SAS® Factory Miner 14.1

User's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2015. *SAS® Factory Miner 14.1: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Factory Miner 14.1: User's Guide

Copyright © 2015, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a) and DFAR 227.7202-4 and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

July 2015

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

Contents

About This Book

Audience

This book is designed for users of SAS Factory Miner. It contains information about creating projects, configuring data sources, choosing model templates, running analyses, viewing and investigating results, and other topics that are designed to help a user perform tasks in SAS Factory Miner.

Chapter 1 Introduction to SAS Factory Miner

Overview Topics for SAS Factory Miner

This section contains the following introductory information about SAS Factory Miner:

- "What Is SAS Factory Miner?" on page 2
- "Benefits of SAS Factory Miner" on page 2
- "Accessing SAS Factory Miner" on page 2
- "Modifying Settings" on page 3
- "Viewing Notifications" on page 4
- "Accessing Help" on page 4
- "Accessibility Information" on page 5
- "Exit SAS Factory Miner" on page 5

See the following chapters for more information about SAS Factory Miner:

- Chapter 2, "The SAS Factory Miner User Interface," on page 7
- Chapter 3, "Building Models with SAS Factory Miner," on page 21
- Chapter 4, "SAS Factory Miner Models," on page 37
- Chapter 5, "Getting Started with SAS Factory Miner," on page 63

What Is SAS Factory Miner?

SAS Factory Miner is an easy-to-use application that enables you to build models to analyze data and examine results. You can build a model by creating a project, configuring a data source, choosing model templates, running your analysis, and viewing results.

You can use the following models to analyze data in SAS Factory Miner:

- Bayesian Network Model
- Decision Tree Model
- Generalized Linear Model
- Gradient Boosting Model
- Neural Network Model
- Random Forest Model
- Regression Model
- Support Vector Machine Model

For more information about the types of models that you can create using SAS Factory Miner, see Chapter 4, "SAS Factory Miner Models," on page 37.

For more information about SAS Factory Miner, click the Help icon, and select **About** from the menu that appears. For more information about accessing help for SAS Factory Miner, see "Accessing Help" on page 4.

Benefits of SAS Factory Miner

SAS Factory Miner helps you perform data mining model creation at a segment level. For example, you can use customer data to investigate which customers are most likely to respond to online offers. Furthermore, you can build separate models for each of the regions in which your clients live. You can run multiple models and examine results to determine which modeling algorithm was most effective for the data that you have and the goals of your analysis. Comparing modeling results can help you make more informed and effective business decisions.

For an example of how you can perform data mining model creation at the segment level and examine results, see Chapter 5, "Getting Started with SAS Factory Miner," on page 63.

Accessing SAS Factory Miner

There are two ways to log on to the SAS Factory Miner application. You can log on directly to the SAS Factory Miner application via a browser URL. You can also access SAS Factory Miner via your site's SAS Home page. SAS Home is a local hub that integrates access to your site's SAS solutions.

To log on to the SAS Factory Miner application directly:

1. In the address bar of your web browser, enter the URL for the SAS Factory Miner application, and press **Enter**. For example, this URL typically takes the following form:

   ```
   http(s)://hostname:port/SASFactoryMiner
   ```

   or

   ```
   http(s)://hostname.com/SASFactoryMiner
   ```

   Note: It's a good idea for regular users to bookmark the SAS Factory Miner logon URL in browsers for ease of use. Contact your system administrator if you do not have the URL for your local installation of the SAS Factory Miner solution.

2. Provide your user ID.
3. Provide your password.
4. Click **SIGN IN**.
5. The SAS Factory Miner application appears and opens to the **Projects** workspace by default.

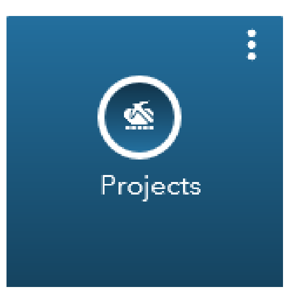
You can also use your site's SAS Home page to access your SAS Factory Miner solution. To log on to SAS Factory Miner via the SAS Home page:

1. In the address bar of your web browser, enter the URL for your site's SAS Home page, and press **Enter**. For example, this URL typically takes the following form:

   ```
   http(s)://servername.com/SASVisualAnalyticsHub
   ```

   Note: Contact your system administrator if you do not have the URL for your installation's SAS Home location.

2. Provide your user ID.
3. Provide your password.
4. Click **SIGN IN**.
5. On the SAS Home page, locate the button labeled **Projects**. Click the **Projects** button to open SAS Factory Miner.

