estimation_from_beginning</i>	<i>booleano</i>	
<i>estimation_to_end</i>	<i>booleano</i>	
<i>estimation_start_offset</i>	<i>entero</i>	
<i>estimation_num_holdouts</i>	<i>entero</i>	
<i>create_future_records</i>	<i>booleano</i>	
<i>num_future_records</i>	<i>entero</i>	
<i>create_future_field</i>	<i>booleano</i>	
<i>future_field_name</i>	<i>cadena</i>	

## Propiedades del nodo transpose



El nodo Transponer intercambia los datos en filas y columnas de manera que los registros se conviertan en campos y los campos en registros.

Tabla 71. Propiedades del nodo transpose.

Propiedades del nodo transpose	Tipo de datos	Descripción de la propiedad
transposed_names	Prefix Leer	Se pueden generar nuevos nombres de campo automáticamente a partir de un prefijo concreto o bien se pueden leer desde un campo existente en los datos.
prefix	<i>cadena</i>	
num_new_fields	<i>entero</i>	Al usar un prefijo, especifica el número máximo de campos nuevos que se van a crear.
read_from_field	<i>field</i>	Campo del que se leen los nombres. Debe tratarse de un campo instanciado o, de lo contrario, se producirá un error al ejecutar el nodo.
max_num_fields	<i>entero</i>	Al leer nombres de un campo, especifica un límite superior con el fin de evitar que se genere un número desmesurado de campos.
transpose_type	Numérico String Custom	De forma predeterminada, solamente los campos continuos (rango numérico) se transponen, si bien se puede elegir un subconjunto personalizado de campos numéricos o, en su lugar, transponer todos los campos de cadena.
transpose_fields	[ <i>field field field</i> ]	Especifica los campos que se van a transponer cuando se usa la opción Custom.
id_field_name	<i>field</i>	

## Propiedades del nodo type



El nodo Tipo especifica propiedades y metadatos de campo. Por ejemplo, puede especificar un nivel de medición (continuo, nominal, ordinal o marca) para cada campo, establecer las opciones para gestionar valores perdidos y nulos del sistema, establecer el rol de un campo con fines de modelado, especificar las etiquetas de valor y campo y especificar los valores de un campo.

Observe que en algunos casos puede que sea necesario instanciar totalmente el nodo Tipo para que otros nodos funcionen adecuadamente, como, por ejemplo, la propiedad `fields from` del nodo Marcas. Puede conectar simplemente un nodo Tabla y ejecutarlo para crear instancias de los campos.

Tabla 72. Propiedades del nodo type.

Propiedades del nodo type	Tipo de datos	Descripción de la propiedad
dirección	Input Destino Both None Partition Split Frequency RecordID	Propiedad con clave para los roles de los campos. <i>Note:</i> Los valores In y Out ahora se han desaprobado. Deberán de ser compatibles en versiones posteriores.
type	Range Flag Set Sin tipo Discrete OrderedSet Predeterminado	El nivel de medición del campo (anteriormente denominado el "tipo" de campo). Si se establece type como Default, se borrará cualquier configuración del parámetro values y si value_mode tiene el valor Specify, se restablecerá a Read. Si value_mode se establece a Pass o Read, el establecimiento de type no afectará a value_mode. Nota: los tipos de datos utilizados internamente son distintos de los que se ven en el nodo tipo. La correspondencia es la siguiente: Rango -> Continuo Conjunto - > Nominal Conjunto ordenado (OrderedSet) -> Ordinal Discreto- > Categórico
storage	Desconocido Cadena Entero Real Hora Fecha Marca de tiempo	Propiedad con clave de solamente lectura para el tipo de almacenamiento de campos.
check	Ninguno Nullify Coerce Discard Warn Abort	Propiedad con clave para la comprobación del rango y el tipo de campo.
values	[value value]	Para un campo continuos, el primer valor es el mínimo y el último es el máximo. Para campos nominales, especifique todos los valores. Para los campos marca, el primer valor representa <i>falso</i> y el último, <i>verdadero</i> . La configuración de esta propiedad establece de forma automática la propiedad value_mode en Specify.
value_mode	Leer Pasar Leer+ Actual Especifica	Determina la forma en la que se establecen los valores. Tenga en cuenta que no puede establecer esta propiedad directamente en Specify. Para utilizar valores específicos, establezca la propiedad values.

Tabla 72. Propiedades del nodo type (continuación).

Propiedades del nodo type	Tipo de datos	Descripción de la propiedad
extend_values	booleano	Se aplica cuando value_mode se establece en Read. Establézcala en T para añadir nuevos valores de lectura a los valores existentes del campo. Establézcala en F para descartar los valores existentes y favorecer a los nuevos valores de lectura.
enable_missing	booleano	Cuando está definida como T, activa el seguimiento de los valores perdidos para el campo.
missing_values	[value value ...]	Especifica los valores de datos que denotan los datos perdidos.
range_missing	booleano	Especifica si se ha definido un rango de valores perdidos (vacíos) para un campo.
missing_lower	cadena	Si range_missing es verdadero, especifica el límite inferior del rango de valores perdidos.
missing_upper	cadena	Si range_missing es verdadero, especifica el límite superior del rango de valores perdidos.
null_missing	booleano	Cuando se establece en T, los valores <i>nulos</i> (valores no definidos que se muestran como \$null\$ en el software) se consideran valores perdidos.
whitespace_missing	booleano	Cuando se establece en T, los valores que sólo contienen un espacio en blanco (espacios, tabulaciones y líneas nuevas) se consideran valores perdidos.
description	cadena	Especifica la descripción de un campo.
value_labels	[[Valor CadenaEtiquetas] { Valor CadenaEtiquetas} ...]	Se utiliza para especificar etiquetas para los pares de valores.
display_places	entero	Establece el número de cifras decimales para el campo cuando se muestra (sólo se aplica a campos con almacenamiento REAL). Un valor de 1 utilizará el valor predeterminado de la ruta.
export_places	entero	Establece el número de cifras decimales para el campo cuando se exporta (sólo se aplica a campos con almacenamiento REAL). Un valor de 1 utilizará el valor predeterminado de la ruta.
decimal_separator	DEFAULT PERIOD COMMA	Establece el separador decimal para el campo (sólo se aplica a campos con almacenamiento REAL).

Tabla 72. Propiedades del nodo type (continuación).

Propiedades del nodo type	Tipo de datos	Descripción de la propiedad
date_format	"DDMAA" "MDDYY" "AAMDD" "YMMDD" "YYYYDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-AAAA" "DD-MES-YY" "DD-MES-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.AAAA" "MM.DD.YY" "MM.DD.YYYY" "DD.MES.YY" "DD.MES.YYYY" "DD/MM/YY" "DD/MM/AAAA" "MM/DD/YY" "MM/DD/YYYY" "DD/MES/YY" "DD/MES/YYYY" MON YYYY q Q YYYY ww WK YYYY	Establece el formato de fecha para el campo (sólo se aplica a campos con almacenamiento FECHA o MARCADETIEMPO).
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"	Establece el formato de hora para el campo (sólo se aplica a campos con almacenamiento TIME o TIMESTAMP).
number_format	DEFAULT STANDARD SCIENTIFIC CURRENCY	Establece el formato de presentación de los números para el campo.
standard_places	entero	Establece el número de cifras decimales para el campo cuando se muestra en formato estándar. Un valor de 1 utilizará el valor predeterminado de la ruta. Tenga en cuenta que el intervalo display_places existente también cambiará esto, pero ahora se ha desaprobadado.
scientific_places	entero	Establece el número de cifras decimales para el campo cuando se muestra en formato científico. Un valor de 1 utilizará el valor predeterminado de la ruta.



Tabla 72. Propiedades del nodo type (continuación).

Propiedades del nodo type	Tipo de datos	Descripción de la propiedad
currency_places	entero	Establece el número de cifras decimales para el campo cuando se muestra en formato moneda. Un valor de 1 utilizará el valor predeterminado de la ruta.
grouping_symbol	DEFAULT NONE LOCALE PERIOD COMMA SPACE	Establece el símbolo de agrupación para el campo.
column_width	entero	Establece el ancho de columna para el campo. Un valor de 1 establecerá el ancho de columna en Auto.
justify	AUTO CENTER LEFT RIGHT	Establece la justificación de columna para el campo.



---

## Capítulo 12. Propiedades de nodos Gráfico

---

### Propiedades comunes de nodos Gráfico

Esta sección describe las propiedades disponibles para los nodos de gráficos, incluidas las comunes y aquellas específicas de cada tipo de nodo.

Tabla 73. Propiedades comunes de nodos de gráficos.

Propiedades comunes de nodos de gráficos	Tipo de datos	Descripción de la propiedad
título	<i>cadena</i>	Especifica el título. Ejemplo: "Esto es un título".
caption	<i>cadena</i>	Especifica el pie. Por ejemplo: "Esto es un pie".
output_mode	Screen File	Determina si el resultado del nodo de gráficos se muestra o si se guarda en un archivo.
output_format	BMP JPEG PNG HTML output (.cou)	Especifica el tipo de resultado. El tipo exacto de resultado permitido para cada nodo varía.
full_filename	<i>cadena</i>	Especifica la ruta de destino y el nombre de archivo del resultado generado por el nodo de gráficos.
use_graph_size	<i>booleano</i>	Controla si el tamaño del gráfico se ha establecido de manera explícita mediante las propiedades de ancho y altura a continuación. Afecta solamente a los gráficos que tienen salida a pantalla. No disponible para el nodo Distribución.
graph_width	<i>número</i>	Cuando use_graph_size es True, establece el ancho del gráfico en píxeles.
graph_height	<i>número</i>	Cuando use_graph_size es True, establece la altura del gráfico en píxeles.

#### Notas

**Desactivación de los campos opcionales.** Los campos opcionales, como un campo de superposición para gráficas, se pueden desactivar estableciendo el valor de la propiedad a " " (cadena vacía).

**Especificación de los colores.** Los colores de los títulos, pies, fondos y etiquetas se pueden especificar mediante las cadenas hexadecimales que comiencen con el símbolo almohadilla (#).

Los dos primeros dígitos especifican el contenido rojo, los dos dígitos en el medio, especifican el contenido verde y los dos últimos dígitos especifican el contenido azul. Cada dígito puede tomar un valor del rango 0-9 o A-F. Juntos, estos valores pueden especificar red-green-blue, o RGB o color.

*Note:* al especificar colores en RVA, puede utilizar el selector de campos en la interfaz de usuario para determinar el código de color correcto. Basta con colocarse sobre el color para ver un texto con la información deseada.

## Propiedades del nodo collection



El nodo Colección muestra la distribución de valores de un campo numérico relativo a los valores de otro. (Crea gráficos parecidos a los histogramas.) Es útil para ilustrar una variable o un campo cuyos valores cambian con el tiempo. Con los gráficos 3D también puede incluir un eje simbólico que muestra las distribuciones por categoría.

Tabla 74. Propiedades del nodo collection.

Propiedades del nodo collection	Tipo de datos	Descripción de la propiedad
over_field	field	
over_label_auto	booleano	
over_label	cadena	
collect_field	field	
collect_label_auto	booleano	
collect_label	cadena	
three_D	booleano	
by_field	field	
by_label_auto	booleano	
by_label	cadena	
operation	Sum Media Mín Máx SDev	
color_field	cadena	
panel_field	cadena	
animation_field	cadena	
range_mode	Automatic UserDefined	
range_min	número	
range_max	número	
bins	ByNumber ByWidth	
num_bins	número	
bin_width	número	
use_grid	booleano	
graph_background	color	Al principio de esta sección se describen los colores de gráficos estándar.
page_background	color	Al principio de esta sección se describen los colores de gráficos estándar.

## Propiedades del nodo distribution



El nodo Distribución muestra las instancias de valores simbólicos (categóricos), como el tipo de hipoteca o el género. Normalmente, podría usar el nodo Distribución para mostrar los desequilibrios de los datos, que pueden rectificarse mediante el nodo Equilibrar antes de crear un modelo.

Tabla 75. Propiedades del nodo distribution.

Propiedades del nodo distribution	Tipo de datos	Descripción de la propiedad
plot	SelectedFields Flags	
x_field	field	
color_field	field	Campo de superposición.
normalize	booleano	
sort_mode	ByOccurrence Alphabetic	
use_proportional_scale	booleano	

## Propiedades del nodo evaluation



El nodo Evaluación ayuda a evaluar y comparar modelos predictivos. El diagrama de evaluación muestra la calidad con que los modelos predicen resultados particulares. Ordena registros en función del valor predicho y la confianza de la predicción. Divide el registro en grupos de igual tamaño (**cuantiles**) y, a continuación, representa el valor del criterio de negocio de cada cuantil de mayor a menor. El gráfico muestra múltiples modelos como líneas independientes.

Tabla 76. Propiedades del nodo evaluation.

Propiedades del nodo evaluation	Tipo de datos	Descripción de la propiedad
chart_type	Gains Response Lift Profit ROI ROC	
inc_baseline	booleano	
field_detection_method	Metadata Name	
use_fixed_cost	booleano	
cost_value	número	
cost_field	cadena	
use_fixed_revenue	booleano	
revenue_value	número	
revenue_field	cadena	
use_fixed_weight	booleano	
weight_value	número	
weight_field	field	
n_tile	Quartiles Quintiles Deciles Vingtiles Percentiles 1000-tiles	
cumulative	flag	

Tabla 76. Propiedades del nodo evaluation (continuación).

Propiedades del nodo evaluation	Tipo de datos	Descripción de la propiedad
style	Line Point	
point_type	Rectángulo Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
export_data	booleano	
data_filename	cadena	
delimiter	cadena	
new_line	booleano	
inc_field_names	booleano	
inc_best_line	booleano	
inc_business_rule	booleano	
business_rule_condition	cadena	
plot_score_fields	booleano	
score_fields	[campo1 ... campoN]	
target_field	field	
use_hit_condition	booleano	
hit_condition	cadena	
use_score_expression	booleano	
score_expression	cadena	
caption_auto	booleano	

## Propiedades del nodo graphboard



El nodo Tablero ofrece muchos tipos diferentes de gráficos en un único nodo. Con este nodo puede seleccionar los campos de datos que desee explorar y seleccionar un gráfico de los disponibles para los datos seleccionados. El nodo filtra automáticamente cualquier tipo de gráfico que no sea compatible con las selecciones de campo.

*Note:* Si establece una propiedad que no es válida para el tipo de gráfico (por ejemplo, si especifica `y_field` para un histograma), se ignora dicha propiedad.

Tabla 77. Propiedades del nodo graphboard.

Propiedades del nodo graphboard	Tipo de datos	Descripción de la propiedad
graph_type	2DDotplot 3DArea 3DBar 3DDensity 3DHistogram 3DPie 3DScatterplot Área ArrowMap Bar BarCounts BarCountsMap BarMap BinnedScatter Diagramas de caja Bubble ChoroplethMeans ChoroplethMedians ChoroplethSums ChoroplethValues ChoroplethCounts CoordinateMap CoordinateChoroplethMeans CoordinateChoroplethMedians CoordinateChoroplethSums CoordinateChoroplethValues CoordinateChoroplethCounts Dotplot Heatmap HexBinScatter Histograma Line LineChartMap LineOverlayMap Parallel Path Pie PieCountMap PieCounts PieMap PointOverlayMap PolygonOverlayMap Ribbon Scatterplot SPL0M Surface	Identifica el tipo de gráfico.
x_field	<i>field</i>	Especifica una etiqueta personalizada para el eje x. Disponible solamente para etiquetas.
y_field	<i>field</i>	Especifica una etiqueta personalizada para el eje y. Disponible solamente para etiquetas.
z_field	<i>field</i>	Se utiliza en algunos gráficos 3D.
color_field	<i>field</i>	Se utiliza en mapas de calor.

Tabla 77. Propiedades del nodo graphboard (continuación).

Propiedades del nodo graphboard	Tipo de datos	Descripción de la propiedad
size_field	field	Se utiliza en gráficos de burbujas.
categories_field	field	
values_field	field	
rows_field	field	
columns_field	field	
fields	field	
start_longitude_field	field	Se utiliza con flechas en un mapa de referencia.
end_longitude_field	field	
start_latitude_field	field	
end_latitude_field	field	
data_key_field	field	Se utiliza en diversos mapas.
panelrow_field	cadena	
panelcol_field	cadena	
animation_field	cadena	
longitude_field	field	Se utiliza en mapas de coordenadas.
latitude_field	field	
map_color_field	field	

## Propiedades del nodo histogram



El nodo Histograma muestra las instancias de valores de los campos numéricos. Se suele utilizar para explorar los datos antes de las manipulaciones y la generación de modelos. Al igual que con el nodo Distribución, con frecuencia el nodo Histograma detecta desequilibrios en los datos.

Tabla 78. Propiedades del nodo histogram.

Propiedades del nodo histogram	Tipo de datos	Descripción de la propiedad
campo	field	
color_field	field	
panel_field	field	
animation_field	field	
range_mode	Automatic UserDefined	
range_min	número	
range_max	número	
bins	ByNumber ByWidth	
num_bins	número	
bin_width	número	
normalize	booleano	
separate_bands	booleano	



Tabla 78. Propiedades del nodo histogram (continuación).

Propiedades del nodo histogram	Tipo de datos	Descripción de la propiedad
x_label_auto	booleano	
x_label	cadena	
y_label_auto	booleano	
y_label	cadena	
use_grid	booleano	
graph_background	color	Al principio de esta sección se describen los colores de gráficos estándar.
page_background	color	Al principio de esta sección se describen los colores de gráficos estándar.
normal_curve	booleano	Indica si se debe mostrar la curva de distribución normal en los resultados.

## Propiedades del nodo multiplot



El nodo G. múltiple crea un gráfico que muestra varios campos Y sobre un campo X único. Los campos Y están representados como líneas coloreadas; cada uno equivale a un nodo Gráfico con el estilo establecido en **Línea** y el Modo para X establecido en **Ordenar**. Los gráficos múltiples son útiles cuando quiere explorar la fluctuación de varias variables a través del tiempo.

Tabla 79. Propiedades del nodo multiplot.

Propiedades del nodo multiplot	Tipo de datos	Descripción de la propiedad
x_field	field	
y_fields	[field field field]	
panel_field	field	
animation_field	field	
normalize	booleano	
use_overlay_expr	booleano	
overlay_expression	cadena	
records_limit	número	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	booleano	
x_label	cadena	
y_label_auto	booleano	
y_label	cadena	
use_grid	booleano	
graph_background	color	Al principio de esta sección se describen los colores de gráficos estándar.
page_background	color	Al principio de esta sección se describen los colores de gráficos estándar.

## Propiedades del nodo plot



El nodo Gráfico muestra la relación entre los campos numéricos. Puede crear un gráfico mediante puntos (un diagrama de dispersión) o líneas.

Tabla 80. Propiedades del nodo plot.

Propiedades del nodo plot	Tipo de datos	Descripción de la propiedad
x_field	field	Especifica una etiqueta personalizada para el eje x. Disponible solamente para etiquetas.
y_field	field	Especifica una etiqueta personalizada para el eje y. Disponible solamente para etiquetas.
three_D	booleano	Especifica una etiqueta personalizada para el eje y. Disponible sólo para etiquetas en gráficos 3D.
z_field	field	
color_field	field	Campo de superposición.
size_field	field	
shape_field	field	
panel_field	field	Especifica un campo de marcas o nominal para crear un gráfico independiente para cada categoría. Los gráficos aparecerán juntos en una ventana de resultados.
animation_field	field	Especifica un campo de marcas o nominal para ilustrar las categorías de los valores de datos creando una serie de gráficos secuenciados mediante la animación.
transp_field	field	Especifica un campo para ilustrar las categorías de los valores de datos utilizando un nivel de transparencia distinto para cada categoría. No disponible para gráficos de líneas.
overlay_type	None Smoother Función	Determina si se muestra una función superpuesta o suavizamiento LOESS.
overlay_expression	cadena	Especifica la expresión utilizada cuando overlay_type se establece en Function.
style	Point Line	

Tabla 80. Propiedades del nodo plot (continuación).

Propiedades del nodo plot	Tipo de datos	Descripción de la propiedad
point_type	Rectángulo Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
x_mode	Ordenar Overlay AsRead	
x_range_mode	Automatic UserDefined	
x_range_min	número	
x_range_max	número	
y_range_mode	Automatic UserDefined	
y_range_min	número	
y_range_max	número	
z_range_mode	Automatic UserDefined	
z_range_min	número	
z_range_max	número	
jitter	booleano	
records_limit	número	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	booleano	
x_label	cadena	
y_label_auto	booleano	
y_label	cadena	
z_label_auto	booleano	
z_label	cadena	
use_grid	booleano	
graph_background	color	Al principio de esta sección se describen los colores de gráficos estándar.

Tabla 80. Propiedades del nodo plot (continuación).

Propiedades del nodo plot	Tipo de datos	Descripción de la propiedad
page_background	color	Al principio de esta sección se describen los colores de gráficos estándar.
use_overlay_expr	booleano	Desaprobado en favor de overlay_type.

## Propiedades del nodo timeplot



El nodo Gráfico de tiempo muestra uno o más conjuntos de datos de series temporales. Normalmente, primero se utilizaría un nodo Intervalos de tiempo para crear un campo *EtiquetaTiempo*, que se utilizaría para etiquetar el eje *x*.

Tabla 81. Propiedades del nodo timeplot.

Propiedades del nodo timeplot	Tipo de datos	Descripción de la propiedad
plot_series	Series Models	
use_custom_x_field	booleano	
x_field	field	
y_fields	[field field field]	
panel	booleano	
normalize	booleano	
line	booleano	
points	booleano	
point_type	Rectángulo Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
suavizamiento	booleano	Puede añadir suavizamientos al gráfico únicamente si establece panel en True.
use_records_limit	booleano	
records_limit	entero	
symbol_size	número	Especifica el tamaño del símbolo.

Tabla 81. Propiedades del nodo timeplot (continuación).

Propiedades del nodo timeplot	Tipo de datos	Descripción de la propiedad
panel_layout	Horizontal Vertical	

## Propiedades del nodo web



El nodo Malla muestra la fuerza de las relaciones entre los valores de dos o más campos simbólicos (categóricos). El gráfico utiliza líneas de varios anchos para indicar la fuerza de la conexión. Podría utilizar un nodo Malla, por ejemplo, para explorar las relaciones existentes entre la compra de un conjunto de elementos en un sitio de comercio electrónico.

Tabla 82. propiedades del nodo web.

Propiedades del nodo web	Tipo de datos	Descripción de la propiedad
use_directed_web	booleano	
fields	[field field field]	
to_field	field	
from_fields	[field field field]	
true_flags_only	booleano	
line_values	Absolute OverallPct PctLarger PctSmaller	
strong_links_heavier	booleano	
num_links	ShowMaximum ShowLinksAbove ShowAll	
max_num_links	número	
links_above	número	
discard_links_min	booleano	
links_min_records	número	
discard_links_max	booleano	
links_max_records	número	
weak_below	número	
strong_above	número	
link_size_continuous	booleano	
web_display	Circular Red Directed Grid	
graph_background	color	Al principio de esta sección se describen los colores de gráficos estándar.
symbol_size	número	Especifica el tamaño del símbolo.



---

## Capítulo 13. Propiedades de nodos de modelado

---

### Propiedades comunes de nodos de modelado

Las siguientes propiedades son comunes a algunos o todos los nodos de modelado. Las excepciones se indican en la documentación de los nodos de modelado individuales según sea adecuado.

Tabla 83. Propiedades comunes de nodos de modelado.

Propiedad	Valores	Descripción de la propiedad
custom_fields	booleano	Si es verdadero, le permite especificar el objetivo, la entrada y otros campos del nodo actual. Si es falso, se utiliza la configuración actual de un nodo Tipo situado en un punto anterior de la ruta.
target o targets	field o [field1 ... fieldN]	Especifica un único campo objetivo o varios campos objetivo dependiendo del tipo de modelo.
inputs	[field1 ... fieldN]	Campos de entrada o predictor utilizados por el modelo.
partición	field	
use_partitioned_data	booleano	Si se ha definido un campo de partición, esta opción garantiza que sólo se utilizarán los datos de la partición de entrenamiento para la generación del modelo.
use_split_data	booleano	
splits	[campo1 ... campoN]	Especifica el campo o campos para utilizar en el modelado de divisiones. Sólo funciona si use_split_data está establecido como True.
use_frequency	booleano	Los modelos específicos utilizan campos de ponderación y frecuencia como se indica en cada tipo de modelo.
frequency_field	field	
use_weight	booleano	
weight_field	field	
use_model_name	booleano	
model_name	cadena	Nombre personalizado para nuevo modelo.
mode	Simple Expert	

---

### Propiedades del nodo anomalydetection



El nodo Detección de anomalías identifica casos extraños, o valores atípicos, que no se ajustan a patrones de datos “normales”. Con este nodo, es posible identificar valores atípicos aunque no se ajusten a ningún patrón previamente conocido o no se realice una búsqueda exacta.

Tabla 84. Propiedades del nodo anomalydetection.

Propiedades del nodo anomalydetection	Valores	Descripción de la propiedad
inputs	[campo1 ... campoN]	Los modelos de detección de anomalías criban registros basándose en los campos de entrada especificados. No utilizan un campo objetivo. Los campos de ponderación y frecuencia tampoco se usan. Consulte el tema “Propiedades comunes de nodos de modelado” en la página 129 para obtener más información.
mode	Expert Simple	
anomaly_method	IndexLevel PerRecords NumRecords	Especifica el método utilizado para determinar el valor de corte para marcar los registros como anómalos.
index_level	número	Especifica el valor de corte mínimo con el que se van a marcar anomalías.
percent_records	número	Establece el umbral para marcar registros de acuerdo al porcentaje de registros en los datos de entrenamiento.
num_records	número	Establece el umbral para marcar registros de acuerdo al número de registros en los datos de entrenamiento.
num_fields	entero	El número de campos de los que se va a informar por cada registro anómalo.
impute_missing_values	booleano	
adjustment_coeff	número	Valor que se usa para equilibrar la ponderación relativa asignada a los campos categóricos y continuos al calcular la distancia.
peer_group_num_auto	booleano	Calcula automáticamente el número de grupos de homólogos.
min_num_peer_groups	entero	Especifica el número mínimo de grupos de homólogos empleado cuando peer_group_num_auto se establece en True.
max_num_per_groups	entero	Especifica el número máximo de grupos de homólogos.
num_peer_groups	entero	Especifica el número de grupos de homólogos empleado cuando peer_group_num_auto se establece en False.
noise_level	número	Determina el modo en que se tratan los valores atípicos durante el clúster. Especifique un valor entre 0 y 0,5.
noise_ratio	número	Especifica la parte de memoria asignada al componente que debe usarse para el almacenamiento en búfer de ruido. Especifique un valor entre 0 y 0,5.



## Propiedades del nodo apriori



El nodo Apriori extrae un conjunto de reglas de los datos y destaca aquellas reglas con un mayor contenido de información. Apriori ofrece cinco métodos diferentes para la selección de reglas y utiliza un sofisticado esquema de indización para procesar eficientemente grandes conjuntos de datos. En los problemas de mucho volumen, Apriori se entrena más rápidamente, no tiene un límite arbitrario para el número de reglas que puede retener y puede gestionar reglas que tengan hasta 32 precondiciones. Apriori requiere que todos los campos de entrada y salida sean categóricos, pero ofrece un mejor rendimiento ya que está optimizado para este tipo de datos.

Tabla 85. Propiedades del nodo apriori.

Propiedades del nodo apriori	Valores	Descripción de la propiedad
consequents	<i>field</i>	Los modelos apriori utilizan Consecuentes y Antecedentes en lugar de los campos estándar objetivo y de entrada. Los campos de ponderación y frecuencia no se usan. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
antecedents	[ <i>field1 ... fieldN</i> ]	
min_supp	<i>número</i>	
min_conf	<i>número</i>	
max_antecedents	<i>número</i>	
true_flags	<i>booleano</i>	
optimize	Speed Memory	
use_transactional_data	<i>booleano</i>	
contiguous	<i>booleano</i>	
id_field	<i>cadena</i>	
content_field	<i>cadena</i>	
mode	Simple Expert	
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	
lower_bound	<i>número</i>	
optimize	Speed Memory	Se utiliza para especificar si la generación del modelo se debe optimizar para la velocidad o la memoria.

## Propiedades del nodo autoclassifier



El nodo Clasificador automático crea y compara varios modelos diferentes para obtener resultados binarios (sí o no, abandono o no de clientes, etc.), lo que le permite seleccionar el mejor enfoque para un análisis determinado. Son compatibles varios algoritmos de modelado, por lo que es posible seleccionar los métodos que desee utilizar, las opciones específicas para cada uno y los criterios para comparar los resultados. El nodo genera un conjunto de modelos basado en las opciones especificadas y clasifica los mejores candidatos en función de los criterios que especifique.

Tabla 86. Propiedades del nodo autoclassifier.

Propiedades del nodo autoclassifier	Valores	Descripción de la propiedad
target	field	En objetivos de marca, el nodo Clasificador binario requiere un único campo objetivo y uno o más campos de entrada. También se puede especificar campos de ponderación y frecuencia. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
ranking_measure	Accuracy Area_under_curve Profit Lift Num_variables	
ranking_dataset	Training Test	
number_of_models	entero	Número de modelos que se incluirán en el nugget de modelo. Especifique un entero entre 1 y 100.
calculate_variable_importance	booleano	
enable_accuracy_limit	booleano	
accuracy_limit	entero	Entero entre 0 y 100.
enable_area_under_curve_limit	booleano	
area_under_curve_limit	número	Número real entre 0,0 y 1,0.
enable_profit_limit	booleano	
profit_limit	número	Entero mayor que 0.
enable_lift_limit	booleano	
lift_limit	número	Número real mayor que 1,0.
enable_number_of_variables_limit	booleano	
number_of_variables_limit	número	Entero mayor que 0.
use_fixed_cost	booleano	
fixed_cost	número	Número real mayor que 0.0.
variable_cost	field	
use_fixed_revenue	booleano	
fixed_revenue	número	Número real mayor que 0.0.
variable_revenue	field	
use_fixed_weight	booleano	

Tabla 86. Propiedades del nodo *autoclassifier* (continuación).

Propiedades del nodo <i>autoclassifier</i>	Valores	Descripción de la propiedad
<code>fixed_weight</code>	<i>número</i>	Número real mayor que 0,0
<code>variable_weight</code>	<i>field</i>	
<code>lift_percentile</code>	<i>número</i>	Entero entre 0 y 100.
<code>enable_model_build_time_limit</code>	<i>booleano</i>	
<code>model_build_time_limit</code>	<i>número</i>	Entero que indica el número máximo de minutos que se puede tardar en generar cada uno de los modelos.
<code>enable_stop_after_time_limit</code>	<i>booleano</i>	
<code>stop_after_time_limit</code>	<i>número</i>	Número real que indica el número máximo de horas que puede tardar una ejecución del clasificador automático.
<code>enable_stop_after_valid_model_production</code>	<i>booleano</i>	
<code>use_costs</code>	<i>booleano</i>	
<code>&lt;algorithm&gt;</code>	<i>booleano</i>	Activa o desactiva el uso de un determinado algoritmo.
<code>&lt;algorithm&gt;.&lt;property&gt;</code>	<i>cadena</i>	Define un valor de propiedad para un algoritmo específico. Consulte el tema "Propiedades de ajustes de algoritmo" para obtener más información.

## Propiedades de ajustes de algoritmo

Los nombres de algoritmos del nodo Clasificador automático son `cart`, `chaid`, `quest`, `c50`, `logreg`, `decisionlist`, `bayesnet`, `discriminant`, `svm` y `knn`.

Los nombres de algoritmos del nodo Autonumérico son `cart`, `chaid`, `neuralnetwork`, `genlin`, `svm`, `regression`, `linear` y `knn`.

Los nombres de algoritmos del nodo Autoclúster son `twostep`, `k-means` y `kohonen`.

Los nombres de las propiedades son los nombres estándar, según se han documentado para cada nodo de algoritmo.

Las propiedades de algoritmos que contienen puntos u otros signos de puntuación deben encerrarse entre comillas simples.

También se pueden asignar varios valores a una propiedad.

*Notas:*

- se deben utilizar minúsculas al utilizar los valores `true` y `false` (en vez de `False`).
- En los casos en los que determinadas opciones de algoritmos no están disponibles en el nodo Clasificador automático o cuando sólo se puede especificar un único valor, en lugar de un intervalo de valores, se aplican los mismos límites que tienen los scripts cuando se accede al nodo de la manera estándar.

## Propiedades del nodo autocluster



El nodo Agrupación en clústeres automática calcula y compara los modelos de agrupación en clústeres que identifican grupos de registros con características similares. El nodo funciona de la misma manera que otros nodos de modelado automático, permitiéndole experimentar con múltiples combinaciones de opciones en una única pasada de modelado. Los modelos se pueden comparar utilizando medidas básicas con las que se intenta filtrar y definir la utilidad de los modelos de clúster y proporcionar una medida según la importancia de campos concretos.

Tabla 87. Propiedades del nodo autocluster.

