IBM SPSS Modeler 18.3 In-Database Mining Guide

Note

Before you use this information and the product it supports, read the information in “Notices” on page 93.

Product Information

This edition applies to version 18, release 3, modification 0 of IBM® SPSS® Modeler and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Preface

IBM SPSS Modeler is the IBM enterprise-strength data mining workbench. SPSS Modeler helps organizations to improve customer and citizen relationships through an in-depth understanding of data. Organizations use the insight gained from SPSS Modeler to retain profitable customers, identify cross-selling opportunities, attract new customers, detect fraud, reduce risk, and improve government service delivery.

SPSS Modeler's visual interface invites users to apply their specific business expertise, which leads to more powerful predictive models and shortens time-to-solution. SPSS Modeler offers many modeling techniques, such as prediction, classification, segmentation, and association detection algorithms. Once models are created, IBM SPSS Modeler Solution Publisher enables their delivery enterprise-wide to decision makers or to a database.

About IBM Business Analytics

IBM Business Analytics software delivers complete, consistent and accurate information that decision-makers trust to improve business performance. A comprehensive portfolio of business intelligence, predictive analytics, financial performance and strategy management, and analytic applications provides clear, immediate and actionable insights into current performance and the ability to predict future outcomes. Combined with rich industry solutions, proven practices and professional services, organizations of every size can drive the highest productivity, confidently automate decisions and deliver better results.

As part of this portfolio, IBM SPSS Predictive Analytics software helps organizations predict future events and proactively act upon that insight to drive better business outcomes. Commercial, government and academic customers worldwide rely on IBM SPSS technology as a competitive advantage in attracting, retaining and growing customers, while reducing fraud and mitigating risk. By incorporating IBM SPSS software into their daily operations, organizations become predictive enterprises - able to direct and automate decisions to meet business goals and achieve measurable competitive advantage. For further information or to reach a representative, visit http://www.ibm.com/spss.

Technical support

Technical support is available to maintenance customers. Customers may contact Technical Support for assistance in using IBM products or for installation help for one of the supported hardware environments. To reach Technical Support, see the IBM web site at http://www.ibm.com/support. Be prepared to identify yourself, your organization, and your support agreement when requesting assistance.

Chapter 1. About IBM SPSS Modeler

IBM SPSS Modeler is a set of data mining tools that enable you to quickly develop predictive models using business expertise and deploy them into business operations to improve decision making. Designed around the industry-standard CRISP-DM model, IBM SPSS Modeler supports the entire data mining process, from data to better business results.

IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics. The methods available on the Modeling palette allow you to derive new information from your data and to develop predictive models. Each method has certain strengths and is best suited for particular types of problems.

SPSS Modeler can be purchased as a standalone product, or used as a client in combination with SPSS Modeler Server. A number of additional options are also available, as summarized in the following sections. For more information, see https://www.ibm.com/analytics/us/en/technology/spss/.

IBM SPSS Modeler Products

The IBM SPSS Modeler family of products and associated software comprises the following.

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console (included with IBM SPSS Deployment Manager)
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- IBM SPSS Modeler Server adapters for IBM SPSS Collaboration and Deployment Services

IBM SPSS Modeler

SPSS Modeler is a functionally complete version of the product that you install and run on your personal computer. You can run SPSS Modeler in local mode as a standalone product, or use it in distributed mode along with IBM SPSS Modeler Server for improved performance on large data sets.

With SPSS Modeler, you can build accurate predictive models quickly and intuitively, without programming. Using the unique visual interface, you can easily visualize the data mining process. With the support of the advanced analytics embedded in the product, you can discover previously hidden patterns and trends in your data. You can model outcomes and understand the factors that influence them, enabling you to take advantage of business opportunities and mitigate risks.

SPSS Modeler is available in two editions: SPSS Modeler Professional and SPSS Modeler Premium. See the topic “ IBM SPSS Modeler Editions” on page 2 for more information.

IBM SPSS Modeler Server

SPSS Modeler uses a client/server architecture to distribute requests for resource-intensive operations to powerful server software, resulting in faster performance on larger data sets.

SPSS Modeler Server is a separately-licensed product that runs continually in distributed analysis mode on a server host in conjunction with one or more IBM SPSS Modeler installations. In this way, SPSS Modeler Server provides superior performance on large data sets because memory-intensive operations can be done on the server without downloading data to the client computer. IBM SPSS Modeler Server also provides support for SQL optimization and in-database modeling capabilities, delivering further benefits in performance and automation.

IBM SPSS Modeler Administration Console

The Modeler Administration Console is a graphical user interface for managing many of the SPSS Modeler Server configuration options, which are also configurable by means of an options file. The console is included in IBM SPSS Deployment Manager, can be used to monitor and configure your SPSS Modeler Server installations, and is available free-of-charge to current SPSS Modeler Server customers. The application can be installed only on Windows computers; however, it can administer a server installed on any supported platform.

IBM SPSS Modeler Batch

While data mining is usually an interactive process, it is also possible to run SPSS Modeler from a command line, without the need for the graphical user interface. For example, you might have long-running or repetitive tasks that you want to perform with no user intervention. SPSS Modeler Batch is a special version of the product that provides support for the complete analytical capabilities of SPSS Modeler without access to the regular user interface. SPSS Modeler Server is required to use SPSS Modeler Batch.

IBM SPSS Modeler Solution Publisher

SPSS Modeler Solution Publisher is a tool that enables you to create a packaged version of an SPSS Modeler stream that can be run by an external runtime engine or embedded in an external application. In this way, you can publish and deploy complete SPSS Modeler streams for use in environments that do not have SPSS Modeler installed. SPSS Modeler Solution Publisher is distributed as part of the IBM SPSS Collaboration and Deployment Services - Scoring service, for which a separate license is required. With this license, you receive SPSS Modeler Solution Publisher Runtime, which enables you to execute the published streams.

For more information about SPSS Modeler Solution Publisher, see the IBM SPSS Collaboration and Deployment Services documentation. The IBM SPSS Collaboration and Deployment Services IBM Documentation contains sections called "IBM SPSS Modeler Solution Publisher" and "IBM SPSS Analytics Toolkit."

IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services

A number of adapters for IBM SPSS Collaboration and Deployment Services are available that enable SPSS Modeler and SPSS Modeler Server to interact with an IBM SPSS Collaboration and Deployment Services repository. In this way, an SPSS Modeler stream deployed to the repository can be shared by multiple users, or accessed from the thin-client application IBM SPSS Modeler Advantage. You install the adapter on the system that hosts the repository.

IBM SPSS Modeler Editions

SPSS Modeler is available in the following editions.

SPSS Modeler Professional

SPSS Modeler Professional provides all the tools you need to work with most types of structured data, such as behaviors and interactions tracked in CRM systems, demographics, purchasing behavior and sales data.

SPSS Modeler Premium

SPSS Modeler Premium is a separately-licensed product that extends SPSS Modeler Professional to work with specialized data and with unstructured text data. SPSS Modeler Premium includes IBM SPSS Modeler Text Analytics:

IBM SPSS Modeler Text Analytics uses advanced linguistic technologies and Natural Language Processing (NLP) to rapidly process a large variety of unstructured text data, extract and organize the key concepts, and group these concepts into categories. Extracted concepts and categories can be combined with existing structured data, such as demographics, and applied to modeling using the full suite of IBM SPSS Modeler data mining tools to yield better and more focused decisions.

IBM SPSS Modeler Subscription

IBM SPSS Modeler Subscription provides all the same predictive analytics capabilities as the traditional IBM SPSS Modeler client. With the Subscription edition, you can download product updates regularly.

Documentation

Documentation is available from the Help menu in SPSS Modeler. This opens the online IBM Documentation, which is always available outside the product.

Complete documentation for each product (including installation instructions) is also available in PDF format, in a separate compressed folder, as part of the product download. Or the latest PDF documents can be downloaded from the web at https://www.ibm.com/support/pages/spss-modeler-1822-documentation.

SPSS Modeler Professional Documentation

The SPSS Modeler Professional documentation suite (excluding installation instructions) is as follows.

- **IBM SPSS Modeler User's Guide.** General introduction to using SPSS Modeler, including how to build data streams, handle missing values, build CLEM expressions, work with projects and reports, and package streams for deployment to IBM SPSS Collaboration and Deployment Services or IBM SPSS Modeler Advantage.
- **IBM SPSS Modeler Source, Process, and Output Nodes.** Descriptions of all the nodes used to read, process, and output data in different formats. Effectively this means all nodes other than modeling nodes.
- **IBM SPSS Modeler Modeling Nodes.** Descriptions of all the nodes used to create data mining models. IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics.
- **IBM SPSS Modeler Applications Guide.** The examples in this guide provide brief, targeted introductions to specific modeling methods and techniques. An online version of this guide is also available from the Help menu. See the topic "Application examples" on page 4 for more information.
- **IBM SPSS Modeler Python Scripting and Automation.** Information on automating the system through Python scripting, including the properties that can be used to manipulate nodes and streams.
- **IBM SPSS Modeler Deployment Guide.** Information on running IBM SPSS Modeler streams as steps in processing jobs under IBM SPSS Deployment Manager.
- **IBM SPSS Modeler CLEF Developer's Guide.** CLEF provides the ability to integrate third-party programs such as data processing routines or modeling algorithms as nodes in IBM SPSS Modeler.
- **IBM SPSS Modeler In-Database Mining Guide.** Information on how to use the power of your database to improve performance and extend the range of analytical capabilities through third-party algorithms.
- **IBM SPSS Modeler Server Administration and Performance Guide.** Information on how to configure and administer IBM SPSS Modeler Server.
- **IBM SPSS Deployment Manager User Guide.** Information on using the administration console user interface included in the Deployment Manager application for monitoring and configuring IBM SPSS Modeler Server.
- **IBM SPSS Modeler CRISP-DM Guide.** Step-by-step guide to using the CRISP-DM methodology for data mining with SPSS Modeler.

- **IBM SPSS Modeler Batch User's Guide.** Complete guide to using IBM SPSS Modeler in batch mode, including details of batch mode execution and command-line arguments. This guide is available in PDF format only.

SPSS Modeler Premium Documentation

The SPSS Modeler Premium documentation suite (excluding installation instructions) is as follows.

- **SPSS Modeler Text Analytics User's Guide.** Information on using text analytics with SPSS Modeler, covering the text mining nodes, interactive workbench, templates, and other resources.

Application examples

While the data mining tools in SPSS Modeler can help solve a wide variety of business and organizational problems, the application examples provide brief, targeted introductions to specific modeling methods and techniques. The data sets used here are much smaller than the enormous data stores managed by some data miners, but the concepts and methods that are involved are scalable to real-world applications.

To access the examples, click **Application Examples** on the Help menu in SPSS Modeler.

The data files and sample streams are installed in the `Demos` folder under the product installation directory. For more information, see "Demos Folder" on page 4.

Database modeling examples. See the examples in the *IBM SPSS Modeler In-Database Mining Guide*.

Scripting examples. See the examples in the *IBM SPSS Modeler Scripting and Automation Guide*.

Demos Folder

The data files and sample streams that are used with the application examples are installed in the `Demos` folder under the product installation directory (for example: `C:\Program Files\IBM\SPSS\Modeler\<version>\Demos`). This folder can also be accessed from the IBM SPSS Modeler program group on the Windows Start menu, or by clicking `Demos` on the list of recent directories in the **File** > **Open Stream** dialog box.

License tracking

When you use SPSS Modeler, license usage is tracked and logged at regular intervals. The license metrics that are logged are *AUTHORIZED_USER* and *CONCURRENT_USER*, and the type of metric that is logged depends on the type of license that you have for SPSS Modeler.

The log files that are produced can be processed by the IBM License Metric Tool, from which you can generate license usage reports.

The license log files are created in the same directory where SPSS Modeler Client log files are recorded (by default, `%ALLUSERSPROFILE%/IBM/SPSS/Modeler/<version>/log`).

Chapter 2. In-Database Mining

Database Modeling Overview

IBM SPSS Modeler Server supports integration with data mining and modeling tools that are available from database vendors, including IBM Netezza, Oracle Data Miner, and Microsoft Analysis Services. You can build, score, and store models inside the database—all from within the IBM SPSS Modeler application. This allows you to combine the analytical capabilities and ease-of-use of IBM SPSS Modeler with the power and performance of a database, while taking advantage of database-native algorithms provided by these vendors. Models are built inside the database, which can then be browsed and scored through the IBM SPSS Modeler interface in the normal manner and can be deployed using IBM SPSS Modeler Solution Publisher if needed. Supported algorithms are on the Database Modeling palette in IBM SPSS Modeler.

Using IBM SPSS Modeler to access database-native algorithms offers several advantages:

- In-database algorithms are often closely integrated with the database server and may offer improved performance.
- Models built and stored "in database" may be more easily deployed to and shared with any application that can access the database.

SQL generation. In-database modeling is distinct from SQL generation, otherwise known as "SQL pushback". This feature allows you to generate SQL statements for native IBM SPSS Modeler operations that can be "pushed back" to (that is, executed in) the database in order to improve performance. For example, the Merge, Aggregate, and Select nodes all generate SQL code that can be pushed back to the database in this manner. Using SQL generation in combination with database modeling may result in streams that can be run from start to finish in the database, resulting in significant performance gains over streams run in IBM SPSS Modeler.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > **About** > **Additional Details**

If connectivity is enabled, you see the option **Server Enablement** in the License Status tab.

For information on supported algorithms, see the subsequent sections on specific vendors.

What you need

To perform database modeling, you need the following setup:

- An ODBC connection to an appropriate database, with required analytical components installed (Microsoft Analysis Services or Oracle Data Miner).
- In IBM SPSS Modeler, database modeling must be enabled in the Helper Applications dialog box (**Tools > Helper Applications**).
- The **Generate SQL** and **SQL Optimization** settings should be enabled in the User Options dialog box in IBM SPSS Modeler as well as on IBM SPSS Modeler Server (if used). Note that SQL optimization is not strictly required for database modeling to work but is highly recommended for performance reasons.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > **About** > **Additional Details**

If connectivity is enabled, you see the option **Server Enablement** in the License Status tab.

For detailed information, see the subsequent sections on specific vendors.

Model Building

The process of building and scoring models by using database algorithms is similar to other types of data mining in IBM SPSS Modeler. The general process of working with nodes and modeling "nuggets" is similar to any other stream when working in IBM SPSS Modeler. The only difference is that the actual processing and model building is pushed back to the database.

A Database modelling stream is conceptually identical to other data streams in IBM SPSS Modeler; however, this stream performs all operations in a database, including, for example, model-building using the Microsoft Decision Tree node. When you run the stream, IBM SPSS Modeler instructs the database to build and store the resulting model, and details are downloaded to IBM SPSS Modeler. In-database execution is indicated by the use of purple-shaded nodes in the stream.

Data Preparation

Whether or not database-native algorithms are used, data preparations should be pushed back to the database whenever possible in order to improve performance.

- If the original data is stored in the database, the goal is to keep it there by making sure that all required upstream operations can be converted to SQL. This will prevent data from being downloaded to IBM SPSS Modeler—avoiding a bottleneck that might nullify any gains—and allow the entire stream to be run in the database.
- If the original data are *not* stored in the database, database modeling can still be used. In this case, the data preparation is conducted in IBM SPSS Modeler and the prepared dataset is automatically uploaded to the database for model building.

Model scoring

Models generated from IBM SPSS Modeler by using in-database mining are different from regular IBM SPSS Modeler models. Although they appear in the Model manager as generated model “nuggets,” they are actually remote models held on the remote data mining or database server. What you see in IBM SPSS Modeler are simply references to these remote models. In other words, the IBM SPSS Modeler model you see is a "hollow" model that contains information such as the database server hostname, database name, and the model name. This is an important distinction to understand as you browse and score models created using database-native algorithms.

Once you have created a model, you can add it to the stream for scoring like any other generated model in IBM SPSS Modeler. All scoring is done within the database, even if upstream operations are not. (Upstream operations may still be pushed back to the database if possible to improve performance, but this is not a requirement for scoring to take place.) You can also browse the generated model in most cases using the standard browser provided by the database vendor.

For both browsing and scoring, a live a connection to the server running Oracle Data Miner or Microsoft Analysis Services is required.

Viewing results and specifying settings

To view results and specify settings for scoring, double-click the model on the stream canvas. Alternatively, you can right-click the model and choose **Browse** or **Edit**. Specific settings depend on the type of model.

Exporting and saving database models

Database models and summaries can be exported from the model browser in the same manner as other models created in IBM SPSS Modeler, using options on the File menu.

1. From the File menu in the model browser, choose any of the following options:

- **Export Text** exports the model summary to a text file

- **Export HTML** exports the model summary to an HTML file
- **Export PMML** (supported for IBM Db2 IM models only) exports the model as predictive model markup language (PMML), which can be used with other PMML-compatible software.

Note: You can also save a generated model by choosing **Save Node** from the File menu.

Model Consistency

For each generated database model, IBM SPSS Modeler stores a description of the model structure, along with a reference to the model with the same name that is stored within the database. The Server tab of a generated model displays a unique key generated for that model, which matches the actual model in the database.

IBM SPSS Modeler uses this randomly generated key to check that the model is still consistent. This key is stored in the description of a model when it is built. It is a good idea to check that keys match before running a deployment stream.

1. To check the consistency of the model stored in the database by comparing its description with the random key stored by IBM SPSS Modeler, click the **Check** button. If the database model cannot be found or the key does not match, an error is reported.

Viewing and Exporting Generated SQL

The generated SQL code can be previewed prior to execution, which may be useful for debugging purposes.

Chapter 3. Database Modeling with Microsoft Analysis Services

IBM SPSS Modeler and Microsoft Analysis Services

IBM SPSS Modeler supports integration with Microsoft SQL Server Analysis Services. This functionality is implemented as modeling nodes in IBM SPSS Modeler and is available from the Database Modeling palette. If the palette is not visible, you can activate it by enabling MS Analysis Services integration, available on the Microsoft tab, from the Helper Applications dialog box. See the topic "Enabling Integration with Analysis Services" on page 11 for more information.

IBM SPSS Modeler supports integration of the following Analysis Services algorithms:

- Decision Trees
- Clustering
- Association Rules
- Naive Bayes
- Linear Regression
- Neural Network
- Logistic Regression
- Time Series
- Sequence Clustering

The following diagram illustrates the flow of data from client to server where in-database mining is managed by IBM SPSS Modeler Server. Model building is performed using Analysis Services. The resulting model is stored by Analysis Services. A reference to this model is maintained within IBM SPSS Modeler streams. The model is then downloaded from Analysis Services to either Microsoft SQL Server or IBM SPSS Modeler for scoring.

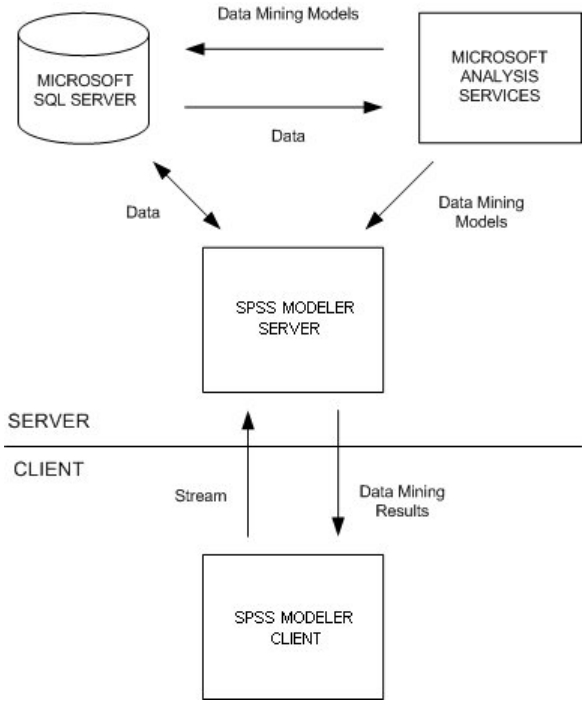

Figure 1. Data flow between IBM SPSS Modeler, Microsoft SQL Server, and Microsoft Analysis Services during model building

Note: The IBM SPSS Modeler Server is not required, though it can be used. IBM SPSS Modeler client is capable of processing in-database mining calculations itself.

Requirements for Integration with Microsoft Analysis Services

The following are prerequisites for conducting in-database modeling using Analysis Services algorithms with IBM SPSS Modeler. You may need to consult with your database administrator to ensure that these conditions are met.

- IBM SPSS Modeler running against an IBM SPSS Modeler Server installation (distributed mode) on Windows. UNIX platforms are not supported in this integration with Analysis Services.

  Important: IBM SPSS Modeler users must configure an ODBC connection using the SQL Native Client driver available from Microsoft at the URL listed below under *Additional IBM SPSS Modeler Server Requirements*. *The driver provided with the IBM SPSS Data Access Pack (and typically recommended for other uses with IBM SPSS Modeler) is not recommended for this purpose.* The driver should be configured to use SQL Server **With Integrated Windows Authentication** enabled, since IBM SPSS Modeler does not support SQL Server authentication. If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.
- SQL Server must be installed, although not necessarily on the same host as IBM SPSS Modeler. IBM SPSS Modeler users must have sufficient permissions to read and write data and drop and create tables and views.

  Note: SQL Server Enterprise Edition is recommended. The Enterprise Edition provides additional flexibility by providing advanced parameters to tune algorithm results. The Standard Edition version provides the same parameters but does not allow users to edit some of the advanced parameters.
- Microsoft SQL Server Analysis Services must be installed on the same host as SQL Server.

Additional IBM SPSS Modeler Server Requirements

To use Analysis Services algorithms with IBM SPSS Modeler Server, the following components must be installed on the IBM SPSS Modeler Server host machine.

Note: If SQL Server is installed on the same host as IBM SPSS Modeler Server, these components will already be available.

- Microsoft SQL Server Analysis Services 10.0 OLE DB Provider (be sure to select the correct variant for your operating system)
- Microsoft SQL Server Native Client (be sure to select the correct variant for your operating system)
- If you are using Microsoft SQL Server 2008 or 2012, you might also require Microsoft Core XML Services (MSXML) 6.0.

To download these components, go to *www.microsoft.com/downloads*, search for **.NET Framework** or (for all other components) **SQL Server Feature Pack**, and select the latest pack for your version of SQL Server.

These may require other packages to be installed first, which should also be available on the Microsoft Downloads web site.

Additional IBM SPSS Modeler Requirements

To use Analysis Services algorithms with IBM SPSS Modeler, the same components must be installed as above, with the addition of the following at the client:

- Microsoft SQL Server Datamining Viewer Controls (be sure to select the correct variant for your operating system) - this also requires:
- Microsoft ADOMD.NET

To download these components, go to *www.microsoft.com/downloads*, search for **SQL Server Feature Pack**, and select the latest pack for your version of SQL Server.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > **About** > **Additional Details**

If connectivity is enabled, you see the option **Server Enablement** in the License Status tab.

Enabling Integration with Analysis Services

To enable IBM SPSS Modeler integration with Analysis Services, you will need to configure SQL Server and Analysis Services, create an ODBC source, enable the integration in the IBM SPSS Modeler Helper Applications dialog box, and enable SQL generation and optimization.

Note: Microsoft SQL Server and Microsoft Analysis Services must be available. See the topic "Requirements for Integration with Microsoft Analysis Services" on page 10 for more information.

Configuring SQL Server

Configure SQL Server to allow scoring to take place within the database.