6. The SAS Factory Miner application appears and opens to the **Projects** workspace by default.

Modifying Settings

Perform the following steps to modify SAS Factory Miner settings:

1. Click the Application options icon .
2. Click **Settings** on the menu that appears.

   The Settings window appears.
3. Modify any settings that you want to change:
   a. Select **Global** ⇨ **General** to modify user locale or theme options.
   b. Select **Global** ⇨ **Side Menu** to display workspace items or to reorder them.
   c. Select **Global** ⇨ **Accessibility** to specify accessibility options.
   d. Select **SAS Factory Miner** ⇨ **Sample Data** to specify whether you want to sample training data. You can specify the sampling method, and whether to use a percentage or number of observations.
   e. Select **SAS Factory Miner** ⇨ **Partition Data** to specify whether you want to partition the data, and, if so, what percentage should be used for training and validation.
   f. Select **SAS Factory Miner** ⇨ **Rules** to specify the minimum number of observations, the minimum event rate, the interval champion criterion, and the nominal champion criterion. The minimum number of observations and the minimum event rate affect which segments will be modeled. The champion criterion determines how the best model will be determined for each segment.
4. Click **Done** when you finish modifying settings.

Viewing Notifications

You can access SAS Factory Miner notifications by clicking the Unread notifications icon .

Accessing Help

Perform the following steps to access SAS Factory Miner help:

1. Click the Help icon .
2. Click **Help Center** in the menu that appears.

   SAS Factory Miner Help appears in a new tab.
3. Expand **SAS Factory Miner: Help and Documentation**.

   Note: If the top of the **Help** menu is not visible, you might need to click the Show table of contents icon to expand it.
4. Select a topic that you want to know more about.

   Note: You might need to drill down into a topic.

Information about the selected topic appears.

5. (Optional) Search for a Help topic by clicking the Show search pane icon , and entering text in the text box that appears. After you enter text to search for in the text box, click the Start search icon in the text box. Topics that match your search terms appear below the search text box. To navigate to one of the Help topics that match your search terms, just click it in the list. To clear your search terms from the search text box, click the Clear search text icon .

   Note: For help with searching for a Help topic, click the Help icon , and select **Search Tips** from the menu that appears.

Accessibility Information

SAS Factory Miner includes accessibility and compatibility features that improve the usability of the product for users with disabilities. These features are related to accessibility standards for electronic information technology that were adopted by the U.S. Government under Section 508 of the U.S. Rehabilitation Act of 1973, as amended.

SAS Factory Miner includes keyboard shortcuts. For example, you can use the **Tab** key to navigate from one component to another.

The following is a list of accessibility limitations and workaround steps in SAS Factory Miner:

- If you have trouble accessing all options from the keyboard with the **Tab** key, try using reverse tab (**Shift** + **Tab**) to access other options.
- The JAWS screen reader does not function with Firefox or Chrome browsers.
- Graphs and other visualizations of data might not be read correctly by screen readers.
- Screen readers might not read labels associated with some fields or calendars.

For specific information about accessibility features, refer to your operating system's help. If you have questions or concerns about the accessibility of SAS products, send an email to accessibility@sas.com.

Exit SAS Factory Miner

To exit SAS Factory Miner:

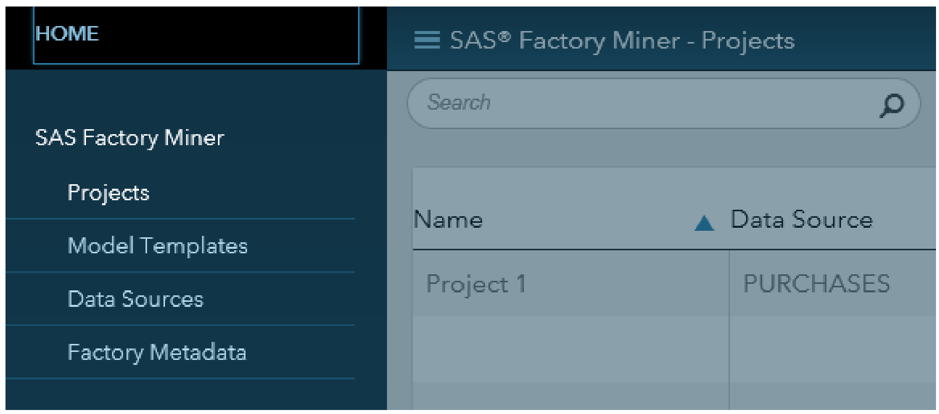
1. Click the Application options icon .
2. Click **Sign Out** on the menu that appears.

Chapter 2

The SAS Factory Miner User Interface

Overview of the SAS Factory Miner User Interface

You can use the SAS Factory Miner application to manage projects, model templates, data sources, and factory metadata. Click the Side menu icon to access a menu that enables you to navigate to the **Projects**, **Model Templates**, **Data Sources**, or **Factory Metadata** workspace.



