IBM SPSS Modeler 15 Applications Guide



Note: Before using this information and the product it supports, read the general information under Notices on p. 385.

This edition applies to IBM SPSS Modeler 15 and to all subsequent releases and modifications until otherwise indicated in new editions.

Adobe product screenshot(s) reprinted with permission from Adobe Systems Incorporated.

Microsoft product screenshot(s) reprinted with permission from Microsoft Corporation.

Licensed Materials - Property of IBM

© Copyright IBM Corporation 1994, 2012.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Preface

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

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

About IBM Business Analytics

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

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

Technical support

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

Contents

1 About IBM SPSS Modeler

IBM SPSS Modeler Products	1
IBM SPSS Modeler	1
IBM SPSS Modeler Server	2
IBM SPSS Modeler Administration Console	
IBM SPSS Modeler Batch	2
IBM SPSS Modeler Solution Publisher	2
IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services .	
IBM SPSS Modeler Editions	3
IBM SPSS Modeler Documentation	4
SPSS Modeler Professional Documentation	
SPSS Modeler Premium Documentation	
Application Examples	
Demos Folder	6

Part I: Introduction and Getting Started

2 IBM SPSS Modeler Overview

Getting Started
Starting IBM SPSS Modeler
Launching from the Command Line
Connecting to IBM SPSS Modeler Server
Changing the Temp Directory
Starting Multiple IBM SPSS Modeler Sessions
IBM SPSS Modeler Interface at a Glance
IBM SPSS Modeler Stream Canvas
Nodes Palette
IBM SPSS Modeler Managers
IBM SPSS Modeler Projects
IBM SPSS Modeler Toolbar
Customizing the Toolbar
Customizing the IBM SPSS Modeler Window
Changing the icon size for a stream
Using the Mouse in IBM SPSS Modeler22
Using Shortcut Keys
Printing
Automating IBM SPSS Modeler

3 Introduction to Modeling

Building the Stream	27
Browsing the Model	33
Evaluating the Model	37
Scoring Records	40
Summary	41

4 Automated Modeling for a Flag Target

Modeling Customer Response (Auto Classifier).	42
Historical Data	42
Building the Stream	43
Generating and Comparing Models	48
Summary	53

5 Automated Modeling for a Continuous Target

Property Values (Auto Numeric)	54
Training Data	54
Building the Stream	55
Comparing the Models	59
Summary	61

Part II: Data Preparation Examples

6	Automated Data Preparation (ADP)	63	
	Building the Stream	63	
	Comparing Model Accuracy	69	
7	Preparing Data for Analysis (Data Audit)	72	
	Building the Stream	72	

v

42

Browsing Statistics and Charts	76
Handling Outliers and Missing Values	79

8 Drug Treatments (Exploratory Graphs/C5.0)

Reading in Text Data	84
Adding a Table	88
Creating a Distribution Graph	89
Creating a Scatterplot	91
Creating a Web Graph	93
Deriving a New Field	94
Building a Model	97
Browsing the Model	99
Using an Analysis Node	01

9 Screening Predictors (Feature Selection)

Building the Stream	104
Building the Models	106
Comparing the Results	107
Summary 1	109

10 Reducing Input Data String Length (Reclassify Node) 110

Reducing Input Data String Length (Reclassify)	• `	110
Reclassifying the Data	. '	110

Part III: Modeling Examples

11	Modeling	Customer	Response	(Decision List)) 116
----	----------	----------	----------	-----------------	-------

Historical Data	17
Building the Stream	18
Creating the Model	21

Calculating Custom Measures Using Excel	134
Modifying the Excel template	140
Saving the Results	143

12 Classifying Telecommunications Customers (Multinomial Logistic Regression) 144

Building the Stream	145
Browsing the Model	149

13 Telecommunications Churn (Binomial Logistic Regression) 154

Building the Stream	154
Browsing the Model	162

14 Forecasting Bandwidth Utilization (Time Series)168

Forecasting with the Time Series Node
Creating the Stream
Examining the Data
Defining the Dates
Defining the Targets
Setting the Time Intervals
Creating the Model
Examining the Model
Summary
Reapplying a Time Series Model
Retrieving the Stream
Retrieving the Saved Model
Generating a Modeling Node
Generating a New Model
Examining the New Model
Summary

15	Forecasting Catalog Sales (Time Series)	199
	Creating the Stream	
	Examining the Data	
	Exponential Smoothing	
	ARIMA	
16	Making Offers to Customers (Self-Learning)	215
	Building the Stream	216
	Browsing the Model	222
17	Predicting Loan Defaulters (Bayesian Network)	227
	Building the Stream	
	Browsing the Model	232
18	Retraining a Model on a Monthly Basis (Bayesian Net	work)237
	Building the Stream	
	Evaluating the Model	241
19	Retail Sales Promotion (Neural Net/C&RT)	249
	Examining the Data	249
	Learning and Testing	252
<i>20</i>	Condition Monitoring (Neural Net/C5.0)	254
	Examining the Data	
	Data Preparation	
	Learning	
	Testing	
	viii	

21 Classifying Telecommunications Customers (Discriminant Analysis) 260

Creating the Stream
Examining the Model
Stepwise Discriminant Analysis
A Note of Caution Concerning Stepwise Methods
Checking Model Fit
Structure Matrix
Territorial Map
Classification Results
Summary

22 Analyzing Interval-Censored Survival Data (Generalized Linear Models) 272

Creating the Stream	2
Tests of Model Effects	8
Fitting the Treatment-Only Model	8
Parameter Estimates	0
Predicted Recurrence and Survival Probabilities	1
Modeling the Recurrence Probability by Period	5
Tests of Model Effects	1
Fitting the Reduced Model	1
Parameter Estimates	3
Predicted Recurrence and Survival Probabilities	4
Summary	8

23 Using Poisson Regression to Analyze Ship Damage Rates (Generalized Linear Models) 300

Fitting an "Overdispersed" Poisson Regression	300
Goodness-of-Fit Statistics.	305
Omnibus Test	305
Tests of Model Effects	306
Parameter Estimates	306
Fitting Alternative Models	307

Goodness-of-Fit Statistics	310
Summary	311

24 Fitting a Gamma Regression to Car Insurance Claims (Generalized Linear Models)

Creating the Stream	312
Parameter Estimates	316
Summary	316

25 Classifying Cell Samples (SVM)

Creating the Stream	318
Examining the Data	323
Trying a Different Function	325
Comparing the Results	327
Summary	328

26 Using Cox Regression to Model Customer Time to Churn 329

Building a Suitable Model	329
Censored Cases	333
Categorical Variable Codings	
Variable Selection	
Covariate Means	
Survival Curve	
Hazard Curve	
Evaluation	341
Tracking the Expected Number of Customers Retained	346
Scoring	360
Summary	365

27 Market Basket Analysis (Rule Induction/C5.0)

Accessing the Data	366
Discovering Affinities in Basket Contents	. <mark>368</mark>

317

312

Profiling the Customer Groups	371
Summary	372

373

28 Assessing New Vehicle Offerings (KNN)

Creating the Stream	374
Examining the Output	379
Predictor Space	380
Peers Chart	381
Neighbor and Distance Table	384
Summary	384

Appendix

A Notices	Notices	385
	Bibliography	388
	Index	389

About IBM SPSS Modeler

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

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

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

IBM SPSS Modeler Products

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

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- IBM SPSS Modeler Server adapters for IBM SPSS Collaboration and Deployment Services

IBM SPSS Modeler

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

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

SPSS Modeler is available in two editions: SPSS Modeler Professional and SPSS Modeler Premium. For more information, see the topic IBM SPSS Modeler Editions on p. 3.

IBM SPSS Modeler Server

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

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

IBM SPSS Modeler Administration Console

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

IBM SPSS Modeler Batch

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

IBM SPSS Modeler Solution Publisher

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

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

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

IBM SPSS Modeler Editions

SPSS Modeler is available in the following editions.

SPSS Modeler Professional

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

SPSS Modeler Premium

SPSS Modeler Premium is a separately-licensed product that extends SPSS Modeler Professional to work with specialized data such as that used for entity analytics or social networking, and with unstructured text data. SPSS Modeler Premium comprises the following components.

IBM® SPSS® Modeler Entity Analytics adds a completely new dimension to IBM® SPSS® Modeler predictive analytics. Whereas predictive analytics attempts to predict future behavior from past data, entity analytics focuses on improving the coherence and consistency of current data by resolving identity conflicts within the records themselves. An identity can be that of an individual, an organization, an object, or any other entity for which ambiguity might exist. Identity resolution can be vital in a number of fields, including customer relationship management, fraud detection, anti-money laundering, and national and international security.

IBM SPSS Modeler Social Network Analysis transforms information about relationships into fields that characterize the social behavior of individuals and groups. Using data describing the relationships underlying social networks, IBM® SPSS® Modeler Social Network Analysis identifies social leaders who influence the behavior of others in the network. In addition, you can determine which people are most affected by other network participants. By combining these results with other measures, you can create comprehensive profiles of individuals on which to base your predictive models. Models that include this social information will perform better than models that do not.

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

IBM SPSS Modeler Documentation

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

Complete documentation for each product (including installation instructions) is available in PDF format under the \Documentation folder on each product DVD. Installation documents can also be downloaded from the web at http://www-01.ibm.com/support/docview.wss?uid=swg27023172.

Documentation in both formats is also available from the SPSS Modeler Information Center at *http://publib.boulder.ibm.com/infocenter/spssmodl/v15r0m0/.*

SPSS Modeler Professional Documentation

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

- IBM SPSS Modeler User's Guide. General introduction to using SPSS Modeler, including how to build data streams, handle missing values, build CLEM expressions, work with projects and reports, and package streams for deployment to IBM SPSS Collaboration and Deployment Services, Predictive Applications, or IBM SPSS Modeler Advantage.
- IBM SPSS Modeler Source, Process, and Output Nodes. Descriptions of all the nodes used to read, process, and output data in different formats. Effectively this means all nodes other than modeling nodes.
- IBM SPSS Modeler Modeling Nodes. Descriptions of all the nodes used to create data mining models. IBM® SPSS® Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics.
- **IBM SPSS Modeler Algorithms Guide.** Descriptions of the mathematical foundations of the modeling methods used in SPSS Modeler. This guide is available in PDF format only.
- IBM SPSS Modeler Applications Guide. The examples in this guide provide brief, targeted introductions to specific modeling methods and techniques. An online version of this guide is also available from the Help menu. For more information, see the topic Application Examples on p. 5.
- **IBM SPSS Modeler Scripting and Automation.** Information on automating the system through scripting, including the properties that can be used to manipulate nodes and streams.
- IBM SPSS Modeler Deployment Guide. Information on running SPSS Modeler streams and scenarios as steps in processing jobs under IBM® SPSS® Collaboration and Deployment Services Deployment Manager.
- **IBM SPSS Modeler CLEF Developer's Guide.** CLEF provides the ability to integrate third-party programs such as data processing routines or modeling algorithms as nodes in SPSS Modeler.
- IBM SPSS Modeler In-Database Mining Guide. Information on how to use the power of your database to improve performance and extend the range of analytical capabilities through third-party algorithms.
- IBM SPSS Modeler Server Administration and Performance Guide. Information on how to configure and administer IBM® SPSS® Modeler Server.

- IBM SPSS Modeler Administration Console User Guide. Information on installing and using the console user interface for monitoring and configuring SPSS Modeler Server. The console is implemented as a plug-in to the Deployment Manager application.
- IBM SPSS Modeler Solution Publisher Guide. SPSS Modeler Solution Publisher is an add-on component that enables organizations to publish streams for use outside of the standard SPSS Modeler environment.
- **IBM SPSS Modeler CRISP-DM Guide.** Step-by-step guide to using the CRISP-DM methodology for data mining with SPSS Modeler.
- IBM SPSS Modeler Batch User's Guide. Complete guide to using IBM SPSS Modeler in batch mode, including details of batch mode execution and command-line arguments. This guide is available in PDF format only.

SPSS Modeler Premium Documentation

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

- IBM SPSS Modeler Entity Analytics User Guide. Information on using entity analytics with SPSS Modeler, covering repository installation and configuration, entity analytics nodes, and administrative tasks.
- IBM SPSS Modeler Social Network Analysis User Guide. A guide to performing social network analysis with SPSS Modeler, including group analysis and diffusion analysis.
- SPSS Modeler Text Analytics User's Guide. Information on using text analytics with SPSS Modeler, covering the text mining nodes, interactive workbench, templates, and other resources.
- IBM SPSS Modeler Text Analytics Administration Console User Guide. Information on installing and using the console user interface for monitoring and configuring IBM® SPSS® Modeler Server for use with SPSS Modeler Text Analytics. The console is implemented as a plug-in to the Deployment Manager application.

Application Examples

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

You can access the examples by clicking Application Examples on the Help menu in SPSS Modeler. The data files and sample streams are installed in the *Demos* folder under the product installation directory. For more information, see the topic Demos Folder on p. 6.

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

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

Demos Folder

The data files and sample streams used with the application examples are installed in the *Demos* folder under the product installation directory. This folder can also be accessed from the IBM SPSS Modeler 15 program group on the Windows Start menu, or by clicking *Demos* on the list of recent directories in the File Open dialog box.

Figure 1-1

Selecting the Demos folder from the list of recently-used directories

📀 Open		S
Look In: 🛅 I	BMModeler14.1	◆ ← ← 👔 🚮 🔯 🧱 🗄 ◯ Installation Directory
Accessibili	ιy	📀 Demos
config		
🚞 Demos		
📄 DTD		
🚞 ext		
🚞 Help		
ire 🔁 jre		
📄 lib		
🚞 libServer		
icenses 🚞		
🚞 Media		
🚞 scripts		
	[
File <u>N</u> ame:		
Files of Type:	Stream Files (*.str)	•
		Open Insert Cancel

Part I: Introduction and Getting Started

IBM SPSS Modeler Overview

Getting Started

As a data mining application, IBM® SPSS® Modeler offers a strategic approach to finding useful relationships in large data sets. In contrast to more traditional statistical methods, you do not necessarily need to know what you are looking for when you start. You can explore your data, fitting different models and investigating different relationships, until you find useful information.

Starting IBM SPSS Modeler

To start the application, click: Start > [All] Programs > IBM SPSS Modeler15 > IBM SPSS Modeler15

The main window is displayed after a few seconds.

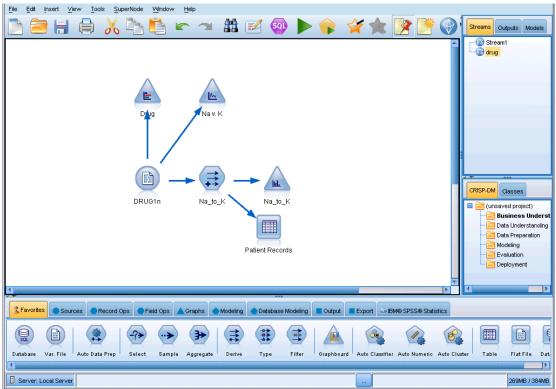


Figure 2-1 IBM SPSS Modeler main application window

© Copyright IBM Corporation 1994, 2012.

9

Launching from the Command Line

You can use the command line of your operating system to launch IBM® SPSS® Modeler as follows:

- On a computer where IBM® SPSS® Modeler is installed, open a DOS, or command-prompt, window.
- ► To launch the SPSS Modeler interface in interactive mode, type the modelerclient command followed by the required arguments; for example:

modelerclient -stream report.str -execute

The available arguments (flags) allow you to connect to a server, load streams, run scripts, or specify other parameters as needed.

Connecting to IBM SPSS Modeler Server

IBM® SPSS® Modeler can be run as a standalone application, or as a client connected to IBM® SPSS® Modeler Server directly or to an SPSS Modeler Server or server cluster through the Coordinator of Processes plug-in from IBM® SPSS® Collaboration and Deployment Services. The current connection status is displayed at the bottom left of the SPSS Modeler window.

Whenever you want to connect to a server, you can manually enter the server name to which you want to connect or select a name that you have previously defined. However, if you have IBM SPSS Collaboration and Deployment Services, you can search through a list of servers or server clusters from the Server Login dialog box. The ability to browse through the Statistics services running on a network is made available through the Coordinator of Processes.

Figure 2-2
Server Login dialog box

Server Login 🛛 🛛 🗙							
			which you want as Default is use		ne table. By		
Default	Server	Name	Description	Port	Add		
	Local S	Server					
	localho	st		28049	Edit		
					X Delete		
					Search		
Default_dat							
👿 Set Cre	edentials						
User ID:		djones					
Password							
Domain:							
Cancel Help							

To Connect to a Server

- ► On the Tools menu, click Server Login. The Server Login dialog box opens. Alternatively, double-click the connection status area of the SPSS Modeler window.
- Using the dialog box, specify options to connect to the local server computer or select a connection from the table.
 - Click Add or Edit to add or edit a connection. For more information, see the topic Adding and Editing the IBM SPSS Modeler Server Connection on p. 10.
 - Click Search to access a server or server cluster in the Coordinator of Processes. For more information, see the topic Searching for Servers in IBM SPSS Collaboration and Deployment Services on p. 12.

Server table. This table contains the set of defined server connections. The table displays the default connection, server name, description, and port number. You can manually add a new connection, as well as select or search for an existing connection. To set a particular server as the default connection, select the check box in the Default column in the table for the connection.

Default data path. Specify a path used for data on the server computer. Click the ellipsis button (...) to browse to the required location.

Set Credentials. Leave this box unchecked to enable the **single sign-on** feature, which attempts to log you in to the server using your local computer username and password details. If single sign-on is not possible, or if you check this box to disable single sign-on (for example, to log in to an administrator account), the following fields are enabled for you to enter your credentials.

User ID. Enter the user name with which to log on to the server.

Password. Enter the password associated with the specified user name.

Domain. Specify the domain used to log on to the server. A domain name is required only when the server computer is in a different Windows domain than the client computer.

Click OK to complete the connection.

To Disconnect from a Server

- ► On the Tools menu, click Server Login. The Server Login dialog box opens. Alternatively, double-click the connection status area of the SPSS Modeler window.
- ▶ In the dialog box, select the Local Server and click OK.

Adding and Editing the IBM SPSS Modeler Server Connection

You can manually edit or add a server connection in the Server Login dialog box. By clicking Add, you can access an empty Add/Edit Server dialog box in which you can enter server connection details. By selecting an existing connection and clicking Edit in the Server Login dialog box, the Add/Edit Server dialog box opens with the details for that connection so that you can make any changes.

Note: You cannot edit a server connection that was added from IBM® SPSS® Collaboration and Deployment Services, since the name, port, and other details are defined in IBM SPSS Collaboration and Deployment Services.

Figure	2-3				
Server	Login	Add/Edit	Server	dialog	box

💟 Server Login: Add/Edit Server 🛛 🛛 🔀						
Server:	localhost 💌					
Port:	28049					
Description:						
Ensure secure connection (use SSL)						

To Add Server Connections

- ▶ On the Tools menu, click Server Login. The Server Login dialog box opens.
- ▶ In this dialog box, click Add. The Server Login Add/Edit Server dialog box opens.
- Enter the server connection details and click OK to save the connection and return to the Server Login dialog box.
 - Server. Specify an available server or select one from the list. The server computer can be identified by an alphanumeric name (for example, *myserver*) or an IP address assigned to the server computer (for example, 202.123.456.78).
 - Port. Give the port number on which the server is listening. If the default does not work, ask your system administrator for the correct port number.
 - **Description.** Enter an optional description for this server connection.
 - Ensure secure connection (use SSL). Specifies whether an SSL (Secure Sockets Layer) connection should be used. SSL is a commonly used protocol for securing data sent over a network. To use this feature, SSL must be enabled on the server hosting IBM® SPSS® Modeler Server. If necessary, contact your local administrator for details.

To Edit Server Connections

- ▶ On the Tools menu, click Server Login. The Server Login dialog box opens.
- ► In this dialog box, select the connection you want to edit and then click Edit. The Server Login Add/Edit Server dialog box opens.
- Change the server connection details and click OK to save the changes and return to the Server Login dialog box.

Searching for Servers in IBM SPSS Collaboration and Deployment Services

Instead of entering a server connection manually, you can select a server or server cluster available on the network through the Coordinator of Processes, available in IBM® SPSS® Collaboration and Deployment Services. A server cluster is a group of servers from which the Coordinator of Processes determines the server best suited to respond to a processing request.

Although you can manually add servers in the Server Login dialog box, searching for available servers lets you connect to servers without requiring that you know the correct server name and port number. This information is automatically provided. However, you still need the correct logon information, such as username, domain, and password.

Note: If you do not have access to the Coordinator of Processes capability, you can still manually enter the server name to which you want to connect or select a name that you have previously defined. For more information, see the topic Adding and Editing the IBM SPSS Modeler Server Connection on p. 10.





To search for servers and clusters

- ▶ On the Tools menu, click Server Login. The Server Login dialog box opens.
- ► In this dialog box, click Search to open the Search for Servers dialog box. If you are not logged on to IBM SPSS Collaboration and Deployment Services when you attempt to browse the Coordinator of Processes, you will be prompted to do so.
- Select the server or server cluster from the list.
- Click OK to close the dialog box and add this connection to the table in the Server Login dialog box.

Changing the Temp Directory

Some operations performed by IBM® SPSS® Modeler Server may require temporary files to be created. By default, IBM® SPSS® Modeler uses the system temporary directory to create temp files. You can alter the location of the temporary directory using the following steps.

• Create a new directory called *spss* and subdirectory called *servertemp*.

13

- ► Edit *options.cfg*, located in the */config* directory of your SPSS Modeler installation directory. Edit the temp_directory parameter in this file to read: temp_directory, "C:/spss/servertemp".
- ► After doing this, you must restart the SPSS Modeler Server service. You can do this by clicking the Services tab on your Windows Control Panel. Just stop the service and then start it to activate the changes you made. Restarting the machine will also restart the service.

All temp files will now be written to this new directory.

Note: The most common error when you are attempting to do this is to use the wrong type of slashes. Because of SPSS Modeler's UNIX history, forward slashes are used.

Starting Multiple IBM SPSS Modeler Sessions

If you need to launch more than one IBM® SPSS® Modeler session at a time, you must make some changes to your IBM® SPSS® Modeler and Windows settings. For example, you may need to do this if you have two separate server licenses and want to run two streams against two different servers from the same client machine.

To enable multiple SPSS Modeler sessions:

- Click: Start > [All] Programs > IBM SPSS Modeler15
- ▶ On the IBM SPSS Modeler15 shortcut (the one with the icon), right-click and select Properties.
- ▶ In the Target text box, add -noshare to the end of the string.
- In Windows Explorer, select: Tools > Folder Options...
- ▶ On the File Types tab, select the SPSS Modeler Stream option and click Advanced.
- ▶ In the Edit File Type dialog box, select Open with SPSS Modeler and click Edit.
- ▶ In the Application used to perform action text box, add -noshare before the -stream argument.

IBM SPSS Modeler Interface at a Glance

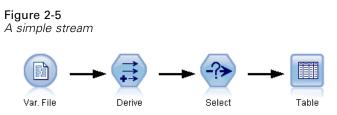
At each point in the data mining process, IBM® SPSS® Modeler's easy-to-use interface invites your specific business expertise. Modeling algorithms, such as prediction, classification, segmentation, and association detection, ensure powerful and accurate models. Model results can easily be deployed and read into databases, IBM® SPSS® Statistics, and a wide variety of other applications.

Working with SPSS Modeler is a three-step process of working with data.

- First, you read data into SPSS Modeler.
- Next, you run the data through a series of manipulations.
- Finally, you send the data to a destination.

14

This sequence of operations is known as a **data stream** because the data flows record by record from the source through each manipulation and, finally, to the destination—either a model or type of data output.



IBM SPSS Modeler Stream Canvas

The stream canvas is the largest area of the IBM® SPSS® Modeler window and is where you will build and manipulate data streams.

Streams are created by drawing diagrams of data operations relevant to your business on the main canvas in the interface. Each operation is represented by an icon or **node**, and the nodes are linked together in a **stream** representing the flow of data through each operation.

You can work with multiple streams at one time in SPSS Modeler, either in the same stream canvas or by opening a new stream canvas. During a session, streams are stored in the Streams manager, at the upper right of the SPSS Modeler window.

Nodes Palette

Most of the data and modeling tools in IBM® SPSS® Modeler reside in the **Nodes Palette**, across the bottom of the window below the stream canvas.

For example, the Record Ops palette tab contains nodes that you can use to perform operations on the data **records**, such as selecting, merging, and appending.

To add nodes to the canvas, double-click icons from the Nodes Palette or drag and drop them onto the canvas. You then connect them to create a **stream**, representing the flow of data.

Figure 2-6

Record Ops tab on the nodes palette	Э
-------------------------------------	---

🙎 Favorite	s Sou	rces 🔵	Record Ops	Field Ops	A Graphs	Modeling	🔷 Database Mode	eling Output	Export	 IBM® SPSS® Statistics
_?>	••>	•1>	€	RFM	€	>>	▶ ↔			
Select	Sample	Balance	Aggregate	RFM Aggregate	Sort	Merge	Append Distinct			

Each palette tab contains a collection of related nodes used for different phases of stream operations, such as:

- **Sources.** Nodes bring data into SPSS Modeler.
- Record Ops. Nodes perform operations on data records, such as selecting, merging, and appending.

- **Field Ops.** Nodes perform operations on data **fields**, such as filtering, deriving new fields, and determining the measurement level for given fields.
- **Graphs.** Nodes graphically display data before and after modeling. Graphs include plots, histograms, web nodes, and evaluation charts.
- Modeling. Nodes use the modeling algorithms available in SPSS Modeler, such as neural nets, decision trees, clustering algorithms, and data sequencing.
- Database Modeling. Nodes use the modeling algorithms available in Microsoft SQL Server, IBM DB2, and Oracle databases.
- **Output.** Nodes produce a variety of output for data, charts, and model results that can be viewed in SPSS Modeler.
- **Export.** Nodes produce a variety of output that can be viewed in external applications, such as IBM® SPSS® Data Collection or Excel.
- SPSS Statistics. Nodes import data from, or export data to, IBM® SPSS® Statistics, as well as running SPSS Statistics procedures.

As you become more familiar with SPSS Modeler, you can customize the palette contents for your own use.

Located below the Nodes Palette, a report pane provides feedback on the progress of various operations, such as when data is being read into the data stream. Also located below the Nodes Palette, a status pane provides information on what the application is currently doing, as well as indications of when user feedback is required.

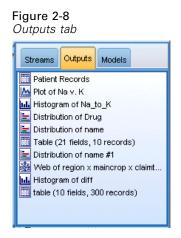
IBM SPSS Modeler Managers

At the top right of the window is the managers pane. This has three tabs, which are used to manage streams, output and models.

You can use the Streams tab to open, rename, save, and delete the streams created in a session.



The Outputs tab contains a variety of files, such as graphs and tables, produced by stream operations in IBM® SPSS® Modeler. You can display, save, rename, and close the tables, graphs, and reports listed on this tab.



The Models tab is the most powerful of the manager tabs. This tab contains all model **nuggets**, which contain the models generated in SPSS Modeler, for the current session. These models can be browsed directly from the Models tab or added to the stream in the canvas.

Figure 2-9 Models tab containing model nuggets

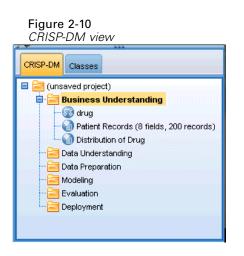


IBM SPSS Modeler Projects

On the lower right side of the window is the project pane, used to create and manage data mining **projects** (groups of files related to a data mining task). There are two ways to view projects you create in IBM® SPSS® Modeler—in the Classes view and the CRISP-DM view.

The CRISP-DM tab provides a way to organize projects according to the Cross-Industry Standard Process for Data Mining, an industry-proven, nonproprietary methodology. For both experienced and first-time data miners, using the CRISP-DM tool will help you to better organize and communicate your efforts.

IBM SPSS Modeler Overview



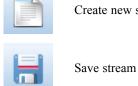
The Classes tab provides a way to organize your work in SPSS Modeler categorically-by the types of objects you create. This view is useful when taking inventory of data, streams, and models.





IBM SPSS Modeler Toolbar

At the top of the IBM® SPSS® Modeler window, you will find a toolbar of icons that provides a number of useful functions. Following are the toolbar buttons and their functions.



Create new stream

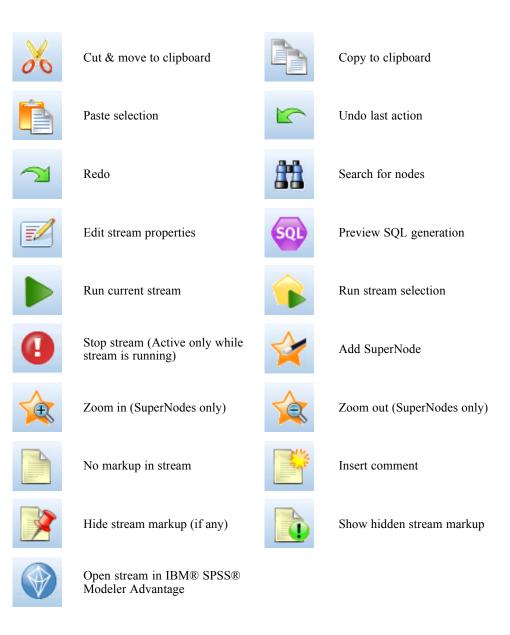


Open stream



Print current stream

•



Stream markup consists of stream comments, model links, and scoring branch indications.

Model links are described in the IBM SPSS Modeling Nodes guide.

IBM SPSS Modeler Overview

Customizing the Toolbar

You can change various aspects of the toolbar, such as:

- Whether it is displayed
- Whether the icons have tooltips available
- Whether it uses large or small icons

To turn the toolbar display on and off:

On the main menu, click:
 View > Toolbar > Display

To change the tooltip or icon size settings:

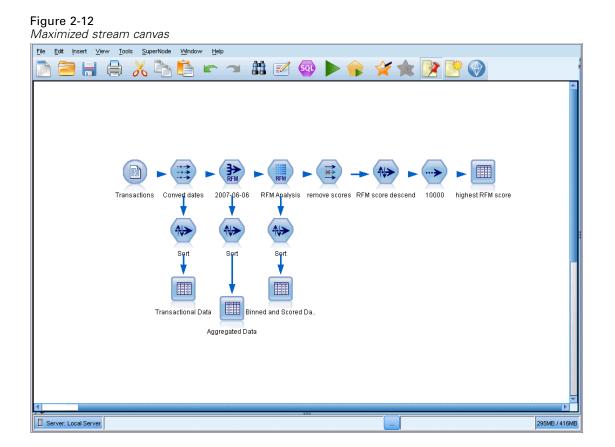
On the main menu, click:
 View > Toolbar > Customize

Click Show ToolTips or Large Buttons as required.

Customizing the IBM SPSS Modeler Window

Using the dividers between various portions of the IBM® SPSS® Modeler interface, you can resize or close tools to meet your preferences. For example, if you are working with a large stream, you can use the small arrows located on each divider to close the nodes palette, managers pane, and project pane. This maximizes the stream canvas, providing enough work space for large or multiple streams.

Alternatively, on the View menu, click Nodes Palette, Managers, or Project to turn the display of these items on or off.



As an alternative to closing the nodes palette, and the managers and project panes, you can use the stream canvas as a scrollable page by moving vertically and horizontally with the scrollbars at the side and bottom of the SPSS Modeler window.

You can also control the display of screen markup, which consists of stream comments, model links, and scoring branch indications. To turn this display on or off, click: View > Stream Markup

Changing the icon size for a stream

You can change the size of the stream icons in the following ways.

- Through a stream property setting
- Through a pop-up menu in the stream
- Using the keyboard

You can scale the entire stream view to one of a number of sizes between 8% and 200% of the standard icon size.

Figure 2-13 Changing the icon size

	🔚 <u>S</u> ave Stream	Ctrl+S	◎ 200%		
tree_credit.sev Type Creditating	S <u>a</u> ve Stream As		0 150%		
	😻 Store as Stream		🔘 100% (Standard)		
Tudda	Deploy		O 92%		
9	Clos <u>e</u> Stream		0 83%		
Artalysis	New Stream	Ctrl+N	🔘 75% (Small)		
	🚐 — 🔁 Open Stream	Ctrl+O	0 67%		
			0 58%		
	Stream.	0 50%			
	Ne <u>w</u> Comment		0 46%		
	👗 Cu <u>t</u>	Ctrl+X	O 42%		
	Copy	Ctrl+C	© 37%		
	💼 <u>P</u> aste	Ctrl+V	◎ 33%		
	× <u>D</u> elete	Delete	© 25%		
	Select <u>A</u> ll	Ctrl+A	© 17%		
	Automatic Layou	t	© 12%		
	lcon Si <u>z</u> e	•	0 8%		
	📥 Print	Ctrl+P			

To scale the entire stream (stream properties method)

- From the main menu, choose
 Tools > Stream Properties > Options > Layout.
- Choose the size you want from the Icon Size menu.
- Click Apply to see the result.
- Click OK to save the change.

To scale the entire stream (menu method)

- ▶ Right-click the stream background on the canvas.
- Choose Icon Size and select the size you want.

To scale the entire stream (keyboard method)

- ▶ Press Ctrl + [-] on the main keyboard to zoom out to the next smaller size.
- ▶ Press Ctrl + Shift + [+] on the main keyboard to zoom in to the next larger size.

This feature is particularly useful for gaining an overall view of a complex stream. You can also use it to minimize the number of pages needed to print a stream.

Using the Mouse in IBM SPSS Modeler

The most common uses of the mouse in IBM® SPSS® Modeler include the following:

- **Single-click.** Use either the right or left mouse button to select options from menus, open pop-up menus, and access various other standard controls and options. Click and hold the button to move and drag nodes.
- Double-click. Double-click using the left mouse button to place nodes on the stream canvas and edit existing nodes.
- Middle-click. Click the middle mouse button and drag the cursor to connect nodes on the stream canvas. Double-click the middle mouse button to disconnect a node. If you do not have a three-button mouse, you can simulate this feature by pressing the Alt key while clicking and dragging the mouse.

Using Shortcut Keys

Many visual programming operations in IBM® SPSS® Modeler have shortcut keys associated with them. For example, you can delete a node by clicking the node and pressing the Delete key on your keyboard. Likewise, you can quickly save a stream by pressing the S key while holding down the Ctrl key. Control commands like this one are indicated by a combination of Ctrl and another key—for example, Ctrl+S.

There are a number of shortcut keys used in standard Windows operations, such as Ctrl+X to cut. These shortcuts are supported in SPSS Modeler along with the following application-specific shortcuts.

Note: In some cases, old shortcut keys used in SPSS Modeler conflict with standard Windows shortcut keys. These old shortcuts are supported with the addition of the Alt key. For example, Ctrl+Alt+C can be used to toggle the cache on and off.

Shortcut Key	Function
Ctrl+A	Select all
Ctrl+X	Cut
Ctrl+N	New stream
Ctrl+O	Open stream
Ctrl+P	Print
Ctrl+C	Сору
Ctrl+V	Paste
Ctrl+Z	Undo
Ctrl+Q	Select all nodes downstream of the selected node
Ctrl+W	Deselect all downstream nodes (toggles with Ctrl+Q)
Ctrl+E	Run from selected node
Ctrl+S	Save current stream
Alt+Arrow keys	Move selected nodes on the stream canvas in the direction of the arrow used
Shift+F10	Open the pop-up menu for the selected node

Table 2-1Supported shortcut keys

IBM SPSS Modeler Overview

Table 2-2Supported shortcuts for old hot keys

Shortcut Key	Function
Ctrl+Alt+D	Duplicate node
Ctrl+Alt+L	Load node
Ctrl+Alt+R	Rename node
Ctrl+Alt+U	Create User Input node
Ctrl+Alt+C	Toggle cache on/off
Ctrl+Alt+F	Flush cache
Ctrl+Alt+X	Expand SuperNode
Ctrl+Alt+Z	Zoom in/zoom out
Delete	Delete node or connection

Printing

The following objects can be printed in IBM® SPSS® Modeler:

- Stream diagrams
- Graphs
- Tables
- Reports (from the Report node and Project Reports)
- Scripts (from the stream properties, Standalone Script, or SuperNode script dialog boxes)
- Models (Model browsers, dialog box tabs with current focus, tree viewers)
- Annotations (using the Annotations tab for output)

To print an object:

- To print without previewing, click the Print button on the toolbar.
- To set up the page before printing, select Page Setup from the File menu.
- To preview before printing, select Print Preview from the File menu.
- To view the standard print dialog box with options for selecting printers, and specifying appearance options, select Print from the File menu.

Automating IBM SPSS Modeler

Since advanced data mining can be a complex and sometimes lengthy process, IBM® SPSS® Modeler includes several types of coding and automation support.

Control Language for Expression Manipulation (CLEM) is a language for analyzing and manipulating the data that flows along SPSS Modeler streams. Data miners use CLEM extensively in stream operations to perform tasks as simple as deriving profit from cost and 24

revenue data or as complex as transforming web log data into a set of fields and records with usable information.

Scripting is a powerful tool for automating processes in the user interface. Scripts can perform the same kinds of actions that users perform with a mouse or a keyboard. You can set options for nodes and perform derivations using a subset of CLEM. You can also specify output and manipulate generated models.

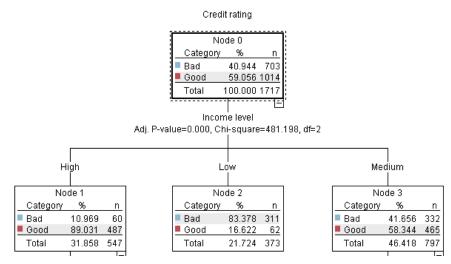
Introduction to Modeling

A model is a set of rules, formulas, or equations that can be used to predict an outcome based on a set of input fields or variables. For example, a financial institution might use a model to predict whether loan applicants are likely to be good or bad risks, based on information that is already known about past applicants.

The ability to predict an outcome is the central goal of predictive analytics, and understanding the modeling process is the key to using IBM® SPSS® Modeler.

Figure 3-1

A simple decision tree model



This example uses a **decision tree** model, which classifies records (and predicts a response) using a series of decision rules, for example:

IF income = Medium AND cards <5 THEN -> 'Good'

While this example uses a CHAID (Chi-squared Automatic Interaction Detection) model, it is intended as a general introduction, and most of the concepts apply broadly to other modeling types in SPSS Modeler.

To understand any model, you first need to understand the data that go into it. The data in this example contain information about the customers of a bank. The following fields are used:

Field name	Description
Credit_rating	Credit rating: 0=Bad, 1=Good, 9=missing values
Age	Age in years

© Copyright IBM Corporation 1994, 2012.

Field name	Description
Income	Income level: 1=Low, 2=Medium, 3=High
Credit_cards	Number of credit cards held: 1=Less than five, 2=Five or more
Education	Level of education: 1=High school, 2=College
Car_loans	Number of car loans taken out: 1=None or one, 2=More than two

The bank maintains a database of historical information on customers who have taken out loans with the bank, including whether or not they repaid the loans (Credit rating = Good) or defaulted (Credit rating = Bad). Using this existing data, the bank wants to build a model that will enable them to predict how likely future loan applicants are to default on the loan.

Using a decision tree model, you can analyze the characteristics of the two groups of customers and predict the likelihood of loan defaults.

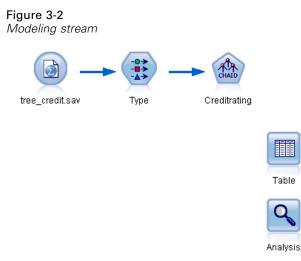
This example uses the stream named *modelingintro.str*, available in the *Demos* folder under the *streams* subfolder. The data file is *tree_credit.sav*. For more information, see the topic Demos Folder in Chapter 1 on p. 6.

Let's take a look at the stream.

- Choose the following from the main menu:
 File > Open Stream
- Click the gold nugget icon on the toolbar of the Open dialog box and choose the Demos folder.
- ► Double-click the *streams* folder.
- ▶ Double-click the file named *modelingintro.str*.

Introduction to Modeling

Building the Stream



To build a stream that will create a model, we need at least three elements:

- A source node that reads in data from some external source, in this case an IBM® SPSS® Statistics data file.
- A source or Type node that specifies field properties, such as measurement level (the type of data that the field contains), and the role of each field as a target or input in modeling.
- A modeling node that generates a model nugget when the stream is run.

In this example, we're using a CHAID modeling node. CHAID, or Chi-squared Automatic Interaction Detection, is a classification method that builds decision trees by using a particular type of statistics known as chi-square statistics to work out the best places to make the splits in the decision tree.

If measurement levels are specified in the source node, the separate Type node can be eliminated. Functionally, the result is the same.

This stream also has Table and Analysis nodes that will be used to view the scoring results after the model nugget has been created and added to the stream.

The Statistics File source node reads data in SPSS Statistics format from the *tree_credit.sav* data file, which is installed in the *Demos* folder. (A special variable named *\$CLEO_DEMOS* is used to reference this folder under the current IBM® SPSS® Modeler installation. This ensures the path will be valid regardless of the current installation folder or version.)

Figure 3-3 Reading data with a Statistics File source node

📀 tree_credit.sav	
Preview Refresh	0
\$CLEO_DEMOS'tree_credit.sav	
Data Filter Types Annotations	
Import file: \$CLEO_DEMOStree_credit.sav	
Variable names: \bigcirc Read names and labels $ extbf{@}$ Read labels as names	
Values: O Read data and labels O Read labels as data	
Use field format information to determine storage	
OK Cancel	Apply Reset

The Type node specifies the **measurement level** for each field. The measurement level is a category that indicates the type of data in the field. Our source data file uses three different measurement levels.

A **Continuous** field (such as the *Age* field) contains continuous numeric values, while a **Nominal** field (such as the *Credit rating* field) has two or more distinct values, for example *Bad*, *Good*, or *No credit history*. An **Ordinal** field (such as the *Income level* field) describes data with multiple distinct values that have an inherent order—in this case *Low*, *Medium* and *High*.

Figure 3-4

Setting the target and input fields with the Type node

Type President	Annotations				
√ , 000 €	🔊 🜓 Read Value	es Clear '	/alues	Clear All Value	es
Field -	Measurement	Values	Missing	Check	Role
A Credit rating	💑 Nominal	Bad,Good	*	None	🔘 Target
🛞 Age	🔗 Continuous	[20.00269		None	🔪 Input
A Income level	📊 Ordinal	High,Low,		None	🔪 Input
A Number of	💑 Nominal	"Less tha		None	🔪 Input
A Education	💑 Nominal	"High sch		None	🔪 Input
\Lambda Car loans	💑 Nominal	"None or		None	🔪 Input
© View current f	ields 🔘 View unused	t field settings			Apply Reset

For each field, the Type node also specifies a **role**, to indicate the part that each field plays in modeling. The role is set to *Target* for the field *Credit rating*, which is the field that indicates whether or not a given customer defaulted on the loan. This is the **target**, or the field for which we want to predict the value.

Role is set to *Input* for the other fields. Input fields are sometimes known as **predictors**, or fields whose values are used by the modeling algorithm to predict the value of the target field.

The CHAID modeling node generates the model.

On the Fields tab in the modeling node, the option Use predefined roles is selected, which means the target and inputs will be used as specified in the Type node. We could change the field roles at this point, but for this example we'll use them as they are.

• Click the Build Options tab.

Figure 3-5

CHAID modeling node, Fields tab

📀 Creditrating	
	0
Objective: Standard model	
Fields Build Options Model Options Annotations	
 Use predefined roles Use custom field assignments 	
Fields:	
Sort: None 🔻 🔷 📢	
Credit rating	
Duadiatava (lavuta)t	
Predictors (Inputs)*:	
ncome level	
Number of credit cards	
Education	
	8 & 1 8 1 1
Analysis Weight:	
	8 8 1 8 1
OK Run Cancel	Apply Reset

Here there are several options where we could specify the kind of model we want to build.

We want a brand-new model, so we'll use the default option Build new model.

We also just want a single, standard decision tree model without any enhancements, so we'll also leave the default objective option Build a single tree.

While we can optionally launch an interactive modeling session that allows us to fine-tune the model, this example simply generates a model using the default mode setting Generate model.

📀 Creditrating							
CHAID							
Objective:	Standard model						
Fields Build Opt	Model Options Annotations						
Select an item:							
Objective	What do you want to do?						
Basics	Build new model Continue training existing model						
Stopping Rules	What is your main objective?						
Costs	Buil <u>d</u> a single tree						
Ensembles	Single Tree						
Advanced	Mode: OGenerate model C Launch interactive session						
	Use tree directives Directive						
	◯ Enhance model accuracy (boosting)						
	C Enhance model stability (bagging)						
	\bigcirc Create a model for <u>v</u> ery large datasets (requires Server)						
	Description						
	Creates a single, standard model to explain relationships between fields. Standard models are easier to interpret and can be faster to score than boosted, bagged, or large dataset ensembles.						
OK 🕨 Ru	n Cancel Apply Reset						

Figure 3-6 CHAID modeling node, Build Options tab

For this example, we want to keep the tree fairly simple, so we'll limit the tree growth by raising the minimum number of cases for parent and child nodes.

- On the Build Options tab, select Stopping Rules from the navigator pane on the left.
- ► Select the Use absolute value option.
- ► Set Minimum records in parent branch to 400.

► Set Minimum records in child branch to 200.

Figure 3-7

Setting the stopping criteria for decision tree building

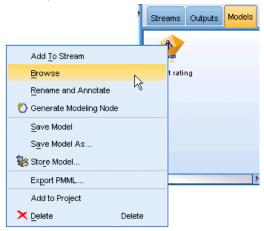
🕜 Cred	litrating					
CHAID	Objective: Sta	ndard model				0
Fields	Build Options	Model Options	Annotations			
<u>S</u> elect an	item:					
Objectiv	e	O Use perce	entage			
Basics Stopping	Rules	<u>M</u> inimum	records in par	ent branch(%):	2.0 🌲	
	, rtuics	Mi <u>n</u> imum	records in chil	d branch(%):	1.0 🚔	
Costs		🔘 Use abso	ute value			
Ensembl	es	Minimum	re <u>c</u> ords in par	ent brench:	400 🚔	
Advance	ed	Miniman	re <u>c</u> orus in par			
		Minimum	records in chil	d branch:	200 ≑	
ОК	🕨 🕨 Run	Cancel				Apply Reset

We can use all the other default options for this example, so click Run to create the model. (Alternatively, right-click on the node and choose Run from the context menu, or select the node and choose Run from the Tools menu.)

Browsing the Model

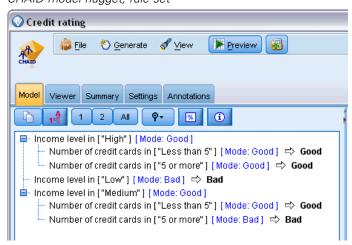
When execution completes, the model nugget is added to the Models palette in the upper right corner of the application window, and is also placed on the stream canvas with a link to the modeling node from which it was created. To view the model details, right-click on the model nugget and choose Browse (on the models palette) or Edit (on the canvas).

Figure 3-8 Models palette



In the case of the CHAID nugget, the Model tab displays the details in the form of a rule set—essentially a series of rules that can be used to assign individual records to child nodes based on the values of different input fields.

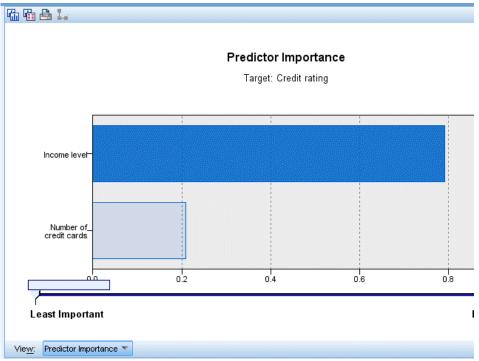
Figure 3-9 CHAID model nugget, rule set



For each decision tree terminal node—meaning those tree nodes that are not split further—a prediction of *Good* or *Bad* is returned. In each case the prediction is determined by the **mode**, or most common response, for records that fall within that node.

To the right of the rule set, the Model tab displays the Predictor Importance chart, which shows the relative importance of each predictor in estimating the model. From this we can see that *Income level* is easily the most significant in this case, and that the only other significant factor is *Number of credit cards*.





The Viewer tab in the model nugget displays the same model in the form of a tree, with a node at each decision point. Use the Zoom controls on the toolbar to zoom in on a specific node or zoom out to see the more of the tree.

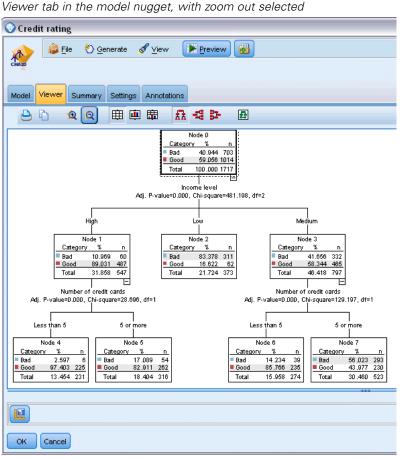


Figure 3-11

Looking at the upper part of the tree, the first node (Node 0) gives us a summary for all the records in the data set. Just over 40% of the cases in the data set are classified as a bad risk. This is quite a high proportion, so let's see if the tree can give us any clues as to what factors might be responsible.

We can see that the first split is by *Income level*. Records where the income level is in the *Low* category are assigned to Node 2, and it's no surprise to see that this category contains the highest percentage of loan defaulters. Clearly lending to customers in this category carries a high risk.

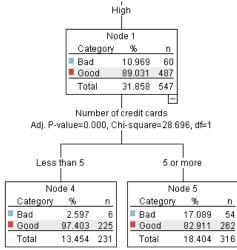
However, 16% of the customers in this category actually *didn't* default, so the prediction won't always be correct. No model can feasibly predict every response, but a good model should allow us to predict the *most likely* response for each record based on the available data.

In the same way, if we look at the high income customers (Node 1), we see that the vast majority (89%) are a good risk. But more than 1 in 10 of these customers has also defaulted. Can we refine our lending criteria to minimize the risk here?

Notice how the model has divided these customers into two sub-categories (Nodes 4 and 5), based on the number of credit cards held. For high-income customers, if we lend only to those with fewer than 5 credit cards, we can increase our success rate from 89% to 97%—an even more satisfactory outcome.

Figure 3-12



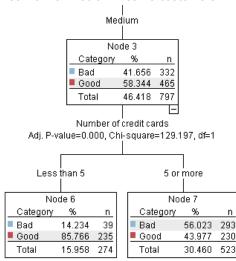


But what about those customers in the Medium income category (Node 3)? They're much more evenly divided between Good and Bad ratings.

Again, the sub-categories (Nodes 6 and 7 in this case) can help us. This time, lending only to those medium-income customers with fewer than 5 credit cards increases the percentage of Good ratings from 58% to 85%, a significant improvement.

Figure 3-13

Tree view of medium-income customers



So, we've learnt that every record that is input to this model will be assigned to a specific node, and assigned a prediction of *Good* or *Bad* based on the most common response for that node.

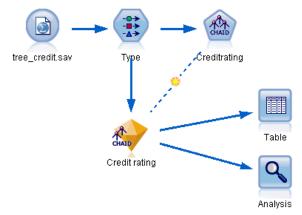
This process of assigning predictions to individual records is known as **scoring**. By scoring the same records used to estimate the model, we can evaluate how accurately it performs on the training data—the data for which we know the outcome. Let's look at how to do this.

Evaluating the Model

We've been browsing the model to understand how scoring works. But to evaluate *how accurately* it works, we need to score some records and compare the responses predicted by the model to the actual results. We're going to score the same records that were used to estimate the model, allowing us to compare the observed and predicted responses.



Attaching the model nugget to output nodes for model evaluation



► To see the scores or predictions, attach the Table node to the model nugget, double-click the Table node and click Run.

The table displays the predicted scores in a field named *\$R-Credit rating*, which was created by the model. We can compare these values to the original *Credit rating* field that contains the actual responses.

By convention, the names of the fields generated during scoring are based on the target field, but with a standard prefix such as R- for predictions or RC- for confidence values. Different models types use different sets of prefixes. A **confidence value** is the model's own estimation, on a scale from 0.0 to 1.0, of how accurate each predicted value is.

Figure 3-15

Table showing generated scores and confidence values

			_	
Number of credit cards	Education	Car loans	\$R-Credit rating	\$RC-Credit rating
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	High school	More than 2	Bad	0.832
5 or more	College	None or 1	Bad	0.832
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	High school	More than 2	Bad	0.832
5 or more	High school	More than 2	Bad	0.832
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.832
5 or more	High school	More than 2	Bad	0.832
5 or more	High school	More than 2	Bad	0.560
5 or more	College	None or 1	Bad	0.832
5 or more	High school	More than 2	Bad	0.832
5 or more	College	More than 2	Bad	0.832
5 or more	College	More than 2	Bad	0.832
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Good	0.827

As expected, the predicted value matches the actual responses for many records but not all. The reason for this is that each CHAID terminal node has a mix of responses. The prediction matches the *most common* one, but will be wrong for all the others in that node. (Recall the 16% minority of low-income customers who did not default.)

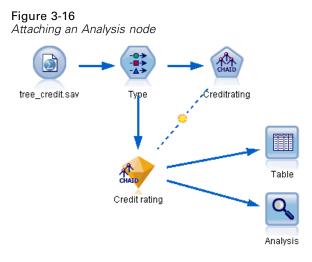
To avoid this, we could continue splitting the tree into smaller and smaller branches, until every node was 100% pure—all *Good* or *Bad* with no mixed responses. But such a model would be extremely complicated and would probably not generalize well to other datasets.

To find out exactly how many predictions are correct, we could read through the table and tally the number of records where the value of the predicted field *\$R-Credit rating* matches the value of *Credit rating*. Fortunately, there's a much easier way—we can use an Analysis node, which does this automatically.

• Connect the model nugget to the Analysis node.

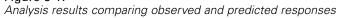
Introduction to Modeling

• Double-click the Analysis node and click Run.



The analysis shows that for 1899 out of 2464 records—over 77%—the value predicted by the model matched the actual response.

Figure 3-17



🔦 Analysis of [Cred	it rating]	_ 🗆 🔀
🐞 File 📄 Edit [0 ×
Analysis Annotations		
😵 Collapse All	Expand All	
Results for output fie	-	
	redit rating with Credit rating	
Correct	1,899 77.07%	
Wrong	565 22.93%	
Total	2,464	
		ОК

40

This result is limited by the fact that the records being scored are the same ones used to estimate the model. In a real situation, you could use a Partition node to split the data into separate samples for training and evaluation.

By using one sample partition to generate the model and another sample to test it, you can get a much better indication of how well it will generalize to other datasets.

The Analysis node allows us to test the model against records for which we already know the actual result. The next stage illustrates how we can use the model to score records for which we don't know the outcome. For example, this might include people who are not currently customers of the bank, but who are prospective targets for a promotional mailing.

Scoring Records

Earlier, we scored the same records used to estimate the model in order to evaluate how accurate the model was. Now we're going to see how to score a different set of records from the ones used to create the model. This is the goal of modeling with a target field: Study records for which you know the outcome, to identify patterns that will allow you to predict outcomes you don't yet know.

Figure 3-18

Attaching new data for scoring



You could update the Statistics File source node to point to a different data file, or you could add a new source node that reads in the data you want to score. Either way, the new dataset must contain the same input fields used by the model (*Age, Income level, Education* and so on) but not the target field *Credit rating*.

Alternatively, you could add the model nugget to any stream that includes the expected input fields. Whether read from a file or a database, the source type doesn't matter as long as the field names and types match those used by the model.

You could also save the model nugget as a separate file, export the model in PMML format for use with other applications that support this format, or store the model in an IBM® SPSS® Collaboration and Deployment Services repository, which offers enterprise-wide deployment, scoring, and management of models.

Regardless of the infrastructure used, the model itself works in the same way.

Summary

This example demonstrates the basic steps for creating, evaluating, and scoring a model.

- The modeling node estimates the model by studying records for which the outcome is known, and creates a model nugget. This is sometimes referred to as training the model.
- The model nugget can be added to any stream with the expected fields to score records. By scoring the records for which you already know the outcome (such as existing customers), you can evaluate how well it performs.
- Once you are satisfied that the model performs acceptably well, you can score new data (such as prospective customers) to predict how they will respond.
- The data used to train or estimate the model may be referred to as the analytical or historical data; the scoring data may also be referred to as the operational data.

Automated Modeling for a Flag Target

Modeling Customer Response (Auto Classifier)

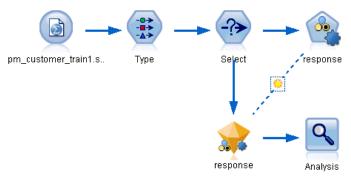
The Auto Classifier node enables you to automatically create and compare a number of different models for either flag (such as whether or not a given customer is likely to default on a loan or respond to a particular offer) or nominal (set) targets. In this example we'll search for a flag (yes or no) outcome. Within a relatively simple stream, the node generates and ranks a set of candidate models, chooses the ones that perform the best, and combines them into a single aggregated (Ensembled) model. This approach combines the ease of automation with the benefits of combining multiple models, which often yield more accurate predictions than can be gained from any one model.

This example is based on a fictional company that wants to achieve more profitable results by matching the right offer to each customer.

This approach stresses the benefits of automation. For a similar example that uses a continuous (numeric range) target, see Chapter 5 on p. 54.

Figure 4-1

Auto Classifier sample stream



This example uses the stream *pm binaryclassifier.str*, installed in the Demo folder under *streams*. The data file used is *pm* customer train1.sav.

Historical Data

The file *pm customer train1.sav* has historical data tracking the offers made to specific customers in past campaigns, as indicated by the value of the *campaign* field. The largest number of records fall under the Premium account campaign.

The values of the *campaign* field are actually coded as integers in the data (for example $2 = Premium \ account$). Later, you'll define labels for these values that you can use to give more meaningful output.

	(31 fields,	21,721						
🝃 Eile 📄 Edit 🖏 Generate 🛛 👪 🔛 🕢 🐼								
Table A	nnotations							
	customer_id	campaign	response	response_date	purchase	purchase_date	product_id	Rowid
1	7	2	0	\$null\$	0	\$null\$	\$null\$	1
2	13	2	0	\$null\$	0	\$null\$	\$null\$	2
3	15	2	0	\$null\$	0	\$null\$	\$null\$	3
4	16	2	1	2006-07-05 00:00:00	0	\$null\$	183	761
5	23	2	0	\$null\$	0	\$null\$	\$null\$	4
3	24	2	0	\$null\$	0	\$null\$	\$null\$	5
7	30	2	0	\$null\$	0	\$null\$	\$null\$	6
3	30	3	0	\$null\$	0	\$null\$	\$null\$	7
э 🛛	33	2	0	\$null\$	0	\$null\$	\$null\$	8
10	42	3	0	\$null\$	0	\$null\$	\$null\$	9
11	42	2	0	\$null\$	0	\$null\$	\$null\$	10
12	52	2	0	\$null\$	0	\$null\$	\$null\$	11
13	57	2	0	\$null\$	0	\$null\$	\$null\$	12
14	63	2	1	2006-07-14 00:00:00	0	\$null\$	183	1501
15	74	2	0	\$null\$	0	\$null\$	\$null\$	13
16	74	3	0	\$null\$	0	\$null\$	\$null\$	14
17	75	2	0	\$null\$	0	\$null\$	\$null\$	15
18	82	2	0	\$null\$	0	\$null\$	\$null\$	16
19	89	3	0	\$null\$	0	\$null\$	\$null\$	17
20	89	2	0	\$null\$	0	\$null\$	\$null\$	18

The file also includes a *response* field that indicates whether the offer was accepted (0 = no, and 1 = yes). This will be the **target field**, or value, that you want to predict. A number of fields containing demographic and financial information about each customer are also included. These can be used to build or "train" a model that predicts response rates for individuals or groups based on characteristics such as income, age, or number of transactions per month.

Building the Stream

Figure 4-2

► Add a Statistics File source node pointing to *pm_customer_train1.sav*, located in the *Demos* folder of your IBM® SPSS® Modeler installation. (You can specify \$CLE0_DEMOS/ in the file

path as a shortcut to reference this folder. Note that a forward slash—rather than a backslash—must be used in the path, as shown.)

Figure 4-3 Reading in the	e data		
📀 pm_c ustome	er_train1.sav		
	review 😰 Refresh		0
	_DEMOS/pm_customer_train1.	sav	
Data Filter Ty	pes Annotations		
Import file: \$CL	EO_DEMOS/pm_customer_trai	n1.sav	
Variable names:	Read names and labels	🔘 Read labels as names	
Values:	Read data and labels	🔘 Read labels as data	
👿 Use field forma	at information to determine sto	rage	
OK Cancel			<u>Apply</u> <u>R</u> eset

► Add a Type node, and select *response* as the target field (Role = Target). Set the Measurement for this field to Flag.