1. Create the following registry key on the SQL Server host machine:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\MSOLAP
```

2. Add the following DWORD value to this key:

```
AllowInProcess 1
```

3. Restart SQL Server after making this change.

Configuring Analysis Services

Before IBM SPSS Modeler can communicate with Analysis Services, you must first manually configure two settings in the Analysis Server Properties dialog box:

1. Log in to the Analysis Server through the MS SQL Server Management Studio.
2. Access the Properties dialog box by right-clicking the server name and choosing **Properties**.

3. Select the **Show Advanced (All) Properties** check box.
4. Change the following properties:

- Change the value for `DataMining\AllowAdHocOpenRowsetQueries` to `True` (the default value is `False`).
- Change the value for `DataMining\AllowProvidersInOpenRowset` to `[all]` (there is no default value).

Creating an ODBC DSN for SQL Server

To read or write to a database, you must have an ODBC data source installed and configured for the relevant database, with read or write permissions as needed. The Microsoft SQL Native Client ODBC driver is required and automatically installed with SQL Server. *The driver provided with the IBM SPSS Data Access Pack (and typically recommended for other uses with IBM SPSS Modeler) is not recommended for this purpose.* If IBM SPSS Modeler and SQL Server reside on different hosts, you can download the Microsoft SQL Native Client ODBC driver. See the topic "Requirements for Integration with Microsoft Analysis Services" on page 10 for more information.

If you have questions about creating or setting permissions for ODBC data sources, contact your database administrator.

1. Using the Microsoft SQL Native Client ODBC driver, create an ODBC DSN that points to the SQL Server database used in the data mining process. The remaining default driver settings should be used.
2. For this DSN, ensure that **With Integrated Windows Authentication** is selected.

- If IBM SPSS Modeler and IBM SPSS Modeler Server are running on different hosts, create the same ODBC DSN on each of the hosts. Ensure that the same DSN name is used on each host.

Enabling the Analysis Services Integration in IBM SPSS Modeler

To enable IBM SPSS Modeler to use Analysis Services, you must first provide server specifications in the Helper Applications dialog box.

1. From the IBM SPSS Modeler menus choose:

   Tools > **Options** > **Helper Applications**
2. Click the **Microsoft** tab.

- **Enable Microsoft Analysis Services Integration.** Enables the Database Modeling palette (if not already displayed) at the bottom of the IBM SPSS Modeler window and adds the nodes for Analysis Services algorithms.
- **Analysis Server Host.** Specify the name of the machine on which Analysis Services is running.
- **Analysis Server Database.** Select the desired database by clicking the ellipsis (...) button to open a subdialog box in which you can choose from available databases. The list is populated with databases available to the specified Analysis server. Since Microsoft Analysis Services stores data mining models within named databases, you should select the appropriate database in which Microsoft models built by IBM SPSS Modeler are stored.
- **SQL Server Connection.** Specify the DSN information used by the SQL Server database to store the data that are passed into the Analysis server. Choose the ODBC data source that will be used to provide the data for building Analysis Services data mining models. If you are building Analysis Services models from data supplied in flat files or ODBC data sources, the data will be automatically uploaded to a temporary table created in the SQL Server database to which this ODBC data source points.
- **Warn when about to overwrite a data mining model.** Select to ensure that models stored in the database are not overwritten by IBM SPSS Modeler without warning.

Note: Settings made in the Helper Applications dialog box can be overridden inside the various Analysis Services nodes.

Enabling SQL Generation and Optimization

1. From the IBM SPSS Modeler menus choose:

Tools > **Stream Properties** > **Options**

2. Click the **Optimization** option in the navigation pane.
3. Confirm that the **Generate SQL** option is enabled. This setting is required for database modeling to function.
4. Select **Optimize SQL Generation** and **Optimize other execution** (not strictly required but strongly recommended for optimized performance).

Building Models with Analysis Services

Analysis Services model building requires that the training dataset be located in a table or view within the SQL Server database. If the data is not located in SQL Server or need to be processed in IBM SPSS Modeler as part of data preparation that cannot be performed in SQL Server, the data is automatically uploaded to a temporary table in SQL Server before model building.

Managing Analysis Services Models

Building an Analysis Services model via IBM SPSS Modeler creates a model in IBM SPSS Modeler and creates or replaces a model in the SQL Server database. The IBM SPSS Modeler model references the content of a database model stored on a database server. IBM SPSS Modeler can perform consistency checking by storing an identical generated model key string in both the IBM SPSS Modeler model and the SQL Server model.

The **MS Decision Tree** modeling node is used in predictive modeling of both categorical and continuous attributes. For categorical attributes, the node makes predictions based on the relationships between input columns in a dataset. For example, in a scenario to predict which customers are likely to purchase a bicycle, if nine out of ten younger customers buy a bicycle but only two out of ten older customers do so, the node infers that age is a good predictor of bicycle purchase. The decision tree makes predictions based on this tendency toward a particular outcome. For continuous attributes, the algorithm uses linear regression to determine where a decision tree splits. If more than one column is set to predictable, or if the input data contains a nested table that is set to predictable, the node builds a separate decision tree for each predictable column.

The **MS Clustering** modeling node uses iterative techniques to group cases in a dataset into clusters that contain similar characteristics. These groupings are useful for exploring data, identifying anomalies in the data, and creating predictions. Clustering models identify relationships in a dataset that you might not logically derive through casual observation. For example, you can logically discern that people who commute to their jobs by bicycle do not typically live a long distance from where they work. The algorithm, however, can find other characteristics about bicycle commuters that are not as obvious. The clustering node differs from other data mining nodes in that no target field is specified. The clustering node trains the model strictly from the relationships that exist in the data and from the clusters that the node identifies.

The **MS Association Rules** modeling node is useful for recommendation engines. A recommendation engine recommends products to customers based on items they have already purchased or in which they have indicated an interest. Association models are built on datasets that contain identifiers both for individual cases and for the items that the cases contain. A group of items in a case is called an **itemset**. An association model is made up of a series of itemsets and the rules that describe how those items are grouped together within the cases. The rules that the algorithm identifies can be used to predict a customer's likely future purchases, based on the items that already exist in the customer's shopping cart.

The **MS Naive Bayes** modeling node calculates the conditional probability between target and predictor fields and assumes that the columns are independent. The model is termed naïve because it treats all proposed prediction variables as being independent of one another. This method is computationally less intense than other Analysis Services algorithms and therefore useful for quickly discovering relationships during the preliminary stages of modeling. You can use this node to do initial explorations of data and then apply the results to create additional models with other nodes that may take longer to compute but give more accurate results.

The **MS Linear Regression** modeling node is a variation of the Decision Trees node, where the `MINIMUM_LEAF_CASES` parameter is set to be greater than or equal to the total number of cases in the dataset that the node uses to train the mining model. With the parameter set in this way, the node will never create a split and therefore performs a linear regression.

The **MS Neural Network** modeling node is similar to the MS Decision Tree node in that the MS Neural Network node calculates probabilities for each possible state of the input attribute when given each state of the predictable attribute. You can later use these probabilities to predict an outcome of the predicted attribute, based on the input attributes.

The **MS Logistic Regression** modeling node is a variation of the MS Neural Network node, where the `HIDDEN_NODE_RATIO` parameter is set to 0. This setting creates a neural network model that does not contain a hidden layer and therefore is equivalent to logistic regression.

The **MS Time Series** modeling node provides regression algorithms that are optimized for the forecasting of continuous values, such as product sales, over time. Whereas other Microsoft algorithms, such as decision trees, require additional columns of new information as input to predict a trend, a time series model does not. A time series model can predict trends based only on the original dataset that is used to create the model. You can also add new data to the model when you make a prediction and automatically incorporate the new data in the trend analysis. See the topic "MS Time Series Node" on page 16 for more information.

The **MS Sequence Clustering** modeling node identifies ordered sequences in data, and combines the results of this analysis with clustering techniques to generate clusters based on the sequences and other attributes. See the topic "MS Sequence Clustering Node" on page 18 for more information.

You can access each node from the Database Modeling palette at the bottom of the IBM SPSS Modeler window.

Settings Common to All Algorithm Nodes

The following settings are common to all Analysis Services algorithms.

Server Options

On the Server tab, you can configure the Analysis server host, database, and the SQL Server data source. Options specified here overwrite those specified on the Microsoft tab in the Helper Applications dialog box. See the topic "Enabling Integration with Analysis Services" on page 11 for more information.

Note: A variation of this tab is also available when scoring Analysis Services models. See the topic "Analysis Services Model Nugget Server Tab" on page 19 for more information.

Model Options

In order to build the most basic model, you need to specify options on the Model tab before proceeding. Scoring method and other advanced options are available on the Expert tab.

The following basic modeling options are available:

Model name. Specifies the name assigned to the model that is created when the node is executed.

- **Auto.** Generates the model name automatically based on the target or ID field names or the name of the model type in cases where no target is specified (such as clustering models).
- **Custom.** Allows you to specify a custom name for the model created.

Use partitioned data. Splits the data into separate subsets or samples for training, testing, and validation based on the current partition field. Using one sample to create the model and a separate sample to test it may provide an indication of how well the model will generalize to larger datasets that are similar to the current data. If no partition field is specified in the stream, this option is ignored.

With Drillthrough. If shown, this option enables you to query the model to learn details about the cases included in the model.

Unique field. From the drop-down list, select a field that uniquely identifies each case. Typically, this is an ID field, such as **CustomerID**.

MS Decision Tree Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

MS Clustering Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

MS Naive Bayes Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

MS Linear Regression Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

MS Neural Network Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

MS Logistic Regression Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

MS Association Rules Node

The MS Association Rules modeling node is useful for recommendation engines. A recommendation engine recommends products to customers based on items they have already purchased or in which they have indicated an interest. Association models are built on datasets that contain identifiers both for individual cases and for the items that the cases contain. A group of items in a case is called an **itemset**.

An association model is made up of a series of itemsets and the rules that describe how those items are grouped together within the cases. The rules that the algorithm identifies can be used to predict a customer's likely future purchases, based on the items that already exist in the customer's shopping cart.

For tabular format data, the algorithm creates scores that represent probability ($MP-*field*) for each generated recommendation ($M-*field*). For transactional format data, scores are created for support ($MS-*field*), probability ($MP-*field*) and adjusted probability ($MAP-*field*) for each generated recommendation ($M-*field*).

Requirements

The requirements for a transactional association model are as follows:

- **Unique field.** An association rules model requires a key that uniquely identifies records.
- **ID field.** When building an MS Association Rules model with transactional format data, an ID field that identifies each transaction is required. ID fields can be set to the same as the unique field.
- **At least one input field.** The Association Rules algorithm requires at least one input field.
- **Target field.** When building an MS Association model with transactional data, the target field must be the transaction field, for example products that a user bought.

MS Association Rules Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

MS Time Series Node

The MS Time Series modeling node supports two types of predictions:

- future
- historical

Future predictions estimate target field values for a specified number of time periods beyond the end of your historical data, and are always performed. **Historical predictions** are estimated target field values for a specified number of time periods for which you have the actual values in your historical data. You can use historical predictions to asses the quality of the model, by comparing the actual historical values with the predicted values. The value of the start point for the predictions determines whether historical predictions are performed.

Unlike the IBM SPSS Modeler Time Series node, the MS Time Series node does not need a preceding Time Intervals node. A further difference is that by default, scores are produced only for the predicted rows, not for all the historical rows in the time series data.

Requirements

The requirements for an MS Time Series model are as follows:

- **Single key time field.** Each model must contain one numeric or date field that is used as the case series, defining the time slices that the model will use. The data type for the key time field can be either a datetime data type or a numeric data type. However, the field must contain continuous values, and the values must be unique for each series.
- **Single target field.** You can specify only one target field in each model. The data type of the target field must have continuous values. For example, you can predict how numeric attributes, such as income, sales, or temperature, change over time. However, you cannot use a field that contains categorical values, such as purchasing status or level of education, as the target field.
- **At least one input field.** The MS Time Series algorithm requires at least one input field. The data type of the input field must have continuous values. Non-continuous input fields are ignored when building the model.
- **Dataset must be sorted.** The input dataset must be sorted (on the key time field), otherwise model building will be interrupted with an error.

MS Time Series Model Options

Model name. Specifies the name assigned to the model that is created when the node is executed.

- **Auto.** Generates the model name automatically based on the target or ID field names or the name of the model type in cases where no target is specified (such as clustering models).
- **Custom.** Allows you to specify a custom name for the model created.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

With Drillthrough. If shown, this option enables you to query the model to learn details about the cases included in the model.

Unique field. From the drop-down list, select the key time field, which is used to build the time series model.

MS Time Series Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

If you are making historical predictions, the number of historical steps that can be included in the scoring result is decided by the value of (HISTORIC_MODEL_COUNT * HISTORIC_MODEL_GAP). By default, this limitation is 10, meaning that only 10 historical predictions will be made. In this case, for example, an error occurs if you enter a value of less than -10 for **Historical prediction** on the Settings tab of the model nugget (see “MS Time Series Model Nugget Settings Tab” on page 20). If you want to see more historical predictions, you can increase the value of HISTORIC_MODEL_COUNT or HISTORIC_MODEL_GAP, but this will increase the build time for the model.

MS Time Series Settings Options

Begin Estimation. Specify the time period where you want predictions to start.

- **Start From: New Prediction.** The time period at which you want future predictions to start, expressed as an offset from the last time period of your historical data. For example, if your historical data ended at 12/99 and you wanted to begin predictions at 01/00, you would use a value of 1; however, if you wanted predictions to start at 03/00, you would use a value of 3.

- **Start From: Historical Prediction.** The time period at which you want historical predictions to start, expressed as a negative offset from the last time period of your historical data. For example, if your historical data ended at 12/99 and you wanted to make historical predictions for the last five time periods of your data, you would use a value of -5.

End Estimation. Specify the time period where you want predictions to stop.

- **End step of the prediction.** The time period at which you want predictions to stop, expressed as an offset from the last time period of your historical data. For example, if your historical data end at 12/99 and you want predictions to stop at 6/00, you would use a value of 6 here. For future predictions, the value must always be greater than or equal to the **Start From** value.

MS Sequence Clustering Node

The MS Sequence Clustering node uses a sequence analysis algorithm that explores data containing events that can be linked by following paths, or *sequences*. Some examples of this might be the click paths created when users navigate or browse a Web site, or the order in which a customer adds items to a shopping cart at an online retailer. The algorithm finds the most common sequences by grouping, or *clustering*, sequences that are identical.

Requirements

The requirements for a Microsoft Sequence Clustering model are:

- **ID field.** The Microsoft Sequence Clustering algorithm requires the sequence information to be stored in transactional format. For this, an ID field that identifies each transaction is required.
- **At least one input field.** The algorithm requires at least one input field.
- **Sequence field.** The algorithm also requires a sequence identifier field, which must have a measurement level of Continuous. For example, you can use a Web page identifier, an integer, or a text string, as long as the field identifies the events in a sequence. Only one sequence identifier is allowed for each sequence, and only one type of sequence is allowed in each model. The Sequence field must be different from the ID and Unique fields.
- **Target field.** A target field is required when building a sequence clustering model.
- **Unique field.** A sequence clustering model requires a key field that uniquely identifies records. You can set the Unique field to be the same as the ID field.

MS Sequence Clustering Fields Options

All modeling nodes have a Fields tab, where you specify the fields to be used in building the model.

Before you can build a sequence clustering model, you need to specify which fields you want to use as targets and as inputs. Note that for the MS Sequence Clustering node, you cannot use field information from an upstream Type node; you must specify the field settings here.

ID. Select an ID field from the list. Numeric or symbolic fields can be used as the ID field. Each unique value of this field should indicate a specific unit of analysis. For example, in a market basket application, each ID might represent a single customer. For a Web log analysis application, each ID might represent a computer (by IP address) or a user (by login data).

Inputs. Select the input field or fields for the model. These are the fields that contain the events of interest in sequence modeling.

Sequence. Choose a field from the list, to be used as the sequence identifier field. For example, you can use a Web page identifier, an integer, or a text string, provided that the field identifies the events in a sequence. Only one sequence identifier is allowed for each sequence, and only one type of sequence is allowed in each model. The Sequence field must be different from the ID field (specified on this tab) and the Unique field (specified on the Model tab).

Target. Choose a field to be used as the target field, that is, the field whose value you are trying to predict based on the sequence data.

MS Sequence Clustering Expert Options

The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

Scoring Analysis Services Models

Model scoring occurs within SQL Server and is performed by Analysis Services. The dataset may need to be uploaded to a temporary table if the data originates within IBM SPSS Modeler or needs to be prepared within IBM SPSS Modeler. Models that you create from IBM SPSS Modeler, using in-database mining, are actually a remote model held on the remote data mining or database server. This is an important distinction to understand as you browse and score models created using Microsoft Analysis Services algorithms.

In IBM SPSS Modeler, generally only a single prediction and associated probability or confidence is delivered.

For model scoring examples, see "Analysis Services Mining Examples" on page 21.

Settings Common to All Analysis Services Models

The following settings are common to all Analysis Services models.

Analysis Services Model Nugget Server Tab

The Server tab is used to specify connections for in-database mining. The tab also provides the unique model key. The key is randomly generated when the model is built and is stored both within the model in IBM SPSS Modeler and also within the description of the model object stored in the Analysis Services database.

On the Server tab, you can configure the Analysis server host and database and the SQL Server data source for the scoring operation. Options specified here overwrite those specified in the Helper Applications or Build Model dialog boxes in IBM SPSS Modeler. See the topic "Enabling Integration with Analysis Services" on page 11 for more information.

Model GUID. The model key is shown here. The key is randomly generated when the model is built and is stored both within the model in IBM SPSS Modeler and also within the description of the model object stored in the Analysis Services database.

Check. Click this button to check the model key against the key in the model stored in the Analysis Services database. This allows you to verify that the model still exists within the Analysis server and indicates that the structure of the model has not changed.

Note: The Check button is available only for models added to the stream canvas in preparation for scoring. If the check fails, investigate whether the model has been deleted or replaced by a different model on the server.

View. Click for a graphical view of the decision tree model. The Decision Tree Viewer is shared by other decision tree algorithms in IBM SPSS Modeler and the functionality is identical.

Analysis Services Model Nugget Summary Tab

The Summary tab of a model nugget displays information about the model itself (*Analysis*), fields used in the model (*Fields*), settings used when building the model (*Build Settings*), and model training (*Training Summary*).

When you first browse the node, the Summary tab results are collapsed. To see the results of interest, use the expander control to the left of an item to unfold it or click the **Expand All** button to show all results. To hide the results when you have finished viewing them, use the expander control to collapse the specific results you want to hide or click the **Collapse All** button to collapse all results.

Analysis. Displays information about the specific model. If you have executed an Analysis node attached to this model nugget, information from that analysis will also appear in this section.

Fields. Lists the fields used as the target and the inputs in building the model.

Build Settings. Contains information about the settings used in building the model.

Training Summary. Shows the type of model, the stream used to create it, the user who created it, when it was built, and the elapsed time for building the model.

MS Time Series Model Nugget

The MS Time Series model produces scores for only for the predicted time periods, not for the historical data.

The following table shows the fields that are added to the model.

Table 1. Fields added to the model

Field Name	Description
$M-*field*	Predicted value of *field*
$Var-*field*	Computed variance of *field*
$Stdev-*field*	Standard deviation of *field*

MS Time Series Model Nugget Server Tab

The Server tab is used to specify connections for in-database mining. The tab also provides the unique model key. The key is randomly generated when the model is built and is stored both within the model in IBM SPSS Modeler and also within the description of the model object stored in the Analysis Services database.

On the Server tab, you can configure the Analysis server host and database and the SQL Server data source for the scoring operation. Options specified here overwrite those specified in the Helper Applications or Build Model dialog boxes in IBM SPSS Modeler. See the topic "Enabling Integration with Analysis Services" on page 11 for more information.

Model GUID. The model key is shown here. The key is randomly generated when the model is built and is stored both within the model in IBM SPSS Modeler and also within the description of the model object stored in the Analysis Services database.

Check. Click this button to check the model key against the key in the model stored in the Analysis Services database. This allows you to verify that the model still exists within the Analysis server and indicates that the structure of the model has not changed.

Note: The Check button is available only for models added to the stream canvas in preparation for scoring. If the check fails, investigate whether the model has been deleted or replaced by a different model on the server.

View. Click for a graphical view of the time series model. Analysis Services displays the completed model as a tree. You can also view a graph that shows the historical value of the target field over time, together with predicted future values.

For more information, see the description of the Time Series viewer in the MSDN library at http://msdn.microsoft.com/en-us/library/ms175331.aspx.

MS Time Series Model Nugget Settings Tab

Begin Estimation. Specify the time period where you want predictions to start.

- **Start From: New Prediction.** The time period at which you want future predictions to start, expressed as an offset from the last time period of your historical data. For example, if your historical data ended

at 12/99 and you wanted to begin predictions at 01/00, you would use a value of 1; however, if you wanted predictions to start at 03/00, you would use a value of 3.

- **Start From: Historical Prediction.** The time period at which you want historical predictions to start, expressed as a negative offset from the last time period of your historical data. For example, if your historical data ended at 12/99 and you wanted to make historical predictions for the last five time periods of your data, you would use a value of -5.

End Estimation. Specify the time period where you want predictions to stop.

- **End step of the prediction.** The time period at which you want predictions to stop, expressed as an offset from the last time period of your historical data. For example, if your historical data end at 12/99 and you want predictions to stop at 6/00, you would use a value of 6 here. For future predictions, the value must always be greater than or equal to the **Start From** value.

MS Sequence Clustering Model Nugget

The following table shows the fields that are added to the MS Sequence Clustering model (where *field* is the name of the target field).

Table 2. Fields added to the model

Field Name	Description
$MC-*field*	Prediction of the cluster to which this sequence belongs.
$MCP-*field*	Probability that this sequence belongs to the predicted cluster.
$MS-*field*	Predicted value of *field*
$MSP-*field*	Probability that $MS-*field* value is correct.

Exporting Models and Generating Nodes

You can export a model summary and structure to text and HTML format files. You can generate the appropriate Select and Filter nodes where appropriate.

Similar to other model nuggets in IBM SPSS Modeler, the Microsoft Analysis Services model nuggets support the direct generation of record and field operations nodes. Using the model nugget Generate menu options, you can generate the following nodes:

- Select node (only if an item is selected on the Model tab)
- Filter node

Analysis Services Mining Examples

Included are a number of sample streams that demonstrate the use of MS Analysis Services data mining with IBM SPSS Modeler. These streams can be found in the IBM SPSS Modeler installation folder under:

\Demos\Database_Modelling\Microsoft

Note: The Demos folder can be accessed from the IBM SPSS Modeler program group on the Windows Start menu.

Example Streams: Decision Trees

The following streams can be used together in sequence as an example of the database mining process using the Decision Trees algorithm provided by MS Analysis Services.

Table 3. Decision Trees - example streams	
Stream	**Description**
1_upload_data.str	Used to clean and upload data from a flat file into the database.
2_explore_data.str	Provides an example of data exploration with IBM SPSS Modeler
3_build_model.str	Builds the model using the database-native algorithm.
4_evaluate_model.str	Used as an example of model evaluation with IBM SPSS Modeler
5_deploy_model.str	Deploys the model for in-database scoring.

Note: In order to run the example, streams must be executed in order. In addition, source and modeling nodes in each stream must be updated to reference a valid data source for the database you want to use.

The dataset used in the example streams concerns credit card applications and presents a classification problem with a mixture of categorical and continuous predictors. For more information about this dataset, see the *crx.names* file in the same folder as the sample streams.

This dataset is available from the UCI Machine Learning Repository at *ftp://ftp.ics.uci.edu/pub/machine-learning-databases/credit-screening/*.

Example Stream: Upload Data

The first example stream, *1_upload_data.str*, is used to clean and upload data from a flat file into SQL Server.

Since Analysis Services data mining requires a key field, this initial stream uses a Derive node to add a new field to the dataset called *KEY* with unique values *1,2,3* using the IBM SPSS Modeler `@INDEX` function.

The subsequent Filler node is used for missing-value handling and replaces empty fields read in from the text file *crx.data* with *NULL* values.

Example Stream: Explore Data

The second example stream, *2_explore_data.str*, is used to demonstrate use of a Data Audit node to gain a general overview of the data, including summary statistics and graphs.

Double-clicking a graph in the Data Audit Report produces a more detailed graph for deeper exploration of a given field.

Example Stream: Build Model

The third example stream, *3_build_model.str*, illustrates model building in IBM SPSS Modeler. You can attach the database model to the stream and double-click to specify build settings.

On the Model tab of the dialog box, you can specify the following:

1. Select the **Key** field as the Unique ID field.

On the Expert tab, you can fine-tune settings for building the model.

Before running, ensure that you have specified the correct database for model building. Use the Server tab to adjust any settings.

Example Stream: Evaluate Model

The fourth example stream, *4_evaluate_model.str*, illustrates the advantages of using IBM SPSS Modeler for in-database modeling. Once you have executed the model, you can add it back to your data stream and evaluate the model using several tools offered in IBM SPSS Modeler.

Viewing Modeling Results

You can double-click the model nugget to explore your results. The Summary tab provides a rule-tree view of your results. You can also click the **View** button, located on the Server tab, for a graphical view of the Decision Trees model.

Evaluating Model Results

The Analysis node in the sample stream creates a coincidence matrix showing the pattern of matches between each predicted field and its target field. Execute the Analysis node to view the results.

The Evaluation node in the sample stream can create a gains chart designed to show accuracy improvements made by the model. Execute the Evaluation node to view the results.

Example Stream: Deploy Model

Once you are satisfied with the accuracy of the model, you can deploy it for use with external applications or for publishing back to the database. In the final example stream, *5_deploy_model.str*, data is read from the table CREDIT and then scored and published to the table CREDITSCORES using a Database Export node.

Running the stream generates the following SQL:

```
DROP TABLE CREDITSCORES

CREATE TABLE CREDITSCORES ( "field1" varchar(1),"field2" varchar(255),"field3" f
loat,"field4" varchar(1),"field5" varchar(2),"field6" varchar(2),"field7" varcha
r(2),"field8" float,"field9" varchar(1),"field10" varchar(1),"field11" int,"fiel
d12" varchar(1),"field13" varchar(1),"field14" int,"field15" int,"field16" varch
ar(1),"KEY" int,"$M-field16" varchar(9),"$MC-field16" float )

INSERT INTO CREDITSCORES ("field1","field2","field3","field4","field5","field6","field7","field8",
"field9","field10","field11","field12","field13","field14","field15","field16",
"KEY","$M-field16","$MC-field16")
SELECT T0.C0 AS C0,T0.C1 AS C1,T0.C2 AS C2,T0.C3 AS C3,T0.C4 AS C4,T0.C5 AS C5,
    T0.C6 AS C6,T0.C7 AS C7,T0.C8 AS C8,T0.C9 AS C9,T0.C10 AS C10,
    T0.C11 AS C11,T0.C12 AS C12,T0.C13 AS C13,T0.C14 AS C14,
    T0.C15 AS C15,T0.C16 AS C16,T0.C17 AS C17,T0.C18 AS C18
FROM (
    SELECT CONVERT(NVARCHAR,[TA].[field1])  AS C0, CONVERT(NVARCHAR,[TA].[field2])  AS C1,
        [TA].[field3] AS C2, CONVERT(NVARCHAR,[TA].[field4])  AS C3,
        CONVERT(NVARCHAR,[TA].[field5])  AS C4, CONVERT(NVARCHAR,[TA].[field6]) AS C5,
        CONVERT(NVARCHAR,[TA].[field7])  AS C6, [TA].[field8] AS C7,
        CONVERT(NVARCHAR,[TA].[field9])  AS C8, CONVERT(NVARCHAR,[TA].[field10])  AS C9,
        [TA].[field11] AS C10, CONVERT(NVARCHAR,[TA].[field12])  AS C11,
        CONVERT(NVARCHAR,[TA].[field13])  AS C12, [TA].[field14] AS C13,
        [TA].[field15] AS C14, CONVERT(NVARCHAR,[TA].[field16])  AS C15,
        [TA].[KEY] AS C16, CONVERT(NVARCHAR,[TA].[$M-field16])  AS C17,
        [TA].[$MC-field16] AS C18
    FROM openrowset('MSOLAP',
        'Datasource=localhost;Initial catalog=FoodMart 2000',
        'SELECT [T].[C0] AS [field1],[T].[C1] AS [field2],[T].[C2] AS [field3],
            [T].[C3] AS [field4],[T].[C4] AS [field5],[T].[C5] AS [field6],
            [T].[C6] AS [field7],[T].[C7] AS [field8],[T].[C8] AS [field9],
            [T].[C9] AS [field10],[T].[C10] AS [field11],[T].[C11] AS [field12],
            [T].[C12] AS [field13],[T].[C13] AS [field14],[T].[C14] AS [field15],
            [T].[C15] AS [field16],[T].[C16] AS [KEY],[CREDIT1].[field16] AS [$M-field16],
            PredictProbability([CREDIT1].[field16]) AS [$MC-field16]
        FROM [CREDIT1] PREDICTION JOIN
            openrowset(''MSDASQL'',
            ''Dsn=LocalServer;Uid=;pwd='',''SELECT T0."field1" AS C0,T0."field2" AS C1,
            T0."field3" AS C2,T0."field4" AS C3,T0."field5" AS C4,T0."field6" AS C5,
            T0."field7" AS C6,T0."field8" AS C7,T0."field9" AS C8,T0."field10" AS C9,
            T0."field11" AS C10,T0."field12" AS C11,T0."field13" AS C12,
            T0."field14" AS C13,T0."field15" AS C14,T0."field16" AS C15,
            T0."KEY" AS C16 FROM "dbo".CREDITDATA T0'') AS [T]
        ON [T].[C2] = [CREDIT1].[field3] and [T].[C7] = [CREDIT1].[field8]
            and [T].[C8] = [CREDIT1].[field9] and [T].[C9] = [CREDIT1].[field10]
            and [T].[C10] = [CREDIT1].[field11] and [T].[C11] = [CREDIT1].[field12]
            and [T].[C14] = [CREDIT1].[field15]') AS [TA]
)  T0
```

Chapter 4. Database Modeling with Oracle Data Mining

About Oracle Data Mining

IBM SPSS Modeler supports integration with Oracle Data Mining (ODM), which provides a family of data mining algorithms tightly embedded within the Oracle RDBMS. These features can be accessed through the IBM SPSS Modeler graphical user interface and workflow-oriented development environment, allowing customers to use the data mining algorithms offered by ODM.

IBM SPSS Modeler supports integration of the following algorithms from Oracle Data Mining:

- Naive Bayes
- Adaptive Bayes
- Support Vector Machine (SVM)
- Generalized Linear Models (GLM)*
- Decision Tree
- O-Cluster
- k-Means
- Nonnegative Matrix Factorization (NMF)
- Apriori
- Minimum Descriptor Length (MDL)
- Attribute Importance (AI)

* 11*g* R1 only

Requirements for Integration with Oracle

The following conditions are prerequisites for conducting in-database modeling using Oracle Data Mining. You may need to consult with your database administrator to ensure that these conditions are met.

- IBM SPSS Modeler running in local mode or against an IBM SPSS Modeler Server installation on Windows or UNIX.
- Oracle 10*g*R2 or 11*g*R1 (10.2 Database or higher) with the Oracle Data Mining option.

Note: 10*g*R2 provides support for all database modeling algorithms except Generalized Linear Models (requires 11*g*R1).

- An ODBC data source for connecting to Oracle as described below.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > **About** > **Additional Details**

If connectivity is enabled, you see the option **Server Enablement** in the License Status tab.

Enabling Integration with Oracle

To enable the IBM SPSS Modeler integration with Oracle Data Mining, you'll need to configure Oracle, create an ODBC source, enable the integration in the IBM SPSS Modeler Helper Applications dialog box, and enable SQL generation and optimization.

Configuring Oracle

To install and configure Oracle Data Mining, see the Oracle documentation—in particular, the *Oracle Administrator's Guide*—for more details.

Creating an ODBC Source for Oracle

To enable the connection between Oracle and IBM SPSS Modeler, you need to create an ODBC system data source name (DSN).

Before creating a DSN, you should have a basic understanding of ODBC data sources and drivers, and database support in IBM SPSS Modeler.

If you are running in distributed mode against IBM SPSS Modeler Server, create the DSN on the server computer. If you are running in local (client) mode, create the DSN on the client computer.

1. Install the ODBC drivers. These are available on the IBM SPSS Data Access Pack installation disk shipped with this release. Run the *setup.exe* file to start the installer, and select all the relevant drivers. Follow the on-screen instructions to install the drivers.
   a. Create the DSN.

      Note: The menu sequence depends on your version of Windows.

      - **Windows XP.** From the Start menu, choose **Control Panel**. Double-click **Administrative Tools**, and then double-click **Data Sources (ODBC)**.
      - **Windows Vista.** From the Start menu, choose **Control Panel**, then **System Maintenance**. Double-click **Administrative Tools**, select**Data Sources (ODBC)**, then click **Open**.
      - **Windows 7.** From the Start menu, choose **Control Panel**, then **System & Security**, then **Administrative Tools**. Select**Data Sources (ODBC)**, then click **Open**.

   b. Go to the **System DSN** tab, and then click **Add**.
2. Select the **SPSS OEM 6.0 Oracle Wire Protocol** driver.
3. Click **Finish**.
4. In the ODBC Oracle Wire Protocol Driver Setup screen, enter a data source name of your choosing, the hostname of the Oracle server, the port number for the connection, and the SID for the Oracle instance you are using.

   The hostname, port, and SID can be obtained from the *tnsnames.ora* file on the server machine if you have implemented TNS with a *tnsnames.ora* file. Contact your Oracle administrator for more information.
5. Click the **Test** button to test the connection.

Enabling Oracle Data Mining Integration in IBM SPSS Modeler

1. From the IBM SPSS Modeler menus choose:

   Tools > **Options** > **Helper Applications**
2. Click the **Oracle** tab.

Enable Oracle Data Mining Integration. Enables the Database Modeling palette (if not already displayed) at the bottom of the IBM SPSS Modeler window and adds the nodes for Oracle Data Mining algorithms.

Oracle Connection. Specify the default Oracle ODBC data source used for building and storing models, along with a valid user name and password. This setting can be overridden on the individual modeling nodes and model nuggets.

Note: The database connection used for modeling purposes may or may not be the same as the connection used to access data. For example, you might have a stream that accesses data from one Oracle database, downloads the data to IBM SPSS Modeler for cleaning or other manipulations, and then uploads the data to a different Oracle database for modeling purposes. Alternatively, the original data might reside in a flat file or other (non-Oracle) source, in which case it would need to be uploaded to Oracle for modeling. In all cases, the data will be automatically uploaded to a temporary table created in the database that is used for modeling.

Warn when about to overwrite an Oracle Data Mining model. Select this option to ensure that models stored in the database are not overwritten by IBM SPSS Modeler without warning.

List Oracle Data Mining Models. Displays available data mining models.

Enable launch of Oracle Data Miner. (optional) When enabled, allows IBM SPSS Modeler to launch the Oracle Data Miner application. Refer to "Oracle Data Miner" on page 41 for more information.

Path for Oracle Data Miner executable. (optional) Specifies the physical location of the Oracle Data Miner for Windows executable file (for example, *C:\odm\bin\odminerw.exe*). Oracle Data Miner is not installed with IBM SPSS Modeler; you must download the correct version from the Oracle Web site (*http://www.oracle.com/technology/products/bi/odm/odminer.html*) and install it at the client.

Enabling SQL Generation and Optimization

1. From the IBM SPSS Modeler menus choose:

   Tools > **Stream Properties** > **Options**
2. Click the **Optimization** option in the navigation pane.
3. Confirm that the **Generate SQL** option is enabled. This setting is required for database modeling to function.
4. Select **Optimize SQL Generation** and **Optimize other execution** (not strictly required but strongly recommended for optimized performance).

Building Models with Oracle Data Mining

Oracle model-building nodes work just like other modeling nodes in IBM SPSS Modeler, with a few exceptions. You can access these nodes from the Database Modeling palette across the bottom of the IBM SPSS Modeler window.

Data Considerations

Oracle requires that categorical data be stored in a string format (either CHAR or VARCHAR2). As a result, IBM SPSS Modeler will not allow numeric storage fields with a measurement level of *Flag* or *Nominal* (categorical) to be specified as input to ODM models. If necessary, numbers can be converted to strings in IBM SPSS Modeler by using the Reclassify node.

Target field. Only one field may be selected as the output (target) field in ODM classification models.

Model name. From Oracle 11gR1 onwards, the name `unique` is a keyword and cannot be used as a custom model name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

General Comments

- PMML Export/Import is not provided from IBM SPSS Modeler for models created by Oracle Data Mining.
- Model scoring always happens within ODM. The dataset may need to be uploaded to a temporary table if the data originate, or need to be prepared, within IBM SPSS Modeler.
- In IBM SPSS Modeler, generally only a single prediction and associated probability or confidence is delivered.

- IBM SPSS Modeler restricts the number of fields that can be used in model building and scoring to 1,000.
- IBM SPSS Modeler can score ODM models from within streams published for execution by using IBM SPSS Modeler Solution Publisher.

Oracle Models Server Options

Specify the Oracle connection that is used to upload data for modeling. If necessary, you can select a connection on the Server tab for each modeling node to override the default Oracle connection specified in the Helper Applications dialog box. See the topic "Enabling Integration with Oracle" on page 26 for more information.

Comments

- The connection used for modeling may or may not be the same as the connection used in the source node for a stream. For example, you might have a stream that accesses data from one Oracle database, downloads the data to IBM SPSS Modeler for cleaning or other manipulations, and then uploads the data to a different Oracle database for modeling purposes.
- The ODBC data source name is effectively embedded in each IBM SPSS Modeler stream. If a stream that is created on one host is executed on a different host, the name of the data source must be the same on each host. Alternatively, a different data source can be selected on the Server tab in each source or modeling node.

Misclassification Costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

With the exception of C5.0 models, misclassification costs are not applied when scoring a model and are not taken into account when ranking or comparing models using an Auto Classifier node, evaluation chart, or Analysis node. A model that includes costs may not produce fewer errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of *less expensive* errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select **Use misclassification costs** and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying *A* as *B* to be 2.0, the cost of misclassifying *B* as *A* will still have the default value of 1.0 unless you explicitly change it as well.

Note: Only the Decision Trees model allows costs to be specified at build time.

Oracle Naive Bayes

Naive Bayes is a well-known algorithm for classification problems. The model is termed *naïve* because it treats all proposed prediction variables as being independent of one another. Naive Bayes is a fast, scalable algorithm that calculates conditional probabilities for combinations of attributes and the target attribute. From the training data, an independent probability is established. This probability gives the likelihood of each target class, given the occurrence of each value category from each input variable.

- Cross-validation is used to test model accuracy on the same data that were used to build the model. This is particularly useful when the number of cases available to build a model is small.

- The model output can be browsed in a matrix format. The numbers in the matrix are conditional probabilities that relate the predicted classes (columns) and predictor variable-value combinations (rows).

Naive Bayes Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Naive Bayes Expert Options

When the model is built, individual predictor attribute values or value pairs are ignored unless there are enough occurrences of a given value or pair in the training data. The thresholds for ignoring values are specified as fractions based on the number of records in the training data. Adjusting these thresholds may reduce noise and improve the model's ability to generalize to other datasets.

- **Singleton Threshold.** Specifies the threshold for a given predictor attribute value. The number of occurrences of a given value must equal or exceed the specified fraction or the value is ignored.
- **Pairwise Threshold.** Specifies the threshold for a given attribute and predictor value pair. The number of occurrences of a given value pair must equal or exceed the specified fraction or the pair is ignored.

Prediction probability. Allows the model to include the probability of a correct prediction for a possible outcome of the target field. To enable this feature, choose **Select**, click the **Specify** button, choose one of the possible outcomes, and then click **Insert**.

Use Prediction Set. Generates a table of all the possible results for all possible outcomes of the target field.

Oracle Adaptive Bayes

Adaptive Bayes Network (ABN) constructs Bayesian Network classifiers by using Minimum Description Length (MDL) and automatic feature selection. ABN does well in certain situations where Naive Bayes does poorly and does at least as well in most other situations, although performance may be slower. The ABN algorithm provides the ability to build three types of advanced, Bayesian-based models, including simplified decision tree (single-feature), pruned Naive Bayes, and boosted multifeature models.

Note: The Oracle Adaptive Bayes algorithm has been dropped in Oracle 12C and is not supported in IBM SPSS Modeler when using Oracle 12C. See http://docs.oracle.com/database/121/DMPRG/release_changes.htm#DMPRG726.

Generated Models

In single-feature build mode, ABN produces a simplified decision tree, based on a set of human-readable rules, that allows the business user or analyst to understand the basis of the model's predictions and act or explain to others accordingly. This may be a significant advantage over Naive Bayes and multifeature

models. These rules can be browsed like a standard rule set in IBM SPSS Modeler. A simple set of rules might look like this:

```
IF MARITAL_STATUS = "Married"
AND EDUCATION_NUM = "13-16"
THEN CHURN= "TRUE"
Confidence = .78, Support = 570 cases
```

Pruned Naive Bayes and multifeature models cannot be browsed in IBM SPSS Modeler.

Adaptive Bayes Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Model Type

You can choose from three different modes for building the model.

- **Multi-feature.** Builds and compares a number of models, including an NB model and single and multifeature product probability models. This is the most exhaustive mode and typically takes the longest to compute as a result. Rules are produced only if the single feature model turns out to be best. If a multifeature or NB model is chosen, no rules are produced.
- **Single-feature.** Creates a simplified decision tree based on a set of rules. Each rule contains a condition together with probabilities associated with each outcome. The rules are mutually exclusive and are provided in a format that can be read by humans, which may be a significant advantage over Naive Bayes and multifeature models.
- **Naive Bayes.** Builds a single NB model and compares it with the global sample prior (the distribution of target values in the global sample). The NB model is produced as output only if it turns out to be a better predictor of the target values than the global prior. Otherwise, no model is produced as output.

Adaptive Bayes Expert Options

Limit execution time. Select this option to specify a maximum build time in minutes. This makes it possible to produce models in less time, although the resulting model may be less accurate. At each milestone in the modeling process, the algorithm checks whether it will be able to complete the next milestone within the specified amount of time before continuing and returns the best model available when the limit is reached.

Max Predictors. This option allows you to limit the complexity of the model and improve performance by limiting the number of predictors used. Predictors are ranked based on an MDL measure of their correlation to the target as a measure of their likelihood of being included in the model.

Max Naive Bayes Predictors. This option specifies the maximum number of predictors to be used in the Naive Bayes model.

Oracle Support Vector Machine (SVM)

Support Vector Machine (SVM) is a classification and regression algorithm that uses machine learning theory to maximize predictive accuracy without overfitting the data. SVM uses an optional nonlinear transformation of the training data, followed by the search for regression equations in the transformed data to separate the classes (for categorical targets) or fit the target (for continuous targets). Oracle's implementation of SVM allows models to be built by using one of two available kernels, linear or

Gaussian. The linear kernel omits the nonlinear transformation altogether so that the resulting model is essentially a regression model.

For more information, see the *Oracle Data Mining Application Developer's Guide* and *Oracle Data Mining Concepts*.

Oracle SVM Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Active Learning. Provides a way to deal with large build sets. With active learning, the algorithm creates an initial model based on a small sample before applying it to the complete training dataset and then incrementally updates the sample and model based on the results. The cycle is repeated until the model converges on the training data or the maximum allowed number of support vectors is reached.

Kernel Function. Select **Linear** or **Gaussian**, or leave the default **System Determined** to allow the system to choose the most suitable kernel. Gaussian kernels are able to learn more complex relationships but generally take longer to compute. You may want to start with the linear kernel and try the Gaussian kernel only if the linear kernel fails to find a good fit. This is more likely to happen with a regression model, where the choice of kernel matters more. Also, note that SVM models built with the Gaussian kernel cannot be browsed in IBM SPSS Modeler. Models built with the linear kernel can be browsed in IBM SPSS Modeler in the same manner as standard regression models.

Normalization Method. Specifies the normalization method for continuous input and target fields. You can choose **Z-Score**, **Min-Max**, or **None**. Oracle performs normalization automatically if the **Auto Data Preparation** check box is selected. Uncheck this box to select the normalization method manually.

Oracle SVM Expert Options

Kernel Cache Size. Specifies, in bytes, the size of the cache to be used for storing computed kernels during the build operation. As might be expected, larger caches typically result in faster builds. The default is 50MB.

Convergence Tolerance. Specifies the tolerance value that is allowed before termination for the model build. The value must be between 0 and 1. The default value is 0.001. Larger values tend to result in faster building but less-accurate models.

Specify Standard Deviation. Specifies the standard deviation parameter used by the Gaussian kernel. This parameter affects the trade-off between model complexity and ability to generalize to other datasets (overfitting and underfitting the data). Higher standard deviation values favor underfitting. By default, this parameter is estimated from the training data.

Specify Epsilon. For regression models only, specifies the value of the interval of the allowed error in building epsilon-insensitive models. In other words, it distinguishes small errors (which are ignored) from large errors (which aren't). The value must be between 0 and 1. By default, this is estimated from the training data.

Specify Complexity Factor. Specifies the complexity factor, which trades off model error (as measured against the training data) and model complexity in order to avoid overfitting or underfitting the data. Higher values place a greater penalty on errors, with an increased risk of overfitting the data; lower values place a lower penalty on errors and can lead to underfitting.

Specify Outlier Rate. Specifies the desired rate of outliers in the training data. Only valid for One-Class SVM models. Cannot be used with the **Specify Complexity Factor** setting.

Prediction probability. Allows the model to include the probability of a correct prediction for a possible outcome of the target field. To enable this feature, choose **Select**, click the **Specify** button, choose one of the possible outcomes, and then click **Insert**.

Use Prediction Set. Generates a table of all the possible results for all possible outcomes of the target field.

Oracle SVM Weights Options

In a classification model, using weights enables you to specify the relative importance of the various possible target values. Doing so might be useful, for example, if the data points in your training data are not realistically distributed among the categories. Weights enable you to bias the model so that you can compensate for those categories that are less well represented in the data. Increasing the weight for a target value should increase the percentage of correct predictions for that category.

There are three methods of setting weights:

- **Based on training data.** This is the default. Weights are based on the relative frequencies of the categories in the training data.
- **Equal for all classes.** Weights for all categories are defined as 1/*k*, where *k* is the number of target categories.
- **Custom.** You can specify your own weights. Starting values for weights are set as equal for all classes. You can adjust the weights for individual categories to user-defined values. To adjust a specific category's weight, select the Weight cell in the table corresponding to the desired category, delete the contents of the cell, and enter the desired value.

The weights for all categories should sum to 1.0. If they do not sum to 1.0, a warning is displayed, with an option to automatically normalize the values. This automatic adjustment preserves the proportions across categories while enforcing the weight constraint. You can perform this adjustment at any time by clicking the **Normalize** button. To reset the table to equal values for all categories, click the **Equalize** button.

Oracle Generalized Linear Models (GLM)

(11g only) Generalized linear models relax the restrictive assumptions made by linear models. These include, for example, the assumptions that the target variable has a normal distribution, and that the effect of the predictors on the target variable is linear in nature. A generalized linear model is suitable for predictions where the distribution of the target is likely to have a non-normal distribution, such as a multinomial or a Poisson distribution. Similarly, a generalized linear model is useful in cases where the relationship, or link, between the predictors and the target is likely to be non-linear.

For more information, see the *Oracle Data Mining Application Developer's Guide* and *Oracle Data Mining Concepts*.

Oracle GLM Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Normalization Method. Specifies the normalization method for continuous input and target fields. You can choose **Z-Score**, **Min-Max**, or **None**. Oracle performs normalization automatically if the **Auto Data Preparation** check box is selected. Uncheck this box to select the normalization method manually.

Missing Value Handling. Specifies how to process missing values in the input data:

- **Replace with mean or mode** replaces missing values of numerical attributes with the mean value, and replaces missing values of categorical attributes with the mode.
- **Only use complete records** ignores records with missing values.

Oracle GLM Expert Options

Use Row Weights. Check this box to activate the adjacent drop-down list, from where you can select a column that contains a weighting factor for the rows.

Save Row Diagnostics to Table. Check this box to activate the adjacent text field, where you can enter the name of a table to contain row-level diagnostics.

Coefficient Confidence Level. The degree of certainty, from 0.0 to 1.0, that the value predicted for the target will lie within a confidence interval computed by the model. Confidence bounds are returned with the coefficient statistics.

Reference Category for Target. Select **Custom** to choose a value for the target field to use as a reference category, or leave the default value **Auto**.

Ridge Regression. Ridge regression is a technique that compensates for the situation where there is too high a degree of correlation in the variables. You can use the **Auto** option to allow the algorithm to control the use of this technique, or you can control it manually by means of the **Disable** and **Enable** options. If you choose to enable ridge regression manually, you can override the system default value for the ridge parameter by entering a value in the adjacent field.

Produce VIF for Ridge Regression. Check this box if you want to produce Variance Inflation Factor (VIF) statistics when ridge is being used for linear regression.

Prediction probability. Allows the model to include the probability of a correct prediction for a possible outcome of the target field. To enable this feature, choose **Select**, click the **Specify** button, choose one of the possible outcomes, and then click **Insert**.

Use Prediction Set. Generates a table of all the possible results for all possible outcomes of the target field.

Oracle GLM Weights Options

In a classification model, using weights enables you to specify the relative importance of the various possible target values. Doing so might be useful, for example, if the data points in your training data are not realistically distributed among the categories. Weights enable you to bias the model so that you can compensate for those categories that are less well represented in the data. Increasing the weight for a target value should increase the percentage of correct predictions for that category.

There are three methods of setting weights:

- **Based on training data.** This is the default. Weights are based on the relative frequencies of the categories in the training data.
- **Equal for all classes.** Weights for all categories are defined as $1/k$, where k is the number of target categories.
- **Custom.** You can specify your own weights. Starting values for weights are set as equal for all classes. You can adjust the weights for individual categories to user-defined values. To adjust a specific category's weight, select the Weight cell in the table corresponding to the desired category, delete the contents of the cell, and enter the desired value.

The weights for all categories should sum to 1.0. If they do not sum to 1.0, a warning is displayed, with an option to automatically normalize the values. This automatic adjustment preserves the proportions across

categories while enforcing the weight constraint. You can perform this adjustment at any time by clicking the **Normalize** button. To reset the table to equal values for all categories, click the **Equalize** button.

Oracle Decision Tree

Oracle Data Mining offers a classic Decision Tree feature, based on the popular Classification and Regression Tree algorithm. The ODM Decision Tree model contains complete information about each node, including Confidence, Support, and Splitting Criterion. The full Rule for each node can be displayed, and in addition, a surrogate attribute is supplied for each node, to be used as a substitute when applying the model to a case with missing values.

Decision trees are popular because they are so universally applicable, easy to apply and easy to understand. Decision trees sift through each potential input attribute searching for the best "splitter," that is, attribute cutpoint (`AGE > 55`, for example) that splits the downstream data records into more homogeneous populations. After each split decision, ODM repeats the process growing out the entire tree and creating terminal "leaves" that represent similar populations of records, items, or people. Looking down from the root tree node (for example, the total population), decision trees provide human readable rules of `IF A, then B` statements. These decision tree rules also provide the support and confidence for each tree node.

While Adaptive Bayes Networks can also provide short simple rules that can be useful in providing explanations for each prediction, Decision Trees provide full Oracle Data Mining rules for each splitting decision. Decision Trees are also useful for developing detailed profiles of the best customers, healthy patients, factors associated with fraud, and so on.

Decision Tree Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Impurity metric. Specifies which metric is used for seeking the best test question for splitting data at each node. The best splitter and split value are those that result in the largest increase in target value homogeneity for the entities in the node. Homogeneity is measured in accordance with a metric. The supported metrics are **gini** and **entropy**.

Decision Tree Expert Options

Maximum Depth. Sets the maximum depth of the tree model to be built.

Minimum percentage of records in a node. Sets the percentage of the minimum number of records per node.

Minimum percentage of records for a split. Sets the minimum number of records in a parent node expressed as a percent of the total number of records used to train the model. No split is attempted if the number of records is below this percentage.

Minimum records in a node. Sets the minimum number of records returned.

Minimum records for a split. Sets the minimum number of records in a parent node expressed as a value. No split is attempted if the number of records is below this value.

Rule identifier. If checked, includes in the model a string to identify the node in the tree at which a particular split is made.

Prediction probability. Allows the model to include the probability of a correct prediction for a possible outcome of the target field. To enable this feature, choose **Select**, click the **Specify** button, choose one of the possible outcomes, and then click **Insert**.

Use Prediction Set. Generates a table of all the possible results for all possible outcomes of the target field.

Oracle O-Cluster

The Oracle O-Cluster algorithm identifies naturally occurring groupings within a data population. Orthogonal partitioning clustering (O-Cluster) is an Oracle proprietary clustering algorithm that creates a hierarchical grid-based clustering model, that is, it creates axis-parallel (orthogonal) partitions in the input attribute space. The algorithm operates recursively. The resulting hierarchical structure represents an irregular grid that tessellates the attribute space into clusters.

The O-Cluster algorithm handles both numeric and categorical attributes and ODM will automatically select the best cluster definitions. ODM provides cluster detail information, cluster rules, cluster centroid values, and can be used to score a population on their cluster membership.

O-Cluster Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Maximum number of clusters. Sets the maximum number of generated clusters.

O-Cluster Expert Options

Maximum Buffer. Sets the maximum buffer size.

Sensitivity. Sets a fraction that specifies the peak density required for separating a new cluster. The fraction is related to the global uniform density.

Oracle k-Means

The Oracle k-Means algorithm identifies naturally occurring groupings within a data population. The k-Means algorithm is a distance-based clustering algorithm that partitions the data into a predetermined number of clusters (provided there are enough distinct cases). Distance-based algorithms rely on a distance metric (function) to measure the similarity between data points. Data points are assigned to the nearest cluster according to the distance metric used. ODM provides an enhanced version of k-Means.

The k-Means algorithm supports hierarchical clusters, handles numeric and categorical attributes, and cuts the population into the user-specified number of clusters. ODM provides cluster detail information, cluster rules, cluster centroid values, and can be used to score a population on their cluster membership.

k-Means Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Number of clusters. Sets the number of generated clusters

Distance Function. Specifies which distance function is used for k-Means Clustering.

Split criterion. Specifies which split criterion is used for k-Means Clustering.

Normalization Method. Specifies the normalization method for continuous input and target fields. You can choose **Z-Score**, **Min-Max**, or **None**.

k-Means Expert Options

Iterations. Sets the number of iterations for the k-Means algorithm.

Convergence tolerance. Sets the convergence tolerance for the k-Means algorithm.

Number of bins. Specifies the number of bins in the attribute histogram produced by k-Means. The bin boundaries for each attribute are computed globally on the entire training dataset. The binning method is equi-width. All attributes have the same number of bins with the exception of attributes with a single value that have only one bin.

Block growth. Sets the growth factor for memory allocated to hold cluster data.

Minimum Percent Attribute Support. Sets the fraction of attribute values that must be non-null in order for the attribute to be included in the rule description for the cluster. Setting the parameter value too high in data with missing values can result in very short or even empty rules.

Oracle Nonnegative Matrix Factorization (NMF)

Nonnegative Matrix Factorization (NMF) is useful for reducing a large dataset into representative attributes. Similar to Principal Components Analysis (PCA) in concept, but able to handle larger amounts of attributes and in an additive representation model, NMF is a powerful, state-of-the-art data mining algorithm that can be used for a variety of use cases.

NMF can be used to reduce large amounts of data, text data for example, into smaller, more sparse representations that reduce the dimensionality of the data (the same information can be preserved using far fewer variables). The output of NMF models can be analyzed using supervised learning techniques such as SVMs or unsupervised learning techniques such as clustering techniques. Oracle Data Mining uses NMF and SVM algorithms to mine unstructured text data.

NMF Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Normalization Method. Specifies the normalization method for continuous input and target fields. You can choose **Z-Score**, **Min-Max**, or **None**. Oracle performs normalization automatically if the **Auto Data Preparation** check box is selected. Uncheck this box to select the normalization method manually.

NMF Expert Options

Specify number of features. Specifies the number of features to be extracted.

Random seed. Sets the random seed for the NMF algorithm.

Number of iterations. Sets the number of iterations for the NMF algorithm.

Convergence tolerance. Sets the convergence tolerance for the NMF algorithm.

Display all features. Displays the feature ID and confidence for all features, instead of those values for only the best feature.

Oracle Apriori

The Apriori algorithm discovers association rules in data. For example, "if a customer purchases a razor and after shave, then that customer will purchase shaving cream with 80% confidence." The association mining problem can be decomposed into two subproblems:

- Find all combinations of items, called frequent itemsets, whose support is greater than the minimum support.
- Use the frequent itemsets to generate the desired rules. The idea is that if, for example, ABC and BC are frequent, then the rule "A implies BC" holds if the ratio of `support(ABC)` to `support(BC)` is at least as large as the minimum confidence. Note that the rule will have minimum support because ABCD is frequent. ODM Association only supports single consequent rules (ABC implies D).

The number of frequent itemsets is governed by the minimum support parameters. The number of rules generated is governed by the number of frequent itemsets and the confidence parameter. If the confidence parameter is set too high, there may be frequent itemsets in the association model but no rules.

ODM uses an SQL-based implementation of the Apriori algorithm. The candidate generation and support counting steps are implemented using SQL queries. Specialized in-memory data structures are not used. The SQL queries are fine-tuned to run efficiently in the database server by using various hints.

Apriori Fields Options

All modeling nodes have a Fields tab, where you can specify the fields to be used in building the model.

Before you can build an Apriori model, you need to specify which fields you want to use as the items of interest in association modeling.

Use type node settings. This option tells the node to use field information from an upstream Type node. This is the default.

Use custom settings. This option tells the node to use field information specified here instead of that given in any upstream Type node(s). After selecting this option, specify the remaining fields on the dialog, which depend on whether you are using transactional format.

If you are *not* using transactional format, specify:

- **Inputs.** Select the input field(s). This is similar to setting a field role to *Input* in a Type node.
- **Partition.** This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building.

If you *are* using transactional format, specify:

Use transactional format. Use this option if you want to transform data from a row per item, to a row per case.

Selecting this option changes the field controls in the lower part of this dialog box:

For transactional format, specify:

- **ID.** Select an ID field from the list. Numeric or symbolic fields can be used as the ID field. Each unique value of this field should indicate a specific unit of analysis. For example, in a market basket application, each ID might represent a single customer. For a Web log analysis application, each ID might represent a computer (by IP address) or a user (by login data).
- **Content.** Specify the content field for the model. This field contains the item of interest in association modeling.
- **Partition.** This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to create the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)

Apriori Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Maximum rule length. Sets the maximum number of preconditions for any rule, an integer from 2 to 20. This is a way to limit the complexity of the rules. If the rules are too complex or too specific, or if your rule set is taking too long to train, try decreasing this setting.

Minimum confidence. Sets the minimum confidence level, a value between 0 and 1. Rules with lower confidence than the specified criterion are discarded.

Minimum support. Sets the minimum support threshold, a value between 0 and 1. Apriori discovers patterns with frequency above the minimum support threshold.

Oracle Minimum Description Length (MDL)

The Oracle Minimum Description Length (MDL) algorithm helps to identify the attributes that have the greatest influence on a target attribute. Oftentimes, knowing which attributes are most influential helps you to better understand and manage your business and can help simplify modeling activities. Additionally, these attributes can indicate the types of data you may wish to add to augment your models. MDL might be used, for example, to find the process attributes most relevant to predicting the quality of a manufactured part, the factors associated with churn, or the genes most likely involved in the treatment of a particular disease.

Oracle MDL discards input fields that it regards as unimportant in predicting the target. With the remaining input fields it then builds an unrefined model nugget that is associated with an Oracle model, visible in Oracle Data Miner. Browsing the model in Oracle Data Miner displays a chart showing the remaining input fields, ranked in order of their significance in predicting the target.

Negative ranking indicates noise. Input fields ranked at zero or less do not contribute to the prediction and should probably be removed from the data.

To display the chart

1. Right-click on the unrefined model nugget in the Models palette and choose **Browse**.

2. From the model window, click the button to launch Oracle Data Miner.
3. Connect to Oracle Data Miner. See the topic "Oracle Data Miner" on page 41 for more information.
4. In the Oracle Data Miner navigator panel, expand **Models**, then **Attribute Importance**.
5. Select the relevant Oracle model (it will have the same name as the target field you specified in IBM SPSS Modeler). If you are not sure which is the correct one, select the Attribute Importance folder and look for a model by creation date.

MDL Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. IBM SPSS Modeler imposes a restriction that this key field must be numeric.

Note: This field is optional for all Oracle nodes except Oracle Adaptive Bayes, Oracle O-Cluster and Oracle Apriori.

Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

Oracle Attribute Importance (AI)

The objective of attribute importance is to find out which attributes in the data set are related to the result, and the degree to which they influence the final outcome. The Oracle Attribute Importance node analyzes data, finds patterns, and predicts outcomes or results with an associated level of confidence.

AI Model Options

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

AI Selection Options

The Options tab allows you to specify the default settings for selecting or excluding input fields in the model nugget. You can then add the model to a stream to select a subset of fields for use in subsequent model-building efforts. Alternatively, you can override these settings by selecting or deselecting additional fields in the model browser after generating the model. However, the default settings make it possible to apply the model nugget without further changes, which may be particularly useful for scripting purposes.

The following options are available:

All fields ranked. Selects fields based on their ranking as *important*, *marginal*, or *unimportant*. You can edit the label for each ranking as well as the cutoff values used to assign records to one rank or another.

Top number of fields. Selects the top *n* fields based on importance.

Importance greater than. Selects all fields with importance greater than the specified value.

The target field is always preserved regardless of the selection.

AI Model Nugget Model Tab

The Model tab for an Oracle AI model nugget displays the rank and importance of all inputs, and allows you to select fields for filtering by using the check boxes in the column on the left. When you run the stream, only the checked fields are preserved, together with the target prediction. The other input fields are discarded. The default selections are based on the options specified in the modeling node, but you can select or deselect additional fields as needed.

- To sort the list by rank, field name, importance, or any of the other displayed columns, click on the column header. Alternatively, select the desired item from the list next to the Sort By button, and use the up and down arrows to change the direction of the sort.
- You can use the toolbar to check or uncheck all fields and to access the Check Fields dialog box, which allows you to select fields by rank or importance. You can also press the Shift or Ctrl keys while clicking on fields to extend the selection.
- The threshold values for ranking inputs as important, marginal, or unimportant are displayed in the legend below the table. These values are specified in the modeling node.

Managing Oracle Models

Oracle models are added to the Models palette just like other IBM SPSS Modeler models and can be used in much the same way. However, there are a few important differences, given that each Oracle model created in IBM SPSS Modeler actually references a model stored on a database server.

Oracle Model Nugget Server Tab

Building an ODM model via IBM SPSS Modeler creates a model in IBM SPSS Modeler and creates or replaces a model in the Oracle database. An IBM SPSS Modeler model of this kind references the content of a database model stored on a database server. IBM SPSS Modeler can perform consistency checking by storing an identical generated **model key** string in both the IBM SPSS Modeler model and the Oracle model.

The key string for each Oracle model is displayed under the *Model Information* column in the List Models dialog box. The key string for an IBM SPSS Modeler model is displayed as the **Model Key** on the Server tab of an IBM SPSS Modeler model (when placed into a stream).

The Check button on the Server tab of a model nugget can be used to check that the model keys in the IBM SPSS Modeler model and the Oracle model match. If no model of the same name can be found in Oracle or if the model keys do not match, the Oracle model has been deleted or rebuilt since the IBM SPSS Modeler model was built.

Oracle Model Nugget Summary Tab

The Summary tab of a model nugget displays information about the model itself (*Analysis*), fields used in the model (*Fields*), settings used when building the model (*Build Settings*), and model training (*Training Summary*).

When you first browse the node, the Summary tab results are collapsed. To see the results of interest, use the expander control to the left of an item to unfold it or click the **Expand All** button to show all results. To hide the results when you have finished viewing them, use the expander control to collapse the specific results you want to hide or click the **Collapse All** button to collapse all results.

Analysis. Displays information about the specific model. If you have executed an Analysis node attached to this model nugget, information from that analysis will also appear in this section.

Fields. Lists the fields used as the target and the inputs in building the model.

Build Settings. Contains information about the settings used in building the model.

Training Summary. Shows the type of model, the stream used to create it, the user who created it, when it was built, and the elapsed time for building the model.

Oracle Model Nugget Settings Tab

The Settings tab on the model nugget allows you to override the setting of certain options on the modeling node for scoring purposes.

Oracle Decision Tree

Use misclassification costs. Determines whether to use misclassification costs in the Oracle Decision Tree model. See the topic "Misclassification Costs" on page 28 for more information.

Rule identifier. If selected (checked), adds a rule identifier column to the Oracle Decision Tree model. The rule identifier identifies the node in the tree at which a particular split is made.

Oracle NMF

Display all features. If selected (checked), displays the feature ID and confidence for all features, instead of those values for only the best feature, in the Oracle NMF model.

Listing Oracle Models

The List Oracle Data Mining Models button launches a dialog box that lists the existing database models and allows models to be removed. This dialog box can be launched from the Helper Applications dialog box and from the build, browse, and apply dialog boxes for ODM-related nodes.

The following information is displayed for each model:

- **Model Name.** Name of the model, which is used to sort the list
- **Model Information.** Model key information composed of build date/time and target column name
- **Model Type.** Name of the algorithm that built this model

Oracle Data Miner

Oracle Data Miner is the user interface to Oracle Data Mining (ODM) and replaces the previous IBM SPSS Modeler user interface to ODM. Oracle Data Miner is designed to increase the analyst's success rate in properly utilizing ODM algorithms. These goals are addressed in several ways:

- Users need more assistance in applying a methodology that addresses both data preparation and algorithm selection. Oracle Data Miner meets this need by providing Data Mining Activities to step users through the proper methodology.
- Oracle Data Miner includes improved and expanded heuristics in the model building and transformation wizards to reduce the chance of error in specifying model and transformation settings.

Defining an Oracle Data Miner Connection

1. Oracle Data Miner can be launched from all Oracle build, apply nodes, and output dialog boxes via the **Launch Oracle Data Miner** button.

   

   Figure 2. Launch Oracle Data Miner Button
2. The Oracle Data Miner **Edit Connection** dialog box is presented to the user before the Oracle Data Miner external application is launched (provided the Helper Application option is properly defined).

   Note: This dialog box only displays in the absence of a defined connection name.

   - Provide a Data Miner connection name and enter the appropriate Oracle 10gR1 or 10gR2 server information. The Oracle server should be the same server specified in IBM SPSS Modeler.
3. The Oracle Data Miner **Choose Connection** dialog box provides options for specifying which connection name, defined in the above step, is used.

Refer to Oracle Data Miner on the Oracle Web site for more information regarding Oracle Data Miner requirements, installation, and usage.

Preparing the Data

Two types of data preparation may be useful when you are using the Naive Bayes, Adaptive Bayes, and Support Vector Machine provided with Oracle Data Mining algorithms in modeling:

- **Binning**, or conversion of continuous numeric range fields to categories for algorithms that cannot accept continuous data.
- **Normalization**, or transformations applied to numeric ranges so that they have similar means and standard deviations.

Binning

IBM SPSS Modeler's Binning node offers a number of techniques for performing binning operations. A binning operation is defined that can be applied to one or many fields. Executing the binning operation on a dataset creates the thresholds and allows an IBM SPSS Modeler Derive node to be created. The derive operation can be converted to SQL and applied prior to model building and scoring. This approach creates a dependency between the model and the Derive node that performs the binning but allows the binning specifications to be reused by multiple modeling tasks.

Normalization

Continuous (numeric range) fields that are used as inputs to Support Vector Machine models should be normalized prior to model building. In the case of regression models, normalization must also be reversed to reconstruct the score from the model output. The SVM model settings allow you to choose **Z-Score**, **Min-Max**, or **None**. The normalization coefficients are constructed by Oracle as a step in the model-building process, and the coefficients are uploaded to IBM SPSS Modeler and stored with the model. At apply time, the coefficients are converted into IBM SPSS Modeler derive expressions and used to prepare the data for scoring before passing the data to the model. In this case, normalization is closely associated with the modeling task.

Oracle Data Mining Examples

A number of sample streams are included that demonstrate the use of ODM with IBM SPSS Modeler. These streams can be found in the IBM SPSS Modeler installation folder under *\Demos\Database_Modelling\Oracle Data Mining*.

Note: The Demos folder can be accessed from the IBM SPSS Modeler program group on the Windows Start menu.

The streams in the following table can be used together in sequence as an example of the database mining process, using the Support Vector Machine (SVM) algorithm that is provided with Oracle Data Mining:

Table 4. Database mining - example streams

Stream	Description
1_upload_data.str	Used to clean and upload data from a flat file into the database.
2_explore_data.str	Provides an example of data exploration with IBM SPSS Modeler
3_build_model.str	Builds the model using the database-native algorithm.
4_evaluate_model.str	Used as an example of model evaluation with IBM SPSS Modeler
5_deploy_model.str	Deploys the model for in-database scoring.

Note: In order to run the example, streams must be executed in order. In addition, source and modeling nodes in each stream must be updated to reference a valid data source for the database you want to use.

The dataset used in the example streams concerns credit card applications and presents a classification problem with a mixture of categorical and continuous predictors. For more information about this dataset, see the *crx.names* file in the same folder as the sample streams.

This dataset is available from the UCI Machine Learning Repository at *ftp://ftp.ics.uci.edu/pub/machine-learning-databases/credit-screening/*.

Example Stream: Upload Data

The first example stream, *1_upload_data.str*, is used to clean and upload data from a flat file into Oracle.

Since Oracle Data Mining requires a unique ID field, this initial stream uses a Derive node to add a new field to the dataset called *ID*, with unique values 1,2,3, using the IBM SPSS Modeler `@INDEX` function.

The Filler node is used for missing-value handling and replaces empty fields that are read from the text file *crx.data* with *NULL* values.

Example Stream: Explore Data

The second example stream, *2_explore_data.str*, is used to demonstrate use of a Data Audit node to gain a general overview of the data, including summary statistics and graphs.

Double-clicking a graph in the Data Audit Report produces a more detailed graph for deeper exploration of a given field.

Example Stream: Build Model

The third example stream, *3_build_model.str*, illustrates model building in IBM SPSS Modeler. Double-click the Database source node (labeled CREDIT) to specify the data source. To specify build settings, double-click the build node (initially labeled CLASS, which changes to FIELD16 when the data source is specified).

On the Model tab of the dialog box:

1. Ensure that **ID** is selected as the Unique field.
2. Ensure that **Linear** is selected as the kernel function and **Z-Score** is the normalization method.

Example Stream: Evaluate Model

The fourth example stream, *4_evaluate_model.str*, illustrates the advantages of using IBM SPSS Modeler for in-database modeling. Once you have executed the model, you can add it back to your data stream and evaluate the model by using several tools offered in IBM SPSS Modeler.

Viewing Modeling Results

Attach a Table node to the model nugget to explore your results. The **$O-field16** field shows the predicted value for *field16* in each case, and the **$OC-field16** field shows the confidence value for this prediction.

Evaluating Model Results

You can use the Analysis node to create a coincidence matrix showing the pattern of matches between each predicted field and its target field. Run the Analysis node to see the results.

You can use the Evaluation node to create a gains chart designed to show accuracy improvements made by the model. Run the Evaluation node to see the results.

Example Stream: Deploy Model

Once you are satisfied with the accuracy of the model, you can deploy it for use with external applications or for publishing back to the database. In the final example stream, *5_deploy_model.str*, data are read from the table `CREDITDATA` and then scored and published to the table `CREDITSCORES` using the Publisher node called *deploy solution*.

Chapter 5. Database Modeling with IBM Data Warehouse and IBM Netezza Analytics

SPSS Modeler with IBM Data Warehouse and IBM Netezza Analytics

IBM SPSS Modeler supports integration with IBM Data Warehouse and IBM Netezza® Analytics, which provides the ability to run advanced analytics on those IBM servers. These features can be accessed through the IBM SPSS Modeler graphical user interface and workflow-oriented development environment, allowing you to run the data mining algorithms directly in the IBM Netezza or IBM Data Warehouse environment.

SPSS Modeler supports integration of the following algorithms from **IBM Netezza Analytics**:

- Decision Trees
- K-Means
- TwoStep
- Bayes Net
- Naive Bayes
- KNN
- Divisive Clustering
- PCA
- Regression Tree
- Linear Regression
- Time Series
- Generalized Linear

For more information about these algorithms, see the *IBM Netezza Analytics Developer's Guide* and the *IBM Netezza Analytics Reference Guide*.

SPSS Modeler supports integration of the following algorithms from **IBM Data Warehouse** (Bayes Net, Divisive Clustering, and Time Series are not supported):

- Decision Trees
- K-Means
- TwoStep
- Naive Bayes
- KNN
- PCA
- Regression Tree
- Linear Regression
- Generalized Linear

Note: AIX isn't supported.

Integration requirements

The following conditions are prerequisites for conducting in-database modeling using IBM Netezza Analytics or IBM Data Warehouse. You may need to consult with your database administrator to ensure that these conditions are met.

- IBM SPSS Modeler running against an IBM SPSS Modeler Server installation on Windows or UNIX (except zLinux, for which IBM Netezza ODBC drivers are not available).
- IBM Netezza Performance Server, running theIBM Netezza Analytics package.

  Note: The minimum version of Netezza Performance Server (NPS) that is required depends on the version of INZA that is required and is as follows:

  – Anything greater than NPS 6.0.0 P8 will support INZA versions prior to 2.0.

  – To use INZA 2.0 or greater requires NPS 6.0.5 P5 or greater.

  Netezza Generalized Linear and Netezza Time Series require INZA 2.0 and above to be functional. All the other Netezza In-Database nodes need INZA 1.1 or later.
- An ODBC data source for connecting to an IBM Netezza database. See the topic "Enabling integration" on page 46 for more information.
- An ODBC data source for connecting to an IBM Data Warehouse database.
- SQL generation and optimization enabled in IBM SPSS Modeler. See the topic "Enabling integration" on page 46 for more information.

Note: Database modeling and SQL optimization require that IBM SPSS Modeler Server connectivity be enabled on the IBM SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from IBM SPSS Modeler, and access IBM SPSS Modeler Server. To verify the current license status, choose the following from the IBM SPSS Modeler menu.

Help > **About** > **Additional Details**

If connectivity is enabled, you see the option **Server Enablement** in the License Status tab.

Enabling integration

Enabling integration with IBM Netezza Analytics or IBM Data Warehouse consists of the following steps.

- Configuring IBM Netezza Analytics or IBM Data Warehouse
- Creating an ODBC source
- Enabling the integration in IBM SPSS Modeler
- Enabling SQL generation and optimization in IBM SPSS Modeler

These are described in the sections that follow.

Configuring IBM Netezza Analytics or IBM Data Warehouse

To install and configure IBM Netezza Analytics or IBM Data Warehouse, refer to the appropriate IBM documentation. For example, for IBM Netezza Analytics, see the *IBM Netezza Analytics Installation Guide* provided with that product. The section *Setting Database Permissions* in that guide contains details of scripts that need to be run to allow IBM SPSS Modeler streams to write to the database.

Note: If you will be using nodes that rely on matrix calculation, the Matrix Engine must be initialized by running `CALL NZM..INITIALIZE();` otherwise execution of stored procedures will fail. Initialization is a one-time setup step for each database.

Creating an ODBC Source for IBM Netezza Analytics

To enable the connection between the IBM Netezza database and IBM SPSS Modeler, you need to create an ODBC system data source name (DSN).

Before creating a DSN, you should have a basic understanding of ODBC data sources and drivers, and database support in IBM SPSS Modeler.

If you are running in distributed mode against IBM SPSS Modeler Server, create the DSN on the server computer. If you are running in local (client) mode, create the DSN on the client computer.

Windows clients

1. From your *Netezza Client* CD, run the *nzodbcsetup.exe* file to start the installer. Follow the on-screen instructions to install the driver. For full instructions, see the IBM Netezza ODBC, JDBC, and OLE DB Installation and Configuration Guide.
   a. Create the DSN.

      Note: The menu sequence depends on your version of Windows.

      - **Windows XP.** From the Start menu, choose **Control Panel**. Double-click **Administrative Tools**, and then double-click **Data Sources (ODBC)**.
      - **Windows Vista.** From the Start menu, choose **Control Panel**, then **System Maintenance**. Double-click **Administrative Tools**, select**Data Sources (ODBC)**, then click **Open**.
      - **Windows 7.** From the Start menu, choose **Control Panel**, then **System & Security**, then **Administrative Tools**. Select**Data Sources (ODBC)**, then click **Open**.

   b. Go to the **System DSN** tab, and then click **Add**.
2. Select **NetezzaSQL** from the list and click **Finish**.
3. On the **DSN Options** tab of the Netezza ODBC Driver Setup screen, type a data source name of your choosing, the hostname or IP address of the IBM Netezza server, the port number for the connection, the database of the IBM Netezza instance you are using, and your username and password details for the database connection. Click the **Help** button for an explanation of the fields.
4. Click the **Test Connection** button and ensure that you can connect to the database.
5. When you have a successful connection, click **OK** repeatedly to exit from the ODBC Data Source Administrator screen.

Windows servers

The procedure for Windows Server is the same as the client procedure for Windows XP.

UNIX or Linux servers

The following procedure applies to UNIX or Linux servers (except zLinux, for which IBM Netezza ODBC drivers are not available).

1. From your Netezza Client CD/DVD, copy the relevant `<platform>cli.package.tar.gz` file to a temporary location on the server.
2. Extract the archive contents by means of the **`gunzip`** and **`untar`** commands.
3. Add execute permissions to the *unpack* script that is extracted.
4. Run the script, answering the on-screen prompts.
5. Edit the `modelersrv.sh` file to include the following lines.

