Automated Data Mining with IBM SPSS Modeler

Student Guide

Course Code: 0A0G2

ERC 1.0

Automated Data Mining with IBM SPSS Modeler

0A0G2

Published October 2010

Table of Contents

Lesson 1: Course Introduction

1.1 *Introduction*

The focus of this one-day course is on the use of IBM® SPSS® Modeler to demonstrate how to complete an automated data mining project. All phases of the project will be illustrated so that the student gains a practical understanding of the stages of a project and the details of how to use the nodes in Modeler to complete these operations.

1.2 *Course Objectives*

After completing this course students will be able to:

- Use Modeler to perform an automated data mining project

To support the achievement of this primary objective, students will also be able to:

- Understand the principles of data mining
- Use the user interface of Modeler to create basic Modeler streams (flow of operations)
- Read a IBM® SPSS® Statistics data file into Modeler and define data characteristics
- Review and explore data to look at data distributions and to identify data problems, including missing values
- Use the Automated Data Prep node to further prepare data for modeling
- Use a Partition node to create training and testing data subsets
- Use the Feature Selection node to select inputs for modeling
- Use the Auto Classifier node to create a model to predict a categorical target
- Evaluate and understand the predictions of a model
- Use the Auto Numeric node to create a model to predict a continuous target
- Use a model to score new data

Note

In this lesson and in the remaining lesson within this course, a few conventions have been followed for product references.

- Collaboration and Deployment Services is abbreviated as **C&DS** in following lessons.
- The corporate prefix, *i.e.* IBM® SPSS® has been left off of the following product names:
  - IBM® SPSS® Modeler => Modeler
  - IBM® SPSS® Modeler Advantage => Modeler Advantage
  - IBM® SPSS® Statistics => Statistics
  - IBM® SPSS® Collaboration and Deployment Services => C&DS
  - IBM® SPSS® Text Analytics => Text Analytics

1.3 *About SPSS*

SPSS Inc. an IBM® Company is a leading global provider of predictive analytics software and solutions. The Company's complete portfolio of products - data collection, statistics, modeling and deployment - captures people's attitudes and opinions, predicts outcomes of future customer interactions, and then acts on these insights by embedding analytics into business processes. SPSS® solutions address interconnected business objectives across an entire organization by focusing on the convergence of analytics, IT architecture and business process. Commercial, government and academic customers worldwide rely on SPSS technology as a competitive advantage in attracting, retaining and growing customers, while reducing fraud and mitigating risk. SPSS was acquired by IBM in October 2009. For more information, visit http://www.spss.com.

1.4 *Supporting Materials*

Supporting Materials

The following materials are used in this course. All of the files can be found in the *c:\Train\Modeler_AutomatedDM* directory. All paths in the following lessons will be relative to this directory. If the course is being conducted in a non-SPSS, an IBM Company sponsored facility; the instructor will define the base directory for the files.

Data Files:

- *CharityBig.sav*
- *Charity_New.sav*
- *Customer_offers.sav*
- *Customer_offers_new.sav*

Stream Files:

- *Backup_Customer_Offers.str*
- *Backup_Customer_Offers_Data Audit.str*
- *Backup_Customer_Offers_Data Audit & ADP.str*
- *Backup_Customer_Offers_Partition.str*
- *Backup_Customer_Offers_Feature Selection.str*
- *Backup_Customer_Offers_Auto Classifier.str*
- *Backup_Customer_Offers_Model Evaluation.str*
- *Backup_Lesson 4 Exercise.str*
- *Backup_Lesson 5 Exercise.str*
- *Backup_Lesson 6 Exercise.str*
- *Backup_Lesson 7 Exercise.str*
- *Backup_Lesson 8 Exercise.str*
- *Backup_Lesson 9 Exercise.str*
- *Backup_Lesson 10 Exercise.str*
- *Customer_Offers_Data Audit_Complete.str*
- *Customer_Offers_Auto Numeric.str*
- *Customer_Offers_Tenure.str*
- *Customer_Offers_Scoring.str*
- *Lesson 11 Exercise.str*
- *Lesson 12 Exercise.str*

External Files:

- *Customer_Offers Field Labels.doc*.

1.5 *Course Assumptions*

The following points are assumed for completion of this course:

- Attendees have access to Modeler 14.0 or later version
- Attendees have access to the supporting course files listed in the Supporting Materials section above
- No prior experience with Modeler is required
- No experience with data mining or statistical analysis is required, although some prior experience with analytics would be helpful

Lesson 2: Introduction to Data Mining

2.1 *Objectives*

After completing this lesson students will be able to:

- Understand the principles of data mining

To support the achievement of this primary objective, students will also be able to:

- Describe the features included with Modeler to automate data mining
- Describe the phases of the CRISP-DM process model for data mining

2.2 *Introduction*

Data seems to be everywhere, and there is more of it, all the time. Electronic methods of data collection have led to an explosion of data in all types of businesses and organizations. With this has come the need, spurred on by competitive pressures, to use this information to identify useful patterns and actionable relationships that can be applied to business operations.

Data mining is a general term which encompasses a number of techniques to extract useful information from (large) data files, without necessarily having preconceived notions about what will be discovered. The useful information often consists of patterns and relationships in the data that were previously unknown or even unsuspected. Data mining is also sometimes called *Knowledge Discovery in Databases* (KDD). Another more technical name for these techniques is “machine learning.”

Data mining is an interactive and iterative process. Business or domain expertise must be used jointly with advanced algorithms to identify underlying relationships and features in the data.

Several of the common data-mining techniques require a different approach to model generation and testing compared to standard parametric statistics. Existing data is used to “train” a model, and then ‘test” or validate it to determine whether it should be deemed acceptable and likely to generalize to the population of interest. Due to the typically large files and weak assumptions made about the distribution of the data, data mining tends to be less focused on statistical significance tests and more on practical importance. However, standard statistical methods are increasingly being used alongside newer techniques, and both types are incorporated into Modeler.

2.3 *Automated Data Mining*

The use of data-mining methods and software has become quite common in the twenty-first century, and more and more, this has meant providing end-users access to techniques that allow them to conduct data mining without deep analytical backgrounds. The Modeler software has been designed from inception with this goal at the forefront, and the automation of data mining, including data exploration, preparation, and modeling, has been an underlying theme of the Modeler development.

These features include:

- A visual programming environment that uses icons which represent operations to be carried out on data. The icons are referred to as *nodes*. The flow of data is immediately apparent, as are the model *nuggets* developed to make predictions. This flow of data composed of nodes and model nuggets can be saved in *streams* for further modeling and deployment.
- A Data Audit node to automate the exploration of data
- An Auto Data Prep node to automate the preparation of data for modeling

- A Feature Selection node to automate the selection of predictors for models
- A Partition node to separate data into training and testing sets
- Auto Classifier and Auto Numeric modeling nodes to easily develop ensemble models to predict categorical and continuous targets, respectively.
- An Analysis node to automate the evaluation of model predictions

In addition to these nodes, there are many other features in Modeler that make data mining more efficient and effective. These include:

- The generation of new nodes from existing nodes to do various types of operations on data, such as impute missing data
- The ability of models to use upstream definitions of the predictors and target (outcome) field to avoid re-specification
- The ability to easily score new data with a model

We use all of these nodes and features in this course to demonstrate how to perform automated data mining with Modeler. Our examples will include all the common steps of a data-mining project, from reading data into Modeler, to eventually using a model to score new data. By the end of the class, the student should feel comfortable using the Modeler environment, and also understand its capabilities to automate a data-mining project.

2.4 *A Strategy for Data Mining: CRISP-DM*

As with most business endeavors, data mining is much more effective if done in a planned, systematic way. Even with tools such as Modeler, an analyst still needs to plan the work carefully and gather the correct information and define project requirements, including the requirements of a successful model.

The typical data-mining process can become complicated very quickly. There is a lot to keep track of—complex business problems, multiple data sources, varying data quality across data sources, an array of data-mining techniques, different ways of measuring data mining success, and so on.

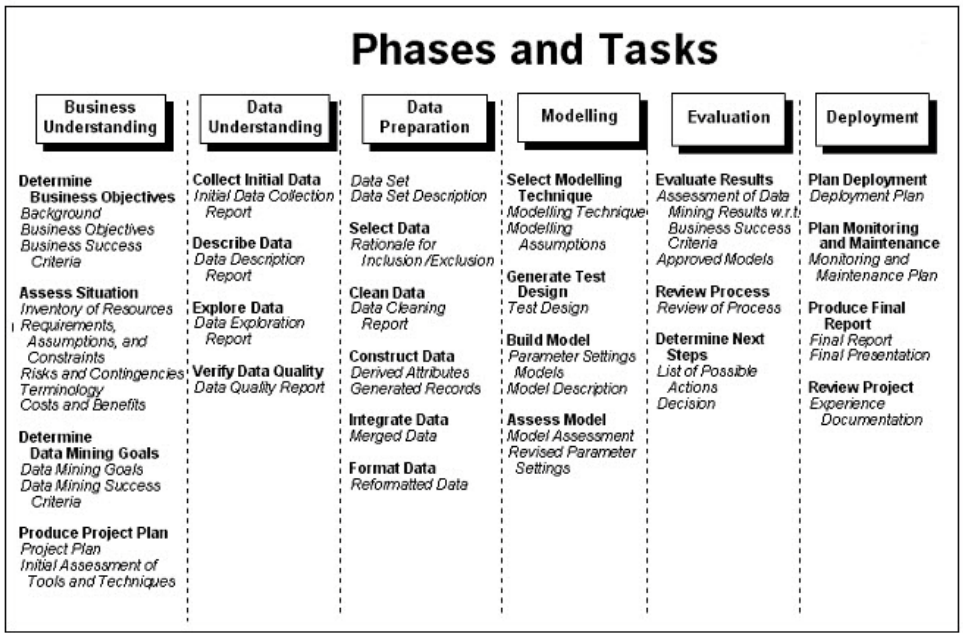
To stay on track, it helps to have an explicitly defined process model for data mining. The process model makes sure that the important points are addressed. It serves as a data mining road map so that the user won't lose their way as they dig into the complexities of their data.

The data mining process model recommended for use with Modeler is the Cross-Industry Standard Process for Data Mining (CRISP-DM). As we can tell from the name, it is designed as a general model that can be applied to a wide variety of industries and business problems. The CRISP-DM reference is included with the Modeler documentation (*CRISP-DM 1.0*) and can be downloaded from www.crisp-dm.org.

The general CRISP-DM process model includes six phases that address the main issues in data mining. The six phases fit together in a cyclical process.

These six phases cover the full data mining process, including how to incorporate data mining into overall business practices. These phases are listed in the diagram in the figure below.

Figure 2.1 CRISP-DM Model



The six phases include:

Business understanding. This is perhaps the most important phase of data mining. Business understanding includes determining business objectives, assessing the current situation, determining data-mining goals, and producing a project plan.

Data understanding. Data provides the "raw materials" of data mining. This phase addresses the need to understand what data resources are available and the characteristics of those resources. It includes collecting initial data, describing data, exploring data, and verifying data quality.

Data preparation. After cataloging the existing data resources, the data will need to be prepared for data mining. Preparations include selecting, cleaning, constructing, integrating, and formatting data. These tasks will likely be performed multiple times, and not necessarily in any prescribed order. These tasks can be very time consuming but are critical for the success of the data-mining project.

Modeling. This phase involves selecting modeling techniques, generating test designs, and building and assessing models. Developing a model is an iterative process—as it can be in standard statistical modeling—and we should expect to try several models, and modeling techniques, before finding a best model. As we demonstrate in this course, another feature that separates data-mining from other approaches is the use of multiple models to make predictions, building on the strengths of each technique.

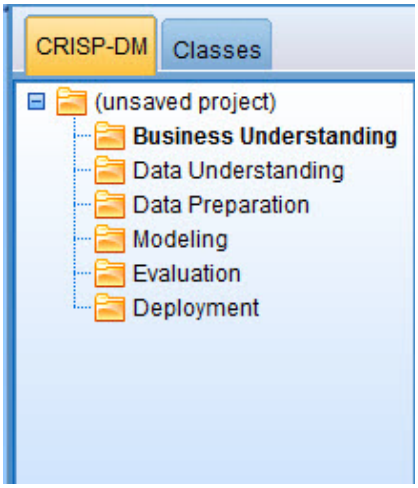
Evaluation. Once the models have been chosen we are ready to evaluate how the data-mining results can help to achieve our business objectives. At this stage in the project, we have built a model (or models) that appears to have high quality, from a data analysis perspective. Before writing final reports and deploying the model, it is important to more thoroughly evaluate the model, and review the steps executed to construct the model, to be certain it properly achieves the business objectives.

Deployment. In this phase, a successful model is applied to new data to make predictions. This might be relatively simple if done within the data-mining software (and Modeler allows the user to easily score new data), or more complex if the model is to be applied directly against an existing

database. A plan should be developed to monitor the model's predictions over time in order to verify that the model still holds true.

To assist the user in organizing their Modeler programs (streams) around the CRISP-DM framework, Modeler contains a Project window (see figure below). In the Project window, a project folder contains subfolders corresponding to the phases in CRISP-DM. This makes it easier to organize Modeler programs and other documents that are associated with a data-mining project. The user can save a project file that contains links to Modeler streams and other documents.

Figure 2.2 Modeler Projects Manager



The lessons in this course are organized around the CRISP-DM phases. We will complete a full data mining project in this course, albeit on a reduced scale and time frame. But the student will be exposed to the essentials of an automated data-mining project.

2.5 *Lesson Summary*

In this lesson we introduced data mining and the CRISP-DM methodology.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Understand the principles of data mining

To support the achievement of the primary objective, students should now also be able to:

- Describe the features included with Modeler to automate data mining
- Describe the phases of the CRISP-DM process model for data mining

Lesson 3: The Basics of Using IBM SPSS Modeler

3.1 *Objectives*

After completing this lesson students will be able to:

- Use the user interface of Modeler to create basic Modeler streams

To support the achievement of this primary objective, students will also be able to:

- Describe the components of the user interface of Modeler
- Place nodes on the stream canvas
- Connect and disconnect nodes
- Edit and rename nodes

3.2 *Introduction*

Modeler has been designed with ease of use in mind. The software design employs a type of visual programming by using icons to represent program actions, connected by data flows via graphic arrows. Rather than conventional menus, most program actions are requested within each icon, or node, by making selections in dialog boxes. In this lesson we will review the basic features of the Modeler user interface and show how to perform common actions.

Supporting Materials

None

3.3 *The Modeler User Interface*

To run Modeler:

1) From the Start button, click **All Programs…SPSS Inc.…PASW Modeler 14… PASW Modeler 14**

At the start of a session, we see the Modeler User Interface.

Modeler enables the user to mine data visually using the Stream Canvas. This is the main work area in Modeler and can be thought of as a surface on which to place icons. These icons represent operations to be carried out on the data and are often referred to as *nodes*.

- The nodes are contained in *palettes*, located across the bottom of the Modeler window.
- Each palette contains a related group of nodes that can be added to the data stream. For example, the Sources palette contains nodes that can be used to read data into Modeler, and the Graphs palette contains nodes that can be used to explore the data visually.
- The Modeling palette contains many nodes so it is grouped into sub palettes.
- The icons that are visible depend on the active, selected palette.

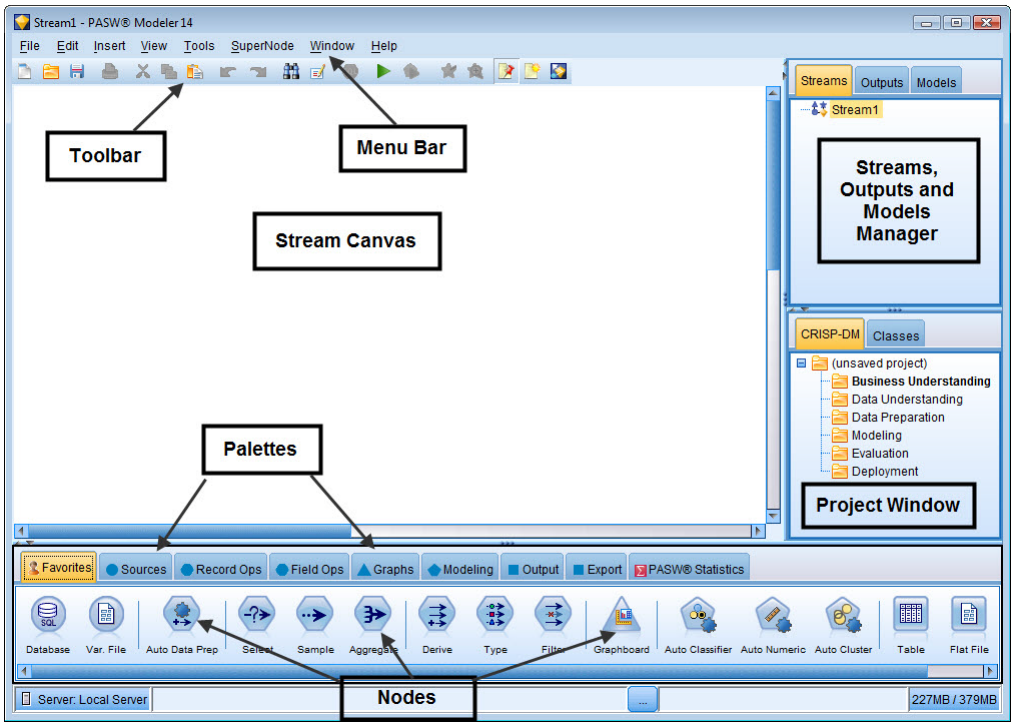
The Favorites palette is a customizable collection of nodes that are used most frequently. It contains a default collection of nodes, but these can be easily modified within the Palette Manager (accessed by selecting Tools…Manage Palettes).

Once nodes have been placed on the Stream Canvas, they can be linked together to form a *stream*. A stream represents a flow of data through a number of operations (nodes) to a destination that can be in the form of output (either text or chart), a model, or the export of data to another format (e.g., a Statistics data file or a database).

At the upper right of the Modeler window, there are three types of manager tabs. Each tab (Streams, Outputs, and Models) is used to view and manage the corresponding type of object.

- We can use the Streams tab to open, rename, save, and delete streams created in a session.
- Modeler output, such as graphs and tables, is stored in the Outputs tab. We can save output objects directly from this manager.
- The Models tab is the most important of the manager tabs as it contains the results of the machine learning and modeling done in Modeler. These models can be browsed directly from the Models tab or on the stream, where they are placed automatically.

Figure 3.1 Modeler User Interface



At the lower right of the Modeler window is the Projects window. This window offers the user a best-practice way to organize their data mining work. The CRISP-DM tab helps the user to organize streams, output, and annotations according to the phases of the CRISP-DM process model (mentioned in Lesson 2). Even though some items do not typically involve work in Modeler, the CRISP-DM tab includes all six phases of the CRISP-DM process model so the user has a central location for storing and tracking all materials associated with the project.

The Classes tab in the Project window organizes the work in Modeler categorically by the type of objects created. Objects can be added to any of the following categories:

- Streams
- Nodes
- Generated Models
- Tables, Graphs, Reports
- Other (non-Modeler files, such as slide shows or white papers relevant to the data mining project)

There are eight menu choices in the Modeler menu bar:

- **File** allows the user to create, open and save Modeler streams and projects. Streams can also be printed from this menu.
- **Edit** allows the user to perform editing operations: for example, copy/paste objects; clear manager tabs; edit individual nodes.
- **Insert** allows the user to insert a particular node, as alternative to dragging a node from the palette.
- **View** allows the user to toggle between hiding and displaying items (for example, the toolbar or the Project window).
- **Tools** allows the user to manipulate the environment in which Modeler works and provides facilities for working with scripts.
- **Supernode** allows the user to create, edit and save a condensed stream to save space on the Canvas.
- **Window** allows the user to close related windows (for example, all open output windows), or switch between open windows.
- **Help** allows the user to access help on a variety of topics or view a tutorial.

3.4 *Visual Programming*

As mentioned earlier, data mining is performed by creating a stream of nodes through which the data pass. A stream, at its simplest, will include a source node, which reads the data into Modeler, and a destination, which can be an output node, such as a table, a graph, or a modeling operation.

When building streams, mouse buttons are used in the following ways:

Left button	Used for icon or node selection, placement and positioning on the Stream Canvas.
Right button	Used to invoke Context (pop-up) menus that, among other options, allow editing, renaming, deletion and execution of the nodes.
Middle button (optional)	Used to connect two nodes and modify these connections. (When using a two-button mouse, the user can right-click on a node, select Connect from the context menu, and then click on the second node to establish a connection.)

Adding a Node

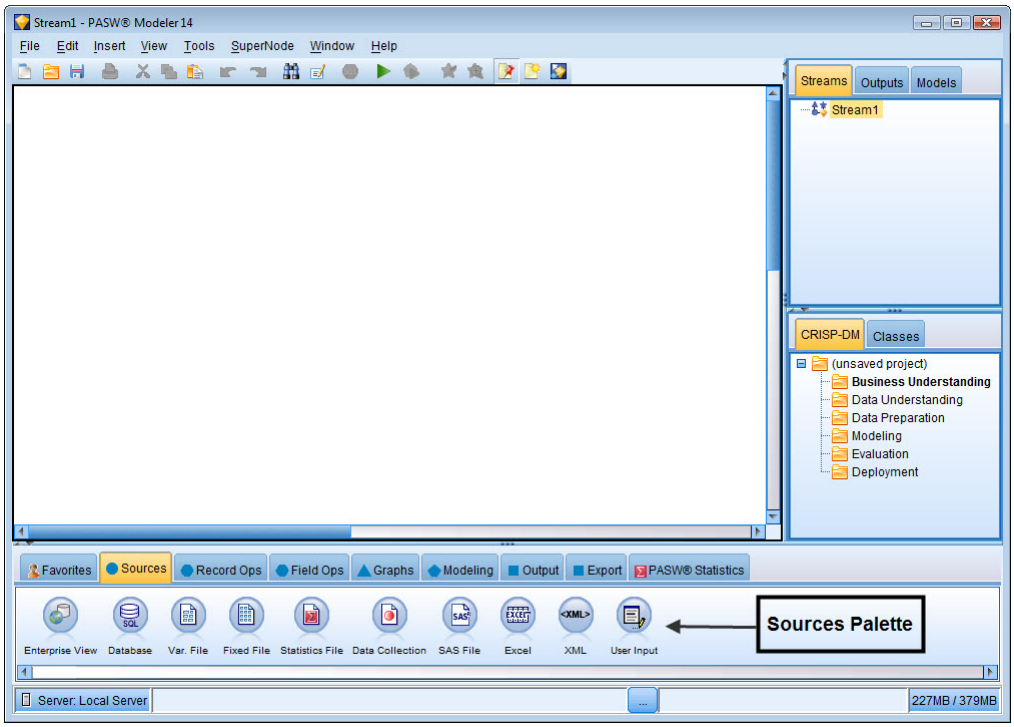
To begin a new stream, a node from the Sources palette needs to be placed on the Stream Canvas. There are three ways to add nodes to a stream from the nodes palette:

- Double-click a node on the palette. Note: Double-clicking a node automatically adds it to the current stream, connecting it to the current selected node, if possible.
- Drag and drop a node from the palette to the stream canvas.
- Click a node on the palette, and then click on the stream canvas.

In this example we will illustrate the third of these methods. We will also assume that data are being read from a previously saved Statistics data file.

1) Activate the Sources palette by clicking the **Sources** tab

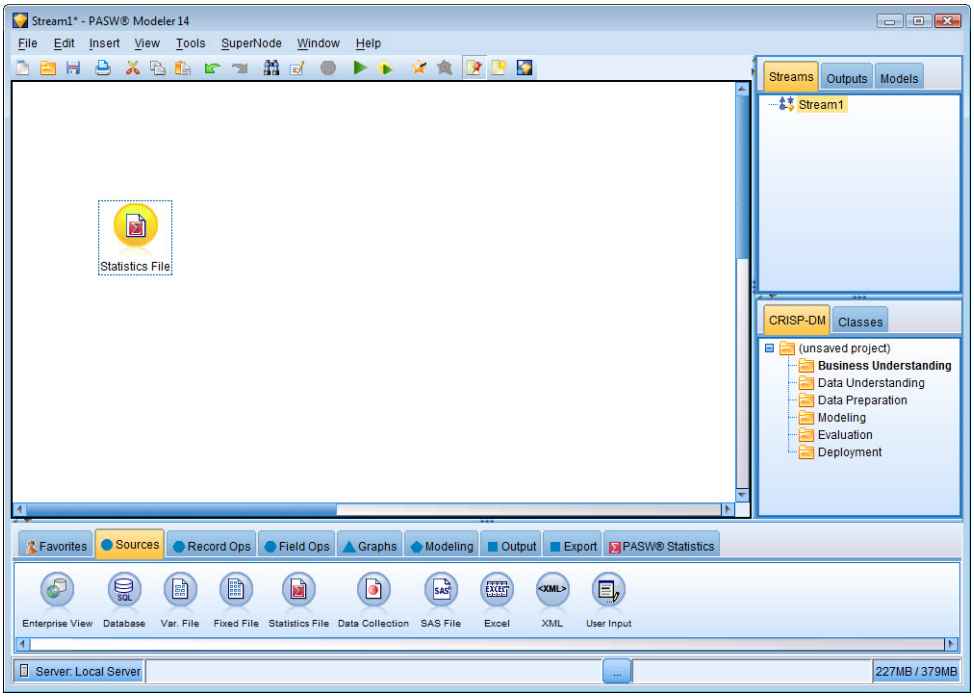
Figure 3.2 Sources Palette



2) Select the **Statistics File** node from the Sources palette by clicking
3) Move the cursor over the Stream Canvas
4) Click **anywhere** in the Stream Canvas

A copy of the icon should appear on the Stream Canvas. This node now represents the action of reading data into Modeler from a Statistics data file.

Figure 3.3 Placing a Source Node on the Stream Canvas



Moving a Node

To move the node within the Stream Canvas, select it (using the left mouse button), and while holding this button down, drag the node to its new position.

Actions to Take with a Node

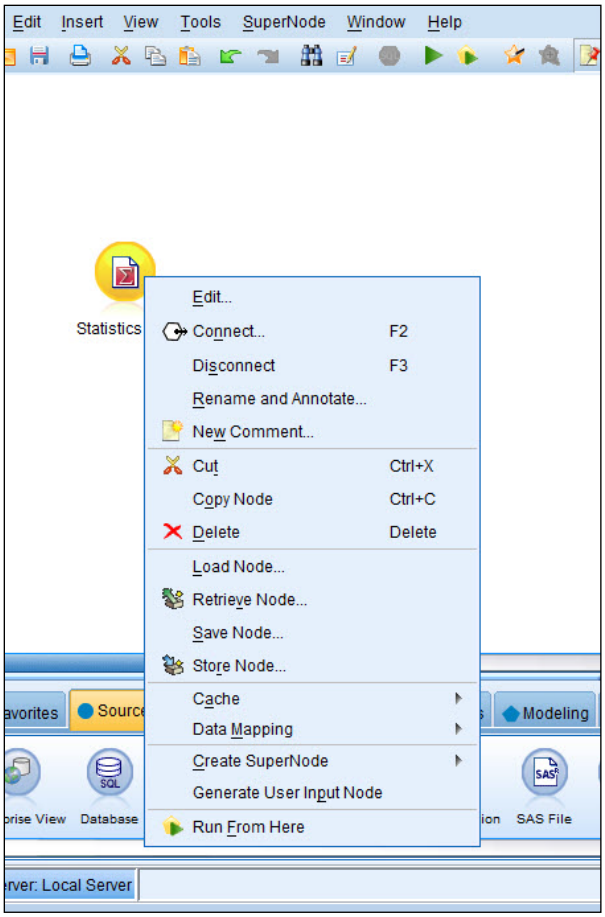
In order to view the action options for a node, right-click on the icon to reveal a Context (pop-up) menu.

1) Right-click on the **Statistics File** node  in the Stream Canvas

The most common action is to edit the node, which opens a dialog box specific to each node type. Double-clicking on the node also opens the dialog box.

Many other actions are available, including connecting and disconnecting nodes, renaming and annotating nodes, copying, saving, and caching the node.

Figure 3.4 Context Menu When a Source Node is Right-Clicked



3.5 *Building Streams with Modeler*

Once two or more nodes have been placed on the Stream Canvas, they need to be connected to produce a stream. This can be thought of as representing a flow of data through the nodes.

To demonstrate this we will place a Table node in the Stream Canvas next to the Statistics File node. The Table node displays the data in a table format, similar to a spreadsheet view.

1) Click the **Output** tab to activate the Output palette
2) Click on the **Table** node  in the Output palette
3) Place this node to the **right** of the **Statistics File** node by clicking in the Stream Canvas

Figure 3.5 Table Node Added to Stream Canvas

Connecting Nodes

There are a number of ways to connect nodes to form a stream: double-clicking, using the middle mouse button, or manually.

The simplest way to form a stream is to double-click nodes on the palette. This method automatically connects the new node to the currently selected node on the stream canvas (the one outlined in the blue-dotted box). For example, if the canvas contains a Database node, the user can select this node and then double-click an appropriate node from the palette, such as a Derive node. This action automatically connects the Derive node to the existing Database node.

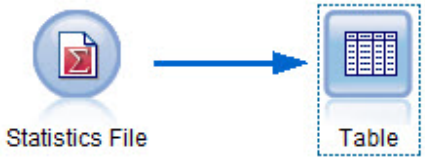
To manually connect two nodes:

1) Right-click on the Statistics File node, and then select **Connect** from the context menu (note the cursor changes to include a connection icon )
2) Click the **Table** node

Alternatively, with a three-button mouse:

1) Click with the **middle mouse** button on the Statistics File node
2) While holding the **middle button** down, drag the cursor over the Table node
3) Release the middle mouse button

Figure 3.6 Statistics File Node Connected To Table Node



A connecting arrow appears between the nodes. The head of the arrow indicates the data flow direction.

Disconnecting Nodes

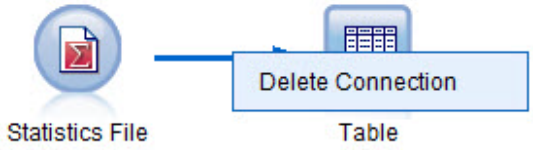
Nodes can be disconnected in several ways:

- By right-clicking on one of the nodes and selecting the Disconnect option from the context menu
- By right-clicking on the actual connection and selecting the Delete Connection option
- By double-clicking with the middle mouse button on one of the connected nodes (for intermediate nodes this will make existing arrows “bypass” the node)

We will demonstrate one of these alternatives.

1) Right-click on the **connecting arrow**
2) Select **Delete Connection**

Figure 3.7 Disconnecting Nodes



Stream Data Flow

Data follows the connection arrows in a stream. Data normally enter a stream from a data source, and eventually enter a modeling node (such as Auto Classifier).

When connecting nodes, there are several guidelines to follow. Some nodes can only send data downstream (source node), while other nodes can only receive a data input from upstream (a *terminal* node). And other nodes can have both an input and an output.

An error message is displayed if the user attempts to make any of the following types of connections:

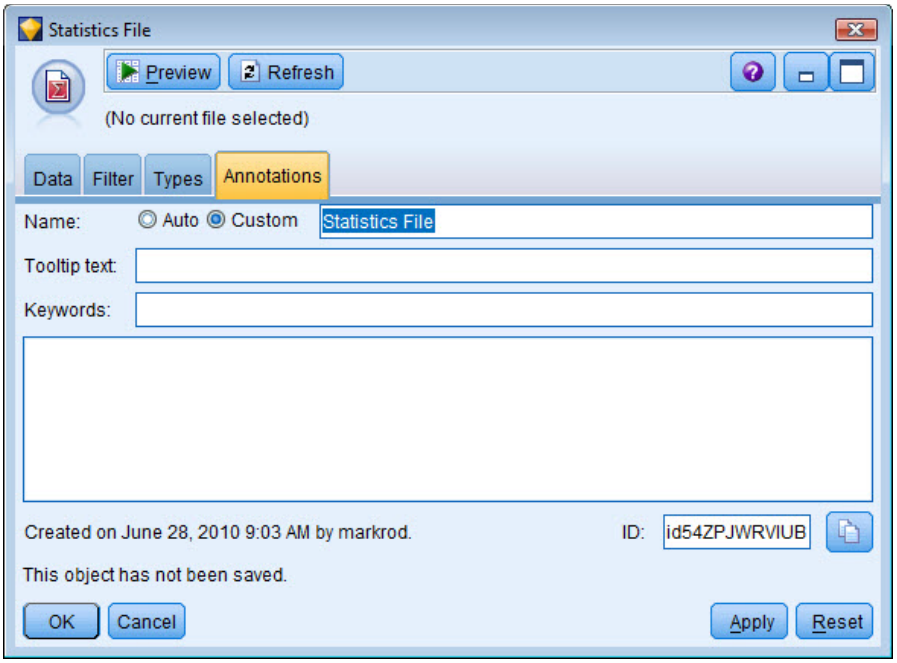
- A connection leading to a source node
- A connection leading from a terminal node
- A node having more than its maximum number of input connections
- Connecting two nodes that are already connected
- Circularity (data returns to a node from which it has already flowed)

Renaming and Annotating a Node

Nodes can be labeled with a more descriptive name by using one of the context menu options.

1) Right-click on the **Statistics File** node
2) Select **Rename and Annotate** on the context menu

Figure 3.8 Rename and Annotate Dialog



We can specify a name and even tooltip text for the node. The tooltip text feature is useful to aid in distinguishing between similar nodes on the stream canvas. In the text area at the bottom, additional information can be attached to the node in order to aid interpretation or to act as a reminder to what it represents. The *Keywords* text box allows the user to enter keywords that are used in project reports and to search or track objects in the Model Manager repository (keywords can be specified for a stream, model, or output object in Modeler).

Tip

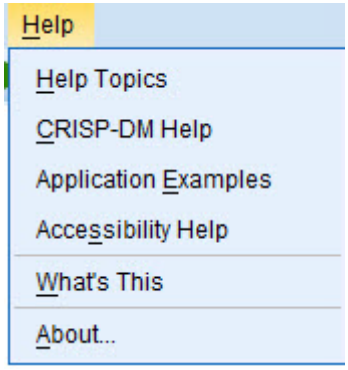
A comment can be added to a node by right-clicking the node and select New Comment from the context menu; Alternatively, select the node and click the Insert new comment button in the Tool bar. In the same way, a stream can be annotated with comments.

3.6 *Modeler Help*

There is a variety of both operational and modeling help available from the Help menu:

1) Select **Help** on the main menu

Figure 3.9 Help Menu



The Help menu contains several options. The Help Topics choice goes directly to the Help system. CRISP-DM Help gives an introduction to the CRISP-DM methodology. Application Examples accesses a variety of real-life examples of using common data mining techniques of data preparation and modeling. Accessibility Help lists keyboard alternatives to using the mouse. What's This changes the cursor into a question mark and provides information about any Modeler item or object the user selects.

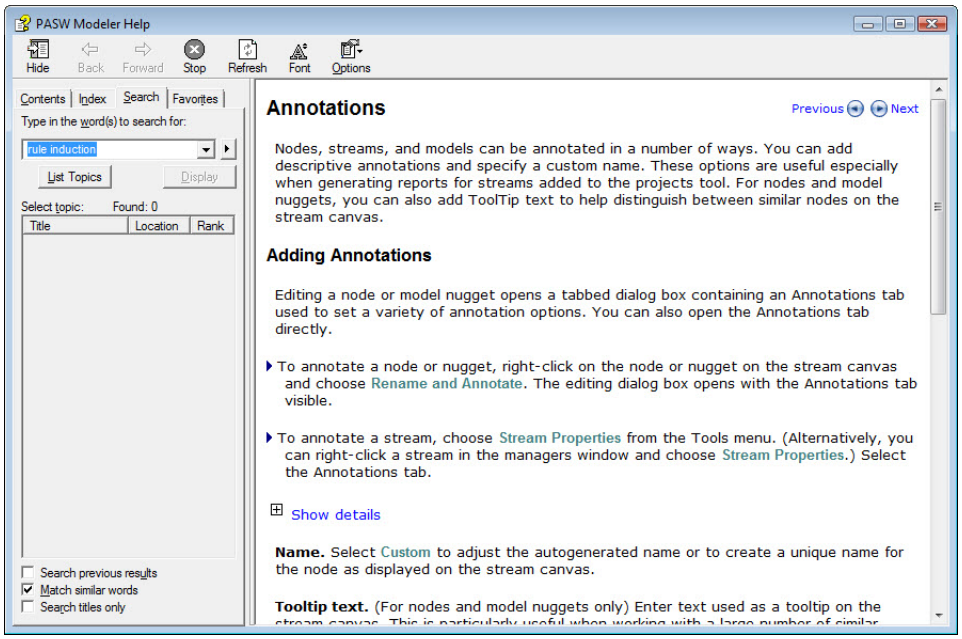
Besides the help provided by the Help menu, context sensitive help is always available in whatever dialog box is open. As an example we look at the help when we rename or annotate the Statistics File node.

1) Click away from the Help menu

2) In the Open Rename and Annotate dialog box, select the **Help** button 

Information about the options in this specific dialog box can be accessed in this manner, or in any dialog box.

Figure 3.10 Context Sensitive Help on Annotating Nodes



3) Close the Help window, and close the Rename and Annotate dialog box

Although there are many other actions that can be taken with nodes and streams in Modeler, including executing a stream to read or modify data and create a model, the basic operations needed to use Modeler have been reviewed in this lesson.

3.7 *Lesson Summary*

In this lesson we introduced the Modeler user interface.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Use the user interface of Modeler to create basic Modeler streams

To support the achievement of the primary objective, students should now also be able to:

- Describe the components of the user interface of Modeler
- Place nodes on the stream canvas
- Connect and disconnect nodes
- Edit and rename nodes

3.8 Learning Activity

The overall goal of this learning activity is to practice creating adding nodes and creating streams in Modeler.

Supporting Materials

None

1. Start Modeler if not already open.
2. Review the various Palettes in Modeler. The names of the nodes usually reflect what operation or model is accomplished with that node.
3. Select the Var. File node from the Sources palette and place it on the Stream Canvas.
4. Select the Table node from the Output palette. Use several methods of placing a Table node on the Canvas.
5. Connect the Var. File node to the Table node, using at least two methods. Disconnect the nodes after each connection.
6. Try to connect the Table node to the Var. File node (the arrow runs from the former to the latter). What happens?
7. Rename the Var. File node to read “My Data Source.”

Lesson 4: Reading Data Files

4.1 *Objectives*

After completing this lesson students will be able to:

- Read a Statistics data file into Modeler and define data characteristics

To support the achievement of this primary objective, students will also be able to:

- Use a Statistics File node to read a Statistics data file
- Use the Filter tab to filter and rename fields
- Use the Types tab to view measurement level and set field role
- Save a Modeler stream file

4.2 *Introduction*

The first step in creating a stream is getting data into Modeler. Modeler reads a variety of different file types, including data stored in spreadsheets and databases, using the nodes within the Sources palette.

Data can be read in from text files, in either free-field or fixed-field format, using the Var. File and Fixed File source nodes. Statistics and SAS data files can be directly read into Modeler using the Statistics File and SAS File nodes, respectively. Excel files can be read directly with the Excel node.

If the user has data in an ODBC (Open Database Connectivity) source, the Database source node can be used to import data from server databases, such as Oracle™ or SQL Server™ and from a variety of other packages including dBase™.

Supporting Materials

The *customer_offers.sav* Statistics data file is used in this lesson. These data are from former and current customers of a telecommunications company. The data includes both demographic and account-related information.

4.3 *Reading Data from Statistics Files*

The Statistics File node reads data from an IBM SPSS Statistics data file. We use a Statistics data file in these lessons because this type of file contains both data values and labels for categorical fields, making the task of reviewing and understanding the data much easier.

The file contains demographic information including age, income, gender, marital status, and education, and also information about whether the customer is current or not (*churn*). There is a mix of continuous fields (e.g., *age*, *income*) and categorical fields (e.g., *gender*, *marital*, *churn*). There are many fields on the usage of telecommunication services.

There are two key decisions when reading Statistics data files:

- Specifying whether the variable names and labels should be read, or just the labels

- Specifying whether the data and data value labels should be read, or just the labels as data

We are using a Statistics data file because it already contains labels, so we will use the options to read names and labels, and data and labels.

Note

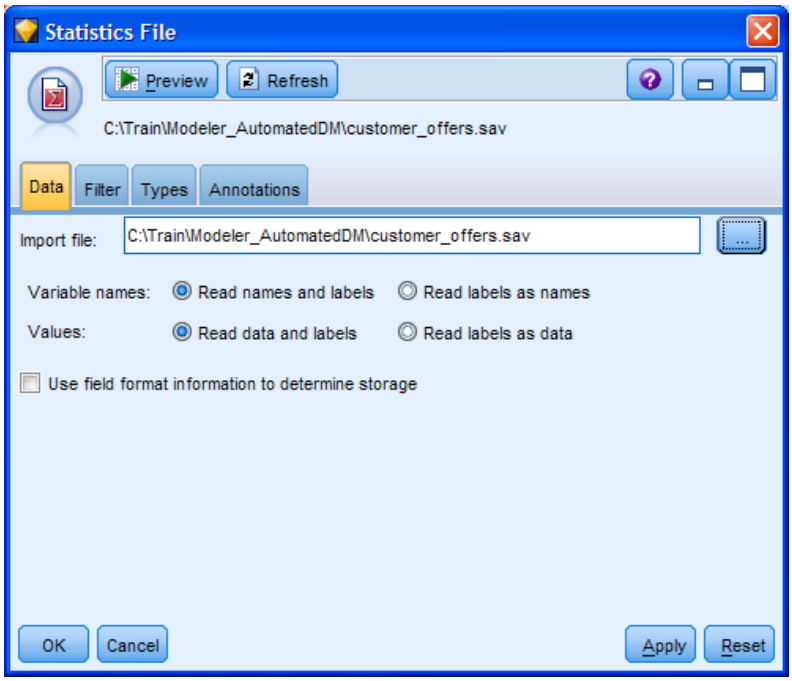
Course files are stored in the folder *c:\Train\Modeler_AutomatedDM*. All paths in the following lessons will be relative to this directory. If the course is being conducted in a non-SPSS, an IBM Company sponsored-facility, the instructor will define the base directory for the files.

4.4 *Operation: Reading a Statistics File*

We follow these steps to read a Statistics data file into Modeler.

1) Place a **Statistics File** source node in a new stream
2) Edit the **Statistics File** node
3) Select the file list button  and then navigate to the appropriate folder where the data file is stored.
4) Select the data file and select the **Open** button (not shown)

Figure 4.1 Statistics File Node Dialog



The default choice under Variable Names is *Read names and labels*. The default choice under Values is *Read data and labels*. These are the settings we require, per our discussion above.

5) Select **Use field format information to determine storage**

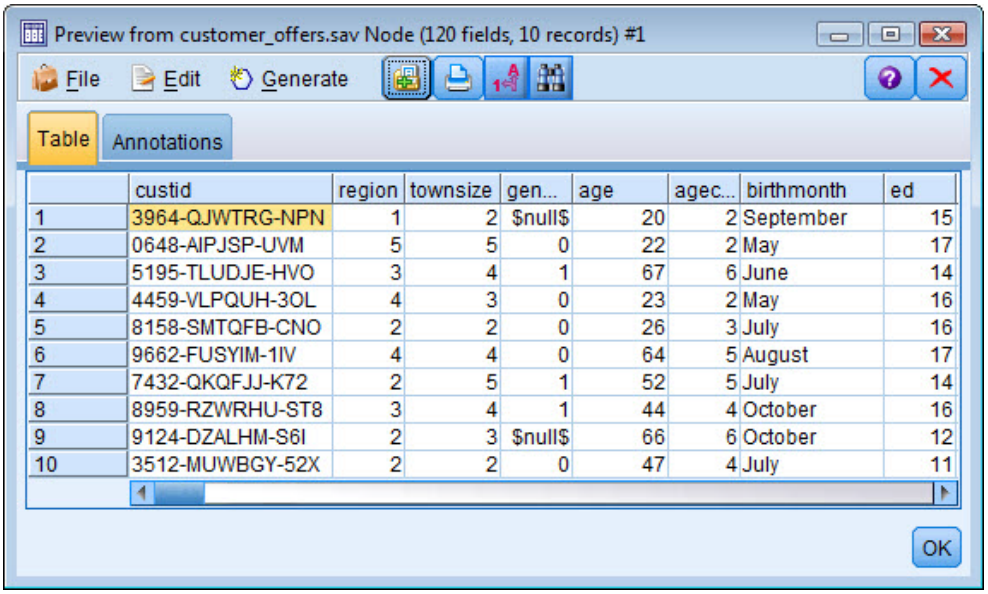
Data storage concerns the way data are stored internally for a field. For example, a field with values of 0 and 1 stores integer data. This is distinct from the *measurement level*, which describes the usage of the data by Modeler for data manipulation and modeling. For example, we might want to set the measurement level for an integer field with values of 1 and 0 to Flag. This usually indicates that 1 = True and 0 = False. While storage must be determined at the source node, measurement level can be changed there or in a Type node downstream.

The data can be previewed with the Preview button.

6) Select the **Preview** button

This preview shows how Modeler will read each field, displaying 10 data records and all fields. It can be used to determine whether the current settings are correct or need to be modified.

Figure 4.2 Preview of customer_offers.sav Data



	custid	region	townsize	gen...	age	agec...	birthmonth	ed
1	3964-QJWTRG-NPN	1	2	$null$	20	2	September	15
2	0648-AIPJSP-UVM	5	5	0	22	2	May	17
3	5195-TLUDJE-HVO	3	4	1	67	6	June	14
4	4459-VLPQUH-3OL	4	3	0	23	2	May	16
5	8158-SMTQFB-CNO	2	2	0	26	3	July	16
6	9662-FUSYIM-1IV	4	4	0	64	5	August	17
7	7432-QKQFJJ-K72	2	5	1	52	5	July	14
8	8959-RZWRHU-ST8	3	4	1	44	4	October	16
9	9124-DZALHM-S6I	2	3	$null$	66	6	October	12
10	3512-MUWBGY-52X	2	2	0	47	4	July	11

4.5 *Demonstration: Reading customer_offers Data File*

We now demonstrate the specific steps to read the file *customer_offers.sav* file.

Detailed Steps for Reading the Statistics File

1) Place a **Statistics File** source node in a new stream
2) Edit the **Statistics File** node
3) Select the file **customer_offers.sav** from the c:\Train\Modeler_AutomatedDM directory
4) Select the **Use field format information to determine storage** check box

These actions are all that is required to successfully read the *customer_offers.sav* file into Modeler. There are several other data characteristics that should be reviewed or set for the data, which we turn to next.