Propiedades del nodo autocluster	Valores	Descripción de la propiedad
evaluation	field	Note: nodo Agrupación en clústeres automática únicamente. Identifica el campo cuyo valor de importancia se calculará. Como alternativa, se puede utilizar para identificar la calidad con la que el clúster diferencia el valor de este campo y, por lo tanto; la calidad con la que el modelo predecirá este campo.
ranking_measure	Silhouette Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Importance	
ranking_dataset	Training Test	
summary_limit	entero	Número de modelos que se incluirán en el informe. Especifique un entero entre 1 y 100.
enable_silhouette_limit	booleano	
silhouette_limit	entero	Entero entre 0 y 100.
enable_number_less_limit	booleano	
number_less_limit	número	Número real entre 0,0 y 1,0.
enable_number_greater_limit	booleano	
number_greater_limit	número	Entero mayor que 0.
enable_smallest_cluster_limit	booleano	
smallest_cluster_units	Percentage Counts	
smallest_cluster_limit_percentage	número	
smallest_cluster_limit_count	entero	Entero mayor que 0.
enable_largest_cluster_limit	booleano	
largest_cluster_units	Percentage Counts	
largest_cluster_limit_percentage	número	
largest_cluster_limit_count	entero	
enable_smallest_largest_limit	booleano	
smallest_largest_limit	número	

Tabla 87. Propiedades del nodo autocluster (continuación).

Propiedades del nodo autocluster	Valores	Descripción de la propiedad
enable_importance_limit	booleano	
importance_limit_condition	Greater_than Less_than	
importance_limit_greater_than	número	Entero entre 0 y 100.
importance_limit_less_than	número	Entero entre 0 y 100.
<algorithm>	booleano	Activa o desactiva el uso de un determinado algoritmo.
<algorithm>.<property>	cadena	Define un valor de propiedad para un algoritmo específico. Consulte el tema "Propiedades de ajustes de algoritmo" en la página 133 para obtener más información.

## Propiedades del nodo autonumeric



El nodo Autonumérico calcula y compara modelos para resultados de rango numérico continuo utilizando cierto número de métodos diferentes. El nodo funciona de la misma manera que el nodo Clasificador automático, lo que le permite seleccionar los algoritmos que desee utilizar y experimentar con varias combinaciones de opciones en una única pasada de modelado. Los algoritmos admitidos incluyen redes neuronales, C&RT, CHAID, regresión lineal, regresión lineal generalizada y máquinas de vectores de soporte (SVM). Los modelos se pueden comparar basándose en la correlación, el error relativo o el número de variables utilizado.

Tabla 88. Propiedades del nodo autonumeric.

Propiedades del nodo autonumeric	Valores	Descripción de la propiedad
custom_fields	booleano	Si es verdadero, se utilizará la configuración de campos personalizada en lugar de la configuración del nodo Tipo.
target	field	El nodo Autonumérico requiere un único campo objetivo y uno o más campos de entrada. También se puede especificar campos de ponderación y frecuencia. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
inputs	[campo1 ... campo2]	
partición	field	
use_frequency	booleano	
frequency_field	field	
use_weight	booleano	
weight_field	field	
use_partitioned_data	booleano	Si se ha definido un campo de partición, sólo se utilizarán los datos de entrenamiento para la generación del modelo.
ranking_measure	Correlation NumberOfFields	

Tabla 88. Propiedades del nodo autonumeric (continuación).

Propiedades del nodo autonumeric	Valores	Descripción de la propiedad
ranking_dataset	Test Training	
number_of_models	entero	Número de modelos que se incluirán en el nugget de modelo. Especifique un entero entre 1 y 100.
calculate_variable_importance	booleano	
enable_correlation_limit	booleano	
correlation_limit	entero	
enable_number_of_fields_limit	booleano	
number_of_fields_limit	entero	
enable_relative_error_limit	booleano	
relative_error_limit	entero	
enable_model_build_time_limit	booleano	
model_build_time_limit	entero	
enable_stop_after_time_limit	booleano	
stop_after_time_limit	entero	
stop_if_valid_model	booleano	
<algorithm>	booleano	Activa o desactiva el uso de un determinado algoritmo.
<algorithm>.<property>	cadena	Define un valor de propiedad para un algoritmo específico. Consulte el tema "Propiedades de ajustes de algoritmo" en la página 133 para obtener más información.

## Propiedades del nodo bayesnet



El nodo Red bayesiana le permite crear un modelo de probabilidad combinando pruebas observadas y registradas con conocimiento del mundo real para establecer la probabilidad de instancias. El nodo se centra en las redes Naïve Bayes aumentado a árbol (TAN) y de manto de Markov que se utilizan principalmente para la clasificación.

Tabla 89. Propiedades del nodo bayesnet.

Propiedades del nodo bayesnet	Valores	Descripción de la propiedad
inputs	[campo1 ... campoN]	Los modelos de red bayesiana utilizan un único campo objetivo y uno o más campos de entrada. Los campos continuos se establecen en intervalos automáticamente. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
continue_training_existing_model	booleano	
structure_type	TAN MarkovBlanket	Seleccione la estructura que desea utilizar cuando cree la red bayesiana.
use_feature_selection	booleano	

Tabla 89. Propiedades del nodo bayesnet (continuación).

Propiedades del nodo bayesnet	Valores	Descripción de la propiedad
parameter_learning_method	Likelihood Bayes	Especifica el método utilizado para calcular las tablas de probabilidad condicional entre nodos donde se conocen los valores de los elementos padre.
mode	Expert Simple	
missing_values	booleano	
all_probabilities	booleano	
independence	Likelihood Pearson	Especifica el método utilizado para determinar si las observaciones relacionadas de dos variables son independientes entre sí.
significance_level	número	Especifica el valor de corte para determinar la independencia.
maximal_conditioning_set	número	Establece el número máximo de variables de condición que se utilizarán para la comprobación de la independencia.
inputs_always_selected	[campo1 ... campoN]	Especifica qué campos del conjunto de datos se deben utilizar siempre al generar la red bayesiana. <i>Note:</i> el campo objetivo siempre está seleccionado.
maximum_number_inputs	número	Especifica el número máximo de campos de entrada que se deben utilizar al generar la red bayesiana.
calculate_variable_importance	booleano	
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	
adjusted_propensity_partition	Test Validation	

## Propiedades del nodo buildr



El nodo Crear R le permite especificar script R personalizado para realizar la creación de modelos y la puntuación de modelos desplegados en IBM SPSS Modeler.

Tabla 90. Propiedades de buildr.

Propiedades del nodo buildr	Valores	Descripción de la propiedad
build_syntax	cadena	Sintaxis de scripts R para la creación del modelo.
score_syntax	cadena	Sintaxis de scripts R para la puntuación del modelo.
convert_flags	StringsAndDoubles LogicalValues	Opción para convertir campos de distintivos.

Tabla 90. Propiedades de buildr (continuación).

Propiedades del nodo buildr	Valores	Descripción de la propiedad
convert_datetime	booleano	Opción para convertir las variables con los formatos de fecha o de fecha y hora para formatos de fecha/hora R.
convert_datetime_class	POSIXct POSIX1t	Opciones para especificar a qué formato se convierten las variables con los formatos de fecha o de fecha y hora.
convert_missing	booleano	Opción para convertir los valores que faltan al valor R NA.
output_html	booleano	Opción para visualizar gráficos en una pestaña en el nugget de modelo R.
output_text	booleano	Opción para escribir salida de texto de la consola R en una pestaña del modelo R.

## Propiedades del nodo c50



El nodo C5.0 genera un árbol de decisión o un conjunto de reglas. El modelo divide la muestra basándose en el campo que ofrece la máxima ganancia de información en cada nivel. El campo objetivo debe ser categórico. Se permiten varias divisiones en más de dos subgrupos.

Tabla 91. Propiedades del nodo c50.

Propiedades del nodo c50	Valores	Descripción de la propiedad
target	field	Los modelos C50 utilizan un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de ponderación. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
output_type	DecisionTree RuleSet	
group_symbolics	booleano	
use_boost	booleano	
boost_num_trials	número	
use_xval	booleano	
xval_num_folds	número	
mode	Simple Expert	
favor	Accuracy Generality	Generalización o precisión de favor.
expected_noise	número	
min_child_records	número	
pruning_severity	número	
use_costs	booleano	
costs	estructurado	Ésta es una propiedad estructurada.
use_winning	booleano	



Tabla 91. Propiedades del nodo c50 (continuación).

Propiedades del nodo c50	Valores	Descripción de la propiedad
use_global_pruning	booleano	Activado (True) de forma predeterminada.
calculate_variable_importance	booleano	
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	
adjusted_propensity_partition	Test Validation	

## Propiedades del nodo carma



El modelo CARMA extrae un conjunto de reglas de los datos sin necesidad de especificar campos de entrada ni de objetivo. A diferencia de Apriori el nodo CARMA ofrece configuraciones de generación basadas en el soporte de las reglas (soporte tanto para el antecedente como el consecuente) en lugar de hacerlo sólo respecto al soporte del antecedente. Esto significa que las reglas generadas se pueden utilizar en una gama de aplicaciones más amplia, por ejemplo, para buscar una lista de productos o servicios (antecedentes) cuyo consecuente es el elemento que se desea promocionar durante esta temporada de vacaciones.

Tabla 92. Propiedades del nodo carma.

Propiedades del nodo carma	Valores	Descripción de la propiedad
inputs	[field1 ... fieldn]	Los modelos CARMA utilizan una lista de campos de entrada, pero no de campos objetivo. Los campos de ponderación y frecuencia no se usan. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
id_field	field	Campo utilizado como el campo de ID para la generación del modelo.
contiguous	booleano	Se utiliza para especificar si los ID del campo de ID son contiguos.
use_transactional_data	booleano	
content_field	field	
min_supp	number(percent)	Está relacionado con el soporte de regla en lugar de con el soporte de antecedentes. El valor por omisión es 20%.
min_conf	number(percent)	El valor por omisión es 20%.
max_size	número	El valor por omisión es 10.
mode	Simple Expert	El valor predeterminado es Simple.
exclude_multiple	booleano	Excluye las reglas con varios consecuentes. El valor predeterminado es False.
use_pruning	booleano	El valor predeterminado es False.
pruning_value	número	El valor predeterminado es 500.
vary_support	booleano	
estimated_transactions	entero	

Tabla 92. Propiedades del nodo carma (continuación).

Propiedades del nodo carma	Valores	Descripción de la propiedad
rules_without_antecedents	booleano	

## Propiedades del nodo cart



El nodo de árbol de clasificación y regresión (C&R) genera un árbol de decisión que permite predecir o clasificar observaciones futuras. El método utiliza la partición reiterada para dividir los registros de entrenamiento en segmentos minimizando las impurezas en cada paso, donde un nodo se considera “puro” si el 100% de los casos del nodo corresponden a una categoría específica del campo objetivo. Los campos de entrada y objetivo pueden ser continuos (rango numérico) o categóricos (nominal, ordinal o marca). Todas las divisiones son binarias (sólo se crean dos subgrupos).

Tabla 93. Propiedades del nodo cart.

Propiedades del nodo cart	Valores	Descripción de la propiedad
target	field	Los modelos de árbol C&R requieren un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de frecuencia. Consulte el tema “Propiedades comunes de nodos de modelado” en la página 129 para obtener más información.
continue_training_existing_model	booleano	
objective	Standard Aumento Agregación autodocimante psm	psm se utiliza para conjuntos de datos de grandes dimensiones y requiere una conexión al Servidor.
model_output_type	Single InteractiveBuilder	
use_tree_directives	booleano	
tree_directives	cadena	Especifique directivas para desarrollar el árbol. Las directivas se pueden escribir entre comillas triples para evitar comillas o saltos de línea no deseados. Recuerde que las directivas pueden ser muy sensibles a las pequeñas modificaciones de las opciones de modelado o los datos y es posible que no se puedan generalizar para otros conjuntos de datos.
use_max_depth	Predeterminado Custom	
max_depth	entero	Máxima profundidad del árbol, desde 0 a 1000. Sólo se utiliza si use_max_depth = Custom.
prune_tree	booleano	Poda del árbol para evitar sobreajustes.
use_std_err	booleano	Use la diferencia máxima en riesgos (en errores estándar).
std_err_multiplier	número	Diferencia máxima.
max_surrogates	número	Número máximo de sustitutos.
use_percentage	booleano	

Tabla 93. Propiedades del nodo cart (continuación).

Propiedades del nodo cart	Valores	Descripción de la propiedad
min_parent_records_pc	número	
min_child_records_pc	número	
min_parent_records_abs	número	
min_child_records_abs	número	
use_costs	booleano	
costs	estructurado	Propiedad estructurada.
priors	Datos Equal Custom	
custom_priors	estructurado	Propiedad estructurada.
adjust_priors	booleano	
trails	número	Número de modelos de componente para un aumento o agregación autodocimante.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Regla de combinación predeterminada para objetivos categóricos.
range_ensemble_method	Media Mediana	Regla de combinación predeterminada para objetivos continuos.
large_boost	booleano	Aplicar aumento a conjunto de datos muy grandes.
min_impurity	número	
impurity_measure	Gini Twoing Ordered	
train_pct	número	Conjunto de prevención sobreajustado.
set_random_seed	booleano	Opción replicar resultados.
seed	número	
calculate_variable_importance	booleano	
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	
adjusted_propensity_partition	Test Validation	

## Propiedades del nodo chaid



El nodo CHAID genera árboles de decisión utilizando estadísticos de chi-cuadrado para identificar las divisiones óptimas. A diferencia de los nodos C&RT y QUEST, CHAID puede generar árboles no binarios, lo que significa que algunas divisiones generarán más de dos ramas. Los campos de entrada y objetivo pueden ser continuos (rango numérico) o categóricos. CHAID exhaustivo es una modificación de CHAID que examina con mayor precisión todas las divisiones posibles, aunque necesita más tiempo para realizar los cálculos.

Tabla 94. Propiedades del nodo chaid.

Propiedades del nodo chaid	Valores	Descripción de la propiedad
target	<i>field</i>	Los modelos CHAID requieren un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de frecuencia. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
continue_training_existing_model	<i>booleano</i>	
objective	Standard Aumento Agregación autodocimante psm	psm se utiliza para conjuntos de datos de grandes dimensiones y requiere una conexión al Servidor.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>booleano</i>	
tree_directives	<i>cadena</i>	
método	Chaid ExhaustiveChaid	
use_max_depth	Predeterminado Custom	
max_depth	<i>entero</i>	Máxima profundidad del árbol, desde 0 a 1000. Sólo se utiliza si use_max_depth = Custom.
use_percentage	<i>booleano</i>	
min_parent_records_pc	<i>número</i>	
min_child_records_pc	<i>número</i>	
min_parent_records_abs	<i>número</i>	
min_child_records_abs	<i>número</i>	
use_costs	<i>booleano</i>	
costs	<i>estructurado</i>	Propiedad estructurada.
trails	<i>número</i>	Número de modelos de componente para un aumento o agregación autodocimante.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Regla de combinación predeterminada para objetivos categóricos.
range_ensemble_method	Media Mediana	Regla de combinación predeterminada para objetivos continuos.
large_boost	<i>booleano</i>	Aplicar aumento a conjunto de datos muy grandes.
split_alpha	<i>número</i>	Nivel de significancia para división.
merge_alpha	<i>número</i>	Nivel de significancia para fusión.
bonferroni_adjustment	<i>booleano</i>	Los valores de significancia de ajuste utilizando el método de Bonferroni.
split_merged_categories	<i>booleano</i>	Permitir segunda división de categorías fusionadas.

Tabla 94. Propiedades del nodo chaid (continuación).

Propiedades del nodo chaid	Valores	Descripción de la propiedad
chi_square	Pearson LR	Método usado para calcular la estadística de chi cuadrado: Pearson o Razón de verosimilitud
epsilon	número	Cambio mínimo en frecuencias de casillas esperadas.
max_iterations	número	Número máximo de iteraciones para la convergencia.
set_random_seed	entero	
seed	número	
calculate_variable_importance	booleano	
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	
adjusted_propensity_partition	Test Validation	
maximum_number_of_models	entero	

## Propiedades del nodo coxreg



El nodo Regresión de Cox le permite crear un modelo de supervivencia para datos de tiempo hasta el evento en presencia de registros censurados. El modelo produce una función de supervivencia que predice la probabilidad de que el evento de interés se haya producido en el momento dado ( $t$ ) para valores determinados de las variables de entrada.

Tabla 95. Propiedades del nodo coxreg.

Propiedades del nodo coxreg	Valores	Descripción de la propiedad
survival_time	field	Los modelos de regresión de Cox requieren un único campo con los tiempos de supervivencia.
target	field	Los modelos de regresión de Cox requieren un único campo objetivo y uno o más campos de entrada. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
método	Intro Stepwise BackwardsStepwise	
groups	field	
model_type	MainEffects Custom	
custom_terms	["BP*Sexo" "BP*Edad"]	
mode	Expert Simple	
max_iterations	número	

Tabla 95. Propiedades del nodo coxreg (continuación).

Propiedades del nodo coxreg	Valores	Descripción de la propiedad
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
removal_criterion	LR Wald Conditional	
probability_entry	número	
probability_removal	número	
output_display	EachStep LastStep	
ci_enable	booleano	
ci_value	90 95 99	
correlación	booleano	
display_baseline	booleano	
survival	booleano	
hazard	booleano	
log_minus_log	booleano	
one_minus_survival	booleano	
separate_line	field	
value	number o string	Si no se especifica ningún valor para un campo, se utilizará la opción predeterminada "Mean" para dicho campo.

## Propiedades del nodo decisionlist



El nodo Lista de decisiones identifica subgrupos, o segmentos, que muestran una mayor o menor posibilidad de proporcionar un resultado binario relacionado con la población global. Por ejemplo, puede buscar clientes que tengan menos posibilidades de abandonar o más posibilidades de responder favorablemente a una campaña. Puede incorporar su conocimiento empresarial al modelo añadiendo sus propios segmentos personalizados y previzualizando modelos alternativos uno junto a otro para comparar los resultados. Los modelos de listas de decisiones constan de una lista de reglas en las que cada regla tiene una condición y un resultado. Las reglas se aplican en orden, y la primera regla que coincide determina el resultado.

Tabla 96. propiedades del nodo de decisionlist.

Propiedades del nodo decisionlist	Valores	Descripción de la propiedad
target	<i>field</i>	Los modelos de listas de decisiones utilizan un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de frecuencia. Consulte el tema “Propiedades comunes de nodos de modelado” en la página 129 para obtener más información.
model_output_type	Modelo InteractiveBuilder	
search_direction	Up Down	Hace referencia a la localización de segmentos, donde Up es el equivalente a Alta probabilidad y Down es el equivalente a Baja probabilidad.
target_value	<i>cadena</i>	Si no se especifica, se supondrá el valor true para las marcas.
max_rules	<i>entero</i>	Número máximo de segmentos sin incluir el resto.
min_group_size	<i>entero</i>	Tamaño mínimo del segmento.
min_group_size_pct	<i>número</i>	Tamaño mínimo del segmento como porcentaje.
confidence_level	<i>número</i>	Umbral mínimo que un campo de entrada tiene que mejorar la probabilidad de la respuesta (aumentar la elevación) para que merezca la pena añadirlo a la definición de un segmento.
max_segments_per_rule	<i>entero</i>	
mode	Simple Expert	
bin_method	EqualWidth EqualCount	
bin_count	<i>número</i>	
max_models_per_cycle	<i>entero</i>	Amplitud de búsqueda de las listas.
max_rules_per_cycle	<i>entero</i>	Amplitud de búsqueda de las reglas de segmentación.
segment_growth	<i>número</i>	
include_missing	<i>booleano</i>	
final_results_only	<i>booleano</i>	

Tabla 96. propiedades del nodo de decisionlist (continuación).

Propiedades del nodo decisionlist	Valores	Descripción de la propiedad
reuse_fields	booleano	Permite la reutilización de los atributos (los campos de entrada que aparecen en las reglas).
max_alternatives	entero	
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	
adjusted_propensity_partition	Test Validation	

## Propiedades del nodo discriminant



El análisis discriminante realiza más supuestos rigurosos que regresiones logísticas, pero puede ser una alternativa o un suplemento valioso al análisis de regresión logística si se cumplen dichos supuestos.

Tabla 97. Propiedades del nodo discriminant.

Propiedades del nodo discriminant	Valores	Descripción de la propiedad
target	field	Los modelos discriminantes requieren un único campo objetivo y uno o más campos de entrada. Los campos de ponderación y frecuencia no se usan. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
método	Intro Stepwise	
mode	Simple Expert	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups SeparateGroups	
means	booleano	Opciones de estadísticos del cuadro de diálogo Salida avanzada.
univariate_anovas	booleano	
box_m	booleano	
within_group_covariance	booleano	
within_groups_correlation	booleano	
separate_groups_covariance	booleano	
total_covariance	booleano	
fishers	booleano	
unstandardized	booleano	
casewise_results	booleano	Opciones de clasificación del cuadro de diálogo Salida avanzada.
limit_to_first	número	El valor predeterminado es 10.



Tabla 97. Propiedades del nodo discriminant (continuación).

Propiedades del nodo discriminant	Valores	Descripción de la propiedad
summary_table	booleano	
leave_one_classification	booleano	
combined_groups	booleano	
separate_groups_covariance	booleano	Opción de matrices <b>Covarianza de grupos separados</b> .
territorial_map	booleano	
combined_groups	booleano	Opción de gráfico <b>Grupos combinados</b> .
separate_groups	booleano	Opción de gráfico <b>Grupos separados</b> .
summary_of_steps	booleano	
F_pairwise	booleano	
stepwise_method	WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	número	
criteria	UseValue UseProbability	
F_value_entry	número	El valor predeterminado es 3.84.
F_value_removal	número	El valor predeterminado es 2.71.
probability_entry	número	El valor predeterminado es 0.05.
probability_removal	número	El valor predeterminado es 0,10.
calculate_variable_importance	booleano	
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	
adjusted_propensity_partition	Test Validation	

## Propiedades del nodo factor



El nodo PCA/Factorial proporciona técnicas eficaces de reducción de datos para reducir la complejidad de los datos. Análisis de componentes principales (PCA) busca combinaciones lineales de los campos de entrada que realizan el mejor trabajo a la hora de capturar la varianza en todo el conjunto de campos, en el que los componentes son ortogonales (perpendiculares) entre ellos. Análisis factorial intenta identificar factores subyacentes que expliquen el patrón de correlaciones dentro de un conjunto de campos observados. Para los dos métodos, el objetivo es encontrar un número pequeño de campos derivados que resuma de forma eficaz la información del conjunto original de campos.

Tabla 98. Propiedades del nodo factor.

Propiedades del nodo factor	Valores	Descripción de la propiedad
inputs	[field1 ... fieldN]	Los modelos PCA/Factorial utilizan una lista de campos de entrada, pero no de campos objetivo. Los campos de ponderación y frecuencia no se usan. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
método	PC ULS GLS ML PAF Alpha Image	
mode	Simple Expert	
max_iterations	número	
complete_records	booleano	
matrix	Correlation Covariance	
extract_factors	ByEigenvalues ByFactors	
min_eigenvalue	número	
max_factor	número	
rotation	Ninguno Varimax DirectOblimin Equamax Quartimax Promax	
delta	número	Si selecciona DirectOblimin como el tipo de datos de rotación, podrá especificar un valor para delta.  Si no especifica ningún valor, se utilizará el valor predeterminado para delta.
kappa	número	Si selecciona Promax como el tipo de datos de rotación, podrá especificar un valor para kappa.  Si no especifica ningún valor, se utilizará el valor predeterminado para kappa.
sort_values	booleano	
hide_values	booleano	
hide_below	número	

## Propiedades del nodo featureselection



El nodo Selección de características filtra los campos de entrada para su eliminación en función de un conjunto de criterios (como el porcentaje de valores perdidos); a continuación, clasifica el grado de importancia del resto de entradas de acuerdo con un objetivo específico. Por ejemplo, a partir de un conjunto de datos dado con cientos de entradas potenciales, ¿cuáles tienen mayor probabilidad de ser útiles para el modelado de resultados de pacientes?

Tabla 99. Propiedades del nodo featureselection.

Propiedades del nodo featureselection	Valores	Descripción de la propiedad
target	field	Los modelos de selección de características ordenan predictores por rangos con respecto a su objetivo específico. Los campos de ponderación y frecuencia no se usan. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
screen_single_category	booleano	En caso de establecer True, filtra campos que tienen demasiados registros dentro de la misma categoría respecto al número total de registros.
max_single_category	número	Especifica el umbral que se utiliza cuando screen_single_category es True.
screen_missing_values	booleano	Si se establece en True, filtra campos con demasiados valores perdidos, expresados como un porcentaje del número total de registros.
max_missing_values	número	
screen_num_categories	booleano	Si se establece en True, filtra campos con demasiadas categorías respecto al número total de registros.
max_num_categories	número	
screen_std_dev	booleano	Si se establece en True, filtra campos con una desviación estándar menor o igual que el mínimo especificado.
min_std_dev	número	
screen_coeff_of_var	booleano	Si se establece en True, filtra campos con un coeficiente de varianza menor o igual que el mínimo especificado.
min_coeff_of_var	número	
criteria	Pearson Likelihood CramersV Lambda	Al clasificar los predictores categóricos en función de un objetivo categórico, especifica la medida en la que se basa el valor de importancia.
unimportant_below	número	Especifica los valores $p$ de umbral utilizados para clasificar las variables como importantes, marginales o sin importancia. Acepta valores de 0,0 a 1,0.
important_above	número	Acepta valores de 0,0 a 1,0.
unimportant_label	cadena	Especifica la etiqueta para la clasificación como 'Sin importancia'.

Tabla 99. Propiedades del nodo featureselection (continuación).

Propiedades del nodo featureselection	Valores	Descripción de la propiedad
marginal_label	cadena	
important_label	cadena	
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	booleano	Si selection_mode se establece en ImportanceLevel, determina si se seleccionan los campos importantes.
select_marginal	booleano	Si selection_mode se establece en ImportanceLevel, determina si se seleccionan los campos marginales.
select_unimportant	booleano	Si selection_mode se establece en ImportanceLevel, determina si se seleccionan los campos sin importancia.
importance_value	número	Si selection_mode se establece en ImportanceValue, determina el valor de corte que se va a usar. Acepta valores de 0 a 100.
top_n	entero	Si selection_mode se establece en TopN, determina el valor de corte que se va a usar. Acepta valores de 0 a 1000.

## Propiedades del nodo genlin



El modelo lineal generalizado amplía el modelo lineal general, de manera que la variable dependiente está relacionada linealmente con los factores y las covariables mediante una determinada función de enlace. Además, el modelo permite que la variable dependiente tenga una distribución no normal. Cubre la funcionalidad de un amplio número de modelos estadísticos, incluyendo regresión lineal, regresión logística, modelos log lineales para recuento de datos y modelos de supervivencia censurados por intervalos.

Tabla 100. Propiedades del nodo genlin.

Propiedades del nodo genlin	Valores	Descripción de la propiedad
target	field	Los modelos lineales generalizados requieren un único campo objetivo, que debe ser un campo nominal o marca, y uno o más campos de entrada. También se puede especificar un campo de ponderación. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
use_weight	booleano	
weight_field	field	El tipo de campo es únicamente continuo.
target_represents_trials	booleano	
trials_type	Variable FixedValue	
trials_field	field	El tipo de campo es continuo, marca u ordinal.

Tabla 100. Propiedades del nodo `genlin` (continuación).

Propiedades del nodo <code>genlin</code>	Valores	Descripción de la propiedad
<code>trials_number</code>	<i>número</i>	El valor predeterminado es 10.
<code>model_type</code>	MainEffects MainAndAllTwoWayEffects	
<code>offset_type</code>	Variable FixedValue	
<code>offset_field</code>	<i>field</i>	El tipo de campo es únicamente continuo.
<code>offset_value</code>	<i>número</i>	Debe ser un número real.
<code>base_category</code>	Last Primero	
<code>include_intercept</code>	<i>booleano</i>	
<code>mode</code>	Simple Expert	
<code>distribution</code>	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL	IGAUSS: De Gauss inversa. NEGBIN: Binomial negativa.
<code>negbin_para_type</code>	Especifica Estimate	
<code>negbin_parameter</code>	<i>número</i>	El valor predeterminado es 1. Debe contener un número real no negativo.
<code>tweedie_parameter</code>	<i>número</i>	
<code>link_function</code>	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPower PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG: log-log complementario. LOGC: complemento log. NEGBIN: Binomial negativa. NLOGLOG: Log-log negativo. CUMCAUCHIT: Cauchit acumulada. CUMCLOGLOG: Log-log complementario acumulado. CUMLOGIT: Logit acumulado. CUMNLOGLOG: Log-log negativo acumulado. CUMPROBIT: Probit acumulado.
<code>power</code>	<i>número</i>	El valor debe ser real y distinto de cero.
<code>método</code>	Hybrid Fisher NewtonRaphson	
<code>max_fisher_iterations</code>	<i>número</i>	El valor predeterminado es 1; sólo se admiten enteros positivos.

Tabla 100. Propiedades del nodo genlin (continuación).

Propiedades del nodo genlin	Valores	Descripción de la propiedad
scale_method	MaxLikelihoodEstimate Deviance PearsonChiSquare FixedValue	
scale_value	número	El valor predeterminado es 1; debe ser mayor que 0.
covariance_matrix	ModelEstimator RobustEstimator	
max_iterations	número	El valor predeterminado es 100; sólo enteros no negativos.
max_step_halving	número	El valor predeterminado es 5; sólo enteros positivos.
check_separation	booleano	
start_iteration	número	El valor predeterminado es 20; sólo se admiten enteros positivos.
estimates_change	booleano	
estimates_change_min	número	El valor predeterminado es 1E-006; sólo se admiten números positivos.
estimates_change_type	Absolute Relative	
loglikelihood_change	booleano	
loglikelihood_change_min	número	Sólo se admiten números positivos.
loglikelihood_change_type	Absolute Relative	
hessian_convergence	booleano	
hessian_convergence_min	número	Sólo se admiten números positivos.
hessian_convergence_type	Absolute Relative	
case_summary	booleano	
contrast_matrices	booleano	
descriptive_statistics	booleano	
estimable_functions	booleano	
model_info	booleano	
iteration_history	booleano	
goodness_of_fit	booleano	
print_interval	número	El valor predeterminado es 1; debe ser un entero positivo.
model_summary	booleano	
lagrange_multiplier	booleano	
parameter_estimates	booleano	
include_exponential	booleano	
covariance_estimates	booleano	
correlation_estimates	booleano	

Tabla 100. Propiedades del nodo *genlin* (continuación).

Propiedades del nodo <i>genlin</i>	Valores	Descripción de la propiedad
<i>analysis_type</i>	TypeI TypeIII TypeIAndTypeIII	
<i>statistics</i>	Wald LR	
<i>citype</i>	Wald Profile	
<i>tolerancelevel</i>	número	El valor predeterminado es 0.0001.
<i>confidence_interval</i>	número	El valor predeterminado es 95.
<i>loglikelihood_function</i>	Completa 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>	booleano	
<i>calculate_raw_propensities</i>	booleano	
<i>calculate_adjusted_propensities</i>	booleano	
<i>adjusted_propensity_partition</i>	Test Validation	

## Propiedades del nodo *glmm*



Un modelo lineal mixto generalizado (GLMM) amplía el modelo lineal de modo que el objetivo pueda tener una distribución no normal, esté linealmente relacionado con los factores y covariables mediante una función de enlace especificada y las observaciones se puedan correlacionar. Los modelos lineales mixtos generalizados cubren una amplia variedad de modelos, desde modelos de regresión lineal simple hasta modelos multinivel complejos para datos longitudinales no normales.

Tabla 101. Propiedades del nodo *glmm*.

Propiedades del nodo <i>glmm</i>	Valores	Descripción de la propiedad
<i>residual_subject_spec</i>	<i>estructurado</i>	La combinación de valores de los campos categóricos especificados que definen de forma exclusiva los sujetos del conjunto de datos.
<i>repeated_measures</i>	<i>estructurado</i>	Campos utilizados para identificar observaciones repetidas.
<i>residual_group_spec</i>	[ <i>field1 ... fieldN</i> ]	Campos que definen conjuntos independientes de parámetros de covarianza de efectos repetidos.

Tabla 101. Propiedades del nodo glmm (continuación).

Propiedades del nodo glmm	Valores	Descripción de la propiedad
residual_covariance_type	Diagonal AR1 ARMA11 COMPOUND_SYMMETRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS	Especifica la estructura de covarianza de residuos.
custom_target	booleano	Indica si se puede utilizar un objetivo definido en el nodo anterior (false) o un objetivo personalizado especificado por target_field (true).
target_field	field	Campo a utilizar como objetivo si custom_target es true.
use_trials	booleano	Indica si hay que utilizar el campo adicional o el valor que especifica el número de ensayos cuando la respuesta objetivo es un número de eventos que tienen lugar en un conjunto de ensayos. El valor predeterminado es false.
use_field_or_value	Campo Value	Indica si se utiliza el campo (valor predeterminado) o valor para especificar el número de ensayos.
trials_field	field	Campo a utilizar para especificar el número de ensayos.
trials_value	entero	Valor a utilizar para especificar el número de ensayos. Si se especifica, el valor mínimo es 1.
use_custom_target_reference	booleano	Indica si hay que utilizar la categoría de referencia personalizada para un objetivo categórico. El valor predeterminado es false.
target_reference_value	cadena	Categoría de referencia a utilizar si use_custom_target_reference es true.
dist_link_combination	Nominal Logit GammaLog BinomialLogit PoissonLog BinomialProbit NegbinLog BinomialLogC Custom	Modelos comunes para la distribución de valores de objetivo. Seleccione Custom para especificar una distribución de la lista proporcionada por target_distribution.
target_distribution	Normal Binomial Multinomial Gamma Inverso NegativeBinomial Poisson	La distribución de valores de objetivo cuando dist_link_combination es Custom.