```
. <SDAP Install Path>/odbc.sh
LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH:/usr/local/nz/lib64; export LD_LIBRARY_PATH_64
NZ_ODBC_INI_PATH=<SDAP Install Path>; export NZ_ODBC_INI_PATH
```

   For example:

```
. /usr/IBM/SPSS/SDAP/odbc.sh
LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH:/usr/local/nz/lib64; export LD_LIBRARY_PATH_64
NZ_ODBC_INI_PATH=/usr/IBM/SPSS/SDAP; export NZ_ODBC_INI_PATH
```

6. Locate the file `/usr/local/nz/lib64/odbc.ini` and copy its contents into the `odbc.ini` file that is installed with SDAP (the one defined by the $ODBCINI environment variable).

   Note: For 64-bit Linux systems, the **`Driver`** parameter incorrectly references the 32-bit driver. When you copy the `odbc.ini` contents in the previous step, edit the path within this parameter accordingly, for example:

   ```
   /usr/local/nz/lib64/libnzodbc.so
   ```

7. Edit the parameters in the Netezza DSN definition to reflect the database to be used.
8. Restart IBM SPSS Modeler Server and test the use of the Netezza in-database mining nodes on the client.

Enabling integration in SPSS Modeler

1. From the IBM SPSS Modeler main menu, choose

   Tools > **Options** > **Helper Applications.**

2. Click the **IBM Data Warehouse** tab.

Enable IBM Data Warehouse Analytics Integration. Enables the Database Modeling palette (if not already displayed) at the bottom of the IBM SPSS Modeler window and adds the nodes for IBM Data Warehouse and Netezza Data Mining algorithms.

IBM Data Warehouse Connection. Click the **Edit** button and choose the IBM Data Warehouse connection string that you set up when creating the ODBC source. For more information, see the IBM Data Warehouse admin console.

Enabling SQL Generation and Optimization

Because of the likelihood of working with very large data sets, for performance reasons you should enable the SQL generation and optimization options in IBM SPSS Modeler.

1. From the IBM SPSS Modeler menus choose:

   Tools > **Stream Properties** > **Options**

2. Click the **Optimization** option in the navigation pane.
3. Confirm that the **Generate SQL** option is enabled. This setting is required for database modeling to function.
4. Select **Optimize SQL Generation** and **Optimize other execution** (not strictly required but strongly recommended for optimized performance).

Building models with IBM Netezza Analytics and IBM Data Warehouse

Each of the supported algorithms has a corresponding modeling node. You can access the IBM Data Warehouse and IBM Netezza modeling nodes from the **Database Modeling** tab on the nodes palette.

Data considerations

Fields in the data source can contain variables of various data types, depending on the modeling node. In IBM SPSS Modeler, data types are known as *measurement levels*. The Fields tab of the modeling node uses icons to indicate the permitted measurement level types for its input and target fields.

Target field The target field is the field whose value you are trying to predict. Where a target can be specified, only one of the source data fields can be selected as the target field.

Record ID field Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. If the source data does not include an ID field, you can create this field by means of a Derive node, as the following procedure shows.

1. Select the source node.
2. From the Field Ops tab on the nodes palette, double-click the Derive node.
3. Open the Derive node by double-clicking its icon on the canvas.
4. In the **Derive field** field, type (for example) `ID`.
5. In the **Formula** field, type `@INDEX` and click **OK**.
6. Connect the Derive node to the rest of the stream.

Note: If you retrieve long numeric data from a Netezza database by using the `NUMERIC(18,0)` data type, SPSS Modeler can sometimes round up the data during import. To avoid this issue, store your data using either the `BIGINT`, or `NUMERIC(36,0)` data type.

Note: Due to the limitations on the types of fields that can be used, a field with a typeless Measurement level and a role of Record ID does not appear in a Netezza In-Database modeling node (for example, K-Means).

Handling null values

If the input data contains null values, use of some Netezza nodes may result in error messages or long-running streams, so we recommend removing records containing null values. Use the following method.

1. Attach a Select node to the source node.
2. Set the **Mode** option of the Select node to **Discard**.
3. Enter the following in the **Condition** field:

   ```
   @NULL(field1) [or @NULL(field2)[... or @NULL(fieldN]])
   ```

   Be sure to include every input field.
4. Connect the Select node to the rest of the stream.

Model output

It is possible for a stream containing a Data Warehouse or Netezza modeling node to produce slightly different results each time it is run. This is because the order in which the node reads the source data is not always the same, as the data is read into temporary tables before model building. However, the differences produced by this effect are negligible.

General comments

- In IBM SPSS Collaboration and Deployment Services, it is not possible to create scoring configurations using streams containing IBM Data Warehouse or IBM Netezza database modeling nodes.
- PMML export or import is not possible for models created by the Data Warehouse or Netezza nodes.

Field options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the **All** button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Target. Choose one field as the target for the prediction. For Generalized Linear models, see also the **Trials** field on this screen.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

Server options

On the Server tab, you specify the IBM Data Warehouse database where the model is to be built.

IBM Data Warehouse Server Details. Here you specify the connection details for the database you want to use for the model.

- **Use upstream connection**. (default) Uses the connection details specified in an upstream node, for example the Database source node. This option works only if all upstream nodes are able to use SQL pushback. In this case there is no need to move the data out of the database, as the SQL fully implements all of the upstream nodes.
- **Move data to connection**. Moves the data to the database you specify here. Doing so allows modeling to work if the data is in another IBM Data Warehouse database, or a database from another vendor, or even if the data is in a flat file. In addition, data is moved back to the database specified here if the data has been extracted because a node did not perform SQL pushback. Click the **Edit** button to browse for and select a connection.

CAUTION: IBM Netezza Analytics and IBM Data Warehouse is typically used with very large data sets. Transferring large amounts of data between databases, or out of and back into a database, can be very time-consuming and should be avoided where possible.

Note: The ODBC data source name is effectively embedded in each IBM SPSS Modeler stream. If a stream that is created on one host is executed on a different host, the name of the data source must be the same on each host. Alternatively, a different data source can be selected on the Server tab in each source or modeling node.

Model options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also set default values for scoring options.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Replace existing if the name has been used. If you select this check box, any existing model of the same name will be overwritten.

Make Available for Scoring. You can set the default values here for the scoring options that appear on the dialog for the model nugget. For details of the options, see the help topic for the Settings tab of that particular nugget.

Managing models

Building an IBM Netezza or IBM Data Warehouse model via SPSS Modeler creates a model in SPSS Modeler and creates or replaces a model in the IBM Data Warehouse database. The SPSS Modeler model of this kind references the content of a database model stored on a database server. SPSS Modeler can perform consistency checking by storing an identical generated model key string in both the SPSS Modeler model and the Netezza or Data Warehouse model.

The model name for each Netezza or Data Warehouse model is displayed under the *Model Information* column in the Listing Database Models dialog box. The model name for an SPSS Modeler model is displayed as the Model Key on the Server tab of an SPSS Modeler model (when placed into a stream).

The Check button can be used to check that the model keys in the SPSS Modeler model and the Netezza or Data Warehouse model match. If no model of the same name can be found in Netezza or Data Warehouse, or if the model keys do not match, the Netezza or Data Warehouse model has been deleted or rebuilt since the SPSS Modeler model was built.

Listing database models

SPSS Modeler provides a dialog box for listing the models that are stored in IBM Data Warehouse and enables models to be deleted. This dialog box is accessible from the IBM Helper Applications dialog box and from the build, browse, and apply dialog boxes for IBM Data Warehouse and IBM Netezza data mining-related nodes. The following information is displayed for each model:

- Model name (name of the model, which is used to sort the list).
- Owner name.
- The algorithm used in the model.
- The current state of the model; for example, Complete.
- The date on which the model was created.

IBM Data WH Regression Tree

A regression tree is a tree-based algorithm that splits a sample of cases repeatedly to derive subsets of the same kind, based on values of a numeric target field. As with decision trees, regression trees decompose the data into subsets in which the leaves of the tree correspond to sufficiently small or sufficiently uniform subsets. Splits are selected to decrease the dispersion of target attribute values, so that they can be reasonably well predicted by their mean values at leaves.

IBM Data WH Regression Tree Build Options - Tree Growth

You can set build options for tree growth and tree pruning.

The following build options are available for tree growth:

Maximum tree depth. The maximum number of levels to which the tree can grow below the root node, that is, the number of times the sample is split recursively. The default is 62, which is the maximum tree depth for modeling purposes.

Note: If the viewer in the model nugget shows the textual representation of the model, a maximum of 12 levels of the tree is displayed.

Splitting Criteria. These options control when to stop splitting the tree. If you do not want to use the default values, click **Customize** and change the values.

- **Split evaluation measure.** This class evaluation measure evaluates the best place to split the tree.

  Note: Currently, variance is the only possible option.
- **Minimum improvement for splits.** The minimum amount by which impurity must be reduced before a new split is created in the tree. The goal of tree building is to create subgroups with similar output values to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the amount that is specified by the splitting criteria, the branch is not split.
- **Minimum number of instances for a split.** The minimum number of records that can be split. When fewer than this number of unsplit records remain, no further splits are made. You can use this field to prevent the creation of small subgroups in the tree.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- **All.** All column-related statistics and all value-related statistics are included.

  Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify **None**.

- **Columns.** Column-related statistics are included.
- **None.** Only statistics that are required to score the model are included.

IBM Data WH Tree Build Options - Tree Pruning

You can use the pruning options to specify pruning criteria for the regression tree. The intention of pruning is to reduce the risk of overfitting by removing overgrown subgroups that do not improve the expected accuracy on new data.

Pruning measure. The pruning measure ensures that the estimated accuracy of the model remains within acceptable limits after removing a leaf from the tree. You can select one of the following measures.

- **mse.** Mean squared error - (default) measures how close a fitted line is to the data points.
- **r2.** R-squared - measures the proportion of variation in the dependent variable explained by the regression model.
- **Pearson.** Pearson's correlation coefficient - measures the strength of relationship between linearly dependent variables that are normally distributed.
- **Spearman.** Spearman's correlation coefficient - detects nonlinear relationships that appear weak according to Pearson’s correlation, but which may actually be strong.

Data for pruning. You can use some or all of the training data to estimate the expected accuracy on new data. Alternatively, you can use a separate pruning dataset from a specified table for this purpose.

- **Use all training data**. This option (the default) uses all the training data to estimate the model accuracy.
- **Use % of training data for pruning**. Use this option to split the data into two sets, one for training and one for pruning, using the percentage specified here for the pruning data.

  Select **Replicate results** if you want to specify a random seed to ensure that the data is partitioned in the same way each time you run the stream. You can either specify an integer in the **Seed used for pruning** field, or click **Generate**, which will create a pseudo-random integer.
- **Use data from an existing table**. Specify the table name of a separate pruning dataset for estimating model accuracy. Doing so is considered more reliable than using training data. However, this option may result in the removal of a large subset of data from the training set, thus reducing the quality of the decision tree.

Netezza Divisive Clustering

Divisive clustering is a method of cluster analysis in which the algorithm is run repeatedly to divide clusters into subclusters until a specified stopping point is reached.

Cluster formation begins with a single cluster containing all training instances (records). The first iteration of the algorithm divides the data set into two subclusters, with subsequent iterations dividing these into further subclusters. The stopping criteria are specified as a maximum number of iterations, a maximum number of levels to which the data set is divided, and a minimum required number of instances for further partitioning.

The resulting hierarchical clustering tree can be used to classify instances by propagating them down from the root cluster, as in the following example.

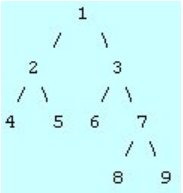

Figure 3. Example of a divisive clustering tree

At each level, the best matching subcluster is chosen with respect to the distance of the instance from the subcluster centers.

When the instances are scored with an applied hierarchy level of -1 (the default), the scoring returns only a leaf cluster, as leaves are designated by a negative number. In the example, this would be one of clusters 4, 5, 6, 8, or 9. However, if the hierarchy level is set to 2, for example, scoring would return one of the clusters at the second level below the root cluster, namely 4, 5, 6, or 7.

Netezza Divisive Clustering Field Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the **All** button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

Netezza Divisive Clustering Build Options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the **Run** button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- **Euclidean**. (default) The distance between two points is computed by joining them with a straight line.
- **Manhattan**. The distance between two points is calculated as the sum of the absolute differences between their co-ordinates.
- **Canberra**. Similar to Manhattan distance, but more sensitive to data points closer to the origin.
- **Maximum**. The distance between two points is calculated as the greatest of their differences along any coordinate dimension.

Maximum number of iterations. The algorithm operates by performing several iterations of the same process. This option allows you to stop model training after the number of iterations specified.

Maximum depth of cluster trees. The maximum number of levels to which the data set can be subdivided.

Replicate results. Check this box if you want to set a random seed, which will enable you to replicate analyses. You can either specify an integer or click **Generate**, which creates a pseudo-random integer.

Minimum number of instances for a split. The minimum number of records that can be split. When fewer than this number of unsplit records remain, no further splits will be made. You can use this field to prevent the creation of very small subgroups in the cluster tree.

IBM Data WH Generalized Linear

Linear regression is a long-established statistical technique for classifying records based on the values of numeric input fields. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values. Linear models are useful in modeling a wide range of real-world phenomena owing to their simplicity in both training and model application. However, linear models assume a normal distribution in the dependent (target) variable and a linear impact of the independent (predictor) variables on the dependent variable.

There are many situations where a linear regression is useful but the above assumptions do not apply. For example, when modeling consumer choice between a discrete number of products, the dependent variable is likely to have a multinomial distribution. Equally, when modeling income against age, income typically increases as age increases, but the link between the two is unlikely to be as simple as a straight line.

For these situations, a generalized linear model can be used. Generalized linear models expand the linear regression model so that the dependent variable is related to the predictor variables by means of a specified link function, for which there is a choice of suitable functions. Moreover, the model allows for the dependent variable to have a non-normal distribution, such as Poisson.

The algorithm iteratively seeks the best-fitting model, up to a specified number of iterations. In calculating the best fit, the error is represented by the sum of squares of the differences between the predicted and actual value of the dependent variable.

IBM Data WH Generalized Linear Model Field Options

On the Fields tab, you choose whether you want to use the field role settings that are already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings, such as targets or predictors from an upstream Type node, or the Types tab of an upstream source node.

Use custom field assignments. Choose this option if you want to assign targets, predictors, and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the **All** button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Target. Choose one field as the target for the prediction.

Record ID. The field that is to be used as the unique record identifier. The values of this field must be unique for each record, for example, customer ID numbers.

Instance Weight. Specify a field to use instance weights. An instance weight is a weight per row of input data. By default, all input records are assumed to have equal relative importance. You can change the importance by assigning individual weights to the input records. The field that you specify must contain a numeric weight for each row of input data.

Predictors (Inputs). Select the input field or fields. This action is similar to setting the field role to *Input* in a Type node.

IBM Data WH Generalized Linear Model Options - General

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also make various settings relating to the model, the link function, the input field interactions (if any), and set default values for scoring options.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Field options. You can specify the roles of the input fields for building the model.

General Settings. These settings relate to the stopping criteria for the algorithm.

- **Maximum number of iterations.** The maximum number of iterations the algorithm will perform; minimum is 1, default is 20.
- **Maximum error (1e).** The maximum error value (in scientific notation) at which the algorithm should stop finding the best fit model. Minimum is 0, default is -3, meaning 1E-3, or 0.001.

- **Insignificant error values threshold (1e).** The value (in scientific notation) below which errors are treated as having a value of zero. Minimum is -1, default is -7, meaning that error values below 1E-7 (or 0.0000001) are counted as insignificant.

Distribution Settings. These settings relate to the distribution of the dependent (target) variable.

- **Distribution of response variable.** The distribution type; one of **Bernoulli** (default), **Gaussian**, **Poisson**, **Binomial**, **Negative binomial**, **Wald** (Inverse Gaussian), and **Gamma**.
- **Parameters.** (Poisson or binomial distribution only) You must specify one of the following options in the **Specify parameter** field:
  - To automatically have the parameter estimated from data, select **Default**.
  - To allow optimization of the distribution quasi-likelihood, select **Quasi**.
  - To explicitly specify the parameter value, select **Explicit**.

  (Binomial distribution only) You must specify the input table column that is to be used as the trials field as required by binomial distribution. This column contains the number of trials for the binomial distribution.

  (Negative binomial distribution only) You can use the default of -1 or specify a different parameter value.

Link Function Settings. These settings relate to the link function, which relates the dependent variable to the predictor variables.

- **Link function.** The function to be used; one of **Identity**, **Inverse**, **Invnegative**, **Invsquare**, **Sqrt**, **Power**, **Oddspower**, **Log**, **Clog**, **Loglog**, **Cloglog**, **Logit** (default), **Probit**, **Gaussit**, **Cauchit**, **Canbinom**, **Cangeom**, **Cannegbinom**.
- **Parameters.** (Power or Oddspower link functions only) You can specify a parameter value if the link function is **Power** or **Oddspower**. Choose to either specify a value, or use the default of 1.

IBM Data WH Generalized Linear Model Options - Interaction

The Interaction panel contains the options for specifying interactions (that is, multiplicative effects between input fields).

Column Interaction. Select this check box to specify interactions between input fields. Leave the box cleared if there are no interactions.

Enter interactions into the model by selecting one or more fields in the source list and dragging to the interactions list. The type of interaction created depends upon the hotspot onto which you drop the selection.

- **Main.** Dropped fields appear as separate main interactions at the bottom of the interactions list.
- **2-way.** All possible pairs of the dropped fields appear as 2-way interactions at the bottom of the interactions list.
- **3-way.** All possible triplets of the dropped fields appear as 3-way interactions at the bottom of the interactions list.
- ***.** The combination of all dropped fields appear as a single interaction at the bottom of the interactions list.

Include Intercept. The intercept is usually included in the model. If you can assume the data passes through the origin, you can exclude the intercept.

Dialog box buttons

The buttons to the right of the display enable you to make changes to the terms used in the model.

Figure 4. Delete button

Delete terms from the model by selecting the terms you want to delete and clicking the delete button.

Figure 5. Reorder buttons

Reorder the terms within the model by selecting the terms you want to reorder and clicking the up or down arrow.

Figure 6. Custom interaction button

Add a Custom Term

You can specify custom interactions in the form *n1*x1*x1*x1...* Select a field from the **Fields** list, click the right-arrow button to add the field to **Custom Term**, click **By***, select the next field, click the right-arrow button, and so on. When you have built the custom interaction, click **Add term** to return it to the Interaction panel.

IBM Data WH Generalized Linear Model Options - Scoring Options

Make Available for Scoring. You can set the default values here for the scoring options that appear on the dialog for the model nugget. See the topic "IBM Data WH Generalized Linear Model Nugget - Settings tab" on page 78 for more information.

- **Include input fields.** Select this check box if you want to display the input fields in the model output as well as the predictions.

IBM Data WH Decision Trees

A decision tree is a hierarchical structure that represents a classification model. With a decision tree model, you can develop a classification system to predict or classify future observations from a set of training data. The classification takes the form of a tree structure in which the branches represent split points in the classification. The splits break the data down into subgroups recursively until a stopping point is reached. The tree nodes at the stopping points are known as **leaves**. Each leaf assigns a label, known as a **class label**, to the members of its subgroup, or class.

Instance weights and class weights

By default, all input records and classes are assumed to have equal relative importance. You can change this by assigning individual weights to the members of either or both of these items. Doing so might be useful, for example, if the data points in your training data are not realistically distributed among the categories. Weights enable you to bias the model so that you can compensate for those categories that are less well represented in the data. Increasing the weight for a target value should increase the percentage of correct predictions for that category.

In the Decision Tree modeling node, you can specify two types of weights. **Instance weights** assign a weight to each row of input data. The weights are typically specified as 1.0 for most cases, with higher or lower values given only to those cases that are more or less important than the majority, as shown in the following table.

Table 5. Instance weight example		
Record ID	**Target**	**Instance Weight**
1	drugA	1.1
2	drugB	1.0
3	drugA	1.0
4	drugB	0.3

Class weights assign a weight to each category of the target field, as shown in the following table.

Table 6. Class weight example	
Class	**Class Weight**
drugA	1.0
drugB	1.5

Both types of weights can be used at the same time, in which case they are multiplied together and used as instance weights. Thus if the two previous examples were used together, the algorithm would use the instance weights as shown in the following table.

Table 7. Instance weight calculation example		
Record ID	**Calculation**	**Instance Weight**
1	1.1*1.0	1.1
2	1.0*1.5	1.5
3	1.0*1.0	1.0
4	0.3*1.5	0.45

Netezza Decision Tree Field Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. To manually assign targets, predictors and other roles, select this option.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

Target. Select one field as the target for the prediction.

Record ID. The field that is to be used as the unique record identifier. The values of this field must be unique for each record (for example, customer ID numbers).

Instance Weight. Specifying a field here enables you to use instance weights (a weight per row of input data) instead of, or in addition to, the default, class weights (a weight per category for the target field). The field you specify here must be one that contains a numeric weight for each row of input data. See the topic “Instance weights and class weights” on page 56 for more information.

Predictors (Inputs). Select the input field or fields. This is similar to setting the field role to *Input* in a Type node.

IBM Data WH Decision Tree Build Options

The following build options are available for tree growth:

Growth Measure. These options control the way tree growth is measured.

- **Impurity Measure.** This measure evaluates the best place to split the tree. It is a measurement of the variability in a subgroup or segment of data. A low impurity measurement indicates a group where most members have similar values for the criterion or target field.

  The supported measurements are **Entropy** and **Gini**. These measurements are based on probabilities of category membership for the branch.
- **Maximum tree depth.** The maximum number of levels to which the tree can grow below the root node, that is, the number of times the sample is split recursively. The default value of this property is 10, and the maximal value that you can set for this property is 62.

  Note: If the viewer in the model nugget shows the textual representation of the model, a maximum of 12 levels of the tree is displayed.

Splitting Criteria. These options control when to stop splitting the tree.

- **Minimum improvement for splits.** The minimum amount by which impurity must be reduced before a new split is created in the tree. The goal of tree building is to create subgroups with similar output values to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the amount that is specified by the splitting criteria, the branch is not split.
- **Minimum number of instances for a split.** The minimum number of records that can be split. When fewer than this number of unsplit records remain, no further splits are made. You can use this field to prevent the creation of small subgroups in the tree.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- **All.** All column-related statistics and all value-related statistics are included.

  Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify **None**.
- **Columns.** Column-related statistics are included.
- **None.** Only statistics that are required to score the model are included.

IBM Data WH Decision Tree Node - Class Weights

Here you can assign weights to individual classes. The default is to assign a value of 1 to all classes, making them equally weighted. By specifying different numerical weights for different class labels, you instruct the algorithm to weight the training sets of particular classes accordingly.

To change a weight, double-click it in the **Weight** column and make the changes you want.

Value. The set of class labels, derived from the possible values of the target field.

Weight. The weighting to be assigned to a particular class. Assigning a higher weight to a class makes the model more sensitive to that class relative to the other classes.

You can use class weights in combination with instance weights. See the topic "Instance weights and class weights" on page 56 for more information.

IBM Data WH Decision Tree Node - Tree Pruning

