IBM SPSS Modeler 14.2 In-Database Mining Guide

IBM®

Preface

IBM® SPSS® Modeler is the IBM Corp. 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 Corp. products or for installation help for one of the supported hardware environments. To reach Technical Support, see the IBM Corp. web site at *http://www.ibm.com/support*. Be prepared to identify yourself, your organization, and your support agreement when requesting assistance.

Contents

4   Database Modeling with Oracle Data Mining   49

5 Database Modeling with IBM InfoSphere Warehouse   98

6 Database Modeling with IBM Netezza Analytics 149

Appendix

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, SPSS Modeler supports the entire data mining process, from data to better business results.

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 in combination with SPSS Modeler Server. A number of additional options are also available, as summarized in the following sections. For more information, see *http://www.ibm.com/software/analytics/spss/products/modeler/*.

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. Additional products or updates beyond those listed here may also be available. For more information, see *http://www.ibm.com/software/analytics/spss/products/modeler/*.

SPSS Modeler. SPSS Modeler is a functionally complete version of the product that is installed and run on the user's desktop computer. It can be run in local mode as a standalone product or in distributed mode along with IBM® SPSS® Modeler Server for improved performance on large data sets.

SPSS Modeler Server. SPSS Modeler Server runs continually in distributed analysis mode together with one or more IBM® SPSS® Modeler installations, providing superior performance on large data sets because memory-intensive operations can be done on the server without downloading data to the client computer. SPSS Modeler Server also provides support for SQL optimization and in-database modeling capabilities, delivering further benefits in performance and automation. At least one SPSS Modeler installation must be present to run an analysis.

IBM SPSS Modeler Options

The following components and features can be separately purchased and licensed for use with SPSS Modeler. Note that additional products or updates may also become available. For more information, see *http://www.ibm.com/software/analytics/spss/products/modeler/*.

■ SPSS Modeler Server access, providing improved scalability and performance on large data sets, as well as support for SQL optimization and in-database modeling capabilities.

- SPSS Modeler Solution Publisher, for real-time or automated scoring outside the SPSS Modeler environment.

- Adapters to enable deployment to IBM SPSS Collaboration and Deployment Services or the thin-client application IBM SPSS Modeler Advantage.

IBM SPSS Text Analytics

IBM® SPSS® Text Analytics is a fully integrated add-on for SPSS Modeler that 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.

- The Text Mining node offers concept and category modeling, as well as an interactive workbench where you can perform advanced exploration of text links and clusters, create your own categories, and refine the linguistic resource templates.

- A number of import formats are supported, including blogs and other web-based sources.

- Custom templates, libraries, and dictionaries for specific domains, such as CRM and genomics, are also included.

Note: A separate license is required to access this component. For more information, see *http://www.ibm.com/software/analytics/spss/products/modeler/*.

IBM SPSS Modeler Documentation

Complete documentation in online help format is available from the Help menu of SPSS Modeler. This includes documentation for SPSS Modeler, SPSS Modeler Server, and SPSS Modeler Solution Publisher, as well as the Applications Guide and other supporting materials.

Complete documentation for each product in PDF format is available under the *\Documentation* folder on each product DVD.

- **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, Predictive Applications, 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 Algorithms Guide.** Descriptions of the mathematical foundations of the modeling methods used in SPSS Modeler.

- **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. For more information, see the topic Application Examples on p. 3.

- **IBM SPSS Modeler Scripting and Automation.** Information on automating the system through scripting, including the properties that can be used to manipulate nodes and streams.

- **IBM SPSS Modeler Deployment Guide.** Information on running SPSS Modeler streams and scenarios as steps in processing jobs under IBM® SPSS® Collaboration and Deployment Services 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 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 and Performance Guide.** Information on how to configure and administer IBM® SPSS® Modeler Server.

- **IBM SPSS Modeler Administration Console User Guide.** Information on installing and using the console user interface for monitoring and configuring SPSS Modeler Server. The console is implemented as a plug-in to the Deployment Manager application.

- **IBM SPSS Modeler Solution Publisher Guide.** SPSS Modeler Solution Publisher is an add-on component that enables organizations to publish streams for use outside of the standard SPSS Modeler environment.

- **IBM SPSS Modeler CRISP-DM Guide.** Step-by-step guide to using the CRISP-DM methodology for data mining with SPSS Modeler.

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 involved should be scalable to real-world applications.

You can access the examples by clicking 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 the topic Demos Folder on p. 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 used with the application examples are installed in the *Demos* folder under the product installation directory. This folder can also be accessed from the IBM SPSS Modeler 14.2 program group on the Windows Start menu, or by clicking *Demos* on the list of recent directories in the File Open dialog box.

Figure 1-1
Selecting the Demos folder from the list of recently-used directories
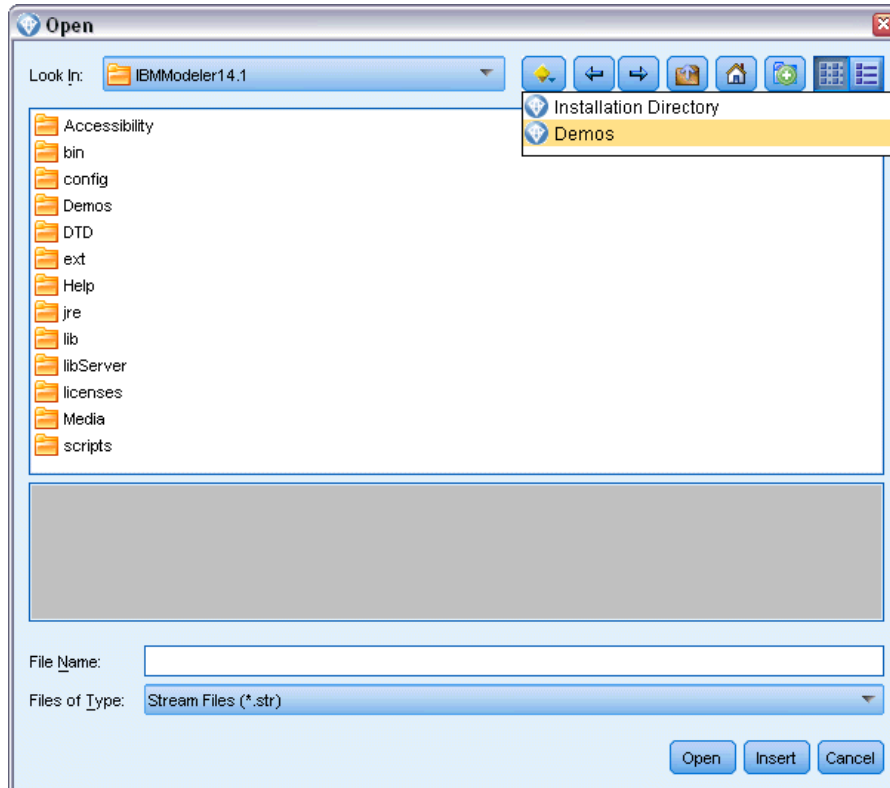
In-Database Mining

Database Modeling Overview

IBM® SPSS® Modeler Server supports integration with data mining and modeling tools that are available from database vendors, including Oracle Data Miner, IBM DB2 InfoSphere Warehouse, 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 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 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 SPSS Modeler.

Using 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 optimization or "SQL pushback". This feature allows you to generate SQL statements for native SPSS Modeler operations that can be "pushed back" to 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 SPSS Modeler.

Note: Database modeling and SQL optimization require that 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 SPSS Modeler, and access SPSS Modeler Server. To verify the current license status, click About on the Help menu in SPSS Modeler.

Figure 2-1
Database Modeling palette
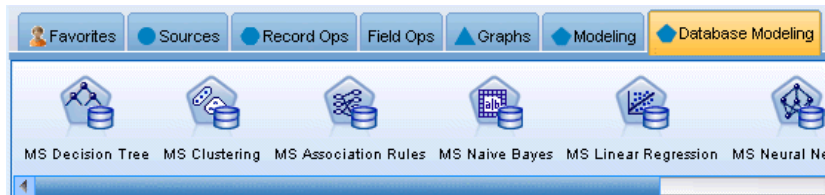
For information on supported algorithms, see the subsequent sections on specific vendors.

Figure 2-2
Viewer providing a graphical view of Microsoft Analysis Services Association Rules model results
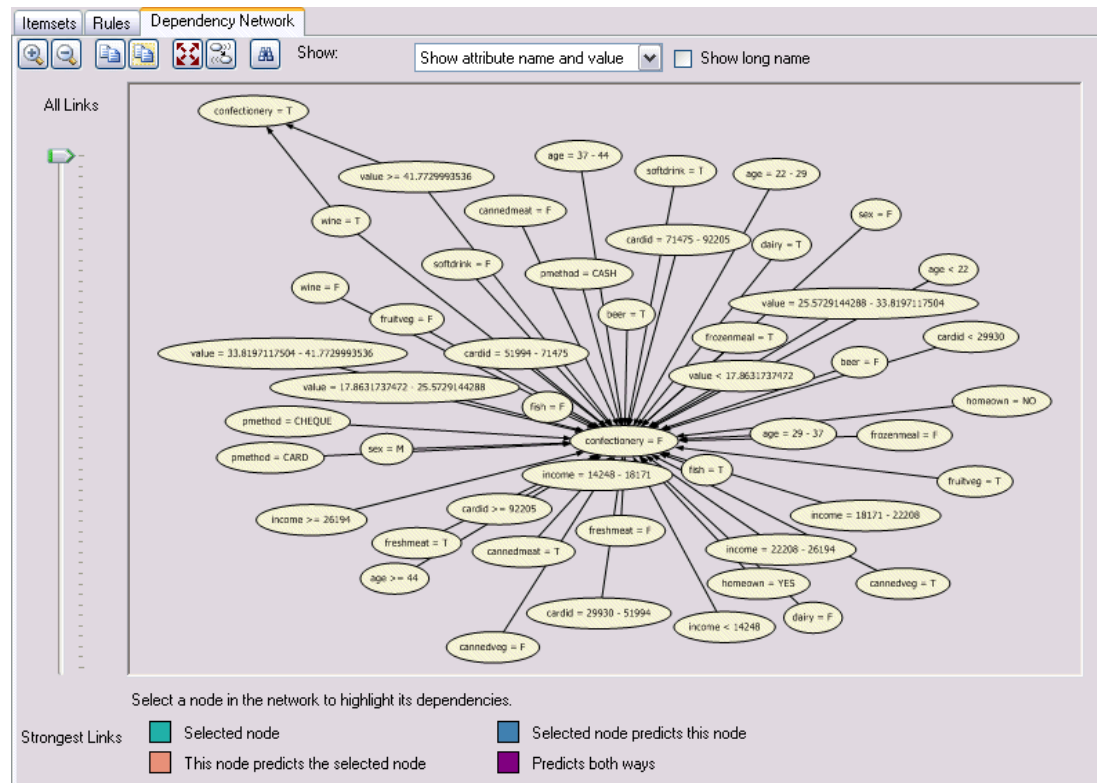
What You Need

To perform database modeling, you need the following setup:

- An ODBC connection to an appropriate database, with required analytical components installed (Microsoft Analysis Services, Oracle Data Miner, or IBM DB2 InfoSphere Warehouse).

- 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 SPSS Modeler Server connectivity be enabled on the SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from SPSS Modeler, and access SPSS Modeler Server. To verify the current license status, click About on the Help menu in SPSS Modeler.

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 SPSS Modeler. The only difference is that the actual processing and model building is pushed back to the database.

For example, the following stream is conceptually identical to other data streams in SPSS Modeler. However, this stream performs all operations in a database, including model-building, using the Microsoft Decision Tree node. When you run the stream, SPSS Modeler instructs the database to build and store the resulting model, and details are downloaded to SPSS Modeler.

Figure 2-3
Database modeling example stream where purple-shaded nodes indicate in-database execution
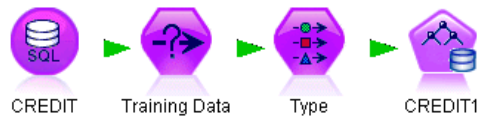
Data Preparation

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 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 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 SPSS Modeler are simply references to these remote models. In other words, the 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.

Figure 2-4
Generated model "nugget" for Microsoft Decision Trees

Once you have created a model, you can add it to the stream for scoring like any other generated model in 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, IBM DB2 InfoSphere Warehouse, 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.

Figure 2-5
Exporting a Microsoft Decision Trees model summary as HTML
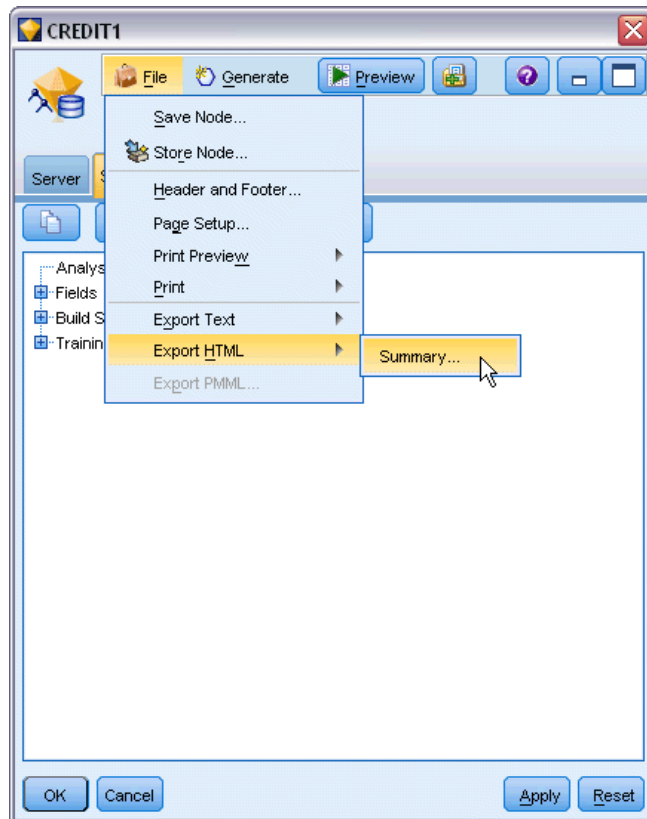
▶ From the File menu in the model browser, choose any of the following options:

■ 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.

Figure 2-6
Generated model key and checking options
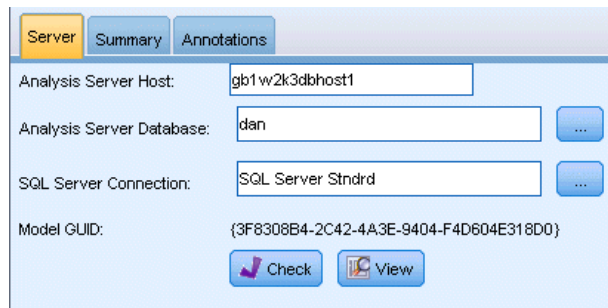
SPSS Modeler uses this randomly generated key to check that the model is still consistent. This key is stored in the description of a model when it is built. It is a good idea to check that keys match before running a deployment stream.

▶ Click the Check button to check the consistency of the model stored in the database by comparing its description with the random key stored by SPSS Modeler. 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.

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 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. For more information, see the topic Enabling Integration with Analysis Services on p. 13.

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 SPSS Modeler streams. The model is then downloaded from Analysis Services to either Microsoft SQL Server or SPSS Modeler for scoring.

10

Figure 3-1
Data flow between IBM SPSS Modeler, Microsoft SQL Server, and Microsoft Analysis Services during model building
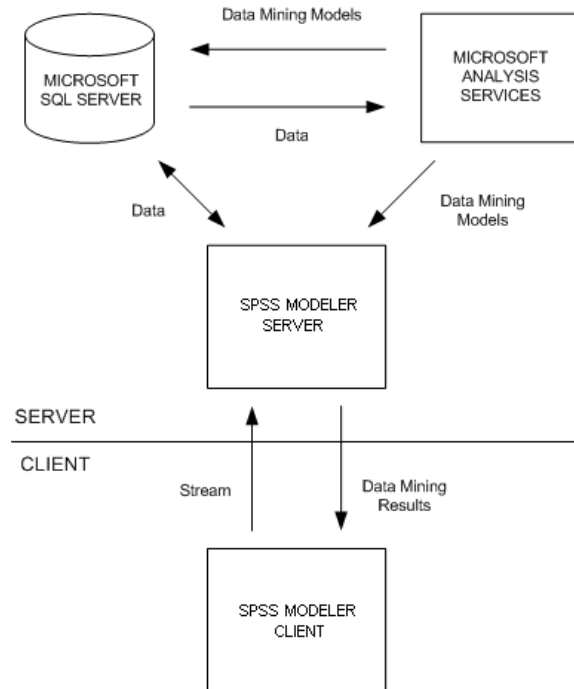
Note: The SPSS Modeler Server is not required, though it can be used. IBM® SPSS® Modeler client is capable of processing in-database mining calculations itself.

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: SPSS Modeler users must configure an ODBC connection using the SQL Native Client driver available from Microsoft at the URL listed below under *Additional SPSS Modeler Server Requirements. The driver provided with the IBM® SPSS® Data Access Pack (and typically recommended for other uses with SPSS Modeler) is not recommended for this purpose.* The driver should be configured to use SQL Server With Integrated Windows Authentication enabled, since 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 2005 or 2008 must be installed, although not necessarily on the same host as SPSS Modeler. 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 SPSS Modeler Server, the following components must be installed on the SPSS Modeler Server host machine.

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

■ Microsoft .NET Framework Version 2.0 Redistributable Package (x86)

■ Microsoft Core XML Services (MSXML) 6.0

■ Microsoft SQL Server 2008 Analysis Services 10.0 OLE DB Provider (be sure to select the correct variant for your operating system)

■ Microsoft SQL Server 2008 Native Client (be sure to select the correct variant for your operating system)

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 SPSS Modeler, the same components must be installed as above, with the addition of the following at the client:

■ Microsoft SQL Server 2008 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 SPSS Modeler Server connectivity be enabled on the SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from SPSS Modeler, and access SPSS Modeler Server. To verify the current license status, click About on the Help menu in SPSS Modeler.

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 SPSS Modeler Helper Applications dialog box, and enable SQL generation and optimization.

Note: Microsoft SQL Server and Microsoft Analysis Services must be available. For more information, see the topic Requirements for Integration with Microsoft Analysis Services on p. 11.

Configuring SQL Server

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

▶ Create the following registry key on the SQL Server host machine:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\MSOLAP

▶ Add the following DWORD value to this key:

AllowInProcess 1

▶ Restart SQL Server after making this change.

Configuring Analysis Services

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

▶ Log in to the Analysis Server through the MS SQL Server Management Studio.

▶ Access the Properties dialog box by right-clicking the server name and choosing Properties.

▶ Select the Show Advanced (All) Properties check box.

▶ 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 SPSS Modeler) is not recommended for this purpose.* If SPSS Modeler and SQL Server reside on different hosts, you can download the Microsoft SQL Native Client ODBC driver. For more information, see the topic Requirements for Integration with Microsoft Analysis Services on p. 11.

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

▶ 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.

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

▶ From the SPSS Modeler menus choose:
Tools > Options > Helper Applications

Figure 3-2
Helper Applications dialog box

▶ Click the Microsoft tab.

■ **Enable Microsoft Analysis Services Integration.** Select to add the Database Modeling tab to the node palettes at the bottom of the SPSS Modeler window. You can use this tab to access Analysis Services related nodes (such as the MS Decision Tree modeling node).

Figure 3-3
Database Modeling tab

- **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 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 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

▶ From the SPSS Modeler menus choose:
Tools > Options > User Options

Figure 3-4
Optimization settings
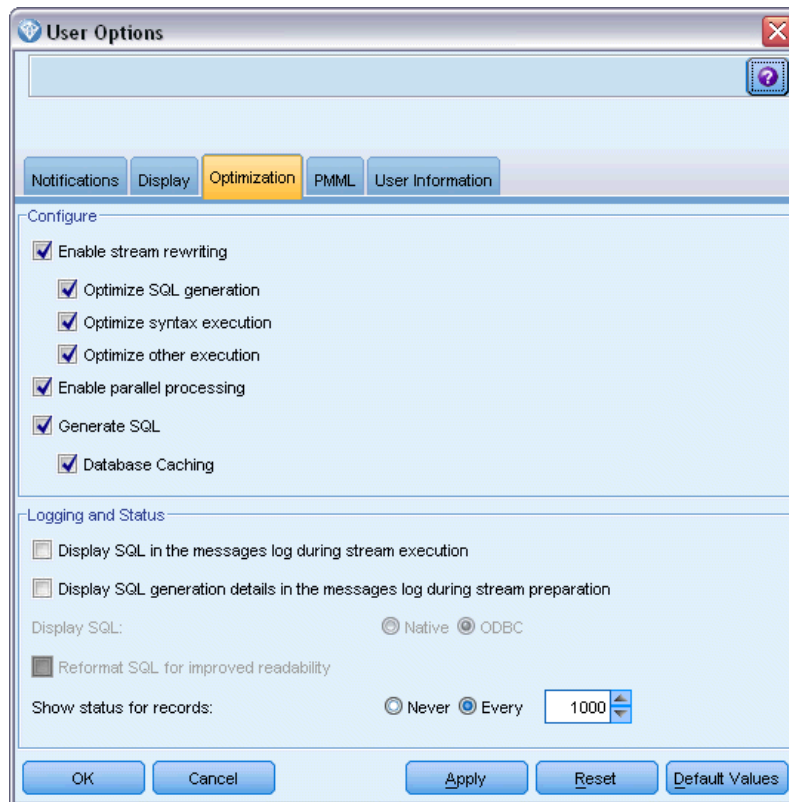
▶ Click the Optimization tab.

▶ Confirm that the Generate SQL option is enabled. This setting is required for database modeling to function.

▶ 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 SPSS Modeler and creates or replaces a model in the SQL Server database. The SPSS Modeler model 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 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. For more information, see the topic MS Time Series Node on p. 27.

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. For more information, see the topic MS Sequence Clustering Node on p. 31.

You can access each node from the Database Modeling palette at the bottom of the 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. For more information, see the topic Enabling Integration with Analysis Services on p. 13.

Figure 3-5
Server options
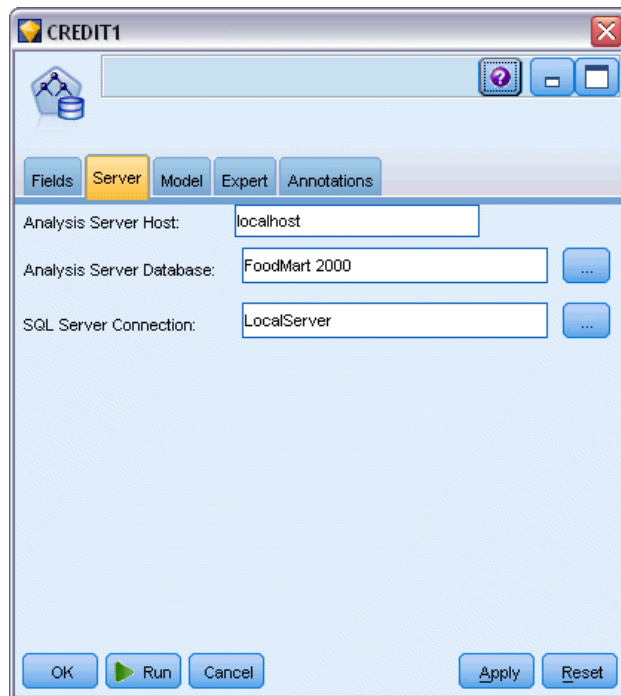
Note: A variation of this tab is also available when scoring Analysis Services models. For more information, see the topic Analysis Services Model Nugget Server Tab on p. 34.

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.

Figure 3-6
Model options
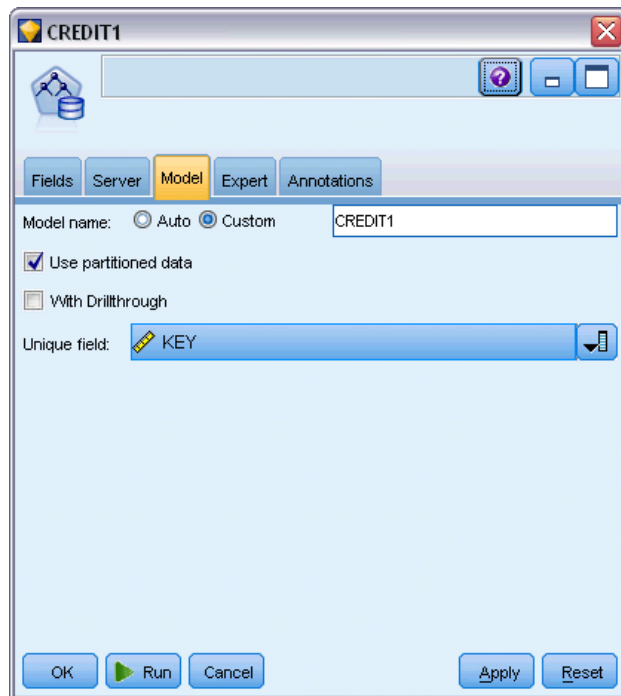
The following basic modeling options are available:

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

Figure 3-7
MS Decision Tree Expert options
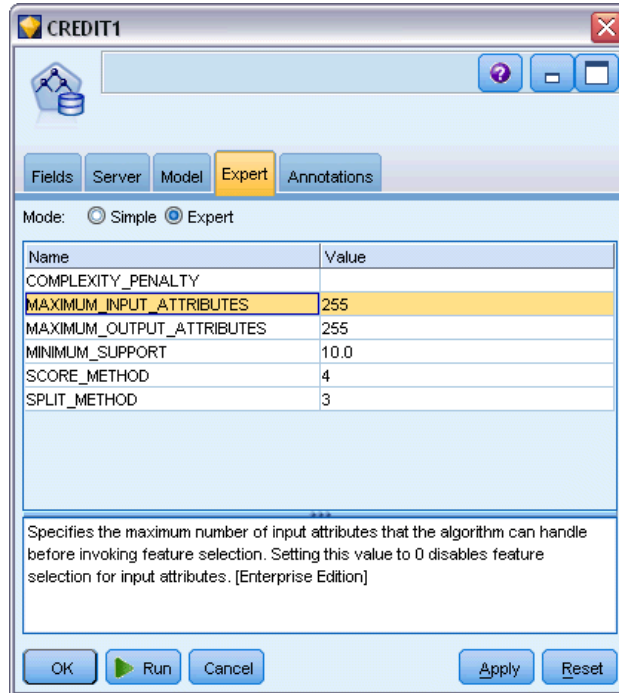
The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

MS Clustering Expert Options

Figure 3-8
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

Figure 3-9
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

Figure 3-10
MS Linear Regression Expert options
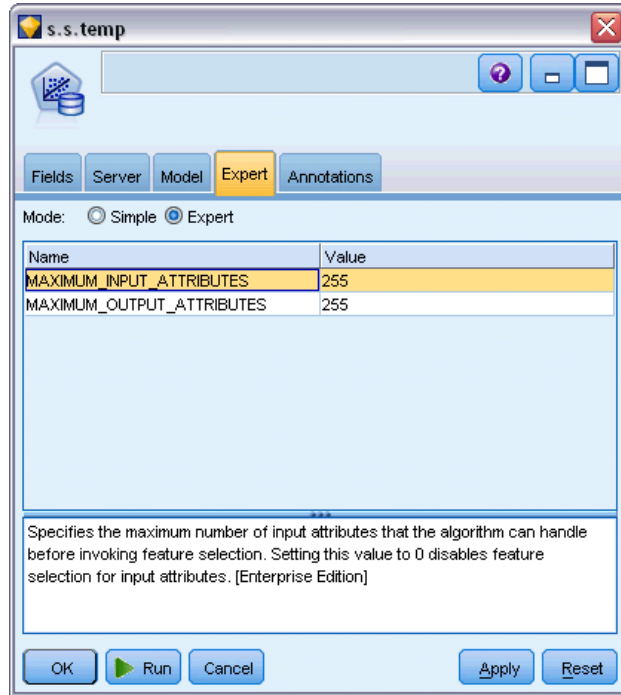
The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

MS Neural Network Expert Options

Figure 3-11
MS Neural Network Expert options
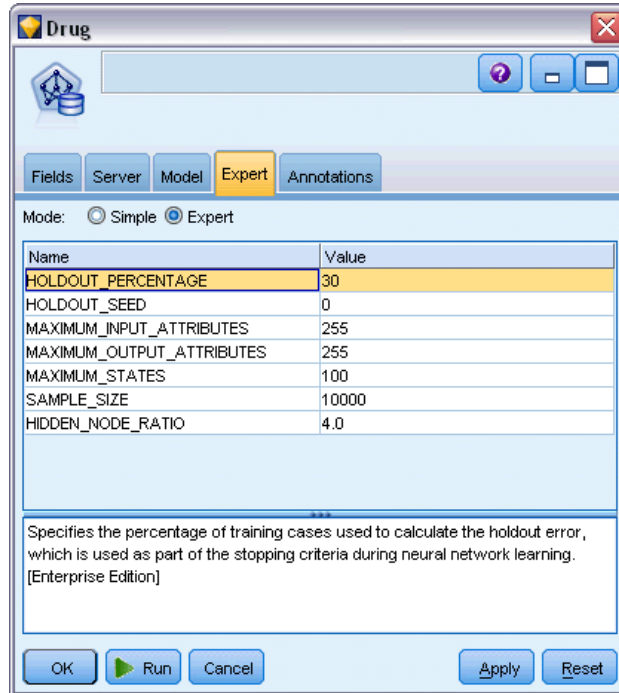
The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

MS Logistic Regression Expert Options

Figure 3-12
MS Logistic Regression Expert options
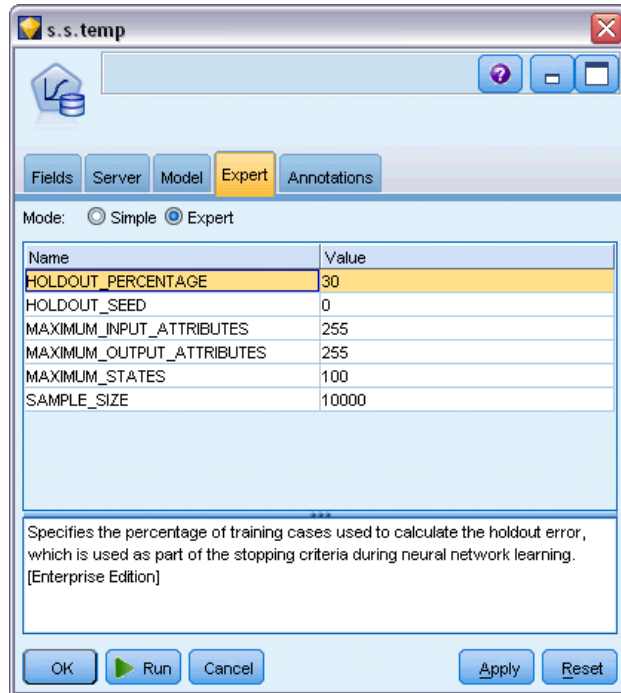
The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

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.
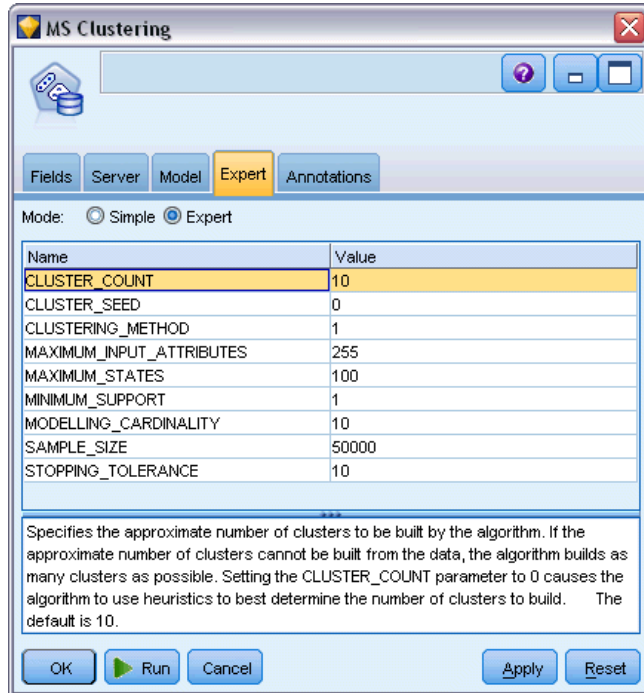
MS Association Rules Expert Options

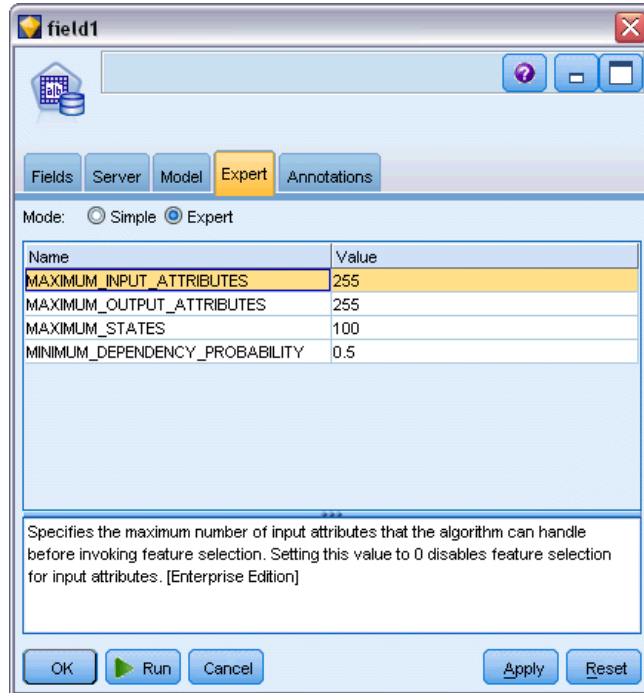Figure 3-13
MS Association Rules Expert options
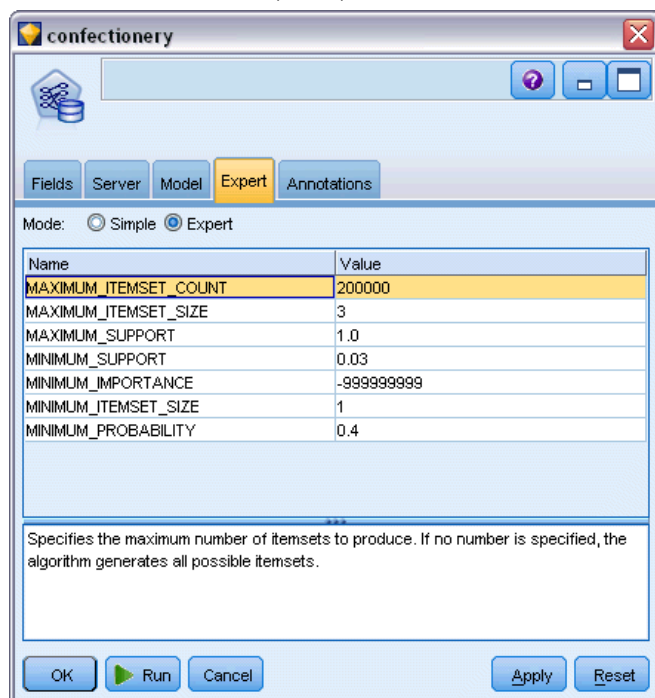
The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

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

Figure 3-14
MS Time Series Model options
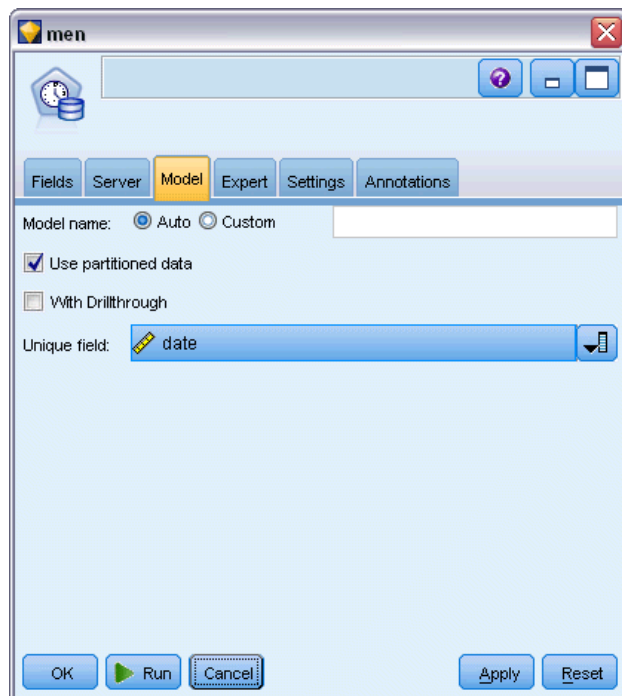
Model name. Specifies the name assigned to the model that is created when the node is executed.