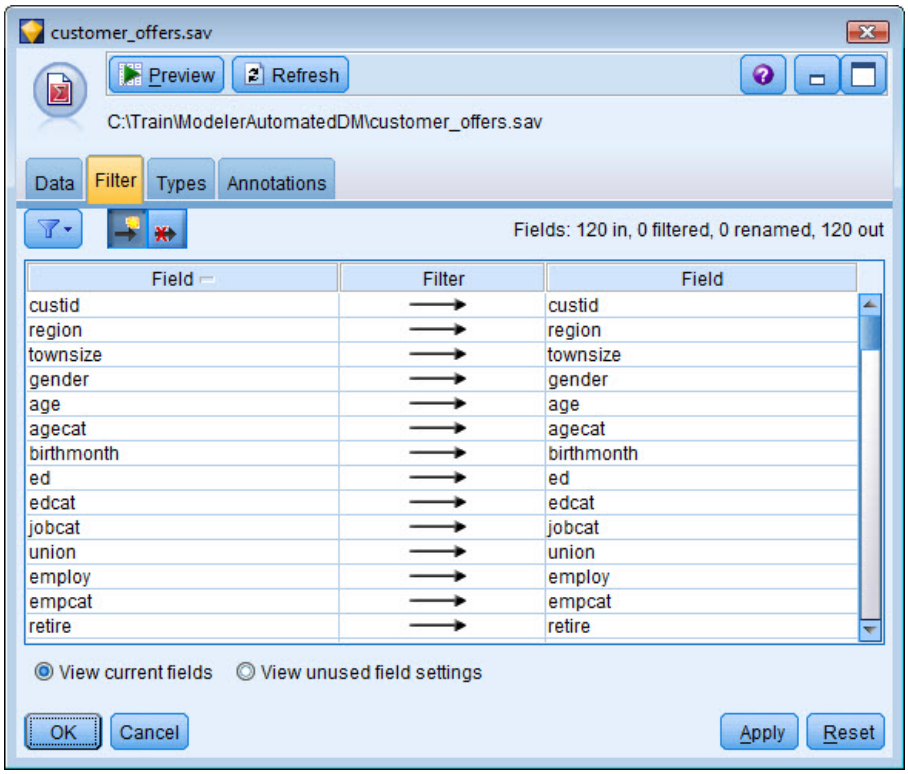
Filter Tab

We may want to edit one or more field names or even decide that some fields should not be read into Modeler and retained downstream. The Filter tab allows us to take these actions.

1) Select the **Filter** tab

The left column contains the field names as read from the data file. We can specify new names in the right column. The middle column shows an arrow that can be interpreted as "becomes".

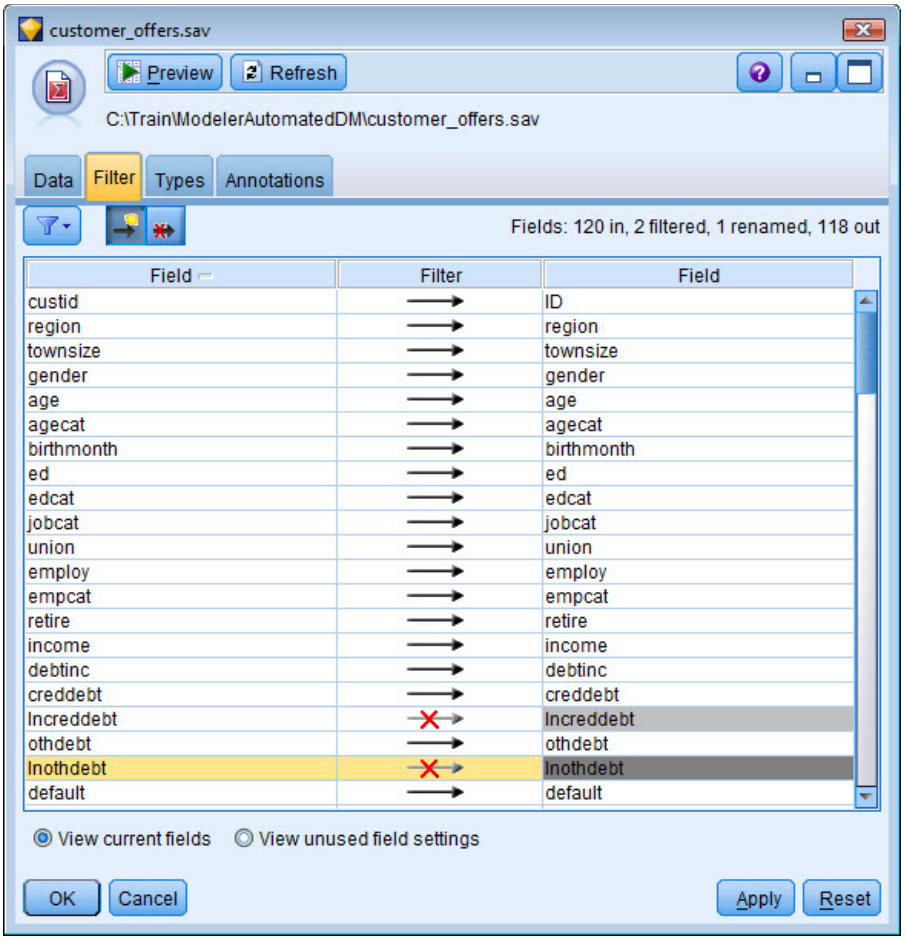
Figure 4.3 Filter Tab



As an example, suppose we would like to rename *custid* to *ID* and would like to exclude the fields *Increddebt* and *Inothdebt*.

1) Double-click on **custid** in the right Field column
2) Change the text to **ID**
3) Select the **arrow** in the **Increddebt** row
4) Select the **arrow** in the **Inothdebt** row

Figure 4.4 Filter Tab Changing Field Names and Excluding a Field



The crossed red arrows indicate that data for that field won't be read into Modeler.

Within the Filter tab, the user can sort the fields (just click on column header Field), exclude all fields at once by clicking the button or include all fields at once by clicking the button. Furthermore, the filter menu options button gives access to numerous filter options such as including/excluding all fields, toggling between fields, removing or renaming duplicates automatically, and truncating fieldnames.
The Preview button can be used at any time to preview the current state of the data to be read into Modeler.

4.6 *Defining Field Type*

A very important step in reading data is to define the measurement level for each of the fields within the data. The measurement level for each field must be set before the fields can be used in constructing model, and with some other nodes.

However, when doing data exploration, it isn't necessary to specify the field type. But in this instance, the Statistics data file contained metadata information for each field that has helped Modeler begin to type the data, so we'll follow through here and complete the process.

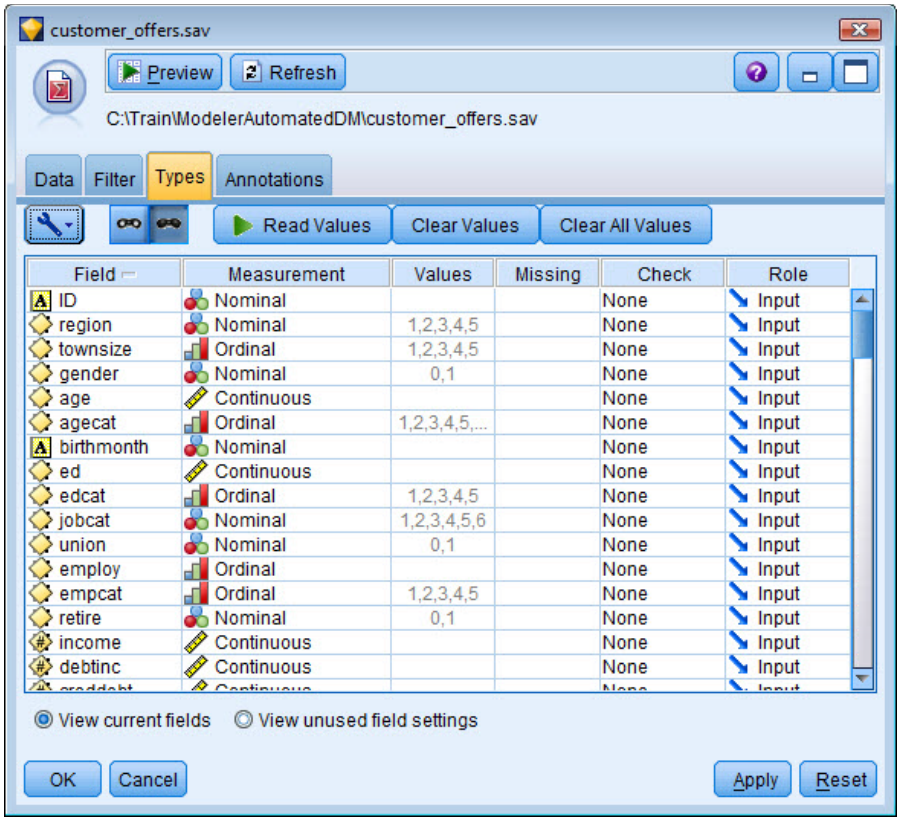
Measurement level can be set in most source nodes (the Types tab) at the same time the data are defined, or in a Type node (located in the Field Ops palette) if one needs to define a measurement level for the field later in a stream (perhaps because a new field was created).

1) Select the **Types** tab

The Types tab in a data source node or the Type node controls the properties of each field: measurement level, data values, role, and missing value definitions. This node also has a Check facility that, when turned on, examines fields to ensure that they conform to specified settings, such as checking whether all the values in a field are within a specified range. This option can be useful for cleaning up data sets in a single operation.

In this section we concentrate on the measurement level and role definitions. Other specifications (missing values) will be discussed in later lessons.

Figure 4.5 Types Tab



Measurement Level Definition

The measurement column in the Types tab of source nodes describes the measurement level of the field, which determines how Modeler will use the field. Modeler distinguishes among several different measurement levels.

- **Continuous.** Used to describe numeric values such as a range of 0-100 or 0.75-1.25. A continuous value may be an integer, real number, or date/time.
- **Categorical.** Used for string values when an exact number of distinct values is unknown. This is an uninstantiated data type, meaning that all possible information about the storage and usage of the data is not yet known.

- **Flag.** Used for data with two distinct values such as Yes/No or 1 and 2.
- **Nominal.** Used to describe data with multiple distinct values, each treated as a member of a set, such as married, single, divorced, etc.
- **Ordinal.** Used to describe data with multiple distinct values that have an inherent order, such as *1 low*, *2 medium*, and *3 high*. Notice, the order is defined by the natural sort order of the data elements. For example, *1*, *3*, *5* is the default sort order for a set of integers, while *high*, *low*, *medium* (ascending alphabetically) is the order for a string field. So, make sure the categories of the field will be ordered correctly when the field is defined as ordinal.
- **Typeless.** Used for data that does not conform to any of the above measurement levels or for a categorical field with too many values. It is useful for cases in which the measurement level would otherwise be categorical with many values (such as an account number). When Typeless is selected for a field's measurement level, the field role is automatically set to None (meaning the field cannot be used in modeling).

Further Information

The default maximum size for sets is 250 unique values. This number can be adjusted or disabled in the Stream Properties dialog.

At this stage the fields in the *customer_offers.sav* file are in a partially instantiated state. *Instantiation* refers to the process of reading or specifying information such as measurement level and values for a field. Fields with totally unknown measurement level are considered *uninstantiated*. Fields are referred to as *partially instantiated* if the program has some information about how they are stored (string or numeric), but the details are incomplete. For example, the Categorical measurement level is temporarily assigned to a string field until it can be determined if it is either a Flag, Nominal or Ordinal measurement level. The Continuous measurement level is given to all numeric fields, whether they are fully instantiated or not.

Another reason a field is partially instantiated is when the measurement level is defined but the range of data (in the Values column) is not known because all the data records have not been read. That is the case for all the fields in the current data file, as the Values cells are either blank or listed in grey.

In reading the data values through the source node, Modeler identifies the measurement level of each field (when the field's Values property is set to *Read* or *Read+*).

- We can force data through the source node by placing and executing a node downstream of it.
- Alternatively we can select the Read Values button, which reads the data completely into the source node or Type node.

We'll use the second option.

1) Select **Read Values** button

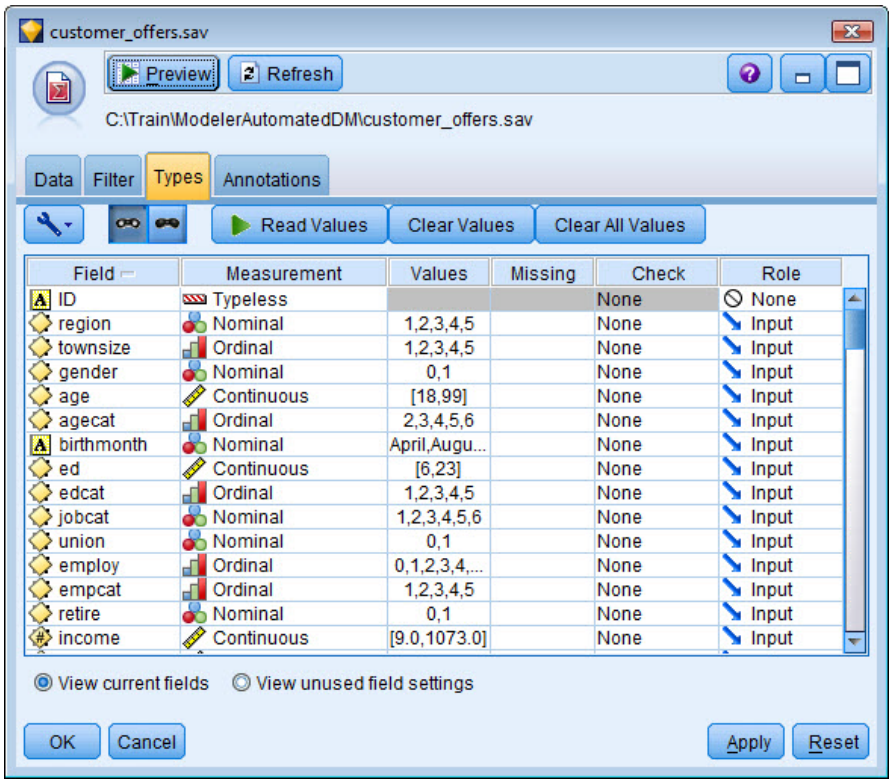
When we do so, rather than immediately reading the data, Modeler supplies an information message, seen in the figure below. This message notes that you can proceed by selecting OK and reading values for all fields, but if instead you meant to read values for only certain fields, you can select Cancel and make alternate specifications.

Figure 4.6 Read Values Information Message

2) Select **OK**

Once this action is completed, each of the fields had values listed. The continuous fields now have a range of values, displayed in brackets. Other fields, such as *edcat*, which is ordinal in measurement level, have its values listed, which are numeric. In general, in the *customer_offers.sav* data file, all the data are numeric, but the categorical fields, including *edcat*, have value labels to identify the category.

Figure 4.7 Fields Fully Instantiated



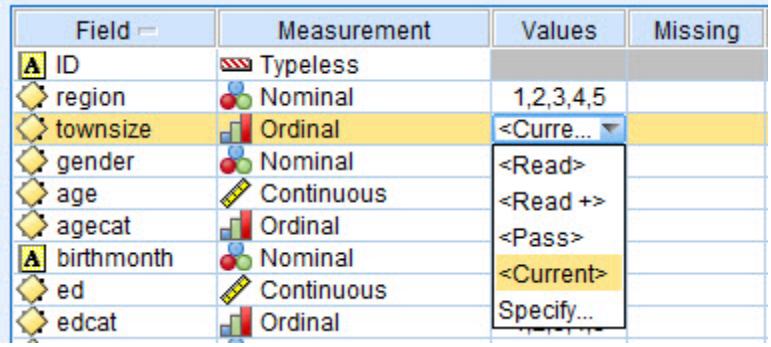
At this point, the user should:

- Review the measurement level for each field to insure that it matches the data or the intent for how that field is to be used (an ordinal field could be treated as nominal, for example)
- Review the set or range of values for each field

Although we aren't familiar with these data, the measurement levels appear generally correct. Let's review a couple of the fields.

1) Click in the Values cell for **townsize**
2) Select **Specify**

Figure 4.8 Values Dropdown Choices

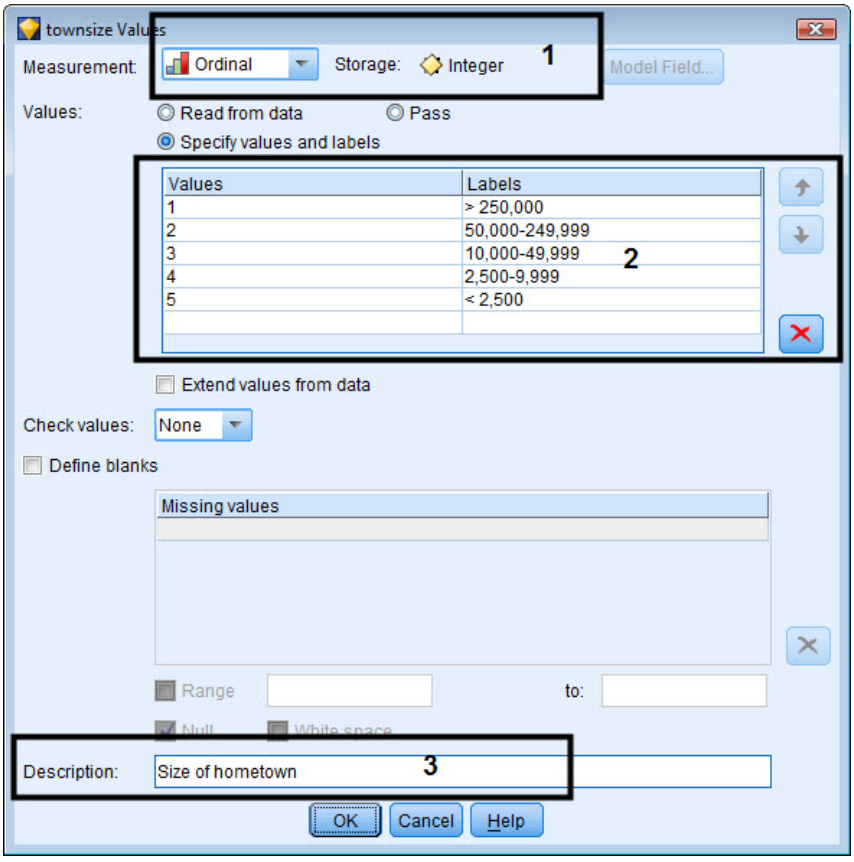


The Values dialog contains several areas listing various characteristics of a field. These include:

1) The Measurement and Storage settings. For *townsize*, the measurement is ordinal, and the storage is integer.
2) The Values area contains the data values and labels for those values. Here the data are integers from 1 to 5. The labels importantly identify the population size corresponding to each value. We also see that, perhaps contrary to expectation, the smallest integer corresponds to the greatest population value. This is obviously critical for data interpretation.
3) The field label is included in the Description box. Here we see that this field stores the population of the customer's home town.

There is also a section dealing with blanks (missing values) that we will use in the next lesson.

Figure 4.9 Values Dialog for townsize

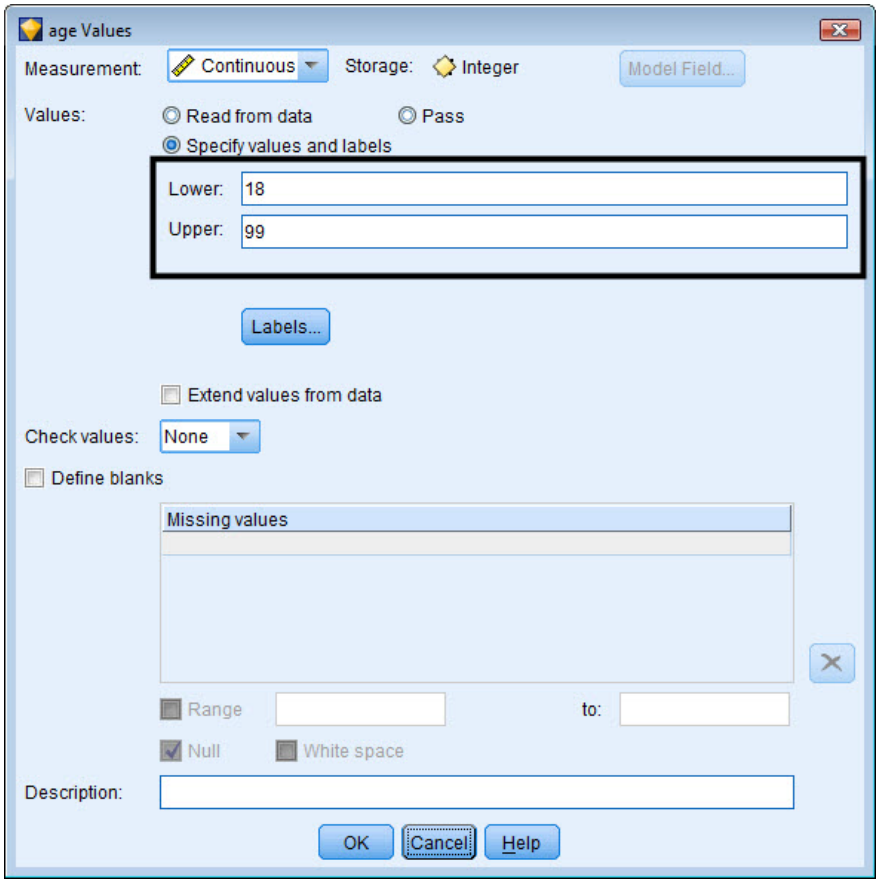


1) Select **OK**
2) Select **Specify** in the Values cell for **age**

The field *age* is continuous in measurement. For this type of field, separate values are not listed. Instead, as highlighted in the figure below, the lower and upper values in the data are listed. Value labels are not needed for a continuous field.

Otherwise, the Values dialog is similar in features to that for categorical fields.

Figure 4.10 Values Dialog for age



3) Select **OK**

We turn next to the concept of field role.

4.7 *Field Role*

The role of a field defines how that field will be used in modeling, or nodes related to modeling, such as Feature Selection. The available roles are:

Table 4.1 Available Field Roles

Input	The field will be used as an input or predictor to a modeling technique. (i.e., a value on which predictions will be based).
Target	The field will be the target for a modeling technique. (i.e. the field to be predicted).
Both	Allows the field to be both an input and a target in an association rule. All other modeling techniques will ignore the field.
None	The field will not be used in modeling.
Partition	Indicates a field used to partition the data into separate samples for training, testing, and (optional) validation purposes. We will discuss this option in a later lesson.
Split	Only available for categorical (flag, nominal, ordinal) fields. Specifies that a model is to be built for each possible value of the split field.
Frequency	Only available for numeric fields. Setting this role enables the field value to be used as a frequency weighting factor for the record.
Record ID	Only relevant for the Linear node, where the specified field will be used as the unique record identifier.

Setting the role for a field is done by clicking on the Role cell for a field and choosing the appropriate role from the drop-down list. Multiple fields can be selected and properties like role or measurement level changed from the context menu (right-click on any of the selected fields).

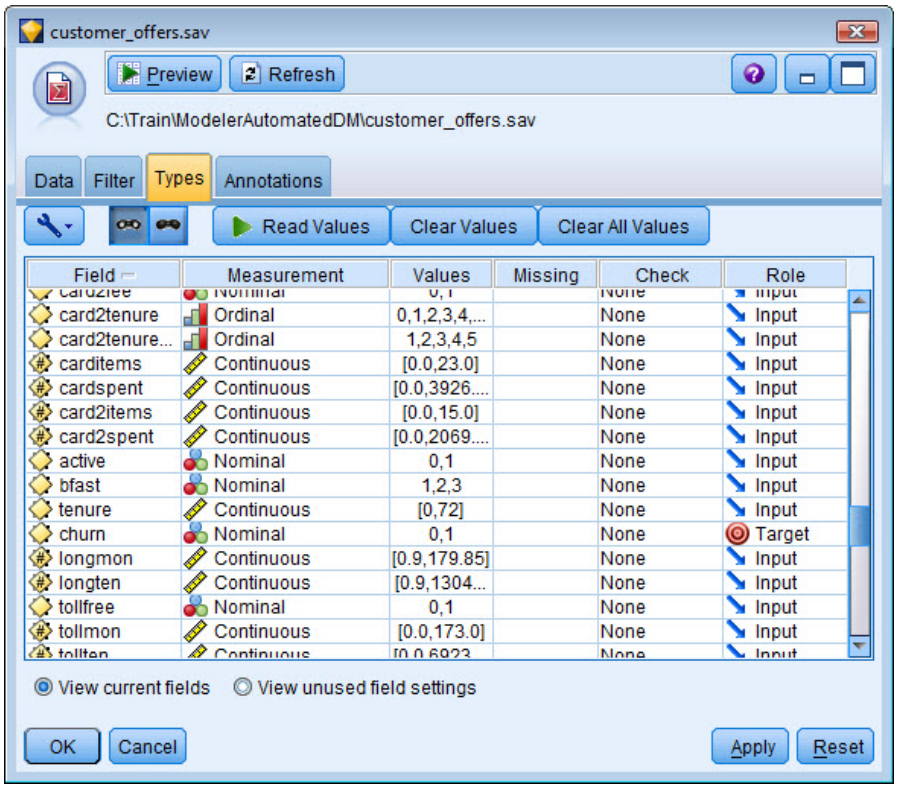
The target and input fields can be set separately in each modeling node, but there is a great advantage in setting these roles in the Types tab or a Type node. The roles are automatically picked up by any modeling node downstream and so don't need to be set every time a new model is added.

Note that the role for *ID* has been automatically set to None. This is because it contains more than 250 distinct values, and so its measurement level was set to Typeless. And that forced its role, correctly, to None.

Let's make one change, setting the role of *churn* to Target, as we will eventually attempt to predict customer churn for the telecommunications firm.

1) Select the Role cell for **churn** and set its role to **Target**

Figure 4.11 Setting Role for churn



Another Review of the Data

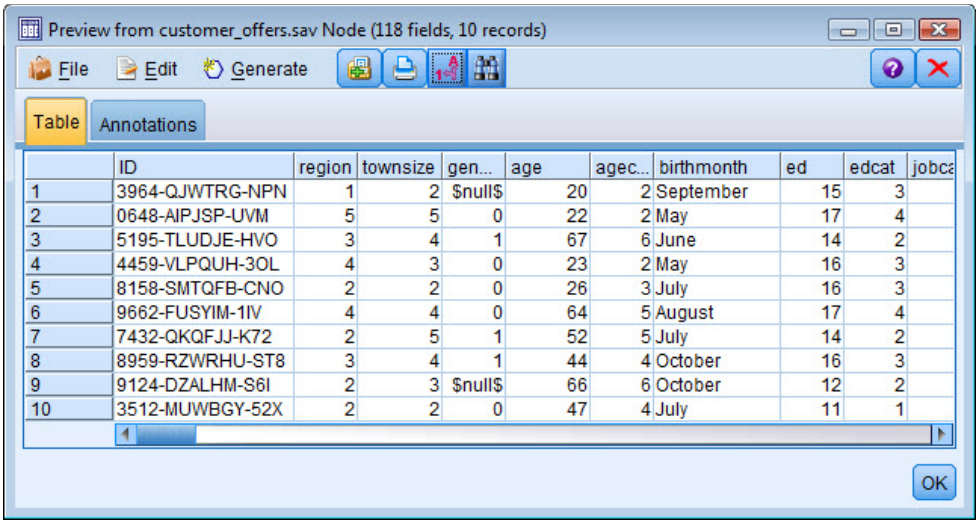
Once some changes have been made in the Type node, it is often advisable to check whether Modeler is accessing the data correctly. As we did earlier, we can use the Preview button.

1) Select **Preview**

The data preview displays all the fields and the first 10 records. There are some missing values for *gender*, including the first record in the file. Because *gender* has numeric storage, with data values of 1 and 2, the missing data has a value of $null$. For numeric fields, when there is no value in the original data source, or the existing value is invalid for a numeric field, Modeler assigns a missing value, represented by $null$.

We will need to take this into account as we work with this field, and any others with a similar missing value.

Figure 4.12 Preview of Data



	ID	region	townsize	gen...	age	agec...	birthmonth	ed	edcat	jobca
1	3964-QJWTRG-NPN	1	2	$null$	20	2	September	15	3	
2	0648-AIPJSP-UVM	5	5	0	22	2	May	17	4	
3	5195-TLUDJE-HVO	3	4	1	67	6	June	14	2	
4	4459-VLPQUH-3OL	4	3	0	23	2	May	16	3	
5	8158-SMTQFB-CNO	2	2	0	26	3	July	16	3	
6	9662-FUSYIM-1IV	4	4	0	64	5	August	17	4	
7	7432-QKQFJJ-K72	2	5	1	52	5	July	14	2	
8	8959-RZWRHU-ST8	3	4	1	44	4	October	16	3	
9	9124-DZALHM-S6I	2	3	$null$	66	6	October	12	2	
10	3512-MUWBGY-52X	2	2	0	47	4	July	11	1	

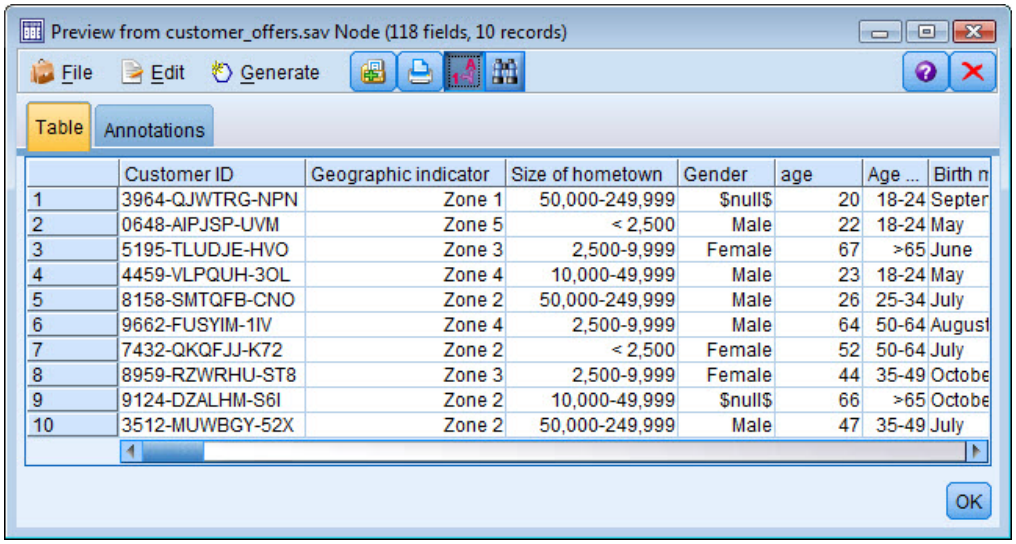
The value labels can be displayed from a Table window instead of the values, as well as the longer field labels.

2) Select the **Display Field and Value Labels** button on the toolbar 

Figure 4.13 Display Field and Value Labels Button on Toolbar

The data are now much easier to read and review. The user can usually toggle between data and field names, or their labels, in output windows.

Figure 4.14 Data with Field Labels and Value Labels Displayed



	Customer ID	Geographic indicator	Size of hometown	Gender	age	Age ...	Birth n
1	3964-QJWTRG-NPN	Zone 1	50,000-249,999	$null$	20	18-24	Septer
2	0648-AIPJSP-UVM	Zone 5	< 2,500	Male	22	18-24	May
3	5195-TLUDJE-HVO	Zone 3	2,500-9,999	Female	67	>65	June
4	4459-VLPQUH-3OL	Zone 4	10,000-49,999	Male	23	18-24	May
5	8158-SMTQFB-CNO	Zone 2	50,000-249,999	Male	26	25-34	July
6	9662-FUSYIM-1IV	Zone 4	2,500-9,999	Male	64	50-64	August
7	7432-QKQFJJ-K72	Zone 2	< 2,500	Female	52	50-64	July
8	8959-RZWRHU-ST8	Zone 3	2,500-9,999	Female	44	35-49	Octobe
9	9124-DZALHM-S6I	Zone 2	10,000-49,999	$null$	66	>65	Octobe
10	3512-MUWBGY-52X	Zone 2	50,000-249,999	Male	47	35-49	July

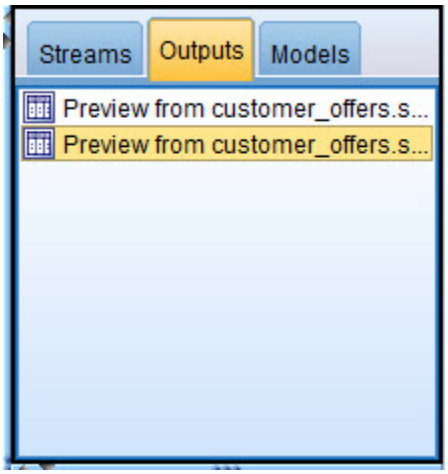
Now that we have checked that the data file has been correctly read by Modeler, we will close the window and return to the Stream Canvas.

Close the Table preview window

Although we have closed the table, the table is still available in the Outputs manager, so we don't have to run the table again to see the data. To activate the Outputs manager:

Click the **Outputs** tab in the upper right of the Modeler window

Figure 4.15 Outputs Tab in Manager



The preview table is still available from the manager. In fact, each output produced (table or chart) will automatically be added as a separate item in the Outputs tab and is available for later use.

Note

For easy reference, a Microsoft Word file with a table of the field names and labels is included in the course files. The file name is *Customer_Offers Field Labels.doc*.

4.8 *Saving a Modeler Stream*

Streams can be saved at any point. Saving a stream is equivalent to saving a set of programming statements, represented by the nodes, to read data, perform data manipulation, and run models. Streams are saved in a file with an extension of "str".

Saving a stream, like saving any program, is *not* equivalent to saving the data the stream has accessed. In fact, data are often not saved separately in Modeler, and there is no special Modeler data format in which to save a data file (more on this in a later lesson).

To save our Modeler stream for later work:

Select **File…Save Stream As.**
Navigate to the **c:\Train\Modeler_AutomatedDM** directory (if necessary)
Type **Customer_Offers** in the File name text box (not shown)
Select the **Save** button

The File menu also allows the user to save (and Open) a State file (which contains the stream and any models stored in the Models palette) and a Project file (which can contain streams, graphs, reports, and generated models, thus organizing elements related to a project). Also, the user can add the saved stream to the current project by clicking the *Add file to project* check box.

Further Information

Additional information on how to import data from various formats into Modeler is available in the *IBM SPSS Modeler User's Guide*.

Apply Your Knowledge

1) True or False? All fields must be read from a data source into Modeler.

2) In what tab in a Source node can a field be renamed?
   a. Field
   b. Data
   c. Filter
   d. Types

3) What Role should be given to fields to be used as predictors in a model?
   a. Both
   b. Independent
   c. Input
   d. Predictor

4.9 *Lesson Summary*

In this lesson we demonstrated how to read a text data file into Modeler.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Read text data files into Modeler and define data characteristics

To support the achievement of the primary objective, students should now also be able to:

- Use a Statistics File node to read a Statistics data file
- Use the Filter tab to filter and rename fields
- Use the Types tab to view measurement level and set field role
- Save a stream file

4.10 Learning Activity

The overall goal of this learning activity is to practice reading a data file and typing the data in the Source node.

Supporting Materials

The Statistics data file *CharityBig.sav*, which contains data from those who responded to a mailed fundraising campaign from a non-profit organization. Information is included on the response, the respondent's previous donations, and basic demographics.

The fields included in the file and their definition are:

Response	Response to campaign; the target
Orispend	Pre-campaign expenditure
Orivisit	Pre-campaign visits
Spendb	Pre-campaign spending category
Visitb	Pre-campaign visits category
Promspd	Post-campaign expenditure
Promvis	Post-campaign visits
Promspdb	Post-campaign spending category
Promvisb	Post-campaign visit category
Totvisit	Total number of visits
Totspend	Total spending
Forpcode	Postal Code
Mos	Mosaic Groups (geodemographic segmentation)
Mosgroup	Mosaic Bands (binned Mosaic groups)
Title	Title
Sex	Gender
Yob	Year of Birth
Age	Age
Ageband	Age Category

1. Clear the Stream Canvas in Modeler, or start with a new stream.
2. Select a Statistics File node and place it on the Stream Canvas.
3. Edit this node and set the file to *CharityBig.sav* in the folder c:\Train|Modeler_AutomatedDM. Use the option to use field information to determine storage. Otherwise use the defaults.
4. In the Types tab, use the Read Values button to instantiate the data.
5. Review each of the fields and their measurement level. Do they all seem appropriate? If not, change the level here. Which field was set to Typeless? Why?
6. Review the labels for categorical fields to see their definitions.
7. Change the Role of *response* (which records who responded to the fundraising campaign) to Target.
8. Save the stream as *Lesson 4 Exercise.str*.

Lesson 5: Data Exploration

5.1 *Objectives*

After completing this lesson students will be able to:

- Review and explore data to look at data distributions and to identify data problems, including missing values

To support the achievement of this primary objective, students will also be able to:

- Describe the types of missing values for fields
- Set missing values for fields
- Use the Data Audit node to explore data distributions
- Use the Data Audit node to impute missing data
- Use the Table node to view the data file

5.2 *Introduction*

Once the data have been accessed in Modeler with a Source node, we are ready for the Data Understanding phase in the CRISP-DM process. In this phase, we are concerned with exploring and becoming thoroughly familiar with the characteristics of our data. We should review the distribution of each field, its range (for continuous fields), outliers, anomalies, and missing values (type and amount). At this time, we can also begin looking for interesting simple patterns in the data, especially relationships between a predictor and a target field.

In this lesson, we will consider issues of data quality and postpone the search for relationships until later.

Data sets always contain problems or errors such as missing information and/or spurious values. Therefore, before data mining can begin, the quality of the data must be assessed. As a general point, the higher the quality of the data used in data mining, the more accurate the predictions and the more useful the results.

Modeler provides several nodes that can be used to automate the investigation of data quality and integrity. In this lesson we introduce the Data Audit node to study several characteristics of each field.

Supporting Materials

The *customer_offers.sav* Statistics data file is used in this lesson. These data are from former and current customers of a telecommunications company. The data includes both demographic and account-related information.

The *customer_offers.str* Modeler stream file is used as the starting point for data exploration and reading the data.

5.3 *Missing Data in Modeler*

Missing data is ubiquitous in almost every data file. It arises for a variety of reasons, but it must be considered carefully for any data mining project. Although we might expect that missing data should be discarded for modeling; that is not always the case. Many analysts have found that missing data

can be useful information. For example, in database marketing, not knowing something about a potential customer (e.g., income) may still be predictive of behavior.

Whether we find missing data to be helpful or not, the first step is to assess the type and amount of missing data for each field. Only then can we decide how to handle it (thus, if there is little missing data, perhaps it will be simpler to discard those records).

In Modeler, missing data is generically labeled *blanks*. It is important to distinguish between the normal use of the word "blank" and Modeler's labeling of missing values with that same term.

In Modeler there are a number of different types of missing data.

1) First, a field may contain no information. Modeler calls such missing information *white space* if the field is string (and it contains one or more blank spaces) and *null value* (non-numeric) if the field is numeric. In addition, if a non-numeric character appears in a numeric field, Modeler also treats this as a null value since no valid value can be created. For numeric fields, this type of missing has the value $null$, as we saw for *gender* in the previous lesson.
2) Second, a string field may be empty, which means that it contains nothing (this is common in databases). This type of missing is called an *empty string*.
3) Finally, predefined codes may be used to represent missing or invalid information. Modeler refers to such codes as *value blanks*.

The Data Audit node reports on such missing values even if they are not declared as missing in the Types tab of a source node (or a Type node), with the exception of value blanks, which the Data Audit node will not recognize unless defined. However, if one knows that such values should be identified as missing values, then there is an advantage in declaring them as data are read, since they then will not appear on the Values list for the field.

Following this logic, we will identify the value of 99 as a value blank for *age*.

Defining Missing Values

Missing values are defined in the Types tab in the source node. If necessary, open the *Customer_Offers.str* stream file.

1) Select **File…Open Stream**
2) Navigate to the c:\Train\Modeler_AutomatedDM folder
3) Select **Customer_Offers.str**

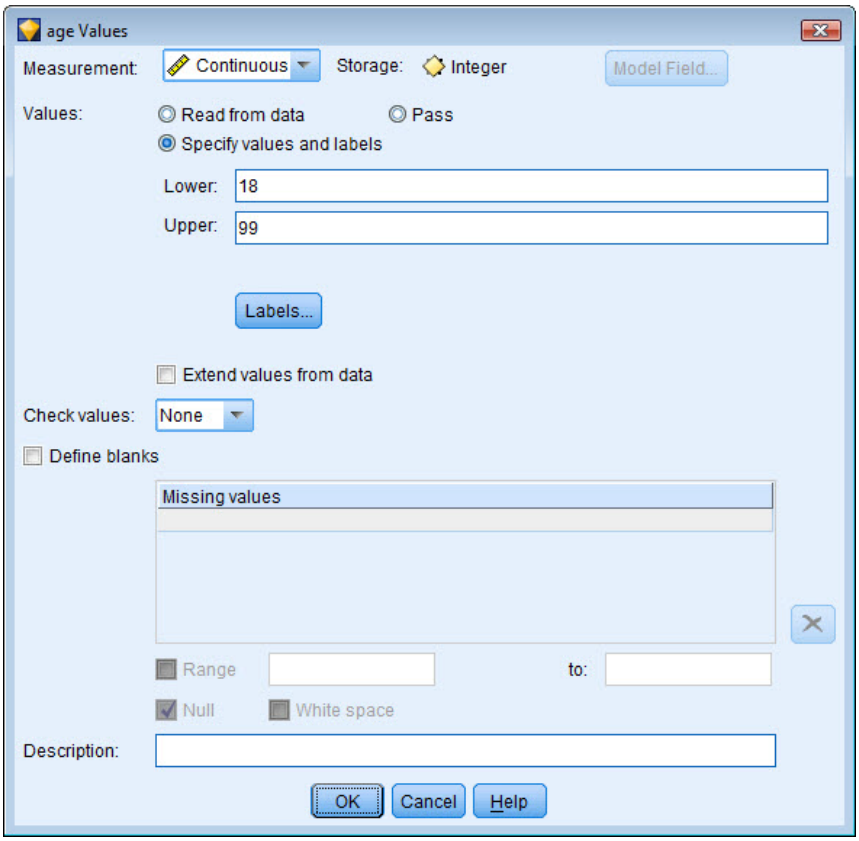
After the stream is open:

4) Edit the **customer_offers.sav** node (double-click the node)
5) Select the **Types** tab

The Missing column in the Types tabs controls whether some data values within a field will be defined as missing. We can open the Values dialog box to set the missing value.

Click the cell in the **Missing** column and **age** row
Select **Specify** from the drop-down menu

Figure 5.1 Values Dialog Box for age



The range for *age* is from 18 to 99. However, the upper value of 99 is not an actual age; instead, any customer whose age is unknown was given this value. It is thus important that this be defined as a value blank in Modeler.

The bottom half of the dialog box includes the section on defining blanks (missing values).

The following options are available.

1) Select the **Define blanks** check box to activate the controls that enable the user to define missing values
2) The **Missing values** table allows the user to define specific values by entering them directly into the text box
3) The **Range** check box and text boxes allow the user to specify a range of values, including for date/time variables and strings (alphabetic order is used)
4) The **Null** and **White space** check boxes allow the user to specify the system null ($null$) and white space (blank or empty strings) as missing. Note that the Null check box is selected by default.

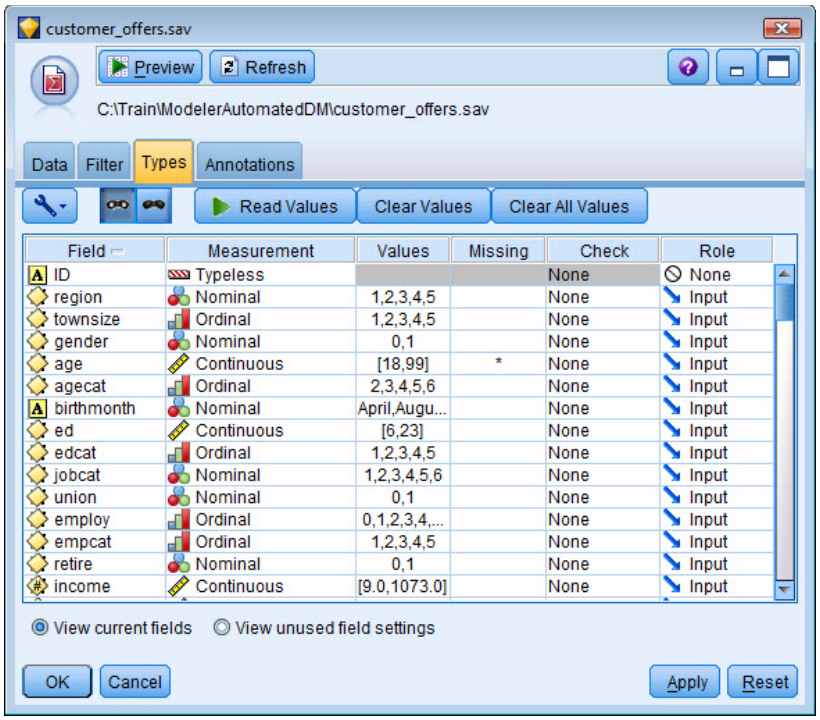
We now define 99 as a value blank for *age*.

1) Select the **Define blanks** check box
2) Enter **99** in the cell under Missing values
3) Select **OK**

The cell in the Missing column for *age* has an asterisk. This indicates that missing values have been defined for this field.

There may well be other missing values that need to be defined, but we'll determine that after exploring the data with the Data Audit node.

Figure 5.2 Missing Value Defined for age



4) Select **OK** to close the source node

5.4 *The Data Audit Node*

The Data Audit node provides a comprehensive first look at the data read into Modeler, presented in an easy-to-read matrix that can be sorted and used to generate full-size graphs and a variety of data preparation nodes. For example, the user can generate a Filter node that excludes fields with too many missing values to be useful in modeling.

When a data field is of categorical measurement, it is of primary interest to see how many unique values there are and how the records are distributed among the categories of that field. We look for categories with either very few, or very many, records. In either instance, this can be a problem for modeling. For numeric fields there is usually interest in the distribution of the data values (histogram) and summary statistics (mean, minimum, maximum, and standard deviation). We look for odd distributions (such as those that are highly skewed). Odd distributions don't hinder some data mining algorithms (such as decision trees) but can be a problem for the more statistically based techniques (such as linear regression).

The Data Audit node is located in the Output palette and is a terminal node (no connections can lead from it).

1) Place a **Data Audit** node from the Output palette into the Stream Canvas and **connect** the **customer_offers.sav** node to it
2) Edit the Data Audit node

The default view shows the usual box in which to select fields for analysis, plus check boxes to control the statistics and graphs produced.

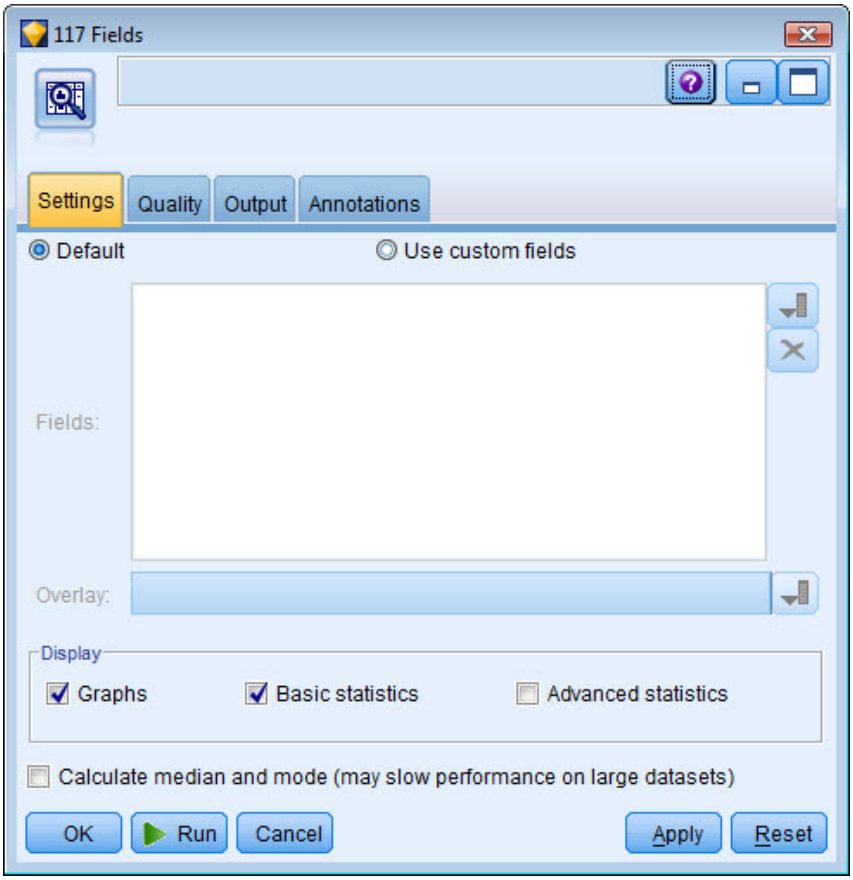
When using large data files, a Sample node may be used to reduce processing time during the initial exploration by selecting only a subset of records for use by the Data Audit node.