You can use the pruning options to specify pruning criteria for the decision tree. The intention of pruning is to reduce the risk of overfitting by removing overgrown subgroups that do not improve the expected accuracy on new data.

Pruning measure. The default pruning measure, **Accuracy**, ensures that the estimated accuracy of the model remains within acceptable limits after removing a leaf from the tree. Use the alternative, **Weighted Accuracy**, if you want to take the class weights into account while applying pruning.

Data for pruning. You can use some or all of the training data to estimate the expected accuracy on new data. Alternatively, you can use a separate pruning dataset from a specified table for this purpose.

- **Use all training data**. This option (the default) uses all the training data to estimate the model accuracy.
- **Use % of training data for pruning**. Use this option to split the data into two sets, one for training and one for pruning, using the percentage specified here for the pruning data.

  Select **Replicate results** if you want to specify a random seed to ensure that the data is partitioned in the same way each time you run the stream. You can either specify an integer in the **Seed used for pruning** field, or click **Generate**, which will create a pseudo-random integer.
- **Use data from an existing table**. Specify the table name of a separate pruning dataset for estimating model accuracy. Doing so is considered more reliable than using training data. However, this option may result in the removal of a large subset of data from the training set, thus reducing the quality of the decision tree.

IBM Data WH Linear Regression

Linear models predict a continuous target based on linear relationships between the target and one or more predictors. While limited to directly modeling linear relationships only, linear regression models are relatively simple and give an easily interpreted mathematical formula for scoring. Linear models are fast, efficient and easy to use, although their applicability is limited compared to those produced by more refined regression algorithms.

IBM Data WH Linear Regression Build Options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the **Run** button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

Use Singular Value Decomposition to solve equations. Using the Singular Value Decomposition matrix instead of the original matrix has the advantage of being more robust against numerical errors, and can also speed up computation.

Include intercept in the model. Including the intercept increases the overall accuracy of the solution.

Calculate model diagnostics. This option causes a number of diagnostics to be calculated on the model. The results are stored in matrices or tables for later review. The diagnostics include r-squared, residual sum-of-squares, estimation of variance, standard deviation, *p*-value, and *t*-value.

These diagnostics relate to the validity and usefulness of the model. You should run separate diagnostics on the underlying data to ensure that it meets linearity assumptions.

IBM Data WH KNN

Nearest Neighbor Analysis is a method for classifying cases based on their similarity to other cases. In machine learning, it was developed as a way to recognize patterns of data without requiring an exact match to any stored patterns, or cases. Similar cases are near each other and dissimilar cases are distant from each other. Thus, the distance between two cases is a measure of their dissimilarity.

Cases that are near each other are said to be “neighbors.” When a new case (holdout) is presented, its distance from each of the cases in the model is computed. The classifications of the most similar cases – the nearest neighbors – are tallied and the new case is placed into the category that contains the greatest number of nearest neighbors.

You can specify the number of nearest neighbors to examine; this value is called *k*. The pictures show how a new case would be classified using two different values of *k*. When $k = 5$, the new case is placed in category *1* because a majority of the nearest neighbors belong to category *1*. However, when $k = 9$, the new case is placed in category *0* because a majority of the nearest neighbors belong to category *0*.

Nearest neighbor analysis can also be used to compute values for a continuous target. In this situation, the average or median target value of the nearest neighbors is used to obtain the predicted value for the new case.

IBM Data WH KNN Model Options - General

On the Model Options - General tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also set options that control how the number of nearest neighbors is calculated, and set options for enhanced performance and accuracy of the model.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Neighbors

Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- **Euclidean**. (default) The distance between two points is computed by joining them with a straight line.
- **Manhattan**. The distance between two points is calculated as the sum of the absolute differences between their co-ordinates.
- **Canberra**. Similar to Manhattan distance, but more sensitive to data points closer to the origin.
- **Maximum**. The distance between two points is calculated as the greatest of their differences along any coordinate dimension.

Number of Nearest Neighbors (k). The number of nearest neighbors for a particular case. Note that using a greater number of neighbors will not necessarily result in a more accurate model.

The choice of *k* controls the balance between the prevention of overfitting (this may be important, particularly for "noisy" data) and resolution (yielding different predictions for similar instances). You will usually have to adjust the value of *k* for each data set, with typical values ranging from 1 to several dozen.

Enhance Performance and Accuracy

Standardize measurements before calculating distance. If selected, this option standardizes the measurements for continuous input fields before calculating the distance values.

Use coresets to increase performance for large datasets. If selected, this option uses core set sampling to speed up the calculation when large data sets are involved.

IBM Data WH KNN Model Options - Scoring Options

On the Model Options - Scoring Options tab, you can set the default value for a scoring option, and assign relative weights to individual classes.

Make Available for Scoring

Include input fields. Specifies whether the input fields are included in scoring by default.

Class Weights

Use this option if you want to change the relative importance of individual classes in building the model.

Note: This option is enabled only if you are using KNN for classification. If you are performing regression (that is, if the target field type is Continuous), the option is disabled.

The default is to assign a value of 1 to all classes, making them equally weighted. By specifying different numerical weights for different class labels, you instruct the algorithm to weight the training sets of particular classes accordingly.

To change a weight, double-click it in the **Weight** column and make the changes you want.

Value. The set of class labels, derived from the possible values of the target field.

Weight. The weighting to be assigned to a particular class. Assigning a higher weight to a class makes the model more sensitive to that class relative to the other classes.

IBM Data WH K-Means

The K-Means node implements the *k*-means algorithm, which provides a method of cluster analysis. You can use this node to cluster a data set into distinct groups.

The algorithm is a distance-based clustering algorithm that relies on a distance metric (function) to measure the similarity between data points. The data points are assigned to the nearest cluster according to the distance metric used.

The algorithm operates by performing several iterations of the same basic process, in which each training instance is assigned to the closest cluster (with respect to the specified distance function, applied to the instance and cluster center). All cluster centers are then recalculated as the mean attribute value vectors of the instances assigned to particular clusters.

IBM Data WH K-Means Field Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the **All** button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM Data WH K-Means Build Options Tab

By setting the build options, you can customize the build of the model for your own purposes.

If you want to build a model with the default options, click **Run**.

Distance measure. This parameter defines the method of measure for the distance between data points. Greater distances indicate greater dissimilarities. Select one of the following options:

- **Euclidean.** The Euclidean measure is the straight-line distance between two data points.
- **Normalized Euclidean.** The Normalized Euclidean measure is similar to the Euclidean measure but it is normalized by the squared standard deviation. Unlike the Euclidean measure, the Normalized Euclidean measure is also scale-invariant.
- **Mahalanobis.** The Mahalanobis measure is a generalized Euclidean measure that takes correlations of input data into account. Like the Normalized Euclidean measure, the Mahalanobis measure is scale-invariant.
- **Manhattan.** The Manhattan measure is the distance between two data points that is calculated as the sum of the absolute differences between their coordinates.
- **Canberra.** The Canberra measure is similar to the Manhattan measure but it is more sensitive to data points that are closer to the origin.
- **Maximum.** The Maximum measure is the distance between two data points that is calculated as the greatest of their differences along any coordinate dimension.

Number of clusters. This parameter defines the number of clusters to be created.

Maximum number of iterations. The algorithm does several iterations of the same process. This parameter defines the number of iterations after which model training stops.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- **All.** All column-related statistics and all value-related statistics are included.

  Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify **None**.
- **Columns.** Column-related statistics are included.
- **None.** Only statistics that are required to score the model are included.

Replicate results. Select this check box if you want to set a random seed to replicate analyses. You can specify an integer, or you can create a pseudo-random integer by clicking **Generate**.

IBM Data WH Naive Bayes

Naive Bayes is a well-known algorithm for classification problems. The model is termed *naïve* because it treats all proposed prediction variables as being independent of one another. Naive Bayes is a fast, scalable algorithm that calculates conditional probabilities for combinations of attributes and the target attribute. From the training data, an independent probability is established. This probability gives the likelihood of each target class, given the occurrence of each value category from each input variable.

Netezza Bayes Net

A Bayesian network is a model that displays variables in a data set and the probabilistic, or conditional, independencies between them. Using the Netezza Bayes Net node, you can build a probability model by combining observed and recorded evidence with common-sense real-world knowledge to establish the likelihood of occurrences by using seemingly unlinked attributes.

Netezza Bayes Net Field Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

For this node, the target field is needed only for scoring, so it is not displayed on this tab. You can set or change the target on a Type node, on the Model Options tab of this node, or on the Settings tab of the model nugget. See the topic “Netezza Bayes Net Nugget - Settings Tab” on page 73 for more information.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the **All** button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

Netezza Bayes Net Build Options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the **Run** button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

Base index. The numeric identifier to be assigned to the first attribute (input field) for easier internal management.

Sample size. The size of the sample to take if the number of attributes is so large that it would cause an unacceptably long processing time.

Display additional information during execution. If this box is checked (default), additional progress information is displayed in a message dialog box.

Netezza Time Series

A **time series** is a sequence of numerical data values, measured at successive (though not necessarily regular) points in time--for example, daily stock prices or weekly sales data. Analyzing such data can be useful, for example, in highlighting behavior such as trends and seasonality (a repeating pattern), and in predicting future behavior from past events.

Netezza Time Series supports the following time series algorithms.

- spectral analysis
- exponential smoothing
- AutoRegressive Integrated Moving Average (ARIMA)
- seasonal trend decomposition

These algorithms break a time series down into a trend and a seasonal component. These components are then analyzed in order to build a model that can be used for prediction.

Spectral analysis is used to identify periodic behavior in time series. For time series composed of multiple underlying periodicities or when a considerable amount of random noise is present in the data, spectral analysis provides the clearest means of identifying periodic components. This method detects the frequencies of periodic behavior by transforming the series from the time domain into a series of the frequency domain.

Exponential smoothing is a method of forecasting that uses weighted values of previous series observations to predict future values. With exponential smoothing, the influence of observations decreases over time in an exponential way. This method forecasts one point at a time, adjusting its forecasts as new data comes in, taking into account addition, trend, and seasonality.

ARIMA models provide more sophisticated methods for modeling trend and seasonal components than do exponential smoothing models. This method involves explicitly specifying autoregressive and moving average orders as well as the degree of differencing.

Note: In practical terms, ARIMA models are most useful if you want to include predictors that may help to explain the behavior of the series being forecast, such as the number of catalogs mailed or the number of hits to a company Web page. Exponential smoothing models describe the behavior of the time series without attempting to explain why it behaves as it does.

Seasonal trend decomposition removes periodic behavior from the time series in order to perform a trend analysis and then selects a basic shape for the trend, such as a quadratic function. These basic shapes have a number of parameters whose values are determined so as to minimize the mean squared error of the residuals (that is, the differences between the fitted and observed values of the time series).

Interpolation of Values in Netezza Time Series

Interpolation is the process of estimating and inserting missing values in time series data.

If the intervals of the time series are regular but some values are simply not present, the missing values can be estimated using linear interpolation. Consider the following series of monthly passenger arrivals at an airport terminal.

Table 8. Monthly arrivals at a passenger terminal	
Month	**Passengers**
3	3,500,000
4	3,900,000
5	-
6	3,400,000
7	4,500,000
8	3,900,000
9	5,800,000
10	6,000,000

In this case, linear interpolation would estimate the missing value for month 5 as 3,650,000 (the mid-point between months 4 and 6).

Irregular intervals are handled differently. Consider the following series of temperature readings.

Table 9. Temperature readings		
Date	**Time**	**Temperature**
2011-07-24	7:00	57
2011-07-24	14:00	75
2011-07-24	21:00	72
2011-07-25	7:15	59
2011-07-25	14:00	77
2011-07-25	20:55	74
2011-07-27	7:00	60
2011-07-27	14:00	78
2011-07-27	22:00	74

Here we have readings taken at three points during three days, but at various times, only some of which are common between days. In addition, only two of the days are consecutive.

This situation can be handled in one of two ways: calculating aggregates, or determining a step size.

Aggregates might be daily aggregates calculated according to a formula based on semantic knowledge of the data. Doing so could result in the following data set.

Table 10. Temperature readings (aggregated)		
Date	**Time**	**Temperature**
2011-07-24	24:00	69
2011-07-25	24:00	71
2011-07-26	24:00	null
2011-07-27	24:00	72

Alternatively, the algorithm can treat the series as a distinct series and determine a suitable step size. In this case, the step size determined by the algorithm might be 8 hours, resulting in the following.

Table 11. Temperature readings with step size calculated		
Date	**Time**	**Temperature**
2011-07-24	6:00	
2011-07-24	14:00	75
2011-07-24	22:00	
2011-07-25	6:00	
2011-07-25	14:00	77
2011-07-25	22:00	
2011-07-26	6:00	
2011-07-26	14:00	
2011-07-26	22:00	
2011-07-27	6:00	
2011-07-27	14:00	78
2011-07-27	22:00	74

Here, only four readings correspond to the original measurements, but with the help of the other known values from the original series, the missing values can again be calculated by interpolation.

Netezza Time Series Field Options

On the Fields tab, you specify roles for the input fields in the source data.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Target. Choose one field as the target for the prediction. This must be a field with a measurement level of Continuous.

(Predictor) Time Points. (required) The input field containing the date or time values for the time series. This must be a field with a measurement level of Continuous or Categorical and a data storage type of Date, Time, Timestamp, or Numeric. The data storage type of the field you specify here also defines the input type for some fields on other tabs of this modeling node.

(Predictor) Time Series IDs (By). A field containing time series IDs; use this if the input contains more than one time series.

Netezza Time Series Build Options

There are two levels of build options:

- Basic - settings for the algorithm choice, interpolation, and time range to be used.
- Advanced - settings for forecasting

This section describes the basic options.

The Build Options tab is where you set all the options for building the model. You can, of course, just click the **Run** button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

Algorithm

These are the settings relating to the time series algorithm to be used.

Algorithm Name. Choose the time series algorithm you want to use. The available algorithms are **Spectral Analysis**, **Exponential Smoothing** (default), **ARIMA**, or **Seasonal Trend Decomposition**. See the topic "Netezza Time Series" on page 63 for more information.

Trend. (Exponential Smoothing only) Simple exponential smoothing does not perform well if the time series exhibits a trend. Use this field to specify the trend, if any, so that the algorithm can take account of it.

- **System Determined.** (default) The system attempts to find the optimal value for this parameter.
- **None(N).** The time series does not exhibit a trend.
- **Additive(A).** A trend that steadily increases over time.
- **Damped Additive(DA).** An additive trend that eventually disappears.
- **Multiplicative(M).** A trend that increases over time, typically more rapidly than a steady additive trend.
- **Damped Multiplicative(DM).** A multiplicative trend that eventually disappears.

Seasonality. (Exponential Smoothing only) Use this field to specify whether the time series exhibits any seasonal patterns in the data.

- **System Determined.** (default) The system attempts to find the optimal value for this parameter.
- **None(N).** The time series does not exhibit seasonal patterns.
- **Additive(A).** The pattern of seasonal fluctuations exhibits a steady upward trend over time.
- **Multiplicative(M).** Same as additive seasonality, but in addition the amplitude (the distance between the high and low points) of the seasonal fluctuations increases relative to the overall upward trend of the fluctuations.

Use system determined settings for ARIMA. (ARIMA only) Choose this option if you want the system to determine the settings for the ARIMA algorithm.

Specify. (ARIMA only) Choose this option and click the button to specify the ARIMA settings manually.

Interpolation

If the time series source data has missing values, choose a method for inserting estimated values to fill the gaps in the data. See the topic "Interpolation of Values in Netezza Time Series" on page 63 for more information.

- **Linear.** Choose this method if the intervals of the time series are regular but some values are simply not present.
- **Exponential Splines.** Fits a smooth curve where the known data point values increase or decrease at a high rate.
- **Cubic Splines.** Fits a smooth curve to the known data points to estimate the missing values.

Time Range

Here you can choose whether to use the full range of data in the time series, or a contiguous subset of that data, to create the model. Valid input for these fields is defined by the data storage type of the field specified for Time Points on the Fields tab. See the topic "Netezza Time Series Field Options" on page 65 for more information.

- **Use earliest and latest times available in data.** Choose this option if you want to use the full range of the time series data.
- **Specify time window.** Choose this option if you want to use only a portion of the time series. Use the **Earliest time (from)** and **Latest time (to)** fields to specify the boundaries.

ARIMA Structure

Specify the values of the various non-seasonal and seasonal components of the ARIMA model. In each case, set the operator to **=** (equal to) or **<=** (less than or equal to), then specify the value in the adjacent field. Values must be non-negative integers specifying the degrees.

Non seasonal. The values for the various nonseasonal components of the model.

- **Degrees of autocorrelation (p).** The number of autoregressive orders in the model. Autoregressive orders specify which previous values from the series are used to predict current values. For example, an autoregressive order of 2 specifies that the value of the series two time periods in the past be used to predict the current value.
- **Derivation (d).** Specifies the order of differencing applied to the series before estimating models. Differencing is necessary when trends are present (series with trends are typically nonstationary and ARIMA modeling assumes stationarity) and is used to remove their effect. The order of differencing corresponds to the degree of series trend--first-order differencing accounts for linear trends, second-order differencing accounts for quadratic trends, and so on.
- **Moving average (q).** The number of moving average orders in the model. Moving average orders specify how deviations from the series mean for previous values are used to predict current values. For example, moving-average orders of 1 and 2 specify that deviations from the mean value of the series from each of the last two time periods be considered when predicting current values of the series.

Seasonal. Seasonal autocorrelation (SP), derivation (SD), and moving average (SQ) components play the same roles as their nonseasonal counterparts. For seasonal orders, however, current series values are affected by previous series values separated by one or more seasonal periods. For example, for monthly data (seasonal period of 12), a seasonal order of 1 means that the current series value is affected by the series value 12 periods prior to the current one. A seasonal order of 1, for monthly data, is then the same as specifying a nonseasonal order of 12.

The seasonal settings are considered only if seasonality is detected in the data, or if you specify Period settings on the Advanced tab.

Netezza Time Series Build Options - Advanced

You can use the advanced settings to specify options for forecasting.

Use system determined settings for model build options. Choose this option if you want the system to determine the advanced settings.

Specify. Choose this option if you want to specify the advanced options manually. (The option is not available if the algorithm is Spectral Analysis.)

- **Period/Units for period.** The period of time after which some characteristic behavior of the time series repeats itself. For example, for a time series of weekly sales figures you would specify 1 for the period and `Weeks` for the units. **Period** must be a non-negative integer; **Units for period** can be one of **Milliseconds**, **Seconds**, **Minutes**, **Hours**, **Days**, **Weeks**, **Quarters**, or **Years**. Do not set **Units for period** if **Period** is not set, or if the time type is not numeric. However, if you specify **Period**, you must also specify **Units for period**.

Settings for forecasting. You can choose to make forecasts up to a particular point in time, or at specific time points. Valid input for these fields is defined by the data storage type of the field specified for Time Points on the Fields tab. See the topic "Netezza Time Series Field Options" on page 65 for more information.

- **Forecast horizon.** Choose this option if you want to specify only an end point for forecasting. Forecasts will be made up to this point in time.
- **Forecast times.** Choose this option to specify one or more points in time at which to make forecasts. Click **Add** to add a new row to the table of time points. To delete a row, select the row and click **Delete**.

Netezza Time Series Model Options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also set default values for the model output options.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Make Available for Scoring. You can set the default values here for the scoring options that appear on the dialog for the model nugget.

- **Include historical values in outcome.** By default, the model output does not include the historical data values (the ones used to make the prediction). Select this check box to include these values.
- **Include interpolated values in outcome.** If you choose to include historical values in the output, select this box if you also want to include the interpolated values, if any. Note that interpolation works only on historical data, so this box is unavailable if **Include historical values in outcome** is not selected. See the topic "Interpolation of Values in Netezza Time Series" on page 63 for more information.

IBM Data WH TwoStep

The TwoStep node implements the TwoStep algorithm that provides a method to cluster data over large data sets.

You can use this node to cluster data while available resources, for example, memory and time constraints, are considered.

The TwoStep algorithm is a database-mining algorithm that clusters data in the following way:

1. A clustering feature (CF) tree is created. This high-balanced tree stores clustering features for hierarchical clustering where similar input records become part of the same tree nodes.
2. The leaves of the CF tree are clustered hierarchically in-memory to generate the final clustering result. The best number of clusters is determined automatically. If you specify a maximum number of clusters, the best number of clusters within the specified limit is determined.
3. The clustering result is refined in a second step where an algorithm that is similar to the K-Means algorithm is applied to the data.

IBM Data WH TwoStep Field Options

By setting the field options, you can specify to use the field role settings that are defined in upstream nodes. You can also make the field assignments manually.

Select an item. Choose this option to use the role settings from an upstream Type node or from the Types tab of an upstream source node. Role settings are, for example, targets and predictors.

Use custom field assignments. Choose this option if you want to assign targets, predictors, and other roles manually.

Fields. Use the arrows to assign items manually from this list to the role fields on the right. The icons indicate the valid measurement levels for each role field.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM Data WH TwoStep Build Options

By setting the build options, you can customize the build of the model for your own purposes.

If you want to build a model with the default options, click **Run**.

Distance measure. This parameter defines the method of measure for the distance between data points. Greater distances indicate greater dissimilarities. The options are:

- **Log-likelihood.** The likelihood measure places a probability distribution on the variables. Continuous variables are assumed to be normally distributed, while categorical variables are assumed to be multinomial. All variables are assumed to be independent.
- **Euclidean.** The Euclidean measure is the straight-line distance between two data points.
- **Normalized Euclidean.** The Normalized Euclidean measure is similar to the Euclidean measure but it is normalized by the squared standard deviation. Unlike the Euclidean measure, the Normalized Euclidean measure is also scale-invariant.

Cluster Number. This parameter defines the number of clusters to be created. The options are:

- **Automatically calculate number of clusters.** The number of clusters is calculated automatically. You can specify the maximum number of clusters in the **Maximum** field.
- **Specify number of clusters.** Specify how many clusters should be created.

Statistics. This parameter defines how many statistics are included in the model. The options are:

- **All.** All column-related statistics and all value-related statistics are included.

  Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify **None**.
- **Columns.** Column-related statistics are included.
- **None.** Only statistics that are required to score the model are included.

Replicate results. Select this check box if you want to set a random seed to replicate analyses. You can specify an integer, or you can create a pseudo-random integer by clicking **Generate**.

IBM Data WH PCA

Principal component analysis (PCA) is a powerful data-reduction technique designed to reduce the complexity of data. PCA finds linear combinations of the input fields that do the best job of capturing the variance in the entire set of fields, where the components are orthogonal to (not correlated with) each other. The goal is to find a small number of derived fields (the principal components) that effectively summarize the information in the original set of input fields.