For information about how you can use these workspaces, see the following:

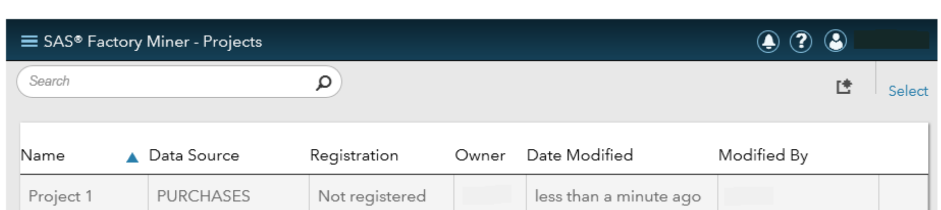
- "Managing Projects" on page 8
- "Managing Model Templates" on page 12
- "Managing Data Sources" on page 16
- "Managing Factory Metadata" on page 18

Managing Projects

Overview of Managing Projects

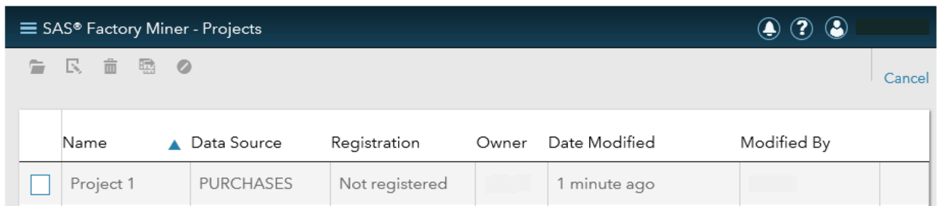
In SAS Factory Miner, a project is a user-created entity that contains the components required to create data mining models. When you create a project, you must specify the name, location, data source, and whether the SAS Factory Miner application should perform the partitioning for you. The variables in your data source are checked against the factory metadata, and any applicable definitions are applied.

When you first open SAS Factory Miner, the **Projects** workspace appears.



The **Projects** workspace lists the projects created by every user. The workspace indicates the data source, owner, when it was most recently modified, who made the most recent modification, and whether the project is locked. In the **Projects** workspace, you can open an existing project, create a new project, delete one or more projects, rename a project, register a project to the SAS model repository, cancel a running registration process, and search for a project.

You can click **Select** in the **Projects** workspace to select a project, and choose an option from the toolbar.



To navigate back to the **Projects** workspace from another workspace, click the Side menu icon , and select **Projects** from the menu that appears.

Note: You can use the Side menu icon to navigate to another workspace in SAS Factory Miner, or another installed product, such as SAS Visual Analytics.

Create a Project

To create a project in the **Projects** workspace:

1. Click the Create a new Factory Miner project icon in the main menu.

   The New Project window appears.

2. Enter a name for your project.
3. Specify a location for your project.

   Note: If a location has already been specified, you can use the default path. Alternatively, you can click **Browse** to navigate to the path that you want to select.

4. Select a data source from the menu.

   If you want to use a data source that is not in the menu, click **New data source** and specify a data source in the window that appears.

   Note: When there are variables with a role of Segment or Target that have factory metadata definitions, those variables and their roles are listed in the New Project window. If you want to alter the assignments made by the factory metadata, you can do so after the project is created.

5. Specify whether you want to partition data.

   Note: If a project partitions the data, the data is split into training and validation data sets. The default setting when creating a new project is to partition the data.

6. Click **Save**.

   Your project appears and opens on the **Data** tab. A project has four primary views:

   - **Data** — displays variables in the input data source and some descriptive statistics for each variable.

- **Profile** — displays information about the segments.
- **Model Templates** — displays the model templates that are used by this project.
- **Results** — displays the model building progress and results.

For information about how to use and navigate each of these tabs, see Chapter 3, "Building Models with SAS Factory Miner," on page 21.

Warnings might appear on the **Data** tab to notify you to select a segment variable and a target variable. For more information about how to configure a data source, see "Data Tab" on page 22.

Open a Project

To open a project in the **Projects** workspace, position the mouse pointer over the project that you want to open in the table and select that row.

Note: You can also open a project by clicking **Select**, clicking the check box next to the project that you want to open, and then clicking the Open a project icon .

To navigate back to the **Projects** workspace after you have opened a project, click the View all items icon on the **Data** tab.

Rename a Project

To rename a project in the **Projects** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each project in the table.
2. Click the check box for the project that you want to rename.
3. Click the Rename a project icon in the main menu.

   The Rename window appears.
4. Provide the new name for your project.
5. Click **OK**.

Delete a Project

To delete a project in the **Projects** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each project in the table.
2. Click the check box for the projects that you want to delete.
3. Click the Delete a project icon in the main menu.

   A confirmation window appears.
4. Click **Yes**.

Register a Project

To register a project in the **Projects** workspace:

Note: You can only register a project that has been run and has models. Registering a project requires a SAS Model Manager license.

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each project in the table.

2. Click the check box for the project that you want to register.
3. Click the Register project to SAS model repository icon .

   The Register project to SAS model repository window appears.

4. Specify whether you want to register all models or only champion models.
5. Click **Register**.

Cancel a Running Registration Process

To cancel a running registration process in the **Projects** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each project in the table.

2. Click the check box for the project whose running registration you want to cancel.
3. Click the Cancel running registration process icon .

Search for a Project

You can search for a project using the **Search** text box:

After you enter your text, click the Start search icon . The table will update with projects that match your search terms. When you want to clear your search from the **Search** text box, and return to the default view of projects, click the Clear search text icon .