Tabla 101. Propiedades del nodo glmm (continuación).

Propiedades del nodo glmm	Valores	Descripción de la propiedad
link_function_type	IDENTITY LOGC LOG CLOGLOG LOGIT NLOGLOG PROBIT POWER CAUCHIT	Función de enlace para relacionar valores de objetivo a los predictores. Si target_distribution es Binomial podrá utilizarse cualquiera de las funciones de enlace listadas. Si target_distribution es Multinomial podrán utilizarse CLOGLOG, CAUCHIT, LOGIT, NLOGLOG o PROBIT. Si target_distribution es cualquier cosa distinta de Binomial o Multinomial, podrán utilizarse IDENTITY, LOG o POWER.
link_function_param	número	Valor del parámetro de función de enlace que hay que utilizar. Sólo es aplicable si normal_link_function o link_function_type es POWER.
use_predefined_inputs	booleano	Indica si los campos de efectos fijos deben ser aquellos definidos anteriormente como campos de entrada (true) o han de ser los campos fixed_effects_list (false). El valor predeterminado es false.
fixed_effects_list	estructurado	Si use_predefined_inputs es falso, especifica los campos de entrada que se han de utilizar como campos de efectos fijos.
use_intercept	booleano	Si es true, el valor predeterminado. incluye la interceptación en el modelo.
random_effects_list	estructurado	Lista de campos para especificar como efectos aleatorios.
regression_weight_field	field	Campo a utilizar como campo de ponderación de análisis.
use_offset	None offset_value offset_field	Indica cómo se especifica la compensación. El valor None significa que no se ha utilizado compensación.
offset_value	número	El valor que se ha de utilizar para desplazamiento si use_offset se establece en offset_value.
offset_field	field	El valor que se ha de utilizar para desplazamiento si use_offset se establece en offset_field.
target_category_order	Ascending Descending Datos	Orden de clasificación para objetivos categóricos. El valor Data especifica que se utiliza el orden de clasificación de los datos. El valor predeterminado es Ascending.
inputs_category_order	Ascending Descending Datos	Orden de clasificación para predictores categóricos. El valor Data especifica que se utiliza el orden de clasificación de los datos. El valor predeterminado es Ascending.

Tabla 101. Propiedades del nodo glmm (continuación).

Propiedades del nodo glmm	Valores	Descripción de la propiedad
max_iterations	entero	Número máximo de iteraciones que ejecutará el algoritmo. Un número entero no negativo; el valor predeterminado es 100.
confidence_level	entero	Nivel de confianza utilizado para calcular estimaciones de intervalo de los coeficientes del modelo. Un número entero no negativo; el valor máximo es 100, el valor predeterminado es 95.
degrees_of_freedom_method	Fixed Varied	Especifica cómo se calculan los grados de libertad para la prueba de significación.
test_fixed_effects_coefficients	Modelo Robust	Método para calcular la matriz de covarianza de las estimaciones de los parámetros.
use_p_converge	booleano	Opción para la convergencia de parámetros.
p_converge	número	Blanco, o cualquier valor positivo.
p_converge_type	Absolute Relative	
use_l_converge	booleano	Opción para la convergencia log-likelihood.
l_converge	número	Blanco, o cualquier valor positivo.
l_converge_type	Absolute Relative	
use_h_converge	booleano	Opción para la convergencia hessiana.
h_converge	número	Blanco, o cualquier valor positivo.
h_converge_type	Absolute Relative	
max_fisher_steps	entero	
singularity_tolerance	número	
use_model_name	booleano	Indica si hay que especificar un nombre personalizado para el modelo (true) o si se ha de utilizar el nombre generado por el sistema (false). El valor predeterminado es false.
model_name	cadena	Si use_model_name es true, especifica el nombre de modelo que se va a utilizar.
confidence	onProbability onIncrease	Base para calcular el valor de confianza de la puntuación: probabilidad más alta predicha, o la diferencia entre la probabilidad más alta predicha y la segunda probabilidad más alta.
score_category_probabilities	booleano	Si es true, genera las probabilidades predichas para objetivos categóricos. El valor predeterminado es false.
max_categories	entero	Si score_category_probabilities es true, especifica el número máximo de categorías que se han de guardar.

Tabla 101. Propiedades del nodo glmm (continuación).

Propiedades del nodo glmm	Valores	Descripción de la propiedad
score_propensity	booleano	Si es true, produce puntuaciones de propensión para campos de objetivo de marca que indican la probabilidad del resultado "true" para el campo.
emeans	estructura	Para cada campo categórico de la lista de efectos fijos, especifica si hay que producir medias marginales estimadas.
covariance_list	estructura	Para cada campo continuo de la lista de efectos fijos, especifica si hay que usar la media o un valor personalizado al calcular medias marginales estimadas.
mean_scale	Original Transformed	Especifica si las medias marginales estimadas se calculan basándose en la escala original del objetivo (valor predeterminado) o en la transformación de la función de enlace.
comparison_adjustment_method	DMS SEQBONFERRONI SEQSIDAK	Método de ajuste que hay que utilizar al realizar pruebas de hipótesis con varios contrastes.

## Propiedades del nodo kmeans



El nodo K-medias agrupa conjuntos de datos en grupos distintos (o clústeres). El método define un número fijo de clústeres, de forma iterativa asigna registros a los clústeres y ajusta los centros de los clústeres hasta que no se pueda mejorar el modelo. En lugar de intentar predecir un resultado, los modelos de *k*-medias utilizan un proceso conocido como aprendizaje no supervisado para revelar los patrones del conjunto de campos de entrada.

Tabla 102. propiedades del nodo kmeans.

Propiedades del nodo kmeans	Valores	Descripción de la propiedad
inputs	[field1 ... fieldN]	Los modelos de K-Medias realizan un análisis de clústeres en un conjunto de campos de entrada pero no utilizan ningún campo objetivo. Los campos de ponderación y frecuencia no se usan. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
num_clusters	número	
gen_distance	booleano	
cluster_label	Cadena Number	
label_prefix	cadena	
mode	Simple Expert	
stop_on	Predeterminado Custom	
max_iterations	número	
tolerance	número	

Tabla 102. propiedades del nodo kmeans (continuación).

Propiedades del nodo kmeans	Valores	Descripción de la propiedad
encoding_value	número	
optimize	Speed Memory	Se utiliza para especificar si la generación del modelo se debe optimizar para la velocidad o la memoria.

## Propiedades del nodo knn



El nodo  $k$  de modelado de vecino (KNN) asocia el nuevo caso con la categoría o valor de los objetos  $k$  junto a él en el espacio de predictores, donde  $k$  es un entero. Los casos parecidos están próximos y los que no lo son están alejados entre sí.

Tabla 103. Propiedades del nodo knn.

Propiedades del nodo knn	Valores	Descripción de la propiedad
analysis	PredictTarget IdentifyNeighbors	
objective	Balance Speed Accuracy Custom	
normalize_ranges	booleano	
use_case_labels	booleano	Seleccione esta casilla de verificación para activar la siguiente opción.
case_labels_field	field	
identify_focal_cases	booleano	Seleccione esta casilla de verificación para activar la siguiente opción.
focal_cases_field	field	
automatic_k_selection	booleano	
fixed_k	entero	Se activa únicamente si el valor de automatic_k_selection es False.
minimum_k	entero	Se activa únicamente si el valor de automatic_k_selection es True.
maximum_k	entero	
distance_computation	Euclidean CityBlock	
weight_by_importance	booleano	
range_predictions	Media Mediana	
perform_feature_selection	booleano	
forced_entry_inputs	[field1 ... fieldN]	
stop_on_error_ratio	booleano	
number_to_select	entero	
minimum_change	número	
validation_fold_assign_by_field	booleano	

Tabla 103. Propiedades del nodo knn (continuación).

Propiedades del nodo knn	Valores	Descripción de la propiedad
number_of_folds	entero	Sólo se activa si el valor de validation_fold_assign_by_field es False
set_random_seed	booleano	
random_seed	número	
folds_field	field	Sólo se activa si el valor de validation_fold_assign_by_field es True
all_probabilities	booleano	
save_distances	booleano	
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	
adjusted_propensity_partition	Test Validation	

## Propiedades del nodo kohonen



El nodo Kohonen genera un tipo de red neuronal que se puede usar para agrupar un conjunto de datos en grupos distintos. Cuando la red se termina de entrenar, los registros que son similares se deberían cerrar juntos en el mapa de resultados, mientras que los registros que son diferentes aparecerían aparte. Puede observar el número de observaciones capturadas por cada unidad en el nugget de modelo para identificar unidades fuertes. Esto le proporcionará una idea del número apropiado de clústeres.

Tabla 104. propiedades del nodo kohonen.

Propiedades del nodo kohonen	Valores	Descripción de la propiedad
inputs	[field1 ... fieldN]	Los modelos Kohonen utilizan una lista de campos de entrada, pero no de campos objetivo. Los campos de frecuencia y ponderación no se usan. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
continue	booleano	
show_feedback	booleano	
stop_on	Predeterminado Hora	
time	número	
optimize	Speed Memory	Se utiliza para especificar si la generación del modelo se debe optimizar para la velocidad o la memoria.
cluster_label	booleano	
mode	Simple Expert	
width	número	
length	número	
decay_style	Lineal Exponential	

Tabla 104. propiedades del nodo kohonen (continuación).

Propiedades del nodo kohonen	Valores	Descripción de la propiedad
phase1_neighborhood	número	
phase1_eta	número	
phase1_cycles	número	
phase2_neighborhood	número	
phase2_eta	número	
phase2_cycles	número	

## Propiedades del nodo lineal



Los modelos de regresión lineal predicen un objetivo continuo tomando como base las relaciones lineales entre el destino y uno o más predictores.

Tabla 105. Propiedades de nodo lineal.

Propiedades del nodo lineal	Valores	Descripción de la propiedad
target	field	Especifica un campo de objetivo único.
inputs	[field1 ... fieldN]	Campos de predictor utilizados por el modelo.
continue_training_existing_model	booleano	
objective	Standard Agregación autodocimante Aumento psm	psm se utiliza para conjuntos de datos de grandes dimensiones y requiere una conexión al Servidor.
use_auto_data_preparation	booleano	
confidence_level	número	
model_selection	ForwardStepwise BestSubsets None	
criteria_forward_stepwise	AICC Estadísticas F R cuadrado corregido ASE	
probability_entry	número	
probability_removal	número	
use_max_effects	booleano	
max_effects	número	
use_max_steps	booleano	
max_steps	número	
criteria_best_subsets	AICC R cuadrado corregido ASE	

Tabla 105. Propiedades de nodo lineal (continuación).

Propiedades del nodo lineal	Valores	Descripción de la propiedad
combining_rule_continuous	Media Mediana	
component_models_n	número	
use_random_seed	booleano	
random_seed	número	
use_custom_model_name	booleano	
custom_model_name	cadena	
use_custom_name	booleano	
custom_name	cadena	
tooltip	cadena	
keywords	cadena	
annotation	cadena	

## Propiedades del nodo logreg



La regresión logística es una técnica de estadístico para clasificar los registros en función los valores de los campos de entrada. Es análoga a la regresión lineal pero toma un campo objetivo categórico en lugar de uno numérico.

Tabla 106. Propiedades del nodo logreg.

Propiedades del nodo logreg	Valores	Descripción de la propiedad
target	field	Los modelos de regresión logística requieren un único campo objetivo y uno o más campos de entrada. Los campos de frecuencia y ponderación no se usan. Consulte el tema “Propiedades comunes de nodos de modelado” en la página 129 para obtener más información.
logistic_procedure	Binomial Multinomial	
include_constant	booleano	
mode	Simple Expert	
método	Intro Stepwise Forwards Backwards BackwardsStepwise	
binomial_method	Intro Forwards Backwards	

Tabla 106. Propiedades del nodo logreg (continuación).

Propiedades del nodo logreg	Valores	Descripción de la propiedad
model_type	MainEffects FullFactorial Custom	Si FullFactorial se especifica como el tipo de modelo, no se ejecutarán los métodos por pasos, aunque así se indique. En su lugar, el método utilizado será Enter.  Si el tipo de modelo se establece en Custom pero no se ha especificado ningún campo personalizado, se generará un modelo de efectos principales.
custom_terms	[{BP Sexo}{BP{Edad}}	
multinomial_base_category	cadena	Especifica cómo se determina la categoría de referencia.
binomial_categorical_input	cadena	
binomial_input_contrast	Indicator Simple Diferencia Helmert Repeated Polinómico Desviación	Propiedad con clave para la entrada categórica que especifica cómo se determina el contraste.
binomial_input_category	Primero Last	Propiedad con clave para la entrada categórica que especifica cómo se determina la categoría de referencia.
scale	None UserDefined Pearson Deviance	
scale_value	número	
all_probabilities	booleano	
tolerance	1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10	
min_terms	número	
use_max_terms	booleano	
max_terms	número	
entry_criterion	Puntuación LR	
removal_criterion	LR Wald	
probability_entry	número	
probability_removal	número	
binomial_probability_entry	número	
binomial_probability_removal	número	
requirements	HierarchyDiscrete HierarchyAll Containment Ninguno	



Tabla 106. Propiedades del nodo logreg (continuación).

Propiedades del nodo logreg	Valores	Descripción de la propiedad
max_iterations	número	
max_steps	número	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
delta	número	
iteration_history	booleano	
history_steps	número	
summary	booleano	
likelihood_ratio	booleano	
asymptotic_correlation	booleano	
goodness_fit	booleano	
parameters	booleano	
confidence_interval	número	
asymptotic_covariance	booleano	
classification_table	booleano	
stepwise_summary	booleano	
info_criteria	booleano	
monotonicity_measures	booleano	
binomial_output_display	at_each_step at_last_step	
binomial_goodness_of_fit	booleano	
binomial_parameters	booleano	
binomial_iteration_history	booleano	
binomial_classification_plots	booleano	
binomial_ci_enable	booleano	
binomial_ci	número	
binomial_residual	valores atípicos all	
binomial_residual_enable	booleano	
binomial_outlier_threshold	número	
binomial_classification_cutoff	número	
binomial_removal_criterion	LR Wald Conditional	

Tabla 106. Propiedades del nodo logreg (continuación).

Propiedades del nodo logreg	Valores	Descripción de la propiedad
calculate_variable_importance	booleano	
calculate_raw_propensities	booleano	

## Propiedades del nodo neuralnet

**Precaución:** Una versión más reciente del nodo de modelado Red neural, con características mejoradas, está disponible en esta versión y se describe en la sección siguiente (*neuralnetwork*). Aunque aún puede generar y puntuar un modelo con la versión anterior, recomendamos que actualice sus scripts para que se use la nueva versión. Los detalles de la versión anterior se conservan aquí como referencia.

Tabla 107. Propiedades del nodo neuralnet.

Propiedades del nodo neuralnet	Valores	Descripción de la propiedad
targets	[field1 ... fieldN]	El nodo Red neuronal espera uno o varios campos objetivo y uno o más campos de entrada. Los campos de frecuencia y ponderación se omiten. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
método	Quick Dynamic Multiple Prune ExhaustivePrune RBFN	
prevent_overtrain	booleano	
train_pct	número	
set_random_seed	booleano	
random_seed	número	
mode	Simple Expert	
stop_on	Predeterminado Exactitud Cycles Hora	Modo de parada.
accuracy	número	Precisión de parada.
cycles	número	Ciclos para entrenar.
time	número	Tiempo para entrenar (minutos).
continue	booleano	
show_feedback	booleano	
binary_encode	booleano	
use_last_model	booleano	
gen_logfile	booleano	
logfile_name	cadena	
alpha	número	
initial_eta	número	

Tabla 107. Propiedades del nodo neuralnet (continuación).

Propiedades del nodo neuralnet	Valores	Descripción de la propiedad
high_eta	número	
low_eta	número	
eta_decay_cycles	número	
hid_layers	One Two Three	
hl_units_one	número	
hl_units_two	número	
hl_units_three	número	
~prsistence	número	
m_topologies	cadena	
m_non_pyramids	booleano	
m_persistence	número	
p_hid_layers	One Two Three	
p_hl_units_one	número	
p_hl_units_two	número	
p_hl_units_three	número	
p_persistence	número	
p_hid_rate	número	
p_hid_pers	número	
p_inp_rate	número	
p_inp_pers	número	
p_overall_pers	número	
p_persistence	número	
r_num_clusters	número	
r_eta_auto	booleano	
r_alpha	número	
r_eta	número	
optimize	Speed Memory	Se utiliza para especificar si la generación del modelo se debe optimizar para la velocidad o la memoria.
calculate_variable_importance	booleano	Nota: La propiedad sensitivity_analysis utilizada en versiones anteriores se ha desaprobadado en favor de esta propiedad. La propiedad anterior se sigue admitiendo, pero se recomienda calculate_variable_importance.
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	
adjusted_propensity_partition	Test Validation	

## Propiedades del nodo neuralnetwork

El nodo Red neuronal utiliza un modelo simplificado que emula el modo en que el cerebro humano procesa la información: Funciona simultaneando un número elevado de unidades simples de procesamiento interconectadas que parecen versiones abstractas de neuronas. Las redes neuronales son dispositivos eficaces de cálculo de funciones generales y requieren un conocimiento matemático o estadístico mínimo para entrenarlas o aplicarlas.

Tabla 108. Propiedades del nodo neuralnetwork.

Propiedades del nodo neuralnetwork	Valores	Descripción de la propiedad
targets	[field1 ... fieldN]	Especifica campos objetivo.
inputs	[field1 ... fieldN]	Campos de predictor utilizados por el modelo.
splits	[campo1 ... campoN]	Especifica el campo o campos para utilizar en el modelado de divisiones.
use_partition	booleano	Si se ha definido un campo de partición, esta opción garantiza que sólo se utilizarán los datos de la partición de entrenamiento para la generación del modelo.
continue	booleano	Continuar entrenando modelo existente.
objective	Standard Agregación autodocimante Aumento psm	psm se utiliza para conjuntos de datos de grandes dimensiones y requiere una conexión al Servidor.
método	MultilayerPerceptron RadialBasisFunction	
use_custom_layers	booleano	
first_layer_units	número	
second_layer_units	número	
use_max_time	booleano	
max_time	número	
use_max_cycles	booleano	
max_cycles	número	
use_min_accuracy	booleano	
min_accuracy	número	
combining_rule_categorical	Voting HighestProbability HighestMeanProbability	
combining_rule_continuous	Media Mediana	
component_models_n	número	
overfit_prevention_pct	número	
use_random_seed	booleano	
random_seed	número	

Tabla 108. Propiedades del nodo neuralnetwork (continuación).

Propiedades del nodo neuralnetwork	Valores	Descripción de la propiedad
missing_values	listwiseDeletion missingValueImputation	
use_custom_model_name	booleano	
custom_model_name	cadena	
confidence	onProbability onIncrease	
score_category_probabilities	booleano	
max_categories	número	
score_propensity	booleano	
use_custom_name	booleano	
custom_name	cadena	
tooltip	cadena	
keywords	cadena	
annotation	cadena	

## Propiedades del nodo quest



El nodo QUEST proporciona un método de clasificación binario para generar árboles de decisión; está diseñado para reducir el tiempo de procesamiento necesario para realizar los análisis de C&RT y reducir la tendencia de los métodos de clasificación de árboles para favorecer a las entradas que permitan realizar más divisiones. Los campos de entrada pueden ser continuos (rango numérico), sin embargo el campo objetivo debe ser categórico. Todas las divisiones son binarias.

Tabla 109. Propiedades del nodo quest.

Propiedades del nodo quest	Valores	Descripción de la propiedad
target	field	Los modelos QUEST requieren un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de frecuencia. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
continue_training_existing_model	booleano	
objective	Standard Aumento Agregación autodocimante psm	psm se utiliza para conjuntos de datos de grandes dimensiones y requiere una conexión al Servidor.
model_output_type	Single InteractiveBuilder	
use_tree_directives	booleano	
tree_directives	cadena	
use_max_depth	Predeterminado Custom	

Tabla 109. Propiedades del nodo quest (continuación).

Propiedades del nodo quest	Valores	Descripción de la propiedad
max_depth	entero	Máxima profundidad del árbol, desde 0 a 1000. Sólo se utiliza si use_max_depth = Custom.
prune_tree	booleano	Poda del árbol para evitar sobreajustes.
use_std_err	booleano	Use la diferencia máxima en riesgos (en errores estándar).
std_err_multiplier	número	Diferencia máxima.
max_surrogates	número	Número máximo de sustitutos.
use_percentage	booleano	
min_parent_records_pc	número	
min_child_records_pc	número	
min_parent_records_abs	número	
min_child_records_abs	número	
use_costs	booleano	
costs	estructurado	Propiedad estructurada.
priors	Datos Equal Custom	
custom_priors	estructurado	Propiedad estructurada.
adjust_priors	booleano	
trails	número	Número de modelos de componente para un aumento o agregación autodocimante.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	Regla de combinación predeterminada para objetivos categóricos.
range_ensemble_method	Media Mediana	Regla de combinación predeterminada para objetivos continuos.
large_boost	booleano	Aplicar aumento a conjunto de datos muy grandes.
split_alpha	número	Nivel de significancia para división.
train_pct	número	Conjunto de prevención sobreajustado.
set_random_seed	booleano	Opción replicar resultados.
seed	número	
calculate_variable_importance	booleano	
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	
adjusted_propensity_partition	Test Validation	

## Propiedades del nodo regression



La regresión lineal es una técnica de estadístico común utilizada para resumir datos y realizar predicciones ajustando una superficie o línea recta que minimice las discrepancias existentes entre los valores de salida reales y los predichos.

*Note:* el nodo Lineal reemplazará al nodo Regresión en una versión futura. Recomendamos que a partir de ahora utilice modelos lineales para la regresión lineal.

Tabla 110. Propiedades del nodo regression.

Propiedades del nodo regression	Valores	Descripción de la propiedad
target	field	Los modelos de regresión requieren un único campo objetivo y uno o más campos de entrada. También se puede especificar un campo de ponderación. Consulte el tema “Propiedades comunes de nodos de modelado” en la página 129 para obtener más información.
método	Intro Stepwise Backwards Forwards	
include_constant	booleano	
use_weight	booleano	
weight_field	field	
mode	Simple Expert	
complete_records	booleano	
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	Utilice comillas dobles para los argumentos.
stepping_method	useP useF	useP : utilizar la probabilidad de F useF: utilizar el valor F
probability_entry	número	
probability_removal	número	
F_value_entry	número	
F_value_removal	número	
selection_criteria	booleano	
confidence_interval	booleano	
covariance_matrix	booleano	
collinearity_diagnostics	booleano	

Tabla 110. Propiedades del nodo regression (continuación).

Propiedades del nodo regression	Valores	Descripción de la propiedad
regression_coefficients	booleano	
exclude_fields	booleano	
durbin_watson	booleano	
model_fit	booleano	
r_squared_change	booleano	
p_correlations	booleano	
descriptives	booleano	
calculate_variable_importance	booleano	

## Propiedades del nodo sequence



El nodo Secuencia encuentra reglas de asociación en datos secuenciales o en datos ordenados en el tiempo. Una secuencia es una lista de conjuntos de elementos que tiende a producirse en un orden previsible. Por ejemplo, si un cliente compra una cuchilla y una loción para después del afeitado, probablemente comprará crema para afeitarse la próxima vez que vaya a comprar. El nodo Secuencia se basa en el algoritmo de reglas de asociación de CARMA, que utiliza un método de dos pasos para encontrar las secuencias.

Tabla 111. Propiedades del nodo sequence.

Propiedades del nodo sequence	Valores	Descripción de la propiedad
id_field	field	Para crear un modelo de secuencias, es necesario especificar un campo de ID, un campo de tiempo opcional y uno o varios campos de contenido. Los campos de ponderación y frecuencia no se usan. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
time_field	field	
use_time_field	booleano	
content_fields	[field1 ... fieldn]	
contiguous	booleano	
min_supp	número	
min_conf	número	
max_size	número	
max_predictions	número	
mode	Simple Expert	
use_max_duration	booleano	
max_duration	número	
use_gaps	booleano	
min_item_gap	número	
max_item_gap	número	
use_pruning	booleano	



Tabla 111. Propiedades del nodo *sequence* (continuación).

Propiedades del nodo <i>sequence</i>	Valores	Descripción de la propiedad
<code>pruning_value</code>	<i>número</i>	
<code>set_mem_sequences</code>	<i>booleano</i>	
<code>mem_sequences</code>	<i>entero</i>	

## Propiedades del nodo *slrm*



El nodo Modelo de respuesta de autoaprendizaje (SLRM) permite crear un modelo en el que un solo caso nuevo o un pequeño número de casos nuevos se pueden utilizar para volver a calcular el modelo sin tener que entrenar de nuevo el modelo utilizando todos los datos.

Tabla 112. Propiedades del nodo *slrm*.

Propiedades del nodo <i>slrm</i>	Valores	Descripción de la propiedad
<code>target</code>	<i>field</i>	El campo objetivo debe ser un campo nominal o marca. También se puede especificar un campo de frecuencia. Consulte el tema “Propiedades comunes de nodos de modelado” en la página 129 para obtener más información.
<code>target_response</code>	<i>field</i>	El tipo debe ser marca.
<code>continue_training_existing_model</code>	<i>booleano</i>	
<code>target_field_values</code>	<i>booleano</i>	Utilizar todos: Usar todos los valores del origen.  Especifique: Son necesarios determinados valores.
<code>target_field_values_specify</code>	<i>[campo1 ... campoN]</i>	
<code>include_model_assessment</code>	<i>booleano</i>	
<code>model_assessment_random_seed</code>	<i>número</i>	Debe ser un número real.
<code>model_assessment_sample_size</code>	<i>número</i>	Debe ser un número real.
<code>model_assessment_iterations</code>	<i>número</i>	Número de iteraciones.
<code>display_model_evaluation</code>	<i>booleano</i>	
<code>max_predictions</code>	<i>número</i>	
<code>randomization</code>	<i>número</i>	
<code>scoring_random_seed</code>	<i>número</i>	
<code>sort</code>	Ascending Descending	Especifica si se mostrarán primero las ofertas con las puntuaciones más altas o más bajas.
<code>model_reliability</code>	<i>booleano</i>	
<code>calculate_variable_importance</code>	<i>booleano</i>	

## Propiedades del nodo statisticsmodel



El nodo Modelo Statistics permite analizar y trabajar con sus datos ejecutando los procedimientos de IBM SPSS Statistics que producen PMML. Este nodo requiere una copia de IBM SPSS Statistics con licencia.

Las propiedades de este nodo están descritas en “Propiedades del nodo statisticsmodel” en la página 240.

## Propiedades del nodo svm



El nodo Máquina de vectores de soporte (SVM) le permite clasificar datos en uno o dos grupos sin que haya un ajuste por exceso. SVM funciona bien con conjuntos de datos grandes, como aquellos con un gran número de campos de entrada.

Tabla 113. Propiedades del nodo svm.

Propiedades del nodo svm	Valores	Descripción de la propiedad
all_probabilities	booleano	
stopping_criteria	1.0E-1 1.0E-2 1.0E-3 (valor predeterminado) 1.0E-4 1.0E-5 1.0E-6	Determina cuándo detener el algoritmo de optimización.
regularization	número	También se conoce como el parámetro C.
precision	número	Sólo se utiliza si el nivel de medición del campo objetivo es Continuo.
kernel	RBF (valor predeterminado) Polinómico Sigmoide Lineal	Tipo de función kernel utilizada para la transformación.
rbf_gamma	número	Sólo se utiliza si kernel es RBF.
gamma	número	Sólo se utiliza si kernel es Polinómico o Sigmoide.
bias	número	
grado	número	Sólo se utiliza si kernel es Polinómico.
calculate_variable_importance	booleano	
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	
adjusted_propensity_partition	Test Validation	

## Propiedades del nodo timeseries



El nodo Serie temporal estima modelos de suavizado exponencial, modelos autorregresivos integrados de media móvil (ARIMA) univariados y modelos ARIMA (o de función de transferencia) multivariados para series temporales y genera previsiones. Un nodo Serie temporal debe ir siempre precedido por un nodo Intervalos de tiempo.

Tabla 114. Propiedades del nodo timeseries.

Propiedades del nodo timeseries	Valores	Descripción de la propiedad
targets	<i>field</i>	El nodo Serie temporal prevé uno o más objetivos, utilizando opcionalmente uno o más campos de entrada como predictores. Los campos de frecuencia y ponderación no se usan. Consulte el tema "Propiedades comunes de nodos de modelado" en la página 129 para obtener más información.
continue	<i>booleano</i>	
método	ExpertModeler Exsmooth Arima Reuse	
expert_modeler_method	<i>booleano</i>	
consider_seasonal	<i>booleano</i>	
detect_outliers	<i>booleano</i>	
expert_outlier_additive	<i>booleano</i>	
expert_outlier_level_shift	<i>booleano</i>	
expert_outlier_innovational	<i>booleano</i>	
expert_outlier_level_shift	<i>booleano</i>	
expert_outlier_transient	<i>booleano</i>	
expert_outlier_seasonal_additive	<i>booleano</i>	
expert_outlier_local_trend	<i>booleano</i>	
expert_outlier_additive_patch	<i>booleano</i>	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima_p	<i>entero</i>	
arima_d	<i>entero</i>	
arima_q	<i>entero</i>	

Tabla 114. Propiedades del nodo timeseries (continuación).

Propiedades del nodo timeseries	Valores	Descripción de la propiedad
arima_sp	entero	
arima_sd	entero	
arima_sq	entero	
arima_transformation_type	None SquareRoot NaturalLog	
arima_include_constant	booleano	
tf_arima_p. nombrecampo	entero	Para funciones de transferencia.
tf_arima_d. nombrecampo	entero	Para funciones de transferencia.
tf_arima_q. nombrecampo	entero	Para funciones de transferencia.
tf_arima_sp. nombrecampo	entero	Para funciones de transferencia.
tf_arima_sd. fieldname	entero	Para funciones de transferencia.
tf_arima_sq. nombrecampo	entero	Para funciones de transferencia.
tf_arima_delay. nombrecampo	entero	Para funciones de transferencia.
tf_arima_transformation_type. nombrecampo	None SquareRoot NaturalLog	Para funciones de transferencia.
arima_detect_outlier_mode	None Automatic	
arima_outlier_additive	booleano	
arima_outlier_level_shift	booleano	
arima_outlier_innovational	booleano	
arima_outlier_transient	booleano	
arima_outlier_seasonal_additive	booleano	
arima_outlier_local_trend	booleano	
arima_outlier_additive_patch	booleano	
conf_limit_pct	real	
max_lags	entero	
events	fields	
scoring_model_only	booleano	Se utiliza para modelos con cifras muy grandes (cientos de miles) o series temporales.

## Propiedades del nodo twostep



El nodo Bietápico es un método de agrupación en clústeres de dos pasos. El primer paso es hacer una única pasada por los datos para comprimir los datos de entrada de la fila en un conjunto de subclústeres administrable. El segundo paso utiliza un método de agrupación en clústeres jerárquica para fundir progresivamente los subclústeres en clústeres cada vez más grandes. El bietápico tiene la ventaja de estimar automáticamente el número óptimo de clústeres para los datos de entrenamiento. Puede gestionar tipos de campos mixtos y grandes conjuntos de datos eficazmente.

Tabla 115. Propiedades de nodo twostep.

Propiedades del nodo twostep	Valores	Descripción de la propiedad
inputs	[field1 ... fieldN]	Los modelos bietápicos utilizan una lista de campos de entrada, pero no de campos objetivo. Los campos de ponderación y frecuencia no se reconocen. Consulte el tema “Propiedades comunes de nodos de modelado” en la página 129 para obtener más información.
standardize	booleano	
exclude_outliers	booleano	
percentage	número	
cluster_num_auto	booleano	
min_num_clusters	número	
max_num_clusters	número	
num_clusters	número	
cluster_label	String Number	
label_prefix	cadena	
distance_measure	Euclidean Loglikelihood	
clustering_criterion	AIC BIC	



---

## Capítulo 14. Propiedades de nodos de nugget de modelo

Los nodos de nugget de modelo comparten las mismas propiedades comunes que los otros nodos. Consulte el tema “Propiedades de nodos comunes” en la página 58 para obtener más información.

---

### Propiedades del nodo `applyanomalydetection`

Los nodos de modelado Detección de anomalías pueden utilizarse para generar un nugget de modelo Detección de anomalías. El nombre de script de este nugget de modelo es `applyanomalydetection`. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo `anomalydetection`” en la página 129.

Tabla 116. Propiedades del nodo `applyanomalydetection`.

Propiedades del nodo <code>applyanomalydetection</code>	Valores	Descripción de la propiedad
<code>anomaly_score_method</code>	FlagAndScore FlagOnly ScoreOnly	Determina que resultados se crean para puntuación.
<code>num_fields</code>	<i>entero</i>	Campos para informar.
<code>discard_records</code>	<i>booleano</i>	Indica si los registros se descartan del resultado o no.
<code>discard_anomalous_records</code>	<i>booleano</i>	Indicador de cuando descartar los registros anómalos o <i>no</i> anómalos. El valor predeterminado es <i>off</i> , que significa que se descartan los registros <i>no</i> anómalos. En caso contrario, si es <i>on</i> , se descartan los registros anómalos. Esta propiedad se activa sólo si la propiedad <code>discard_records</code> se activa.