Figure 4-4

Setting the measurement level and role

Type							
~	🔹 🚺 🕨 Read Valu	es Clear	Values	Clear All Val	ues		
Field -	Measurement	Values	Missing	Check	Role		
父 customer_id 🐰	🔗 Continuous	[7,116993]		None	🛇 None 🛛 🔺		
🔷 campaign 🛛 🌘	🇞 Nominal	1,2,3,4		None	🔪 Input		
🔆 response	🎖 Flag	1/0		None	🔘 Target 📃		
🚾 response 🗸	🔗 Continuous	[2006-04		None	🛇 None		
🔆 purchase 🛛 🗸	🔗 Continuous	[0,1]		None	🛇 None		
🚾 purchase 🗸	🖉 Continuous	[2006-04		None	🛇 None		
🔆 product_id 🛛	🖉 Continuous	[183,421]		None	🛇 None		
🔆 Rowid 🛛	🖉 Continuous	[1,19599]		None	🛇 None 📃		
ana 🛆	Continuous	14 O GE1		None	🔪 loout 🛛 🚬		
 View current f OK Cancel 	iields 🔘 View unuse	ed field settin	gs	(Apply <u>R</u> eset		

- ► Set the role to None for the following fields: *customer_id*, *campaign*, *response_date*, *purchase*, *purchase_date*, *product_id*, *Rowid*, and *X_random*. These fields will be ignored when you are building the model.
- Click the Read Values button in the Type node to make sure that values are instantiated.

As we saw earlier, our source data includes information about four different campaigns, each targeted to a different type of customer account. These campaigns are coded as integers in the data, so to make it easier to remember which account type each integer represents, let's define labels for each one.

Figure 4-5 Choosing to specify values for a field

Type						
Read Val	ues Clear	^r Values	Clear All Va	lues		
Field - Measurement	Values	Missing	Check	Role		
🔆 customer_id 🔗 Continuous	[7,116993]		None	S None		
🔿 campaign 🛛 🍪 Nominal	<curr td="" 💌<=""><td colspan="2"><curr td="" 🔽<=""><td>🔪 Input</td><td></td></curr></td></curr>	<curr td="" 🔽<=""><td>🔪 Input</td><td></td></curr>		🔪 Input		
🔆 response 🛛 🎖 Flag	<read></read>		None	🔘 Target		
🚾 response 💉 Continuous	<read +=""></read>	<read +=""></read>		🛇 None		
🔆 purchase 🛛 🛷 Continuous	<pass></pass>	<pass> N</pass>		🛇 None		
🚾 purchase 💉 Continuous	<current></current>		None	🛇 None		
今 product_id 🛛 🔗 Continuous	SpecifyN		None	🛇 None		
🔆 Rowid 🛛 🔗 Continuous	11,19599	•	None	🛇 None		
🛆 ana 🛛 🔊 Continuous	140.061		None	N Innut		
View current fields View unus OK Cancel	ed field settin	gs		Apply Re	eset	

- On the row for the campaign field, click the entry in the Values column.
- Choose Specify from the drop-down list.

Figure 4	-6				
Defining	labels	for	the	field	values

😡 campaign	n Values	X
Measurement:	😞 Nominal 🔽 Storage: 🔆 Integer 🛛 Model Field	
Values:	 Read from data Pass Specify values and labels 	
	Values Labels 1 Standard account 2 Premium account 3 Gold account 4 Platinum account	↑ ↓ ×
Check values:	Extend values from data None wks	
	Missing values	×
	Range to:	
Description:		

- ▶ In the Labels column, type the labels as shown for each of the four values of the campaign field.
- ► Click OK.

Automated Modeling for a Flag Target

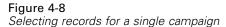
×

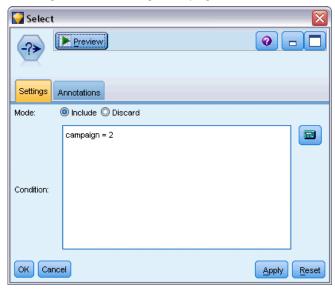
N £ 41 . .

違 File			ds) #3				
1	📄 Edit 🛛 💐) <u>G</u> enerate 🛛 👔		A #1			ଟ
Table A	nnotations						
	customer_id	campaign	response	response_date	purchase	purchase_date	product
1	7	Premium account	0	\$null\$	0	\$null\$	\$null\$
2	13	Premium account	0	\$null\$	0	\$null\$	\$null\$
3	15	Premium account	0	\$null\$	0	\$null\$	\$null\$
4	16	Premium account	1	2006-07-05 00:00:00	0	\$null\$	183
5	23	Premium account	0	\$null\$	0	\$null\$	\$null\$
6	24	Premium account	0	\$null\$	0	\$null\$	\$null\$
7	30	Premium account	0	\$null\$	0	\$null\$	\$null\$
8	30	Gold account	0	\$null\$	0	\$null\$	\$null\$
9	33	Premium account	0	\$null\$	0	\$null\$	\$null\$
10	42	Gold account	0	\$null\$	0	\$null\$	\$null\$
11	42	Premium account	0	\$null\$	0	\$null\$	\$null\$
12	52	Premium account	0	\$null\$	0	\$null\$	\$null\$
13	57	Premium account	0	\$null\$	0	\$null\$	\$null\$
14	63	Premium account	1	2006-07-14 00:00:00	0	\$null\$	183
15	74	Premium account	0	\$null\$	0	\$null\$	\$null\$
16	74	Gold account	0	\$null\$	0	\$null\$	\$null\$
17	75	Premium account	0	\$null\$	0	\$null\$	\$null\$
18	82	Premium account	0	\$null\$	0	\$null\$	\$null\$
19	89	Gold account	0	\$null\$	0	\$null\$	\$null\$
20	89	Premium account	0	\$null\$	0	\$null\$	\$null\$

- Attach a Table node to the Type node. ►
- Open the Table node and click Run. ►
- On the output window, click the Display field and value labels toolbar button to display the labels. ►
- Click OK to close the output window. ►

Although the data includes information about four different campaigns, you will focus the analysis on one campaign at a time. Since the largest number of records fall under the Premium account campaign (coded campaign=2 in the data), you can use a Select node to include only these records in the stream.





Generating and Comparing Models

• Attach an Auto Classifier node, and select Overall Accuracy as the metric used to rank models.

Automated Modeling for a Flag Target

Set the Number of models to use to 3. This means that the three best models will be built when you execute the node.

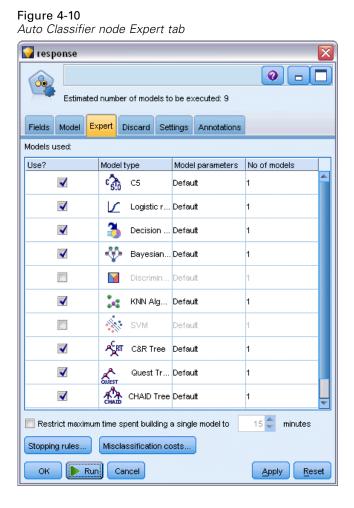
Figure Auto C		ier noa	le Moa	lel tab					
😡 resp	onse						×		
						0	- (
	Estim	ated numb	er of mod	els to be e:	(ecuted: 9				
Fields	Model	Expert	Discard	Settings	Annotations				
Model na	ame:	، (Auto 🔘 C	ustom					
📝 Use	partition	ied data							
📝 Build	model	for each s	plit						
Rank mo	dels by	: Ove	erall accur	acy 🔻					
Rank mo	dels usi	ing: 🔘 '	Fraining pa	artition 🤇	Test partition				
Number	of mode	els to use:			3 ≑				
📝 Calci	ulate pro	edictor imp	ortance						
Profit C	riteria (v	alid only f	or flag tar	gets)					
Costs:	0) Fixed		5.0 ≑	🔘 Variable		-		
Revenu	ue: 🧕) Fixed		10.0 ≑	🔘 Variable		-1		
Weight	Weight: Fixed 1.0 Variable Variable								
-Lift Crite	⊢Lift Criteria (valid only for flag targets)								
Percen	tile to u	se for lift c	alculation:	30					
ОК		Run	Cancel			Apply	Reset		

On the Expert tab you can choose from up to 11 different model algorithms.

Deselect the Discriminant and SVM model types. (These models take longer to train on these data, so deselecting them will speed up the example. If you don't mind waiting, feel free to leave them selected.)

50

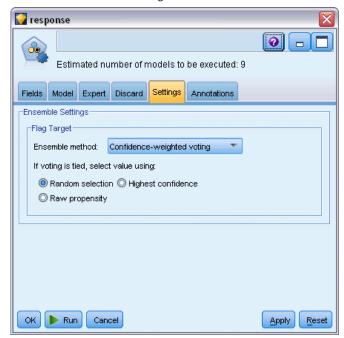
Because you set Number of models to use to 3 on the Model tab, the node will calculate the accuracy of the remaining nine algorithms and build a single model nugget containing the three most accurate.



► On the Settings tab, for the ensemble method, select Confidence-weighted voting. This determines how a single aggregated score is produced for each record.

With simple voting, if two out of three models predict *yes*, then *yes* wins by a vote of 2 to 1. In the case of confidence-weighted voting, the votes are weighted based on the confidence value for each prediction. Thus, if one model predicts *no* with a higher confidence than the two *yes* predictions combined, then *no* wins.

Figure 4-11 Auto Classifier node: Settings tab



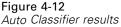
Click Run.

After a few minutes, the generated model nugget is built and placed on the canvas, and on the Models palette in the upper right corner of the window. You can browse the model nugget, or save or deploy it in a number of other ways.

Open the model nugget; it lists details about each of the models created during the run. (In a real situation, in which hundreds of models may be created on a large dataset, this could take many hours.) See Figure 4-1 on p. 42.

If you want to explore any of the individual models further, you can double-click on a model nugget icon in the Model column to drill down and browse the individual model results; from there you can generate modeling nodes, model nuggets, or evaluation charts. In the Graph column, you can double-click on a thumbnail to generate a full-sized graph.





By default, models are sorted based on overall accuracy, because this was the measure you selected on the Auto Classifier node Model tab. The C51 model ranks best by this measure, but the C&R Tree and CHAID models are nearly as accurate.

You can sort on a different column by clicking the header for that column, or you can choose the desired measure from the Sort by drop-down list on the toolbar.

Based on these results, you decide to use all three of these most accurate models. By combining predictions from multiple models, limitations in individual models may be avoided, resulting in a higher overall accuracy.

In the Use? column, select the C51, C&R Tree, and CHAID models.

Attach an Analysis node (Output palette) after the model nugget. Right-click on the Analysis node and choose Run to run the stream.

The aggregated score generated by the ensembled model is shown in a field named *\$XF-response*. When measured against the training data, the predicted value matches the actual response (as recorded in the original *response* field) with an overall accuracy of 92.82%.

While not quite as accurate as the best of the three individual models in this case (92.86% for C51), the difference is too small to be meaningful. In general terms, an ensembled model will typically be more likely to perform well when applied to datasets other than the training data.

🔍 Analysis of [response] 0 × 📦 File 📄 Edit Analysis Annotations Po Expand All 8 Collapse All E-Results for output field response Comparing \$XF-response with response Correct 12,534 92.82% Wrong 970 7.18% Total 13,504 OK

Figure 4-13 Analysis of the three ensembled models

Summary

To sum up, you used the Auto Classifier node to compare a number of different models, used the three most accurate models and added them to the stream within an ensembled Auto Classifier model nugget.

- Based on overall accuracy, the C51, C&R Tree, and CHAID models performed best on the training data.
- The ensembled model performed nearly as well as the best of the individual models and may perform better when applied to other datasets. If your goal is to automate the process as much as possible, this approach allows you to obtain a robust model under most circumstances without having to dig deeply into the specifics of any one model.

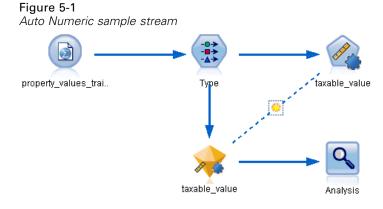
Automated Modeling for a Continuous Target

Property Values (Auto Numeric)

The Auto Numeric node enables you to automatically create and compare different models for continuous (numeric range) outcomes, such as predicting the taxable value of a property. With a single node, you can estimate and compare a set of candidate models and generate a subset of models for further analysis. The node works in the same manner as the Auto Classifier node, but for continuous rather than flag or nominal targets.

The node combines the best of the candidate models into a single aggregated (Ensembled) model nugget. This approach combines the ease of automation with the benefits of combining multiple models, which often yield more accurate predictions than can be gained from any one model.

This example focuses on a fictional municipality responsible for adjusting and assessing real estate taxes. To do this more accurately, they will build a model that predicts property values based on building type, neighborhood, size, and other known factors.



This example uses the stream *property_values_numericpredictor.str*, installed in the Demos folder under *streams*. The data file used is *property values train.sav*.

Training Data

The data file includes a field named *taxable_value*, which is the **target field**, or value, that you want to predict. The other fields contain information such as neighborhood, building type, and interior volume and may be used as predictors.

Field name	Label
property_id	Property ID

Field name	Label
neighborhood	Area within the city
building_type	Type of building
year_built	Year built
volume_interior	Volume of interior
volume_other	Volume of garage and extra buildings
lot_size	Lot size
taxable_value	Taxable value

A scoring data file named *property_values_score.sav* is also included in the Demos folder. It contains the same fields but without the *taxable_value* field. After training models using a dataset where the taxable value is known, you can score records where this value is not yet known.

Building the Stream

Add a Statistics File source node pointing to *property_values_train.sav*, located in the *Demos* folder of your IBM® SPSS® Modeler installation. (You can specify \$CLE0_DEMOS/ in the file path as a shortcut to reference this folder. Note that a forward slash—rather than a backslash—must be used in the path, as shown.)

Figure 5-2 Reading in the data								
📀 property_values_train.sav 🛛 🛛 🔀								
Preview 2 Refresh								
\$CLEO_DEMOS/property_values_train.sav								
Data Fitter Types Annotations								
Import file: \$CLEO_DEMOS/property_values_train.sav								
Variable names: Read names and labels Read labels as names 								
Values: Read data and labels Read labels as data 								
☑ Use field format information to determine storage								
OK Cancel	Apply Reset							

Add a Type node, and select *taxable_value* as the target field (Role = Target). Role should be set to Input for all other fields, indicating that they will be used as predictors.

Type						
-4-5					^	
Types Format Annotations						
🔨 🚾 🕶 🕨 Read Values Clear Values Clear All Values						
Field 💳	Measurement	Values	Missing	Check	Role	
🔅 property_id 🛛	🔗 Continuous	[2,21418]		None	🔪 Input	
A neighborhood	🇞 Nominal	Bloemenb		None	🔪 Input	
A building_type	💑 Nominal	"2-onder		None	🔪 Input	
决 year_built 🛛 .	🔗 Continuous	[1870,1992]	*	None	🔪 Input	
💭 volume_inte	🔗 Continuous	[138,1901]	*	None	🔪 Input	
🔅 volume_other .	🔗 Continuous	[0,496]		None	🔪 Input	
💭 lot_size 🛛 .	🔗 Continuous	[55,1310]	*	None	🔪 Input	
🔅 taxable_value .	🔗 Continuous	[40000,66	*	None	🔘 Target	
View current f		l field settings				

► Attach an Auto Numeric node, and select Correlation as the metric used to rank models.

Automated Modeling for a Continuous Target

Set the Number of models to use to 3. This means that the three best models will be built when you execute the node.

Figure 5-4 Auto Numeric node Model tab	
🙀 taxable_value	
	0 - 🗆
Estimated number of models to be ex	ecuted: 7
Fields Model Expert Settings Annotation	ns
Model name: 💿 Auto 🔘 Custom	
Vse partitioned data	
🗹 Build model for each split	
Rank models by: Correlation	
Rank models using: O Training partition	Test partition
Number of models to use:	3
Calculate predictor importance	
Do not keep models if:	
Correlation is less than	0.8 🖨
Number of fields is greater than	10 🗬
Relative error is greater than	1.0 🚔
OK 🕨 Run Cancel	Apply Reset

On the Expert tab, leave the default settings in place; the node will estimate a single model for each algorithm, for a total of seven models. (Alternatively, you can modify these settings to compare multiple variants for each model type.)

58

Because you set Number of models to use to 3 on the Model tab, the node will calculate the accuracy of the seven algorithms and build a single model nugget containing the three most accurate.

Figure 5-5 Auto Numeric node Expert tab							
😡 taxable_value				X			
				0			
Estimated number of models to be executed: 7							
Fields Model Exper	Settings	Annotation	าร				
Models used:							
Use?	Model type		Model parameters	No of models			
	🐹 Reg	ression	Default	1			
	🞉 Ger	eralized	Default	1			
	te KNN	I Algorithm	Default	1			
	🔅 svi	M	Default	1			
	ACRT C&F	R Tree	Default	1			
	СНА СНА	ID Tree	Default	1			
	🏠 Neu	ral Net	Default	1 🚽			
🔲 Restrict maximum tim	ie spent build	ling a single	model to 15 🌲	minutes			
Stopping rules							
OK 🕨 Run	Cancel			Apply Reset			

► On the Settings tab, leave the default settings in place. Since this is a continuous target, the ensemble score is generated by averaging the scores for the individual models.

Figure Auto N		ic node	e Settir	ngs tab		
😡 taxa	ible_va	lue				
						0
	Estim	nated nu	mber of r	nodels to be	executed: 6	
Fields	Model	Expert	Settings	Annotations		
Ensem	ble Settir	ngs				
The e	nsemble	scores f	or a contin	uous target wil	I be generated by averaging	g.
🔽 Ca	ilculate s	tandard e	error			
ОК	▶ Run	Cano	;el]			Apply Reset

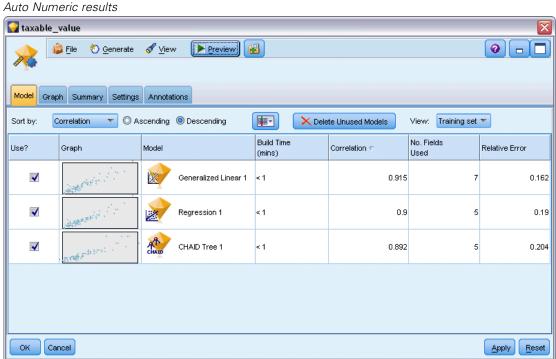
Comparing the Models

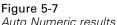
► Click the Run button.

The model nugget is built and placed on the canvas, and also on the Models palette in the upper right corner of the window. You can browse the nugget, or save or deploy it in a number of other ways.

Open the model nugget; it lists details about each of the models created during the run. (In a real situation, in which hundreds of models are estimated on a large dataset, this could take many hours.) See Figure 5-1 on p. 54.

If you want to explore any of the individual models further, you can double-click on a model nugget icon in the Model column to drill down and browse the individual model results; from there you can generate modeling nodes, model nuggets, or evaluation charts.





By default, models are sorted by correlation because this was the measure you selected in the Auto Numeric node. For purposes of ranking, the absolute value of the correlation is used, with values closer to 1 indicating a stronger relationship. The Generalized Linear model ranks best on this measure, but several others are nearly as accurate. The Generalized Linear model also has the lowest relative error.

You can sort on a different column by clicking the header for that column, or you can choose the desired measure from the Sort by list on the toolbar.

Each graph displays a plot of observed values against predicted values for the model, providing a quick visual indication of the correlation between them. For a good model, points should cluster along the diagonal, which is true for all the models in this example.

In the Graph column, you can double-click on a thumbnail to generate a full-sized graph.

Based on these results, you decide to use all three of these most accurate models. By combining predictions from multiple models, limitations in individual models may be avoided, resulting in a higher overall accuracy.

In the Use? column, ensure that all three models are selected.

Attach an Analysis node (Output palette) after the model nugget. Right-click on the Analysis node and choose Run to run the stream.

The averaged score generated by the ensembled model is added in a field named *\$XR-taxable_value*, with a correlation of 0.922, which is higher than those of the three individual models. The ensemble scores also show a low mean absolute error and may perform better than any of the individual models when applied to other datasets.

Figure 5-8 Auto Numeric sample stream

🔦 Analysi	s of [taxable_value]		
違 <u>F</u> ile 🛛	🖹 Edit 🔳 🕒 📢		0 X
Analysis	Annotations		
8 Collaps	e All 🦗 Expand All		
Results	for output field taxable_value		
🖻 Corr	paring \$XR-taxable_value wit	h taxable_value	
	Minimum Error	-156049.854	
	Maximum Error	176856.403	
	Mean Error	0.014	
	Mean Absolute Error	21353.824	
	Standard Deviation	30815.028	
	Linear Correlation	0.922	
	Occurrences	1,138	
			ок

Summary

To sum up, you used the Auto Numeric node to compare a number of different models, selected the three most accurate models and added them to the stream within an ensembled Auto Numeric model nugget.

- Based on overall accuracy, the Generalized Linear, Regression, and CHAID models performed best on the training data.
- The ensembled model showed performance that was better than two of the individual models and may perform better when applied to other datasets. If your goal is to automate the process as much as possible, this approach allows you to obtain a robust model under most circumstances without having to dig deeply into the specifics of any one model.

Part II: Data Preparation Examples

Automated Data Preparation (ADP)

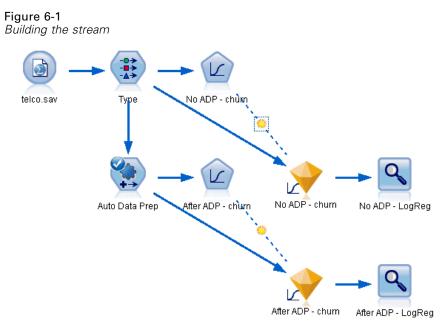
Preparing data for analysis is one of the most important steps in any data-mining project—and traditionally, one of the most time consuming. The Automated Data Preparation (ADP) node handles the task for you, analyzing your data and identifying fixes, screening out fields that are problematic or not likely to be useful, deriving new attributes when appropriate, and improving performance through intelligent screening techniques. You can use the node in fully automated fashion, allowing the node to choose and apply fixes, or you can preview the changes before they are made and accept or reject them as desired.

Using the ADP node enables you to make your data ready for data mining quickly and easily, without needing to have prior knowledge of the statistical concepts involved. If you run the node with the default settings, models will tend to build and score more quickly.

This example uses the stream named *ADP_basic_demo.str*, which references the data file named *telco.sav* to demonstrate the increased accuracy that may be found by using the default ADP node settings when building models. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *ADP_basic_demo.str* file is in the *streams* directory.

Building the Stream

► To build the stream, add a Statistics File source node pointing to *telco.sav* located in the *Demos* directory of your IBM® SPSS® Modeler installation.



• Attach a Type node to the source node, set the measurement level for the *churn* field to Flag, and set the role to Target. All other fields should have their role set to Input.

	review				0 -	
Types Format	Annotations	lues Clear	^r Values	Clear All Va	alues	
Field -	Measurement	Values	Missing	Check	Role	1
🖅 enill	💓 NUMINAI	0.0,1.0		NULLE	a inpur	-
A				None	🔪 Input	
🛞 loglong	Continuous	[-0.10536				-
🖗 logtoll	Continuous	[1.74919		None	🔪 Input	
🖗 logtoll 👰 logequi	Continuous	[1.74919 [2.73436		None None	> Input	
 logtoll logequi logcard 	Continuous Continuous Continuous	[1.74919 [2.73436 [1.01160		None None None	Input Input Input	
 logtoll logequi logcard logwire 	Continuous Continuous Continuous	[1.74919 [2.73436 [1.01160 [2.70136		None None None None	Input Input Input	
logtoll logequi logequi logcard logwire lninc	Continuous Continuous Continuous Continuous Continuous	[1.74919 [2.73436 [1.01160 [2.70136 [2.19722		None None None None None	Input Input Input Input Input	
 logtoll logequi logcard logwire 	Continuous Continuous Continuous	[1.74919 [2.73436 [1.01160 [2.70136		None None None None	Input Input Input	

• Attach a Logistic node to the Type node.

Automated Data Preparation (ADP)

► In the Logistic node, click the Model tab and select the Binomial procedure. In the *Model name* field, select Custom and enter No ADP - churn.

Figure 6-3 Choosing model opt	ions	
😡 No ADP - churn		X
Fields Model Expert A	nalyze Annotations	
Model name: 🔘 Auto 🔘 🤇	Custom	No ADP - churn
👿 Use partitioned data		
👿 Build model for each split		
Procedure: O Multinomia	I	Binomial
Binomial Procedure Method: Enter Categorical Inputs:)	
Field Name	Contrast	Base Category
		×
👿 Include constant in equat	ion	
OK 🕨 Run Cancel)	Apply Reset

- ► Attach an ADP node to the Type node. On the Objectives tab, leave the default settings in place to analyze and prepare your data by balancing both speed and accuracy.
- ► At the top of the Objectives tab, click Analyze Data to analyze and process your data.

Other options on the ADP node enable you to specify that you want to concentrate more on accuracy, more on the speed of processing, or to fine tune many of the data preparation processing steps.

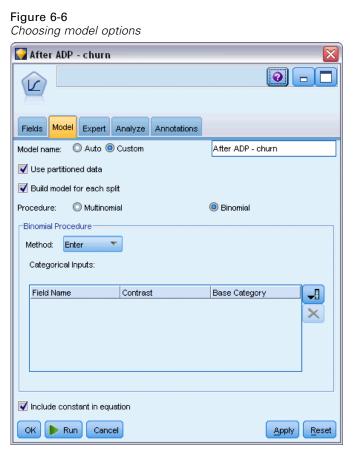
Figure 6-4 ADP default objectives
🚺 Auto Data Prep 🛛 🔀
Cenerate S View Preview Analyze Data X Clear Analysis
Objectives Fields Settings Analysis Annotations
Automated Data Preparation can recommend data preparation steps that will speed up model building and improve predictive power. This can include transforming, constructing and selecting features. The target can also be transformed.
What is your objective?
Balance speed and accuracy
Transform the data with an emphasis on building models with a balance of speed and accuracy.
Optimize for speed
Transform the data with an emphasis on building models as quickly as possible.
O Optimize for accuracy
Transform the data with an emphasis on building models with the greatest predictive power.
O Custom analysis
Choose this option to fine tune the algorithm on the Settings tab.
OK Cancel

The results of the data processing are displayed on the Analysis tab. The Field Processing Summary shows that of the 41 data features brought in to the ADP node, 19 have been transformed to aid processing, and 3 have been discarded as unused.

Figure 6-5 Summary of data processing	9			
😵 Auto Data Prep				×
Objectives Fields Settings Analysis Annotat		🗙 Clear Ani	natysis	
Objectives Fields Settings Analysis Annotat	ions		16 6 A L	
	ssing Summary	N	Predictors Recommended for Use in Ar Predictive Power	nalysis
Target			- Target: churn	
Target		1		
Predictors		41	transformed Equipment renta	
	Total	38	Internet 💫	
	Original fields (untransformed)	19	logiong transformed Electronic billing	
Predictors recommended for use in analysis	Transformations of original fields	19	Calling card service	
	Derived from dates and times	0	Level of education transformed	
	Constructed	0	employ transformed Customer category	
Predictors not used		3		
			Least Important Most Im	
View: Field Processing Summary 🔻 Reset			View: Predictive Power 🐨	
OK Cancel				pply <u>R</u> eset

• Attach a Logistic node to the ADP node.

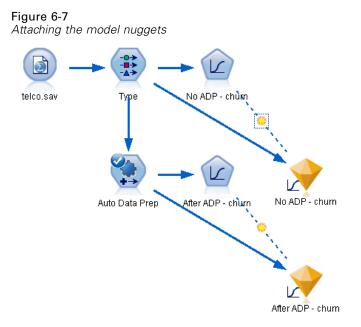
► In the Logistic node, click the Model tab and select the Binomial procedure. In the *Modeling name* field, select Custom and enter After ADP - churn.



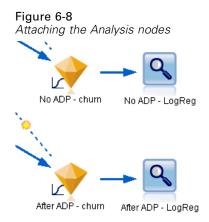
Automated Data Preparation (ADP)

Comparing Model Accuracy

Run both Logistic nodes to create the model nuggets, which are added to the stream and to the Models palette in the upper-right corner.



▶ Attach Analysis nodes to the model nuggets and run the Analysis nodes using their default settings.



The Analysis of the non ADP-derived model shows that just running the data through the Logistic Regression node with its default settings gives a model with low accuracy - just 10.6%.

F igure 6-9 Non ADP-deri	ved model re	sults	
🔍 No ADP - Lo	gReg		
🐞 File 🛛 📄 Ed	it 📳 🕒 📭	4 0	×
Analysis Annot	ations		
😵 Collapse All	Co Expand All		
-Results for ou	tput field churn		
	g \$L-churn with chur	rn	
Cori		10.6%	
Wro		89.4%	
Tota	il 1,000		
			OK

The Analysis of the ADP-derived model shows that running the data through the default ADP settings, you have built a much more accurate model that is 78.8% correct.

Figure 6-10 ADP-derived mo	del results	
🔍 After ADP - Log	Reg	
違 File 📄 Edit		0 ×
Analysis Annotation	31	
Scollapse All	Po Expand All	
Results for output		
	churn with churn	
Correct	788 78.8%	
Wrong	212 21.2%	
Total	1,000	
		ок

In summary, by just running the ADP node to fine tune the processing of your data, you were able to build a more accurate model with little direct data manipulation.

Obviously, if you are interested in proving or disproving a certain theory, or want to build specific models, you may find it beneficial to work directly with the model settings; however, for those with a reduced amount of time, or with a large amount of data to prepare, the ADP node may give you an advantage.

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the SPSS Modeler Algorithms Guide, available from the \Documentation directory of the installation disk.

Note that the results in this example are based on the training data only. To assess how well models generalize to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.

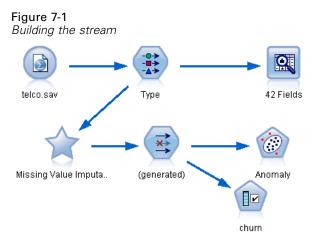
Preparing Data for Analysis (Data Audit)

The Data Audit node provides a comprehensive first look at the data you bring into IBM® SPSS® Modeler. Often used during the initial data exploration, the data audit report shows summary statistics as well as histograms and distribution graphs for each data field, and it allows you to specify treatments for missing values, outliers, and extreme values.

This example uses the stream named *telco_dataaudit.str*, which references the data file named *telco.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the SPSS Modeler program group on the Windows Start menu. The *telco_dataaudit.str* file is in the *streams* directory.

Building the Stream

► To build the stream, add a Statistics File source node pointing to *telco.sav* located in the *Demos* directory of your IBM® SPSS® Modeler installation.



Add a Type node to define fields, and specify *churn* as the target field (Role = Target). Role should be set to Input for all of the other fields so that this is the only target.

😩 🕒	Preview				
-4>					
Types Format	Annotations				
4. 00	Pead Va		Y	-	
1 . 000	🕿 [🕨 Read Va	lues Clear	Values	Clear All Va	llues
Field 🗂	Measurement	Values	Missing	Check	Role
ecilii	i⊛_ riag	170		NUTE	🔳 input
🤀 loglong	🔗 Continuous	[-0.10536		None	🔪 Input
🤣 logtoll	🔗 Continuous	[1.74919		None	🔪 Input
🋞 logequi	🔗 Continuous	[2.73436		None	🔪 Input
🛞 logcard	Continuous	[1.01160		None	🔪 Input
nogwire	Continuous	[2.70136		None	🔪 Input
	Continuous	[2.19722		None	🔪 Input
ninc 🖗		1,2,3,4		None	🔪 Input
X	本 Nominal	1-1-1-1		None	O Target
🖗 Ininc	Flag	1/0		NULLE	www.ranget

Confirm that field measurement levels are defined correctly. For example, most fields with values 0 and 1 can be regarded as flags, but certain fields, such as gender, are more accurately viewed as a nominal field with two values.

Figure 7-3 *Setting measurement levels*

Type	eview				0
∛ - ∞ •	Read Val	ues Clear	Values	Clear All Va	lues
Field 🗂	Measurement	Values	Missing	Check	Role
父 ed	📲 Ordinal	1,2,3,4,5		None	🔪 Input 🛛
🔆 employ	🔗 Continuous	[0,47]		None	🔪 Input
🛞 retire	💑 Nominal	0.0,1.0		None	🔪 Input
🔆 gender	💑 Nominal	0,1		None	🔪 input
🔆 reside	📶 Ordinal	1,2,3,4,5,		None	🔪 Input
🔆 tollfree	🎖 Flag	1/0		None	🔪 Input
🔆 equip	🎖 Flag 🎖 Flag	1/0		None	🔪 Input
🔆 callcard	🎖 Flag	1/0		None	🔪 Input
🔿 wireless	2 Flag	1/0		None	🔪 Innut 🔼
OK Cancel	fields 🔘 View unus	ed field settin	gs		Apply Rese

74

Tip: To change properties for multiple fields with similar values (such as 0/1), click the *Values* column header to sort fields by that column, and use the Shift key to select all of the fields you want to change. You can then right-click on the selection to change the measurement level or other attributes for all selected fields.

► Attach a Data Audit node to the stream. On the Settings tab, leave the default settings in place to include all fields in the report. Since *churn* is the only target field defined in the Type node, it will automatically be used as an overlay.

Dala Au		ie, sei	lings lab			
😡 42 Fie	elds					X
					Į.	
Settings	Quality	Output	Annotations			
🔘 Default			O U:	se custo	m fields	
Fields:						×
Overlay:						-1
-Display -			asic statistics		Advanced stat	
	ke mediar	Cancel	e (may slow p	ertorma	nce on large datase	 <u>R</u> eset

Figure 7-4 Data Audit node, Settings tab On the Quality tab, leave the default settings for detecting missing values, outliers, and extreme values in place, and click Run.

Figure 7 Data Au		le, Qua	ality tab				
😡 42 Fi	əlds						X
						0	
Settings	Quality	Output	Annotations				
	nt of reco		alid values ecords with in	valid values			
	& Extreme						
i i i i i i i i i i i i i i i i i i i	andard de	eviation fro	om mean				
Our	tliers:	3.0 🖨	Extremes:	5.0 ≑			
O Int	erquartile	ranges fr	om upper/lovv	er quartiles			
Out	tliers:	1.5 🌲	Extremes:	3.0 🌲			
Note:	Selecting	Interquart	ile range may	slow performa	ance on lar	rge datasets	
ок	Run	Cancel				Apply	Reset

Browsing Statistics and Charts

The Data Audit browser is displayed, with thumbnail graphs and descriptive statistics for each field.

🛛 Data Au	dit of [42 fields]								
📦 <u>F</u> ile 🛛 📑	🛓 Edit 🛛 🕙 Generate								0
Audit Qual	ity Annotations								
Field 📼	Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
🔆 region		💑 Nominal	1	3				з	1000
🔆 tenure		🛷 Continuous	1	72	35.526	21.360	0.112		1000
🔷 age		🛷 Continuous	18	77	41.684	12.559	0.357		1000
🔆 marital		🎖 Flag	0	1				2	1000
🔷 address		🔗 Continuous	0	55	11.551	10.087	1.106		1000
🋞 income		🔗 Continuous	9.000	1668.000	77.535	107.044	6.643		1000
Indicates a r	nuttimode result ² Indi	icates a sampled result							
									O

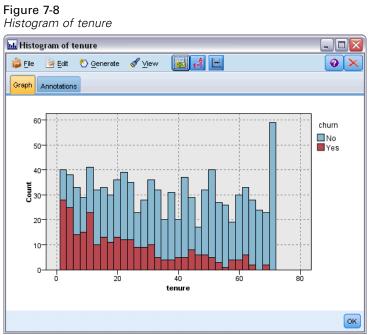
Figure 7-6

Use the toolbar to display field and value labels, and to toggle the alignment of charts from horizontal to vertical (for categorical fields only).

▶ You can also use the toolbar or Edit menu to choose the statistics to display.

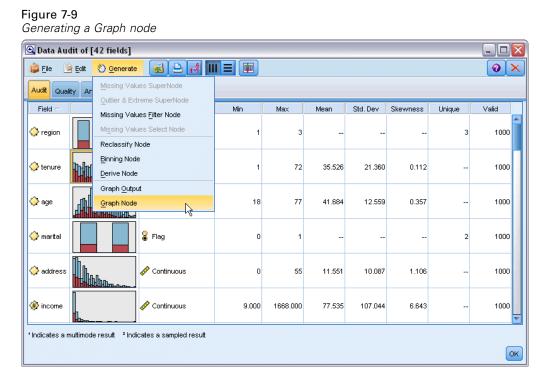
🔍 D	🔍 Display Statistics 🛛 🛛 🔀						
	Statistic						
-	Min						
-	Max						
	Sum						
	Range						
-	Mean						
	Mean Std. Err.						
-	Standard deviation						
	Variance						
-	Skewness						
	Skewness Std. Err.						
	Kurtosis						
	Kurtosis Std. Err.						
-	Unique						
-	Valid						

Double-click on any thumbnail graph in the audit report to view a full-sized version of that chart. Because *churn* is the only target field in the stream, it is automatically used as an overlay. You can toggle the display of field and value labels using the graph window toolbar, or click the Edit mode button to further customize the chart.



78

Alternatively, you can select one or more thumbnails and generate a Graph node for each. The generated nodes are placed on the stream canvas and can be added to the stream to re-create that particular graph.



Handling Outliers and Missing Values

The Quality tab in the audit report displays information about outliers, extremes, and missing values.

Figure 7-10 Data Audit browser, Quality tab

👂 File 🛛 📄 Edi	t 🕙 <u>G</u> enerate					0
udit Quality	Annotations					
omplete fields (9	6): 90.476190 Com	iplete records (%):	13.1			
Field 📼	Measurement	Outliers	Extremes	Action	Impute Missing	Method
> region	💑 Nominal				Never	Fixed
> tenure	🔗 Continuous	0	0	None	Never	Fixed
> age	🔗 Continuous	0	0	None	Never	Fixed
> marital	🎖 Flag				Never	Fixed
> address	🔗 Continuous	12	0	None	Never	Fixed
🖗 income	🔗 Continuous	9	6	None	Never	Fixed
> ed	🚮 Ordinal				Never	Fixed
> employ	🔗 Continuous	8	0	None	Never	Fixed
👂 retire	💑 Nominal				Never	Fixed
🔉 gender	💑 Nominal				Never	Fixed
> reside	🚮 Ordinal				Never	Fixed
> tollfree	🎖 Flag				Never	Fixed
ݤ equip	🎖 Flag 🎖 Flag				Never	Fixed
ݤ callcard					Never	Fixed
> wireless	🎖 Flag				Never	Fixed
🚯 longmon	🔗 Continuous	18	4	None	Never	Fixed
🖗 tollmon	🔗 Continuous	9	1	None	Never	Fixed
🖗 equipmon	🔗 Continuous	2	0	None	Never	Fixed
cardmon 🚯	🖉 Continuous	11	3	None	Never	Fixed

80

You can also specify methods for handling these values and generate SuperNodes to automatically apply the transformations. For example you can select one or more fields and choose to impute or replace missing values for these fields using a number of methods, including the C&RT algorithm.

Figure 7-11 Choosing an impute method

🕽 Eile 🛛 📄 Eo	lit 🕙 Generate						
udit Quality	Annotations						
omplete fields (%): 90.47619(Con	nplete records (%):	13.1				
Field 🗂	Measurement	Outliers	Extremes	Action	Impute Missing	Method	%
> region	💑 Nominal				Never	Fixed	
> tenure	🔗 Continuous	0	0	None	Never	Fixed	
age	🔗 Continuous	0	0	None	Never	Fixed	
> marital	🎖 Flag				Blank & Null Values	Fixed 💌	
address	🔗 Continuous	12	0	None	Never	Fixed	
income	🔗 Continuous	9	6	None	Never	Random	
) ed	📶 Ordinal				Never	Expression	
employ	🔗 Continuous	8	0	None	Never	Algorithm N	
> retire	🂑 Nominal				Never	Specify	
👌 gender	💑 Nominal				Never	Fixed	
k reside	📶 Ordinal				Never	Fixed	
tollfree	🎖 Flag				Never	Fixed	
🕨 equip	🎖 Flag				Never	Fixed	
callcard	🎖 Flag				Never	Fixed	
> wireless	🎖 Flag				Never	Fixed	
longmon	🖉 Continuous	18	4	None	Never	Fixed	
tollmon	🖉 Continuous	9	1	None	Never	Fixed	
👌 equipmon	🔗 Continuous	2	0	None	Never	Fixed	
cardmon	🔗 Continuous	11	3	None	Never	Fixed	
							•

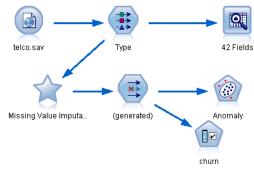
After specifying an impute method for one or more fields, to generate a Missing Values SuperNode, from the menus choose:

Generate > Missing Values SuperNode

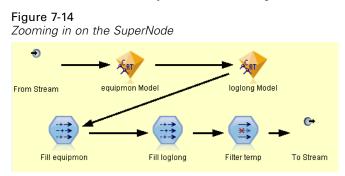
Data Audit of	f [42 fields] #2	2				
🍃 <u>F</u> ile 🛛 📄 Edit	🕙 <u>G</u> enerate					0
Audit Quality		es SuperNode eme SuperNode				
Complete fields (%): Missing Value	es <u>F</u> ilter Node	1			
Field -	Mi <u>s</u> sing Value	es Select Node	xtremes	Action	Impute Missing	Method
region	Reclassify No	ode		-	Never	Fixed
tenure			10	lone	Never	Fixed
age .	Binning Node		10	lone	Never	Fixed
marital	Derive Node			-	Blank & Null Values	Fixed 🔻
address ,			10	lone	Never	Fixed
income	Graph Output		18	lone	Never	Fixed
ed ,	Graph Node			-	Never	Fixed
employ	Continuous	8	10	lone	Never	Fixed
retire (Nominal			-	Never	Fixed
gender	💑 Nominal			-	Never	Fixed
reside	Ordinal			-	Never	Fixed
tollfree	🎖 Flag			-	Never	Fixed
equip	🎖 Flag			-	Never	Fixed
callcard	🖁 Flag			-	Never	Fixed
> wireless	🖁 Flag			-	Never	Fixed
longmon .	Continuous	18	4 1	lone	Never	Fixed
tollmon .	🖉 Continuous	9	11	lone	Never	Fixed
equipmon	🖉 Continuous	2	10	lone	Never	Fixed
cardmon	Continuous	11	31	lone	Never	Fixed

The generated SuperNode is added to the stream canvas, where you can attach it to the stream to apply the transformations.

Figure 7-13 Stream with Missing Values SuperNode



The SuperNode actually contains a series of nodes that perform the requested transformations. To understand how it works, you can edit the SuperNode and click Zoom In.



For each field imputed using the algorithm method, for example, there will be a separate C&RT model, along with a Filler node that replaces blanks and nulls with the value predicted by the model. You can add, edit, or remove specific nodes within the SuperNode to further customize the behavior.

Alternatively, you can generate a Select or Filter node to remove fields or records with missing values. For example, you can filter any fields with a quality percentage below a specified threshold.

Figure 7-15 Generating a Filter node

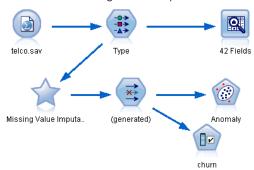


Outliers and extreme values can be handled in a similar manner. Specify the action you want to take for each field—either coerce, discard, or nullify—and generate a SuperNode to apply the transformations.

Data Audit o	f [4	12 fields] #4					
🍃 File 🛛 📄 Edil	t	🕙 <u>G</u> enerate					0
Audit Quality	Ar	Missing Value	s SuperNode				
	AI	Outlier & Extre	me SuperNode				
omplete fields (%	ຄ	— Missing Value	2	\$ 1			
	1		-				
Field		Missing Value	s Select Node	xtremes	Action	Impute Missing	Method
> region	6	Reclassify No	de			Never	Fixed
tenure	1			0	None	Never	Fixed
> age	1	Binning Node		0	None	Never	Fixed
> marital	8	Derive Node				Never	Fixed
address	1	Oversle Overset		0	Coerce	Blank & Null Val	Fixed
🖗 income	1	Graph Output		6	None	Never	Fixed
> ed	-	<u>G</u> raph Node				Never	Fixed
> employ	Ø	Continuous	8	0	None	Never	Fixed
🖗 retire	*	Nominal				Never	Fixed
> gender	h	Nominal				Never	Fixed
> reside	-	Ordinal				Never	Fixed
决 tollfree	8	Flag				Never	Fixed
决 equip		Flag				Never	Fixed
决 callcard		Flag				Never	Fixed
🔉 wireless	8	Flag				Never	Fixed
🚯 longmon	Þ	Continuous	18	4	None	Never	Fixed
🚯 tollmon	A	Continuous	9	1	None	Never	Fixed
🚯 equipmon	Þ	Continuous	2	0	None	Never	Fixed
🚯 cardmon	s.	Continuous	11	3	None	Never	Fixed
🚯 wiremon	R	Continuous	8	1	None	Never	Fixed
🚯 longten	A	Continuous	20	4	None	Never	Fixed
🖗 toliten	R	Continuous	18	2	None	Never	Fixed
🖗 equipten	R	Continuous	16	3	None	Never	Fixed
🚯 cardten	A	Continuous	11	6	None	Never	Fixed
🚯 wireten	A	Continuous	22	3	None	Never	Fixed
🔆 multline	8	Flag				Never	Fixed

After completing the audit and adding the generated nodes to the stream, you can proceed with your analysis. Optionally, you may want to further screen your data using Anomaly Detection, Feature Selection, or a number of other methods.

Figure 7-17 Stream with Missing Values SuperNode



Drug Treatments (Exploratory Graphs/C5.0)

For this section, imagine that you are a medical researcher compiling data for a study. You have collected data about a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of five medications. Part of your job is to use data mining to find out which drug might be appropriate for a future patient with the same illness.

This example uses the stream named *druglearn.str*, which references the data file named *DRUG1n*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *druglearn.str* file is in the *streams* directory.

Data field	Description
Age	(Number)
Sex	M or F
BP	Blood pressure: HIGH, NORMAL, or LOW
Cholesterol	Blood cholesterol: NORMAL or HIGH
Na	Blood sodium concentration
K	Blood potassium concentration
Drug	Prescription drug to which a patient responded

The data fields used in the demo are:

Reading in Text Data

You can read in delimited text data using a **Variable File node**. You can add a Variable File node from the palettes—either click the Sources tab to find the node or use the Favorites tab, which includes this node by default. Next, double-click the newly placed node to open its dialog box.

Drug Treatments (Exploratory Graphs/C5.0)



Click the button just to the right of the File box marked with an ellipsis (...) to browse to the directory in which IBM® SPSS® Modeler is installed on your system. Open the *Demos* directory and select the file called *DRUG1n*.

Ensuring that Read field names from file is selected, notice the fields and values that have just been loaded into the dialog box.

Figure 8-2 Variable File dialog box
🔽 Var. File 🛛 🔀
Preview 2 Refresh
\$CLEO_DEMOS\DRUG1n
File Data Filter Types Annotations
File: \$CLEO_DEMOS\DRUG1n
Age, Sex, BP, Cholesterol, Na, K, Drug 23, F, HIGH, HIGH, 0. 792535, 0. 031258, drugY 47, M, LOW, HIGH, 0. 739309, 0. 056468, drugC 47, M, LOW, HIGH, 0. 697269, 0. 068944, drugC
Read field names from file Specify number of fields 1 Skip header characters: 0 EOL comment characters: Strip lead and trail spaces: Image: None Image: Comment Characters image: Comm
Invalid characters: O Discard O Replace with Encoding: Stream default Decimal symbol: Stream default
Delimiters Lines to scan for type: 50 = Space Comma Tab Automatically recognize dates and times
Newline Other Non-printing characters Single quotes: Allow multiple blank delimiters Double quotes:
OK Cancel Apply Reset

Drug Treatments (Exploratory Graphs/C5.0)



Figure 8-3 Changing the storage type for a field

🙀 DRUG1 n			
Preview 2 Refres	sh		0 - 🗖
SCLEO_DEMOS/DRUG1	n		
File Data Filter Types Annotat	ions		
Field -	Override	Storage	Input Format
Age		🔆 Integer	
Sex		A String	
BP		A String	
Cholesterol Na		🔥 String 💌	
K		(Unknown)	
Drug			
eray	trand	V integer	
		🛞 Real	
		🕒 Time	
		Date	
		💽 Timestamp	
I view current fields ○ View un	used field setting	8	
OK Cancel			Apply Reset

Figure 8-4

Selecting Value options on the Types tab

😡 DRUG1 n					
	view 😰 Refresh				
	DEMOS/DRUG1n				
File Data Filter	Types Annotations				
₹ • ∞ ∞	🕨 Read Values	Clear Values	Clear All	Values	
Field -	Measurement	Values	Missing	Check	Role
🔆 Age	🔗 Continuous	[15,74]		None	🔪 Input
A Sex	🎖 Flag	M/F		None	🔪 Input
A BP	💑 Nominal	HIGH,LOW,		None	🔪 Input
A Cholesterol	🎖 Flag	NORMAL/HI	Off 🔷 🔻	None	🔪 Input
🛞 Na	🔗 Continuous	[0.500169,0	On (*)	None	🔪 Input
🛞 К	🔗 Continuous	[0.020022,0		None	🔪 Input
A Drug	💑 Nominal	drugA,drug	Specify	None	🔪 Input
View current field	elds 🛛 View unused fiel	d settings			
OK Cancel					Apply Reset

Click the Data tab to override and change **Storage** for a field. Note that storage is different from **Measurement**, that is, the measurement level (or usage type) of the data field. The Types tab helps you learn more about the type of fields in your data. You can also choose Read Values to view the actual values for each field based on the selections that you make from the *Values* column. This process is known as **instantiation**.

Adding a Table

Now that you have loaded the data file, you may want to glance at the values for some of the records. One way to do this is by building a stream that includes a Table node. To place a Table node in the stream, either double-click the icon in the palette or drag and drop it on to the canvas.

Figure 8-5 Table node connected to the data source



Drug Treatments (Exploratory Graphs/C5.0)

Figure 8-6

ile Edit Insert Y	view <u>T</u> ools Supe	erNode Wir	ndow			#1 🗾] ડિવા		-	**
								Ru	n the curr	ent stream
		Tabl	e (7 f	ields	. 200 ree	cords) #2				
		😺 <u>F</u> ile		Edit	🕙 <u>G</u> ener	rate 🔡				() ×
		Table	0l							
\frown		Table	Annote	ations						
(副) —			Age	Sex	BP	Cholesterol	Na	к	Drug	
		1	23	F	HIGH	HIGH	0.793	0.031	drugY	4
		2	47	M	LOW	HIGH	0.739	0.056	drugC	
DRUG1n	Table	3	47	M	LOW	HIGH	0.697	0.069	drugC	
		4	28	F	NORMAL	HIGH	0.564	0.072	drugX	
		5	61	F	LOW	HIGH	0.559	0.031	drugY	
		6	22	F	NORMAL	HIGH	0.677	0.079	drugX	
		7	49	F	NORMAL	HIGH	0.790	0.049	drugY	
		8	41	M	LOW	HIGH	0.767	0.069	drugC	
		9	60	M	NORMAL	HIGH	0.777	0.051	drugY	
		10	43	M	LOW	NORMAL	0.526	0.027	drugY	
		11	47	F	LOW	HIGH	0.896	0.076	drugC	
		12	34	F	HIGH	NORMAL	0.668	0.035	drugY	
		13	43	M	LOW	HIGH	0.627	0.041	drugY	
		14	74	F	LOW	HIGH	0.793	0.038	drugY	
		15	50	F	NORMAL	HIGH	0.828	0.065	drugX	
		16	16	F	HIGH	NORMAL	0.834	0.054	drugY	
		17	69	м	LOW	NORMAL	0.849	0.074	drugX	
		18	43	м	HIGH	HIGH	0.656	0.047	drugA	
		19	23	М	LOW	HIGH	0.559	0.077	drugC	
		20	32	F	HIGH	NORMAL	0.643	0.025	drugY	

Running a stream from the toolbar

Double-clicking a node from the palette will automatically connect it to the selected node in the stream canvas. Alternatively, if the nodes are not already connected, you can use your middle mouse button to connect the Source node to the Table node. To simulate a middle mouse button, hold down the Alt key while using the mouse. To view the table, click the green arrow button on the toolbar to run the stream, or right-click the Table node and choose Run.

Creating a Distribution Graph

During data mining, it is often useful to explore the data by creating visual summaries. IBM® SPSS® Modeler offers several different types of graphs to choose from, depending on the kind of data that you want to summarize. For example, to find out what proportion of the patients responded to each drug, use a Distribution node.

Add a Distribution node to the stream and connect it to the Source node, then double-click the node to edit options for display.

Select *Drug* as the target field whose distribution you want to show. Then, click Run from the dialog box.

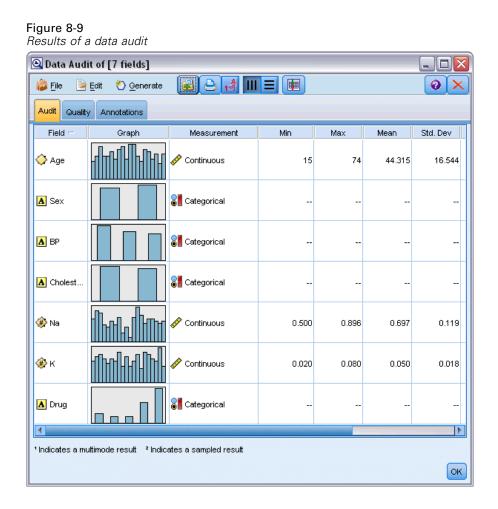
Figure 8-7 Selecting drug as the target field	
🚰 Drug	×
Field: Drug	כ
Plot Appearance Output Annotations	
Plot: O Selected fields O All flags (true values)	
Field: 🖁 Drug	Ð
Overla Natural Name Type Type Color: Color: (none) Sex BP Cholesterol	
Sort: Broportional scale Drug	
OK Run Cancel Apply Res	et

The resulting graph helps you see the "shape" of the data. It shows that patients responded to drug Y most often and to drugs B and C least often.

Figure 8-8 Distribution of response to drug type

崖 Distributio	n of Drug #1		
🐞 File 🛛 🖹 E	-	8	0×
Table Graph	Annotations		
Value 🗠	Proportion	%	Count
drugA		11.5	23
drugB		8.0	16
drugC		8.0	16
drugX		27.0	54
drugY		45.5	91
			ок

Drug Treatments (Exploratory Graphs/C5.0)



Alternatively, you can attach and execute a Data Audit node for a quick glance at distributions and histograms for all fields at once. The Data Audit node is available on the Output tab.

Creating a Scatterplot

Now let's take a look at what factors might influence *Drug*, the target variable. As a researcher, you know that the concentrations of sodium and potassium in the blood are important factors. Since these are both numeric values, you can create a scatterplot of sodium versus potassium, using the drug categories as a color overlay.

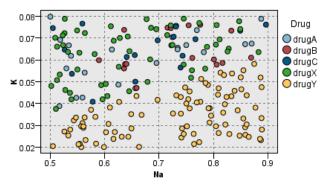
Place a Plot node in the workspace and connect it to the Source node, and double-click to edit the node.

On the Plot tab, select *Na* as the X field, *K* as the Y field, and *Drug* as the overlay field. Then, click Run.



The plot clearly shows a threshold above which the correct drug is always drug Y and below which the correct drug is never drug Y. This threshold is a ratio—the ratio of sodium (*Na*) to potassium (*K*).

Figure 8-11 Scatterplot of drug distribution

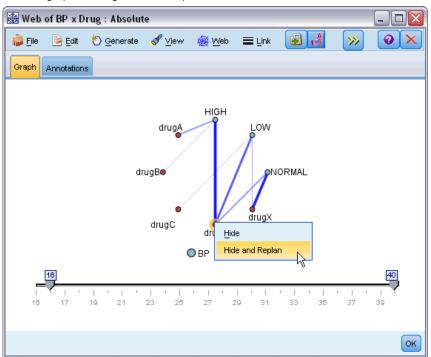


Creating a Web Graph

Since many of the data fields are categorical, you can also try plotting a web graph, which maps associations between different categories. Start by connecting a Web node to the Source node in your workspace. In the Web node dialog box, select *BP* (for blood pressure) and *Drug*. Then, click Run.

From the plot, it appears that drug Y is associated with all three levels of blood pressure. This is no surprise—you have already determined the situation in which drug Y is best. To focus on the other drugs, you can hide drug Y. On the View menu, choose Edit Mode, then right-click over the drug Y point and choose Hide and Replan.

Figure 8-12 Web graph of drugs vs. blood pressure



In the simplified plot, drug Y and all of its links are hidden. Now, you can clearly see that only drugs A and B are associated with high blood pressure. Only drugs C and X are associated with low blood pressure. And normal blood pressure is associated only with drug X. At this point,

though, you still don't know how to choose between drugs A and B or between drugs C and X, for a given patient. This is where modeling can help.

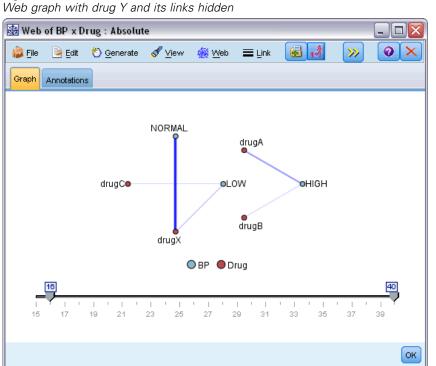


Figure 8-13

Deriving a New Field

Since the ratio of sodium to potassium seems to predict when to use drug *Y*, you can derive a field that contains the value of this ratio for each record. This field might be useful later when you build a model to predict when to use each of the five drugs. To simplify the stream layout, start by deleting all the nodes except the DRUG1n source node. Attach a Derive node (Field Ops tab) to DRUG1n, then double-click the Derive node to edit it.

Drug Treatments (Exploratory Graphs/C5.0)

Figure 8-14 Editing the Derive node

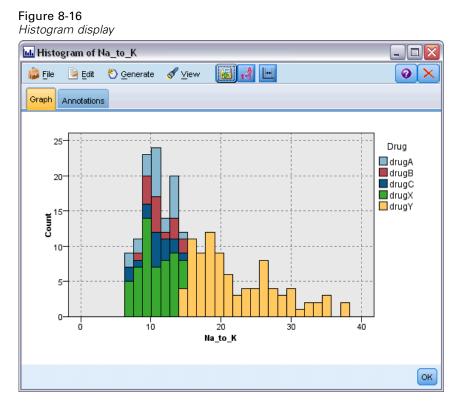
Derive	
	0
Derive as: Formula	
Settings Annotations	
Mode: 💿 Single 🔘 Multiple	
Derive field:	
Na_to_K	
Derive as: Formula	
Field type: 🧳 <default> 🤝</default>	
Field type: 🖌 <default> 🔽</default>	
Formula:	
Na/K	
]
OK Cancel	Apply Reset

Name the new field *Na_to_K*. Since you obtain the new field by dividing the sodium value by the potassium value, enter Na/K for the formula. You can also create a formula by clicking the icon just to the right of the field. This opens the Expression Builder, a way to interactively create expressions using built-in lists of functions, operands, and fields and their values.

You can check the distribution of your new field by attaching a Histogram node to the Derive node. In the Histogram node dialog box, specify *Na_to_K* as the field to be plotted and *Drug* as the overlay field.



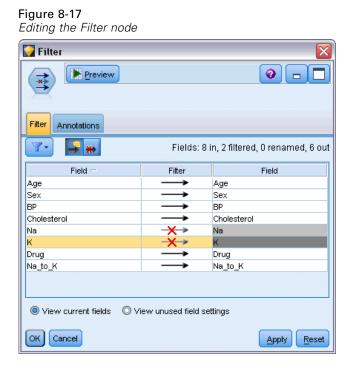
When you run the stream, you get the graph shown here. Based on the display, you can conclude that when the Na_to_K value is about 15 or above, drug Y is the drug of choice.



Building a Model

By exploring and manipulating the data, you have been able to form some hypotheses. The ratio of sodium to potassium in the blood seems to affect the choice of drug, as does blood pressure. But you cannot fully explain all of the relationships yet. This is where modeling will likely provide some answers. In this case, you will use try to fit the data using a rule-building model, C5.0.

Since you are using a derived field, *Na_to_K*, you can filter out the original fields, *Na* and *K*, so that they are not used twice in the modeling algorithm. You can do this using a Filter node.



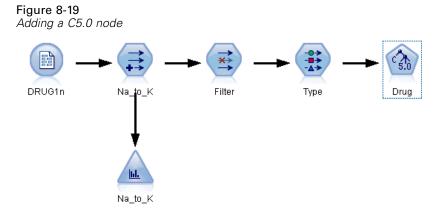
On the Filter tab, click the arrows next to *Na* and *K*. Red Xs appear over the arrows to indicate that the fields are now filtered out.

Next, attach a Type node connected to the Filter node. The Type node allows you to indicate the types of fields that you are using and how they are used to predict the outcomes.