Managing Model Templates

Overview of the Standard Model Templates

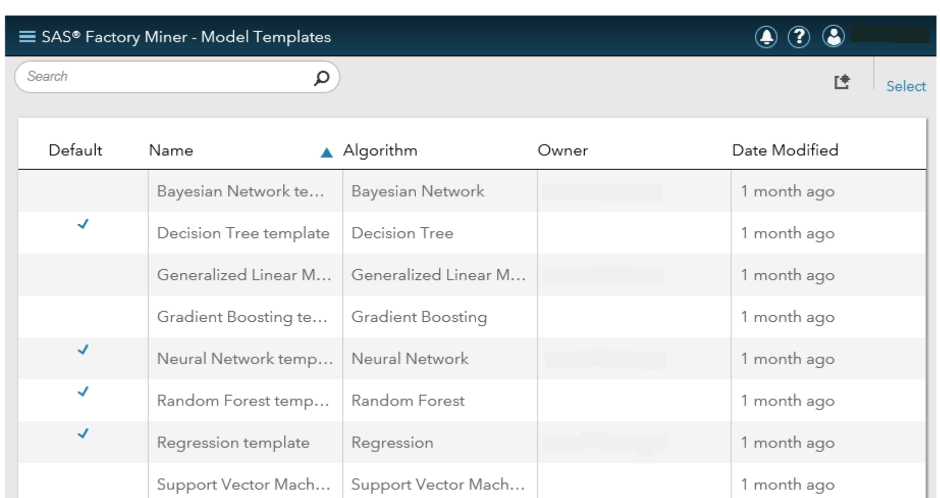
A SAS Factory Miner model template is a collection of data manipulation rules and model properties that are applied to a data segment in order to build a model. By default, SAS Factory Miner includes a model template for each available predictive model. These model templates are included so that you can begin modeling immediately, without needing to create your own templates. However, the standard model templates might not meet your specific business needs.

Note: See “Create a Custom Model Template” on page 13 for information about creating custom model templates.

The following are the standard model templates in SAS Factory Miner:

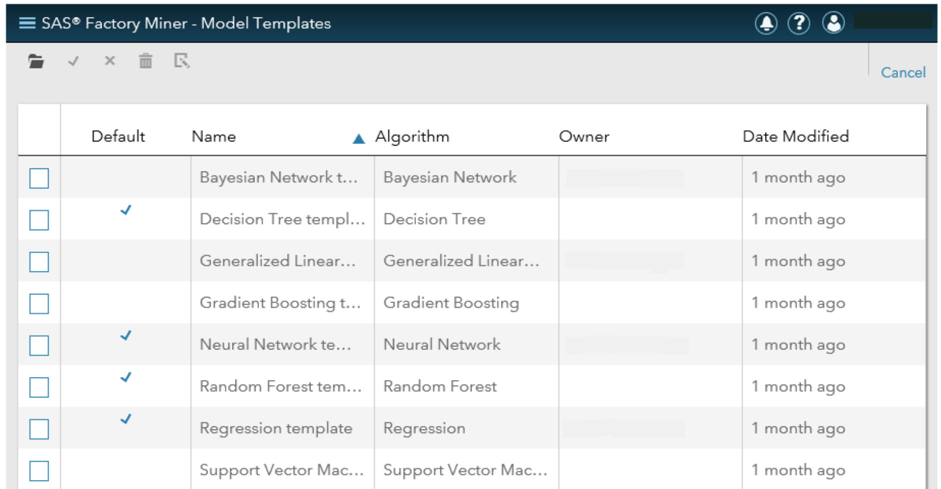
- Bayesian Network template
- Decision Tree template
- Generalized Linear Model template
- Gradient Boosting template
- Neural Network template
- Random Forest template
- Regression template
- Support Vector Machine template

To navigate to the **Model Templates** workspace, click the Side menu icon , and select **Model Templates** from the menu that appears.



Default	Name	Algorithm	Owner	Date Modified
	Bayesian Network te...	Bayesian Network		1 month ago
✓	Decision Tree template	Decision Tree		1 month ago
	Generalized Linear M...	Generalized Linear M...		1 month ago
	Gradient Boosting te...	Gradient Boosting		1 month ago
✓	Neural Network temp...	Neural Network		1 month ago
✓	Random Forest temp...	Random Forest		1 month ago
✓	Regression template	Regression		1 month ago
	Support Vector Mach...	Support Vector Mach...		1 month ago

You can click **Select** in the **Model Templates** workspace to select a model template, and choose an option from the toolbar.



For more information about the standard model templates, see Chapter 4, “SAS Factory Miner Models,” on page 37.

Create a Custom Model Template

If you have a specific set of modeling requirements or a trusted model that you want to use frequently, you can create your own model template using the **Model Templates** workspace.

1. Click the Create a new model template icon  in the main menu of the **Model Templates** workspace.

   The New Model Template window appears.

2. Provide a name for the model template.

3. Select a model algorithm from the menu.

4. Specify whether the model template should be marked as default. This determines whether the new template will be used automatically when you create new projects.