- **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

Figure 3-15
MS Time Series Expert options
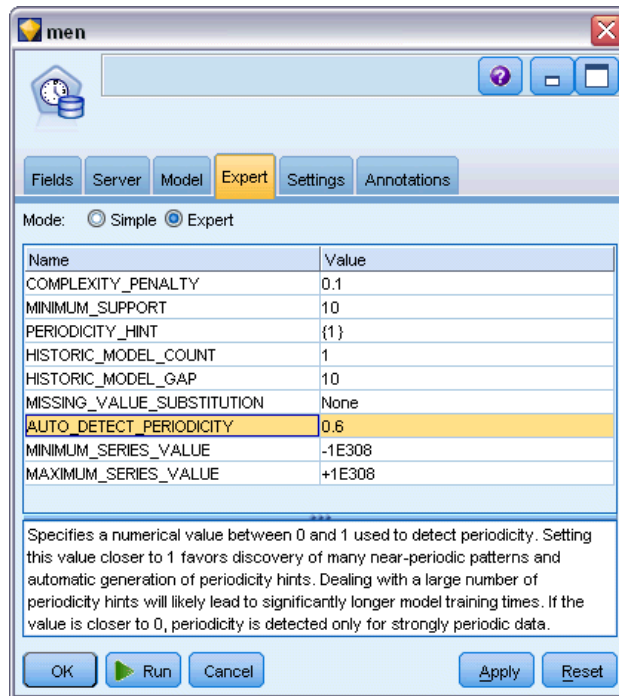
The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

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 p. 40). 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

Figure 3-16
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 (see). 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.

Figure 3-17
Specifying the fields for MS sequence clustering
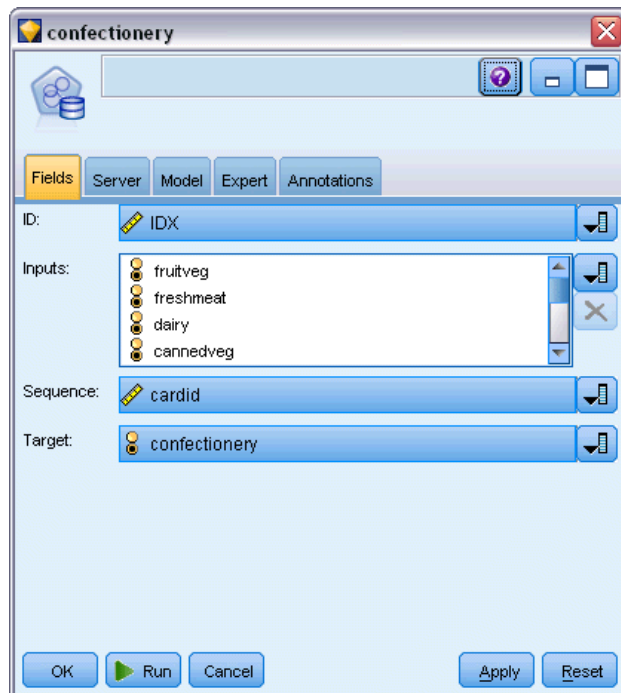
Before you can build a sequence clustering model, you need to specify which fields you want to use as targets and as inputs. Note that for the MS Sequence Clustering node, you cannot use field information from an upstream Type node; you must specify the field settings here.

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

Figure 3-18
Specifying expert options for MS sequence clustering
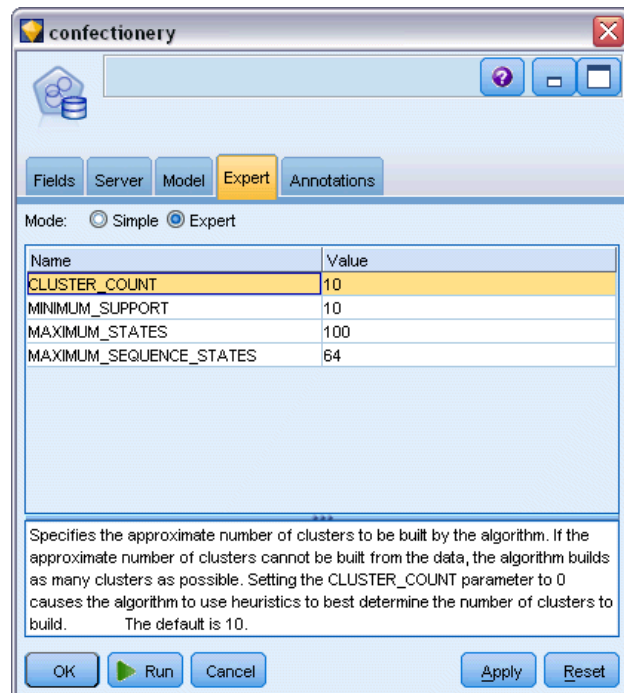
The options available on the Expert tab can fluctuate depending on the structure of the selected stream. Refer to the user interface field-level help for full details regarding expert options for the selected Analysis Services model node.

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 SPSS Modeler. Models that you create from 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 SPSS Modeler, generally only a single prediction and associated probability or confidence is delivered.

For model scoring examples, see Analysis Services Mining Examples on p. 41.

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.

Figure 3-19
Server options for the MS Decision Tree model nugget

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. For more information, see the topic Enabling Integration with Analysis Services on p. 13.

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 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.

Figure 3-20
Results of checking the model keys
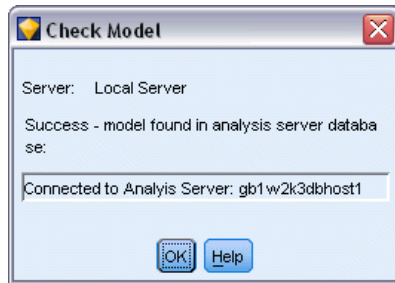
Note: The Check button is available only for models added to the stream canvas in preparation for scoring. If the check fails, investigate whether the model has been deleted or replaced by a different model on the server.

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

Analysis Services Model Nugget Summary Tab

Figure 3-21
Summary options for the MS Decision Tree model nugget

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 fields are 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.

Figure 3-22
*Server options for the MS Time Series model nugget*

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. For more information, see the topic Enabling Integration with Analysis Services on p. 13.

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 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.

Figure 3-23
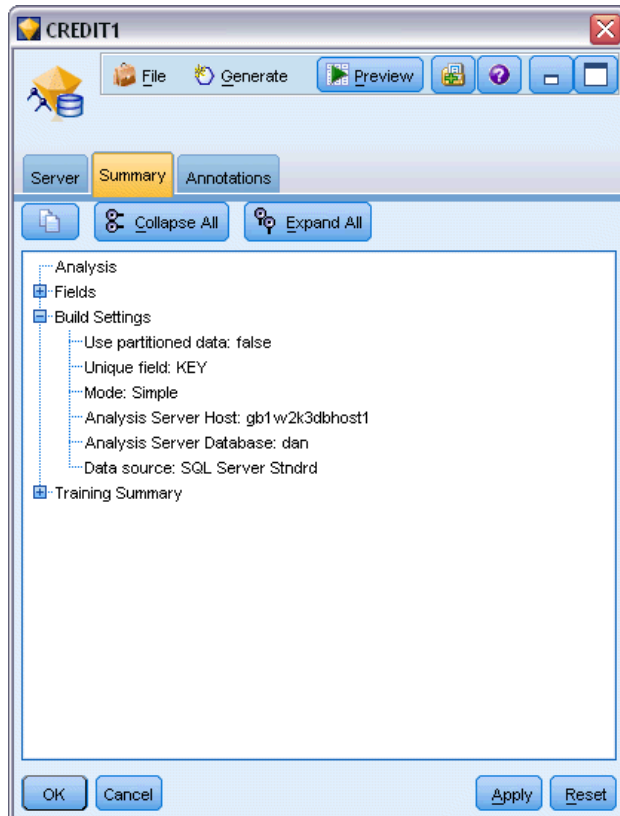*Results of checking the model keys*
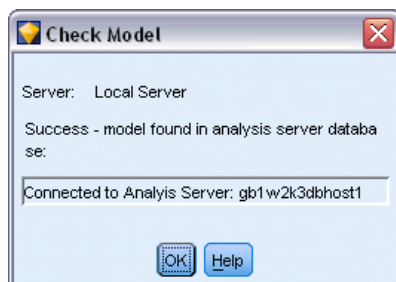
Note: The Check button is available only for models added to the stream canvas in preparation for scoring. If the check fails, investigate whether the model has been deleted or replaced by a different model on the server.

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.

Figure 3-24
MS Time Series viewer with historical (solid line) and predicted future values (dotted line)
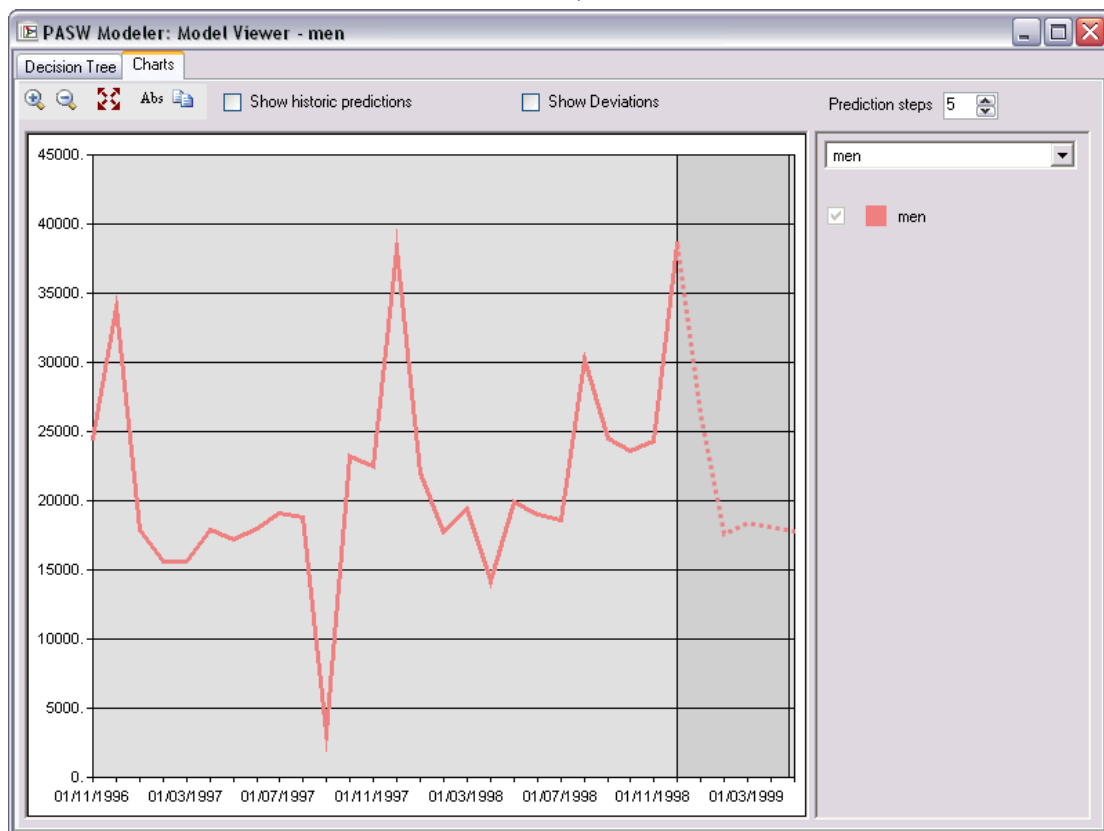
For more information, see the description of the Time Series viewer in the MSDN library at *http://msdn.microsoft.com/en-us/library/ms175331.aspx*.

MS Time Series Model Nugget Settings Tab

Figure 3-25
Settings options for the MS Time Series model nugget

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 fields are added to the MS Sequence Clustering model (where *field* is the name of the target field):

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 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.

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.

Stream	Description
4_evaluate_model.str	Used as an example of model evaluation with 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.

Figure 3-26
Example stream used to upload data
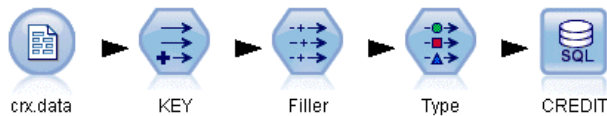
Since Analysis Services data mining requires a key field, this initial stream uses a Derive node to add a new field to the dataset called *KEY* with unique values *1,2,3* using the IBM® SPSS® Modeler@INDEX function.

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.

Figure 3-27
Data Audit results
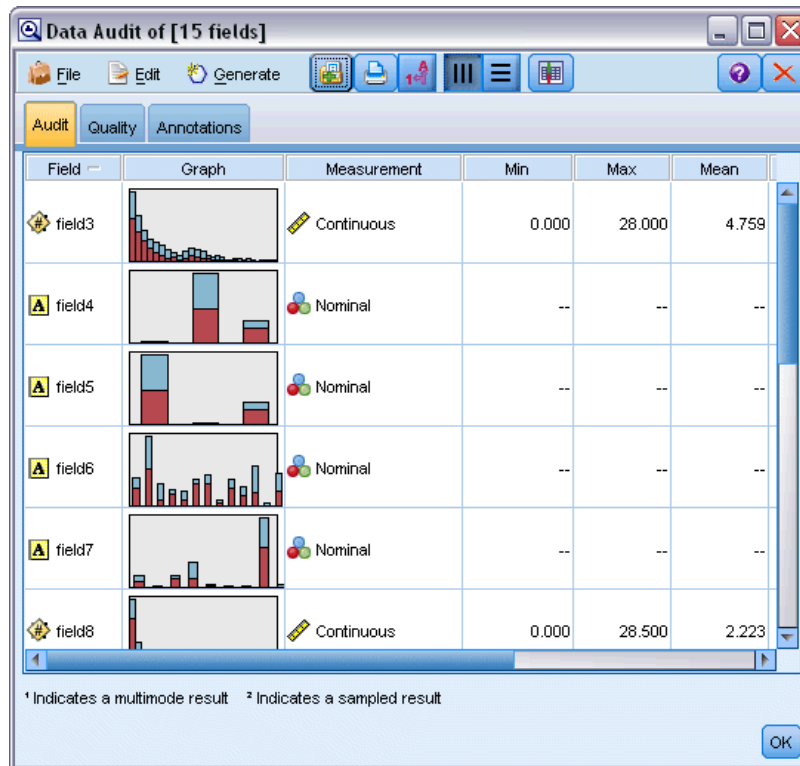
Double-clicking a graph in the Data Audit Report produces a more detailed graph for deeper exploration of a given field.

Figure 3-28
Histogram created by double-clicking a graph in the Data Audit window
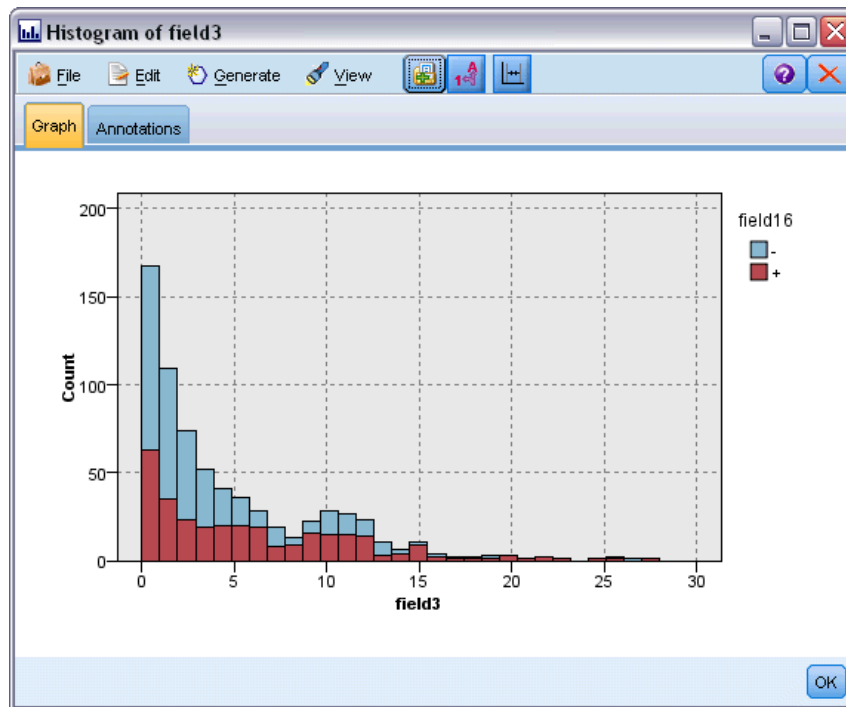
Example Stream: Build Model

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.

Figure 3-29
Database modeling example stream, where purple-shaded nodes indicate in-database execution
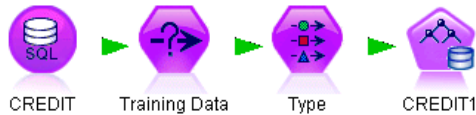
On the Model tab of the dialog box, you can specify the following:

▶ 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 SPSS Modeler.

Figure 3-30
Example stream used for model evaluation
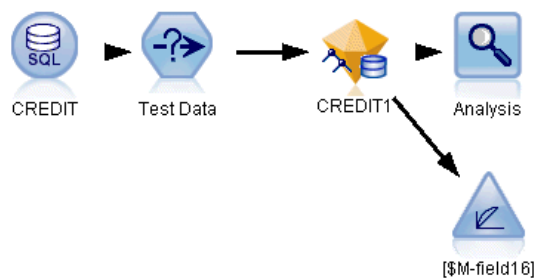
Viewing Modeling Results

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.

Figure 3-31
Viewer providing a graphical view of MS Decision Tree model results
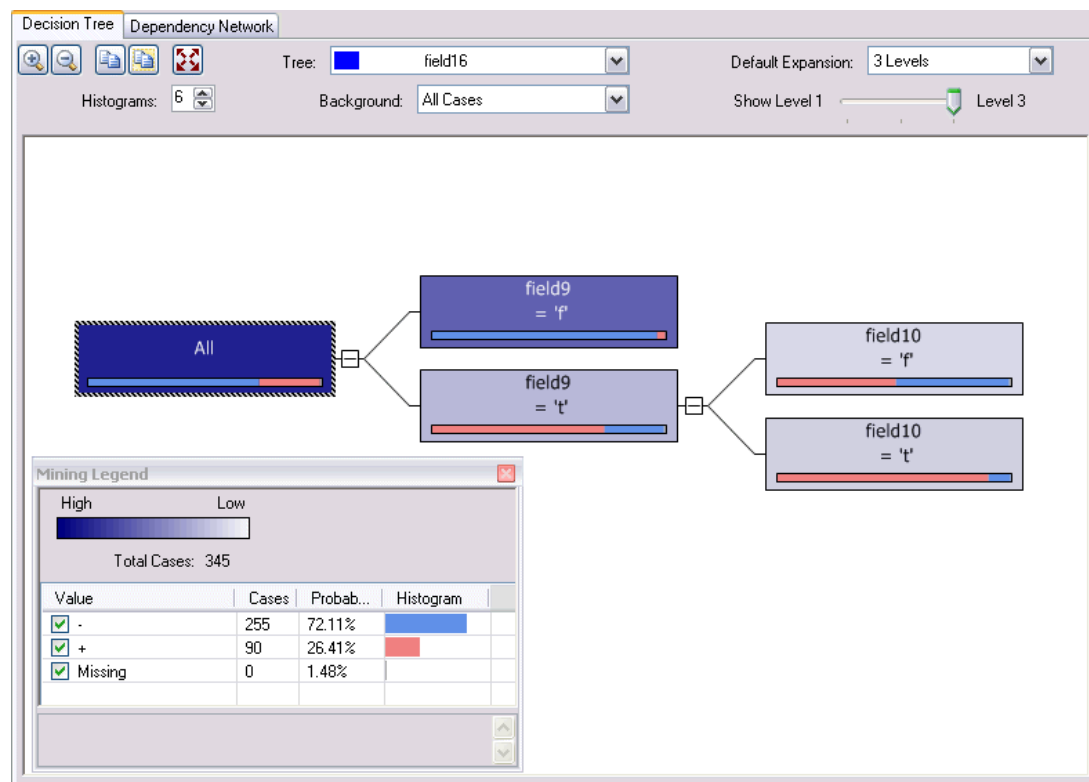
Evaluating Model Results

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.

Figure 3-32
Analysis node results
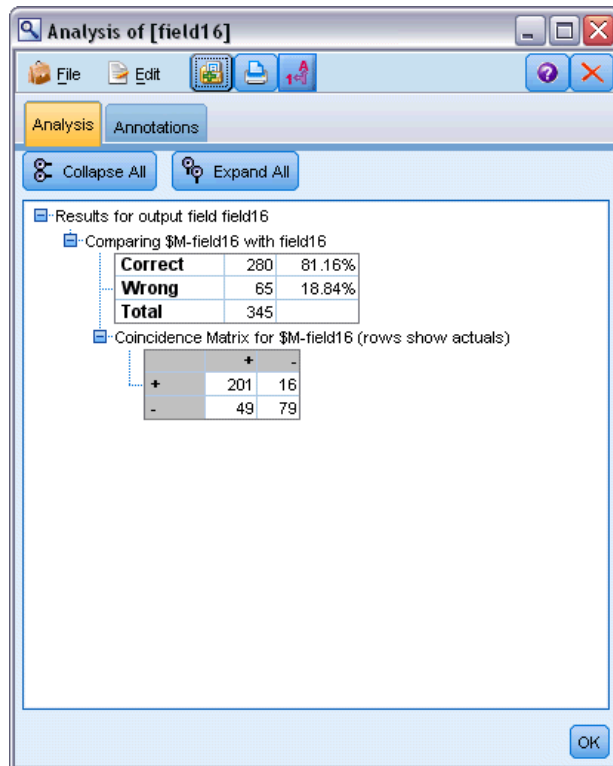
The table indicates that 81.16% of the predictions generated by the MS Decision Tree algorithm were correct.

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.

Figure 3-33
Gains chart generated using the Evaluation node
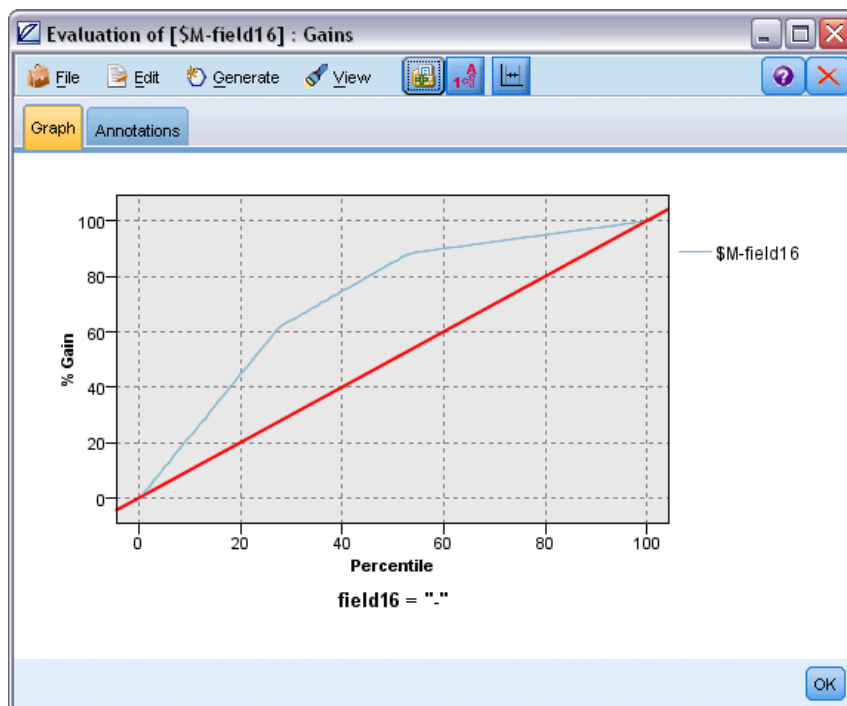
Example Stream: Deploy Model

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.

Figure 3-34
Example stream used to deploy the model
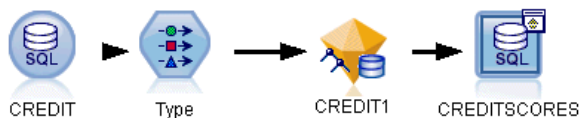
Running the stream generates the following SQL:

```
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
```

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 SPSS Modeler's graphical user interface and workflow-oriented development environment, allowing customers to use the data mining algorithms offered by ODM.

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 SPSS Modeler Server connectivity be enabled on the SPSS Modeler computer. With this setting enabled, you can access database algorithms, push back SQL directly from SPSS Modeler, and access SPSS Modeler Server. To verify the current license status, click About on the Help menu in SPSS Modeler.

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 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 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 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.

▶ 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.

▶ 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, selectData Sources (ODBC), then click Open.

- **Windows 7.** From the Start menu, choose Control Panel, then System & Security, then Administrative Tools. SelectData Sources (ODBC), then click Open.

▶ Click the System DSN tab, and then click Add.

▶ Select the SPSS OEM 6.0 Oracle Wire Protocol driver.

▶ Click Finish.

▶ 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.

▶ Click the Test button to test the connection.

Enabling Oracle Data Mining Integration in IBM SPSS Modeler

▶ From the SPSS Modeler menus choose:
Tools > Options > Helper Applications

▶ Click the Oracle tab.

Figure 4-1
Options for enabling database modeling with Oracle Data Mining
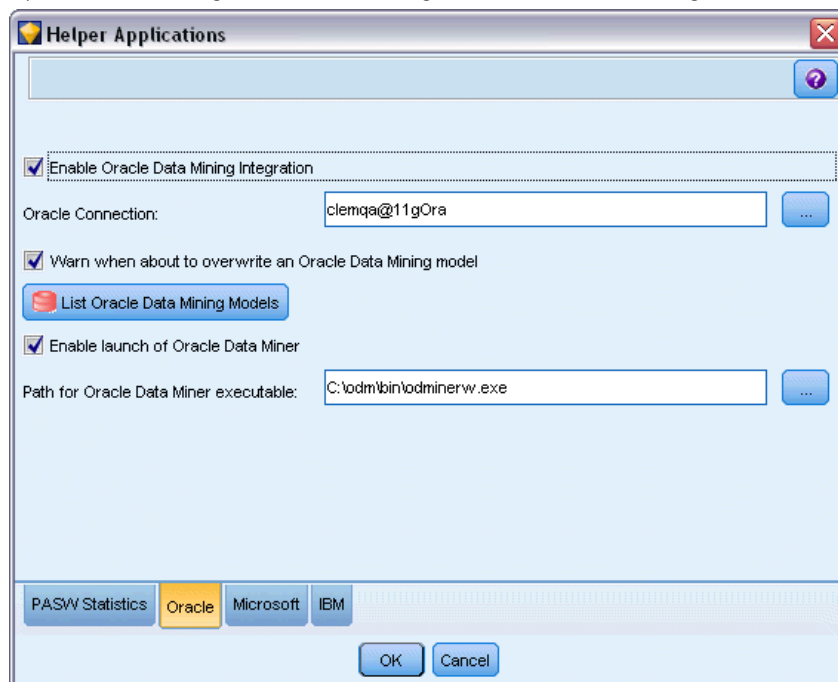
Enable Oracle Data Mining Integration. Enables the Database Modeling palette (if it is not already displayed) and adds modeling nodes for Oracle Naive Bayes, Adaptive Bayes, and Support Vector Machine. (Once enabled, this palette is displayed with others across the bottom of the SPSS Modeler window.)

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 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 SPSS Modeler without warning.

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

Enable launch of Oracle Data Miner. (optional) When enabled, allows SPSS Modeler to launch the Oracle Data Miner application. Refer to Oracle Data Miner on p. 88 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 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

► From the SPSS Modeler menus choose:
Tools > Options > User Options

Figure 4-2
Optimization settings
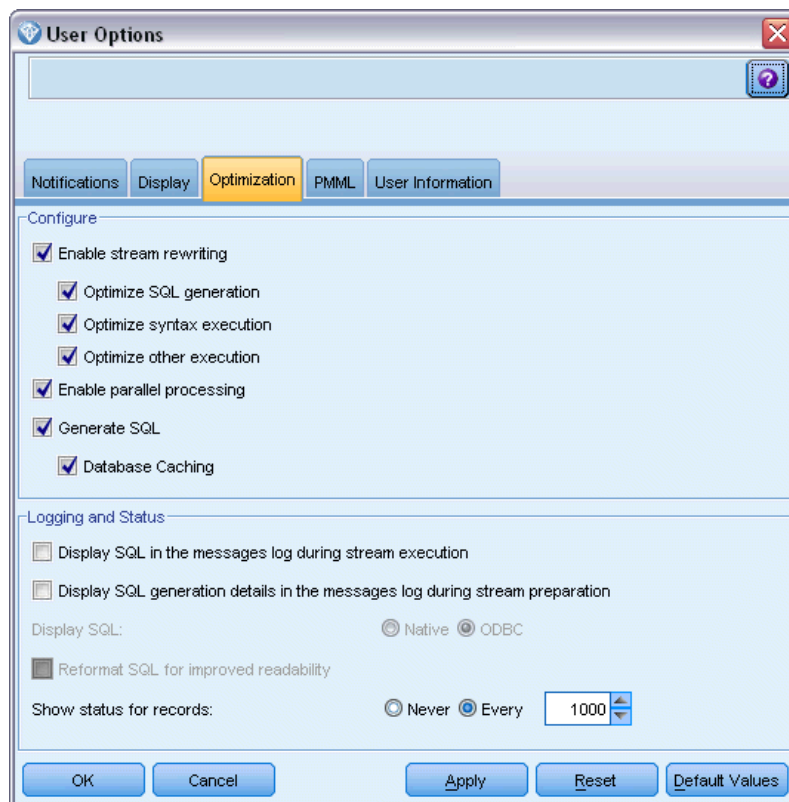
▶ Click the Optimization tab.

▶ Confirm that the Generate SQL option is enabled. This setting is required for database modeling to function.

▶ 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 SPSS Modeler window.

Figure 4-3
Database Modeling palette

Data Considerations

Oracle requires that categorical data be stored in a string format (either CHAR or VARCHAR2). As a result, 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 SPSS Modeler by using the Reclassify node.

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

Unique field. Specifies the field used to uniquely identify each case. For example, this might be an ID field, such as *CustomerID*. 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 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 SPSS Modeler.
- In SPSS Modeler, generally only a single prediction and associated probability or confidence is delivered.
- SPSS Modeler restricts the number of fields that can be used in model building and scoring to 1,000.
- 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. For more information, see the topic Enabling Integration with Oracle on p. 50.

Figure 4-4
Oracle Server options

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 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

Figure 4-5
Oracle costs options

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

Figure 4-6
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

Figure 4-7
Naive Bayes expert options
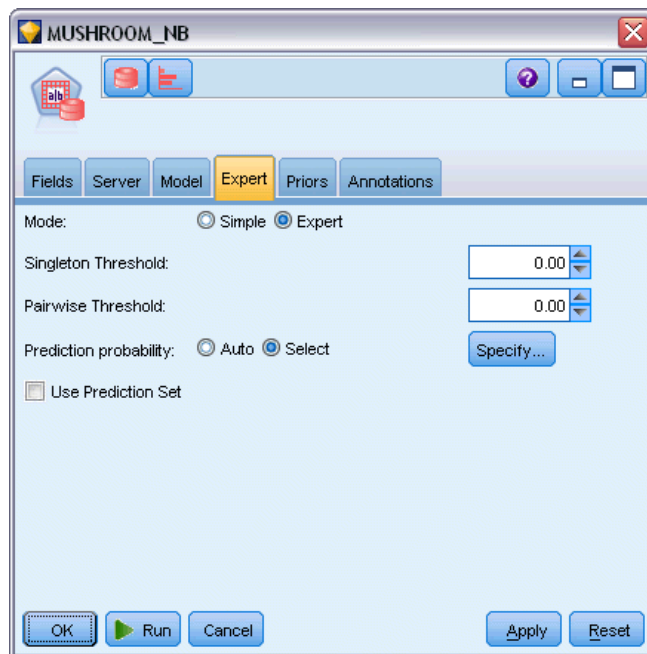
When the model is built, individual predictor attribute values or value pairs are ignored unless there are enough occurrences of a given value or pair in the training data. The thresholds for ignoring values are specified as fractions based on the number of records in the training data. Adjusting these 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 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.

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 SPSS Modeler.

Adaptive Bayes Model Options

Figure 4-8
Adaptive Bayes model options
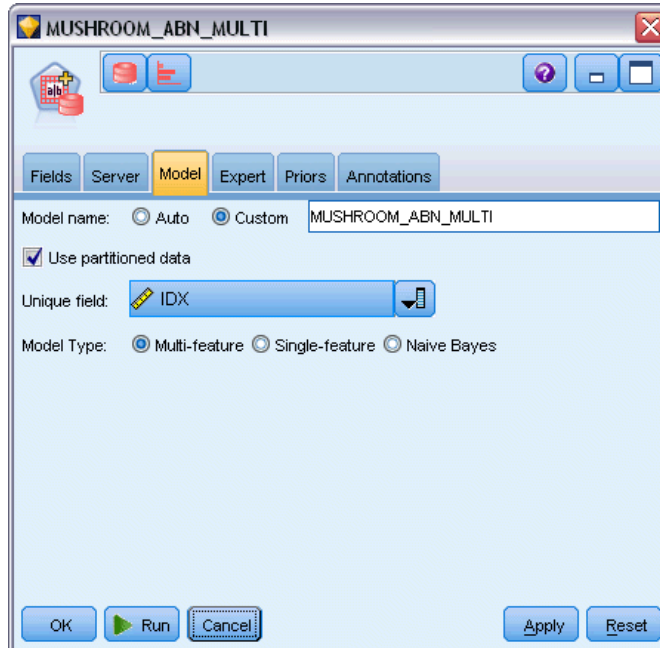
Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

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

Figure 4-9
Adaptive Bayes expert options
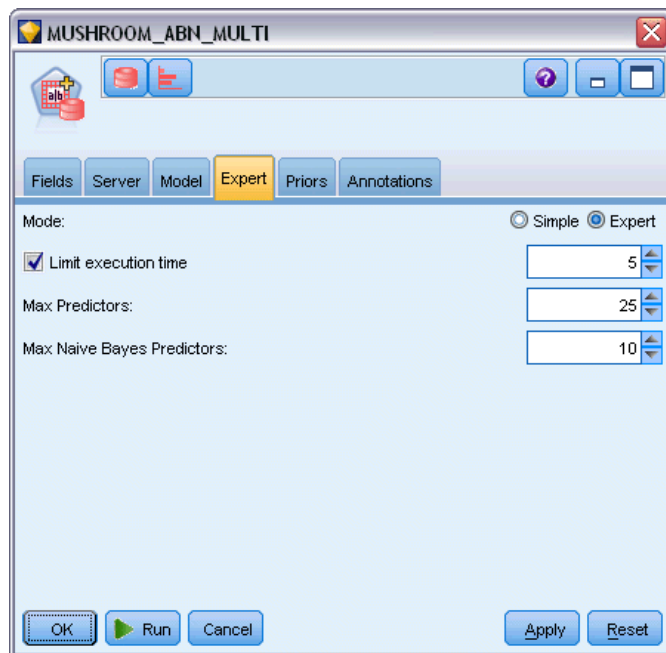
Limit execution time. Select this option to specify a maximum build time in minutes. This makes it possible to produce models in less time, although the resulting model may be less accurate. At each milestone in the modeling process, the algorithm checks whether it will be able to complete the next milestone 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.
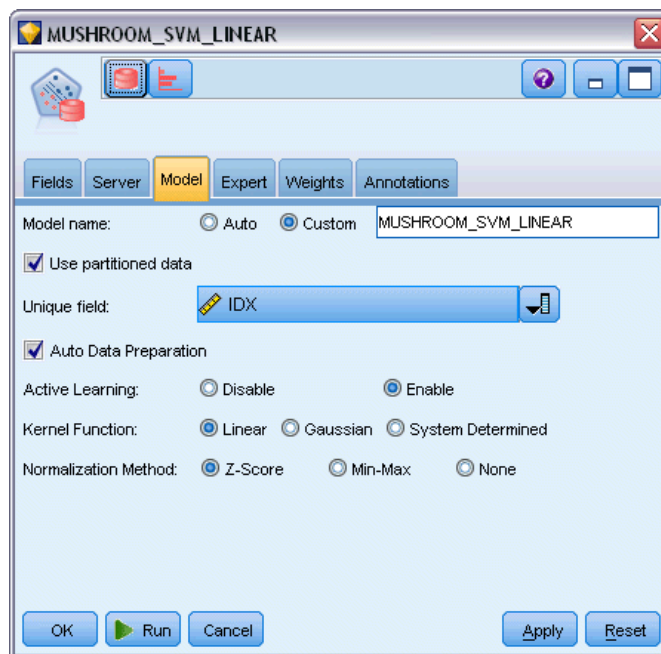
Oracle Support Vector Machine (SVM)

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

Figure 4-10
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 SPSS Modeler. Models built with the linear kernel can be browsed in 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

Figure 4-11
SVM expert options
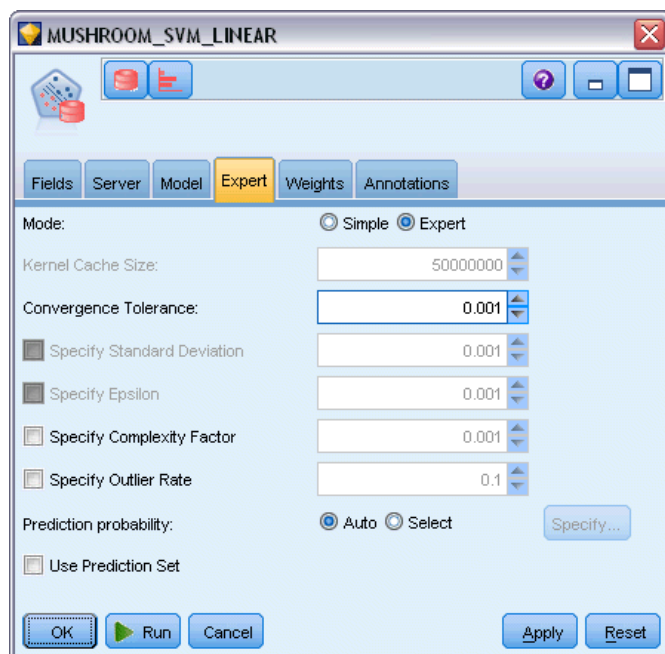
Kernel Cache Size. Specifies, in bytes, the size of the cache to be used for storing computed kernels during the build operation. As might be expected, larger caches typically result in faster builds. The default is 50MB.