Tip

Note the following:

- If there are no Type node settings, all fields are included in the report.
- If there are Type settings (regardless of whether or not they are instantiated), all fields with roles of Input, Target, and Both are included in the analysis.
- If there is a single Target field, it will be used as the Overlay field (here that will be *churn*).

Figure 5.3 Data Audit Node Dialog



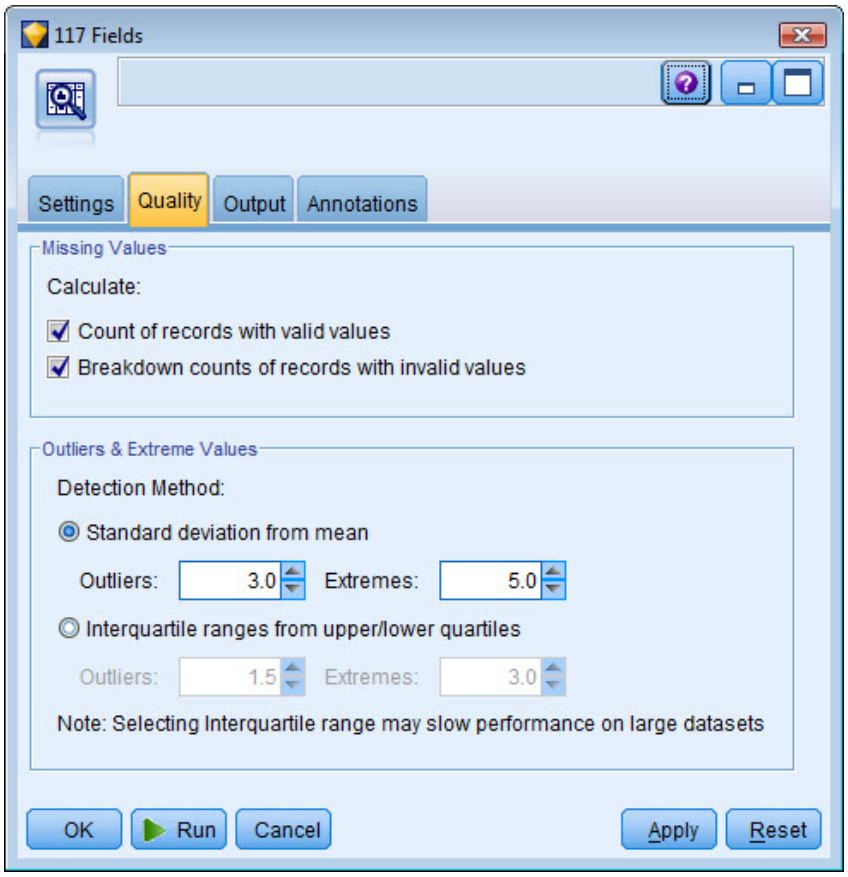
3) Select the **Quality** tab

The options in the Quality tab control checking for missing data and also what values are considered to be outliers and extreme for data checking. Normally, the default selections can be used.

The Missing Values check boxes include:

- **Count of records with valid values.** This option shows the total number of records with valid values for each selected field. Null values, value blanks, white space, and empty strings are always treated as invalid values.
- **Breakdown counts of records with invalid values**. This option shows the number of records with each type of invalid value for each field.

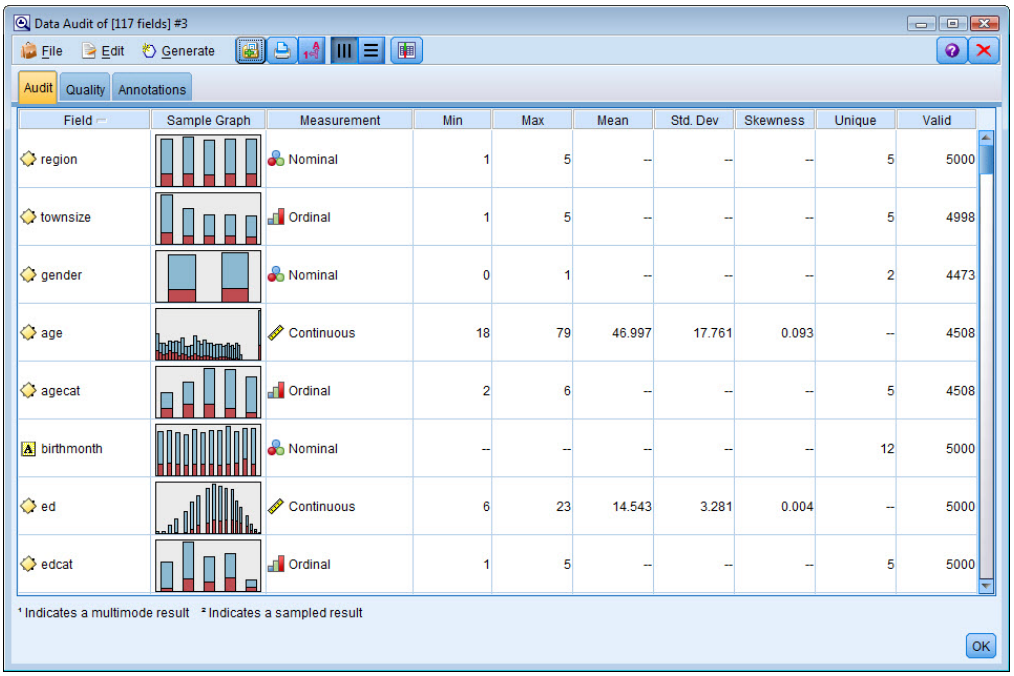
Figure 5.4 Quality Tab in Data Audit Node



We can execute the node with the default settings.

4) Select the **Run** button

Figure 5.5 Data Audit Report on the Audit Tab



The Data Audit browser window provides two views of the data. The Audit tab displays thumbnail graphs, storage icons, and statistics for all fields, while the Quality tab displays information about outliers, extremes, and missing values.

Each row of the Data Audit output represents a field and the columns contain graphs, type information, and statistical summaries. Under default settings in the Data Audit node, every field will have a graph, type information, and a summary of the number of records with valid values for that field (*Valid* column), For continuous fields, the graph in the *Graph* column is a histogram, while categorical (flag, nominal, ordinal) fields are graphed using bar charts (in Modeler they are called distribution charts). Graphs are displayed as thumbnails in the initial report, but full-sized graphs and graph nodes can also be generated by double-clicking on the thumbnails.

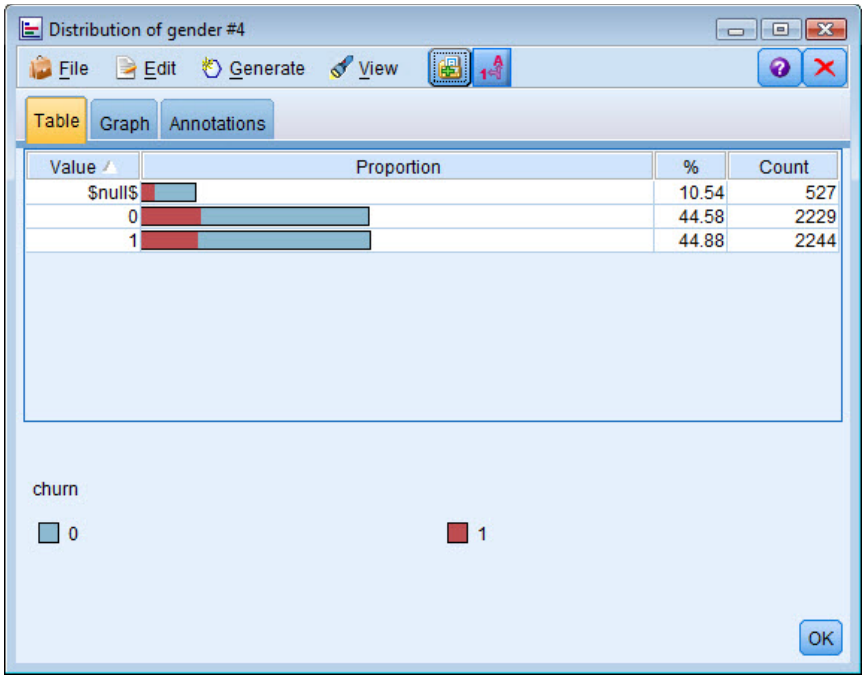
The Unique column lists the number of unique data values for categorical fields.

There are 5,000 records in the file, and most of the fields are listed as having 5,000 valid values. That is not true for *gender* and *age*, which have about 500 missing values each. Let's take a look at the distributions for these fields. Double-clicking on the thumbnail graph opens up a full graph window.

1) Double-click on the Sample Graph thumbnail for **gender**

For distribution tables, count and percent are displayed for each category.

Figure 5.6 Distribution Chart of gender with Overlay of churn



Although the thumbnail graph displayed only two categories, the table view of the field shows all three values, including the missing value of $null$. There are almost an equal number of females and males.

In data mining, we often can use missing values in models for categorical fields, or alternatively, leave the missing values "as is" and let the modeling nodes handle the data as just another value. This strategy can be effective for flag and nominal fields. For ordinal fields, it may or may not be appropriate.

What we don't want to do is select out the missing values from our data stream. For *gender* that would reduce file size by over 10%, which is not acceptable. We return to this issue below.

The target field *churn* is overlaid on the bars, and has values of *No* and *Yes*. It doesn't appear that there is much of a relationship between *gender* and *churn* because the proportion of no and yes in each bar is about the same, including for the $null$ category.

Further Information

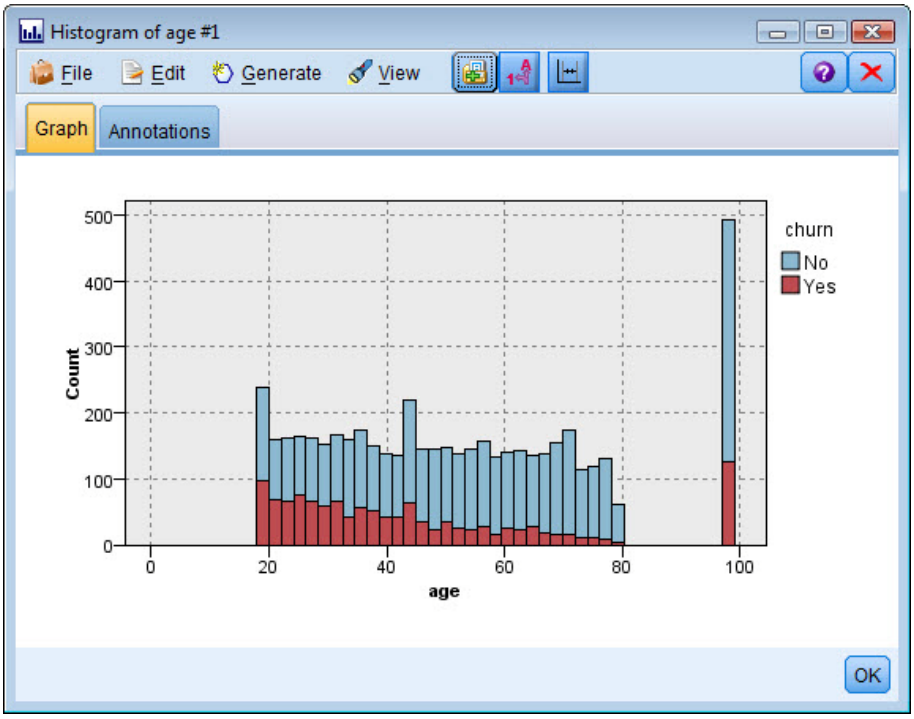
The Graph tab in the Distribution window lets the user view and edit just the bar chart for the displayed field.

Next we'll look at a histogram for *age*.

1) Close the Distribution window
2) Double-click on the Sample Graph thumbnail for **age**

The Histogram shows the frequency of occurrence of values for numeric fields. In a histogram, the range of data values is split into bands (buckets) and bars representing the number of records falling into each band are displayed.

Figure 5.7 Histogram of age with overlay of churn



Perhaps surprisingly, the value of 99 which we defined as missing is still displayed. Histograms, and charts in general, will display data values, even when they are defined as missing. This allows the user to view the full distribution. But the number of valid values for *age* tells us indirectly that 99 has indeed been defined as a value blank.

Age is distributed quite evenly over its range. Intriguingly, there is a definite relationship between *churn* and *age*. As *age* increase, the proportion of those who churn (Yes) decreases fairly steadily. This indicates that we should use *age* when predicting *churn*.

However, as with *gender*, about 10% of the records are missing *age*. As with *gender*, we don't want to drop these cases when constructing models, but here, we cannot treat 99 as a valid value. We will need to *impute* or estimate the value of *age*. We discuss this more when viewing the Quality tab output next.

3) Close the Histogram window

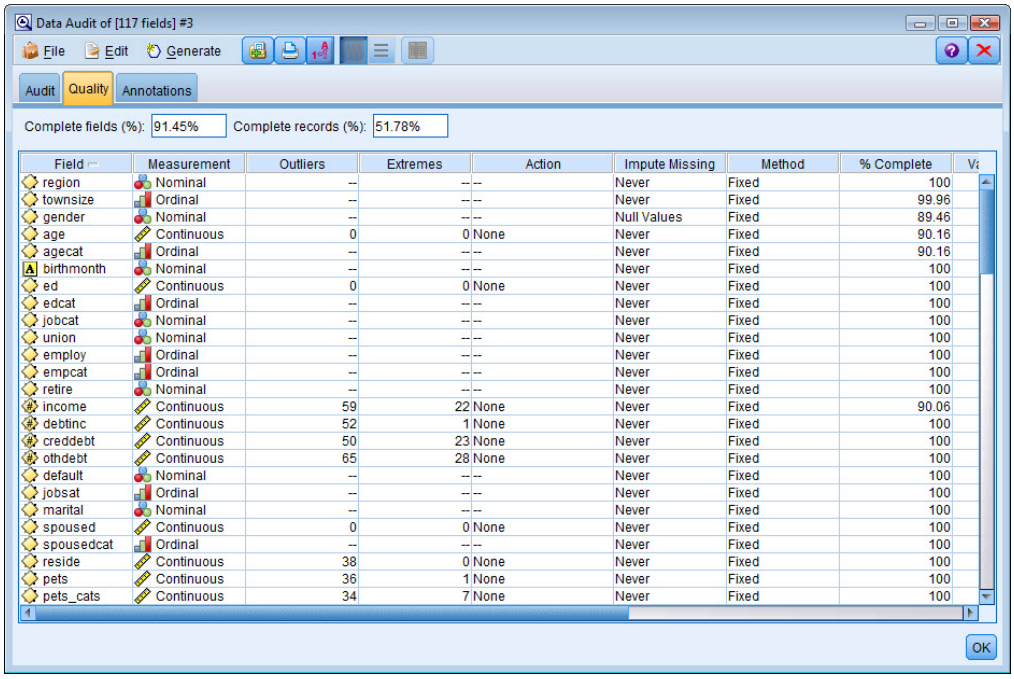
5.5 *The Quality Tab*

The Quality tab in the Data Audit browser window displays information by field on outliers and extreme data values for continuous fields, and it also displays information on the various types of missing data for a field. Along with these are options for handling missing values, outliers, and extreme values.

1) Select the **Quality** tab

As defined in the Quality tab of the Data Audit node itself, the number of records that are outliers, or extreme, values are listed for continuous fields. For example, *income* has 59 outliers and 22 extreme values.

Figure 5.8 Quality Tab in Data Audit Browser



Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method	% Complete	V
region	Nominal	--	--	--	Never	Fixed	100	
townsize	Ordinal	--	--	--	Never	Fixed	99.96	
gender	Nominal	--	--	--	Null Values	Fixed	89.46	
age	Continuous	0	0	None	Never	Fixed	90.16	
agecat	Ordinal	--	--	--	Never	Fixed	90.16	
birthmonth	Nominal	--	--	--	Never	Fixed	100	
ed	Continuous	0	0	None	Never	Fixed	100	
edcat	Ordinal	--	--	--	Never	Fixed	100	
jobcat	Nominal	--	--	--	Never	Fixed	100	
union	Nominal	--	--	--	Never	Fixed	100	
employ	Ordinal	--	--	--	Never	Fixed	100	
empcat	Ordinal	--	--	--	Never	Fixed	100	
retire	Nominal	--	--	--	Never	Fixed	100	
income	Continuous	59	22	None	Never	Fixed	90.06	
debtinc	Continuous	52	1	None	Never	Fixed	100	
creddebt	Continuous	50	23	None	Never	Fixed	100	
othdebt	Continuous	65	28	None	Never	Fixed	100	
default	Nominal	--	--	--	Never	Fixed	100	
jobsat	Ordinal	--	--	--	Never	Fixed	100	
marital	Nominal	--	--	--	Never	Fixed	100	
spoused	Continuous	0	0	None	Never	Fixed	100	
spousedcat	Ordinal	--	--	--	Never	Fixed	100	
reside	Continuous	38	0	None	Never	Fixed	100	
pets	Continuous	36	1	None	Never	Fixed	100	
pets_cats	Continuous	34	7	None	Never	Fixed	100	

The column labeled Action provides options to handle these values. Often for modeling, it is best to reduce the impact of outliers for a field, as they can skew model effects.

The following actions, accessed by clicking in the Actions cell for a field, are available for handling outliers and extreme values:

- **Coerce.** Replaces outliers and extreme values with the nearest value that would not be considered extreme. For example if an outlier is defined to be anything above or below three standard deviations, then all outliers would be replaced with the highest or lowest value within this range.
- **Discard.** Discards records with outlying or extreme values for the specified field.
- **Nullify.** Replaces outliers and extremes with the null value.
- **Coerce outliers / discard extremes.** Discards extreme values only.
- **Coerce outliers / nullify extremes.** Nullifies extreme values only

After making selections in the Action column, a Supernode must be generated from the Generate…Outlier & Extreme Supernode menu selection that will apply the necessary data transformations. This node must be added to the stream between the data source and any modeling nodes.

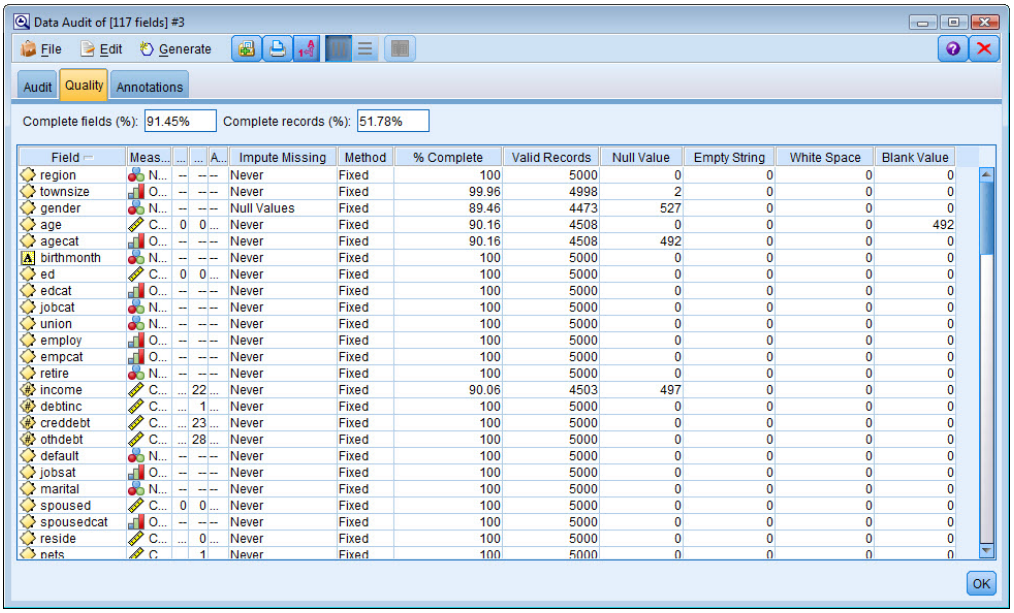
Some types of models are less affected by extreme data values, such as CHAID. But most models, including Neural Net, Logistic, and C&R Tree, will be affected by outliers on the predictors.

Tip

Often a good strategy is to try a model with the data "as is," and then another model with the outliers transformed.

To the right are columns that supply information on the missing data. Listed are the percentages of complete records for each field, along with the number of valid, null, and blank values. The user can choose to impute missing values for specific fields as appropriate from the Impute Missing column and then generate a SuperNode to apply these transformations.

Figure 5.9 Missing Value Information for Fields



Field	Meas...	...	...	A...	Impute Missing	Method	% Complete	Valid Records	Null Value	Empty String	White Space	Blank Value
region	N...	--	--	--	Never	Fixed	100	5000	0	0	0	0
townsize	O...	--	--	--	Never	Fixed	99.96	4998	2	0	0	0
gender	N...	--	--	--	Null Values	Fixed	89.46	4473	527	0	0	0
age	C...	0	0	...	Never	Fixed	90.16	4508	0	0	0	492
agecat	O...	--	--	--	Never	Fixed	90.16	4508	492	0	0	0
birthmonth	N...	--	--	--	Never	Fixed	100	5000	0	0	0	0
ed	C...	0	0	...	Never	Fixed	100	5000	0	0	0	0
edcat	O...	--	--	--	Never	Fixed	100	5000	0	0	0	0
jobcat	N...	--	--	--	Never	Fixed	100	5000	0	0	0	0
union	N...	--	--	--	Never	Fixed	100	5000	0	0	0	0
employ	O...	--	--	--	Never	Fixed	100	5000	0	0	0	0
empcat	O...	--	--	--	Never	Fixed	100	5000	0	0	0	0
retire	N...	--	--	--	Never	Fixed	100	5000	0	0	0	0
income	C...	...	22	...	Never	Fixed	90.06	4503	497	0	0	0
debtinc	C...	...	1	...	Never	Fixed	100	5000	0	0	0	0
creddebt	C...	...	23	...	Never	Fixed	100	5000	0	0	0	0
othdebt	C...	...	28	...	Never	Fixed	100	5000	0	0	0	0
default	N...	--	--	--	Never	Fixed	100	5000	0	0	0	0
jobsat	O...	--	--	--	Never	Fixed	100	5000	0	0	0	0
marital	N...	--	--	--	Never	Fixed	100	5000	0	0	0	0
spoused	C...	0	0	...	Never	Fixed	100	5000	0	0	0	0
spousedcat	O...	--	--	--	Never	Fixed	100	5000	0	0	0	0
reside	C...	...	0	...	Never	Fixed	100	5000	0	0	0	0
pets	C		1		Never	Fixed	100	5000	0	0	0	0

We see that *gender* has 527 null values, as we learned when viewing its Distribution graph above. With the value of 99 defined as blank for *age*, that field is listed as having only 4508 valid records, and the 492 missing values are categorized correctly as Blank Values. Scrolling through the fields, the missing data are almost entirely comprised of null values.

At the top of the window, the percentage of fields with no missing data (Complete fields (%)) and the number of records with no missing data (Complete records (%)) is displayed.

This latter number can be quite critical for modeling. Although we will not use every available field for modeling, if we don't adjust for missing data and use many of these fields in a model, we could lose a substantial fraction of the available records. Or, alternatively, lots of missing data would have to be imputed. This is a problem when there is a pattern to the missing data such that there are differences between the records with missing data and those with valid data, especially with respect to the target field. If there is, a model can be misestimated and not perform well on future data.

Imputing Missing Data

Although some models automatically adjust for missing data, or use it directly, sometimes it is a good idea to take control of this operation. In addition, the missing data for variables that are nominal or ordinal in measurement may not be properly handled by modeling algorithms, which often set missing

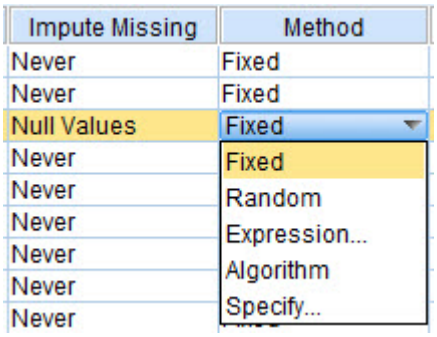
data to the mode or median, respectively. But this type of action wouldn't be appropriate for *gender*, for example.

The Quality tab in the Method column allows the user to choose which missing values to estimate (impute) and then select a method. To illustrate this, we will set the $null$ missing value for *gender* to 9. This will simply make it another value to be used in modeling, and since the measurement level is defined as nominal, it will not be treated as the highest numeric value, but simply another categorical value for *gender*. Further, in this way, those models that drop records with missing data for a field will not drop those records for *gender*. We can also label this value in a Type node downstream.

1) Select the Impute Missing cell for **gender** and select **Null Values**
2) Select the Method cell for **gender**

The dropdown list in the Method cell shows several options to impute missing values, including a constant (Fixed), a random value (based on the mean and standard deviation of the field), an equation (Expression), or an algorithm (a model, C&R Tree).

Figure 5.10 Missing Value Imputation Methods

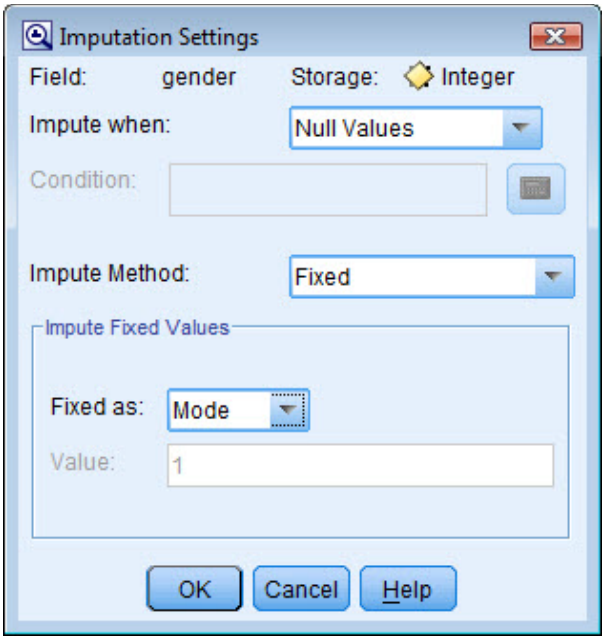


We need to set the fixed value.

3) Select **Specify**

The default constant for *gender* is the mode because the measurement level is set to nominal, but as we have mentioned, that isn't appropriate.

Figure 5.11 Imputation Settings Dialog



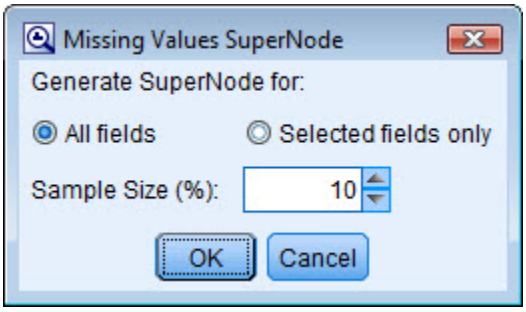
4) Select **Constant** from the Fixed as: dropdown
5) Input **9** in the Value box (not shown)
6) Select **OK**

This sets the condition for doing imputation, but we need to generate a Supernode to use in the stream that will do the actual transformation.

7) Select **gender** to highlight its row
8) Select **Generate…Missing Values SuperNode** from the menu

The intermediate dialog box that appears allows the user to select the fields for which a SuperNode will be generated. In this case, we only want to generate one for the selected field *gender*.

Figure 5.12 Missing Values SuperNode Generate Dialog



9) Select **Selected fields only**
10) Select **OK**
11) Close the Data Audit browser window

The Supernode we need has been automatically created and added to the Stream Canvas. Briefly, a SuperNode groups multiple nodes into a single node by encapsulating sections of a data stream (although in this instance all that is used is a Filler node). A SuperNode has a special icon with a star shape.

Figure 5.13 Missing Values Imputation SuperNode Added to Stream

We can now check to insure that it does what we requested. To do this, we use a Table node.

5.6 *Viewing Data with the Table Node*

The Table node, stored in the Output palette, displays the data in spreadsheet-like format, with one row per record, and field names or labels heading the columns. Since it is an often-used node, the Table node is included on the Favorites palette.

We can now add a Table node to the stream and view the full data file.

1) Move the **SuperNode** to the right of the customer_offers.sav source node
2) **Connect** the customer_offers.sav node to the SuperNode
3) Add a **Table** node to the stream from the Output palette
4) **Connect** the SuperNode to the Table node (not shown)

By default the Table node will display its output on the screen, although it can be sent to a file in several different formats.

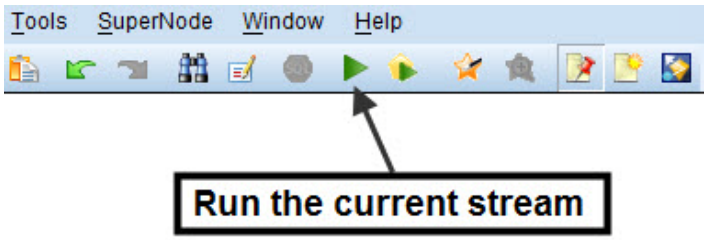
Executing a Node

We don't need to edit a node to execute it.

- When editing a node, we use the Run button to execute our instructions in the node
- Alternatively, a node can be executed by using the context menu and selecting Run (the equivalent keystroke sequence is Ctrl+E).

When there is only one terminal node in the stream (Table is a terminal node), the *Run the current stream* button can also be used—see figure below). However, if there are multiple terminal nodes, all of them will be executed when the Run button is selected, and the Data Audit node is also a terminal node included in this stream.

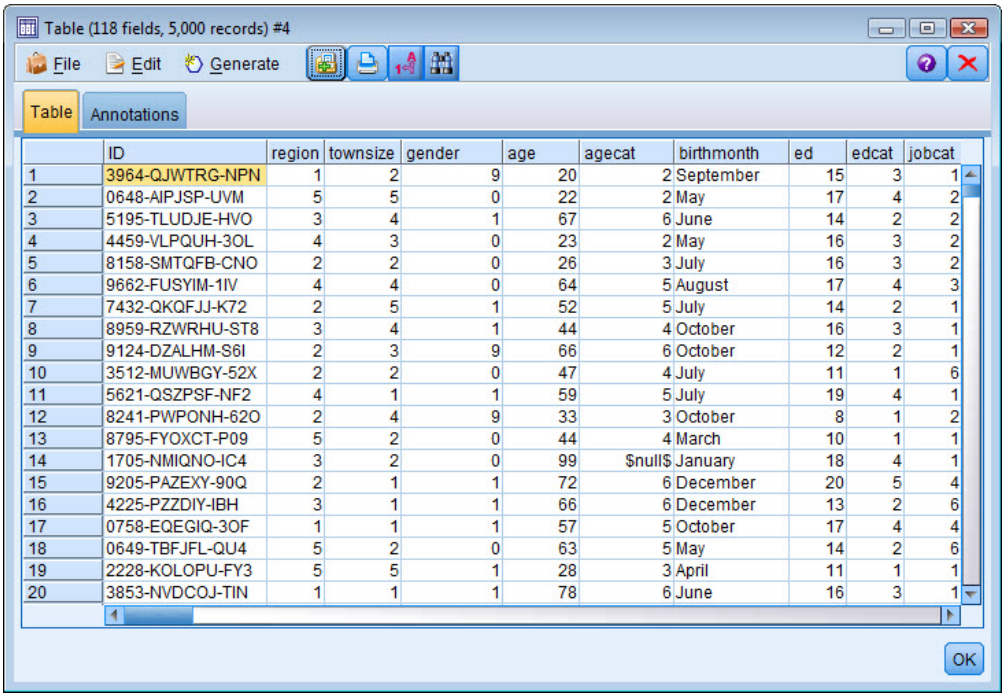
Figure 5.14 Run Button on Toolbar



1) Right-click on the Table node
2) Select **Run** from the Context menu

The title bar of the Table window displays the number of fields and records read into the table. The Table is similar to the Preview option in a node, with the exception that it displays all the records.

Figure 5.15 Table Window Showing Data from customer_offers.sav



	ID	region	townsize	gender	age	agecat	birthmonth	ed	edcat	jobcat
1	3964-QJWTRG-NPN	1	2	9	20	2	September	15	3	1
2	0648-AIPJSP-UVM	5	5	0	22	2	May	17	4	2
3	5195-TLUDJE-HVO	3	4	1	67	6	June	14	2	2
4	4459-VLPQUH-3OL	4	3	0	23	2	May	16	3	2
5	8158-SMTQFB-CNO	2	2	0	26	3	July	16	3	2
6	9662-FUSYIM-1IV	4	4	0	64	5	August	17	4	3
7	7432-QKQFJJ-K72	2	5	1	52	5	July	14	2	1
8	8959-RZWRHU-ST8	3	4	1	44	4	October	16	3	1
9	9124-DZALHM-S6I	2	3	9	66	6	October	12	2	1
10	3512-MUWBGY-52X	2	2	0	47	4	July	11	1	6
11	5621-QSZPSF-NF2	4	1	1	59	5	July	19	4	1
12	8241-PWPONH-62O	2	4	9	33	3	October	8	1	2
13	8795-FYOXCT-P09	5	2	0	44	4	March	10	1	1
14	1705-NMIQNO-IC4	3	2	0	99	$null$	January	18	4	1
15	9205-PAZEXY-90Q	2	1	1	72	6	December	20	5	4
16	4225-PZZDIY-IBH	3	1	1	66	6	December	13	2	6
17	0758-EQEGIQ-3OF	1	1	1	57	5	October	17	4	4
18	0649-TBFJFL-QU4	5	2	0	63	5	May	14	2	6
19	2228-KOLOPU-FY3	5	5	1	28	3	April	11	1	1
20	3853-NVDCOJ-TIN	1	1	1	78	6	June	16	3	1

Looking at the first record, we see that the value of *gender* is now 9, as we intended.

The File menu in the Table window allows the user to save the table, print the table, export the table, or publish the table to the web. Using the Edit menu the user can copy values or fields. The Generate menu allows the user to automatically generate Select (data selection) and Derive (field calculation) nodes.

Now that we have checked that the data file has been correctly read into Modeler, we will close the window and return to the Stream Canvas.

Close the Table window

We can save the file for later use.

Select **File…Save Stream As.**
Navigate to the **c:\Train\Modeler_AutomatedDM** directory (if necessary)
Type **Customer_Offers_Data Audit** in the File name text box (not shown)
Select the **Save** button

Note on Data Preparation

To save time in data preparation, we will use a stream file in the next lesson named *Customer_Offers_Data Audit_Complete.str*. It has adjustments for other nominal fields with missing data similar to what we did for *gender*, and imputes data for *age* and *income*, all done from the Data Audit node. We'll review those actions in the next lesson.

Apply Your Knowledge

1) Which of these are types of missing values in Modeler? Select all that apply.
    a. Null
    b. White space
    c. Invalid
    d. Blank

2) What is the default method to define outlier and extreme values within the Data Audit node?
    a. Based on the interquartile range
    b. Based on the standard deviation
    c. Based on the skewness

3) Which of these is not an imputation method for missing data?
    a. Algorithm
    b. Random
    c. Mean
    d. Fixed

5.7 *Lesson Summary*

In this lesson we demonstrated how to explore data to be used for modeling.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Review and explore data to look at data distributions and to identify data problems, including missing values

To support the achievement of the primary objective, students should now also be able to:

- Describe the types of missing values for fields
- Set missing values for fields
- Use the Data Audit node to explore data distributions
- Use the Data Audit node to impute missing data
- Use the Table node to view the data file

5.8 Learning Activity

The overall goal of this learning activity is to review missing values in Modeler and to use the Data Audit node on the charity data.

Supporting Materials

The stream file *Lesson 4 Exercise.str*. If this file was not created, you can use *Backup_Lesson 4 Exercise.str*.

1. Open the stream file *Lesson 4 Exercise.str*.
2. Edit the Source node. On the Types tab, note that blanks are already defined for two fields. Which ones?
3. What types of blank values are defined for each field?
4. Add a Data Audit node to the stream and connect it to the Source node. Run the Data Audit node.
5. Review the distributions for each of the fields? What percentage of people responded to the charity's fundraising campaign (*response*)? Do you find any possible issues or problems when using them to predict *response*? Which fields are skewed? Which fields have outliers and extreme values?
6. Open the Quality tab. Is there any missing data in the file? Do you need to impute missing data?
7. Save the stream file as *Lesson 5 Exercise.str*.

Lesson 6: Automated Data Preparation

6.1 *Objectives*

After completing this lesson students will be able to:

- Use the Automated Data Prep node to further prepare data for modeling

To support the achievement of this primary objective, students will also be able to:

- Use the Type node to set characteristics for fields
- Describe the various features and capabilities of the Automated Data Prep node
- Use settings of the Automated Data Prep node that are appropriate for the data and modeling objectives
- Describe the types of output produced by the Automated Data Prep node

6.2 *Introduction*

Preparing data for analysis is one of the most important steps in any project—and traditionally, one of the most time consuming. Modeler provides several methods of automating this step in the CRISP-DM process. In this lesson we focus on Automated Data Preparation (ADP), which can analyze the data and screen out fields that are problematic or not likely to be useful, derive new attributes when appropriate, and improve performance through various screening techniques. We can use the node in fully automatic fashion, allowing it to choose and apply changes, or we can use it in interactive fashion, previewing the changes before they are made.

Using ADP enables us to make our data ready for model building more quickly. As a consequence, models will tend to build and score more rapidly.

Like the Data Audit node, ADP can make substitutions for missing data as part of its data preparation, and can even adjust the measurement level of fields automatically. It can also extract information from date and time fields for use in modeling.

In this lesson we will apply ADP to the telecommunications customer data to continue the process of data preparation.

	The *customer_offers.sav* data file. These data are from former and current customers of a telecommunications company. The data includes both demographic and account-related information.
Supporting Materials	The stream file *Customer_Offers_Data Audit_Complete.str*, which contains the result of previous data preparation.

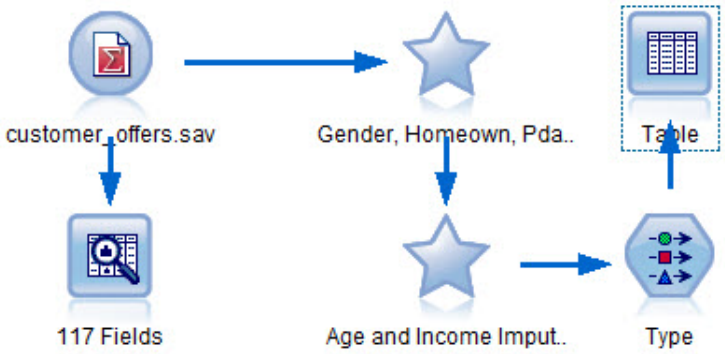
6.3 *The Type Node*

We learned about the data preparation features of the Data Audit node in the previous lesson and used some of them with the *customer_offers.sav* data file. In the interests of time, we didn't take all the actions necessary with the data, so a stream has been created that takes additional steps. The

stream is *Customer_Offers_Data Audit_Complete.str*, and we'll open it and review the current status of the data.

1) Open the stream file **Customer_Offers_Data Audit_Complete.str**

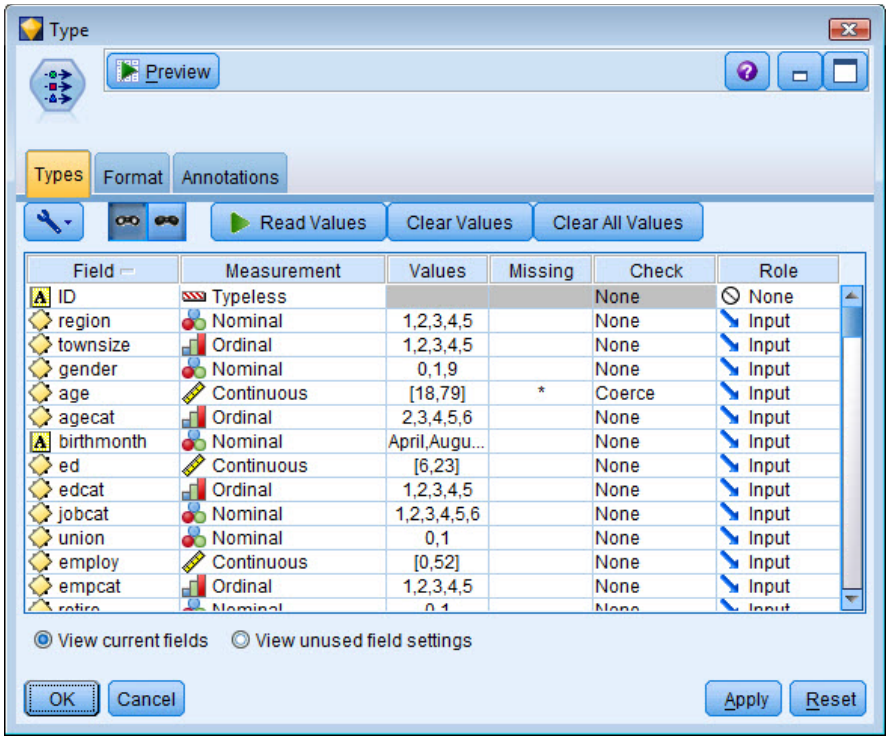
Figure 6.1 Customer_Offers_Data Audit_Complete.str Stream File



To the stream saved in the Data Audit lesson, (*Customer_Offers_Data Audit.str*), a Type node has been added after the two SuperNodes (discussed below).

2) Edit the Type node

Figure 6.2 Type Node Showing Additional Modifications



The Type node is stored in the Fields palette (because it operates on fields). The Type node dialog should look very familiar because it essentially is identical to the Type tab dialog in a source node. Type nodes are used downstream from a source node when new fields have been created, or existing fields modified, because all fields to be used in modeling must be fully instantiated. Invariably the user will use one or more Type nodes when doing automated data mining.

Type nodes downstream pick up typing information from upstream Source nodes or other Type nodes.

In this instance, there were several things to be done to conclude our initial work of data review and preparation. Some of them were done in the Var. File source node that reads the *customer_offers.sav* data file; others were accomplished in the Type node added to the stream.

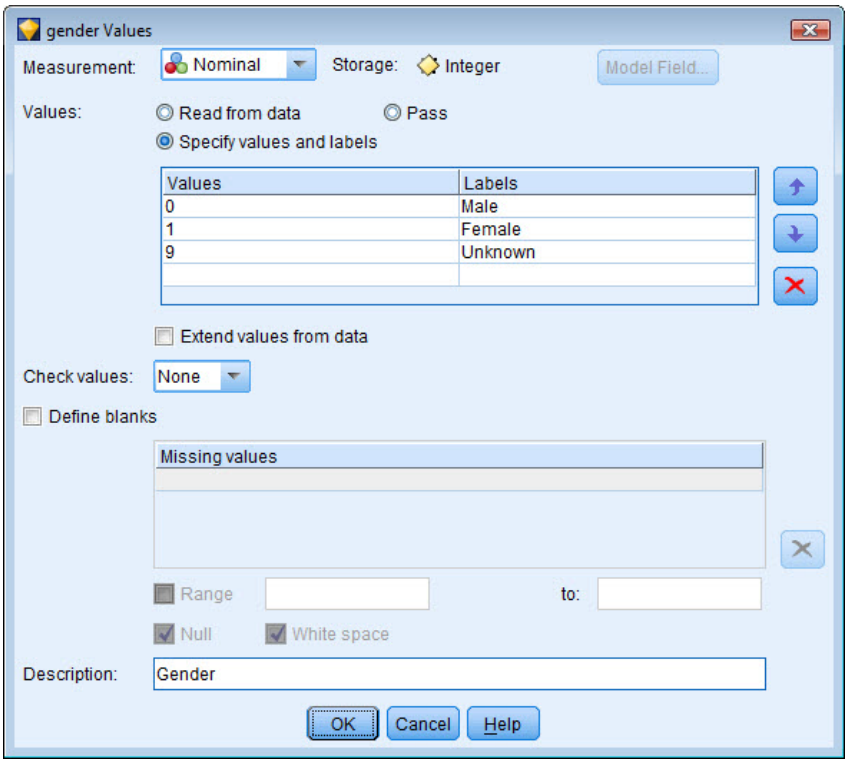
Measurement Level. Some fields were originally typed as ordinal in measurement level but should be continuous. These include *employ*, *address*, *cardtenure*, and *card2tenture*. Their measurement level was changed manually in the original source node.

Imputed Missing Values for Categorical Fields. Following the logic for *gender*, where the value of $null$ was set to 9, missing values for *homeown* and *ownpda* were also set to 9. All of this was done in one SuperNode.

Adding Labels for Categorical Fields. Labels were added for these three fields to identify the values, including 9. This was done in the Type node in the Value dialog.

3) Select the Value cell for **gender** and select **Specify**

Figure 6.3 Labels Added for the Values of gender



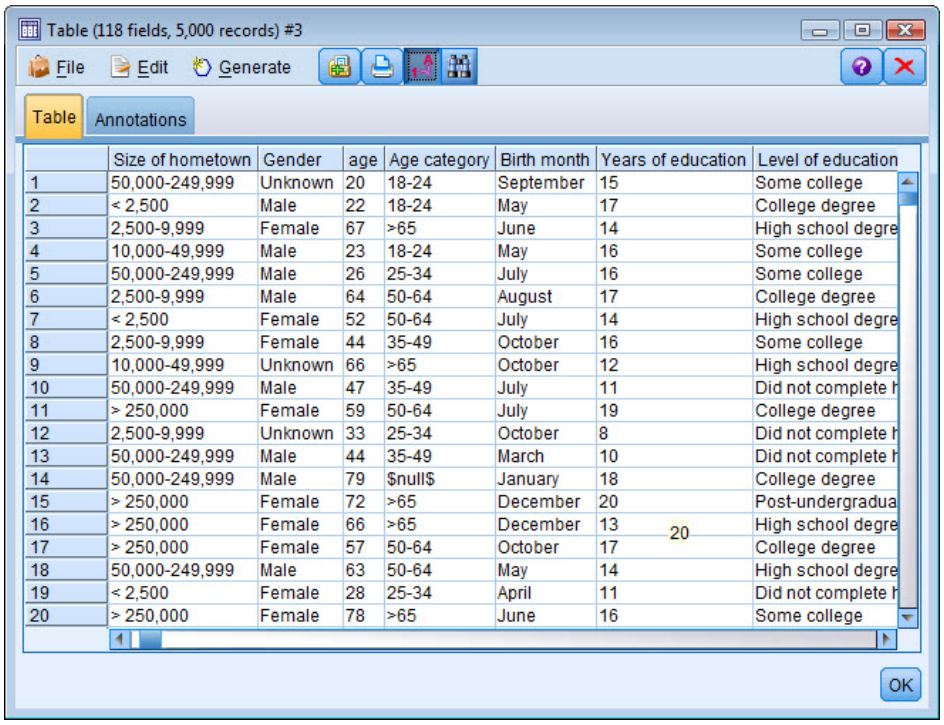
Imputed Missing Values for Continuous Fields. The random method was used to impute missing data for *age* and *income*. This retains the same mean and standard deviation for the fields. This was done in one SuperNode.