5. Click **Save**.

   A window appears with property options depending on the algorithm that you selected.

6. Specify the values that you want to use for the various property options.

   a. Specify which components you want to use.

      i. Select a component that you want to use in the **Ignore** list.

      ii. Drag the component to the **Drag components here** text area in the **Use** list.

      Note: If a component has already been dragged to the **Use** list, the **Drag components here** text area might not be visible. If the **Drag components here** text area is not visible, then drag the component to where another component is toward the top of the **Use** list. Components will be applied to a model in the order in which they are displayed in the **Use** list.

      The component is added to the **Use** list.

   b. Specify values for each component.

      i. Select the component whose values you want to modify in the **Use** list.

The current specifications for that property appear in the window.

ii. Select, specify, or provide values that you want to use for that component in the window.

For property information for each model, see Chapter 4, “SAS Factory Miner Models,” on page 37.

iii. Repeat the previous two steps for each component whose values you want to modify.

7. Click **Save**.

   The new model template is added to the table of model templates in the **Model Templates** workspace.

Note: You can also create a new model template from an algorithm or from a global template on the **Model Templates** tab of an opened project. For more information about how to create model templates on the **Model Templates** tab, see “Create a New Model Template from an Algorithm” on page 26, and “Create a New Model Template from a Global Template” on page 27.

Open a Model Template

To open a model template in the **Model Templates** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each model template in the table.

2. Click the check box for the model template that you want to open.

3. Click the Open model template icon in the main menu.

   The model template opens in a window.

Note: Alternatively, you can open a model template by clicking the name of the template in the table of model templates when the column of check boxes is not visible for each template.

Mark a Model Template as Default

To mark a model template as the default in the **Model Templates** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each model template in the table.

2. Click the check box for the model template that you want to mark as the default.

3. Click the Mark as default icon in the main menu.

   The model template is marked as the default.

Unmark a Model Template as Default

To unmark a model template as the default in the **Model Templates** workspace:

1. Click **Select** in the main menu.

A selection box appears in the left-most column for each model template in the table.

2. Click the check box for the model template that you want to unmark as the default.
3. Click the Unmark as default icon in the main menu.

   The model template is unmarked as the default.

Delete a Model Template

To delete a model template in the **Model Templates** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each model template in the table.
2. Click the check box for the model template that you want to delete.
3. Click the Delete icon in the main menu.

Rename a Model Template

To rename a model template in the **Model Templates** workspace:

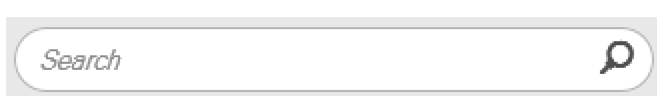
1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each model template in the table.
2. Select the check box for the model template that you want to rename.
3. Click the Rename icon in the main menu.

   The Rename window appears.
4. Enter a new name.
5. Click **OK**.

Search for a Model Template

You can search for a model template using the **Search** text box:



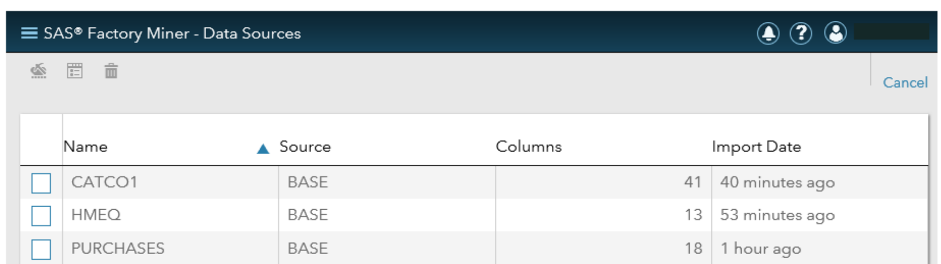
After you enter your text, click the Start search icon . The table will update with model templates that match your search terms. When you want to clear your search from the **Search** text box, and return to the default view of model templates, click the Clear search text icon .

Managing Data Sources

Overview of Managing Data Sources

In SAS Factory Miner, a data source is a logically defined collection of information for use with your models. To navigate to the **Data Sources** workspace, click the Side menu icon, and select **Data Sources** from the menu that appears.

You can click **Select** in the **Data Sources** workspace to select a data source, and choose an option from the toolbar.



Create a Data Source

To create a new data source in the **Data Sources** workspace:

1. Click the Add a new data source icon .

   A window appears.

2. Navigate to the table that you want to use as a data source, and select it.

3. Click **Open**.

Create a New Project with a Data Source

To create a new project with a data source in the **Data Sources** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each data source in the table.

2. Click the check box for a data source that you want to create a new project with.

3. Click the Create a new Factory Miner project icon  .

   The New Project window appears with the selected data source specified.

4. Provide the name for your new project.

5. Specify a location for your project.

   Note: If a location has already been specified, you can use the default path. Alternatively, you can click **Browse** to navigate to the path that you want to select.