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 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

Figure 4-12
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.
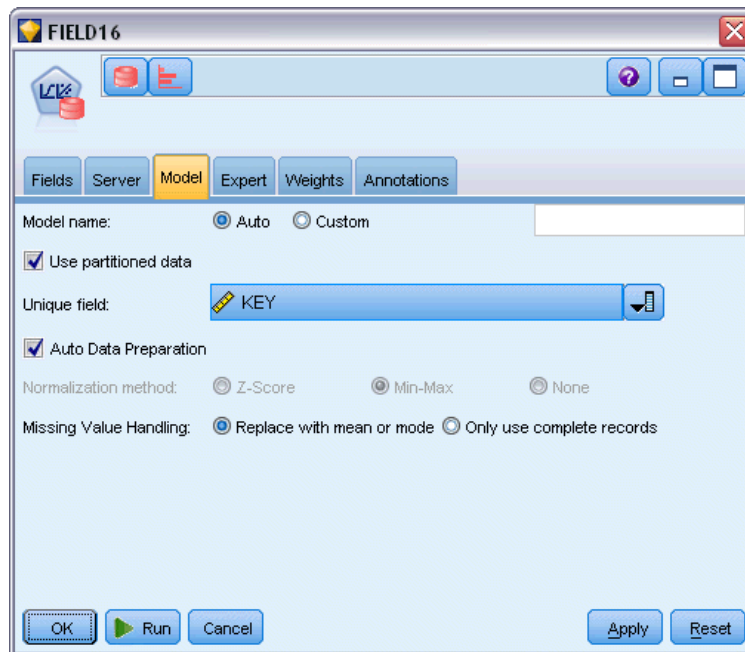
Oracle Generalized Linear Models (GLM)

(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

Figure 4-13
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

Figure 4-14
GLM expert options
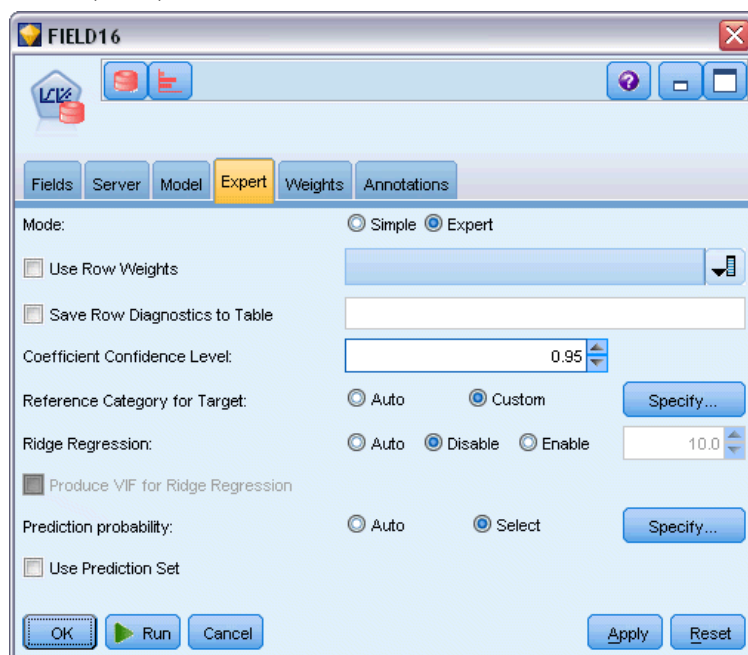
Use Row Weights. Check this box to activate the adjacent drop-down list, from where you can select a column that contains a weighting factor for the rows.

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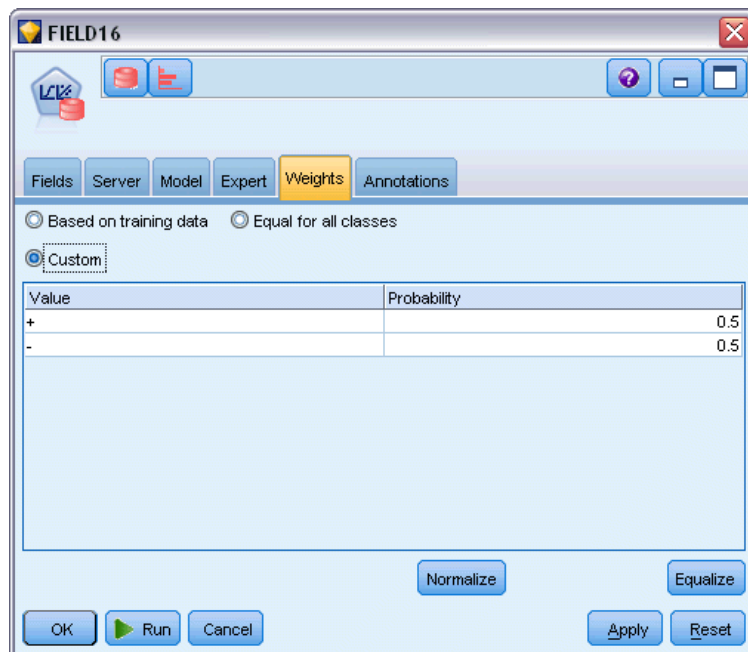
Produce VIF for Ridge Regression. Check this box if you want to produce Variance Inflation Factor (VIF) statistics when ridge is being used for linear regression.

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 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

Figure 4-15
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.
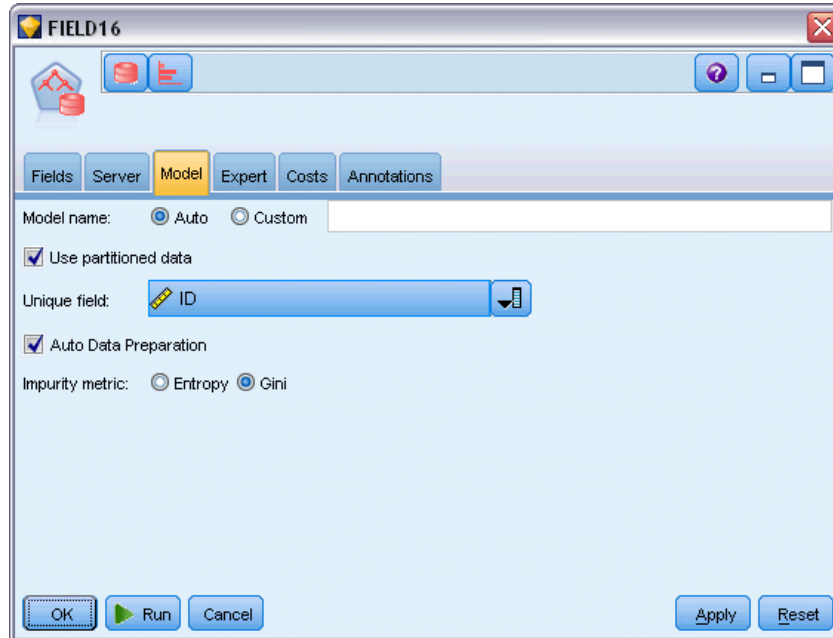
Oracle Decision Tree

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

Figure 4-16
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

Figure 4-17
Decision Tree expert options
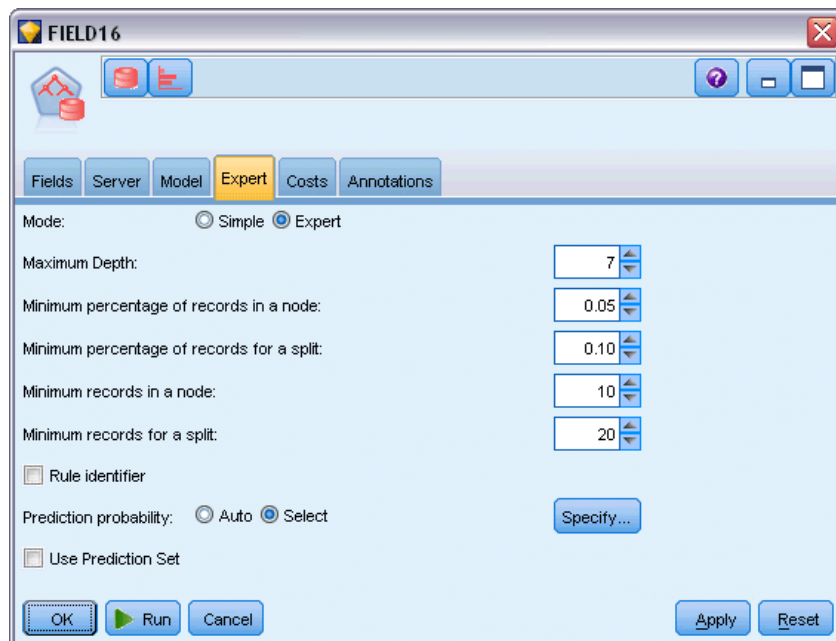
Maximum Depth. Sets the maximum depth of the tree model to be built.

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 click Insert.

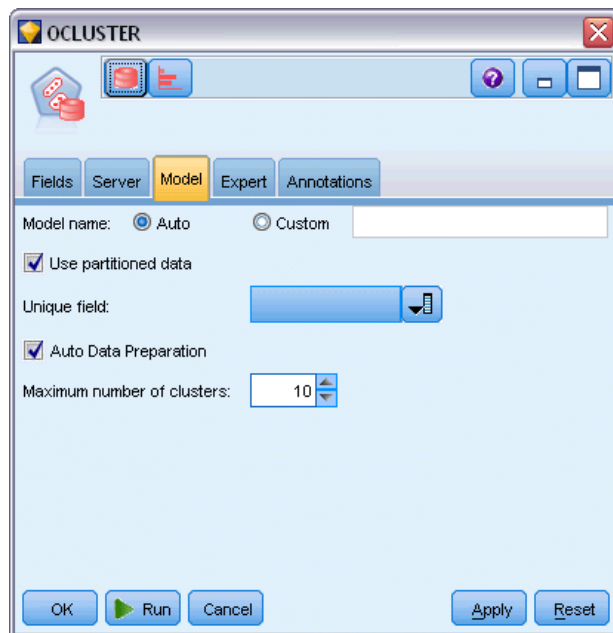**Use Prediction Set.** Generates a table of all the possible results for all possible outcomes of the target field.

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

Figure 4-18
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

Figure 4-19
O-Cluster expert options
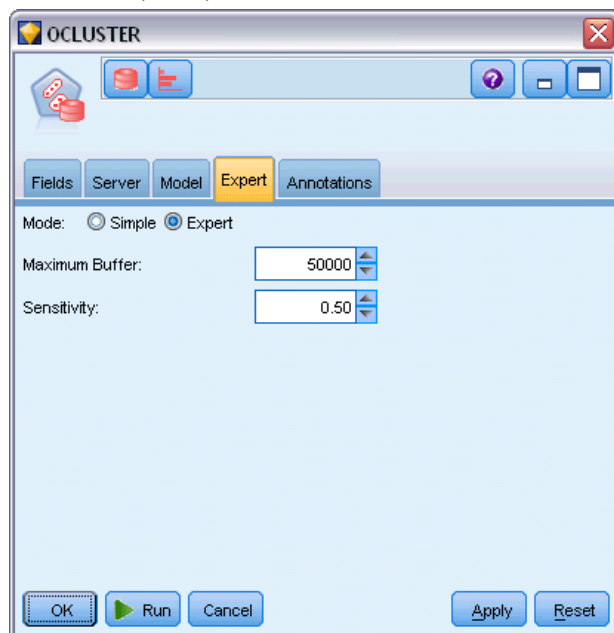
Maximum Buffer. Sets the maximum buffer size.

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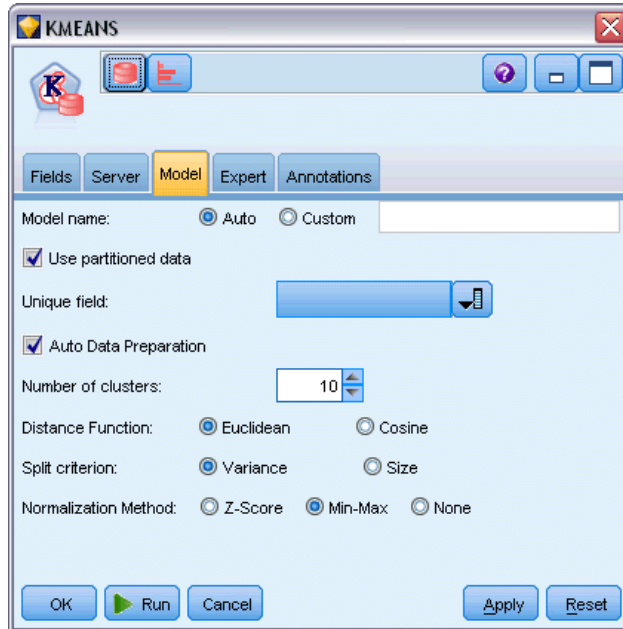
Oracle k-Means

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

Figure 4-20
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

Figure 4-21
k-Means expert options
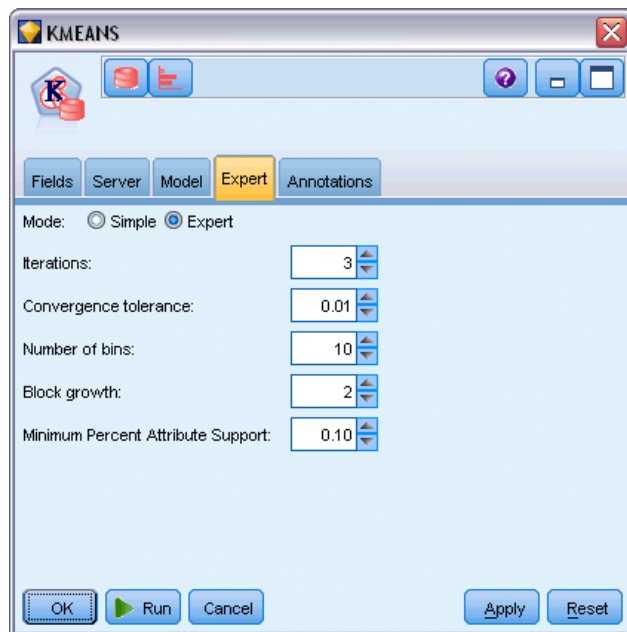
Iterations. Sets the number of iterations for the k-Means algorithm.

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.
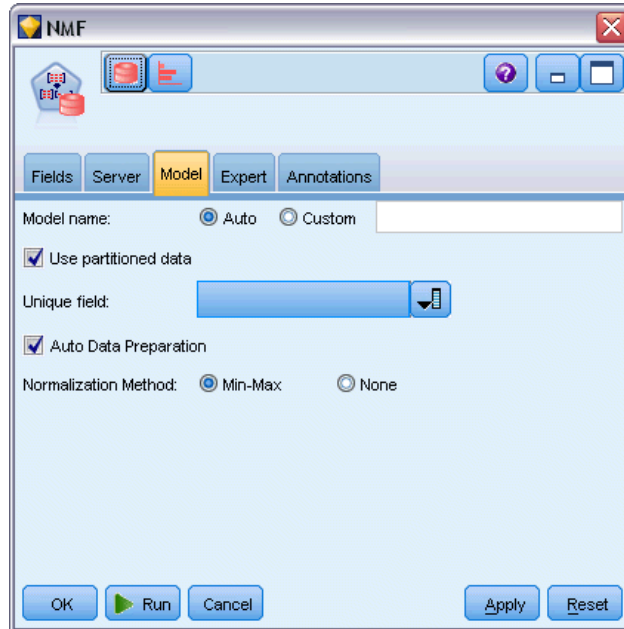
Oracle Nonnegative Matrix Factorization (NMF)

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

Figure 4-22
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

Figure 4-23
NMF expert options
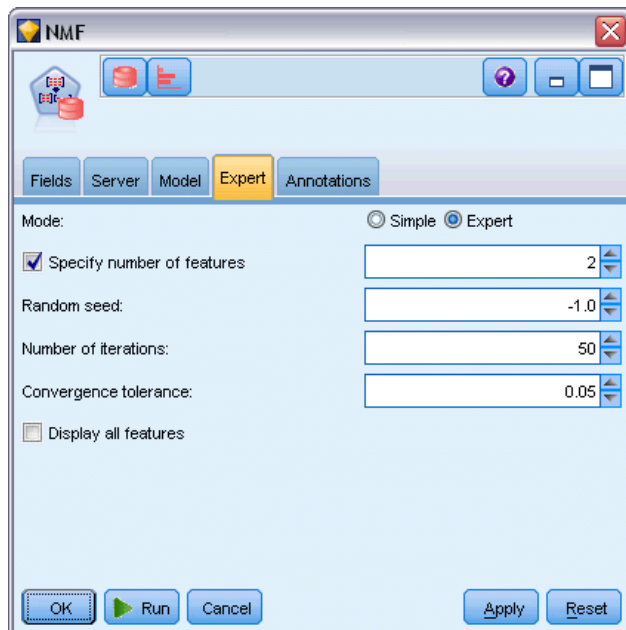
Specify number of features. Specifies the number of features to be extracted.

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.

Figure 4-24
Default custom field settings
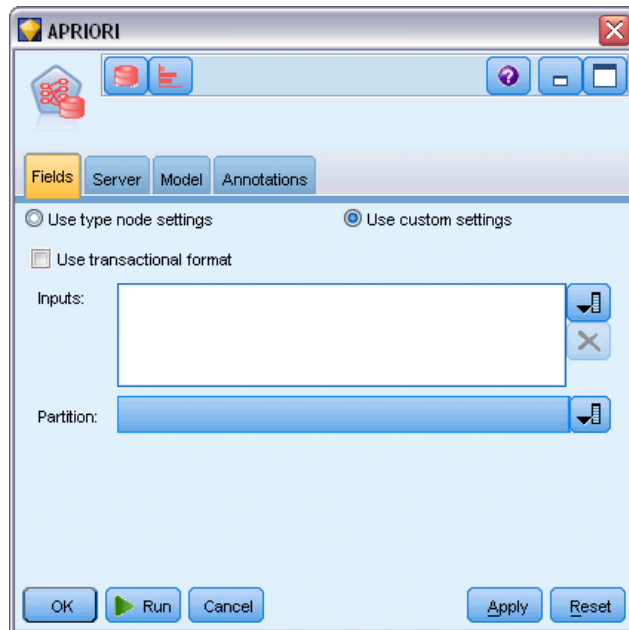
If you are *not* using transactional format, specify:

■ **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:

Figure 4-25
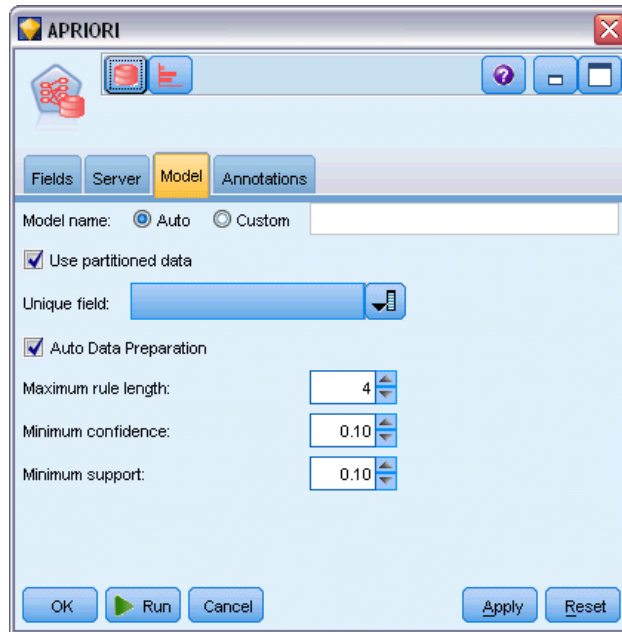*Field settings for transactional format*
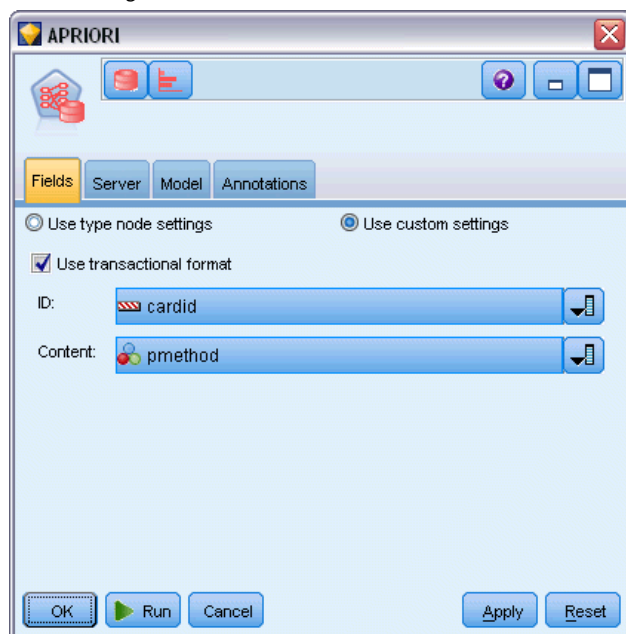
For transactional format, specify:

■ **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

Figure 4-26
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.

Figure 4-27
Using Oracle MDL chart showing the relative importance of input fields in predicting a target
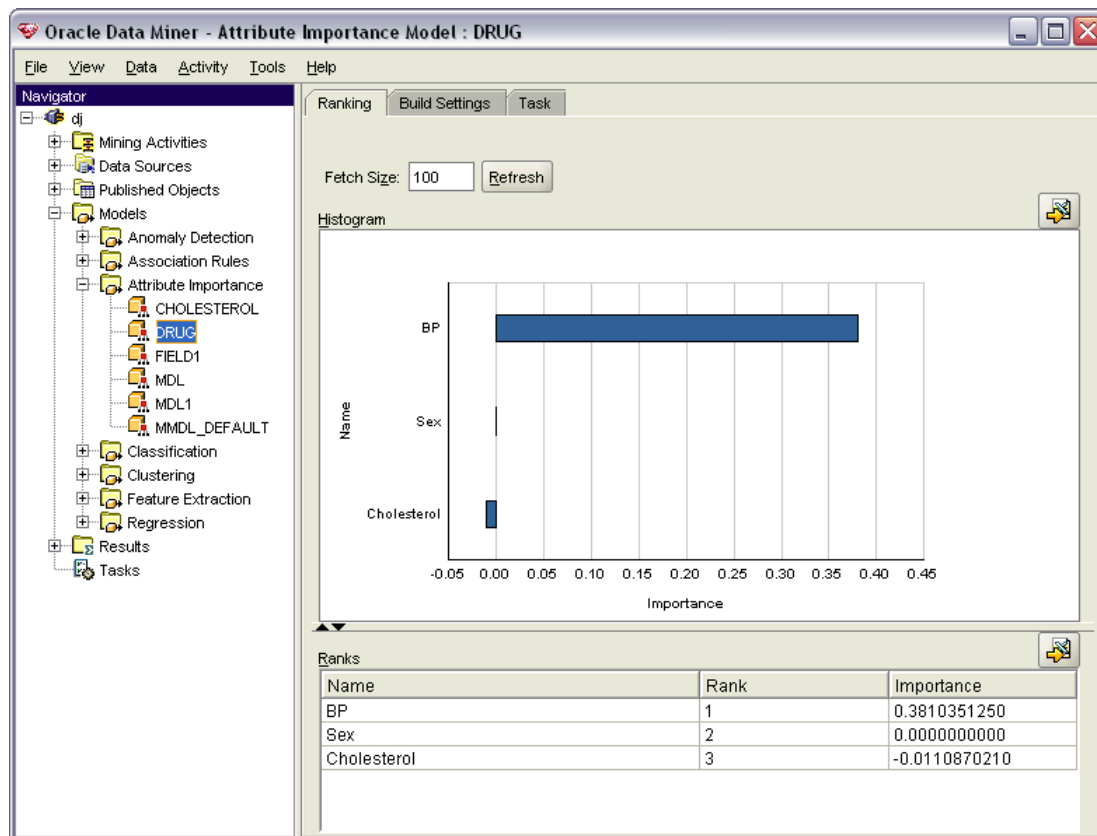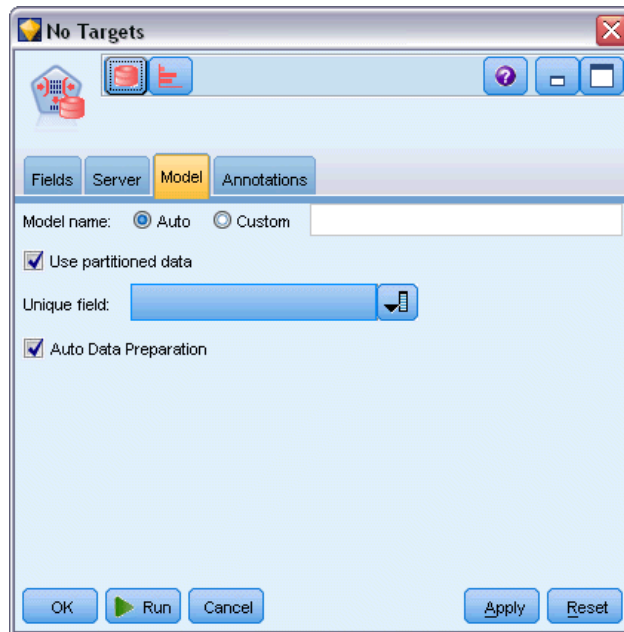

Negative ranking indicates noise. Input fields ranked at zero or less do not contribute to the prediction and should probably be removed from the data.

To display the chart

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

▶ From the model window, click the button to launch Oracle Data Miner.

▶ Connect to Oracle Data Miner. For more information, see the topic Oracle Data Miner on p. 88.

▶ In the Oracle Data Miner navigator panel, expand Models, then Attribute Importance.

▶ 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

Figure 4-28
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.
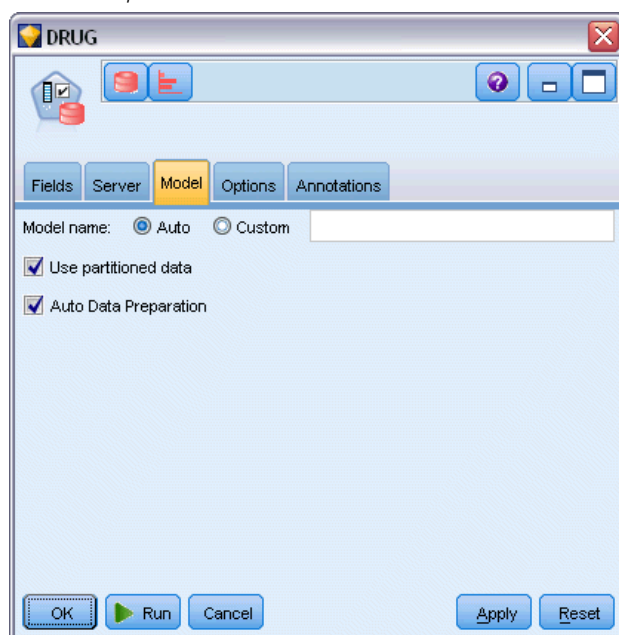
Auto Data Preparation. (11*g* only) Enables (default) or disables the automated data preparation mode of Oracle Data Mining. If this box is checked, ODM automatically performs the data transformations required by the algorithm. For more information, see *Oracle Data Mining Concepts*.

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

Figure 4-29
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.

Figure 4-30
AI selection options
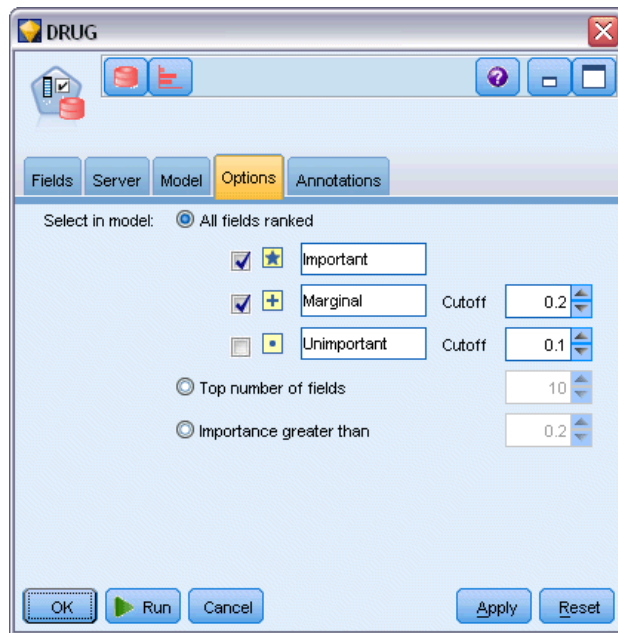
The following options are available:

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.

Figure 4-31
AI model nugget
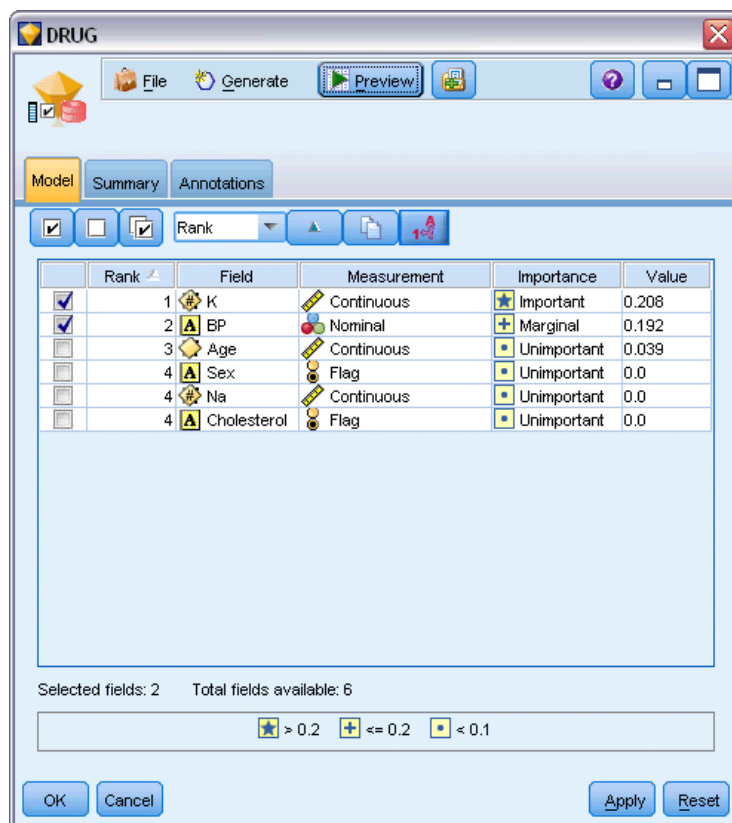
- To sort the list by rank, field name, importance, or any of the other displayed columns, click on the column header. Alternatively, select the desired item from the list next to the Sort By button, and use the up and down arrows to change the direction of the sort.

- 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 SPSS Modeler actually references a model stored on a database server.
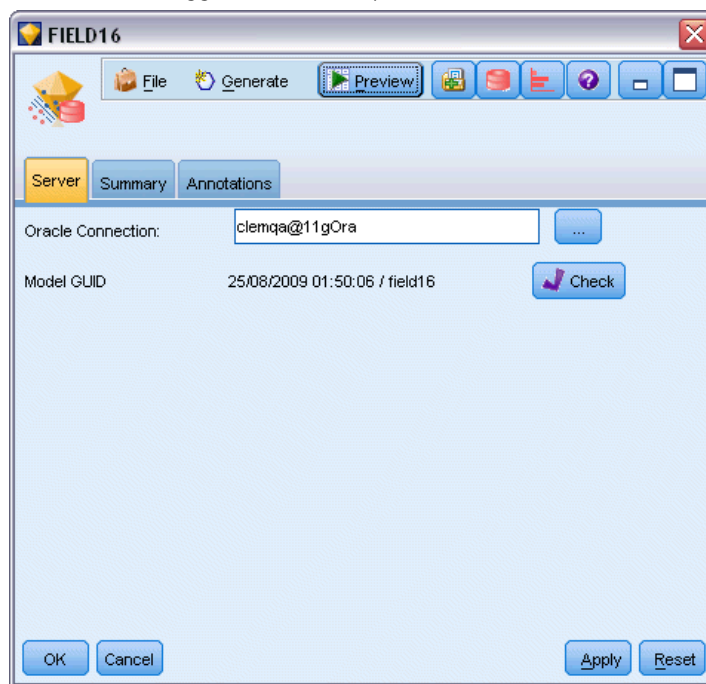
Oracle Model Nugget Server Tab

Building an ODM model via IBM® SPSS® Modeler creates a model in SPSS Modeler and creates or replaces a model in the Oracle database. An 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 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 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 on the Server tab of a model nugget can be used to check that the model keys in the 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 SPSS Modeler model was built.

Figure 4-32
Oracle model nugget Server tab options

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. For more information, see the topic Misclassification Costs on p. 56.

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.