---

### Propiedades del nodo `applyapriori`

Los nodos de modelado Apriori pueden utilizarse para generar un nugget de modelo Apriori. El nombre de script de este nugget de modelo es `applyapriori`. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo `apriori`” en la página 131.

Tabla 117. Propiedades del nodo `applyapriori`.

Propiedades del nodo <code>applyapriori</code>	Valores	Descripción de la propiedad
<code>max_predictions</code>	<i>número (entero)</i>	
<code>ignore_unmatched</code>	<i>booleano</i>	
<code>allow_repeats</code>	<i>booleano</i>	
<code>check_basket</code>	NoPredictions Predictions NoCheck	
<code>criterio</code>	Confianza Soporte RuleSupport Lift Capacidad de despliegue	

---

## Propiedades del nodo applyautoclassifier

Los nodos de modelado de clasificador automático se pueden utilizar para crear un nugget de modelo Clasificador automático. El nombre de script de este nugget de modelo es *applyautoclassifier*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo autoclassifier” en la página 132.

Tabla 118. Propiedades del nodo applyautoclassifier.

Propiedades del nodo applyautoclassifier	Valores	Descripción de la propiedad
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence AverageRawPropensity	Especifica el método utilizado para determinar la puntuación del conjunto. Este conjunto sólo se aplica si el objetivo seleccionado es un campo de marca.
flag_voting_tie_selection	Random HighestConfidence RawPropensity	Si se selecciona un método de votación, especifica cómo se resolverán los empates. Este conjunto sólo se aplica si el objetivo seleccionado es un campo de marca.
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	Especifica el método utilizado para determinar la puntuación del conjunto. Este conjunto sólo se aplica si el objetivo seleccionado es un campo de conjunto.
set_voting_tie_selection	Random HighestConfidence	Si se selecciona un método de votación, especifica cómo se resolverán los empates. Este conjunto sólo se aplica si el objetivo seleccionado es un campo nominal.

---

## Propiedades del nodo applyautocluster

Los nodos de modelado de Clúster automático se pueden utilizar para crear un nugget de modelo Clúster automático. El nombre de script de este nugget de modelo es *applyautocluster*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo autocluster” en la página 134.

---

## Propiedades del nodo applyautonumeric

Los nodos de modelado autonumérico se pueden utilizar para crear un nugget de modelo Autonumérico. El nombre de script de este nugget de modelo es *applyautonumeric*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo autonumeric” en la página 135.

Tabla 119. Propiedades del nodo applyautonumeric.

Propiedades del nodo applyautonumeric	Valores	Descripción de la propiedad
calculate_standard_error	booleano	



---

## Propiedades del nodo applybayesnet

Los nodos de modelado de red bayesiana pueden utilizarse para generar un nugget de modelo de red bayesiana. El nombre de script de este nugget de modelo es *applybayesnet*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo bayesnet” en la página 136.

Tabla 120. Propiedades del nodo *applybayesnet*.

Propiedades del nodo <b>applybayesnet</b>	Valores	Descripción de la propiedad
<code>all_probabilities</code>	<i>booleano</i>	
<code>raw_propensity</code>	<i>booleano</i>	
<code>adjusted_propensity</code>	<i>booleano</i>	
<code>calculate_raw_propensities</code>	<i>booleano</i>	
<code>calculate_adjusted_propensities</code>	<i>booleano</i>	

---

## Propiedades de nodo de applyc50

Los nodos de modelado C5.0 pueden utilizarse para generar un nugget de modelo C5.0. El nombre de script de este nugget de modelo es *applyc50*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo c50” en la página 138.

Tabla 121. Propiedades de nodo de *applyc50*.

Propiedades del nodo <b>applyc50</b>	Valores	Descripción de la propiedad
<code>sql_generate</code>	Nunca NoMissingValues	Se utiliza para establecer las opciones de generación de SQL durante la ejecución del conjunto de reglas.
<code>calculate_conf</code>	<i>booleano</i>	Disponible cuando la generación de SQL está activada. Esta propiedad incluye los cálculos de confianza en el árbol generado.
<code>calculate_raw_propensities</code>	<i>booleano</i>	
<code>calculate_adjusted_propensities</code>	<i>booleano</i>	

---

## Propiedades del nodo applycarma

Los nodos de modelado CARMA pueden utilizarse para generar un nugget de modelo CARMA. El nombre de script de este nugget de modelo es *applycarma*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo carma” en la página 139.

---

## Propiedades del nodo applycart

Se pueden utilizar los nodos de modelado C&RT para generar un nugget de modelo C&RT. El nombre de script de este nugget de modelo es *applycart*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo cart” en la página 140.

Tabla 122. Propiedades del nodo *applycart*.

Propiedades del nodo <b>applycart</b>	Valores	Descripción de la propiedad
<code>sql_generate</code>	Nunca MissingValues NoMissingValues	Se utiliza para establecer las opciones de generación de SQL durante la ejecución del conjunto de reglas.

Tabla 122. Propiedades del nodo applycart (continuación).

Propiedades del nodo applycart	Valores	Descripción de la propiedad
calculate_conf	booleano	Disponible cuando la generación de SQL está activada. Esta propiedad incluye los cálculos de confianza en el árbol generado.
display_rule_id	booleano	Añade un campo en el resultado de puntuación que indica el ID para el nodo terminal al que se asigna cada registro.
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	

## Propiedades del nodo applychaid

Los nodos de modelado CHAID pueden utilizarse para generar un nugget de modelo CHAID. El nombre de script de este nugget de modelo es *applychaid*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo chaid” en la página 141.

Tabla 123. Propiedades del nodo applychaid.

Propiedades del nodo applychaid	Valores	Descripción de la propiedad
sql_generate	Nunca MissingValues	
calculate_conf	booleano	
display_rule_id	booleano	Añade un campo en el resultado de puntuación que indica el ID para el nodo terminal al que se asigna cada registro.
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	

## Propiedades del nodo applycoxreg

Los nodos de modelado Cox pueden utilizarse para generar un nugget de modelo Cox. El nombre de script de este nugget de modelo es *applycoxreg*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo coxreg” en la página 143.

Tabla 124. Propiedades del nodo applycoxreg.

Propiedades del nodo applycoxreg	Valores	Descripción de la propiedad
future_time_as	Intervalos Campos	
time_interval	número	
num_future_times	entero	
time_field	field	
past_survival_time	field	
all_probabilities	booleano	
cumulative_hazard	booleano	

---

## Propiedades del nodo applydecisionlist

Los nodos de modelado Lista de decisiones pueden utilizarse para generar un nugget de modelo Lista de decisiones. El nombre de script de este nugget de modelo es *applydecisionlist*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo decisionlist” en la página 145.

Tabla 125. Propiedades del nodo *applydecisionlist*.

Propiedades del nodo <b>applydecisionlist</b>	Valores	Descripción de la propiedad
enable_sql_generation	<i>booleano</i>	Cuando se establece en true, IBM SPSS Modeler intenta enviar el modelo Lista de decisiones a SQL.
calculate_raw_propensities	<i>booleano</i>	
calculate_adjusted_propensities	<i>booleano</i>	

---

## Propiedades del nodo applydiscriminant

Los nodos de modelado Discriminante pueden utilizarse para generar un nugget de modelo Discriminante. El nombre de script de este nugget de modelo es *applydiscriminant*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo discriminant” en la página 146.

Tabla 126. Propiedades del nodo *applydiscriminant*.

Propiedades del nodo <b>applydiscriminant</b>	Valores	Descripción de la propiedad
calculate_raw_propensities	<i>booleano</i>	
calculate_adjusted_propensities	<i>booleano</i>	

---

## Propiedades del nodo applyfactor

Los nodos de modelado PCA/Factorial pueden utilizarse para generar un nugget de modelo PCA/Factorial. El nombre de script de este nugget de modelo es *applyfactor*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo factor” en la página 147.

---

## Propiedades del nodo applyfeatureselection

Los nodos de modelado Selección de características pueden utilizarse para generar un nugget de modelo Selección de características. El nombre de script de este nugget de modelo es *applyfeatureselection*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo featureselection” en la página 149.

Tabla 127. Propiedades del nodo *applyfeatureselection*.

Propiedades del nodo <b>applyfeatureselection</b>	Valores	Descripción de la propiedad
selected_ranked_fields		especifica qué campos clasificados se comprueban en el explorador de modelos.
selected_screened_fields		Especifica qué campos filtrados se comprueban en el explorador de modelos.

---

## Propiedades del nodo applygeneralizedlinear

Los nodos de modelado lineal generalizado (genlin) pueden utilizarse para generar un nugget de modelo lineal generalizado. El nombre de script de este nugget de modelo es *applygeneralizedlinear*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo genlin” en la página 150.

Tabla 128. Propiedades del nodo applygeneralizedlinear.

Propiedades del nodo applygeneralizedlinear	Valores	Descripción de la propiedad
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	

---

## Propiedades del nodo applyglm

Los nodos de modelado GLMM pueden utilizarse para generar un nugget de modelo GLMM. El nombre de script de este nugget de modelo es *applyglm*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo glm” en la página 153.

Tabla 129. Propiedades del nodo applyglm.

Propiedades del nodo applyglm	Valores	Descripción de la propiedad
confidence	onProbability onIncrease	Base para calcular el valor de confianza de la puntuación: probabilidad más alta predicha, o la diferencia entre la probabilidad más alta predicha y la segunda probabilidad más alta.
score_category_probabilities	booleano	Si es True, genera las probabilidades predichas para objetivos categóricos. Se crea un campo para cada categoría. El valor predeterminado es False.
max_categories	entero	Número máximo de categorías para el que se van a predecir las probabilidades. Sólo se utiliza si score_category_probabilities es True.
score_propensity	booleano	Si se establece en True, genera puntuaciones de propensión en bruto (probabilidad de resultado "True") para modelos con objetivos de marca. Si las particiones están en vigor, también genera puntuaciones de propensión ajustadas en función de la partición de prueba. El valor predeterminado es False.

---

## Propiedades del nodo applykmeans

Los nodos de modelado K-medias pueden utilizarse para generar un nugget de modelo K-medias. El nombre de script de este nugget de modelo es *applykmeans*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo kmeans” en la página 157.

---

## Propiedades del nodo applyknn

Los nodos de modelado KNN pueden utilizarse para generar un nugget de modelo KNN. El nombre de script de este nugget de modelo es *applyknn*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo knn” en la página 158.

Tabla 130. Propiedades del nodo applyknn.

Propiedades del nodo applyknn	Valores	Descripción de la propiedad
all_probabilities	booleano	
save_distances	booleano	

---

## Propiedades del nodo applykohonen

Los nodos de modelado Kohonen pueden utilizarse para generar un nugget de modelo Kohonen. El nombre de script de este nugget de modelo es *applykohonen*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo kohonen” en la página 159.

---

## Propiedades del nodo applylinear

Los nodos de modelado lineal pueden utilizarse para generar un nugget de modelo lineal. El nombre de script de este nugget de modelo es *applylinear*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo linear” en la página 160.

Tabla 131. Propiedades del nodo applylinear.

Propiedades del nodo applylinear	Valores	Descripción de la propiedad
use_custom_name	booleano	
custom_name	cadena	
enable_sql_generation	booleano	

---

## Propiedades del nodo applylogreg

Los nodos de modelado Regresión logística pueden utilizarse para generar un nugget de modelo Regresión logística. El nombre de script de este nugget de modelo es *applylogreg*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo logreg” en la página 161.

Tabla 132. *applylogreg* node properties.

Propiedades del nodo applylogreg	Valores	Descripción de la propiedad
calculate_raw_propensities	booleano	
calculate_conf	booleano	
enable_sql_generation	booleano	

---

## Propiedades del nodo applyneuralnet

Los nodos de modelado Red neuronal pueden utilizarse para generar un nugget de modelo Red neuronal. El nombre de script de este nugget de modelo es *applyneuralnet*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo neuralnet” en la página 164.

**Precaución:** Una versión más reciente del nodo de modelado Red neuronal, con características mejoradas, está disponible en esta versión y se describe en la sección siguiente (*applyneuralnetwork*). Aunque la versión anterior sigue estando disponible, le recomendamos actualizar sus scripts para que se usen la nueva versión. En este documento se incluyen detalles de la versión anterior como referencia, pero en versiones futuras dejará de ser compatible.

Tabla 133. Propiedades del nodo *applyneuralnet*.

Propiedades del nodo <i>applyneuralnet</i>	Valores	Descripción de la propiedad
calculate_conf	booleano	Disponible cuando la generación de SQL está activada. Esta propiedad incluye los cálculos de confianza en el árbol generado.
enable_sql_generation	booleano	
nn_score_method	Diferencia SoftMax	
calculate_raw_propensities	booleano	
calculate_adjusted_propensities	booleano	

## Propiedades del nodo *applyneuralnetwork*

Los nodos de modelado Red neuronal pueden utilizarse para generar un nugget de modelo Red neuronal. El nombre de script de este nugget de modelo es *applyneuralnetwork*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo *neuralnetwork*” en la página 166.

Tabla 134. Propiedades del nodo *applyneuralnetwork*.

Propiedades del nodo <i>applyneuralnetwork</i>	Valores	Descripción de la propiedad
use_custom_name	booleano	
custom_name	cadena	
confidence	onProbability onIncrease	
score_category_probabilities	booleano	
max_categories	número	
score_propensity	booleano	

## Propiedades del nodo *applyquest*

Los nodos de modelado QUEST pueden utilizarse para generar un nugget de modelo QUEST. El nombre de script de este nugget de modelo es *applyquest*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo *quest*” en la página 167.

Tabla 135. Propiedades del nodo *applyquest*.

Propiedades del nodo <i>applyquest</i>	Valores	Descripción de la propiedad
sql_generate	Nunca MissingValues NoMissingValues	
calculate_conf	booleano	

Tabla 135. Propiedades del nodo *applyquest* (continuación).

Propiedades del nodo <i>applyquest</i>	Valores	Descripción de la propiedad
<code>display_rule_id</code>	<i>booleano</i>	Añade un campo en el resultado de puntuación que indica el ID para el nodo terminal al que se asigna cada registro.
<code>calculate_raw_propensities</code>	<i>booleano</i>	
<code>calculate_adjusted_propensities</code>	<i>booleano</i>	

## Propiedades del nodo *applyregression*

Los nodos de modelado Regresión lineal pueden utilizarse para generar un nugget de modelo Regresión lineal. El nombre de script de este nugget de modelo es *applyregression*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo *regression*” en la página 169.

## Propiedades del nodo *applyr*

Los nodos de modelado R pueden utilizarse para generar un nugget de modelo R. El nombre de script de este nugget de modelo es *applyr*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo *buildr*” en la página 137.

Tabla 136. Propiedades del nodo *applyr*

Propiedades del nodo <i>applyr</i>	Valores	Descripción de la propiedad
<code>score_syntax</code>	<i>cadena</i>	Sintaxis de scripts R para la puntuación del modelo.
<code>convert_flags</code>	StringsAndDoubles LogicalValues	Opción para convertir campos de distintivos.
<code>convert_datetime</code>	<i>booleano</i>	Opción para convertir las variables con los formatos de fecha o de fecha y hora para formatos de fecha/hora R.
<code>convert_datetime_class</code>	POSIXct POSIXlt	Opciones para especificar a qué formato se convierten las variables con los formatos de fecha o de fecha y hora.
<code>convert_missing</code>	<i>booleano</i>	Opción para convertir los valores que faltan al valor R NA.

## Propiedades del nodo *applyselflearning*

Los nodos de modelado de modelo de respuesta de autoaprendizaje (SLRM) pueden utilizarse para generar un nugget de modelo SLRM. El nombre de script de este nugget de modelo es *applyselflearning*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo *slrm*” en la página 171.

Tabla 137. Propiedades del nodo *applyselflearning*.

Propiedades del nodo <i>applyselflearning</i>	Valores	Descripción de la propiedad
<code>max_predictions</code>	<i>número</i>	
<code>randomization</code>	<i>número</i>	
<code>scoring_random_seed</code>	<i>número</i>	

Tabla 137. Propiedades del nodo *applyselflearning* (continuación).

Propiedades del nodo <i>applyselflearning</i>	Valores	Descripción de la propiedad
sort	ascending descending	Especifica si se mostrarán primero las ofertas con las puntuaciones más altas o más bajas.
model_reliability	<i>booleano</i>	Tiene en cuenta la opción de fiabilidad del modelo de la pestaña Configuración.

## Propiedades del nodo *applysequence*

Los nodos de modelado Secuencia pueden utilizarse para generar un nugget de modelo Secuencia. El nombre de script de este nugget de modelo es *applysequence*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo *sequence*” en la página 170.

## Propiedades del nodo *applysvm*

Los nodos de modelado SVM pueden utilizarse para generar un nugget de modelo SVM. El nombre de script de este nugget de modelo es *applysvm*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo *svm*” en la página 172.

Tabla 138. Propiedades del nodo *applysvm*.

Propiedades del nodo <i>applysvm</i>	Valores	Descripción de la propiedad
all_probabilities	<i>booleano</i>	
calculate_raw_propensities	<i>booleano</i>	
calculate_adjusted_propensities	<i>booleano</i>	

## Propiedades del nodo *applytimeseries*

Los nodos de modelado Serie temporal pueden utilizarse para generar un nugget de modelo Serie temporal. El nombre de script de este nugget de modelo es *applytimeseries*. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo *timeseries*” en la página 173.

Tabla 139. Propiedades del nodo *applytimeseries*.

Propiedades del nodo <i>applytimeseries</i>	Valores	Descripción de la propiedad
calculate_conf	<i>booleano</i>	
calculate_residuals	<i>booleano</i>	

## Propiedades del nodo *applytwostep*

Los nodos de modelado Bietápico pueden utilizarse para generar un nugget de modelo Bietápico. El nombre de script de este nugget de modelo es *applytwostep*. No existe ninguna otra propiedad para este nugget de modelo. Para obtener más información sobre los scripts para el propio nodo de modelado, consulte el tema “Propiedades del nodo *twostep*” en la página 175.



---

## Capítulo 15. Propiedades de nodos de modelado de bases de datos

IBM SPSS Modeler admite la integración con herramientas de modelado y minería de datos ofrecidas por proveedores de bases de datos, incluidos Microsoft SQL Server Analysis Services, Oracle Data Mining, IBM DB2 InfoSphere Warehouse, y IBM Netezza Analytics. Podrá crear y almacenar modelos mediante algoritmos nativos de bases de datos, todo desde la aplicación IBM SPSS Modeler. Los modelos de base de datos también se pueden crear y manipular a través de scripts utilizando las propiedades descritas en esta sección.

---

### Propiedades de nodos de modelado de Microsoft

#### Propiedades de nodos de modelado de Microsoft

Propiedades comunes

Las siguientes propiedades son comunes a los nodos de modelado de bases de datos de Microsoft.

Tabla 140. Propiedades comunes de nodos de Microsoft.

Propiedades comunes de nodo de Microsoft	Valores	Descripción de la propiedad
analysis_database_name	<i>cadena</i>	Nombre de la base de datos de Analysis Services.
analysis_server_name	<i>cadena</i>	Nombre del host de Analysis Services.
use_transactional_data	<i>booleano</i>	Especifica si los datos de entrada están en formato tabular o transaccional.
inputs	<i>[campo campo campo]</i>	Campos de entrada de datos tabulares.
target	<i>field</i>	Campo predicho (no aplicable a nodo Clúster de MS o nodos de Agrupación en clústeres de secuencias).
unique_field	<i>field</i>	Campos clave.
msas_parameters	<i>estructurado</i>	Parámetros del algoritmo. Consulte el tema "Parámetros del algoritmo" en la página 188 para obtener más información.
with_drillthrough	<i>booleano</i>	Opción Con exploración.

Árbol de decisión de MS

No hay propiedades específicas para los nodos del tipo *mstree*. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

Clúster de MS

No hay propiedades específicas para los nodos del tipo *mscluster*. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

Reglas de asociación de MS

Las siguientes propiedades específicas están disponibles para los nodos del tipo *msassoc*:

Tabla 141. Propiedades del nodo msassoc.

Propiedades del nodo msassoc	Valores	Descripción de la propiedad
id_field	field	Identifica todas las transacciones en los datos.
trans_inputs	[campo campo campo]	Los campos de entrada de datos transaccionales.
transactional_target	field	Campo predicho (datos transaccionales).

#### Bayesiano ingenuo de MS

No hay propiedades específicas para los nodos del tipo msbayes. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

#### Regresión lineal de MS

No hay propiedades específicas para los nodos del tipo msregression. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

#### Red neuronal de MS

No hay propiedades específicas para los nodos del tipo msneuralnetwork. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

#### Regresión logística de MS

No hay propiedades específicas para los nodos del tipo mslogistic. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

#### Series temporales de MS

No hay propiedades específicas para los nodos del tipo mstimeseries. Consulte las propiedades comunes de Microsoft que se indican al comienzo de esta sección.

#### Clúster de secuencias de MS

Las siguientes propiedades específicas están disponibles para los nodos del tipo mssequencecluster:

Tabla 142. Propiedades del nodo mssequencecluster.

Propiedades del nodo mssequencecluster	Valores	Descripción de la propiedad
id_field	field	Identifica todas las transacciones en los datos.
input_fields	[campo campo campo]	Los campos de entrada de datos transaccionales.
sequence_field	field	Identificador de secuencia.
target_field	field	Campo predicho (datos tabulares).

## Parámetros del algoritmo

Cada tipo de modelo de base de datos de Microsoft tiene parámetros concretos que se pueden establecer mediante la propiedad msas\_parameters.

Estos parámetros se derivan de SQL Server. Para ver los parámetros relevantes para cada nodo:

1. Coloque un nodo de origen de base de datos en el lienzo.

2. Abra el nodo de origen de base de datos.
3. Seleccione un origen válido en la lista desplegable **Origen de datos**.
4. Seleccione una tabla válida en la lista **Nombre de tabla**.
5. Pulse en **Aceptar** para cerrar el nodo de origen de base de datos.
6. Conecte un nodo de modelado de bases de datos de Microsoft cuyas propiedades desee conocer.
7. Abra el nodo de modelado de bases de datos.
8. Seleccione la pestaña **Experto**.

Aparecerán las propiedades `msas_parameters` disponibles de este nodo.

## Propiedades de nugget de modelo de Microsoft

Las siguientes propiedades son para los nugget de modelo creados mediante los nodos de modelado de bases de datos de Microsoft.

Árbol de decisión de MS

Tabla 143. Propiedades de Árbol de decisión de MS.

Propiedades del nodo <code>appliedtree</code>	Valores	Descripción
<code>analysis_database_name</code>	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta.  Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
<code>analysis_server_name</code>	<i>cadena</i>	Nombre del host del servidor de análisis.
<code>datasource</code>	<i>cadena</i>	Nombre del origen de datos (DSN) ODBC de SQL Server.
<code>sql_generate</code>	<i>booleano</i>	Activa la generación de SQL.

Regresión lineal de MS

Tabla 144. Propiedades de Regresión lineal de MS.

Propiedades del nodo <code>appliedregression</code>	Valores	Descripción
<code>analysis_database_name</code>	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta.  Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
<code>analysis_server_name</code>	<i>cadena</i>	Nombre del host del servidor de análisis.

Red neuronal de MS

Tabla 145. Propiedades de Red neuronal de MS.

Propiedades del nodo <code>appliedneuralnetwork</code>	Valores	Descripción
<code>analysis_database_name</code>	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta.  Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
<code>analysis_server_name</code>	<i>cadena</i>	Nombre del host del servidor de análisis.

Regresión logística de MS

Tabla 146. Propiedades de Regresión logística de MS.

Propiedades del nodo <b>applieslogistic</b>	Valores	Descripción
analysis_database_name	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta.  Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
analysis_server_name	<i>cadena</i>	Nombre del host del servidor de análisis.

## Series temporales de MS

Tabla 147. Propiedades de MS Time Series.

Propiedades del nodo <b>appliestimeseries</b>	Valores	Descripción
analysis_database_name	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta.  Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
analysis_server_name	<i>cadena</i>	Nombre del host del servidor de análisis.
start_from	new_prediction historical_prediction	Especifica si se realizarán predicciones futuras o históricas.
new_step	<i>número</i>	Define el período de tiempo inicial de predicciones futuras.
historical_step	<i>número</i>	Define el período de tiempo inicial de predicciones históricas.
end_step	<i>número</i>	Define el período de tiempo final de las predicciones.

## Clúster de secuencias de MS

Tabla 148. Propiedades de Agrupación en clústeres de secuencias de MS.

Propiedades del nodo <b>appliessequencecluster</b>	Valores	Descripción
analysis_database_name	<i>cadena</i>	Este nodo se puede puntuar directamente en una ruta.  Esta propiedad se utiliza para identificar el nombre de la base de datos de Analysis Services.
analysis_server_name	<i>cadena</i>	Nombre del host del servidor de análisis.

## Propiedades de nodos de modelado de Oracle

### Propiedades de nodos de modelado de Oracle

Las siguientes propiedades son comunes a los nodos de modelado de bases de datos de Oracle.

Tabla 149. Propiedades comunes de nodos de Oracle.

Propiedades comunes de nodos de Oracle	Valores	Descripción de la propiedad
target	<i>field</i>	
inputs	<i>Lista de campos</i>	
partición	<i>field</i>	Campo usado para dividir los datos en muestras independientes para las fases de entrenamiento, comprobación y validación en la generación del modelo.
datasource		
username		
password		
epassword		
use_model_name	<i>booleano</i>	
model_name	<i>cadena</i>	Nombre personalizado para nuevo modelo.
use_partitioned_data	<i>booleano</i>	Si se ha definido un campo de partición, esta opción garantiza que sólo se utilizarán los datos de la partición de entrenamiento para la generación del modelo.
unique_field	<i>field</i>	
auto_data_prep	<i>booleano</i>	Activa o desactiva la característica de preparación de datos automática de Oracle (solamente para bases de datos 11g).
costs	<i>estructurado</i>	Propiedad estructurada.
mode	Simple Expert	Hace que se ignoren ciertas propiedades si se establece como Simple, como se indica en las propiedades de nodos individuales.
use_prediction_probability	<i>booleano</i>	
prediction_probability	<i>cadena</i>	
use_prediction_set	<i>booleano</i>	

### Bayesiano ingenuo de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo oranb.

Tabla 150. Propiedades del nodo oranb.

Propiedades del nodo oranb	Valores	Descripción de la propiedad
singleton_threshold	<i>número</i>	0.0–1.0.*
pairwise_threshold	<i>número</i>	0.0–1.0.*
priors	Datos Equal Custom	
custom_priors	<i>estructurado</i>	Propiedad estructurada.

\* Propiedad ignorada si Modo se establece como Simple.

#### Bayesiano adaptativo de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo oraabn.

Tabla 151. Propiedades del nodo oraabn.

Propiedades del nodo oraabn	Valores	Descripción de la propiedad
model_type	SingleFeature MultiFeature NaiveBayes	
use_execution_time_limit	booleano	*
execution_time_limit	entero	El valor debe ser mayor que 0.*
max_naive_bayes_predictors	entero	El valor debe ser mayor que 0.*
max_predictors	entero	El valor debe ser mayor que 0.*
priors	Datos Equal Custom	
custom_priors	estructurado	Propiedad estructurada.

\* Propiedad ignorada si Modo se establece como Simple.

#### Máquinas de vectores de soporte de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo orasvm.

Tabla 152. Propiedades del nodo orasvm.

Propiedades del nodo orasvm	Valores	Descripción de la propiedad
active_learning	Enable Disable	
kernel_function	Lineal Gaussian Sistema	
normalization_method	zscore minmax ninguno	
kernel_cache_size	entero	Solamente kernel gaussiano. El valor debe ser mayor que 0.*
convergence_tolerance	número	El valor debe ser mayor que 0.*
use_standard_deviation	booleano	Solamente kernel gaussiano.*
standard_deviation	número	El valor debe ser mayor que 0.*
use_epsilon	booleano	Solamente modelos de regresión.*
epsilon	número	El valor debe ser mayor que 0.*
use_complexity_factor	booleano	*
complexity_factor	número	*
use_outlier_rate	booleano	Solamente variantes de una clase.*
outlier_rate	número	Solamente variantes de una clase. 0.0–1.0.*

Tabla 152. Propiedades del nodo orasvm (continuación).

Propiedades del nodo orasvm	Valores	Descripción de la propiedad
weights	Datos Equal Custom	
custom_weights	<i>estructurado</i>	Propiedad estructurada.

\* Propiedad ignorada si Modo se establece como Simple.

#### Modelos lineales generalizados de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo oraglm.

Tabla 153. Propiedades del nodo oraglm.

Propiedades del nodo oraglm	Valores	Descripción de la propiedad
normalization_method	zscore minmax ninguno	
missing_value_handling	ReplaceWithMean UseCompleteRecords	
use_row_weights	<i>booleano</i>	*
row_weights_field	<i>field</i>	*
save_row_diagnostics	<i>booleano</i>	*
row_diagnostics_table	<i>cadena</i>	*
coefficient_confidence	<i>número</i>	*
use_reference_category	<i>booleano</i>	*
reference_category	<i>cadena</i>	*
ridge_regression	Auto Off On	*
parameter_value	<i>número</i>	*
vif_for_ridge	<i>booleano</i>	*

\* Propiedad ignorada si Modo se establece como Simple.

#### Árbol de decisión de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo oradecisiontree.

Tabla 154. Propiedades del nodo oradecisiontree.

Propiedades del nodo oradecisiontree	Valores	Descripción de la propiedad
use_costs	<i>booleano</i>	
impurity_metric	Entropy Gini	
term_max_depth	<i>entero</i>	2–20.*
term_minpct_node	<i>número</i>	0.0–10.0.*
term_minpct_split	<i>número</i>	0.0–20.0.*

Tabla 154. Propiedades del nodo oradecisiontree (continuación).

Propiedades del nodo oradecisiontree	Valores	Descripción de la propiedad
term_minrec_node	entero	El valor debe ser mayor que 0.*
term_minrec_split	entero	El valor debe ser mayor que 0.*
display_rule_ids	booleano	*

\* Propiedad ignorada si Modo se establece como Simple.

#### O-clúster de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo oracluster.

Tabla 155. Propiedades del nodo oracluster.

Propiedades del nodo oracluster	Valores	Descripción de la propiedad
max_num_clusters	entero	El valor debe ser mayor que 0.
max_buffer	entero	El valor debe ser mayor que 0.*
sensitivity	número	0.0–1.0.*

\* Propiedad ignorada si Modo se establece como Simple.

#### K-medias de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo orakmeans.

Tabla 156. Propiedades del nodo orakmeans.

Propiedades del nodo orakmeans	Valores	Descripción de la propiedad
num_clusters	entero	El valor debe ser mayor que 0.
normalization_method	zscore minmax ninguno	
distance_function	Euclidean Cosine	
iteraciones	entero	0–20.*
conv_tolerance	número	0.0–0.5.*
split_criterion	Variance Size	El valor predeterminado es Variance.*
num_bins	entero	El valor debe ser mayor que 0.*
block_growth	entero	1–5.*
min_pct_attr_support	número	0.0–1.0.*

\* Propiedad ignorada si Modo se establece como Simple.

#### NMF de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo oranmf.



Tabla 157. Propiedades del nodo oranmf.

Propiedades del nodo oranmf	Valores	Descripción de la propiedad
normalization_method	minmax ninguno	
use_num_features	booleano	*
num_features	entero	0-1. El algoritmo estima el valor predeterminado a partir de los datos.*
random_seed	número	*
num_iterations	entero	0-500.*
conv_tolerance	número	0.0-0.5.*
display_all_features	booleano	*

\* Propiedad ignorada si Modo se establece como Simple.

### Apriori de Oracle

Las siguientes propiedades están disponibles para los nodos del tipo oraapriori.

Tabla 158. Propiedades del nodo oraapriori.

Propiedades del nodo oraapriori	Valores	Descripción de la propiedad
content_field	field	
id_field	field	
max_rule_length	entero	2-20.
min_confidence	número	0.0-1.0.
min_support	número	0.0-1.0.
use_transactional_data	booleano	

### Longitud mínima de la descripción de Oracle (LMD)

No hay propiedades específicas para los nodos del tipo oramd1. Consulte las propiedades comunes de Oracle que se indican al comienzo de esta sección.

### Importancia del atributo de Oracle (AI)

Las siguientes propiedades están disponibles para los nodos del tipo oraai.

Tabla 159. Propiedades del nodo oraai.

Propiedades del nodo oraai	Valores	Descripción de la propiedad
custom_fields	booleano	Si es verdadero, le permite especificar el objetivo, la entrada y otros campos del nodo actual. Si es falso, se utiliza la configuración actual de un nodo Tipo situado en un punto anterior de la ruta.
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	booleano	Si selection_mode se establece en ImportanceLevel, determina si se seleccionan los campos importantes.
important_label	cadena	Especifica la etiqueta para la clasificación como "important".

Tabla 159. Propiedades del nodo *oraai* (continuación).