On the Types tab, set the role for the *Drug* field to Target, indicating that *Drug* is the field you want to predict. Leave the role for the other fields set to Input so they will be used as predictors.

Age Age Continuous [15,74] None In A Sex Flag M/F None In A BP Nominal HIGH,LO None In A Cholesterol Flag Norminal drugA,dru None In None In None In A Drug Nominal drugA,dru None In None In Drug Nominal In Drug Nominal In Drug Nominal In Drug None In Drug Nome In Drug Nominal In Drug Nominal In Drug None In	
Age Continuous [15,74] None In A Sex Flag M/F None In A BP Nominal HIGH,LO None In A Cholesterol Flag NORMAL/ None In A Drug Nominal drugA,dru None In None In None In None In P Na_to_K Continuous [6.268724	
A Sex Flag M/F None In A BP Nominal HIGH,LO None In A Cholesterol Flag NORMAL/ None In A Drug Nominal drugA,dru None In A Drug Continuous [6.268724] None In Image: Sex of the s	Role
ABP Norminal HIGH,LO None In A Cholesterol Flag NORMAL/ None In A Drug Nominal drugA,dru None In Wa_to_K Continuous [6.268724] None In	nput
A BP Nominal HIGH,LO None In A Cholesterol Flag NORMAL/ None In A Drug Nominal drugA,dru None In A Drug Orninal drugA,dru None In Image: A state of the state of th	nput
A Drug 💑 Nominal drugA,dru None 🕥 Ir ④ Na_to_K 🖉 Continuous [6.268724 None 🕥 Ir ⑥ T ⑧ B	
🛞 Na_to_K 🖉 Continuous [6.268724 None 🕥 In 🎯 T 🕲 B	nput
(©т))© в	nput 🔻
🐌 в	nput
Harry Kang, Kang	arget
	Both
○ N	lone
View current fields O View unused field settings	artition
	Split
OK Cancel	

To estimate the model, place a C5.0 node in the workspace and attach it to the end of the stream as shown. Then click the green Run toolbar button to run the stream.



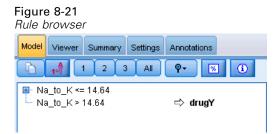
Browsing the Model

Figure 8-18

When the C5.0 node is executed, the model nugget is added to the stream, and also to the Models palette in the upper-right corner of the window. To browse the model, right-click either of the icons and choose Edit or Browse from the context menu.

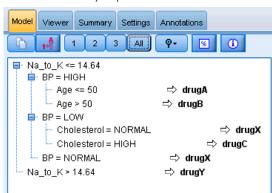
Figure 8-20 Browsing the model			
	Strea	ims Outputs Models	
Add <u>T</u> o Stream			
Browse	N	9	
Rename and Annotate	43		
🏷 Generate Modeling Node	e		
Save Model		-	
Save Model As			
饕 Store Model			
Export PMML			
Add to Project			
	Delete		

The Rule browser displays the set of rules generated by the C5.0 node in a decision tree format. Initially, the tree is collapsed. To expand it, click the All button to show all levels.



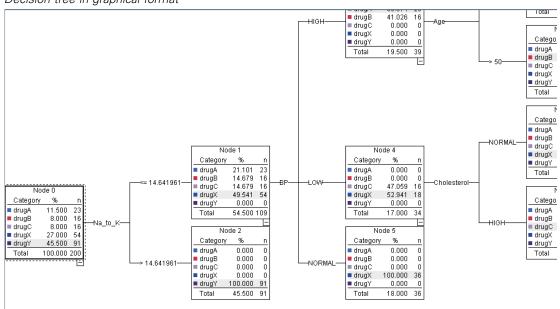
Now you can see the missing pieces of the puzzle. For people with an *Na*-to-*K* ratio less than 14.64 and high blood pressure, age determines the choice of drug. For people with low blood pressure, cholesterol level seems to be the best predictor.

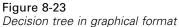
Figure 8-22 Rule browser fully expanded



100

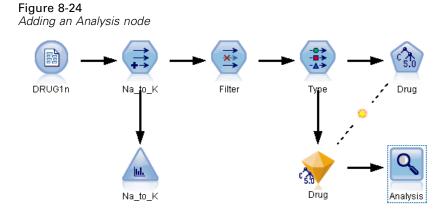
The same decision tree can be viewed in a more sophisticated graphical format by clicking the Viewer tab. Here, you can see more easily the number of cases for each blood pressure category, as well as the percentage of cases.





Using an Analysis Node

You can assess the accuracy of the model using an analysis node. Attach an Analysis node (from the Output node palette) to the model nugget, open the Analysis node and click Run.



The Analysis node output shows that with this artificial dataset, the model correctly predicted the choice of drug for every record in the dataset. With a real dataset you are unlikely to see 100% accuracy, but you can use the Analysis node to help determine whether the model is acceptably accurate for your particular application.

Figure 8-25
Analysis node output

🔦 Analysis of [Drug]	_				
🐞 File 🎅 Edit 🔃		0 ×			
Analysis Annotations					
Collapse All 🖗 Expand All					
■-Results for output field	Drug				
😑 Comparing \$C-Drug	g with Drug				
Correct	200 100%				
Wrong Total	0 0%				
Total	200				
		ок			

Screening Predictors (Feature Selection)

The Feature Selection node helps you to identify the fields that are most important in predicting a certain outcome. From a set of hundreds or even thousands of predictors, the Feature Selection node screens, ranks, and selects the predictors that may be most important. Ultimately, you may end up with a quicker, more efficient model—one that uses fewer predictors, executes more quickly, and may be easier to understand.

The data used in this example represent a data warehouse for a hypothetical telephone company and contain information about responses to a special promotion by 5,000 of the company's customers. The data include a large number of fields containing customers' age, employment, income, and telephone usage statistics. Three "target" fields show whether or not the customer responded to each of three offers. The company wants to use this data to help predict which customers are most likely to respond to similar offers in the future.

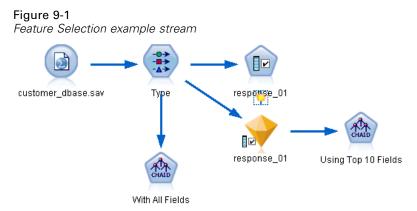
This example uses the stream named *featureselection.str*, which references the data file named *customer_dbase.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *featureselection.str* file is in the *streams* directory.

This example focuses on only one of the offers as a target. It uses the CHAID tree-building node to develop a model to describe which customers are most likely to respond to the promotion. It contrasts two approaches:

- Without feature selection. All predictor fields in the dataset are used as inputs to the CHAID tree.
- With feature selection. The Feature Selection node is used to select the top 10 predictors. These are then input into the CHAID tree.

By comparing the two resulting tree models, we can see how feature selection produces effective results.

Building the Stream



- Place a Statistics File source node onto a blank stream canvas. Point this node to the example data file *customer_dbase.sav*, available in the *Demos* directory under your IBM® SPSS® Modeler installation. (Alternatively, open the example stream file *featureselection.str* in the *streams* directory.)
- ► Add a Type node. On the Types tab, scroll down to the bottom and change the role for response_01 to Target. Change the role to None for the other response fields (response_02 and response_03) as well as for the customer ID (custid) at the top of the list. Leave the role set to Input for all other fields, and click the Read Values button, then click OK.
- ► Add a Feature Selection modeling node to the stream. On this node, you can specify the rules and criteria for screening, or disqualifying, fields.
- ▶ Run the stream to create the Feature Selection model nugget.

Screening Predictors (Feature Selection)

 Right-click the model nugget on the stream or in the Models palette and choose Edit or Browse to look at the results.

		ature Selectic					- fe
resp	onse_01						
	🐞 <u>F</u> ile	🕙 <u>G</u> enerate 🏾	🕨 <u>P</u> r	eview) 🔛		0	
Aodel	Summary	Annotations					
		Rank 🔻 🔺		14			
	Rank 🗠	Field	I	Measurement	Importance	Value	
-	1 <	🔆 ed	🖉 Ca	ontinuous	📩 Important	1.0	
-	2 <	🔆 ownpc	6.5	ominal	🚼 Important	1.0	
-	3 <	🔆 edcat	1 0	rdinal	🚼 Important	1.0	
-	4 <	🔅 internet	💑 No	ominal	📩 Important	1.0	
-	5 <	🔆 equip	💑 No	ominal	★ Important	1.0	
\checkmark	6 <	🔆 owngame	💑 No	ominal	★ Important	1.0	
\checkmark	7 🤇	🋞 equipmon	🔗 Co	ontinuous	📩 Important	1.0	
\checkmark	8 <	🔆 confer	💑 No	ominal	📩 Important	1.0	
\checkmark	9 <	칮 ebill	💑 No	ominal	📩 Important	1.0	
\checkmark	10 <	决 callwait	💑 No	ominal	📩 Important	1.0	
\checkmark	11 📢	💭 forward	💑 No	ominal	📩 Important	1.0	
\checkmark	12 📢	🋞 tollmon	🔗 Co	ontinuous	📩 Important	1.0	
\checkmark	13 <	🔆 multline	💑 No	ominal	📩 Important	1.0	
\checkmark	14 <	💭 ownipod	💑 No	ominal	📩 Important	1.0	
\checkmark	15 📢	칮 callid	💑 No	ominal	📩 Important	1.0	
\checkmark	16 <	🛞 equipten	🖉 Ca	ontinuous	📩 Important	1.0	
\checkmark		ؼ tollfree	💑 No	ominal	📩 Important	1.0	
\checkmark	18 <	🛞 tollten	🖉 Co	ontinuous	📩 Important	1.0	
\checkmark	19 📢	💭 churn	💑 No	ominal	📩 Important	1.0	
1	20 (🔿 snousedcat		rdinal	🛨 Important	10	-
Selecte	ed fields: 34	Total fields availa					
		★ > 0.9	5 🕂	<= 0.95 • < 0.	9		
			9 Scre	ened Fields			
	Field 🔻	Measuremer	nt		Reason		
	🔆 ownvcr	💑 Nominal		Single category to	o large		4
	🔷 owntv 🛛	💑 Nominal		Single category to	o large		
	🔆 owndvd	💑 Nominal		Single category to	o large		
	🔆 owned	💑 Nominal		Single category to	o large		
	🛞 Inwiretei			Too many missing	values		
	🛞 Inwirem.			Too many missing	values		
1000	🛞 Inequip	. 🔊 Continuous		Coefficient of varia	ation below thre	shold	T

Figure 9-2

The top panel shows the fields found to be useful in the prediction. These are ranked based on importance. The bottom panel shows which fields were screened from the analysis and why. By examining the fields in the top panel, you can decide which ones to use in subsequent modeling sessions.

- ► Now we can select the fields to use downstream. Although 34 fields were originally identified as important, we want to reduce the set of predictors even further.
- ► Select only the top 10 predictors using the check marks in the first column to deselect the unwanted predictors. (Click the check mark in row 11, hold down the Shift key and click the check mark in row 34.) Close the model nugget.

- ► To compare results without feature selection, you must add two CHAID modeling nodes to the stream: one that uses feature selection and one that does not.
- Connect one CHAID node to the Type node, and the other one to the Feature Selection model nugget.
- Open each CHAID node, select the Build Options tab and ensure that the options Build new model, Build a single tree and Launch interactive session are selected in the Objectives pane.

On the Basics pane, make sure that Maximum Tree Depth is set to 5.

Figure 9-3

🖓 With All Fields 🛛 🛛 🔀					
CHAID					
Fields Build Optic	Model Options Annotations				
Objective	What do you want to do?				
Basics	I guild new model O Continue training existing model				
Stopping Rules	What is your main objective?				
Costs	Buil <u>d</u> a single tree				
Ensembles	Single Tree				
Advanced	Mode: O Generate model O Launch interactive session				
	Use tree directives				
	C Enhance model accuracy (boosting)				
	\bigcirc Enhance model stability (bagging)				
	O Create a model for very large datasets (requires Server)				
	Description				
	Creates a single, standard model to explain relationships between fields. Standard models are easier to interpret and can be faster to score than boosted, bagged, or large dataset ensembles.				
OK 🕨 Run	Cancel Apply Reset				

Objectives settings for CHAID modeling node for all predictor fields

Building the Models

Execute the CHAID node that uses all of the predictors in the dataset (the one connected to the Type node). As it runs, notice how long it takes to execute. The results window displays a table.

▶ From the menus, choose Tree > Grow Tree to grow and display the expanded tree.

Growing the tree in the Tree Builder	
The active Tree of CHAID #6	
🙀 Eile 📄 Edit 💰 View Iree 🖏 Generate 🕘 🕒 📢	X
Viewer Gains Risks Annotations	
	Î.
response_01	
Node 0	
Category % n	
0.000 91.640 4582 1.000 8.360 418	
Total 100.000 5000	
ownpc	
Adj. P-value=0.000, Chi-square=57.452, df=1	
	-
E S S S S S S S S S S S S S S S S S S S	
	ОК
	UN

Figure 9-4 *Growing the tree in the Tree Builder*

▶ Now do the same for the other CHAID node, which uses only 10 predictors. Again, grow the tree when the Tree Builder opens.

The second model should have executed faster than the first one. Because this dataset is fairly small, the difference in execution times is probably a few seconds; but for larger real-world datasets, the difference may be very noticeable—minutes or even hours. Using feature selection may speed up your processing times dramatically.

The second tree also contains fewer tree nodes than the first. It is easier to comprehend. But before you decide to use it, you need to find out whether it is effective and how it compares to the model that uses all predictors.

Comparing the Results

To compare the two results, we need a measure of effectiveness. For this, we will use the Gains tab in the Tree Builder. We will look at **lift**, which measures how much more likely the records in a node are to fall under the target category when compared to all records in the dataset. For example, a lift value of 148% indicates that records in the node are 1.48 times more likely

Figure 9-5

to fall under the target category than all records in the dataset. Lift is indicated in the *Index* column on the Gains tab.

- ► In the Tree Builder for the full set of predictors, click the Gains tab. Change the target category to 1.0. Change the display to quartiles by first clicking the Quantiles toolbar button. Then select Quartile from the drop-down list to the right of this button.
- Repeat this procedure in the Tree Builder for the set of 10 predictors so that you have two similar Gains tables to compare, as shown in the following figures.

Y Interactive	Tree of CHAID					
違 File 🛛 📄 Ed	it 🧳 <u>V</u> iew <u>T</u> ree	🕙 Generate 🛛				0
Viewer Gains	Risks Annotations					
74 <u>11</u> 12 (Quartile	▼ Ⅲ ∠ G	ains	👻 🍫 Targ	et category 1.0	- I A 7
Target variable: response_01 Target category: 1.0 Training Sample						
Nodes	Percentile	Percentile: n	Gain: n	Gain (%)	Response (%)	Index (%)
	53,45,49,33 25.00	1250.00	231.00	55.29	18,49	221.17
	41,40,51, 50.00	2500.00	358.00	85.54	14.30	171.09
54,47,32,55,58,19		3750.00	407.00	97.45	10.86	129.94
	,39,35,57, 100.00	5000.00	418.00	100.00	8.36	100.00
		🖔 Generate 🛛 🚺				0
	it ∦ ⊻iew <u>T</u> ree					
Viewer Gains	it <u>Yew</u> <u>ree</u> Risks Annotations Quartile	▼ Ⅲ ∠ 0	ains	▼ <mark>\$%</mark> Tar <u>c</u>	jet category 1.0	•
Viewer Gains	Risks Annotations			▼ \$ Targ 01 Target catego		• I A 7
Viewer Gains	Risks Annotations				ry: 1.0	Index (%)
Viewer Gains	Risks Annotations	Target va	ariable: response_	01 Target catego	ry: 1.0	Index (%)
Viewer Gains	Risks Annotations Quartile Percentile	Target va	ariable: response_ Gain: n	01 Target catego Gain (%)	ry: 1.0 Response (%)	
Viewer Gains	Risks Annotations Quartile Percentile 25.00	Target va Percentile: n 1250.00	ariable: response_ Gain: n 203.00	01 Target catego Gain (%) 48.45	ry: 1.0 Response (%) 16.20	193.81
Viewer Gains	Risks Annotations Quartile Percentile 25.00 50.00	Target v: Percentile: n 1250.00 2500.00	ariable: response_ Gain: n 203.00 308.00	01 Target catego Gain (%) 48.45 73.57	ry: 1.0 Response (%) 16.20 12.30	193.81 147.14

Each Gains table groups the terminal nodes for its tree into quartiles. To compare the effectiveness of the two models, look at the lift (*Index* value) for the top quartile in each table.

When all predictors are included, the model shows a lift of 221%. That is, cases with the characteristics in these nodes are 2.2 times more likely to respond to the target promotion. To see what those characteristics are, click to select the top row. Then switch to the Viewer tab, where the corresponding nodes are now outlined in black. Follow the tree down to each highlighted terminal node to see how the predictors were split. The top quartile alone includes 10 nodes. When translated into real-world scoring models, 10 different customer profiles can be difficult to manage.

With only the top 10 predictors (as identified by feature selection) included, the lift is nearly 194%. Although this model is not quite as good as the model that uses all predictors, it is certainly useful. Here, the top quartile includes only four nodes, so it is simpler. Therefore, we can determine that the feature selection model is preferable to the one with all predictors.

Summary

Let's review the advantages of feature selection. Using fewer predictors is less expensive. It means that you have less data to collect, process, and feed into your models. Computing time is improved. In this example, even with the extra feature selection step, model building was noticeably faster with the smaller set of predictors. With a larger real-world dataset, the time savings should be greatly amplified.

Using fewer predictors results in simpler scoring. As the example shows, you might identify only four profiles of customers who are likely to respond to the promotion. Note that with larger numbers of predictors, you run the risk of overfitting your model. The simpler model may generalize better to other datasets (although you would need to test this to be sure).

You could have used a tree-building algorithm to do the feature selection work, allowing the tree to identify the most important predictors for you. In fact, the CHAID algorithm is often used for this purpose, and it is even possible to grow the tree level-by-level to control its depth and complexity. However, the Feature Selection node is faster and easier to use. It ranks all of the predictors in one fast step, allowing you to identify the most important fields quickly. It also allows you to vary the number of predictors to include. You could easily run this example again using the top 15 or 20 predictors instead of 10, comparing the results to determine the optimal model.

Reducing Input Data String Length (Reclassify Node)

Reducing Input Data String Length (Reclassify)

For binomial logistic regression, and auto classifier models that include a binomial logistic regression model, string fields are limited to a maximum of eight characters. Where strings are more than eight characters, they can be recoded using a Reclassify node.

This example uses the stream named *reclassify_strings.str*, which references the data file named *drug_long_name*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *reclassify strings.str* file is in the *streams* directory.

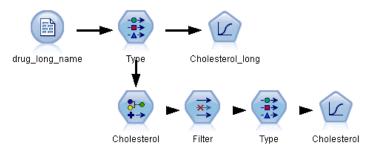
This example focuses on a small part of a stream to show the sort of errors that may be generated with overlong strings and explains how to use the Reclassify node to change the string details to an acceptable length. Although the example uses a binomial Logistic Regression node, it is equally applicable when using the Auto Classifier node to generate a binomial Logistic Regression model.

Reclassifying the Data

▶ Using a Variable File source node, connect to the dataset *drug_long_name* in the *Demos* folder.

Figure 10-1

Sample stream showing string reclassification for binomial logistic regression



- ► Add a Type node to the Source node and select Cholesterol_long as the target.
- ► Add a Logistic Regression node to the Type node.

Reducing Input Data String Length (Reclassify Node)

▶ In the Logistic Regression node, click the Model tab and select the Binomial procedure.

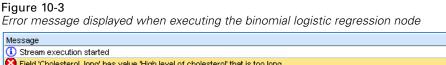
Figure 10-2

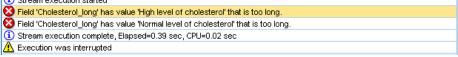
Long string details in the	"Cholesterol_long" field
----------------------------	--------------------------

Type Types Format Annotations					
<u></u>	🗢 🛛 🕨 Read Value:	s 📔 Clear V	'alues	Clear All Values	s
Field -	Measurement	Values	Missing	Check	Role
🚫 Age	🔗 Continuous	[15,74]		None	🔪 Input
A Sex	🎖 Flag	M/F		None	🔪 Input
A BP	💑 Nominal	HIGH,LO		None	🔪 Input
🛞 Na	🔗 Continuous	[0.500517		None	🔪 Input
🛞 К	🔗 Continuous	[0.020152		None	🔪 Input
A Drug	💑 Nominal	drugA,dru		None	🔪 Input
🔥 Cholesterol	🎖 Flag	"Normal le		None	🔘 Target
View current OK Cancel	fields 🔘 View unused	field settings	:	<u>A</u>	pply <u>R</u> eset

► When you execute the Logistic Regression node in *reclassify_strings.str*, an error message is displayed warning that the Cholesterol_long string values are too long.

If you encounter this type of error message, follow the procedure explained in the rest of this example to modify your data.





- ► Add a Reclassify node to the Type node.
- ► In the Reclassify field, select Cholesterol_long.
- ► Type Cholesterol as the new field name.
- Click the Get button to add the Cholesterol_long values to the original value column.

► In the new value column, type High next to the original value of High level of cholesterol and Normal next to the original value of Normal level of cholesterol.

🏑 Chole	esterol			
**	Pr	eview		0
Settings	Annota	ations		
		Mode:	🔘 Single 🔘 Multiple	
		Reclassify into:	New field C Existing field	ld
Reclassif	fy field:			
8 Cho	lesterol	_long		-
New field	d name:			
Choleste	rol			
Choleste		:		
Reclassit		с Хору	Clear new	🗳 Auto
Reclassit	fy values Get		Clear new New value	4 Auto
Reclassit	fy values Get	>> Copy		4 Auto
Reclassif	fy values Get Origina vel of cho	>> Copy	New value	<pre># Auto</pre>
Reclassif	fy values Get Origina vel of cho	>> Copy al value	New value	# Auto
Reclassif	fy values Get Origina vel of cho	>> Copy al value	New value	# Auto
Reclassif	fy values Get Origina vel of cho	>> Copy al value	New value	

• Add a Filter node to the Reclassify node.

Reducing Input Data String Length (Reclassify Node)

▶ In the Filter column, click to remove Cholesterol_long.

Figure 10-6

Figure 10-5 Filtering the "Cholesterol_long" field from the data

💟 Filter		$\overline{\mathbf{X}}$			
Filter Annotations					
7- 두 🗰	Fields: 8	3 in, 1 filtered, 0 renamed, 7 out			
Field -	Filter	Field			
Age	\rightarrow	Age			
Sex	\rightarrow	Sex			
BP	\rightarrow	BP			
Na	\rightarrow	Na			
к	\rightarrow	к			
Drug	\rightarrow	Drug			
Cholesterol_long	— × →	Cholesterol_long			
Cholesterol	\rightarrow	Cholesterol			
Cholesterol Cholesterol View current fields View unused field settings OK Cancel <u>Apply</u> <u>Reset</u>					

► Add a Type node to the Filter node and select Cholesterol as the target.

	eview				
Types Format	Annotations				
₹ -	🍽 🚺 🕨 Read Valu	es 🛛 Clear V	/alues	Clear All Value	5
Field 🗂	Measurement	Values	Missing	Check	Role
🔆 Age	🔗 Continuous	[15,74]		None	🔪 Input
A Sex	🎖 Flag	MÆ		None	🔪 Input
A BP	💑 Nominal	HIGH,LO		None	🔪 Input
🤣 Na	🔗 Continuous	[0.500517		None	🔪 Input
⊛к	🔗 Continuous	[0.020152		None	🔪 Input
A Drug	💑 Nominal	drugA,dru		None	🔪 Input
A Cholesterol	🎖 Flag	Normal/High		None	🎯 Target
View current	fields 🔘 View unuse	ed field settings	:		

Short string details in the "Cholesterol" field

- Add a Logistic Node to the Type node. ►
- ► In the Logistic node, click the Model tab and select the Binomial procedure.

► You can now execute the Binomial Logistic node and generate a model without displaying an error message.

Figure 10-7 Choosing Binomial a	as the procedure		
😡 Cholesterol			X
Fields Model Expert	Analyze Annotations	0	
Model name: 💿 Auto 🛇	Custom		
🔽 Use partitioned data			
🔽 Build model for each spli	t		
Procedure: 🔘 Multinomi	al	Binomial	
Binomial Procedure Method: Enter Categorical Inputs:			
Field Name	Contrast	Base Category	
			×
V Include constant in equa	tion		
OK 🕨 Run Cance		Apply	Reset

This example only shows part of a stream. If you require further information about the types of streams in which you may need to reclassify long strings, the following examples are available:

- Auto Classifier node. For more information, see the topic Modeling Customer Response (Auto Classifier) in Chapter 4 on p. 42.
- Binomial Logistic Regression node. For more information, see the topic Telecommunications Churn (Binomial Logistic Regression) in Chapter 13 on p. 154.

More information on how to use IBM® SPSS® Modeler, such as a user's guide, node reference, and algorithms guide, are available from the *Documentation* directory of the installation disk.

Part III: Modeling Examples

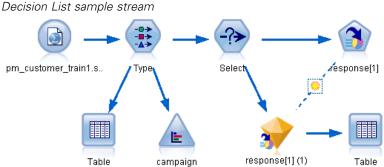
Modeling Customer Response (Decision List)

The Decision List algorithm generates rules that indicate a higher or lower likelihood of a given binary (yes or no) outcome. Decision List models are widely used in customer relationship management, such as call center or marketing applications.

This example is based on a fictional company that wants to achieve more profitable results in future marketing campaigns by matching the right offer to each customer. Specifically, the example uses a Decision List model to identify the characteristics of customers who are most likely to respond favorably, based on previous promotions, and to generate a mailing list based on the results.

Decision List models are particularly well suited to interactive modeling, allowing you to adjust parameters in the model and immediately see the results. For a different approach that allows you to automatically create a number of different models and rank the results, the Auto Classifier node can be used instead.

Figure 11-1



This example uses the stream *pm_decisionlist.str*, which references the data file *pm_customer_train1.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *pm_decisionlist.str* file is in the *streams* directory.

Historical Data

The file *pm_customer_train1.sav* has historical data tracking the offers made to specific customers in past campaigns, as indicated by the value of the *campaign* field. The largest number of records fall under the *Premium account* campaign.

Figure 11-2 Data about previous promotions

🛄 Table	(31 fields,	21,927 record	ds)				_ 🗆 🛛
じ <u>F</u> ile	📄 <u>E</u> dit 🛛 💐) <u>G</u> enerate 🧯		ata			0 ×
Table 🛛	Annotations						
	customer_id	campaign	response	response_date	purchase	purchase_date	product_id I
1	7	Premium account	0	\$null\$	0	\$null\$	\$null\$ 1🗲
2	13	Premium account	0	\$null\$	0	\$null\$	\$null\$ 2
3	15	Premium account	0	\$null\$	0	\$null\$	\$null\$ 3
4	16	Premium account	1	2006-07-05 00:00:00	0	\$null\$	183 7
5	23	Premium account	0	\$null\$	0	\$null\$	\$null\$ 4
6	24	Premium account	0	\$null\$	0	\$null\$	\$null\$ 5
7	30	Premium account	0	\$null\$	0	\$null\$	\$null\$ 6
8	30	Gold card	0	\$null\$	0	\$null\$	\$null\$ 7
9	33	Premium account	0	\$null\$	0	\$null\$	\$null\$ 8
10	42	Gold card	0	\$null\$	0	\$null\$	\$null\$ §
11	42	Premium account	0	\$null\$	0	\$null\$	\$null\$ 1
12	52	Premium account	0	\$null\$	0	\$null\$	\$null\$ 1
13	57	Premium account	0	\$null\$	0	\$null\$	\$null\$ 1
14	63	Premium account	1	2006-07-14 00:00:00	0	\$null\$	183 1
15	74	Premium account	0	\$null\$	0	\$null\$	\$null\$ 1
16	74	Gold card	0	\$null\$	0	\$null\$	\$null\$ 1
17	75	Premium account	0	\$null\$	0	\$null\$	\$null\$ 1
18	82	Premium account	0	\$null\$	0	\$null\$	\$null\$ 1
19	89	Gold card	0	\$null\$	0	\$null\$	\$null\$ 1
20	89	Premium account	0	\$null\$	0	\$null\$	\$null\$ 1
	4						•
							_
							ок

The values of the *campaign* field are actually coded as integers in the data, with labels defined in the Type node (for example, $2 = Premium \ account$). You can toggle display of value labels in the table using the toolbar.

The file also includes a number of fields containing demographic and financial information about each customer that can be used to build or "train" a model that predicts response rates for different groups based on specific characteristics.

Building the Stream

► Add a Statistics File node pointing to *pm_customer_train1.sav*, located in the *Demos* folder of your IBM® SPSS® Modeler installation. (You can specify \$CLE0_DEMOS/ in the file path as a shortcut to reference this folder.)

Figure 11-3 Reading in the data	
📀 pm_customer_train1.sav	
Preview Refresh	0
\$CLEO_DEMOS/pm_customer_train1.sav	
Data Filter Types Annotations	
Import file: \$CLEO_DEMOS/pm_customer_train1.sav	
Variable names:	
Values: Read data and labels © Read labels as data	
Vse field format information to determine storage	
OK Cancel	Apply Reset

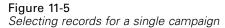
► Add a Type node, and select *response* as the target field (Role = Target). Set the measurement level for this field to Flag.

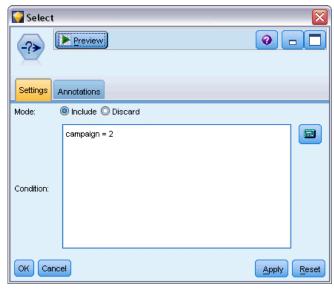
	eview				0-	
Types Format	Annotations Read Va		Values	Clear All Va		
Field -	Measurement	Values	Missing	Check	Role	Ļ
customer_id	Continuous	[7,116993]		None	O None	
ݤ campaign 🔰	nominal	1,2,3,4		None	None None	
response	🍯 Flag	1/0		None	🔘 Target	-
🖸 response	🖉 Continuous	[2006-04		None	O None	
ݤ purchase 🛛 .	🖉 Continuous	[0,1]		None	🛇 None	
💽 purchase	🖉 Continuous	[2006-04		None	🛇 None	
🗦 product_id 🛛 .	🔗 Continuous	[183,421]		None	🛇 None	
🔉 Rowid 🛛 🕴	🔗 Continuous	[1,19599]		None	🛇 None	4
ana 🐔	Continuous	F1 0 0E1		None	🔪 loout	
View current 1	iields 🔘 View unu:	sed field settin	gs			

Figure 11-4

- Set the role to None for the following fields: customer_id, campaign, response_date, purchase, purchase_date, product_id, Rowid, and X_random. These fields all have uses in the data but will not be used in building the actual model.
- Click the Read Values button in the Type node to make sure that values are instantiated.

Although the data includes information about four different campaigns, you will focus the analysis on one campaign at a time. Since the largest number of records fall under the Premium campaign (coded *campaign* = 2 in the data), you can use a Select node to include only these records in the stream.





Creating the Model

Attach a Decision List node to the stream. On the Model tab, set the Target value to 1 to indicate the outcome you want to search for. In this case, you are looking for customers who responded *Yes* to a previous offer.

Figure 11-6 Decision List node, Model tab

🖓 response[1]	X
3	
Fields Model Expert Analyze Annotations	
Model name: O Auto O Custom	
✓ Use partitioned data	
I Build model for each split	
Mode: O Generate model O Launch interactive session	
Target value: 1	
Find segments with: High probability Maximum number of segments: 3	
Minimum segment size As percentage of previous segment (%): 5.0 As absolute value (N): 50	
Segment rules Maximum number of attributes: 5 Allow attribute re-use Confidence interval for new conditions (%): 85.0	
OK Run Cancel	eset

- ► Select Launch interactive session.
- ▶ To keep the model simple for purposes of this example, set the maximum number of segments to 3.
- Change the confidence interval for new conditions to 85%.

► On the Expert tab, set the Mode to Expert.

Figure 11-7 Decision List node, Expert tab

😡 response[1]	X
Fields Model Expert Analyze	Annotations
Mode: 🔘 Simple 🔘 Expert	
Binning method:	Equal Count 👻
Number of bins:	10 🗲
Model search width:	5 🗲
Rule search width:	5 🗲
Bin merging factor:	2.0
Allow missing values in condition	ons
V Discard intermediate results	
Interactive mode only	
Maximum number of atternatives:	3
OK 🕨 Run Cancel	Apply Reset

- ► Increase the Maximum number of alternatives to 3. This option works in conjunction with the Launch interactive session setting that you selected on the Model tab.
- Click Run to display the Interactive List viewer.

Modeling Customer Response (Decision List)

	ractive List: response[1] #1				_	_) C
File	🛛 📄 Edit 💰 View Tools 🖔	Generate		9) 🛛 📼 🗣 🐓			9
iewer	Gains Annotations						
	Take Snapshot			Segment Finder			
_	rake Shapshot			Find segments with:	High Probability 🤝	Settings	
Targe	et field: 🔾 response			Max. no. of new segments:		Find Segments	
Targe	et value: 1					Find Segments	
id	Segment Rules					Probability	חר
	All segments including Remainder		Score	Cover (n) F 13,504	requency 1,952		
	Remainder			13,504	1,952		
							ł
Model	I Summary, Cover 0: Frequency 0: Proba	bility 0%					
Model	I Summary; Cover 0: Frequency 0: Proba	bility 0%					

Since no segments have yet been defined, all records fall under the remainder. Out of 13,504 records in the sample, 1,952 said *Yes*, for an overall hit rate of 14.45%. You want to improve on this rate by identifying segments of customers more (or less) likely to give a favorable response.

 In the Interactive List viewer, from the menus choose: Tools > Find Segments

Figure 11-9
Interactive List viewer

nieractive List view	vei				
😂 Interactive List: respons	se[1] #1				
🔓 Eile 📄 Edit 💰 View	Tools 🖏 Generate 🛛 📓 🕒) 🗷 🖬 🖷 💲		0 ×
Viewer Gains Annotations	Find Segments				
Take Snapshot	 Organize <u>M</u> odel Measures		Segment Finder	High Probability 🔻	0.1 11-112
Target field: 🔾 response	 Organize <u>Data Selections</u> Change Target <u>Value</u> 		Max. no. of new segment		Settings
Target value: 1	Take Snapshot				
id Segment Rules	Si	core	Cover (n)	Frequency	Probability
All segments including R	emainder	(S)	13,504	1,952	
Remainder			13,504	1,952	14.45%
Model Summary, Cover 0: Frequ	vency 0: Probability 0%				
					ок

This runs the default mining task based on the settings you specified in the Decision List node. The completed task returns three alternative models, which are listed in the Alternatives tab of the Model Albums dialog box.

Name	Target		No. of Segmer	ıts	Cover		Freq.	Prob.
Alternative 1	1			3		2,375		
Alternative 2	1			3		2,368		
Alternative 3	1			3		2,380	1,329	55.849
Alternative Pre	eview							
id	Segment Rules			Score		Cover (n)	Frequency	Probability
	All segments inclu	iding Remain	ıder			13,504	1,952	14.45%
	🗆 income, nur	nber prod	ucts					
1		5267.000 an		1		912	795	87.17%
		oducts > 1.0						
_	🗏 rfm_score,							
2		> 12.333 and		1		737	360	48.85%
	number_tra	nsactions >	2.000					
	🗏 number_tra	nsactions,	income					
	number_tra	nsactions >	0.000 and			704	474	00.000
3	number_tra	nsactions <	= 1.000 and	1		731	174	23.80%
	income > 4	5072.000						
	Remainder					11,124	623	5.60%
1 Load								
Louis								
Alternatives S	Snapshots							

Figure 11-10 Available alternative models

Select the first alternative from the list; its details are shown in the Alternative Preview panel.
 Figure 11-11

Name	Target	o. of Segments	Cover	F	Freq.	Prob.
Alternative 1	1.0	3 3		2,375	1,348	56,76%
Alternative 2	1.0	3		2,368	1,326	56.00%
Alternative 3	1.0	3		2,380	1,329	55.84% 🔽
Alternative P	review					
id	Segment Rules	Scor	e (Cover (n)	Frequency	Probability
	All segments including Remaind	er		13,504	1,952	14.45%
	🗉 income, number_produ	cts				
1	income > 55267.000 and	1.0		912	795	87.17%
	number_products > 1.000	D				
	🖃 rfm_score, number_trai	nsactions				
2	rfm_score > 10.535 and	1.0		725	357	49.24%
-	number_transactions > 3					10.2170
	= average#balance#feed#i					
	average#balance#feed#i			700	196	20.000
3	average#balance#feed#i			738	196	26.56%
	number_products <= 2.00	UU and				
	rfm_score > 9.239					
	Remainder			11,129	604	5.43%

The Alternative Preview panel allows you to quickly browse any number of alternatives without changing the working model, making it easy to experiment with different approaches.

Note: To get a better look at the model, you may want to maximize the Alternative Preview panel within the dialog, as shown here. You can do this by dragging the panel border.

Using rules based on predictors, such as income, number of transactions per month, and RFM score, the model identifies segments with response rates that are higher than those for the sample overall. When the segments are combined, this model suggests that you could improve your hit rate to 56.76%. However, the model covers only a small portion of the overall sample, leaving over 11,000 records—with several hundred hits among them—to fall under the remainder. You want a model that will capture more of these hits while still excluding the low-performing segments.

 To try a different modeling approach, from the menus choose: Tools > Settings

reate/Edit Mining Task: res	ponse[1]					
.oad Settings: response[1]	•	New 🗙				
Target Field: OB response Target Value: 1						
Simple Settings						
Find segments with:		High Probability 🔻				
Maximum number of new segmen	its:	3 🚔				
Minimum segment size						
As percentage of previous se	gment (%):	5.0 🗲				
As absolute value (N):	50 🚔					
Maximum number of alternatives:	3 🗲					
Maximum attributes per segment:	5 🚔					
📝 Allow attribute re-use with	in segment					
Confidence interval for new cond	litions (%):	85.0 🗲				
Expert Settings						
Binning method:	Equal Count	Number of bins:	10			
Model search width:	5	Rule search width:	5			
Bin merging factor:	2.00					
Allow missing values in condition:	s: True	Discard intermediate results:	Tru			
		Edit				
Data						
Build Selection: All Data	▼ 🖽					
Available fields: 🔘 All fields 🔘) Custom	t				

Click the New button (upper right corner) to create a second mining task, and specify *Down Search* as the task name in the New Settings dialog box.

Figure 11-13

Figure 11-13	
Create/Edit Mining	Task dialog box

Create/Edit Mining Task: response[1]							
Load Settings: Down Search Target							
⊚ Target Field: O⊛ response 🛛 Target Value: 1							
Simple Settings							
Find segments with: Low Probability T							
Maximum number of new segments	:	3 🗲					
Minimum segment size							
As percentage of previous segn	nent (%):	5.0 ≑					
As absolute value (N):		1,000 ≑					
Maximum number of alternatives:		3 🖨					
Maximum attributes per segment:		5					
📝 Allow attribute re-use within	segment						
Confidence interval for new condition	ons (%):	85.0 ≑					
Expert Settings							
Binning method:	Equal Count	Number of bins:	10				
Model search width:	5	Rule search width:	5				
Bin merging factor:	2.00						
Allow missing values in conditions:	True	Discard intermediate results:	True				
Data		_					
Build Selection: All Data	▼ 🖽						
Available fields: 💿 All fields 🔘 C	ustom Edit						
O	Cancel	Help					

- Change the search direction to Low probability for the task. This will cause the algorithm to search for segments with the *lowest* response rates rather than the highest.
- ▶ Increase the minimum segment size to 1,000. Click OK to return to the Interactive List viewer.
- ▶ In Interactive List viewer, make sure that the *Segment Finder* panel is displaying the new task details and click Find Segments.

Figure 11-14 Find segments in n	ew mining task	
Segment Finder		
Find segments with:	Low Probability 🔻	Settings
Max. no. of new segments:	3 🖨	🕨 Find Segments 🔳

The task returns a new set of alternatives, which are displayed in the Alternatives tab of the Model Albums dialog box and can be previewed in the same manner as previous results.

Name	Target	No. of Segments		Cover		Freq.	Prob.
Alternative 1	1		3		9,183		2.539
Alternative 2	1		3		9,183	232	2.539
Alternative 3	1		3		8,749	144	1.659
Alternative Pr	1						
id	Segment Rules		Score				Probability
	All segments including Remain	der			13,504	1,952	14.45%
1	months_customer	1			1,747	0	0.00%
	months_customer = "0"	ľ			1,141		0.00%
	🗆 rfm_score						
2	rfm_score <= 0.000	1			6,003	0	0.00%
	□ income, rfm_score						
3	income > 40297.000 and				1,433	232	16.19%
3	income <= 55267.000 ar	na i			1,435	232	10.19%
	rfm_score > 0.000 and rfm_score <= 10.535						
	the second s						
	Remainder				4,321	1,720	39.81%
1 Load							
) Loud							

This time each model identifies segments with low response probabilities rather than high. Looking at the first alternative, simply excluding these segments will increase the hit rate for the remainder to 39.81%. This is lower than the model you looked at earlier but with higher coverage (meaning more total hits).

By combining the two approaches—using a Low Probability search to weed out uninteresting records, followed by a High Probability search—you may be able to improve this result.

 Click Load to make this (the first Down Search alternative) the working model and click OK to close the Model Albums dialog box.

Figure 11-16 Excluding a segment

rge	<mark>Take Snapshot</mark> et field: ●● response et value: 1			Segment Finder Find segments with Max. no. of new se	,	Settings Find Segments
	Segment Rules	Score	Cover (n	1	Frequency	Probability
	All segments including Remainder			13,504	1,952	14.45%
1	months_customer months_customer = "0"	Excluded		1,747	C	0.00%
2	☐ rfm_score rfm_score <= 0.000	Excluded		6,003	a	0.00% >
3	☐ income, rfm_score income > 40287 000 and income <= 55267 000 and rfm_score > 0.000 and rfm_score <= 10.535	1		1,433	232	16.19%
	Remainder			4,321	1,720	39.81%

- ► Right-click on each of the first two segments and select Exclude Segment. Together, these segments capture almost 8,000 records with zero hits between them, so it makes sense to exclude them from future offers. (Excluded segments will be scored as null to indicate this.)
- Right-click on the third segment and select Delete Segment. At 16.19%, the hit rate for this segment is not that different than the baseline rate of 14.45%, so it doesn't add enough information to justify keeping it in place.

Note: Deleting a segment is not the same as excluding it. Excluding a segment simply changes how it is scored, while deleting it removes it from the model entirely.

Having excluded the lowest-performing segments, you can now search for high-performing segments in the remainder.

Modeling Customer Response (Decision List)

• Click on the remainder row in the table to select it, so that the next mining task will apply to the remainder only.

	eractive List: response[1] #2			<u>, </u>					
File	e 📄 Edit 💰 View Tools 🐑 Generate 🛛 🔠 🤱					0			
we	Gains Annotations								
Take Snapshot									
				Find segments with	Low Probability 🔻	Settings			
	et field: 🔎 response			Max. no. of new se	gments: 3 🚔	Find Segments			
nrg	et value: 1								
4	Segment Rules	Score	Cover (n)	Frequency	Probability			
	All segments including Remainder			13,504	1,952	2 14.45%			
1	months_customer months_customer = "0"	Excluded		1,747	C	0.00%			
2	□ rfm_score rfm_score <= 0.000	Excluded		6,003	C	0.00%			
	Remainder			5,754	1,952	2 33.92%			

- ▶ With the remainder selected, click Settings to reopen the Create/Edit Mining Task dialog box.
- ► At the top, in Load Settings, select the default mining task: response[1].
- Edit the Simple Settings to increase the number of new segments to 5 and the minimum segment size to 500.

• Click OK to return to the Interactive List viewer.

Figure 11-18

Selecting the default mining task

Create/Edit M	ining Task: Dow	n Search		X		
Load Settings:	response[1]	*	New 🗙			
Target						
	🍥 Target Field: 🛛	Target Value: 1				
Simple Settings						
Find segments with: High Probability 🔻						
Maximum numb	er of new segments	:	5 🚔			
Minimum segme	ent size					
As percents	age of previous segr	ne⊓t (%):	5.0 🗲			
As absolute	e value (N):		500 🗲			
Maximum numb	er of alternatives:		3 🗲			
Maximum attrib	utes per segment:	5 🗲				
📝 Allow at	ttribute re-use within	segment				
Confidence inte	erval for new conditi	ons (%):	85.0 ≑			
Expert Settings						
Binning method	ł:	Equal Count	Number of bins:	10		
Model search v	width:	5	Rule search width:	5		
Bin merging fac	ctor:	2.00				
Allow missing	values in conditions:	True	Discard intermediate results:	True		
_Data						
Build Selection:	All Data					
Available fields	: 💿 All fields 🔘 🤇	Custom Edit				
	O	Cancel !	<u>delb</u>			

► Click Find Segments.

This displays yet another set of alternative models. By feeding the results of one mining task into another, these latest models contain a mix of high- and low-performing segments. Segments with low response rates are excluded, which means that they will be scored as null, while included segments will be scored as 1. The overall statistics reflect these exclusions, with the first

alternative model showing a hit rate of 45.63%, with higher coverage (1,577 hits out of 3,456 records) than any of the previous models.

Name		Target	No. of Segm	ents	Cover	r	Freq.	F	rob.	
Alternative 1	1			7		37		,577	45.6	3%
Alternative 2	1			7		3,	456 1	,577	45.6	3%
Alternative 3	1			7		3,	456 1	,577	45.6	39
Alternative Pr	eview		•	••						
id	Segmen	t Rules		Score		Cover (n)	Frequency	Prob	ability	
	All segm	ents including Remain	der			13,504	1,952	2	14.45%	4
1		n ths_customer onths_customer = "0"		Excluded	I	1,747	C)	0.00%	
2	⊟ rfm_ rfr	_ score n_score <= 0.000		Excluded	I	6,003	c)	0.00%	
3	rfr	_score, income m_score > 12.333 and come > 52213.000		1		555	456	3	82.16%	
4	🗆 inco inc	me come > 55267.000		1		643	551		85.69%	
5	nu	n ber_transactions, i imber_transactions > : n_score > 12.333		1		533	206	3	38.65%	-
1 Load										
Alternatives	Snapshot									

Figure 11-19 Alternatives for combined model

▶ Preview the first alternative and then click Load to make it the working model.

Calculating Custom Measures Using Excel

• To gain a bit more insight as to how the model performs in practical terms, choose Organize Model Measures from the Tools menu.

Figure 11-20 Organizing model measures

ile	📄 Edit 🛛 💰 View	Tools 🖏 Generate 🛛 🗐 🛃			🛛 🗖	þ 🖷 🐦				
iewer	Gains Annotations	Find Segments Settings								
💕 T <u>a</u>	ke Snapshot	📝 Organize <u>M</u> odel Measures	D.			Segment Finder				
Tarnet	field: 🔘 response	Organize Data Selections	N			Find segments wit	h:	High Probability 🥆		Settings
		Ohange Target <u>V</u> alue				Max. no. of new s	egments:	5 🚔	F	ind Segments 🔳
Target	value: 1	📽 Take Snapshot								
id :	Segment Rules		Score		Cover (n)		Frequenc	ÿ	Probability	
1	All segments including Re	mainder	8			13,504		1,952		14.45%
1	months_customer months_customer =		Exclud	led		1,747		0		0.00%
2	<pre>Ifm_score ffm_score <= 0.000</pre>)	Excluc	led		6,003		0		0.00%
3	rfm_score, income rfm_score > 12.333 income > 52213.00	3 and	1			555		456		82.16%
4	income > 55267.00	0	1			643		551		85.69%
5	number_transaction number_transaction rfm_score > 12.333	ns > 2.000 and	1			533		206		38.65%

The Organize Model Measures dialog box allows you to choose the measures (or columns) to show in the Interactive List viewer. You can also specify whether measures are computed against all records or a selected subset, and you can choose to display a pie chart rather than a number, where applicable.

Figure 11-21 Organize Model Measures dialog box

Name	Туре	Display	Data Selection	Show
Cover	Coverage	Pie Chart	All Data	
Cover (n)	Coverage	Numeric	All Data	
Frequency	Frequency	Numeric	All Data	
Probability	Probability	Numeric	All Data	
Error	Error	Numeric	All Data	
				1
	Description			Show
Name				
Name				
Ivame				
Name				

In addition, if you have Microsoft Excel installed, you can link to an Excel template that will calculate custom measures and add them to the interactive display.

- ▶ In the Organize Model Measures dialog box, set Calculate custom measures in Excel (TM) to Yes.
- Click Connect to Excel (TM)
- Select the *template_profit.xlt* workbook, located under *streams* in the *Demos* folder of your IBM® SPSS® Modeler installation, and click Open to launch the spreadsheet.

Figure 11-22

Exce	ei ivi						
× 1	Micro	soft Exc	el - template	e_profit1			🛛
:2	Eile	<u>E</u> dit	<u>V</u> iew <u>I</u> nsert	F <u>o</u> rmat <u>T</u> ools	<u>D</u> ata <u>W</u> indow <u>H</u> elp A	do <u>b</u> e PDF	_ & ×
	1	Arial	•	10 - B I	· U 📰 🚍 🔄 🔤 🛛	9 % , 🔝 🚝 🛄 🗸	🔕 - 🗛 - 🚆
1	1 🟞	€.					
_	F4		▼ fx =	=IF(H4="",0,L4)-	·Settings!FIX_1		
	Α	В	С	D	E	F	G 🔨
							∃
1							
	1						
2							
2			Metric:	Imported Me			
-	#	Use	Metric: Frequency	Imported Me Cover	etric: Calculated Metri Profit Margin	ic: Calculated Metric: Cumulative Profit	Target
2	#	Use					Target
2	#	Use					Target
2	#	Use					
2		Use				Cumulative Profit	
2		Use				Cumulative Profit	
2 3 4	1	Use				Cumulative Profit	
2		Use				Cumulative Profit	0
2 3 4	1		Frequency		Profit Margin	Cumulative Profit	

The Excel template contains three worksheets:

- Model Measures displays model measures imported from the model and calculates custom measures for export back to the model.
- Settings contains parameters to be used in calculating custom measures.
- Configuration defines the measures to be imported from and exported to the model.

The metrics exported back to the model are:

- **Profit Margin**. Net revenue from the segment
- **Cumulative Profit.** Total profit from campaign

As defined by the following formulas:

Profit Margin = Frequency * Revenue per respondent - Cover * Variable cost

Cumulative Profit = Total Profit Margin - Fixed cost

Note that Frequency and Cover are imported from the model.

The cost and revenue parameters are specified by the user on the Settings worksheet.

_				kshee			_								_
×	ticros	oft Ex	xcel-t	template	e_profit	t1								-)0	×
:1	Eile	<u>E</u> dit	⊻iew	Insert	F <u>o</u> rmat	<u>T</u> ools	<u>D</u> ata	<u>W</u> indow	Help	Ado <u>b</u> e PD	F			- 8	×
: 🔿	1	Arial			• 10 •	BI	Ū		+a+	9 %	• ^{€.0}	Image: Second	• 🖑 •	A -	++ ₹
	1														
	J12		-	fx											
			A			В	D	E		F	G	Н			~
4															
5															
6															_
7															
8											_				
9															-
10															-
11															-
12			reveni	ue							_				- =
	- Fixe				2	2,500.00									-
	- Vari				-	0.50									-
15	- Revi	enue j	per res	pondent		100.00									-
16															-
17															-
18 19															-
20											_				
20															
	► H	∖ Me	odel Me	easures	∫Settin	igs / Cor	nfigur	ation /	<	1				>	È
Read												NUM			

Figure 11-23 Excel Settings worksheet

Fixed cost is the setup cost for the campaign, such as design and planning.

Variable cost is the cost of extending the offer to each customer, such as envelopes and stamps.

Revenue per respondent is the net revenue from a customer who responds to the offer.

► To complete the link back to the model, use the Windows taskbar (or press Alt+Tab) to navigate back to the Interactive List viewer.

Figure 11-24 Choosing inputs	for custom measures
Choose inputs for	Custom Measures 🛛 🗙
Hint: Use this dialog to inputs to calculate cus	choose which model measures will be used by Excel (TM) as tom measures.
Input	Model Measure
Frequency	Frequency
Cover	Cover (n)

The Choose Inputs for Custom Measures dialog box is displayed, allowing you to map inputs from the model to specific parameters defined in the template. The left column lists the available measures, and the right column maps these to spreadsheet parameters as defined in the Configuration worksheet.

► In the Model Measures column, select Frequency and Cover (n) against the respective inputs and click OK.

In this case, the parameter names in the template—Frequency and Cover (n)—happen to match the inputs, but different names could also be used.

► Click OK in the Organize Model Measures dialog box to update the Interactive List viewer.

Figure 11-25 Organize Model Measures dialog box showing custom measures from Excel × Organize Model Measures Int: Use this dialog to define the Model Measures which are displayed in the Viewer table. Name Display Data Selection Show Туре * Cover Pie Chart All Data Coverage + Cover (n) ✓ ✓ ✓ Coverage Numeric All Data Frequency Frequency Numeric All Data ÷ Probability All Data Probability Numeric Error Error All Data × Numeric Custom Measures Calculate custom measures in Excel (TM): 🔘 Yes 🔘 No Connect to Excel (TM)... Workbook: Files\SPSSInc\PASVModeler14\Demos\Classification_Module\template_profit.xt Name Description Show Profit margin Excel calculated profit margin 3 Cumulative profit Excel calculated cumulative profit OK Cancel Help

The new measures are now added as new columns in the window and will be recalculated each time the model is updated.

Figure 11-26

Custom measures from Excel displayed in the Interactive List viewer

ırg	Take Snapshot et field: I response et value: 1		F	egment Finder ind segments with: lax. no. of new segi		obability 🔻	Find	Segments
1	Segment Rules	Score	Cover (n)	Frequency	F	Probability	Profit margin	Cumulative
	All segments including Remainder			3,504	1,952	14.45%		
1	months_customer months_customer = "0"	Excluded		1,747	0	0.00%	-873.5	-2,500
2	l = rfm_score rfm_score <= 0.000	Excluded		6,003	0	0.00%	-3,001.5	-2,500
3	G rfm_score, income rfm_score > 12.333 and income > 52213.000	1		555	456	82.16%	45,322.5	42,822.5
4	☐ income income > 55267.000	1		643	551	85.69%	54,778.5	97,601
5	number_transactions, rfm_score number_transactions > 2.000 and rfm_score > 12.333	1		533	206	38.65%	20,333.5	117,934.5
nde	Summary, Cover 3,456: Frequency 1,577: Probabili	ty 45.63%						

By editing the Excel template, any number of custom measures can be created.

Modifying the Excel template

Although IBM® SPSS® Modeler is supplied with a default Excel template to use with the Interactive List viewer, you may want to change the settings or add your own. For example, the costs in the template may be incorrect for your organization and need amending.

Note: If you do modify an existing template, or create you own, remember to save the file with an Excel 2003 *.xlt* suffix.

To modify the default template with new cost and revenue details and update the Interactive List viewer with the new figures:

- ▶ In the Interactive List viewer, choose Organize Model Measures from the Tools menu.
- ▶ In the Organize Model Measures dialog box, click Connect to Excel[™].
- ► Select the *template_profit.xlt* workbook, and click Open to launch the spreadsheet.
- ► Select the Settings worksheet.

• Edit the Fixed costs to be 3,250.00, and the Revenue per respondent to be 150.00.

NIOC	lified values	on Excel 3	Settings wor	ksh	eet					
N	licrosoft Excel	- template_	_profit1.xlt							
:2	<u>File E</u> dit <u>V</u> iev	w <u>I</u> nsert I	F <u>o</u> rmat <u>T</u> ools	<u>D</u> ata	<u>W</u> indow <u>H</u> elp	Ado <u>b</u> e PDF			_ 6	×
: 🖪	🚆 Arial	-	10 - B <i>I</i>	U	토 플 클 클	🥶 % ,	*.0 .00	- <u>-</u>	🖏 - <u>A</u> -	
	🔁 🖏 📮									
	GAIN_1 -	<i>f</i> × 15	50							
	A		В	D	E	F	G	H		~
4										
5										
6										
7										_
8										_
9										_
10										_
11										_
12	Costs and reve	enue	0.050.00							- =
	- Fixed costs		3,250.00							-
14 15	- Variable cost - Revenue per r		0.50							-
16	- Revenue per r	espondent	150.00							-
17										-
18										
19										
20										-
21										~
14 4	🕨 🕨 🔪 Model	Measures λ	Settings / Cor	nfigur	ation /		1111		>	
Read	ly							NUM		

Figure 11-27 *Modified values on Excel Settings worksheet*

► Save the modified template with a unique, relevant filename. Ensure it has an Excel 2003 *.xlt* extension.

Figure 11-28 Saving modifi	ied Excel te	emplate					
Save As							? 🔀
Save in:	🛅 Classifical	tion_Module	🖌 🚱 .	- 🔰 🖏	× 📴 🎟 •	r Tools -	
My Recent Documents Desktop My Documents My Computer	template_p						
S	File <u>n</u> ame:	template_profit_3250,x	lt		~	<u>5</u> a	ve
My Network Places	Save as <u>t</u> ype:	Template (*.xlt)			~		ncel

- Use the Windows taskbar (or press Alt+Tab) to navigate back to the Interactive List viewer.
 In the Choose Inputs for Custom Measures dialog box, select the measures you want to display and click OK.
- ► In the Organize Model Measures dialog box, click OK to update the Interactive List viewer.

Obviously, this example has only shown one simple way of modifying the Excel template; you can make further changes that pull data from, and pass data to, the Interactive List viewer, or work within Excel to produce other output, such as graphs.

Figure 11-29

Modified custom measures from Excel displayed in the Interactive List viewer

rge	ake Snapshot t field: •• response t value: 1		Find	ment Finder segments with: Hig . no. of new segments:	n Probability 🤝	Find	Segments
	Segment Rules	Score	Cover (n)	Frequency	Probability	Profit margin	Cumulative
	All segments including Remainder		13,504	1,952	14.45%	0	0 📤
1	months_customer months_customer = "0"	Excluded	1,747	0	0.00%	-873.5	-3,250
2	□ rfm_score rfm_score <= 0.000	Excluded	6,003	3 O	0.00%	-3,001.5	-3,250
3	☐ rfm_score, income rfm_score > 12.333 and income > 52213.000	1	556	5 456	82.16%	68,122.5	64,872.5
4	☐ income income > 55267.000	1	643	551	85.69%	82,328.5	147,201
5	number_transactions, rfm_score number_transactions > 2.000 and rfm_score > 12.333	1	533	3 206	38.65%	30,633.5	177,834.5

Saving the Results

To save a model for later use during your interactive session, you can take a snapshot of the model, which will be listed on the Snapshots tab. You can return to any saved snapshot at any time during the interactive session.

Continuing in this manner, you can experiment with additional mining tasks to search for additional segments. You can also edit existing segments, insert custom segments based on your own business rules, create data selections to optimize the model for specific groups, and customize the model in a number of other ways. Finally, you can explicitly include or exclude each segment as appropriate to specify how each will be scored.

When you are satisfied with your results, you can use the Generate menu to generate a model that can be added to streams or deployed for purposes of scoring.

Alternatively, to save the current state of your interactive session for another day, choose Update Modeling Node from the File menu. This will update the Decision List modeling node with the current settings, including mining tasks, model snapshots, data selections, and custom measures. The next time you run the stream, just make sure that Use saved session information is selected in the Decision List modeling node to restore the session to its current state.



Classifying Telecommunications Customers (Multinomial Logistic Regression)

Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric one.

For example, suppose a telecommunications provider has segmented its customer base by service usage patterns, categorizing the customers into four groups. If demographic data can be used to predict group membership, you can customize offers for individual prospective customers.

This example uses the stream named *telco_custcat.str*, which references the data file named *telco.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *telco_custcat.str* file is in the *streams* directory.

The example focuses on using demographic data to predict usage patterns. The target field *custcat* has four possible values that correspond to the four customer groups, as follows:

Value	Label
1	Basic Service
2	E-Service
3	Plus Service
4	Total Service

Because the target has multiple categories, a multinomial model is used. In the case of a target with two distinct categories, such as yes/no, true/false, or churn/don't churn, a binomial model could be created instead. For more information, see the topic Telecommunications Churn (Binomial Logistic Regression) in Chapter 13 on p. 154.

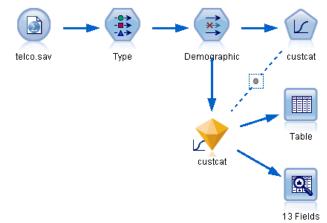
Classifying Telecommunications Customers (Multinomial Logistic Regression)

Building the Stream

▶ Add a Statistics File source node pointing to *telco.sav* in the *Demos* folder.

Figure 12-1

Sample stream to classify customers using multinomial logistic regression



► Add a Type node and click Read Values, making sure that all measurement levels are set correctly. For example, most fields with values 0 and 1 can be regarded as flags.