Figure 4-33
Oracle List Models dialog box
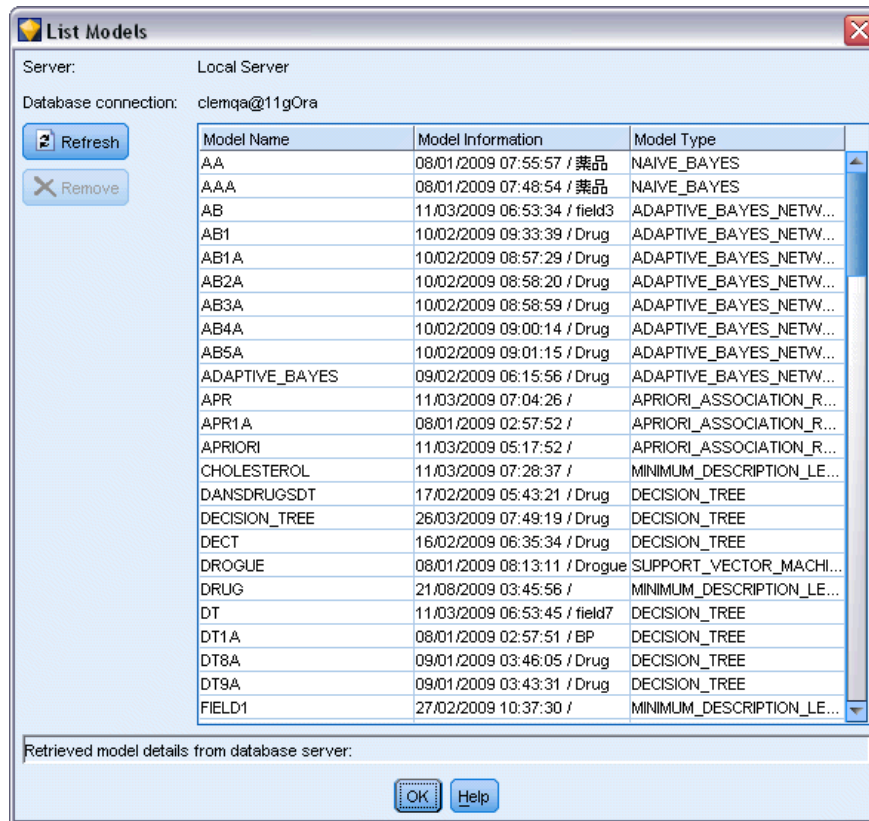
The following information is displayed for each model:

- **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

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

Figure 4-34
Launch Oracle Data Miner Button

▶ 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.

Figure 4-35
Oracle Data Miner Edit Connection dialog box
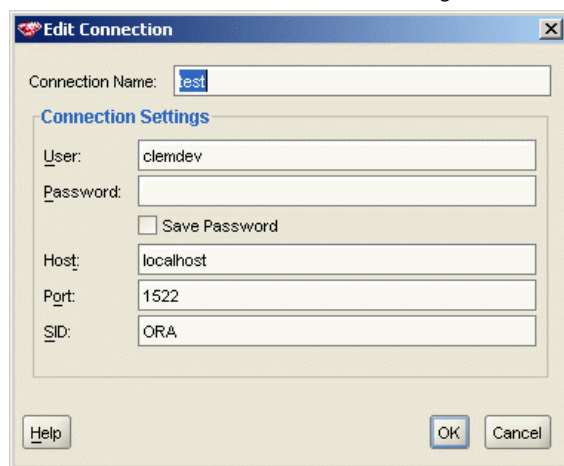
- Provide a Data Miner connection name and enter the appropriate Oracle 10gR1 or 10gR2 server information. The Oracle server should be the same server specified in SPSS Modeler.

▶ The Oracle Data Miner **Choose Connection** dialog box provides options for specifying which connection name, defined in the above step, is used.

Figure 4-36
Oracle Data Miner Choose Connection dialog box
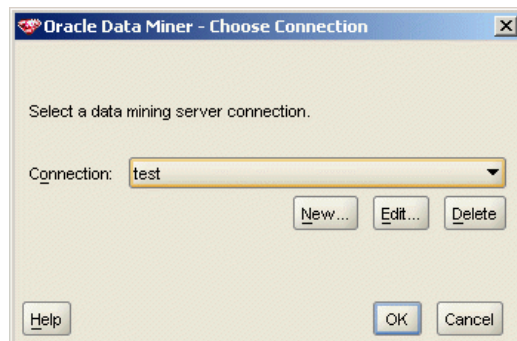
Refer to Oracle Data Miner
(*http://www.oracle.com/technology/products/bi/odm/odminer/odminer_install_102.htm*) on the
Oracle Web site for more information regarding Oracle Data Miner requirements,
installation, and usage.

Figure 4-37
Oracle Data Miner user interface
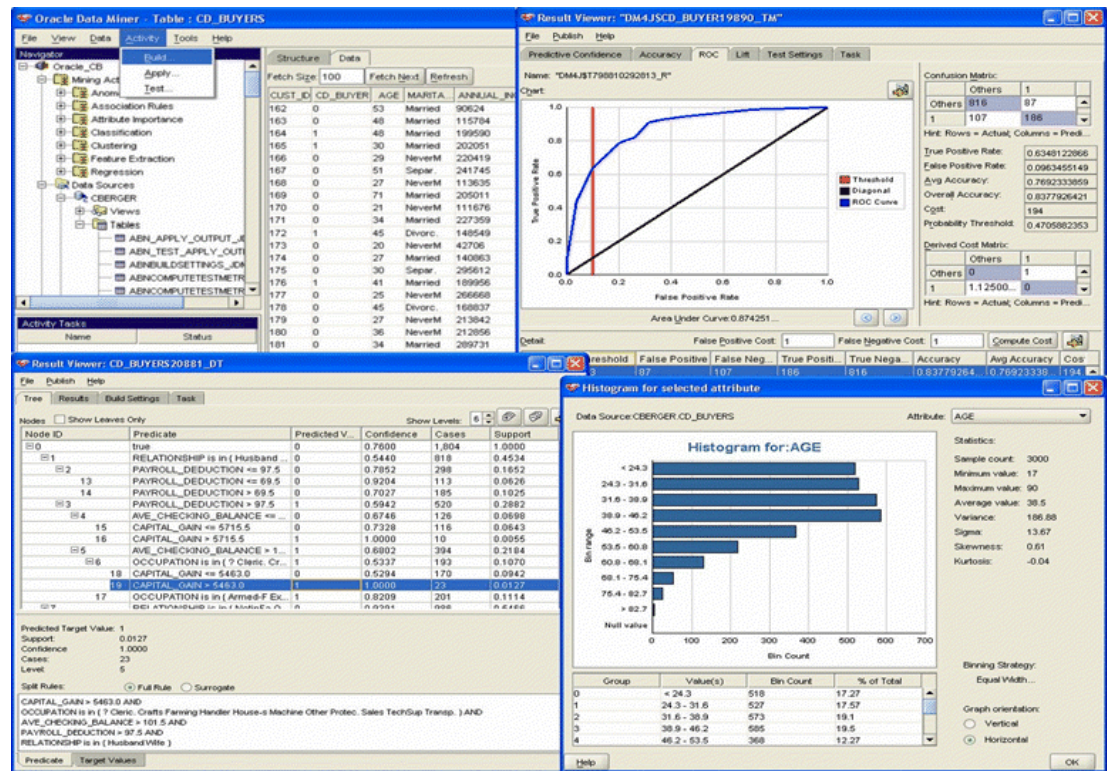
Preparing the Data

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 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 SPSS Modeler and stored with the model. At apply time, the coefficients are converted into 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 SPSS Modeler installation folder under *\Demos\Database_Modelling\Oracle Data Mining*.

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

The following streams 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:

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 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 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.
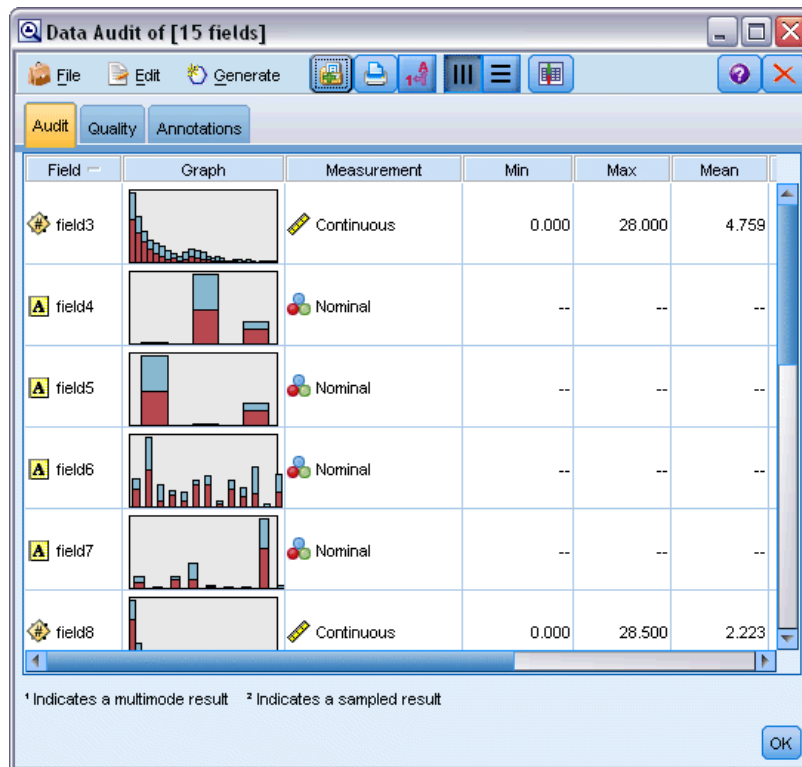
Figure 4-38
Example stream used to upload data
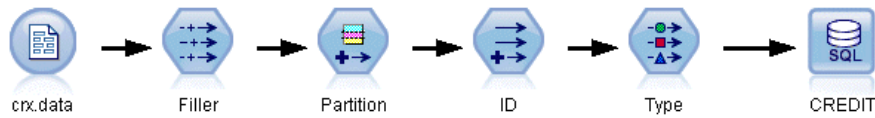
Since Oracle Data Mining requires a unique ID field, this initial stream uses a Derive node to add a new field to the dataset called *ID*, with unique values 1,2,3, using the IBM® SPSS® Modeler@INDEX function.

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.

Figure 4-39
Data Audit results

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

Figure 4-40
Histogram created by double-clicking a graph in the Data Audit window
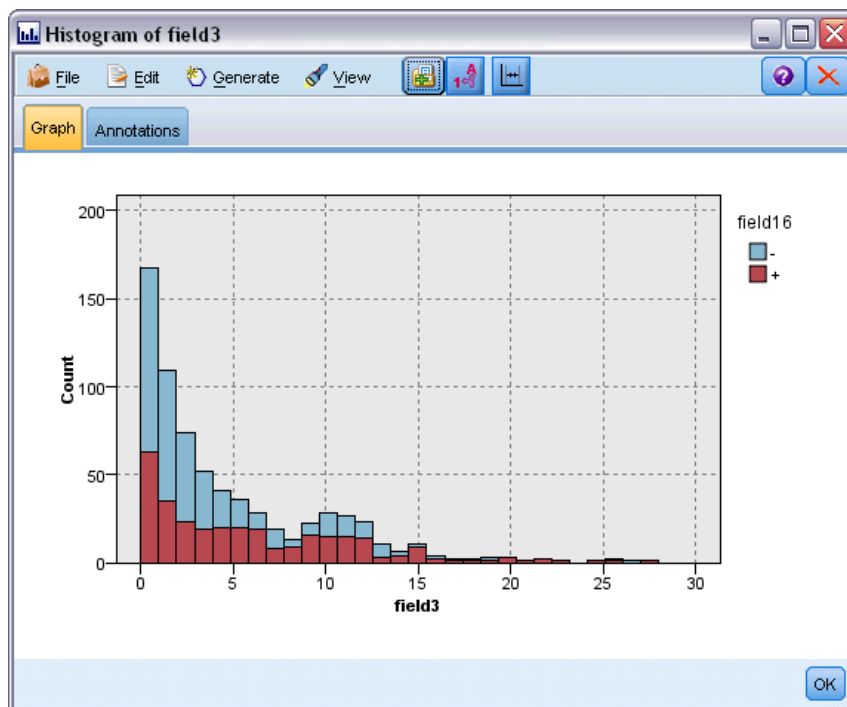
Example Stream: Build Model

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).

Figure 4-41
Database modeling example stream
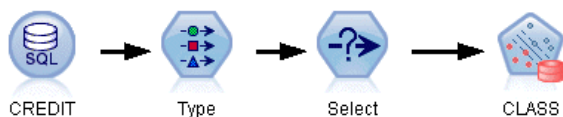
On the Model tab of the dialog box:

► Ensure that ID is selected as the Unique field.

► Ensure that Linear is selected as the kernel function and Z-Score is the normalization method.

Figure 4-42
Model options for Oracle SVM
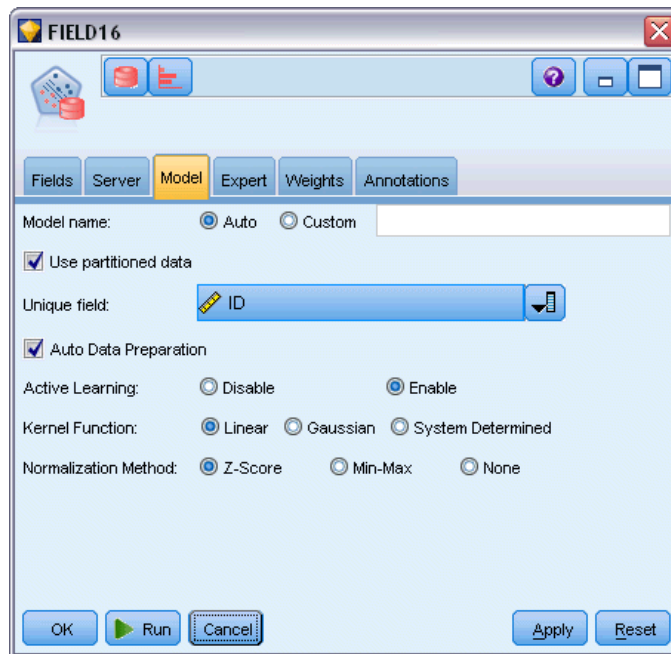
Example Stream: Evaluate Model

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 SPSS Modeler.

Figure 4-43
Example stream used for model evaluation
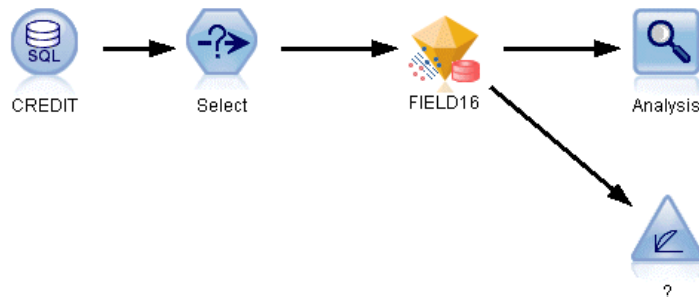
Viewing Modeling Results

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.

Figure 4-44
Table with information about generated predictions

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.

Figure 4-45
Analysis tab with information about analysis results
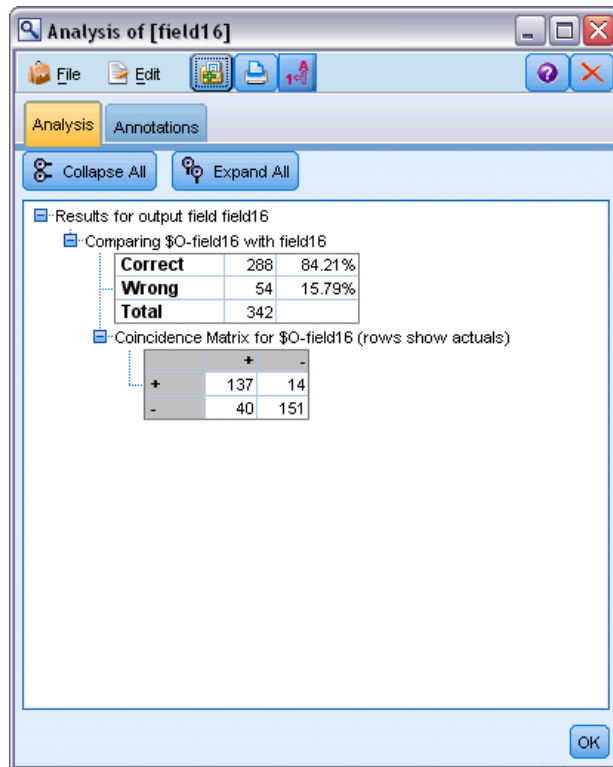
The table indicates that 84.21% of the predictions created by the Oracle SVM algorithm were correct.

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.

Figure 4-46
Gains chart with information about model accuracy improvements
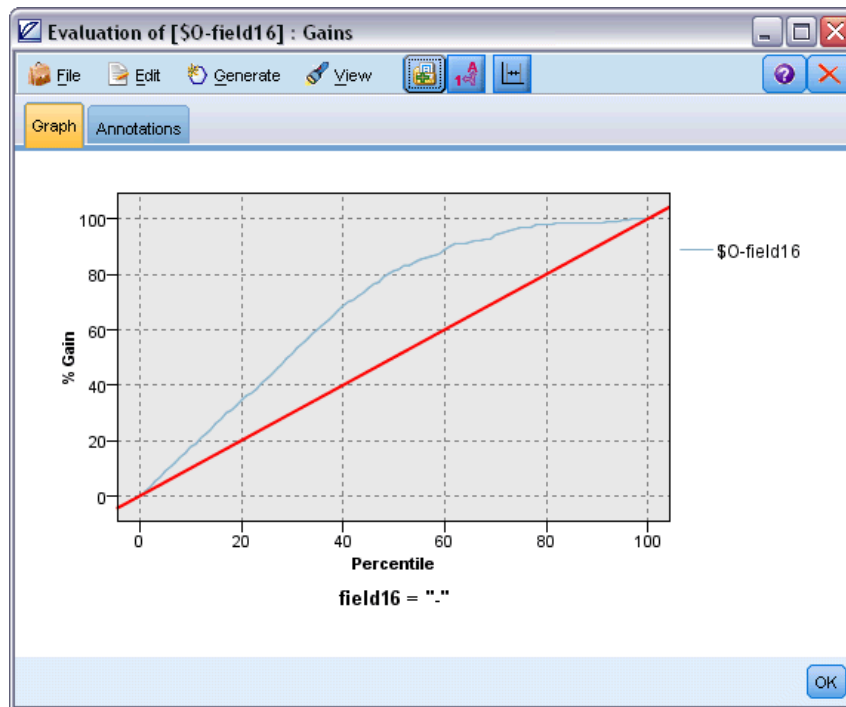
Example Stream: Deploy Model

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*.

Figure 4-47
Database modeling example stream
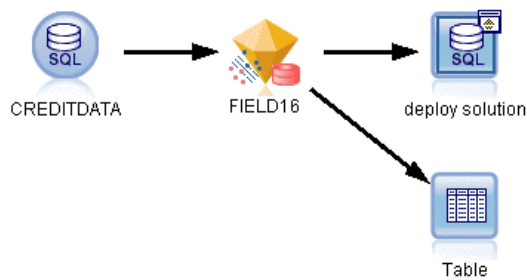
Database Modeling with IBM InfoSphere Warehouse

IBM InfoSphere Warehouse and IBM SPSS Modeler

IBM InfoSphere Warehouse (ISW) provides a family of data mining algorithms embedded within IBM's DB2 RDBMS. IBM® SPSS® Modeler provides nodes that support integration of the following IBM algorithms:

- Decision Trees
- Association Rules
- Demographic Clustering
- Kohonen Clustering
- Sequence Rules
- Transform Regression
- Linear Regression
- Polynomial Regression
- Naive Bayes
- Logistic Regression
- Time Series

For more information about these algorithms, see the documentation that accompanies your IBM InfoSphere Warehouse installation.

Requirements for Integration with InfoSphere Warehouse

The following conditions are prerequisites for conducting in-database modeling by using InfoSphere Warehouse Data Mining. 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.
- IBM DB2 Data Warehouse Edition Version 9.1

  or
- IBM InfoSphere Warehouse Version 9.5 Enterprise Edition
- An ODBC data source for connecting to DB2 as described below.

98

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

Enabling Integration with InfoSphere Warehouse

To enable the IBM® SPSS® Modeler integration with IBM InfoSphere Warehouse (ISW) Data Mining, you will need to configure ISW, create an ODBC source, enable the integration in the SPSS Modeler Helper Applications dialog box, and enable SQL generation and optimization.

Configuring ISW

To install and configure ISW, follow the instructions in the *InfoSphere Warehouse Installation* guide.

Creating an ODBC Source for ISW

To enable the connection between ISW and 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 SPSS Modeler.

If IBM® SPSS® Modeler Server and InfoSphere Warehouse Data Mining are running on different hosts, create the same ODBC DSN on each of the hosts. Be sure to use the same name for this DSN on each host.

► 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.

► 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, selectData Sources (ODBC), then click Open.

■ **Windows 7.** From the Start menu, choose Control Panel, then System & Security, then Administrative Tools. SelectData Sources (ODBC), then click Open.

► Click the System DSN tab, and then click Add.

► Select the SPSS OEM 6.0 DB2 Wire Protocol driver.

► Click Finish.

► In the ODBC DB2 Wire Protocol Driver Setup dialog box:

■ Specify a data source name.

- For the IP address, specify the hostname of the server that the DB2 RDBMS is on.
- Accept the default for the TCP port (50000).
- Specify the name of the database to which you will connect.

▶ Click Test Connection.

▶ In the Logon to DB2 Wire Protocol dialog box, enter the user name and password given to you by the database administrator, and then click OK.

The message Connection established! will appear.

IBM DB2 ODBC DRIVER. If your ODBC driver is IBM DB2 ODBC DRIVER, follow these steps to create an ODBC DSN:

▶ In the ODBC Data Source Administrator, click the System DSN tab, and then click Add.

▶ Select IBM DB2 ODBC DRIVER, and click Finish.

▶ In the IBM DB2 ODBC DRIVER—Add window, enter a data source name, and then for the database alias, click Add.

▶ In the CLI/ODBC Settings—<Data source name> window, on the Data Source tab, enter the user ID and password given to you by the database administrator, and then click the TCP/IP tab.

▶ On the TCP/IP tab, enter:
- The name of the database to which you want to connect.
- A database alias name (no more than eight characters).
- The hostname of the database server to which you want to connect.
- The port number for the connection.

▶ Click the Security Options tab and select Specify the security options (Optional), and then accept the default (Use authentication value in server's DBM Configuration).

▶ Click the Data Source tab and click Connect.

The message Connection tested successfully will appear.

Configure ODBC for Feedback (Optional)

To receive feedback from InfoSphere Warehouse Data Mining during model building and to enable SPSS Modeler to cancel model building, follow the steps below to configure the ODBC data source that was created in the previous section. Note that this configuration step allows SPSS Modeler to read DB2 data that may not be committed to the database by concurrently executing transactions. If in doubt about the implications of this change, please consult your database administrator.
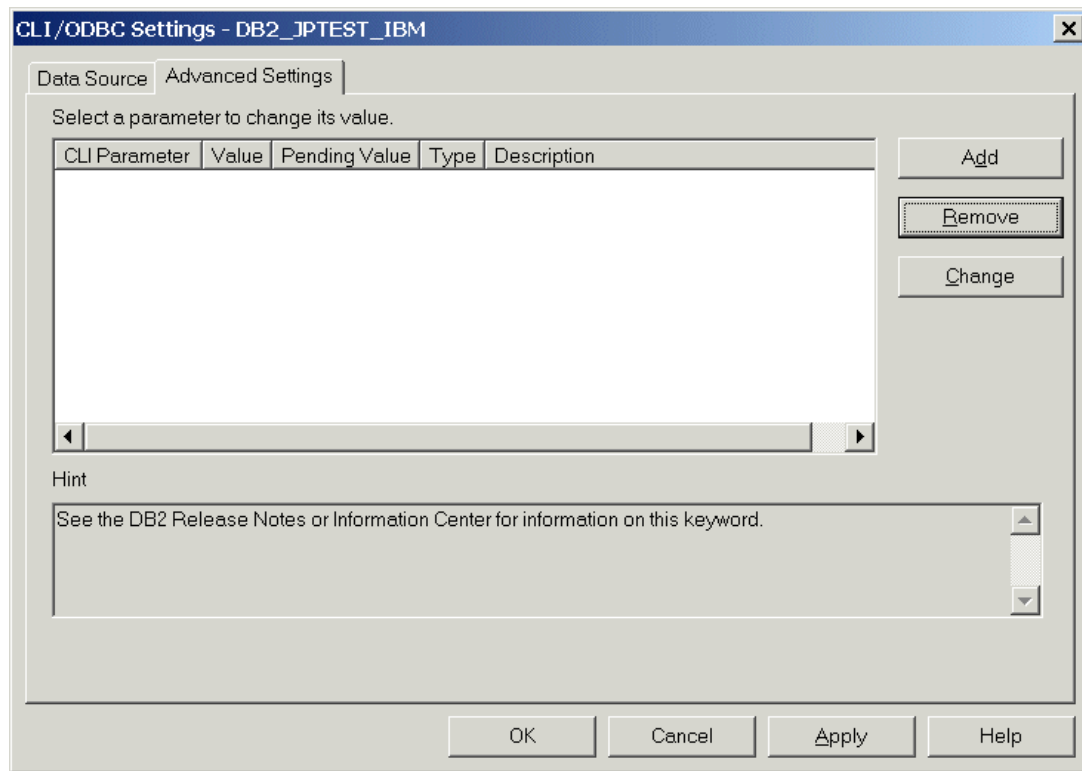
Figure 5-1
ODBC DB2 Wire Protocol Driver Setup dialog box, Advanced tab
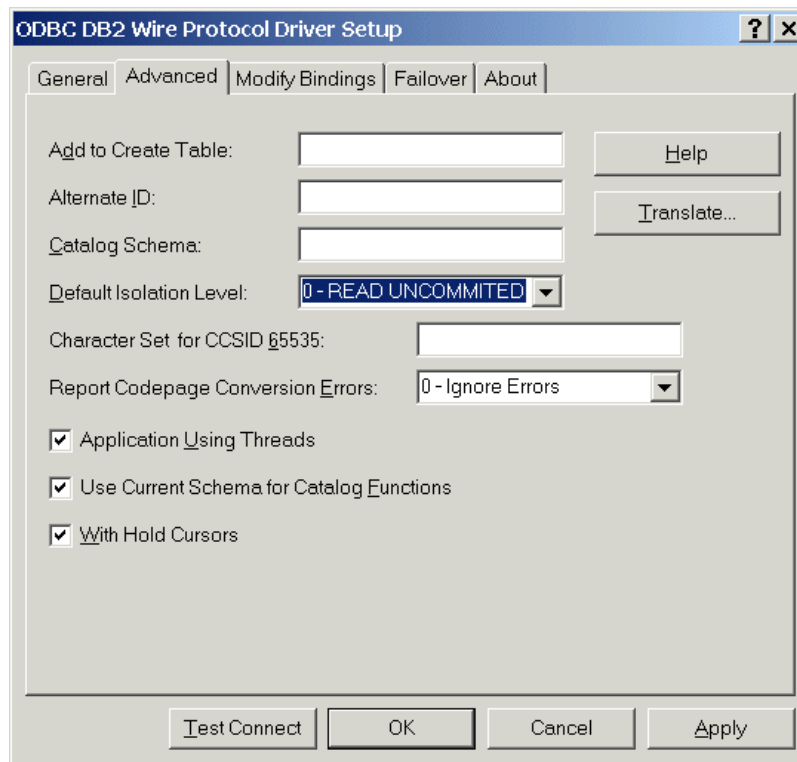
SPSS OEM 6.0 DB2 Wire Protocol Driver. For the Connect ODBC driver, follow these steps:

▶ Start the ODBC Data Source Administrator, select the data source that was created in the previous section, and click the Configure button.

▶ In the ODBC DB2 Wire Protocol Driver Setup dialog box, click the Advanced tab.

▶ Set the default isolation level to 0-READ UNCOMMITED, and then click OK.

Figure 5-2
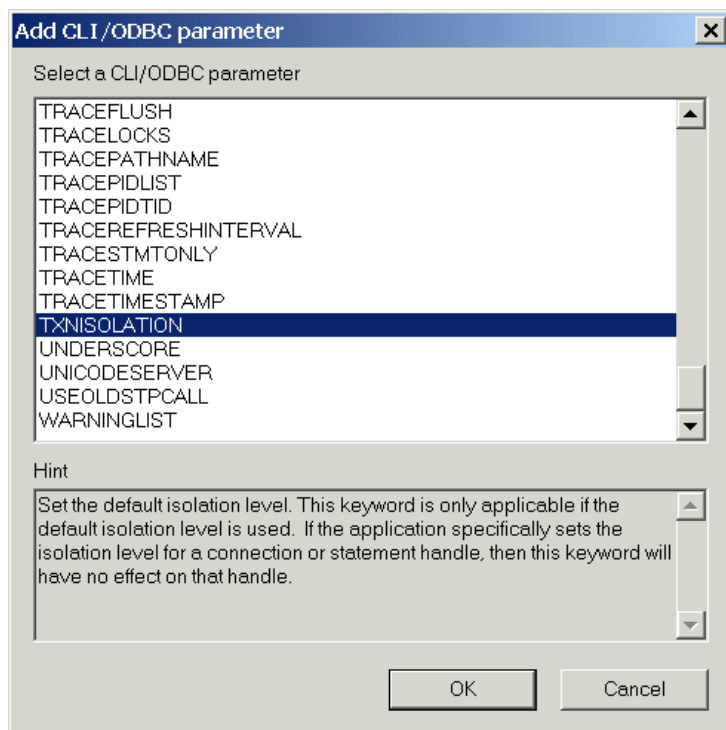*CLI/ODBC Settings dialog box, Advanced Settings tab*

IBM DB2 ODBC Driver. For the IBM DB2 driver, follow these steps:

► Start the ODBC Data Source Administrator, select the data source that was created in the previous section, and then click the Configure button.

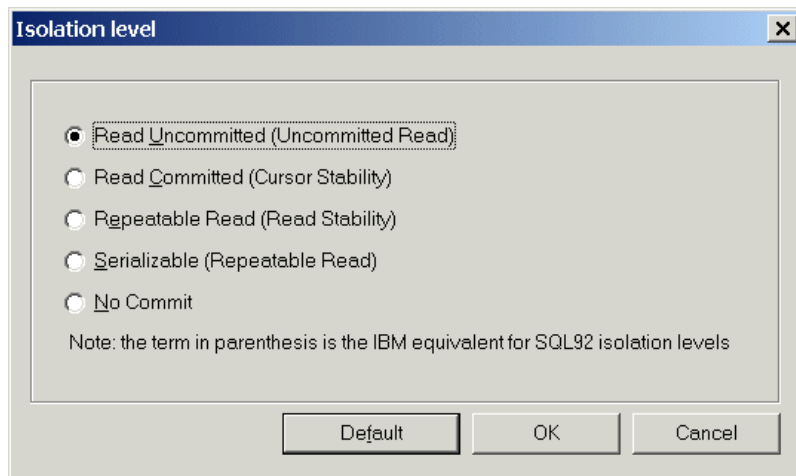► In the CLI/ODBC Settings dialog box, click the Advanced Settings tab, and then click the Add button.

Figure 5-3
CLI/ODBC Parameter dialog box

▶ In the Add CLI/ODBC Parameter dialog box, select the parameter TXNISOLATION, and then click OK.

Figure 5-4
Isolation level dialog box

▶ In the Isolation level dialog box, select Read Uncommitted, and then click OK.

▶ In the CLI/ODBC Settings dialog box, click OK to complete the configuration.

Note that feedback reported by InfoSphere Warehouse Data Mining appears in the following format:

<ITERATIONNO> / <PROGRESS> / <KERNELPHASE>

where:

■ <ITERATIONNO> indicates the number of the current pass over the data, starting at 1.

■ <PROGRESS> indicates the progress of the current iteration as a number between 0.0 and 1.0.

■ <KERNELPHASE> describes the current phase of the mining algorithm.

Enable InfoSphere Warehouse Data Mining Integration in IBM SPSS Modeler

To enable SPSS Modeler to use DB2 with InfoSphere Warehouse Data Mining, you first have to provide some specifications in the Helper Applications dialog box.

Figure 5-5
Helper Applications dialog box, IBM tab
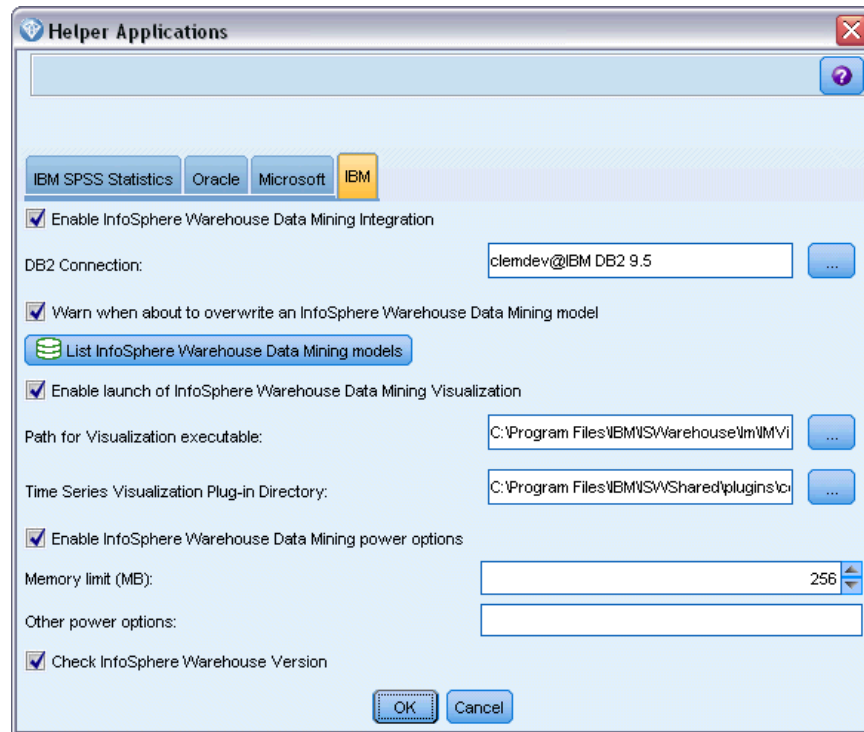
▶ From the SPSS Modeler menus choose:
Tools > Options > Helper Applications

▶ Click the IBM tab.

Enable InfoSphere Warehouse Data Mining Integration. Enables the Database Modeling palette (if not already displayed) and adds modeling nodes for ISW Data Mining Integration algorithms, including Decision Trees, Association, Sequence, Clustering, and Regression. (Once enabled, this palette is displayed with the others across the bottom of the SPSS Modeler window.)

DB2 Connection. Specifies the default DB2 ODBC data source used for building and storing models. This setting can be overridden on the individual model building and generated model nodes. Click the ellipsis (...) button to choose the data source.

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 DB2 database, downloads the data to SPSS Modeler for cleaning or other manipulation, and then uploads the data to a different DB2 database for modeling purposes. Alternatively, the original data might reside in a flat file or other (non-DB2) source, in which case the data would need to be uploaded to DB2 for modeling. In all cases, the data will be automatically uploaded to a temporary table created in the database that is used for modeling, if necessary.

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

List InfoSphere Warehouse Data Mining Models. This option allows you to list and delete the models stored in DB2. For more information, see the topic Listing Database Models on p. 108.

Enable launch of InfoSphere Warehouse Data Mining Visualization. If you have installed the Visualization module, you must enable it here for SPSS Modeler use.

Path for Visualization executable. The location of the Visualization module executable (if installed), for example *C:\Program Files\IBM\ISWarehouse\Im\IMVisualization\bin\imvisualizer.exe*.

Time Series Visualization Plug-in Directory. The location of the Time Series Visualization flash plug-in (if installed), for example *C:\Program Files\IBM\ISWShared\plugins\com.ibm.datatools.datamining.imvisualization.flash_2.2.1.v20091111_0915*.

Enable InfoSphere Warehouse Data Mining Power Options. You can set a memory consumption limit on an in-database mining algorithm and specify other arbitrary options in command-line form for specific models. The memory limit allows you to control memory consumption and specify a value for the power option `-buf`. Other power options can be specified here in command-line form and are passed on to InfoSphere Warehouse Data Mining. For more information, see the topic Power Options on p. 111.