Propiedades del nodo <i>oraai</i>	Valores	Descripción de la propiedad
<code>select_marginal</code>	<i>booleano</i>	Si <code>selection_mode</code> se establece en <code>ImportanceLevel</code> , determina si se seleccionan los campos marginales.
<code>marginal_label</code>	<i>cadena</i>	Especifica la etiqueta para la clasificación como "marginal".
<code>important_above</code>	<i>número</i>	0.0–1.0.
<code>select_unimportant</code>	<i>booleano</i>	Si <code>selection_mode</code> se establece en <code>ImportanceLevel</code> , determina si se seleccionan los campos sin importancia.
<code>unimportant_label</code>	<i>cadena</i>	Especifica la etiqueta para la clasificación como "unimportant".
<code>unimportant_below</code>	<i>número</i>	0.0–1.0.
<code>importance_value</code>	<i>número</i>	Si <code>selection_mode</code> se establece en <code>ImportanceValue</code> , determina el valor de corte que se va a usar. Acepta valores de 0 a 100.
<code>top_n</code>	<i>número</i>	Si <code>selection_mode</code> se establece en <code>TopN</code> , determina el valor de corte que se va a usar. Acepta valores de 0 a 1000.

## Propiedades de nugget de modelo de Oracle

Las siguientes propiedades son para los nugget de modelo creados mediante los modelos de Oracle.

Bayesiano ingenuo de Oracle

No hay propiedades específicas para los nodos del tipo `applyoranb`.

Bayesiano adaptativo de Oracle

No hay propiedades específicas para los nodos del tipo `applyoraabn`.

Máquinas de vectores de soporte de Oracle

No hay propiedades específicas para los nodos del tipo `applyorasvm`.

Árbol de decisión de Oracle

Nas siguientes propiedades están disponibles para los nodos del tipo `applyoradecisiontree`.

Tabla 160. Propiedades del nodo *applyoradecisiontree*.

Propiedades del nodo <i>applyoradecisiontree</i>	Valores	Descripción de la propiedad
<code>use_costs</code>	<i>booleano</i>	
<code>display_rule_ids</code>	<i>booleano</i>	

O-clúster de Oracle

No hay propiedades específicas para los nodos del tipo `applyoraocluster`.

K-medias de Oracle

No hay propiedades específicas para los nodos del tipo `applyorakmeans`.

NMF de Oracle

La siguiente propiedad está disponible para los nodos del tipo `applyoranmf`.

Tabla 161. Propiedades del nodo `applyoranmf`.

Propiedades del nodo <code>applyoranmf</code>	Valores	Descripción de la propiedad
<code>display_all_features</code>	<i>booleano</i>	

Apriori de Oracle

Este nugget de modelo no se puede aplicar en los scripts.

LMD de Oracle

Este nugget de modelo no se puede aplicar en los scripts.

---

## Propiedades de nodos para modelado de IBM DB2

### Propiedades de nodos de modelado de IBM DB2

Las siguientes propiedades son comunes a los nodos de modelado de bases de datos de IBM InfoSphere Warehouse (ISW).

Tabla 162. Propiedades comunes de nodo de ISW.

Propiedades comunes de nodo de ISW	Valores	Descripción de la propiedad
<code>inputs</code>	<i>Lista de campos</i>	
<code>datasource</code>		
<code>username</code>		
<code>password</code>		
<code>epassword</code>		
<code>enable_power_options</code>	<i>booleano</i>	
<code>power_options_max_memory</code>	<i>entero</i>	El valor debe ser mayor que 32.
<code>power_options_cmdline</code>	<i>cadena</i>	
<code>mining_data_custom_sql</code>	<i>cadena</i>	
<code>logical_data_custom_sql</code>	<i>cadena</i>	
<code>mining_settings_custom_sql</code>		

Árbol de decisión de ISW

Las siguientes propiedades están disponibles para los nodos del tipo `db2imtree`.

Tabla 163. Propiedades del nodo `db2imtree`.

Propiedades del nodo <code>db2imtree</code>	Valores	Descripción de la propiedad
<code>target</code>	<i>field</i>	
<code>perform_test_run</code>	<i>booleano</i>	
<code>use_max_tree_depth</code>	<i>booleano</i>	
<code>max_tree_depth</code>	<i>entero</i>	Valor mayor que 0.

Tabla 163. Propiedades del nodo db2imtree (continuación).

Propiedades del nodo db2imtree	Valores	Descripción de la propiedad
use_maximum_purity	booleano	
maximum_purity	número	Número entre 0 y 100.
use_minimum_internal_cases	booleano	
minimum_internal_cases	entero	Valor mayor que 1.
use_costs	booleano	
costs	estructurado	Propiedad estructurada.

### Asociación de ISW

Las siguientes propiedades están disponibles para los nodos del tipo db2imassoc.

Tabla 164. Propiedades del nodo db2imassoc.

Propiedades del nodo db2imassoc	Valores	Descripción de la propiedad
use_transactional_data	booleano	
id_field	field	
content_field	field	
data_table_layout	básico limited_length	
max_rule_size	entero	El valor debe ser mayor que 2.
min_rule_support	número	0–100%
min_rule_confidence	número	0–100%
use_item_constraints	booleano	
item_constraints_type	Incluir Exclude	
use_taxonomy	booleano	
taxonomy_table_name	cadena	Nombre de la tabla de DB2 para almacenar los detalles de taxonomía.
taxonomy_child_column_name	cadena	Nombre de la columna hija de la tabla de taxonomía. La columna hija contiene los nombres de los elementos o de las categorías.
taxonomy_parent_column_name	cadena	Nombre de la columna padre en la tabla de taxonomía. La columna padre contiene los nombres de las categorías.
load_taxonomy_to_table	booleano	Comprueba si la información sobre taxonomía almacenada en IBM SPSS Modeler se debe cargar en la tabla de taxonomía en el momento de la generación del modelo. Tenga en cuenta que si la tabla de taxonomía ya existe, se eliminará. La información de taxonomía se almacena con el nodo de generación de modelos y se puede editar mediante los botones <b>Editar categorías</b> y <b>Editar taxonomía</b> .

### Secuencia de ISW

Las siguientes propiedades están disponibles para los nodos del tipo db2imsequence.

Tabla 165. Propiedades del nodo db2imsequence.

Propiedades del nodo db2imsequence	Valores	Descripción de la propiedad
id_field	field	
group_field	field	
content_field	field	
max_rule_size	entero	El valor debe ser mayor que 2.
min_rule_support	número	0–100%
min_rule_confidence	número	0–100%
use_item_constraints	booleano	
item_constraints_type	Incluir Exclude	
use_taxonomy	booleano	
taxonomy_table_name	cadena	Nombre de la tabla de DB2 para almacenar los detalles de taxonomía.
taxonomy_child_column_name	cadena	Nombre de la columna hija de la tabla de taxonomía. La columna hija contiene los nombres de los elementos o de las categorías.
taxonomy_parent_column_name	cadena	Nombre de la columna padre en la tabla de taxonomía. La columna padre contiene los nombres de las categorías.
load_taxonomy_to_table	booleano	Comprueba si la información sobre taxonomía almacenada en IBM SPSS Modeler se debe cargar en la tabla de taxonomía en el momento de la generación del modelo. Tenga en cuenta que si la tabla de taxonomía ya existe, se eliminará. La información de taxonomía se almacena con el nodo de generación de modelos y se puede editar mediante los botones <b>Editar categorías</b> y <b>Editar taxonomía</b> .

## Regresión de ISW

Las siguientes propiedades están disponibles para los nodos del tipo db2imreg.

Tabla 166. Propiedades del nodo db2imreg.

Propiedades del nodo db2imreg	Valores	Descripción de la propiedad
target	field	
regression_method	transform lineal polynomial rbf	Consulte en la tabla siguiente propiedades que sólo se aplican si regression_method se establece en rbf.
perform_test_run	field	
limit_rsquared_value	booleano	
max_rsquared_value	número	Valor entre 0,0 y 1,0.
use_execution_time_limit	booleano	
execution_time_limit_mins	entero	Valor mayor que 0.
use_max_degree_polynomial	booleano	
max_degree_polynomial	entero	

Tabla 166. Propiedades del nodo db2imreg (continuación).

Propiedades del nodo db2imreg	Valores	Descripción de la propiedad
use_intercept	booleano	
use_auto_feature_selection_method	booleano	
auto_feature_selection_method	normal adjusted	
use_min_significance_level	booleano	
min_significance_level	número	
use_min_significance_level	booleano	

Las siguientes propiedades solo se aplican si regression\_method se establece en rbf.

Tabla 167. Propiedades del nodo db2imreg si regression\_method se establece en rbf.

Propiedades del nodo db2imreg	Valores	Descripción de la propiedad
use_output_sample_size	booleano	Si es verdadero, el valor se ajusta automáticamente como el valor predeterminado.
output_sample_size	entero	El valor predeterminado es 2.  El mínimo es 1.
use_input_sample_size	booleano	Si es verdadero, el valor se ajusta automáticamente como el valor predeterminado.
input_sample_size	entero	El valor predeterminado es 2.  El mínimo es 1.
use_max_num_centers	booleano	Si es verdadero, el valor se ajusta automáticamente como el valor predeterminado.
max_num_centers	entero	El valor predeterminado es 20.  El mínimo es 1.
use_min_region_size	booleano	Si es verdadero, el valor se ajusta automáticamente como el valor predeterminado.
min_region_size	entero	El valor por omisión es 15.  El mínimo es 1.
use_max_data_passes	booleano	Si es verdadero, el valor se ajusta automáticamente como el valor predeterminado.
max_data_passes	entero	El valor por omisión es de 5.  El valor mínimo es 2.
use_min_data_passes	booleano	Si es verdadero, el valor se ajusta automáticamente como el valor predeterminado.
min_data_passes	entero	El valor por omisión es de 5.  El valor mínimo es 2.

### Agrupación en clústeres de ISW

Las siguientes propiedades están disponibles para los nodos del tipo db2imcluster.

Tabla 168. Propiedades del nodo db2imcluster.

Propiedades del nodo db2imcluster	Valores	Descripción de la propiedad
cluster_method	demográfico kohonen birch	
kohonen_num_rows	entero	
kohonen_num_columns	entero	
kohonen_passes	entero	
use_num_passes_limit	booleano	
use_num_clusters_limit	booleano	
max_num_clusters	entero	Valor mayor que 1.
birch_dist_measure	log_likelihood euclidean	El valor predeterminado es log_likelihood.
birch_num_cfleaves	entero	El valor predeterminado es 1000.
birch_num_refine_passes	entero	El valor predeterminado es 3; el mínimo es 1.
use_execution_time_limit	booleano	
execution_time_limit_mins	entero	Valor mayor que 0.
min_data_percentage	número	0-100%
use_similarity_threshold	booleano	
similarity_threshold	número	Valor entre 0,0 y 1,0.

### Bayesiano ingenuo ISW

Las siguientes propiedades están disponibles para los nodos del tipo db2imnbs.

Tabla 169. Propiedades del nodo db2imnb.

Propiedades del nodo db2imnb	Valores	Descripción de la propiedad
perform_test_run	booleano	
probability_threshold	número	El valor predeterminado es 0.001. El valor mínimo es 0; el valor máximo es 1.000.
use_costs	booleano	
costs	estructurado	Propiedad estructurada.

### Regresión logística ISW

Las siguientes propiedades están disponibles para los nodos del tipo db2imlog.

Tabla 170. Propiedades del nodo db2imlog.

Propiedades del nodo db2imlog	Valores	Descripción de la propiedad
perform_test_run	booleano	
use_costs	booleano	
costs	estructurado	Propiedad estructurada.

### Serie temporal ISW

*Note:* el parámetro de campos de entrada no se utiliza para este nodo. Si se encuentra el parámetro de campos de entrada en el script, se muestra una advertencia para indicar que el nodo tiene *tiempo* y *objetivos* como campos entrantes, pero no de entrada.

Las siguientes propiedades están disponibles para los nodos del tipo db2imtimeseries.

Tabla 171. Propiedades del nodo db2imtimeseries.

Propiedades del nodo db2imtimeseries	Valores	Descripción de la propiedad
time	field	Se permite entero, hora o fecha.
targets	lista de campos	
forecasting_algorithm	arima exponential_smoothing seasonal_trend_decomposition	
forecasting_end_time	auto entero date (fecha) time	
use_records_all	booleano	Si es falso, deben establecerse use_records_start y use_records_end.
use_records_start	entero / hora / fecha	Depende del tipo de campo de tiempo.
use_records_end	entero / hora / fecha	Depende del tipo de campo de tiempo.
interpolation_method	ninguno lineal exponential_splines cubic_splines	

## Propiedades de nugget de modelo de IBM DB2

Las siguientes propiedades son para los nugget de modelo creados mediante los modelos de IBM DB2 ISW.

Árbol de decisión de ISW

No hay propiedades específicas para los nodos del tipo applydb2imtree.

Asociación de ISW

Este nugget de modelo no se puede aplicar en los scripts.

Secuencia de ISW

Este nugget de modelo no se puede aplicar en los scripts.

Regresión de ISW

No hay propiedades específicas para los nodos del tipo applydb2imreg.

Agrupación en clústeres de ISW

No hay propiedades específicas para los nodos del tipo applydb2imcluster.



Bayesiano ingenuo ISW

No hay propiedades específicas para los nodos del tipo `app1ydb2imnb`.

Regresión logística ISW

No hay propiedades específicas para los nodos del tipo `app1ydb2imlog`.

Serie temporal ISW

Este nugget de modelo no se puede aplicar en los scripts.

## Propiedades de nodos de modelado de IBM Netezza Analytics

### Propiedades de nodos de modelado de Netezza

Las siguientes propiedades son comunes a los nodos de modelado de bases de datos de IBM Netezza.

Tabla 172. Propiedades comunes de nodos de Netezza.

Propiedades comunes de nodos de Netezza	Valores	Descripción de la propiedad
<code>custom_fields</code>	<i>booleano</i>	Si es verdadero, le permite especificar el objetivo, la entrada y otros campos del nodo actual. Si es falso, se utiliza la configuración actual de un nodo Tipo situado en un punto anterior de la ruta.
<code>inputs</code>	<i>[campo1 ... campoN]</i>	Campos de entrada o predictor utilizados por el modelo.
<code>target</code>	<i>field</i>	Campo de destino (continuo o categórico).
<code>record_id</code>	<i>field</i>	El campo que se debe utilizar como identificador de registros exclusivo.
<code>use_upstream_connection</code>	<i>booleano</i>	Si es verdadero (valor predeterminado), los detalles de conexión especificados en un nodo anterior. No se utiliza si se especifica <code>move_data_to_connection</code> .
<code>move_data_connection</code>	<i>booleano</i>	Si es verdadero, transfiere los datos a la base de datos especificada mediante <code>connection</code> . No se utiliza si se especifica <code>use_data_upstream_connection</code> .
<code>connection</code>	<i>estructurado</i>	La cadena de conexión para la base de datos de Netezza donde se almacena el modelo. Propiedad estructurada de la siguiente forma: <code>['odbc' '&lt;dsn&gt;' '&lt;username&gt;' '&lt;psw&gt;' '&lt;catname&gt;' '&lt;conn_attribs&gt;' {true false}]</code>  donde: <dsn> es el nombre del origen de datos <username> y <psw> son el nombre de usuario y la contraseña para la base de datos <catname> es el nombre de catálogo <conn_attribs> son los atributos de conexión true   false indica si la contraseña es necesaria.
<code>table_name</code>	<i>cadena</i>	Nombre de la base de datos donde se debe almacenar el modelo.
<code>use_model_name</code>	<i>booleano</i>	Si es true, utiliza el nombre que especifica mediante <code>model_name</code> como el nombre del modelo, de lo contrario el sistema crea el nombre del modelo.
<code>model_name</code>	<i>cadena</i>	Nombre personalizado para nuevo modelo.

Tabla 172. Propiedades comunes de nodos de Netezza (continuación).

Propiedades comunes de nodos de Netezza	Valores	Descripción de la propiedad
include_input_fields	booleano	Si es verdadero, transmite todos los campos de entrada siguientes, de lo contrario solo transmite record_id y los campos generados por el modelo.

### Árbol de decisión de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo netezadectree.

Tabla 173. Propiedades del nodo netezadectree.

Propiedades del nodo netezadectree	Valores	Descripción de la propiedad
impurity_measure	Entropy Gini	La medición de impureza, utilizada para valorar la mejor ubicación para dividir el árbol.
max_tree_depth	entero	Número máximo de niveles que puede alcanzar el crecimiento de árbol. El valor predeterminado es 62 (el máximo posible).
min_improvement_splits	número	Mejoras mínimas que se pueden realizar en la impureza de la división. El valor predeterminado es 0.01.
min_instances_split	entero	Número mínimo de registros por dividir antes de realizar la división. El valor predeterminado es 2 (el mínimo posible).
weights	estructurado	Ponderaciones relativas para clases. Propiedad estructurada. El valor predeterminado es la ponderación de 1 para todas las clases.
pruning_measure	Acc wAcc	El valor predeterminado es Acc (precisión). wAcc alternativo (precisión ponderada) tiene en cuenta las ponderaciones de clase mientras se aplica la poda.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	El valor predeterminado es utilizar allTrainingData para calcular la precisión del modelo. Utilice partitionTrainingData para especificar un porcentaje de datos de prueba por utilizar, o useOtherTable para utilizar un conjunto de datos de prueba desde una tabla específica de la base de datos.
perc_training_data	número	Si prune_tree_options se establece en partitionTrainingData, especifica el porcentaje de datos que se utilizará para entrenamiento.

Tabla 173. Propiedades del nodo *netezadectree* (continuación).

Propiedades del nodo <i>netezadectree</i>	Valores	Descripción de la propiedad
<code>prune_seed</code>	<i>entero</i>	Semilla aleatoria que se debe utilizar para replicar los resultados del análisis si <code>prune_tree_options</code> se establece en <code>partitionTrainingData</code> ; el valor predeterminado es 1.
<code>pruning_table</code>	<i>cadena</i>	Nombre de tabla de un conjunto de datos de poda separado para estimar la precisión del modelo.
<code>compute_probabilities</code>	<i>booleano</i>	Si es verdadero, produce un campo de nivel de confianza (probabilidad) y el campo de predicción.

### K-medias de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo *netezzakmeans*.

Tabla 174. Propiedades del nodo *netezzakmeans*.

Propiedades del nodo <i>netezzakmeans</i>	Valores	Descripción de la propiedad
<code>distance_measure</code>	Euclidean Manhattan Canberra máximo	Método que se debe utilizar para medir la distancia entre puntos de datos.
<code>num_clusters</code>	<i>entero</i>	Número de clústeres que se deben crear; el valor predeterminado es 3.
<code>max_iterations</code>	<i>entero</i>	Número de iteraciones de algoritmos después de los cuáles se debe detener la prueba del modelo; el valor predeterminado es 5.
<code>rand_seed</code>	<i>entero</i>	Semilla aleatoria que se debe utilizar para replicar los resultados del análisis; el valor predeterminado es 12345.

### Red bayesiana de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo *netezabayes*.

Tabla 175. Propiedades del nodo *netezabayes*.

Propiedades del nodo <i>netezabayes</i>	Valores	Descripción de la propiedad
<code>base_index</code>	<i>entero</i>	Identificador numérico asignado al primer campo de entrada de gestión interna; el valor predeterminado es 777.
<code>sample_size</code>	<i>entero</i>	Tamaño de la muestra que se tomará si el número de atributos es muy grande; el valor predeterminado es 10.000.
<code>display_additional_information</code>	<i>booleano</i>	Si es verdadero, muestra información adicional sobre el progreso en un cuadro de diálogo de mensaje.
<code>type_of_prediction</code>	best vecinos nn-neighbors	Tipo de algoritmo de predicción que se utilizará: el mejor (vecino más correlacionado), vecinos (predicción ponderada de vecinos), o vecinos NN (vecinos no nulos).

## Bayesiano ingenuo de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo `netezzanaiwebayes`.

Tabla 176. Propiedades del nodo `netezzanaiwebayes`.

Propiedades del nodo <code>netezzanaiwebayes</code>	Valores	Descripción de la propiedad
<code>compute_probabilities</code>	<i>booleano</i>	Si es verdadero, produce un campo de nivel de confianza (probabilidad) y el campo de predicción.
<code>use_m_estimation</code>	<i>booleano</i>	Si es verdadero, utiliza la técnica m-estimation para evitar probabilidades de cero durante el cálculo.

## KNN de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo `netezzaknn`.

Tabla 177. Propiedades del nodo `netezzaknn`.

Propiedades del nodo <code>netezzaknn</code>	Valores	Descripción de la propiedad
<code>weights</code>	<i>estructurado</i>	Propiedad estructurada que se utiliza para asignar ponderaciones a clases individuales.
<code>distance_measure</code>	Euclidean Manhattan Canberra Máximo	Método que se utiliza para medir la distancia entre puntos de datos.
<code>num_nearest_neighbors</code>	<i>entero</i>	Número de vecinos más próximos de un caso concreto; el valor predeterminado es 3.
<code>standardize_measurements</code>	<i>booleano</i>	Si es verdadero, estandariza las mediciones de campos de entrada continuos antes de calcular los valores de distancia.
<code>use_coresets</code>	<i>booleano</i>	Si es verdadero, utiliza el muestreo del conjunto principal para acelerar el cálculo de conjuntos de datos grandes.

## Clúster divisivo de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo `netez zadivcluster`.

Tabla 178. Propiedades del nodo `netez zadivcluster`.

Propiedades del nodo <code>netez zadivcluster</code>	Valores	Descripción de la propiedad
<code>distance_measure</code>	Euclidean Manhattan Canberra Máximo	Método que se utiliza para medir la distancia entre puntos de datos.
<code>max_iterations</code>	<i>entero</i>	Número máximo de iteraciones de algoritmo que se ejecutarán antes de detener el entrenamiento del modelo; el valor predeterminado es 5.
<code>max_tree_depth</code>	<i>entero</i>	El número máximo de niveles en los que se puede subdividir el conjunto de datos; el valor predeterminado es 3.

Tabla 178. Propiedades del nodo netezadivcluster (continuación).

Propiedades del nodo netezadivcluster	Valores	Descripción de la propiedad
rand_seed	entero	Semilla aleatoria, se utiliza para replicar los análisis; el valor predeterminado es 12345.
min_instances_split	entero	El número mínimo de registros que se pueden dividir, el valor predeterminado es 5.
nivel	entero	El nivel de jerarquía en el que se guardan los registros; el valor predeterminado es -1.

## PCA de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo netezzapca.

Tabla 179. Propiedades del nodo netezzapca.

Propiedades del nodo netezzapca	Valores	Descripción de la propiedad
center_data	booleano	Si es verdadero (opción predeterminada), ejecuta el centrado de datos (también conocido como "sustracción de media") antes del análisis.
perform_data_scaling	booleano	Si es verdadero, ejecuta la adaptación de los datos antes del análisis. De esta forma el análisis será menos arbitrario si las diferentes variables se miden en unidades diferentes.
force_eigensolve	booleano	Si es verdadero, utiliza un método menos preciso, pero más rápido para encontrar componentes principales.
pc_number	entero	El número principal de componentes al que se reducirá el conjunto de datos; el valor predeterminado es 1.

## Árbol de regresión de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo netezzaregtree.

Tabla 180. Propiedades del nodo netezzaregtree.

Propiedades del nodo netezzaregtree	Valores	Descripción de la propiedad
max_tree_depth	entero	Número máximo de niveles que puede crecer el árbol por debajo del nodo raíz; el valor predeterminado es 10.
split_evaluation_measure	Variance	Medida de clase de impureza, se utiliza para evaluar la mejor ubicación para dividir el árbol; el valor predeterminado (y la única opción actualmente) es Variance.
min_improvement_splits	número	Cantidad mínima para reducir la impureza antes de que se cree la nueva división en el árbol.
min_instances_split	entero	El número mínimo de registros que se pueden dividir.

Tabla 180. Propiedades del nodo netezzaregtree (continuación).

Propiedades del nodo netezzaregtree	Valores	Descripción de la propiedad
pruning_measure	mse r2 pearson spearman	Método que se utilizará para la poda.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	El valor predeterminado es utilizar allTrainingData para calcular la precisión del modelo. Utilice partitionTrainingData para especificar un porcentaje de datos de prueba por utilizar, o useOtherTable para utilizar un conjunto de datos de prueba desde una tabla específica de la base de datos.
perc_training_data	número	Si prune_tree_options se establece en PercTrainingData, especifica el porcentaje de datos que se utilizará para entrenamiento.
prune_seed	entero	Semilla aleatoria que se debe utilizar para replicar los resultados del análisis si prune_tree_options se establece en PercTrainingData; el valor predeterminado es 1.
pruning_table	cadena	Nombre de tabla de un conjunto de datos de poda separado para estimar la precisión del modelo.
compute_probabilities	booleano	Si es verdadero, especifica que las varianzas de las clases asignadas se deben incluir en el resultado.

## Regresión lineal de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo netezزالineregression.

Tabla 181. Propiedades del nodo netezزالineregression.

Propiedades del nodo netezزالineregression	Valores	Descripción de la propiedad
use_svd	booleano	Si es verdadero, utiliza la matriz de descomposición de valores singulares en lugar de la matriz original, para mayor velocidad y precisión numérica.
include_intercept	booleano	Si es verdadero (valor predeterminado), aumenta la precisión global de la solución.
calculate_model_diagnostics	booleano	Si es verdadero, calcula el diagnóstico del modelo.

## Series temporales de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo netezzatimeseries.

Tabla 182. Propiedades del nodo `netezzatimeseries`.

Propiedades del nodo <code>netezzatimeseries</code>	Valores	Descripción de la propiedad
<code>time_points</code>	<i>field</i>	El campo de entrada que contiene los valores de fecha u hora de la serie temporal.
<code>time_series_ids</code>	<i>field</i>	El campo de entrada que contiene diversos ID de series temporales; utilice esta opción si la entrada contiene más de una serie temporal.
<code>model_table</code>	<i>field</i>	Nombre de la tabla de base de datos en la que se guardará el modelo de series temporales de Netezza.
<code>description_table</code>	<i>field</i>	Nombre de la tabla de entrada que contiene los nombres y las descripciones de las series temporales.
<code>seasonal_adjustment_table</code>	<i>field</i>	Nombre de la tabla de salida en la que se guardarán los valores ajustados calculados por los algoritmos de suavizado exponencial o de descomposición de tendencia estacional.
<code>algorithm_name</code>	SpectralAnalysis o spectral ExponentialSmoothing o esmoothing ARIMA SeasonalTrendDecomposition o std	Algoritmo que hay que utilizar para el modelado de series temporales.
<code>trend_name</code>	N A DA M DM	Tipo de tendencia del suavizado exponencial: N - none A - aditivo DA -aditivo amortiguado M - multiplicativo DM - multiplicativo amortiguado
<code>seasonality_type</code>	N A M	Tipo de estacionalidad del suavizado exponencial: N - none A - aditivo M - multiplicativo
<code>interpolation_method</code>	lineal cubicspline exponentialspline	Método de interpolación que hay que utilizar.
<code>timerange_setting</code>	SD SP	Valor de rango de tiempo que se debe utilizar: SD - determinado por el sistema (utiliza el rango completo de datos de series de tiempo) SP - especificado por el usuario mediante <code>earliest_time</code> y <code>latest_time</code>

Tabla 182. Propiedades del nodo netezatimeseries (continuación).

Propiedades del nodo netezatimeseries	Valores	Descripción de la propiedad
earliest_time	Fecha	Hora de inicio y finalización si timerange_setting es SP.  Formato: <aaaa>-<mm>-<dd>
latest_time		
arima_setting	SD SP	Valor para el algoritmo ARIMA (sólo se utiliza si algorithm_name se establece en ARIMA): SD - determinado por el sistema SP - especificado por el usuario  Si se utiliza arima_setting = SP, utilice los parámetros siguientes para establecer los valores estacionales y no estacionales.
p_symbol	sin eq lesseq	ARIMA - operador para los parámetros p, d, q, sp, sd y sq: less - menor que eq - igual a lesseq - menor o igual que
d_symbol		
q_symbol		
sp_symbol		
sd_symbol		
sq_symbol		
p	entero	ARIMA: grados no estacionales de autocorrelación.
q	entero	ARIMA: valor de derivación no estacional.
d	entero	ARIMA: número no estacional de órdenes de media móvil presentes en el modelo.
sp	entero	ARIMA: grados estacionales de autocorrelación.
sq	entero	ARIMA: valor de derivación estacional.
sd	entero	ARIMA: número estacional de órdenes de media móvil presentes en el modelo.
advanced_setting	SD SP	Determina cómo se manejan los valores avanzados: SD - determinado por el sistema SP - especificado por el usuario mediante period, units_period y forecast_setting.
punto	entero	Longitud de ciclo estacional, especificado junto con units_period. No aplicable para análisis espectrales.



Tabla 182. Propiedades del nodo *netezzatimeseries* (continuación).

Propiedades del nodo <i>netezzatimeseries</i>	Valores	Descripción de la propiedad
<i>units_period</i>	ms s min h d semana q y	Unidades en que se expresa <i>period</i> : ms - milisegundos s - segundos min - minutos h - horas d - días wk - semanas q - trimestres y - años  Por ejemplo, para una serie temporal semanal utilice 1 para <i>period</i> y wk para <i>units_period</i> .
<i>forecast_setting</i>	<i>forecasthorizon</i> <i>forecasttimes</i>	Especifica cómo se han de realizar las previsiones.
<i>forecast_horizon</i>	<i>cadena</i>	Si <i>forecast_setting</i> = <i>forecasthorizon</i> , especifica el punto final para la previsión.  Formato: <aaa>-<mm>-<dd>
<i>forecast_times</i>	[{'date'}, {'date'},..., {'date'}]	Si <i>forecast_setting</i> = <i>forecasttimes</i> , especifica los valores de tiempo que se utilizarán para hacer previsiones.  Formato: <aaa>-<mm>-<dd>
<i>include_history</i>	<i>booleano</i>	Indica si se deben incluir los valores históricos en los resultados.
<i>include_interpolated_values</i>	<i>booleano</i>	Indica si se deben incluir los valores interpolados en los resultados. No es aplicable si <i>include_history</i> es false.

### Lineal generalizado de Netezza

Las siguientes propiedades están disponibles para los nodos del tipo *netezzaglm*.

Tabla 183. Propiedades del nodo *netezzaglm*.

Propiedades del nodo <i>netezzaglm</i>	Valores	Descripción de la propiedad
<i>dist_family</i>	bernoulli de gauss poisson negativebinomial wald gamma	El tipo de distribución. El valor predeterminado es <i>bernoulli</i> .
<i>dist_params</i>	<i>número</i>	Valor del parámetro de distribución que hay que utilizar. Sólo se aplica si <i>distribution</i> es <i>Negativebinomial</i> .

Tabla 183. Propiedades del nodo netezzaglm (continuación).

Propiedades del nodo netezzaglm	Valores	Descripción de la propiedad
trials	entero	Sólo se aplica si <code>distribution</code> es Binomial. Cuando la respuesta objetivo es un número de eventos que tienen lugar en un conjunto de ensayos, el campo <code>target</code> contiene el número de eventos, y el campo <code>trials</code> contiene el número de ensayos.
model_table	field	Nombre de la tabla de base de datos en la que se guardará el modelo lineal generalizado de Netezza.
maxit	entero	Número máximo de iteraciones que debe ejecutar el algoritmo; el valor predeterminado es 20.
eps	número	Valor del error máximo (en notación científica) en el que el algoritmo debería dejar de buscar el modelo de mejor ajuste. El valor predeterminado es $-3$ , lo que significa $1E-3$ o 0,001.
tol	número	El valor (en notación científica) por debajo del que los errores se tratan como si su valor fuera cero. El valor predeterminado es $-7$ , lo que significa que los valores de error por debajo de $1E-7$ (o 0,0000001) se cuentan como insignificantes.
link_func	identidad inverse invnegative invsquare sqrt power oddspower registro clog loglog cloglog logit probit gaussit cauchit canbinom cangeom cannegbinom	Función de enlace que se ha de utilizar; el valor predeterminado es <code>logit</code> .
link_params	número	Valor del parámetro de función de enlace que hay que utilizar. Sólo se aplica si <code>link_function</code> es <code>power</code> u <code>oddspower</code> .
interaction	[[[ <i>colnames1</i> ],[ <i>levels1</i> ]],[[ <i>colnames2</i> ],[ <i>levels2</i> ]],...,[[ <i>colnamesN</i> ],[ <i>levelsN</i> ]],]	Especifica las interacciones entre los campos. <i>colnames</i> es una lista de campos de entrada, y <i>level</i> es siempre 0 para cada campo.
intercept	booleano	Si es <code>true</code> , incluye la interceptación en el modelo.

## Propiedades de nugget de modelo de Netezza

Las siguientes propiedades son comunes a los nuggets del modelo de la base de datos de Netezza.

Tabla 184. Propiedades comunes de nugget de nodos de Netezza.

Propiedades comunes de nugget de modelo de Netezza	Valores	Descripción de la propiedad
connection	cadena	La cadena de conexión para la base de datos de Netezza donde se almacena el modelo.
table_name	cadena	Nombre de la base de datos donde se almacenará el modelo.

Otras las propiedades del nugget de modelo son las mismas que las del nodo de modelado correspondiente.