😡 Туре								×
Types Format	Annotations	ead Value	es Clea	r Values	СІ	(ear All Valu	Ies	
Field -	Measurer	nent	Values	Missing		Check	Role	
찾 gender	💑 Nominal		0,1		No	ne	🔪 Input	<u></u>
🚫 reside	🤣 Continuou:	s	[1,8]		No	ne	🔪 Input	
📿 tollfree	🎖 Flag		1/0		No	ne	🔪 Input	
📿 equip	⊌ Flag ⊌ Flag ⊌ Flag		1/0		No	ne	🔪 Input	
📿 callcard	😸 Flag		1/0		N	<default></default>	,	
📿 wireless	Flag		1/0		NI			
🛞 longmon	🖉 Continuou	Sel	ect All			Continuo	us	
🛞 tollmon	🔗 Continuou	Sel	ect None			Categorio	al	
View current	fields 🔘 Vi	Sel	ect Fields	•		Flag	R	
		Cop	ру	Ctrl+C		Nominal	N	
OK Cancel		-	ste Special	. Ctrl+V		Ordinal		Reset

Figure 12-2 Setting the measurement level for multiple fields

Tip: To change properties for multiple fields with similar values (such as 0/1), click the *Values* column header to sort fields by value, and then hold down the shift key while using the mouse or arrow keys to select all the fields you want to change. You can then right-click on the selection to change the measurement level or other attributes of the selected fields.

Notice that *gender* is more correctly considered as a field with a set of two values, instead of a flag, so leave its Measurement value as Nominal.

• Set the role for the *custcat* field to Target. All other fields should have their role set to Input.

Figure 12-3 Setting field r	ole					
😡 Туре						X
	review				0	
Types Format	Annotations					
~	💌 🚺 🕨 Read Valı	ues Clear	r Values	Clear All Va	lues	
Field 🗆	Measurement	Values	Missing	Check	Role	
	● riag	I/O		None	niput	1
	Continuous	[1.74919		None		
(#) logtoll	Continuous	[1.74919		None	> Input	
Iogequi	Continuous	[2.73436		None	Input	-
logwire	Continuous	[2.70136		None	Input	
	Continuous	[2.19722		None	Input	-
Custcat	Nominal	1,2,3,4		None	O Target	
Cusical Churn	An Nominal	0,1		None	Input	-
© View current	fields O View unus		gs			set

Classifying Telecommunications Customers (Multinomial Logistic Regression)

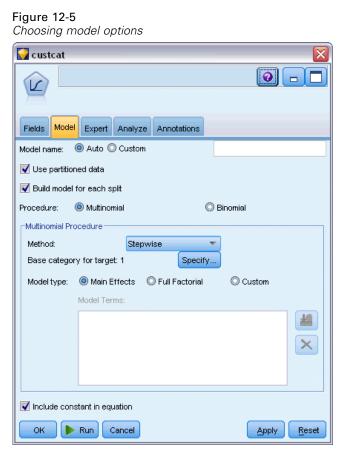
Since this example focuses on demographics, use a Filter node to include only the relevant fields (*region, age, marital, address, income, ed, employ, retire, gender, reside, and custcat*). Other fields can be excluded for the purpose of this analysis.

Z Demographic		
Filter Annotations		0 - [
7- 📑 🗰	Fields: 4	42 in, 31 filtered, 0 renamed, 11 d
Field	Filter	Field
region	\rightarrow	region
tenure	→	tenure
age	\rightarrow	age
marital	\rightarrow	marital
address	\rightarrow	address
income	\rightarrow	income
ed	\rightarrow	ed
employ	\rightarrow	employ
retire	\rightarrow	retire
gender	\rightarrow	gender
	View unused field s	settings

Figure 12-4

(Alternatively, you could change the role to None for these fields rather than exclude them, or select the fields you want to use in the modeling node.)

► In the Logistic node, click the Model tab and select the Stepwise method. Select Multinomial, Main Effects, and Include constant in equation as well.



Leave the Base category for target as 1. The model will compare other customers to those who subscribe to the Basic Service.

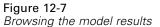
 On the Expert tab, select the Expert mode, select Output, and, in the Advanced Output dialog box, select Classification table.

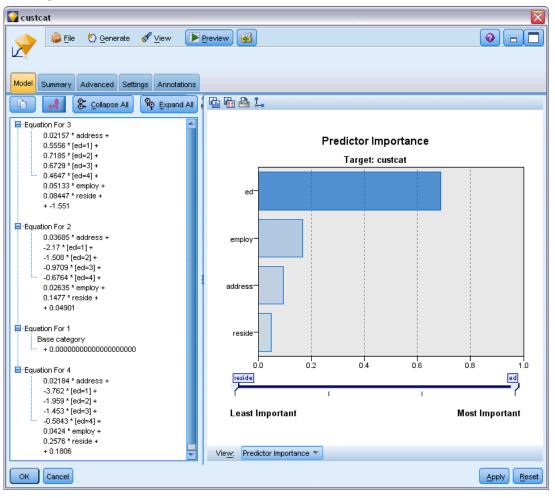
Figure 12-6 Choosing output options		
😡 Logistic Regression: Advance	d Output	
Summary statistics	🔲 Parameter estimates	
🔲 Likelihood ratio tests	Confidence interval:	95.0 🌲
Asymptotic correlation	Asymptotic covariance	
Coodness of fit chi-square statistics	🔽 Classification table	
teration history for every	1 荣	step(s)
E Stepwise variable loadings	Monotonicity measures	
Information criteria		
Са	ncel Help	

Browsing the Model

• Execute the node to generate the model, which is added to the Models palette in the upper-right corner. To view its details, right-click on the generated model node and choose Browse.

The model tab displays the equations used to assign records to each category of the target field. There are four possible categories, one of which is the base category for which no equation details are shown. Details are shown for the remaining three equations, where category 3 represents Plus Service, and so on.





Classifying Telecommunications Customers (Multinomial Logistic Regression)

The Summary tab shows (among other things) the target and inputs (predictor fields) used by the model. Note that these are the fields that were actually chosen based on the Stepwise method, not the complete list submitted for consideration.

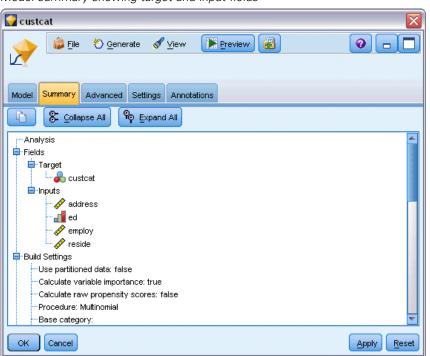


Figure 12-8 Model summary showing target and input fields

The items shown on the Advanced tab depend on the options selected on the Advanced Output dialog box in the modeling node.

One item that is always shown is the Case Processing Summary, which shows the percentage of records that falls into each category of the target field. This gives you a null model to use as a basis for comparison.

Without building a model that used predictors, your best guess would be to assign all customers to the most common group, which is the one for Plus service.

Figure 12- Case proce	9 essing summary							
🜍 custcat								
	🝺 File 👋 Generate 🖋 View	Previo	ew 🛃 🛛 🖓					
	mary Advanced Settings Annote	ations						
Case Proce	ssing Summary			-				
		N	Marginal Percentage					
	Basic service	266	26.6%					
custcat	E-service	217	21.7%					
Plus service 281 28.1%								
	Total service	236	23.6%					

Based on the training data, if you assigned all customers to the null model, you would be correct 281/1000 = 28.1% of the time. The Advanced tab contains further information that enables you to examine the model's predictions. You can then compare the predictions with the null model's results to see how well the model works with your data.

At the bottom of the Advanced tab, the Classification table shows the results for your model, which is correct 39.9% of the time.

In particular, your model excels at identifying Total Service customers (category 4) but does a very poor job of identifying E-service customers (category 2). If you want better accuracy for customers in category 2, you may need to find another predictor to identify them.

gure 12-10 lassification tab	ole				
🛛 custcat					X
Pile 🖉	N Generate	∛ ⊻iew	Preview	v 🛃	0
Model Summary A	dvanced Sett	ings Annot	ations		
@					
Classification					
			Predicted	1	
Observed	Basic service	E-service	Plus service	Total service	Percent Correct
Observed Basic service		E-service			
	service		service	service	Correct
Basic service	service 122	8	service 75	service 61	Correct 45.9%
Basic service E-service	service 122 58	8	service 75 68	service 61 81	Correct 45.9% 4.6%
Basic service E-service Plus service	service 122 58 89	8 10 8	service 75 68 133	service 61 81 51	Correct 45.9% 4.6% 47.3%

Depending on what you want to predict, the model may be perfectly adequate for your needs. For example, if you are not concerned with identifying customers in category 2, the model may be accurate enough for you. This may be the case where the E-service is a loss-leader that brings in little profit.

If, for example, your highest return on investment comes from customers who fall into category 3 or 4, the model may give you the information you need.

To assess how well the model actually fits the data, a number of diagnostics are available in the Advanced Output dialog box when you are building the model. Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *SPSS Modeler Algorithms Guide*, available from the *\Documentation* directory of the installation disk.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you can use a Partition node to hold out a subset of records for purposes of testing and validation.

Telecommunications Churn (Binomial Logistic Regression)

Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric one.

Chapter

This example uses the stream named *telco_churn.str*, which references the data file named *telco.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *telco_churn.str* file is in the *streams* directory.

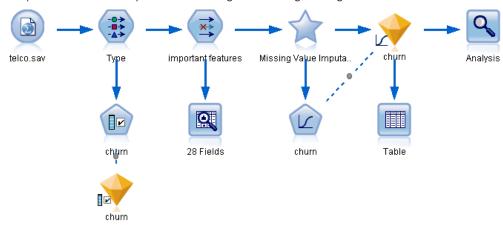
For example, suppose a telecommunications provider is concerned about the number of customers it is losing to competitors. If service usage data can be used to predict which customers are liable to transfer to another provider, offers can be customized to retain as many customers as possible.

This example focuses on using usage data to predict customer loss (churn). Because the target has two distinct categories, a binomial model is used. In the case of a target with multiple categories, a multinomial model could be created instead. For more information, see the topic Classifying Telecommunications Customers (Multinomial Logistic Regression) in Chapter 12 on p. 144.

Building the Stream

► Add a Statistics File source node pointing to *telco.sav* in the *Demos* folder.

Figure 13-1 Sample stream to classify customers using binomial logistic regression



► Add a Type node to define fields, making sure that all measurement levels are set correctly. For example, most fields with values 0 and 1 can be regarded as flags, but certain fields, such as gender, are more accurately viewed as a nominal field with two values.

🛛 Туре									2
	nnotations					(0		
√ - ∞ ∞	🕨 🕨 R	ead Value	es Clear	' Values	C	lear All Valu	ies		
Field -	Measurer	nent	Values	Missing		Check		Role	
🔆 gender 🛛 😹	Nominal		0,1		No	ne	>	Input	4
🔆 reside 🛛 🔗	Continuou	5	[1,8]		No	ne	1	Input	
🔿 tollfree 🛛 🖁	Flag		1/0		No	ne	>	Input	
🗘 tollfree 🛛 🖁	Flag		1/0		No	ne	2	Input	
🔿 callcard 🛛 🎖	Flag		1/0		N¢	<default></default>			
🔿 wireless 💦 🎖	Flag		1/0		NIZ	Speraulta			
🛞 longmon 🛛 🔗	Continuou	Sel	ect All			Continuou	IS		
🛞 tolimon 🛛 🔗	Continuou	Sel	ect None			Categoric	al		-
	• • •					_			
() View current fiel	ds 🔘 Vi	Sel	ect Fields	,		Flag		R	
		Cop	ογ	Ctrl+C		Nominal		N	
OK Cancel									Reset

Figure 13-2 Setting the measurement level for multiple fields

Tip: To change properties for multiple fields with similar values (such as 0/1), click the *Values* column header to sort fields by value, and then hold down the Shift key while using the mouse or arrow keys to select all of the fields that you want to change. You can then right-click on the selection to change the measurement level or other attributes of the selected fields.

Figure 13-3

156

• Set the measurement level for the *churn* field to Flag, and set the role to Target. All other fields should have their role set to Input.

Туре					
	review				
/pes Format	Annotations				
- 00 6	🍋 🚺 🍺 Read Va	lues Clear	Values	Clear All Va	alues
Field -	Measurement	Values	Missing	Check	Role
riela	iveasurement i riag	T/U	Missing	NUTE	I IIIput
loglong	🖉 Continuous	[-0.10536		None	🔪 Input
logtoll	🔗 Continuous	[1.74919		None	🔪 Input
logequi	🔗 Continuous	[2.73436		None	🔪 Input
logcard	🖉 Continuous	[1.01160		None	🔪 Input
logwire	🔗 Continuous	[2.70136		None	🔪 Input
> Ininc	🔗 Continuous	[2.19722		None	🔪 Input
custcat	💑 Nominal	1,2,3,4		None	🔪 Input
churn	🎖 Flag	1/0		None	🔘 Target
View current	fielde 🔘 View upu	and field anttin			
	tields 🛛 View unu	sed field settin	gs		

• Add a Feature Selection modeling node to the Type node.

Using a Feature Selection node enables you to remove predictors or data that do not add any useful information with respect to the predictor/target relationship.

▶ Run the stream.

Telecommunications Churn (Binomial Logistic Regression)

Open the resulting model nugget, and from the Generate menu, choose Filter to create a Filter node.
 Figure 13-4

😪 chu							
Chu	i u						
	🐞 <u>F</u> ile	O Generate	Pr	eview 🛃		0	
		🏷 Generate	Modeling	Node			
		Model to	Palette				
Model	Summary	Filter					
		-	R				
		Rank 🔽	A 4	<u>1</u>			
	Rank 🛆	Field	Me	asurement	Importance	Value	
-	1	🛞 tenure	🔗 Contir	nuous	🔀 Important	1.0	-
-	2	🛞 loglong	🔗 Contir	nuous	★ Important	1.0	
-		🛞 equip	💑 Nomir	nal	🚖 Important	1.0	
-	4 •	🛞 longten	🔗 Contir	nuous	🚖 Important	1.0	
-	5	🛞 employ	🔗 Contir	nuous	★ Important	1.0	
-	6	🛞 longmon	🔗 Contir	nuous	★ Important	1.0	
-	7 1	🛞 internet	💑 Nomir	nal	📩 Important	1.0	
-		🛞 equipmon 👘	🔗 Contir	nuous	📩 Important	1.0	
-	9	🛞 age	🔗 Contir	nuous	📩 Important	1.0	
\checkmark	10	🛞 ebill	💑 Nomir	nal	📩 Important	1.0	
-	11	🛞 address 👘	🔗 Contir	nuous	📩 Important	1.0	
-	12	🛞 callcard	🢑 Nomir	nal	📩 Important	1.0	
-		🛞 cardten	ntir 🖉 🖉		📩 Important	1.0	
-		🛞 ed	📑 Ordin	al	📩 Important	1.0	
-		🛞 tollten	🔗 Contir		📩 Important	1.0	
-		🛞 custcat	💑 Nomir		📩 Important	1.0	
-		🛞 voice	💑 Nomir		📩 Important	1.0	
-		👾 cardmon	n Contir		📩 Important	1.0	
-		🛞 logtoll	🖉 Contir		📩 Important	1.0	
-	20	👾 wireless	💑 Nomir	nal	📩 Important	1.0	-
Selecte	ed fields: 27	Total fields av	/ailable: 41				
		>	0.95 🕂	<= 0.95 💽 🔹	< 0.9		
			3 Scre	ened Fields			
	Field 7	Maarin			Basar-		
	Field 🔽	Measure	ment	Cincila antorrow	Reason		
	retire	Nominal	-	Single category			
	🛞 logwire 🋞 logequi	Continuou 🔗 Continuou		Too many miss Coefficient of v	ing values /ariation below th	reshold	
		1.					
ок	Cancel					Apply	Reset

Generating a Filter node from a Feature Selection node

Not all of the data in the *telco.sav* file will be useful in predicting churn. You can use the filter to only select data considered to be important for use as a predictor.

▶ In the Generate Filter dialog box, select All fields marked: Important and click OK.

► Attach the generated Filter node to the Type node.

Figure 13-5 Selecting important field	ls
😡 Generate Filter from Fe	atu 区
Mode: O Include	O Exclude
O Selected fields	
All fields marked:	
👿 📩 Important	
📄 🛨 Marginal	
📃 💽 Unimportant	
O Top number of fields	10 🌲
O Importance greater than:	0.667 韋
OK Cancel He	q

► Attach a Data Audit node to the generated Filter node.

Open the Data Audit node and click Run.

- ▶ On the Quality tab of the Data Audit browser, click the % *Complete* column to sort the column by ascending numerical order. This lets you identify any fields with large amounts of missing data; in this case the only field you need to amend is *logtoll*, which is less than 50% complete.
- ▶ In the *Impute Missing* column for *logtoll*, click Specify.

```
Figure 13-6
Imputing missing values for logtoll
```

🔍 Data Audit	of [28 fields] #2	2						
诊 <u>F</u> ile 🛛 📄 E	dit 🕙 <u>G</u> enerate		11					
Audit Quality	Annotations							
Complete fields ((%): 96.43% Co	mplete record	is (%): 47.5%	, ,				
Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method	% Complete 🔺	Vali
📿 logtoll	🔗 Continuous	2	0	None	Never 💌	Fixed	47.5	
🔆 tenure	🔗 Continuous	0	0	None	Never	Fixed	100	
🔆 age	🔗 Continuous	0	0	None	Blank Values	Fixed	100	
🛞 address	🔗 Continuous	12	0	None	Null Values	Fixed	100	
즞 income	🔗 Continuous	9	6	None	Blank & Null Value	Fixed	100	
즞 ed	📶 Ordinal			-	Condition	Fixed	100	
📿 employ	🔗 Continuous	8	0	None	Specify	Fixed	100	
즞 equip	🎖 Flag			-	Never Ve	Fixed	100	
决 callcard	🎖 Flag			-	Never	Fixed	100	
🛞 wireless	🎖 Flag			-	Never	Fixed	100	
🛞 longmon	🔗 Continuous	18	4	None	Never	Fixed	100	
🛞 tolimon	🔗 Continuous	9	1	None	Never	Fixed	100	
🛞 equipmon	🔗 Continuous	2	0	None	Never	Fixed	100	
🛞 cardmon	🔗 Continuous	11	3	None	Never	Fixed	100	
🛞 wiremon	🔗 Continuous	8	1	None	Never	Fixed	100	
🛞 longten	🔗 Continuous	20	4	None	Never	Fixed	100	
🛞 toliten	🔗 Continuous	18	2	None	Never	Fixed	100	
🔆 cardten	🔗 Continuous	11	6	None	Never	Fixed	100	
🔿 voice	🎖 Flaq			-	Never	Fixed	100	

► For Impute when, select Blank and Null values. For Fixed As, select Mean and click OK.

Selecting Mean ensures that the imputed values do not adversely affect the mean of all values in the overall data.

Figure 13 Selecting	-7 imputation se	ettings	
🔍 Imputat	tion Settings		
Field: I	ogtoll	Storage: 🚸 Real	
Impute when	:	Blank & Null Values 🔽	
Condition:			
Impute Metho	d:	Fixed	~
Impute Fixe	d Values		
Fixed as:	Mean 💌		
Value:	Mean Mid-Range 😽		
	Constant		
	ОК	Cancel <u>H</u> elp	

On the Data Audit browser Quality tab, generate the Missing Values SuperNode. To do this, from the menus choose:

Generate > Missing Values SuperNode

Figure 13-8

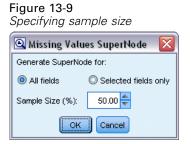
Generating a missing values SuperNode

logtoll Rectassify Node 0 None Blank & Null Val Fixed 47.5 tenure Binning Node 0 None Never Fixed 100 age Derive Node 0 None Never Fixed 100 address Derive Node 0 None Never Fixed 100 income Graph Output 6 None Never Fixed 100 ed Graph Output 6 None Never Fixed 100 equip Mominal Never Fixed 100 equip Mominal Never Fixed 100 vireless Nominal Never Fixed 100 olagmon Continuous 18 4 None Never Fixed 100 olagmon Continuous 2 0 None Never Fixed 100 cardmon Continuous 11 3 None Never Fixed 100 vireless Continuous 8 1 None Never Fixed 100	👂 File 🛛 📄 Ed	lit 🕙 <u>G</u> enerate								0)>
Outlier & Extreme SuperNode S% Signalete fields (%) Missing Values Filter Node S% Field Missing Values Select Node Extremes Action Impute Missing Method % Complete / Val Iogolal Reclassify Node O None Blank & Null Val Fixed 47.5 age Derive Node O None Never Fixed 100 address Derive Node O None Never Fixed 100 ed Graph Output None Never Fixed 100 equip Nominal None Never Fixed 100 callaard Nominal None Never Fixed 100 equip Nominal None Never Fixed 100 ordinaus 18 4 None Never Fixed 100 outlints 18 4 None Never Fixed 100 <tholl< th=""></tholl<>	Audit Quality	An Missing Value	s SuperNode							
Field Missing Values Select Node Extremes Action Impute Missing Method % Complete / Values logtol & Reclassify Node 0 None Blank & Null Val Fixed 47.5 age Binning Node 0 None Never Fixed 100 address Derive Node 0 None Never Fixed 100 income Graph Output 6 None Never Fixed 100 ed Graph Output - - Never Fixed 100 equip Oranh Output - - Never Fixed 100 calcard Nominal - - - Never Fixed 100 virielss Nominal - - - Never Fixed 100 0 virielss Nominal - - - Never Fixed 100 0 virielss Nominal - - - Never Fixed 100 0 virielss Nominal - -		Outlier & Extre	me SuperNode		_					-
Pick Pick <th< th=""><th>Complete fields ('</th><th>^{%):} Missing Value</th><th>s <u>F</u>ilter Node</th><th>.5%</th><th></th><th></th><th></th><th></th><th></th><th></th></th<>	Complete fields ('	^{%):} Missing Value	s <u>F</u> ilter Node	.5%						
tenure Binning Node 0 None Never Fixed 100 address Derive Node 0 None Never Fixed 100 address Derive Node 0 None Never Fixed 100 oncome Oraph Output 6 None Never Fixed 100 ed Graph Output 6 None Never Fixed 100 equip Nominal Never Fixed 100 equip Nominal Never Fixed 100 callcard Nominal Never Fixed 100 wireless Nominal Never Fixed 100 vireless Nominal Never Fixed 100 tollmon Continuous 18 4 None Never Fixed 100 cardmon Continuous 2 0 None Never Fixed 100 wirenon Continuous 11 3 None Never Fix	Field	Missing Value	s Select Node	Extremes		Action	Impute Missing	Method	% Complete 🗠	Valid
age Binning Node O None Never Fixed 100 address Derive Node 0 None Never Fixed 100 income Graph Output 6 None Never Fixed 100 ed Graph Output Never Fixed 100 equip Nominal Never Fixed 100 calcard Nominal Never Fixed 100 vireless Nominal Never Fixed 100 longmon Continuous 18 4 None Never Fixed 100 longmon Continuous 9 1 None Never Fixed 100 cardmon Continuous 10 None Never Fixed 100 equipmon Continuous 1 None Never Fixed 100 viresmon Continuous 1 None Never Fixed 100 viresmon Continuous 18 None Never Fix	🌶 logtoll	🞸 Reclassify No	de		0 None		Blank & Null Val	Fixed	47.5	
age Divone Never Fixed 100 address Derive Node 0 None Never Fixed 100 income Graph Output 0 None Never Fixed 100 ed Graph Output 0 None Never Fixed 100 equip O Noninal Never Fixed 100 equip O Noninal Never Fixed 100 callcard Nominal Never Fixed 100 vireless Nominal Never Fixed 100 vireless Nominal Never Fixed 100 longmon Continuous 18 4 None Never Fixed 100 cardmon Continuous 9 1 None Never Fixed 100 vireness Continuous 11 3 None Never Fixed 100 vireness Continuous 2 0 None Never Fixed 100 cardmon Continuous 11 3 None Never Fixed 100 vireness Continuous	🎐 tenure	Ø Dississ Made			0 None		Never	Fixed	100	
Income 6 None Never Fixed 100 engloy Graph Output Never Fixed 100 engloy Graph Node 0 None Never Fixed 100 equip Nominal Never Fixed 100 callcard Nominal Never Fixed 100 vireless Nominal Never Fixed 100 longmon Continuous 18 4 None Never Fixed 100 longmon Continuous 9 1 None Never Fixed 100 longmon Continuous 11 3 None Never Fixed 100 virences Continuous 11 3 None Never Fixed 100 virences Continuous 11 3 None Never Fixed 100 virences Continuous 18 2 None Never Fixed 100 voice Nominal Never <	🌶 age	Binning Node			0 None		Never	Fixed	100	
ed Graph Output	address	Derive Node			0 None		Never	Fixed	100	
ed	income	Graph Order d			6 None		Never	Fixed	100	
equip Nominal - - - Never Fixed 100 calcard Nominal - - Never Fixed 100 vireless Nominal - - Never Fixed 100 longmon Continuous 18 4 None Never Fixed 100 longmon Continuous 9 1 None Never Fixed 100 equipmon Continuous 9 1 None Never Fixed 100 equipmon Continuous 11 3 None Never Fixed 100 viremon Continuous 8 1 None Never Fixed 100 viremon Continuous 20 4 None Never Fixed 100 toitten Continuous 18 2 None Never Fixed 100 toitten Continuous 11 6 None Never Fixed 100 conten Continuous 11 6 None Never Fixed 100	ed						Never	Fixed	100	
Calicard Nominal Never Fixed 100 Wreless Nominal Never Fixed 100 Iongmon Continuous 18 4 None Never Fixed 100 Iongmon Continuous 9 1 None Never Fixed 100 Lollmon Continuous 9 1 None Never Fixed 100 Logland Continuous 2 0 None Never Fixed 100 Continuous 11 3 None Never Fixed 100 Veremon Continuous 8 1 None Never Fixed 100 Veremon Continuous 20 4 None Never Fixed 100 Iongten Continuous 18 2 None Never Fixed 100 Iongten Continuous 11 6 None Never Fixed 100 Iongten Continuous 11 6 None Never Fixed 100 Voice Nominal -	employ	🞸 Graph Node			0 None		Never	Fixed	100	
wireless Nominal Never Fixed 100 longmon Continuous 18 4 None Never Fixed 100 tolmon Continuous 9 1 None Never Fixed 100 tolmon Continuous 9 1 None Never Fixed 100 continuous 2 0 None Never Fixed 100 cardmon Continuous 11 3 None Never Fixed 100 ordmon Continuous 8 1 None Never Fixed 100 ordernon Continuous 8 1 None Never Fixed 100 ordernon Continuous 18 2 None Never Fixed 100 cardten Continuous 11 6 None Never Fixed 100 cardten Continuous 11 6 None Never Fixed 100 cardten Nominal Never Fixed 100 reardten Nominal	👌 equip	on Nominal					Never	Fixed	100	
longmon Continuous 18 4 None Never Fixed 100 longmon Continuous 9 1 None Never Fixed 100 equipmon Continuous 2 0 None Never Fixed 100 equipmon Continuous 11 3 None Never Fixed 100 wiremon Continuous 8 1 None Never Fixed 100 wiremon Continuous 8 1 None Never Fixed 100 longten Continuous 18 2 None Never Fixed 100 cardten Continuous 18 2 None Never Fixed 100 cardten Continuous 18 2 None Never Fixed 100 cardten Continuous 11 6 None Never Fixed 100 voice Nominal Never Fixed 100 redrem Noninal Never Fixed 100 cardtw	callcard	💑 Nominal					Never	Fixed	100	
tollmon Continuous 9 1 None Never Fixed 100 equipmon Continuous 2 0 None Never Fixed 100 cardmon Continuous 11 3 None Never Fixed 100 wiremon Continuous 8 1 None Never Fixed 100 ongten Continuous 8 1 None Never Fixed 100 olongten Continuous 20 4 None Never Fixed 100 tollten Continuous 18 2 None Never Fixed 100 cardten Continuous 11 6 None Never Fixed 100 voice Nominal Never Fixed 100 outcet Nominal Never Fixed 100 internet Nominal Never Fixed 100 callwait Nominal Never Fixed 100 confer	> wireless	💑 Nominal					Never	Fixed	100	
equipmon Continuous 2 0 None Never Fixed 100 cardmon Continuous 11 3 None Never Fixed 100 wiremon Continuous 8 1 None Never Fixed 100 longten Continuous 20 4 None Never Fixed 100 totaten Continuous 20 4 None Never Fixed 100 totaten Continuous 18 2 None Never Fixed 100 cardten Continuous 11 6 None Never Fixed 100 voice Nominal Never Fixed 100 voice Nominal Never Fixed 100 internet Nominal Never Fixed 100 cardwait Nominal Never Fixed 100 conter Nominal Never Fixed 100 conter Nomi	longmon	🔗 Continuous	18		4 None		Never	Fixed	100	
Verardmon Continuous 11 3 None Never Fixed 100 Verardmon Continuous 8 1 None Never Fixed 100 Iongten Continuous 20 4 None Never Fixed 100 Iongten Continuous 20 4 None Never Fixed 100 Iongten Continuous 18 2 None Never Fixed 100 voice Nominal Never Fixed 100 voice Nominal Never Fixed 100 reardward Nominal Never Fixed 100 reardward Nominal Never Fixed 100 callwait Nominal Never Fixed 100 confer Nominal Never Fixed 100 confer Nominal Never Fixed 100 ebill Nominal Never Fixed 100 <t< td=""><td>tollmon</td><td>🔗 Continuous</td><td>9</td><td></td><td>1 None</td><td></td><td>Never</td><td>Fixed</td><td>100</td><td></td></t<>	tollmon	🔗 Continuous	9		1 None		Never	Fixed	100	
wiremon Continuous 8 1 None Never Fixed 100 longten Continuous 20 4 None Never Fixed 100 longten Continuous 20 4 None Never Fixed 100 longten Continuous 18 2 None Never Fixed 100 cardten Continuous 11 6 None Never Fixed 100 voice Nominal Never Fixed 100 pager Nominal Never Fixed 100 internet Nominal Never Fixed 100 callwait Nominal Never Fixed 100 confer Nominal <	👌 equipmon	🔗 Continuous	2		0 None		Never	Fixed	100	
Iongten Continuous 20 4 None Never Fixed 100 Ioliten Continuous 18 2 None Never Fixed 100 Cardten Continuous 11 6 None Never Fixed 100 Voice Nominal Never Fixed 100 Voice Nominal Never Fixed 100 Internet Nominal Never Fixed 100 Callwait Nominal Never Fixed 100 Confer Nominal Never Fixed 100 Coldon Never Fixed 100	cardmon 🏷	🔗 Continuous	11		3 None		Never	Fixed	100	
tollen Continuous 18 2 None Never Fixed 100 carden Continuous 11 6 None Never Fixed 100 voice Nominal Never Fixed 100 voice Nominal Never Fixed 100 voice Nominal Never Fixed 100 internet Nominal Never Fixed 100 callwat Nominal Never Fixed 100 callwat Nominal Never Fixed 100 callwat Nominal Never Fixed 100 confer Nominal Never Fixed 100 coll Sonfer Nominal Never Fixed 100 coll Sonfe Nominal	> wiremon	🔗 Continuous	8		1 None		Never	Fixed	100	
Continuous 11 6 None Never Fixed 100 voice Nominal Never Fixed 100 pager Nominal Never Fixed 100 internet Nominal Never Fixed 100 callwait Nominal Never Fixed 100 confer Nominal Never Fixed 100 confer Nominal Never Fixed 100 confer Nominal Never Fixed 100 bill Nominal Never Fixed 100 confer Nominal Never Fixed 100	longten	🔗 Continuous	20		4 None		Never	Fixed	100	
voice Nominal Never Fixed 100 pager Nominal Never Fixed 100 Internet Nominal Never Fixed 100 Internet Nominal Never Fixed 100 callwait Nominal Never Fixed 100 confer Nominal Never Fixed 100 ebill Nominal Never Fixed 100 loglong Continuous 4 0 None Never Fixed 100	toliten	🔗 Continuous	18		2 None		Never	Fixed	100	
pager Nominal Never Fixed 100 Internet Nominal Never Fixed 100 Internet Nominal Never Fixed 100 Callwait Nominal Never Fixed 100 Confer Nominal Never Fixed 100 belil Nominal Never Fixed 100 logiong Confinuous 4 O None Never Fixed 100	cardten	🔗 Continuous	11		6 None		Never	Fixed	100	
Internet Nominal Never Fixed 100 callwait Nominal Never Fixed 100 callwait Nominal Never Fixed 100 confer Vominal Never Fixed 100	> voice	💑 Nominal					Never	Fixed	100	
Collwait Mominal Never Fixed 100 Confer Nominal Never Fixed 100 Solid Never Fixed 100 Solid Never Fixed 100 Solid Never Fixed 100	> pager	💑 Nominal					Never	Fixed	100	
Confer Nominal Never Fixed 100 bebili Nominal Never Fixed 100 boglong Continuous 4 0 None Never Fixed 100	👌 internet	💑 Nominal					Never	Fixed	100	
Debili Image: Second sec	👌 callvvait	💑 Nominal					Never	Fixed	100	
loglong 🔗 Continuous 4 0 None Never Fixed 100	confer >	💑 Nominal					Never	Fixed	100	
	👌 ebill	💑 Nominal					Never	Fixed	100	
	loglong		4		0 None		Never	Fixed	100	
Ininc 🖉 Continuous 9 0 None Never Fixed 100	Ininc	🔗 Continuous	9		0 None		Never	Fixed	100	

In the Missing Values SuperNode dialog box, increase the Sample Size to 50% and click OK.

The SuperNode is displayed on the stream canvas, with the title: Missing Value Imputation.

• Attach the SuperNode to the Filter node.



► Add a Logistic node to the SuperNode.

Telecommunications Churn (Binomial Logistic Regression)

► In the Logistic node, click the Model tab and select the Binomial procedure. In the *Binomial Procedure* area, select the Forwards method.

Figure 13-10 Choosing model of	otions			
😡 churn				X
			0	-
Fields Model Expert	Analyze	Annotations		
Model name: 💿 Auto 🔇) Custom			
👿 Use partitioned data				
🛛 🗹 Build model for each s	plit			
Procedure: O Multinor	nial		Binomial	
Binomial Procedure Method: Forwards Categorical Inputs:	~			
Field Name	Contrast		Base Category	
Include constant in equ	uation			
OK 🕨 Run 🔇	Cancel		Apply	Reset

 On the Expert tab, select the Expert mode and then click Output. The Advanced Output dialog box is displayed.

► In the Advanced Output dialog, select At each step as the *Display* type. Select Iteration history and Parameter estimates and click OK.

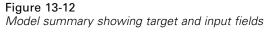
💡 Logistic R	egression: Advance	d Output 🛛 🛛 🔀
Display:	🔘 At each step	🔘 At last step
👿 Iteration hist	ory	🔽 Parameter estimates
📃 Classificatio	n plots	📕 Hosmer-Lemeshow goodness-of-fit
CI for exp(B) (%)	95 🗬
🗾 Residual Diag	nosis	
🔘 Outlie	ers outside (std. dev.):	2.0 🜲
🔘 All ca	ases	
Classification cut	off:	0.5 🚔

Browsing the Model

• On the Logistic node, click Run to create the model.

The model nugget is added to the stream canvas, and also to the Models palette in the upper-right corner. To view its details, right-click on the model nugget and select Edit or Browse.

The Summary tab shows (among other things) the target and inputs (predictor fields) used by the model. Note that these are the fields that were actually chosen based on the Forwards method, not the complete list submitted for consideration.





The items shown on the Advanced tab depend on the options selected on the Advanced Output dialog box in the Logistic node. One item that is always shown is the Case Processing Summary, which shows the number and percentage of records included in the analysis. In addition, it lists

164

the number of missing cases (if any) where one or more of the input fields are unavailable and any cases that were not selected.

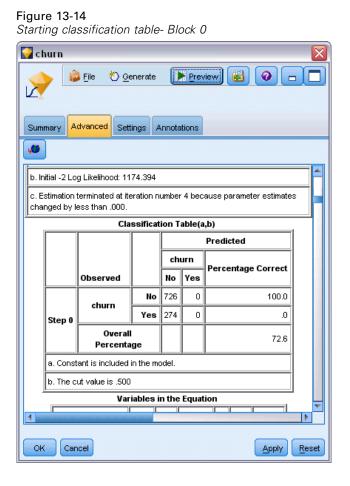
Figure 13-13 Case processing sun	nmary								
🔽 churn 🛛 🔀									
File 🖔 Generate 🕞 Preview 👜 🥝 🗖 🗖									
Summary Advanced Set	tings Ani	notations							
Logistic Regression									
Unweighted Cases(a)									
	Include	d in Anaț	ysis	1000	100.0				
Selected Cases Missing Cases				0	.0				
	Total								
Unselected Cases 0 .0									
Total 1000 100.0									
a. If weight is in effect, see classification table for the total number of cases.									
	ndent Var			1					
Origi	nal Value	Internal	Value						
	No		0			-			
1		1	_		•				
OK Cancel Apply Reset									

Scroll down from the Case Processing Summary to display the Classification Table under Block
 0: Beginning Block.

The Forward Stepwise method starts with a null model - that is, a model with no predictors - that can be used as a basis for comparison with the final built model. The null model, by convention, predicts everything as a 0, so the null model is 72.6% accurate simply because the 726 customers

Telecommunications Churn (Binomial Logistic Regression)

who didn't churn are predicted correctly. However, the customers who did churn aren't predicted correctly at all.



▶ Now scroll down to display the Classification Table under Block 1: Method = Forward Stepwise.

This Classification Table shows the results for your model as a predictor is added in at each of the steps. Already, in the first step - after just one predictor has been used - the model has increased the accuracy of the churn prediction from 0.0% to 29.9%

churi	n						
	Ŵ	File 🛛 🕙 Ge	enerate	D	Prev	riew) 🔒 📀 🗖	
-Y							
ummary	, A	dvanced Set	ings A	Innota	tions		
		CI	assifica	tion 1	ſable(a)	-
						Predicted	
					urn	Percentage Correct	rrect
		Observed		No	Yes		
		churn	No	668	58	92.0	-
Ste	ep 1 📙		Yes	192	82	29.9	
		Overal Percenta				75.0	
		churn	No	657	69	90.5	
Ste	ep 2	churn	Yes	160	114	41.6	
	-	Overal Percenta				77.1	
		churn	No	661	65	91.0	
Ste	ep 3		Yes	153	121	44.2	,

► Scroll down to the bottom of this Classification Table.

The Classification Table shows that the last step is step 8. At this stage the algorithm has decided that it no longer needs to add any further predictors into the model. Although the accuracy of the non-churning customers has decreased a little to 91.2%, the accuracy of the prediction for those

Telecommunications Churn (Binomial Logistic Regression)

who did churn has risen from the original 0% to 47.1%. This is a significant improvement over the original null model that used no predictors.

Figure 13-16 Classification table - Block 1										
🔽 churn 🛛 🔀										
File 🖏 Generate 🕞 Preview 🗟 🥥 🗖 🗖										
Summary Advanced Settings Annotations										
		je					78.7			
		churn	No	657	69					
	Step 7		Yes	144	130		47.4			
		Overall Percentag	je				78.7			
		No	662	64		91.2				
	Step 8	churn	Yes	145	129		47.1			
		Overall Percentag	je				79.1			
	a. The cut value is .500									
		Varia	ables i		<u> </u>					1
			B	S.E		ald	đf	Sig.	Exp(B)	
	Step 1(a)		046			.346	1	.000	.955	-
4		Constant	463	136	SII 11	574	1	001	1 587	•
OK Cancel Apply Reset										

For a customer who wants to reduce churn, being able to reduce it by nearly half would be a major step in protecting their income streams.

Note: This example also shows how taking the Overall Percentage as a guide to a model's accuracy may, in some cases, be misleading. The original null model was 72.6% accurate overall, whereas the final predicted model has an overall accuracy of 79.1%; however, as we have seen, the accuracy of the actual individual category predictions were vastly different.

To assess how well the model actually fits the data, a number of diagnostics are available in the Advanced Output dialog box when you are building the model. Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the *SPSS Modeler Algorithms Guide*, available from the *\Documentation* directory of the installation disk.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.

Forecasting Bandwidth Utilization (Time Series)

Forecasting with the Time Series Node

An analyst for a national broadband provider is required to produce forecasts of user subscriptions in order to predict utilization of bandwidth. Forecasts are needed for each of the local markets that make up the national subscriber base. You will use time series modeling to produce forecasts for the next three months for a number of local markets. A second example shows how you can convert source data if it is not in the correct format for input to the Time Series node.

These examples use the stream named *broadband_create_models.str*, which references the data file named *broadband_l.sav*. These files are available from the *Demos* folder of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *broadband_create_models.str* file is in the *streams* folder.

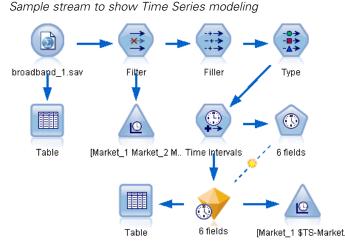
The last example demonstrates how to apply the saved models to an updated dataset in order to extend the forecasts by another three months.

In SPSS Modeler, you can produce multiple time series models in a single operation. The source file you'll be using has time series data for 85 different markets, although for the sake of simplicity you will only model five of these markets, plus the total for all markets.

The *broadband_l.sav* data file has monthly usage data for each of 85 local markets. For the purposes of this example, only the first five series will be used; a separate model will be created for each of these five series, plus a total.

The file also includes a date field that indicates the month and year for each record. This field will be used in a Time Intervals node to label records. The date field reads into SPSS Modeler as a string, but in order to use the field in SPSS Modeler you will convert the storage type to numeric Date format using a Filler node.

Figure 14-1



[©] Copyright IBM Corporation 1994, 2012.

The Time Series node requires that each series be in a separate column, with a row for each interval. SPSS Modeler provides methods for transforming data to match this format if necessary.

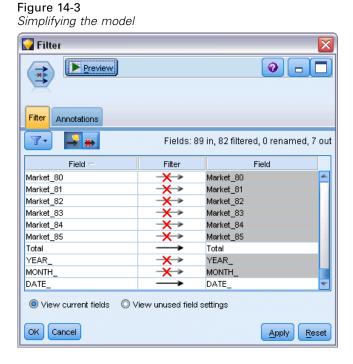
違 File	📄 Edit	🏷 <u>G</u> ener	rate 🪺		1 4 88			G))>
Table ,	Annotations								
	Market_1	Market_2	Market_3	Market_4	Market_5	Market_6	Market_7	Market_8	Mar
1	3750	11489	11659	4571	2205	5488	6144	2363	504:4
2	3846	11984	12228	4825	2301	5672	6390	2404	5160
3	3894	12266	12897	5041	2352	5802	6670	2469	5232
4	4010	12801	13716	5211	2490	5899	6929	2574	540:
5	4147	13291	14647	5383	2534	6017	7312	2654	554:
6	4335	13828	15419	5496	2664	6137	7493	2699	577:
7	4554	14273	16108	5747	2738	6250	7702	2786	5904
8	4744	14664	16958	5885	2754	6439	7965	2847	6032
9	4885	15130	17642	6053	2874	6701	8107	2967	6150
10	5020	15851	18453	6229	2975	6957	8366	3099	634:
11	5208	16509	19181	6320	3042	7111	8684	3195	663:
12	5379	17225	19885	6499	3095	7275	8997	3341	676
13	5574	18173	20565	6593	3199	7380	9326	3376	7021
14	5828	19287	21155	6680	3207	7633	9543	3443	733
15	5942	20171	21655	6757	3298	7985	9673	3617	749
16	6139	21379	21964	6804	3387	8236	9934	3732	7716
17	6244	22067	22756	6915	3450	8464	10211	3831	7946
18	6274	23074	23464	7035	3528	8575	10440	3886	829:
19	6347	23729	24324	7151	3546	8817	10763	3938	8584
20	6399	24803	25351	7304	3604	9041	11012	3953	8711
	4								

Figure 14-2 Monthly subscription data for broadband local markets

Creating the Stream

- Create a new stream and add a Statistics File source node pointing to *broadband_1.sav*.
- ► Use a Filter node to filter out the *Market_6* to *Market_85* fields and the *MONTH_* and *YEAR_* fields to simplify the model.

Tip: To select multiple adjacent fields in a single operation, click the *Market_6* field, hold down the left mouse button and drag the mouse down to the *Market_85* field. Selected fields are highlighted in blue. To add the other fields, hold down the Ctrl key and click the *MONTH_* and *YEAR_* fields.



Examining the Data

It is always a good idea to have a feel for the nature of your data before building a model. Do the data exhibit seasonal variations? Although the Expert Modeler can automatically find the best seasonal or nonseasonal model for each series, you can often obtain faster results by limiting the search to nonseasonal models when seasonality is not present in your data. Without examining the data for each of the local markets, we can get a rough picture of the presence or absence of seasonality by plotting the total number of subscribers over all five markets.

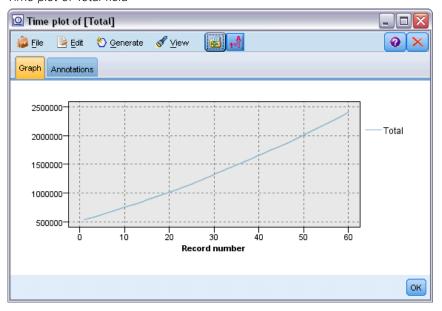
Figure 14-4

Plotting t	he total	number	of subscribe	ərs

[]	X
	0
Plot Appearance Output Annotations	
Plot: O Selected series Selected Time Series models	
Series:	
X axis label: 🔘 Default 🔘 Custom	-
🔲 Display series in separate panels 🛛 🕅 Normalize	
Display: 👿 Line	
Point	
Smoother	
☑ Limit records Maximum number of records to plot: 2	000 🗲
OK Run Cancel	Apply Reset

- ▶ From the Graphs palette, attach a Time Plot node to the Filter node.
- ► Add the *Total* field to the Series list.
- Deselect the Display series in separate panels and Normalize check boxes.
- ► Click Run.

Figure 14-5 Time plot of Total field



The series exhibits a very smooth upward trend with no hint of seasonal variations. There might be individual series with seasonality, but it appears that seasonality is not a prominent feature of the data in general.

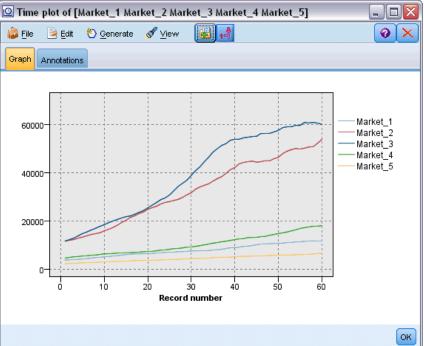
Of course you should inspect each of the series before ruling out seasonal models. You can then separate out series exhibiting seasonality and model them separately.

IBM® SPSS® Modeler makes it easy to plot multiple series together.

Figure 14-6 Plotting multiple time series 😡 [Total] 0 - -0 Plot Appearance Output Annotations Plot: Selected series O Selected Time Series models 🔗 Market_1 Series: 🔗 Market_2 🔗 Market 3 -1 X axis label: 🔘 Default 🔘 Custom 📃 Display series in separate panels 🛛 📃 Normalize Display: 👿 Line 🔲 Point Smoother 2000 ≑ 🔽 Limit records 🛛 Maximum number of records to plot: OK 🕨 Run Cancel Apply Reset

- ► Reopen the Time Plot node.
- Remove the *Total* field from the Series list (select it and click the red X button).
- ► Add the *Market 1* through *Market 5* fields to the list.
- ► Click Run.





Inspection of each of the markets reveals a steady upward trend in each case. Although some markets are a little more erratic than others, there is no evidence of seasonality to be seen.

Defining the Dates

Now you need to change the storage type of the DATE_ field to Date format.

- Attach a Filler node to the Filter node.
- Open the Filler node and click the field selector button.
- ► Select DATE_ to add it to Fill in fields.
- ► Set the Replace condition to Always.

► Set the value of Replace with to to_date(DATE_).

Figure 14-8

Setting the date storage type

🔽 Filler	
Preview)	0
Settings Annotations	
Fill in fields:	
& DATE_	×
Replace: Always	
Condition:	
@BLANK(@FIELD)	
Replace with:	
to_date(DATE_)	
OK Cancel	Apply Reset

Change the default date format to match the format of the Date field. This is necessary for the conversion of the Date field to work as expected.

• On the menu, choose Tools > Stream Properties > Options to display the Stream Options dialog box.

► Set the default Date format to MON YYYY .

Figure 14-9

Setting the date format

😡 broadband_create	e_mode	ls						
								0
		arameters	Deployment	Script	Globals	Search	Comments	Annotations
	Radiar		Degrees					
	Date/T		String					
Date format:	MON YYY	ΥY	*					
Time format:	HH:MM:SS	S	*	C Rol	lover days	s/mins		
Number display format:	Standard	(###.###)	*					
Standard decimal places:	:	3 ≑						
Scientific decimal places:		з≑ Сі	urrency decim	al places		2 ≑		
Decimal symbol:	Period ((.) 🔻 Gr	ouping symbo	Ŀ	None		•	
Date baseline (1st Jan):	1900	0 🚔 2-	digit dates sta	t from:	193	10 ≑		
Encoding:	System d	lefault 🔻						
Maximum number of rows	to show	in Data Prev	view:	1	0 ≑			
📝 Maximum set size				25	50 ≑			
V Limit set size for Neura	al, Kohone	en and K-Me	ans modeling	2	20 ≑			
Ruleset Evaluation:	Voting 📑	•						
Refresh source nodes	on execu	ution						
Display field and value	labels in	output						
Save As Default								
OK Cancel							Ar	pply <u>R</u> eset

Defining the Targets

► Add a Type node and set the role to None for the *DATE*_field. Set the role to Target for all others (the *Market_n* fields plus the *Total* field).

• Click the Read Values button to populate the Values column.

Figure 14-10

Setting the role for multiple fields

	Annotations		0	
×- 00 e	🛰 🚺 🕨 Read Value	es Clear Values	Clear All Values	
Field -	Measurement	Values Missing	Check	Role
🚫 Market_1	🔗 Continuous	[3750,117	None 🧕) Target
🔆 Market_2	🔗 Continuous	[11489,53	None 🧕	Target
🔆 Market_3	🔗 Continuous	[11659,60	None 🧕	Target
🔆 Market_4	🔗 Continuous	[4571,179	None 🧕	Target
🔆 Market_5	🔗 Continuous	[2205,6611]	None 🧕	Target
🔆 Total	🔗 Continuous	[536413,2	None 🧕	Target
DATE_	🔗 Continuous	[1999-01	None 6) None
View current OK Cancel	fields 🔘 View unuser	d field settings	lqq <u>A</u>	y <u>R</u> eset

Setting the Time Intervals

- Add a Time Intervals node (from the Field Operations palette).
- ▶ On the Intervals tab, select Months as the time interval.
- ► Select the Build from data option.

► Select DATE_ as the build field.

-			
Figure	14-1		
Setting	the	time	interval

🚺 Time Intervals 🛛 🔯
Periodicity: 12
Intervals Build Estimation Forecast Annotations
Time Interval: Months
◯ Start labeling from first record ◉ Build from data
Field: 🖉 DATE_
New field name extension: \$TIAdd as: Prefix O Suffix
OK Cancel Apply Reset

- On the Forecast tab, select the Extend records into the future check box.
- ► Set the value to 3.

► Click OK.

Figure 14-12 Setting the forecast period	
💟 Time Intervals	
Periodicity: 12	
Intervals Build Estimation Forecast	nnotations
Extend records into the future	
Future indicator field: \$TI_Futu	re
Future Values to use in Forecasting Select fields whose values you wish to adv	d to the data:
Field	Values
	×
OK Cancel	Apply

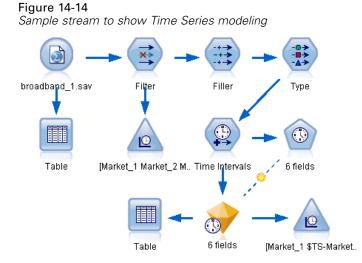
Creating the Model

► From the Modeling palette, add a Time Series node to the stream and attach it to the Time Intervals node.

 Click Run on the Time Series node using all default settings. Doing so enables the Expert Modeler to decide the most appropriate model to use for each time series.

Figure 14-13 Choosing the Expert Modeler for Time Series	
🔽 6 fields	×
Periodicity: 12	0
Fields Model Annotations	
Model name: 🔘 Auto 🔘 Custom	
Continue estimation using existing model(s)	
Method: Expert Modeler Criteria	
Estimate model using all records Forecast 3 records beyond estimation period	
Confidence limit width (%):	
Maximum number of lags in ACF and PACF output:	
Build scoring model only	
OK Run Cancel	Apply <u>R</u> eset

- ► Attach the Time Series model nugget to the Time Intervals node.
- Attach a Table node to the Time Series model and click Run.



There are now three new rows (61 through 63) appended to the original data. These are the rows for the forecast period, in this case January to March 2004.

Several new columns are also present now—a number of *\$TI_* columns added by the Time Intervals node and the *\$TS-* columns added by the Time Series node. The columns indicate the following for each row (i.e., each interval in the time series data):

Column	Description
\$TI_TimeIndex	The time interval index value for this row.
\$TI_TimeLabel	The time interval label for this row.
\$TI_Year	The year and month indicators for the generated data in this row.
\$TI_Month	
\$TI_Count	The number of records involved in determining the new data for this row.
\$TI_Future	Indicates whether this row contains forecast data.
\$TS-colname	The generated model data for each column of the original data.
\$TSLCI-colname	The lower confidence interval value for each column of the generated model data.
\$TSUCI-colname	The upper confidence interval value for each column of the generated model data.
\$TS-Total	The total of the \$TS-colname values for this row.
\$TSLCI-Total	The total of the \$TSLCI- <i>colname</i> values for this row.
\$TSUCI-Total	The total of the \$TSUCI-colname values for this row.

The most significant columns for the forecast operation are the *\$TS-Market_n*, *\$TSLCI-Market_n*, and *\$TSUCI-Market_n* columns. In particular, these columns in rows 61 through 63 contain the user subscription forecast data and confidence intervals for each of the local markets.

Examining the Model

 Double-click the Time Series model nugget to display data about the models generated for each of the markets.

Note how the Expert Modeler has chosen to generate a different type of model for Market 5 from the type it has generated for the other markets.

Figure 14-15

Time Series models generated for the markets

)	e 🏷 <u>G</u> enerate	Previe				0	-)[
1odel Paramete	rs Residuals S	Summary Sett	ings Annotation	s			
Sor	t by Selected	•	View: Sim	ple 🔽	D 14		
imber of records	used in estimation	:60					
Target 🗠	Model	Predictors	StationaryR**2	Q	df	Sig.	
🚺 Market_1	Holts linear tr	0	0.264	8.53	16.0	0.931	
🚺 Market_2	Holts linear tr	0	0.121	35.9	16.0	0.003	
🚺 Market_3	Holts linear tr	0	0.258	15.76	16.0	0.47	
🚺 Market_4	Holts linear tr	0	0.25	27.714	16.0	0.034	
🚺 Market_5	Winters addit	0	0.544	11.888	15.0	0.688	
	Winters addit Holts linear tr	0 0	0.544 0.049	11.888 27.616	15.0 16.0	0.688 0.035	
				27.616			
			0.049 Summary Statist	27.616		0.035	
	Holts linear tr		0.049	27.616	16.0 df		
Total	Hotts linear tr		0.049 Summary Statist StationaryR**2	27.616 ics Q 21.235	16.0 df 15.833	0.035 Sig.	
Z Total	Hotts linear tr		0.049 Summary Statist StationaryR**2 0.247	27.616 ics Q 21.235 10.738	16.0 df 15.833 0.408	0.035 Sig. 0.36	
Total SUMMARY SUMMARY	Hotts linear tr Statistic MEAN SE		0.049 Summary Statist StationaryR**2 0.247 0.169	27.616 ics Q 21.235 10.738 8.53	16.0 df 15.833 0.408 15	0.035 Sig. 0.36 0.396	
Total SUMMARY SUMMARY SUMMARY	Holts linear tr Statistic MEAN SE MINIMUM		0.049 Summary Statist StationaryR**2 0.247 0.169 0.049	27.616 ics 21.235 10.738 8.53 35.9	16.0 df 15.833 0.408 15 15	0.035 Sig. 0.36 0.396 0.003	
Total SUMMARY SUMMARY SUMMARY SUMMARY	Holts linear tr Statistic MEAN SE MINIMUM MAXIMUM		0.049 Summary Statist StationaryR**2 0.247 0.169 0.049 0.544	27.616 Q 21.235 10.738 8.53 35.9 8.53	16.0 df 15.833 0.408 15 16 15	0.035 Sig. 0.36 0.396 0.003 0.931	
Total SUMMARY SUMMARY SUMMARY SUMMARY	Holts linear tr Statistic MEAN SE MINIMUM MAXIMUM PERCENTILE 5		0.049 Summary Statist StationaryR**2 0.247 0.169 0.049 0.544 0.049	27.616 ics Q 21.235 10.738 8.53 35.9 8.53 8.53	16.0 df 15.833 0.408 15 16 15 15	0.035 Sig. 0.36 0.396 0.003 0.931 0.003	
Total SUMMARY SUMMARY SUMMARY SUMMARY SUMMARY	Holts linear tr Statistic MEAN SE MINIMUM MAXIMUM PERCENTILE 5 PERCENTILE		0.049 Summary Statist StationaryR**2 0.247 0.169 0.049 0.544 0.049 0.049	27.616 ics Q 21.235 10.738 8.53 355.9 8.53 8.53 8.53 11.048	16.0 df 15.833 0.408 15 16 15 15 15 15,75	0.035 Sig. 0.36 0.396 0.003 0.931 0.003 0.003	
Total SUMMARY SUMMARY SUMMARY SUMMARY SUMMARY SUMMARY	Holts linear tr Statistic MEAN SE MINIMUM MAXIMUM PERCENTILE 5 PERCENTILE PERCENTILE		0.049 Summary Statist StationaryR**2 0.247 0.169 0.049 0.049 0.049 0.049 0.049	27.616 Q 21.235 10.738 8.53 35.9 8.53 10.9 8.53 11.048 21.688	16.0 df 15.833 0.408 15 16 15 15 15 15,75	0.035 Sig. 0.396 0.033 0.003 0.003 0.003 0.003 0.003 0.003	
Total SUMMARY SUMMARY SUMMARY SUMMARY SUMMARY SUMMARY SUMMARY	Holts linear tr Statistic MEAN SE MINIMUM MAXIMUM PERCENTILE 5 PERCENTILE PERCENTILE		0.049 Summary Statist StationaryR**2 0.247 0.169 0.049 0.049 0.049 0.049 0.049 0.049 0.049 0.049	27.616 Q 21.235 10.738 8.53 35.9 8.53 8.53 8.53 11.048 21.688 29.761	16.0 df 15.833 0.408 15 15 15 15 15 15 15 15 15 16 16 16	0.035 Sig. 0.36 0.036 0.003 0.931 0.003 0.003 0.026 0.252	
Total SUMMARY SUMMARY SUMMARY SUMMARY SUMMARY SUMMARY SUMMARY SUMMARY	Holts linear tr Statistic MEAN SE MINIMUM MAXIMUM PERCENTILE 5 PERCENTILE PERCENTILE PERCENTILE		0.049 Summary Statist StationaryR**2 0.247 0.169 0.049 0.544 0.049 0.049 0.049 0.049 0.103 0.254 0.334	27.616 Q 21.235 10.738 8.53 35.9 8.53 11.048 21.688 29.761 35.9	16.0 df 15.833 0.408 15 15 15 15 15 5 15.75 16 16 16 16	0.035 Sig. 0.36 0.036 0.031 0.031 0.033 0.003 0.026 0.252 0.749	

The Predictors column shows how many fields were used as predictors for each target—in this case, none.

The remaining columns in this view show various goodness-of-fit measures for each model. The Stationary R^{**2} column shows the Stationary *R*-squared value. This statistic provides an estimate of the proportion of the total variation in the series that is explained by the model. The higher the value (to a maximum of 1.0), the better the fit of the model.

The Q, df, and Sig. columns relate to the Ljung-Box statistic, a test of the randomness of the residual errors in the model—the more random the errors, the better the model is likely to be. Q is the Ljung-Box statistic itself, while df (degrees of freedom) indicates the number of model parameters that are free to vary when estimating a particular target.

The Sig. column gives the significance value of the Ljung-Box statistic, providing another indication of whether the model is correctly specified. A significance value less than 0.05 indicates that the residual errors are not random, implying that there is structure in the observed series that is not accounted for by the model.