Coercing Out of Range Values. Some methods of imputing data for continuous fields can create values are outside the original and appropriate data range for a field. To fix this, the original range was specified in the Type node Values area for *age* and *income*. Then the *Coerce* option in the Check cell was selected; this automatically coerces any values outside the upper or lower limits to those values.

We can view some of these changes by running the Table node attached to the Type node.

4) Select **OK** to close the Values dialog, and then select **OK** again to close the Type node
5) Right-click on the Table node and select **Run**
6) Turn on the display of labels by selecting the **Display field and value labels** button on the toolbar

Figure 6.4 Table Output Showing Modified Fields



	Size of hometown	Gender	age	Age category	Birth month	Years of education	Level of education
1	50,000-249,999	Unknown	20	18-24	September	15	Some college
2	< 2,500	Male	22	18-24	May	17	College degree
3	2,500-9,999	Female	67	>65	June	14	High school degre
4	10,000-49,999	Male	23	18-24	May	16	Some college
5	50,000-249,999	Male	26	25-34	July	16	Some college
6	2,500-9,999	Male	64	50-64	August	17	College degree
7	< 2,500	Female	52	50-64	July	14	High school degre
8	2,500-9,999	Female	44	35-49	October	16	Some college
9	10,000-49,999	Unknown	66	>65	October	12	High school degre
10	50,000-249,999	Male	47	35-49	July	11	Did not complete h
11	> 250,000	Female	59	50-64	July	19	College degree
12	2,500-9,999	Unknown	33	25-34	October	8	Did not complete h
13	50,000-249,999	Male	44	35-49	March	10	Did not complete h
14	50,000-249,999	Male	79	$null$	January	18	College degree
15	> 250,000	Female	72	>65	December	20	Post-undergradua
16	> 250,000	Female	66	>65	December	13	High school degre
17	> 250,000	Female	57	50-64	October	17	College degree
18	50,000-249,999	Male	63	50-64	May	14	High school degre
19	< 2,500	Female	28	25-34	April	11	Did not complete h
20	> 250,000	Female	78	>65	June	16	Some college

We see that the original missing data for *gender* now has a label of “Unknown.” Also, if you scroll through the file, you will find no missing data for *age* or *income*.

We are now ready to use the Auto Data Prep node for more automated data preparation.

6.4 *Auto Data Prep Node*

The Auto Data Prep node (hereafter ADP) can take several actions with a data set. These include some of the same actions that can be taken in the Data Audit node, although with different settings. These actions include:

- The creation of a duration field for times or dates from a reference date or time
- The exclusion of fields with too many categories or with too many records in one category
- The exclusion of fields with too much missing data
- The reordering of nominal fields to place the smallest frequency category first, and the largest last

- The replacement of missing values with the mean, median, or mode, as determined by the measurement level
- The replacement of outliers in continuous fields
- The transformation of continuous fields to standardized scores (z-scores)

Additionally, ADP can transform existing fields or create new fields to:

- Maximize the association of a categorical input field to the target field by merging categories
- Maximize the association of a continuous field to the target by binning (grouping) the field
- Perform feature selection to identify fields, including transformed and new fields that have a statistically significant relationship to the target. This applies only to continuous input fields and a continuous target

Further Information

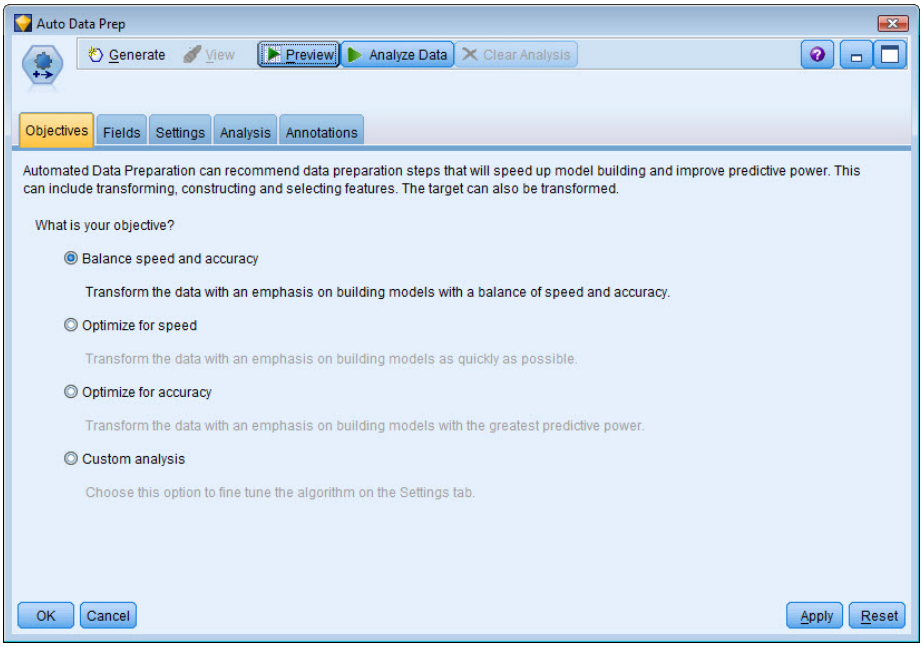
Feature selection to identify important input fields can also be done by the Feature Selection node, which we use in a later lesson for that task.

The Auto Data Prep node does not, by default, create new nodes to transform fields or to make substitutions for missing data. Instead, the node, which is located in the Fields palette, does its work internally, and data are passed through the node downstream.

We'll add an ADP node to the stream and review its features.

1) Add an **Auto Data Prep** node from the Fields palette to the stream to the right of the Type node
2) **Connect** the Type node to the ADP node
3) Edit the Auto Data Prep node

Figure 6.5 Auto Data Prep Node Objectives Tab



The Objectives tab provides four options for general data preparation: balancing speed and accuracy; optimizing for one or the other; or allowing the analyst to choose specific options.

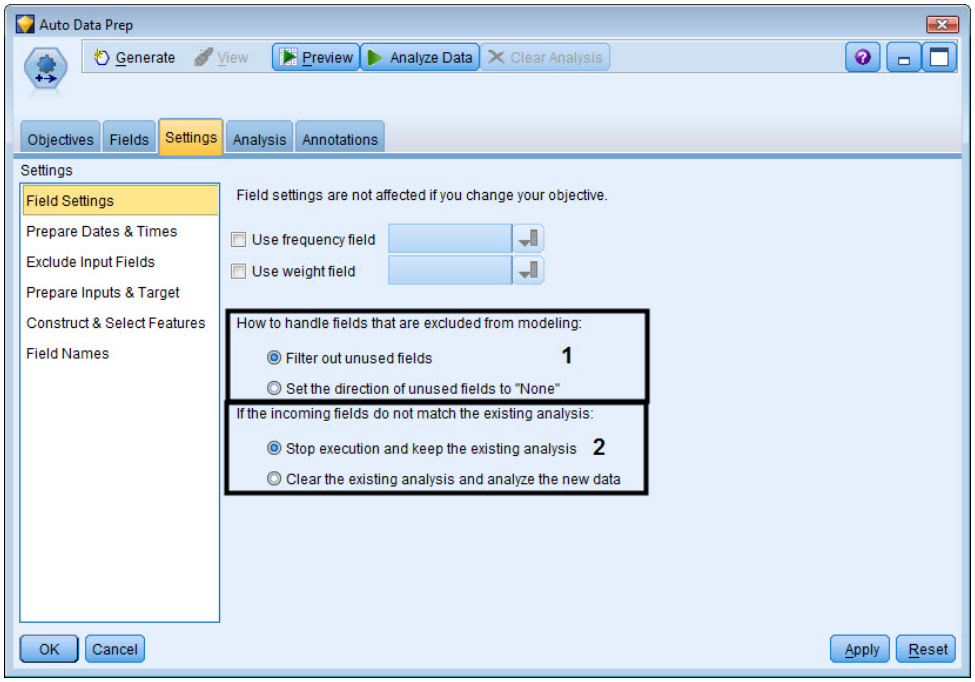
In theory, all that is required to run the node is to select one of these options. However, some changes are invariably made based on specific data characteristics.
To learn about the capabilities of the node, we'll choose a custom analysis.

4) Select **Custom analysis**

The fields used by the ADP node are, by default, taken from current Type node settings. Since we've set the inputs and targets already, we don't need to use the Fields tab.

5) Select the **Settings** tab

Figure 6.6 Auto Data Prep Node Settings Field Settings

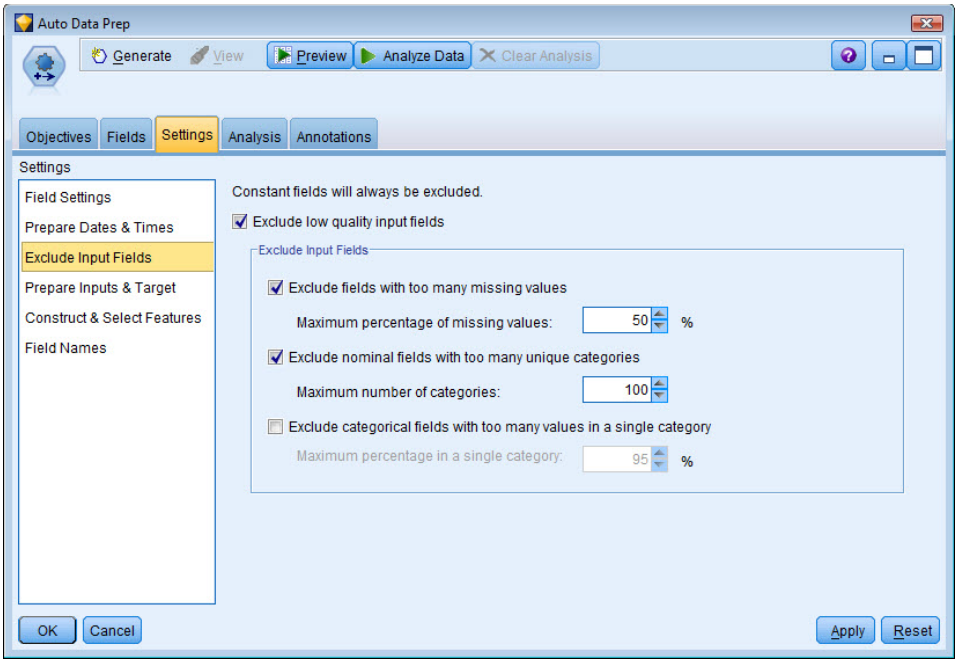


The first section in this tab is Field Settings

1. When the ADP node does data preparation, it can determine that some fields should not be used for modeling. Those fields can either be filtered, so they won't appear downstream, or set to a role (direction) of None so they won't be used in modeling. Unless data sets are very large, we generally recommend the second option, since it allows us to use these fields for reporting and other analyses later.
2. Eventually, a created model will be used to score (make predictions) on new data. The incoming fields for scoring must pass through the ADP node so that the data are prepared in a similar manner. Thus, the fields should match the original fields used to create the scoring model. If they don't, the ADP node will stop execution and not send the data downstream, by default. Or the existing analysis can be cleared and a new analysis done.

6) Select the **Exclude Input Fields** section

Figure 6.7 Auto Data Prep Node Exclude Input Fields Settings



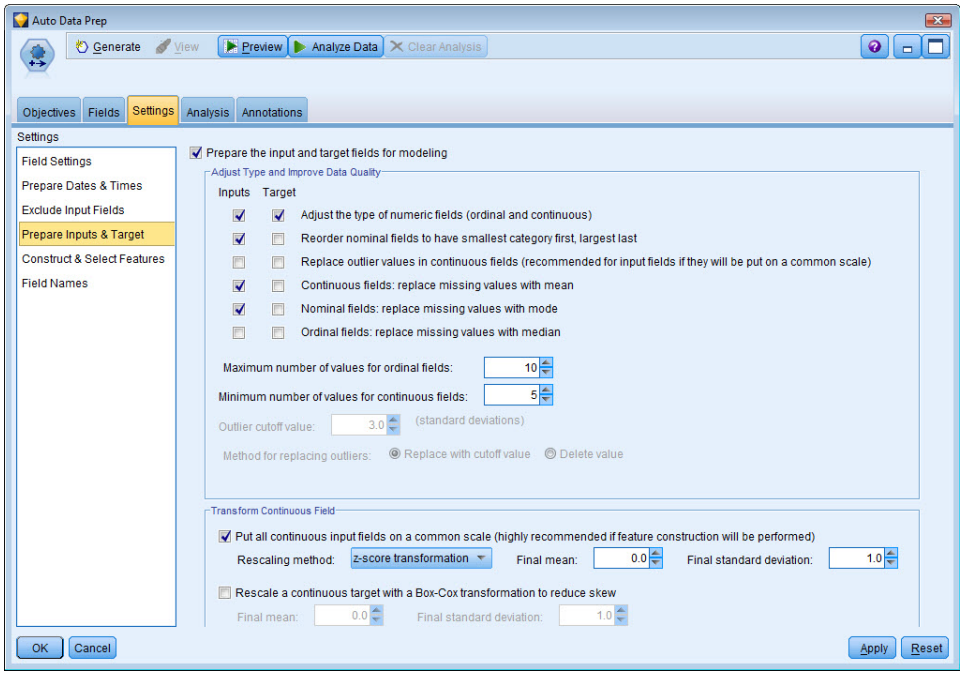
The user can set criteria in this section for excluding fields with too much missing data, nominal fields with too many unique categories, and any categorical fields with too concentrated a distribution in a single category.

Recommendations for using these settings include:

- 50% is often too lenient a setting for excluding fields with missing data. This might be reduced to 30%, or even lower, depending on other factors
- The number of unique categories to exclude a nominal field should be based on the type of models to be used, and on whether such a field is critical from a business perspective. Even fields with many categories can be important, although sometimes the categories can be combined beforehand (with the Reclassify node) into logical groupings
- The percentage of values necessary in a single category to exclude a field should probably be set lower than 95%, which again is quite lenient.

7) Select the **Prepare Inputs & Target** section

Figure 6.8 Auto Data Prep Node Prepare Inputs & Target Settings



Many actions can be taken based on the settings in this area. The default actions include:

- Adjusting the type of numeric fields, based on the settings below for the maximum number of values for ordinal fields and minimum number for continuous fields
- Reorder nominal input fields from lowest to highest frequency categories
- Replace missing values for continuous and nominal input fields
- Transform all continuous input fields to z-scores, with a mean of 0 and standard deviation of 1.

Other data preparation can be done, including replacing outlier values in continuous fields, based on the *Outlier cutoff value* and method listed in the lower half of the dialog. And all these actions can be taken for the target, as well as the inputs.

Best Practice

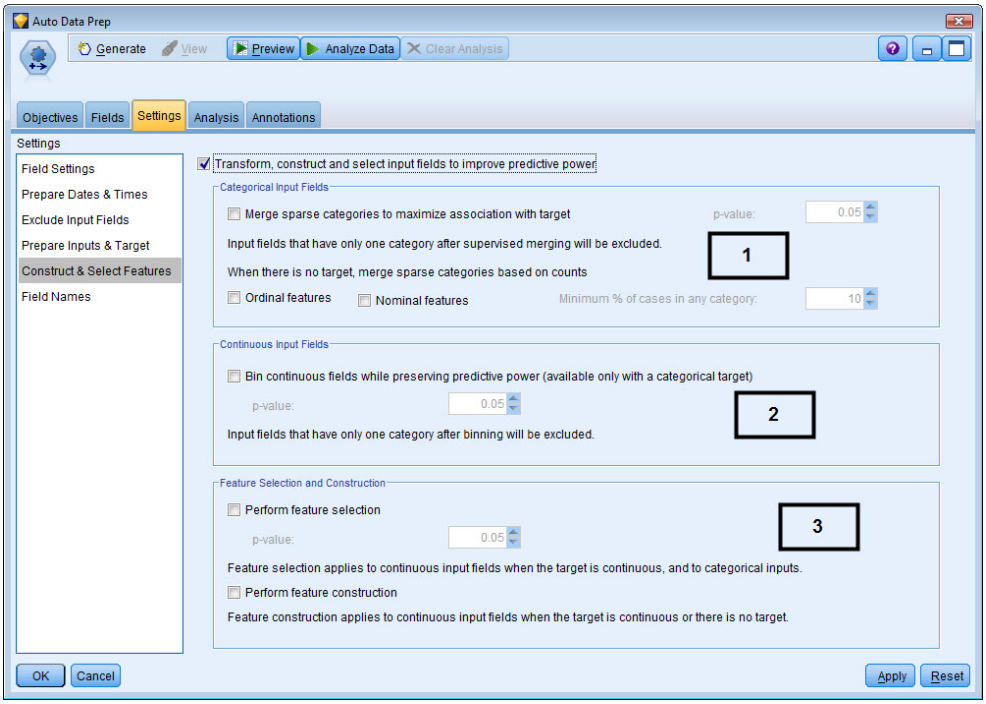
We normally do not replace missing values for target fields, or reorder their categories from low to high, although a continuous target might have outliers reduced in value because that can result in a more accurate model. However, doing so does change the data range to which the model applies.

The missing value options in the ADP node complement those in the Data Audit node. Here, random and model imputations are not available, but the common substitution methods of mean, median, or mode are, depending on the field measurement level.

One of the key advantages of the ADP node for data preparation is illustrated by the fact that these settings apply to all the fields with a certain role, or measurement level, at once. In the Data Audit node, to trim outliers or do mean substitution, the analyst has to specify these options for each field. The ADP node requires much less user intervention.

8) Select the **Construct & Select Features** section
9) Select the **Transform, construct and select input fields to improve predictive power** check box

Figure 6.9 Auto Data Prep Node Construct & Select Features Settings



There are three areas within this section.

1) The Categorical Input Fields area allows the user to merge sparse categories to increase association with a target field
2) The Continuous Input Fields area allows the user to group (bin) continuous fields to preserve predictive power. Doing so can make a model easier to interpret.
3) The Feature Selection and Construction area allows the user to identify input fields that are statistically associated with the target. This is done for all categorical inputs; it is only done for continuous inputs with a continuous target.

In this same area, the *Perform feature construction* check box requests the ADP node to combine input fields to create new fields when the inputs are highly correlated.

Note on Feature Construction and Preparation

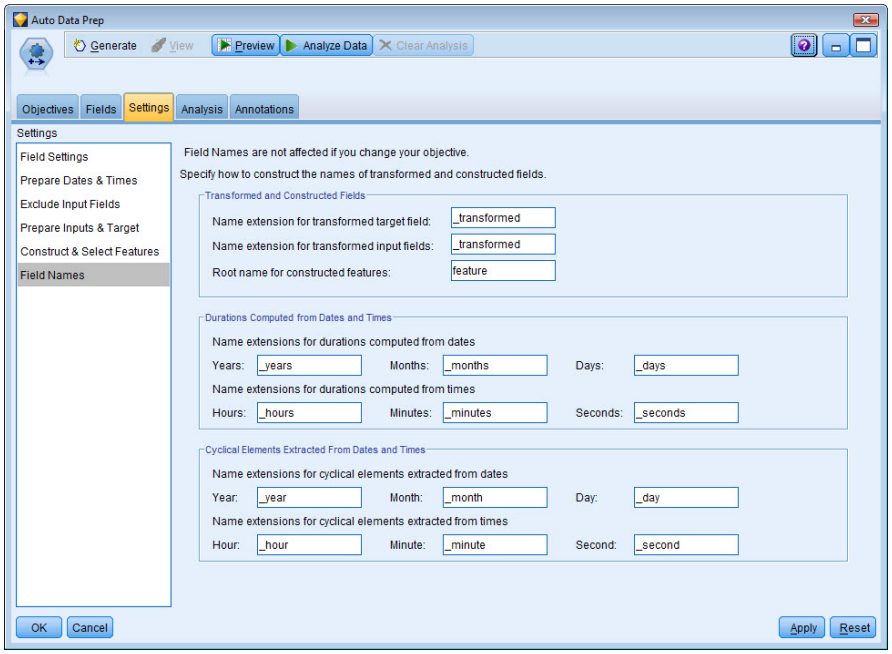
The ADP node does not honor a Partition field (see the next lesson) which creates training and testing data subsets. Because of this, all the data coming into the ADP node is used for all actions, including those taken in the Construct & Select Features section. This means that, for example, all the data would be used to merge sparse categories to maximize association with the target. However, it is a principle of data mining that the model testing data should be left pristine and not used in any way to create models.

Because of this, we only recommend using the first two features in this section when the data have been split physically into training and testing files, either at the original source or within the Modeler stream. Then all the data preparation can be done on the training data.

After a model is developed, the testing data can be run through the ADP node to prepare it in the same way. The specified transformations are applied to the data without further analysis as long as the upstream data does not change its characteristics, such as measurement level or role.

10) Select the **Field Names** section

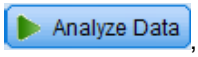
Figure 6.10 Auto Data Prep Node Field Name Settings



To easily identify new and transformed features, the ADP node creates new names for fields with an added suffix (extension). The user can change extensions in the text boxes. To specify completely different names, use a Filter node downstream.

Creating new fields with this method preserves the original fields, whose role will be set to None, in case the user wishes to use them downstream.

ADP Node Execution

The Auto Data Prep node is a process node, making data modifications. It does not create output, like a terminal node, or a model. Because it is important that the user review the results of the data preparation done by the node, the Auto Data Prep node provides an Analysis tab to view the actions taken by the node, after selecting the Analyze Data button at the top of the dialog 

, to perform the data preparation requested.

We will cancel our actions in the Auto Data Prep node, then reopen it to make appropriate selections for the telecommunications data file.

11) Select **Cancel**

6.5 *Operation: Using the Auto Data Prep Node*

We follow these steps to use the Auto Data Prep node with the data coming from the Type node.

1) Edit the Auto Data Prep node
2) Select **Custom analysis**

3) Select the **Settings** tab
4) Select **Set the direction of unused fields to "None"**
5) Select the **Exclude Input Fields** section
6) Change the Maximum percentage of missing values to **30**
7) Select the **Exclude categorical fields with too many values in a single category** check box
8) Change the Maximum percentage in a single category to **85**
9) Select the **Prepare Inputs & Target** section
10) Deselect **all checkboxes** in the Adjust Type and Improve Data Quality area except for **Continuous fields: replace missing values with mean** for the Input
11) Select the **Inputs** check box for **Ordinal fields: replace missing value with median**
12) Select the **Inputs** check box for **Replace outlier values in continuous fields**
13) Change the Outlier cutoff value to **5.0**
14) Deselect the **Put all continuous input fields on a common scale** check box

We will not construct or perform feature selection since we are using all the data.

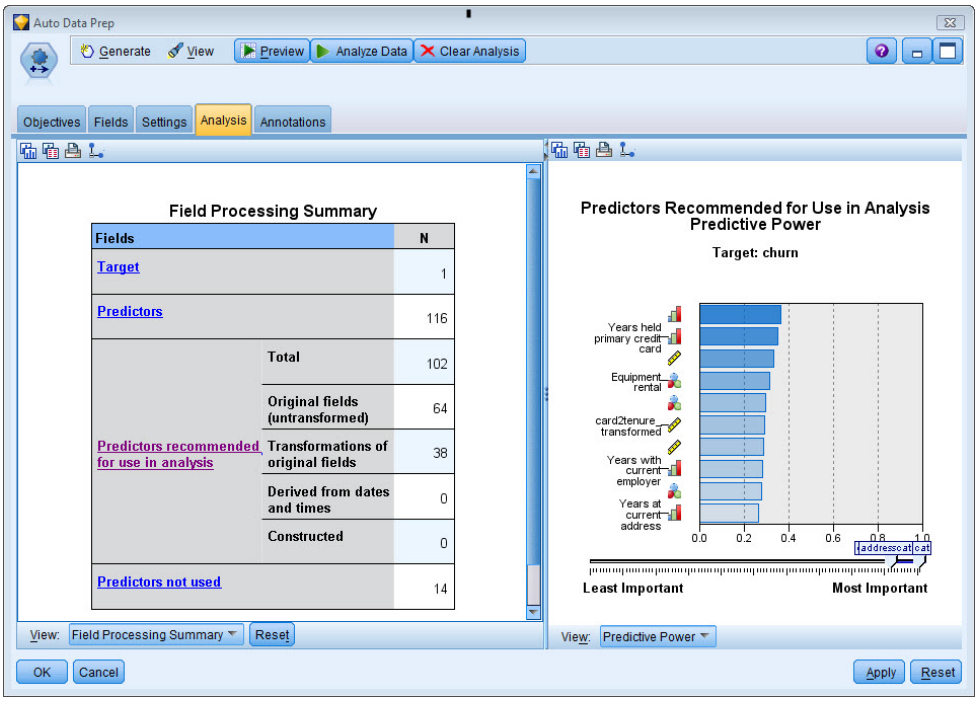
15) Select **Analyze Data** button

Auto Data Prep Node Analysis

After Modeler has finished executing, the Clear Analysis button is activated to delete the current analysis. To view the analysis:

1) Select the **Analysis** tab

Figure 6.11 Field Processing Summary of ADP Analysis



The Analysis tab contains both tabular and graphical output that summarizes the processing of the data and displays recommendations as to how the data may be modified or improved for scoring. The user can then review and either accept or reject those recommendations.

The Analysis tab is made up of two panels, the main view on the left and the linked, or auxiliary, view on the right. There are three main views: Field Processing Summary, Fields, and Action Summary. There are four linked/auxiliary views: Predictive Power, Fields Table, Field Details, and Action Details.

Within the main view, underlined text in the tables controls the display in the linked view. Clicking on the text allows the user to get details on a particular field, set of fields, or processing step.

With regard to the input fields, the ADP node:

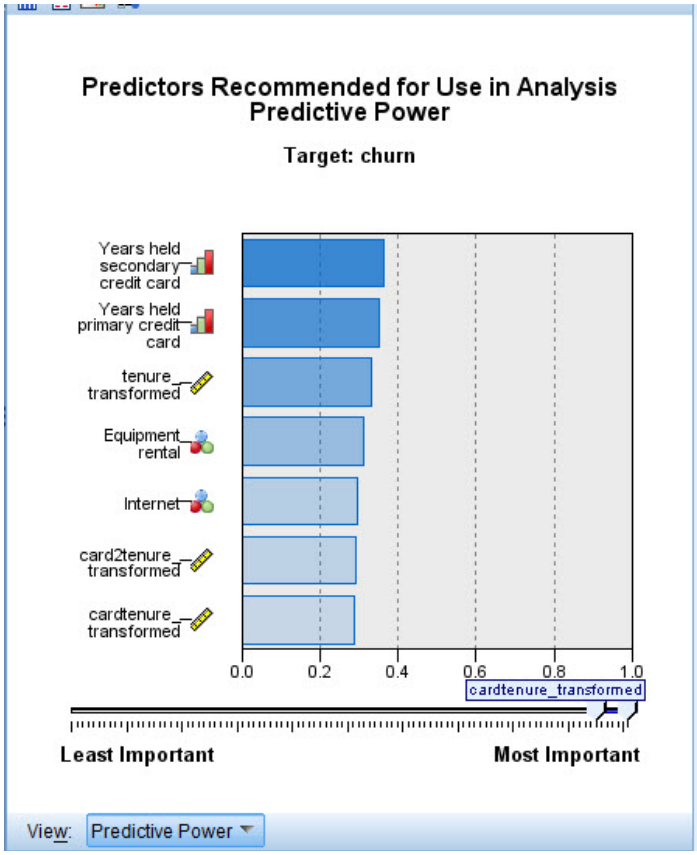
- Retained 64 untransformed fields
- Transformed 38 fields
- Marked 14 inputs (predictors) as not to be used

On the right in the current view, only the most important predictors are displayed.

1) Move the **left Importance** slider **toward the right** so only about 7 fields are displayed in the Predictive Power chart.

We see that the best predictors (in a bivariate relationship) with *churn* are either financial-related fields (years held primary and secondary credit cards), or fields measuring use of the telecommunication services (equipment rental, number of months as a customer, or internet service). They are not demographic fields, such as age or gender.

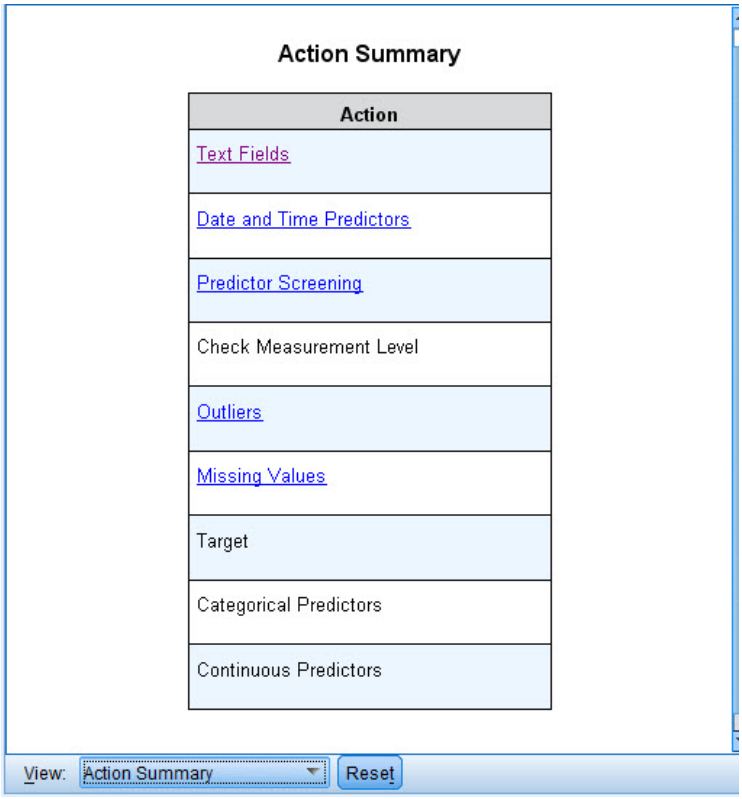
Figure 6.12 Top Predictors for churn



In automated data mining, there is no need to review every field and the action taken by the ADP node. We can, though, review a few details to learn about how the analysis output is organized. Let's look next at the Action summary.

1) In the Main view on the left, select **Action Summary** from the View dropdown

Figure 6.13 Action Summary View



Any underlined text can be selected to display that class of action.

2) Select **Predictor Screening**

We observe that 12 predictors were screened as not useful because they had too many records in a single category (above the 85% level we specified).

Figure 6.14 Predictor Screening Details

Predictor Screening

Predictors Excluded	N
Constant	0
Too many missing values	0
Too many records in a single category (> 85.0% of records)	12
Nominal fields with too many categories	0
Total	12

3) Select **Outliers** in the Main view

In the Outliers action, we see that there are many fields with large outliers (almost all positive in these data). For these fields, outliers were trimmed per our specification. When this occurred, two of the fields became constant and so were excluded.

Figure 6.15 Outlier Handling Details

Outliers

Outlier Handling	N
Continuous fields for which outliers were found and trimmed	38
Continuous fields excluded because they were constant after outlier handling	2

Outlier cutoff: 5.00 standard deviations above or below field mean.

Finally, we can review the missing value handling.

4) Select **Missing Values** in the Main view

Five fields had missing values replaced. None were nominal in measurement, as we had deselected this option.

Figure 6.16 Missing Value Details

Missing Values

Fields With Missing Values Replaced		N
Predictors	Nominal	0
	Ordinal	2
	Continuous	3
Total		5

To see which fields are not used, we need to switch views back to Field Processing Summary.

1) In the Main view on the left, select **Field Processing Summary** from the View dropdown
2) Select **Predictors not used**
3) Click the **Measurement Level** column header to sort by measurement level so Nominal is listed first

The predictors that are nominal are not used because they had too many records in a single category. Most of these don't seem as if they would be useful to predict *churn* anyway, such as *owntv*, *ownvcr*, or *commutebike*.

Figure 6.17 Predictors Not Used Table

Predictors Not Used

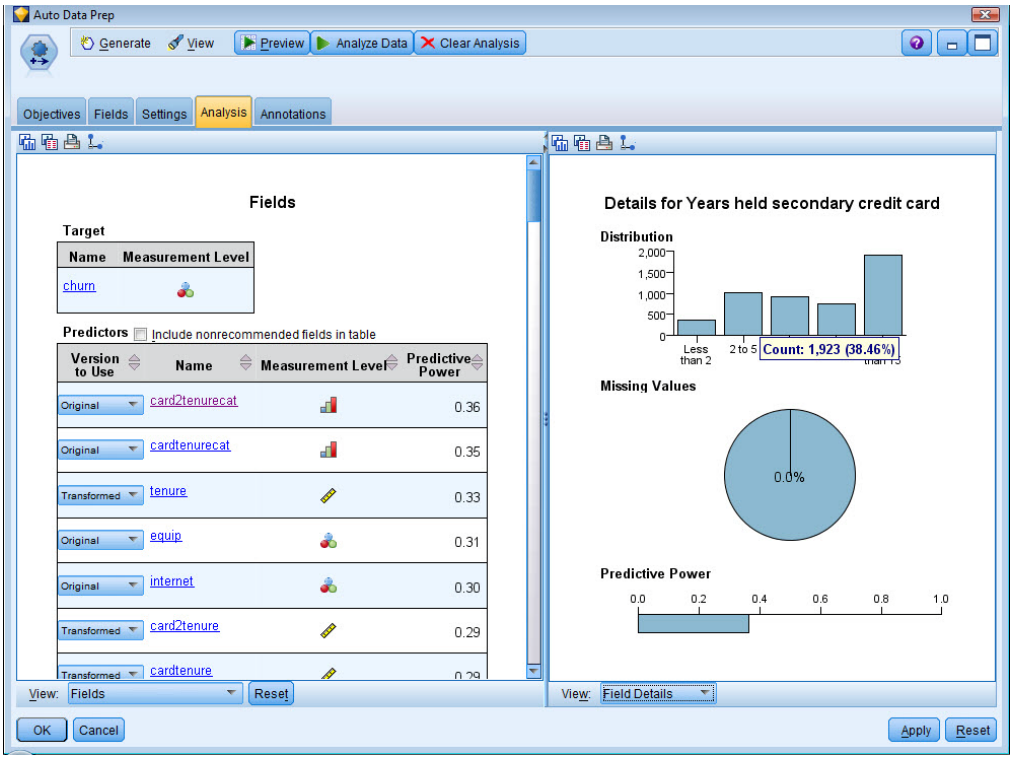
Name	Measurement Level
owntv	Nominal
ownvcr	Nominal
commutemotorcycle	Nominal
commutepublic	Nominal
commutebike	Nominal
commutenonmotor	Nominal
owndvd	Nominal
retire	Nominal
response_01	Nominal

Finally, we can look briefly at the Fields view.

1) In the Main view on the left, select **Fields** from the View dropdown

The Fields main view displays the fields and whether ADP recommends using them in downstream models. The user can override the recommendation for any field from the dropdown list for each field. If a field has been transformed, we can decide whether to accept the suggested transformation or use the original version. The user can sort these columns to easily locate fields.

Figure 6.18 Fields View in ADP Analysis Tab

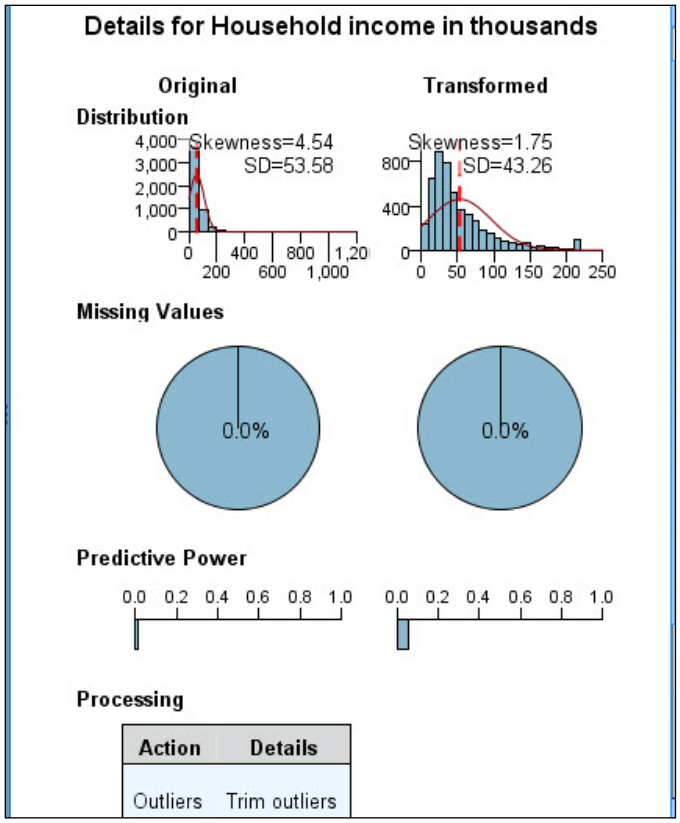


The Field Details view on the right contains distribution, missing values, and predictive power charts (if recommended to be used) for the selected field. In addition, the processing history for the field and the name of the transformed field are also shown (if applicable). For each chart, if a field was transformed, two versions are shown side by side to compare the field with and without transformations applied.

To see where outlier trimming made a difference, we can view the *income* field.

2) Scroll down and select the field **income**

Figure 6.19 Details for income Field



Trimming outliers reduced the skewness and standard deviation quite a bit, and the predictive power increased. This is the intention of the actions taken by the ADP node.

Further Information

Even if outliers are not trimmed from a field, the ADP node will still list a field as "transformed" since it was reviewed for the existence of outliers. There is no harm in using the transformed field, since it will be identical to the original field.

We have seen no evidence that the recommendations of the ADP node should be modified, so we will accept all of them. In the next lesson we continue with additional data preparation.

1) Select **OK** to close the ADP node

The ADP node retains the analysis when it is closed.

We'll save the stream for future work.

1) Select **File…Save Stream As**
2) Name the stream **Customer_Offers_Data Audit & ADP.str**
3) Select **Save**

Apply Your Knowledge

1) In which palette is the Type node located?
    a. Record Ops
    b. Field Ops
    c. Modeling
    d. Output

2) What options does the ADP node provide for handling fields excluded from modeling? Select all that apply.
    a. Filter out the unused fields
    b. Retain the field but mark it as “excluded”
    c. Set the Role of the field to None
    d. Rename the field to identify it as not excluded

3) Which of these is not a method of data preparation supported by the ADP node?
    a. Missing value substitution
    b. Outlier handling
    c. Excluding fields with too many categories
    d. Excluding fields with too few categories

6.6 *Lesson Summary*

In this lesson we demonstrated how to use the Auto Data Prep node to automatically prepare data to be used for modeling.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Use the Auto Data Prep node to further prepare data for modeling

To support the achievement of the primary objective, students should now also be able to:

- Use the Type node to set characteristics for fields
- Describe the various features and capabilities of the Automated Data Prep node
- Use settings of the Automated Data Prep node that are appropriate for the data and modeling objectives
- Describe the types of output produced by the Automated Data Prep node

6.7 Learning Activity

The overall goal of this learning activity is to practice using the ADP node to prepare data for modeling.

Supporting Materials

The Modeler stream file *Lesson 5 Exercise.str*. If this file was not created, you can use *Backup_Lesson 5 Exercise.str*.

1. Open the stream file *Lesson 5 Exercise.str*.

2. Add an ADP node to the stream. Connect the Source node to the ADP node.

3. Edit the ADP node. Request a custom analysis. Also request that

    a. The direction of unused fields be set to None
    b. Exclude categorical fields with more than 80% in a single category
    c. Turn off all check boxes in the Adjust Type and Improve Data Quality area, except for replacing outlier values in continuous fields for the Inputs
    d. Change the outlier cutoff value to 4.0
    e. Turn off the option to put all continuous fields on the same scale

4. Run the Analysis in the ADP node. How many predictors are not used? Why? Which continuous fields were transformed? Which one had the greatest proportional decrease in standard deviation?

5. Save the stream file as *Lesson 6 Exercise.str*.

Lesson 7: Data Partitioning

7.1 *Objectives*

After completing this lesson students will be able to:

- Use a Partition node to create training and testing data subsets

To support the achievement of this primary objective, students will also be able to:

- Describe rationale and use of a Partition node to create data subsets
- Set sizes of the training and testing partitions and other partition characteristics
- Use a Distribution node to view the distribution of a categorical field

7.2 *Introduction*

Models that are built (train) must be assessed with a separate testing data file that was not used to create the model. The training and testing data should be created randomly from the original data file. There are many methods that could be used to create the two datasets, but to automate the process, Modeler includes the Partition node, which provides great flexibility in creating the data files.

With the Partition node, Modeler has the capability to directly create a field that can split records between training, testing (and validation) data files. Partition nodes generate a partition field that splits the data into separate subsets or samples for the training and testing stages of model building.

In this lesson, we will continue with the data that have been prepared in earlier lessons with the Data Audit and Automated Data Prep nodes, adding a Partition node to the stream.

	The *customer_offers.sav* Statistics data file. These data are from former and current customers of a telecommunications company. The data includes both demographic and account-related information.
Supporting Materials	The Modeler stream file *Customer_Offers_Data Audit & ADP.str*, which contains the results of previous data preparation.

7.3 *Data to Train and Test Models*

Up until this point in the course, we have had one data stream, and all the data from the *customer_offers.sav* data file have been used in data preparation. At some point, though, the data need to be split into separate subsets, one for building models, and at least one more to validate, or test, the models.

There are generally two strategies when creating training and testing data files.

- Construct two completely separate data files. Use one to prepare the data and create models. Then add the second to the stream to test the models.
- Use a Partition node to split (partition) a single data file into subsets. Use the ability of Modeler to create models on the training partition and validate the model on the testing partition.

For automating the data mining process, the second option and use of a Partition node is definitely the better choice.

One decision to make is exactly when to create these separate streams, or partitions. It can be done early in the data preparation process, but eventually, the data preparation done on the training data must also be done on the testing data. So there is some logic in not partitioning the data immediately. Care must be taken not to use information about the target field to modify the predictor or input fields in the testing data.

7.4 *The Partition Node*

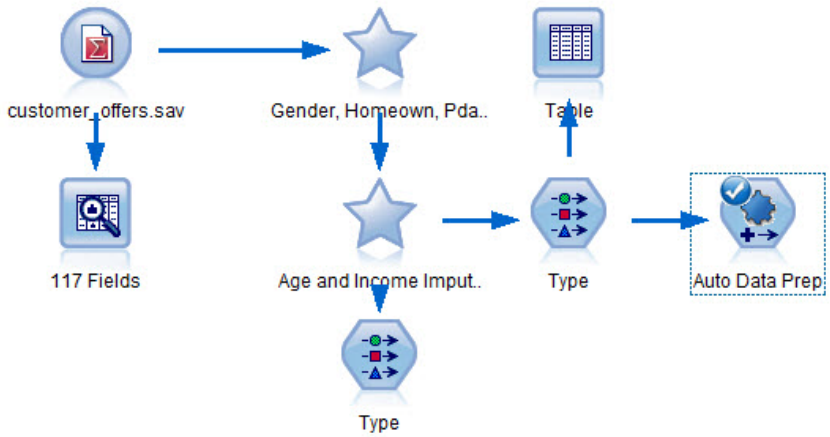
The Partition Node, found in the Field Ops palette, generates a partition field that divides the data into separate subsamples for the Training, Testing (and Validation) stages of model building. The partition field will have its role set to Partition. By default, the node randomly selects 50% of the cases for training purposes and reserves the other 50% for testing the model. These proportions can be altered.

If one prefers, they can subdivide the cases into three samples instead of just two, one for Training, one for Testing, and one for Validation. The model will still be built with the Training sample and tested with the Testing sample. However, the Testing sample can then be used to help further refine the model, and once satisfied that the model is the best you can get, the Validation sample is then used to see how well the model performs against yet unseen data. In this lesson, we will use just training and testing partitions.

We need to open the stream file saved in Lesson 6.

1) Open the stream file **Customer_Offers_Data Audit & ADP.str**

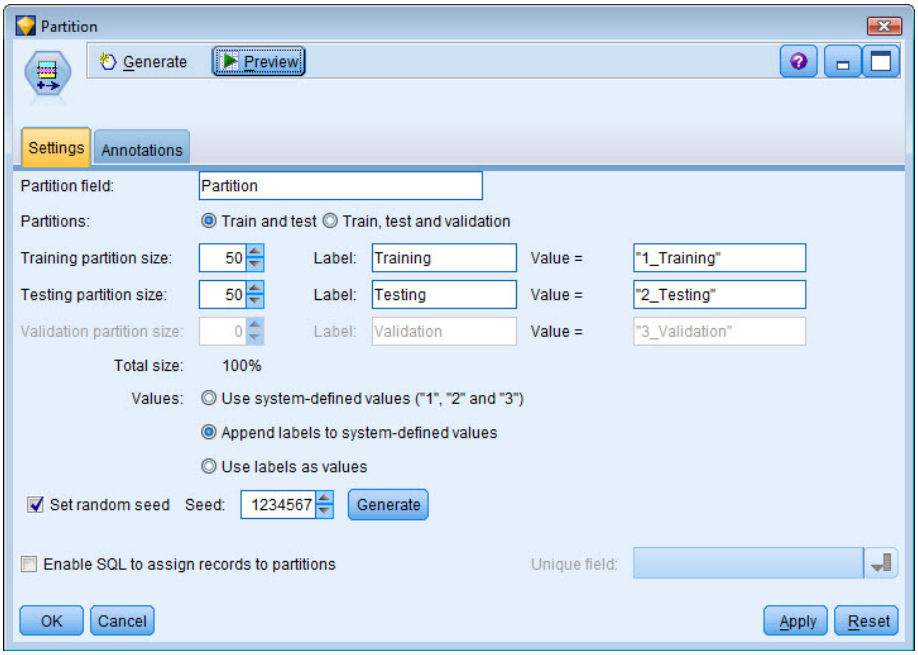
Figure 7.1 Customer_Offers_Data Audit & ADP.str Stream File



Now we can add a Partition node to the stream.

2) Add a **Partition** node from the Fields Ops palette to the right of the ADP node
3) **Connect** the ADP node to the Partition node
4) Edit the Partition node

Figure 7.2 Partition Node Settings Tab



To use the Partition node, all that is required is to decide on two key features:

1. Partitions: Either use the default of *Train and test*, or select *Train, test and validation*.
2. Partition size: Set the training partition size and the testing partition sizes (and if being used, the validation partition size) to the desired proportions.

Normally, more data should be used to develop the model than is used to test the model. With moderate sized datasets (a few tens of thousands of records), a typical split is 70% for training and 30% for testing. The total size of the partitions should add to 100%, and Modeler will provide a warning message if they do not.

Modeler will label the partitions with the labels, and values, shown in the text boxes, but these can be changed as preferred.