6. Specify whether you want to partition data.

7. Click **Save**.

   Your project appears and opens on the **Data** tab.

   For more information about how to configure a project using the **Data**, **Profile**, **Model Templates**, and **Results** tabs, see Chapter 3, “Building Models with SAS Factory Miner,” on page 21.

Define a Data Source

To define a data source in the **Data Sources** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each data source in the table.

2. Click the check box for a data source that you want to define.

3. Click the Define factory metadata icon  .

   The Define Data window appears.

4. Select the check box for a variable that you want to assign a role for.

5. Click a role option. Available role options include the following:

   - Frequency
   - ID
   - Input
   - Key
   - Rejected
   - Segment
   - Target

6. Repeat the previous two steps for all variables that you want to assign a role for.

   Note: You can remove a role that you assigned for a variable by selecting the check box for the variable, and then clicking the Reset icon 

.

7. Click **Save** when you have finished assigning roles for the data source.

Note: The roles that you select for a data source will be selected for all new projects that are based on that data source. This means that when you create a project using that data source, you do not have to define roles on the **Data** tab after that project is created.

Remove a Data Source

To remove a data source in the **Data Sources** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each data source in the table.

2. Click the check box for a data source that you want to remove.
3. Click the Remove a data source icon in the main menu.

Search for a Data Source

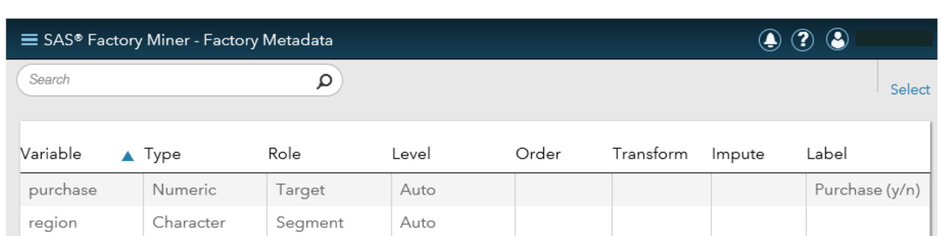
You can search for a data source in the **Data Sources** workspace using the **Search** text box:

After you enter your text, click the Start search icon . The table will update with data sources that match your search terms. When you want to clear your search from the **Search** text box, and return to the default view of data sources, click the Clear search text icon .

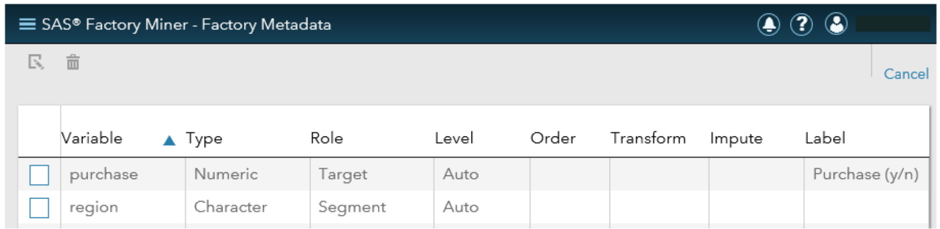
Managing Factory Metadata

Overview of Managing Factory Metadata

In SAS Factory Miner, factory metadata consists of variables that you can use to help you with your models. To navigate to the **Factory Metadata** workspace, click the Side menu icon , and select **Factory Metadata** from the menu that appears. The variables that are displayed in the **Factory Metadata** workspace are those that were added when you defined factory metadata from the **Data Sources** workspace.



You can click **Select** in the **Factory Metadata** workspace to select a factory metadata item, and choose an option from the toolbar.



Edit Metadata

To edit metadata in the **Factory Metadata** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each variable in the table.

2. Click the check box for a variable that you want to modify.
3. Click the Edit global metadata icon .

   A window appears where you can modify variable information.

4. Modify any fields that you want to change.
5. Click **Save**.

Remove Factory Metadata

To remove metadata in the **Factory Metadata** workspace:

1. Click **Select** in the main menu.

   A selection box appears in the left-most column for each variable in the table.

2. Click the check box for a variable that you want to remove.
3. Click the Remove a metadata entry icon in the main menu.

Search for Factory Metadata

You can search for metadata in the **Factory Metadata** workspace using the **Search** text box:

After you enter your text, click the Start search icon . The table will update with metadata that match your search terms. When you want to clear your search from the **Search** text box, and return to the default view of metadata, click the Clear search text icon .

Chapter 3

Building Models with SAS Factory Miner

Overview of Building a Model Factory

After you have created a project, you can build a collection of models in the project. To build models, first select the project that you want to open in the **Projects** workspace.

Note: For more information about creating a project, see "Create a Project" on page 9.

The project appears in a new window. After your project opens, you can configure your model by specifying information using the **Data**, **Profile**, **Model Templates**, and **Results** tabs.

You can also retrain a project with a new data source, download project score code, edit project settings, build a profile, and run all models after a project has been opened.

After you create a project, you can define a target and a segment variable. After you define a target and a segment variable, you can build a profile. After you build a profile, you can run models.