Taking both the Stationary *R*-squared and Significance values into account, the models that the Expert Modeler has chosen for *Market_1*, *Market_3*, and *Market_5* are quite acceptable. The Sig. values for *Market_2* and *Market_4* are both less than 0.05, indicating that some experimentation with better-fitting models for these markets might be necessary.

The summary values in the lower part of the display provide information on the distribution of the statistics across all models. For example, the mean Stationary *R*-squared value across all the models is 0.247, while the minimum such value is 0.049 (that of the *Total* model) and the maximum is 0.544 (the value for *Market_5*).

SE denotes the standard error across all the models for each statistic. For example, the standard error for Stationary *R*-squared across all models is 0.169.

The summary section also includes percentile values that provide information on the distribution of the statistics across models. For each percentile, that percentage of models have a value of the fit statistic below the stated value.

Thus for example, only 25% of the models have a Stationary *R*-squared value that is less than 0.121.

Click the View drop-down list and select Advanced.

The display shows a number of additional goodness-of-fit measures. R^{*2} is the *R*-squared value, an estimation of the total variation in the time series that can be explained by the model. As the maximum value for this statistic is 1.0, our models are fine in this respect.

6 fie	lds							2
)	i 🖗 E	ile 🕙 <u>G</u> ene	erate [🔭 Pr	eview 🛃				》□□□
Model	Paramet				tations			
umber o		ort by Selectors		View:	Advanced		A 1억	
M	APE	MAE	MaxAPE	MaxAE	Norm. BIC	Q	df	Sig.
,	0.94	73.869	9 2.147	224.517	9.15	8.53	3 16.0	0.93
ì	0.94	314.721	1.867	927.949	12.059	35.9	9 16.0	0.00
1	0.776	306.877	1.918	1,030.105	12.1	15.76	3 16.0	0.4
;	0.78	79.49	9 1.942	233.544	9.329	27.714	l 16.0	0.03
	0.936	39.963	3 2.481	137.633	8.114			
	0.094	1,326.071	0.299	7,062.662	15.243	27.616	6 16.0	
				Summary S	Statistics			
MAPE		MAE	MaxAPE	MaxAE	Norm. BIC	Q	df	Sig.
	0.744	356.832	1.776	1,602.735	10.999	21.235	15.833	0.36
	0.328	490.119	0.758	2,702.397	2.641	10.738	0.408	0.396
	0.094	39.963	0.299	137.633	8.114	8.53	15	0.003
	0.94	1,326.071	2.481	7,062.662	15.243	35.9	16	0.931
	0.094	39.963	0.299	137.633	8.114	8.53	15	0.003
	0.094	39.963	0.299	137.633	8.114	8.53	15	0.003
	0.605	65.393	1.475	202.796	8.891	11.048	15.75	0.026
	0.858	193.183	1.93	580.747	10.694	21.688	16	0.252
	0.94	567.559	2.231	2,538.245	12.886	29.761	16	0.749
	0.94	1,326.071	2.481	7,062.662	15.243	35.9	16	0.931
	0.94	1,326.071	2.481	7,062.662	15.243	35.9	16	0.931
4								•

Figure 14-16 Time Series models advanced display

RMSE is the root mean square error, a measure of how much the actual values of a series differ from the values predicted by the model, and is expressed in the same units as those used for the series itself. As this is a measurement of an error, we want this value to be as low as possible. At first sight it appears that the models for *Market_2* and *Market_3*, while still acceptable according to the statistics we have seen so far, are less successful than those for the other three markets.

These additional goodness-of-fit measure include the mean absolute percentage errors (MAPE) and its maximum value (MaxAPE). Absolute percentage error is a measure of how much a target series varies from its model-predicted level, expressed as a percentage value. By examining the mean and maximum across all models, you can get an indication of the uncertainty in your predictions.

The MAPE value shows that all models display a mean uncertainty of less than 1%, which is very low. The MaxAPE value displays the maximum absolute percentage error and is useful for imagining a worst-case scenario for your forecasts. It shows that the largest percentage error for each of the models falls in the range of roughly 1.8 to 2.5%, again a very low set of figures.

The MAE (mean absolute error) value shows the mean of the absolute values of the forecast errors. Like the RMSE value, this is expressed in the same units as those used for the series itself. MaxAE shows the largest forecast error in the same units and indicates worst-case scenario for the forecasts.

Interesting though these absolute values are, it is the values of the percentage errors (MAPE and MaxAPE) that are more useful in this case, as the target series represent subscriber numbers for markets of varying sizes.

Do the MAPE and MaxAPE values represent an acceptable amount of uncertainty with the models? They are certainly very low. This is a situation in which business sense comes into play, because acceptable risk will change from problem to problem. We'll assume that the goodness-of-fit statistics fall within acceptable bounds and go on to look at the residual errors.

Examining the values of the autocorrelation function (ACF) and partial autocorrelation function (PACF) for the model residuals provides more quantitative insight into the models than simply viewing goodness-of-fit statistics.

A well-specified time series model will capture all of the nonrandom variation, including seasonality, trend, and cyclic and other factors that are important. If this is the case, any error should not be correlated with itself (autocorrelated) over time. A significant structure in either of the autocorrelation functions would imply that the underlying model is incomplete.

Click the Residuals tab to display the values of the autocorrelation function (ACF) and partial autocorrelation function (PACF) for the residual errors in the model for the first of the local markets.

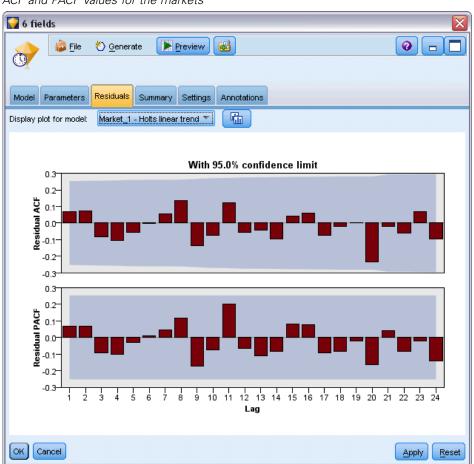


Figure 14-17 ACF and PACF values for the markets

In these plots, the original values of the error variable have been lagged by up to 24 time periods and compared with the original value to see if there is any correlation over time. For the model to be acceptable, none of the bars in the upper (ACF) plot should extend outside the shaded area, in either a positive (up) or negative (down) direction.

Should this occur, you would need to check the lower (PACF) plot to see whether the structure is confirmed there. The PACF plot looks at correlations after controlling for the series values at the intervening time points.

The values for *Market_l* are all within the shaded area, so we can continue and check the values for the other markets.

 Click the Display plot for model drop-down list to display these values for the other markets and the totals. The values for *Market_2* and *Market_4* give a little cause for concern, confirming what we suspected earlier from their Sig. values. We'll need to experiment with some different models for those markets at some point to see if we can get a better fit, but for the rest of this example, we'll concentrate on what else we can learn from the *Market_1* model.

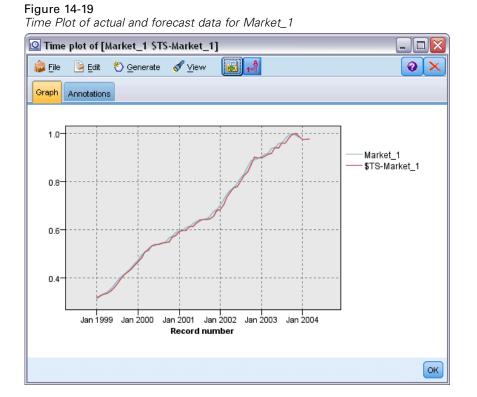
- ▶ From the Graphs palette, attach a Time Plot node to the Time Series model nugget.
- On the Plot tab, uncheck the Display series in separate panels check box.
- ► At the Series list, click the field selector button, select the *Market_1* and *\$TS-Market_1* fields, and click OK to add them to the list.
- Click Run to display a line graph of the actual and forecast data for the first of the local markets.

Selecting the fields to plot
[1] 🔽
Plot Appearance Output Annotations
Plot: O Selected series Selected Time Series models
Series: Series:
X axis label: O Default O Custom
🔲 Display series in separate panels 🛛 Normalize
Display: 👿 Line
Point
Smoother
☑ Limit records Maximum number of records to plot: 2000 🗲
OK Run Cancel <u>Apply</u> <u>R</u> eset

Figure 14-18

Notice how the forecast (*\$TS-Market_1*) line extends past the end of the actual data. You now have a forecast of expected demand for the next three months in this market.

The lines for actual and forecast data over the entire time series are very close together on the graph, indicating that this is a reliable model for this particular time series.



Save the model in a file for use in a future example:

- Click OK to close the current graph.
- ► Open the Time Series model nugget.
- ► Choose File > Save Node and specify the file location.
- Click Save.

You have a reliable model for this particular market, but what margin of error does the forecast have? You can get an indication of this by examining the confidence interval.

- Double-click the last Time Plot node in the stream (the one labeled Market_1 \$TS-Market_1) to open its dialog box again.
- Click the field selector button and add the *\$TSLCI-Market_1* and *\$TSUCI-Market_1* fields to the Series list.

► Click Run.

Figure 14-2 Adding more	0 e fields to plot	
😡 [Market_1	\$TS-Market_1 \$TSLCI-	Market_1 \$TSUCI-Marke 🔀
		0 - 🗆
Plot Appeara	nce Output Annotations	
Plot: 🔘 Sele	cted series 🔘 Selected Time	Series models
Series:	 ✓ Market_1 ✓ \$TS-Market_1 ✓ \$TSLCI-Market 1 	
X axis label:	🖲 Default 🔘 Custom	-
📄 Display ser	ies in separate panels 🛛 👿 N	ormalize
Display: 👿 l	ine	
🗖 F	Point	
: 🗖	Smoother	
👿 Limit record	ls Maximum number of recor	rds to plot: 2000 🗲
OK 🕨 Ru	Cancel	Apply Reset

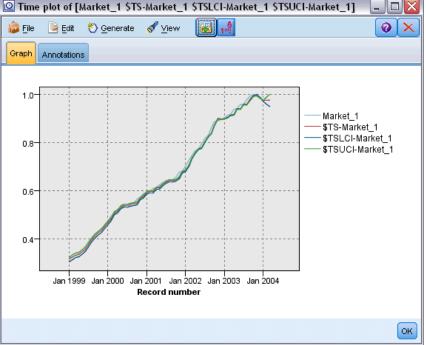
Now you have the same graph as before, but with the upper (*\$TSUCI*) and lower (*\$TSLCI*) limits of the confidence interval added.

Notice how the boundaries of the confidence interval diverge over the forecast period, indicating increasing uncertainty as you forecast further into the future.

However, as each time period goes by, you will have another (in this case) month's worth of actual usage data on which to base your forecast. You can read the new data into the stream and reapply your model now that you know it is reliable. For more information, see the topic Reapplying a Time Series Model on p. 190.

Figure 14-21





Summary

You have learned how to use the Expert Modeler to produce forecasts for multiple time series, and you have saved the resulting models to an external file.

In the next example, you will see how to transform nonstandard time series data into a format suitable for input to a Time Series node.

Reapplying a Time Series Model

This example applies the time series models from the first time series example but can also be used independently. For more information, see the topic Forecasting with the Time Series Node on p. 168.

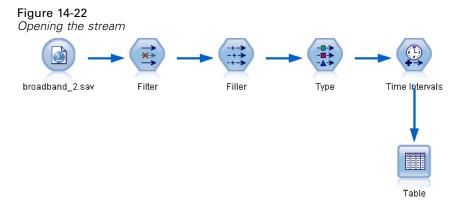
As in the original scenario, an analyst for a national broadband provider is required to produce monthly forecasts of user subscriptions for each of a number of local markets, in order to predict bandwidth requirements. You have already used the Expert Modeler to create models and to forecast three months into the future. Your data warehouse has now been updated with the actual data for the original forecast period, so you would like to use that data to extend the forecast horizon by another three months.

This example uses the stream named *broadband_apply_models.str*, which references the data file named *broadband_2.sav*. These files are available from the *Demos* folder of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *broadband_apply_models.str* file is in the *streams* folder.

Retrieving the Stream

In this example, you'll be recreating a Time Series node from the Time Series model saved in the first example. Don't worry if you don't have a model saved—we've provided one in the *Demos* folder.

▶ Open the stream *broadband apply models.str* from the *streams* folder under *Demos*.



Fi	guı	ſе	14	-23	

Updated sales data

違 File		🛓 Edit 🛛 🕙	<u>G</u> enerate						0	×
Table	Ann	otations								
	1	Market_82	Market_83	Market_84	Market_85	Total	YEAR_	MONTH_	DATE_	
44		58820	20482	14326	16935	17917	2002	8	AUG 2002	4
45		60119	21211	14349	17179	18249	2002	9	SEP 2002	Γ
46		61320	21893	14333	17601	18601	2002	10	OCT 2002	L
47		63099	22471	14229	17816	18945	2002	11	NOV 2002	
48		64687	23112	14514	17937	19343	2002	12	DEC 2002	
49		65518	23686	14856	18003	19752	2003	1	JAN 2003	
50		65570	24669	15182	17875	20148	2003	2	FEB 2003	
51		66567	25469	15709	18214	20540	2003	3	MAR 2003	
52		67527	25868	16155	18557	20922	2003	4	APR 2003	L
53		67724	26284	16521	19190	21300	2003	5	MAY 2003	
54		68644	26468	16567	19938	21669	2003	6	JUN 2003	L
55		69878	26781	16618	20876	22004	2003	7	JUL 2003	L
56		71538	27566	16553	21514	22398	2003	8	AUG 2003	Ľ
57		73162	28164	16597	21779	22773	2003	9	SEP 2003	
58		74167	28693	16669	22266	23160	2003	10	OCT 2003	
59		76036	28922	16748	22559	23616	2003	11	NOV 2003	
60		76630	29811	16798	23018	24067	2003	12	DEC 2003	
61		79002	30034	17122	23160	24509	2004	1	JAN 2004	
62		81123	30091	17581	23698	24968	2004	2	FEB 2004	
63		83909	30162	17894	24355	25383	2004	3	MAR 2004	
	1									

The updated monthly data is collected in *broadband_2.sav*.

Attach a Table node to the IBM® SPSS® Statistics File source node, open the Table node and click Run.

Note: The data file has been updated with the actual sales data for January through March 2004, in rows 61 to 63.

- Open the Time Intervals node on the stream.
- ► Click the Forecast tab.

• Ensure that Extend records into the future is set to 3.

Figure 14-24

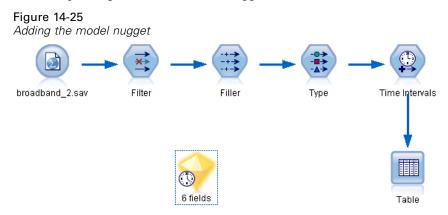
Checking the setting of the forecast period

💟 Time Intervals	\mathbf{X}
Preview	0
Periodicity: 12	
Intervals Build Estimation Forecast Annotations	
Extend records into the future 3	
Future indicator field: \$TI_Future	
Future Values to use in Forecasting Select fields whose values you wish to add to the data:	
Field Values	
Titita Yandos	
OK Cancel	Apply Reset

Retrieving the Saved Model

On the IBM® SPSS® Modeler menu, choose Insert > Node From File and select the *TSmodel.nod* file from the *Demos* folder (or use the Time Series model you saved in the first time series example).

This file contains the time series models from the previous example. The insert operation places the corresponding Time Series model nugget on the canvas.

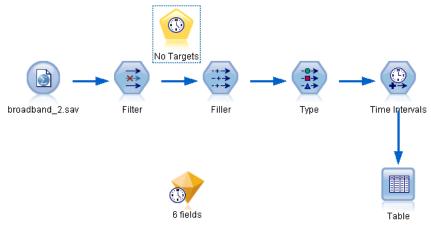


Generating a Modeling Node

▶ Open the Time Series model nugget and choose Generate > Generate Modeling Node.

This places a Time Series modeling node on the canvas.

Figure 14-26Generating a modeling node from the model nugget



Generating a New Model

• Close the Time Series model nugget and delete it from the canvas.

The old model was built on 60 rows of data. You need to generate a new model based on the updated sales data (63 rows).

► Attach the newly generated Time Series build node to the stream.

Figure 14-27 Attaching the modeling node to the stream



Figure 14-28

Reusing stored settings for the time series model

🔽 6 fields	
	0
Periodicity: 12	
Fields Model Annotations	
Model name: 💿 Auto 🔘 Custom	
Continue estimation using existing model(s)	
Method: Expert Modeler Criteria	
Confidence limit width (%):	
Maximum number of lags in ACF and PACF output: 24 🖨	
Build scoring model only	
	\square
OK Nun Cancel	Apply Reset

- ► Open the Time Series node.
- On the Model tab, ensure that Continue estimation using existing models is checked.
- Click Run to place a new model nugget on the canvas and in the Models palette.

Examining the New Model

Figure	14-29
--------	-------

Table showing new forecast

じ File	📄 <u>E</u> dit 🛛 🕙	<u>G</u> enerate		1.4 8	8		0 ×
Table 🗛	nnotations						
2//////////////////////////////////////	\$TI_TimeLabel	\$TI_Year	\$TI_Month	\$TI_Count	\$TI_Future	\$TS-Market_1	\$TSLCI-Market_1
47	Nov 2002	2002	11	1	0	10552	10365 🔶
48	Dec 2002	2002	12	1	0	10593	10406
49	Jan 2003	2003	1	1	0	10653	10466
50	Feb 2003	2003	2	1	0	10740	10553
51	Mar 2003	2003	3	1	0	10851	10664
52	Apr 2003	2003	4	1	0	10909	10722
53	May 2003	2003	5	1	0	11153	10966
54	Jun 2003	2003	6	1	0	11178	10991
55	Jul 2003	2003	7	1	0	11382	11195
56	Aug 2003	2003	8	1	0	11408	11221
57	Sep 2003	2003	9	1	0	11627	11440
58	Oct 2003	2003	10	1	0	11795	11608
59	Nov 2003	2003	11	1	0	11869	11682
60	Dec 2003	2003	12	1	0	11793	11607
61	Jan 2004	2004	1	1	0	11686	11500
62	Feb 2004	2004	2	1	0	11896	11710
63	Mar 2004	2004	3	1	0	11996	11810
64	Apr 2004	2004	4	0	1	12278	12056
65	May 2004	2004	5	0	1	12416	12100
66	Jun 2004	2004	6	0	1	12553	12167 🔫
	4						

- Attach a Table node to the new Time Series model nugget on the canvas.
- Open the Table node and click Run.

The new model still forecasts three months ahead because you're reusing the stored settings. However, this time it forecasts April through June because the estimation period (specified on the Time Intervals node) now ends in March instead of January.

(DVC)

Figure 14-30 Specifying fields to plot

	_
	0
Plot Appearance Output Annotations	
Plot: O Selected series O Selected Time Series models	
Series:	
X axis label: 🔘 Default 🔘 Custom	-
👿 Display series in separate panels 🛛 🗑 Normalize	
Display: 👿 Line	
Point Distance Contemporation	
Smoother	
Limit records Maximum number of records to plot:	2000 🗲
OK 🕨 Run Cancel	Apply Reset

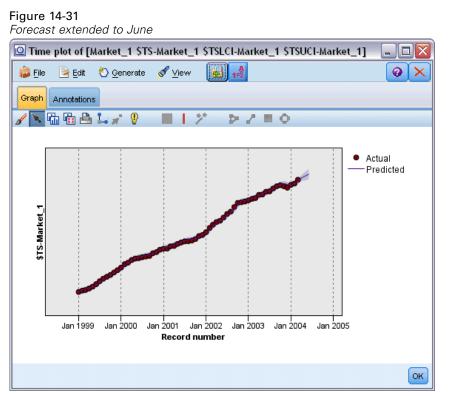
► Attach a Time Plot graph node to the Time Series model nugget.

This time we'll use the time plot display designed especially for time series models.

- ▶ On the Plot tab, choose the Selected Time Series models option.
- ► At the Series list, click the field selector button, select the *\$TS-Market_1* field and click OK to add it to the list.
- ► Click Run.

Now you have a graph that shows the actual sales for *Market_l* up to March 2004, together with the forecast (Predicted) sales and the confidence interval (indicated by the blue shaded area) up to June 2004.

As in the first example, the forecast values follow the actual data closely throughout the time period, indicating once again that you have a good model.



Summary

You have learned how to apply saved models to extend your previous forecasts when more current data becomes available, and you have done this without rebuilding your models. Of course, if there is reason to think that a model has changed, you should rebuild it.



Forecasting Catalog Sales (Time Series)

A catalog company is interested in forecasting monthly sales of its men's clothing line, based on their sales data for the last 10 years.

This example uses the stream named *catalog_forecast.str*, which references the data file named *catalog_seasfac.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *catalog_forecast.str* file is in the *streams* directory.

We've seen in an earlier example how you can let the Expert Modeler decide which is the most appropriate model for your time series. Now it's time to take a closer look at the two methods that are available when choosing a model yourself—exponential smoothing and ARIMA.

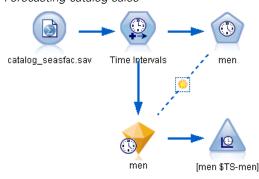
To help you decide on an appropriate model, it's a good idea to plot the time series first. Visual inspection of a time series can often be a powerful guide in helping you choose. In particular, you need to ask yourself:

- Does the series have an overall trend? If so, does the trend appear constant or does it appear to be dying out with time?
- Does the series show seasonality? If so, do the seasonal fluctuations seem to grow with time or do they appear constant over successive periods?

Creating the Stream

Create a new stream and add a Statistics File source node pointing to catalog seasfac.sav.

Figure 15-1 Forecasting catalog sales



Catalog_seasfac.sav Preview Preview Refresh \$CLEO_DEMOS/catalog_seasfac.sav										
Data Filter <mark>1</mark>	ypes Annotations	alues Clea	ar Values	Clear All V	alues					
Field -	Measurement	Values	Missing	Check	Role					
🖬 date	🔗 Continuous	[0000-12		None	None					
🋞 men	🔗 Continuous	[3245.18,		None	🔘 Target					
🛞 women	🔗 Continuous	[16578.9		None	O None					
🋞 jewel	🔗 Continuous	[5983.55,		None	🛇 None					
🔆 mail	🔗 Continuous	[1147,15		None	🛇 None					
🔆 page	🔗 Continuous	[51,114]		None	🛇 None					
🔆 phone	🔗 Continuous	[17,59]		None	🛇 None					
🤣 print	🔗 Continuous	[18061.2,		None	🛇 None					
🔆 service	🔗 Continuous	[15,68]		None	🛇 None					
VEAR_ 💑 Nominal 1989,199 None 🛇 None 🚽										
View current fields View unused field settings										

- ▶ Open the IBM® SPSS® Statistics File source node and select the Types tab.
- ► Click Read Values, then OK.
- Click the *Role* column for the *men* field and set the role to Target.
- ► Set the role for all the other fields to None, and click OK.

Forecasting Catalog Sales (Time Series)

Figure 15-3 Setting the time interval

💟 Time Intervals	
(B) Preview	0
Periodicity: 12	
Intervals Build Estimation Forecast Annotations	
Time Interval: Months	
◯ Start labeling from first record ⊚ Build from data	
Field: 🔗 date	-
New field name extension: \$TI_	Add as: 💿 Prefix 🔘 Suffix
OK Cancel	Apply Reset

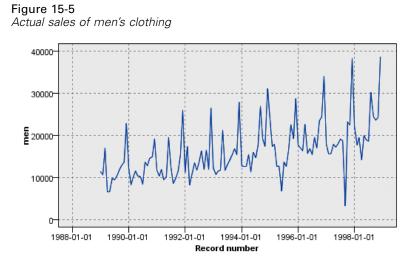
- Attach a Time Intervals node to the SPSS Statistics File source node.
- ► Open the Time Intervals node and set Time Interval to Months.
- Select Build from data.
- ► Set Field to date, and click OK.

Figure 15-4
Plotting the time series

[9] 🛛 🔀
Plot Appearance Output Annotations
Plot: Selected series Selected Time Series models
Series:
X axis label: Default O Custom
☑ Display series in separate panels Normalize
Display: 👿 Line
🥅 Point
Smoother
Limit records Maximum number of records to plot: 2000 😴
OK Run Cancel Apply Reset

- Attach a Time Plot node to the Time Intervals node.
- ► On the Plot tab, add men to the Series list.
- ► Deselect the Normalize check box.
- ► Click Run.

Examining the Data



The series shows a general upward trend; that is, the series values tend to increase over time. The upward trend is seemingly constant, which indicates a linear trend.

The series also has a distinct seasonal pattern with annual highs in December, as indicated by the vertical lines on the graph. The seasonal variations appear to grow with the upward series trend, which suggests multiplicative rather than additive seasonality.

Click OK to close the plot.

Now that you've identified the characteristics of the series, you're ready to try modeling it. The exponential smoothing method is useful for forecasting series that exhibit trend, seasonality, or both. As we've seen, your data exhibit both characteristics.

Exponential Smoothing

Building a best-fit exponential smoothing model involves determining the model type—whether the model needs to include trend, seasonality, or both—and then obtaining the best-fit parameters for the chosen model.

The plot of men's clothing sales over time suggested a model with both a linear trend component and a multiplicative seasonality component. This implies a Winters model. First, however, we will explore a simple model (no trend and no seasonality) and then a Holt model (incorporates linear trend but no seasonality). This will give you practice in identifying when a model is not a good fit to the data, an essential skill in successful model building.

Figure 15-6 Specifying exponential smoothing

🔽 men	×
	0
Periodicity: 12	
Fields Model Annotations	
Model name: 🔘 Auto 🔘 Custom	
Continue estimation using existing model(s)	
Method: Exponential Smoothing Criteria	
Estimate model using all records No forecast period specified	
Confidence limit width (%):	
Maximum number of lags in ACF and PACF output: 24 🚔	
Build scoring model only	
OK F Run Cancel	Apply Reset

We'll start with a simple exponential smoothing model.

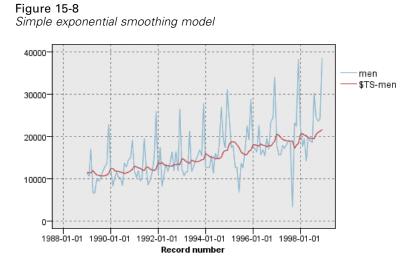
- Attach a Time Series node to the Time Intervals node. ►
- On the Model tab, set Method to Exponential Smoothing. ►
- Click Run to create the model nugget.

Forecasting Catalog Sales (Time Series)



🔽 [men \$TS-men]
Plot Appearance Output Annotations
Plot: Selected series Selected Time Series models
A men
Series: Series:
X axis label: 💿 Default 🔘 Custom 📃 🚽
Display series in separate panels
Display: 👿 Line
Point
Smoother
Limit records Maximum number of records to plot: 2000
OK Run Cancel Apply Reset

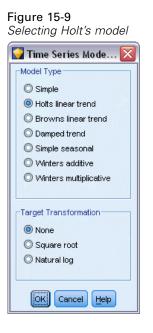
- Attach a Time Plot node to the model nugget.
- ▶ On the Plot tab, add *men* and *\$TS-men* to the Series list.
- Deselect the Display series in separate panels and Normalize check boxes.
- ► Click Run.



The men plot represents the actual data, while \$TS-men denotes the time series model.

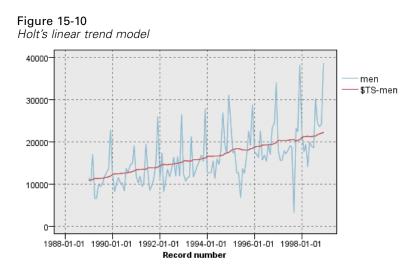
Although the simple model does, in fact, exhibit a gradual (and rather ponderous) upward trend, it takes no account of seasonality. You can safely reject this model.

► Click OK to close the time plot window.



Let's try Holt's linear model. This should at least model the trend better than the simple model, although it too is unlikely to capture the seasonality.

- ► Reopen the Time Series node.
- ▶ On the Model tab, with Exponential Smoothing still selected as the method, click Criteria.
- ▶ On the Exponential Smoothing Criteria dialog box, choose Holts linear trend.
- ► Click OK to close the dialog box.
- ► Click Run to re-create the model nugget.
- ▶ Re-open the Time Plot node and click Run.



Holt's model displays a smoother upward trend than the simple model but it still takes no account of the seasonality, so you can discard this one too.

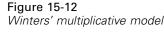
► Close the time plot window.

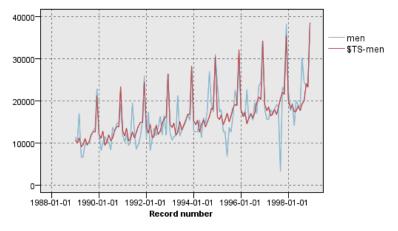
You may recall that the initial plot of men's clothing sales over time suggested a model incorporating a linear trend and multiplicative seasonality. A more suitable candidate, therefore, might be Winters' model.

Figure 15-11 Selecting Winters' model
🔽 Time Series Mode 🔀
Model Type
O Simple
O Holts linear trend
O Browns linear trend
O Damped trend
O Simple seasonal
Winters additive
Winters multiplicative
Target Transformation
O None
🔘 Square root
◯ Natural log
OK Cancel Help

- ► Reopen the Time Series node.
- On the Model tab, with Exponential Smoothing still selected as the method, click Criteria.
- On the Exponential Smoothing Criteria dialog box, choose Winters multiplicative.

- Click OK to close the dialog box.
- Click Run to re-create the model nugget.
- Open the Time Plot node and click Run.





This looks better-the model reflects both the trend and the seasonality of the data.

The dataset covers a period of 10 years and includes 10 seasonal peaks occurring in December of each year. The 10 peaks present in the predicted results match up well with the 10 annual peaks in the real data.

However, the results also underscore the limitations of the Exponential Smoothing procedure. Looking at both the upward and downward spikes, there is significant structure that is not accounted for.

If you are primarily interested in modeling a long-term trend with seasonal variation, then exponential smoothing may be a good choice. To model a more complex structure such as this one, we need to consider using the ARIMA procedure.

ARIMA

The ARIMA procedure allows you to create an autoregressive integrated moving-average (ARIMA) model suitable for finely tuned modeling of time series. ARIMA models provide more sophisticated methods for modeling trend and seasonal components than do exponential smoothing models, and they allow the added benefit of including predictor variables in the model.

Continuing the example of the catalog company that wants to develop a forecasting model, we have seen how the company has collected data on monthly sales of men's clothing along with several series that might be used to explain some of the variation in sales. Possible predictors include the number of catalogs mailed and the number of pages in the catalog, the number of phone lines open for ordering, the amount spent on print advertising, and the number of customer service representatives.

Are any of these predictors useful for forecasting? Is a model with predictors really better than one without? Using the ARIMA procedure, we can create a forecasting model with predictors, and see if there is a significant difference in predictive ability over the exponential smoothing model with no predictors.

The ARIMA method enables you to fine-tune the model by specifying orders of autoregression, differencing, and moving average, as well as seasonal counterparts to these components. Determining the best values for these components manually can be a time-consuming process involving a good deal of trial and error, so for this example, we'll let the Expert Modeler choose an ARIMA model for us.

We'll try to build a better model by treating some of the other variables in the dataset as predictor variables. The ones that seem most useful to include as predictors are the number of catalogs mailed (*mail*), the number of pages in the catalog (*page*), the number of phone lines open for ordering (*phone*), the amount spent on print advertising (*print*), and the number of customer service representatives (*service*).

Setting the p	oredictor fields					
📀 catalog_se	asfac.sav				8	
	Preview 2 Refresh)			0	
	D_DEMOS/catalog_seasf	ac.sav				
~	🗪 🚺 🕨 Read Val	ues Clea	r Values	Clear All Va	alues	
Field -	Measurement	Values	Missing	Check	Role	
🗰 date	🔗 Continuous	[0000-12	_	None	🛇 None 🛛 🔼	
🛞 men	🔗 Continuous	[3245.18,		None	🔘 Target 🛛	
🛞 women	🔗 Continuous	[16578.9		None	🛇 None	
🛞 jewel	🔗 Continuous	[5983.55,		None	🛇 None	
🔷 mail	🔗 Continuous	[1147,15		None	🔪 Input	
🚫 page	🔗 Continuous	[51,114]		None	🔪 Input	
🚫 phone	🔗 Continuous	[17,59]		None	🔪 Input 👘	
🛞 print	🔗 Continuous	[18061.2,		None	🔪 Input	
🔆 service	🔗 Continuous	[15,68]		None	🔪 Input	
🔆 YEAR_	💑 Nominal	1989,199		None	🛇 None 🛛 🔽	
View current fields View unused field settings						
OK Cance	el				Apply Reset	

- ► Open the IBM® SPSS® Statistics File source node.
- ▶ On the Types tab, set the *Role* for *mail*, *page*, *phone*, *print*, and *service* to Input.
- Ensure that the role for men is set to Target and that all the remaining fields are set to None.
- ► Click OK.

Figure 15-13



🔽 men	×
	0
Periodicity: 12	
Fields Model Annotations	
Model name: 🔘 Auto 🔘 Custom	
Continue estimation using existing model(s)	
Method: Expert Modeler Criteria	
Estimate model using all records No forecast period specified	
Confidence limit width (%):	
Maximum number of lags in ACF and PACF output: 24	
Build scoring model only	
OK Run Cancel	Apply Reset

- ► Open the Time Series node.
- ► On the Model tab, set Method to Expert Modeler and click Criteria.

Forecasting Catalog Sales (Time Series)

	e Series	Modeler: Expert Modeler Cri	teria
Model	Outliers		
Model	Туре		
O AI	l models		
O E>	ponential	smoothing models only	
() AI	RIMA mode	els only	
VE:	kpert Mode	eler considers seasonal models	
Events	s and Inter	ventions	
	Fie	eld	
	and interv	vention fields are special independant	fields that are
Event		effects of external occurrences such	
	to model e		as a nood, strike,
used			
used		of a new product line. Check all fields	that you want

- On the Expert Modeler Criteria dialog box, choose the ARIMA models only option and ensure that Expert Modeler considers seasonal models is checked.
- Click OK to close the dialog box.
- Click Run on the Model tab to re-create the model nugget.

Figure 15-16	
Expert Modeler chooses to	vo predictors
💟 men	

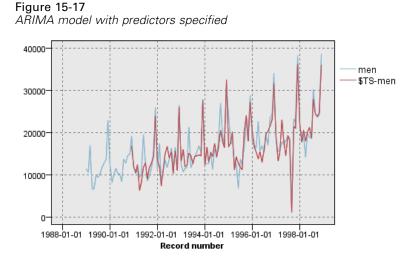
🔶 🚺 Eile 🔌 Ge	nerate [🔭 Preview) 🧯			0	
Model Parameters Residu	als Summary Settings /	Innotations			
Sort by Sele	cted 🔽 🔽 Vi	ew: Simple 💌	1		
Number of records used in est	mation:120				
Target 🚣 🛛 Mode	I Predictors Station	aryR**2 Q	df	Sig.	
📝 men 🛛 ARIMA(0,	D,O 2	0.731 19.455	i 17.0	0.303	

► Open the model nugget.

Notice how the Expert Modeler has chosen only two of the five specified predictors as being significant to the model.

• Click OK to close the model nugget.

▶ Open the Time Plot node and click Run.



This model improves on the previous one by capturing the large downward spike as well, making it the best fit so far.

We could try refining the model even further, but any improvements from this point on are likely to be minimal. We've established that the ARIMA model with predictors is preferable, so let's use the model we have just built. For the purposes of this example, we'll forecast sales for the coming year.

- ► Click OK to close the time plot window.
- Open the Time Intervals node and select the *Forecast* tab.
- Select the *Extend records into the future* checkbox and set its value to 12.

The use of predictors when forecasting requires you to specify estimated values for those fields in the forecast period, so that the modeler can more accurately forecast the target field.

Specifying future values for pred	dictor fields
🙀 Time Intervals	×
Preview	0
Periodicity: 12	
Intervals Build Estimation Forecast A	nnotations
Extend records into the future	
Future indicator field: \$TI_Futu	ure
Select fields whose values you wish to add	d to the data:
mail	Mean of recent points
page	Mean of recent points
phone	Mean of recent points
print	Mean of recent points
service	Mean of recent points
	Blank
	Mean of recent points Most recent value
	Specify
	Specify
OK Cancel	Apply

- ► In the Future Values to use in Forecasting group, click the field selector button to the right of the Values column.
- ▶ On the Select Fields dialog box, select mail through service and click OK.

In the real world, you would specify the future values manually at this point, since these five predictors all relate to items that are under your control. For the purposes of this example, we'll use one of the predefined functions, to save having to specify 12 values for each predictor. (When you're more familiar with this example, you might want to try experimenting with different future values to see what effect they have on the model.)

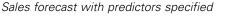
- ► For each field in turn, click the Values field to display the list of possible values and choose Mean of recent points. This option calculates the mean of the last three data points for this field and uses that as the estimated value in each case.
- ► Click OK.

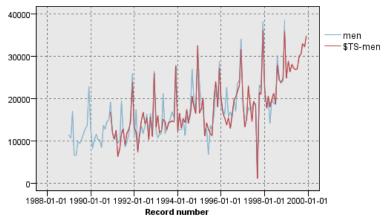
Figure 15-18

- ▶ Open the Time Series node and click Run to re-create the model nugget.
- ▶ Open the Time Plot node and click Run.

The forecast for 1999 looks good—as expected, there's a return to normal sales levels following the December peak, and a steady upward trend in the second half of the year, with sales in general significantly above those for the previous year.

Figure 15-19





Summary

You have successfully modeled a complex time series, incorporating not only an upward trend but also seasonal and other variations. You have also seen how, through trial and error, you can get closer and closer to an accurate model, which you have then used to forecast future sales.

In practice, you would need to reapply the model as your actual sales data are updated—for example, every month or every quarter—and produce updated forecasts. For more information, see the topic Reapplying a Time Series Model in Chapter 14 on p. 190.



Making Offers to Customers (Self-Learning)

The Self-Learning Response Model (SLRM) node generates and enables the updating of a model that allows you to predict which offers are most appropriate for customers and the probability of the offers being accepted. These sorts of models are most beneficial in customer relationship management, such as marketing applications or call centers.

This example is based on a fictional banking company. The marketing department wants to achieve more profitable results in future campaigns by matching the right offer of financial services to each customer. Specifically, the example uses a Self-Learning Response Model to identify the characteristics of customers who are most likely to respond favorably based on previous offers and responses and to promote the best current offer based on the results.

This example uses the stream *pm_selflearn.str*, which references the data files *pm_customer_train1.sav*, *pm_customer_train2.sav*, and *pm_customer_train3.sav*. These files are available from the *Demos* folder of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *pm_selflearn.str* file is in the *streams* folder.

Existing Data

The company has historical data tracking the offers made to customers in past campaigns, along with the responses to those offers. These data also include demographic and financial information that can be used to predict response rates for different customers.

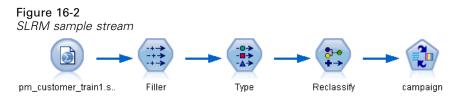
Figure 16-1

Responses to previous offers

길 <u>F</u> ile	📄 <u>E</u> dit 🛛 💐) <u>G</u> enerate						0	>
Table 🔎	Annotations								
	customer_id	campaign	response	response_date	purchase	purchase_date	product_id	Rowid	
	7	2	0	\$null\$	0	\$null\$	\$null\$	1	
2	13	2	0	\$null\$	0	\$null\$	\$null\$	2	F
3	15	2	0	\$null\$	0	\$null\$	\$null\$	3	
1	16	2	1	2006-07-05 00:00:00	0	\$null\$	183	761	
5	23	2	0	\$null\$	0	\$null\$	\$null\$	4	
6	24	2	0	\$null\$	0	\$null\$	\$null\$	5	
7	30	2	0	\$null\$	0	\$null\$	\$null\$	6	
3	30	3	0	\$null\$	0	\$null\$	\$null\$	7	ľ
9	33	2	0	\$null\$	0	\$null\$	\$null\$	8	I
10	42	3	0	\$null\$	0	\$null\$	\$null\$	9	ľ
11	42	2	0	\$null\$	0	\$null\$	\$null\$	10	I
12	52	2	0	\$null\$	0	\$null\$	\$null\$	11	L
13	57	2	0	\$null\$	0	\$null\$	\$null\$	12	L
4	63	2	1	2006-07-14 00:00:00	0	\$null\$	183	1501	I
15	74	2	0	\$null\$	0	\$null\$	\$null\$	13	
16	74	3	0	\$null\$	0	\$null\$	\$null\$	14	
17	75	2	0	\$null\$	0	\$null\$	\$null\$	15	
18	82	2	0	\$null\$	0	\$null\$	\$null\$	16	
19	89	3	0	\$null\$	0	\$null\$	\$null\$	17	
20	89	2	0	\$null\$	0	\$null\$	\$null\$	18	5
	4	and the second second						4	Г

Building the Stream

Add a Statistics File source node pointing to pm_customer_train1.sav, located in the Demos folder of your IBM® SPSS® Modeler installation.



- Add a Filler node and select campaign as the Fill in field.
- ► Select a Replace type of Always.

Making Offers to Customers (Self-Learning)

► In the Replace with text box, enter to_string(campaign) and click OK.

Figure 16-3 Derive a campaign field	
🔽 Filler	X
	0 - 🗖
Settings Annotations	
Fill in fields:	
🖋 campaign	×
Replace: Always	
Condition:	
@BLANK(@FIELD)	▲ ▼
Replace with:	
to_string(campaign)	
OK Cancel	Apply Reset

Figure 16-4

► Add a Type node, and set the *Role* to None for the *customer_id*, *response_date*, *purchase_date*, *product id*, *Rowid*, and *X random* fields.

Туре		<u>٦</u>		_	(
Types Format 4		Itations				0	
₹ - 00 00		► Read Values	Clear V	alues	Clear All Valu	Jes	
Field -		Measurement	Values	Missing	Check	Role	
🔆 customer_id	Ø	Continuous	[7,116993]		None	🛇 None	-
A campaign		Nominal	"1","2","3		None	🔘 Target	
🔆 response	8	Flag	1/0		None	🔘 Target	
🔁 response_date	Ø	Continuous	[2006-04		None	🛇 None	
🔆 purchase		Nominal	0,1		None	🔪 Input	
🔁 purchase_date	P	[*] Continuous	[2006-04		None	🛇 None	
짖 product_id	R	Continuous	[183,421]		None	🛇 None	
📿 Rowid	Ø	' Continuous	[1,19599]		None	🛇 None	
즞 age	Ø	Continuous	[10,96]		None	🔪 Input	
칮 age_youngest	Ø.	' Continuous	[0,66]		None	🔪 Input	-
View current field	lds	O View unused	field settings				

► Set the *Role* to Target for the *campaign* and *response* fields. These are the fields on which you want to base your predictions.

Set the Measurement to Flag for the response field.

► Click Read Values, then OK.

Because the campaign field data show as a list of numbers (1, 2, 3, and 4), you can reclassify the fields to have more meaningful titles.

- ► Add a Reclassify node to the Type node.
- ▶ In the Reclassify into field, select Existing field.
- ▶ In the Reclassify field list, select campaign.
- Click the Get button; the campaign values are added to the *Original value* column.
- ▶ In the *New value* column, enter the following campaign names in the first four rows:
 - Mortgage
 - Car loan
 - Savings
 - Pension

Making Offers to Customers (Self-Learning)

► Click OK.

Figure 16-5

Reclassify the	campaign n	ames	
😡 Reclassify			
	view		0
Settings Annotat	ions		
	Mode:	💿 Single 🔘 Multiple	
	Reclassify into:	🔘 New field 🥥 Existing field	
Reclassify field:			
💑 campaign			-
New field name:			
Reclassify3			
Reclassify values:			
🕨 Get	>>> Copy	🧷 Clear new	🗳 Auto
Original	value 🗆	New value	
1		Mortgage	
2		Car Ioan	
3		Savings	
4		Pension	
For unspecified va	lues use: 🛛 🔘 Or	iginal value 🔘 Default value	undef
OK Cancel			Apply Reset

• Attach an SLRM modeling node to the Reclassify node. On the Fields tab, select campaign for the Target field, and response for the Target response field.

Figure 16-6 Select the target	and target r	esponse	
💟 campaign			
			0
Fields Model Settin	gs Annotations		
Target field:	🗞 campaign		_]
Target response field:	🎖 response		-1
Use type node setting	gs 🔘 Use custom	settings	
Inputs:			-
			×
Partition:			-
Use frequency field			-
OK 🕨 Run Ca	ancel		Apply Reset

▶ On the Settings tab, in the Maximum number of predictions per record field, reduce the number to 2.

This means that for each customer, there will be two offers identified that have the highest probability of being accepted.

Making Offers to Customers (Self-Learning)

• Ensure that Take account of model reliability is selected, and click Run.

Figure 16-7 SLRM node settings			
😡 campaign			
		0	
Fields Model Settings A	nnotations		
Maximum number of predictions	per record: 2 🚔		
Level of randomization :	0.00 ≑		
Set random seed:	876547 ≑		
Sort order:			
	vith highest score will be returne		
	th lowest score will be returned	0	
Preferences for target fields:	Ductours	Aburrus is shude	
Value	Preference	Always include	Add Delete
			Delete
Take account of model relia	bility		
OK 🕨 Run Cancel		Apply	Reset

Browsing the Model

► Open the model nugget. The Model tab initially shows the estimated the accuracy of the predictions for each offer and the relative importance of each predictor in estimating the model.

To display the correlation of each predictor with the target variable, choose Association with Response from the View list in the right-hand pane.

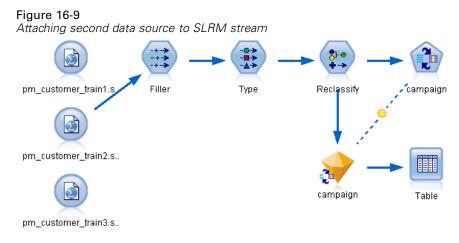
► To switch between each of the four offers for which there are predictions, select the required offer from the View list in the left-hand pane.



- ► Close the model nugget window.
- On the stream canvas, disconnect the IBM® SPSS® Statistics File source node pointing to pm_customer_train1.sav.

Making Offers to Customers (Self-Learning)

► Add a Statistics File source node pointing to *pm_customer_train2.sav*, located in the *Demos* folder of your IBM® SPSS® Modeler installation, and connect it to the Filler node.



• On the Model tab of the SLRM node, select Continue training existing model.

Figure 16-10

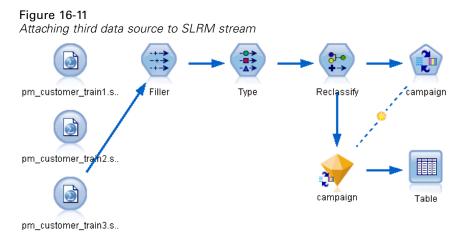
Continue training model

💟 campaign	×
Fields Model Settings Annotations	
Model name: O Auto O Custom	
☑ Use partitioned data	
Continue training existing model	
Target field values: 🔘 Use all 🔘 Specify	
	Add
	Edit
	Delete
Model Assessment	
Include model assessment	
Set random seed: 876547 ≑	
Simulated sample size: 100 🗲	
Number of iterations: 10	
V Display model evaluation	
OK Run Cancel Appl	ly <u>R</u> eset

• Click Run to re-create the model nugget. To view its details, double-click the nugget on the canvas.

The Model tab now shows the revised estimates of the accuracy of the predictions for each offer.

► Add a Statistics File source node pointing to *pm_customer_train3.sav*, located in the *Demos* folder of your SPSS Modeler installation, and connect it to the Filler node.



- Click Run to re-create the model nugget once more. To view its details, double-click the nugget on the canvas.
- ▶ The Model tab now shows the final estimated accuracy of the predictions for each offer.

As you can see, the average accuracy fell slightly (from 86.9% to 85.4%) as you added the additional data sources; however, this fluctuation is a minimal amount and may be attributed to slight anomalies within the available data.

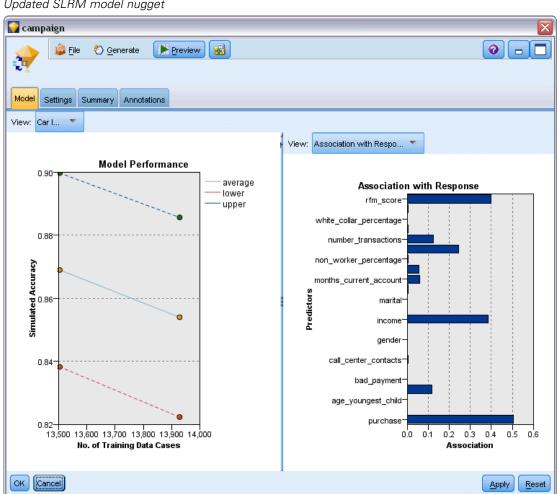


Figure 16-12 Updated SLRM model nugget

- Attach a Table node to the last (third) generated model and execute the Table node.
- Scroll across to the right of the table. The predictions show which offers a customer is most likely to accept and the confidence that they will accept, depending on each customer's details.

For example, in the first line of the table shown, there is only a 13.2% confidence rating (denoted by the value 0.132 in the *\$SC-campaign-1* column)) that a customer who previously took out a car loan will accept a pension if offered one . However, the second and third lines show two more customers who also took out a car loan; in their cases, there is a 95.7% confidence that they, and

226

other customers with similar histories, would open a savings account if offered one, and over 80% confidence that they would accept a pension.

Table (35 fields, 27 records)						
🝃 File 📄 Edit 🥙 Generate 🔢 🕒 📢 🏦						
Table	Annotations					
		X_random	\$S-campaign-1	\$SC-campaign-1	\$S-campaign-2	\$SC-campaign-2
		1	Pension	0.132	Mortgage	0.107
<u></u>		1	Savings	0.957	Pension	0.844
34/////		1	Savings	0.957	Pension	0.802
11/1/11	12	3	Pension	0.132	Mortgage	0.107
34/////		1	Pension	0.805	Savings	0.284
1//////		3	Pension	0.132	Mortgage	0.107
		2	Pension	0.132	Mortgage	0.107
34/////		3	Pension	0.132	Mortgage	0.107
		1	Pension	0.132	Mortgage	0.107
0 /////		1	Pension	0.132	Mortgage	0.107
1/////		2	Pension	0.132	Mortgage	0.107
2		2	Pension	0.132	Mortgage	0.107
3		2	Savings	0.957	Mortgage	0.829
4		2	Savings	0.164	Pension	0.132
5		2	Savings	0.957	Pension	0.868
B ////	12	2	Pension	0.132	Mortgage	0.107
7	22	3	Pension	0.132	Mortgage	0.107
8	<u>//</u>	3	Pension	0.132	Mortgage	0.107
9	12	3	Savings	0.289	Pension	0.132
0	12	2	Pension	0.132	Mortgage	0.107
	4					•
						_
						C

Figure 16-13 Model output - predicted offers and confidences

Explanations of the mathematical foundations of the modeling methods used in SPSS Modeler are listed in the *SPSS Modeler Algorithms Guide*, available from the *\Documentation* directory of the product DVD.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.

Predicting Loan Defaulters (Bayesian Network)

Bayesian networks enable you to build a probability model by combining observed and recorded evidence with "common-sense" real-world knowledge to establish the likelihood of occurrences by using seemingly unlinked attributes.

This example uses the stream named *bayes_bankloan.str*, which references the data file named *bankloan.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation and can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *bayes bankloan.str* file is in the *streams* directory.

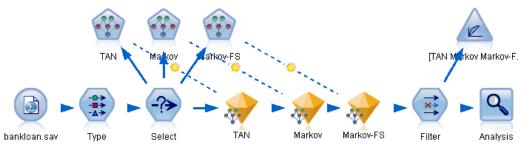
For example, suppose a bank is concerned about the potential for loans not to be repaid. If previous loan default data can be used to predict which potential customers are liable to have problems repaying loans, these "bad risk" customers can either be declined a loan or offered alternative products.

This example focuses on using existing loan default data to predict potential future defaulters, and looks at three different Bayesian network model types to establish which is better at predicting in this situation.

Building the Stream

► Add a Statistics File source node pointing to *bankloan.sav* in the *Demos* folder.

Figure 17-1 Bayesian Network sample stream



Add a Type node to the source node and set the role of the default field to Target. All other fields should have their role set to Input.

228

• Click the Read Values button to populate the *Values* column.

🖌 Туре					
	review				0
Types Format	Annotations				
4- 00	🗪 🚺 🕨 Read Va	lues Clear	^r Values	Clear All Va	alues
Field -	Measurement	Values	Missing	Check	Role
🔆 age	🔗 Continuous	[20,56]		None	🔪 Input 🛛 🧧
🔆 ed	📲 Ordinal	1,2,3,4,5		None	🔪 Input
关 employ	🔗 Continuous	[0,33]		None	🔪 Input
📿 address	🔗 Continuous	[0,34]		None	🔪 Input
🛞 income	🔗 Continuous	[13.0,446		None	🔪 Input
🛞 debtinc	🔗 Continuous	[0.1,41.3]		None	🔪 Input
🛞 creddebt	🔗 Continuous	[0.01169		None	🔪 Input
🛞 othdebt	🔗 Continuous	[0.04558		None	🔪 Input
🔆 default	🎖 Flag	1/0	1/0		🔘 Target
🛞 preddef1	🔗 Continuous	[1.16573		None	🔪 Input 🔄
V					

Cases where the target has a null value are of no use when building the model. You can exclude those cases to prevent them from being used in model evaluation.

- ► Add a Select node to the Type node.
- ► For Mode, select Discard.

Predicting Loan Defaulters (Bayesian Network)

► In the Condition box, enter default = '\$null\$'.

Figure 17- Discarding	3 null targets	
😡 Select		
	Preview	0
	nnotations	
Mode: (🔾 Include 🔘 Discard	
Condition:	default = '\$null\$'	
OK Cance	a)	Apply Reset

Because you can build several different types of Bayesian networks, it is worth comparing several to see which model provides the best predictions. The first one to create is a Tree Augmented Naïve Bayes (TAN) model.

- Attach a Bayesian Network node to the Select node.
- On the Model tab, for Model name, select Custom and enter TAN in the text box.

► For Structure type, select TAN and click OK.

Figure 17-4

Creating a Tree Augmented Naïve Bayes model

💟 TAN	\mathbf{X}
Fields Model Expert Analyze Annotations	0
Model name: O Auto O Custom TAN	
☑ Use partitioned data	
☑ Build model for each split	
To select fields manually, choose "Use custom settings" on the Fields tab-	
Partition:	-1
Splits:	×
Continue training existing model	
Structure type:	
Include feature selection preprocessing step	
Parameter learning method:	r small cell counts
OK Fun Cancel	Apply Reset

The second model type to build has a Markov Blanket structure.

- ► Attach a second Bayesian Network node to the Select node.
- On the Model tab, for Model name, select Custom and enter Markov in the text box.

×

► For Structure type, select Markov Blanket and click OK.

Figure 17-5

😡 Marl	κον							
Fields	Model	Expert	Analyze	Annotations				
Model na	ime:		🔘 At	.to 💿 Custom	Markov			
🔽 Use j	partition	ed data						
🗸 Build	model f	or each s	plit					
-To selec	t fields	manually,	choose "U	se custom settir	ngs" on the	Fields tab—		
Partitio	in:							-
Splits:								
								X
Contin	iue train	ing existir	ng model					
Structure	type:		O TAN	l 🔘 Markov Bla	anket			
	e featur	e selectio	n preproce	ssing step				
🗐 Includ			ł					
🗾 Includ Paramete	er learni	ng method	••					
	er learnii	ng method		num likelihood ()) Bayes a	djustment for	r small cell c	ounts

The third model type to build has a Markov Blanket structure and also uses feature selection preprocessing to select the inputs that are significantly related to the target variable.

- Attach a third Bayesian Network node to the Select node.
- On the Model tab, for Model name, select Custom and enter Markov-FS in the text box.
- ► For Structure type, select Markov Blanket.

► Select Include feature selection preprocessing step and click OK.

Figure 17-6

Creating a Markov Blanket model with Feature Selection preprocessing

😡 Markov-FS		
		0
Fields Model Expert An	alyze Annotations	
Model name:	🔘 Auto 💿 Custom 🛛 Ma	arkov-FS
👿 Use partitioned data		
👿 Build model for each split		
To select fields manually, cho	ose "Use custom settings"	on the Fields tab
Partition:		-
Splits:		×
Continue training existing m	del	
Structure type:	🕽 TAN 🧿 Markov Blanket	
V Include feature selection pro	processing step	
Parameter learning method:		
() Maximum likelihood 🔘 Ba	ayes adjustment for small cell counts
OK 🕨 Run Cancel		Apply Reset

Browsing the Model

Run the stream to create the model nuggets, which are added to the stream and to the Models palette in the upper-right corner. To view their details, double-click on any of the model nuggets in the stream.

The model nugget Model tab is split into two panes. The left pane contains a network graph of nodes that displays the relationship between the target and its most important predictors, as well as the relationship between the predictors.

The right pane shows either *Predictor Importance*, which indicates the relative importance of each predictor in estimating the model, or *Conditional Probabilities*, which contains the conditional probability value for each node value and each combination of values in its parent nodes.

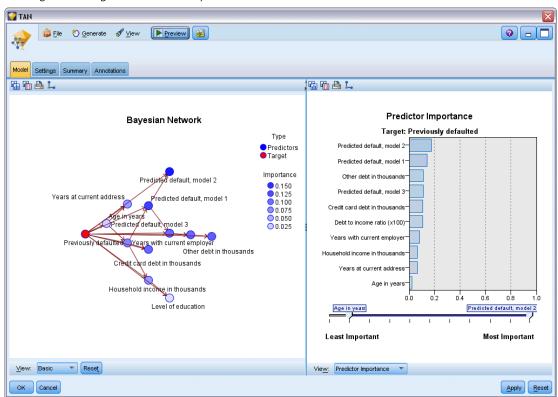
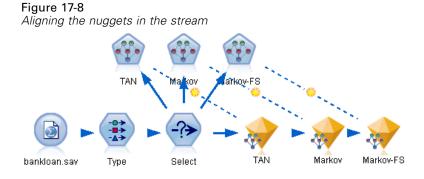


Figure 17-7 Viewing a Tree Augmented Naïve Bayes model

- ► Connect the TAN model nugget to the Markov nugget (choose Replace on the warning dialog).
- Connect the Markov nugget to the Markov-FS nugget (choose Replace on the warning dialog).
- ► Align the three nuggets with the Select node for ease of viewing.



► To rename the model outputs for clarity on the Evaluation graph that you'll be creating, attach a Filter node to the Markov-FS model nugget.

► In the right *Field* column, rename \$B-default as TAN, \$B1-default as Markov, and \$B2-default as Markov-FS.

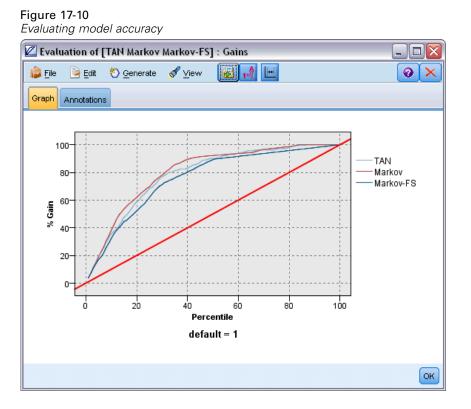
Figure 17-9 Rename model field nam	nes	
🜍 Filter		
Filter Annotations		0
	Fields	: 18 in, 0 filtered, 3 renamed, 18 out
Field	Filter	Field
default	\rightarrow	default 🖌
preddef1	\rightarrow	preddef1
preddef2	\rightarrow	preddef2
preddef3	\rightarrow	preddef3
\$B-default	\rightarrow	TAN
\$BP-default	\rightarrow	\$BP-default
\$B1-default	\rightarrow	Markov
\$BP1-default	\rightarrow	\$BP1-default
\$B2-default	\rightarrow	Markov-FS
\$BP2-default	\rightarrow	\$BP2-default
View current fields View	w unused field settir	ngs
		<u>Apply</u> <u>R</u> eset

To compare the models' predicted accuracy, you can build a gains chart.

• Attach an Evaluation graph node to the Filter node and execute the graph node using its default settings.

Predicting Loan Defaulters (Bayesian Network)

The graph shows that each model type produces similar results; however, the Markov model is slightly better.



To check how well each model predicts, you could use an Analysis node instead of the Evaluation graph. This shows the accuracy in terms of percentage for both correct and incorrect predictions.

• Attach an Analysis node to the Filter node and execute the Analysis node using its default settings.

As with the Evaluation graph, this shows that the Markov model is slightly better at predicting correctly; however, the Markov-FS model is only a few percentage points behind the Markov model. This may mean it would be better to use the Markov-FS model since it uses fewer inputs to calculate its results, thereby saving on data collection and entry time and processing time.