Los nombres de script de los nuggets de modelo son los siguientes.

Tabla 185. Nombres de script de nuggets de modelos Netezza.

Nugget de modelo	Nombre de script
Árbol de decisiones	aplynetezzadectree
K-medias	aplynetezzakmeans
Red bayesiana	aplynetezzabayes
bayesiano ingenuo	aplynetezzanaivebayes
KNN	aplynetezzaknn
Clúster divisivo	aplynetezzadivcluster
PCA	aplynetezzapca
Árbol de regresión	aplynetezzaregtree
Regresión lineal	aplynetezzalineregression
Serie temporal	aplynetezzatimeseries
Lineal generalizado	aplynetezzaglm



---

## Capítulo 16. Propiedades de los nodos de resultados

Las propiedades de nodos de resultados se diferencian un poco de las de otros tipos de nodos. En lugar de hacer referencia a una opción determinada de nodo, las propiedades de nodos de resultados almacenan una referencia en el objeto de resultado. Esto resulta útil al tomar un valor de una tabla y establecerlo como un parámetro de ruta.

Esta sección describe las propiedades de scripts disponibles para los nodos de resultados.

---

### Propiedades del nodo analysis



El nodo Análisis evalúa la capacidad de los modelos predictivos para generar predicciones precisas. Los nodos Análisis realizan varias comparaciones entre los valores predichos y los valores reales para uno o más nugget de modelo. También pueden comparar modelos predictivos entre sí.

Tabla 186. Propiedades del nodo analysis.

Propiedades del nodo analysis	Tipo de datos	Descripción de la propiedad
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
use_output_name	booleano	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_format	Text (.txt) HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado.
by_fields	[field field field]	
full_filename	cadena	Nombre del archivo de resultados, si se trata de resultados HTML, de datos o de disco.
coincidence	booleano	
performance	booleano	
evaluation_binary	booleano	
confidence	booleano	
umbral	número	
improve_accuracy	número	
inc_user_measure	booleano	
user_if	expr	
user_then	expr	
user_else	expr	
user_compute	[Mean Sum Min Max SDev]	

## Propiedades del nodo dataaudit



El nodo Auditoría de datos permite echar un primer vistazo exhaustivo a los datos, incluyendo estadísticos de resumen, histogramas y distribución para cada campo, así como información sobre valores atípicos, valores perdidos y extremos. Los resultados se muestran en una matriz fácil de leer que se puede ordenar y utilizar para generar nodos de preparación de datos y gráficos de tamaño completo.

Tabla 187. Propiedades del nodo dataaudit.

Propiedades del nodo dataaudit	Tipo de datos	Descripción de la propiedad
custom_fields	booleano	
fields	[campo1 ... campoN]	
overlay	field	
display_graphs	booleano	Se utiliza para activar o desactivar la representación de gráficos en la matriz de resultados.
basic_stats	booleano	
advanced_stats	booleano	
median_stats	booleano	
calculate	Recuento Breakdown	Se utiliza para calcular valores perdidos. Seleccione uno de los métodos de cálculo, ambos o ninguno.
outlier_detection_method	std iqr	Se utiliza para especificar el método de detección de valores atípicos y extremos.
outlier_detection_std_outlier	número	Si outlier_detection_method es std, especifica el número que se utilizará para definir los valores atípicos.
outlier_detection_std_extreme	número	Si outlier_detection_method es std, especifica el número que se utilizará para definir los valores extremos.
outlier_detection_iqr_outlier	número	Si outlier_detection_method es iqr, especifica el número que se utilizará para definir los valores atípicos.
outlier_detection_iqr_extreme	número	Si outlier_detection_method es iqr, especifica el número que se utilizará para definir los valores extremos.
use_output_name	booleano	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.

Tabla 187. Propiedades del nodo dataaudit (continuación).

Propiedades del nodo dataaudit	Tipo de datos	Descripción de la propiedad
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado.
paginate_output	booleano	Si output_format es HTML, los resultados se separarán por páginas.
lines_per_page	número	Si se usa con paginate_output, especifica las líneas por página del resultado.
full_filename	cadena	

## Propiedades del nodo matrix



El nodo Matriz crea una tabla que muestra las relaciones entre campos. Se suele utilizar normalmente para mostrar las relaciones entre dos campos simbólicos, pero también puede mostrar relaciones entre campos de marcas o entre campos numéricos.

Tabla 188. Propiedades del nodo matrix.

Propiedades del nodo matrix	Tipo de datos	Descripción de la propiedad
fields	Seleccionado Flags Numerics	
row	field	
column	field	
include_missing_values	booleano	Determina si los valores perdidos por el sistema (nulos) o no especificados por el usuario (vacíos) se incluyen en los resultados de fila y columna.
cell_contents	CrossTabs Función	
function_field	cadena	
function	Sum Media Mín Máx SDev	
sort_mode	Unsorted Ascending Descending	
highlight_top	número	Si no es cero, es verdadero.
highlight_bottom	número	Si no es cero, es verdadero.

Tabla 188. Propiedades del nodo matrix (continuación).

Propiedades del nodo matrix	Tipo de datos	Descripción de la propiedad
display	[Counts Expected Residuos RowPct ColumnPct TotalPct]	
include_totals	booleano	
use_output_name	booleano	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado. Los formatos Formatted y Delimited aceptan el modificador transposed, mediante el cual se transponen filas y columnas en la tabla.
paginate_output	booleano	Si output_format es HTML, los resultados se separarán por páginas.
lines_per_page	número	Si se usa con paginate_output, especifica las líneas por página del resultado.
full_filename	cadena	

## Propiedades del nodo means



El nodo Medias compara las medias de grupos independientes o de pares de campos relacionados para probar si existen diferencias significativas. Por ejemplo, puede comparar los ingresos medios antes y después de poner en marcha una promoción o comparar los ingresos de los clientes que no recibieron esa promoción con los que sí lo hicieron.

Tabla 189. Propiedades del nodo means.

Propiedades del nodo means	Tipo de datos	Descripción de la propiedad
means_mode	BetweenGroups BetweenFields	Especifica el tipo de estadístico de las medias que se va a ejecutar en los datos.
test_fields	[field1 ... fieldn]	Especifica el campo de prueba si means_mode se establece en BetweenGroups.
grouping_field	field	Especifica el campo de agrupación.
paired_fields	[{field1 field2} {field3 field4} ...]	Especifica los pares de campos que se usan si means_mode se establece en BetweenFields.



Tabla 189. Propiedades del nodo means (continuación).

Propiedades del nodo means	Tipo de datos	Descripción de la propiedad
label_correlations	booleano	Determina si las etiquetas de correlación se muestran en el resultado. Esta configuración se aplica únicamente si means_mode se establece en BetweenFields.
correlation_mode	Probability Absolute	Determina si las correlaciones deben etiquetarse según la probabilidad o según el valor absoluto.
weak_label	cadena	
medium_label	cadena	
strong_label	cadena	
weak_below_probability	número	Si correlation_mode se establece en Probability, determina el valor de corte para las correlaciones débiles. Debe tratarse de un valor comprendido entre 0 y 1; por ejemplo, 0,90.
strong_above_probability	número	Valor de corte para correlaciones fuertes.
weak_below_absolute	número	Si correlation_mode se establece en Absolute, especifica el valor de corte para las correlaciones débiles. Debe tratarse de un valor comprendido entre 0 y 1; por ejemplo, 0,90.
strong_above_absolute	número	Valor de corte para correlaciones fuertes.
unimportant_label	cadena	
marginal_label	cadena	
important_label	cadena	
unimportant_below	número	Valor de corte para una importancia del campo baja. Debe tratarse de un valor comprendido entre 0 y 1; por ejemplo, 0,90.
important_above	número	
use_output_name	booleano	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Nombre que se va a usar.
output_mode	Screen File	Especifica la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Especifica el tipo de resultado.
full_filename	cadena	
output_view	Simple Advanced	Determina si el resultado muestra la vista simple o la avanzada.

## Propiedades del nodo report



El nodo Informe crea informes con formato que contienen texto fijo, así como datos y otras expresiones derivadas de los datos. Puede especificar el formato del informe utilizando plantillas de texto para definir el texto fijo y las construcciones de resultados de datos. Puede proporcionar formato de texto personalizado utilizando etiquetas HTML de la plantilla y configurando opciones en la pestaña Resultado. Puede incluir valores de datos y otros resultados condicionales mediante el uso de expresiones CLEM en la plantilla.

Tabla 190. Propiedades del nodo report.

Propiedades del nodo report	Tipo de datos	Descripción de la propiedad
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	HTML (.html) Text (.txt) Output (.cou)	Se utiliza para especificar el tipo de resultado.
use_output_name	booleano	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
text	cadena	
full_filename	cadena	
highlights	booleano	
título	cadena	
lines_per_page	número	

## Propiedades del nodo Routput



El nodo Routput permite analizar datos y resultados de la puntuación del modelo utilizando su propio script R personalizado. El resultado del análisis puede ser en texto o en gráficos. El resultado se añade a la pestaña **Resultado** del panel de gestor. De forma alternativa, el resultado se puede redirigir a un archivo.

Tabla 191. Propiedades del nodo Routput.

Propiedades del nodo Routput	Tipo de datos	Descripción de la propiedad
syntax	cadena	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	booleano	
convert_datetime_class	POSIXct POSIXlt	
convert_missing	booleano	

Tabla 191. Propiedades del nodo Routput (continuación).

Propiedades del nodo Routput	Tipo de datos	Descripción de la propiedad
output_name	Auto Personalizado.	
custom_name	cadena	
output_to	Screen File	
output_type	Graph Text	
full_filename	cadena	
graph_file_type	HTML COU	
text_file_type	HTML TXT COU	

## Propiedades del nodo setglobals



El nodo Val. globales explora los datos y calcula los valores de resumen que se pueden utilizar en expresiones CLEM. Por ejemplo, puede utilizar este nodo para calcular estadísticos para un campo denominado *edad* y, a continuación, utilizar la media global de *edad* en expresiones CLEM insertando la función @GLOBAL\_MEAN(*edad*).

Tabla 192. Propiedades del nodo setglobals.

Propiedades del nodo setglobals	Tipo de datos	Descripción de la propiedad
globals	[Sum Mean Min Max SDev]	Propiedad estructurada
clear_first	booleano	
show_preview	booleano	

## Propiedades de nodo simeval



El nodo de evaluación de simulación evalúa un campo de destino predicho y presenta información sobre la distribución y correlación del campo de destino.

Tabla 193. Propiedades del nodo simeval.

Propiedades del nodo simeval	Tipo de datos	Descripción de la propiedad
target	campo	
iteration	campo	
presorted_by_iteration	booleano	
max_iterations	número	
tornado_fields	[field1...fieldN]	
plot_pdf	booleano	
plot_cdf	booleano	

Tabla 193. Propiedades del nodo simeval (continuación).

Propiedades del nodo simeval	Tipo de datos	Descripción de la propiedad
show_ref_mean	booleano	
show_ref_median	booleano	
show_ref_sigma	booleano	
num_ref_sigma	número	
show_ref_pct	booleano	
ref_pct_bottom	número	
ref_pct_top	número	
show_ref_custom	booleano	
ref_custom_values	[number1...numberN]	
category_values	Category Probabilities Both	
category_groups	Categories Iterations	
create_pct_table	booleano	
pct_table	Quartiles Intervals Personalizado.	
pct_intervals_num	número	
pct_custom_values	[number1...numberN]	

## Propiedades del nodo simfit

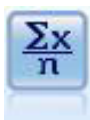


El nodo de simulación de ajuste examina la distribución de las estadísticas de los datos de cada campo y genera (o actualiza) un nodo de generación de simulación, con la mejor distribución de ajuste asignada a cada campo. El nodo de generación de simulación se puede utilizar, a continuación, para generar datos simulados.

Tabla 194. Propiedades del nodo simfit.

Propiedades del nodo simfit	Tipo de datos	Descripción de la propiedad
build	Nodo XMLExport Both	
use_source_node_name	booleano	
source_node_name	cadena	El nombre personalizado del nodo de origen que se está generando o actualizando.
use_cases	All LimitFirstN	
use_case_limit	entero	
fit_criterion	AndersonDarling KolmogorovSmirnov	
num_bins	entero	
parameter_xml_filename	cadena	
generate_parameter_import	booleano	

## Propiedades del nodo statistics



El nodo Estadísticos ofrece información básica de resumen acerca de los campos numéricos. Calcula estadísticos de resumen para campos individuales y correlaciones entre campos.

Tabla 195. Propiedades del nodo statistics.

Propiedades del nodo statistics	Tipo de datos	Descripción de la propiedad
use_output_name	booleano	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	Text (.txt) HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado.
full_filename	cadena	
examine	[field field field]	
correlate	[field field field]	
statistics	[Count Mean Sum Min Max Range Variance SDev SErr Median Mode]	
correlation_mode	Probability Absolute	Determina si las correlaciones deben etiquetarse según la probabilidad o según el valor absoluto.
label_correlations	booleano	
weak_label	cadena	
medium_label	cadena	
strong_label	cadena	
weak_below_probability	número	Si correlation_mode se establece en Probability, determina el valor de corte para las correlaciones débiles. Debe tratarse de un valor comprendido entre 0 y 1; por ejemplo, 0,90.
strong_above_probability	número	Valor de corte para correlaciones fuertes.
weak_below_absolute	número	Si correlation_mode se establece en Absolute, especifica el valor de corte para las correlaciones débiles. Debe tratarse de un valor comprendido entre 0 y 1; por ejemplo, 0,90.
strong_above_absolute	número	Valor de corte para correlaciones fuertes.

---

## Propiedades del nodo statisticsoutput



El nodo Resultados de Statistics le permite llamar a un procedimiento de IBM SPSS Statistics para analizar los datos de IBM SPSS Modeler. Se puede acceder a una gran variedad de procedimientos analíticos de IBM SPSS Statistics. Este nodo requiere una copia de IBM SPSS Statistics con licencia.

Las propiedades de este nodo están descritas en “Propiedades del nodo statisticsoutput” en la página 240.

---

## Propiedades del nodo table



El nodo Tabla muestra los datos en formato de tabla, que también se puede escribir en un archivo. Esto es útil en cualquier momento en que necesite inspeccionar sus valores de datos o exportarlos en un formato fácilmente legible.

Tabla 196. Propiedades del nodo table.

Propiedades del nodo table	Tipo de datos	Descripción de la propiedad
full_filename	<i>cadena</i>	Nombre del archivo de resultados, si se trata de resultados HTML, de datos o de disco.
use_output_name	<i>booleano</i>	Especifica si se utiliza un nombre de resultado personalizado.
output_name	<i>cadena</i>	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado.
transpose_data	<i>booleano</i>	Transpone los datos antes de exportarlos de manera que las filas representan campos y las columnas, registros.
paginate_output	<i>booleano</i>	Si output_format es HTML, los resultados se separarán por páginas.
lines_per_page	<i>número</i>	Si se usa con paginate_output, especifica las líneas por página del resultado.
highlight_expr	<i>cadena</i>	
output	<i>cadena</i>	Propiedad de sólo lectura que mantiene una referencia en la última tabla creada por el nodo.
value_labels	<i>{Valor CadenaEtiquetas}</i> <i>{Valor CadenaEtiquetas} ...]</i>	Se utiliza para especificar etiquetas para los pares de valores.

Tabla 196. Propiedades del nodo table (continuación).

Propiedades del nodo table	Tipo de datos	Descripción de la propiedad
display_places	entero	Establece el número de cifras decimales para el campo cuando se muestra (sólo se aplica a campos con almacenamiento REAL). Un valor de -1 utilizará el valor predeterminado de la ruta.
export_places	entero	Establece el número de cifras decimales para el campo cuando se exporta (sólo se aplica a campos con almacenamiento REAL). Un valor de -1 utilizará el valor predeterminado de la ruta.
decimal_separator	DEFAULT PERIOD COMMA	Establece el separador decimal para el campo (sólo se aplica a campos con almacenamiento REAL).
date_format	"DDMMAA" "MMDDYY" "AAMMDD" "YYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-AAAA" "DD-MES-YY" "DD-MES-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.AAAA" "MM.DD.YY" "MM.DD.YYYY" "DD.MES.YY" "DD.MES.YYYY" "DD/MM/YY" "DD/MM/AAAA" "MM/DD/YY" "MM/DD/YYYY" "DD/MES/YY" "DD/MES/YYYY" MON YYYY q Q YYYY ww WK YYYY	Establece el formato de fecha para el campo (sólo se aplica a campos con almacenamiento FECHA o MARCADETIEMPO).
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"	Establece el formato de hora para el campo (sólo se aplica a campos con almacenamiento TIME o TIMESTAMP).

Tabla 196. Propiedades del nodo table (continuación).

Propiedades del nodo table	Tipo de datos	Descripción de la propiedad
column_width	entero	Establece el ancho de columna para el campo. Un valor de -1 establecerá el ancho de columna en Auto.
justify	AUTO CENTER LEFT RIGHT	Establece la justificación de columna para el campo.

## Propiedades del nodo transform



El nodo Transformación permite seleccionar y previsualizar los resultados de las transformaciones antes de aplicarlas a los campos seleccionados.

Tabla 197. Propiedades de nodo de transformación.

Propiedades del nodo transform	Tipo de datos	Descripción de la propiedad
fields	[ field1... fieldn]	Campos que se utilizarán en la transformación.
formula	All Select	Indica si se deben calcular todas las transformaciones o sólo las seleccionadas.
formula_inverse	booleano	Indica si se debe utilizar la transformación inversa.
formula_inverse_offset	número	Indica el desplazamiento de los datos que se utilizará en la fórmula. De forma predeterminada es 0, a menos que el usuario especifique un valor.
formula_log_n	booleano	Indica si debe utilizarse la transformación $\log_n$ .
formula_log_n_offset	número	
formula_log_10	booleano	Indica si debe utilizarse la transformación $\log_{10}$ .
formula_log_10_offset	número	
formula_exponential	booleano	Indica si se debe utilizar la transformación exponencial ( $e^x$ ).
formula_square_root	booleano	Indica si se debe utilizar la transformación de raíz cuadrada.
use_output_name	booleano	Especifica si se utiliza un nombre de resultado personalizado.
output_name	cadena	Si use_output_name es verdadero, especifica el nombre que se va a utilizar.
output_mode	Screen File	Se utiliza para especificar la ubicación objetivo para el resultado generado desde el nodo de resultados.



Tabla 197. Propiedades de nodo de transformación (continuación).

Propiedades del nodo transform	Tipo de datos	Descripción de la propiedad
output_format	HTML (.html) Output (.cou)	Se utiliza para especificar el tipo de resultado.
paginate_output	booleano	Si output_format es HTML, los resultados se separarán por páginas.
lines_per_page	número	Si se usa con paginate_output, especifica las líneas por página del resultado.
full_filename	cadena	Indica el nombre de archivo que se utilizará para el resultado de archivo.



---

## Capítulo 17. Propiedades de nodos Exportar

---

### Propiedades de nodos Exportar comunes

Las siguientes propiedades son comunes a todos los nodos de exportación:

Tabla 198. Propiedades comunes de nodos de exportación.

Propiedad	Valores	Descripción de la propiedad
publish_path	<i>cadena</i>	Introduzca el nombre raíz que se utilizará para la imagen publicada y los archivos de parámetros.
publish_metadata	<i>booleano</i>	Especifica si un archivo de metadatos se crea y que describe las entradas y los resultados de la imagen y sus modelos de datos.
publish_use_parameters	<i>booleano</i>	Especifica si se incluyen parámetros de ruta en el archivo *.par.
publish_parameters	<i>string list</i>	Especifica los parámetros que se van a incluir.
execute_mode	export_data publish	Especifica si el nodo se ejecuta sin publicar la ruta, o si la ruta se publica automáticamente cuando se ejecuta el nodo.

---

### Propiedades del nodo asexport

La exportación de Analytic Server permite ejecutar una ruta en el sistema de archivos distribuido de Hadoop (HDFS).

Tabla 199. propiedades del nodo asexport.

Propiedades del nodo asexport	Tipo de datos	Descripción de la propiedad
data_source	<i>cadena</i>	Nombre del origen de datos.
export_mode	<i>cadena</i>	Especifica si se <b>añaden</b> (append) los datos exportados al origen de datos existente o si se <b>sobrescriben</b> (overwrite) al origen de datos existente.
host	<i>cadena</i>	El nombre del host de Analytic Server.
port	<i>entero</i>	El puerto por el que escucha Analytic Server.
tenant	<i>cadena</i>	En un entorno de varios consumidores, el nombre del consumidor al que se pertenece. En un entorno de un único consumidor, toma el valor predeterminado de <b>ibm</b> .
set_credentials	<i>booleano</i>	Si la autenticación de usuario en Analytic Server es la misma que en el servidor SPSS Modeler, asigne a esta propiedad el valor <b>false</b> . En caso contrario, asígnele <b>true</b> .

Tabla 199. propiedades del nodo asexport (continuación).

Propiedades del nodo asexport	Tipo de datos	Descripción de la propiedad
user_name	cadena	El nombre de usuario de inicio de sesión de Analytic Server. Solo es necesario si set_credentials es true.
contraseña	cadena	La contraseña de inicio de sesión de Analytic Server. Solo es necesario si set_credentials es true.

## Propiedades del nodo cognosexport



El nodo Exportar de IBM Cognos BI exporta datos en un formato que pueden leer las bases de datos de Cognos BI.

*Nota:* Para este nodo, debe definir una conexión de Cognos y una conexión ODBC.

### Conexión de Cognos

Las propiedades de la conexión de Cognos son las siguientes.

Tabla 200. Propiedades del nodo cognosexport.

Propiedades del nodo cognosexport	Tipo de datos	Descripción de la propiedad
cognos_connection	{"campo", "campo", ..., "campo"}	Una propiedad de la lista que contiene los detalles de conexión para el servidor de Cognos. El formato es:  {"Cognos_server_URL", login_mode, "namespace", "username", "password"}  donde: Cognos_server_URL es la URL del servidor de Cognos al que está realizando la exportación login_mode indica si se utiliza el inicio de sesión anónimo, y es true o false; si se establece como true, los campos siguientes deben establecerse en "" namespace especifica el proveedor de autenticación de seguridad utilizado para registrarse en el servidor username y password son los utilizados para registrarse en el servidor de Cognos
cognos_package_name	cadena	La ruta y el nombre del paquete de Cognos al que está exportando datos, por ejemplo: /Public Folders/MyPackage
cognos_datasource	cadena	
cognos_export_mode	Publicar ExportFile	
cognos_filename	cadena	

### Conexión ODBC

Las propiedades de la conexión ODBC son idénticas a las indicadas para databaseexport en la sección siguiente, a excepción de la propiedad datasource, que no es válida.

## Propiedades del nodo databaseexport



El nodo Exportar base de datos escribe datos en orígenes de datos relacionales compatibles con ODBC. Para escribir en un origen de datos ODBC, el origen de datos debe existir y debe tener permiso para escribir en él.

Tabla 201. Propiedades del nodo databaseexport.

Propiedades del nodo databaseexport	Tipo de datos	Descripción de la propiedad
datasource	<i>cadena</i>	
username	<i>cadena</i>	
password	<i>cadena</i>	
epassword	<i>cadena</i>	Este intervalo es de sólo lectura durante la ejecución. Para generar una contraseña codificada, utilice la herramienta Contraseña disponible del menú Herramientas. Consulte el tema “Generación de una contraseña codificada” en la página 47 para obtener más información.
table_name	<i>cadena</i>	
write_mode	Create Append Fundir	
map	<i>cadena</i>	Correlaciona un nombre de campo de ruta a un nombre de columna de la base de datos (sólo es válido si write_mode es Merge). Para una fusión, todos los campos se deben estar correlacionados para que se exporten. Los nombres de campos que no existen en la base de datos se añaden como nuevas columnas.
key_fields	<i>[field field ... field]</i>	Especifica el campo de ruta que se utiliza para la clave; la propiedad map muestra los elementos que se corresponden con la base de datos.
join	Base de datos Añadir	
drop_existing_table	<i>booleano</i>	
delete_existing_rows	<i>booleano</i>	
default_string_size	<i>entero</i>	
type		Propiedad estructurada que se utiliza para establecer el tipo de esquema.
generate_import	<i>booleano</i>	

Tabla 201. Propiedades del nodo databaseexport (continuación).

Propiedades del nodo databaseexport	Tipo de datos	Descripción de la propiedad
use_custom_create_table_command	booleano	Utilice el intervalo <i>custom_create_table</i> para modificar el comando de SQL estándar CREATE TABLE.
custom_create_table_command	cadena	Especifica el comando de cadena a utilizar en lugar del comando de SQL estándar CREATE TABLE.
use_batch	booleano	Las siguientes propiedades son opciones avanzadas para la carga masiva de la base de datos. Un valor verdadero para Use_batch desactiva fila a fila las confirmaciones en la base de datos.
batch_size	número	Especifica el número de registros para enviar a la base de datos antes de confirmar en la memoria.
bulk_loading	Off ODBC External	Especifica el tipo de carga masiva. A continuación se muestran las opciones adicionales para ODBC y External.
not_logged	booleano	
odbc_binding	Row Columna	Especifique el enlace a lo largo de las filas o de las columnas para la carga masiva a través de ODBC.
loader_delimit_mode	Tabulador Space Other	Especifique el tipo de delimitador para la carga masiva a través de un programa externo. Seleccione Other junto con la propiedad loader_other_delimiter para especificar los delimitadores, como la coma (,).
loader_other_delimiter	cadena	
specify_data_file	booleano	Una marca verdadera activa la siguiente propiedad data_file, en la que puede especificar el nombre de archivo y la ruta de acceso en la que se va a escribir al realizar la carga masiva en la base de datos.
data_file	cadena	
specify_loader_program	booleano	Una marca verdadera activa la siguiente propiedad loader_program, en la que puede especificar el nombre y la ubicación de un programa o script del cargador externo.
loader_program	cadena	
gen_logfile	booleano	Una marca verdadera activa la siguiente propiedad logfile_name, en la que puede especificar el nombre de un archivo en el servidor para generar un registro de errores.
logfile_name	cadena	

Tabla 201. Propiedades del nodo databaseexport (continuación).

Propiedades del nodo databaseexport	Tipo de datos	Descripción de la propiedad
check_table_size	booleano	Una marca verdadera permite la comprobación de la tabla para garantizar que el aumento del tamaño de la tabla de la base de datos se corresponde con el número de filas exportadas desde IBM SPSS Modeler.
loader_options	cadena	Especifique los argumentos adicionales, como -comment y -specialdir, en el programa cargador.
export_db_primarykey	booleano	Determina si el campo especificado es una clave primaria.
use_custom_create_index_command	booleano	Si se establece true, se activa SQL personalizado para todos los índices.
custom_create_index_command	cadena	Especifica el comando de SQL empleado para crear índices cuando SQL personalizado está activado. (Este valor puede anularse para determinados índices, tal y como se indica a continuación.)
indexes.INDEXNAME.fields		Crea el índice especificado si procede y enumera los nombres de campos que se van a incluir en él.
indexes.INDEXNAME.use_custom_create_index_command	booleano	Se usa para activar o desactivar el SQL personalizado para un índice específico.
indexes.INDEXNAME.custom_create_command		Especifica el SQL personalizado que se usa para el índice específico.
indexes.INDEXNAME.remove	booleano	Si se establece True, se elimina el índice específico del grupo de índices.
table_space	cadena	Especifica el espacio de tabla que se creará.
use_partition	booleano	Especifica que se utilizará el campo Distribuir Hash.
partition_field	cadena	Especifica el contenido del campo Distribuir Hash.

Note: Para algunas bases de datos, puede especificar que se crearán tablas de bases de datos para la exportación con compresión (por ejemplo, el equivalente a CREATE TABLE MYTABLE (...) COMPRESS YES; en SQL). Las propiedades use\_compression y compression\_mode se proporcionan para dar soporte a esta característica, como se indica a continuación.

Tabla 202. Propiedades del nodo databaseexport utilizando funciones de compresión.

Propiedades del nodo databaseexport	Tipo de datos	Descripción de la propiedad
use_compression	booleano	Si se establece en true, crea tablas para la exportación con compresión.

Tabla 202. Propiedades del nodo databaseexport utilizando funciones de compresión (continuación).

Propiedades del nodo databaseexport	Tipo de datos	Descripción de la propiedad
compression_mode	Row Page	Establece el nivel de compresión de las bases de datos de SQL Server.
	Predeterminado Direct_Load_Operations All_Operations Básico OLTP Query_High Query_Low Archive_High Archive_Low	Establece el nivel de compresión de las bases de datos de Oracle. Tenga en cuenta que los valores OLTP, Query_High, Query_Low, Archive_High y Archive_Low requieren un mínimo de Oracle 11gR2.

## Propiedades del nodo datacollectionexport



El nodo de exportación IBM SPSS Data Collection abre los datos en el formato utilizado por el software de investigación de mercados IBM SPSS Data Collection. Se debe instalar la biblioteca de datos de IBM SPSS Data Collection para utilizar este nodo.

Tabla 203. Propiedades del nodo datacollectionexport.

Propiedades del nodo datacollectionexport	Tipo de datos	Descripción de la propiedad
metadata_file	cadena	Nombre del archivo de metadatos que se va a exportar.
merge_metadata	Overwrite MergeCurrent	
enable_system_variables	booleano	Especifica si el archivo .mdd exportado debe incluir las variables de sistema de IBM SPSS Data Collection.
casedata_file	cadena	Nombre del archivo .sav donde se exportan los datos de casos.
generate_import	booleano	

## Propiedades del nodo excelexport



El nodo de exportación Excel ofrece los datos resultantes en formato de Microsoft Excel (.xls). Si lo desea, puede elegir iniciar automáticamente Excel y abrir el archivo exportado cuando se ejecute el nodo.

Tabla 204. Propiedades del nodo excelexport.

Propiedades del nodo excelexport	Tipo de datos	Descripción de la propiedad
full_filename	cadena	
excel_file_type	Excel2003 Excel2007	
export_mode	Create Append	



Tabla 204. Propiedades del nodo excelexport (continuación).

Propiedades del nodo excelexport	Tipo de datos	Descripción de la propiedad
inc_field_names	booleano	Especifica si los nombres de campos deben incluirse en la primera fila de la hoja de trabajo.
start_cell	cadena	Especifica la casilla de inicio de la exportación.
worksheet_name	cadena	Nombre de la hoja de trabajo que se va a escribir.
launch_application	booleano	Determina si Excel debe invocarse para el archivo resultante. Tenga en cuenta que deberá especificar la ruta para iniciar Excel en el cuadro de diálogo Aplicaciones de ayuda (menú Herramientas, Aplicaciones de ayuda).
generate_import	booleano	Determina si debe crearse un nodo Importar a Excel que leerá el archivo de datos exportado.

## Propiedades del nodo outputfile



El nodo Archivo sin formato produce datos en un archivo de texto delimitado. Esto es útil para exportar datos que se pueden leer con otro software de hoja de cálculo o de análisis.

Tabla 205. Propiedades del nodo outputfile.

Propiedades del nodo outputfile	Tipo de datos	Descripción de la propiedad
full_filename	cadena	Nombre del archivo de resultados.
write_mode	Overwrite Append	
inc_field_names	booleano	
use_newline_after_records	booleano	
delimit_mode	Comma Tab Space Other	
other_delimiter	char	
quote_mode	Ninguno Single Double Other	
other_quote	booleano	
generate_import	booleano	
codificación	StreamDefault SystemDefault "UTF-8"	

## Propiedades del nodo sasexport



El nodo Exportar SAS produce datos en formato SAS, para leerlos en SAS o en un paquete de software compatible con SAS. Están disponibles tres formatos de archivo SAS: SAS para Windows/OS2, SAS para UNIX o SAS versión 7/8.

Tabla 206. Propiedades del nodo sasexport.

Propiedades del nodo sasexport	Tipo de datos	Descripción de la propiedad
format	Windows UNIX SAS7 SAS8	Campos de etiquetas de propiedad de variantes.
full_filename	<i>cadena</i>	
export_names	NamesAndLabels NamesAsLabels	Se utiliza para correlacionar los nombres de campos de IBM SPSS Modeler que se vayan a exportar a nombres de variables de IBM SPSS Statistics o SAS.
generate_import	<i>booleano</i>	

## Propiedades de nodo statisticsexport



El nodo Exportar Statistics ofrece los resultados en formato *.sav* de IBM SPSS Statistics. Los archivos *.sav* se pueden leer con IBM SPSS Statistics Base y otros productos. Este es también el formato utilizado para los archivos caché de IBM SPSS Modeler.

Las propiedades de este nodo están descritas en “Propiedades de nodo statisticsexport” en la página 240.

## Propiedades del nodo xmlexport



El nodo de exportación XML exporta datos a un archivo en formato XML. También puede crear un nodo de origen XML para leer los datos exportados a la ruta.

Tabla 207. Propiedades del nodo xmlexport.

Propiedades del nodo xmlexport	Tipo de datos	Descripción de la propiedad
full_filename	<i>cadena</i>	(obligatorio) Ruta completa y nombre de archivo del archivo XML para exportar.
use_xml_schema	<i>booleano</i>	Especifica si utilizar un esquema XML (archivo XSD o DTD) para controlar la estructura de los datos exportados.
full_schema_filename	<i>cadena</i>	Ruta completa y nombre de archivo del archivo XSD o DTD que se quiere utilizar. Es obligatorio si use_xml_schema está establecido como true.