Check InfoSphere Warehouse Version. Checks the version of InfoSphere Warehouse you are using, and reports an error if you try to use a data mining feature that is not supported in your version.

Enabling SQL Generation and Optimization

▶ From the SPSS Modeler menus choose:
Tools > Options > User Options

Figure 5-6
Optimization settings
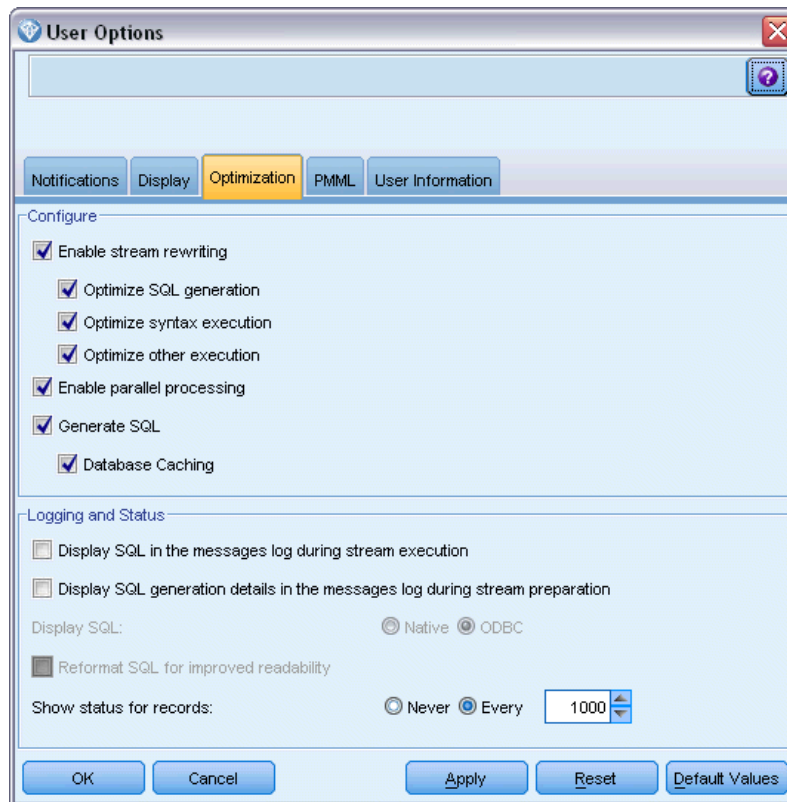
▶ Click the Optimization tab.

▶ Confirm that the Generate SQL option is enabled. This setting is required for database modeling to function.

▶ Select Optimize SQL Generation and Optimize other execution (not strictly required but strongly recommended for optimized performance).

Building Models with InfoSphere Warehouse Data Mining

InfoSphere Warehouse Data Mining model building requires that the training dataset be located in a table or view within the DB2 database. If the data are not located in DB2 or need to be processed in IBM® SPSS® Modeler as part of data preparation that cannot be performed in DB2, the data are automatically uploaded to a temporary table in DB2 before model building.

Model Scoring and Deployment

Model scoring always occurs within DB2 and is always performed by InfoSphere Warehouse Data Mining. The dataset may need to be uploaded to a temporary table if the data originates, or needs to be prepared, within IBM® SPSS® Modeler. For Decision Tree, Regression and Clustering models in SPSS Modeler, generally only a single prediction and associated probability

or confidence is delivered. In addition, a user option to display confidences for each possible outcome (similar to that found in logistic regression) is a score time option available on the model nugget Settings tab (the Include confidences for all classes check box). For Association and Sequence models in SPSS Modeler, several values are delivered. SPSS Modeler can score InfoSphere Warehouse Data Mining models from within streams published for execution by using IBM® SPSS® Modeler Solution Publisher.

The following fields are generated by scoring models:

Table 5-1
Model-scoring fields

Model Type	Score Columns	Meaning
Decision Trees	$I-*field*	Best prediction for *field*.
	$IC-*field*	Confidence of best prediction for *field*.
	$IC-*value1*, ..., $IC-*valueN*	(optional) Confidence of each of *N* possible values for *field*.
Regression	$I-*field*	Best prediction for *field*.
	$IC-*field*	Confidence of best prediction for *field*.
Clustering	$I-*model_name*	Best cluster assignment for input record.
	$IC-*model_name*	Confidence of best cluster assignment for input record.
Association	$I-*model_name*	Identifier of matching rule.
	$IH-*model_name*	Head item.
	$IHN-*model_name*	Name of head item.
	$IS-*model_name*	Support value of matching rule.
	$IC-*model_name*	Confidence value of matching rule.
	$IL-*model_name*	Lift value of matching rule.
	$IMB-*model_name*	Number of matching body items or body item sets (because all body items or body item sets must match this number, it is equal to the number of body items or body item sets).

Model Type	Score Columns	Meaning
Sequence	$I-*model_name*	Identifier of matching rule
	$IH-*model_name*	Head item set of matching rule
	$IHN-*model_name*	Names of items in head item set of matching rule
	$IS-*model_name*	Support value of matching rule
	$IC-*model_name*	Confidence value of matching rule
	$IL-*model_name*	Lift value of matching rule
	$IMB-*model_name*	Number of matching body items or body item sets (because all body items or body item sets must match this number, it is equal to the number of body items or body item sets)
Naive Bayes	$I-*field*	Best prediction for *field*.
	$IC-*field*	Confidence of best prediction for *field*.
Logistic Regression	$I-*field*	Best prediction for *field*.
	$IC-*field*	Confidence of best prediction for *field*.

Managing DB2 Models

Building an InfoSphere Warehouse Data Mining model via IBM® SPSS® Modeler creates a model in SPSS Modeler and creates or replaces a model in the DB2 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 DB2 model.

The key string for each DB2 model is displayed under the *Model Information* column in the Listing Database Models dialog box. The key string 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 DB2 model match. If no model of the same name can be found in DB2, or if the model keys do not match, the DB2 model has been deleted or rebuilt since the SPSS Modeler model was built. For more information, see the topic

Listing Database Models

IBM® SPSS® Modeler provides a dialog box for listing the models that are stored in InfoSphere Warehouse Data Mining and enables models to be deleted.

Figure 5-7
DB2 List Models dialog box
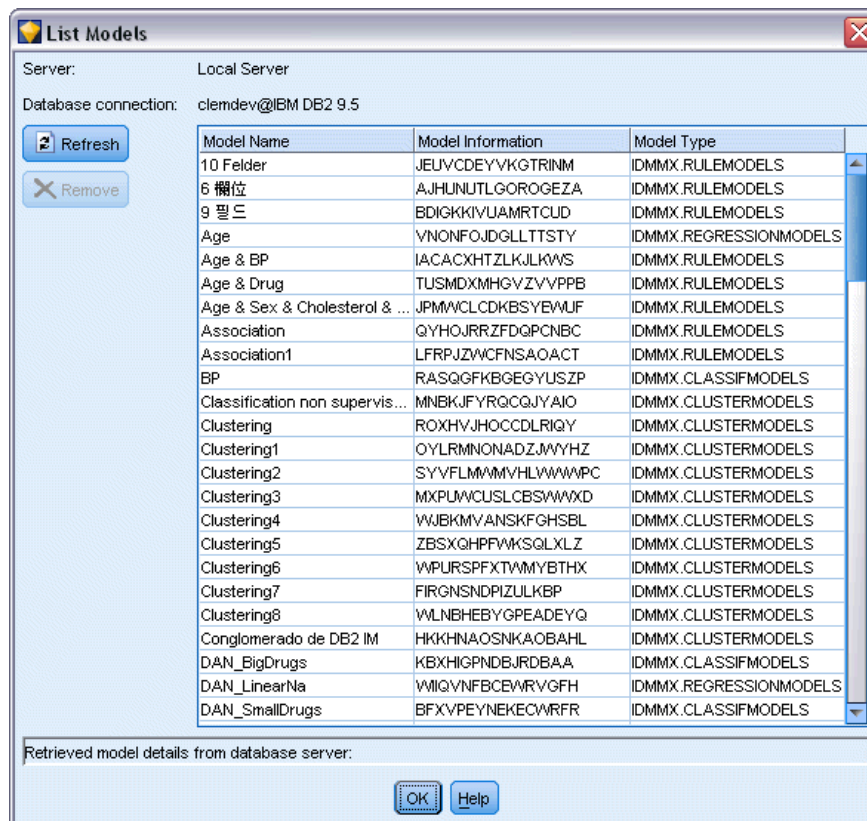
This dialog box is accessible from the IBM Helper Applications dialog box and from the build, browse, and apply dialog boxes for InfoSphere Warehouse Data Mining-related nodes. The following information is displayed for each model:

■ Model name (name of the model, which is used to sort the list).

■ Model information (model key information from a random key generated when SPSS Modeler built the model).

■ Model type (the DB2 table in which InfoSphere Warehouse Data Mining has stored the model).

Browsing Models

The Visualizer tool is the only method for browsing InfoSphere Warehouse Data Mining models. The tool can be installed optionally with InfoSphere Warehouse Data Mining. For more information, see the topic Enabling Integration with InfoSphere Warehouse on p. 99.

■ Click View to launch the visualizer tool. What the tool displays depends on the generated node type. For example, the visualizer tool will return a Predicted Classes View when launched from an ISW Decision Tree model nugget.

■ Click Test Results (Decision Trees and Sequence only) to launch the visualizer tool and view the generated model's overall quality.

Exporting Models and Generating Nodes

You can perform PMML import and export actions on InfoSphere Warehouse Data Mining models. The PMML that is exported is the original PMML generated by InfoSphere Warehouse Data Mining. The export function returns the model in PMML format.

You can export a model summary and structure to text and HTML format files. You can generate the appropriate Filter, Select, and Derive nodes where appropriate. For more information, see "Exporting Models" in the *IBM® SPSS® Modeler User's Guide*.

Node Settings Common to All Algorithms

The following settings are common to many of the InfoSphere Warehouse Data Mining algorithms:

Target and predictors. You can specify a target and predictors by using the Type node or manually by using the Fields tab of the model builder node, as is standard in IBM® SPSS® Modeler.

ODBC data source. This setting allows the user to override the default ODBC data source for the current model. (The default is specified in the Helper Applications dialog box. For more information, see the topic Enabling Integration with InfoSphere Warehouse on p. 99.)

ISW Server Tab Options

You can specify the DB2 connection 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 DB2 connection specified in the Helper Applications dialog box. For more information, see the topic Enabling Integration with InfoSphere Warehouse on p. 99.

Figure 5-8
ISW Server tab
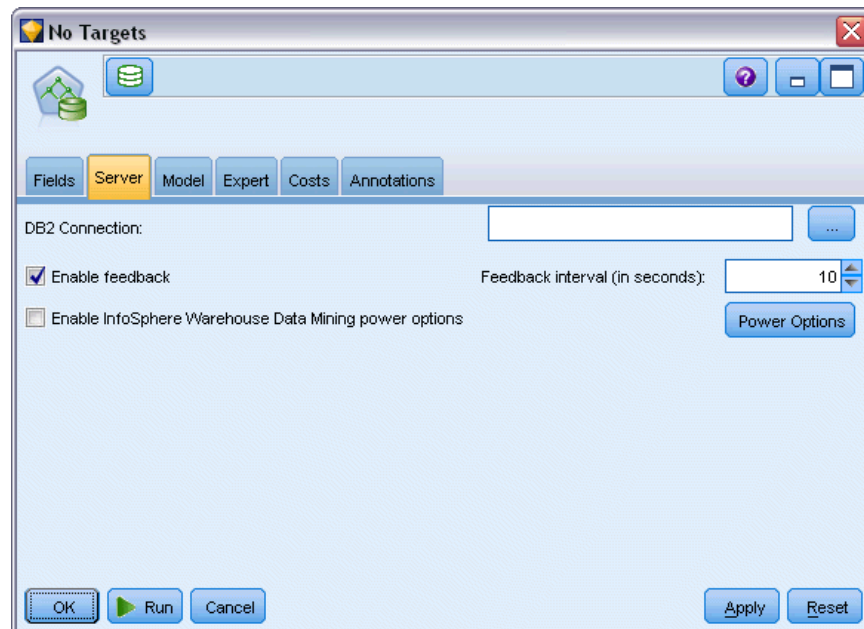
The connection used for modeling may or may not be the same as the connection used in the source node for a stream. For example, you might have a stream that accesses data from one DB2 database, downloads the data to IBM® SPSS® Modeler for cleaning or other manipulation, and then uploads the data to a different DB2 database for modeling purposes.

The ODBC data source name is effectively embedded into each SPSS Modeler stream. If a stream 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.

You can get feedback as you build a model, using the following options:

- **Enable feedback.** Select this option to get feedback during a model build (default is off).
- **Feedback interval (in seconds).** Specify how often SPSS Modeler retrieves feedback on the progress of the model build.

Enable InfoSphere Warehouse Data Mining Power Options. Select this option to enable the Power Options button, which allows you to specify a number of advanced options such as a memory limit and custom SQL. For more information, see the topic Power Options on p. 111.

The Server tab for a generated node includes an option to perform consistency checking by storing an identical generated model key string in both the SPSS Modeler model and the DB2 model. For more information, see the topic ISW Model Nugget Server Tab on p. 140.

Power Options

The Server tab for all algorithms includes a check box for enabling ISW modeling power options. When you click the Power Options button, the ISW Power Options dialog appears, with options for:

- Memory limit.
- Other power options.
- Mining data custom SQL.
- Logical data custom SQL.
- Mining settings custom SQL.

Figure 5-9
ISW Power Options settings
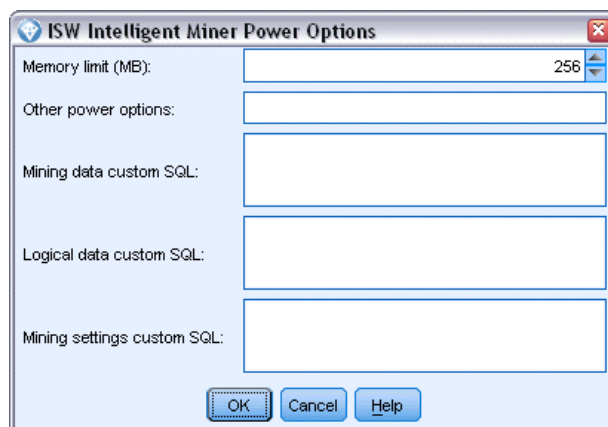
Memory Limit. Limits the memory consumption of a model-building algorithm. Note that the standard power option sets a limit on the number of discrete values in categorical data.

Other power options. Allows arbitrary power options to be specified in command-line form for specific models or solutions. Specifics may vary, depending on the implementation or solution. You can manually extend the SQL generated by IBM® SPSS® Modeler to define a model-building task.

Mining data custom SQL. You can add method calls to modify the DM_MiningData object. For example, entering the following SQL adds a filter based on a field called *Partition* to the data used in model building:

..DM_setWhereClause('"Partition" = 1')

Logical data custom SQL. You can add method calls to modify the DM_LogicalDataSpec object. For example, the following SQL removes a field from the set of fields used for model building:

..DM_remDataSpecFld('field6')

Mining settings custom SQL. You can add method calls to modify the DM_ClasSettings/DM_RuleSettings/DM_ClusSettings/DM_RegSettings object. For example, entering the following SQL instructs InfoSphere Warehouse Data Mining to set the field *Partition* to active (meaning that it is always included in the resulting model):

..DM_setFldUsageType('Partition',1)

ISW Costs Options

On the Costs tab, you can adjust the misclassification costs, which allow you to specify the relative importance of different kinds of prediction errors.

Figure 5-10
ISW Costs tab

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.

ISW Decision Tree

Decision tree models allow you to develop classification systems that predict or classify future observations based on a set of decision rules. If you have data divided into classes that interest you (for example, high- versus low-risk loans, subscribers versus nonsubscribers, voters versus nonvoters, or types of bacteria), you can use your data to build rules that you can use to classify old or new cases with maximum accuracy. For example, you might build a tree that classifies credit risk or purchase intent based on age and other factors.

The ISW Decision Tree algorithm builds classification trees on categorical input data. The resulting decision tree is binary. Various settings, including misclassification costs, can be applied to build the model.

The ISW Visualizer tool is the only method for browsing InfoSphere Warehouse Data Mining models.

ISW Decision Tree Model Options

Figure 5-11
ISW Decision Tree node Model tab
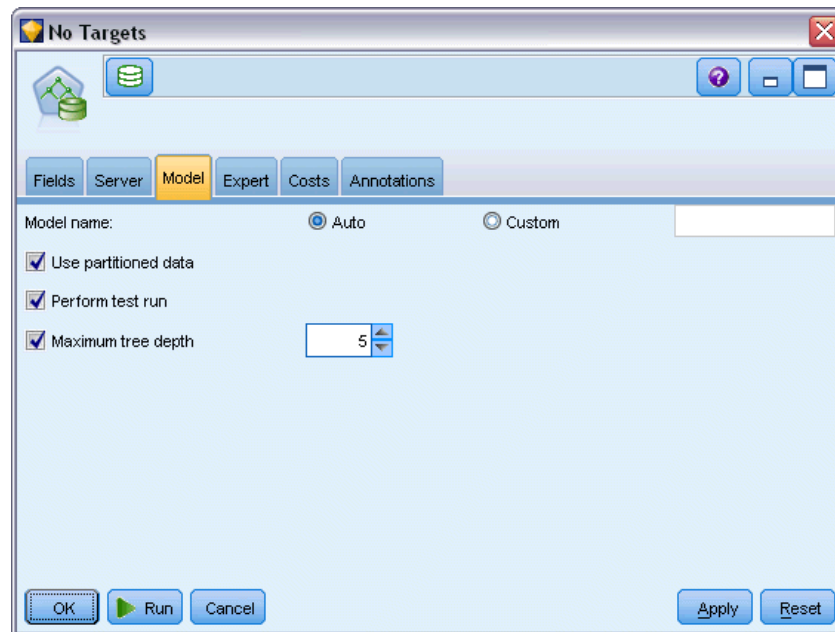
Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Use partitioned data. If you define a partition field, select Use partitioned data.

Perform test run. You can choose to perform a test run. An InfoSphere Warehouse Data Mining test run is then executed after the model is built on the training partition. This performs a pass over the test partition to establish model quality information, lift charts, and so on.

Maximum tree depth. You can specify the maximum tree depth. This limits the depth of the tree to the specified number of levels. If this option is not selected, no limit is enforced. To avoid overly complex models, a value greater than 5 is seldom recommended.

ISW Decision Tree Expert Options

Figure 5-12
ISW Decision Tree node Expert tab
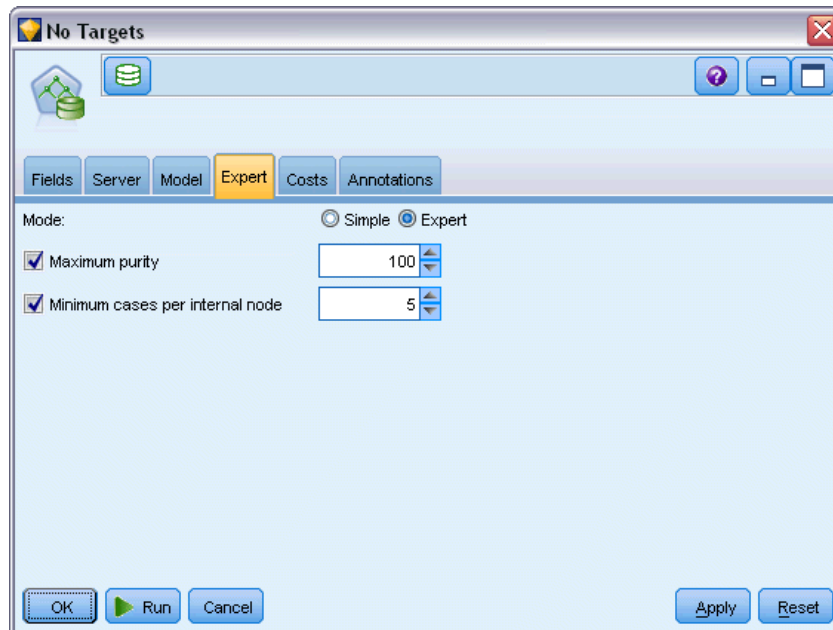
Maximum purity. This option sets the maximum purity for internal nodes. If splitting a node causes one of the children to exceed the specified purity measure (for example, more than 90% of cases fall into a specified category), the node will not be split.

Minimum cases per internal node. If splitting a node results in a node with fewer cases than the specified minimum, the node will not be split.

ISW Association

You can use the ISW Association node to find association rules among items that are present in a set of groups. Association rules associate a particular conclusion (for example, the purchase of a particular product) with a set of conditions (for example, the purchase of several other products).

You can choose to include or exclude association rules from the model by specifying **constraints**. If you choose to include a particular input field, association rules that contain at least one of the specified items are included in the model. If you exclude an input field, association rules that contain any of the specified items are discarded from the results.

The ISW Association and Sequence algorithms can make use of **taxonomies**. Taxonomies map individual values to higher-level concepts. For example, pens and pencils can be mapped to a stationery category.

Association rules have a single consequent (the conclusion) and multiple antecedents (the set of conditions). An example is as follows:

[Bread, Jam] ▯ [Butter]

[Bread, Jam]
 ▯ [Margarine]

Here, Bread and Jam are the antecedents (also known as the **rule body**) and Butter or Margarine are each examples of a consequent (also known as the **rule head**). The first rule indicates that a person who bought bread and jam also bought butter at the same time. The second rule identifies a customer who, when buying the same combination (bread and jam) also bought margarine in the same visit to the store.

The Visualizer tool is the only method for browsing InfoSphere Warehouse Data Mining models.

ISW Association Model Options

Figure 5-13
ISW Association node Model tab

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.

Minimum rule support (%). Minimum support level for association or sequence rules. Only rules that achieve at least this level of support are included in the model. The value is calculated as A/B*100, where A is the number of groups containing all the items that appear in the rule, and B is the total number of all the groups that are considered. If you want to focus on more common associations or sequences, increase this setting.

Minimum rule confidence (%). Minimum confidence level for association or sequence rules. Only rules that achieve at least this level of confidence are included in the model. The value is calculated as $m/n*100$, where m is the number of groups containing the joined rule head (consequent) and rule body (antecedent), and n is the number of groups containing the rule body. If you are getting too many associations or sequences, or uninteresting ones, try increasing this setting. If you are getting too few associations or sequences, try decreasing this setting.

Maximum rule size. Maximum number of items allowed in a rule, including the consequent item. If the associations or sequences of interest are relatively short, you can decrease this setting to speed up building the set.

Note: Only nodes with transactional input format are scored; truth table (tabular data) formats remain unrefined.

ISW Association Expert Options

On the Expert tab of the Association node, you can specify which association rules are to be included in the results, or excluded from the results. If you decide to include specified items, the rules that contain at least one of the specified items are included in the model. If you decide to exclude specified items, the rules that contain any of the specified items are discarded from the results.

Figure 5-14
ISW Association node Expert tab
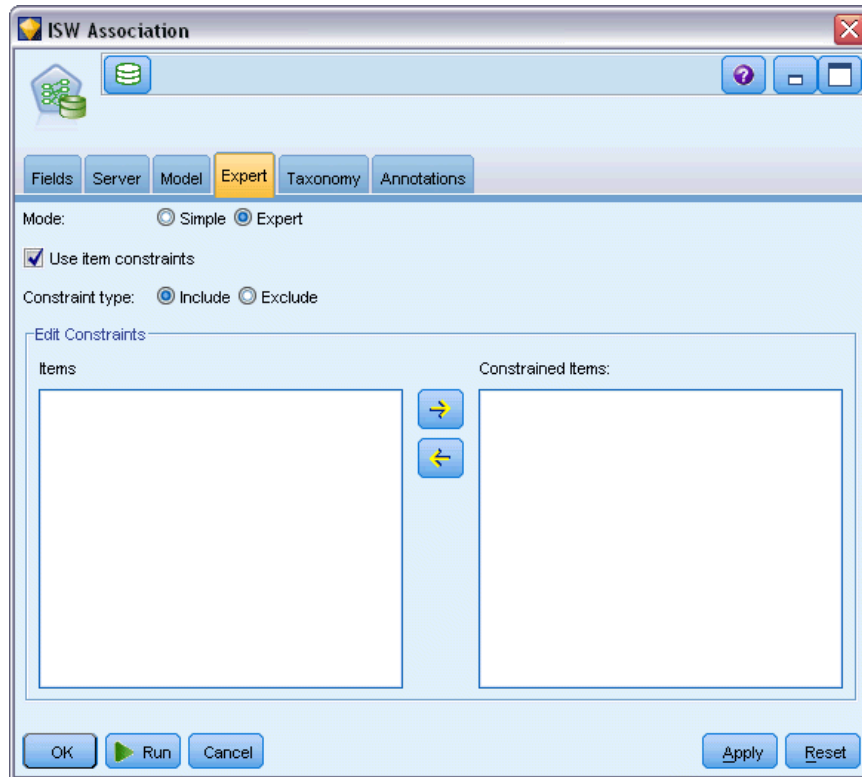
When Use item constraints is selected, any items you have added to the constraints list will be included or excluded from the results, depending on your setting for Constraint type.

Constraint type. Choose whether you want to include or exclude from the results those association rules containing the specified items.

Edit Constraints. To add an item to the list of constrained items, select it in the Items list and click the right-arrow button.

ISW Taxonomy Options

The ISW Association and Sequence algorithms can make use of **taxonomies**. Taxonomies map individual values to higher-level concepts. For example, pens and pencils can be mapped to a stationery category.

On the Taxonomy tab, you can define category maps to express taxonomies across the data. For example, a taxonomy might create two categories (Staple and Luxury) and then assign basic items to each of these categories. For example, wine is assigned toLuxury and bread is assigned to Staple. The taxonomy has a parent-child structure, as follows:

Child	Parent
wine	Luxury
bread	Staple

With this taxonomy in place, you can build an association or sequence model that includes rules involving the categories as well as the basic items.

Note: To activate the options on this tab, the source data must be in transactional format, and you must select Use transactional format on the Fields tab, then select Use Taxonomy on this tab.

Figure 5-15
ISW Association node Taxonomy tab
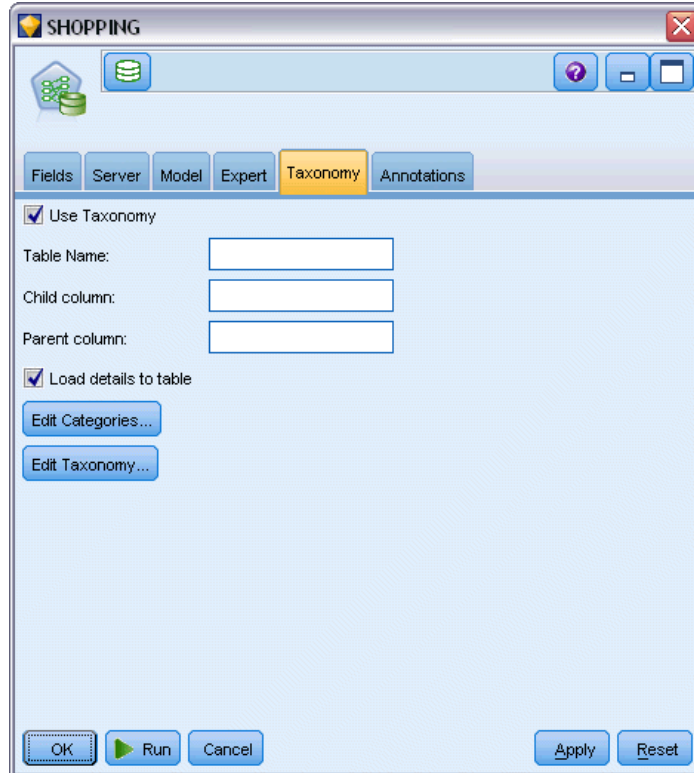
Table Name. This option specifies the name of the DB2 table to store taxonomy details.

Child column. This option specifies the name of the child column in the taxonomy table. The child column contains the item names or category names.

Parent column. This option specifies the name of the parent column in the taxonomy table. The parent column contains the category names.

Load details to table. Select this option if taxonomy information stored in IBM® SPSS® Modeler should be uploaded to the taxonomy table at model build time. Note that the taxonomy table is dropped if it already exists. Taxonomy information is stored with the model build node and is edited by using the Edit Categories and Edit Taxonomy buttons.

Categories Editor

The Edit Categories dialog box allows categories to be added to and deleted from a sorted list.

Figure 5-16
Taxonomy category editor
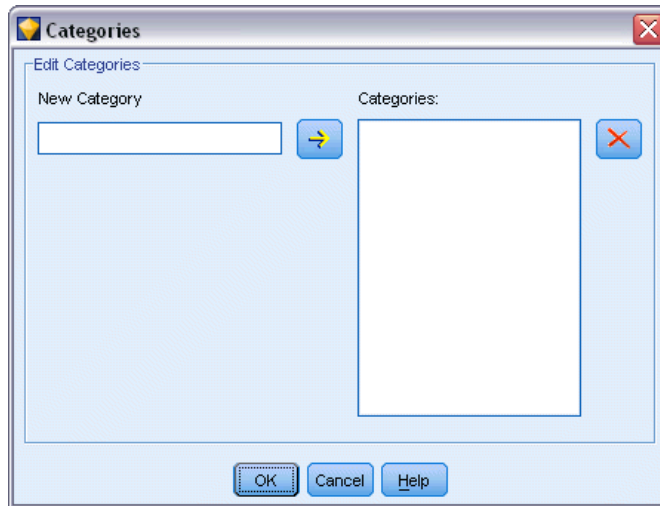
To add a category, type its name in the New Category field and click the arrow button to move it to the Categories list.

To remove a category, select it in the Categories list and click the adjacent Delete button.

Taxonomy Editor

The Edit Taxonomy dialog box allows the set of basic items defined in the data and the set of categories to be combined to build a taxonomy. To add entries to the taxonomy, select one or more items or categories from the list on the left and one or more categories from the list on the right, and then click the arrow button. Note that if any additions to the taxonomy result in a conflict (for example, specifying cat1 -> cat2 and the opposite, cat2 -> cat1), those additions are not made.

Figure 5-17
Taxonomy editor
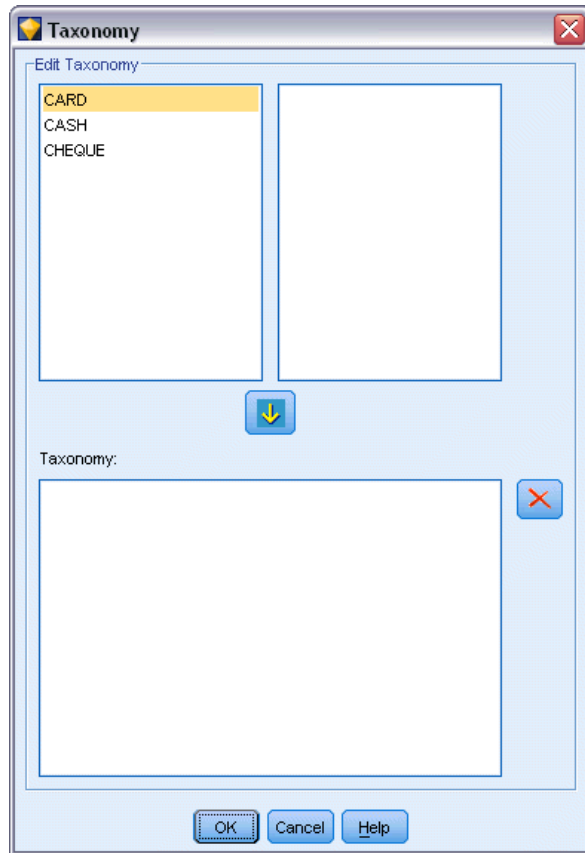
ISW Sequence

The Sequence node discovers patterns in sequential or time-oriented data, in the format `bread -> cheese`. The elements of a sequence are **item sets** that constitute a single transaction. For example, if a person goes to the store and purchases bread and milk and then a few days later returns to the store and purchases some cheese, that person's buying activity can be represented as two item sets. The first item set contains bread and milk, and the second one contains cheese. A **sequence** is a list of item sets that tend to occur in a predictable order. The Sequence node detects frequent sequences and creates a generated model node that can be used to make predictions.

You can use the Sequence Rules mining function in various business areas. For example, in the retail industry, you can find typical series of purchases. These series show you the different combinations of customers, products, and time of the purchase. With this information, you can identify potential customers for a particular product who have not yet bought the product. Furthermore, you can offer products to the potential customers in due time.

A sequence is an ordered set of item sets. Sequences contain the following grouping levels:

- Events that happen simultaneously form a single transaction or an item set.
- Each item or each item set belongs to a transaction group. For example, a purchased article belongs to a customer, a particular page click belongs to a Web surfer, or a component belongs to a produced car. Several item sets that occur at different times and belong to the same transaction group form a sequence.

ISW Sequence Model Options

Figure 5-18
ISW Sequence node Model tab

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.

Minimum rule support (%). Minimum support level for association or sequence rules. Only rules that achieve at least this level of support are included in the model. The value is calculated as A/B*100, where A is the number of groups containing all the items that appear in the rule, and B is the total number of all the groups that are considered. If you want to focus on more common associations or sequences, increase this setting.

Minimum rule confidence (%). Minimum confidence level for association or sequence rules. Only rules that achieve at least this level of confidence are included in the model. The value is calculated as $m/n*100$, where m is the number of groups containing the joined rule head (consequent) and rule body (antecedent), and n is the number of groups containing the rule body. If you are getting too many associations or sequences, or uninteresting ones, try increasing this setting. If you are getting too few associations or sequences, try decreasing this setting.

Maximum rule size. Maximum number of items allowed in a rule, including the consequent item. If the associations or sequences of interest are relatively short, you can decrease this setting to speed up building the set.

Note: Only nodes with transactional input format are scored; truth table (tabular data) formats remain unrefined.

ISW Sequence Expert Options

You can specify which sequence rules are to be included in the results, or excluded from the results. If you decide to include specified items, the rules that contain at least one of the specified items are included in the model. If you decide to exclude specified items, the rules that contain any of the specified items are discarded from the results.

Figure 5-19
ISW Sequence node Expert tab
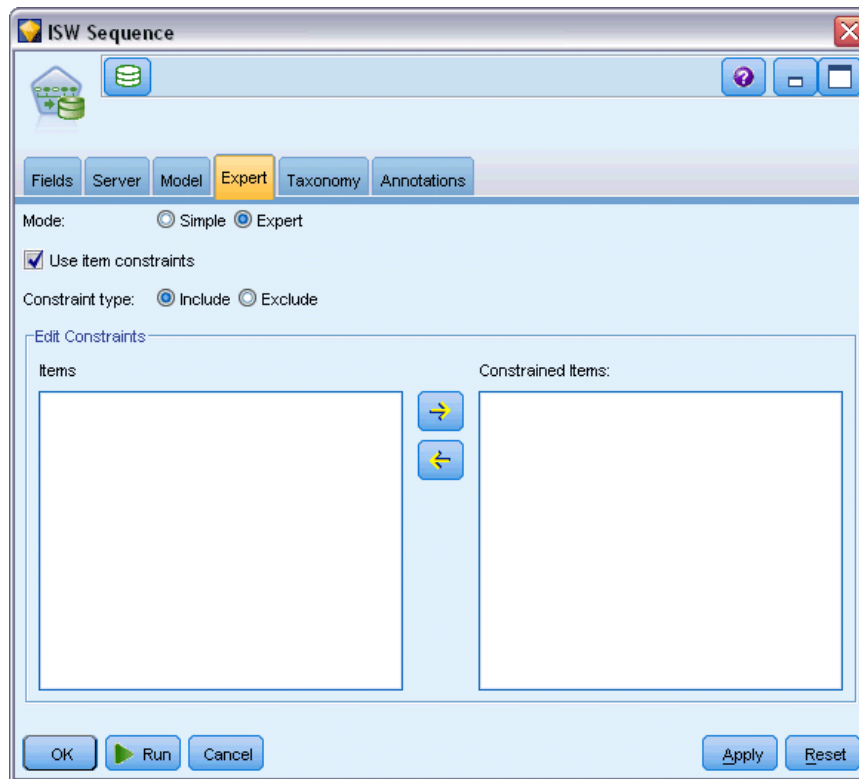
When Use item constraints is selected, any items you have added to the constraints list will be included or excluded from the results, depending on your setting for Constraint type.