Figure 17-11 *Analyzing model accuracy*

🛾 Analysis of [default]				_ 0 🗙			
😰 File 📄 Edit 🔞	b 14	ł		0 ×			
Analysis Annotations							
😵 Collapse All 🛛 🌳 Exp	and All						
E-Results for output field de	fault						
🖨 Individual Models	🖨 Individual Models						
😑 Comparing TAN v	vith defau	ılt					
Correct	593	84.71	%				
Wrong	107	15.29	%				
Total	700						
🖨 Comparing Marko	v with de	fault					
Correct	604	86.29	%				
Wrong	96	13.71	%				
Total	700						
😑 Comparing Marko	v-FS with	n default					
Correct	573	81.86	%				
Wrong	127	18.14	%				
Total	700						
😑 Agreement between	TAN Mar	kov Marko	v-FS				
Agree	606	86.57%					
Disagree	94	13.43%					
Total	700						
😑 Comparing Agree	ment with	n default					
Correct	541	89.27	%				
Wrong	65	10.73	%				
Total	606						
				ок			
				ON			

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the SPSS Modeler Algorithms Guide, available from the \Documentation directory of the installation disk.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.

Retraining a Model on a Monthly Basis (Bayesian Network)

Bayesian networks enable you to build a probability model by combining observed and recorded evidence with "common-sense" real-world knowledge to establish the likelihood of occurrences by using seemingly unlinked attributes.

This example uses the stream named *bayes_churn_retrain.str*, which references the data files named *telco_Jan.sav* and *telco_Feb.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation and can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *bayes_churn_retrain.str* file is in the *streams* directory.

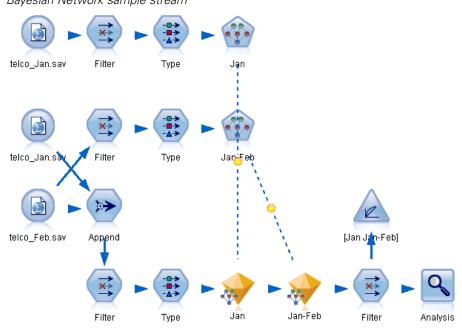
For example, suppose that a telecommunications provider is concerned about the number of customers it is losing to competitors (churn). If historic customer data can be used to predict which customers are more likely to churn in the future, these customers can be targeted with incentives or other offers to discourage them from transferring to another service provider.

This example focuses on using an existing month's churn data to predict which customers may be likely to churn in the future and then adding the following month's data to refine and retrain the model.

Building the Stream

Add a Statistics File source node pointing to *telco_Jan.sav* in the *Demos* folder.

Figure 18-1 Bayesian Network sample stream



© Copyright IBM Corporation 1994, 2012.

Previous analysis has shown you that several data fields are of little importance when predicting churn. These fields can be filtered from your data set to increase the speed of processing when you are building and scoring models.

- ► Add a Filter node to the Source node.
- Exclude all fields except *address*, *age*, *churn*, *custcat*, *ed*, *employ*, *gender*, *marital*, *reside*, *retire*, and *tenure*.
- ► Click OK.

Figure 18-2 Filtering unnecessary fields

Filter		
Filter Annotations		
7.	Fields:	42 in, 31 filtered, 0 renamed, 11 out
Field	Filter	Field
region	→	region 🗲
tenure	\rightarrow	tenure
age	\rightarrow	age 📃
marital	\rightarrow	marital
address	\rightarrow	address
income	_ ★ →	income
ed	\rightarrow	ed
employ	\rightarrow	employ
retire	\rightarrow	retire
gender	\rightarrow	gender 🔽
View current fields View	v unused field settir	igs
OK Cancel		Apply Reset

- ► Add a Type node to the Filter node.
- Open the Type node and click the Read Values button to populate the *Values* column.

► In order that the Evaluation node can assess which value is true and which is false, set the measurement level for the *churn* field to Flag, and set its role to Target. Click OK.

Type								
Preview 0								
A								
Types Format Annotations								
🔧 🤷 🕋 🕨 kead Values 🛛 Clear Values 🖉 Clear All Values								
Field —	Measurement	Values	Missing	Check	Role			
X mantai	пау	170		NULLE	a input			
ݤ address	Continuous	[0,55]		None	> Input			
🜔 ed	nominal	1,2,3,4,5		None	🔪 Input			
X	🖋 Continuous	[0,47]		None	🔪 Input			
Ç employ	X.			None	🔪 Input			
X	🏅 Flag	1.0/0.0						
Ç employ	Flag	1.0/0.0 0,1		None	🔪 Input			
employ				None None	Input			
employ retire gender	💑 Nominal	0,1						
employ retire gender reside	Nominal	0,1 1,2,3,4,5,		None	🔪 Input			

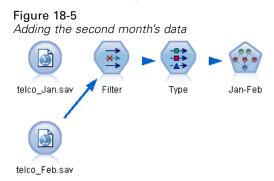
You can build several different types of Bayesian networks; however, for this example you are going to build a Tree Augmented Naïve Bayes (TAN) model. This creates a large network and ensures that you have included all possible links between data variables, thereby building a robust initial model.

- Attach a Bayesian Network node to the Type node.
- On the Model tab, for Model name, select Custom and enter Jan in the text box.
- ► For Parameter learning method, select Bayes adjustment for small cell counts.

Click Run. The model nugget is added to the stream, and also to the Models palette in the upper-right corner.

Figure 18-4 Creating a Tree Augmented Naïve Bayes model								
😡 Jan					X			
					0			
Fields Mo	del Expert	Analyze	Annotations					
Model name:		🔘 Au	.to 💿 Custom	Jan				
👿 Use parti	itioned data							
☑ Build model for each split								
To select fields manually, choose "Use custom settings" on the Fields tab								
Partition:					-			
Splits:					×			
Continue training existing model								
Structure type: 💿 TAN 🔘 Markov Blanket								
Include feature selection preprocessing step								
Parameter learning method:								
O Maximum likelihood 💿 Bayes adjustment for small cell counts								
ОК	▶ Run 🛛	Cancel			Apply Reset			

- ▶ Add a Statistics File source node pointing to *telco* Feb.sav in the Demos folder.
- Attach this new source node to the Filter node (on the warning dialog, choose Replace to replace the connection to the previous source node).



- On the Model tab of the Bayesian Network node, for Model name, select Custom and enter Jan-Feb in the text box.
- ► Select Continue training existing model.

Click Run. The model nugget overwrites the existing one in the stream, but is also added to the Models palette in the upper-right corner.



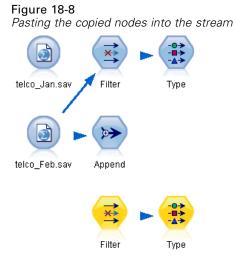
Evaluating the Model

To compare the models, you must combine the two datasets.

Add an Append node and attach both the *telco_Jan.sav* and *telco_Feb.sav* source nodes to it.
 Figure 18-7

Append the two data sources							
😡 Append		$\overline{\mathbf{X}}$					
Annend 2 datacet							
Inputs Append Annotations							
Match fields by: 🛛 🔘 Positio	on 💿 Name 📃 Matc	h case					
Preview of field matches an	d structure						
Output Field -	1[telco_Jan.sav:telco_Jan.s	2[telco_Feb.sav:telco_Feb					
🔆 region	region	🔆 region 🛛 🔺					
🔆 tenure	🔆 tenure	🔆 tenure					
🔆 age	🔆 age	🔷 age 🗌					
🔆 marital	🔆 marital	🔆 marital					
🔆 address	🔆 address	🔆 address					
🛞 income	🛞 income	🛞 income					
今 ed	🔷 ed	🔆 ed					
今 employ	🔆 employ	🔆 employ 📃					
4							
Include fields from: 💿 Main dataset only 🛇 All datasets							
	Tag records by including source dataset in field Input						
OK Cancel		Apply Reset					

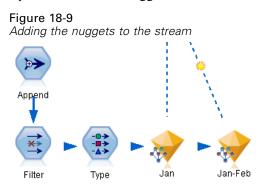
- Copy the Filter and Type nodes from earlier in the stream and paste them onto the stream canvas.
- Attach the Append node to the newly copied Filter node.



The nuggets for the two Bayesian Network models are located in the Models palette in the upper-right corner.

Double-click the Jan model nugget to bring it into the stream, and attach it to the newly copied Type node.

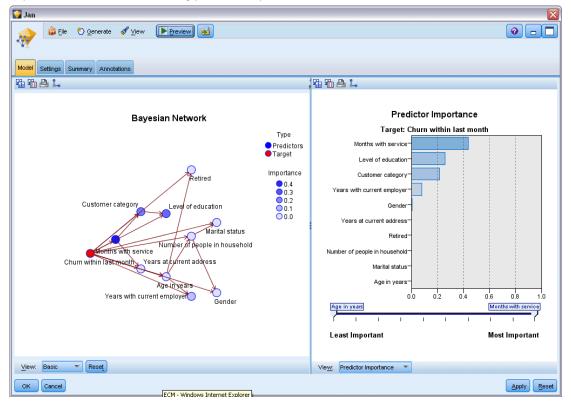
- Attach the Jan-Feb model nugget already in the stream to the Jan model nugget.
- ► Open the Jan model nugget.



The Bayesian Network model nugget Model tab is split into two columns. The left column contains a network graph of nodes that displays the relationship between the target and its most important predictors, as well as the relationship between the predictors.

The right column shows either *Predictor Importance*, which indicates the relative importance of each predictor in estimating the model, or *Conditional Probabilities*, which contains the conditional probability value for each node value and each combination of values in its parent nodes.

Figure 18-10



Bayesian Network model showing predictor importance

To display the conditional probabilities for any node, click on the node in the left column. The right column is updated to show the required details.

The conditional probabilities are shown for each bin that the data values have been divided into relative to the node's parent and sibling nodes.

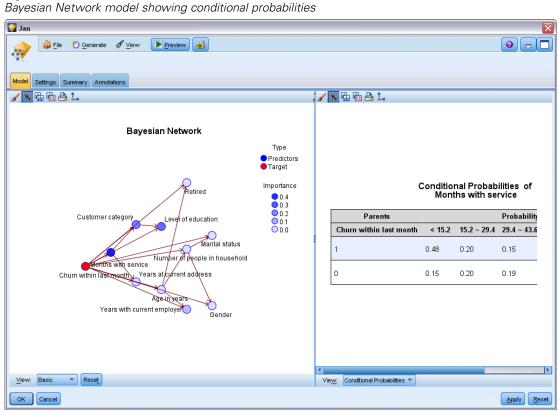


Figure 18-11 Bayesian Network model showing conditional probabilities

▶ To rename the model outputs for clarity, attach a Filter node to the Jan-Feb model nugget.

246

▶ In the right *Field* column, rename \$B-churn as Jan and \$B1-churn as Jan-Feb.

Figure 18-12 Rename model field names							
🔽 Filter							
Preview Preview							
Filter Annotations Tr Fields: 15 in, 0 filtered, 2 renamed, 15 out							
Field -	Filter	Field					
employ	\rightarrow	employ 🛌					
retire	\rightarrow	retire					
gender	\rightarrow	gender					
reside	\rightarrow	reside 🔚					
custcat	\rightarrow	custcat					
churn	\rightarrow	churn					
\$B-churn	\rightarrow	Jan					
\$BP-churn	\rightarrow	\$BP-churn					
\$B1-churn	\rightarrow	Jan-Feb					
\$BP1-churn	\rightarrow	\$BP1-churn					
	w unused field settin	igs					
OK Cancel	OK Cancel Apply Reset						

To check how well each model predicts churn, use an Analysis node; this shows the accuracy in terms of percentage for both correct and incorrect predictions.

- ► Attach an Analysis node to the Filter node.
- ► Open the Analysis node and click Run.

Retraining a Model on a Monthly Basis (Bayesian Network)

This shows that both models have a similar degree of accuracy when predicting churn.

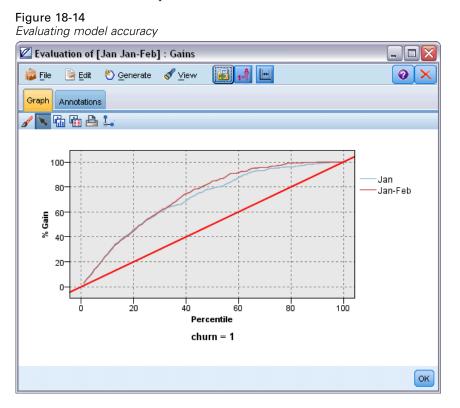
Figure 18-13 Analyzing model accuracy 🔍 Analysis of [churn] _ 0 0 × 👜 File 📄 Edit Analysis Annotations Po Expand All 8 Collapse All E-Results for output field churn 📮 Individual Models Comparing Jan with churn Correct 771 77.1% Wrong 229 22.9% Total 1,000 Comparing Jan-Feb with churn 765 76.5% Correct Wrong 235 23.5% Total 1,000 🖻 Agreement between Jan Jan-Feb Agree 882 88.2% Disagree 118 11.8% Total 1,000 Comparing Agreement with churn Correct 710 80.5% Wrong 172 19.5% Total 882 OK

As an alternative to the Analysis node, you can use an Evaluation graph to compare the models' predicted accuracy by building a gains chart.

• Attach an Evaluation graph node to the Filter node.

and execute the graph node using its default settings.

As with the Analysis node, the graph shows that each model type produces similar results; however, the retrained model using both months' data is slightly better because it has a higher level of confidence in its predictions.



Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the SPSS Modeler Algorithms Guide, available from the \Documentation directory of the installation disk.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.



Retail Sales Promotion (Neural Net/C&RT)

This example deals with data that describes retail product lines and the effects of promotion on sales. (This data is fictitious.) Your goal in this example is to predict the effects of future sales promotions. Similar to the condition monitoring example, the data mining process consists of the exploration, data preparation, training, and test phases.

This example uses the streams named *goodsplot.str* and *goodslearn.str*, which reference the data files named *GOODS1n* and *GOODS2n*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The stream *goodsplot.str* is in the *streams* folder, while the *goodslearn.str* file is in the *streams* directory.

Examining the Data

Each record contains:

- Class. Product type.
- Cost. Unit price.
- *Promotion*. Index of amount spent on a particular promotion.
- *Before*. Revenue before promotion.
- *After*. Revenue after promotion.

The stream *goodsplot.str* contains a simple stream to display the data in a table. The two revenue fields (*Before* and *After*) are expressed in absolute terms; however, it seems likely that the increase in revenue after the promotion (and presumably as a result of it) would be a more useful figure.

Figure 19-1 Effects of promotion on product sales
mp

🖩 Promotions 📃 🗆 🔀							
違 <u>F</u> ile	🔓 File 📄 Edit 🖏 Generate 🐻 🕒 📢 🏦 🛛 🥥						
Table 🛛	nnotations						
	Class	Cost	Promotion	Before	After		
1	Confection	23.990	1467	114957	122762	-	
2	Drink	79.290	1745	123378	137097		
3	Luxury	81.990	1426	135246	141172		
4	Confection	74.180	1098	231389	244456		
5	Confection	90.090	1968	235648	261940		
6	Meat	69.850	1486	148885	156232		
7	Meat	100.1	1248	123760	128441		
8	Luxury	21.010	1364	251072	268134		
9	Luxury	87.320	1585	287043	310857		
10	Drink	26.580	1835	240805	272863		
11	Drink	65.230	1194	212406	227836		
12	Meat	79.820	1596	174022	181489		
13	Confection	41.390	1161	270631	283189		
14	Meat	36.820	1151	231281	235722		
15	Meat	44.050	1482	178138	185934		
16	Drink	84.620	1623	247885	278031		
17	Confection	51.820	1969	148597	165598		
18	Confection	90.080	1462	215102	228696		
19	Luxury	57.300	1842	246885	270082		
20	Drink	11.020	1370	164984	176802	-	
						ок	

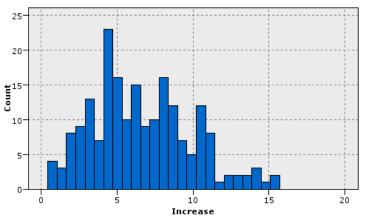
goodsplot.str also contains a node to derive this value, expressed as a percentage of the revenue before the promotion, in a field called *Increase* and displays a table showing this field.

違 File	otions Presidit	🕙 <u>G</u> ene	erate [14	1	
Table 🖌	Annotations						
	Class	Cost	Promotion	Before	After	Increase	
1	Confection	23.990	1467	114957	122762	6.789	
2	Drink	79.290	1745	123378	137097	11.119	
3	Luxury	81.990	1426	135246	141172	4.382	
4	Confection	74.180	1098	231389	244456	5.647	
5	Confection	90.090	1968	235648	261940	11.157	
6	Meat	69.850	1486	148885	156232	4.935	
7	Meat	100.1	1248	123760	128441	3.782	
8	Luxury	21.010	1364	251072	268134	6.796	
9	Luxury	87.320	1585	287043	310857	8.296	
10	Drink	26.580	1835	240805	272863	13.313	
11	Drink	65.230	1194	212406	227836	7.264	
12	Meat	79.820	1596	174022	181489	4.291	
13	Confection	41.390	1161	270631	283189	4.640	
14	Meat	36.820	1151	231281	235722	1.920	
15	Meat	44.050	1482	178138	185934	4.376	
16	Drink	84.620	1623	247885	278031	12.161	
17	Confection	51.820	1969	148597	165598	11.441	
18	Confection	90.080	1462	215102	228696	6.320	
19	Luxury	57.300	1842	246885	270082	9.396	
20	Drink	11.020	1370	164984	176802	7.163	

Figure 19-2 Increase in revenue after promotion

In addition, the stream displays a histogram of the increase and a scatterplot of the increase against the promotion costs expended, overlaid with the category of product involved.

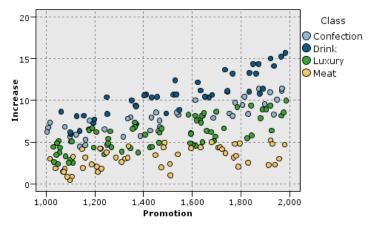




The scatterplot shows that for each class of product, an almost linear relationship exists between the increase in revenue and the cost of promotion. Therefore, it seems likely that a decision tree or neural network could predict, with reasonable accuracy, the increase in revenue from the other available fields.

Figure 19-4

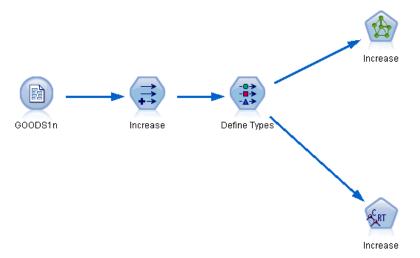
Revenue increase versus promotional expenditure



Learning and Testing

The stream *goodslearn.str* trains a neural network and a decision tree to make this prediction of revenue increase.

Figure 19-5 *Modeling stream goodslearn.str*



Once you have executed the model nodes and generated the actual models, you can test the results of the learning process. You do this by connecting the decision tree and network in series between the Type node and a new Analysis node, changing the input (data) file to *GOODS2n*, and

executing the Analysis node. From the output of this node, in particular from the linear correlation between the predicted increase and the correct answer, you will find that the trained systems predict the increase in revenue with a high degree of success.

Further exploration could focus on the cases where the trained systems make relatively large errors; these could be identified by plotting the predicted increase in revenue against the actual increase. Outliers on this graph could be selected using IBM® SPSS® Modeler's interactive graphics, and from their properties, it might be possible to tune the data description or learning process to improve accuracy.

20 Condition Monitoring (Neural Net/C5.0)

This example concerns monitoring status information from a machine and the problem of recognizing and predicting fault states. The data is created from a fictitious simulation and consists of a number of concatenated series measured over time. Each record is a snapshot report on the machine in terms of the following:

- *Time*. An integer.
- Power. An integer.
- Temperature. An integer.
- Pressure. 0 if normal, 1 for a momentary pressure warning.
- *Uptime*. Time since last serviced.
- *Status*. Normally 0, changes to error code on error (101, 202, or 303).
- *Outcome*. The error code that appears in this time series, or 0 if no error occurs. (These codes are available only with the benefit of hindsight.)

This example uses the streams named *condplot.str* and *condlearn.str*, which reference the data files named *COND1n* and *COND2n*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *condplot.str* and *condlearn.str* files are in the *streams* directory.

For each time series, there is a series of records from a period of normal operation followed by a period leading to the fault, as shown in the following table:

Time	Power	Temperature	Pressure	Uptime	Status	Outcome
0	1059	259	0	404	0	0
1	1059	259	0	404	0	0
51	1059	259	0	404	0	0
52	1059	259	0	404	0	0
53	1007	259	0	404	0	303
54	998	259	0	404	0	303
89	839	259	0	404	0	303
90	834	259	0	404	303	303
0	965	251	0	209	0	0
1	965	251	0	209	0	0
51	965	251	0	209	0	0
52	965	251	0	209	0	0
53	938	251	0	209	0	101
54	936	251	0	209	0	101

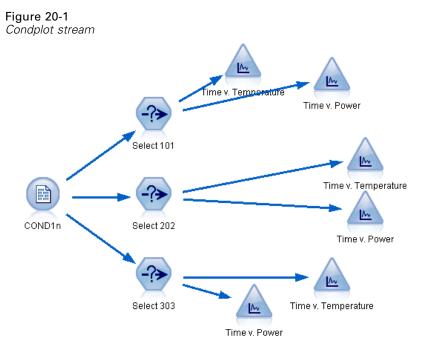
Time	Power	Temperature	Pressure	Uptime	Status	Outcome
208	644	251	 0	209	0	101
209	640	251	0	209	101	101

The following process is common to most data mining projects:

- Examine the data to determine which attributes may be relevant to the prediction or recognition of the states of interest.
- Retain those attributes (if already present), or derive and add them to the data, if necessary.
- Use the resultant data to train rules and neural nets.
- Test the trained systems using independent test data.

Examining the Data

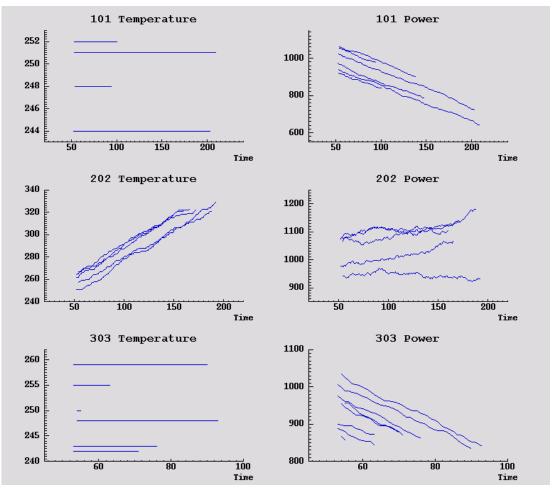
The file *condplot.str* illustrates the first part of the process. It contains a stream that plots a number of graphs. If the time series of temperature or power contains visible patterns, you could differentiate between impending error conditions or possibly predict their occurrence. For both temperature and power, the stream below plots the time series associated with the three different error codes on separate graphs, yielding six graphs. Select nodes separate the data associated with the different error codes.



The results of this stream are shown in this figure.

Figure 20-2

Temperature and power over time

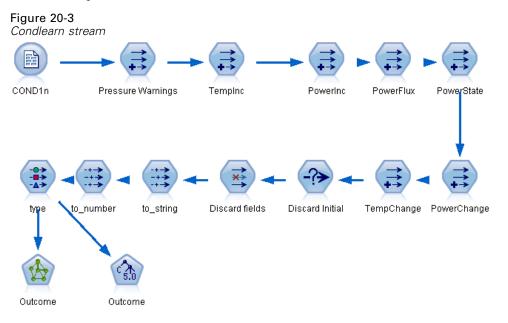


The graphs clearly display patterns distinguishing 202 errors from 101 and 303 errors. The 202 errors show rising temperature and fluctuating power over time; the other errors do not. However, patterns distinguishing 101 from 303 errors are less clear. Both errors show even temperature and a drop in power, but the drop in power seems steeper for 303 errors.

Based on these graphs, it appears that the presence and rate of change for both temperature and power, as well as the presence and degree of fluctuation, are relevant to predicting and distinguishing faults. These attributes should therefore be added to the data before applying the learning systems.

Data Preparation

Based on the results of exploring the data, the stream *condlearn.str* derives the relevant data and learns to predict faults.



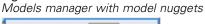
The stream uses a number of Derive nodes to prepare the data for modeling.

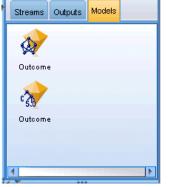
- Variable File node. Reads data file *COND1n*.
- Derive Pressure Warnings. Counts the number of momentary pressure warnings. Reset when time returns to 0.
- **Derive TempInc.** Calculates momentary rate of temperature change using @DIFF1.
- **Derive PowerInc**. Calculates momentary rate of power change using @DIFF1.
- Derive PowerFlux. A flag, true if power varied in opposite directions in the last record and this one; that is, for a power peak or trough.
- Derive PowerState. A state that starts as *Stable* and switches to *Fluctuating* when two successive power fluxes are detected. Switches back to *Stable* only when there hasn't been a power flux for five time intervals or when *Time* is reset.
- **PowerChange**. Average of *PowerInc* over the last five time intervals.
- **TempChange**. Average of *TempInc* over the last five time intervals.
- Discard Initial (select). Discards the first record of each time series to avoid large (incorrect) jumps in *Power* and *Temperature* at boundaries.
- Discard fields. Cuts records down to Uptime, Status, Outcome, Pressure Warnings, PowerState, PowerChange, and TempChange.
- **Type**. Defines the role of *Outcome* as Target (the field to predict). In addition, defines the measurement level of *Outcome* as Nominal, *Pressure Warnings* as Continuous, and *PowerState* as Flag.

Learning

Running the stream in *condlearn.str* trains the C5.0 rule and neural network (net). The network may take some time to train, but training can be interrupted early to save a net that produces reasonable results. Once the learning is complete, the Models tab at the upper right of the managers window flashes to alert you that two new nuggets were created: one represents the neural net and one represents the rule.

Figure 20-4





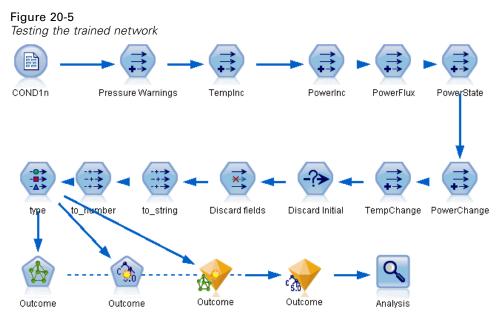
The model nuggets are also added to the existing stream, enabling us to test the system or export the results of the model. In this example, we will test the results of the model.

Testing

The model nuggets are added to the stream, both of them connected to the Type node.

- Reposition the nuggets as shown, so that the Type node connects to the neural net nugget, which connects to the C5.0 nugget.
- ► Attach an Analysis node to the C5.0 nugget.

• Edit the original source node to read the file *COND2n* (instead of *COND1n*), as *COND2n* contains unseen test data.



• Open the Analysis node and click Run.

Doing so yields figures reflecting the accuracy of the trained network and rule.



Classifying Telecommunications Customers (Discriminant Analysis)

Discriminant analysis is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric one.

For example, suppose a telecommunications provider has segmented its customer base by service usage patterns, categorizing the customers into four groups. If demographic data can be used to predict group membership, you can customize offers for individual prospective customers.

This example uses the stream named *telco_custcat_discriminant.str*, which references the data file named *telco.sav*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *telco_custcat_discriminant.str* file is in the *streams* directory.

The example focuses on using demographic data to predict usage patterns. The target field *custcat* has four possible values which correspond to the four customer groups, as follows:

Value	Label
1	Basic Service
2	E-Service
3	Plus Service
4	Total Service

Creating the Stream

First, set the stream properties to show variable and value labels in the output. From the menus, choose:

File > Stream Properties... > Options > General

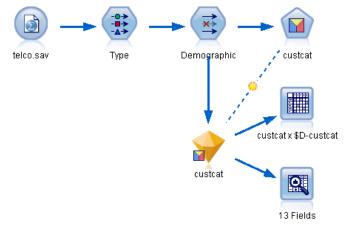
► Make sure that Display field and value labels in output is selected and click OK.

Figure 21-1 Stream propertie	25						
📀 telco_custcat_o	🕅 telco_custcat_discriminant 🛛 🛛 🔀						
	0						
Options Messages	Parameters Deployment Script Globals Search Comments Annotations						
Select a setting:							
General	These are general settings that apply to the current stream. Click Save As Default to use these settings as the default for all your streams.						
Date/Time Number formats	Decimal symbol:						
Optimization	Grouping symbol:						
Logging and Status	Encoding: System default 🔽						
Layout	Ruleset Evaluation: Voting						
	Maximum number of rows to show in Data Preview:						
	Maximum members for nominal fields						
	🗹 Limit set size for Neural, Kohonen and K-Means modeling 🛛 🔁 🗲						
	Refresh source nodes on execution						
	☑ Display field and value labels in output						
	Save As Default						
OK Cancel	Apply Reset						

▶ Add a Statistics File source node pointing to *telco.sav* in the *Demos* folder.

Figure 21-2

Sample stream to classify customers using discriminant analysis



► Add a Type node and click Read Values, making sure that all measurement levels are set correctly. For example, most fields with values 0 and 1 can be regarded as flags.

Figure 21-3

Setting the measurement level for multiple fields

Type Types Format	review Annotations					(0		
<u>-</u>	🍋 🚺 🕨 R	ead Value	es Clear	r Values 🚶	C	lear All Valu	Jes)	
Field -	Measurer	nent	Values	Missing		Check		Role	
🔆 gender	💑 Nominal		0,1		No	ne	>	Input	<u></u>
🚫 reside	🔗 Continuous	3	[1,8]		No	ne		Input	
🔿 tollfree	🎖 Flag		1/0		No	ne	2	Input	
🔿 equip	ă Flag S Flag S Flag		1/0		No	ne	N	Input	
📿 callcard	🎖 Flag		1/0		N¢	<default></default>			
📿 wireless	🍯 Flag 🛛		1/0		Nic				
🛞 longmon	🔗 Continuou	Sel	ect All			Continuo	us		
tollmon	Continuou	Sel	ect None			Categoric	al		-
O View current	fields 🔘 Vi	Sel	ect Fields	•		Flag		6	
		Cop	ру	Ctrl+C		Nominal		n	
OK Cancel]	<u>वि</u> Pas	ste Special	. Ctrl+V		Ordinal			Reset

Tip: To change properties for multiple fields with similar values (such as 0/1), click the *Values* column header to sort fields by value, and then hold down the shift key while using the mouse or arrow keys to select all the fields you want to change. You can then right-click on the selection to change the measurement level or other attributes of the selected fields.

Notice that *gender* is more correctly considered as a field with a set of two values, instead of a flag, so leave its Measurement value as Nominal.

Classifying Telecommunications Customers (Discriminant Analysis)

• Set the role for the *custcat* field to Target. All other fields should have their role set to Input.

gure 21-4 etting field role						
	review				0-1	
Types Format	Annotations	Υ				
4. 00	🍋 🜓 Read Val	ues Clear	r Values	Clear All Va	alues	
Field -	Measurement	Values	Missing	Check	Role]
X eom	🕘 пау	170		NULLE	a inpur	4
👾 loglong	Continuous	[-0.10536		None	🔪 Input	Г
🐲 logtoli	🖉 Continuous	[1.74919		None	🔪 Input	
🛞 logequi	🔗 Continuous	[2.73436		None	🔪 Input	
🛞 logcard	🔗 Continuous	[1.01160		None	🔪 Input	
🛞 logwire	🔗 Continuous	[2.70136		None	🔪 Input	
🛞 Ininc	🔗 Continuous	[2.19722		None	🔪 Input	L
🔆 custcat	🂑 Nominal	1,2,3,4		None	O Target	
🔆 churn	💑 Nominal	0,1		None	S Input	
View current fields View unused field settings						
OK Cancel Apply Reset						

Since this example focuses on demographics, use a Filter node to include only the relevant fields (*region, age, marital, address, income, ed, employ, retire, gender, reside, and custcat*). Other fields can be excluded for the purpose of this analysis.

Figure 21-5

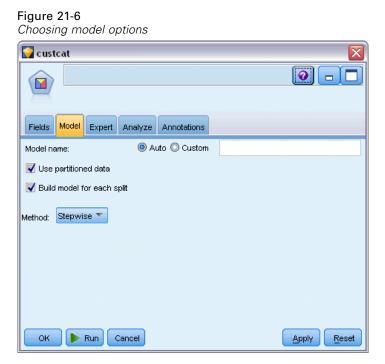
Filtering on demographic fields

Demographic					
	Fields: 4:	2 in, 31 filtered, 0 renamed, 11 out			
Field -	Filter	Field			
region	\rightarrow	region 🔄			
tenure	$\xrightarrow{\times}$	tenure			
age	\rightarrow	age			
marital	\rightarrow	marital			
address	\rightarrow	address			
income	\rightarrow	income			
ed	\rightarrow	ed			
employ	\rightarrow	employ			
retire	\rightarrow	retire			
gender	\rightarrow	gender 🔽			
View current fields View unused field settings OK Cancel Apply Reset					

264

(Alternatively, you could change the role to None for these fields rather than exclude them, or select the fields you want to use in the modeling node.)

▶ In the Discriminant node, click the Model tab and select the Stepwise method.



• On the Expert tab, set the mode to Expert and click Output.

Classifying Telecommunications Customers (Discriminant Analysis)

 Select Summary table, Territorial map, and Summary of Steps in the Advanced Output dialog box, then click OK.

😪 Discriminant: Advanc	ed Output —	X
Statistics		
Descriptives:	Matrices:	
🔲 Means	🗾 Within-gro	ups correlation
🔲 Univariate ANOVAS	📃 Within-gro	up covariance
🔲 Box's M	🗾 Separate-g	groups covariance
Function Coefficients:	📃 Total cova	riance
🧾 Fisher's		
🔲 Unstandardized		
Classification		
Casewise results		Plots:
Limit cases to first:	10 🖨	📝 Territorial map
📝 Summary table		Combined-groups
📃 Leave-one-out classification	n	🗾 Separate-groups
Stepwise		
V Summary of Steps		
📕 F for pairwise distances		

Examining the Model

Click Run to create the model, which is added to the stream and to the Models palette in the upper-right corner. To view its details, double-click on the model nugget in the stream.

The Summary tab shows (among other things) the target and the complete list of inputs (predictor fields) submitted for consideration.



For details of the discriminant analysis results:

- ► Click the Advanced tab.
- Click the "Launch in external browser" button (just below the Model tab) to view the results in your Web browser.

Classifying Telecommunications Customers (Discriminant Analysis)

Stepwise Discriminant Analysis

Figure 21-9

Variables not in the analysis, step 0

Step		Tolerance	Min. Tolerance	F to Enter	Wilks' Lambda
0	Age in years	1.000	1.000	7.521	.978
	Marital status	1.000	1.000	3.500	.990
	Years at current address	1.000	1.000	8.433	.975
	Household income in thousands	1.000	1.000	6.689	.980
	Level of education	1.000	1.000	61.454	.844
	Retired	1.000	1.000	3.005	.991
	Years with current employer	1.000	1.000	16.976	.951
	Gender	1.000	1.000	.373	.999
	Number of people in household	1.000	1.000	3.976	.988

When you have a lot of predictors, the stepwise method can be useful by automatically selecting the "best" variables to use in the model. The stepwise method starts with a model that doesn't include any of the predictors. At each step, the predictor with the largest F to Enter value that exceeds the entry criteria (by default, 3.84) is added to the model.

Figure 21-10

Variables not in the analysis, step 3

			Min.		Wilks'
Step		Tolerance	Tolerance	F to Enter	Lambda
3	Age in years	.535	.535	.252	.795
	Marital status	.605	.593	1.507	.792
	Years at current address	.776	.771	3.514	.787
	Household income in thousands	.688	.657	.687	.794
	Retired	.917	.880	.353	.795
	Gender	.997	.931	.395	.795

The variables left out of the analysis at the last step all have *F* to *Enter* values smaller than 3.84, so no more are added.

Figure 21-11 Variables in the analysis

Oton		Tolerance	F to Remove	Wilks' Lambda
Step		Tolerance	F to Remove	Lambua
1	Level of education	1.000	61.454	
2	Level of education	.953	59.108	.951
	Years with current employer	.953	14.933	.844
3	Level of education	.951	60.046	.940
	Years with current employer	.934	15.824	.834
	Number of people in household	.979	4.841	.807

268

This table displays statistics for the variables that are in the analysis at each step. *Tolerance* is the proportion of a variable's variance not accounted for by other independent variables in the equation. A variable with very low tolerance contributes little information to a model and can cause computational problems.

F to Remove values are useful for describing what happens if a variable is removed from the current model (given that the other variables remain). *F to Remove* for the entering variable is the same as *F to Enter* at the previous step (shown in the Variables Not in the Analysis table).

A Note of Caution Concerning Stepwise Methods

Stepwise methods are convenient, but have their limitations. Be aware that because stepwise methods select models based solely upon statistical merit, it may choose predictors that have no **practical significance**. If you have some experience with the data and have expectations about which predictors are important, you should use that knowledge and eschew stepwise methods. If, however, you have many predictors and no idea where to start, running a stepwise analysis and adjusting the selected model is better than no model at all.

Checking Model Fit

Figure 21-12 Eigenvalues

Function	Eigenvalue	% of Variance	Cumulative %	Canonical Correlation
1	.198	80.2	80.2	.407
2	.048	19.4	99.6	.214
3	.001	.4	100.0	.031

Nearly all of the variance explained by the model is due to the first two discriminant functions. Three functions are fit automatically, but due to its minuscule eigenvalue, you can fairly safely ignore the third.

Figure 21-13 Wilks' lambda

Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1 through 3	.796	227.345	9	.000
2 through 3	.953	47.486	4	.000
3	.999	.929	1	.335

Wilks' lambda agrees that only the first two functions are useful. For each set of functions, this tests the hypothesis that the means of the functions listed are equal across groups. The test of function 3 has a significance value greater than 0.10, so this function contributes little to the model.

Structure Matrix

Figure 21-14 Structure matrix

		Function	
	1	2	3
Level of education	.966*	090	244
Years with current employer	182	.964*	193
Age in years ^a	162	.598*	285
Household income in thousands	.109	.514*	190
Years at current address ^a	151	.394*	214
Retired ^a	108	.230*	137
Gendera	.008	.054*	.009
Number of people in household	.232	.097	.968*
Marital statusª	.132	.134	.600*

Pooled within-groups correlations between discriminating variables and standardized canonical discriminant functions Variables ordered by absolute size of correlation within function.

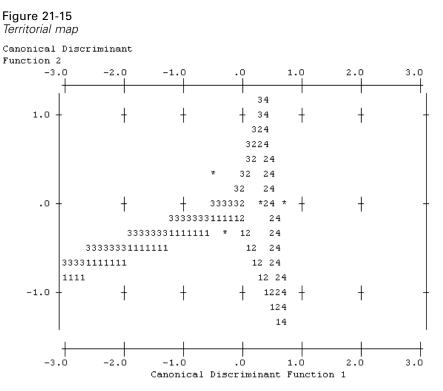
 Largest absolute correlation between each variable and any discriminant function

a. This variable not used in the analysis.

When there is more than one discriminant function, an asterisk(*) marks each variable's largest absolute correlation with one of the canonical functions. Within each function, these marked variables are then ordered by the size of the correlation.

- *Level of education* is most strongly correlated with the first function, and it is the only variable most strongly correlated with this function.
- Years with current employer, Age in years, Household income in thousands, Years at current address, Retired, and Gender are most strongly correlated with the second function, although Gender and Retired are more weakly correlated than the others. The other variables mark this function as a "stability" function.
- Number of people in household and Marital status are most strongly correlated with the third discriminant function, but this is a useless function, so these are nearly useless predictors.

Territorial Map



The territorial map helps you to study the relationships between the groups and the discriminant functions. Combined with the structure matrix results, it gives a graphical interpretation of the relationship between predictors and groups. The first function, shown on the horizontal axis, separates group 4 (*Total service* customers) from the others. Since *Level of education* is strongly positively correlated with the first function, this suggests that your *Total service* customers are, in general, the most highly educated. The second function separates groups 1 and 3 (*Basic service* and *Plus service* customers). *Plus service* customers are not separated well from the others, although the map suggests that they tend to be well educated with a moderate amount of work experience.

In general, the closeness of the group centroids, marked with asterisks (*), to the territorial lines suggests that the separation between all groups is not very strong.

Only the first two discriminant functions are plotted, but since the third function was found to be rather insignificant, the territorial map offers a comprehensive view of the discriminant model.

Classification Results

Figure 21-16

			F	Predicted Gro	up Membership		
		Customer category	Basic service	E-service	Plus service	Total service	Total
Original	Count	Basic service	125	11	61	69	266
		E-service	49	15	58	95	217
		Plus service	102	14	112	53	281
		Total service	40	16	37	143	236
	%	Basic service	47.0	4.1	22.9	25.9	100.0
		E-service	22.6	6.9	26.7	43.8	100.0
		Plus service	36.3	5.0	39.9	18.9	100.0
		Total service	16.9	6.8	15.7	60.6	100.0

a. 39.5% of original grouped cases correctly classified.

From Wilks' lambda, you know that your model is doing better than guessing, but you need to turn to the classification results to determine how much better. Given the observed data, the "null" model (that is, one without predictors) would classify all customers into the modal group, *Plus service*. Thus, the null model would be correct 281/1000 = 28.1% of the time. Your model gets 11.4% more or 39.5% of the customers. In particular, your model excels at identifying *Total service* customers. However, it does an exceptionally poor job of classifying *E-service* customers. You may need to find another predictor in order to separate these customers.

Summary

You have created a discriminant model that classifies customers into one of four predefined "service usage" groups, based on demographic information from each customer. Using the structure matrix and territorial map, you identified which variables are most useful for segmenting your customer base. Lastly, the classification results show that the model does poorly at classifying *E-service* customers. More research is required to determine another predictor variable that better classifies these customers, but depending on what you are looking to predict, the model may be perfectly adequate for your needs. For example, if you are not concerned with identifying *E-service* customers the model may be accurate enough for you. This may be the case where the E-service is a loss-leader which brings in little profit. If, for example, your highest return on investment comes from *Plus service* or *Total service* customers, the model may give you the information you need.

Also note that these results are based on the training data only. To assess how well the model generalizes to other data, you can use a Partition node to hold out a subset of records for purposes of testing and validation.

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the SPSS Modeler Algorithms Guide. This is available from the *\Documentation* directory of the installation disk.



Analyzing Interval-Censored Survival Data (Generalized Linear Models)

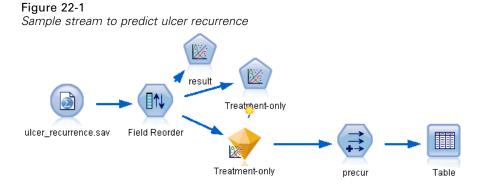
When analyzing survival data with interval censoring—that is, when the exact time of the event of interest is not known but is known only to have occurred within a given interval—then applying the Cox model to the hazards of events in intervals results in a complementary log-log regression model.

Partial information from a study designed to compare the efficacy of two therapies for preventing the recurrence of ulcers is collected in *ulcer_recurrence.sav*. This dataset has been presented and analyzed elsewhere . Using generalized linear models, you can replicate the results for the complementary log-log regression models.

This example uses the stream named *ulcer_genlin.str*, which references the data file *ulcer_recurrence.sav*. The data file is in the *Demos* folder and the stream file is in the *streams* subfolder. For more information, see the topic Demos Folder in Chapter 1 on p. 6.

Creating the Stream

▶ Add a Statistics File source node pointing to *ulcer_recurrence.sav* in the *Demos* folder.



• On the Filter tab of the source node, filter out *id* and *time*.

Figure 22-2 Filter unwanted fields					
📀 ulcer_recurrence.sav					
CLEO_DEMOSAulcer_recu		0-0			
Data Filter Types Annotations					
7		Fields: 6 in, 2 filtered, 0 renamed, 4 out			
Field -	Filter	Field			
id	_ × →	id			
age	\rightarrow	age			
duration	\rightarrow	duration			
treatment	\rightarrow	treatment			
time	— × →	time			
result	\rightarrow	result			
	nused field settings				
ОК Сапсе		Apply Reset			

► On the Types tab of the source node, set the role for the *result* field to Target and set its measurement level to Flag. A result of 1 indicates that the ulcer has recurred. All other fields should have their role set to Input.

• Click Read Values to instantiate the data.

\$CLEO	rrence.sav review 2 Refresh _DEMOS/ulcer_recurrence /pes Annotations	ce.sav			0
4.	🗪 🚺 🕨 Read Valu	les Clear	Values	Clear All Valu	es
Field 😑	Measurement	Values	Missing	Check	Role
决 age	🔗 Continuous	[23,76]		None	🔪 Input
🔉 duration	📲 Ordinal	1,2		None	🔪 Input
💭 treatment	💑 Nominal	0,1		None	🔪 Input
决 result	🎖 Flag	1/0		None	🔘 Target

▶ Add a Field Reorder node and specify *duration*, *treatment*, and *age* as the order of inputs. This determines the order in which fields are entered in the model and will help you try to replicate Collett's results.

Reorde	ring fields so they are entered	d into the model as desi	ired
😡 Field	d Reorder		X
	Preview		0
Reorder	Annotations		
🔘 Custo	om Order	O Automatic Sort	
Туре:	🔺 🔻 Name: 🔺 🔻 Storage: 🚺	×	
Туре	Field	Storage	
	[other fields][other fields]		
	treatment	Integer Integer	于
-	age	o Integer	•
	<u> </u>	-	
			±
Clear L	Inused		
Note: Fi	elds added down stream of this node are no	t reordered.	
ОК	Cancel		Apply Reset

Figure 22-4

- Attach a GenLin node to the source node; on the GenLin node, click the Model tab.
- Select First (Lowest) as the reference category for the target. This indicates that the second category is the event of interest, and its effect on the model is in the interpretation of parameter estimates. A continuous predictor with a positive coefficient indicates increased probability of recurrence

276

with increasing values of the predictor; categories of a nominal predictor with larger coefficients indicate increased probability of recurrence with respect to other categories of the set.

Figure 22-5 Choosing model options	
💟 result	
	0 - 🗆
Fields Model Expert Analyze Annotations	
Model name: O Auto O Custom	
☑ Use partitioned data	
📝 Build model for each split	
Model type: $^{igodoldolde{}}$ Main effects only $^{igodolde{}}$ Main effects and all two-way interactions	
Offset:	
Variable	
Offset field:	_
◎ Fixed value	
Value: 0.0 🖨	
Base category for flag target: First (Lowest) 🔽	
Include intercept in model	
OK Fun Cancel	Apply Reset

- Click the Expert tab and select Expert to activate the expert modeling options.
- Select Binomial as the distribution and Complementary log-log as the link function.
- Select Fixed value as the method for estimating the scale parameter and leave the default value of 1.0.

Select Descending as the category order for factors. This indicates that the first category of each factor will be its reference category; the effect of this selection on the model is in the interpretation of parameter estimates.

Figure 22-6 Choosing expert options	
😡 result	X
Fields Model Expert Analyze Annotations	
Mode: 🔘 Simple 💿 Expert	
Target Field Distribution and Link Function	
The distribution that you choose determines which link	functions are available.
Distribution: Binomial 🔻	Parameters
	Parameter for negative binomial:
	Specify value Value: 1.0 🗧
	© Estimate
	Parameter for Tweedie:
Link function: Complementary log-log	Power: 0.0
Method and iteration settings are not available if Distributi Function = Identity. ┌Parameter Estimation	on = Normal and Link
Method:	Maximum Fisher scoring iterations:
Scale parameter method: Fixed value	▼ Value: 1.0 €
Covariance matrix: Model-based estimator (Ĵ Robust estimator
Iterations Output	
Singularity tolerance: 1E-007 🔽	
Value order for categorical inputs: O Ascending I D	escending 🔘 Use data order
OK FRun Cancel	Apply Reset

Run the stream to create the model nugget, which is added to the stream canvas, and also to the Models palette in the upper right corner. To view the model details, right-click the nugget and choose Edit or Browse.

Tests of Model Effects

Figure 22-7

Tests of model effects for main-effects model

		Type III	
	Wald		
Source	Chi-Square	df	Sig.
(Intercept)	.536	1	.464
duration	.003	1	.958
treatment	.382	1	.537
age	.358	1	.550
Dependent V	ariable: Result		

Model: (Intercept), duration, treatment, age

None of the model effects is statistically significant; however, any observable differences in the treatment effects are of clinical interest, so we will fit a reduced model with just the treatment as a model term.

Fitting the Treatment-Only Model

- On the Fields tab of the GenLin node, click Use custom settings.
- ► Select *result* as the target.

• Select *treatment* as the sole input.

Figure 22-8 Choosing field options							
🔽 Treatment-only	×						
	0						
Fields Model Expert Analyze Annotations							
O Use type node settings O Use custom settings							
Target: 💡 result	-						
Inputs: Streatment	×						
Partition:	-						
Splits:	×						
Use weight field	-						
Target field represents number of events occurring in a set of trials Wariable							
Trials field:	-						
Fixed value Number of trials:							
OK Run Cancel	Apply Reset						

• Run the stream and open the resulting model nugget.

On the model nugget, select the Advanced tab and scroll to the bottom.

Parameter Estimates

Figure 22-9

Parameter estimates for treatment-only model

			95%				
			Confidence	e Interval	Hypot	hesis Tes	rt 🛛
					Wald		
Parameter	B	Std. Error	Lower	Upper	Chi-Square	df	Sig.
(Intercept)	-1.442	.5012	-2.425	460	8.282	1	.004
[treatment=1]	.378	.6288	855	1.610	.361	1	.548
[treatment=0]	0 ^a						
(Scale)	1 ^b						

Dependent Variable: Result Model: (Intercept), treatment

a. Set to zero because this parameter is redundant.

b. Fixed at the displayed value.

The treatment effect (the difference of the linear predictor between the two treatment levels; that is, the coefficient for [treatment=1]) is still not statistically significant, but only suggestive that treatment *A* [treatment=0] may be better than *B* [treatment=1] because the parameter estimate for treatment *B* is larger than that for *A*, and is thus associated with an increased probability of recurrence in the first 12 months. The linear predictor, (intercept + treatment effect) is an estimate of log($-log(1-P(recur_{12,t}))$, where P(recur_{12,t}) is the probability of recurrence at 12 months for treatment t(=*A* or *B*). These predicted probabilities are generated for each observation in the dataset.

Predicted Recurrence and Survival Probabilities

Figure 22-10 Derive node settings options	
🔽 Derive	×
Preview	0
Derive as: Conditional	
Settings Annotations	
Mode: 💿 Single 🔘 Multiple	
Derive field:	
precur	
Derive as: Conditional	
Field type: <u></u> y <default></default>	
Then:	
Else:	
OK Cancel	Apply Reset

- ► For each patient, the model scores the predicted result and the probability of that predicted result. In order to see the predicted recurrence probabilities, copy the generated model to the palette and attach a Derive node.
- ▶ In the Settings tab, type precur as the derive field.
- Choose to derive it as Conditional.
- Click the calculator button to open the Expression Builder for the If condition.

Figure 22-11

Derive node: Expression Builder for If condition

General Functions		*	+ **	13	Fields		
Function	Return			Туре	Field -	Storage	
_integer(ITEM)	Boolean	-	± rem	8	result	Integer	
:_real(ITEM)	Boolean		/ mod	-	duration	Integer	
_number(ITEM)	Boolean			a	treatment	Integer	
_string(ITEM)	Boolean		122	A Company	age	Integer	
_date(ITEM)	Boolean			8	\$G-result	Integer	
_time(ITEM)	Boolean				\$GP-result	Real	
_timestamp(ITEM)	Boolean		and or		\$GP-0	Real	
_datetime(ITEM)	Boolean		not() ><		\$GP-1	Real	
_integer(ITEM)	Integer	-			\$GRP-result	Real	
= integer(ITEM) eturns a value of true if ITI	EM type is an int	eger.	Otherwise, ret	urns a val	lue of false.		

- ► Insert the *\$G*-result field into the expression.
- ► Click OK.

The derive field *precur* will take the value of the Then expression when G-result equals 1 and the value of the Else expression when it is 0.

Analyzing Interval-Censored Survival Data (Generalized Linear Models)

Figure 22-12

Derive node: Expression Builder for Then expression

Ceneral Functions	-		**	15	Fields		•
Function -	Return		div	Туре	Field -	Storage	
s_integer(ITEM)	Boolean		rem	8	result	Integer	
s_real(ITEM)	Boolean		nod	-	duration	Integer	
s_number(ITEM)	Boolean		>=		treatment	Integer	
s_string(ITEM)	Boolean				age	Integer	
s_date(ITEM)	Boolean		1	8	\$G-result	Integer	
s_time(ITEM)	Boolean	القار	<u> </u>	ß	\$GP-result	Real	
s_timestamp(ITEM)	Boolean	and	or	ß	\$GP-0	Real	
s_datetime(ITEM)	Boolean	not()	1	× ۲	\$GP-1	Real	
o_integer(ITEM)	Integer 🗧		~		\$GRP-result	Real	
s_integer(ITEM) leturns a value of true if IT ✔ Check expression befo		r. Otherwis	e, retu	rns a val	ue of false.		

- Click the calculator button to open the Expression Builder for the Then expression.
- ▶ Insert the *\$GP-result* field into the expression.
- ► Click OK.

Figure 22-13

Derive node: Expression Builder for Else expression

	😪 Expression Builder - Derive : Else							
l-'\$GP-result'								
Ceneral Functions		•	+ **	15	Fields		-	
Function -	Return			Туре	Field -	Storage		
is_integer(ITEM)	Boolean	-	* rem	8	result	Integer	-	
is_real(ITEM)	Boolean		/ mod	-1	duration	Integer		
is_number(ITEM)	Boolean		>>=		treatment	Integer		
is_string(ITEM)	Boolean		राख		age	Integer		
is_date(ITEM)	Boolean			8	\$G-result	Integer		
is_time(ITEM)	Boolean			Ø	\$GP-result	Real		
is_timestamp(ITEM)	Boolean		and or		\$GP-0	Real		
is_datetime(ITEM)	Boolean		not() ><		\$GP-1	Real		
to_integer(ITEM)	Integer	Ŧ		1	\$GRP-result	Real		
is_integer(ITEM) Returns a value of true if ITE Check expression befor OK Cancel		eger.	Otherwise, reti	urns a vai	lue of false.	Check H	Help	

• Click the calculator button to open the Expression Builder for the Else expression.

- ► Type 1- in the expression and then insert the *\$GP*-result field into the expression.
- ► Click OK.

Figure 22-14	
Derive node settings options	
🔽 precur	

😪 precur		×
	0	
Derive as: Conditional		
Settings Annotations		
Mode: 💿 Single 🔘 Multiple		
Derive field:		
precur		
Derive as: Conditional T Field type:		
lf.		
'\$G-result'		
Then:		
'\$GP-result'		
Else:		
1-'\$GP-result'		
OK Cancel	Apply	Reset

• Attach a table node to the Derive node and execute it.

Analyzing Interval-Censored Survival Data (Generalized Linear Models)

違 <u>F</u> ile	<u> </u>	lit 🕙 🤆	≧enerate			4		0	
Table ,	Annotati	ons							
	result	duration	treatment	age	\$G-result	\$GP-result	\$GP-0	\$GP-1	
1	1	2	1	48	0	0.708	0.708	0.292	
2	0	1	1	73	0	0.708	0.708	0.292	
3	0	1	1	54	0	0.708	0.708	0.292	
4	0	2	1	58	0	0.708	0.708	0.292	
5	0	1	0	56	0	0.789	0.789	0.211	
6	0	2	0	49	0	0.789	0.789	0.211	
7	0	1	1	71	0	0.708	0.708	0.292	
8	0	1	0	41	0	0.789	0.789	0.211	
9	0	1	1	23	0	0.708	0.708	0.292	
10	1	1	1	37	0	0.708	0.708	0.292	
11	0	1	1	38	0	0.708	0.708	0.292	
12	0	2	1	76	0	0.708	0.708	0.292	
13	0	2	0	38	0	0.789	0.789	0.211	
14	1	1	0	27	0	0.789	0.789	0.211	
15	1	1	1	47	0	0.708	0.708	0.292	
16	0	1	0	54	0	0.789	0.789	0.211	
17	1	1	1	38	0	0.708	0.708	0.292	
18	1	2	1	27	0	0.708	0.708	0.292	
19	0	2	0	58	0	0.789	0.789	0.211	
20	0	1	1	75	0	0.708	0.708	0.292	
	•								

Figure 22-	-15
Predicted	probabilities

There is an estimated 0.211 probability that patients assigned to treatment A will experience a recurrence in the first 12 months; 0.292 for treatment B. Note that $1-P(\text{recur}_{12, t})$ is the survivor probability at 12 months, which may be of more interest to survival analysts.

Modeling the Recurrence Probability by Period

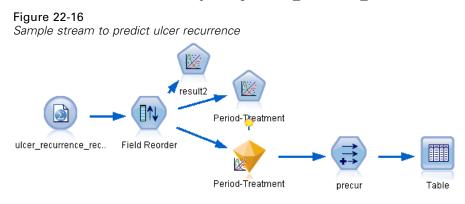
A problem with the model as it stands is that it ignores the information gathered at the first examination; that is, that many patients did not experience a recurrence in the first six months. A "better" model would model a binary response that records whether or not the event occurred during each interval. Fitting this model requires a reconstruction of the original dataset, which can be found in *ulcer_recurrence_recoded.sav*. For more information, see the topic Demos Folder in Chapter 1 on p. 6. This file contains two additional variables:

- *Period*, which records whether the case corresponds to the first examination period or the second.
- *Result by period*, which records whether there was a recurrence for the given patient during the given period.

Each original case (patient) contributes one case per interval in which it remains in the risk set. Thus, for example, patient 1 contributes two cases; one for the first examination period in which no recurrence occurred, and one for the second examination period, in which a recurrence was recorded. Patient 10, on the other hand, contributes a single case because a recurrence was 286

recorded in the first period. Patients 16, 28, and 34 dropped out of the study after six months, and thus contribute only a single case to the new dataset.

► Add a Statistics File source node pointing to *ulcer_recurrence_recoded.sav* in the *Demos* folder.



• On the Filter tab of the source node, filter out *id*, *time*, and *result*.

😡 ulcer_recurrence_recoded	.sav	
🕞 [Preview] 💈 Refre	sh	0
\$CLEO_DEMOSAlcer_recu	rrence_recoded.sav	
Data Fitter Types Annotations		
T		Fields: 8 in, 3 filtered, 0 renamed, 5 out
Field -	Filter	Field
id	$\xrightarrow{*}$	id
age	\rightarrow	age
duration	\rightarrow	duration
treatment	\rightarrow	treatment
time	→	time
result	** **	result
period	\rightarrow	period
result2	\rightarrow	result2
View current fields View ur	nused field settings	

• On the Types tab of the source node, set the role for the *result2* field to Target and set its measurement level to Flag. All other fields should have their role set to Input.

	rence_recoded.sav	e_recoded.s	av	0	
	💌 🜗 Read Value	s Clear	Values	Clear All Values	5
Field -	Measurement	Values	Missing	Check	Role
🔆 age	🔗 Continuous	[23,76]		None	🔪 Input
🔆 duration	📶 Ordinal	1,2		None	🔪 Input
🔆 treatment	💑 Nominal	0,1		None	🔪 Input
즞 period	📲 Ordinal	1,2		None	🔪 Input
📿 result2	🎖 Flag	1/0		None	🔘 Target
View current :	fields 🔘 View unused	field setting:	3		
OK Cancel)			A	pply <u>R</u> eset

 Add a Field Reorder node and specify *period*, *duration*, *treatment*, and *age* as the order of inputs. Making *period* the first input (and not including the intercept term in the model) will allow you to fit a full set of dummy variables to capture the period effects.

Figure 22-19

🜍 Field	Reorder		×
	Preview		0
Reorder	Annotations		
🔘 Custo	m Order	O Automatic Sort	
Type:	🔺 🔻 Name: 🔺 🔻 Storage:	A V	
Туре	Field	Storage	
	·····[other fields] ·····	-	
	period	🔆 Integer	
	duration	🔆 Integer	
	treatment	关 Integer	•
4	age	关 Integer	
			Ŧ
Clear U	nused		
Note: Fie	lds added down stream of this node are not	reordered.	
ок	Cancel		Apply Reset

Reordering fields so they are entered into the model as desired

Analyzing Interval-Censored Survival Data (Generalized Linear Models)

► On the GenLin node, click the Model tab.

Figure 22-20 Choosing model options	
🚰 result2	X
Fields Model Expert Analyze Annotations	
Model name: O Custom	
✓ Use partitioned data	
I Build model for each split	
Model type: Main effects only Main effects and all two-way interactions 	
Offset:	
Variable	
Offset field:	
C Fixed value	
Value: 0.0 🖨	
Base category for flag target: First (Lowest) 🔻	
Include intercept in model	
OK Run Cancel	Apply Reset

- Select First (Lowest) as the reference category for the target. This indicates that the second category is the event of interest, and its effect on the model is in the interpretation of parameter estimates.
- ► Deselect Include intercept in model.