Tabla 207. Propiedades del nodo `xmlexport` (continuación).

Propiedades del nodo <code>xmlexport</code>	Tipo de datos	Descripción de la propiedad
<code>generate_import</code>	<i>booleano</i>	Genera un nodo de origen XML que leerá el archivo de datos exportados de nuevo en la ruta.
<code>records</code>	<i>cadena</i>	Expresión de XPath que denota el límite de registro.
<code>map</code>	<i>cadena</i>	Correlaciona el nombre de campo a la estructura XML.



---

## Capítulo 18. Propiedades de nodos de IBM SPSS Statistics

---

### Propiedades del nodo statisticsimport



El nodo Archivo Statistics lee los datos desde un formato de archivo *.sav* que utiliza IBM SPSS Statistics y archivos caché guardados en IBM SPSS Modeler, que también puede utilizar el mismo formato.

Tabla 208. propiedades del nodo statisticsimport.

Propiedades del nodo statisticsimport	Tipo de datos	Descripción de la propiedad
full_filename	<i>cadena</i>	El nombre completo del archivo, incluyendo la ruta.
import_names	NamesAndLabels LabelsAsNames	Método para gestionar nombres y etiquetas de variables.
import_data	DataAndLabels LabelsAsData	Método para gestionar valores y etiquetas.
use_field_format_for_storage	<i>Booleana</i>	Especifica si se utiliza la información de formato de campo de IBM SPSS Statistics al importar.

---

### Propiedades del nodo statisticstransform



El nodo Transformación Statistics ejecuta una selección de comandos de sintaxis de IBM SPSS Statistics en los orígenes de datos de IBM SPSS Modeler. Este nodo requiere una copia de IBM SPSS Statistics con licencia.

Tabla 209. Propiedades del nodo statisticstransform.

Propiedades del nodo statisticstransform	Tipo de datos	Descripción de la propiedad
syntax	<i>cadena</i>	
check_before_saving	<i>booleano</i>	Valida la sintaxis introducida antes de guardar las entradas. Muestra un mensaje de error si la sintaxis no es válida.
default_include	<i>booleano</i>	Consulte el tema “Propiedades del nodo filter” en la página 101 para obtener más información.
include	<i>booleano</i>	Consulte el tema “Propiedades del nodo filter” en la página 101 para obtener más información.
new_name	<i>cadena</i>	Consulte el tema “Propiedades del nodo filter” en la página 101 para obtener más información.

---

## Propiedades del nodo `statisticsmodel`



El nodo Modelo Statistics permite analizar y trabajar con sus datos ejecutando los procedimientos de IBM SPSS Statistics que producen PMML. Este nodo requiere una copia de IBM SPSS Statistics con licencia.

Propiedades del nodo <code>statisticsmodel</code>	Tipo de datos	Descripción de la propiedad
<code>syntax</code>	<i>cadena</i>	
<code>default_include</code>	<i>booleano</i>	Consulte el tema "Propiedades del nodo filter" en la página 101 para obtener más información.
<code>include</code>	<i>booleano</i>	Consulte el tema "Propiedades del nodo filter" en la página 101 para obtener más información.
<code>new_name</code>	<i>cadena</i>	Consulte el tema "Propiedades del nodo filter" en la página 101 para obtener más información.

---

## Propiedades del nodo `statisticsoutput`



El nodo Resultados de Statistics le permite llamar a un procedimiento de IBM SPSS Statistics para analizar los datos de IBM SPSS Modeler. Se puede acceder a una gran variedad de procedimientos analíticos de IBM SPSS Statistics. Este nodo requiere una copia de IBM SPSS Statistics con licencia.

Tabla 210. Propiedades del nodo `statisticsoutput`.

Propiedades del nodo <code>statisticsoutput</code>	Tipo de datos	Descripción de la propiedad
<code>mode</code>	Dialog Syntax	Selecciona la opción "cuadro de diálogo de IBM SPSS Statistics" o el editor Sintaxis
<code>syntax</code>	<i>cadena</i>	
<code>use_output_name</code>	<i>booleano</i>	
<code>output_name</code>	<i>cadena</i>	
<code>output_mode</code>	Screen File	
<code>full_filename</code>	<i>cadena</i>	
<code>file_type</code>	HTML SPV SPW	

---

## Propiedades de nodo `statisticsexport`



El nodo Exportar Statistics ofrece los resultados en formato `.sav` de IBM SPSS Statistics. Los archivos `.sav` se pueden leer con IBM SPSS Statistics Base y otros productos. Este es también el formato utilizado para los archivos caché de IBM SPSS Modeler.

Tabla 211. Propiedades del nodo *statisticsexport*.

Propiedades del nodo <i>statisticsexport</i>	Tipo de datos	Descripción de la propiedad
<i>full_filename</i>	<i>cadena</i>	
<i>launch_application</i>	<i>booleano</i>	
<i>export_names</i>	NamesAndLabels NamesAsLabels	Se utiliza para correlacionar los nombres de campos de IBM SPSS Modeler que se vayan a exportar a nombres de variables de IBM SPSS Statistics o SAS.
<i>generate_import</i>	<i>booleano</i>	





---

## Capítulo 19. Propiedades de Supernodos

En las siguientes tablas se describen las propiedades específicas de los Supernodos. Tenga en cuenta que las propiedades de nodos comunes se aplican también a los Supernodos.

Tabla 212. Propiedades del supernodo de terminal.

Nombre de la propiedad	Tipo de propiedad / Lista de valores	Descripción de la propiedad
execute_method	Script Normal	
script	<i>cadena</i>	
script_language	Python Legacy	Establece el lenguaje de script para el script Supernodo.

### Parámetros de Supernodos

Pueden utilizarse scripts para crear o establecer parámetros de Supernodo utilizando las mismas funciones que se utilizan para modificar los parámetros de ruta. Consulte el tema “Parámetros de ruta, sesión y Supernodo” en la página 40 para obtener más información.

### Configuración de las propiedades de nodos encapsulados

Para establecer las propiedades de los nodos contenidos en el Supernodo, debe acceder al diagrama propiedad de dicho Supernodo y, a continuación utilizar los diferentes métodos find (tal como `findByName()` y `findById()`) para localizar los nodos. Por ejemplo, en un script Supernodo que incluye un nodo Tipo individual:

```
supernode = modeler.script.supernode()
diagram = supernode.getCompositeProcessorDiagram()
#FFind the type node within the supernode internal diagram
typenode = diagram.findByName("type", None)
typenode.setKeyedProperty("direction", "Drug", "Input")
typenode.setKeyedProperty("direction", "Age", "Target")
```

**Limitaciones de los scripts de Supernodos.** Los Supernodos no pueden manipular otras rutas y no pueden cambiar la ruta actual.



---

## Apéndice A. Referencia de nombres de nodo

Esta sección ofrece una referencia de todos los nombres de script de los nodos de IBM SPSS Modeler.

---

### Nombres de nugget de modelo

Se puede hacer referencia a los nugget de modelo (también denominados modelos generados) según el tipo, como con los objetos de nodo y de resultado. Las siguientes tablas muestran los nombres de referencia de los objetos del modelo.

Tenga en cuenta que estos nombres se utilizan específicamente para hacer referencia a los nugget de modelo en la paleta Modelos (en la esquina superior derecha de la ventana de IBM SPSS Modeler). Para hacer referencia a los nodos de modelo que se han añadido a una ruta para la puntuación, se utiliza un conjunto diferente de nombres con el prefijo `apply...`. Consulte el tema Capítulo 14, “Propiedades de nodos de nugget de modelo”, en la página 177 para obtener más información.

*Nota:* En circunstancias normales, se recomienda hacer referencia a los modelos por nombre y tipo para evitar confusiones.

*Tabla 213. Nombres de nugget de modelo (paleta de modelado).*

Nombre del modelo	Modelo
anomalydetection	Anomalía
a priori	A priori
autoclassifier	Clasificador automático
autocluster	Agrupación en clústeres automática
autonumeric	Autonumérico
bayesnet	Red bayesiana
c50	C5.0
carma	Carma
árbol cr	Árbol C&R
chaid	CHAID
coxreg	Regresión de Cox
decisionlist	Lista de decisiones
discriminant	Discriminante
factor	PCA/Factorial
featureselection	Sel. características
genlin	Regresión lineal generalizada
glmm	GLMM
kmeans	K-medias
knn	<i>k</i> : vecino más cercano
kohonen	Kohonen
lineal	Lineal
logreg	Regresión logística
neuralnetwork	Red neuronal

Tabla 213. Nombres de nugget de modelo (paleta de modelado) (continuación).

Nombre del modelo	Modelo
quest	QUEST
regresión	Regresión lineal
secuencia	Secuencia
slrm	Modelo de respuesta de autoaprendizaje
statisticsmodel	Modelo de IBM SPSS Statistics
svm	Máquina de vectores de soporte
timeseries	Serie temporal
twostep	Dos fases

Tabla 214. Nombres de nugget de modelo (paleta de modelado de bases de datos).

Nombre del modelo	Modelo
db2imcluster	Clúster de IBM ISW
db2imlog	Regresión logística de IBM ISW
db2imnb	Bayesiano ingenuo de IBM ISW
db2imreg	Regresión de IBM ISW
db2imtree	Árbol de decisión de IBM ISW
msassoc	Reglas de asociación de MS
msbayes	Bayesiano ingenuo de MS
mscluster	Clúster de MS
mslogistic	Regresión logística de MS
msneuralnetwork	Red neuronal de MS
msregression	Regresión lineal de MS
mssequencecluster	Clúster de secuencias de MS
mstimeseries	Serie temporales de MS
mstree	Árbol de decisión de MS
netezzabayes	Red bayesiana de Netezza
netezzadectree	Árbol de decisión de Netezza
netezzadivcluster	Clúster divisivo de Netezza
netezzaglm	Lineal generalizado de Netezza
netezzakmeans	K-medias de Netezza
netezzaknn	KNN de Netezza
netezzalineression	Regresión lineal de Netezza
netezzanaivebayes	Bayesiano ingenuo de Netezza
netezzapca	PCA de Netezza
netezzaregtree	Árbol de regresión de Netezza
netezzatimeseries	Serie temporales de Netezza
oraabn	Bayesiano adaptativo de Oracle
oraai	Oracle AI
oradecisiontree	Árbol de decisión de Oracle
oraglm	GLM de Oracle

Tabla 214. Nombres de nugget de modelo (paleta de modelado de bases de datos) (continuación).

Nombre del modelo	Modelo
orakmeans	K-medias de Oracle
oranb	Bayesiano ingenuo de Oracle
oranmf	NMF de Oracle
oracluster	O-clúster de Oracle
orasvm	SVM de Oracle

## Evitar nombres duplicados del modelo

Al utilizar los scripts para manipular los modelos generados, debe tener en cuenta que el hecho de permitir nombres de modelo duplicados puede originar referencias ambiguas. Para evitarlo, resulta útil utilizar nombres exclusivos para los modelos generados en los scripts.

Para configurar las opciones de los nombres de modelo duplicados:

1. Seleccione en los menús:  
**Herramientas > Opciones de usuario**
2. Pulse en la pestaña **Notificaciones**.
3. Seleccione **Sustituir modelo anterior** para restringir los nombres duplicados de los modelos generados.

El comportamiento de la ejecución de scripts puede variar entre SPSS Modeler y IBM SPSS Collaboration and Deployment Services cuando haya referencias de modelo ambiguas. El cliente de SPSS Modeler incluye la opción "Reemplazar modelo anterior", que reemplaza automáticamente los modelos que tengan el mismo nombre (por ejemplo, cuando un script se itera a través de un bucle para producir un modelo diferente cada vez). Sin embargo, esta opción no está disponible cuando el mismo script se ejecuta en IBM SPSS Collaboration and Deployment Services. Puede evitar esta situación cambiando el nombre del modelo generado en cada iteración para evitar referencias ambiguas a los modelos o borrando el modelo actual (por ejemplo, añadiendo una instrucción `clear generated palette`) antes del final del bucle.

## Nombres de tipo de resultados

La siguiente tabla indica los tipos de objetos de resultados y los nodos que los crean. Para obtener una lista completa de los formatos de exportación disponibles para cada tipo de objeto de salida, consulte la descripción de las propiedades del nodo que crea el tipo de salida, disponible en "Propiedades comunes de nodos Gráfico" en la página 117 y en Capítulo 16, "Propiedades de los nodos de resultados", en la página 215.

Tabla 215. Tipos de objeto de salida y los nodos que los crean.

Tipo de objeto de resultado	Nodo
analysisoutput	Análisis
collectionoutput	Colección
dataauditoutput	Auditoría de datos
distributionoutput	Distribución
evaluationoutput	Evaluación
histogramoutput	Histograma
matrixoutput	Matriz
meansoutput	Medias
multiplotoutput	G. múltiple

Tabla 215. Tipos de objeto de salida y los nodos que los crean (continuación).

Tipo de objeto de resultado	Nodo
plotoutput	Gráfico
qualityoutput	Calidad
reportdocumentoutput	Este tipo de objeto no es de un nodo, es un resultado creado por un informe de proyecto
reportoutput	Informe
statisticsprocedureoutput	Resultado de Estadísticas
statisticsoutput	Estadísticos
tableoutput	Tabla
timeplotoutput	Gráfico de tiempo
weboutput	Web

---

## Apéndice B. Migración desde scripts de herencia a scripts Python

---

### Visión general de la migración de scripts de herencia

Esta sección proporciona un resumen de las diferencias entre el script de Python y el script de herencia en IBM SPSS Modeler y proporciona información acerca de cómo migrar los scripts de herencia a scripts Python. En esta sección encontrar una lista de los comandos de herencia de SPSS Modeler estándar y los comandos Python equivalentes.

---

### Diferencias generales

Una gran parte del diseño de los scripts de herencia se debe a los scripts de comandos del sistema operativo. Los scripts de herencia están orientados a líneas y, aunque existen algunas estructuras de bloque, por ejemplo `if...then...else...endif` y `for...endfor`, generalmente la indentación no es importante.

En los scripts Python, la indentación es importante y las líneas que pertenecen al mismo bloque lógico se deben indentar en el mismo nivel.

**Nota:** Debe prestar atención cuando copie y pegue el código Python. En el editor, una línea que se ha indentado utilizando pestañas puede parecer la misma que una línea que se ha indentado utilizando espacios. Sin embargo, el script Python generará un error porque no se considera que la indentación de las líneas sea la misma.

---

### El contexto de los scripts

El contexto de script define el entorno en el que se ejecuta un script como, por ejemplo, la ruta o Supernodo que ejecuta el script. En los scripts heredados el contexto es implícito, lo que significa que, por ejemplo, se asume que toda referencia a un nodo de una ruta está dentro de la ruta que ejecuta el script.

En los scripts Python el contexto de script se proporciona de forma explícita mediante el módulo `modeler.script`. Por ejemplo, un script Python de ruta puede acceder a la ruta que ejecuta el script mediante el código siguiente:

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

A continuación podrán invocarse funciones relacionadas con la ruta a través del objeto devuelto.

---

### Comparativa de comandos y funciones

Los scripts heredados están orientados a comando. Esto significa que cada línea del script suele comenzar con el comando a ejecutar seguido de los parámetros, por ejemplo:

```
connect 'Type':typenode to :filternode  
rename :derivenode as "Compute Total"
```

Python utiliza funciones que suelen invocarse a través de un objeto (módulo, clase u objeto) que define la función, por ejemplo:

```
ruta = modeler.script.stream()  
nodotipo= ruta.findByName("type", "Type")  
nodofiltro = ruta.findByName("filter", None)  
ruta.link(nodotipo, nodofiltro)  
derive.setLabel("Compute Total")
```

## Literales y comentarios

Algunos de los literales y comandos de comentarios que normalmente se utilizan en IBM SPSS Modeler tienen sus equivalentes en los scripts Python. Esto puede ayudarle a convertir los scripts de SPSS Modeler de herencia existentes en scripts Python para utilizarlos en IBM SPSS Modeler 16.

Tabla 216. Correlación de scripts de herencia con scripts Python para literales y comentarios.

Scripts de herencia	scripts Python
Entero, por ejemplo 4	El mismo
Flotante, por ejemplo, 0,003	El mismo
Cadenas entre comillas simples, por ejemplo, 'Hola'	El mismo <b>Nota:</b> Los literales de cadena que contengan caracteres que no sean ASCII deberán tener el prefijo u para garantizar que se representen en Unicode.
Cadenas entre comillas dobles, por ejemplo, "Hola de nuevo"	El mismo <b>Nota:</b> Los literales de cadena que contengan caracteres que no sean ASCII deberán tener el prefijo u para garantizar que se representen en Unicode.
Cadenas largas, por ejemplo, """Estas es una cadena que abarca varias líneas"""	El mismo
Listas, por ejemplo, [1 2 3]	[1, 2, 3]
Referencia de variable, por ejemplo, set x = 3	x = 3
Continuación de línea (\), por ejemplo, set x = [1 2 \ 3 4]	x = [ 1, 2,\n3, 4]
Comentario de bloque, por ejemplo, /* Éste es un comentario largo a través de una línea. */	/* Éste es un comentario largo a través de una línea. */
Comentario de línea, por ejemplo, set x = 3 # make x 3	x = 3 # make x 3
undef	None
true	True
false	False

## Operadores

Algunos de los comandos de operadores que normalmente se utilizan en IBM SPSS Modeler tienen sus comandos equivalentes en los scripts Python. Esto puede ayudarle a convertir los scripts de SPSS Modeler de herencia existentes en scripts Python para utilizarlos en IBM SPSS Modeler 16.

Tabla 217. Correlación de scripts de herencia con scripts Python para operadores.

Scripts de herencia	scripts 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



Tabla 217. Correlación de scripts de herencia con scripts Python para operadores (continuación).

Scripts de herencia	scripts 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
and or not(EXPR)	and or not EXPR

## Comandos condicionales y de bucle

Algunos comandos condicionales y de bucle utilizados habitualmente en IBM SPSS Modeler tienen sus comandos equivalentes en los scripts Python. Esto puede ayudarle a convertir los scripts de SPSS Modeler de herencia existentes en scripts Python para utilizarlos en IBM SPSS Modeler 16.

Tabla 218. Correspondencia de scripts de herencia con scripts Python en lo referente a comandos condicionales y de bucle.

Scripts de herencia	scripts Python
for VAR from INT1 to INT2 ... endfor	for VAR in range(INT1, INT2): ...  o VAR = INT1 while VAR <= INT2: ... VAR += 1
for VAR in LIST ... endfor	for VAR in LIST: ...
for VAR in_fields_to NODE ... endfor	for VAR in NODE.getInputDataModel(): ...
for VAR in_fields_at NODE ... endfor	for VAR in NODE.getOutputDataModel(): ...
if...then ... elseif...then ... else ... endif	if ...: ... elif ...: ... else: ...
with TYPE OBJECT ... endwith	Sin equivalente

Tabla 218. Correspondencia de scripts de herencia con scripts Python en lo referente a comandos condicionales y de bucle (continuación).

Scripts de herencia	scripts Python
var VARI	La declaración de variables no es obligatoria

---

## Variables

En los scripts heredados, las variables se declaran antes de ser referenciadas, por ejemplo:

```
var minodo
set minodo = create typenode at 96 96
```

En los scripts Python, las variables se crean la primera vez que se referencian, por ejemplo:

```
minodo = stream.createAt("type", "Type", 96, 96)
```

En los scripts heredados, las referencias a variables deben eliminarse explícitamente mediante el operador `^`, por ejemplo:

```
var minodo
set minodo = create typenode at 96 96
set ^minodo.direction."Age" = Input
```

Al igual que en la mayoría de lenguajes de script, esto no es necesario en los scripts Python, por ejemplo:

```
minodo = stream.createAt("type", "Type", 96, 96)
minodo.setKeyedPropertyValue("direction", "Age", "Input")
```

---

## Tipos modelo, resultado y nodo

En los scripts heredados, a los distintos tipos de objeto (nodo, resultado y modelo) se les suele añadir el tipo al tipo de objeto. Por ejemplo, el nodo Derivar es de tipo `derivenode`:

```
set feature_name_node = create derivenode at 96 96
```

El API de IBM SPSS Modeler en Python no incluye el sufijo `node`, de modo que el nodo Derivar tiene el tipo `derive`, por ejemplo:

```
feature_name_node = stream.createAt("derive", "Feature", 96, 96)
```

La única diferencia en los tipos de nombre entre los scripts Python y los heredados es la ausencia del sufijo de tipo.

---

## Nombres de propiedades

Los nombres de las propiedades son los mismos en scripts heredados y en scripts Python. Por ejemplo, en el nodo Archivo variable, la propiedad que define la ubicación del archivo es `full_filename` en ambos entornos de creación de scripts.

---

## Referencias de nodos

Muchos scripts de herencia utilizan una búsqueda implícita para buscar y acceder al nodo que se ha de modificar. Por ejemplo, los comandos siguientes buscan en la ruta actual un nodo Type con la etiqueta "Type", a continuación, establecen la dirección (o el rol de modelado) del campo "Age" como entrada y el campo "Drug" como destino, esto es, el valor predicho:

```
set 'Type':typenode.direction."Age" = Input
set 'Type':typenode.direction."Drug" = Target
```

En los scripts Python, los objetos de nodo se han de localizar de forma explícita antes de llamar a la función para establecer el valor de propiedad, por ejemplo:

```
typenode = stream.findByType("type", "Type")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
typenode.setKeyedPropertyValue("direction", "Drug", "Target")
```

**Nota:** En este caso, "Target" debe estar encerrado entre comillas en la cadena.

Los scripts Python pueden utilizar de forma alternativa la enumeración `ModelingRole` del paquete `modeler.api`.

Aunque la versión de los scripts Python puede ser más verbosa, el rendimiento de tiempo de ejecución es mejor ya que la búsqueda del nodo generalmente solo se realiza una vez. En el ejemplo de scripts de herencia, la búsqueda del nodo se realiza para cada comando.

También está soportado buscar nodos por ID (el ID de nodo se puede ver en la pestaña Anotaciones del diálogo del nodo). Por ejemplo, en los scripts de herencia:

```
# id65EMPB9VL87 es el ID de un nodo Type
set @id65EMPB9VL87.direction."Age" = Input
```

El script siguiente muestra el mismo ejemplo en scripts Python:

```
typenode = stream.findByID("id65EMPB9VL87")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
```

---

## Obtener y establecer propiedades

Los scripts de herencia utilizan el comando `set` para asignar un valor. El término que sigue al comando `set` puede ser una definición de propiedad. El script siguiente muestra dos formatos de script posibles para establecer una propiedad:

```
set <referencia de nodo>.<propiedad> = <valor>
set <referencia de nodo>.<propiedad-con claves>.<clave> = <valor>
```

En los scripts Python, se obtiene el mismo resultado utilizando las funciones `setProperty()` y `setKeyedPropertyValue()`, por ejemplo:

```
objeto.setProperty(propiedad, valor)
objeto.setKeyedPropertyValue(propiedad-con claves, clave, valor)
```

En los scripts de herencia, se puede acceder a los valores de las propiedades utilizando el comando `get`, por ejemplo:

```
var n v
set n = get node :filternode
set v = ^n.name
```

En los scripts Python, se obtiene el mismo resultado utilizando la función `getProperty()`, por ejemplo:

```
n = stream.findByName("filter", None)
v = n.getProperty("name")
```

---

## Edición de rutas

En los scripts de herencia, se utiliza el comando `create` para crear un nodo nuevo, por ejemplo:

```
var agg select
set agg = create aggregatenode at 96 96
set select = create selectnode at 164 96
```

En los scripts Python, las rutas tienen varios métodos para crear nodos, por ejemplo:

```
stream = modeler.script.stream()
agg = stream.createAt("aggregate", "Aggregate", 96, 96)
select = stream.createAt("select", "Select", 164, 96)
```

En los scripts de herencia, se utiliza el comando `connect` para crear enlaces entre nodos, por ejemplo:  
`connect ^agg to ^select`

En los scripts Python, se utiliza el método `link` para crear enlaces entre nodos, por ejemplo:  
`stream.link(agg, select)`

En los scripts de herencia, se utiliza el comando `disconnect` para eliminar enlaces entre nodos, por ejemplo:  
`disconnect ^agg from ^select`

En los scripts Python, se utiliza el método `unlink` para eliminar enlaces entre nodos, por ejemplo:  
`stream.unlink(agg, select)`

En los scripts de herencia, se utiliza el comando `position` para posicionar los nodos en el lienzo de rutas o entre nodos, por ejemplo:  
`position ^agg at 256 256`  
`position ^agg between ^myselect and ^mydistinct`

En los scripts Python, se obtiene el mismo resultado utilizando dos métodos separados: `setXYPosition` y `setPositionBetween`. Por ejemplo:  
`agg.setXYPosition(256, 256)`  
`agg.setPositionBetween(myselect, mydistinct)`

## Operaciones de nodo

Algunos de los comandos de operaciones de nodo que normalmente se utilizan en IBM SPSS Modeler tienen sus comandos equivalentes en los scripts Python. Esto puede ayudarle a convertir los scripts de SPSS Modeler de herencia existentes en scripts Python para utilizarlos en IBM SPSS Modeler 16.

Tabla 219. Correlación de scripts de herencia con scripts Python para operaciones de nodo.

Scripts de herencia	scripts Python
<code>create especificaciónodo at x y</code>	<code>ruta.create(tipo, nombre)</code> <code>ruta.createAt(tipo, nombre, x, y)</code> <code>ruta.createBetween(tipo, nombre, preNode, postNode)</code> <code>ruta.createModelApplier(modelo, nombre)</code>
<code>connect desdeNodo to aNodo</code>	<code>ruta.link(desdeNodo, aNodo)</code>
<code>delete nodo</code>	<code>ruta.delete(nodo)</code>
<code>disable nodo</code>	<code>ruta.setEnabled(nodo, False)</code>
<code>enable nodo</code>	<code>ruta.setEnabled(nodo, True)</code>
<code>disconnect desdeNodo from aNodo</code>	<code>ruta.unlink(desdeNodo, aNodo)</code> <code>ruta.disconnect(nodo)</code>
<code>duplicate nodo</code>	<code>nodo.duplicate()</code>
<code>execute nodo</code>	<code>ruta.runSelected(nodos, resultados)</code> <code>ruta.runAll(resultados)</code>
<code>flush nodo</code>	<code>nodo.flushCache()</code>
<code>position nodo at x y</code>	<code>nodo.setXYPosition(x, y)</code>
<code>position nodo between nodo1 and nodo2</code>	<code>nodo.setPositionBetween(nodo1, nodo2)</code>
<code>rename nodo as nombre</code>	<code>nodo.setLabel(nombre)</code>

---

## Bucle

En los scripts de herencia, hay dos opciones de bucle principales a las que se da soporte:

- Bucles de *Valor contado*, en los que una variable de índice se mueve entre dos límites de entero.
- Bucles de *secuencia* que avanzan en bucle por una secuencia de valores, enlazando el valor actual con la variable de bucle.

El script siguiente es un ejemplo de un bucle de valor contado en un script de herencia:

```
for i from 1 to 10
  println ^i
endfor
```

El script siguiente es un ejemplo de un bucle de secuencia en un script de herencia:

```
var items
set items = [a b c d]

for i in items
  println ^i
endfor
```

También existen otros tipos de bucles que se pueden utilizar:

- Iteración por los modelos de la paleta de modelos o por los resultados de la paleta de resultados.
- Iteración por los campos de entrada o salida de un nodo.

Los scripts Python también dan soporte a diferentes tipos de bucles. El script siguiente es un ejemplo de un bucle de valor contado en un script Python:

```
i = 1
while i <= 10:
  print i
  i += 1
```

El script siguiente es un ejemplo de un bucle de secuencia en un script Python:

```
items = ["a", "b", "c", "d"]
for i in items:
  print i
```

El bucle de secuencia es muy flexible y cuando se combina con los métodos de la API de IBM SPSS Modeler puede dar soporte a la mayoría de los casos de uso scripts de herencia. El siguiente ejemplo muestra cómo utilizar un bucle de secuencia en scripts Python para iterar por los campos de salida de un nodo:

```
node = modeler.script.stream().findByType("filter", None)
for column in node.getOutputDataModel().columnIterator():
  print column.getColumnName()
```

---

## Ejecución de rutas

Durante la ejecución de la secuencia, el modelo o los objetos de resultados que se generan se añaden a uno de los gestores de objeto. En el script existente, el script debe localizar los objetos creados desde el gestor de objeto, o acceder al resultado generado más recientemente desde el nodo que ha generado el resultado.

La ejecución de rutas en Python es diferente, ya que cualquier objeto de modelo o resultados que genere la ejecución se devuelve una lista que se pasa a la función de ejecución. Esto hace que resulte más sencillo acceder a los resultados de la ejecución de la ruta.

Los scripts de herencia dan soporte a tres comandos de ejecución de ruta:

- `execute_all` ejecuta todos nodos terminales ejecutables en la ruta.
- `execute_script` ejecuta el script de ruta independientemente del valor de la ejecución del script.
- `execute_nodo` ejecuta el nodo especificado.

Los scripts Python dan soporte a un conjunto de funciones similares:

- `ruta.runAll(lista-resultados)` ejecuta todos los nodos terminales ejecutables de la ruta.
- `ruta.runScript(lista-resultados)` ejecuta el script de ruta independientemente del valor de la ejecución del script.
- `ruta.runSelected(matriz-nodos, lista-resultados)` ejecuta el conjunto de nodos especificados en el orden en que se suministran.
- `nodo.run(lista-resultados)` ejecuta el nodo especificado.

En los scripts de herencia, la ejecución de ruta se puede finalizar con el comando `exit` con un código de entero opcional, por ejemplo:

```
exit 1
```

En los scripts Python, se puede obtener el mismo resultado con el script siguiente:

```
modeler.script.exit(1)
```

---

## Acceso a objetos mediante el sistema de archivos y el repositorio

En los scripts heredados se puede abrir una ruta, un modelo o un resultado existentes mediante el comando `open`, por ejemplo:

```
var s
set s = open stream "c:/my streams/modeling.str"
```

En los scripts Python, existe la clase `TaskRunner`, accesible desde la sesión, que puede utilizarse para realizar tareas similares, por ejemplo:

```
taskrunner = modeler.script.session().getTaskRunner()
s = taskrunner.openStreamFromFile("c:/my streams/modeling.str", True)
```

Para guardar un objeto en los scripts heredados, puede utilizarse el comando `save`, por ejemplo:

```
save stream s as "c:/my streams/new_modeling.str"
```

El enfoque de un script Python consiste en utilizar la clase `TaskRunner`, por ejemplo:

```
taskrunner.saveStreamToFile(s, "c:/my streams/new_modeling.str")
```

Las operaciones basadas en un IBM SPSS Collaboration and Deployment Services Repository se soportan en los scripts heredados mediante los comandos `retrieve` y `store`, por ejemplo:

```
var s
set s = retrieve stream "/my repository folder/my_stream.str"
store stream ^s as "/my repository folder/my_stream_copy.str"
```

En los scripts Python, se accede a la funcionalidad equivalente a través del objeto `Repository` asociado a la sesión, por ejemplo:

```
session = modeler.script.session()
repo = session.getRepository()
s = repo.retrieveStream("/my repository folder/my_stream.str", None, None, True)
repo.storeStream(s, "/my repository folder/my_stream_copy.str", None)
```

**Nota:** El acceso al repositorio exige que la sesión se haya configurado con una conexión de repositorio válida.

## Operaciones de ruta

Algunos comandos de operación de ruta que normalmente se utilizan en IBM SPSS Modeler tienen sus comandos equivalentes en los scripts Python. Esto puede ayudarle a convertir los scripts de SPSS Modeler de herencia existentes en scripts Python para utilizarlos en IBM SPSS Modeler 16.

Tabla 220. Correlación de scripts de herencia con scripts Python para operaciones de ruta.

Scripts de herencia	scripts Python
create stream <i>NOMBREARCHIVO_PREDETERMINADO</i>	<i>ejecutortareas.createStream(nombre, autoConectar, autoGestionar)</i>
close stream	<i>ruta.close()</i>
clear stream	<i>ruta.clear()</i>
get stream <i>ruta</i>	Sin equivalente
load stream <i>vía de acceso</i>	Sin equivalente
open stream <i>vía de acceso</i>	<i>ejecutortareas.openStreamFromFile(vía de acceso, autoGestionar)</i>
save <i>ruta</i> as <i>vía de acceso</i>	<i>ejecutortareas.saveStreamToFile(ruta, vía de acceso)</i>
retrieve stream <i>vía de acceso</i>	<i>repositorio.retrieveStream(vía de acceso, versión, etiqueta, autoGestionar)</i>
store <i>ruta</i> as <i>vía de acceso</i>	<i>repositorio.storeStream(ruta, vía de acceso, etiqueta)</i>

## Operaciones de modelo

Algunos de los comandos de operación de modelo que normalmente se utilizan en IBM SPSS Modeler tienen sus comandos equivalentes en los scripts Python. Esto puede ayudarle a convertir los scripts de SPSS Modeler de herencia existentes en scripts Python para utilizarlos en IBM SPSS Modeler 16.

Tabla 221. Correlación de scripts de herencia con scripts Python para operaciones de modelo.