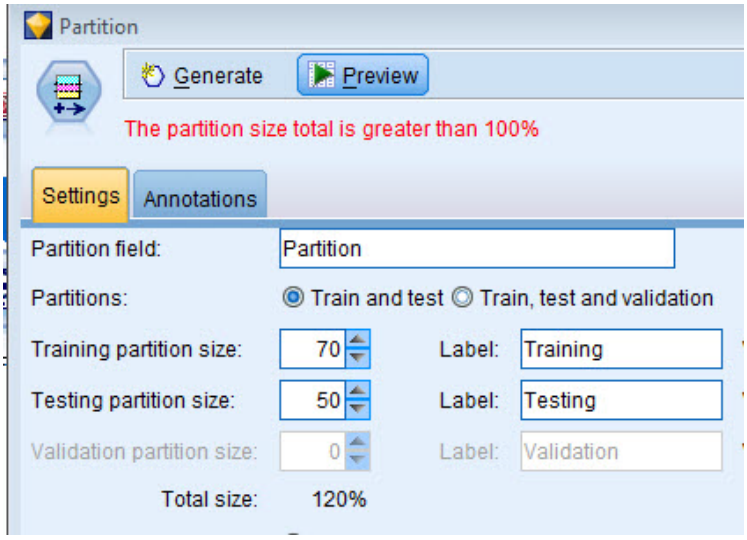
To create the partitions, Modeler uses a random number seed to randomly sample from the data stream. If a different random number seed is used for each execution, then a different random sample would be used each time. This would mean that the records assigned to the training and testing partitions would be different each time data are passed through the Partition node, and this violates the requirement of complete separation of the training and testing data.

To avoid this, Modeler uses a default fixed random seed (*1234567*) to generate random numbers. It is perfectly acceptable to use this seed, and we will do so in this lesson, although the user can specify a custom seed, or use the Generate button to generate a new random seed.

We will make only one change, making the proportion of cases in the training set 70% and the testing set 30%.

1) Set the Training partition size to **70**

Figure 7.3 Modeler Message about Partition Size Total Greater than 100%



When we do so, a message in red appears that currently the partition size is greater than 100%. No such message will appear for sizes totaling less than 100%.

Further Information

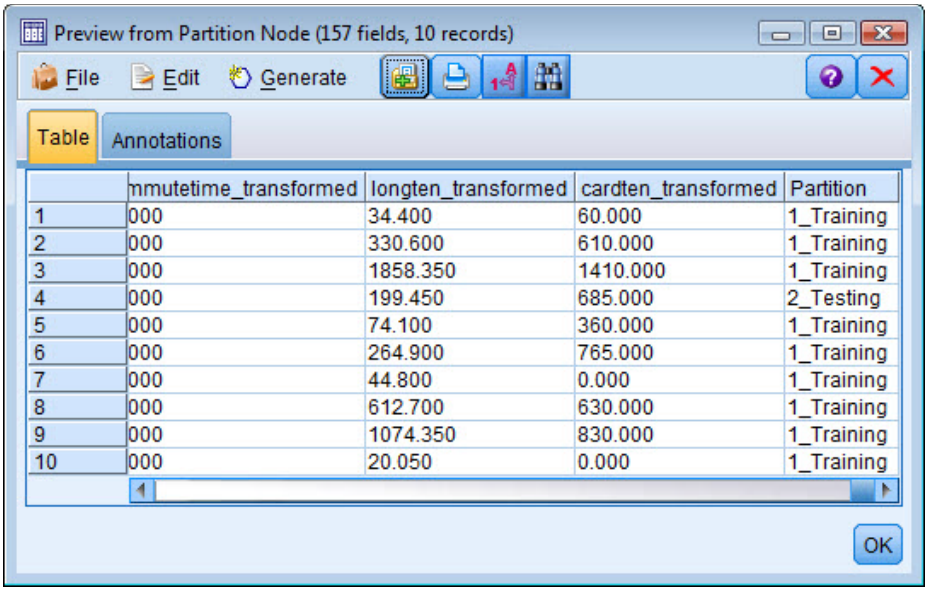
If the sum of the partition sizes is less than 100%, then the records not included in a partition will be discarded. This is another method to sample from a large data file.

2) Set the Testing partition size to **30**

To see the results of this, we can use the data preview.

3) Select the **Preview** button
4) Scroll to the last column in the Table window

Figure 7.4 Partition Field in Data Stream



	nmutetime_transformed	longten_transformed	cardten_transformed	Partition
1	000	34.400	60.000	1_Training
2	000	330.600	610.000	1_Training
3	000	1858.350	1410.000	1_Training
4	000	199.450	685.000	2_Testing
5	000	74.100	360.000	1_Training
6	000	264.900	765.000	1_Training
7	000	44.800	0.000	1_Training
8	000	612.700	630.000	1_Training
9	000	1074.350	830.000	1_Training
10	000	20.050	0.000	1_Training

The new field *Partition* has been added to the data stream. Because of random assignment, 90% of the first 10 records have been assigned to the training partition.

Modeling nodes, and others, are set to recognize the existence of a partition field in the data.

1) Select **OK** to close the Table window
2) Select **OK** to close the Partition node

We can view the complete distribution of the *Partition* field with a Distribution node.

Distribution Node

A Distribution node is used to review the distributions of categorical fields. It displays the equivalent of a frequency table and a bar chart. The Distribution node is located in the Graphs palette.

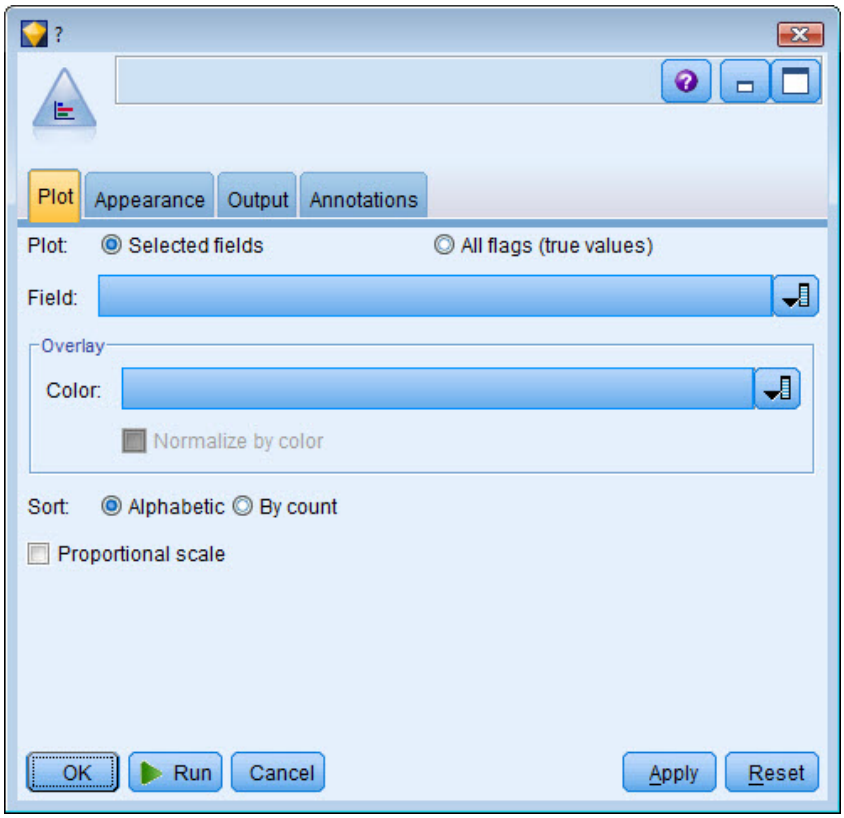
The Data Audit node produced a distribution chart for all the categorical fields in the data file. The Distribution node produces the equivalent for one field at a time. A second categorical field can be overlaid to examine the distribution between the two fields.

We'll add a Distribution node to the stream.

1) Place a **Distribution** node on the stream near the Partition node
2) **Connect** the Partition node to the Distribution node
3) Edit the Distribution node

The only required user action is to select a field in the Field area. The Field Chooser button  is used to open a list of fields in the data stream.

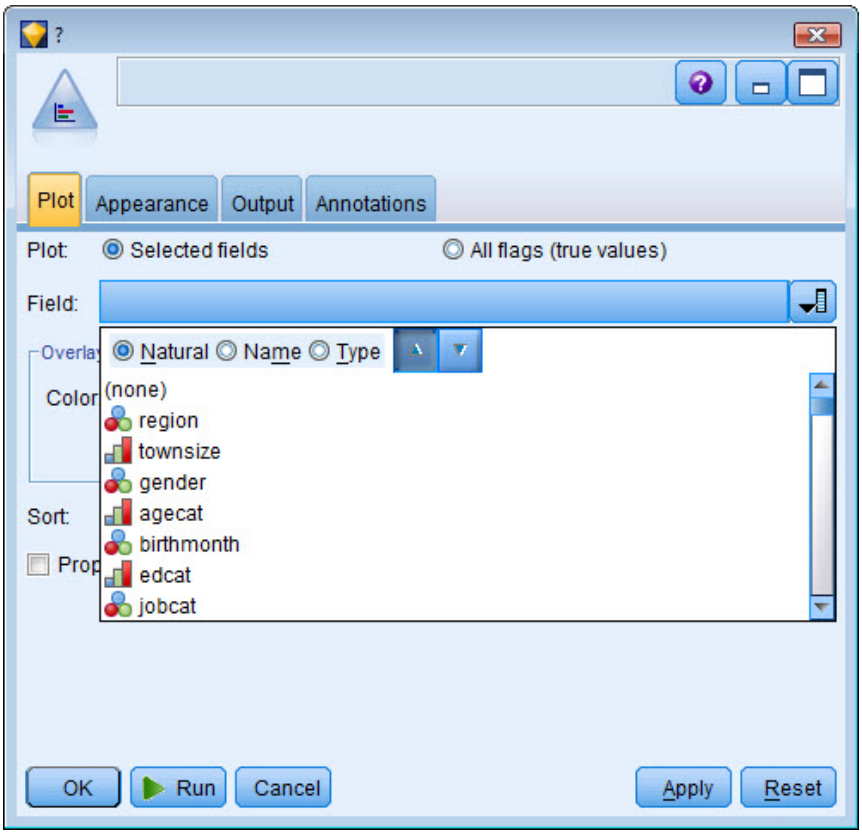
Figure 7.5 Distribution Node



1) Select the Field Chooser button

The fields are listed in their current order in the data stream (Natural order). To more easily locate a field, the list can be sorted in alphabetical order (Name order), or by measurement level.

Figure 7.6 Field Chooser Listing

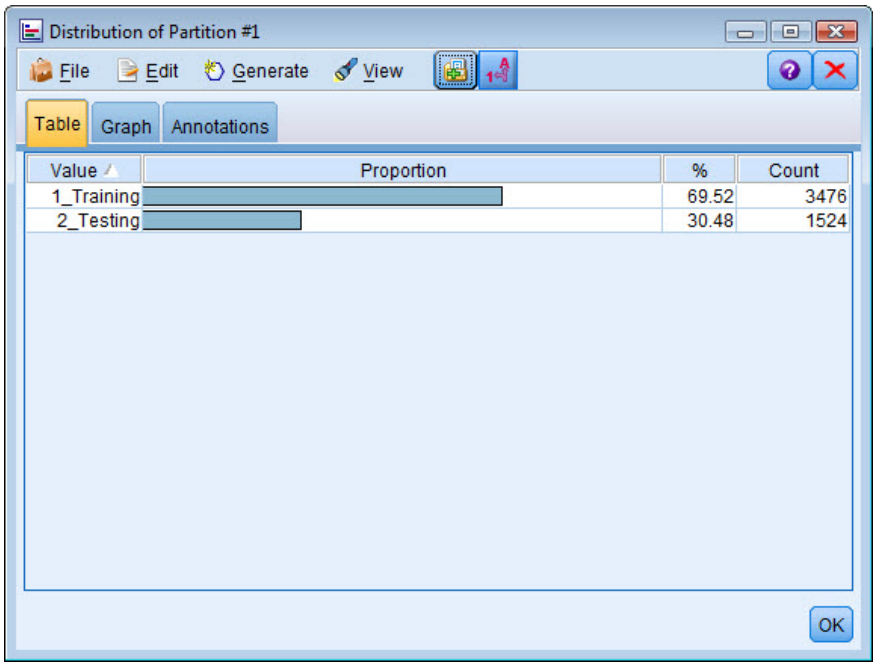


2) Select **Partition**

By default, categories of the selected field will be sorted by alphabetical order in the table and chart. The user can instead sort them by count (frequency). The *Proportional scale* check box makes the largest bar span the full width of the chart. It is normally more useful when there is an overlay field.

3) Select **Run**

Figure 7.7 Distribution Table for Partition Field



We requested that the training partition contain 70% of the records and the testing partition 30%. Of course, these assignments were made randomly, so the percentages don't quite match this distribution, but they are certainly close enough for all purposes.

We can save the stream at this point, with the added Partition node.

1) Select **OK**
2) Select **File…Save Stream As**
3) Name the stream **Customer_Offers_Partition.str**
4) Select the **Save** button

Apply Your Knowledge

1) True or False? The Testing data subset should be larger than the Training data subset?

2) In what palette is the Partition node located?
   a. Record Ops
   b. Output
   c. Source
   d. Field Ops

7.5 *Lesson Summary*

In this lesson we demonstrated how to create training and testing partitions.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Use a Partition node to create training and testing data subsets

To support the achievement of the primary objective, students should now also be able to:

- Describe rationale and use of a Partition node to create data subsets
- Set sizes of the training and testing partitions and other partition characteristics
- Use a Distribution node to view the distribution of a categorical field

7.6 Learning Activity

The overall goal of this learning activity is to use a Partition node to split the charity data for modeling.

Supporting Materials

The Modeler stream file *Lesson 6 Exercise.str*. If this file was not created, you can use *Backup_Lesson 6 Exercise.str*.

1. Open the stream file *Lesson 6 Exercise.str*.
2. Add a Partition node to the stream. Connect the ADP node to the Partition node.
3. Edit the Partition node. Set the size of the Training partition to 70 and the size of the Testing partition to 30.
4. Enter the random seed 444.
5. Add a Distribution node to the stream. Run a distribution analysis for the field *Partition*.
6. Save the stream file as *Lesson 7 Exercise.str*.

Lesson 8: Predictor Selection for Modeling

8.1 *Objectives*

After completing this lesson students will be able to:

- Use the Feature Selection node to select inputs for modeling

To support the achievement of this primary objective, students will also be able to:

- Describe the features and settings of the Feature Selection node
- Describe the model output from Feature Selection
- Generate a Filter node to use the selected fields

8.2 *Introduction*

Data mining problems may involve hundreds, or even thousands, of fields that can potentially be used as inputs. As a result, a great deal of time and effort may be spent examining which fields or variables to include in the model. Although some algorithms can use many inputs, some perform poorly if there are too many inputs, and all algorithms are slower, sometimes substantially so, as the number of inputs is increased.

To help us choose from a long list of fields, the Feature Selection node can identify the fields that are most important for a given analysis. By reducing the number of fields used in the model, the scoring times may also be reduced for many models, and time for future data preparation can be decreased if we can concentrate on only the most critical fields.

We will still need to use our judgment concerning which inputs to include, and those important for operational or other reasons should be included, no matter the results of the Feature Selection analysis.

	The *customer_offers.sav* Statistics data file. These data are from former and current customers of a telecommunications company. The data includes both demographic and account-related information.
Supporting Materials	The stream file *Customer_Offers_Partition.str*, which contains the results of previous data preparation.

8.3 *The Feature Selection Node*

The Feature Selection Node is found in the Modeling palette, perhaps surprisingly. This is because it creates a model nugget, although one that does not add predictions or other derived fields to the stream. Instead, it acts as a filter node, removing unnecessary fields downstream (with parameters under user control).

Feature selection has three steps:

Screening. In this first step, fields are removed that have too much missing data, too little variation, or too many categories. Also, records are removed with excessive missing data.

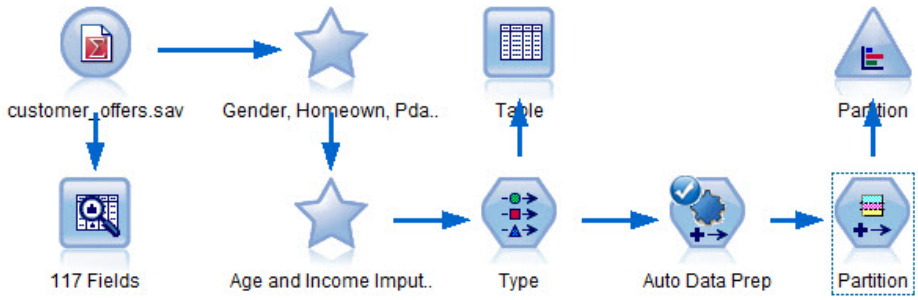
Ranking. In the second step, each predictor is paired with the target and an appropriate test of the bivariate relationship between the two is performed. This can be a chi-square test for categorical fields or a Pearson correlation coefficient if both fields are continuous. The probability values from these bivariate analyses are turned into an importance measure by subtracting the p value of the test from 1 (thus a low p value leads to an importance near 1). The predictors are then ranked on importance.

Selecting. In the final step, a subset of predictors is identified to use in modeling. The number of predictors can be identified automatically by the model, or the user can request a specific number.

We will continue to work with the stream file from the previous lesson, with the Partition node added.

1) Open the stream file **Customer_Offers_Partition.str**

Figure 8.1 Customer_Offers_Partition.str Stream File



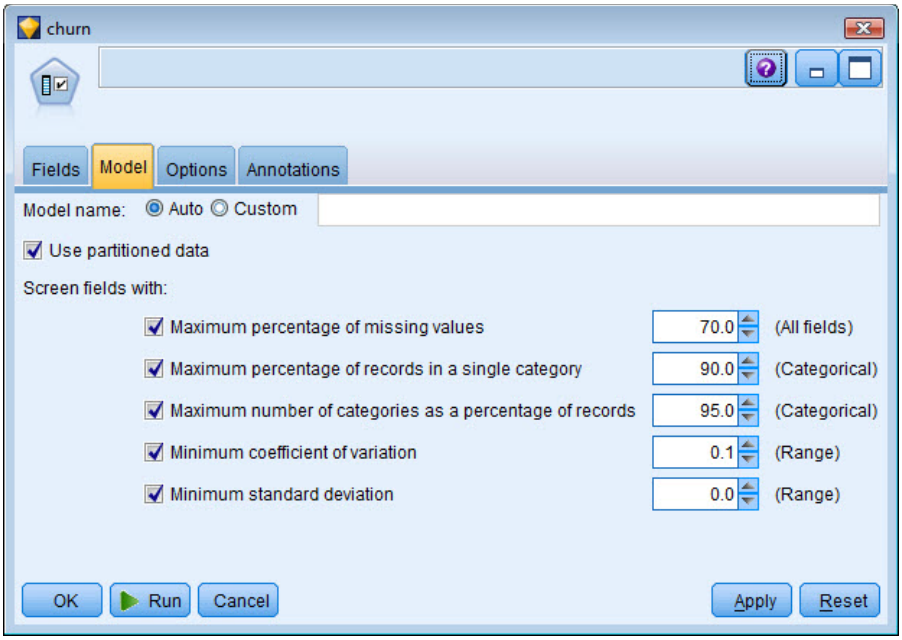
Now we can add a Feature Selection node to the stream.

1) Add a **Feature Selection** node from the Modeling palette to the right of the Partition node
2) **Connect** the Partition node to the Feature Selection node
3) Edit the Feature Selection node

The Feature Selection node immediately recognizes *churn* as the target field.

By default fields will initially be screened based on the various criteria listed in the Model tab. A field can have no more than 70% missing data. There can be no more than 90% of the records with the same value, and the minimum coefficient of variation (standard deviation/mean) is 0.1. All of these are fairly generous standards. Moreover, we have already been preparing the data in the stream and have screened for these same type of criteria. The Feature Selection node includes these just in case we need to do additional screening based on data quality as well as association with the target field.

Figure 8.2 Model Tab in Feature Selection Node

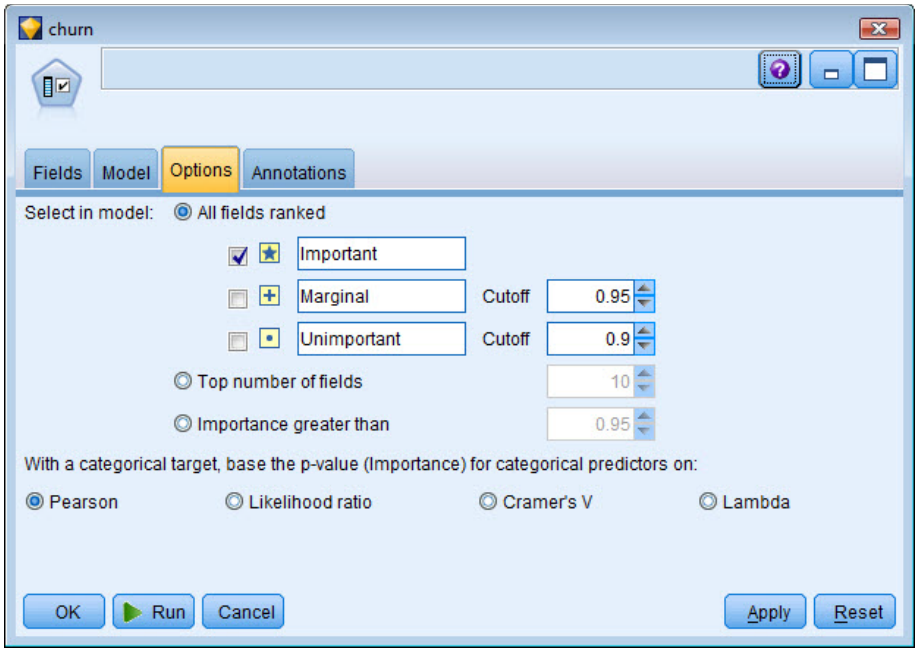


4) Select the **Options** tab

Fields are tested for their bivariate association with the target, with the test used depending on whether the predictors and the target are all categorical, all numeric ranges, or a mix of range and categorical. As stated above, the p value of the test is subtracted from 1.0 and used as a measure of importance of that input.

Four options are available for determining the importance of categorical predictors with a categorical target, with the default being the Pearson chi-square value. Normally there is no reason to change this default.

Figure 8.3 Options Tab in Feature Selection Node



Labels are applied to various categories of the importance values, determined by cutoff values, and the user can change these labels if they wish.

The inputs are ranked based on importance, and only those deemed Important (importance of .95 or above by default) will be selected in the model. The check in the box by the Important label indicates that only this group of inputs will be included.

The user can also change the cutoff values for the various categories. In very large data files (our current file of 5,000 records is not that large), we might increase the cutoff values, otherwise most fields can be labeled as Important.

Alternatively, the user can select fields based on:

- The top N fields, by ranking of importance
- All fields that meet a minimum level of importance

Further Information

The importance values measure whether an input is significantly associated with the target; they do not measure the strength of that association.

8.4 *Feature Selection Model*

The default settings are perfectly appropriate for our analysis, so we can simply execute the node.

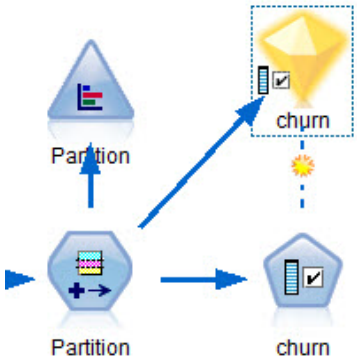
1) Select **Run**

When Feature Selection finishes execution, it creates a model nugget 

. In general, a model nugget contains the results of a model. It allows the user to both browse the results of a model—which come in a variety of formats—and apply the model to the data stream.

Model nuggets are both placed in the Models manager or palette, in the upper right corner of the screen, and on the Stream Canvas. And in the stream, a model nugget is automatically connected to the modeling node that created it (see figure below). Each link contains a symbol to indicate whether the model is replaced when the modeling node is executed. There is also a link to the node that is attached to the modeling node upstream (in this example, the Partition node). All these connections allow the model to be updated easily and applied to the stream data.

Figure 8.4 Feature Selection Model Added to Stream Canvas



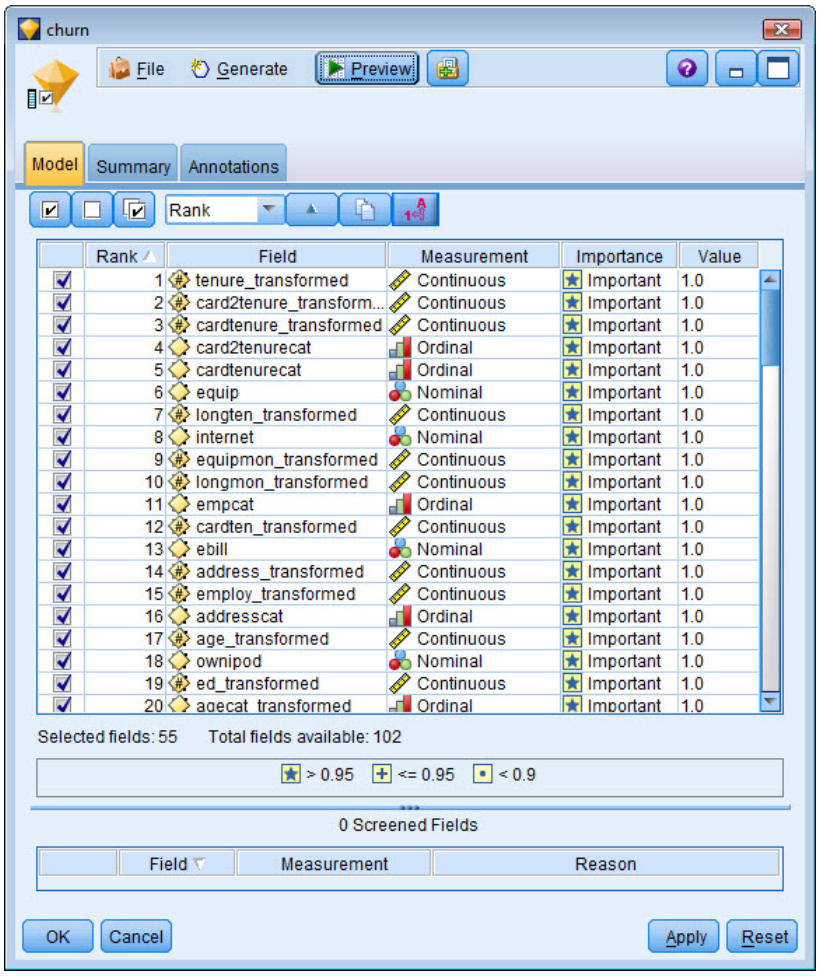
Let's review the Feature Selection results.

2) Right-click the **churn** Feature Selection model nugget in the stream and select **Edit**

The Model tab for a Feature Selection model nugget displays the rank and importance of all inputs in the upper pane and allows the user to select fields for filtering by using the check boxes in the column on the left. When the stream is run, only the checked fields are preserved. The other fields are discarded (filtered). The default selections are based on the options specified in the model-building node, but the user can select or deselect additional fields as needed.

Just as a reminder, the results are based only on the training partition data. This is important so that we don't use the testing data to decide which fields to include when building a model.

Figure 8.5 Feature Selection Model



There are 102 fields currently available as potential inputs. No fields were screened because of too much missing data or too little variation (we have taken care of this already). The model selected 55, or just over half of the fields, as being important. The model ranked the fields by importance (importance is rounded off to a maximum value of 1.00). If you scroll down the list of fields in the upper pane, you will eventually see fields with low values of importance that are unrelated to *churn*. All fields with their box checked will be passed downstream if this node is added to a data stream.

To make it easier to find fields, or review other characteristics of the inputs (e.g., measurement level, the table can be sorted by any of the columns, either from the dropdown list in the toolbar, or by clicking on a column header.

The set of important fields includes a mix of characteristics, but the most important are those measuring aspects of the customer's account or use of services from the firm.

There are two ways to filter fields based on the Feature selection model:

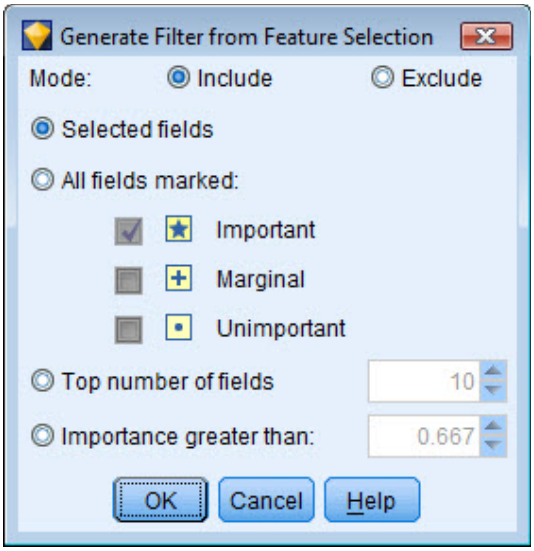
1) Use the model nugget, running data through it and downstream to our preferred modeling node. This is easy to do, but has the disadvantage of filtering out all the fields that we might want to use in other analyses.
2) Generate a Filter node, which we then have to connect in the stream. This has the advantage of allowing us to easily modify the filter settings, or rename fields as well. Although it is not

automatically updated when the Feature Selection node is rerun, normally we run that node only once, so this isn't a drawback.

In this example, we will use the second option.

1) Select **Generate…Filter** from the menu

Figure 8.6 Generate Filter Dialog

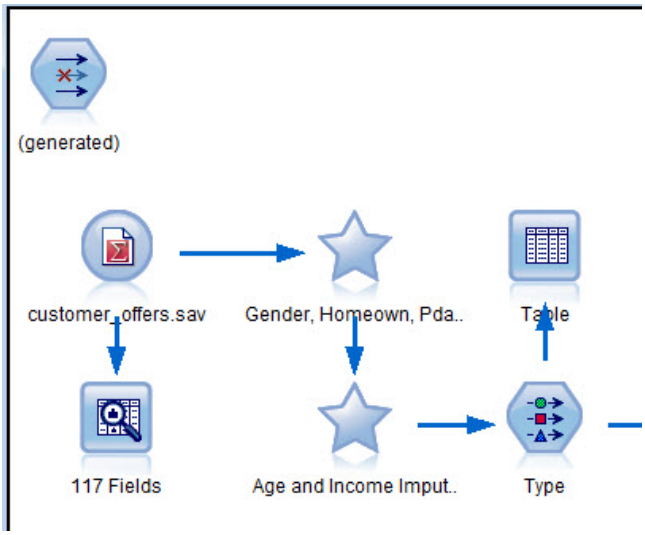


The Generate Filter from Feature Selection intermediate dialog box appears, allowing the user to make changes in which fields will be filtered. We'll use the current setting.

2) Select **OK**
3) Close the Feature Selection model browser

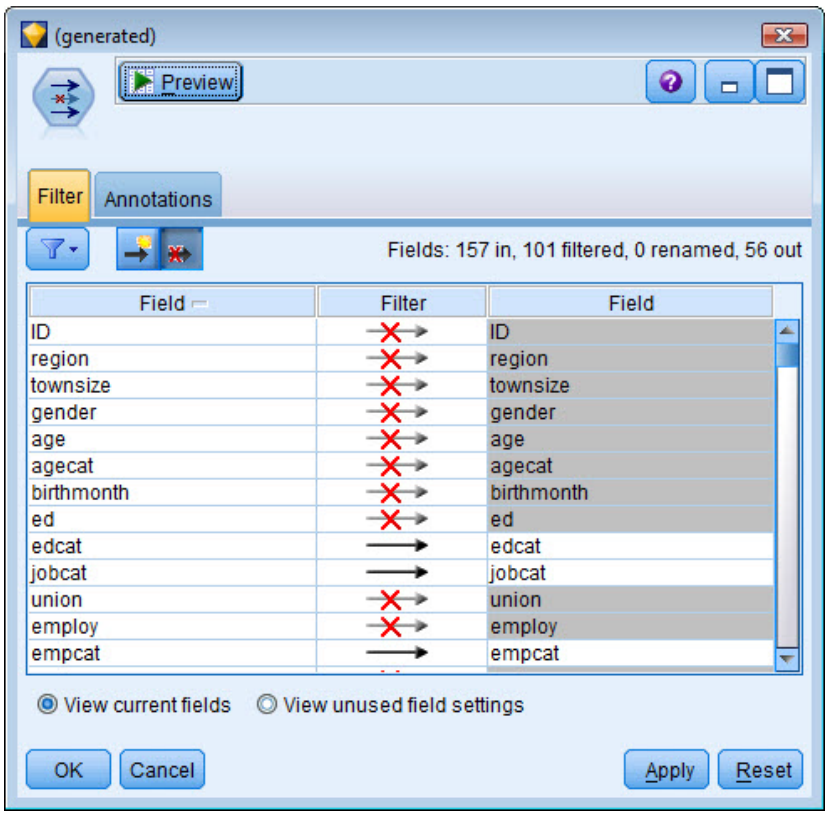
A Filter node is added to the Stream Canvas in the upper left corner. It is not labeled with "Filter" because it has been generated from another node. We will rename it after connecting it to the Partition node.

Figure 8.7 Generated Filter Node Added to Stream Canvas



1) Move the **(generated)** node near the Partition node
2) **Connect** the Partition node to the **(generated)** node
3) Edit the (generated) node

Figure 8.8 Filter Node Generated from Feature Selection Model



The node is a Filter node, with the fields selected by the model retained. There are a few changes we need to make. First, when this node is generated, the field *Partition* is filtered, but of course we need to retain it.

4) Click on the arrow for **Partition** to turn off filtering

Next, there are a few other fields that should not be used for modeling because they may not be available in future data, or because they don't apply to every customer. These include *spousedcat*, *reason*, *polview*, *active*, and *bfast*. We left these in while doing data preparation because it didn't affect the results of those tasks, but now that we are in the final stages of data preparation, we need to remove them.

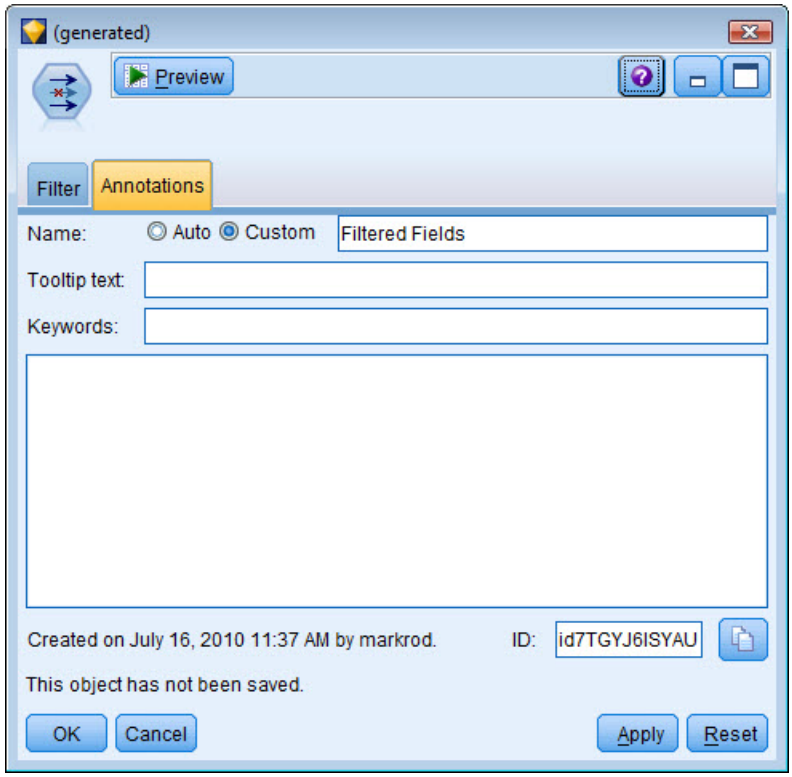
5) Click on the arrow to turn on filtering for **spousedcat**, **reason**, **polview**, **active**, and **bfast** (not shown)

There should now be 52 fields that will be sent out of the node.

Next we want to rename the node.

6) Select the **Annotations** tab
7) Select **Custom** option button
8) Enter the text **Filtered Fields**

Figure 8.9 Renaming Filter Node



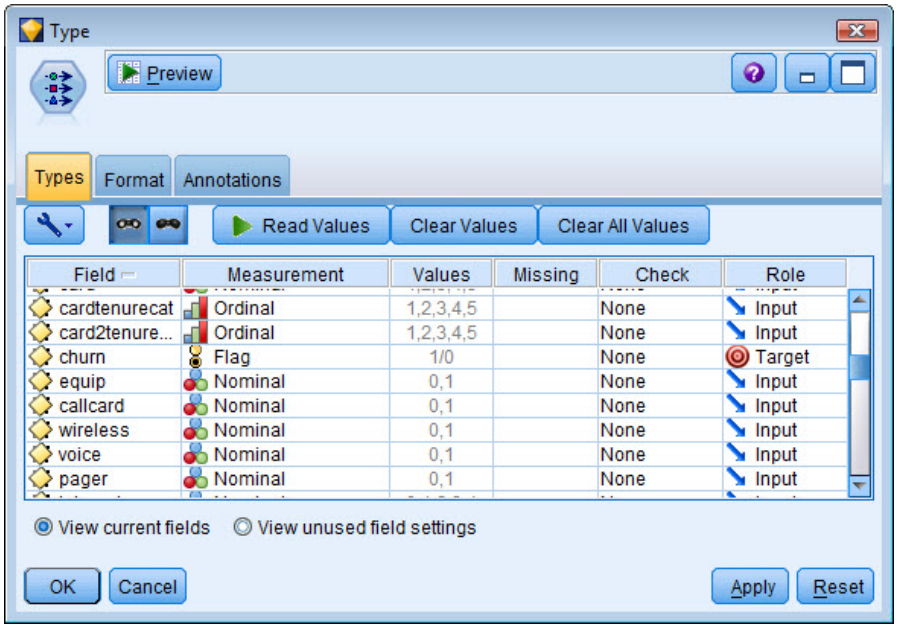
We've made the necessary changes. We have one more task to complete before we can attempt to predict *churn* with the selected fields.

Setting Churn as a Flag Field

Although it has not affected the results of the data preparation, the target field *churn* has only two categories (Yes and No), but has been given the measurement level nominal rather than flag. This difference can have an effect on the models that are developed, and it will definitely change some of the options available in the Auto Classifier modeling node that we will be using in the next lesson. So we need to make that change to *churn*. We do so in a Type node.

1) Add a **Type** node from the Field palette near the Filtered Fields node
2) **Connect** the Filtered Fields node to the Type node
3) Edit the Type node
4) Change the Measurement level of churn to **Flag**

Figure 8.10 Setting Measurement Level of churn to Flag



Several other fields are also truly flag but have measurement levels of nominal (such as *equp*, *callcard*, or *wireless*. But this won't affect how they are used in models so we won't bother to change their measurement level as well.

With data preparation now complete, we can save the stream.

1) Select **OK**
2) Select **File…Save Stream As**
3) Name the stream **Customer_Offers_Feature Selection.str**
4) Select the **Save** button

Apply Your Knowledge

1) True or False? The Feature Selection node uses a Partition field.

2) What is the default level of Importance used to select a field in the Feature Selection node?
   a. 0.05
   b. 0.95
   c. 1.00

8.5 *Lesson Summary*

In this lesson we demonstrated how to create training and testing partitions.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Use the Feature Selection node to select inputs for modeling

To support the achievement of the primary objective, students should now also be able to:

- Describe the features and settings of the Feature Selection node
- Describe the model output from Feature Selection
- Generate a Filter node to use the selected fields

8.6 Learning Activity

The overall goal of this learning activity is to use the Feature Selection node to select fields to predict *response*.

Supporting Materials

The stream file *Lesson 7 Exercise.str*. If this file was not created, you can use *Backup_Lesson 7 Exercise.str*.

1. Open the stream file *Lesson 7 Exercise.str*.
2. Add a Feature Selection node to the stream. Connect it to the Partition node.
3. Change the settings on the Model tab of the Feature Selection node as you desire. Run the node.
4. Are any fields screened out? Why?
5. Out of the remaining fields, how many are considered to be important for predicting *response*? Are any unimportant?
6. Generate a Filter node to use only the important fields. Attach this generated node to the Partition node. Name it “Fields to Predict Response.”
7. Although all these fields were deemed important, in fact, not all of them should be used because of causal order. In the Filter node, filter out these fields because they collect information that occurred after the response to the campaign:

   Promspd
   Promvis
   Promspdb
   Promvisb
   Totvisit
   Totspend

   Also filter out any transformed versions of these same fields. Also filter out *title*. Make sure you turn off filtering for the *partition* field, if necessary.
8. Add a Type node to the stream. Set the measurement level of *response* to Flag. Use the Read Values button to instantiate the data.
9. Save the stream as *Lesson 8 Exercise.str*.

Lesson 9: Automated Models for Categorical Targets

9.1 *Objectives*

After completing this lesson students will be able to:

- Use the Auto Classifier node to create an ensemble model to predict a categorical target

To support the achievement of this primary objective, students will also be able to:

- Describe and use the features and settings of the Auto Classifier node
- Describe and use the components of the model output from the Auto Classifier node

9.2 *Introduction*

Developing more than one model to predict a target is commonplace in data-mining projects. When creating a model, it isn't possible to know in advance which modeling technique will produce the most accurate result. Often several different models may be appropriate for a given data set and target, and normally it is best to try more than one.

For example, suppose we are trying to predict a binary target (buy/not buy). Potentially, we could model the data with a Neural Net, any of the Decision Tree algorithms, an SVM model, a Bayes Net, Logistic Regression, Nearest Neighbor, Decision List, or Discriminant Analysis. Trying each of these models separately can be quite time consuming, and comparing them can be less than efficient, since the results will be in separate model nuggets.

To automate this process, Modeler offers the Auto Classifier node, which can simultaneously build several different types of models to predict categorical fields. The node generates a set of models based on specified options and ranks the best candidate models according to a selected criterion. The node also combines the predictions from the top models.

We will use the Auto Classifier node to predict the customers who churned (cancelled their contract) from the telecommunications firm.

	The *customer_offers.sav* Statistics data file. These data are from former and current customers of a telecommunications company. The data includes both demographic and account-related information.
Supporting Materials	The Modeler stream file *Customer_Offers_Feature Selection.str*, which contains the results of previous data preparation.

9.3 *The Auto Classifier Node*

The Auto Classifier node allows the user to create and compare models for categorical targets using a number of methods all at the same time, and then compare the results. The modeling algorithms to be used and the specific options for each can be selected. The user can also specify multiple variants for each model. For instance, rather than choose between the Multilayer Perceptron or Radial Basis Function methods for a neural net model, both can be tried. The Auto Classifier node generates a set of models based on the specified options and ranks the candidates based on the specified criteria. The supported algorithms include Neural Net, all decision trees (C5.0, C&R Tree, QUEST, and CHAID), Logistic Regression, Decision List, Bayes Net, Discriminant, Nearest Neighbor and SVM.

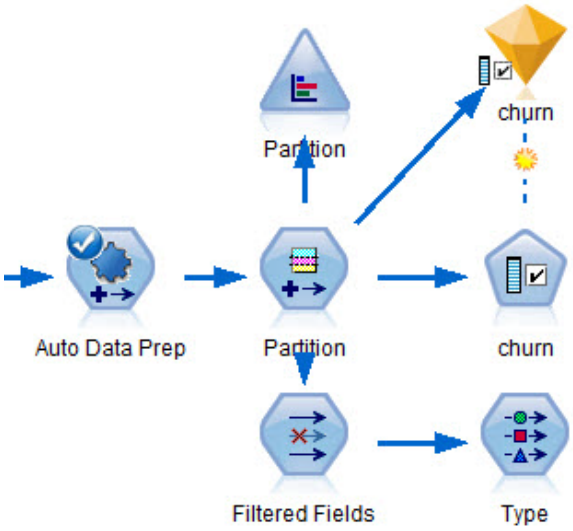
To use this node, a single target field with categorical measurement level (flag, nominal or ordinal) and at least one predictor field are required. Predictor fields can be continuous or categorical, with the limitation that some predictors may not be appropriate for some model types. For example, ordinal fields used as predictors in C&R Tree, CHAID, and QUEST models must have numeric storage (not string), and will be ignored by these models if specified otherwise. Similarly, continuous predictor fields can be binned in some cases (as with CHAID). The requirements are the same as when using the individual modeling nodes. Normally, the individual models will make the appropriate adjustments.

When an automated modeling node is executed, the node estimates candidate models for every possible combination of options, ranks each candidate model based on the specified measure, and saves the best models in a composite automated model nugget.

We will continue to work with the stream file from the previous lesson, with feature selection completed for the inputs that will be used to predict *churn*.

1) Open the stream file **Customer_Offers_Feature Selection.str**

Figure 9.1 Customer_Offers_Partition.str Stream File

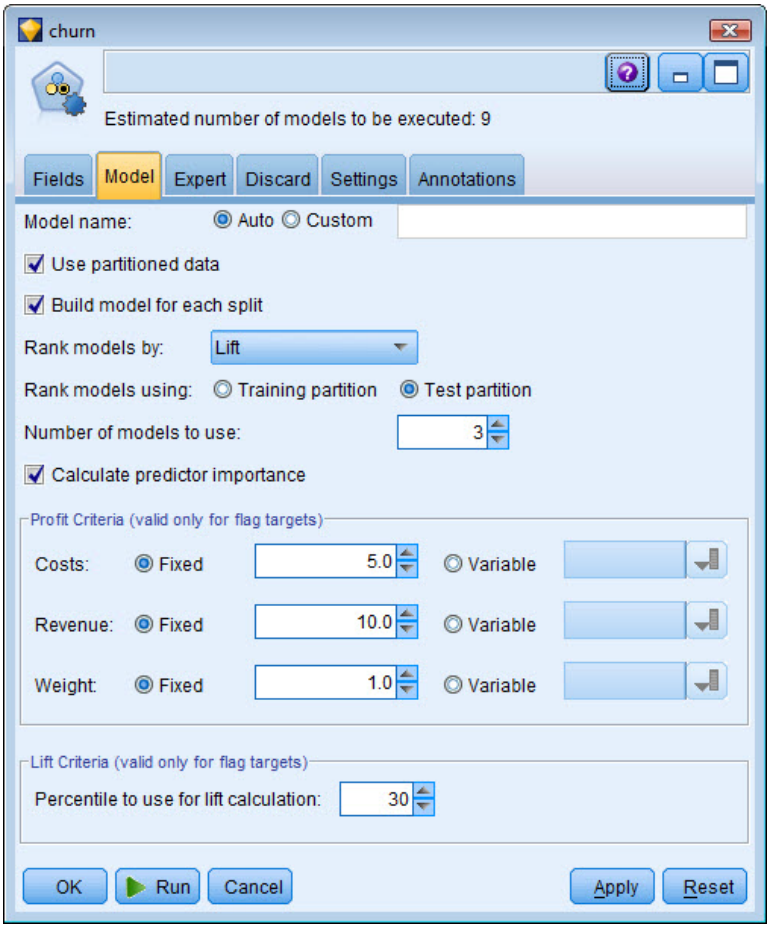


The Auto Classifier node is found in the Modeling palette. We can add an Auto Classifier node to the stream, connecting it to the Type node.

1) Add an **Auto Classifier** node from the Modeling palette to the right of the last Type node
2) **Connect** the Type node to the Auto Classifier node
3) Edit the Auto Classifier node

The node immediately recognizes *churn* as the target field.