This section contains the following information about how you can perform model building in SAS Factory Miner:

- "Data Tab" on page 22
- "Profile Tab" on page 24
- "Model Templates Tab" on page 26
- "Results Tab" on page 29
- "Retrain a Project with a New Data Source" on page 34
- "Download Project Score Code" on page 35
- "Edit Project Settings" on page 35
- "Build a Profile" on page 36
- "Run All Models" on page 36

Data Tab

Overview of the Data Tab

Information about variables appears in a table on the **Data** tab after you open a project.

You can view and configure variable information, search for variables, and modify your view of variables using the **Data** tab.

View Variable Information

On the **Data** tab, you can view information about variables in the data source. Variable information is presented in a table. The table includes the variable's name, label (if any), type, role, level, and other statistical information.

The table also provides graphical distribution information for a variable. To view distribution information for a variable:

1. In the table on the **Data** tab, select the check box next to a variable that you want to view information about.
2. Click **View**.

   A window appears with distribution information for the variable.
3. Click **Close** when you finish.

Configure Variables

After you create a project, or open a project that has not been configured, you might receive a message that you need to specify a target and a segment variable. This section explains how to configure variable information.

To configure a variable:

1. In the table on the **Data** tab, select the check box next to a variable that you want to configure.
2. Click the Edit icon  .

   A window appears.
3. In the window, select values for the role, level, order, transform, and impute settings.

   In many cases, you can use the default setting. But you need to specify the role for one variable in the data source as the target and specify the role for one variable in the data source as segment. In addition to specifying a target and a segment variable, you might want to adjust the other settings for some variables in your data source.

   Note: In order for a value that is set for the **Transform** column to be used (for an interval input), the template must include the **Transform Variables** component. The option that is selected for the **Interval input** property in the **Transform Variables** component is used for all interval inputs that do not have a value set on the **Data** tab. The value on the **Data** tab is used otherwise. This works similarly for the **Impute** column. Information is picked up by templates that include the **Impute** component. If something is set in the **Impute** column on the **Data** tab, it overrides what is set for the various properties in the **Impute** component (**Default Class Input Method**, **Default target method**, **Default interval input method**, and **Default target method**).
4. Click **Save** when you finish specifying settings for the variable.

Search for Variables

You can search for a variable in the table by entering text into the **Filter** text box.

The list of variables in the table will update to match your search terms.

Note: If the list of variables does not automatically update, try clicking the Start search icon .

Click the Clear search text icon to clear the **Filter** text box and reset your view of variables in the table.

Modify Your View of Variables

You can add or remove data source columns from your view of variables in the table on the **Data** tab.

To modify your view of variables in the **Data** tab:

1. Click the Options icon .

   A menu appears.

2. Select **Columns**.

   The Columns window appears.

3. (Optional) Add a hidden column.

   Note: All of the columns might be displayed in default, so there might not be anything hidden to add in your view.

   a. Select a column that you want to view in the **Available columns** list.

   b. Click the Add icon to add the column to the **Selected columns** list.

4. (Optional) Remove a column from your view.

   a. Select a column that you want to hide from view in the **Selected columns** list.

   b. Click the Remove icon to add the column to the **Available columns** list.

5. Click **OK** when you finish.

Profile Tab

Overview of the Profile Tab

After you create a project, and specify a target and a segment variable on the **Data** tab, then you can build a profile. Information about segments appears in a table on the **Profile** tab.

Note: Before you can view and modify information on the **Profile** tab, you might need to build a profile. For more information about building a profile, see "Build a Profile" on page 36.

You can modify training information, specify segments to include or exclude, and modify your view of segments from the **Profile** tab.

Modify Training Information

When you open the **Profile** tab, training information is presented next to a table that contains segment information. You can modify training information by selecting a slider and entering a number in the text box that appears. To reset training information to the default view, click the Reset icon .

Specify Segments to Include or Exclude

To specify a segment to include or exclude on the **Profile** tab:

1. Select the check box next to a segment in the table that you want to include or exclude.

   Note: You can check multiple rows to exclude or include all selected rows.

2. If the segment is currently included, click the **Exclude** button to exclude it. If the segment is currently excluded, click the **Include** button to include it.

Modify Your View of Segments

To add or remove segment columns from your view in the table of segments on the **Profile** tab:

1. Click the Options icon

   A menu appears.

2. Select **Columns**.

   The Columns window appears.

3. (Optional) Add a hidden column.

   Note: All of the columns might be displayed by default, so there might not be anything hidden to add in your view.

   a. Select a column that you want to view in the **Available columns** list.

   b. Click the Add icon to add the column to the **Selected columns** list.

4. (Optional) Remove a column from your view.

   a. Select a column that you want to remove from view in the **Selected columns** list.

   b. Click the Remove icon to add the column to the **Available columns** list.

5. Click **OK** when you finish.

Model Templates Tab

Overview of the Model Templates Tab

You can click **Model Templates** in the menu to open the **Model Templates** tab after you have opened a project.