Scripts de herencia	scripts Python
open model <i>vía de acceso</i>	<i>ejecutortareas.openModelFromFile(vía de acceso, autoGestionar)</i>
save <i>modelo</i> as <i>vía de acceso</i>	<i>ejecutortareas.saveModelToFile(modelo, vía de acceso)</i>
retrieve model <i>vía de acceso</i>	<i>repositorio.retrieveModel(vía de acceso, versión, etiqueta, autoGestionar)</i>
store <i>modelo</i> as <i>vía de acceso</i>	<i>repositorio.storeModel(modelo, vía de acceso, etiqueta)</i>

## Operaciones de resultado de documento

Algunos de los comandos de operaciones de resultado de documento que normalmente se utilizan en IBM SPSS Modeler tienen sus comandos equivalentes en los scripts Python. Esto puede ayudarle a convertir los scripts de SPSS Modeler de herencia existentes en scripts Python para utilizarlos en IBM SPSS Modeler 16.

Tabla 222. Correlación de scripts de herencia con scripts Python para operaciones de resultado de documento.

Scripts de herencia	scripts Python
open output <i>vía de acceso</i>	<i>ejecutortareas.openDocumentFromFile(vía de acceso, autoGestionar)</i>
save <i>resultado</i> as <i>vía de acceso</i>	<i>ejecutortareas.saveDocumentToFile(resultado, vía de acceso)</i>

Tabla 222. Correlación de scripts de herencia con scripts Python para operaciones de resultado de documento (continuación).

Scripts de herencia	scripts Python
retrieve output <i>vía de acceso</i>	<code>repositorio.retrieveDocument(vía de acceso, versión, etiqueta, autoGestionar)</code>
store resultado as <i>vía de acceso</i>	<code>repositorio.storeDocument(resultado, vía de acceso, etiqueta)</code>

---

## Otras diferencias entre scripts heredados y scripts Python

Los scripts heredados soportan la manipulación de proyectos de IBM SPSS Modeler. Los scripts Python no soportan esto actualmente.

Los scripts heredados proporcionan cierto soporte de carga de objetos de *estado* (combinaciones de rutas y modelos). Los objetos de estado han caído en desuso desde IBM SPSS Modeler 8.0. Los scripts Python no soportan objetos de estado.

Los scripts Python proporcionan las siguientes funciones adicionales no disponibles en los scripts heredados:

- Definiciones de clase y función.
- Manejo de errores.
- Soporte más sofisticado de entrada/salida.
- Módulos externos y de terceros.



---

## Avisos

Esta información se ha desarrollado para los productos y servicios ofrecidos en todo el mundo.

Puede que en otros países IBM no ofrezca los productos, servicios ni características que se describen en esta información. Póngase en contacto con el representante local de IBM, que le informará sobre los productos y servicios disponibles actualmente en su área. Las referencias a programas, productos o servicios de IBM no pretenden establecer ni implicar que sólo puedan utilizarse dichos productos, programas o servicios de IBM. En su lugar se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no infrinja ningún derecho de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y comprobar el funcionamiento de todo producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que cubran la materia descrita en esta información. Este documento no le otorga ninguna licencia para estas patentes. Puede enviar preguntas acerca de las licencias, por escrito, a:

IBM Director of Licensing  
IBM Corporation  
North Castle  
Drive  
Armonk, NY 10504-1785  
EE. UU.

Para consultas de licencias relacionadas con información de doble byte (DBCS), póngase en contacto con el Departamento de Propiedad intelectual de IBM de su país o envíe sus consultas, por escrito, a:

Intellectual Property Licensing  
Derecho de propiedad intelectual y legal  
IBM Japan Ltd.  
1623-14,  
Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde dichas disposiciones entren en conflicto la legislación local: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN NINGÚN TIPO DE GARANTÍA, EXPLÍCITA O IMPLÍCITA, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERABILIDAD, COMERCIALIZACIÓN O ADECUACIÓN A UN PROPÓSITO DETERMINADO.

Algunos estados no permiten la renuncia a expresar o a garantías implícitas en determinadas transacciones, por lo tanto, esta declaración no se aplique a usted.

Esta información puede incluir imprecisiones técnicas o errores tipográficos. Periódicamente, se efectúan cambios en la información aquí y estos cambios se incorporarán en nuevas ediciones de la publicación. IBM puede realizar en cualquier momento mejoras o cambios en los productos o programas descritos en esta publicación sin previo aviso.

Cualquier referencia a sitios Web que no sean de IBM en esta información solamente es ofrecida por comodidad y de ningún modo sirve como aprobación de esos sitios Web. Los materiales de dichos sitios web no forman parte de los materiales de este producto IBM y el uso de dichos sitios web se realiza a cuenta y riesgo del usuario.

IBM puede utilizar o distribuir cualquier información que se le proporcione en la forma que considere adecuada, sin incurrir por ello en ninguna obligación para con el remitente.

Los titulares de licencias de este programa que deseen obtener información sobre el mismo con el fin de permitir: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido éste) y (ii) el uso mutuo de información que se haya intercambiado, deben ponerse en contacto con:

Tel. 901 100 400  
ATTN: Licensing  
200 W. Madison St.  
Chicago, IL; 60606  
Estados Unidos de América

Esta información estará disponible, bajo las condiciones adecuadas, incluyendo en algunos casos el pago de una cuota.

El programa bajo licencia descrito en este documento y todo el material bajo licencia disponible se proporcionan bajo los términos de IBM Customer Agreement, IBM International Program License Agreement o cualquier otro acuerdo equivalente entre IBM y el cliente.

Cualquier dato de rendimiento mencionado aquí ha sido determinado en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar de forma significativa. Es posible que algunas mediciones se hayan realizado en sistemas en desarrollo y no existe ninguna garantía de que estas medidas sean las mismas en los sistemas comerciales. Además, es posible que algunas mediciones hayan sido estimadas a través de extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben consultar los datos que corresponden a su entorno específico.

Se ha obtenido información acerca de productos que no son de IBM de los proveedores de esos productos, de sus publicaciones anunciadas o de otros orígenes disponibles públicamente. IBM no ha probado estos productos y no puede confirmar la precisión del rendimiento, la compatibilidad ni ninguna otra afirmación relacionada con productos que no son de IBM. Las preguntas acerca de las aptitudes de productos que no sean de IBM deben dirigirse a los proveedores de dichos productos.

Todas las declaraciones sobre el futuro del rumbo y la intención de IBM están sujetas a cambio o retirada sin previo aviso y representan únicamente metas y objetivos.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlos lo máximo posible, los ejemplos incluyen los nombres de las personas, empresas, marcas y productos. Todos esos nombres son ficticios y cualquier parecido con los nombres y direcciones utilizados por una empresa real es pura coincidencia.

Si está viendo esta información en copia electrónica, es posible que las fotografías y las ilustraciones en color no aparezcan.

---

## **Marcas comerciales**

IBM, el logotipo de IBM e [ibm.com](http://ibm.com) son marcas registradas o marcas comerciales de International Business Machines Corp., registradas en muchas jurisdicciones en todo el mundo. Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras empresas. Hay disponible una lista actual de las marcas registradas de IBM en el sitio web en "Información de copyright y de marcas registradas" en [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Intel, el logotipo de Intel, Intel Inside, el logotipo de Intel Inside, Intel Centrino, el logotipo de Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium y Pentium son marcas comerciales o marcas registradas de Intel Corporation o sus filiales en Estados Unidos y otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos, otros países o ambos.

Microsoft, Windows, Windows NT, y el logotipo de Windows son marcas comerciales de Microsoft Corporation en Estados Unidos, otros países o ambos.

UNIX es una marca registrada de The Open Group en Estados Unidos y otros países.

Java y todas las marcas registradas y logotipos basados en Java son marcas comerciales o marcas registradas de Oracle y/o sus filiales.

El resto de nombres de productos y servicios pueden ser marcas comerciales de IBM o de otras empresas.



---

# Índice

## A

- adición de atributos 22
- API de creación de scripts
  - acceso a objetos generados 38
  - búsqueda 35
  - ejemplo 35
  - introducción 35
  - metadatos 35
  - parámetros de ruta 40
  - parámetros de sesión 40
  - parámetros de Supernodos 40
  - scripts autónomos 45
  - tratamiento de errores 39
  - valores globales 44
  - varias rutas 45
- applylogreg node properties 183
- Árbol de decisión de MS
  - propiedades de scripts de nodos 187, 189
- argumentos
  - archivo de comandos 55
  - conexión con el servidor 54
  - conexión de IBM SPSS Collaboration and Deployment Services Repository 55
  - sistema 52
- atravesar los nodos 31

## B

- bloques de código 17
- bucles en rutas 4, 5
- buscar nodos 27

## C

- cadenas 15
- campos
  - desactivación de los scripts 117
- caracteres que no son ASCII 20
- clave de iteración
  - bucle en scripts 6
- Clúster de secuencias de MS
  - propiedades de scripts de nodos 189
- comando clear generated palette 48
- comentarios 16
- comprobación de errores
  - scripts 48
- configuración de propiedades 28
- contraseñas
  - adición a scripts 47
  - codificadas 54
- contraseñas codificadas
  - adición a scripts 47
- corrientes
  - bucle 4
  - ejecución condicional 4
  - scripts 1, 2, 25
- creación de nodos 29, 31
- crear una clase 22

## D

- definición de atributos 23
- definir métodos 23
- definir una clase 22
- Diagramas 25

## E

- ejecución condicional de rutas 4
- ejecución condicional en rutas 8
- Ejecución de rutas 25
- ejecución de scripts 9
- ejemplos 18
- encontrar y sustituir 10
- expresiones regulares 10

## F

- funciones
  - bucle 251
  - comentarios 250
  - condicionales 251
  - literales 250
  - operaciones de modelo 257
  - operaciones de nodo 254
  - operaciones de resultado de documento 257
  - operaciones de ruta 257
  - operadores 250
  - referencias a objeto 250

## H

- herencia 23

## I

- IBM SPSS Modeler
  - ejecución desde la línea de comandos 51
- identificadores 17
- interrupción de scripts 9

## J

- Jython 13

## L

- línea de comandos
  - ejecutar IBM SPSS Modeler 51
  - lista de argumentos 52, 54, 55
  - parámetros 53
  - scripts 48
  - varios argumentos 55
- listas 14

## M

- marcas
  - argumentos de la línea de comandos 51
  - combinación de varias marcas 55
- métodos matemáticos 19
- migrar
  - acceder a objetos 256
  - bucle 255
  - comandos 249
  - conceptos básicos 249
  - configuración de propiedades 253
  - contexto de los scripts 249
  - diferencias generales 249
  - editar rutas 253
  - ejecución de rutas 255
  - funciones 249
  - nombres de propiedad 252
  - obtener propiedades 253
  - referencias de nodos 252
  - repositorio 256
  - sistema de archivos 256
  - tipos de modelos 252
  - tipos de nodo 252
  - tipos de salida 252
  - variables 252
  - varios 258
- modelado de bases de datos 187
- modelos
  - nombres de scripts 245, 247
- modelos apriori
  - propiedades de scripts de nodos 131, 177
- Modelos Apriori de Oracle
  - propiedades de scripts de nodos 191, 196
- modelos autonuméricos
  - propiedades de scripts de nodos 135
- Modelos autonuméricos
  - propiedades de scripts de nodos 178
- Modelos bayesianos adaptativos de Oracle
  - propiedades de scripts de nodos 191, 196
- modelos bietápicos
  - propiedades de scripts de nodos 175, 186
- modelos C5.0
  - propiedades de scripts de nodos 138, 179
- modelos CARMA
  - propiedades de scripts de nodos 139, 179
- modelos CHAID
  - propiedades de scripts de nodos 141, 180
- Modelos de agrupación en clústeres de IBM ISW
  - propiedades de scripts de nodos 197, 202

Modelos de agrupación en clústeres divisivo de Netezza propiedades de scripts de nodos	203, 213	modelos de Microsoft propiedades de scripts de nodos	187, 189	Modelos GLMM propiedades de scripts de nodos	153, 182
modelos de árbol C&R propiedades de scripts de nodos	140, 179	Modelos de Netezza propiedades de scripts de nodos	203	modelos KNN propiedades de scripts de nodos	183
Modelos de árbol de decisión de IBM ISW propiedades de scripts de nodos	197, 202	Modelos de NMF de Oracle propiedades de scripts de nodos	191, 196	Modelos KNN de Netezza propiedades de scripts de nodos	203, 213
Modelos de árbol de decisión de Oracle propiedades de scripts de nodos	191, 196	modelos de Oracle propiedades de scripts de nodos	191	modelos kohonen propiedades de scripts de nodos	159
modelos de árboles de decisión de Netezza propiedades de scripts de nodos	203, 213	modelos de red bayesiana propiedades de scripts de nodos	136	modelos Kohonen propiedades de scripts de nodos	183
Modelos de árboles de regresión de Netezza propiedades de scripts de nodos	203, 213	Modelos de red bayesiana de Netezza propiedades de scripts de nodos	203, 213	modelos lineales propiedades de scripts de nodos	160, 183
Modelos de asociación de IBM ISW propiedades de scripts de nodos	197, 202	modelos de red neuronal propiedades de scripts de nodos	164, 183	modelos lineales generalizados propiedades de scripts de nodos	150, 182
modelos de Autoclúster propiedades de scripts de nodos	178	Modelos de redes bayesianas propiedades de scripts de nodos	179	Modelos lineales generalizados de Netezza propiedades de scripts de nodos	203
Modelos de bayesiano ingenuo de IBM ISW propiedades de scripts de nodos	197, 202	Modelos de regresión de Cox propiedades de scripts de nodos	143, 180	Modelos lineales generalizados de Oracle propiedades de scripts de nodos	191
Modelos de bayesiano ingenuo de Netezza propiedades de scripts de nodos	203, 213	Modelos de regresión de IBM ISW propiedades de scripts de nodos	197, 202	Modelos Oracle AI propiedades de scripts de nodos	191
Modelos de bayesiano ingenuo de Oracle propiedades de scripts de nodos	191, 196	modelos de regresión lineal propiedades de scripts de nodos	169, 185	Modelos para LMD de Oracle propiedades de scripts de nodos	191, 196
Modelos de clasificador automático propiedades de scripts de nodos	178	Modelos de regresión lineal de Netezza propiedades de scripts de nodos	203, 213	modelos PCA propiedades de scripts de nodos	147, 181
modelos de detección de anomalías propiedades de scripts de nodos	129, 177	modelos de regresión logística propiedades de scripts de nodos	161, 183	Modelos PCA de Netezza propiedades de scripts de nodos	203, 213
modelos de IBM DB2 propiedades de scripts de nodos	197	Modelos de regresión logística de IBM ISW propiedades de scripts de nodos	197, 202	modelos PCA/Factorial propiedades de scripts de nodos	147, 181
modelos de IBM SPSS Statistics propiedades de scripts de nodos	240	modelos de respuesta de autoaprendizaje propiedades de scripts de nodos	171, 185	modelos QUEST propiedades de scripts de nodos	167, 184
modelos de K-Means propiedades de scripts de nodos	157, 182	Modelos de secuencia de IBM ISW propiedades de scripts de nodos	197, 202	modelos SLRM propiedades de scripts de nodos	171, 185
modelos de K-medias de Netezza propiedades de scripts de nodos	203, 213	modelos de secuencias propiedades de scripts de nodos	170, 186	modelos SVM propiedades de scripts de nodos	172
Modelos de K-medias de Oracle propiedades de scripts de nodos	191, 196	modelos de selección de características propiedades de scripts de nodos	149, 181	modificar rutas	29, 31
modelos de la máquina de vectores de soporte propiedades de scripts de nodos	172, 186	modelos de series temporales propiedades de scripts de nodos	173, 186	<b>N</b>	
modelos de listas de decisiones propiedades de scripts de nodos	145, 181	Modelos de series temporales de IBM ISW propiedades de scripts de nodos	197	Nodo Adición de RFM propiedades	85
Modelos de máquinas de vectores de soporte de Oracle propiedades de scripts de nodos	191, 196	Modelos de series temporales de Netezza propiedades de scripts de nodos	203	nodo aggregate propiedades	81
		modelos del vecino más próximo propiedades de scripts de nodos	158	Nodo Agrupación en clústeres automática propiedades de scripts de nodos	134
		modelos discriminantes propiedades de scripts de nodos	146, 181	Nodo Análisis de RFM propiedades	105
		modelos generados nombres de scripts	245, 247	Nodo analysis propiedades	215
				nodo anonymize propiedades	93
				nodo append propiedades	81
				nodo Archivo var. propiedades	77
				Nodo Auditoría de datos propiedades	216

- nodo balance
    - propiedades 82
  - nodo Base de datos
    - propiedades 66
  - nodo binning
    - propiedades 96
  - nodo Cajas-Espacio-Tiempo
    - propiedades 82
  - Nodo Clasificador automático
    - propiedades de scripts de nodos 132
  - nodo Colección
    - propiedades 118
  - nodo de datos de usuario
    - propiedades 76
  - Nodo de exportación de IBM SPSS Statistics Collection
    - propiedades 240
  - nodo de exportación de Recopilación de datos de IBM SPSS
    - propiedades 234
  - Nodo de exportación Excel
    - propiedades 234
  - nodo de exportación XML
    - propiedades 236
  - Nodo de origen de Excel
    - propiedades 70
  - Nodo de origen de IBM Cognos BI
    - propiedades 65
  - Nodo de origen de IBM SPSS Data Collection
    - propiedades 67
  - nodo de origen de IBM SPSS Statistics
    - propiedades 239
  - Nodo de origen SAS
    - propiedades 73
  - Nodo de origen XML
    - propiedades 80
  - Nodo de salida de IBM SPSS Statistics Collection
    - propiedades 240
  - nodo de simulación de ajuste
    - propiedades 222
  - nodo de simulación de evaluación
    - propiedades 221
  - nodo de transformación
    - propiedades 226
  - Nodo de transformación de IBM SPSS Statistics Collection
    - propiedades 239
  - nodo derive
    - propiedades 99
  - nodo distinct
    - propiedades 84
  - nodo distribution
    - propiedades 118
  - nodo ensemble
    - propiedades 100
  - nodo Enterprise View
    - propiedades 70
  - nodo Estadísticos
    - propiedades 223
  - nodo evaluation
    - propiedades 119
  - nodo Exportar base de datos
    - propiedades 231
  - Nodo Exportar SAS
    - propiedades 236
  - nodo filter
    - propiedades 101
  - nodo fixedfile
    - propiedades 71
  - nodo flatfilenode
    - propiedades 235
  - nodo Fundir
    - propiedades 84
  - nodo G. múltiple
    - propiedades 123
  - nodo Generación de análisis de serie temporal
    - propiedades 89
  - nodo Gráfico de tiempo
    - propiedades 126
  - Nodo graphboard
    - propiedades 120
  - nodo histogram
    - propiedades 122
  - nodo history
    - propiedades 102
  - nodo Informe
    - propiedades 220
  - nodo Intervalos de tiempo
    - propiedades 107
  - nodo Malla
    - propiedades 127
  - nodo Malla direccional
    - propiedades 127
  - Nodo Marcas
    - propiedades 106
  - nodo matrix
    - propiedades 217
  - nodo Medias
    - propiedades 218
  - nodo origen Analytic Server
    - propiedades 64
  - nodo partition
    - propiedades 102
  - nodo plot
    - propiedades 124
  - nodo R Build
    - propiedades de scripts de nodos 137
  - nodo Reclasificar
    - propiedades 103
  - nodo Reestructurar
    - propiedades 104
  - nodo Rellenar
    - propiedades 101
  - nodo Reordenar
    - propiedades 104
  - nodo Reorg. campos
    - propiedades 104
  - nodo Routput
    - propiedades 220
  - nodo Rprocess
    - propiedades 87
  - nodo sample
    - propiedades 87
  - nodo Seleccionar
    - propiedades 88
  - nodo Sim Eval
    - propiedades 221
  - nodo Sim Fit
    - propiedades 222
  - nodo Sim Gen
    - propiedades 73
  - nodo Simulación de generación
    - propiedades 73
  - nodo sort
    - propiedades 89
  - nodo Tabla
    - propiedades 224
  - nodo Tipo
    - propiedades 111
  - nodo Transponer
    - propiedades 111
  - nodo Val. globales
    - propiedades 221
  - nodos
    - desenlazar nodos 29
    - eliminación 31
    - enlazar nodos 29
    - importación 31
    - información 32
    - referencia de nombres 245
    - sustitución 31
  - nodos de exportación
    - propiedades de scripts de nodos 229
  - nodos de gráficos
    - propiedades de los scripts 117
  - nodos de modelado
    - propiedades de scripts de nodos 129
  - nodos de origen
    - propiedades 63
  - nodos de resultados
    - propiedades de los scripts 215
  - nugget
    - propiedades de scripts de nodos 177
  - nuggets de modelo
    - nombres de scripts 245, 247
    - propiedades de scripts de nodos 177
- ## O
- O-clúster de Oracle
    - propiedades de scripts de nodos 191, 196
  - objetos de resultados
    - nombres de scripts 247
  - objetos del modelo
    - nombres de scripts 245, 247
  - operaciones 14
  - orden de ejecución
    - modificación con scripts 47
  - orden de ejecución de rutas
    - modificación con scripts 47
  - orientado a objetos 21
- ## P
- palabra clave generada 48
  - parámetros 3, 57, 59
    - scripts 14
    - Supernodos 243
  - parámetros de intervalo 3, 57, 58
  - paso de argumentos 18
  - preparación automática de datos
    - propiedades 93
  - propiedades
    - nodos de modelado de bases de datos 187
    - ruta 59

propiedades (*continuación*)  
 scripts 57, 58, 129, 177, 229  
 scripts comunes 58  
 Supernodos 243  
 propiedades de applyr 185  
 Propiedades de buildr 137  
 propiedades de flatfilenode 235  
 Propiedades de nodo de applyc50 179  
 Propiedades de nodo de ejemplo 87  
 Propiedades de nodo de transformación 226  
 Propiedades de nodo lineal 160  
 Propiedades de nodo sas 73  
 Propiedades de nodo twostep 175  
 Propiedades de nodo xmlimport 80  
 propiedades de scripts de nodos 187  
 nodos de exportación 229  
 nodos de modelado 129  
 nuggets de modelo 177  
 Propiedades del nodo aggregate 81  
 Propiedades del nodo analysis 215  
 propiedades del nodo anomalydetection 129  
 propiedades del nodo anonymize 93  
 Propiedades del nodo append 81  
 propiedades del nodo applyanomalydetection 177  
 Propiedades del nodo applyapriori 177  
 propiedades del nodo applyautoclassifier 178  
 propiedades del nodo applyautocluster 178  
 propiedades del nodo applyautonumeric 178  
 Propiedades del nodo applybayesnet 179  
 propiedades del nodo applycarma 179  
 Propiedades del nodo applycart 179  
 Propiedades del nodo applychaid 180  
 Propiedades del nodo applycoxreg 180  
 propiedades del nodo applydb2imcluster 202  
 propiedades del nodo applydb2imlog 202  
 propiedades del nodo applydb2imnb 202  
 propiedades del nodo applydb2imreg 202  
 Propiedades del nodo applydb2imtree 202  
 propiedades del nodo applydecisionlist 181  
 propiedades del nodo applydiscriminant 181  
 propiedades del nodo applyfactor 181  
 propiedades del nodo applyfeatureselection 181  
 propiedades del nodo applygeneralizedlinear 182  
 Propiedades del nodo applyglm 182  
 propiedades del nodo applykmeans 182  
 Propiedades del nodo applyknn 183  
 propiedades del nodo applykohonen 183  
 propiedades del nodo applylinear 183  
 propiedades del nodo applyml 189  
 propiedades del nodo applymneuralnetwork 189  
 propiedades del nodo applymlregression 189  
 propiedades del nodo applymlsequencecluster 189  
 propiedades del nodo applymltimeseries 189  
 Propiedades del nodo applymltree 189  
 propiedades del nodo applynetezabayes 213  
 propiedades del nodo applynetezadectree 213  
 propiedades del nodo applynetezadivcluster 213  
 propiedades del nodo applynetezakmeans 213  
 propiedades del nodo applynetezaknn 213  
 propiedades del nodo applynetezalineregression 213  
 propiedades del nodo applynetezanaivebayes 213  
 propiedades del nodo applynetezapca 213  
 propiedades del nodo applynetezaregtree 213  
 propiedades del nodo applyneuralnet 183  
 Propiedades del nodo applyneuralnetwork 184  
 propiedades del nodo applyoraabn 196  
 Propiedades del nodo applyoradecisiontree 196  
 propiedades del nodo applyorakmeans 196  
 propiedades del nodo applyoranb 196  
 Propiedades del nodo applyoranmf 196  
 propiedades del nodo applyoracluster 196  
 propiedades del nodo applyorasvm 196  
 Propiedades del nodo applyquest 184  
 propiedades del nodo applyregression 185  
 propiedades del nodo applyselflearning 185  
 Propiedades del nodo applysequence 186  
 propiedades del nodo applysvm 186  
 Propiedades del nodo applytimeseries 186  
 propiedades del nodo applytwostep 186  
 Propiedades del nodo apriori 131  
 propiedades del nodo asexport 229  
 propiedades del nodo asimport 64  
 propiedades del nodo autoclassifier 132  
 Propiedades del nodo autocluster 134  
 propiedades del nodo autodataprep 93  
 Propiedades del nodo autonumeric 135  
 Propiedades del nodo balance 82  
 Propiedades del nodo bayesnet 136  
 Propiedades del nodo binning 96  
 Propiedades del nodo c50 138  
 Propiedades del nodo carma 139  
 Propiedades del nodo cart 140  
 Propiedades del nodo chaid 141  
 Propiedades del nodo cognosimport 65  
 Propiedades del nodo collection 118  
 Propiedades del nodo coxreg 143  
 Propiedades del nodo dataaudit 216  
 Propiedades del nodo database 66  
 Propiedades del nodo databaseexport 231  
 Propiedades del nodo datacollectionexport 234  
 Propiedades del nodo datacollectionimport 67  
 Propiedades del nodo db2imassoc 197  
 Propiedades del nodo db2imcluster 197  
 Propiedades del nodo db2imlog 197  
 Propiedades del nodo db2imnb 197  
 Propiedades del nodo db2imreg 197  
 Propiedades del nodo db2imsequence 197  
 Propiedades del nodo db2imtimeseries 197  
 Propiedades del nodo db2imtree 197  
 propiedades del nodo de decisionlist 145  
 Propiedades del nodo derivar 99  
 propiedades del nodo derive\_stb 82  
 propiedades del nodo directedweb 127  
 Propiedades del nodo discriminant 146  
 Propiedades del nodo distinct 84  
 Propiedades del nodo distribution 118  
 Propiedades del nodo ensemble 100  
 Propiedades del nodo evaluation 119  
 Propiedades del nodo evimport 70  
 Propiedades del nodo excelexport 234  
 Propiedades del nodo excelimport 70  
 Propiedades del nodo factor 147  
 propiedades del nodo featureselection 149  
 Propiedades del nodo filler 101  
 Propiedades del nodo filter 101  
 Propiedades del nodo fixedfile 71  
 propiedades del nodo generación de análisis ts 89  
 Propiedades del nodo genlin 150  
 Propiedades del nodo glm 153  
 Propiedades del nodo graphboard 120  
 Propiedades del nodo histogram 122  
 Propiedades del nodo history 102  
 propiedades del nodo kmeans 157  
 Propiedades del nodo knn 158  
 propiedades del nodo kohonen 159  
 Propiedades del nodo logreg 161  
 Propiedades del nodo matrix 217  
 Propiedades del nodo means 218  
 Propiedades del nodo merge 84  
 Propiedades del nodo msassoc 187  
 propiedades del nodo msbayes 187  
 propiedades del nodo mscluster 187  
 propiedades del nodo mslogistic 187  
 propiedades del nodo msneuralnetwork 187  
 propiedades del nodo msregression 187  
 Propiedades del nodo mssequencecluster 187  
 propiedades del nodo mstimeseries 187  
 propiedades del nodo mstree 187  
 Propiedades del nodo netzplot 123  
 Propiedades del nodo netezabayes 203



- Propiedades del nodo netezadectree 203
- Propiedades del nodo netezadivcluster 203
- Propiedades del nodo netezzaglm 203
- Propiedades del nodo netezzakmeans 203
- Propiedades del nodo netezzakn 203
- Propiedades del nodo netezzalineression 203
- Propiedades del nodo netezanaivebayes 203
- Propiedades del nodo netezzapca 203
- Propiedades del nodo netezzaregtree 203
- Propiedades del nodo netezzatimeseries 203
- Propiedades del nodo neuralnet 164
- Propiedades del nodo neuralnetwork 166
- propiedades del nodo numericpredictor 135
- Propiedades del nodo oraabn 191
- Propiedades del nodo oraai 191
- Propiedades del nodo oraapriori 191
- Propiedades del nodo oradecisiontree 191
- Propiedades del nodo oraglm 191
- Propiedades del nodo orakmeans 191
- propiedades del nodo oramd 191
- Propiedades del nodo oranb 191
- Propiedades del nodo oranmf 191
- Propiedades del nodo oraocluster 191
- Propiedades del nodo orasvm 191
- propiedades del nodo outputfile 235
- Propiedades del nodo outputpartition 102
- Propiedades del nodo plot 124
- Propiedades del nodo quest 167
- Propiedades del nodo reclassify 103
- Propiedades del nodo regression 169
- Propiedades del nodo reorder 104
- Propiedades del nodo report 220
- Propiedades del nodo restructure 104
- Propiedades del nodo rfmaggragate 85
- Propiedades del nodo rfmanalysis 105
- Propiedades del nodo Routput 220
- propiedades del nodo Rprocess 87
- Propiedades del nodo sasexport 236
- Propiedades del nodo select 88
- Propiedades del nodo sequence 170
- Propiedades del nodo setglobals 221
- Propiedades del nodo settoflag 106
- Propiedades del nodo simeval 221
- Propiedades del nodo simfit 222
- Propiedades del nodo simgen 73
- propiedades del nodo slrm 171
- Propiedades del nodo sort 89
- Propiedades del nodo statistics 223
- Propiedades del nodo statisticsexport 240
- propiedades del nodo statisticsimport 239
- propiedades del nodo statisticsmodel 240
- Propiedades del nodo statisticsoutput 240

- Propiedades del nodo statisticstransform 239
- Propiedades del nodo svm 172
- Propiedades del nodo table 224
- Propiedades del nodo timeintervals 107
- Propiedades del nodo timeplot 126
- Propiedades del nodo timeseries 173
- Propiedades del nodo transpose 111
- Propiedades del nodo type 111
- Propiedades del nodo userinput 76
- Propiedades del nodo variablefile 77
- propiedades del nodo web 127
- Propiedades del nodo xmllexport 236
- Python 13
  - scripts 14

## R

- Red neuronal de MS
  - propiedades de scripts de nodos 187, 189
- redes neuronales
  - propiedades de scripts de nodos 166, 184
- referencia a nodos 27
  - buscar nodos 27
  - configuración de propiedades 28
- Regresión lineal de MS
  - propiedades de scripts de nodos 187, 189
- Regresión logística de MS
  - propiedades de scripts de nodos 187, 189
- Repositorio de IBM SPSS Collaboration and Deployment Services
  - argumentos de la línea de comandos 55
- rutas
  - bucle 5
  - comando de conjunto múltiple 57
  - ejecución 25
  - ejecución condicional 8
  - modificación 29
  - propiedades 59
  - scripts 1, 25

## S

- script
  - abreviaturas utilizadas 57
  - bucle visual 5
  - bucles visuales 4
  - clave de iteración 6
  - comprobación de errores 48
  - conceptos básicos 1, 13
  - ejecución condicional 4, 8
  - ejecución de nodo de modelado 47
  - nodos de resultados 215
  - orden de ejecución de rutas 47
  - propiedades comunes 58
  - scripts de herencia 250, 251, 254, 257
  - scripts Python 250, 251, 254, 257
  - selección de campos 8
  - sintaxis 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
  - sustitución de modelo 47

- script (*continuación*)
  - variable de iteración 7
- scripts
  - almacenamiento 2
  - bucle 4, 5
  - clave de iteración 6
  - compatibilidad con versiones anteriores 48
  - contexto 26
  - desde la línea de comandos 48
  - Diagramas 25
  - ejecución 9
  - ejecución condicional 4, 8
  - en Supernodos 3
  - importación desde archivos de texto 2
  - interfaz de usuario 2, 3, 10
  - interrupción 9
  - nodos de gráficos 117
  - rutas 1, 25
  - rutas de archivos 48
  - rutas de supernodo 25
  - Rutas de supernodo 25
  - scripts autónomos 1, 25
  - Scripts de Supernodo 1, 25
  - selección de campos 8
  - variable de iteración 7
  - scripts autónomos 1, 3, 25
  - seguridad
    - contraseñas codificadas 47, 54
  - sentencias 17
  - Serie temporales de MS
    - propiedades de scripts de nodos 189
  - servidor
    - argumentos de la línea de comandos 54
  - sistema
    - argumentos de la línea de comandos 52
  - supernodo 57
  - Supernodo
    - ruta 25
  - Supernodos
    - configuración de propiedades 243
    - parámetros 243
    - propiedades 243
    - rutas 25
    - scripts 1, 3, 25, 243

## V

- variable de iteración
  - bucle en scripts 7
- variables
  - scripts 14
- variables ocultas 23







Impreso en España