Figure 9.2 Auto Classifier Node Model Tab



There are many options that are under user control, but the default settings are often adequate for most situations. We will review of a few of these options.

Ranking models. Models can be ranked by five different criteria, with the default being Lift. Lift is defined as the ratio of correct predictions in cumulative quantiles relative to the overall sample (where quantiles are sorted in terms of model confidence for the prediction). For example, a lift value of 3 for the top quantile indicates a hit rate three times as high as for the overall sample. The other common measure for ranking is Overall Accuracy of the model (accuracy aggregated across all categories of the target).

Some of these criteria, such as lift, are only available for flag targets.

Models can be ranked on either the Training or Testing partition. It is usually better to initially rank the models by the Training partition since the Testing data should only be used after some acceptable models have been achieved, but it is easy to switch the Partition when viewing the results.

Number of models to use. The number of models to use and display in the Auto Classifier is 3 by default, which the user can change. The top-ranking 3 models are listed according to the specified ranking criterion. Except for additional time for scoring, there is usually no penalty in increasing this value.

Profit Criteria. If the resulting cost and profit for correct and incorrect predictions are known—for example, to send offers to customers—they can be defined in the Profit Criteria area. Profit equals the revenue for each record minus the cost for the record. Profits are assumed to apply only to hits, but costs apply to all records. This is often used for direct marketing applications.

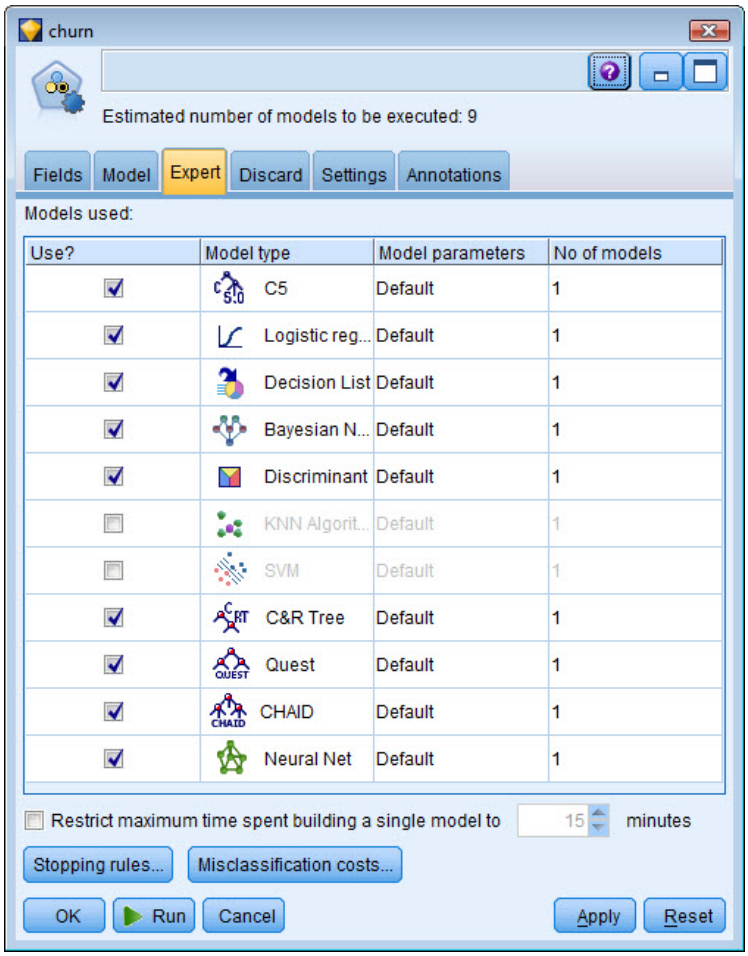
Further Information

For a flag field, the True value defined for the target is assumed to represent a hit when calculating profits, lift, and related statistics.

We will make no changes in this tab.

1) Select the **Expert** tab

Figure 9.3 Auto Classifier Node Expert Tab



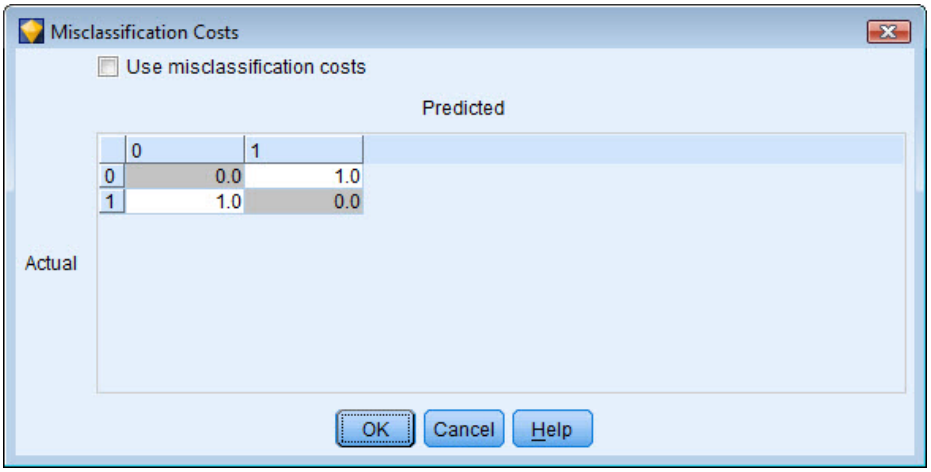
The Expert tab allows the user to select from the available model types and to specify stopping rules and misclassification costs. By default, all models are selected except KNN and SVM. However, it is important to note that the more models selected, the longer the processing time will be. Uncheck a

box to not consider a particular algorithm. The Model parameters option can be used to change the default settings for each algorithm, or to request different versions of the same model type.

To change the options for a model type, select Specify by clicking in the Model parameters cell. The specific options are similar to those available in the separate modeling nodes, with the difference that multiple options or combinations can be selected. For example, if comparing Neural Net models, rather than choosing one of the training methods, choose both of them to train the models in a single pass.

2) Select the **Misclassification costs** button

Figure 9.4 Misclassification Costs Dialog

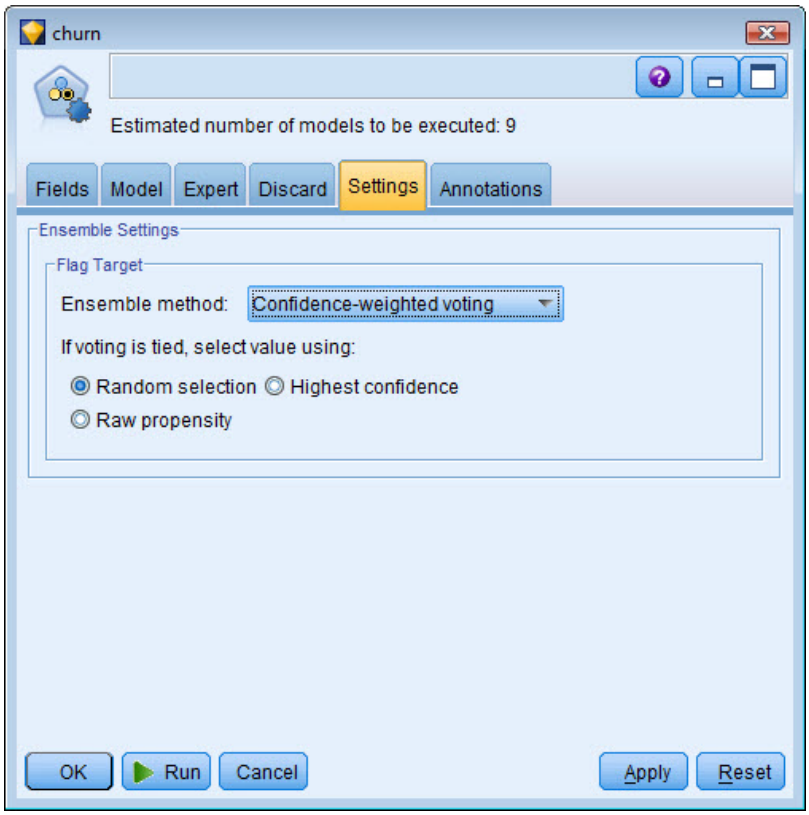


In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow the user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

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, misclassification costs are set to 0.0 for the cells with correct predictions, and 1.0 for cells that represent errors of prediction (misclassification). To enter custom cost values, select Use misclassification costs check box and enter custom values into the cost matrix. Note that these can be relative costs of an error, not absolute values.

3) Select **Cancel**
4) Select **Settings** tab

Figure 9.5 Auto Classifier Node Settings Tab



The Settings tab of the Auto Classifier node allows the user to pre-configure the method that will be used to combine model predictions (for the top three models by default). Flag targets offer more options than targets with three or more categories. The default method is confidence-weighted voting. All models output model confidence for each record, which can be thought of as the certainty of the model prediction. The model predictions are combined after being weighted by this confidence. Other choices include the highest confidence prediction, or simple straight voting. A method of breaking ties is also listed.

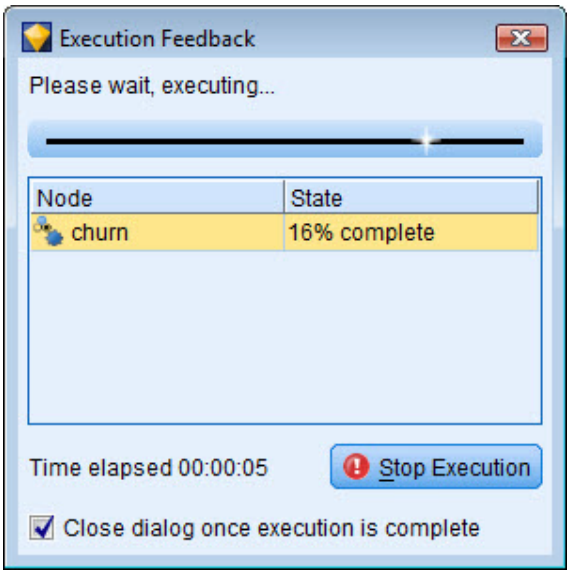
9.4 *Auto Classifier Model*

Although there are many changes that can be made in the Auto Classifier node as the user becomes more comfortable with the settings, or with the specifics of individual models, it is often quite adequate to use the default settings, and we will do that here.

1) Select **Run**

While the Auto Classifier model is running (as is true for any modeling node in Modeler), an Execution Feedback dialog provides information on how much of the modeling process is completed and the total elapsed time. There is also a button to stop execution.

Figure 9.6 Execution Feedback Dialog



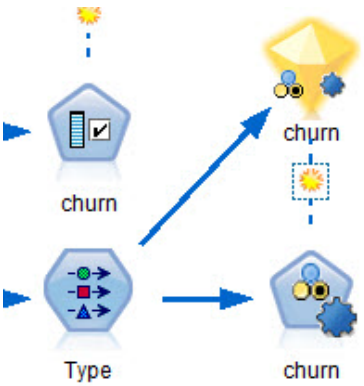
When the Auto Classifier completes execution, like the Feature Selection node, it creates a model nugget 

. Model nuggets are both placed in the Models manager or palette, in the upper right corner of the window, and on the Stream Canvas.

And in the stream, a model nugget is automatically connected to the modeling node that created it (see figure below). The link is initially shown with model replacement turned on, depicted by the small sunburst symbol in the link. In this state, re-executing the modeling node at one end of the link simply updates the nugget at the other end.

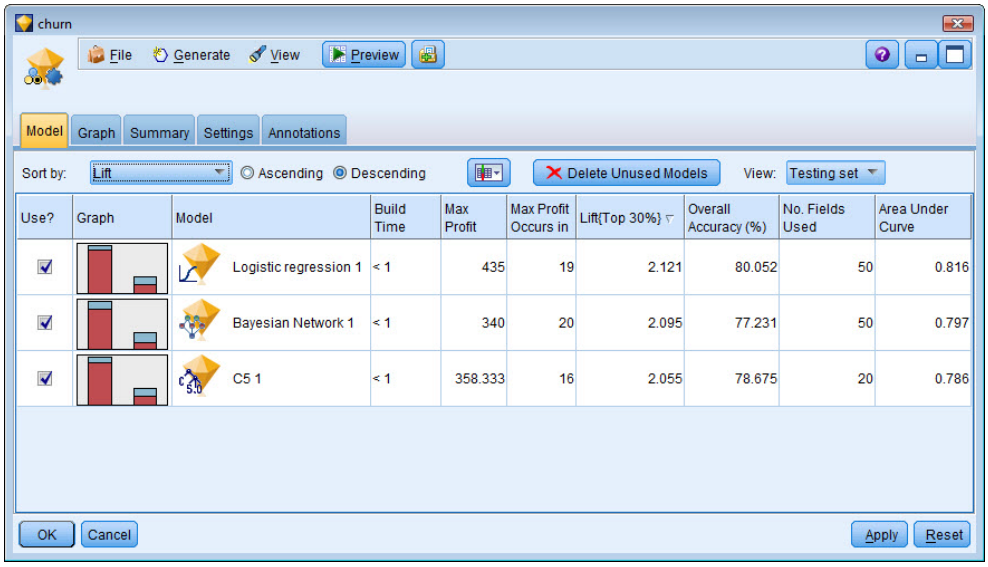
The Auto Classifier model nugget is also connected to the node before the Auto Classifier node (the Type node), so data can immediately be run through the model.

Figure 9.7 Auto Classifier Model Nugget Added and Connected in Stream



1) Edit the Auto Classifier model

Figure 9.8 Auto Classifier Model Results for Testing Partition



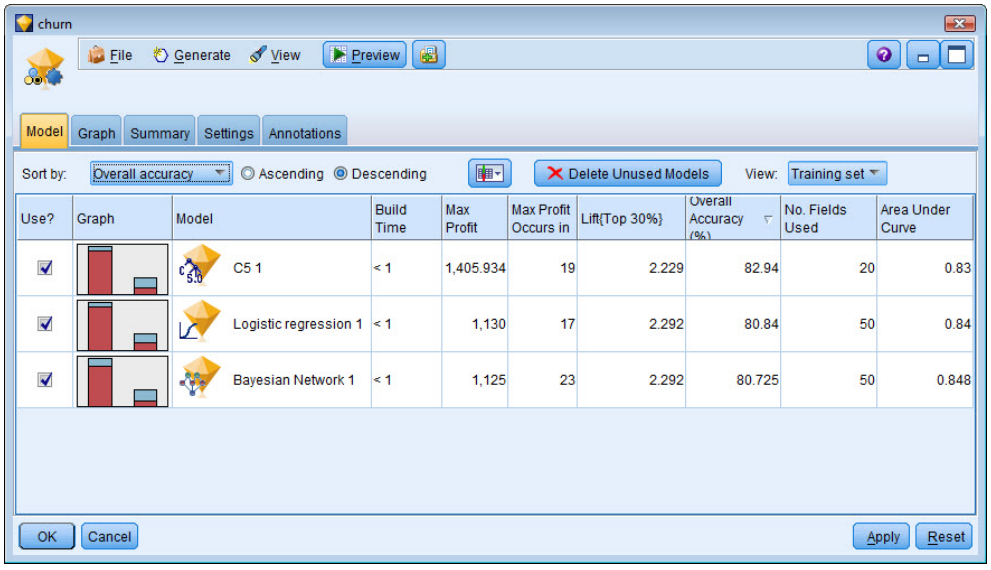
The default View is of the Testing data set (partition). Since the Training data should be reviewed first, we should switch to that before going any further.

2) Select the View: dropdown and select **Training set**

The three top models are currently sorted by Lift, but we are more interested in accuracy, so we'll change the ranking criterion.

3) Select the Sort by: dropdown and select **Overall Accuracy**

Figure 9.9 Auto Classifier Model Results for Training Partition Sorted by Accuracy



The three top performing models included in the model nugget are listed, including C5 (a decision tree-type model), Logistic regression, and a Bayesian Network. The number to the right of the model type indicates whether this is the first, second, etc. model of that type created by the Auto Classifier.

The best model is the C5 model, which has an accuracy of 82.94%. The other two models are close to that.

We can sort the results by the values in any of the columns.

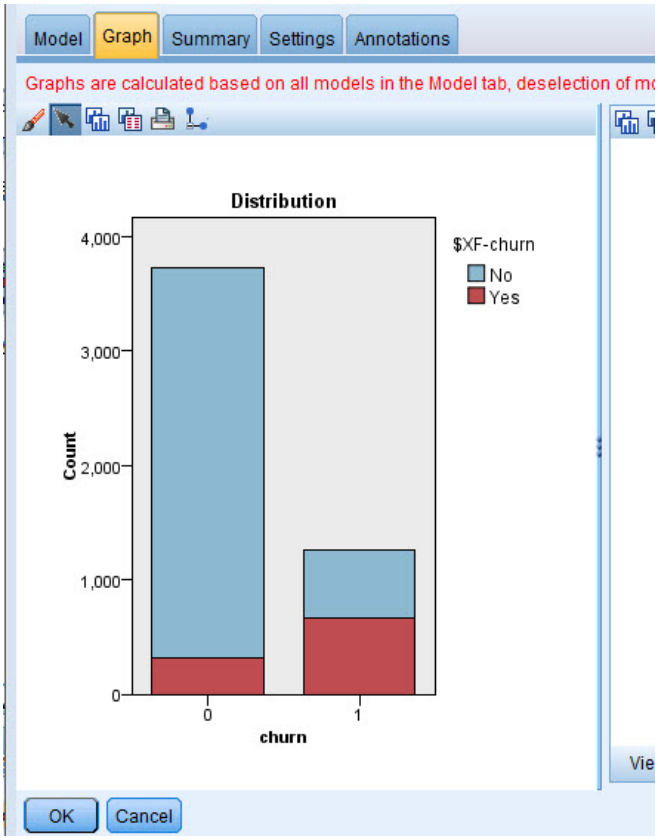
The Model cell for each column can be double-clicked, which will open a model browser window for that model. If we have sufficient knowledge, this can be helpful in determining how a model makes predictions, or how to modify that model's settings. However, remember that the model results will be combined.

In the Graph cell for each model is a thumbnail of a bar chart showing the distribution of actual values, overlaid with the predicted values, to give a quick visual indication of how many records were correctly predicted in each category. The user can double-click on a thumbnail to generate a full-sized graph. The full-sized plot includes up to 1000 records and will be based on a sample if the dataset contains more records.

The Graph tab provides summary information for the combined model, including an equivalent graph for the combination of the three models.

4) Select the **Graph** tab
5) Increase the height of the window

Figure 9.10 Distribution Graph of Predicted and Actual Values of churn

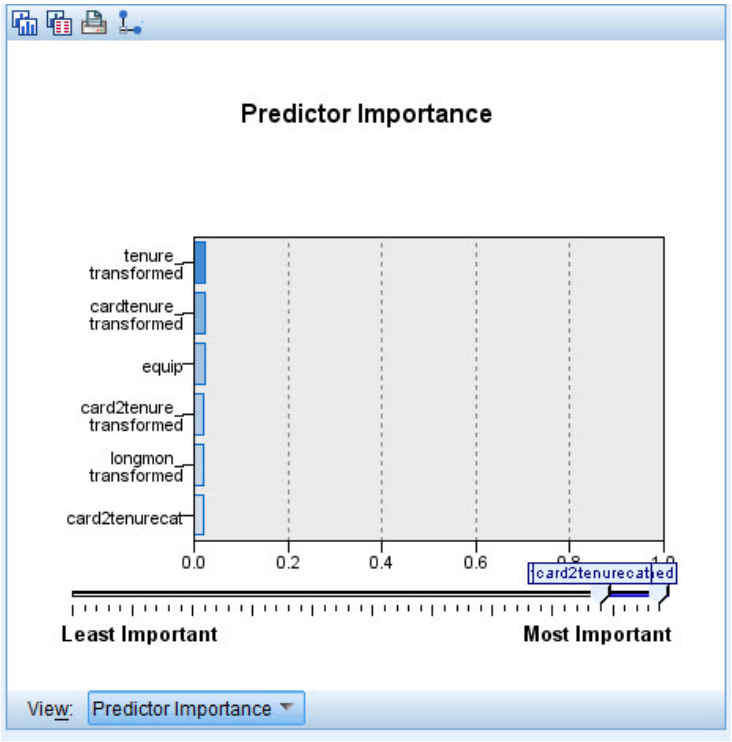


The predicted value (*$XF-churn*) is overlaid on the actual value of the target field. Ideally, if the model was 100% accurate, each bar would be of only one color because the overlap would be perfect. Here, we observe that prediction from the model is more accurate for the *No* category of *churn*, but less accurate for those customers who dropped their service (*Yes*).

On the right half of the model browser window is a Predictor Importance chart for the combined model. There is a slider that can be used to set a lower importance limit to control which fields are displayed. These are the fields that are important at predicting *churn* in the combined model.

The figure below shows the top seven predictors, which are mostly measuring either credit card information or aspects of how the customer used the telecommunications service. The best predictor, *tenure_transformed*, records how many months a customer has had a contract with the company.

Figure 9.11 Predictor Importance Chart

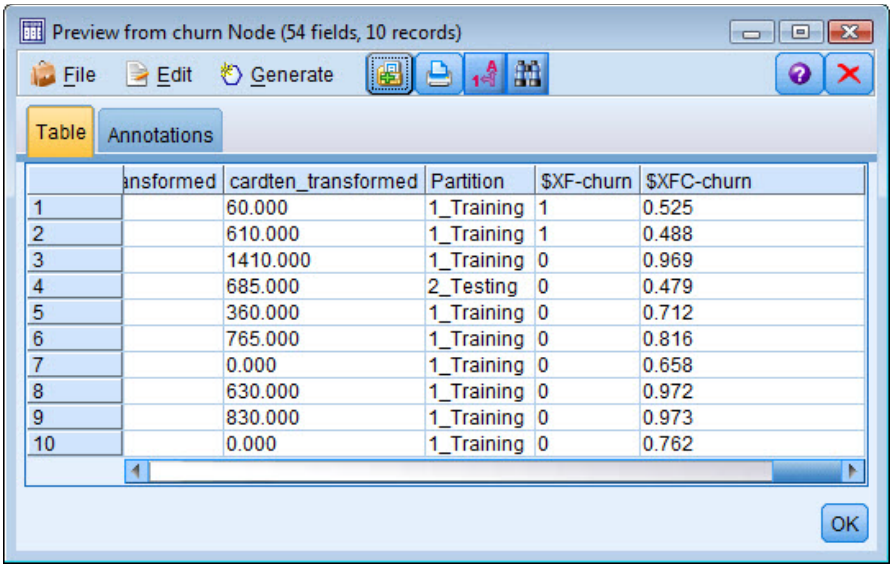


Fields Created by Model

Each model nugget creates fields that are the result of applying the model to the stream. For predictive models, these include the actual model prediction and the confidence of that prediction. Let's review these for the Auto Classifier model.

1) Select the **Preview** button
2) Scroll to the last columns in the Table window

Figure 9.12 Preview of Fields Added to Stream by Auto Classifier Model



	ansformed	cardten_transformed	Partition	$XF-churn	$XFC-churn
1		60.000	1_Training	1	0.525
2		610.000	1_Training	1	0.488
3		1410.000	1_Training	0	0.969
4		685.000	2_Testing	0	0.479
5		360.000	1_Training	0	0.712
6		765.000	1_Training	0	0.816
7		0.000	1_Training	0	0.658
8		630.000	1_Training	0	0.972
9		830.000	1_Training	0	0.973
10		0.000	1_Training	0	0.762

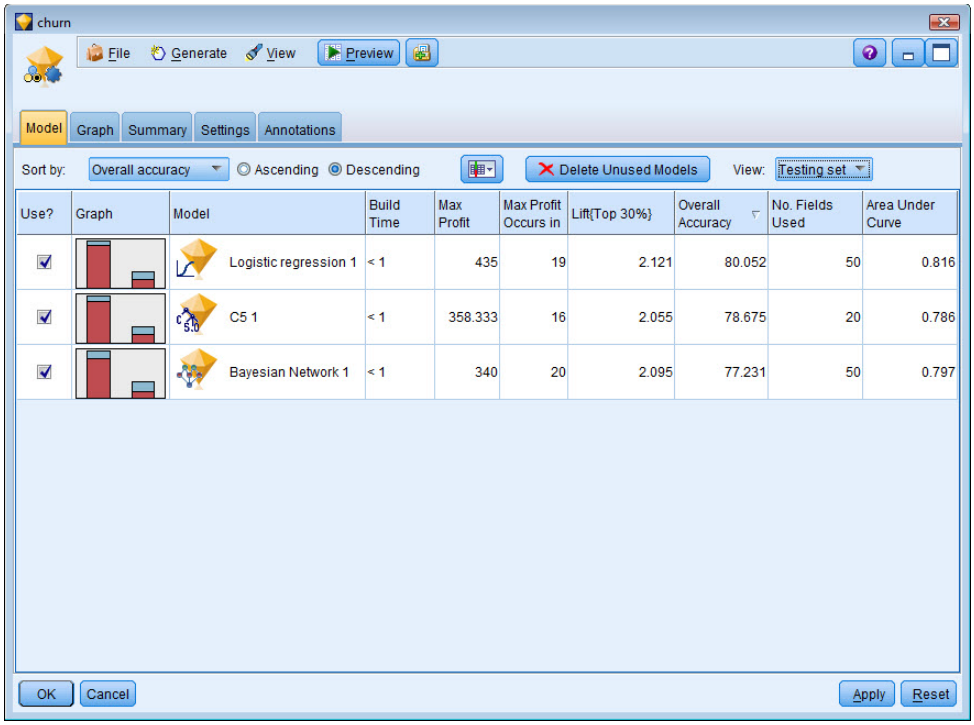
The field *$XF-churn* contains the prediction from the model for each record. The field *$XFC-churn* contains the confidence of that prediction, which can range, in this model, from about 0.33 up to 1.0. Fields created by Modeler are appended with a leading '$'.

Evaluating the Models on the Testing Partition

The real test of the model is how well it performs on the test data set, so we'll investigate that next. Note that the models are not re-estimated on the testing partition. Instead, they are simply applied to those data and predictions are made, and then the various parameters (accuracy, lift, etc.) are calculated.

1) Close the Table window
2) Select the View: dropdown and select **Testing set**

Figure 9.13 Auto Classifier Model Results for Testing Partition



The order of models has shifted slightly, with the logistic regression model now the best. As is typical, the overall accuracy has declined a bit on each of the models. The model should perform somewhat more poorly on the testing data, but still at an acceptable level.

Note

Because the models are not recalculated for the testing partition, they will have the same form, or coefficients, as in the training partition.

We will assume that the overall accuracy is adequate for the moment. The next step is to evaluate the predictions of the model in each of the categories of *churn*, and to investigate how model predictions are related to important inputs. We turn to that in the next lesson.

We will save the stream at this point.

1) Select **OK**
2) Select **File…Save Stream As**
3) Name the stream **Customer_Offers_Auto Classifier.str**
4) Select the **Save** button

Apply Your Knowledge

1) What is the default criteria used to rank models in the Auto Classifier node?
    a. Importance
    b. Accuracy
    c. Lift
    d. Number of fields

2) How does the Auto Classifier node make predictions by default?
    a. Picks the best model to use
    b. Combines the predictions with confidence-weighted voting
    c. Combines the predictions with simple voting
    d. Averages the predictions

9.5 *Lesson Summary*

In this lesson we demonstrated how to use an Auto Classifier modeling node.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Use the Auto Classifier node to create a model to predict a categorical target

To support the achievement of the primary objective, students should now also be able to:

- Describe and use the features and settings of the Auto Classifier node
- Describe and use the components of the model output from the Auto Classifier node

9.6 Learning Activity

The overall goal of this learning activity is to use the Auto Classifier node to construct a model to predict *response*.

Supporting Materials

The stream file *Lesson 8 Exercise.str*. If this file was not created, you can use *Backup_Lesson 8 Exercise.str*.

1. Open the stream file *Lesson 8 Exercise.str*.
2. Add an Auto Classifier node to the stream. Connect it to the Filter node.
3. Run the Auto Classifier node, then edit the model nugget.
4. In the training data, what are the top three models? Which model is best? What is its accuracy?
5. In the overall model, which fields are most important? Is there one field that is greatly important than others?
6. Switch to the testing data. Does the order of models change? How much did accuracy decrease on the testing data?
7. Save the stream as *Lesson 9 Exercise.str*.

Lesson 10: Model Evaluation

10.1 *Objectives*

After completing this lesson students will be able to:

- Evaluate and understand the predictions of a model

To support the achievement of this primary objective, students will also be able to:

- Use the Analysis node to get a summary table of predictions
- Use the Select node to analyze the Testing partition data
- Use a Matrix node to examine the percent accuracy of predictions
- Use a Distribution node to graphically display the relationship between a categorical predictor and the target
- Use a Histogram node to graphically display the relationship between a continuous predictor and the target

10.2 *Introduction*

In the previous lesson we developed a model with the Auto Classifier to predict customer non-retention (*churn*) at a telecommunications firm. Modeler created a combined model using the best 3 individual models. We reviewed some of the most important predictors, as well as individual model accuracy, but there was no report on overall accuracy of the combined models, or other details of the model predictions.

When a model seems satisfactory based on performance, the next phase of the CRISP-DM process is model evaluation. First, we need to examine the accuracy of model predictions in each category of the target. Then, we will want to spend some time exploring how the inputs are related to the target predictions. For example, *tenure_transformed* is the most important predictor, but is this relationship positive or negative? Are people who with lower or greater tenure more or less likely to continue with their contract with the firm?

These are important matters for model understanding, and they also may be critical in gaining model acceptance throughout the organization. Usually we don't want to treat a model as a "black box," which makes predictions in some unknown manner, even if those predictions are accurate. Also, if we are going to use the model to make changes in marketing, sales plans, customer offers, and the like, we will need to understand how the model operates.

In this lesson we will utilize several straightforward methods to explore the model for *churn*. They will provide the foundation for assessing and understanding the model, but there are others that could be used as well. And if this model exploration was unsatisfactory, then we would return to the model building stage to try to construct a model better suited to the data mining project goals as defined in the CRISP-DM methodology.

	The *customer_offers.sav* data file. These data are from former and current customers of a telecommunications company. The data includes both demographic and account-related information.
Supporting Materials	The stream file *Customer_Offers_Auto Classifier.str*, which contains the results of the model predicting *churn*.

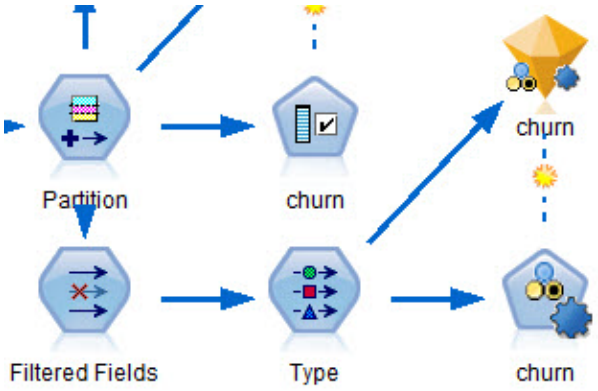
10.3 *Model Predictions with the Analysis Node*

The Analysis node constructs a classification table that shows the accuracy of the combined model, and importantly, how well it does at predicting each of the individual categories of the target. An Analysis node in a stream will report on any model that is upstream in the data flow, so the Analysis node is also useful at comparing two or more separate models.

We will work with the stream file from the previous lesson, which includes the Auto Classifier model nugget predicting *churn*.

1) Open the stream file **Customer_Offers_Auto Classifier.str**

Figure 10.1 Customer_Offers_Auto Classifier.str Stream File

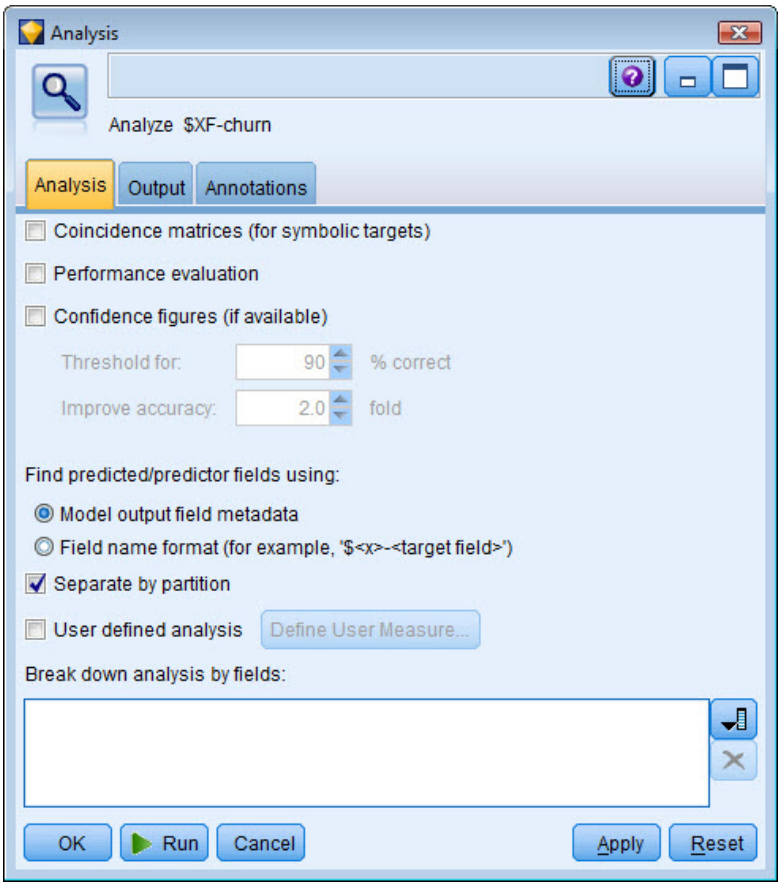


The Analysis node is found in the Output palette.

1) Add an **Analysis** node from the Output palette to the right of the churn model nugget
2) **Connect** the churn model nugget to the Analysis node
3) Edit the Analysis node

There are several types of output available from the Analysis node, but with one exception, the default choices are adequate for categorical targets.

Figure 10.2 Analysis Node Dialog



By default, results will be grouped by the partition field (Separate by partition check box). The analysis can be broken down further into groups defined by one or more categorical fields (such as looking at the model predictions within customer type).

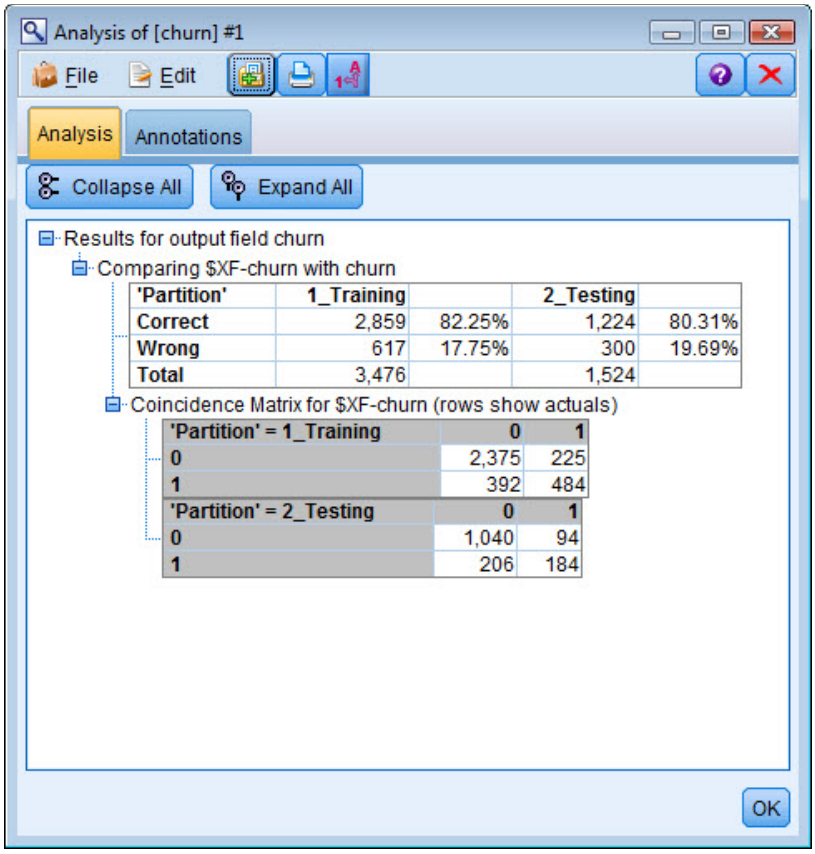
The only necessary change is to request a coincidence matrix for categorical (symbolic) targets. This is the "misclassification" table that shows the predictions in each category of the target field.

1) Select **Coincidence matrices (for symbolic targets)** check box
2) Select **Run**

Note

Valid values for *age* and *income* are imputed using a random function, so the results in the course guide figure below will not exactly match the results when re-running the stream.

Figure 10.3 Analysis Node Output for Auto Classifier Model



The first set of tables (Comparing *$XF-churn* with *churn*) provides overall accuracy results for the combined model. On the training partition, the model was accurate for 82.25% of the records. The accuracy decreased on the testing partition, but only to 80.31%. This is a typical drop for a successful model. The best individual model accuracy on the testing partition was 80.05%, so the combined models are (slightly) better than any single model.

For categorical output fields, the coincidence matrix provides predictions within categories of the target. The rows represent actual observed values of *churn* and the columns represent predicted values. The cell in the table indicates the number of records for each combination of predicted and actual values. Accurate predictions are in the 0,0 and 1,1 cells. There is a table for each partition.

Reading in the rows in the testing partition table, the model is very accurate at predicting those who do not churn (1,040 out of 1,040 + 94). The model is much less accurate at predicting those who do churn (184 out of 206 + 184). The accuracy for those who churned is less than 50%.

Further Information

The distribution of *churn* is somewhat skewed, with about three times as many No values as Yes values. Models have more difficulty predicting categories with fewer records, all things being equal. Ideally, the training data will have an equal number of records in each category of the target. This would require us to over sample records from the smaller category for model training. Note that the model must be tested eventually on data with the population distribution of the target field.

The model may or may not be acceptable, depending on whether the accuracy for those who churned is sufficient. For the moment, we will assume it is and carry on. Even if it was not, we would want to explore the model further before creating a modified model.

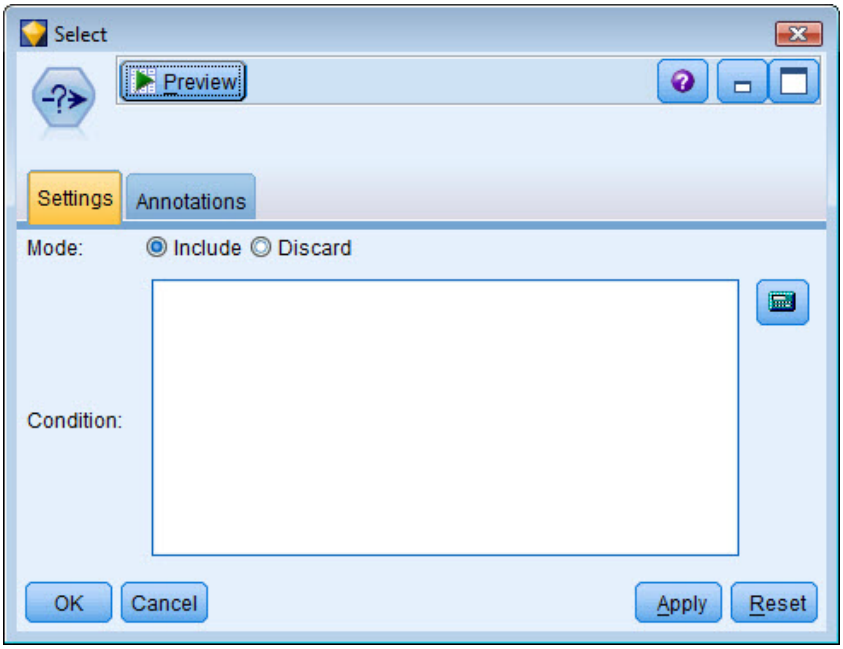
Although the Analysis node output is helpful, the percentages in each category of the target are not calculated automatically. To obtain these we can use a Matrix node, which creates a crosstabulation. To do so, though, we need to first select only the testing partition data.

10.4 *Selecting the Testing Partition Records*

Modeler provides the Select node in the Record Ops palette to do record selection in a stream. We will use it to select the testing partition data to do additional model evaluation.

1) Add a **Select** node to the Stream Canvas near the churn model nugget
2) **Connect** the churn model nugget to the Select node
3) Edit the Select node

Figure 10.4 Select Node Dialog



A Select node can be used to include or discard a subset of records from the data stream. There are two key actions to take:

- Decide on the Mode, which is either to Include or Discard records
- Create the logical condition in the Condition text box that will identify the records to include or discard

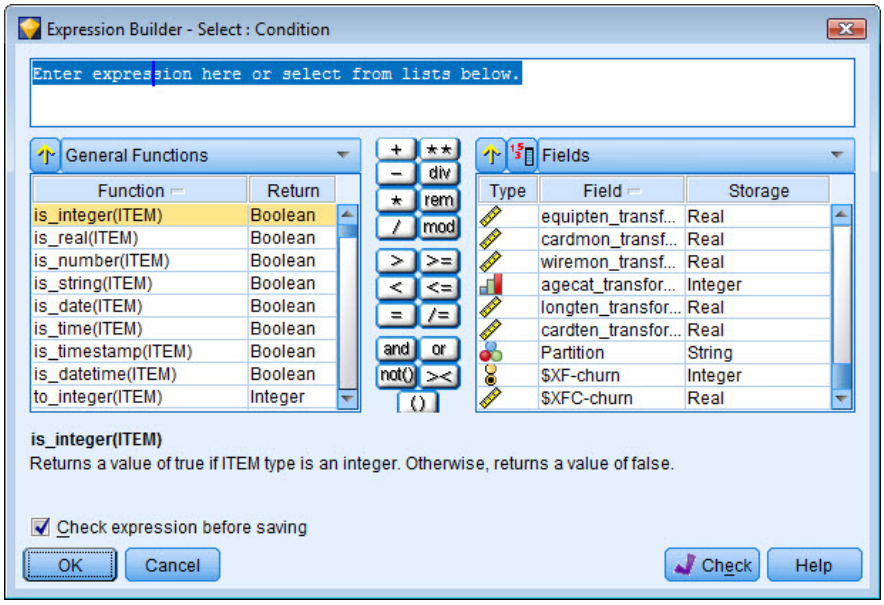
The Select node, and other Modeler nodes that require the creation of an expression, provides the Expression Builder, which makes creating the appropriate CLEM expression (the Modeler language) much less error prone, even for new users.

We will use the default choice of including records.

The Expression Builder is accessed by selecting this button: 

1) Select the **Expression Builder** button

Figure 10.5 Expression Builder Dialog



The Expression Builder provides complete lists of fields, functions, and operators, and also access to data values if the data are instantiated.

The user can type an expression into the text box at the top, or

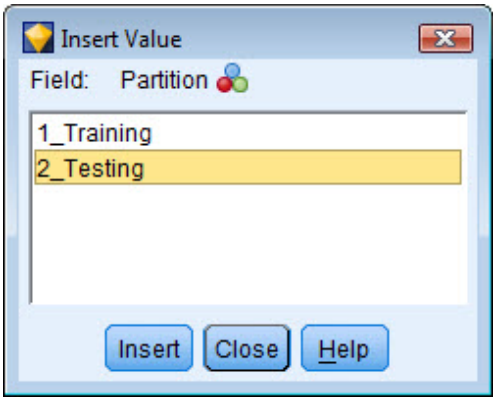
- Select the desired fields and functions from the scrolling lists
- Double-click or click the yellow arrow button to add the field or function to the expression window.
- Use the operand buttons in the center of the dialog box to insert the operations into the expression.

The expression we need to create is simple.

2) Select **Partition** from the Fields list box and move it to the Expression Builder text box

3) Select the (equal sign button)
4) Select the **Select from existing field values button** and select the value **2_Testing** in the Insert Value dialog

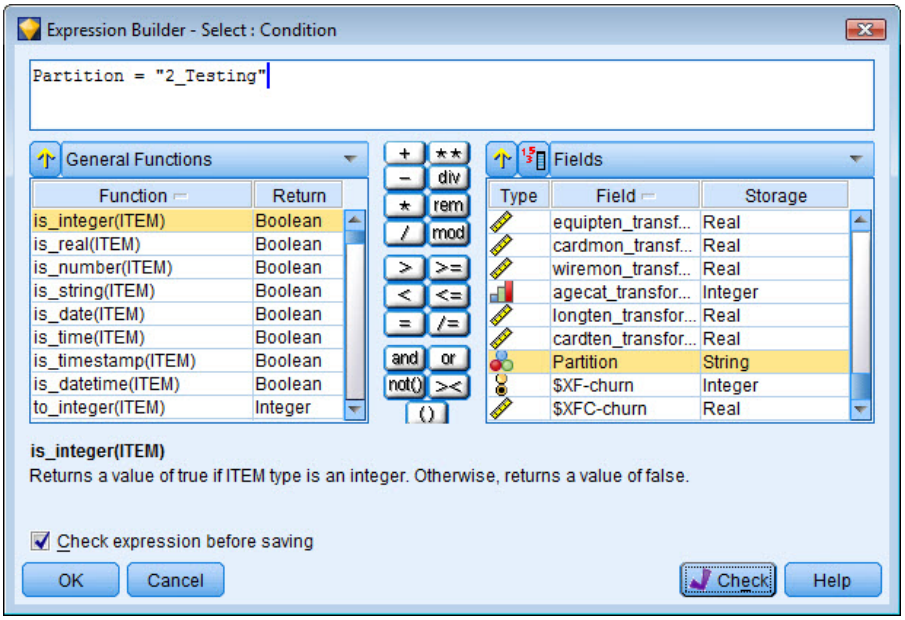
Figure 10.6 Insert Value Dialog for Partition



5) Select **Insert**
6) Select the **Check** button

The Check option will check the CLEM expression to be certain it meets syntax requirements. If the text turns black from red, the expression has no errors.

Figure 10.7 Completed Selection Expression



7) Select **OK**, and then **OK** again to complete the Select node settings

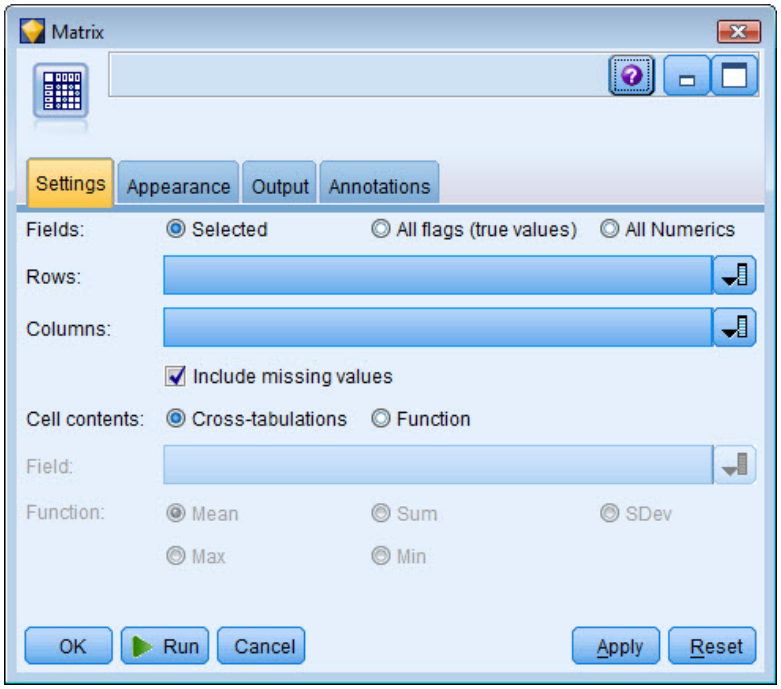
Now we can use the Matrix node.