Constraint type. Choose whether you want to include or exclude from the results those association rules containing the specified items.

Edit Constraints. To add an item to the list of constrained items, select it in the Items list and click the right-arrow button.

ISW Regression

The ISW Regression node supports the following regression algorithms:

- Transform (default)
- Linear
- Polynomial
- RBF

Transform regression

The ISW transform regression algorithm builds models that are decision trees with regression equations at the tree leaves. Note that IBM's Visualizer will not display the structure of these models.

The IBM® SPSS® Modeler browser shows the settings and annotations. However, the model structure cannot be browsed. There are relatively few user-configurable build settings.

Linear regression

The ISW linear regression algorithm assumes a linear relationship between the explanatory fields and the target field. It produces models that represent equations. The predicted value is expected to differ from the observed value, because a regression equation is an approximation of the target field. The difference is called residual.

InfoSphere Warehouse Data Mining modeling recognizes fields that do not have an explanatory value. To determine whether a field has an explanatory value, the linear regression algorithm performs statistical tests in addition to the autonomic variable selection. If you know the fields that do not have an explanatory value, you can automatically select a subset of the explanatory fields for shorter run times.

The linear regression algorithm provides the following methods to automatically select subsets of explanatory fields:

Stepwise regression. For stepwise regression, you must specify a minimum significance level. Only the fields that have a significance level above the specified value are used by the linear regression algorithm.

R-squared regression. The R-squared regression method identifies an optimum model by optimizing a model quality measure. One of the following quality measures are used:

- The squared Pearson correlation coefficient
- The adjusted squared Pearson correlation coefficient.

By default, the linear regression algorithm automatically selects subsets of explanatory fields by using the adjusted squared Pearson correlation coefficient to optimize the quality of the model.

Polynomial regression

The ISW polynomial regression algorithm assumes a polynomial relationship. A polynomial regression model is an equation that consists of the following parts:

- The maximum degree of polynomial regression
- An approximation of the target field
- The explanatory fields.

RBF regression

The ISW RBF regression algorithm assumes a relationship between the explanatory fields and the target field. This relationship can be expressed as a linear combination of Gaussian functions. Gaussian functions are specific Radial Basis Functions.

ISW Regression Model Options

On the Model tab of the ISW Regression node, you can specify the type of regression algorithm to use, as well as:

- Whether to use partitioned data
- Whether to perform a test run
- A limit for the R^2 value
- A limit for execution time

Figure 5-20
ISW Regression node Model tab
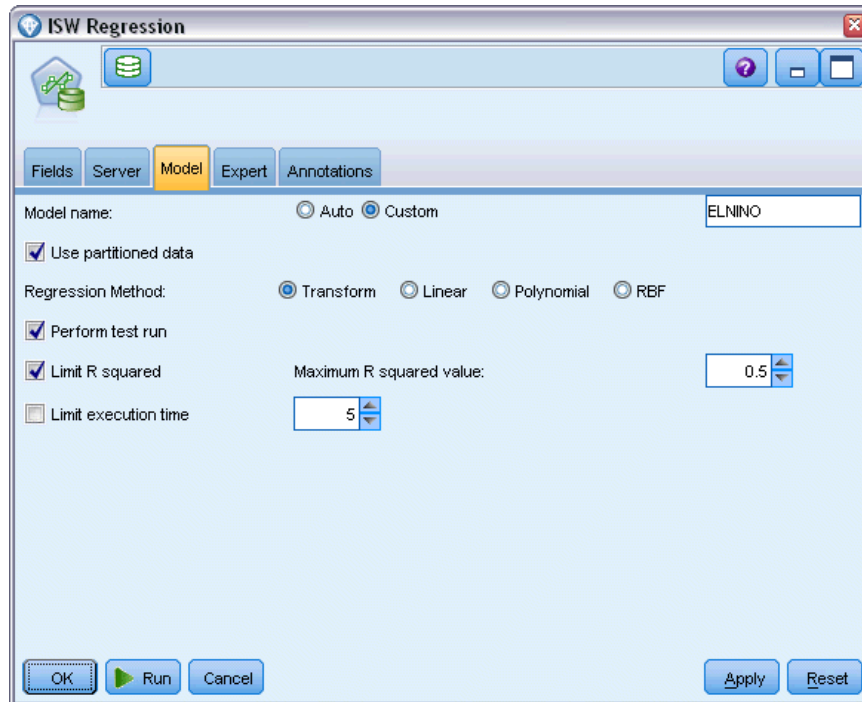
Use partitioned data. If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

Regression method. Choose the type of regression you want to perform. For more information, see the topic ISW Regression on p. 124.

Perform test run. You can choose to perform a test run. An InfoSphere Warehouse Data Mining test run is then executed after the model is built on the training partition. This performs a pass over the test partition to establish model quality information, lift charts, and so on.

Limit R squared. This option specifies the maximum tolerated systematic error (the squared Pearson correlation coefficient, R^2). This coefficient measures the correlation between the prediction error on verification data and the actual target values. It has a value between 0 (no correlation) and 1 (perfect positive or negative correlation). The value you define here sets the upper limit for the acceptable systematic error of the model.

Limit execution time. Specify a desired maximum execution time in minutes.

ISW Regression Expert Options

On the Expert tab of the ISW Regression node, you can specify a number of advanced options for linear, polynomial, or RBF regression.

Expert options for linear or polynomial regression

Figure 5-21
ISW Regression node Expert tab for linear or polynomial regression

Limit degree of polynomial. Sets the maximum degree of polynomial regression. If you set the maximum degree of polynomial regression to 1, the polynomial regression algorithm is identical to the linear regression algorithm. If you specify a high value for the maximum degree of polynomial regression, the polynomial regression algorithm tends to overfit. This means that the resulting model accurately approximates the training data, however, it fails when it is applied to data that is not used for training.

Use intercept. When enabled, forces the regression curve to go through the origin. This means that the model will not contain a constant term.

Use automatic feature selection. When enabled, the algorithm tries to determine an optimal subset of possible predictors if you do not specify a minimum significance level.

Use minimum significance level. When a minimum significance level is specified, Stepwise Regression is used to determine a subset of possible predictors. Only independent fields whose significance is above the specified value contribute to the calculation of the regression model.

Field Settings. To specify options for individual input fields, click the corresponding row in the Settings column of the Field Settings table, and choose <Specify Settings>. For more information, see the topic Specifying Field Settings for Regression on p. 128.

Expert options for RBF regression

Figure 5-22
ISW Regression node Expert tab for RBF regression
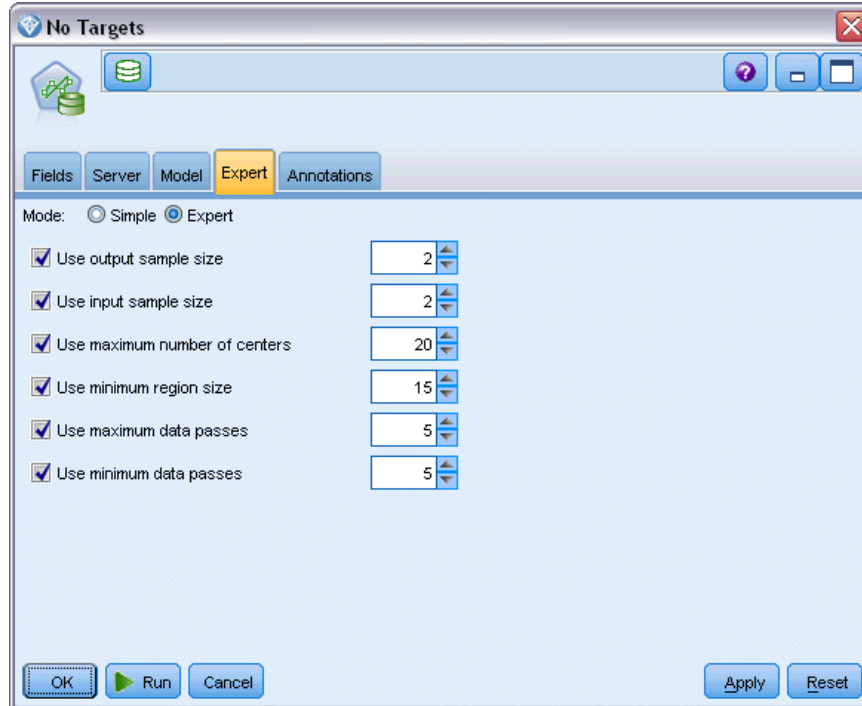
Use output sample size. Defines a 1-in-N sample for model verification and testing.

Use input sample size. Defines a 1-in-N sample for training.

Use maximum number of centers. The maximum number of centers that are built in each pass. Because the number of centers can increase up to twice the initial number during a pass, the actual number of centers might be higher than the number you specify.

Use minimum region size. The minimum number of records that are assigned to a region.

Use maximum data passes. The maximum number of passes through the input data made by the algorithm. If specified, this value must be greater than or equal to the minimum number of passes.

Use minimum data passes. The minimum number of passes through the input data made by the algorithm. Specify a large value only if you have enough training data and if you are sure that a good model exists.

Specifying Field Settings for Regression

Here you can specify the value range for an individual input field.

Figure 5-23
Specifying regression settings for an input field
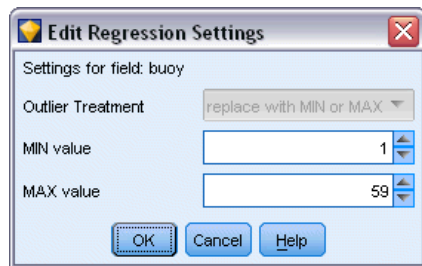
MIN value. The minimum valid value for this input field.

MAX value. The maximum valid value for this input field.

ISW Clustering

The Clustering mining function searches the input data for characteristics that most frequently occur in common. It groups the input data into clusters. The members of each cluster have similar properties. There are no preconceived notions of what patterns exist within the data. Clustering is a discovery process.

The ISW Clustering node gives you the choice of the following clustering methods:

- demographic
- Kohonen

The **demographic clustering** algorithm's technique is distribution-based. Distribution-based clustering provides fast and natural clustering of very large databases. The number of clusters is chosen automatically (you can specify the maximum number of clusters). There is a large number of user-configurable parameters.

The **Kohonen clustering** algorithm's technique is center based. The Kohonen feature map tries to put the cluster centers in places that minimize the overall distance between records and the cluster centers. The separability of clusters is not taken into account. The center vectors are arranged in a map with a certain number of columns and rows. These vectors are interconnected so that not only the winning vector that is closest to a training record is adjusted but also the vectors in its neighborhood. However, the further away the other centers are, the less they are adjusted.

ISW Clustering Model Options

On the Model tab of the Clustering node, you can specify the method to use to create the clusters, together with some related options.

Figure 5-24
ISW Clustering node Model tab
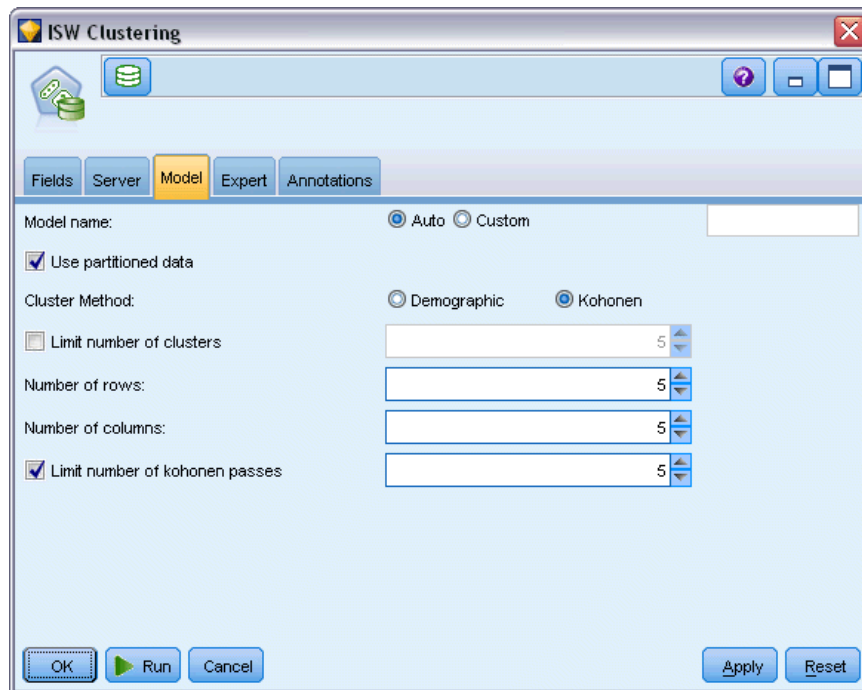
Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

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.

Cluster Method. Choose the method you want to use to create the clusters: Demographicor Kohonen. For more information, see the topic ISW Clustering on p. 129.

Limit number of clusters. Limiting the number of clusters saves run time by preventing the production of many small clusters.

Number of rows/Number of columns. (Kohonen method only) Specifies the number of rows and columns for the Kohonen feature map. (Available only if Limit number of kohonen passes is selected and Limit number of clusters is deselected.)

Max number of passes. (Kohonen method only) Specifies the number of passes the clustering algorithm makes over the data during training runs. With each pass, the center vectors are adjusted to minimize the total distance between cluster centers and records. Also, the amount by which the vectors are adjusted is decreased. In the first pass, the adjustments are rough. In the final pass, the amount by which the centers are adjusted is rather small. Only minor adjustments are done.

ISW Clustering Expert Options

On the Expert tab of the Clustering node, you can specify advanced options such as similarity thresholds, execution time limits, and field weights.

Figure 5-25
ISW Clustering node Expert tab
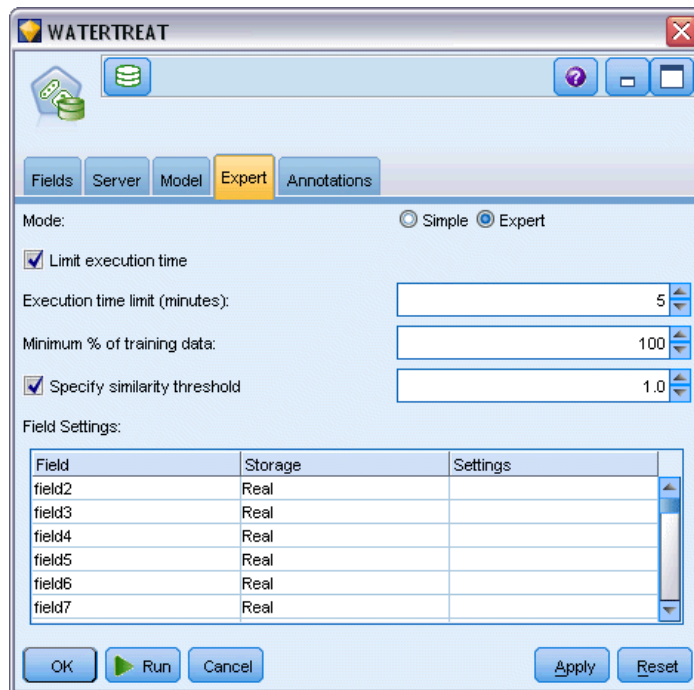
Limit execution time. Check this box to enable options that allow you to control the time taken to create the model. You can specify a time in minutes, a minimum percentage of training data to be processed, or both.

Specify similarity threshold. (Demographic clustering only) The lower limit for the similarity of two data records that belong to the same cluster. For example, a value of 0.25 means that records with values that are 25% similar are likely to be assigned to the same cluster. A value of 1.0 means that records must be identical in order to appear in the same cluster.

Field Settings. To specify options for individual input fields, click the corresponding row in the Settings column of the Field Settings table, and choose <Specify Settings>.

Specifying Field Settings for Clustering

Here you can specify options for individual input fields.

Figure 5-26
Specifying cluster settings for an input field
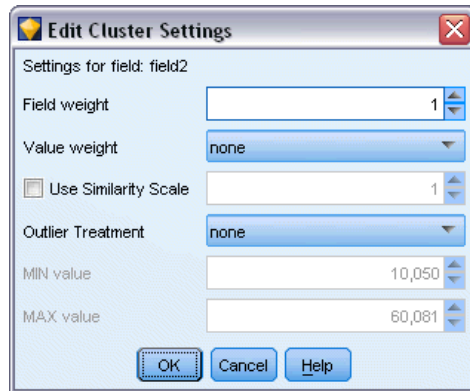
Field weight. Assigns more or less weight to the field during the model building process. For example, if you believe that this field is relatively less important to the model than the other fields, decrease its field weight relative to the other fields.

Value weight. Assigns more or less weight to particular values of this field. Some field values might be more common than other values. The coincidence of rare values in a field might be more significant to a cluster than the coincidence of frequent values. You can choose one of the following methods to weight values for this field (in either case, rare values carry a large weight, while common values carry a small weight):

- **Logarithmic.** Assigns a weight to each value according to the logarithm of its probability in the input data.

- **Probabilistic.** Assigns a weight to each value according to its probability in the input data.

For either method, you can also choose a with compensation option to compensate for the value weighting applied to each field. If you compensate for value weighting, the overall importance of the weighted field is equal to that of an unweighted field. This is so regardless of the number of possible values. Compensated weighting affects only the relative importance of coincidences within the set of possible values.

    Use similarity scale. Check this box if you want to use a similarity scale to control the calculation of the similarity measurement for a field. You specify the similarity scale as an absolute number. The specification is considered only for active numerical fields. If you do not specify a similarity scale, the default value (half of the standard deviation) is used. To obtain a larger number of clusters, decrease the mean similarity between pairs of clusters by smaller similarity scales for numerical fields.

Outlier Treatment. Outliers are field values that lie outside the range of values specified for the field, as defined by MIN value and MAX value. You can choose how you want outlier values to be handled for this field.

- The default, none, means that no special action is taken for outlier values.

■ If you choose replace with MIN or MAX, a field value less than MIN value or greater than MAX value is replaced with the values of MIN or MAX as appropriate. You can set the values of MIN and MAX in this case.

■ If you choose treat as missing, outliers are treated as missing values and disregarded. You can set the values of MIN and MAX in this case.

ISW 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.

The ISW Naive Bayes classification algorithm is a probabilistic classifier. It is based on probability models that incorporate strong independence assumptions.

ISW Naive Bayes Model Options

Figure 5-27
ISW Naive Bayes node Model tab
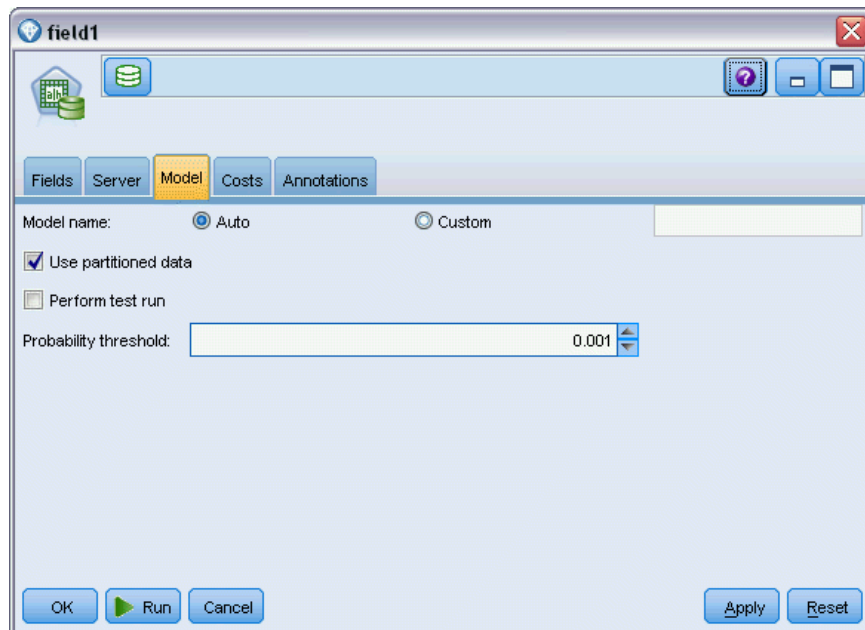
Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

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.

Perform test run. You can choose to perform a test run. An InfoSphere Warehouse Data Mining test run is then executed after the model is built on the training partition. This performs a pass over the test partition to establish model quality information, lift charts, and so on.

Probability threshold. The probability threshold defines a probability for any combinations of predictor and target values that are not seen in the training data. This probability should lie between 0 and 1. The default value is 0.001.

ISW Logistic Regression

Logistic regression, also known as nominal regression, is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression, but the ISW Logistic Regression algorithm takes a flag (binary) target field instead of a numeric one.

ISW Logistic Regression Model Options

Figure 5-28
ISW Logistic Regression node Model tab
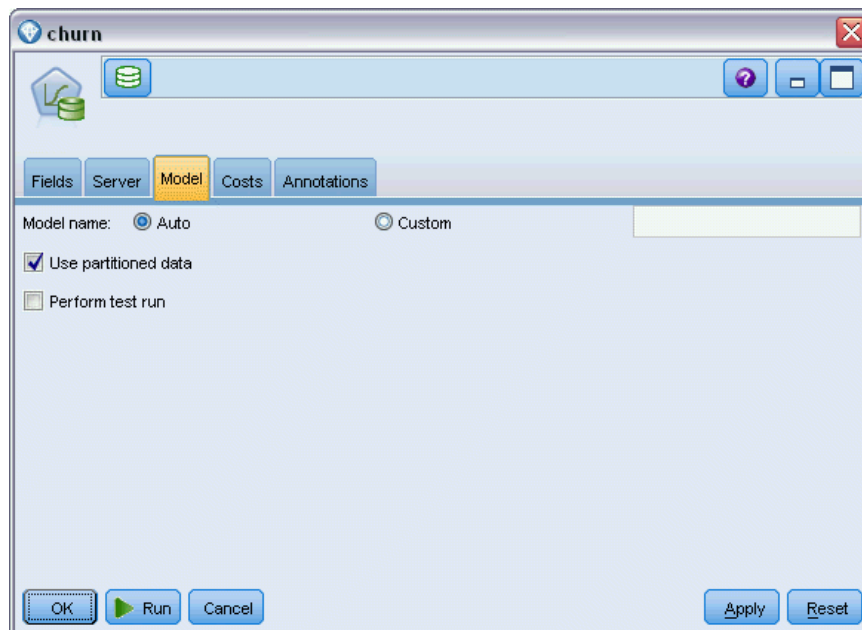
Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

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.

Perform test run. You can choose to perform a test run. An InfoSphere Warehouse Data Mining test run is then executed after the model is built on the training partition. This performs a pass over the test partition to establish model quality information, lift charts, and so on.

ISW Time Series

The ISW Time Series algorithms enable you to predict future events based on known events in the past.

Similar to common regression methods, time series algorithms predict a numerical value. In contrast to common regression methods, time series predictions are focused on future values of an ordered series. These predictions are commonly called forecasts.

The time series algorithms are univariate algorithms. This means that the independent variable is a time column or an order column. The forecasts are based on past values. They are not based on other independent columns.

Time series algorithms are different from common regression algorithms because they do not only predict future values but also incorporate seasonal cycles into the forecast.

The Time Series mining function provides the following algorithms to predict future trends:

- Autoregressive Integrated Moving Average (ARIMA)
- Exponential Smoothing
- Seasonal Trend Decomposition

The algorithm that creates the best forecast of your data depends on different model assumptions. You can calculate all forecasts at the same time. The algorithms calculate a detailed forecast including seasonal behavior of the original time series. If you have the IBM InfoSphere Warehouse Client installed, you can use the Time Series Visualizer to evaluate and compare the resulting curves.

ISW Time Series Fields Options

Figure 5-29
ISW Time Series node Fields tab
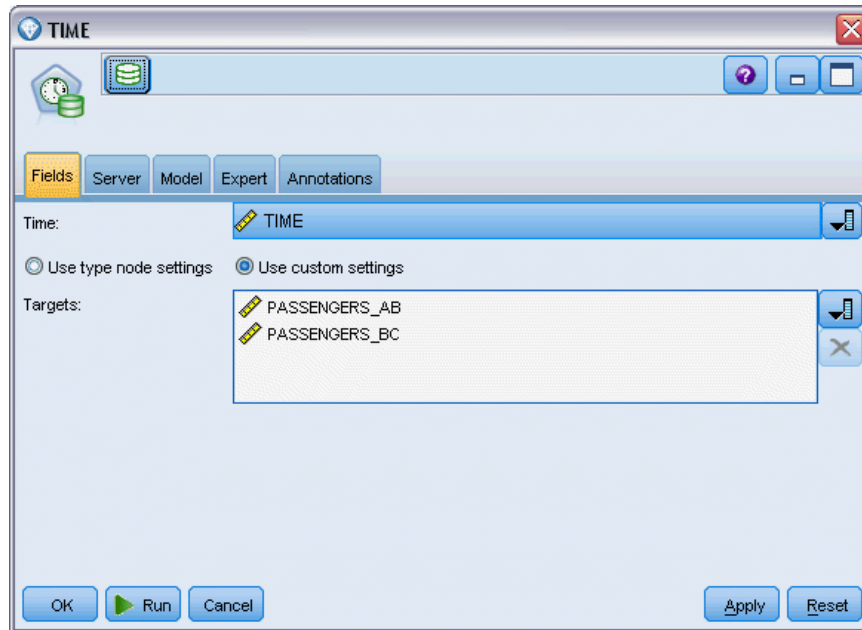
Time. Select the input field that contains the time series. This must be a field with a storage type of Date, Time, Timestamp, Real or Integer.

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 fields below as required.

Targets. Select one or more target fields. This is similar to setting the field role to *Target* in a Type node.

ISW Time Series Model Options

Figure 5-30
ISW Time Series node Model tab
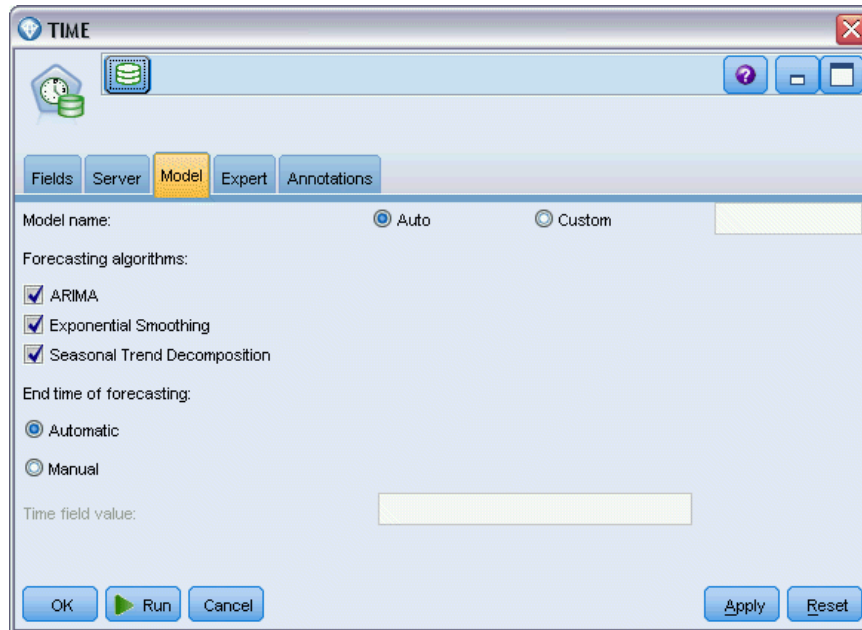
Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

Forecasting algorithms. Select the algorithms to be used for modeling. You can choose one or a mixture of the following:

■ ARIMA

■ Exponential smoothing

■ Seasonal trend decomposition.

End time of forecasting. Specify if the forecast end time is either to be calculated automatically or specified manually.

Time field value. When the End time of forecasting is set to manual, enter the forecasting end time. The value you can enter depends on the time field's type; for example, if the type is an integer representing hours then you may enter 48 to stop forecasting after 48 hours' data is processed. Alternatively, this field may prompt you to enter either a date or time as the end value.

ISW Time Series Expert Options

Figure 5-31
ISW Time Series node Expert tab
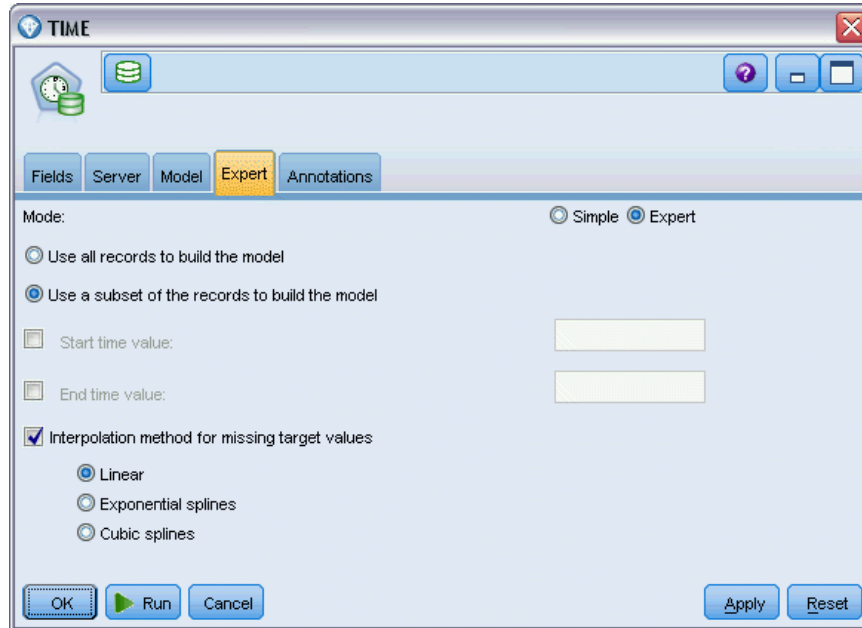
Use all records to build the model. This is the default setting; all records are analyzed when the model is built.

Use a subset of the records to build the model. If you only want to create the model from a portion of the data available, select this option. For example, this might be necessary when you have an excessively large amount of repetitious data.

Enter the Start time value and End time value to identify the data to be used. Note that the values you can enter in these fields depend on the time field's type; for example, this may be a number of hours or days, or specific dates or times.

Interpolation method for missing target values. If you are processing data with one or more missing values, select the method to be used to calculate them. You can choose one of the following:

■ Linear

■ Exponential splines

■ Cubic splines

Displaying ISW Time Series Models

ISW Time Series models are output in the form of an unrefined model, which contains information extracted from the data but is not designed for generating predictions directly.

Figure 5-32
Unrefined model icon

If you have the IBM InfoSphere Warehouse Client installed, you can use the Time Series
Visualizer tool for a graphical display of your time series data.

Figure 5-33
ISW Time Series model displayed in the Visualizer
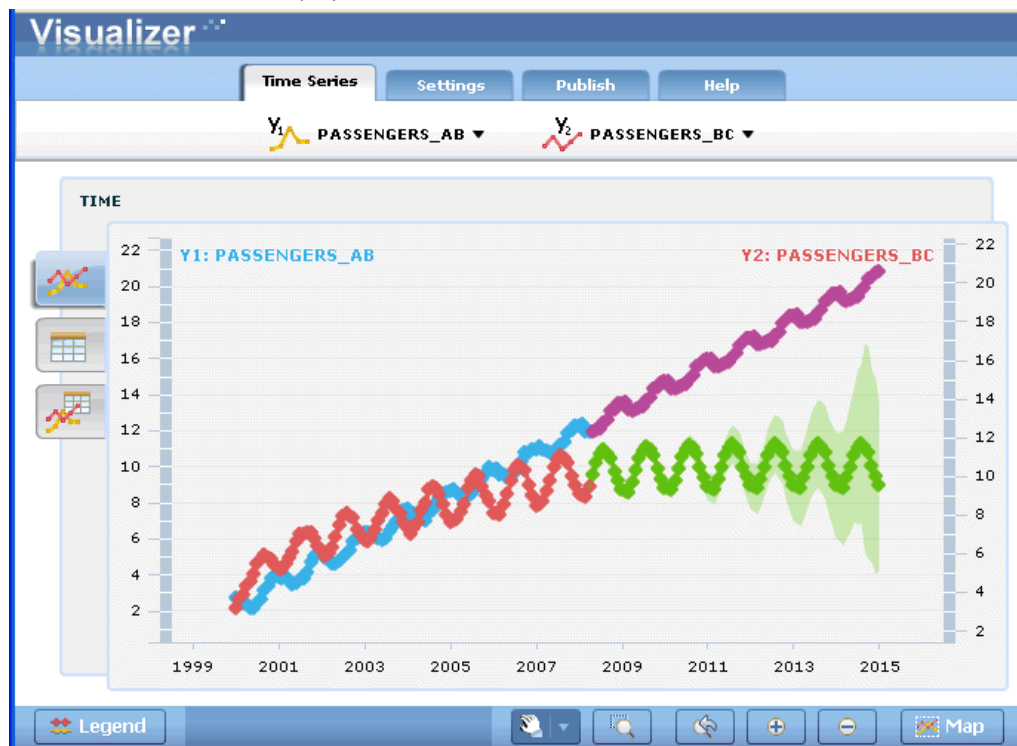
To use the Time Series Visualizer tool:

▶ Make sure that you have completed the tasks for integrating IBM® SPSS® Modeler with IBM
InfoSphere Warehouse. For more information, see the topic Enabling Integration with InfoSphere
Warehouse on p. 99.

▶ Double-click the unrefined model icon in the Models palette.

▶ On the Server tab of the dialog box, click the View button to display the Visualizer in your
default web browser.

ISW Data Mining Model Nuggets

You can create models from the ISW Decision Tree, Association, Sequence, Regression, and
Clustering nodes included with IBM® SPSS® Modeler.

ISW Model Nugget Server Tab

The Server tab provides options for performing consistency checks and launching the IBM Visualizer tool.

Figure 5-34
ISW model nugget Server tab
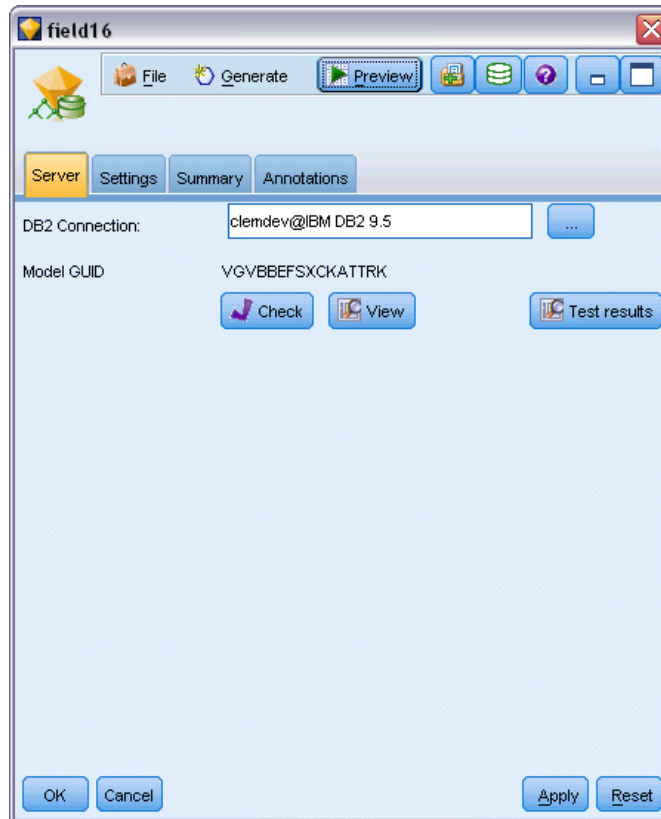
IBM® SPSS® Modeler can perform consistency checking by storing an identical generated model key string in both the SPSS Modeler model and the ISW model. Consistency checking is performed by clicking the Check button on the Server tab. For more information, see the topic Managing DB2 Models on p. 108.

The Visualizer tool is the only method for browsing InfoSphere Warehouse Data Mining models. The tool can be installed optionally with InfoSphere Warehouse Data Mining. For more information, see the topic Enabling Integration with InfoSphere Warehouse on p. 99.

■ Click View to launch the visualizer tool. What the tool displays depends on the generated node type. For example, the visualizer tool will return a Predicted Classes View when launched from an ISW Decision Tree model nugget.