Note: An error may occur when scoring the model if lowercase field names are used. This is a known Db2 Data Warehouse defect, with the workaround being to rename all the fields to uppercase before scoring.

IBM Data WH PCA Field Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the **All** button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM Data WH PCA Build Options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the **Run** button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

Center data before computing PCA. If checked (default), this option performs data centering (also known as "mean subtraction") before the analysis. Data centering is necessary to ensure that the first principal component describes the direction of maximum variance, otherwise the component might correspond more closely to the mean of the data. You would normally uncheck this option only for performance improvement if the data had already been prepared in this way.

Perform data scaling before computing PCA. This option performs data scaling before the analysis. Doing so can make the analysis less arbitrary when different variables are measured in different units. In its simplest form data scaling can be achieved by dividing each variable by its standard variation.

Use less accurate but faster method to compute PCA. This option causes the algorithm to use a less accurate but faster method (forceEigensolve) of finding the principal components.

Managing IBM Data WH and Netezza Models

IBM Data Warehouse and IBM Netezza Analytics models are added to the canvas and the Models palette in the same way as other IBM SPSS Modeler models, and can be used in much the same way. However, there are a few important differences, given that each IBM Data Warehouse or IBM Netezza Analytics model created in IBM SPSS Modeler actually references a model stored on a database server. Thus for a stream to function correctly, it must connect to the database where the model was created, and the model table must not have been changed by an external process.

Scoring IBM Data Warehouse and IBM Netezza Analytics models

Models are represented on the canvas by a gold model nugget icon. The main purpose of a nugget is for scoring data to generate predictions, or to allow further analysis of the model properties. Scores are added in the form of one or more extra data fields that can be made visible by attaching a Table node to the nugget and running that branch of the stream, as described later in this section. Some nugget dialog boxes, such as those for Decision Tree or Regression Tree, additionally have a Model tab that provides a visual representation of the model.

The extra fields are distinguished by the prefix `$<id>-` added to the name of the target field, where `<id>` depends on the model, and identifies the type of information being added. The different identifiers are described in the topics for each model nugget.

To view the scores, complete the following steps:

1. Attach a Table node to the model nugget.
2. Open the Table node.
3. Click **Run**.
4. Scroll to the right of the table output window to view the extra fields and their scores.

IBM Data WH and Netezza model nugget Server tab

On the Server tab, you can set server options for scoring the model. You can either continue to use a server connection that was specified upstream, or you can move the data to another database that you specify here.

IBM Data Warehouse Server Details. Here you specify the connection details for the database you want to use for the model.

- **Use upstream connection**. (default) Uses the connection details specified in an upstream node, for example the Database source node. This option works only if all upstream nodes are able to use SQL pushback. In this case there is no need to move the data out of the database, as the SQL fully implements all of the upstream nodes.
- **Move data to connection**. Moves the data to the database you specify here. Doing so allows modeling to work if the data is in another IBM Data Warehouse database, or a database from another vendor, or even if the data is in a flat file. In addition, data is moved back to the database specified here if the data has been extracted because a node did not perform SQL pushback. Click the **Edit** button to browse for and select a connection.

CAUTION: IBM Netezza Analytics and IBM Data Warehouse is typically used with very large data sets. Transferring large amounts of data between databases, or out of and back into a database, can be very time-consuming and should be avoided where possible.

Model name. The name of the model. The name is shown for your information only; you cannot change it here.

IBM Data WH Decision Tree Model Nuggets

The Decision Tree model nugget displays the output from the modeling operation, and also enables you to set some options for scoring the model.

When you run a stream containing a Decision Tree model nugget, by default the node adds one new field, the name of which is derived from the target name.

Table 12. Model-scoring field for Decision Tree

Name of Added Field	Meaning
$I-*target_name*	Predicted value for current record.

If you select the option **Compute probabilities of assigned classes for scoring records** on either the modeling node or the model nugget and run the stream, a further field is added.

Table 13. Model-scoring field for Decision Tree - additional

Name of Added Field	Meaning
$IP-*target_name*	Confidence value (from 0.0 to 1.0) for the prediction.

IBM Data WH Decision Tree Nugget - Model Tab

The **Model** tab shows the Predictor Importance of the decision tree model in graphical format. The length of the bar represents the importance of the predictor.

Note: When you are working with IBM Netezza Analytics Version 2.x or previous, the content of the decision tree model is shown in textual format only.

For these versions, the following information is shown:

- Each line of text corresponds to a node or a leaf.
- The indentation reflects the tree level.
- For a node, the split condition is displayed.
- For a leaf, the assigned class label is shown.

IBM Data WH Decision Tree Nugget - Settings Tab

The Settings tab enables you to set some options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Compute probabilities of assigned classes for scoring records. (Decision Tree and Naive Bayes only) If selected, this option means that the extra modeling fields include a confidence (that is, a probability) field as well as the prediction field. If you clear this check box, only the prediction field is produced.

Use deterministic input data. If selected, this option ensures that any Netezza algorithm that runs multiple passes of the same view will use the same set of data for each pass. If you clear this check box to show that non-deterministic data is being used a temporary table is created to hold the data output for processing, such as that produced by a partition node; this table is deleted after the model is created.

IBM Data WH Decision Tree Nugget - Viewer Tab

The **Viewer** tab shows a tree presentation of the tree model in the same way as the SPSS Modeler does for its decision tree model.

Note: If the model is built with IBM Netezza Analytics Version 2.x or previous, the **Viewer** tab is empty.

IBM Data WH K-Means Model Nugget

K-Means model nuggets contain all of the information captured by the clustering model, as well as information about the training data and the estimation process.

When you run a stream containing a K-Means model nugget, the node adds two new fields containing the cluster membership and distance from the assigned cluster center for that record. The new field with the name $KM-K-Means is for the cluster membership and the new field with the name $KMD-K-Means is for the distance from the cluster center.

IBM Data WH K-Means Nugget - Model Tab

The **Model** tab contains various graphic views that show summary statistics and distributions for fields of clusters. You can export the data from the model, or you can export the view as a graphic.

When you are working with IBM Netezza Analytics Version 2.x or previous, or when you build the model with Mahalanobis as the distance measure, the content of the K-Means models is shown in textual format only.

For these versions, the following information is shown:

- **Summary Statistics.** For both the smallest and the largest cluster, summary statistics shows the number of records. Summary statistics also shows the percentage of the data set that is taken up by these clusters. The list also shows the size ratio of the largest cluster to the smallest.
- **Clustering Summary.** The clustering summary lists the clusters that are created by the algorithm. For each cluster, the table shows the number of records in that cluster, together with the mean distance from the cluster center for those records.

IBM Data WH K-Means Nugget - Settings Tab

The Settings tab enables you to set some options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- **Euclidean**. (default) The distance between two points is computed by joining them with a straight line.
- **Manhattan**. The distance between two points is calculated as the sum of the absolute differences between their co-ordinates.
- **Canberra**. Similar to Manhattan distance, but more sensitive to data points closer to the origin.
- **Maximum**. The distance between two points is calculated as the greatest of their differences along any coordinate dimension.

Netezza Bayes Net Model Nuggets

The Bayes Net model nugget provides a means of setting options for scoring the model.

When you run a stream containing a Bayes Net model nugget, the node adds one new field, the name of which is derived from the target name.

Table 14. Model-scoring field for Bayes Net

Name of Added Field	Meaning
$BN-*target_name*	Predicted value for current record.

You can view the extra field by attaching a Table node to the model nugget and running the Table node.

Netezza Bayes Net Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Target. If you want to score a target field that is different from the current target, choose the new target here.

Record ID. If no Record ID field is specified, choose the field to use here.

Type of prediction. The variation of the prediction algorithm that you want to use:

- **Best (most correlated neighbor).** (default) Uses the most correlated neighbor node.
- **Neighbors (weighted prediction of neighbors).** Uses a weighted prediction of all neighbor nodes.
- **NN-neighbors (non-null neighbors).** Same as the previous option, except that it ignores nodes with null values (that is, nodes corresponding to attributes that have missing values for the instance for which the prediction is calculated).

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

IBM Data WH Naive Bayes Model Nuggets

The Naive Bayes model nugget provides a means of setting options for scoring the model.

When you run a stream containing a Naive Bayes model nugget, by default the node adds one new field, the name of which is derived from the target name.

Table 15. Model-scoring field for Naive Bayes - default

Name of Added Field	Meaning
$I-*target_name*	Predicted value for current record.

If you select the option **Compute probabilities of assigned classes for scoring records** on either the modeling node or the model nugget and run the stream, two further fields are added.

Table 16. Model-scoring fields for Naive Bayes - additional

Name of Added Field	Meaning
$IP-*target_name*	The Bayesian numerator of the class for the instance (that is, the product of the prior class probability and the conditional instance attribute value probabilities).
$ILP-*target_name*	The natural logarithm of the latter.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

IBM Data WH Naive Bayes Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Compute probabilities of assigned classes for scoring records. (Decision Tree and Naive Bayes only) If selected, this option means that the extra modeling fields include a confidence (that is, a probability) field as well as the prediction field. If you clear this check box, only the prediction field is produced.

Improve probability accuracy for small or heavily unbalanced datasets. When computing probabilities, this option invokes the *m*-estimation technique for avoiding zero probabilities during estimation. This kind

of estimation of probabilities may be slower but can give better results for small or heavily unbalanced datasets.

IBM Data WH KNN Model Nuggets

The KNN model nugget provides a means of setting options for scoring the model.

When you run a stream containing a KNN model nugget, the node adds one new field, the name of which is derived from the target name.

Table 17. Model-scoring field for KNN

Name of Added Field	Meaning
$KNN-*target_name*	Predicted value for current record.

You can view the extra field by attaching a Table node to the model nugget and running the Table node.

IBM Data WH KNN Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- **Euclidean**. (default) The distance between two points is computed by joining them with a straight line.
- **Manhattan**. The distance between two points is calculated as the sum of the absolute differences between their co-ordinates.
- **Canberra**. Similar to Manhattan distance, but more sensitive to data points closer to the origin.
- **Maximum**. The distance between two points is calculated as the greatest of their differences along any coordinate dimension.

Number of Nearest Neighbors (k). The number of nearest neighbors for a particular case. Note that using a greater number of neighbors will not necessarily result in a more accurate model.

The choice of *k* controls the balance between the prevention of overfitting (this may be important, particularly for "noisy" data) and resolution (yielding different predictions for similar instances). You will usually have to adjust the value of *k* for each data set, with typical values ranging from 1 to several dozen.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Standardize measurements before calculating distance. If selected, this option standardizes the measurements for continuous input fields before calculating the distance values.

Use coresets to increase performance for large datasets. If selected, this option uses core set sampling to speed up the calculation when large data sets are involved.

Netezza Divisive Clustering Model Nuggets

The Divisive Clustering model nugget provides a means of setting options for scoring the model.

When you run a stream containing a Divisive Clustering model nugget, the node adds two new fields, the names of which are derived from the target name.

Table 18. Model-scoring fields for Divisive Clustering

Name of Added Field	Meaning
$DC-*target_name*	Identifier of subcluster to which current record is assigned.

Table 18. Model-scoring fields for Divisive Clustering (continued)	
Name of Added Field	**Meaning**
$DCD-*target_name*	Distance from subcluster center for current record.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

Netezza Divisive Clustering Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- **Euclidean**. (default) The distance between two points is computed by joining them with a straight line.
- **Manhattan**. The distance between two points is calculated as the sum of the absolute differences between their co-ordinates.
- **Canberra**. Similar to Manhattan distance, but more sensitive to data points closer to the origin.
- **Maximum**. The distance between two points is calculated as the greatest of their differences along any coordinate dimension.

Applied hierarchy level. The level of hierarchy that should be applied to the data.

IBM Data WH PCA Model Nuggets

The PCA model nugget provides a means of setting options for scoring the model.

When you run a stream containing a PCA model nugget, by default the node adds one new field, the name of which is derived from the target name.

Table 19. Model-scoring field for PCA	
Name of Added Field	**Meaning**
$F-*target_name*	Predicted value for current record.

If you specify a value greater than 1 in the **Number of principal components ...** field on either the modeling node or the model nugget and run the stream, the node adds a new field for each component. In this case the field names are suffixed by *-n*, where *n* is the number of the component. For example, if your model is named *pca* and contains three components, the new fields would be named *$F-pca-1*, *$F-pca-2*, and *$F-pca-3*.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

Note: An error may occur when scoring the model if lowercase field names are used. This is a known Db2 Data Warehouse defect, with the workaround being to rename all the fields to uppercase before scoring.

IBM Data WH PCA Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Number of principal components to be used in projection. The number of principal components to which you want to reduce the data set. This value must not exceed the number of attributes (input fields).

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Netezza Regression Tree Model Nuggets

The Regression Tree model nugget provides a means of setting options for scoring the model.

When you run a stream containing a Regression Tree model nugget, by default the node adds one new field, the name of which is derived from the target name.

Table 20. Model-scoring field for Regression Tree	
Name of Added Field	**Meaning**
$I-*target_name*	Predicted value for current record.

If you select the option **Compute estimated variance** on either the modeling node or the model nugget and run the stream, a further field is added.

Table 21. Model-scoring field for Regression Tree - additional	
Name of Added Field	**Meaning**
$IV-*target_name*	Estimated variances of the predicted value.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

Netezza Regression Tree Nugget - Model Tab

The **Model** tab shows the Predictor Importance of the regression tree model in graphical format. The length of the bar represents the importance of the predictor.

Note: When you are working with IBM Netezza Analytics Version 2.x or previous, the content of the regression tree model is shown in textual format only.

For these versions, the following information is shown:

- Each line of text corresponds to a node or a leaf.
- The indentation reflects the tree level.
- For a node, the split condition is displayed.
- For a leaf, the assigned class label is shown.

Netezza Regression Tree Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Compute estimated variance. Indicates whether the variances of assigned classes should be included in the output.

Netezza Regression Tree Nugget - Viewer Tab

The **Viewer** tab shows a tree presentation of the tree model in the same way as the SPSS Modeler does for its regression tree model.

Note: If the model is built with IBM Netezza Analytics Version 2.x or previous, the **Viewer** tab is empty.

IBM Data WH Linear Regression Model Nuggets

The Linear Regression model nugget provides a means of setting options for scoring the model.

When you run a stream containing a Linear Regression model nugget, the node adds one new field, the name of which is derived from the target name.

Table 22. Model-scoring field for Linear Regression	
Name of Added Field	**Meaning**
$LR-*target_name*	Predicted value for current record.

IBM Data WH Linear Regression Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Include input fields. If selected, this option passes all the original input fields downstream, appending the extra modeling field or fields to each row of data. If you clear this check box, only the Record ID field and the extra modeling fields are passed on, and so the stream runs more quickly.

Netezza Time Series Model Nugget

The model nugget provides access to the output of the time series modeling operation. The output consists of the following fields.

Table 23. Time Series model output fields	
Field	**Description**
TSID	The identifier of the time series; the contents of the field specified for Time Series IDs on the Fields tab of the modeling node. See the topic "Netezza Time Series Field Options" on page 65 for more information.
TIME	The time period within the current time series.
HISTORY	The historical data values (the ones used to make the prediction). This field is included only if the option **Include historical values in outcome** is selected on the Settings tab of the model nugget.
$TS-INTERPOLATED	The interpolated values, where used. This field is included only if the option **Include interpolated values in outcome** is selected on the Settings tab of the model nugget. Interpolation is an option on the Build Options tab of the modeling node.
$TS-FORECAST	The forecast values for the time series.

To view the model output, attach a Table node (from the Output tab of the node palette) to the model nugget and run the Table node.

Netezza Time Series Nugget - Settings tab

On the Settings tab you can specify options for customizing the model output.

Model Name. The name of the model, as specified on the Model Options tab of the modeling node.

The other options are the same as those on the Modeling Options tab of the modeling node.

IBM Data WH Generalized Linear Model Nugget

The model nugget provides access to the output of the modeling operation.

When you run a stream containing a Generalized Linear model nugget, the node adds a new field, the name of which is derived from the target name.

Table 24. Model-scoring field for Generalized Linear	
Name of Added Field	**Meaning**
$GLM-*target_name*	Predicted value for current record.

The Model tab displays various statistics relating to the model.

The output consists of the following fields.

Table 25. Output fields from Generalized Linear model

Output Field	**Description**
Parameter	The parameters (that is, the predictor variables) used by the model. These are the numerical and nominal columns, as well as the intercept (the constant term in the regression model).
Beta	The correlation coefficient (that is, the linear component of the model).
Std Error	The standard deviation for the beta.
Test	The test statistics used to evaluate the validity of the parameter.
p-value	The probability of an error when assuming that the parameter is significant.
Residuals Summary	
Residual Type	The type of residual of the prediction for which summary values are shown.
RSS	The value of the residual.
df	The degrees of freedom for the residual.
p-value	The probability of an error. A high value indicates a poorly-fitting model; a low value indicates a good fit.

IBM Data WH Generalized Linear Model Nugget - Settings tab

On the Settings tab you can customize the model output.

The option is the same as that shown for Scoring Options on the modeling node. See the topic "IBM Data WH Generalized Linear Model Options - Scoring Options" on page 56 for more information.

IBM Data WH TwoStep Model Nugget

When you run a stream that contains a TwoStep model nugget, the node adds two new fields that contain the cluster membership and distance from the assigned cluster center for that record. The new field with the name $TS-Twostep is for the cluster membership, and the new field with the name $TSP-Twostep is for the distance from the cluster center.

IBM Data WH TwoStep Nugget - Model Tab

The **Model** tab contains various graphic views that show summary statistics and distributions for fields of clusters. You can export the data from the model, or you can export the view as a graphic.

Chapter 6. Database modeling with IBM Db2 for z/OS

IBM SPSS Modeler and IBM Db2 for z/OS

SPSS Modeler supports integration with Db2 for z/OS, which provides the ability to run advanced analytics on Db2 for z/OS servers. You can access these features through the SPSS Modeler graphical user interface and workflow-oriented development environment. This way, you can run the data mining algorithms directly in the Db2 for z/OS environment leveraging IBM Db2 Analytics Accelerator.

SPSS Modeler supports integration of the following algorithms from Db2 for z/OS.

- Decision Trees
- K-Means
- Naive Bayes
- Regression Tree
- TwoStep

Requirements for integration with IBM Db2 for z/OS

The following conditions are prerequisites for conducting in-database modeling by using Db2® for z/OS® and IBM Db2 Analytics Accelerator for z/OS. To ensure that these conditions are met, you might need to consult with your database administrator. For detailed requirements, including supported versions, see the Software Product Compatibility Reports.

- IBM SPSS Modeler running in local mode or against an SPSS Modeler Server installation on Windows or UNIX
- Db2 for z/OS together with Db2 Analytics Accelerator for z/OS
- IBM SPSS Data Access Pack
- On the server running SPSS Modeler Server, one of the following systems:
  - IBM Db2 Data Server Driver for ODBC and CLI
  - Any version of Db2 for Linux®, UNIX, and Windows with an ODBC data source that is configured for Db2 for z/OS
- License for Db2 Connect for System z®
- SQL generation and optimization enabled in SPSS Modeler
- Db2 z/OS in-database mining requires either accelerator-only tables (AOT) or accelerated tables, and INZA support. IDAA INZA was introduced in IDAA 5.1.This means that the Db2 z/OS in-database mining nodes will not work with previous versions of IDAA.

  If you use an IDAA-enabled DSN in Modeler, the only tables that will be displayed in the list of tables returned in the Database source node using that DSN will be AOT or accelerated tables.

Enabling integration with IBM Db2 Analytics Accelerator for z/OS

Enabling integration with Db2 Analytics Accelerator for z/OS consists of the following steps:

- Configuring Db2 for z/OS and Db2 Analytics Accelerator for z/OS
- Creating an ODBC source
- Enabling the integration of IBM Db2 for z/OS in IBM SPSS Modeler
- Enabling SQL generation and optimization in SPSS Modeler
- Enabling IBM SPSS Modeler Server Scoring Adapter for Db2 for z/OS

- Configuring DSN using IBM Db2 Client in IBM SPSS Modeler

Configuring IBM Db2 for z/OS and IBM Analytics Accelerator for z/OS

How to configure Db2 for z/OS and Analytics Accelerator for z/OS is described on the following website:

Db2 Analytics Accelerator for z/OS.

Creating an ODBC Source for IBM Db2 for z/OS and IBM Db2 Analytics Accelerator

For information about how to enable a connection between Db2 for z/OS and IBM Db2 Analytics Accelerator, see the following websites:

- For version 4: Db2 Analytics Accelerator for z/OS 4.1.0
- For version 3: Db2 Analytics Accelerator for z/OS 3.1.0
- Enabling query acceleration with IBM Db2 Analytics Accelerator for ODBC and JDBC applications without modifying the applications
- SQL error from ODBC driver when running a query in Db2 Analytics Accelerator for z/OS

Enabling the integration of IBM Db2 for z/OS in IBM SPSS Modeler

To enable the integration of Db2 for z/OS in SPSS Modeler, perform the following steps:

1. From the SPSS Modeler `config` directory, open the `odbc-db2-accelerator-names.cfg` file.

   If the file does not exists, you must create it.
2. Add the names of all data sources and the names of all accelerators. For example:

   ```
   dsn1, acceleratorname1
   dsn2, acceleratorname2
   ```
3. The default CCSID for accelerator only tables (AOT) is Unicode; to override this, modify the entries by adding encoding strings to the accelerator names. For example:

   ```
   dsn1, acceleratorname1, EBCDIC
   dsn2, acceleratorname2, UNICODE
   ```
4. Save and close the `odbc-db2-accelerator-names.cfg` file, then open the `odbc-db2-custom-properties.cfg` file from the same directory.
5. SPSS Modeler uses SQL to set the IDAA registers. If required, you can override these entries by changing the SQL to the required values. For example:

   ```
   current_query_sql_acc, "SET CURRENT QUERY ACCELERATION = ELIGIBLE"
   current_get_archive_acc, "SET CURRENT GET_ACCEL_ARCHIVE = NO"
   ```
6. By default, SPSS Modeler uses SQL to create temporary tables for a database cache. If required, you can override this by specifying the expected database name. For example:

   ```
   [OSZ]
   table_create_temp_sql_acc, 'CREATE TABLE <table-name> <(table-columns)> IN DATABASE
   NAME_OF_DATABASE_FOR_AOT'
   ```
7. By default, SPSS Modeler considers that SQL queries written in an ODBC source node are non-replayable, meaning that the query is considered to return different results when being executed multiple times. However, in some scenarios, this may prevent Modeler from generating SQL for downstream nodes and can be overridden by changing the relevant value to Y. For example:

   ```
   assume_custom_sql_replayable, Y
   ```
8. From the SPSS Modeler main menu, click **Tools** > **Options** > **Helper Applications**.

9. Click the **IBM Db2 for z/OS** tab.
10. Select **Enable IBM Db2 for z/OS Data Mining Integration** and then click **OK**.

Note: You cannot view IDAA and non-IDAA tables at the same time in Modeler.

Enabling SQL Generation and Optimization

Because of the likelihood of working with very large data sets, for performance reasons you should enable the SQL generation and optimization options in IBM SPSS Modeler.

To configure SPSS Modeler, do the following steps:

1. From the IBM SPSS Modeler menus choose **Tools** > **Stream Properties** > **Options**
2. Click the **Optimization** option in the navigation pane.
3. Confirm that the **Generate SQL** option is enabled. This setting is required for database modeling to function.
4. Select **Optimize SQL Generation and Optimize other execution** (not strictly required but strongly recommended for optimized performance).

Configuring DSN using IBM Db2 Client in IBM SPSS Modeler

If required, to configure a data source name (DSN) using Db2 Client for Db2 in SPSS Modeler, complete the following steps:

1. If not already installed, install Db2 Client on the operating system where Modeler Server is installed.
2. Using the **db2 catalog** command, catalog the database and add a new data source to the `db2cli.ini` file in Db2 Client. Be sure to point to the defined database alias.
3. Configure data access; detailed steps are available in the Modeler documentation.

   For more information, see the topic **Architecture and Hardware Recommendations** > **Data Access** in the *Modeler Server Administration and Performance Guide* (ModelerServerAdminPerformance.pdf).
4. Create a new ODBC data source in `odbc.ini` by referencing the database alias defined in step 2.
5. For Linux or UNIX users:
   a. Ensure that the driver library `libdb2o.so` is used (instead of `libdb2.so`), and make sure `'DriverUnicodeType=1'` is defined for the new data source.
   b. In the IBM SPSS Data Access Pack installation, ensure that the library path of Db2 Client is added to `odbc.sh`.
   c. Ensure that Modeler Server uses an ODBC Driver wrapper library with UTF-16 encoding (this is called `'libspssodbc_datadirect_utf16.so'`).
6. Make sure that the user who connects to Db2 has the necessary privileges to run the following query:

```
SELECT ACCELERATORNAME FROM SYSACCEL.SYSACCELERATORS
```

Building models with IBM Db2 for z/OS

Each of the supported algorithms has a corresponding modeling node. You can access the Db2 for z/OS modeling nodes from the Database Modeling tab on the nodes palette.

Data considerations

Fields in the data source can contain variables of various data types, depending on the modeling node. In SPSS Modeler, data types are known as *measurement levels*. The Fields tab of the modeling node uses icons to indicate the permitted measurement level types for its input and target fields.

Target field. The target field is the field whose value you are trying to predict. Where a target can be specified, only one of the source data fields can be selected as the target field.

Record ID field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. If the source data does not include an ID field, you can create this field by means of a Derive node, as the following procedure shows.

1. Select the source node.
2. From the Field Ops tab on the nodes palette, double-click the Derive node.
3. Open the Derive node by double-clicking its icon on the canvas.
4. In the **Derive field** field, type (for example) `ID`.
5. In the **Formula** field, type `@INDEX` and click **OK**.
6. Connect the Derive node to the rest of the stream.

Handling null values

If the input data contains null values, use of some Db2 for z/OS nodes may result in error messages or long-running streams, so we recommend removing records containing null values. Use the following method.

1. Attach a Select node to the source node.
2. Set the **Mode** option of the Select node to **Discard**.
3. Enter the following in the **Condition** field:

   ```
   @NULL(field1) [or @NULL(field2)[... or @NULL(fieldN]])
   ```

   Be sure to include every input field.
4. Connect the Select node to the rest of the stream.

Model output

It is possible for a stream containing a Db2 for z/OS modeling node to produce slightly different results each time it is run. This is because the order in which the node reads the source data is not always the same, as the data is read into temporary tables before model building. However, the differences produced by this effect are negligible.

General comments

- In SPSS Collaboration and Deployment Services, it is not possible to create scoring configurations using streams containing Db2 for z/OS modeling nodes.
- PMML export or import is not possible for models created by the Db2 for z/OS nodes.

IBM Db2 for z/OS models - Field options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the **All** button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Target. Choose one field as the target for the prediction. For Generalized Linear models, see also the **Trials** field on this screen.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM Db2 for z/OS Models - Server Options

On the Server tab, you specify the Db2 for z/OS system where the model is to be built.

- **Use upstream connection.** (default) Uses the connection details specified in an upstream node, for example the Database source node. *Note:* This option works only if all upstream nodes are able to use SQL pushback. In this case, there is no need to move the data out of the database, as the SQL fully implements all of the upstream nodes.
- **Move data to connection**. Moves the data to the database you specify here. Doing so allows modeling to work if the data is in another IBM database, or a database from another vendor, or even if the data is in a flat file. In addition, data is moved back to the database specified here if the data has been extracted because a node did not perform SQL pushback. Click the **Edit** button to browse for and select a connection.

Note: The ODBC data source name is effectively embedded in each SPSS Modeler stream. If a stream that is created on one host is executed on a different host, the name of the data source must be the same on each host. Alternatively, a different data source can be selected on the Server tab in each source or modeling node.

IBM Db2 for z/OS models - Model options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically.

Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Replace existing if the name has been used. If you select this check box, any existing model of the same name will be overwritten.

IBM Db2 for z/OS Models - K-Means

The K-Means node implements the *k*-means algorithm, which provides a method of cluster analysis. You can use this node to cluster a data set into distinct groups.

The algorithm is a distance-based clustering algorithm that relies on a distance metric (function) to measure the similarity between data points. The data points are assigned to the nearest cluster according to the distance metric used.

The algorithm operates by performing several iterations of the same basic process, in which each training instance is assigned to the closest cluster (with respect to the specified distance function, applied to the instance and cluster center). All cluster centers are then recalculated as the mean attribute value vectors of the instances assigned to particular clusters.

IBM Db2 for z/OS models - K-Means Field options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the **All** button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM Db2 for z/OS Models - K-Means build options

By setting the build options, you can customize the build of the model for your own purposes.

If you want to build a model with the default options, click **Run**.

Distance measure. This parameter defines the method of measure for the distance between data points. Greater distances indicate greater dissimilarities. Select one of the following options:

- **Euclidean.** The Euclidean measure is the straight-line distance between two data points.
- **Normalized Euclidean.** The Normalized Euclidean measure is similar to the Euclidean measure but it is normalized by the squared standard deviation. Unlike the Euclidean measure, the Normalized Euclidean measure is also scale-invariant.

Number of clusters. This parameter defines the number of clusters to be created.

Maximum number of iterations. The algorithm does several iterations of the same process. This parameter defines the number of iterations after which model training stops.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- **All.** All column-related statistics and all value-related statistics are included.

  Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify **None**.
- **Columns.** Column-related statistics are included.
- **None.** Only statistics that are required to score the model are included.

Replicate results. Select this check box if you want to set a random seed to replicate analyses. You can specify an integer, or you can create a pseudo-random integer by clicking **Generate**.

IBM Db2 for z/OS models - Naive Bayes

Naive Bayes is a well-known algorithm for classification problems. The model is termed naïve because it treats all proposed prediction variables as being independent of one another. Naive Bayes is a fast, scalable algorithm that calculates conditional probabilities for combinations of attributes and the target attribute. From the training data, an independent probability is established. This probability gives the likelihood of each target class, given the occurrence of each value category from each input variable.

IBM Db2 for z/OS Models - Decision Trees

A decision tree is a hierarchical structure that represents a classification model. With a decision tree model, you can develop a classification system to predict or classify future observations from a set of training data. The classification takes the form of a tree structure in which the branches represent split points in the classification. The splits break the data down into subgroups recursively until a stopping point is reached. The tree nodes at the stopping points are known as *leaves*. Each leaf assigns a label, known as a *class label*, to the members of its subgroup, or class.

IBM Db2 for z/OS models - Decision Tree field options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

Use custom field assignments. Choose this option if you want to assign targets, predictors and other roles manually on this screen.

Fields. Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

Click the **All** button to select all the fields in the list, or click an individual measurement level button to select all fields with that measurement level.

Target. Choose one field as the target for the prediction.

Record ID. The field that is to be used as the unique record identifier. The values of this field must be unique for each record (for example, customer ID numbers).

Instance Weight. Specifying a field here enables you to use instance weights (a weight per row of input data) instead of, or in addition to, the default, class weights (a weight per category for the target field). The field you specify here must be one that contains a numeric weight for each row of input data.

Predictors (Inputs). Select the input field or fields. This is similar to setting the field role to *Input* in a Type node.

IBM Db2 for z/OS Models - Decision Tree Build Options

The following build options are available for tree growth:

Growth Measure. These options control the way tree growth is measured.

- **Impurity Measure.** This measure evaluates the best place to split the tree. It is a measurement of the variability in a subgroup or segment of data. A low impurity measurement indicates a group where most members have similar values for the criterion or target field.

  The supported measurements are **Entropy** and **Gini**. These measurements are based on probabilities of category membership for the branch.
- **Maximum tree depth.** The maximum number of levels to which the tree can grow below the root node, that is, the number of times the sample is split recursively. The default value of this property is 10, and the maximal value that you can set for this property is 62.

  Note: If the viewer in the model nugget shows the textual representation of the model, a maximum of 12 levels of the tree is displayed.

Splitting Criteria. These options control when to stop splitting the tree.

- **Minimum improvement for splits.** The minimum amount by which impurity must be reduced before a new split is created in the tree. The goal of tree building is to create subgroups with similar output values to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the amount that is specified by the splitting criteria, the branch is not split.
- **Minimum number of instances for a split.** The minimum number of records that can be split. When fewer than this number of unsplit records remain, no further splits are made. You can use this field to prevent the creation of small subgroups in the tree.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- **All.** All column-related statistics and all value-related statistics are included.

  Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify **None**.
- **Columns.** Column-related statistics are included.
- **None.** Only statistics that are required to score the model are included.

IBM Db2 for z/OS Models - Decision Tree Node - Class Weights

Here you can assign weights to individual classes. The default is to assign a value of 1 to all classes, making them equally weighted. By specifying different numerical weights for different class labels, you instruct the algorithm to weight the training sets of particular classes accordingly.

To change a weight, double-click it in the **Weight** column and make the changes you want.

Value. The set of class labels, derived from the possible values of the target field.

Weight. The weighting to be assigned to a particular class. Assigning a higher weight to a class makes the model more sensitive to that class relative to the other classes.

You can use class weights in combination with instance weights.

IBM Db2 for z/OS Models - Decision Tree Node - Tree Pruning

You can use the pruning options to specify pruning criteria for the decision tree. The intention of pruning is to reduce the risk of overfitting by removing overgrown subgroups that do not improve the expected accuracy on new data.

Pruning measure. The default pruning measure, **Accuracy**, ensures that the estimated accuracy of the model remains within acceptable limits after removing a leaf from the tree. Use the alternative, **Weighted Accuracy**, if you want to take the class weights into account while applying pruning.

Data for pruning. You can use some or all of the training data to estimate the expected accuracy on new data. Alternatively, you can use a separate pruning dataset from a specified table for this purpose.

- **Use all training data.** This option (the default) uses all the training data to estimate the model accuracy.
- **Use % of training data for pruning.** Use this option to split the data into two sets, one for training and one for pruning, using the percentage specified here for the pruning data.
- Select **Replicate results** if you want to specify a random seed to ensure that the data is partitioned in the same way each time you run the stream. You can either specify an integer in the **Seed used for pruning** field, or click **Generate**, which will create a pseudo-random integer.
- **Use data from an existing table.** Specify the table name of a separate pruning data set for estimating model accuracy. Doing so is considered more reliable than using training data.

IBM Db2 for z/OS models - Regression Tree

A regression tree is a tree-based algorithm that splits a sample of cases repeatedly to derive subsets of the same kind, based on values of a numeric target field. As with decision trees, regression trees decompose the data into subsets in which the leaves of the tree correspond to sufficiently small or sufficiently uniform subsets. Splits are selected to decrease the dispersion of target attribute values, so that they can be reasonably well predicted by their mean values at leaves.

IBM Db2 for z/OS Models - Regression Tree Build Options - Tree Growth

You can set build options for tree growth and tree pruning.

The following build options are available for tree growth:

Maximum tree depth. The maximum number of levels to which the tree can grow below the root node, that is, the number of times the sample is split recursively. The default is 62, which is the maximum tree depth for modeling purposes.

Note: If the viewer in the model nugget shows the textual representation of the model, a maximum of 12 levels of the tree is displayed.

Splitting Criteria. These options control when to stop splitting the tree.

- **Split evaluation measure.** This class evaluation measure evaluates the best place to split the tree.

Note: Currently, variance is the only possible option.

- **Minimum improvement for splits.** The minimum amount by which impurity must be reduced before a new split is created in the tree. The goal of tree building is to create subgroups with similar output values to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the amount that is specified by the splitting criteria, the branch is not split.
- **Minimum number of instances for a split.** The minimum number of records that can be split. When fewer than this number of unsplit records remain, no further splits are made. You can use this field to prevent the creation of small subgroups in the tree.

Statistics. This parameter defines how many statistics are included in the model. Select one of the following options:

- **All.** All column-related statistics and all value-related statistics are included.

  Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify **None**.
- **Columns.** Column-related statistics are included.
- **None.** Only statistics that are required to score the model are included.

IBM Db2 for z/OS models - Regression Tree build options - Tree Pruning

You can use the pruning options to specify pruning criteria for the regression tree. The intention of pruning is to reduce the risk of overfitting by removing overgrown subgroups that do not improve the expected accuracy on new data.

Pruning measure. The pruning measure ensures that the estimated accuracy of the model remains within acceptable limits after removing a leaf from the tree. You can select one of the following measures.

- **mse.** Mean squared error - (default) measures how close a fitted line is to the data points.
- **r2.** R-squared - measures the proportion of variation in the dependent variable explained by the regression model.
- **Pearson.** Pearson's correlation coefficient - measures the strength of relationship between linearly dependent variables that are normally distributed.
- **Spearman.** Spearman's correlation coefficient - detects nonlinear relationships that appear weak according to Pearson’s correlation, but which may actually be strong.

Data for pruning. You can use some or all of the training data to estimate the expected accuracy on new data. Alternatively, you can use a separate pruning dataset from a specified table for this purpose.

- **Use all training data.** This option (the default) uses all the training data to estimate the model accuracy.
- **Use % of training data for pruning.** Use this option to split the data into two sets, one for training and one for pruning, using the percentage specified here for the pruning data.

  Select **Replicate results** if you want to specify a random seed to ensure that the data is partitioned in the same way each time you run the stream. You can either specify an integer in the **Seed used for pruning** field, or click **Generate**, which will create a pseudo-random integer.
- **Use data from an existing table.** Specify the table name of a separate pruning dataset for estimating model accuracy. Doing so is considered more reliable than using training data.

IBM Db2 for z/OS models - TwoStep

The TwoStep node implements the TwoStep algorithm that provides a method to cluster data over large data sets.

You can use this node to cluster data while available resources, for example, memory and time constraints, are considered.

The TwoStep algorithm is a database-mining algorithm that clusters data in the following way:

1. A clustering feature (CF) tree is created. This high-balanced tree stores clustering features for hierarchical clustering where similar input records become part of the same tree nodes.
2. The leaves of the CF tree are clustered hierarchically in-memory to generate the final clustering result. The best number of clusters is determined automatically. If you specify a maximum number of clusters, the best number of clusters within the specified limit is determined.
3. The clustering result is refined in a second step where an algorithm that is similar to the K-Means algorithm is applied to the data.

IBM Db2 for z/OS models - TwoStep field options

By setting the field options, you can specify to use the field role settings that are defined in upstream nodes. You can also make the field assignments manually.

Select an item. Choose this option to use the role settings from an upstream Type node or from the Types tab of an upstream source node. Role settings are, for example, targets and predictors.

Use custom field assignments. Choose this option if you want to assign targets, predictors, and other roles manually.

Fields. Use the arrows to assign items manually from this list to the role fields on the right. The icons indicate the valid measurement levels for each role field.

Record ID. The field that is to be used as the unique record identifier.

Predictors (Inputs). Choose one or more fields as inputs for the prediction.

IBM Db2 for z/OS Models - TwoStep Build Options

By setting the build options, you can customize the build of the model for your own purposes.

If you want to build a model with the default options, click **Run**.

Distance measure. This parameter defines the method of measure for the distance between data points. Greater distances indicate greater dissimilarities. The option is:

- **Log-likelihood.** The likelihood measure places a probability distribution on the variables. Continuous variables are assumed to be normally distributed, while categorical variables are assumed to be multinomial. All variables are assumed to be independent.

Cluster Number. This parameter defines the number of clusters to be created. The options are:

- **Automatically calculate number of clusters.** The number of clusters is calculated automatically. You can specify the maximum number of clusters in the **Maximum** field.
- **Specify number of clusters.** Specify how many clusters should be created.

Statistics. This parameter defines how many statistics are included in the model. The options are:

- **All.** All column-related statistics and all value-related statistics are included.

  Note: This parameter includes the maximum number of statistics and might therefore affect the performance of your system. If you do not want to view the model in graphical format, specify **None**.
- **Columns.** Column-related statistics are included.
- **None.** Only statistics that are required to score the model are included.

Replicate results. Select this check box if you want to set a random seed to replicate analyses. You can specify an integer, or you can create a pseudo-random integer by clicking **Generate**.

IBM Db2 for z/OS Models - TwoStep nugget - Model tab

The **Model** tab contains various graphic views that show summary statistics and distributions for fields of clusters. You can export the data from the model, or you can export the view as a graphic.

Managing IBM Db2 for z/OS Models

Db2 for z/OS models are added to the canvas and the Models palette in the same way as other IBM SPSS Modeler models, and can be used in much the same way.

To score the data directly in Db2 for z/OS, do the following steps:

1. Install SPSS Scoring Adapter in the Db2 for z/OS database where the data is located.
2. Ensure that the stream connects to the Db2 for z/OS database where the data is located.

Scoring IBM Db2 for z/OS Models

Models are represented on the canvas by a gold model nugget icon. The main purpose of a nugget is for scoring data to generate predictions, or to allow further analysis of the model properties. Scores are added in the form of one or more extra data fields that can be made visible by attaching a Table node to the nugget and running that branch of the stream, as described later in this section. Some nugget dialog boxes, such as those for Decision Tree or Regression Tree, additionally have a Model tab that provides a visual representation of the model.

The extra fields are distinguished by the prefix `$<id>-` added to the name of the target field, where `<id>` depends on the model, and identifies the type of information being added. The different identifiers are described in the topics for each model nugget.

To view the scores, complete the following steps:

1. Attach a Table node to the model nugget.
2. Open the Table node.
3. Click **Run**.
4. Scroll to the right of the table output window to view the extra fields and their scores.

Note: The scoring process does not run in the accelerator but in Db2 and consequently requires that the input table for scoring must be physically located in Db2. Therefore, as scoring input, only a Db2-based table or an accelerated table can be used. If the stream uses an accelerator-only table, the following error occurs: `"THE STATEMENT CANNOT BE EXECUTED BY DB2 OR IN THE ACCELERATOR."`

IBM Db2 for z/OS Decision Tree Model Nuggets

The Decision Tree model nugget displays the output from the modeling operation and also enables you to set some options for scoring the model.

When you run a stream that contains a Decision Tree model nugget, the node adds two new fields, the names of which are derived from the target.

Table 26. Model-scoring field for Decision Tree

Name of Added Field	Meaning
$I-*target_name*	Predicted value for current record.
$IP-*target_name*	Confidence value (from 0.0 to 1.0) for the prediction.

Note: Due to limitations in Db2 for z/OS, the column names might be truncated.

IBM Db2 for z/OS Decision Tree Nugget - Model Tab

The **Model** tab shows the Predictor Importance of the decision tree model in graphical format. The length of the bar represents the importance of the predictor.

IBM Db2 for z/OS Decision Tree Nugget - Viewer Tab

The **Viewer** tab shows a tree presentation of the tree model in the same way as the SPSS Modeler does for its decision tree model.

IBM Db2 for z/OS K-Means model nugget

K-Means model nuggets contain all of the information captured by the clustering model, as well as information about the training data and the estimation process.

When you run a stream that contains a K-Means model nugget, the node adds two new fields that contain the cluster membership and distance from the assigned cluster center for that record. The new field names are derived from the model name, prefixed by `$KM-` for the cluster membership and `$KMD-` for the distance from the cluster center. For example, if your model is named `Kmeans`, the new fields would be named `$KM-Kmeans` and `$KMD-Kmeans`.

Note: Due to limitations in Db2 for z/OS, the column names might be truncated.

IBM Db2 for z/OS K-Means nugget - Model tab

The **Model** tab contains various graphic views that show summary statistics and distributions for fields of clusters. You can export the data from the model, or you can export the view as a graphic.

IBM Db2 for z/OS Naive Bayes model nuggets

When you run a stream that contains a Naive Bayes model nugget, the node adds two new fields, the names of which are derived from the target name.

Table 27. Model-scoring field for Naive Bayes

Name of Added Field	Meaning
$I-*target_name*	Predicted value for current record.
$IP-*target_name*	Confidence value (from 0.0 to 1.0) for the prediction.

Note: Due to limitations in Db2 for z/OS, the column names might be truncated.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

IBM Db2 for z/OS Regression Tree model nuggets

When you run a stream that contains a Regression Tree model nugget, the node adds two new fields, the names of which are derived from the target name.

Table 28. Model-scoring field for Regression Tree

Name of Added Field	Meaning
$I-*target_name*	Predicted value for current record.
$IS-*target_name*	Estimated standard deviation of the predicted value.

Note: Due to limitations in Db2 for z/OS, the column names might be truncated.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node.

IBM Db2 for z/OS Regression Tree nugget - Model tab

The **Model** tab shows the Predictor Importance of the regression tree model in graphical format. The length of the bar represents the importance of the predictor.

IBM Db2 for z/OS Regression Tree nugget - Viewer tab

The **Viewer** tab shows a tree presentation of the tree model in the same way as the SPSS Modeler does for its regression tree model.

IBM Db2 for z/OS TwoStep model nugget

When you run a stream that contains a TwoStep model nugget, the node adds two new fields that contain the cluster membership and distance from the assigned cluster center for that record. The new field names are derived from the model name, prefixed by `$TS-` for the cluster membership and `$TSD-` for the distance from the cluster center. For example, if your model is named `MDL`, the new fields would be named `$TS-MDL` and `$TSD-MDL`.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Index

L

M

N

O

P

R

S

T

U

Z