10.5 *Using the Matrix Node for Model Predictions*

The Matrix node performs crosstabulations of two categorical fields, showing how values of one field are related to those of a second field. Cell percentages as well as the default counts can be selected. A third, continuous, field can be included as an overlay field to see how it varies across the categorical pair relationship. The Matrix node is located in the Output palette and is thus a terminal node (since it creates output there is no need for data to pass through it downstream).

1) Add a **Matrix** node from the Output palette to the right of the Select node
2) **Connect** the Select node to the Matrix node
3) Edit the Matrix node

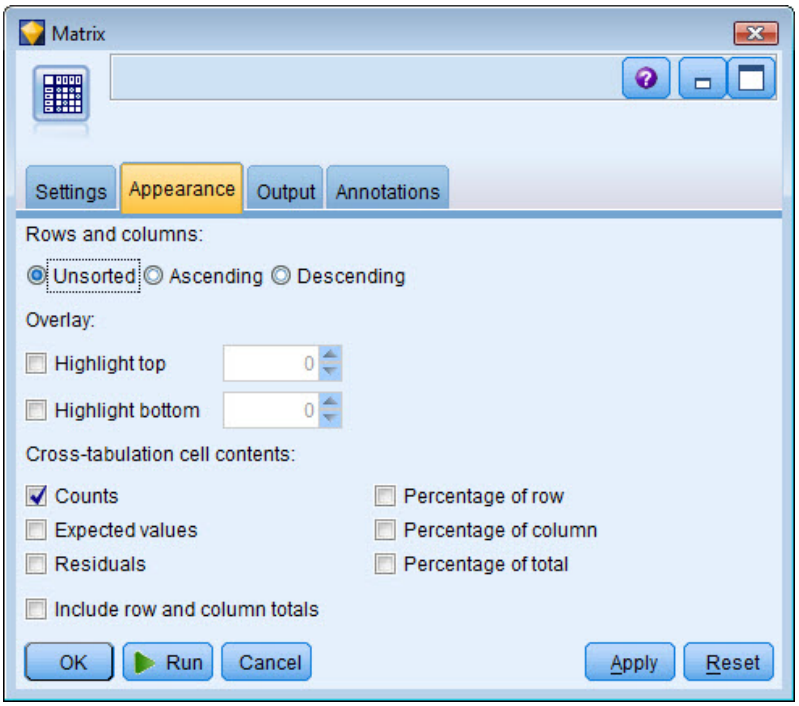
Figure 10.8 Matrix Node Settings Tab



The minimum necessary to run the node is to specify Row and Column fields. The information included in the table and its display is controlled in the Appearance tab.

1) Place the field **churn** in the Rows: list
2) Place the field **$XF-churn** in the Columns: list (not shown)
3) Select the **Appearance** tab

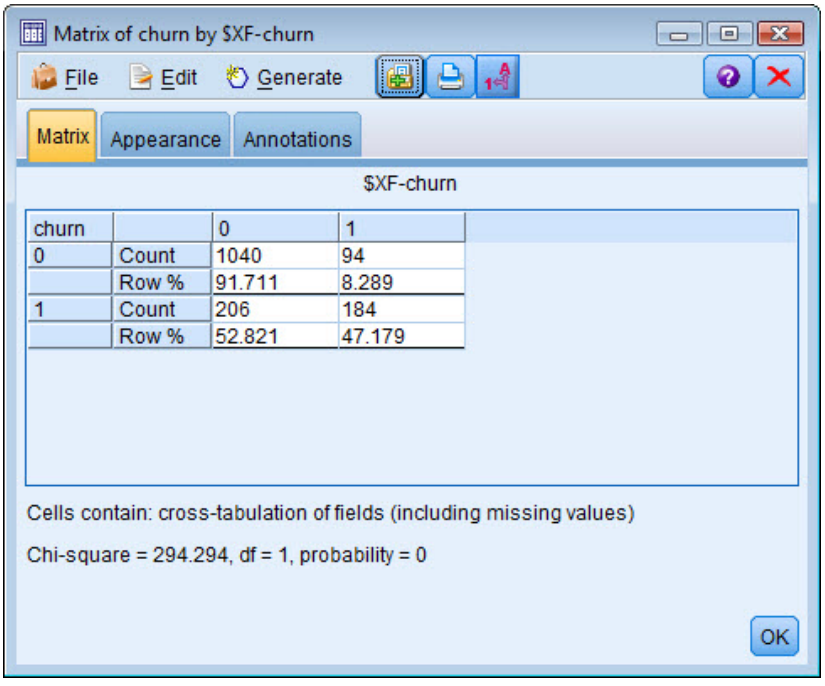
Figure 10.9 Matrix Node Appearance Tab



Looking in the Cross-tabulation cell contents area at the bottom of the dialog box, by default cell counts are displayed. We want to also request row percentages (we percentage by the observed values of *churn*).

4) Select the **Percentage of row** check box
5) Select **Run**

Figure 10.10 Matrix Node Output



The cell counts are the same we saw before with the Analysis node. The addition now is the percentages for each cell. We observe that 91.71% of those who didn't churn are predicted correctly, but only 47.18% of those who did.

The accuracy for churners might be adequate, and it's better than not having a model and therefore no method at predicting this group (a random model would accurately predict only about 25% of this group because that is their overall percentage in the data).

The Matrix node also provides a chi-square test of association, not necessary for this table.

Now that we've reviewed the model accuracy, we can turn to seeing how the model inputs are related to the predictions.

6) Close the Matrix Node Output browser

10.6 *Model Predictions for Categorical Input Fields*

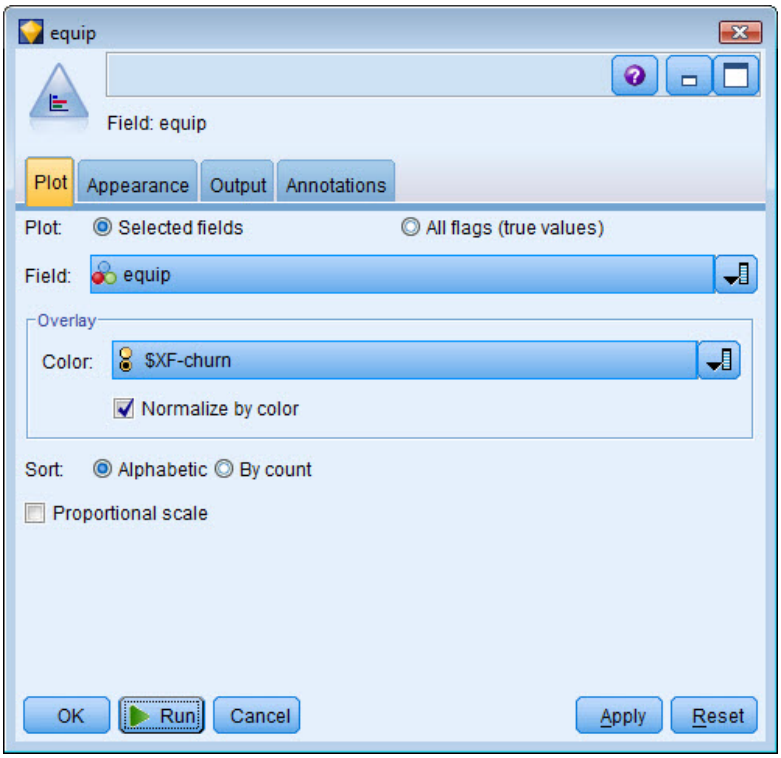
There are several categorical fields that were selected for use in modeling. As an analyst, we will want to explore how these fields are related to the target predictions. We could use a Matrix node, since both the input and target are categorical, but we can also visually view the relationship in a bar chart with the Distribution node, which we have used in a previous lesson.

One of the fields that was identified as important was *equip* (did the customer rent equipment from the firm). We'll see how that relates to predicted customer retention.

1) Add a **Distribution** node from the Graphs palette to the stream near the Select node
2) **Connect** the Select node to the Distribution node
3) Edit the Distribution node
4) Select **equip** as the Field
5) Select **$XF-churn** as the Overlay
6) Select **Normalize by color** check box

The Normalize by color choice sets all bars so that they take up the full width of the graph. The overlay values equal a proportion of each bar, making comparisons across categories easier.

Figure 10.11 Distribution Node Dialog

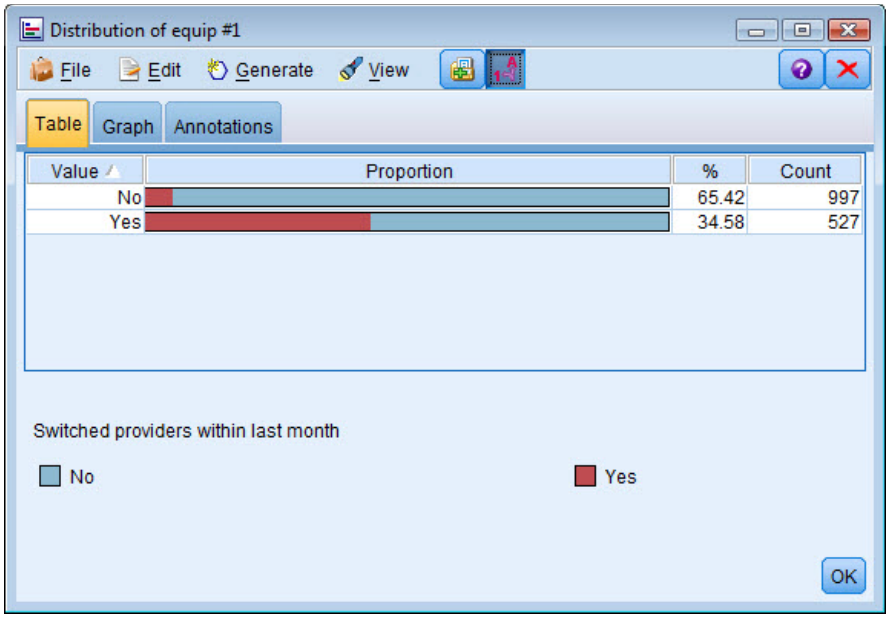


7) Select **Run**

When the Distribution node browser opens:

8) Select the **Display fields and value labels** button 

Figure 10.12 Distribution Graph for equip Overlaid by Predicted churn



Each bar represents a category of *equip*. The colors in the bars represent either those who are predicted not to churn (blue) or churn (red). There is a dramatic difference between the two groups. Those who rented equipment are far more likely to churn. Open the Graph tab to see a larger graph with a percentage scale.

This difference illustrates why *equip* was an important predictor, and it shows how that field is used in the combined model. Given this relationship, the telecommunications firm could consider what might be causing equipment rental customers to be more likely to cancel their contract, and what could be done to prevent that from happening in the future.

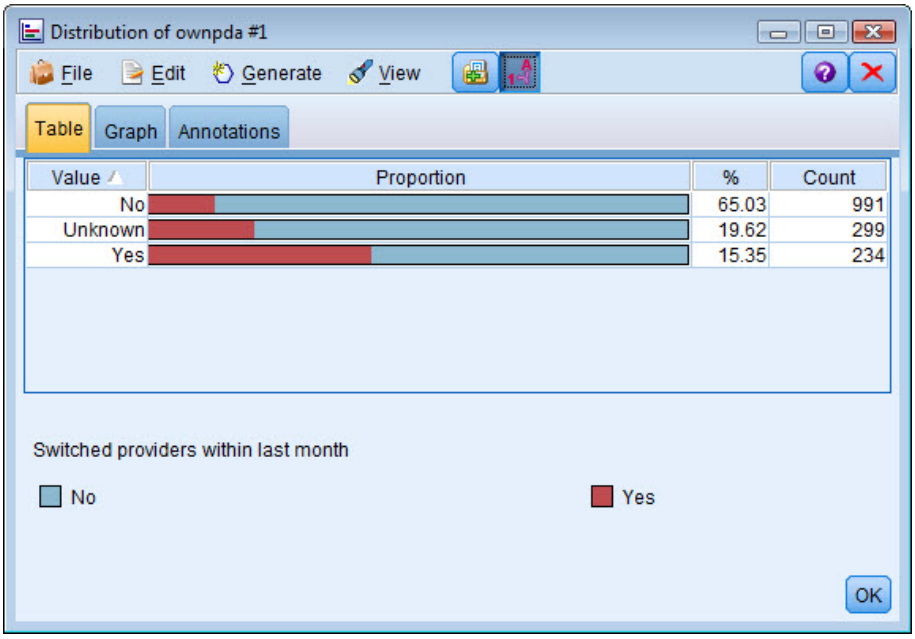
During data preparation we coded the missing values of several categorical variables with the value of 9 and made that a valid value. And we noted that sometimes missing data can be useful in a model. We can illustrate this by creating a Distribution chart for *ownpda*.

1) Select **OK**
2) Edit the Distribution node
3) Change the Field to **ownpda** (not shown)
4) Select **Run**

When the Distribution node browser opens:

5) Select the **Display fields and value labels** button 

Figure 10.13 Distribution Graph for ownpda Overlaid by Predicted churn



We see an interesting relationship between *ownpda* and predicted churn, and it appears that *ownpda* should be another good predictor. Customers who don't own a pda have the lowest churn, customers that do own a pda have the highest churn, and those whose pda ownership is unknown are in between. This indicates that most probably the unknown group is comprised of a mix of those who do and don't own a pda, which makes common sense. But since almost 20% of the customers fall in the Unknown category, it wouldn't have been prudent to drop them from the model.

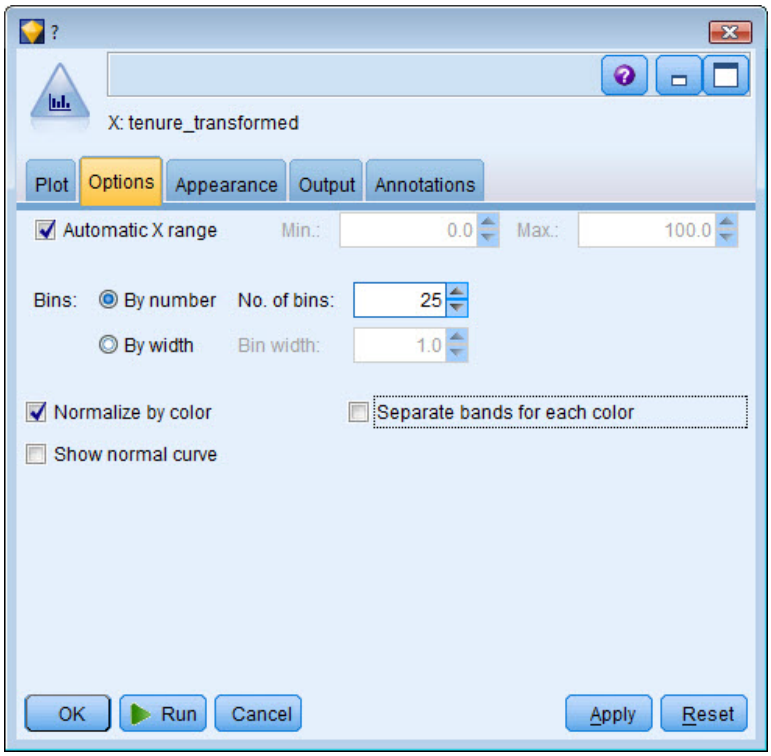
6) Close the Distribution browser

10.7 *Model Predictions for Continuous Input Fields*

We next complete an analogous examination of how the model predictions are related to continuous fields. A histogram can be used to see how a continuous field is related to predicted churn. We can begin with *tenure_transformed*, which was the most important predictor in the model.

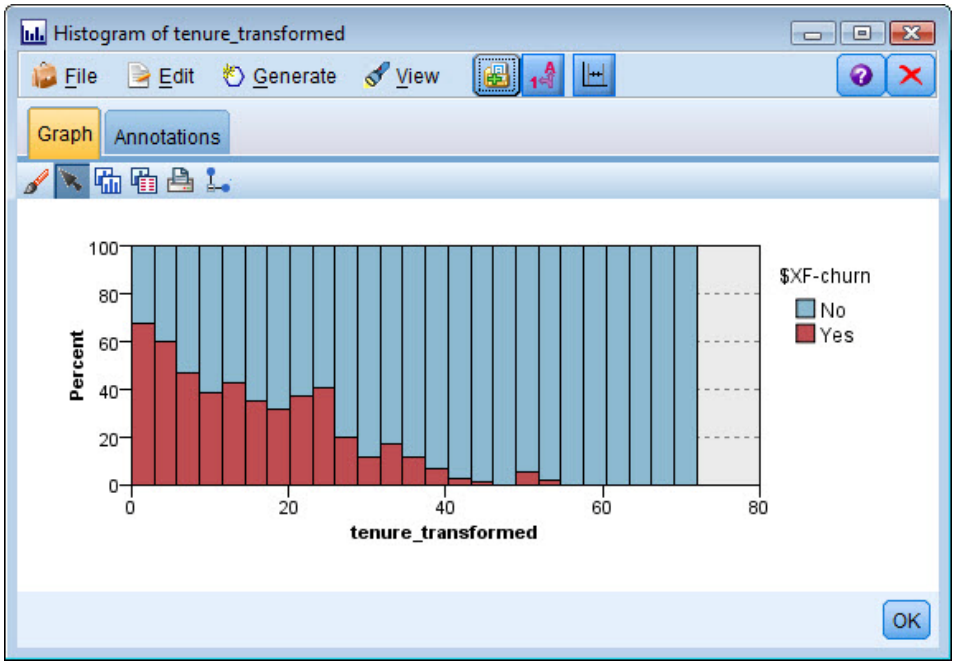
1) Add a **Histogram** node from the Graphs palette to the stream near the Select node
2) **Connect** the Select node to the Histogram node
3) Edit the Histogram node
4) Select **tenure_transformed** as the Field
5) Select **$XF-churn** as the Color Overlay field (not shown)
6) Select the **Options** tab
7) Select **Normalize by color** check box

Figure 10.14 Histogram Options Tab



8) Select **Run**

Figure 10.15 Histogram of tenure_transformed Overlaid with Predicted churn

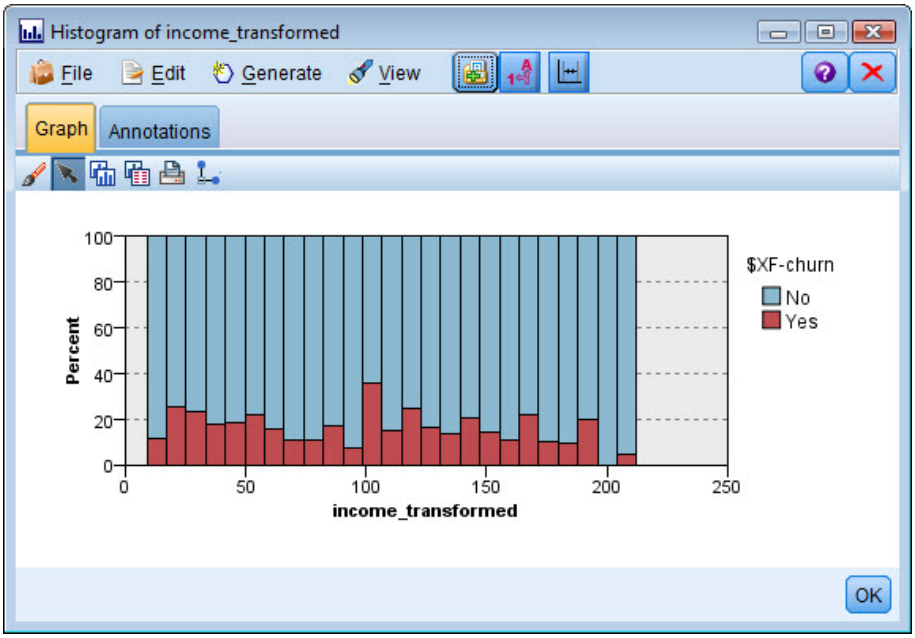


There is a very clear and definite relationship between time as a customer and churn. The more time as a customer (tenure is measured in months), the less likely the model predicts that a customer will churn. Conversely, new customers are quite likely to churn in their first few months (jumping from one company to another?)

What does this type of chart look like when there is no relationship between a predictor and the target? We can answer this question by using *income_transformed* in a histogram.

1) Select **OK**
2) Edit the Histogram node
3) Change the Field to **income_transformed** (not shown)
4) Select **Run**

Figure 10.16 Histogram of income_transformed Overlaid with Predicted churn



The distribution of No and Yes responses for predicted churn is about the same throughout the range of income. This means there is essentially no relationship between the two fields, and so the telecommunications firm could ignore income when applying the results of the model.

Tip

What if we want to investigate the relationship between fields not included in the data stream and the target (recall that fields were filtered)? We can either:

- Edit the Filter node, turning off the filtering for some fields
- Add a copy of the Filter node and create an alternative data stream. We can temporarily disconnect the existing Filter node and connect the new one. Or we can copy all necessary nodes and make two completely separate streams.

In the interests of time we will terminate any further model evaluation, but in practice we would spend a significant amount of time at model evaluation and understanding.

We will save the stream at this point.

1) Select **OK**
2) Select **File…Save Stream As**
3) Name the stream **Customer_Offers_Model Evaluation.str**
4) Select the **Save** button

Apply Your Knowledge

1) How does the Analysis node help understand model predictions? Select all that apply.
    a. It organizes results by partition
    b. It ranks the most important fields
    c. It displays overall percentage accuracy
    d. It displays percentage accuracy within categories of the target

2) True or false? An overlay field in a Distribution or Histogram chart should be categorical.

10.8 *Appendix: Improving the Model*

What can be done to improve a model that is not satisfactory? What strategies should be followed? Here are some suggestions:

- Try the Auto Classifier node with different variants of some of the models
- Try adding the SVM and KNN models to model construction
- Balance the training sample distribution on the target so that it has close to a equal distribution across categories
- Add more variables to be used in the Auto Classifier, even if they were previously filtered out
- Create new variables (such as the sum of long distance and local minutes of usage) that might be an alternative and better predictor

These are just some of the more common approaches to model improvement. Some of them can be done by anyone, even those new to Modeler or data mining. Others take some experience or knowledge of the specific models.

10.9 *Lesson Summary*

In this lesson we demonstrated how to evaluate a model.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Evaluate and understand the predictions of a model

To support the achievement of the primary objective, students should now also be able to:

- Use the Analysis node to get a summary table of predictions
- Use the Select node to analyze the Testing partition data
- Use a Matrix node to examine the percent accuracy of predictions
- Use a Distribution node to graphically display the relationship between a categorical predictor and the target
- Use a Histogram node to graphically display the relationship between a continuous predictor and the target

10.10 Learning Activity

The overall goal of this learning activity is to evaluate the model created to predicte the field *response*.

Supporting Materials

The stream file *Lesson 9 Exercise.str*. If this file was not created, you can use *Backup_Lesson 9 Exercise.str*.

1. Open the stream file *Lesson 9 Exercise.str*.
2. Use an Analysis node to evaluate the model predictions. What is the accuracy on the training data? What is the accuracy on the testing data? Is the overall accuracy less than on the best model? Why might that be?
3. Add a Select node to the stream and select out the testing partition.
4. Add a Matrix node to the stream and calculate the percent correctly predicted in each category of *response*. Which category can be predicted more accurately?
5. There are several categorical inputs in the stream. Use a Distribution node to examine the relationship between these and predicted *response*. Describe the relationships.
6. There are three continuous predictors. Use a Histogram node to examine the relationship between them and predicted *response*. Are the relationships positive or negative?
7. Save the stream as *Lesson 10 Exercise.str*.

Lesson 11: Automated Models for Continuous Targets

11.1 *Objectives*

After completing this lesson students will be able to:

- Use the Auto Numeric node to create an ensemble model to predict a continuous target

To support the achievement of this primary objective, students will also be able to:

- Describe and use the features and settings of the Auto Numeric node
- Describe and use the components of the model output from the Auto Numeric node
- Use various nodes for model evaluation

11.2 *Introduction*

In a previous lesson we learned how to automate the production of models to predict categorical targets with the Auto Classifier node. In this lesson we discuss the Auto Numeric node, which in an analogous manner can automate the production of models for targets that are numeric with a continuous level of measurement.

The Auto Numeric node allows us to create and compare models for continuous targets using a number of methods all at the same time, and then compare the results. The user can select the modeling algorithms to be used and the specific options for each. Multiple variants for each model can also be specified. The supported algorithms include Neural Net, C&R Tree, CHAID, Regression, Linear, Generalized Linear Models, KNN and SVM.

Predictor fields can be categorical or continuous, with the limitation that some predictors may not be appropriate for some model types. For example, C&R Tree models can use categorical string fields as predictors, while regression models cannot use these fields and will ignore them if specified. The requirements are the same as when using the individual modeling nodes.

In the example in this lesson we will attempt to predict the time, in months, that a customer has had a contract with the telecommunications firm (*tenure*).

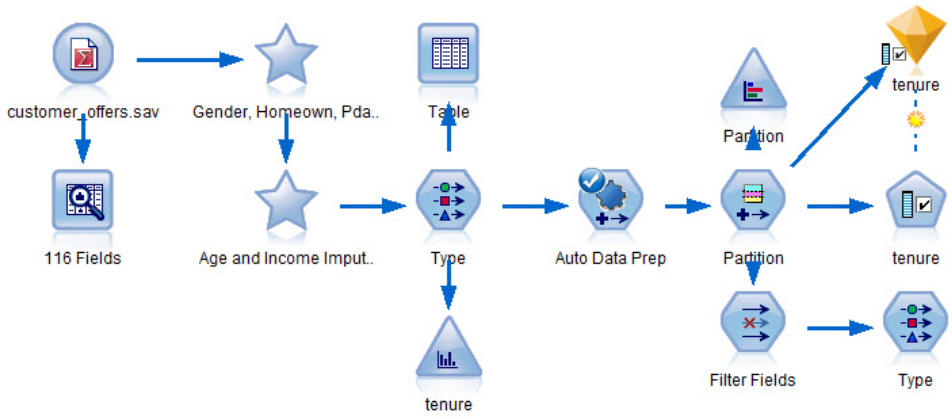
	The *customer_offers.sav* Statistics data file. These data are from former and current customers of a telecommunications company. The data includes both demographic and account-related information.
Supporting Materials	The Modeler stream file *Customer_Offers_Tenure.str*, which contains the data preparation for a model predicting *tenure*.

11.3 *Data Preparation Stream to Predict Tenure*

Rather than work through all the steps of data preparation to create a model to predict customer tenure, we will use an existing stream in this lesson that has made the data ready for modeling. We are using the same *customer_offers.sav* data file that was used to predict *churn*, so several of the steps in data preparation will be the same. The file name is *Customer_Offers_Tenure.str*. Let's open the file and review the nodes.

1) Open the stream file **Customer_Offers_Tenure.str**

Figure 11.1 Customer_Offers_Tenure.str Stream File



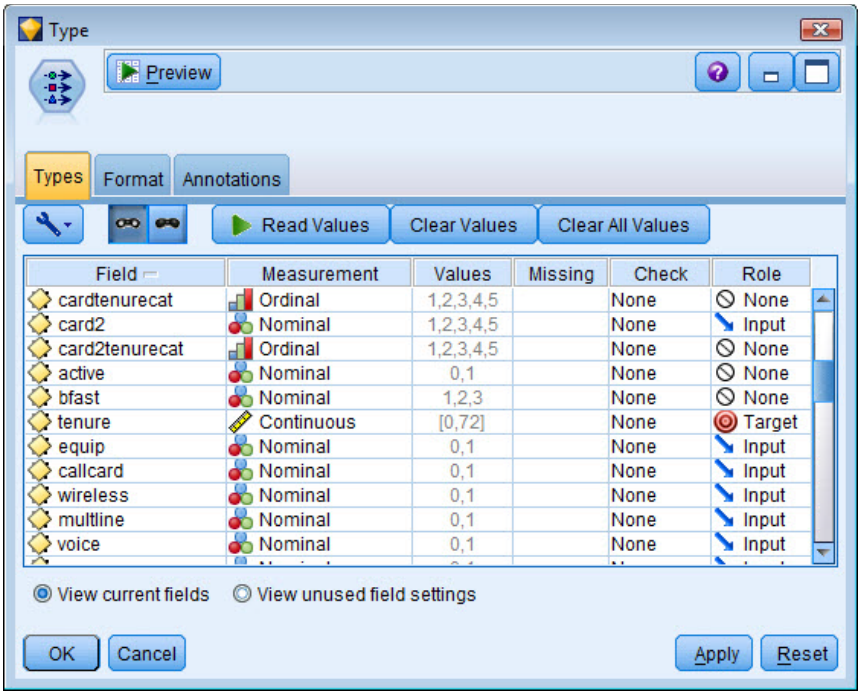
Here is how the data are prepared in this stream:

- The Source node set the target as *tenure* and set the other fields as inputs, except for *churn*
- The Data Audit node was used to impute missing data for *age* and *income* and to set missing values for *gender, homeown*, and *ownpda* to 9
- The first Type node coerced out-of-range values for *age* and *income* into the range endpoints
- The ADP node replaced outlier values in continuous fields with the cutoff value, and replaced missing values with the mean and median, respectively, for continuous and ordinal fields; input fields with low quality, as with the data prepared to predict *churn*, were also excluded
- The Partition node partitions the data with the same split as used previously
- The Feature Selection node was used with default settings, and then a Filter node was generated from the Feature Selection model nugget; 56 fields were selected by the model

The Type node removes additional fields as inputs for the model. The reason is a crucial one for model building, and we'll open the node to discuss this issue.

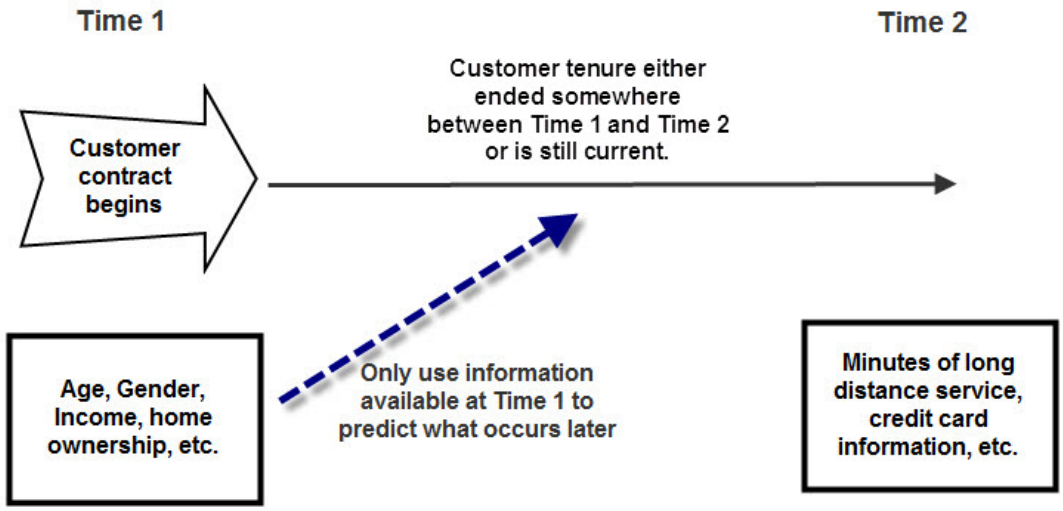
2) Edit the last Type node in the stream
3) Scroll down until the fields concerned with credit cards, e.g., **cardtenurecat**, are visible

Figure 11.2 Type Node with Field Roles



Although Feature selection indicated that, for example, *cardtenurecat* and *card2tenurecat* would be important predictors, they should not be used in predicting *tenure*. The reason is the time ordering of the information. The telecommunications firm, understandably, would like to predict the length of time a customer will use their services soon after they sign up. In this model, we are attempting to predict customer *tenure* with information available at the time, or soon after, a customer signed a contract with the firm. Consequently, information on how long someone has held a credit card, which is current information (at the end of data collection), or the amount of services used (which happened long after a customer signed up), cannot be used for modeling. To do so would be a serious error in causal direction. The next figure illustrates the situation.

Figure 11.3 Causal Diagram for Predicting Customer Tenure



As a consequence, many fields had their role set to None in the Type node so they would not be used for modeling (compare the Filter Fields node to the Type node to see which fields had their role changed).

Additionally, although Feature Selection indicated that *gender* would not be a good predictor of *tenure*, that field was included in the model. It was available when a customer signed up, and the firm wanted to use as many of these fields as possible when developing the model.

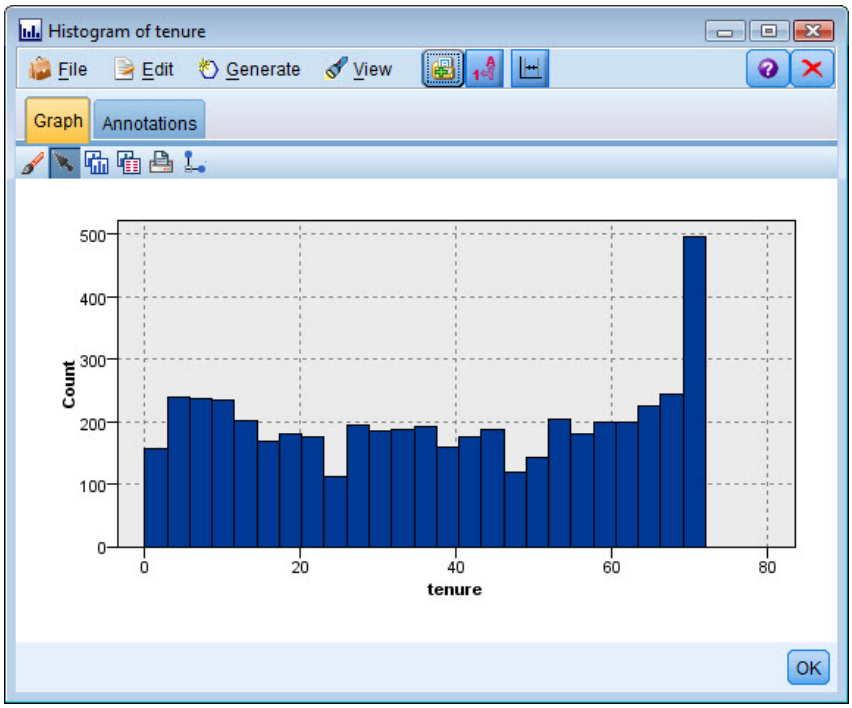
Note

We might want to look at other demographic-type fields that were filtered out by the Feature Selection model to see whether they could be included as well.

Before we use the Auto Numeric node, we should review the distribution of the field *tenure*. There is a predefined Histogram node attached to the first Type node that will create a histogram for *tenure*.

1) Right-click the Histogram node named **tenure** and select **Run**

Figure 11.4 Histogram of tenure



The distribution is quite uniform, except for a spike at just over 70 months (or about 6 years). These data were collected over a six year period, and customers who still had contracts at that point all have the same value for *tenure*. It appears that the rate that customers cancel their contracts is fairly uniform over time.

We are now ready to use the Auto Numeric Node.

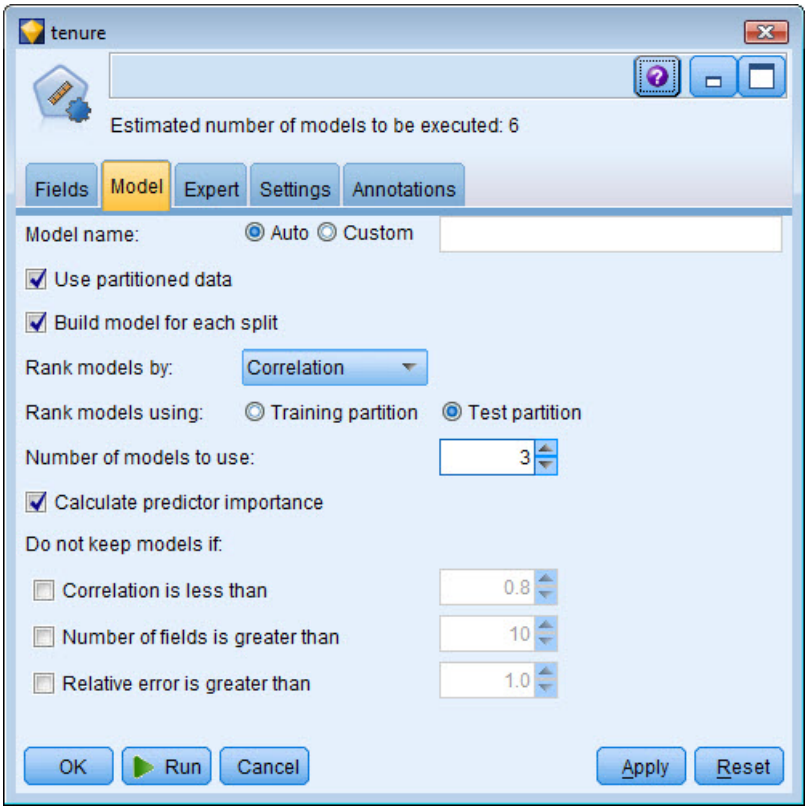
11.4 *The Auto Numeric Node*

The Auto Numeric node allows the user to create and compare models for continuous targets using a number of methods all at the same time, and then compare the results.

We can add an Auto Numeric node to the stream, connecting it to the Type node.

1) Add an **Auto Numeric** node from the Modeling palette to the right of the last Type node
2) **Connect** the Type node to the Auto Numeric node
3) Edit the Auto Numeric node

Figure 11.5 Auto Numeric Node Model Tab



The settings in the node are similar, or identical, to the Auto Classifier node, so we will only discuss the key differences.

Ranking models. Models can be ranked by three different criteria, with the default being the (Pearson) Correlation between the target and the predicted value of the target. The other common measure for ranking is Relative Error, which is the ratio of the variance of the observed values from those predicted by the model to the variance of the observed values from a baseline model that uses the mean value of the target field as the prediction. For a good model, this value should be much less than 1, indicating that the model is more accurate than the baseline model. The third available criterion is the number of fields.

If the relationship between predicted and observed values is not linear, the correlation is not a good measure of fit or ranking. We'll view scatterplots to make that determination.

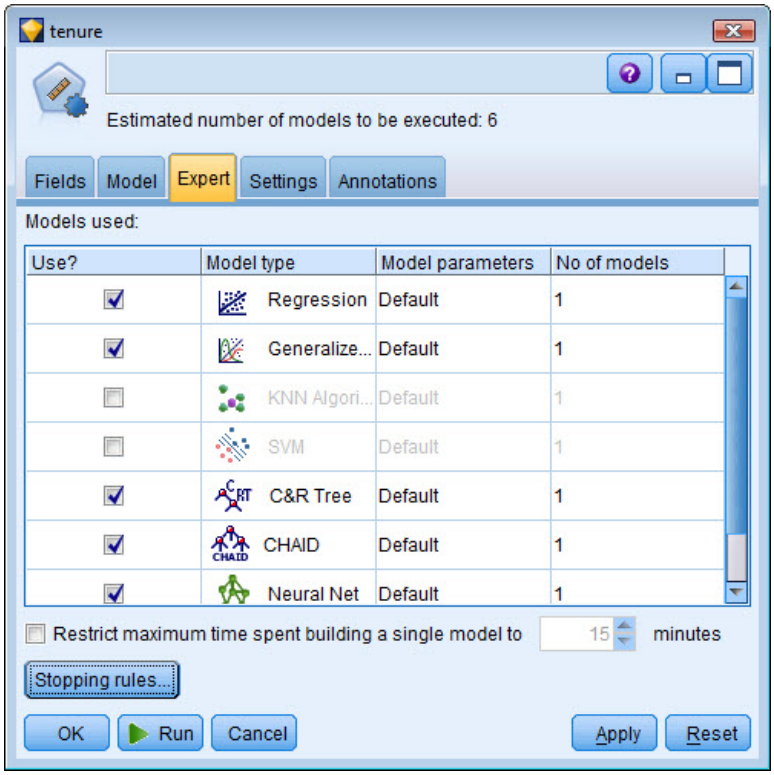
Further Info

The number of models to be used in the ensemble is 3, which can be changed.

Options are available on this tab to discard a model which doesn't meet a criterion based on one or more of the ranking options.

4) Select **Expert** tab

Figure 11.6 Auto Numeric Node Expert Tab

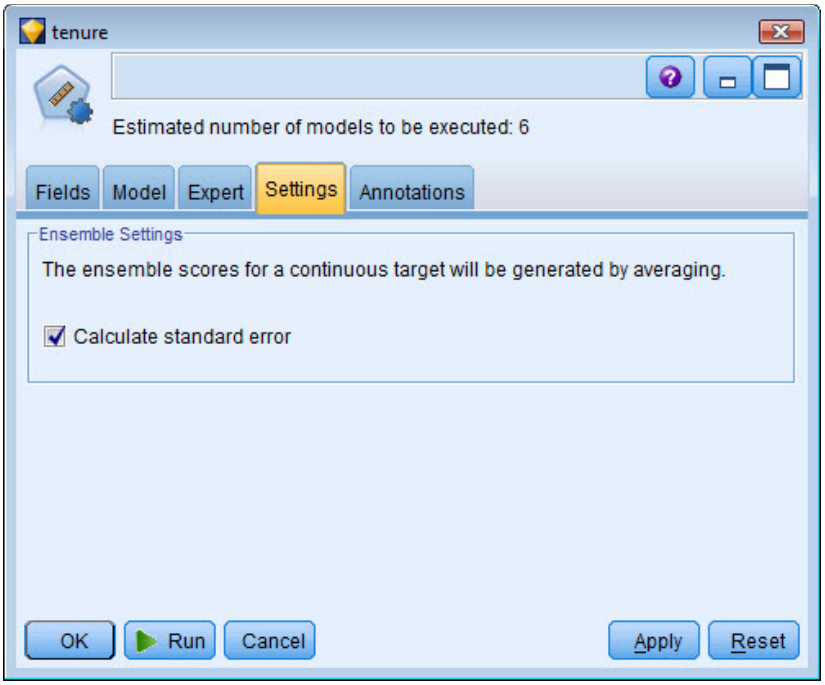


The Expert tab allows the user to select from the available model types and to specify stopping rules. By default, all models are selected except KNN and SVM (6 in total). Uncheck a box to not consider a particular algorithm, or check a box to select one. The Model parameters option can be used to change the default settings for each algorithm, or to request different versions of the same model type.

To change the options for a model type, select Specify by clicking in the Model parameters cell. The specific options are similar to those available in the separate modeling nodes, with the difference that multiple options or combinations can be selected.

5) Select **Settings** tab

Figure 11.7 Auto Numeric Node Settings Tab



As the note states in this tab, model scores (predictions) will be created for each record by averaging the predictions for the ensemble of models (because we are predicting a continuous target). The standard error is calculated by default to measure the difference between the predicted and observed values.

11.5 *Auto Numeric Model*

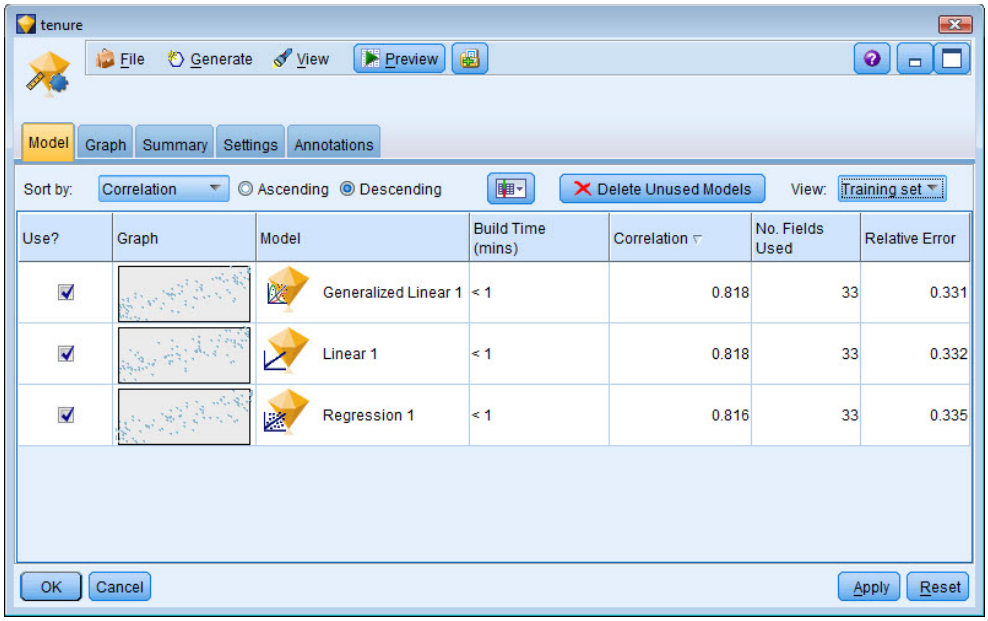
Although there are many changes that can be made in the Auto Numeric node, it is often quite adequate to use the default settings, and we will do that for this example.

1) Select **Run**

After the node has executed:

1) Edit the Auto Numeric model
2) Select the View: dropdown and select **Training set**

Figure 11.8 Auto Numeric Model Results for Training Partition Sorted by Correlation



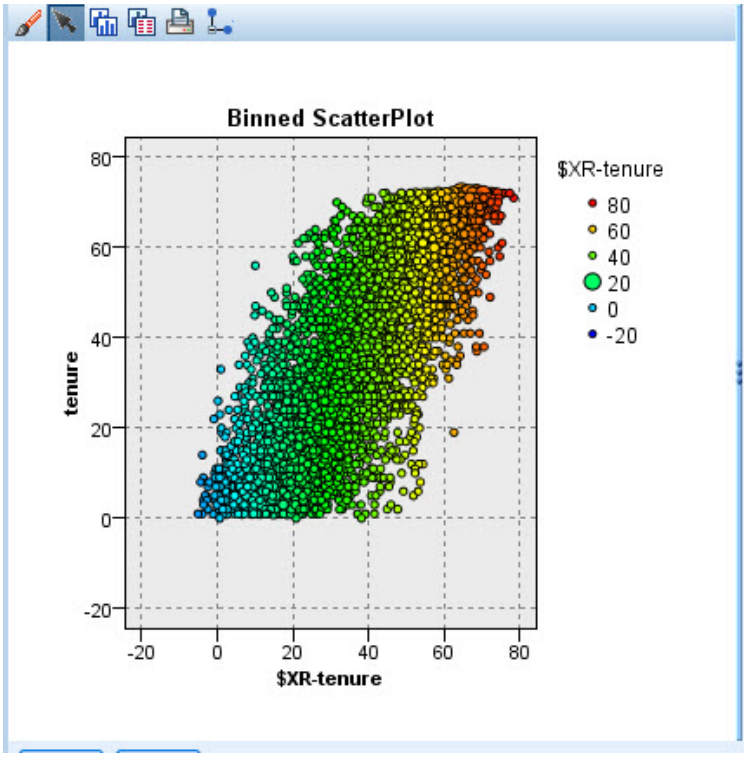
The three top performing models included in the model nugget are listed. They include a Generalized Linear Model, a Linear model, and a Regression model. All of these are variants of standard statistical methods. All the models have very similar correlations with the target (about 0.82) and relative errors (about 0.33). Although there is no universally agreed rule of thumb, in general we would like to have a correlation above about 0.70 or so, as this indicates that about half of the variance in the target can be explained by the model (explained variance equals the correlation squared, so a correlation of .7 gives an explained variance of $(0.7)^2$ ~= .5).