■ Click Test Results (Decision Trees and Sequence only) to launch the visualizer tool and view the generated model's overall quality.

ISW Model Nugget Settings Tab

In IBM® SPSS® Modeler, generally, only a single prediction and associated probability or confidence is delivered. In addition, a user option to display probabilities for each outcome (similar to that found in logistic regression) is a score time option available on the model nugget Settings tab.

Figure 5-35
ISW model nugget Settings tab
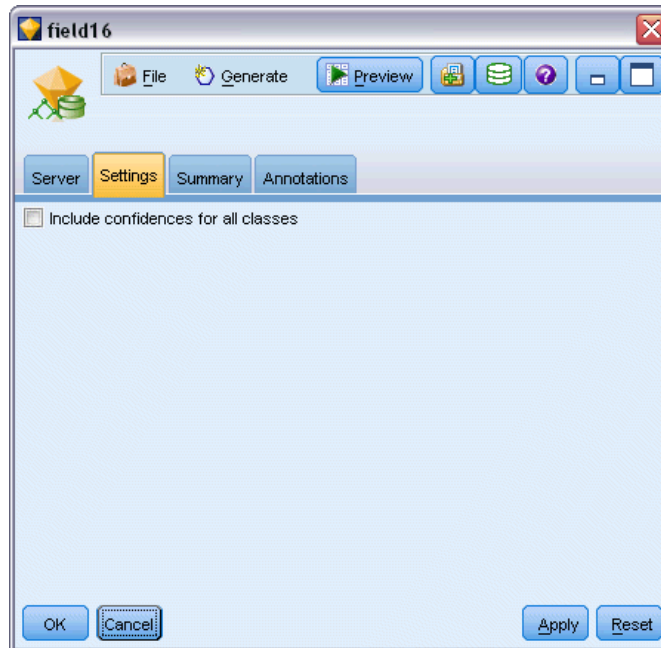
Include confidences for all classes. For each of the possible outcomes for the target field, adds a column giving the confidence level.

ISW Model Nugget Summary Tab

Figure 5-36
ISW 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.

ISW Data Mining Examples

IBM® SPSS® Modeler for Windows ships with a number of demo streams illustrating the process of database mining. These streams can be found in the IBM® SPSS® Modeler installation folder under:

\Demos\Database_Modeling\IBM DB2 ISW

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

The following streams can be used together in sequence as an example of the database mining process:

- *1_upload_data.str*—Used to clean and upload data from a flat file into DB2.
- *2_explore_data.str*—Used as an example of data exploration with SPSS Modeler.
- *3_build_model.str*—Used to build an ISW Decision Tree model.
- *4_evaluate_model.str*—Used as an example of model evaluation with SPSS Modeler.
- *5_deploy_model.str*—Used to deploy the model for in-database scoring.

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 following file in the SPSS Modeler installation folder under:

\Demos\Database_Modeling\IBM DB2 ISW\crx.names

This dataset is available from the UCI Machine Learning Repository at *http://archive.ics.uci.edu/ml/*.

Example Stream: Upload Data

The first example stream, *1_upload_data.str*, is used to clean and upload data from a flat file into DB2.

Figure 5-37
Example stream used to upload data
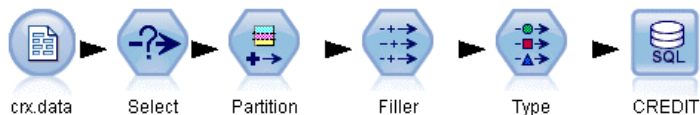
The Filler node is used for missing-value handling and replaces empty fields read from the text file *crx.data* with *NULL* values.

Example Stream: Explore Data

The second example stream, *2_explore_data.str*, is used to demonstrate data exploration in IBM® SPSS® Modeler.

Figure 5-38
Example stream used to explore data

A typical step used during data exploration is to attach a Data Audit node to the data. The Data Audit node is available from the Output nodes palette.

Figure 5-39
Data Audit results
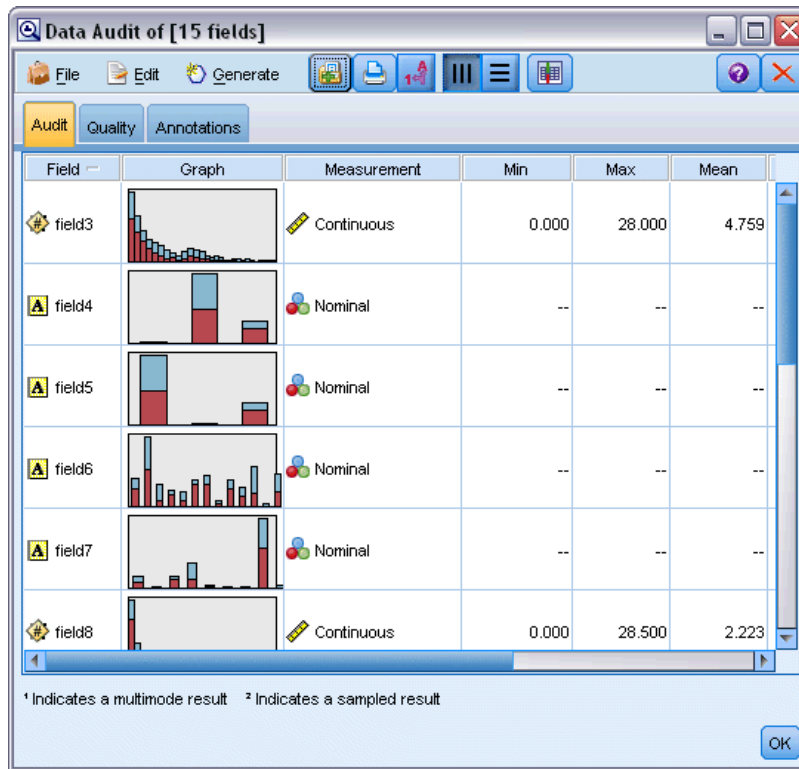
You can use the output from a Data Audit node to gain a general overview of fields and data distribution. Double-clicking a graph in the Data Audit window produces a more detailed graph for deeper exploration of a given field.

Figure 5-40
Histogram created by double-clicking a graph in the Data Audit window
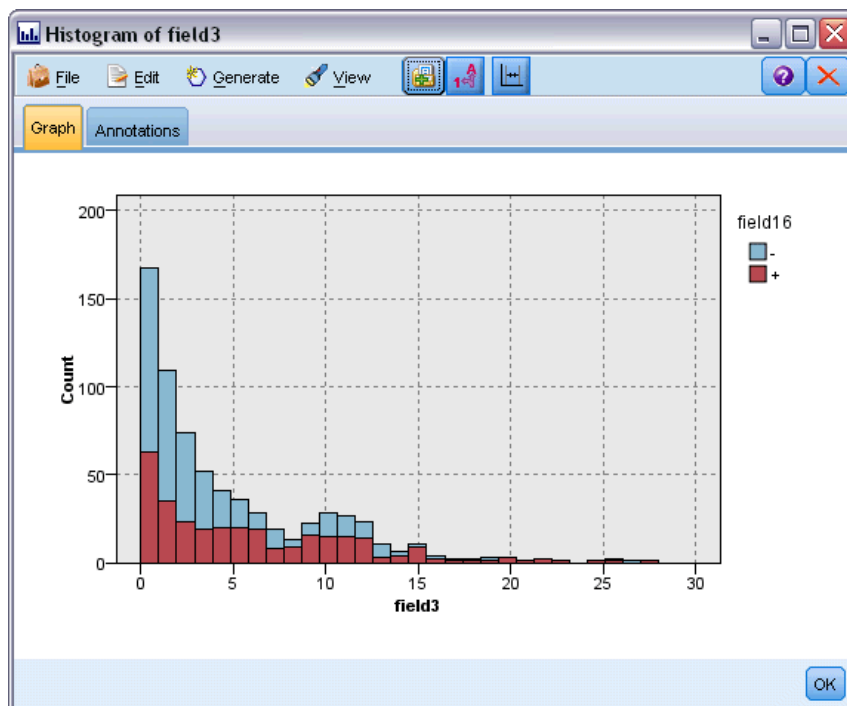
Example Stream: Build Model

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

Figure 5-41
Database modeling example stream, where purple-shaded nodes indicate in-database execution
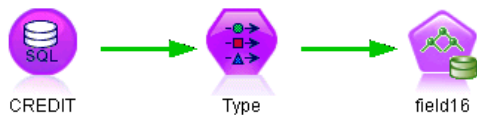
Using the Model and Expert tabs of the modeling node, you can adjust the maximum tree depth, and you can stop further splitting of a node from when the initial decision tree was built by setting the maximum purity and minimum cases per internal node. For more information, see the topic ISW Decision Tree on p. 114.

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 SPSS Modeler.

Figure 5-42
Example stream used for model evaluation
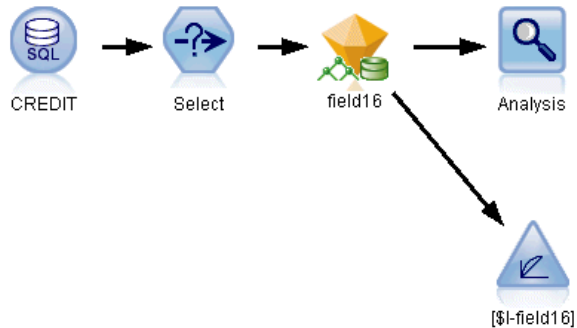
When you first open the stream, the model nugget (*field16*) is not included in the stream. Open the CREDIT source node and ensure that you have specified a data source. Next, provided that you have run the *3_build_model.str* stream to create a *field16* nugget in the Models palette, you can run the disconnected nodes by clicking the Run button on the toolbar (the button with a green triangle). This runs a script that copies the *field16* nugget into the stream, connects it to the existing nodes, and then runs the terminal nodes in the stream.

You can attach an Analysis node (available on the Output palette) to create a coincidence matrix showing the pattern of matches between each generated (predicted) field and its target field. Run the Analysis node to see the results.

Figure 5-43
Analysis node results
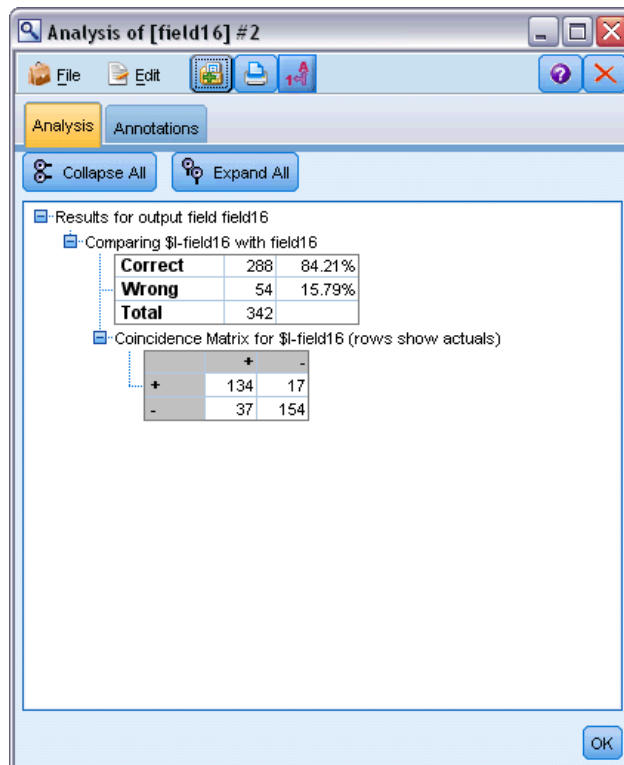
The generated table indicates that 84.21% of the predictions generated by the ISW Decision Tree algorithm were correct.

You can also create a gains chart to show accuracy improvements made by the model. Attach an Evaluation node to the generated model and then run the stream to see the results.

Figure 5-44
Gains chart generated using the Evaluation node
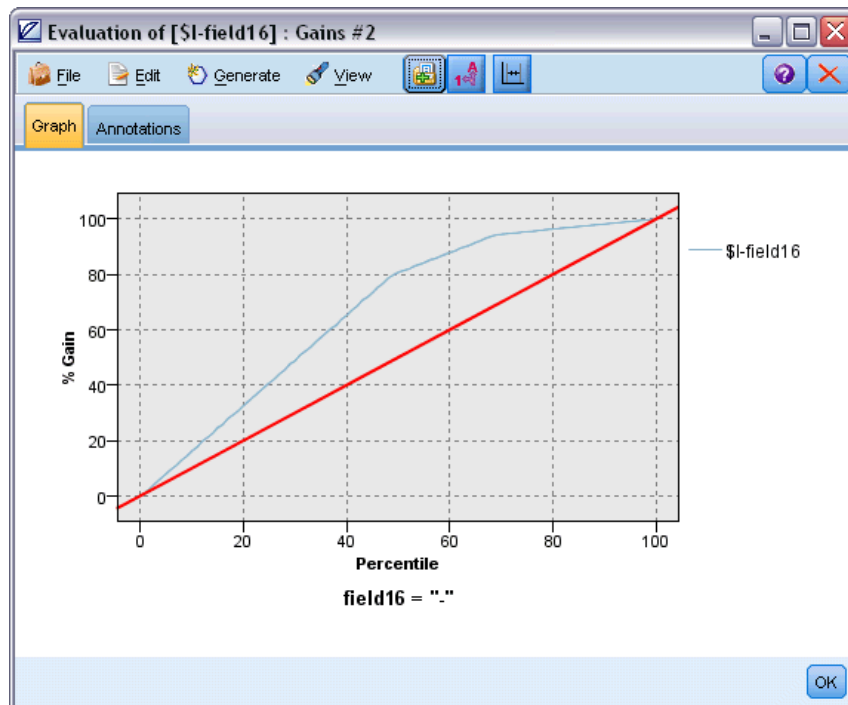
Example Stream: Deploy Model

Once you are satisfied with the accuracy of the model, you can deploy it for use with external applications or for writing scores back to the database. In the example stream *5_deploy_model.str*, data are read from the table CREDIT. When the *deploy solution* Database export node is run, the data are not actually scored. Instead, the stream creates the published image file *credit_scorer.pim* and the published parameter file *credit_scorer.par*.

Figure 5-45
Example stream used to deploy the model
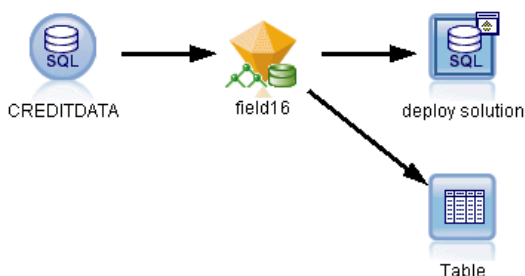
As in the previous example, the stream runs a script that copies the *field16* nugget into the stream from the Models palette, connects it to the existing nodes, and then runs the terminal nodes in the stream. In this case you must first specify a data source in both the Database source and export nodes.

Database Modeling with IBM Netezza Analytics

IBM SPSS Modeler and IBM Netezza Analytics

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

SPSS Modeler supports integration of the following algorithms from Netezza Analytics.

- Decision Trees
- K-Means
- Bayes Net
- Naive Bayes
- KNN
- Divisive Clustering
- PCA
- Regression Tree
- Linear Regression

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

Requirements for Integration with IBM Netezza Analytics

The following conditions are prerequisites for conducting in-database modeling using IBM® Netezza® Analytics. 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 (except zLinux, for which IBM Netezza ODBC drivers are not available).
- IBM Netezza Performance Server 6.0 or later, running the IBM® SPSS® In-Database Analytics package.
- An ODBC data source for connecting to an IBM Netezza database. For more information, see the topic Enabling Integration with IBM Netezza Analytics on p. 150.
- SQL generation and optimization enabled in SPSS Modeler. For more information, see the topic Enabling Integration with IBM Netezza Analytics on p. 150.

Note: Database modeling and SQL optimization require that 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 SPSS Modeler, and access SPSS Modeler Server. To verify the current license status, click About on the Help menu in SPSS Modeler.

Enabling Integration with IBM Netezza Analytics

Configuring IBM Netezza Analytics

To install and configure IBM® Netezza® Analytics, see the Netezza Analytics documentation—in particular, the *Netezza Analytics Installation Guide*—for more details. 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 (Netezza PCA and Netezza Linear Regression), the Netezza 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 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

▶ 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*.

▶ 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, selectData Sources (ODBC), then click Open.

- **Windows 7.** From the Start menu, choose Control Panel, then System & Security, then Administrative Tools. SelectData Sources (ODBC), then click Open.

▶ Click the System DSN tab, and then click Add.

▶ Select NetezzaSQL from the list and click Finish.

▶ 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.

▶ Click the Test Connection button and ensure that you can connect to the database.

▶ 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).

▶ From your *Netezza Client* CD, copy the relevant *<platform>cli.package.tar.gz* file to a temporary location on the server.

▶ Extract the archive contents by means of the gunzip and untar commands.

▶ Add execute permissions to the *unpack* script that is extracted.

▶ Run the script, answering the on-screen prompts.

▶ Edit the *modelersrv.sh* file to include the following lines.

```
. /usr/IBM/SPSS/SDAP61_notfinal/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/SDAP61_notfinal; export NZ_ODBC_INI_PATH
```