• Click the Expert tab and select Expert to activate the expert modeling options.

Figure 22-21 Choosing expert options	
💟 result	
Fields Model Expert Analyze Annotations	
Mode: 🔘 Simple 💿 Expert	
Target Field Distribution and Link Function	
The distribution that you choose determines which link f	
Distribution: Binomial	Parameters
	Parameter for negative binomial:
	Estimate
	Parameter for Tweedie: 1.5
Link function: Complementary log-log	Power: 0.0 🗬
← Method and iteration settings are not available if Distributio Function = Identity. ┌Parameter Estimation	on = Normal and Link
Method:	Maximum Fisher scoring iterations: 📃 🗧
Scale parameter method: Fixed value	▼ Value: 1.0 🗧
Covariance matrix: OModel-based estimator 🤇) Robust estimator
tterations Output	
Singularity tolerance: 1E-007 T	
Value order for categorical inputs: O Ascending O De	escending 🔘 Use data order

- ▶ Select Binomial as the distribution and Complementary log-log as the link function.
- ▶ Select Fixed value as the method for estimating the scale parameter and leave the default value of 1.0.
- Select Descending as the category order for factors. This indicates that the first category of each factor will be its reference category; the effect of this selection on the model is in the interpretation of parameter estimates.
- Run the stream to create the model nugget, which is added to the stream canvas, and also to the Models palette in the upper right corner. To view the model details, right-click the nugget and choose Edit or Browse.

Tests of Model Effects

Figure 22-22

Tests of model effects for main-effects model

	Туре III	
Wald Chi-Square	df	Sig.
.464	1	.496
.000	1	.988
.117	1	.732
.314	1	.575
	Chi-Square .464 .000 .117	Wald df Chi-Square df .464 1 .000 1 .117 1

Dependent Variable: Result by period Model: period, duration, treatment, age

None of the model effects is statistically significant; however, any observable differences in the period and treatment effects are of clinical interest, so we will fit a reduced model with just those model terms.

Fitting the Reduced Model

- ▶ On the Fields tab of the GenLin node, click Use custom settings.
- ► Select *result2* as the target.

► Select *period* and *treatment* as the inputs.

Figure 22-23 Choosing field options

Period-Treatment	$\overline{\mathbf{X}}$
	0
Fields Model Expert Analyze Annotations	
Use type node settings	
Target: 💡 result2	J
Inputs: Inputs: Inputs:	×
Partition:	_]
Splits:	×
Use weight field	.
☐ Target field represents number of events occurring in a set of trials ⓐ ∨ariable	
Trials field:	-
Fixed value Number of trials:	
OK Pan Cancel	Apply Reset

• Execute the node and browse the generated model, and then copy the generated model to the palette, attach a table node, and execute it.

Analyzing Interval-Censored Survival Data (Generalized Linear Models)

Parameter Estimates

Figure 22-24

Parameter estimates for treatment-only model

			95% ۱	Vald			
			Confidence	e Interval	Hypoth	esis Test	
		Std.			Wald		
Parameter	в	Error	Lower	Upper	Chi-Square	df	Sig.
[period=2]	-1.794	.5792	-2.929	659	9.597	1	.002
[period=1]	-2.206	.5912	-3.365	-1.047	13.926	1	.000
[treatment=1]	.195	.6279	-1.035	1.426	.097	1	.756
[treatment=0]	0 ^a						
(Scale)	1 ^b						

Dependent Variable: Result by period

Model: period, treatment

a. Set to zero because this parameter is redundant.

b. Fixed at the displayed value.

The treatment effect is still not statistically significant but only suggestive that treatment *A* may be better than *B* because the parameter estimate for treatment *B* is associated with an increased probability of recurrence in the first 12 months. The period values are statistically significantly different from 0, but this is because of the fact that an intercept term is not fit. The period effect (the difference between the values of the linear predictor for [*period=1*] and [*period=2*]) is not statistically significant, as can be seen in the tests of model effects. The linear predictor (period effect + treatment effect) is an estimate of $\log(-\log(1-P(\text{recur}_{p, t})))$, where $P(\text{recur}_{p, t})$ is the probability of recurrence at the period p(=1 or 2, representing six months or 12 months) given treatment t(=A or B). These predicted probabilities are generated for each observation in the dataset.

Predicted Recurrence and Survival Probabilities

Figure 22-25 Derive node settings options	
🚺 Derive	X
	0
Derive as: Conditional	
Settings Annotations	
Mode: 💿 Single 🔘 Multiple	
Derive field:	
precur	
Derive as: Conditional T Field type: 🖌 <default></default>	
lf:	
Then:	
Else:	
OK Cancel	Apply Reset

- ► For each patient, the model scores the predicted result and the probability of that predicted result. In order to see the predicted recurrence probabilities, copy the generated model to the palette and attach a Derive node.
- ▶ In the Settings tab, type precur as the derive field.
- Choose to derive it as Conditional.
- Click the calculator button to open the Expression Builder for the If condition.

Analyzing Interval-Censored Survival Data (Generalized Linear Models)

Figure 22-26

Derive node: Expression Builder for If condition

			– div	1 3	Fields	
Function =	Return			Type	Field 🗂	Storage
_integer(ITEM)	Boolean	-		() X	result2	Integer
_real(ITEM)	Boolean		/ mod		period	Integer
_number(ITEM)	Boolean		>>=		duration	Integer
_string(ITEM)	Boolean		राख		treatment	Integer
_date(ITEM)	Boolean				age	Integer
_time(ITEM)	Boolean			l 🖁	\$G-result2	Integer
_timestamp(ITEM)	Boolean		and or		\$GP-result2	Real
_datetime(ITEM)	Boolean		not() ><		\$GP-0	Real
integer(ITEM)	Integer	Ŧ			\$GP-1	Real

- ▶ Insert the *\$G-result2* field into the expression.
- ► Click OK.

The derive field *precur* will take the value of the Then expression when G-result2 equals 1 and the value of the Else expression when it is 0.

Figure 22-27

Derive node:		

General Functions		*		1 1	Fields		
Function	Return			Туре	Field	Storage	
s_integer(ITEM)	Boolean	-	* rem	() ă	result2	Integer	
s_real(ITEM)	Boolean		/ mod		period	Integer	
s_number(ITEM)	Boolean		>>=		duration	Integer	
s_string(ITEM)	Boolean] 🛃	treatment	Integer	
s_date(ITEM)	Boolean				age	Integer	
s_time(ITEM)	Boolean				\$G-result2	Integer	
s_timestamp(ITEM)	Boolean		and or		\$GP-result2	Real	
s_datetime(ITEM)	Boolean		not() ><		\$GP-0	Real	
o_integer(ITEM)	Integer				\$GP-1	Real	

- ► Click the calculator button to open the Expression Builder for the Then expression.
- ▶ Insert the *\$GP*-result2 field into the expression.
- ► Click OK.

Figure 22-28

Derive node: Expression Builder for Else expression

🚰 Expression Builder - precur : Else 🛛 👔								
l-'\$GP-result2'								
Ceneral Functions		-	+ **	15	Fields		-	
Function -	Return			Туре	Field -	Storage		
is_integer(ITEM)	Boolean	-	* rem	8	result2	Integer	-	
is_real(ITEM)	Boolean		/ mod	-	period	Integer		
is_number(ITEM)	Boolean		>>=		duration	Integer		
is_string(ITEM)	Boolean		বেৰ	-	treatment	Integer		
is_date(ITEM)	Boolean				age	Integer		
is_time(ITEM)	Boolean				\$G-result2	Integer		
is_timestamp(ITEM)	Boolean		and or	A	\$GP-result2	Real		
is_datetime(ITEM)	Boolean		not() ><		\$GP-0	Real		
to_integer(ITEM)	Integer	Ŧ			\$GP-1	Real	-	
is_integer (ITEM) Returns a value of true if ITEM type is an integer. Otherwise, returns a value of false. Check expression before saving OK Cancel								

• Click the calculator button to open the Expression Builder for the Else expression.

Analyzing Interval-Censored Survival Data (Generalized Linear Models)

- ▶ Type 1- in the expression and then insert the *\$GP-result2* field into the expression.
- ► Click OK.

Figure 22-29 Derive node settings options	
💟 precur	X
	0
Derive as: Conditional	
Settings Annotations	
Mode: 🔘 Single 🔘 Multiple	
Derive field:	
precur	
Derive as: Conditional T Field type: 🖋 <default> T</default>	
lf:	
'\$G-result2'	
Then:	
'\$GP-result2'	
Else:	
1-'\$GP-result2'	
OK Cancel	Apply Reset

► Attach a table node to the Derive node and execute it.

Figure 22	-30
Predicted	probabilities

👂 <u>F</u> ile	📄 Edit	0	<u>G</u> enerate		Ð	14 1			<u></u>
able /	Annotatio	ns							
	result2	period	duration	treatment	age	\$G-result2	\$GP-result2	\$GP-0	\$GP-1
	0	1	2	1	48	0	0.875	0.875	0.125
	1	2	2	1	48	0	0.817	0.817	0.183
	0	1	1	1	73	0	0.875	0.875	0.125
	0	2	1	1	73	0	0.817	0.817	0.183
	0	1	1	1	54	0	0.875	0.875	0.125
	0	2	1	1	54	0	0.817	0.817	0.183
	0	1	2	1	58	0	0.875	0.875	0.125
	0	2	2	1	58	0	0.817	0.817	0.183
	0	1	1	0	56	0	0.896	0.896	0.104
0	0	2	1	0	56	0	0.847	0.847	0.153
1	0	1	2	0	49	0	0.896	0.896	0.104
2	0	2	2	0	49	0	0.847	0.847	0.153
3	0	1	1	1	71	0	0.875	0.875	0.125
4	0	2	1	1	71	0	0.817	0.817	0.183
5	0	1	1	0	41	0	0.896	0.896	0.104
6	0	2	1	0	41	0	0.847	0.847	0.153
7	0	1	1	1	23	0	0.875	0.875	0.125
8	0	2	1	1	23	0	0.817	0.817	0.183
9	1	1	1	1	37	0	0.875	0.875	0.125
0	0	1	1	1	38	0	0.875	0.875	0.125
	4							in the second se	

The estimated recurrence probabilities can be summarized as follows:

Treatment	6 months	12 months
А	0.104	0.153
В	0.125	0.183

From these, the survival probability through 12 months can be estimated as $1-(P(recur_{1, t}) + P(recur_{2, t}) \times (1-P(recur_{1, t})))$; thus, for each treatment:

A: 1 - (0.104 + 0.153*0.896) = 0.759

B: 1 - (0.125 + 0.183 * 0.875) = 0.715

which again shows nonstatistically significant support for A as the better treatment.

Summary

Using Generalized Linear Models, you have fit a series of complementary log-log regression models for interval-censored survival data. While there is some support for choosing treatment A, achieving a statistically significant result may require a larger study. However, there are some further avenues to explore with the existing data.

Analyzing Interval-Censored Survival Data (Generalized Linear Models)

• It may be worthwhile to refit the model with interaction effects, particularly between *Period* and *Treatment group*.

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the SPSS Modeler Algorithms Guide.



Using Poisson Regression to Analyze Ship Damage Rates (Generalized Linear Models)

A generalized linear model can be used to fit a Poisson regression for the analysis of count data. For example, a dataset presented and analyzed elsewhere concerns damage to cargo ships caused by waves. The incident counts can be modeled as occurring at a Poisson rate given the values of the predictors, and the resulting model can help you determine which ship types are most prone to damage.

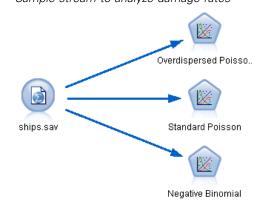
This example uses the stream *ships_genlin.str*, which references the data file *ships.sav*. The data file is in the *Demos* folder and the stream file is in the *streams* subfolder. For more information, see the topic Demos Folder in Chapter 1 on p. 6.

Modeling the raw cell counts can be misleading in this situation because the *Aggregate months of service* varies by ship type. Variables like this that measure the amount of "exposure" to risk are handled within the generalized linear model as offset variables. Moreover, a Poisson regression assumes that the log of the dependent variable is linear in the predictors. Thus, to use generalized linear models to fit a Poisson regression to the accident rates, you need to use *Logarithm of aggregate months of service*.

Fitting an "Overdispersed" Poisson Regression

▶ Add a Statistics File source node pointing to *ships.sav* in the *Demos* folder.

Figure 23-1 Sample stream to analyze damage rates



Using Poisson Regression to Analyze Ship Damage Rates (Generalized Linear Models)

► On the Filter tab of the source node, exclude the field *months_service*. The log-transformed values of this variable are contained in *log months service*, which will be used in the analysis.

Figure 23-2 Filtering an unneeded field	1			
Ships.sav Preview Refree \$CLEO_DEMOS/ships.sav Deta Filter Types Annotations T	esh	Fields: 6 in, 1 filtered, 0 renamed, 5 out		
Field -	Filter	Field		
type	\rightarrow	type		
construction	\rightarrow	construction		
operation	\rightarrow	operation		
months_service	→	months_service		
log_months_service	\rightarrow	log_months_service		
damage_incidents	\rightarrow	damage_incidents		
	nused field settings			
OK Cancel		<u>Apply</u> <u>R</u> eset		

(Alternatively, you could change the role to None for this field on the Types tab rather than exclude it, or select the fields you want to use in the modeling node.)

► On the Types tab of the source node, set the role for the *damage_incidents* field to Target. All other fields should have their role set to Input.

► Click Read Values to instantiate the data.

Figure 23-3 Setting field rol	e						
🚫 ships.sav			_		8		
Scleo_beMOS/ships.sav							
Data Filter Type	S Annotations						
~	🕨 Read Value	es Clear	Values	Clear All Value	s		
Field -	Measurement	Values	Missing	Check	Role		
🔷 type 🛛 🧔	b Nominal	1,2,3,4,5		None	🔪 Input		
📿 📿 construction 🔒	🛛 Ordinal	60,65,70,75		None	🔪 Input		
🛛 쯪 operation 🛛 🚽	🛛 Ordinal	60,75		None	🔪 Input		
🛛 發 log_months 🞸	A	[3.806662		None	O None		
🛛 父 damage_inc 🞸	Continuous	[0,58]		None	🔘 Target		
View current fields View unused field settings							
OK Cancel				<u>A</u>	pply <u>R</u> eset		

• Attach a Genlin node to the source node; on the Genlin node, click the Model tab.

Using Poisson Regression to Analyze Ship Damage Rates (Generalized Linear Models)

► Select *log_months_service* as the offset variable.

Figure 23-4 Choosing model options

💟 Overdispersed Poisson	X
	0 - 0
Fields Model Expert Analyze Annotations	
Model name: O Auto O Custom Overdispersed Poisson	
☑ Use partitioned data	
☑ Build model for each split	
Model type: Main effects only Main effects and all two-way interactions	
Offset:	
Image: State S	
Offset field: 🔗 log_months_service	J
Fixed value Value:	
Base category for flag target: Last (Highest) 🔽	
✓ Include intercept in model	
OK 🕨 Run Cancel	Apply Reset

• Click the Expert tab and select Expert to activate the expert modeling options.

Figure 23-5 Choosing expert options

🔛 Overdispersed Poisson 🛛 🛛 🔀							
Fields Model Expert Analyze Annotations							
lode: OSimple OExpert							
Target Field Distribution and Link Function							
The distribution that you choose determines which link functions are available.							
Distribution: Poisson Parameters							
Parameter for negative binomial:							
Specify value Value: 1.0 🖨							
© Estimate							
Parameter for Tweedie: 1.5 🗢							
Link function: Log Power: 0.0							
ethod and iteration settings are not available if Distribution = Normal and Link unction = Identity. Parameter Estimation							
Method: Hybrid Maximum Fisher scoring iterations:							
Scale parameter method: Pearson Chi-square Value:							
Covariance matrix:							
terations							
ingularity tolerance: 1E-012 T							
alue order for categorical inputs: 🔘 Ascending 🔘 Descending 🔘 Use data order							
OK Run Cancel Apply Reset							

- Select Poisson as the distribution for the response and Log as the link function.
- Select Pearson Chi-Square as the method for estimating the scale parameter. The scale parameter is usually assumed to be 1 in a Poisson regression, but McCullagh and Nelder use the Pearson chi-square estimate to obtain more conservative variance estimates and significance levels.
- Select Descending as the category order for factors. This indicates that the first category of each factor will be its reference category; the effect of this selection on the model is in the interpretation of parameter estimates.
- Click Run to create the model nugget, which is added to the stream canvas, and also to the Models palette in the upper right corner. To view the model details, right-click the nugget and choose Edit or Browse, then click the Advanced tab.

Goodness-of-Fit Statistics

Figure 23-6

Goodness-of-fit statistics

	Value	df	Value/df
Deviance	38.695	25	1.548
Scaled Deviance	22.883	25	
Pearson Chi-Square	42.275	25	1.691
Scaled Pearson Chi-Square	25.000	25	
Log Likelihoodª	-68.281		
Akaike's Information Criterion (AIC)	154.562		
Finite Sample Corrected AIC (AICC)	162.062		
Bayesian Information Criterion (BIC)	168.299		
Consistent AIC (CAIC)	177.299		

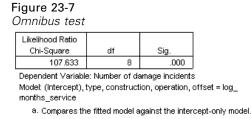
Dependent Variable: Number of damage incidents Model: (Intercept), type, construction, operation, offset = log_months_ service

a. The full log likelihood function is displayed and used in computing information criteria.

b. Information criteria are in small-is-better form.

The goodness-of-fit statistics table provides measures that are useful for comparing competing models. Additionally, the *Value/df* for the Deviance and Pearson Chi-Square statistics gives corresponding estimates for the scale parameter. These values should be near 1.0 for a Poisson regression; the fact that they are greater than 1.0 indicates that fitting the overdispersed model may be reasonable.

Omnibus Test



The omnibus test is a likelihood-ratio chi-square test of the current model versus the null (in this case, intercept) model. The significance value of less than 0.05 indicates that the current model outperforms the null model.

Tests of Model Effects

Figure 23-8

Tests of model effects Type III Wald Source Chi-Square df Sig. (Intercept) 2138.657 1 .000 type 15.415 .004 4 construction 17.242 3 .001 operation 6.249 1 .012

Dependent Variable: Number of damage incidents Model: (Intercept), type, construction, operation, offset = log_months_service

Each term in the model is tested for whether it has any effect. Terms with significance values less than 0.05 have some discernible effect. Each of the main-effects terms contributes to the model.

Parameter Estimates

Figure 23-9 *Parameter estimates*

				Wald dence			
			Interval		Hypothesis Test		
		Std.			Wald		
Parameter	B	Error	Lower	Upper	Chi-Square	df	Sig.
(Intercept)	-6.406	.2828	-6.960	-5.852	513.238	1	.000
[type=5]	.326	.3067	276	.927	1.127	1	.288
[type=4]	076	.3779	817	.665	.040	1	.841
[type=3]	687	.4279	-1.526	.151	2.581	1	.108
[type=2]	543	.2309	996	091	5.536	1	.019
[type=1]	0ª						
[construction=75]	.453	.3032	141	1.048	2.236	1	.135
[construction=70]	.818	.2208	.386	1.251	13.743	1	.000
[construction=65]	.697	.1946	.316	1.079	12.835	1	.000
[construction=60]	0 ^a						
[operation=75]	.384	.1538	.083	.686	6.249	1	.012
[operation=60]	0ª						
(Scale)	1.691 ^b						

Dependent Variable: Number of damage incidents

Model: (Intercept), type, construction, operation, offset = log_months_service

Set to zero because this parameter is redundant.

b. Computed based on the Pearson chi-square.

The parameter estimates table summarizes the effect of each predictor. While interpretation of the coefficients in this model is difficult because of the nature of the link function, the signs of the coefficients for covariates and relative values of the coefficients for factor levels can give important insights into the effects of the predictors in the model.

For covariates, positive (negative) coefficients indicate positive (inverse) relationships between predictors and outcome. An increasing value of a covariate with a positive coefficient corresponds to an increasing rate of damage incidents. For factors, a factor level with a greater coefficient indicates greater incidence of damage. The sign of a coefficient for a factor level is dependent upon that factor level's effect relative to the reference category.

You can make the following interpretations based on the parameter estimates:

- Ship type *B* [*type=2*] has a statistically significantly (*p* value of 0.019) lower damage rate (estimated coefficient of -0.543) than type *A* [*type=1*], the reference category. Type *C* [*type=3*] actually has an estimated parameter lower than *B*, but the variability in *C*'s estimate clouds the effect. See the estimated marginal means for all relations between factor levels.
- Ships constructed between 1965–69 [construction=65] and 1970–74 [construction=70] have statistically significantly (p values <0.001) higher damage rates (estimated coefficients of 0.697 and 0.818, respectively) than those built between 1960–64 [construction=60], the reference category. See the estimated marginal means for all relations between factor levels.
- Ships in operation between 1975–79 [operation=75] have statistically significantly (p value of 0.012) higher damage rates (estimated coefficient of 0.384) than those in operation between 1960–1974 [operation=60].

Fitting Alternative Models

One problem with the "overdispersed" Poisson regression is that there is no formal way to test it versus the "standard" Poisson regression. However, one suggested formal test to determine whether there is overdispersion is to perform a likelihood ratio test between a "standard" Poisson regression and a negative binomial regression with all other settings equal. If there is no overdispersion in the Poisson regression, then the statistic $-2 \times (log-likelihood for Poisson model - log-likelihood for negative binomial model) should have a mixture distribution with half its probability mass at 0 and the rest in a chi-square distribution with 1 degree of freedom.$

Figure 23-10 Expert tab	
🙀 Standard Poisson	
Fields Model Expert Analyze Annotations	
Mode: 🔘 Simple 🖲 Expert	
Target Field Distribution and Link Function	
The distribution that you choose determines which link f	unctions are available. Parameters
Distribution: Poisson 👻	Parameter for negative binomial:
	Specify value Value: 1.0
	Parameter for Tweedie:
Link function:	Power: 0.0
Method and iteration settings are not available if Distributi Function = Identity. Parameter Estimation Method:	on = Normal and Link Maximum Fisher scoring iterations:
Scale parameter method: Fixed value	▼ Value: 1.0 🗧
Covariance matrix:	Ĵ Robust estimator
Iterations Output Singularity tolerance: 1E-007	
	escending 🔘 Use data order
OK 🕨 Run Cancel	Apply Reset

To fit the "standard" Poisson regression, copy and paste the Genlin node, attach it to the source node, open the new node and click the Expert tab.

• Select Fixed value as the method for estimating the scale parameter. By default, this value is 1.

Using Poisson Regression to Analyze Ship Damage Rates (Generalized Linear Models)

Figure 23-11 Expert tab					
😡 Negative Binomial					
Fields Model Expert Analyze Annotations					
Mode: O Simple O Expert					
Target Field Distribution and Link Function					
The distribution that you choose determines which link f	unctions are available.				
Distribution: Negative binomial	Parameters				
	Parameter for negative binomial:				
	◎ Specify value Value: 1.0 🗧				
	◯ Estimate				
	Parameter for Tweedie: 1.5 🗬				
Link function: Log 🔻	Power: 0.0				
Method and iteration settings are not available if Distribution Function = Identity.	on = Normal and Link				
Parameter Estimation					
Method:	Maximum Fisher scoring iterations: 📃 1 🗧				
Scale parameter method: Fixed value	▼ Value: 1.0 🖨				
Covariance matrix: Model-based estimator Robust estimator					
Iterations Output					
Singularity tolerance: 1E-007 💌					
Value order for categorical inputs: 🔘 Ascending 🔘 Descending 🔘 Use data order					
OK 🕨 Run Cancel	Apply Reset				

- To fit the negative binomial regression, copy and paste the Genlin node, attach it to the source node, open the new node and click the Expert tab.
- Select Negative binomial as the distribution. Leave the default value of 1 for the ancillary parameter.
- ▶ Run the stream and browse the Advanced tab on the newly-created model nuggets.

Goodness-of-Fit Statistics

Figure 23-12

Goodness-of-fit statistics for standard Poisson regression

	Value	df	Value/df
Deviance	38.695	25	1.548
Scaled Deviance	38.695	25	
Pearson Chi-Square	42.275	25	1.691
Scaled Pearson Chi-Square	42.275	25	
Log Likelihood ^a	-68.281		
Akaike's Information Criterion (AIC)	154.562		
Finite Sample Corrected AIC (AICC)	162.062		
Bayesian Information Criterion (BIC)	168.299		
Consistent AIC (CAIC)	177.299		

Dependent Variable: Number of damage incidents

Model: (Intercept), type, construction, operation, offset = log_months_ service

 The full log likelihood function is displayed and used in computing information criteria.

b. Information criteria are in small-is-better form.

The log-likelihood reported for the standard Poisson regression is -68.281. Compare this to the negative binomial model.

Figure 23-13

Goodness-of-fit statistics for negative binomial regression

	Value	df	Value/df
Deviance	11.145	25	.446
Scaled Deviance	11.145	25	
Pearson Chi-Square	8.815	25	.353
Scaled Pearson Chi-Square	8.815	25	
Log Likelihood ^a	-83.725		
Akaike's Information Criterion (AIC)	185.450		
Finite Sample Corrected AIC (AICC)	192.950		
Bayesian Information Criterion (BIC)	199.187		
Consistent AIC (CAIC)	208.187		

Dependent Variable: Number of damage incidents Model: (Intercept), type, construction, operation, offset = log_months_

service

a. The full log likelihood function is displayed and used in

computing information criteria.

b. Information criteria are in small-is-better form.

The log-likelihood reported for the negative binomial regression is -83.725. This is actually *smaller* than the log-likelihood for the Poisson regression, which indicates (without the need for a likelihood ratio test) that this negative binomial regression does not offer an improvement over the Poisson regression.

However, the chosen value of 1 for the ancillary parameter of the negative binomial distribution may not be optimal for this dataset. Another way you could test for overdispersion is to fit a negative binomial model with ancillary parameter equal to 0 and request the Lagrange multiplier test on the Output dialog of the Expert tab. If the test is not significant, overdispersion should not be a problem for this dataset.

Using Poisson Regression to Analyze Ship Damage Rates (Generalized Linear Models)

Summary

Using Generalized Linear Models, you have fit three different models for count data. The negative binomial regression was shown not to offer any improvement over the Poisson regression. The overdispersed Poisson regression seems to offer a reasonable alternative to the standard Poisson model, but there is not a formal test for choosing between them.

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the SPSS Modeler Algorithms Guide.



Fitting a Gamma Regression to Car Insurance Claims (Generalized Linear Models)

A generalized linear model can be used to fit a Gamma regression for the analysis of positive range data. For example, a dataset presented and analyzed elsewhere concerns damage claims for cars. The average claim amount can be modeled as having a gamma distribution, using an inverse link function to relate the mean of the dependent variable to a linear combination of the predictors. In order to account for the varying number of claims used to compute the average claim amounts, you specify *Number of claims* as the scaling weight.

This example uses the stream named *car-insurance_genlin.str*, which references the data file named *car_insurance_claims.sav*. The data file is in the *Demos* folder and the stream file is in the *streams* subfolder. For more information, see the topic Demos Folder in Chapter 1 on p. 6.

Creating the Stream

▶ Add a Statistics File source node pointing to *car_insurance_claims.sav* in the *Demos* folder.

Figure 24-1 Sample stream to predict car insurance claims



• On the Types tab of the source node, set the role for the *claimamt* field to Target. All other fields should have their role set to Input.

► Click Read Values to instantiate the data.

SCLEO_	ce_claims.sav eview) 2 Refresh DEMOS/car_insurance_ pes Annotations	claims.sav		6	
٠	💌 🚺 🕨 Read Value	es Clear	Values 🚺	Clear All Value	s
Field 💳	Measurement	Values	Missing	Check	Role
🔆 holderage	📶 Ordinal	1,2,3,4,5,		None	🔪 Input
🔆 vehiclegroup	💑 Nominal	1,2,3,4		None	🔪 Input
🔆 vehicleage	📶 Ordinal	1,2,3,4		None	🔪 Input
즞 claimamt	🔗 Continuous	[11,850]		None	🔘 Target
📿 nclaims	🔗 Continuous	[0,434]		None	🛇 None

• Attach a Genlin node to the source node; in the Genlin node, click the Fields tab.

• Select *nclaims* as the scale weight field.

Figure 24-3 Choosing field options

💟 claimamt	
	0-0
Fields Model Expert Analyze Annotations	
Use type node settings Use custom settings	
Target	-1
Inputs:	-
	\mathbf{x}
Partition:	-
Splits:	×
🗹 Use weight field 🛷 nclaims	_
■ Target field represents number of events occurring in a set of trials	
Trials field:	-
© Fixed value	(-=)
Number of trials: 10	
OK Run Cancel	Apply Reset

Fitting a Gamma Regression to Car Insurance Claims (Generalized Linear Models)

• Click the Expert tab and select Expert to activate the expert modeling options.

Figure 24-4 Choosing e.		otions					
😡 claimamt							X
Fields Mode	Expert	Analyze	Annotations			0	
Mode: O Simp	ole 💿 Expe	rt					
Target Field Di	stribution a	nd Link Fun	ction				
The distribution	h that you c	hoose detei	rmines which li	nk fun	ctions are available.		
Distribution:	Gamma		-	P	arameters		
					Parameter for negative binomial:		
					Specify value	Value:	1.0 🌲
					lestimate		
					Parameter for Tweedie:		1.5 🌲
Link function:	Power					Power:	-1.0 븆
└ Method and itera Function = Identi ┌Parameter Esti	ity.	s are not av	/ailable if Distrit	oution	= Normal and Link		
Method:		Hybrid	Ŧ		Maximum Fisher scoring	iterations:	1 🚔
Scale paramet	er method:	Pearson C	hi-square	-	Value:		1.0 🌲
Covariance ma	atri×:	🔘 Model-	based estimato	r 🔘 F	Robust estimator		
Iterations Singularity tolera Value order for		16	tput -007 🔽 Ascending @) Desc	ending 🔘 Use data order		
ок	Run	Cancel					<u>R</u> eset

- ► Select Gamma as the response distribution.
- Select Power as the link function and type -1.0 as the exponent of the power function. This is an inverse link.
- Select Pearson chi-square as the method for estimating the scale parameter. This is the method used by McCullagh and Nelder, so we follow it here in order to replicate their results.
- Select Descending as the category order for factors. This indicates that the first category of each factor will be its reference category; the effect of this selection on the model is in the interpretation of parameter estimates.
- Click Run to create the model nugget, which is added to the stream canvas, and also to the Models palette in the upper-right corner. To view the model details, right-click the model nugget and choose Edit or Browse, then select the Advanced tab.

316

Parameter Estimates

Figure	24-5
--------	------

Parameter estimates

			95%	95% Wald			
			Confidence	e Interval	Hypoth	esis Tes	t
		Std.			Wald		
Parameter	В	Error	Lower	Upper	Chi-Square	df	Sig.
(Intercept)	.003411	.000418	.002591	.004230	66.593	1	.000
[holderage=8]	.000920	.000416	.000105	.001735	4.898	1	.027
[holderage=7]	.000916	.000408	.000117	.001716	5.046	1	.025
[holderage=6]	.000969	.000405	.000176	.001763	5.740	1	.017
[holderage=5]	.001370	.000419	.000548	.002192	10.682	1	.001
[holderage=4]	.000462	.000411	000342	.001267	1.268	1	.260
[holderage=3]	.000350	.000412	000458	.001158	.720	1	.396
[holderage=2]	.000101	.000436	000754	.000956	.054	1	.816
[holderage=1]	.000000 ^a						
[vehiclegroup=4]	001421	.000181	001775	001067	61.883	1	.000
[vehiclegroup=3]	000614	.000170	000947	000281	13.039	1	.000
[vehiclegroup=2]	.000038	.000169	000293	.000368	.050	1	.823
[vehiclegroup=1]	.000000 ^a						
[vehicleage=4]	.004154	.000442	.003287	.005021	88.175	1	.000
[vehicleage=3]	.001651	.000227	.001207	.002096	53.013	1	.000
[vehicleage=2]	.000366	.000101	.000169	.000564	13.191	1	.000
[vehicleage=1]	.000000 ^a						
(Scale)	1.209 ^b			.001	.0004	.000	.002

Dependent Variable: Average cost of claims

Model: (Intercept), holderage, vehiclegroup, vehicleage a. Set to zero because this parameter is redundant.

b. Computed based on the Pearson chi-square.

The omnibus test and tests of model effects (not shown) indicate that the model outperforms the null model and that each of the main effects terms contribute to the model. The parameter estimates table shows the same values obtained by McCullagh and Nelder for the factor levels and the scale parameter.

Summary

Using Generalized Linear Models, you have fit a gamma regression to the claims data. Note that while the canonical link function for the gamma distribution was used in this model, a log link will also give reasonable results. In general, it is difficult to impossible to directly compare models with different link functions; however, the log link is a special case of the power link where the exponent is 0, thus you can compare the deviances of a model with a log link and a model with a power link to determine which gives the better fit (see, for example, section 11.3 of McCullagh and Nelder).

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the SPSS Modeler Algorithms Guide.



Classifying Cell Samples (SVM)

Support Vector Machine (SVM) is a classification and regression technique that is particularly suitable for wide datasets. A wide dataset is one with a large number of predictors, such as might be encountered in the field of bioinformatics (the application of information technology to biochemical and biological data).

A medical researcher has obtained a dataset containing characteristics of a number of human cell samples extracted from patients who were believed to be at risk of developing cancer. Analysis of the original data showed that many of the characteristics differed significantly between benign and malignant samples. The researcher wants to develop an SVM model that can use the values of these cell characteristics in samples from other patients to give an early indication of whether their samples might be benign or malignant.

This example uses the stream named *svm_cancer.str*, available in the *Demos* folder under the *streams* subfolder. The data file is *cell_samples.data*. For more information, see the topic Demos Folder in Chapter 1 on p. 6.

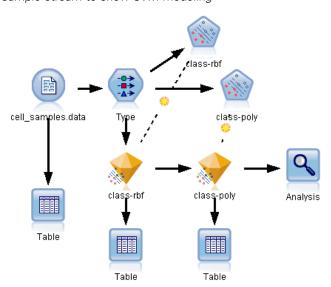
The example is based on a dataset that is publicly available from the UCI Machine Learning Repository (Asuncion and Newman, 2007). The dataset consists of several hundred human cell sample records, each of which contains the values of a set of cell characteristics. The fields in each record are:

Field name	Description
ID	Patient identifier
Clump	Clump thickness
UnifSize	Uniformity of cell size
UnifShape	Uniformity of cell shape
MargAdh	Marginal adhesion
SingEpiSize	Single epithelial cell size
BareNuc	Bare nuclei
BlandChrom	Bland chromatin
NormNucl	Normal nucleoli
Mit	Mitoses
Class	Benign or malignant

For the purposes of this example, we're using a dataset that has a relatively small number of predictors in each record.

Creating the Stream

Figure 25-1 Sample stream to show SVM modeling



Create a new stream and add a Var File source node pointing to *cell_samples.data* in the *Demos* folder of your IBM® SPSS® Modeler installation.

Let's take a look at the data in the source file.

- ► Add a Table node to the stream.
- Attach the Table node to the Var File node and run the stream.

Classifying Cell Samples (SVM)

Figure 25-2
Source data for SVM

違 <u>F</u> ile	📄 Ed	it 🕙 <u>G</u> er	nerate		1				0	>
Table ,	Annotatio	ons								
	hifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class	
1		1	1	2	1	3	1	1	2	-
2		4	5	7	10	3	2	1	2	ľ
3		1	1	2	2	3	1	1	2	l
4		8	1	3	4	3	7	1	2	l
5		1	3	2	1	3	1	1	2	l
6	1	10	8	7	10	9	7	1	4	l
7		1	1	2	10	3	1	1	2	l
8		2	1	2	1	3	1	1	2	
9		1	1	2	1	1	1	5	2	
10		1	1	2	1	2	1	1	2	
11		1	1	1	1	3	1	1	2	
12		1	1	2	1	2	1	1	2	
13		3	3	2	3	4	4	1	4	
14		1	1	2	3	3	1	1	2	I
15		5	10	7	9	5	5	4	4	
16		6	4	6	1	4	3	1	4	
17		1	1	2	1	2	1	1	2	I
18		1	1	2	1	3	1	1	2	
19		7	6	4	10	4	1	2	4	
20		1	1	2	1	3	1	1	2	ŀ
	4									ſ

The *ID* field contains the patient identifiers. The characteristics of the cell samples from each patient are contained in fields *Clump* to *Mit*. The values are graded from 1 to 10, with 1 being the closest to benign.

The *Class* field contains the diagnosis, as confirmed by separate medical procedures, as to whether the samples are benign (value = 2) or malignant (value = 4).

	review				0 - 1
😫 🛄					
Types Format	Annotations				
4. 00 6				01	
	🎮 💫 🕨 🕨 🛤	ues Clear	'Values	Clear All Va	alues
Field -	Measurement	Values	Missing	Check	Role
V OFITISIZE	🖉 conunuous	[1,10]	Missing	NUTE	🛥 mpar
UnifShape			Missing		niput
V OFITISIZE	🖉 conunuous	[1,10]	Missing	NUTE	Input Input
UnifShape	Continuous	[1,10] [1,10]	Missing	None None	Input Input
VonitSize UnifShape MargAdh	Continuous	[1,10] [1,10] [1,10] [1,10]	Missing	None None None	input Input Input Input
UnifShape UnifShape MargAdh SingEpiSize	Continuous Continuous Continuous	[1,10] [1,10] [1,10] [1,10] [1,10] "1","10","	Missing	None None None None	Input Input Input Input Input
UnirSize UnifShape MargAdh SingEpiSize	Continuous Continuous Continuous Continuous Continuous Nominal Continuous Continuous	[1,10] [1,10] [1,10] [1,10] "1","10"," [1,10]	Missing	None None None None None	Input Input Input Input Input Input
UniniSize UnifShape MargAdh SingEpiSize BareNuc BlandChrom	Continuous	[1,10] [1,10] [1,10] [1,10] [1,10] "1","10","	Missing	None None None None None None	Input Input Input Input Input

- ► Add a Type node and attach it to the Var File node.
- ► Open the Type node.

We want the model to predict the value of *Class* (that is, benign (=2) or malignant (=4)). As this field can have one of only two possible values, we need to change its measurement level to reflect this.

- ► In the Measurement column for the *Class* field (the last one in the list), click the value Continuous and change it to Flag.
- Click Read Values.
- ► In the Role column, set the role for *ID* (the patient identifier) to None, as this will not be used either as a predictor or a target for the model.
- Set the role for the target, *Class*, to Target and leave the role of all the other fields (the predictors) as Input.
- ► Click OK.

The SVM node offers a choice of kernel functions for performing its processing. As there's no easy way of knowing which function performs best with any given dataset, we'll choose different functions in turn and compare the results. Let's start with the default, RBF (Radial Basis Function).

Classifying Cell Samples (SVM)

Figure 25-4 Model tab settings

😡 class-r	bf				
Fields Mo	odel Expert	Analyze	Annotations		
Model name	:	🔘 At	uto 🔘 Custom	class-rbf	
👿 Use part	titioned data				
👿 Build mo	del for each :	split			
To select fi	elds manually	, choose "U	se custom settir	ngs" on the Fields t	ab
Partition:					-
Splits:					×
ОК	🕨 Run	Cancel			Apply Reset

- ► From the Modeling palette, attach an SVM node to the Type node.
- Open the SVM node. On the Model tab, click the Custom option for Model name and type *class-rbf* in the adjacent text field.

Figure 25-5	
0	
Default Expert tab settings	;

Class-rbf			
Fields Model Expert Ana	alyze Annotations		
Mode:	🔘 Simple 🔘 Expe	rt	
Append all probabilities (valid	d only for categorica	l targets)	
Stopping criteria:	1.0E-3 💌		
Regularization parameter (C):	10 ≑		
Regression precision (epsilon):	0.1 ≑		
Kernel type:	RBF		
RBF gamma:	0.1 ≑	Bias:	0 ≑
Gamma:	1	Degree:	3 🖨
OK 🕨 Run Cance	21		Apply Reset

• On the Expert tab, set the Mode to Expert for readability but leave all the default options as they are. Note that Kernel type is set to RBF by default. All the options are greyed out in Simple mode.

Figure 25-6 Analyze tab settings
🔽 Class 🛛 🔀
Fields Model Expert Analyze Annotations
Model Evaluation Calculate variable importance Propensity Scores (valid only for flag targets) Calculate raw propensity scores Calculate adjusted propensity scores
Based on Testing partition Validation partition
OK Run Cancel Apply Reset

• On the Analyze tab, select the Calculate variable importance check box.

Classifying Cell Samples (SVM)

- ▶ Click Run. The model nugget is placed in the stream, and in the Models palette at the top right of the screen.
- Double-click the model nugget in the stream. ►

Examining the Data

Figure 25-7 Predictor Importance graph 🙀 class-rbf X 0 🏷 <u>G</u>enerate Preview 違 File 🝼 <u>V</u>iew Model Settings Summary Annotations 6 6 4 1 Predictor Importance Target: Class BareNuc UnifShape Clump BlandChrom NormNucl MargAdh Mit UnifSize SingEpiSize 0.2 0.4 0.6 0.8 0.0 1.0 SingEpiSize BareNuc Least Important Most Important View: Predictor Importance 🔻 OK Cancel Apply Reset

On the Model tab, the Predictor Importance graph shows the relative effect of the various fields on the prediction. This shows us that BareNuc has easily the greatest effect, while UnifShape and *Clump* are also quite significant.

Click OK. ►

- Attach a Table node to the *class-rbf* model nugget.
- ▶ Open the Table node and click Run.

Figure 25-8

Fields added for prediction and confidence value

違 <u>F</u> ile	📄 <u>E</u> dit	🏷 Ger	nerate [2		0	
Table 🛛	Annotation	s							
	gEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class	\$S-Class	\$SP-Class	
		1	3	1	1	2	2	0.992	
2		10	3	2	1	2	4	0.899	
3		2	3	1	1	2	2	0.994	
ł.		4	3	7	1	2	4	0.915	
5		1	3	1	1	2	2	0.992	
6		10	9	7	1	4	4	0.999	
, <u> </u>		10	3	1	1	2	2	0.907	
3		1	3	1	1	2	2	0.997	
)		1	1	1	5	2	2	0.997	
0		1	2	1	1	2	2	0.996	
1		1	3	1	1	2	2	0.999	
2		1	2	1	1	2	2	0.999	
3		3	4	4	1	4	2	0.514	
4		3	3	1	1	2	2	0.989	
5		9	5	5	4	4	4	0.991	
6		1	4	3	1	4	4	0.691	
7		1	2	1	1	2	2	0.997	
8		1	3	1	1	2	2	0.995	
9		10	4	1	2	4	4	0.996	
20		1	3	1	1	2	2	0.986	
	4								۲

▶ The model has created two extra fields. Scroll the table output to the right to see them:

New field name	Description
\$S-Class	Value for <i>Class</i> predicted by the model.
\$SP-Class	Propensity score for this prediction (the likelihood of this prediction being true, a value from 0.0 to 1.0).

Just by looking at the table, we can see that the propensity scores (in the *\$SP-Class* column) for most of the records are reasonably high.

However, there are some significant exceptions; for example, the record for patient 1041801 at line 13, where the value of 0.514 is unacceptably low. Also, comparing *Class* with *\$S-Class*, it's clear that this model has made a number of incorrect predictions, even where the propensity score was relatively high (for example, lines 2 and 4).

Let's see if we can do better by choosing a different function type.

Classifying Cell Samples (SVM)

Trying a Different Function

Figure 25-9 Setting a new name for the model

😡 class-p	oly	
		0
Fields Mo	Ddel Expert Analyze Annotations	
Model name	: 🔘 Auto 💿 Custom 🛛 class-poly	
👿 Use part	titioned data	
👿 Build mo	odel for each split	
To select fi	elds manually, choose "Use custom settings" on the Fields tab-	
Partition:		-
Splits:		×
ок	Run Cancel	Apply Reset

- ► Close the Table output window.
- Attach a second SVM modeling node to the Type node. ►
- ► Open the new SVM node.
- On the Model tab, choose Custom and type *class-poly* as the model name.

Figure 25-10 Expert tab settings for Polynomial

💟 class-poly			
			0
Fields Model Expert Ar	alyze Annotations		
Mode:	🔘 Simple 🔘 Exp	ert	
E Append all probabilities (va	lid only for categoric	al targets)	
Stopping criteria:	1.0E-3 💌		
Regularization parameter (C):	10 ≑		
Regression precision (epsilon):	0.1 ≑		
Kernel type:	Polynomial 🔝		
RBF gamma:	0.1 🌲	Bias:	0 ≑
Gamma:	1 荣	Degree:	3 🖨
OK 🕨 Run Cano			Apply Reset

- ► On the Expert tab, set Mode to Expert.
- ► Set Kernel type to Polynomial and click Run. The *class-poly* model nugget is added to the stream, and also to the Models palette at the top right of the screen.
- Connect the *class-rbf* model nugget to the *class-poly* model nugget (choose Replace at the warning dialog).
- Attach a Table node to the *class-poly* nugget.
- Open the Table node and click Run.

Classifying Cell Samples (SVM)

Comparing the Results

Figure 25-11 Fields added for Polynomial function

違 <u>F</u> ile	📄 Edit	×	<u>)</u> <u>G</u> ene	erate			0 ×
Table	Annotation	s					
	ormNucl	Mit	Class	\$S-Class	\$SP-Class	\$S1-Class	\$SP1-Class
78		1	2	2	0.992	2	0.998
79	1	1	2	2	0.968	2	0.967
80		1	2	2	0.998	2	0.994
81		1	2	2	0.986	2	0.991
82	2	1	2	2	0.996	2	0.997
83		1	2	2	0.991	2	0.998
84	8	1	2	2	0.970	2	0.998
85)	7	4	4	0.992	4	1.000
86)	10	4	4	0.974	4	1.000
87		1	4	4	0.786	4	0.958
88	8	3	4	4	0.988	4	0.935
89	1	1	2	2	0.995	2	0.997
90		1	2	2	0.998	2	0.991
91	2	1	2	2	0.999	2	0.993
92		1	2	2	0.998	2	0.996
93		1	2	2	0.995	2	0.997
94		1	2	2	0.999	2	0.994
95		1	2	2	0.998	2	0.995
96		1	2	2	0.999	2	0.993
97	1	1	2	2	0.999	2	0.995
	4						•

► Scroll the table output to the right to see the newly added fields.

The generated fields for the Polynomial function type are named \$S1-Class and \$SP1-Class.

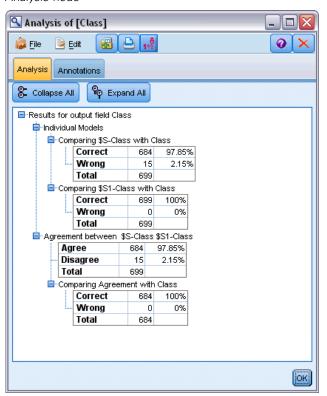
The results for Polynomial look much better. Many of the propensity scores are 0.995 or better, which is very encouraging.

▶ To confirm the improvement in the model, attach an Analysis node to the *class-poly* model nugget.

Open the Analysis node and click Run.

328

Figure 25-12 Analysis node



This technique with the Analysis node enables you to compare two or more model nuggets of the same type. The output from the Analysis node shows that the RBF function correctly predicts 97.85% of the cases, which is still quite good. However, the output shows that the Polynomial function has correctly predicted the diagnosis in every single case. In practice you are unlikely to see 100% accuracy, but you can use the Analysis node to help determine whether the model is acceptably accurate for your particular application.

In fact, neither of the other function types (Sigmoid and Linear) performs as well as Polynomial on this particular dataset. However, with a different dataset, the results could easily be different, so it's always worth trying the full range of options.

Summary

You have used different types of SVM kernel functions to predict a classification from a number of attributes. You have seen how different kernels give different results for the same dataset and how you can measure the improvement of one model over another.



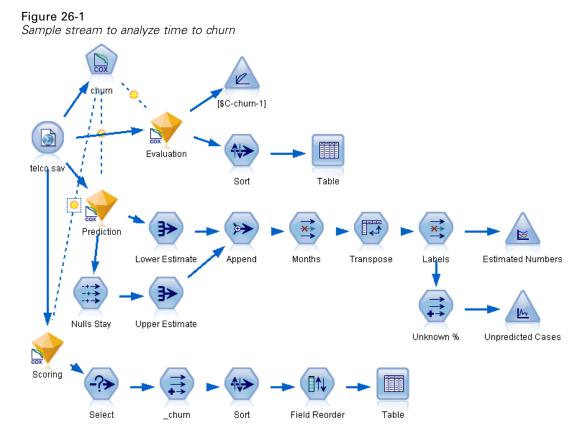
Using Cox Regression to Model Customer Time to Churn

As part of its efforts to reduce customer churn, a telecommunications company is interested in modeling the "time to churn" in order to determine the factors that are associated with customers who are quick to switch to another service. To this end, a random sample of customers is selected and their time spent as customers, whether they are still active customers, and various other fields are pulled from the database.

This example uses the stream *telco_coxreg.str*, which references the data file *telco.sav*. The data file is in the *Demos* folder and the stream file is in the *streams* subfolder. For more information, see the topic Demos Folder in Chapter 1 on p. 6.

Building a Suitable Model

• Add a Statistics File source node pointing to *telco.sav* in the *Demos* folder.



© Copyright IBM Corporation 1994, 2012.

330

• On the Filter tab of the source node, exclude the fields *region*, *income*, *longten* through *wireten*, and *loglong* through *logwire*.

Figure 26-2 Filtering unneeded fields		
📀 telco.sav		
\$CLEO_DEMOS/telco.sav		0 - 1
Data Filter Types Annotations		lds: 42 in, 12 filtered, 0 renamed, 30 out
Field -	Filter	Field
region	→	region 🖌
tenure		tenure
age	\rightarrow	age
marital	\rightarrow	marital
address	\rightarrow	address
income	→	income
ed	\rightarrow	ed
employ	\rightarrow	employ
retire	\rightarrow	retire
gender	\rightarrow	gender 📃 🔽
View current fields View OK Cancel	unused field settings	Apply Reset

(Alternatively, you could change the role to None for these fields on the Types tab rather than exclude it, or select the fields you want to use in the modeling node.)

• On the Types tab of the source node, set the role for the *churn* field to Target and set its measurement level to Flag. All other fields should have their role set to Input.

► Click Read Values to instantiate the data.

Figure 26-4

Velco.sav Preview Refresh \$CLEO_DEMOS/telco.sav								
Data Filter T	ypes Annotations							
4,- 00	🗪 🌓 Read Va	alues Clea	ar Values	Clear All V	alues			
Field 😑	Measurement	Values	Missing	Check	Role			
즞 pager	💑 Nominal	0,1		None	🔪 Input	4		
🔆 internet	💑 Nominal	0,1		None	🔪 Input			
즞 callid	💑 Nominal	0,1		None	🔪 Input			
📿 callwait	nominal 🛃	0,1		None	🔪 Input			
📿 forward	💑 Nominal	0,1		None	🔪 Input			
📿 confer	🂑 Nominal	0,1		None	🔪 Input			
	olominal ស	0,1		None	🔪 Input			
찾 ebill	A			bi-	🔪 Input			
ě Ininc	🔗 Continuous	[2.19722		None		- 12		
X	Continuous	[2.19722 1,2,3,4 1/0		None None None	Input Marget			

- Attach a Cox node to the source node; in the Fields tab, select *tenure* as the survival time variable.
 - Choosing field options 😡 churn X 0--Fields Model Expert Settings Annotations Survival time: 🔗 tenure -1 () Use type node settings O Use custom settings -1 Target: Inputs: -× Partition: -1 Splits: -× 🕨 Run Cancel OK Apply Reset

- ► Click the Model tab.
- ► Select Stepwise as the variable selection method.

Figure Choosii		odel oj	otions		
Chur Chur	'n				
Fields	Model	Expert	Settings	Annotations	
Model na	me:		() A	uto 🔘 Custom	
📝 Use p	partition	ed data			
📝 Build	model f	or each s	plit		
Method:	s	Stepwise		-	
Groups:					J
Model type	e: 🤇) Main ef	fects 🔘 C	Custom	
Model ter	ms:				
					*
					×
ОК		Run	Cancel		Apply Reset

• Click the Expert tab and select Expert to activate the expert modeling options.

Using Cox Regression to Model Customer Time to Churn

► Click Output.

Figure 26-6

Choosing advanced output options

Statistics		
Display:	🔘 At ead	ch step 🔘 At last step
CI for exp(B)	95 🔽	Correlation of estimates
🔲 Display baseline	e function	
🗹 Survival	🗹 Hazard	🔲 Log minus log 📃 One minus survival
Plot a separate line f	for each value:	
Plot a separate line f Value to use for plo Fie	ots:	Value ~
Value to use for plo	ots:	
Value to use for plo	ots:	Value ~
Value to use for plo Fie Contenure	ots:	Value
Value to use for plo Fie ∳ tenure ∳ age	ots:	Value
Value to use for plo Fie fie age age marital	ots:	Value
Value to use for plo Fie & tenure age address	ots:	Value
Value to use for plo	ots:	Value ~
Value to use for plo	ots:	Value ~ Mean Mean Mean Mean Mean

- ► Select Survival and Hazard as plots to produce, then click OK.
- Click Run to create the model nugget, which is added to the stream, and to the Models palette in the upper right corner. To view its details, double-click the nugget on the stream. First, look at the Advanced output tab.

Censored Cases

Figure 26-7 Case process

Jase	processing	summary	

		N	Percent
Cases available in	Event ^a	274	27.4%
analysis	Censored	726	72.6%
	Total	1000	100.0%
Cases dropped	Cases with missing values	0	.0%
	Cases with negative time	0	.0%
	Censored cases before the earliest event in a stratum	0	.0%
	Total	0	.0%
Total		1000	100.0%

a. Dependent Variable: Months with service

The status variable identifies whether the event has occurred for a given case. If the event has not occurred, the case is said to be censored. Censored cases are not used in the computation of the regression coefficients but are used to compute the baseline hazard. The case processing summary shows that 726 cases are censored. These are customers who have not churned.

Categorical Variable Codings

Figure 26-8

Categorical variable codings

		Frequency	(1) ^b	(2)	(3)	(4)
marital ^a	0=Unmarried	505	1			
5ho	1=Married	495	0			
eda	1=Did not complete high school	204	1	0	0	0
	2=High school degree	287	0	1	0	0
	3=Some college	209	0	0	1	0
	4=College degree	234	0	0	0	1
	5=Post-undergraduate degree	66	0	0	0	0
retireª	.00=No	953	1			
	1.00=Yes	47	0			
gender ^a	0=Male	483	1			
	1=Female	517	0			
tollfree ^a	0=No	526	1			
	1=Yes	474	0			
equipª	0=No	614	1			
	1=Yes	386	0			
callcard ^a	0=No	322	1			
	1=Yes	678	0			
wireless ^a	0=No	704	1			
	1=Yes	296	0			
multline ^a	0=No	525	1			
	1=Yes	475	0			
voice ^a	0=No	696	1			
	1=Yes	304	0			
pager ^a	0=No	739	1			
	1=Yes	261	0			
internet ^a	0=No	632	1			
	1=Yes	368	0			
callid ^a	0=No	519	1			
	1=Yes	481	0			
callwait ^a	0=No	515	1			
	1=Yes	485	0			
forward ^a	0=No	507	1			
	1=Yes	493	0			
confer ^a	0=No	498	1			
	1=Yes	502	0			
ebillª	0=No	629	1			
	1=Yes	371	0			
custcat ^a	1=Basic service	266	1	0	0	
	2=E-service	217	0	1	0	
	3=Plus service	281	0	0	1	
	4=Total service	236	0	0	0	

The categorical variable codings are a useful reference for interpreting the regression coefficients for categorical covariates, particularly dichotomous variables. By default, the reference category is the "last" category. Thus, for example, even though *Married* customers have variable values of 1 in the data file, they are coded as 0 for the purposes of the regression.

Variable Selection

Figure 26-9

Omnibus tests

		Overall (score)		Change From Previous Step			Change From Previous Block			
Step	-2 Log Likelihoo d	Chi- square	df	Siq.	Chi- square	df	Siq.	Chi- square	df	Siq.
1ª	3392.536	162.303	1	.000	133.828	1	.000	133.828	1	.000
2 ^b	3087.314	249.392	2	.000	305.222	1	.000	439.050	2	.000
3°	3027.085	328.426	3	.000	60.229	1	.000	499.279	3	.000
4 ^d	2990.790	347.197	4	.000	36.294	1	.000	535.574	4	.000
5 ^e	2973.790	362.673	5	.000	17.000	1	.000	552.574	5	.000
6 ^f	2958.796	376.140	6	.000	14.994	1	.000	567.568	6	.000
79	2945.503	384.717	7	.000	13.293	1	.000	580.861	7	.000
8 ^h	2936.993	417.341	8	.000	8.510	1	.004	589.371	8	.000
9 ⁱ	2926.000	423.911	9	.000	10.994	1	.001	600.364	9	.000
10 ^j	2917.551	428.078	10	.000	8.449	1	.004	608.813	10	.000
11 ^k	2913.308	436.837	11	.000	4.243	1	.039	613.056	11	.000
12 ¹	2908.078	440.158	12	.000	5.230	1	.022	618.286	12	.000

a. Variable(s) Entered at Step Number 1: callcard b. Variable(s) Entered at Step Number 2: longmon c. Variable(s) Entered at Step Number 3: equip d. Variable(s) Entered at Step Number 4: employ e. Variable(s) Entered at Step Number 5: multine

e. variable(s) Entered at Step Number 6: notaline f, Variable(s) Entered at Step Number 6: voice g, Variable(s) Entered at Step Number 7: address h, Variable(s) Entered at Step Number 9: ebill j, Variable(s) Entered at Step Number 10: callid k, Variable(s) Entered at Step Number 11: internet

), Variable(s) Entered at Step Number 11: internet I. Variable(s) Entered at Step Number 12: reside m. Beginning Block Number 0, initial Log Likelihood function: -2 Log likelihood: 3526.364 n. Beginning Block Number 1. Method = Forward Stepwise (Likelihood Ratio)

The model-building process employs a forward stepwise algorithm. The omnibus tests are measures of how well the model performs. The chi-square change from previous step is the difference between the -2 log-likelihood of the model at the previous step and the current step. If the step was to add a variable, the inclusion makes sense if the significance of the change is less than 0.05. If the step was to remove a variable, the exclusion makes sense if the significance of the change is greater than 0.10. In twelve steps, twelve variables are added to the model.

		В	SE	Wald	df	Siq.	Exp(B)
Step 12	address	035	.009	14.543	1	.000	.966
	employ	051	.010	25.767	1	.000	.950
	reside	103	.046	5.037	1	.025	.902
	equip	-1.948	.381	26.180	1	.000	.143
	callcard	.777	.151	26.451	1	.000	2.175
	longmon	233	.022	115.619	1	.000	.792
	equipmon	042	.011	15.377	1	.000	.959
	multline	.612	.145	17.854	1	.000	1.844
	voice	501	.157	10.197	1	.001	.606
	internet	362	.160	5.114	1	.024	.697
	callid	464	.148	9.790	1	.002	.629
	ebill	399	.156	6.557	1	.010	.671

Figure 26	3-10)				
Variables	in 1	the	equation	(step	12	only)

The final model includes *address*, *employ*, *reside*, *equip*, *callcard*, *longmon*, *equipmon*, *multline*, *voice*, *internet*, *callid*, and *ebill*. To understand the effects of individual predictors, look at Exp(B), which can be interpreted as the predicted change in the hazard for a unit increase in the predictor.

- The value of Exp(B) for *address* means that the churn hazard is reduced by 100%–(100%×0.966)=3.4% for each year a customer has lived at the same address. The churn hazard for a customer who has lived at the same address for five years is reduced by 100%–(100%×0.966⁵)=15.88%.
- The value of Exp(B) for *callcard* means that the churn hazard for a customer who does not subscribe to the calling card service is 2.175 times that of a customer with the service. Recall from the categorical variable codings that *No* = 1 for the regression.
- The value of Exp(B) for *internet* means that the churn hazard for a customer who does not subscribe to the internet service is 0.697 times that of a customer with the service. This is somewhat worrisome because it suggests that customers with the service are leaving the company faster than customers without the service.

Figure 26-11

Variables not in the model (step 12 only)

		Score	df	Siq.
Step 12	age	.122	1	.726
	marital	.648	1	.421
	income	1.476	1	.224
	ed	6.328	4	.176
	ed(1)	.007	1	.934
	ed(2)	.203	1	.652
	ed(3)	.835	1	.361
	ed(4)	5.773	1	.016
	retire	.013	1	.908
	gender	.214	1	.644
	tollfree	3.243	1	.072
	wireless	.668	1	.414
	tollmon	.000	1	.987
	cardmon	3.163	1	.075
	wiremon	1.084	1	.298
	pager	1.808	1	.179
	callwait	.266	1	.606
	forward	2.201	1	.138
	confer	2.568	1	.109
	custcat	.864	3	.834
	custcat(1)	.466	1	.495
	custcat(2)	.450	1	.502
	custcat(3)	.019	1	.889

Variables left out of the model all have score statistics with significance values greater than 0.05. However, the significance values for *tollfree* and *cardmon*, while not less than 0.05, are fairly close. They may be interesting to pursue in further studies.