In the Graph cell for each model is a thumbnail of a bar chart showing the distribution of actual values, overlaid with the predicted values, to give a quick visual indication of how many records were correctly predicted in each category. Double-click on a thumbnail to generate a full-sized graph. The full-sized plot includes up to 1000 records and will be based on a sample if the dataset contains more records.

The Graph tab provides summary information for the combined model, including an equivalent graph for the combination of the three models.

3) Select the **Graph** tab
4) Increase the height of the window

Figure 11.9 Scatterplot of Customer Tenure and Predicted Tenure

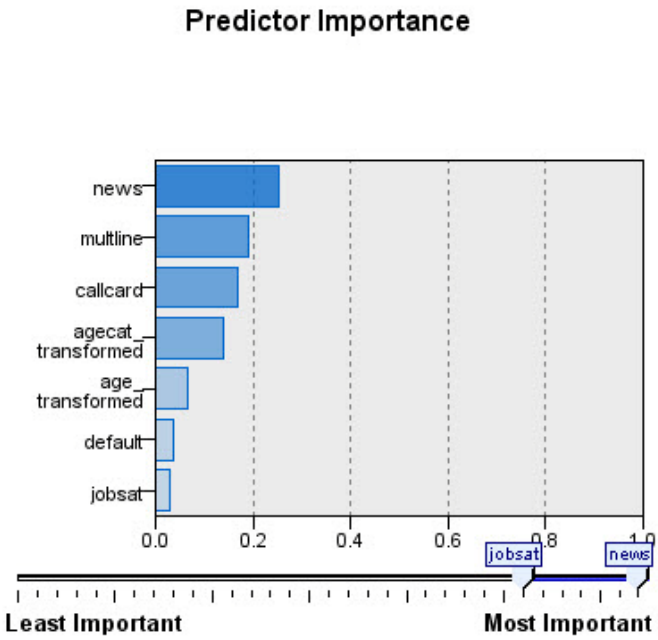


The values for predicted tenure (*$XR-tenure*) are binned, but the key point is the general shape and spread of the graph. For a good model, points should tend to cluster along the diagonal from lower left to upper right rather than be scattered randomly across the graph. Although there might appear to be a fair amount of scatter, this model is actually quite good. We also look for outliers that are difficult to predict.

On the right half of the model browser window is a Predictor Importance chart for the combined model. There is a slider that can be used to set a lower importance limit to control which fields are displayed. These are the fields that are important at predicting *tenure* in the combined model.

The figure below shows the top seven predictors, which are either measuring the types of services that a customer signed up for initially, age (in two versions), and other customer characteristics (had a newspaper subscription or ever defaulted on a loan). Unlike in the model for *churn*, there are some fields here which are much more important than the other fields. As usual, the importance does not tell us the direction of influence: does having a newspaper subscription increase or decrease time as a customer?

Figure 11.10 Predictor Importance for tenure Model

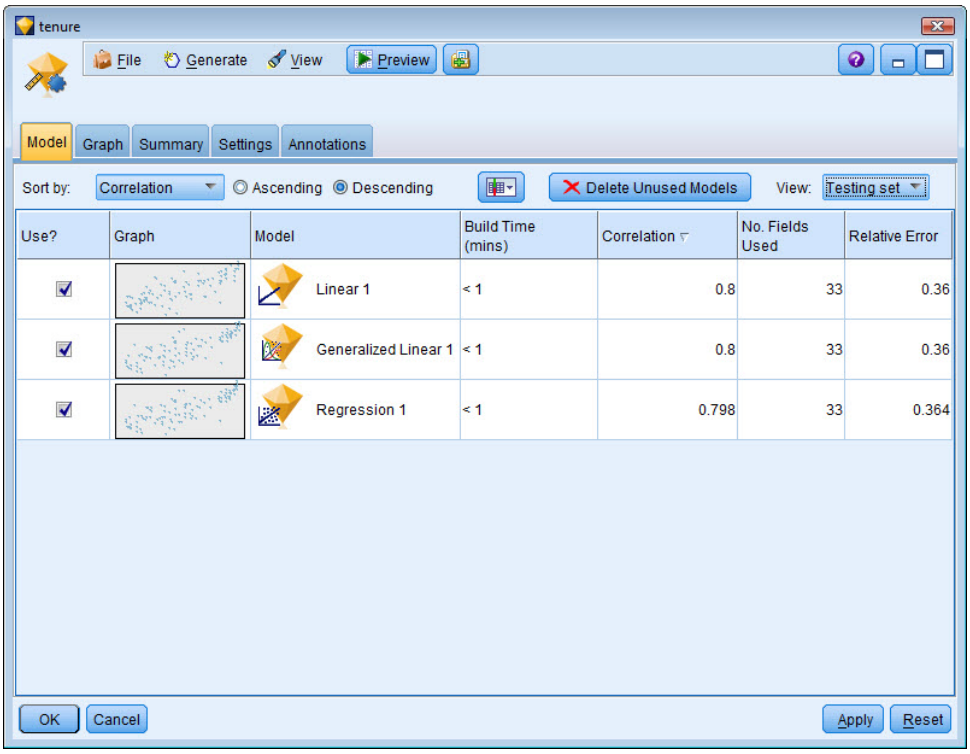


Evaluating the Models on the Testing Partition

As always, the real test of the model is how well it performs on the test data set, so we'll investigate that next.

1) Select the **Model** tab
2) Select the View: dropdown and select **Testing set**

Figure 11.11 Auto Numeric Model Results for Testing Partition



The order of models has shifted slightly, with the Linear model now the best. As is typical, the correlation has declined a bit on each of the models, and the relative error has increased slightly. But both of these measures are still good.

Of course, we really care about the results for the combined model, so we'll spend some time evaluating that model in this lesson, following the same basic process as with the Auto Classifier model.

11.6 *Model Predictions with the Analysis Node*

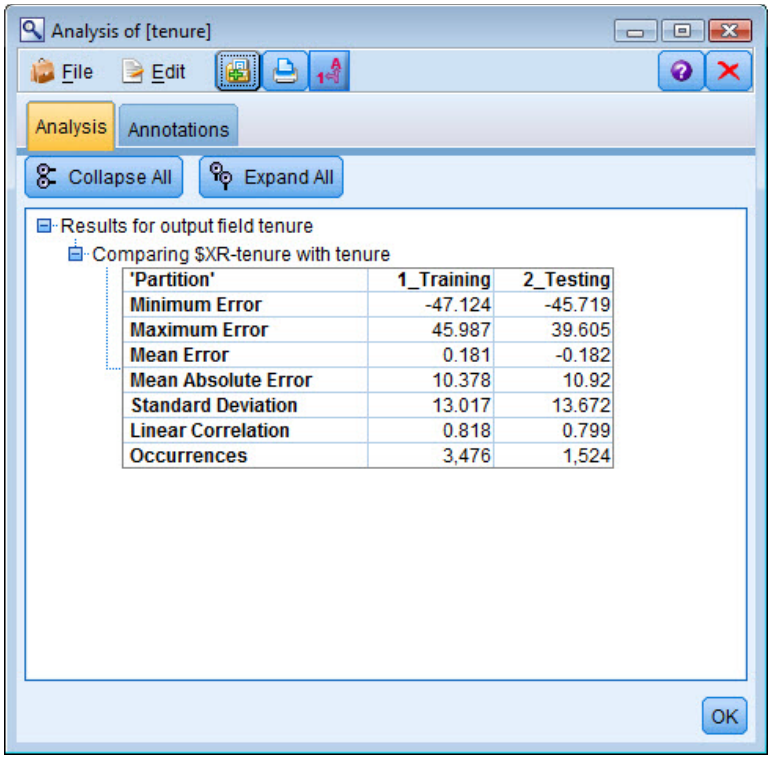
The Analysis node will allow us to assess the overall accuracy of a model created with the Auto Numeric node. The Analysis node calculates a variety of statistics appropriate for continuous targets, separated into the training and testing partitions.

1) Close the Auto Numeric Model browser
2) Add an **Analysis** node from the Output palette to the right of the tenure model nugget
3) **Connect** the tenure model nugget to the Analysis node

The default settings are adequate.

4) Run the **Analysis** node

Figure 11.12 Analysis Node Output for Auto Numeric Model



'Partition'	1_Training	2_Testing
Minimum Error	-47.124	-45.719
Maximum Error	45.987	39.605
Mean Error	0.181	-0.182
Mean Absolute Error	10.378	10.92
Standard Deviation	13.017	13.672
Linear Correlation	0.818	0.799
Occurrences	3,476	1,524

The Analysis node output contains the following information:

- Minimum Error. The largest minimum error for any customer (difference between observed and predicted values) which in practice is usually the largest negative error (the model over predicts).
- Maximum Error. The maximum error (largest positive value) for any customer (the model under predicts).
- Mean Error. The mean of errors across all records. This indicates whether there is a systematic bias (a stronger tendency to overestimate than to underestimate, or vice versa) in the model. The mean error should be reasonably close to zero, which it is for this model.
- Mean Absolute Error. The average of the absolute values of the errors across all records. Indicates the average magnitude of error, independent of the direction. This is often the most usual model measure. Here, the model is off by about 10.92 months, on average, on the testing data.
- Standard Deviation.
- Linear Correlation.

The model performs almost as well on the testing data as the training data, and better on some measures. Whether the model is sufficiently accurate is a judgment call that depends on the goals set initially in the data mining project.

11.7 *Selecting the Testing Partition Records*

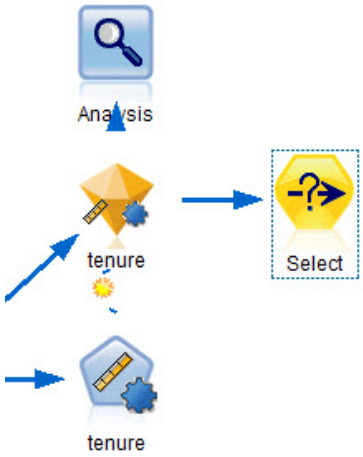
As with the evaluation of the Auto Classifier model for *churn*, to go further we need to select the records in the testing partition to do additional model evaluation. However, rather than recreate the specifications in the Select node, we can copy that node from the stream file saved in that earlier lesson.

1) Select **File…Recent Streams…Customer_Offers_Model Evaluation.str**
2) Right-click on the Select node and select **Copy Node**
3) Select the **Streams** tab in the Manager area in the upper right
4) Select **Customer_Offers_Tenure**
5) Right-click in the stream near the tenure model nugget and select **Paste**

The Select node is pasted into this stream.

6) **Connect** the tenure model nugget to the Select node

Figure 11.13 Select Node Added to Stream



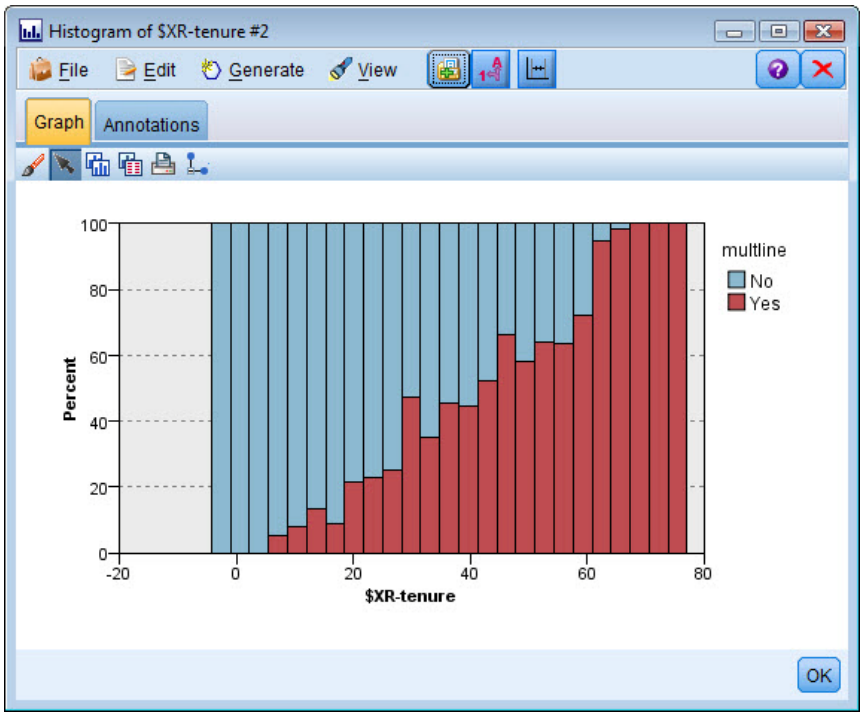
We can now continue with model evaluation.

11.8 *Model Predictions for Categorical Fields*

When the target is continuous, an effective method to investigate the relationship between a categorical input and the target is a histogram with the categorical field used as an overlay. One of the important predictors was *multline*, measuring whether or not a customer had multiline service at their home. We'll see how this relates to predicted tenure.

1) Add a **Histogram** node from the Graphs palette to the stream near the Select node
2) **Connect** the Select node to the Histogram node
3) Edit the Histogram node
4) Select **$XR-tenure** as the Field
5) Select **multline** as the Color Overlay field (not shown)
6) Select the **Options** tab
7) Select **Normalize by color** check box
8) Select **Run**

Figure 11.14 Histogram of Predicted Tenure by multline



There is a very clear and definite relationship between time as a customer and having multiline service. Those customers with multiline service are much more likely to have longer tenure. Moreover, there is almost a linear relationship between the two: as tenure increases, the percent of customers who are multiline also increases.

Note that some predicted values of *tenure* are below 0, and some values are above 72. This is a typical result of models that predict continuous fields, as predictions can regularly occur outside the actual range of the data. When this happens, there are several methods to adjust the predictions. The simplest is to use a Type node to set the range (here 0 to 72) and then coerce the values that are lower or higher to the endpoints of the range.

Further Info

In fact, multiline customers have a tenure 20 months greater, on the average, than non-multiline customers (we used the Means node to make this calculation).

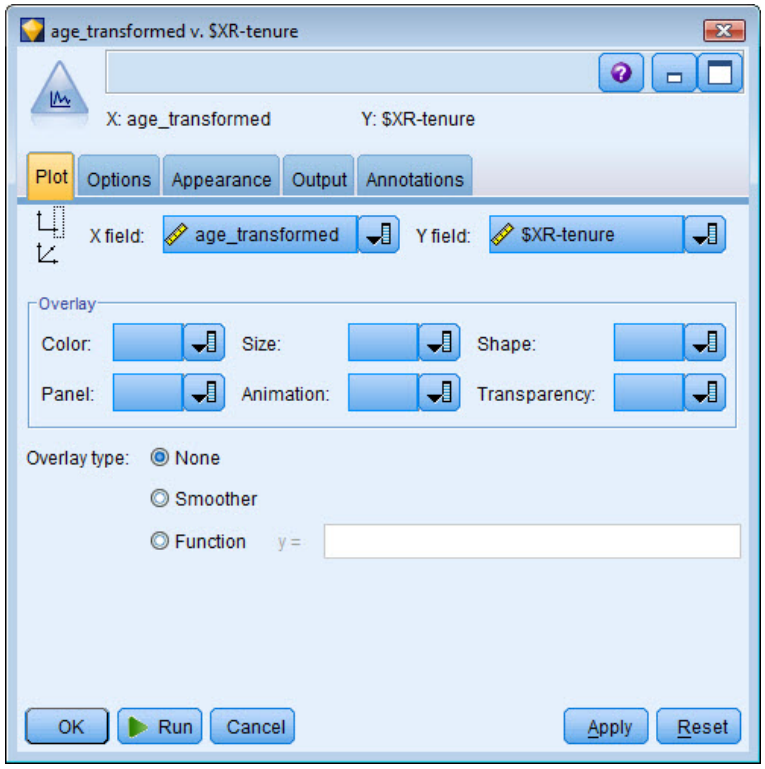
11.9 *Model Predictions for Continuous Fields*

When the target and input are both continuous, an effective method to examine the relationship between them is a scatterplot. One of the important predictors was *age_transformed*, so we'll use a scatterplot to see how it is associated with predicted tenure.

1) Add a **Plot** node from the Graphs palette to the stream near the Select node
2) **Connect** the Select node to the Plot node
3) Edit the Plot node

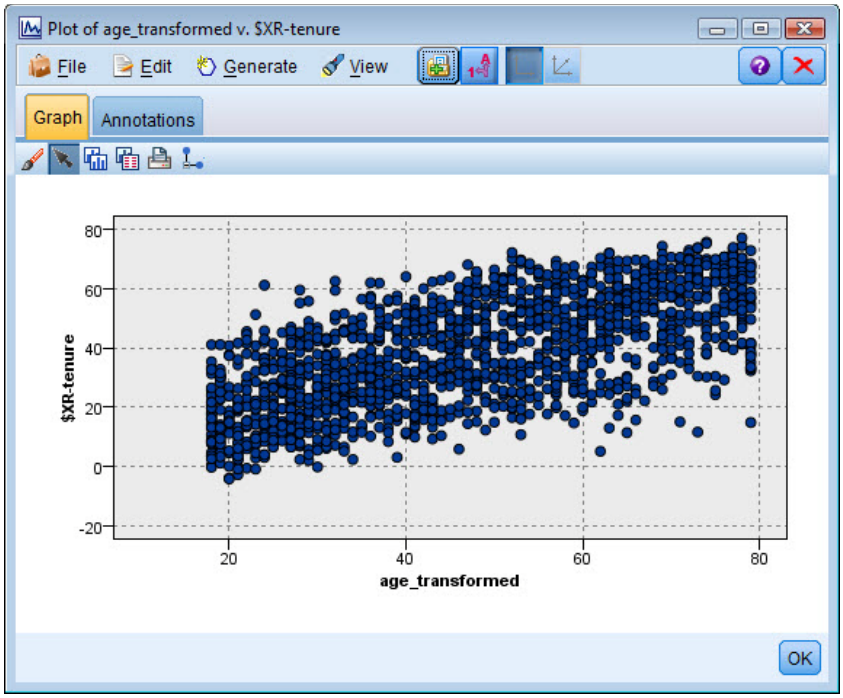
4) Select **age_transformed** as the X field
5) Select **$XR-tenure** as the Y field

Figure 11.15 Plot Node Specifications



6) Select **Run**

Figure 11.16 Scatterplot of age_transformed and Predicted tenure



There is a strong linear relationship between *age_transformed* and *$XR-tenure*. As one increases, so does the other. Therefore, the older a customer when he or she began their service, the longer their tenure.

Model evaluation can continue in this manner, exploring how important predictors are related to the target predictions.

Best Practice

The data for this study were collected by following a large group of customers who all began their service at approximately the same time. This is the best method to build models of churn or customer lifetime. An alternative statistical method to analyze such data is Cox Regression, which is included in Modeler but is beyond the scope of this class.

In the interests of time we will terminate any further model evaluation, but in practice we would spend a significant amount of time at model evaluation and understanding.

We will save the stream at this point.

1) Select **OK**
2) Select **File…Save Stream As**
3) Name the stream **Customer_Offers_Auto Numeric.str**
4) Select the **Save** button

Apply Your Knowledge

1) Which of these measures can the Auto Numeric node use to rank models? Select all that apply.
    a. Overall accuracy
    b. Correlation
    c. Lift
    d. Relative error
    e. Standard deviation

2) True or false? A model for a continuous target can predict outside the range of the original field.

11.10 *Lesson Summary*

In this lesson we demonstrated how to develop a model for a continuous target, and then how to evaluate that model.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Use the Auto Numeric node to create an ensemble model to predict a continuous target

To support the achievement of the primary objective, students should now also be able to:

- Describe and use the features and settings of the Auto Numeric node
- Describe and use the components of the model output from the Auto Numeric node
- Use various nodes for model evaluation

11.11 Learning Activity

The overall goal of this learning activity is to develop a model to predict total spending by the respondent, recorded in the field *totspend*.

 **Supporting Materials**	The Modeler stream file *Lesson 11 Exercise.str*, created beforehand.

1. To save time, as we did in the lesson, a stream file has already been created that prepares the data. Open the stream file *Lesson 11 Exercise.str*.

2. Review the stream to see what changes were made compared to the stream that predicted *response*.

3. Add an Auto Numeric node to the stream and attach to the Filter node named “Fields to Predict Spending.” Run the Auto Numeric node.

4. Evaluate the model on the training data. What are the top three models? What is their correlation to *totspend?* What is their relative error?

5. In the overall model, which fields are most important? Is there one field that is greatly important than others? In the scatterplot of actual and predicted total spending, does the relationship appear linear? What values of *totspend* are more difficult to predict?

6. In the testing data, how does the model performance change? Does this seem adequate?

7. Use an Analysis node to evaluate the Auto Numeric model. What is the absolute mean error on the training data? The testing data?

8. Add a Select node to the stream and select out the testing partition.

9. There are several categorical inputs in the stream. Use a Histogram node to examine the relationship between these and predicted *totspend*. Describe the relationships.

10. There are several continuous predictors. Use a Plot node to examine the relationship between them and predicted *totspend*. Are the relationships positive or negative? Are they linear or not?

11. Save the stream as *Lesson 11 Exercise Complete.str*.

Lesson 12: Deploying Models

12.1 *Objectives*

After completing this lesson students will be able to:

- Use a model to score new data

To support the achievement of this primary objective, students will also be able to:

- Describe what needs to be modified to create a scoring stream for new data
- Describe the deployment options in Modeler
- Export scored data to another file format

12.2 *Introduction*

After a model has been developed, it must be applied to new data. The task of doing so is called *deploying* the model. The task of assigning a prediction to a new record is called *scoring* the model. So we deploy a model to score new data.

Modeler has been designed to make deployment a straightforward task that can be done within its standard environment. The output from streams—a data file—can also be written out to other file formats for use by other applications. There are also optional methods to deploy streams that offer the ability to use the features of the Modeler stream, and model, but work outside the Modeler environment.

In this lesson, we focus on what can be done directly within Modeler. We will use a variant of the stream with an Auto Classifier model to predict customers who churned from the telecommunications firm.

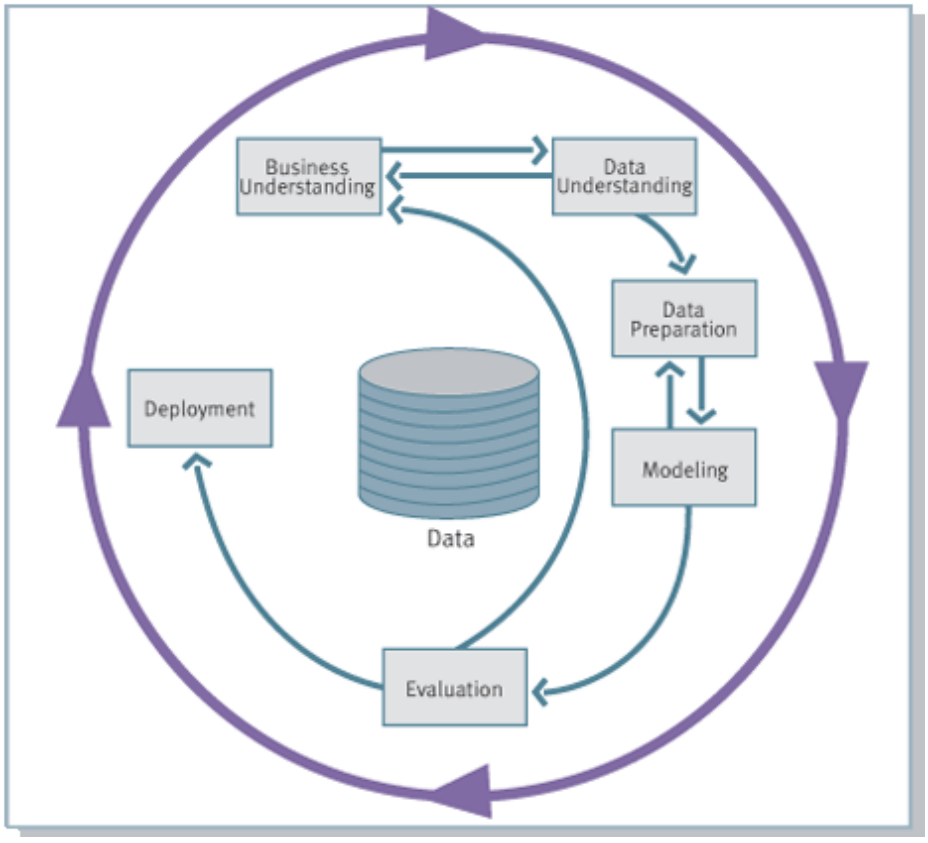
	The *customer_offers_new.sav* Statistics data file. These data are from customers of a telecommunications company. The data includes both demographic and account-related information.
Supporting Materials	The Modeler stream file *Customer_Offers_Scoring.str*, which contains the results of the model predicting churn and has been modified for scoring new data.

12.3 *The Deployment Phase*

The final major stage of the CRISP-DM data-mining process is deployment, as illustrated by the figure below. Deployment takes a stream that prepares and models data and applies it to new data. It then feeds that information back into the business process for decision making. Deployment can be:

- Based on a single customer (record): The focus can be on what offer to make or action to take for a new customer. Data for just that customer could be fed into the stream to generate a prediction, in real-time
- Based on a group of customers (records): Often called “batch scoring,” the model can be deployed on many customers to score all of them at once. This is often done in some regular cycle—daily, weekly, etc.—and the scored records added to databases or data warehouses.

Figure 12.1 Deployment as End Stage of CRISP-DM Process



Note that the Deployment phase includes a Monitoring activity. The performance of a model must be monitored on a regular basis to insure that it still meets the targets set in the CRISP-DM process.

If and when the model performs poorly (and no model lasts forever because of the changing competitive environment as well as new economic conditions), the model must be “refreshed” or reconstructed. This process should take much less time than the initial model development, although it is worth some thought about what new data could be collected, and the data should be audited again to look for differences in distributions, missing data, or other features that could affect model development.

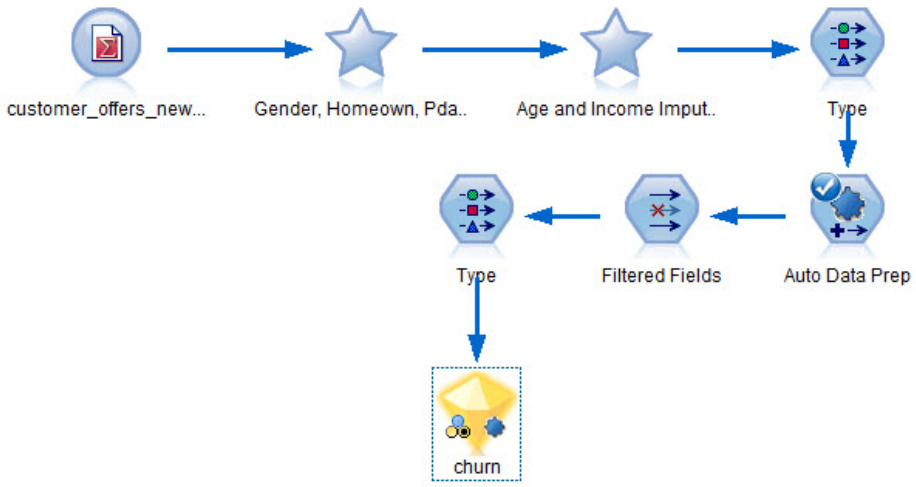
12.4 *Deploying a Model*

The models that Modeler generates are added automatically to a stream and used to make predictions on data, especially on the testing partition not used to develop the model. But models just as easily can be used in a similar stream to make predictions for new data. This is the essence of deploying a stream and scoring new data. Simply use the same stream, with only the essential nodes, and apply it to new data. Streams to score new data don't have training and testing data partitions because there will be no model validation. And that is because there is no target field to check against the model's predictions.

To illustrate this process, we have modified the *Customer_Offers_Auto Classifier.str* constructed to predict customer churn. The new stream is *Customer_Offers_Scoring.str*. We'll open it and review its features.

1) Open the stream file **Customer_Offers_Scoring.str**

Figure 12.2 Customer_Offers_Scoring.str Stream File



Compared to the stream we constructed to model *churn*, this stream:

1. Removes the Data Audit node used to create the SuperNodes
2. Removes the Partition node
3. Removes the Feature Selection node and the Feature selection model used to generate the Filter Fields node
4. Removes the Auto Classifier node used to create the Auto Classifier model for *churn*
5. Removes all nodes used to review or display the data

The result is a smaller stream, with less of a data processing burden. The stream:

- Accesses the new data file and sets blank definitions
- Substitutes for missing data
- Coerces data to the correct range
- Transforms continuous fields
- Filters the fields
- Instantiates the new fields
- Runs the data into the model for *churn* to make predictions

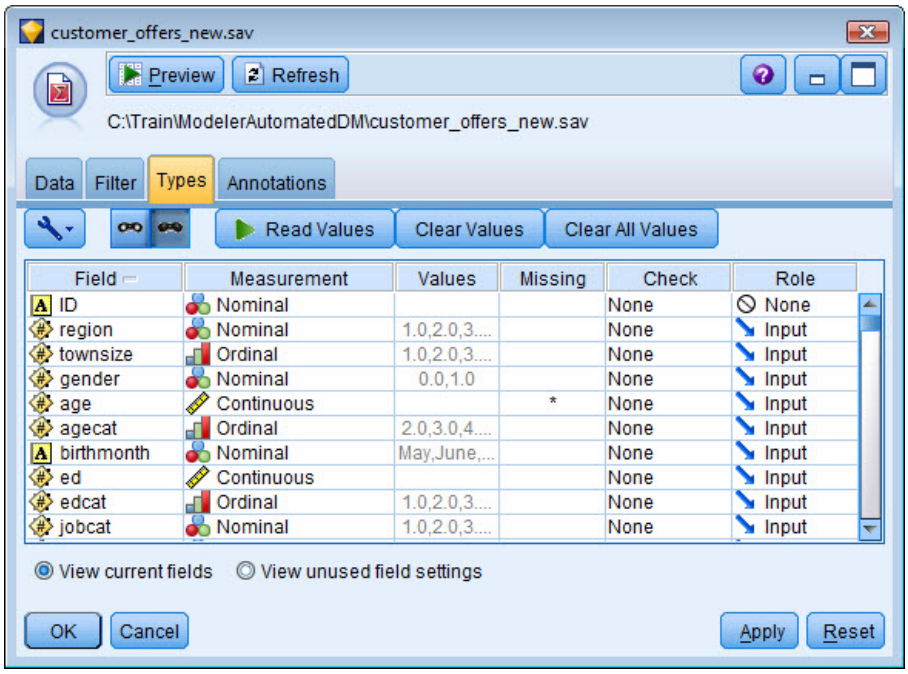
To successfully use new data in the stream, the source node must have the same fields as before, or at least the same ones used downstream in data preparation and modeling. Obviously, if a necessary node is missing, then one or more nodes may not execute.

The new data file, *customer_offers_new.sav*, does not, of course, have a field named *churn* because that is what we are trying to predict. This will not cause Modeler any difficulty. For example, the Type nodes will automatically adjust to the fact that a field is missing from the data stream. The nodes will also not report an error because there is no *partition* field in the data.

It is very important that the same fields in the new data have the same field definitions as the original data used for modeling. Let's examine the source node to illustrate this point.

1) Edit the Source node **customer_offers_new.sav**

Figure 12.3 Type Tab in customer_offers_new.sav Source Node



Recall that in an earlier lesson we defined the value of 99 as a blank (missing value) for *age*. We made the same assignment in this Source node.

If new fields have been added to the data source, we can use the Filter tab in the Source node to remove them from the data stream. Otherwise, they could be a problem for downstream nodes.

It is usually best to instantiate the new data before using it for scoring. We can do that in the Source node.

2) Select **Read Values** button
3) Select **OK** in the Read Values dialog box (not shown)

The data are read, and the Measurement column now lists the categories, or range, for each field, depending on its type.

Data Range in New Data

Ideally, when developing a model, we want to include the full possible range of data for each field as there could be in future data. However, this is not always possible, and so one question that often arises in model scoring is whether the range or categories in the new data must match the old data. The first Type node downstream is used to coerce out-of-range values for *age* and *income*. That's fine, but what happens if there is a value beyond the current range for *cardtenure*, which was an important field in predicting *churn*?

The answer depends on whether the field is categorical or continuous.

Continuous fields. Usually a model will still make a prediction without any problems. Usually low or high values of a continuous field are used in an equation, the equivalent of an equation, or a set of rules where such values will be grouped with other low, or high values. Thus, no problems are encountered.

Now, a value far out of range could still be one that we don't wish to use for modeling, e.g., a very high number of long distance minutes. The model wasn't developed with such a value, and so it might make poor predictions. We can use a Type node to coerce values to the existing range for any continuous fields that have out-of-range values. Alternatively, we can have the Type node discard records with out-of-range values, or set them to missing.

Categorical fields. Here, values not encountered when building a model can cause problems. Consider the field *card*, recording type of credit card. If a new credit card not available before is now available, that value will not be part of the defined values for *card*. Even if the model doesn't fail when attempting to score the data, it certainly won't use this new value correctly.

If there are only a few records with values not used in modeling, they can be deleted from the stream, set to missing, or something else suitable. But if there are enough records so that they cannot be ignored, and if this field is important for modeling, then it may be necessary to develop a new model. And this makes sense, as this situation is just one variant on the rule that eventually, all models will fail or be inapplicable to new data, and so must be refreshed.

Scoring the Model

There is nothing special to do when scoring the model in a stream that has been modified as we have been discussing. A Table node can be used to display predictions for each record, or another node can be used to display the predictions. Usually, there is less interest in the overall grouped predictions compared to predictions for specific records (customers).

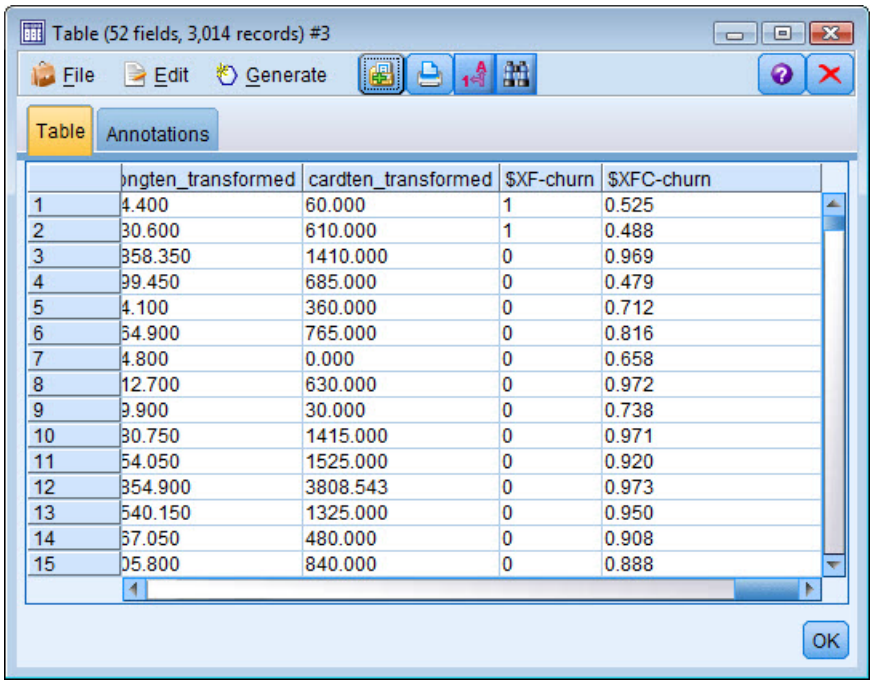
Note

It is true, though, that it can be a good idea to review the overall percentage of predictions for new data in each category of the target to see if it matches what is expected. For example, if almost all the customers were predicted to churn, that would probably be a sign that there is something wrong with the data.

We'll review the predictions by record.

1) Close the Type node by selecting **OK**
2) Add a **Table** node to the stream close to the model for churn
3) **Connect** the churn model to the Table node
4) Run the **Table** node

Figure 12.4 Predictions for churn for New Records



	ngten_transformed	cardten_transformed	$XF-churn	$XFC-churn
1	4.400	60.000	1	0.525
2	30.600	610.000	1	0.488
3	358.350	1410.000	0	0.969
4	99.450	685.000	0	0.479
5	4.100	360.000	0	0.712
6	54.900	765.000	0	0.816
7	4.800	0.000	0	0.658
8	12.700	630.000	0	0.972
9	9.900	30.000	0	0.738
10	30.750	1415.000	0	0.971
11	54.050	1525.000	0	0.920
12	354.900	3808.543	0	0.973
13	540.150	1325.000	0	0.950
14	67.050	480.000	0	0.908
15	05.800	840.000	0	0.888

The usual two fields have been added to the data by the Auto Classifier model. The difference here is that there is no target field to compare to the prediction (this means we cannot use the Analysis node, for example). We use the prediction in the field *$XF-churn*, and we could sort the predictions by their confidence (*$XFC-churn*) for direct marketing applications.

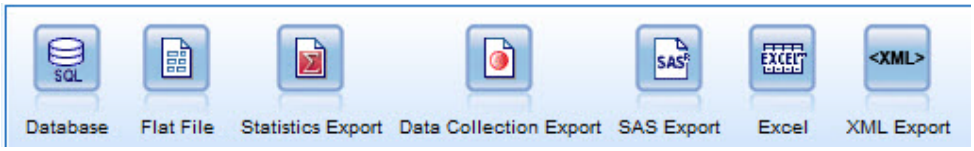
5) Close the Table window

12.5 *Exporting Model Results*

Once predictions have been made on new data, we will need to use them in some fashion. Typically, this means we will want to export the results as a data file to another format, or even write the results directly into a database. Modeler has several types of file exports.

1) Click the **Export** tab in the Palettes area

Figure 12.5 Export Nodes in Modeler



One of these nodes can be attached to a generated model node and used to write data to an external file of that type.

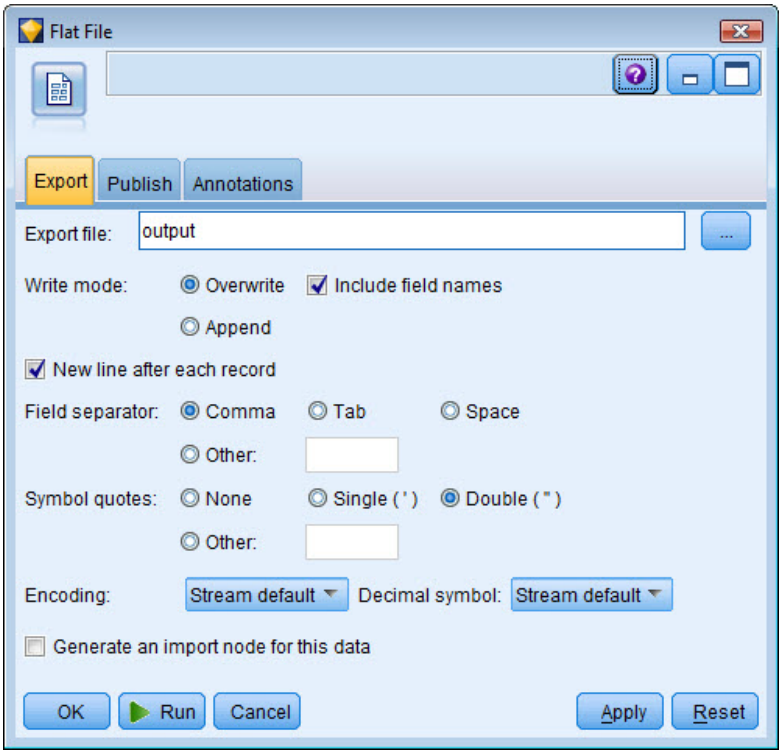
- The Database export node writes data directly into an ODBC-compliant relational data source. This option allows the user to write data into an existing database so predictions can be added to other data.

- The Flat File export node writes data to a delimited text file (with data values separated by commas, tabs, or other characters).
- The Statistics Export node writes data in Statistics file format (.sav). This file can be read by Statistics and other SPSS, an IBM Company products.
- The Excel export node saves data in Microsoft Excel format (.xls). A choice is available to launch Excel automatically and open the exported file when the node is run.
- The XML Export node, new to Modeler 14.0, enables the user to output data in XML format, using UTF-8 encoding.

We'll illustrate the process by writing data to a text file.

2) Add a **Flat File** node to the stream by the model for churn
3) **Connect** the churn model to the Flat File node
4) Edit the **Flat File** node

Figure 12.6 Flat File Node



We should change the file name and location. Otherwise, for this demonstration, we can use the defaults.

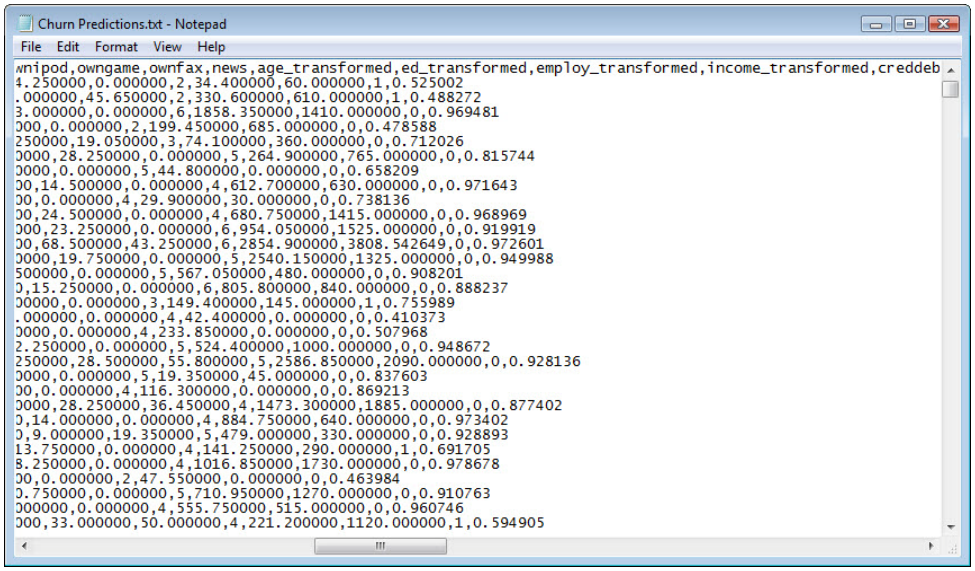
5) Change the Export file: name to **Churn Predictions.txt**
6) Change the file location to **c:\Train\Modeler_AutomatedDM** (not shown)
7) Select **Run**

Further Information

In an actual application, if the other fields aren't needed, you might place a Filter node between the model and Export node and only write out the prediction and its confidence.

If we opened the file *Churn Predictions.txt* in a text editor, we would see the data as in the next figure. The data values are separated by commas.

Figure 12.7 Churn Predictions.txt File Exported



In the future, when more new data need to be scored, the data source node in the stream can be updated, or if the new data are always written to the same data file name (here *customer_offers_new.sav*), no other action need be taken to score those data beyond running the full scoring stream.

We don't need to save the stream file in this lesson.

12.6 *Other Deployment Options*

There are several other more technically sophisticated options available to deploy Modeler streams and models. They require optional software and a server version of Modeler.

For example, the IBM® SPSS® Collaboration and Deployment Services (C&DS) is a platform for the management and deployment of analytical assets in the full enterprise. It provides extensive support as predictive models are prepared, validated and integrated with operational systems to support decision-making within an organization. It enables the organization to manage the life cycle of data mining models and allows these objects to be used by enterprise applications, tools, and solutions.

Deployment can be managed with C&DS, including real-time scoring:

- It can deliver analytical results as customer interactions are occurring through integration with business user systems

- It can be optimized for real time use so that information gathered during the time of the interaction and historical data can determine the score
- It can be integrated with existing applications using standard programming interfaces, such as Web Services

Streams deployed in C&DS are saved as *scenarios*, with the extension *.scn*.

Another deployment option is to use Modeler streams in conjunction with the thin-client application IBM® SPSS® Modeler Advantage. While it is possible to create customized applications entirely within Modeler Advantage, we can also use a stream already created in Modeler as the basis of an application workflow to score new data.

Apply Your Knowledge

1) True or False? New data to be scored must have exactly the same range for continuous input fields.

2) True or False? You can use a model nugget "as is" to make predictions for new data.

12.7 *Lesson Summary*

In this lesson we demonstrated how to deploy a model and score new data.

Lesson Objectives Review

Students who have completed this lesson should now be able to:

- Use a model to score new data

To support the achievement of the primary objective, students should now also be able to:

- Describe what needs to be modified to create a scoring stream for new data
- Describe the deployment options in Modeler
- Export scored data to another file format

12.8 Learning Activity

The overall goal of this learning activity is to use a scoring stream to make predictions for new supporters of the charity for a similar fundraising campaign.

	The new Statistics data file *Charity_New.sav* that contains new data on the charity's supporters to predict *response*.
Supporting Materials	The Modeler stream file *Lesson 12 Exercise.str*, created beforehand.

1. The data file *Charity_New.sav* contains information on new supporters of the charity. All fields are included except *response*. We need to create a stream to score these new data with our existing Auto Classifier model for *response*.
    a. If you feel ambitious, you can attempt to modify the existing stream created in the exercises for Lesson 9 to use this new data to make a prediction for *response*. If you do, delete any extraneous nodes to make the stream as simple as possible.
    b. If you feel less ambitious or have less time, the stream *Lesson 12 Exercise.str* contains the modified stream to use for scoring. You can use it to check your work if you do try to modify the existing stream yourself

2. In either case, score the new data in *Charity_New.sav*. What percentage of supporters are predicted to have a "Yes" response to a similar campaign?

3. Export the predictions to a text file. Filter the fields first so only the prediction and its confidence are included.

Lesson 13: Course Summary

13.1 *Course Objectives Review*

Now that you have completed the course, you should be able to:

- Use Modeler to perform an automated data mining project

And, you should also be able to:

- Understand the principles of data mining
- Use the user interface of Modeler to create basic program streams
- Read a Statistics data file into Modeler and define data characteristics
- Review and explore data to look at data distributions and to identify data problems, including missing values
- Use the Automated Data Prep node to further prepare data for modeling
- Use a Partition node to create training and testing data subsets
- Use the Feature Selection node to select inputs for modeling
- Use the Auto Classifier node to create an ensemble model to predict a categorical target
- Evaluate and understand the predictions of a model
- Use the Auto Numeric node to create an ensemble model to predict a continuous target
- Use a model to score new data

13.2 *Course Review: Discussion Questions*

1. What is the difference between traditional statistical analysis and data mining?
2. What are the features of data that are important to review during data exploration?
3. What types of actions are commonly taken to prepare data for modeling?
4. What are the options to handle missing data in Modeler?
5. Is accuracy the best criterion on which to evaluate a model? What other criteria might you use?

13.3 *Next Steps*

Thought Starters

How can you immediately apply what you have learned about using Modeler for automated data mining to your organization's own predictive analytic needs?

List your top three steps or actions:

1._____

2._____

3._____

Next Courses

This course discussed many topics and methods, but there is much left to learn about Modeler. In this section we provide direction for what courses you can attend to broaden your knowledge in specific areas.

If you want to learn more about this:	Take this course:
More nodes and capabilities of Modeler	Introduction to IBM SPSS Modeler and Data Mining
Advanced data preparation	Advanced Data Preparation with IBM SPSS Modeler
Advanced predictive models	Predictive Modeling with IBM SPSS Modeler
Cluster and association models	Clustering and Association Models with IBM SPSS Modeler