▶ Locate the file */usr/local/nz/lib64/odbc.ini* and copy its contents into the *odbc.ini* file that is installed with SDAP 6.1 (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
```

▶ Edit the parameters in the Netezza DSN definition to reflect the database to be used.

▶ Restart SPSS Modeler Server and test the use of the Netezza in-database mining nodes on the client.

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.

▶ From the SPSS Modeler menus choose:
Tools > Options > User Options

Figure 6-1
Optimization settings
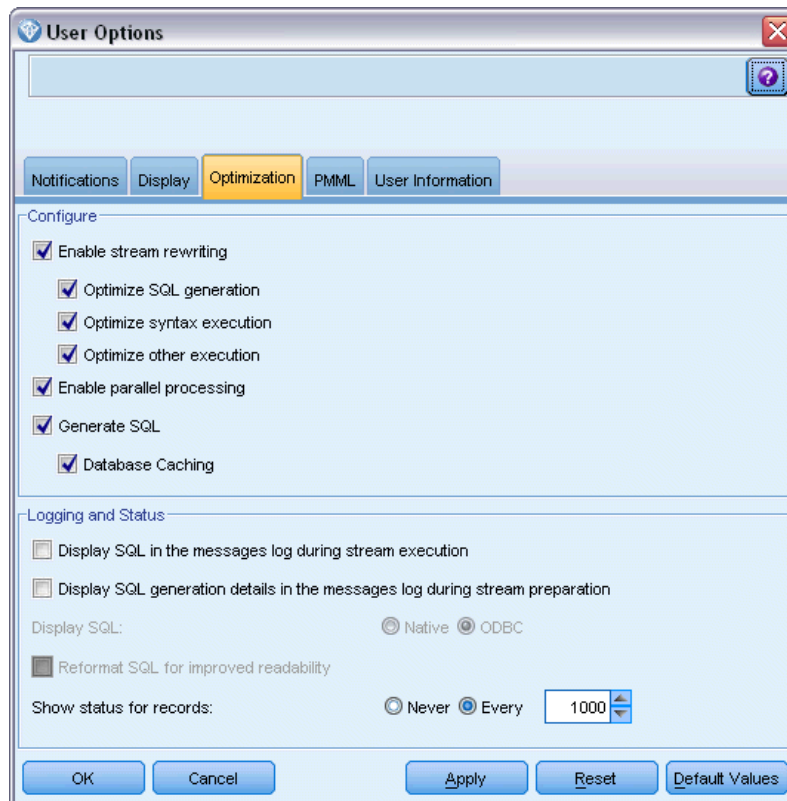
▶ Click the Optimization tab.

▶ Confirm that the Generate SQL option is enabled. This setting is required for database modeling to function.

▶ Select Optimize SQL Generation and Optimize other execution (not strictly required but strongly recommended for optimized performance).

Building Models with IBM Netezza Analytics

Each of the supported algorithms has a corresponding modeling node. You can access the 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.

▶ Select the source node.

▶ From the Field Ops tab on the nodes palette, double-click the Derive node.

▶ Open the Derive node by double-clicking its icon on the canvas.

▶ In the Derive field field, type (for example) ID.

▶ In the Formula field, type @INDEX and click OK.

▶ Connect the Derive node to the rest of the stream.

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.

▶ Attach a Select node to the source node.

▶ Set the Mode option of the Select node to Discard.

▶ Enter the following in the Condition field:

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

Be sure to include every input field.

▶ Connect the Select node to the rest of the stream.

General Comments

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

Netezza 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.

Figure 6-2
Example of Netezza field options

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.

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

Netezza Models - Server Options

On this tab, you specify the IBM Netezza database where the model is to be stored.

Figure 6-3
Example of Netezza server options

Netezza DB 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. *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 Netezza 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 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.

Table Name. The name of the database table where the model is to be stored. *Note*: This must be a new table; you cannot use an existing table for this operation.

Comments

■ The connection used for modeling does not have to be the same as the connection used in the source node for a stream. For example, you could have a stream that accesses data from one IBM Netezza database, downloads the data to IBM® SPSS® Modeler for cleaning or other manipulations, and then uploads the data to a different IBM Netezza database for modeling purposes. Note, however, that such a configuration can adversely affect performance.

■ 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.

Netezza Models - 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.

Figure 6-4
Example of Netezza model options
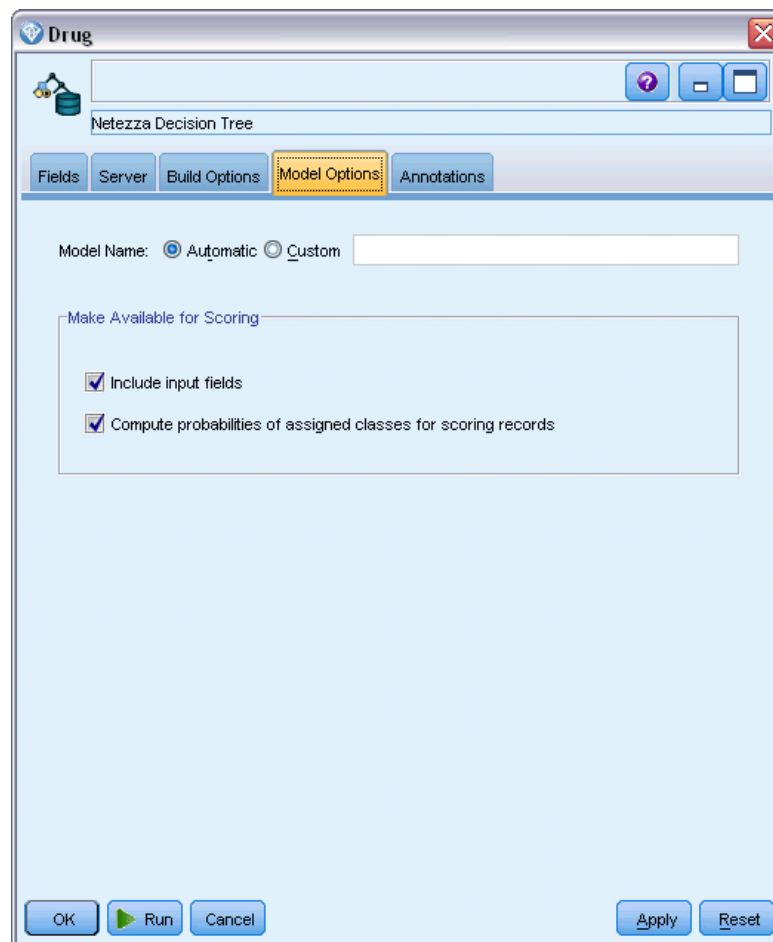
Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

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.

Netezza 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.

The model output takes the form of a textual representation of the tree. Each line of text corresponds to a node or a leaf, and the indentation reflects the tree level. For a node, the split condition is displayed; for a leaf, the assigned class label is shown.

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, for 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, for 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 following instance weights.

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.

Figure 6-5
Decision Tree field options

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. For more information, see the topic Instance Weights and Class Weights on p. 158.

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

Netezza Decision Tree 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.

Figure 6-6
Decision Tree build options for tree growth
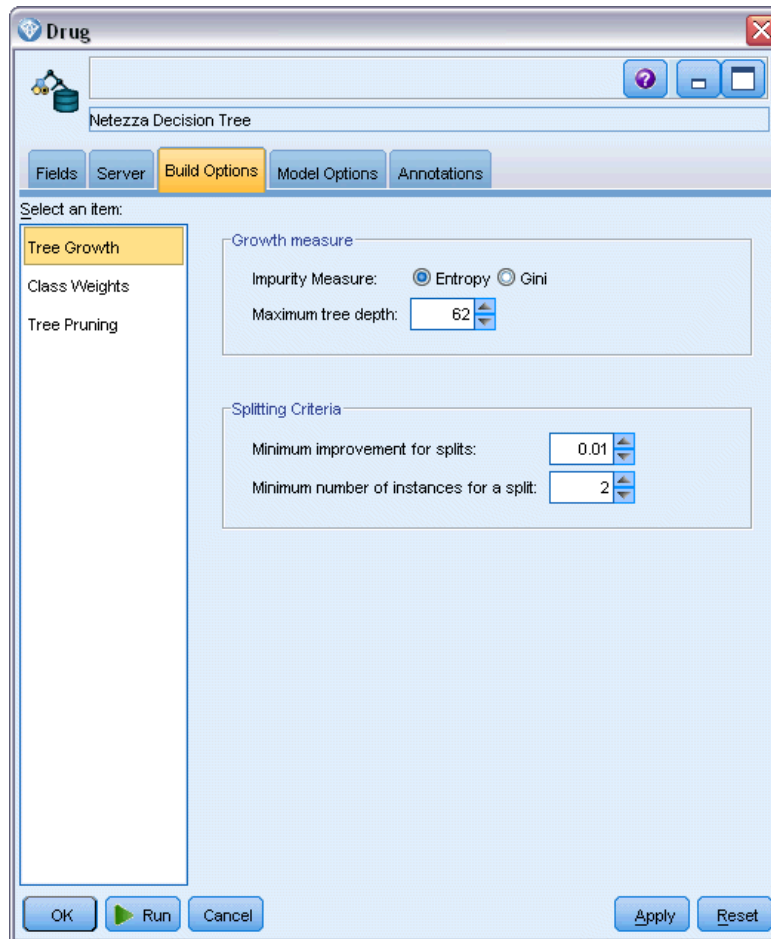
You can set build options for:

- Tree growth

- Weights for class labels
- Tree pruning

The options for tree growth are described in this section.

Growth Measure. These options control the way tree growth is measured. If you do not want to use the default values, click Customize and make the changes.

- **Impurity Measure.** The measurement of impurity, used to evaluate the best place to split the tree. **Impurity** refers to the extent to which subgroups defined by the tree have a wide range of output field values within each group.

  The supported measurements are **Entropy** (default) and **Gini**. These are two popular impurity measures that 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 will be split recursively). The default is 62, which is the maximum possible tree depth for modeling purposes. Note, however, that the viewer in the model nugget can display a maximum of 10 levels.

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

- **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—in other words, to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the amount specified by the splitting criteria, the split will not be made.

- **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 tree.

Netezza 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.

Figure 6-7
Decision Tree class weight options
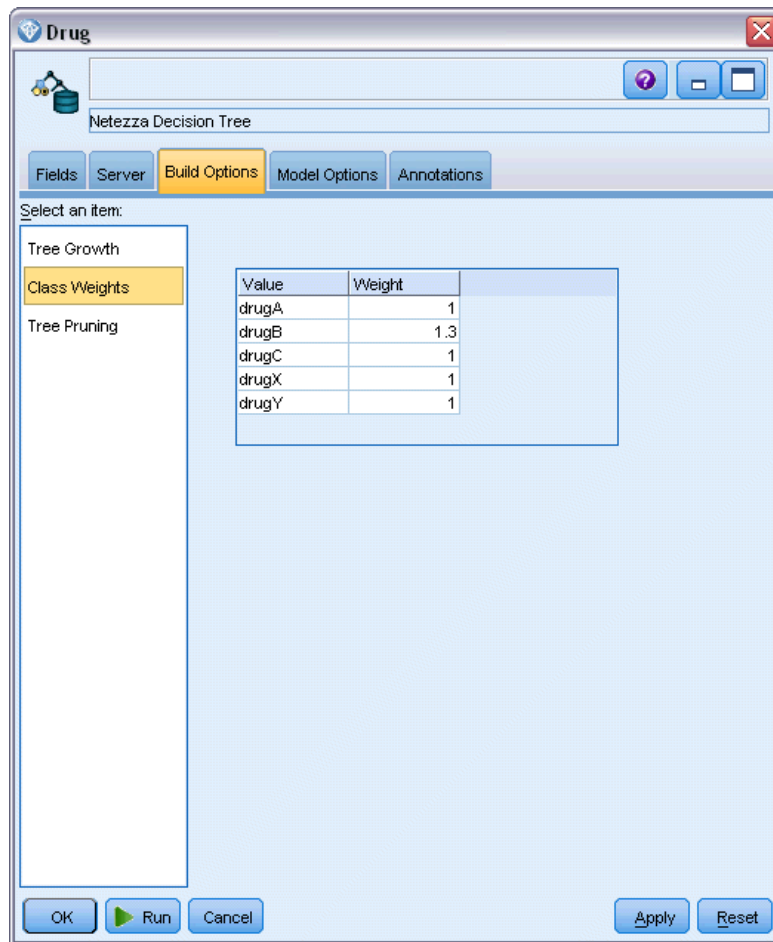
To change a weight, double-click it in the Weight column and make the changes you want.

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. For more information, see the topic Instance Weights and Class Weights on p. 158.

Netezza 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.

Figure 6-8
Decision Tree pruning options
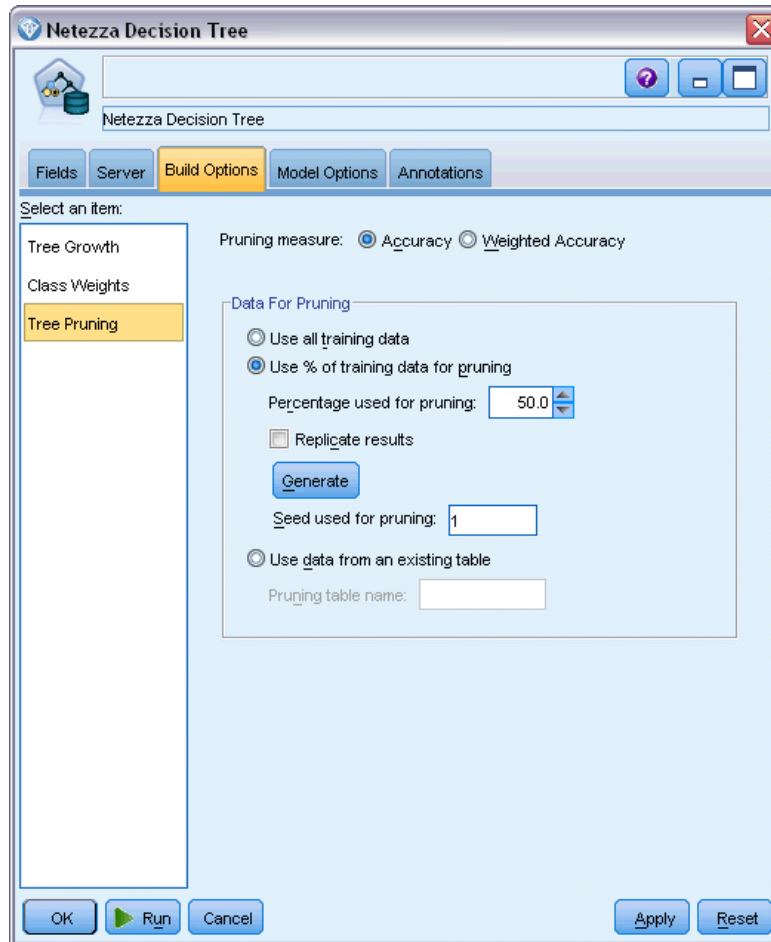
Pruning measure. The default pruning measure, Accuracy, ensures that the estimated accuracy of the model remains within acceptable limits after removing a leaf from the tree. Use the alternative, Weighted Accuracy, if you want to take the class weights into account while applying pruning.

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.

Netezza 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.

Netezza 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.

Figure 6-9
K-Means field options

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 K-Means 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.

Figure 6-10
K-Means build options
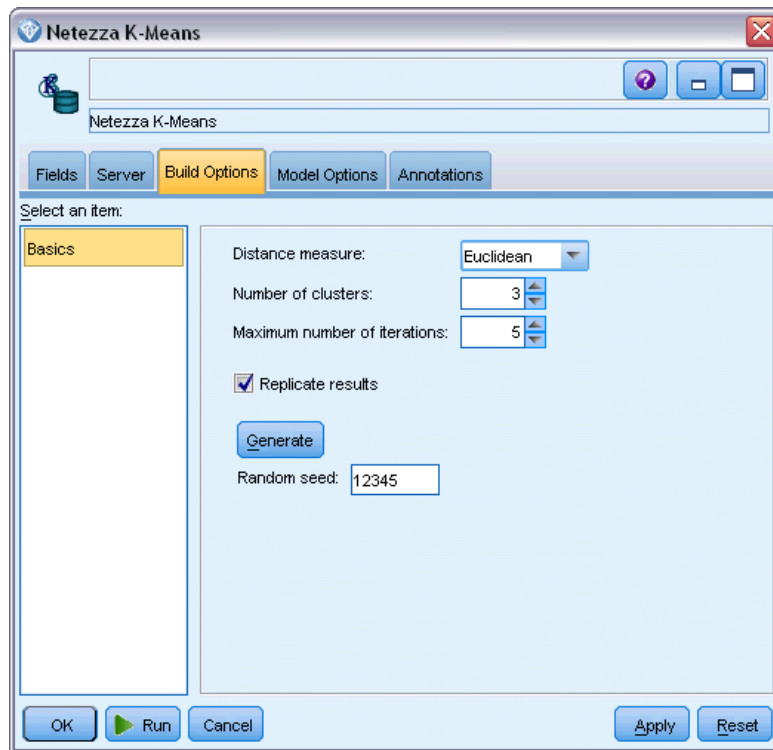
Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- **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 clusters (k). Specify the number of clusters to be created.

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.

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.

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. For more information, see the topic

Figure 6-11
Bayes Net field options
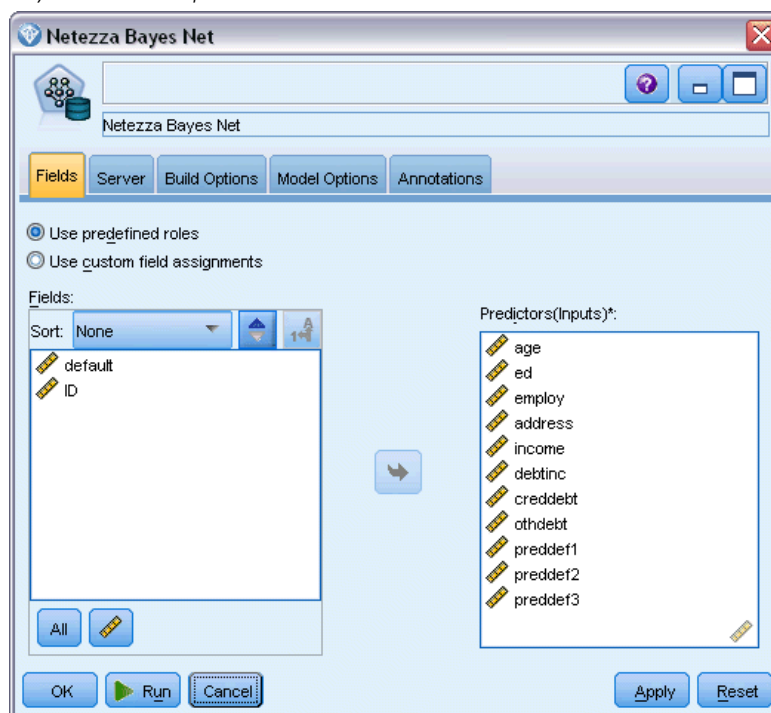
Use predefined roles. This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

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.

Figure 6-12
Bayes Net build options
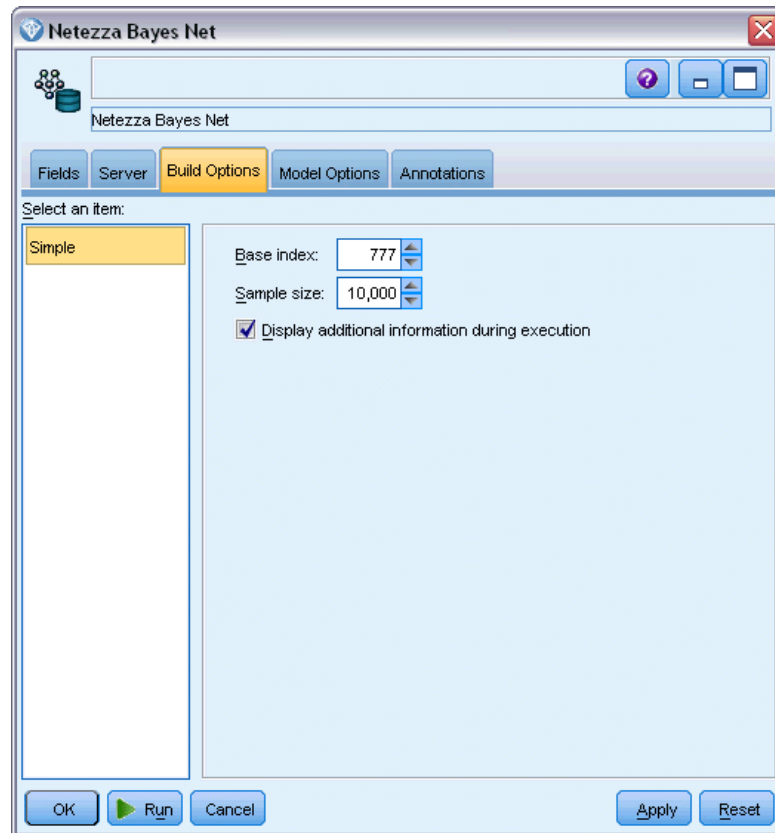
Base index. The numeric identifier to be assigned to the first attribute (input field) for easier internal management.

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 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 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*.

Figure 6-13
The effects of changing k on classification
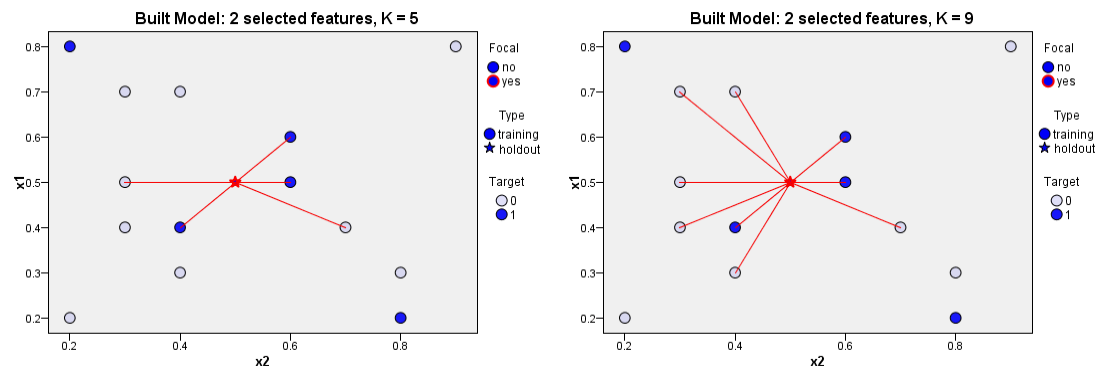
Nearest neighbor analysis can also be used to compute values for a continuous target. In this situation, the average or median target value of the nearest neighbors is used to obtain the predicted value for the new case.

Netezza 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.

Figure 6-14
KNN general model options
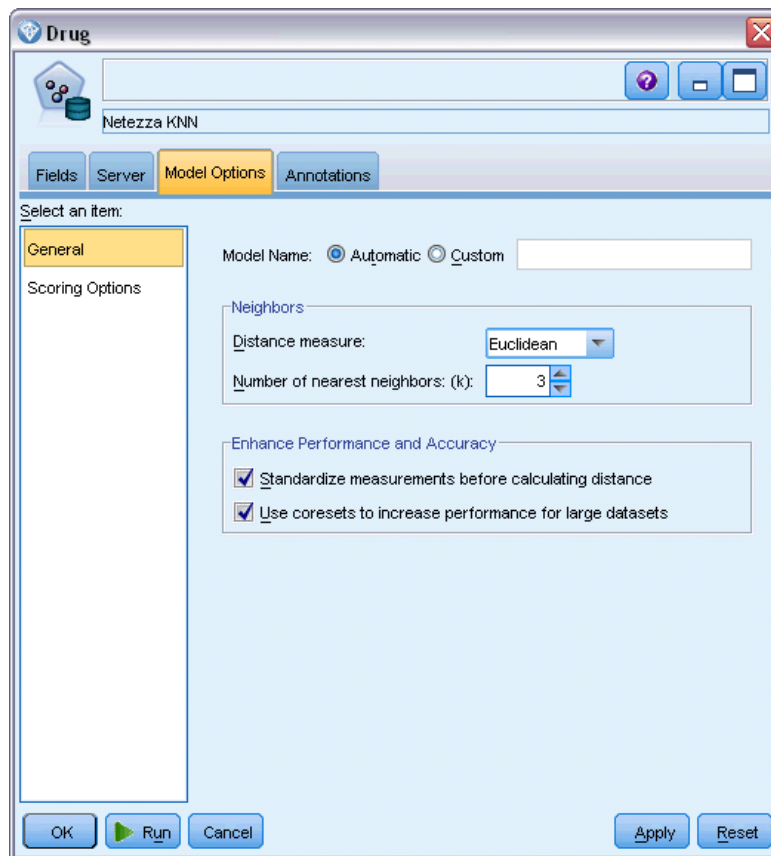
Model name. You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

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.

Netezza 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.

Figure 6-15
KNN general model options
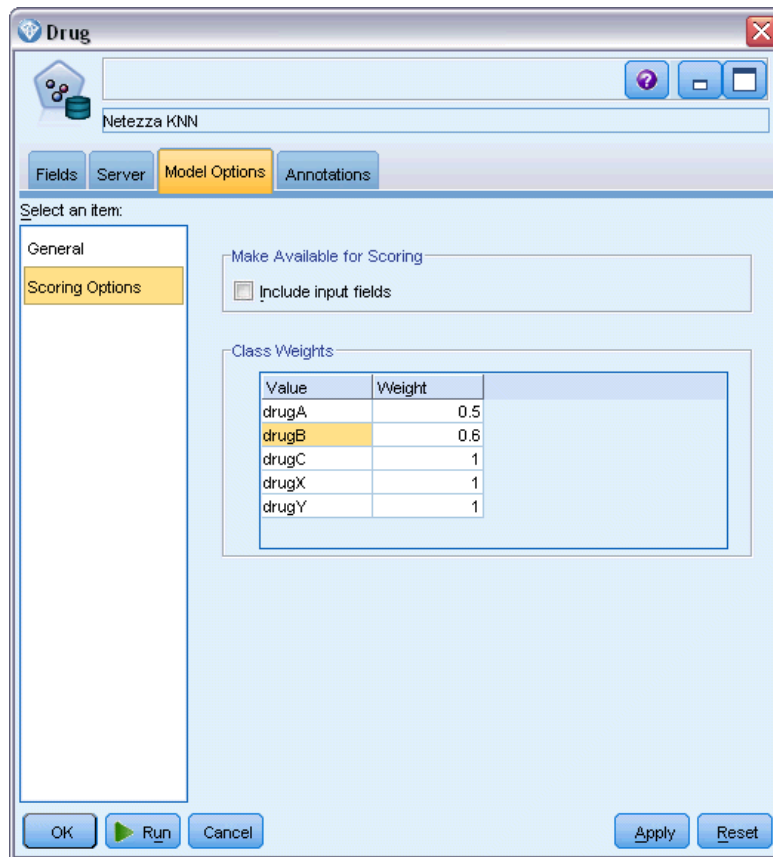
Make Available for Scoring

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.

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 6-16
Example of a divisive clustering tree
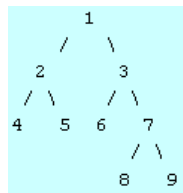
At each level, the best matching subcluster is chosen with respect to the distance of the instance from the subcluster centers.

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.

Figure 6-17
Divisive Clustering field options

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.

Figure 6-18
Divisive Clustering build options
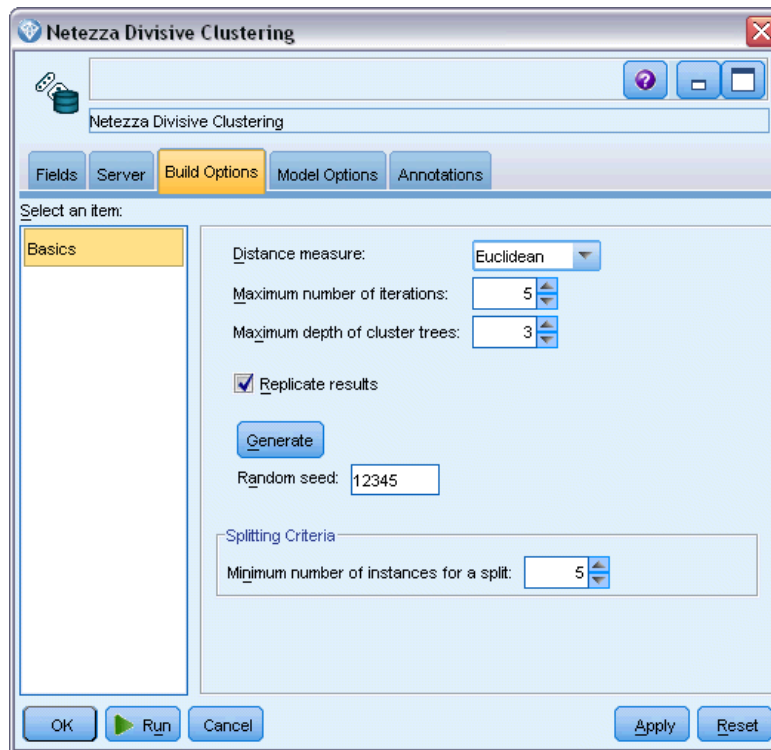
Distance measure. The method to be used for measuring the distance between data points; greater distances indicate greater dissimilarities. The options are:

- **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.

Netezza 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.

Netezza 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.

Figure 6-19
PCA field options

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 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.

Figure 6-20
PCA build options
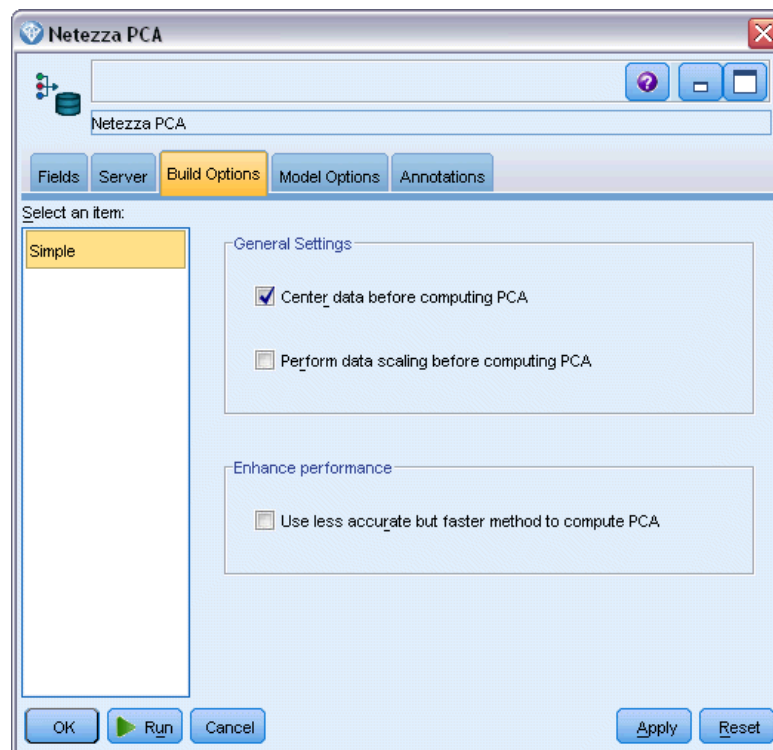
Center data before computing PCA. If checked (default), this option performs data centering (also known as "mean subtraction") before the analysis. Data centering is necessary to ensure that the first principal component describes the direction of maximum variance, otherwise the component might correspond 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.

Netezza 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.

The model output takes the form of a textual representation of the tree. Each line of text corresponds to a node or a leaf, and 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 Build Options - Tree Growth

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.

Figure 6-21
Regression Tree build options for tree growth
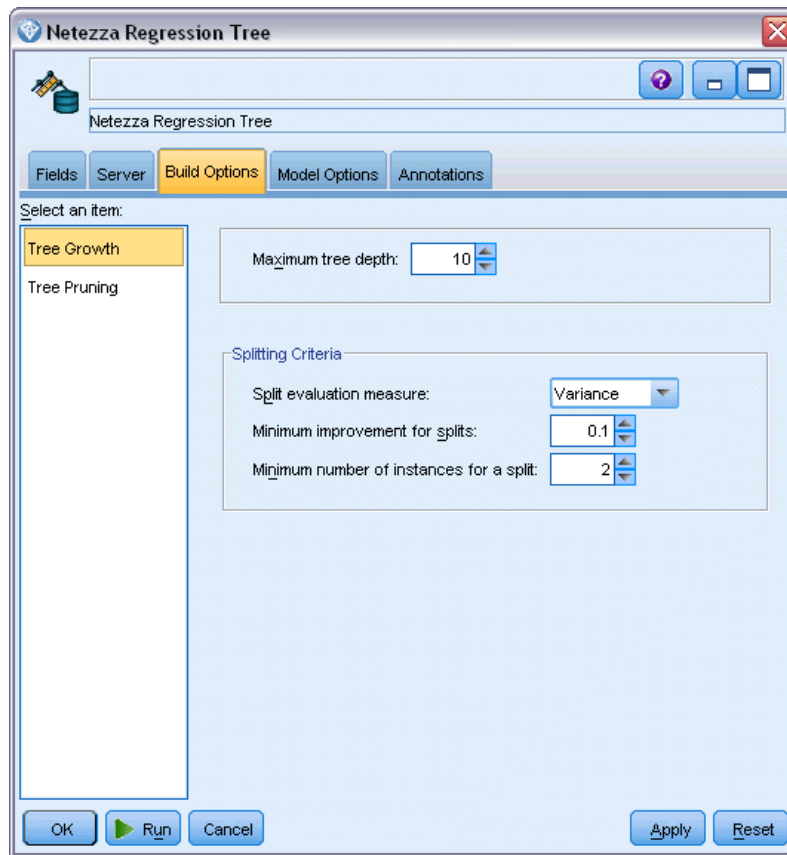
Maximum tree depth. The maximum number of levels to which the tree can grow below the root node (that is, the number of times the sample will be split recursively). The default is 62, which is the maximum possible tree depth for modeling purposes. Note, however, that the viewer in the model nugget can display a maximum of 12 levels.

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

- **Split evaluation measure**. The class impurity measure, used to evaluate 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—in other words, to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the amount specified by the splitting criteria, the split will not be made.

- **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 tree.

Netezza 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.

Figure 6-22
Regression Tree build options for tree pruning
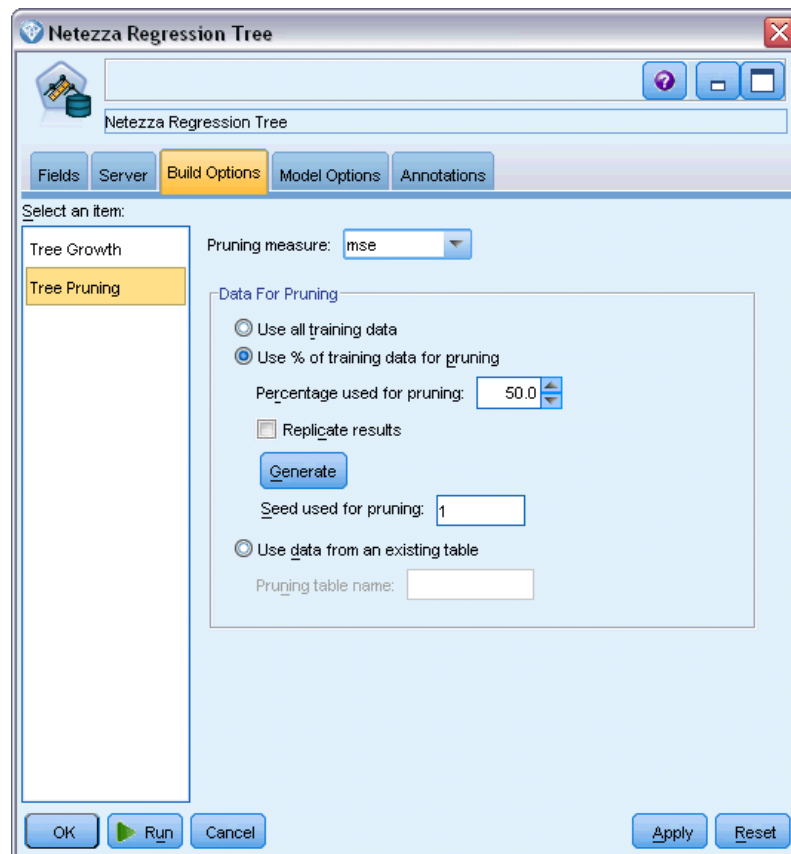
Pruning measure. The pruning measure ensures that the estimated accuracy of the model remains within acceptable limits after removing a leaf from the tree. You can select one of the following measures.

■ **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 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.

Netezza 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.

Figure 6-23
Linear Regression build options
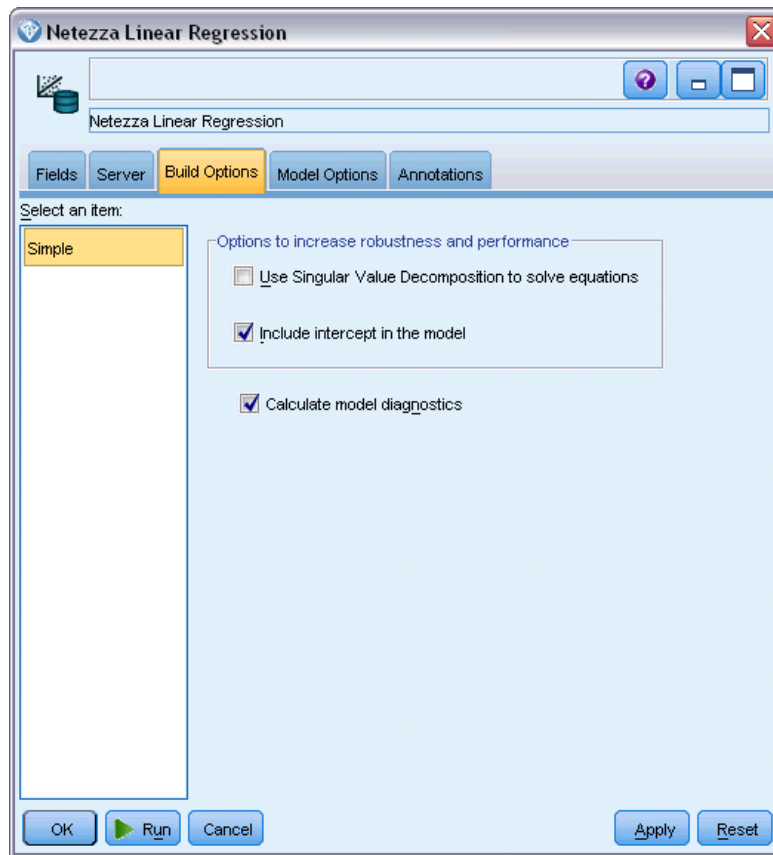
Use Singular Value Decomposition to solve equations. Using the Singular Value Decomposition matrix instead of the original matrix has the advantage of being more robust against numerical errors, and can also speed up computation.

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.

Managing IBM Netezza Analytics Models

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 Netezza Analytics model created in 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 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:

▶ Attach a Table node to the model nugget.

▶ Open the Table node.

▶ Click Run.

▶ Scroll to the right of the table output window to view the extra fields and their scores.

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.

Figure 6-24
Example of Netezza model nugget server options

Netezza DB 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. *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 Netezza 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 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.

Table Name. The name of the database table where the model is stored. This is for information only; you cannot change this name here.

Netezza 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 modeling node, by default the node adds one new field, the name of which is derived from the model name.

Table 6-1
Model-scoring field for Decision Tree
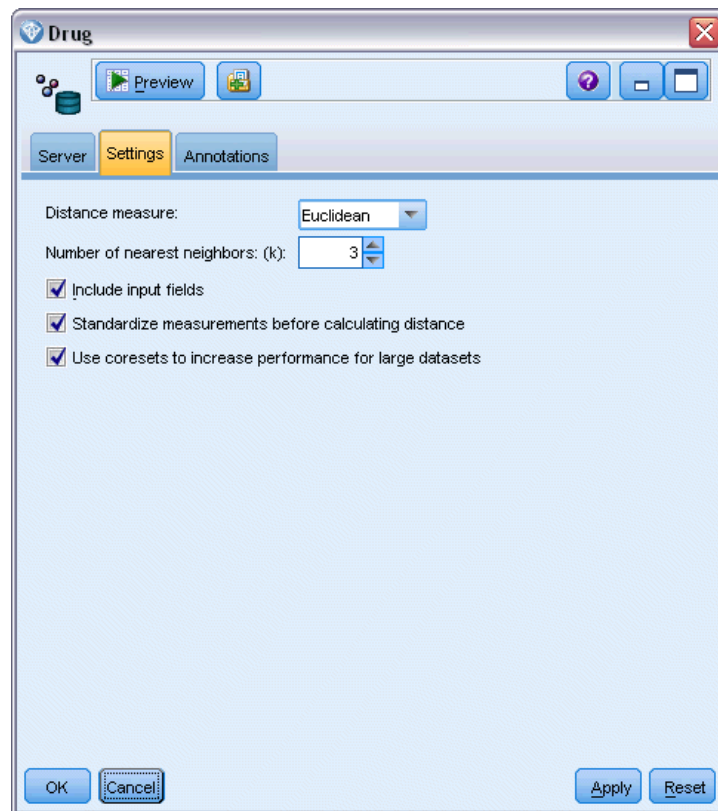
Name of Added Field	Meaning
$I-*model_name*	Predicted value for current record.

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 6-2
Model-scoring field for Decision Tree - additional

Name of Added Field	Meaning
$IP-*model_name*	Confidence value (from 0.0 to 1.0) for the prediction.

Netezza Decision Tree Nugget - Model Tab

Figure 6-25
Decision Tree model output
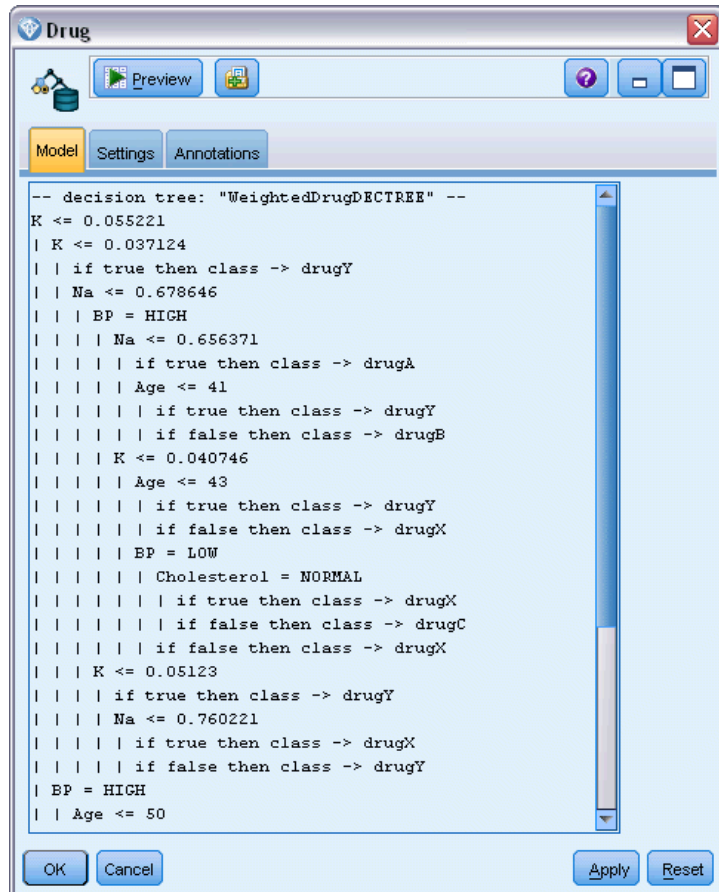
The model output takes the form of a textual representation of the tree. Each line of text corresponds to a node or a leaf, and the indentation reflects the tree level. For a node, the split condition is displayed; for a leaf, the assigned class label is shown.

Netezza 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.

Netezza 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 modeling node, the node adds two new fields containing 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*.

Netezza K-Means Nugget - Model Tab

Figure 6-26
K-Means model output
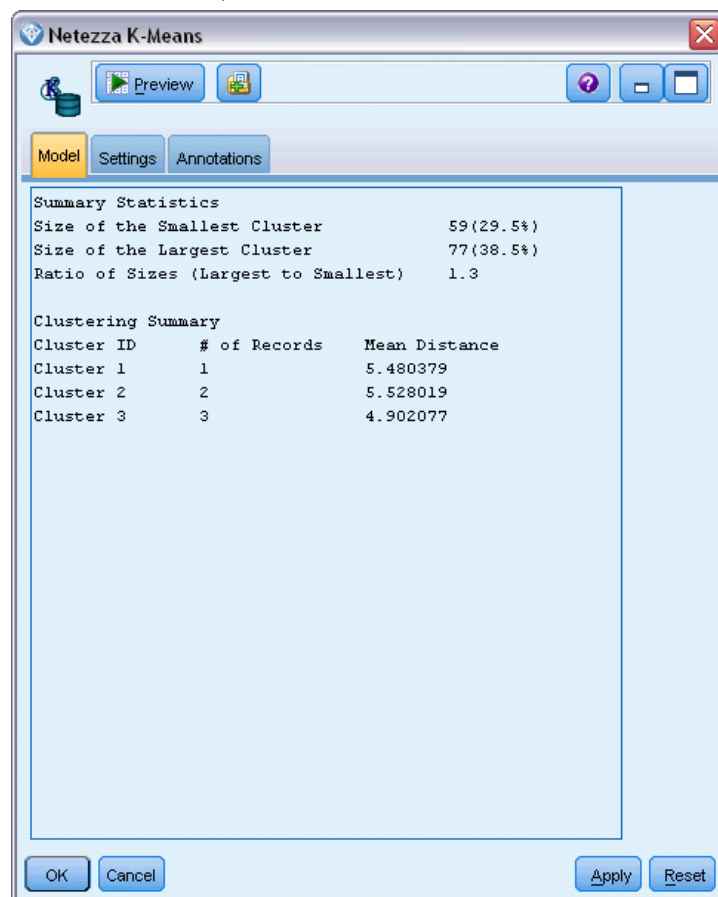
The model output is displayed on the Model tab as follows.

Summary Statistics. For both the smallest and the largest cluster, shows the number of records, and the percentage of the data set, taken up by these clusters. The list also shows the size ratio of the largest cluster to the smallest.

Clustering Summary. Lists the clusters 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.

Netezza 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 modeling node, the node adds one new field, the name of which is derived from the model name.

Table 6-3
Model-scoring field for Bayes Net

Name of Added Field	Meaning
$BN-*model_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. For more information, see the topic Scoring IBM Netezza Analytics Models on p. 183.

Netezza Bayes Net Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Figure 6-27
Bayes Net model settings
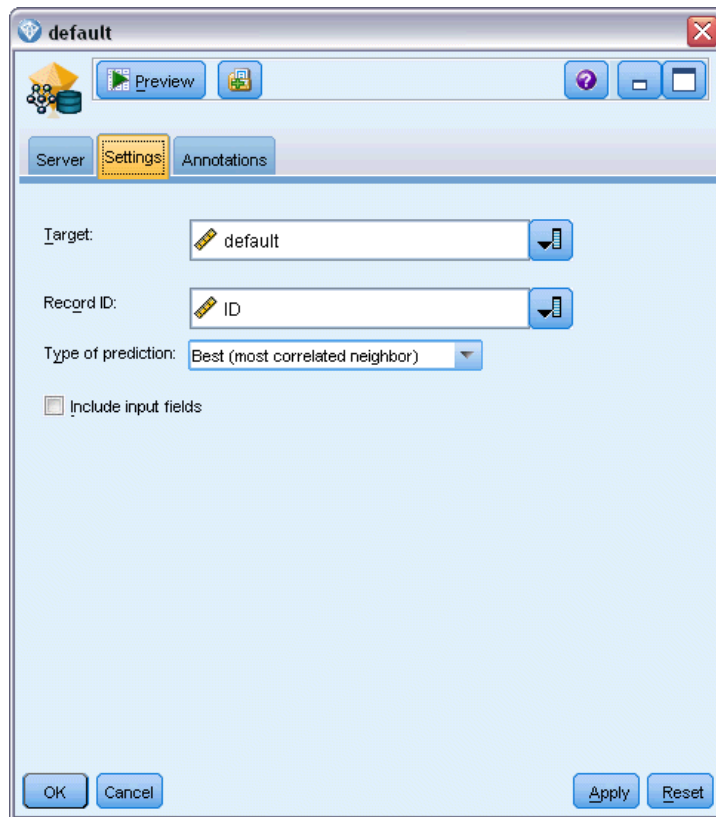
Target. If you want to score a target field that is different from the current target, choose the new target here.

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.

Netezza 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 modeling node, by default the node adds one new field, the name of which is derived from the model name.

Table 6-4
Model-scoring field for Naive Bayes - default

Name of Added Field	Meaning
$I-*model_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 6-5
Model-scoring fields for Naive Bayes - additional

Name of Added Field	Meaning
$IP-*model_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-*model_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. For more information, see the topic

Netezza 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.

Netezza 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 modeling node, the node adds one new field, the name of which is derived from the model name.

Table 6-6
Model-scoring field for KNN

Name of Added Field	Meaning
$KNN-*model_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. For more information, see the topic

Netezza KNN Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Figure 6-28
KNN model settings

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 modeling node, the node adds two new fields, the names of which are derived from the model name.

Table 6-7
Model-scoring fields for Divisive Clustering

Name of Added Field	Meaning
$DC-*model_name*	Identifier of subcluster to which current record is assigned.
$DCD-*model_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. For more information, see the topic Scoring IBM Netezza Analytics Models on p. 183.

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.

Netezza 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 modeling node, by default the node adds one new field, the name of which is derived from the model name.

Table 6-8
Model-scoring field for PCA

Name of Added Field	Meaning
$F-*model_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. For more information, see the topic

Netezza PCA Nugget - Settings Tab

On the Settings tab, you can set options for scoring the model.

Figure 6-29
PCA model settings
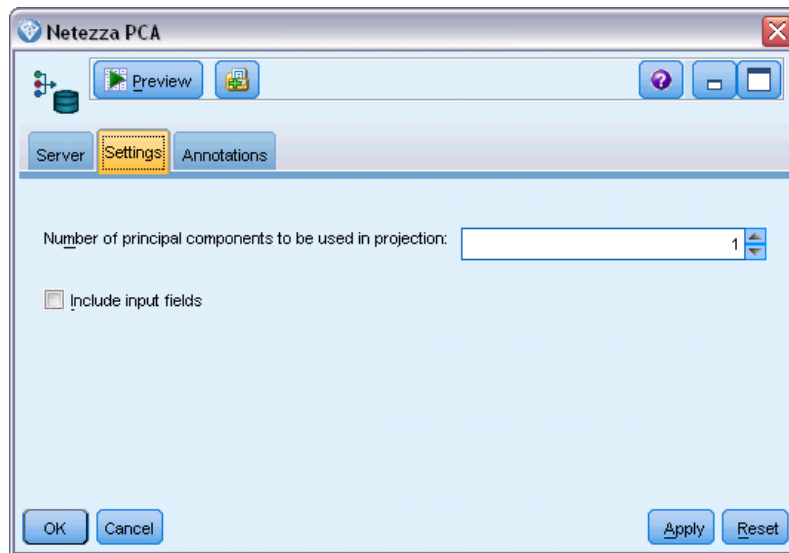
Number of principal components to be used in projection. The number of principal components to which you want to reduce the data set. This value must not exceed the number of attributes (input fields).

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 modeling node, by default the node adds one new field, the name of which is derived from the model name.

Table 6-9
Model-scoring field for Regression Tree

Name of Added Field	Meaning
$I-*model_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 6-10
Model-scoring field for Regression Tree - additional

Name of Added Field	Meaning
$IV-*model_name*	Estimated variances of assigned classes.

You can view the extra fields by attaching a Table node to the model nugget and running the Table node. For more information, see the topic Scoring IBM Netezza Analytics Models on p. 183.

Netezza Regression Tree Nugget - Model Tab

Figure 6-30
Regression Tree model output
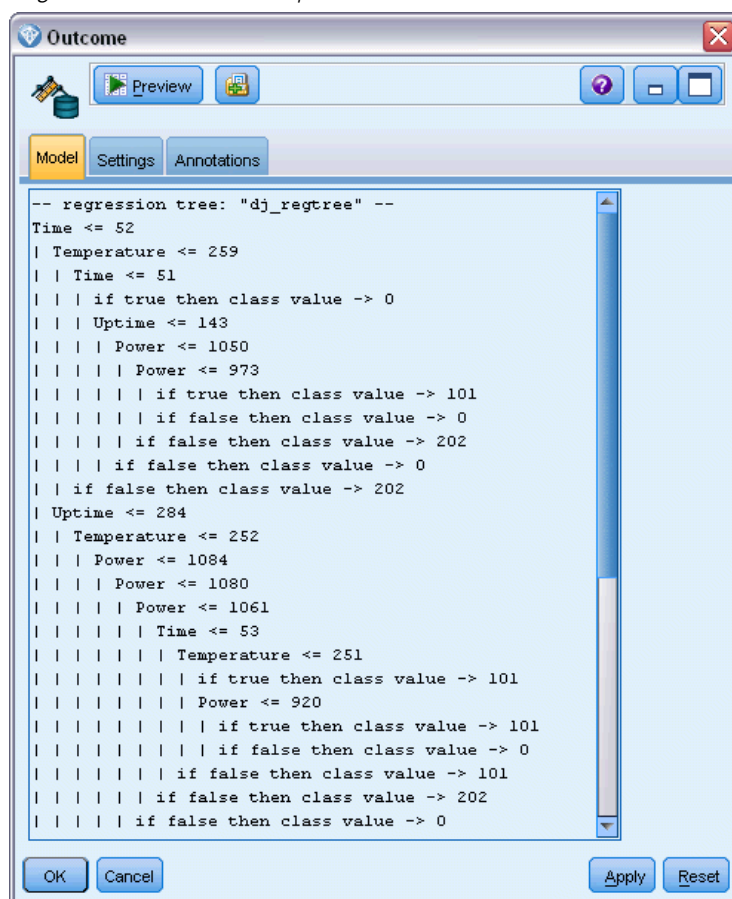
The model output takes the form of a textual representation of the tree. Each line of text corresponds to a node or a leaf, and the indentation reflects the tree level. For a node, the split condition is displayed; for a leaf, the assigned class label is shown.

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 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 modeling node, the node adds one new field, the name of which is derived from the model name.

Table 6-11
Model-scoring field for Linear Regression

Name of Added Field	Meaning
$LR-*model_name*	Predicted value for current record.

Netezza 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.

Notices

This information was developed for products and services offered worldwide.

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, Armonk, NY 10504-1785, U.S.A.

For license inquiries regarding double-byte character set (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., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Japan.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES 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 states 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply 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 Software Group, Attention: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

197

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All 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 the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, and SPSS are trademarks of IBM Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at *http://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.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

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.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Other product and service names might be trademarks of IBM or other companies.

Index