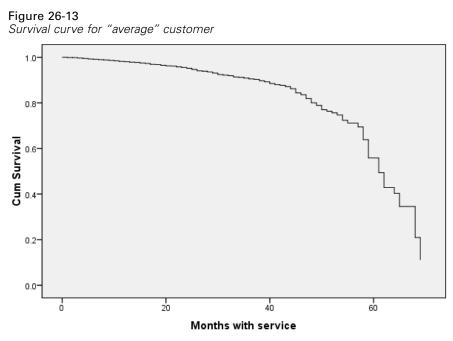
Covariate Means

Figure 26-12 Covariate means

	Mean
age	41.684
marital	.505
address	11.551
income	77.535
ed(1)	.204
ed(2)	.287
ed(3)	.209
ed(4)	.234
employ	10.987
retire	.953
gender	.483
reside	2.331
tollfree	.526
equip	.614
callcard	.322
wireless	.704
longmon	11.723
tollmon	13.274
equipmon	14.220
cardmon	13.781
wiremon	11.584
multline	.525
voice	.696
pager	.739
internet	.632
callid	.519
callwait	.515
forward	.507
confer	.498
ebill	.629
custcat(1)	.266
custcat(2)	.217
custcat(3)	.281

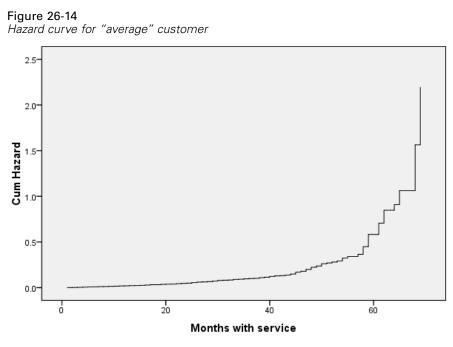
This table displays the average value of each predictor variable. This table is a useful reference when looking at the survival plots, which are constructed for the mean values. Note, however, that the "average" customer doesn't actually exist when you look at the means of indicator variables for categorical predictors. Even with all scale predictors, you are unlikely to find a customer whose covariate values are all close to the mean. If you want to see the survival curve for a particular case, you can change the covariate values at which the survival curve is plotted in the Plots dialog box. If you want to see the survival curve for a particular case, you can change the survival curve is plotted in the Plots group of the Advanced Output dialog.

Survival Curve



The basic survival curve is a visual display of the model-predicted time to churn for the "average" customer. The horizontal axis shows the time to event. The vertical axis shows the probability of survival. Thus, any point on the survival curve shows the probability that the "average" customer will remain a customer past that time. Past 55 months, the survival curve becomes less smooth. There are fewer customers who have been with the company for that long, so there is less information available, and thus the curve is blocky.

Hazard Curve



The basic hazard curve is a visual display of the cumulative model-predicted potential to churn for the "average" customer. The horizontal axis shows the time to event. The vertical axis shows the cumulative hazard, equal to the negative log of the survival probability. Past 55 months, the hazard curve, like the survival curve, becomes less smooth, for the same reason.

Using Cox Regression to Model Customer Time to Churn

Evaluation

The stepwise selection methods guarantee that your model will have only "statistically significant" predictors, but it does not guarantee that the model is actually good at predicting the target. To do this, you need to analyze scored records.

Figure 26-15 Cox nugget: Settings tab

	3	0				
😡 Evalua	ation			_		
COX	違 <u>F</u> ile	🕙 <u>G</u> enerat	e 🌔 Prev	view) 🔒	0	
Settings	Advanced	Summary	Annotations			
Predict sur	vival at futur	e times spec	ified as:			
🔘 Regular	intervals	Tim	e interval:		1.0 🌲	
		Nur	nber of time pe	riods to score:	1 🌲	
Time fiel	ld	🛷 ti	enure			-
Past surviv	al time:					-
📝 Append	l all probabilit	ies				
📃 Calculat	e cumulative	hazard fund	tion			
ОК	Cancel				Apply	Reset

- Place the model nugget on the canvas and attach it to the source node, open the nugget and click the Settings tab.
- Select Time field and specify *tenure*. Each record will be scored at its length of tenure.
- ► Select Append all probabilities.

This creates scores using 0.5 as the cutoff for whether a customer churns; if their propensity to churn is greater than 0.5, they are scored as a churner. There is nothing magical about this number, and a different cutoff may yield more desirable results. For one way to think about choosing a cutoff, use an Evaluation node.

Figure 26-16

Evaluation node: Plot tab 💟 [\$C-churn-1] X 0 - -R Plot Options Appearance Output Annotations 🔘 Gains O Response 🔘 Lift 🔘 Profit OROI Chart type: **V** Cumulative plot 📝 Include baseline 🛛 📝 Include best line Models Find predicted/predictor fields using: Model output field metadata ◯ Field name format (for example, '\$<x>-<target field>') -Other Score Fields Plot score fields -1 × -1 Target: 👿 Separate by partition Percentiles 🔻 Plot: 🔘 Line 🛛 Point Style: -1 5.0 🌲 🔘 Variable Costs: Fixed -1 10.0 🌲 Revenue: Fixed O Variable -1 1.0 韋 Weight: Fixed O Variable OK 🕨 Run Cancel Apply Reset

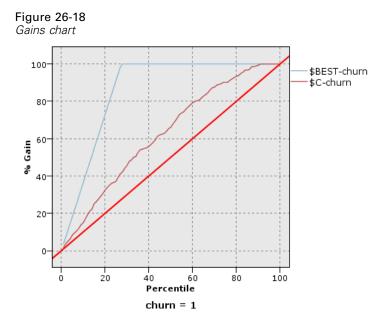
- Attach an Evaluation node to the model nugget; on the Plot tab, select Include best line.
- ► Click the Options tab.

Using Cox Regression to Model Customer Time to Churn

Figure 26-17 Evaluation node: Options tab

		_
😡 [\$C-chur	n-1]	
		0
Plot Options	Appearance Output Annotations	
📃 User define	d hit	
Condition:		
👿 User define	d score	
Expression:	'\$CP-1-1'	
🔲 Include bus	iness rule	
Condition:		
📃 Export resu	lts to file	
Filename:	output.txt	
Delimiter:		
🔽 Include field	I names 🛛 👿 New line after each record	
ок 🕨	Run Cancel	Apply Reset

- ► Select User defined score and type '\$CP-1-1' as the expression. This is a model-generated field that corresponds to the propensity to churn.
- ► Click Run.

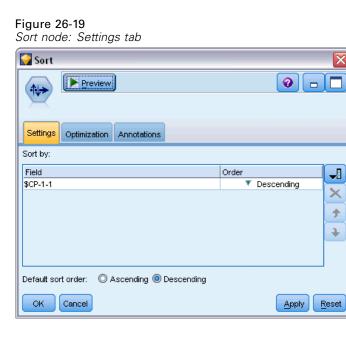


The cumulative gains chart shows the percentage of the overall number of cases in a given category "gained" by targeting a percentage of the total number of cases. For example, one point on the curve is at (10%, 15%), meaning that if you score a dataset with the model and sort all of the cases by predicted propensity to churn, you would expect the top 10% to contain approximately 15% of all of the cases that actually take the category I (churners). Likewise, the top 60% contains approximately 79.2% of the churners. If you select 100% of the scored dataset, you obtain all of the churners in the dataset.

The diagonal line is the "baseline" curve; if you select 20% of the records from the scored dataset at random, you would expect to "gain" approximately 20% of all of the records that actually take the category *I*. The farther above the baseline a curve lies, the greater the gain. The "best" line shows the curve for a "perfect" model that assigns a higher churn propensity score to every churner than every non-churner. You can use the cumulative gains chart to help choose a classification cutoff by choosing a percentage that corresponds to a desirable gain, and then mapping that percentage to the appropriate cutoff value.

What constitutes a "desirable" gain depends on the cost of Type I and Type II errors. That is, what is the cost of classifying a churner as a non-churner (Type I)? What is the cost of classifying a non-churner as a churner (Type II)? If customer retention is the primary concern, then you want to lower your Type I error; on the cumulative gains chart, this might correspond to increased customer care for customers in the top 60% of predicted propensity of *1*, which captures 79.2% of the possible churners but costs time and resources that could be spent acquiring new customers. If lowering the cost of maintaining your current customer base is the priority, then you want to lower your Type II error. On the chart, this might correspond to increased customer care for the top 20%, which captures 32.5% of the churners. Usually, both are important concerns, so you have to choose a decision rule for classifying customers that gives the best mix of sensitivity and specificity.

Using Cox Regression to Model Customer Time to Churn



- ► Say that you have decided that 45.6% is a desirable gain, which corresponds to taking the top 30% of records. To find an appropriate classification cutoff, attach a Sort node to the model nugget.
- ▶ On the Settings tab, choose to sort by *\$CP-1-1* in descending order and click OK.

🛛 Table	(34 fields	s, 1,000 records)		_	
诊 <u>F</u> ile	📄 <u>E</u> dit	🏷 Generate 🛛 👪			0)>
Table					
Table d	Annotations				
	Irn \$C-chur	n-1 \$CP-churn-1	\$CP-0-1	\$CP-1-1	
292	0	0.744	0.744	0.256	1
293	0	0.745	0.745	0.255	
294	0	0.745	0.745	0.255	
295	0	0.746	0.746	0.254	
296	0	0.748	0.748	0.252	
297	0	0.749	0.749	0.251	
298	0	0.749	0.749	0.251	
299	0	0.750	0.750	0.250	
300	0	0.752	0.752	0.248	
301	0	0.752	0.752	0.248	
302	0	0.754	0.754	0.246	
303	0	0.754	0.754	0.246	
304	0	0.755	0.755	0.245	
305	0	0.756	0.756	0.244	
306	0	0.757	0.757	0.243	
307	0	0.757	0.757	0.243	
308	0	0.758	0.758	0.242	
309	0	0.759	0.759	0.241	
310	0	0.761	0.761	0.239	
311	0	0.762	0.762	0.238	
	4				

- Attach a Table node to the Sort node.
- Open the Table node and click Run.

Scrolling down the output, you see that the value of CP-1-1 is 0.248 for the 300th record. Using 0.248 as a classification cutoff should result in approximately 30% of the customers scored as churners, capturing approximately 45% of the actual total churners.

Tracking the Expected Number of Customers Retained

Once satisfied with a model, you want to track the expected number of customers in the dataset that are retained over the next two years. The null values, which are customers whose total tenure (future time + *tenure*) falls beyond the range of survival times in the data used to train the model, present an interesting challenge. One way to deal with them is to create two sets of predictions, one in which null values are assumed to have churned, and another in which they

Using Cox Regression to Model Customer Time to Churn

are assumed to have been retained. In this way you can establish upper and lower bounds on the expected number of customers retained.

Figure 26-21 Cox nugget: Settings	s tab
Prediction	
	enerate Preview 🔒 🧿 🗖 🗖
Settings Advanced Sun	mary Annotations
Predict survival at future time	s specified as:
Regular intervals	Time interval: 1.0 🗧
	Number of time periods to score: 24 🗲
O Time field	
Past survival time:	🔗 tenure
📝 Append all probabilities	
🔲 Calculate cumulative haza	rd function
OK Cancel	Apply Reset

- Double-click the model nugget in the Models palette (or copy and paste the nugget on the stream canvas) and attach the new nugget to the Source node.
- Open the nugget to the Settings tab.
- ► Make sure Regular Intervals is selected, and specify 1.0 as the time interval and 24 as the number of periods to score. This specifies that each record will be scored for each of the following 24 months.
- Select *tenure* as the field to specify the past survival time. The scoring algorithm will take into account the length of each customer's time as a customer of the company.
- Select Append all probabilities.

Figure 26-22		
Aggregate node:	Settings	tab

😡 Lower Estimate						
					0	
Settings Annotations						
Key fields:					📃 Keys are	e contiguous
Aggregate fields:						×
Field	Sum	Mean	Min	Max	SDev	
\$CP-0-1	-					
\$CP-0-10	-					×
\$CP-0-11	-					
\$CP-0-12	\checkmark					
\$CP-0-13	-					
\$CP-0-14	-					-
Default mode: Sum Mean Min Max SDev						
New field name extension	on:		Add as:	Suffix	O Prefix	
Include record count	in field Rec	ord_Count				
\frown					_	

- Attach an Aggregate node to the model nugget; on the Settings tab, deselect Mean as a default mode.
- ► Select *\$CP-0-1* through *\$CP-0-24*, the fields of form *\$CP-0-n*, as the fields to aggregate. This is easiest if, on the Select Fields dialog, you sort the fields by Name (that is, alphabetical order).
- ► Deselect Include record count in field.
- Click OK. This node creates the "lower bound" predictions.

Using Cox Regression to Model Customer Time to Churn

Figure 26-23 Filler node: Settings tab	
Nulls Stay	
Settings Annotations Fill in fields:	
Fill In Tields:	
♦ \$CP-0-10 ♦ \$CP-0-11	×
✓ \$CP-0-12	
Replace: Null values	
Condition:	
@BLANK(@FIELD)	
Replace with:	
1	
OK Cancel	Apply Reset

- ► Attach a Filler node to the Coxreg nugget to which we just attached the Aggregate node; on the Settings tab, select *\$CP-0-1* through *\$CP-0-24*, the fields of form *\$CP-0-n*, as the fields to fill in. This is easiest if, on the Select Fields dialog, you sort the fields by Name (that is, alphabetical order).
- Choose to replace Null values with the value 1.
- ► Click OK.

Figure 26-24		
Aggregate node:	Settings	tab

🜍 Upper Estimate						
)				0	
Settings Annotations						
Key fields:					📃 Keys are	e contiguous
Aggregate fields:						
Field	Sum	Mean	Min	Max	SDev	
\$CP-0-1	-					
\$CP-0-10	\checkmark					X
\$CP-0-11	\checkmark					
\$CP-0-12	-					
\$CP-0-13	-					
\$CP-0-14	-					-
Default mode: Sum Mean Min Max SDev						
New field name extensio	n:		Add as:	Suffix	O Prefix	
E Include record count	in field Rec	ord_Count				
OK Cancel					Apply	/ Reset

- Attach an Aggregate node to the Filler node; on the Settings tab, deselect Mean as a default mode.
- ► Select *\$CP-0-1* through *\$CP-0-24*, the fields of form *\$CP-0-n*, as the fields to aggregate. This is easiest if, on the Select Fields dialog, you sort the fields by Name (that is, alphabetical order).
- ► Deselect Include record count in field.
- ► Click OK. This node creates the "upper bound" predictions.

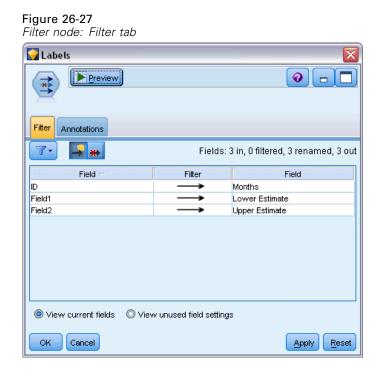
🖓 Months			
Preview		0	
Filter Annotations			
Fields: 24 in, 0 filtered, 24 renamed, 24 out			
Field 📼	Filter	Field	
\$CP-0-1_Sum	\rightarrow	1	- 4
\$CP-0-2_Sum	\rightarrow	2	
\$CP-0-3_Sum	\rightarrow	3	
		4	
\$CP-0-4_Sum		-	
\$CP-0-5_Sum	\rightarrow	5	_
\$CP-0-5_Sum \$CP-0-6_Sum	\rightarrow	6	
\$CP-0-5_Sum \$CP-0-6_Sum \$CP-0-7_Sum	$\begin{array}{c} \uparrow \\ \uparrow $	6 7	
\$CP-0-5_Sum \$CP-0-6_Sum \$CP-0-7_Sum \$CP-0-8_Sum	111 111	6 7 8	
\$CP-0-5_Sum \$CP-0-6_Sum \$CP-0-7_Sum	1111	6 7	

- Attach an Append node to the two Aggregate nodes, then attach a Filter node to the Append node.
- ▶ On the Settings tab of the Filter node, rename the fields to *1* through 24. Through the use of a Transpose node, these field names will become values for the *x*-axis in charts downstream.

Figure 26-26 Transpose node: Settings tab

💟 Transpose	
Previe	
Settings Annotation	3
New field names:	
🔘 Use prefix	Field Number of new fields: 2
C Read from field	
▶ Read Values	New Field Names
	Maximum number of values to read: 500 🗬
Transpose: 🔘 All r	numeric 🔘 All string 🔘 Custom
Fields:	
	×
Row ID name: ID	
OK Cancel	Apply Reset

- ► Attach a Transpose node to the Filter node.
- Type 2 as the number of new fields.



- ► Attach a Filter node to the Transpose node.
- ► On the Settings tab of the Filter node, rename *ID* to *Months*, *Field1* to *Lower Estimate*, and *Field2* to *Upper Estimate*.

Figure 26-28 Multiplot node: Plot tab

🖸 Estimated Numbers 🛛 🛛 🛛
Plot Appearance Output Annotations
X field: 🔊 Months
Y fields: Vpper Estimate
_Overlay
Panel:
Normalize
Overlay function y = :
When number of records greater than: 2000 🗲
OK Run Cancel <u>Apply</u> <u>R</u> eset

- Attach a Multiplot node to the Filter node.
- On the Plot tab, *Months* as the X field, *Lower Estimate* and *Upper Estimate* as the Y fields.

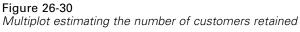
Using Cox Regression to Model Customer Time to Churn

Figure 26-29 Multiplot node: Appearance tab

🚰 Estimated Numbers 🛛 🔀
Plot Appearance Output Annotations
Title: Number of Customers
Subtitle:
Caption: Estimates the number of customers retained
X label: Auto Custom
Y label: 🔘 Auto 🔘 Custom
☑ Display gridline
OK Run Cancel Apply Reset

- Click the Appearance tab.
- Type Number of Customers as the title.
- ► Type Estimates the number of customers retained as the caption.
- ► Click Run.





Estimates the number of customers retained

The upper and lower bounds on the estimated number of customers retained are plotted. The difference between the two lines is the number of customers scored as null, and therefore whose status is highly uncertain. Over time, the number of these customers increases. After 12 months, you can expect to retain between 601 and 735 of the original customers in the dataset; after 24 months, between 288 and 597.

Figure 26-31 Derive node: Settings tab

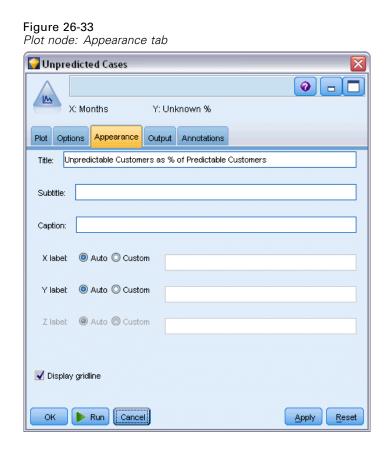
Unknown %	X
Derive as: Formula	0
Settings Annotations	
Mode: 💿 Single 🔘 Multiple	
Derive field:	
Unknown %	
Derive as: Formula Field type: Continuous Formula:	
(100 * ('Upper Estimate' - 'Lower Estimate')) / 'Lower Estimate'	
OK Cancel	Apply Reset

- ► To get another look at how uncertain the estimates of the number of customers retained are, attach a Derive node to the Filter node.
- ▶ On the Settings tab of the Derive node, type *Unknown* % as the derive field.
- ► Select Continuous as the field type.
- ► Type (100 * ('Upper Estimate' 'Lower Estimate')) / 'Lower Estimate' as the formula. Unknown % is the number of customers "in doubt" as a percentage of the lower estimate.
- ► Click OK.

Figure 26-32 Plot node: Plot tab
Unpredicted Cases
X: Months Y: Unknown %
Plot Options Appearance Output Annotations
L X field: Months Y field: V field:
Overlay Color: Image: Size: Image: Size: <tr< td=""></tr<>
Overlay type: None Smoother
Function y =
OK Run Cancel Apply Reset

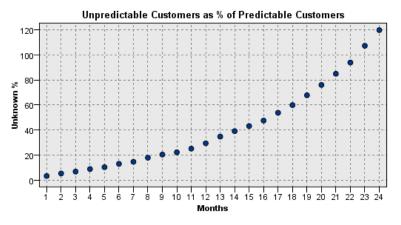
- ► Attach a Plot node to the Derive node.
- On the Plot tab of the Plot node, select *Months* as the X field and *Unknown* % as the Y field.
- Click the Appearance tab.

Using Cox Regression to Model Customer Time to Churn



- ▶ Type Unpredictable Customers as % of Predictable Customers as the title.
- Execute the node.





Through the first year, the percentage of unpredictable customers increases at a fairly linear rate, but the rate of increase explodes during the second year until, by month 23, the number of customers with null values outnumber the expected number of customers retained.

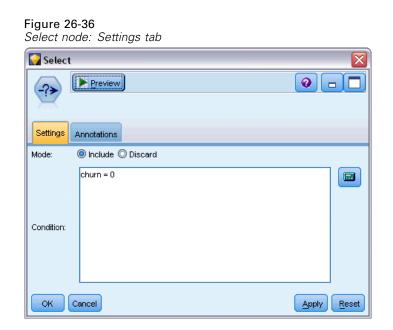
Scoring

Once satisfied with a model, you want to score customers to identify the individuals most likely to churn within the next year, by quarter.

igure 26-35 Coxreg nugget: Settings tab
Scoring 🛛 🕅
👷 🖟 File 🖏 Generate 🕞 Preview) 🐻 🛛 🕤 🗖
Settings Advanced Summary Annotations
Predict survival at future times specified as:
Regular intervals
Number of time periods to score:
🔍 Time field 📃
Past survival time: 🔗 tenure
🖌 Append all probabilities
Calculate cumulative hazard function
OK Cancel Apply Reset

- Attach a third model nugget to the Source node and open the model nugget.
- ► Make sure Regular Intervals is selected, and specify 3.0 as the time interval and 4 as the number of periods to score. This specifies that each record will be scored for the following four quarters.
- ► Select *tenure* as the field to specify the past survival time. The scoring algorithm will take into account the length of each customer's time as a customer of the company.
- Select Append all probabilities. These extra fields will make it easier to sort the records for viewing in a table.

Using Cox Regression to Model Customer Time to Churn



► Attach a Select node to the model nugget; on the Settings tab, type churn=0 as the condition. This removes customers who have already churned from the results table.

Figure 26-37

Derive node: Settings ta	
🔽 _churn	×
	0 - 🗖
Derive as: Conditiona	al
Settings Annotations	
Mode	e: 🔘 Single 🔘 Multiple
Derive from:	
\$CP-1-1 \$CP-1-2	
\$ \$CP-1-2 \$ \$CP-1-3	$\overline{+}$ \times
Field name extension:churn	Add as: 🔘 Suffix 🔘 Prefix
Derive as: Conditional 🔻	TIP: Refer to selected fields by using @FIELD
Field type: 🎖 Flag 🔻	
lf:	
@FIELD>0.248	
Then:	
1	
Else:	
OK Cancel	Apply

- Attach a Derive node to the Select node; on the Settings tab, select Multiple as the mode.
- ► Choose to derive from *\$CP-1-1* through *\$CP-1-4*, the fields of form *\$CP-1-n*, and type _churn as the suffix to add. This is easiest if, on the Select Fields dialog, you sort the fields by Name (that is, alphabetical order).
- Choose to derive the field as a Conditional.
- ► Select Flag as the measurement level.
- Type @FIELD>0.248 as the If condition. Recall that this was the classification cutoff identified during Evaluation.
- ► Type 1 as the Then expression.
- ► Type 0 as the Else expression.
- ► Click OK.

Using Cox Regression to Model Customer Time to Churn

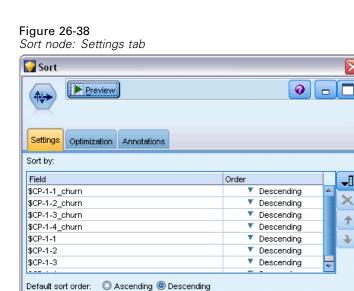
J

×

+

÷

Reset



OK

Cancel

▶ Attach a Sort node to the Derive node; on the Settings tab, choose to sort by \$CP-1-1 churn through \$CP-1-4-churn and then \$CP-1-1 through \$CP-1-4, all in descending order. Customers who are predicted to churn will appear at the top.

Apply

Custom Order O Automatic Sort Type: Name: X Storage: Storage SCP-1-1_churn SCP-1-1_churn SCP-1-1 SCP-1-2_churn SCP-1-2_churn SCP-1-2_churn SCP-1-2_churn SCP-1-3_churn SCP-1-3 SCP-1-3 SCP-1-3	Figure 26-39 Field Reorder node: Reorder	tab	
\$CP-1-1_churn (Unknown) \$CP-1-1 Preal \$CP-1-2_churn (Unknown) \$CP-1-3_churn (Unknown) \$CP-1-3_churn (Unknown) \$CP-1-3_churn (Unknown) \$CP-1-4_churn Preal \$CP-1-4_churn Preal			
Note: Fields added down stream of this node are not reordered.	\$CP-1-1_churn \$CP-1-1 \$CP-1-2_churn \$CP-1-2 \$CP-1-3_churn \$CP-1-3 \$CP-1-4	 (Unknown) Real (Unknown) Real (Unknown) Real (Unknown) Real (Unknown) Real (Unknown) Real 	

▶ Attach a Field Reorder node to the Sort node; on the Reorder tab, choose to place \$*CP-1-1 churn* through \$CP-1-4 in front of the other fields. This simply makes the results table easier to read,

and so is optional. You will need to use the buttons to move the fields into the position shown in the figure.

違 <u>F</u> ile	📄 <u>E</u> dit 🛛 🕙	<u>G</u> enerate		# #					0)>
Table ,	Annotations								
	\$CP-1-1_churn	\$CP-1-1	\$CP-1-2_churn	\$CP-1-2	\$CP-1-3_churn	\$CP-1-3	\$CP-1-4_churn	\$CP-1-4	tenur
255	0	0.032	0	0.075	0	0.147	1	0.298	49 4
256	0	0.027	0	0.064	0	0.127	1	0.260	49
257	0	0.023	0	0.130	0	0.233	1	0.308	53
258	0	0.021	0	0.127	0	0.239	1	0.320	54
259	0	0.021	0	0.125	0	0.237	1	0.318	54
260	0	0.021	0	0.053	0	0.198	1	0.331	50
261	0	0.021	0	0.053	0	0.196	1	0.329	50
262	0	0.020	0	0.050	0	0.189	1	0.317	50
263	0	0.017	0	0.043	0	0.163	1	0.278	50
264	0	0.015	0	0.039	0	0.148	1	0.253	50
265	0	0.197	0	0.197	0	\$null\$		\$null\$	66
266	0	0.109	0	0.109	0	\$null\$	0	\$null\$	66
267	0	0.101	-	0.214	0	\$null\$		\$null\$	65
268	0	0.081		0.137	0	0.194		0.245	23
269	0	0.074		0.159	0	\$null\$		\$null\$	65
270	0	0.070		0.116	0	0.158		0.237	28
271	0	0.070	-	0.128	0	0.189	-	0.234	45
272	0	0.062	0	0.105	0	0.151	0	0.191	23
273	0	0.062		0.130	0	0.163		0.212	44
274	0	0.061	0	0.123	0	0.182	0	0.241	4

Figure 26-40 Table showing customer scores

• Attach a Table node to the Field Reorder node and execute it.

264 customers are expected to churn by the end of the year, 184 by the end of the third quarter, 103 by the second, and 31 in the first. Note that given two customers, the one with a higher propensity to churn in the first quarter does not necessarily have a higher propensity to churn in later quarters; for example, see records 256 and 260. This is likely due to the shape of the hazard function for the months following the customer's current tenure; for example, customers who joined because of a promotion might be more likely to switch early on than customers who joined because of a personal recommendation, but if they do not then they may actually be more loyal for their remaining tenure. You may want to re-sort the customers to obtain different views of the customers most likely to churn.

Figure 26-41

Table showing customers with null values

違 <u>F</u> ile	📄 Edit 🛛 🕙 🤇	⊇enerate		22					ଡ଼ୗୄୖ
Table ,	Annotations								
	\$CP-1-1_churn	\$CP-1-1	\$CP-1-2_churn	\$CP-1-2	\$CP-1-3_churn	\$CP-1-3	\$CP-1-4_churn	\$CP-1-4	tenur
707	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	71 1
708	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	71
709	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	71
710	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	72
711	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	71
712	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	72
713	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	72
714	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	72
715	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	70
716	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	70
717	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	71
718	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	72
719	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	72
720	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	72
721	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	72
722	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	71
723	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	70
724	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	71
725	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	70
726	0	\$null\$	0	\$null\$	0	\$null\$	0	\$null\$	72 •
	4								•

At the bottom of the table are customers with predicted null values. These are customers whose total tenure (future time + *tenure*) falls beyond the range of survival times in the data used to train the model.

Summary

Using Cox regression, you have found an acceptable model for the time to churn, plotted the expected number of customers retained over the next two years, and identified the individual customers most likely to churn in the next year. Note that while this is an acceptable model, it may not be the best model. Ideally you should at least compare this model, obtained using the Forward stepwise method, with one created using the Backward stepwise method.

Explanations of the mathematical foundations of the modeling methods used in IBM® SPSS® Modeler are listed in the SPSS Modeler Algorithms Guide.



Market Basket Analysis (Rule Induction/C5.0)

This example deals with fictitious data describing the contents of supermarket baskets (that is, collections of items bought together) plus the associated personal data of the purchaser, which might be acquired through a loyalty card scheme. The goal is to discover groups of customers who buy similar products and can be characterized demographically, such as by age, income, and so on.

This example illustrates two phases of data mining:

- Association rule modeling and a web display revealing links between items purchased
- C5.0 rule induction profiling the purchasers of identified product groups

Note: This application does not make direct use of predictive modeling, so there is no accuracy measurement for the resulting models and no associated training/test distinction in the data mining process.

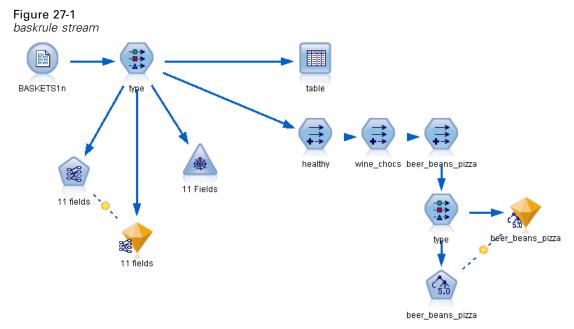
This example uses the stream named *baskrule*, which references the data file named *BASKETS1n*. These files are available from the *Demos* directory of any IBM® SPSS® Modeler installation. This can be accessed from the IBM® SPSS® Modeler program group on the Windows Start menu. The *baskrule* file is in the *streams* directory.

Accessing the Data

Using a Variable File node, connect to the dataset *BASKETS1n*, selecting to read field names from the file. Connect a Type node to the data source, and then connect the node to a Table node. Set the measurement level of the field *cardid* to *Typeless* (because each loyalty card ID occurs only once in

Market Basket Analysis (Rule Induction/C5.0)

the dataset and can therefore be of no use in modeling). Select *Nominal* as the measurement level for the field *sex* (this is to ensure that the Apriori modeling algorithm will not treat *sex* as a flag).



Now run the stream to instantiate the Type node and display the table. The dataset contains 18 fields, with each record representing a basket.

The 18 fields are presented in the following headings.

Basket summary:

- *cardid*. Loyalty card identifier for customer purchasing this basket.
- *value*. Total purchase price of basket.
- *pmethod*. Method of payment for basket.

Personal details of cardholder:

- sex
- *homeown*. Whether or not cardholder is a homeowner.
- income
- age

Basket contents—flags for presence of product categories:

- fruitveg
- freshmeat
- dairy
- cannedveg
- cannedmeat

- frozenmeal
- beer
- wine
- softdrink
- fish
- confectionery

Discovering Affinities in Basket Contents

First, you need to acquire an overall picture of affinities (associations) in the basket contents using Apriori to produce association rules. Select the fields to be used in this modeling process by editing the Type node and setting the role of all of the product categories to *Both* and all other roles to *None*. (*Both* means that the field can be either an input or an output of the resultant model.)

Note: You can set options for multiple fields using Shift-click to select the fields before specifying an option from the columns.

Figure 27-2 Selecting fields for modeling

type Image: Preview Types Format Annotations												
4. 00	🗪 🛛 🕨 Read Valu	ues Clear	^r Values	Clear All Va	lues							
Field -	Measurement	Values	Missing	Check	Role							
A sex	💑 Nominal	F,M		None	🛇 None 🔺							
A homeown	🎖 Flag	YES/NO		None	🛇 None							
父 income	🔗 Continuous	[10200,3		None	🛇 None 🛛							
今 age	🔗 Continuous	[16,50]		None	🛇 None							
A fruitveg	🎖 Flag	T/F		None	🔘 Both 💌							
\Lambda freshmeat	🎖 Flag 🎖 Flag	T/F		None	🔪 Input							
\Lambda dairy	🎖 Flag	T/F		None	🔘 Target							
\Lambda cannedveg	🎖 Flag	T/F		None	Both							
accordment	V Flow	TÆ		blong	O None							
View current	fields 🔘 View unuse	ed field setting	ļS		Partition Split							
OK Cance				(A Frequencet							
					d Record ID							

Once you have specified fields for modeling, attach an Apriori node to the Type node, edit it, select the option Only true values for flags, and click run on the Apriori node. The result, a model on the Models tab at the upper right of the managers window, contains association rules that you can view by using the context menu and selecting Browse.

Figure 27-3 Association rules												
🚺 11 fields 🛛 🛛 🔀												
Elle 🐑 Generate 🕞 Preview) 📾 🥥 🗖 🗖												
Model Settings Summary Annotations												
Sort by: Confidence %												
Consequent	Antecedent	Support %	Confidence %									
frozenmeal	beer cannedveg	16.7	87.425									
cannedveg	beer frozenmeal	17.0	85.882									
beer	frozenmeal cannedveg	17.3	84.393									
OK Cancel			Apply Reset									

These rules show a variety of associations between frozen meals, canned vegetables, and beer. The presence of two-way association rules, such as:

frozenmeal -> beer beer -> frozenmeal

suggests that a web display (which shows only two-way associations) might highlight some of the patterns in this data.

Attach a Web node to the Type node, edit the Web node, select all of the basket contents fields, select Show true flags only, and click run on the Web node.

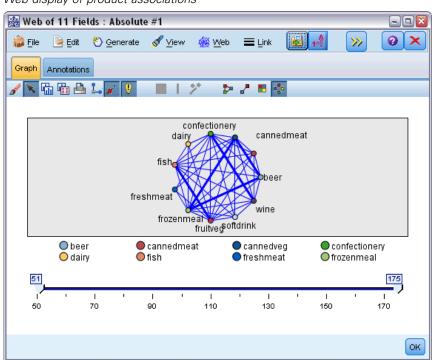
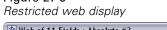
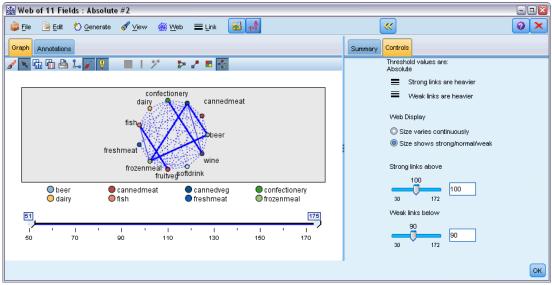


Figure 27-4 *Web display of product associations*

Because most combinations of product categories occur in several baskets, the strong links on this web are too numerous to show the groups of customers suggested by the model.

Figure 27-5





- ► To specify weak and strong connections, click the yellow double arrow button on the toolbar. This expands the dialog box showing the web output summary and controls.
- ► Select Size shows strong/normal/weak.
- ▶ Set weak links below 90.
- ► Set strong links above 100.

In the resulting display, three groups of customers stand out:

- Those who buy fish and fruits and vegetables, who might be called "healthy eaters"
- Those who buy wine and confectionery
- Those who buy beer, frozen meals, and canned vegetables ("beer, beans, and pizza")

Profiling the Customer Groups

You have now identified three groups of customers based on the types of products they buy, but you would also like to know who these customers are—that is, their demographic profile. This can be achieved by tagging each customer with a flag for each of these groups and using rule induction (C5.0) to build rule-based profiles of these flags.

First, you must derive a flag for each group. This can be automatically generated using the web display that you just created. Using the right mouse button, click the link between *fruitveg* and *fish* to highlight it, then right-click and select Generate Derive Node For Link.

Deriving a flag for each customer group

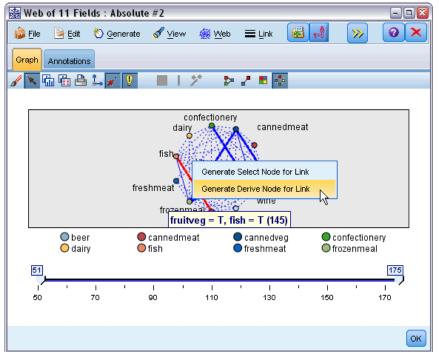


Figure 27-6

Edit the resulting Derive node to change the Derive field name to *healthy*. Repeat the exercise with the link from *wine* to *confectionery*, naming the resultant Derive field *wine_chocs*.

For the third group (involving three links), first make sure that no links are selected. Then select all three links in the *cannedveg*, *beer*, and *frozenmeal* triangle by holding down the shift key while you click the left mouse button. (Be sure you are in Interactive mode rather than Edit mode.) Then from the web display menus choose:

Generate > Derive Node ("And")

Change the name of the resultant Derive field to *beer_beans_pizza*.

To profile these customer groups, connect the existing Type node to these three Derive nodes in series, and then attach another Type node. In the new Type node, set the role of all fields to *None*, except for *value*, *pmethod*, *sex*, *homeown*, *income*, and *age*, which should be set to *Input*, and the relevant customer group (for example, *beer_beans_pizza*), which should be set to *Target*. Attach a C5.0 node, set the Output type to Rule set, and click run on the node. The resultant model (for *beer_beans_pizza*) contains a clear demographic profile for this customer group:

Rule 1 for T: if sex = M and income <= 16,900 then T

The same method can be applied to the other customer group flags by selecting them as the output in the second Type node. A wider range of alternative profiles can be generated by using Apriori instead of C5.0 in this context; Apriori can also be used to profile all of the customer group flags simultaneously because it is not restricted to a single output field.

Summary

This example reveals how IBM® SPSS® Modeler can be used to discover affinities, or links, in a database, both by modeling (using Apriori) and by visualization (using a web display). These links correspond to groupings of cases in the data, and these groups can be investigated in detail and profiled by modeling (using C5.0 rule sets).

In the retail domain, such customer groupings might, for example, be used to target special offers to improve the response rates to direct mailings or to customize the range of products stocked by a branch to match the demands of its demographic base.



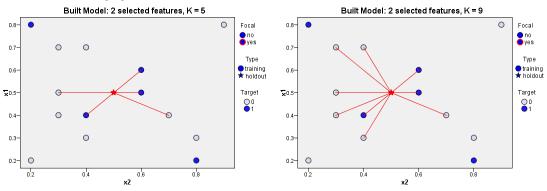
Assessing New Vehicle Offerings (KNN)

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

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

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





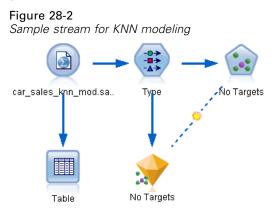
Nearest neighbor analysis can also be used to compute values for a continuous target. In this situation, the average or median target value of the nearest neighbors is used to obtain the predicted value for the new case.

An automobile manufacturer has developed prototypes for two new vehicles, a car and a truck. Before introducing the new models into its range, the manufacturer wants to determine which existing vehicles on the market are most like the prototypes—that is, which vehicles are their "nearest neighbors", and therefore which models they will be competing against.

The manufacturer has collected data about the existing models under a number of categories, and has added the details of its prototypes. The categories under which the models are to be compared include price in thousands (*price*), engine size (*engine_s*), horsepower (*horsepow*), wheelbase (*wheelbas*), width (*width*), length (*length*), curb weight (*curb_wgt*), fuel capacity (*fuel_cap*) and fuel efficiency (*mpg*).

This example uses the stream named *car_sales_knn.str*, available in the *Demos* folder under the *streams* subfolder. The data file is *car_sales_knn_mod.sav*. For more information, see the topic Demos Folder in Chapter 1 on p. 6.

Creating the Stream



Create a new stream and add a Statistics File source node pointing to *car_sales_knn_mod.sav* in the *Demos* folder of your IBM® SPSS® Modeler installation.

First, let's see what data the manufacturer has collected.

- Attach a Table node to the Statistics File source node.
- ▶ Open the Table node and click Run.

Assessing New Vehicle Offerings (KNN)

🥪 File	📄 Edit	🕙 Ger	nerate		<u></u>	14				0
Table 🛛	nnotations							_	_	
	manufact	model	sales	resale	type	price	engine_s	horsepow	wheelbas	width
140	Toyota	Celica	33.269	15.445	0.0	16	1.800	140.000	102.400	68.3
141	Toyota	Tacoma	84.087	9.575	1.0	11	2.400	142.000	103.300	66.5
142	Toyota	Sienna	65.119	\$null\$	1.0	22	3.000	194.000	114.200	73.4
143	Toyota	RAV4	25.106	13.325	1.0	16	2.000	127.000	94.900	66.7
144	Toyota	4Run	68.411	19.425	1.0	22	2.700	150.000	105.300	66.5
145	Toyota	Land	9.835	34.080	1.0	51	4.700	230.000	112.200	76.4
146	Volksw	Golf	9.761	11.425	0.0	14	2.000	115.000	98.900	68.3
147	Volksw	Jetta	83.721	13.240	0.0	16	2.000	115.000	98.900	68.3
148	Volksw	Passat	51.102	16.725	0.0	21	1.800	150.000	106.400	68.5
149	Volksw	Cabrio	9.569	16.575	0.0	19	2.000	115.000	97.400	66.7
150	Volksw	GTI	5.596	13.760	0.0	17	2.000	115.000	98.900	68.3
151	Volksw	Beetle	49.463	\$null\$	0.0	15	2.000	115.000	98.900	67.9
152	Volvo	S40	16.957	\$null\$	0.0	23	1.900	160.000	100.500	67.6
153	Volvo	V40	3.545	\$null\$	0.0	24	1.900	160.000	100.500	67.6
154	Volvo	S70	15.245	\$null\$	0.0	27	2.400	168.000	104.900	69.3
155	Volvo	V70	17.531	\$null\$	0.0	28	2.400	168.000	104.900	69.3
156	Volvo	C70	3.493	\$null\$	0.0	45	2.300	236.000	104.900	71.5
157	Volvo	S80	18.969	\$null\$				201.000	109.900	72.1
158		newC	\$null\$	\$null\$	\$n	21	1.500	76.000	106.300	67.9
159		newT	\$null\$	\$null\$	\$n	34	3.500	167.000	109.800	75.2

Figure 28-3

The details for the two prototypes, named *newCar* and *newTruck*, have been added at the end of the file.

We can see from the source data that the manufacturer is using the classification of "truck" (value of 1 in the *type* column) rather loosely to mean any non-automobile type of vehicle.

The last column, *partition*, is necessary in order that the two prototypes can be designated as holdouts when we come to identify their nearest neighbors. In this way, their data will not influence the calculations, as it is the rest of the market that we want to consider. Setting the *partition* value of the two holdout records to 1, while all the other records have a 0 in this field, enables us to use this field later when we come to set the focal records—the records for which we want to calculate the nearest neighbors.

Leave the table output window open for now, as we'll be referring to it later.

Figure 28-4

	review				0-						
Types Format Image: State	Annotations	ilues Clea	r Values	Clear All V	alues						
Field 💳	Measurement	Values	Missing	Check	Role						
v norsepow	Continuous	[33.0,430 [92.6,138		None	nput						
width	Continuous	· ·	[62.6,79.9]		Input						
length	Continuous	[149.4,22		None	Input						
curb_wqt	Continuous	[1.895,5		None	Input						
fuel cap	Continuous	[10.3,32.0]			Input						
mpg	Continuous	[15.0,46.0]		None	🔪 Input						
linsales	🔗 Continuous	[-2.20727		None	S None						
partition	🎖 Flag	1.0/0.0		None	🔪 Input						
Partition Flag 1.0/0.0 None Input View current fields View unused field settings											

- ► Add a Type node to the stream.
- Attach the Type node to the Statistics File source node.
- ► Open the Type node.

We want to make the comparison only on the fields *price* through *mpg*, so we'll leave the role for all these fields set to Input.

- ▶ Set the role for all the other fields (*manufact* through *type*, plus *lnsales*) to None.
- Set the measurement level for the last field, *partition*, to Flag. Make sure that its role is set to Input.
- Click Read Values to read the data values into the stream.
- ► Click OK.

Assessing New Vehicle Offerings (KNN)

Figure 28-5

Choosing to identify the nearest neighbors

🔽 No Targets 🛛 🔀
Objectives Fields Settings Annotations
The kNN procedure will identify the most similiar training cases (the nearest neighbors) to your cases
of interest. A target field can be predicted based on the neighboring values.
What type of analysis do you want to perform?
O Predict a target field
Only identify the nearest neighbors
What is your objective?
Balance speed and accuracy
Automatically selects the best number of neighbors within a small range.
Speed
Finds a fixed number of neighbors.
O Accuracy
Automatically selects the best number of neighbors within a larger range and uses
variable importance when calculating distances.
◎ Custom analysis
Choose this option to fine tune the algorithm on the Settings tab.
OK Run Cancel <u>Apply</u> <u>R</u> eset

- ► Attach a KNN node to the Type node.
- ► Open the KNN node.

We're not going to be predicting a target field this time, because we just want to find the nearest neighbors for our two prototypes.

- On the Objectives tab, choose Only identify the nearest neighbors.
- ► Click the Settings tab.

Figure 28-6

Using the partition field to identify the focal records

😡 No Target	ts				\mathbf{X}
					0
Objectives F	Fields	Settings A	Annotations		
Settings					
Model		Model name	x:	🔘 Auto 🔘 Custom	
Neighbors		V Use par	titioned data	1	
Feature Selecti	ion	🟹 Build mo	del for each	n split	
Cross-Validatio	n	To select fi	ields manual	lly, choose "Use custom setting	gs" on the Fields tab
Analyze		Partition:			_]
		Splits:			-
					×
		📝 Normaliza	e range inpu	ıts	
		🔲 Use case	e labels		-
		🚺 Identify f	ocal record	🎖 partition	~]
ок 🕨	Run	Cancel			Apply Reset

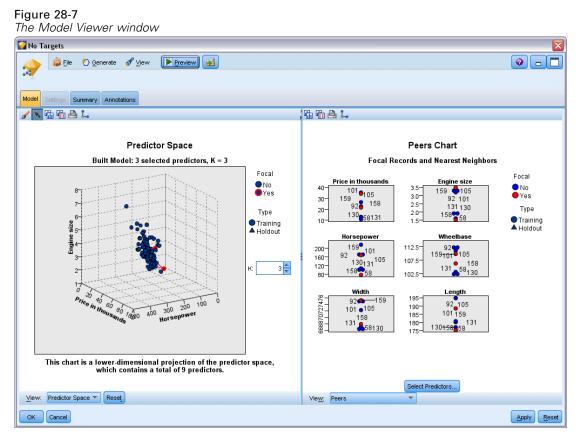
Now we can use the *partition* field to identify the focal records—the records for which we want to identify the nearest neighbors. By using a flag field, we ensure that records where the value of this field is set to 1 become our focal records.

As we've seen, the only records that have a value of 1 in this field are *newCar* and *newTruck*, so these will be our focal records.

- On the Model panel of the Settings tab, select the Identify focal record check box.
- ▶ From the drop-down list for this field, choose partition.
- ► Click the Run button.

Assessing New Vehicle Offerings (KNN)

Examining the Output



A model nugget has been created on the stream canvas and in the Models palette. Open either of the nuggets to see the Model Viewer display, which has a two-panel window:

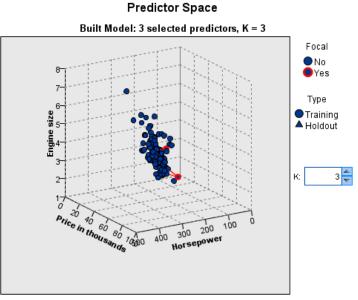
- The first panel displays an overview of the model called the main view. The main view for the Nearest Neighbor model is known as the predictor space.
- The second panel displays one of two types of views:

An auxiliary model view shows more information about the model, but is not focused on the model itself.

A linked view is a view that shows details about one feature of the model when you drill down on part of the main view.

Predictor Space

Figure 28-8 Predictor space chart



This chart is a lower-dimensional projection of the predictor space, which contains a total of 9 predictors.

The predictor space chart is an interactive 3-D graph that plots data points for three features (actually the first three input fields of the source data), representing price, engine size and horsepower.

Our two focal records are highlighted in red, with lines connecting them to their k nearest neighbors.

By clicking and dragging the chart, you can rotate it to get a better view of the distribution of points in the predictor space. Click the Reset button to return it to the default view.

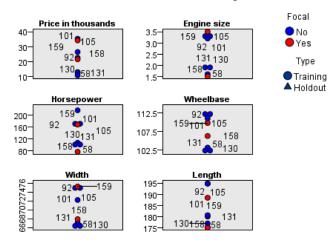
Assessing New Vehicle Offerings (KNN)

Peers Chart

Figure 28-9 Peers chart

Peers Chart

Focal Records and Nearest Neighbors



The default auxiliary view is the peers chart, which highlights the two focal records selected in the predictor space and their k nearest neighbors on each of six features—the first six input fields of the source data.

The vehicles are represented by their record numbers in the source data. This is where we need the output from the Table node to help identify them.

If the Table node output is still available:

- Click the Outputs tab of the manager pane at the top right of the main IBM® SPSS® Modeler window.
- ▶ Double-click the entry Table (16 fields, 159 records).

If the table output is no longer available:

- ▶ On the main SPSS Modeler window, open the Table node.
- ► Click Run.

Figure 28-10 Identifying records by record number

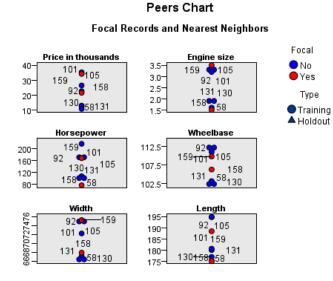
📦 File	(16 field <u> </u> <u> </u>	🕙 Ger				14	44				×
Sec. 1	- Eour	0 20	101 010			191					
Table 🛛	Annotations										
	manufact	model	sales	resale	type	price	engine_s	horsepow	wheelbas	width	
140	Toyota	Celica	33.269	15.445	0.0	16	1.800	140.000	102.400	68.3	- 4
141	Toyota	Tacoma	84.087	9.575	1.0	11	2.400	142.000	103.300	66.5	ſ
142	Toyota	Sienna	65.119	\$null\$	1.0	22	3.000	194.000	114.200	73.4	÷
143	Toyota	RAV4	25.106	13.325	1.0	16	2.000	127.000	94.900	66.7	·
144	Toyota	4Run	68.411	19.425	1.0	22	2.700	150.000	105.300	66.5	·
145	Toyota	Land	9.835	34.080	1.0	51	4.700	230.000	112.200	76.4	·
146	Volksw	Golf	9.761	11.425	0.0	14	2.000	115.000	98.900	68.3	·
147	Volksw	Jetta	83.721	13.240	0.0	16	2.000	115.000	98.900	68.3	÷
148	Volksw	Passat	51.102	16.725	0.0	21	1.800	150.000	106.400	68.5	•
149	Volksw	Cabrio	9.569	16.575	0.0	19	2.000	115.000	97.400	66.7	•
150	Volksw	GTI	5.596	13.760	0.0	17	2.000	115.000	98.900	68.3	•
151	Volksw	Beetle	49.463	\$null\$	0.0	15	2.000	115.000	98.900	67.9	1
152	Volvo	S40	16.957	\$null\$	0.0	23	1.900	160.000	100.500	67.6	1
153	Volvo	V40	3.545	\$null\$	0.0	24	1.900	160.000	100.500	67.6	
154	Volvo			\$null\$	0.0		2.400	168.000	104.900	69.3	•
155	Volvo			\$null\$	0.0		2.400	168.000	104.900	69.3	1
156	Volvo			\$null\$	0.0		2.300	236.000	104.900	71.5	1
157	Volvo			\$null\$	0.0		2.900	201.000	109.900	72.1	
158		newC	\$null\$	\$null\$	\$n		1.500	76.000	106.300	67.9	1
159		newT	\$null\$	\$null\$	\$n	34	3.500	167.000	109.800	75.2	1
	4						1112			•	
										G	OK

Scrolling down to the bottom of the table, we can see that newCar and newTruck are the last two records in the data, numbers 158 and 159 respectively.

Assessing New Vehicle Offerings (KNN)

Figure 28-11

Comparing features on the peers chart



From this we can see on the peers chart, for example, that *newTruck* (159) has a bigger engine size than any of its nearest neighbors, while *newCar* (158) has a smaller engine than any of *its* nearest neighbors.

For each of the six features, you can move the mouse over the individual dots to see the actual value of each feature for that particular case.

But which vehicles are the nearest neighbors for newCar and newTruck?

The peers chart is a little bit crowded, so let's change to a simpler view.

- Click the View drop-down list at the bottom of the peers chart (the entry that currently says Peers).
- ► Select Neighbor and Distance Table.

Neighbor and Distance Table

Figure 28-12 Neighbor and distance table

k Nearest Neighbors and Distances

Displayed for Initial Focal Records											
Freed Derend	Nearest Neighbors Nearest D										
Focal Record	1	2	3	1	2						
158	131	130	58	0.979	0.990						
159	105	92	101	0.580	0.634						

That's better. Now we can see the three models to which each of our two prototypes are closest in the market.

For *newCar* (focal record 158) they are the Saturn SC (131), the Saturn SL (130), and the Honda Civic (58).

No great surprises there—all three are medium-size saloon cars, so *newCar* should fit in well, particularly with its excellent fuel efficiency.

For *newTruck* (focal record 159), the nearest neighbors are the Nissan Quest (105), the Mercury Villager (92), and the Mercedes M-Class (101).

As we saw earlier, these are not necessarily trucks in the traditional sense, but simply vehicles that are classed as not being automobiles. Looking at the Table node output for its nearest neighbors, we can see that *newTruck* is relatively expensive, as well as being one of the heaviest of its type. However, fuel efficiency is again better than its closest rivals, so this should count in its favor.

Summary

We've seen how you can use nearest-neighbor analysis to compare a wide-ranging set of features in cases from a particular data set. We've also calculated, for two very different holdout records, the cases that most closely resemble those holdouts.

Notices

This information was developed for products and services offered worldwide.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing, Legal and Intellectual Property Law, IBM Japan Ltd., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Japan.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group, Attention: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

© Copyright IBM Corporation 1994, 2012.

Appendix A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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

Trademarks

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

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

387 Notices



Bibliography

Asuncion, A., and D. Newman. 2007. "UCI Machine Learning Repository." Available at http://mlearn.ics.uci.edu/MLRepository.html.

Index

adding IBM SPSS Modeler Server connections, 10, 12 analysis node, 101 application examples, 4

canvas, 14 categorical variable codings in Cox regression, 334 censored cases in Cox regression, 333 classes, 16 classification table in Discriminant Analysis, 271 CLEM introduction, 23 command line starting IBM SPSS Modeler, 9 condition monitoring, 254 connections server cluster, 12 to IBM SPSS Modeler Server, 9-10, 12 Coordinator of Processes, 12 COP, 12 copy, 17 covariate means in Cox regression, 338 Cox regression categorical variable codings, 334 censored cases, 333 hazard curve, 340 survival curve, 339 variable selection, 335 CRISP-DM, 16 cut, 17

data

manipulation, 94 modeling, 97, 99, 101 reading, 84 viewing, 88 Decision List models application example, 116 connecting with Excel, 134 custom measures using Excel, 134 generating, 143 modifying the Excel template, 140 saving session information, 143 Decision List node application example, 116 Decision List viewer, 121 derive node, 94 **Discriminant Analysis** classification table, 271 eigenvalues, 268 stepwise methods, 267 structure matrix, 269

territorial map, 270 Wilks' lambda, 268 documentation, 4 domain name (Windows) IBM SPSS Modeler Server, 9 down search Decision List models, 127 eigenvalues in Discriminant Analysis, 268 examples Applications Guide, 4 Bayesian network, 227, 237 catalog sales, 199 cell sample classification, 317 condition monitoring, 254 discriminant analysis, 260 input string length reduction, 110 KNN, 373 market basket analysis, 366 multinomial logistic regression, 144, 154 new vehicle offering assessment, 373 overview, 5 Reclassify node, 110 retail analysis, 249 string length reduction, 110 SVM, 317 telecommunications, 144, 154, 168, 190, 260 Excel connecting with Decision List models, 134 modifying Decision List templates, 140 expression builder, 94

Feature Selection models, 103 Feature Selection node importance, 103 ranking predictors, 103 screening predictors, 103 fields ranking importance, 103 screening, 103 selecting for analysis, 103 filtering, 97

gamma regression in Generalized Linear Models, 312 Generalized Linear Models goodness of fit, 305, 310 omnibus test, 305 parameter estimates, 280, 293, 306, 316 Poisson regression, 300 tests of model effects, 278, 291, 306 generated models palette, 15 goodness of fit in Generalized Linear Models, 305, 310

© Copyright IBM Corporation 1994, 2012.

390 Index

graph nodes, 93 grouped survival data in Generalized Linear Models, 272 hazard curves in Cox regression, 340 host name IBM SPSS Modeler Server, 9-10 hot keys, 22 IBM SPSS Modeler, 1, 13 documentation, 4 getting started, 8 overview, 8 running from command line, 9 IBM SPSS Modeler Server domain name (Windows), 9 host name, 9-10 password, 9 port number, 9-10 user ID, 9 icons setting options, 20 importance ranking predictors, 103 Interactive List viewer application example, 121 Preview pane, 121 working with, 121 interval-censored survival data in Generalized Linear Models, 272 introduction IBM SPSS Modeler, 8 legal notices, 385 logging in to IBM SPSS Modeler Server, 9 low probability search Decision List models, 127 main window, 14 managers, 15 market basket analysis, 366 Microsoft Excel connecting with Decision List models, 134 modifying Decision List templates, 140 middle mouse button simulating, 22 minimizing, 19 mining tasks Decision List models, 121 modeling, 97, 99, 101 mouse

using in IBM SPSS Modeler, 22

multiple IBM SPSS Modeler sessions, 13

negative binomial regression in Generalized Linear Models, 307 nodes, 8 nuggets defined, 16 omnibus test in Generalized Linear Models, 305 omnibus tests in Cox regression, 335 output, 15 palettes, 14 parameter estimates in Generalized Linear Models, 280, 293, 306, 316 password IBM SPSS Modeler Server, 9 paste, 17 Poisson regression in Generalized Linear Models, 300 port number IBM SPSS Modeler Server, 9-10 predictors ranking importance, 103 screening, 103 selecting for analysis, 103 preparing, 94 printing, 23 streams, 20 projects, 16 ranking predictors, 103 remainder Decision List models, 121 resizing, 19 retail analysis, 249 scaling streams to view, 20 screening predictors, 103 scripting, 23 searching COP for connections, 12 segments Decision List models, 121 excluding from scoring, 130 Self-Learning Response Model node application example, 215 browsing the model, 222 building the stream, 216 stream building example, 216 server adding connections, 10 logging in, 9 searching COP for servers, 12 shortcuts keyboard, 22

391 Index

single sign-on, 10 SLRM node application example, 215 browsing the model, 222 building the stream, 216 stream building example, 216 source nodes, 84 SPSS Modeler Server, 2 stepwise methods in Cox regression, 335 in Discriminant Analysis, 267 stop execution, 17 stream, 14 streams, 8 building, 84 scaling to view, 20 structure matrix in Discriminant Analysis, 269 survival curves in Cox regression, 339 table node, 88 temp directory, 12 territorial map in Discriminant Analysis, 270 tests of model effects in Generalized Linear Models, 278, 291, 306 toolbar, 17 trademarks, 386 undo, 17 user ID IBM SPSS Modeler Server, 9 var. file node, 84 visual programming, 13 web node, 93 Wilks' lambda in Discriminant Analysis, 268

zooming, 17