When you open the **Model Templates** tab, the available templates appear in the table.

A Baseline template is run for every segment, even segments that are excluded. The Baseline template provides a baseline to compare other models against. For an interval target, the predicted value is the overall target mean. For a non-interval target, the posterior probabilities are the same as the prior probabilities. Here, the most frequent level is used as the predicted value.

You can create a new model template from an algorithm or a global template, edit the properties of a model template, remove a model template, search for a model template, and modify your view of model templates on the **Model Templates** tab.

Create a New Model Template from an Algorithm

To create a new model template from an algorithm on the **Model Templates** tab:

1. Click the New model template from algorithm icon 

   The New Model Template window appears.
2. Provide a name for the new model template.
3. Select a model algorithm from the list of available algorithms.
4. Specify whether you want to make this template available everywhere.
5. Specify whether you want to mark this template as the default one to use.

   Note: Your choices about making the template available everywhere and marking it as a default one to use determine the scope of a new template. If both are unchecked, then you are creating a local template. A local template is one that is known only to the current project. If you specify to make the template available everywhere, then the template will be available to other projects, and will appear in the list of model templates. If you specify to use the default template, then new projects will use the new template automatically. That is, you will not need to manually include it. A new template can be used by default only if it is also made available everywhere.
6. Click **Save**.

   A window appears where you can specify options for the algorithm that you selected.
7. Select or specify values that the model template will use.

   Note: The options that are available depend on the algorithm that you selected.

   a. Specify which components you want to use.

i. Select a component that you want to use in the **Ignore** list.

ii. Drag the component to the **Drag components here** text area in the **Use** list.

Note: If a component has already been dragged to the **Use** list, the **Drag components here** text area might not be visible. If the **Drag components here** text area is not visible, then drag the component to where another component is toward the top of the **Use** list. Components will be applied to a model in the order in which they are displayed in the **Use** list.

The component is added to the **Use** list.

b. Specify values for each component.

i. Select the component whose values you want to modify in the **Use** list.

The current specifications for that property appear in the window.

ii. Select, specify, or provide values that you want to use for that component in the window.

For property information for each model, see Chapter 4, "SAS Factory Miner Models," on page 37.

iii. Repeat the previous two steps for each component whose values you want to modify.

8. Click **Save**.

The new model template appears in the table.

Create a New Model Template from a Global Template

To create a new model template from a global template on the **Model Templates** tab:

1. Click the Add model template from global template icon  .

   The Add Model Template window appears. The templates that are currently available display a check mark in a column next to their name.

2. Click the check box for a template that you want to use.

3. Click **Save**.

View and Modify the Properties of a Model Template

To view and modify the properties of a model template in the **Model Templates** tab:

1. Select the check box for a model template in the table.

2. Click the Edit Properties icon  .

   The properties for the model template appears in a window.

3. (Optional) Specify which components you want to use.

   a. Select a component that you want to use in the **Ignore** list.

   b. Drag the component to the **Drag components here** text area in the **Use** list.

   Note: If a component has already been dragged to the **Use** list, the **Drag components here** text area might not be visible. If the **Drag components**

here text area is not visible, then drag the component to where another component is toward the top of the **Use** list. Components will be applied to a model in the order in which they are displayed in the **Use** list.

The component is added to the **Use** list.

4. (Optional) Specify values for each component.

   a. Select the component whose values you want to modify in the **Use** list.

      The current specifications for that property appear in the window.

   b. Select, specify, or provide values that you want to use for that component in the window.

      For property information for each model, see Chapter 4, “SAS Factory Miner Models,” on page 37.

   c. Repeat the previous two steps for each component whose values you want to modify.

5. Click **Save** if the template has not yet been used in a run, or **Save As** if the template has already been run, to save your changes when you finish viewing the properties for the model template.

Note: If you have already built your profile when you change an existing template, you need to rebuild the profile in order to pick up the changes.

Remove a Model Template

To remove a model template in the **Model Templates** tab:

1. Select the check box for a model template in the table that you want to remove.
2. Click the Remove Model Template icon .

   A confirmation window appears.

3. Click **Yes**.

Search for a Model Template

You can search for a model template in the table by entering text into the **Filter** text box.

The list of model templates in the table will update to match your search terms.

Note: If the list of model templates does not automatically update, try clicking the Start search icon .

Click the Clear search text icon to clear the **Filter** text box and reset your view of model templates in the table.

Modify Your View of Model Templates

To add or remove columns from your view in the table on the **Model Templates** tab:

1. Click the Options icon

   A menu appears.
2. Select **Columns**.

   The Columns window appears.
3. (Optional) Add a hidden column.

   Note: All of the columns might be displayed in default, so there might not be anything hidden to add in your view.

   a. Select a column that you want to view in the **Available columns** list.

   b. Click the Add icon to add the column to the **Selected columns** list.
4. (Optional) Remove a column from your view.

   a. Select a column that you want to hide from view in the **Selected columns** list.

   b. Click the Remove icon to add the column to the **Available columns** list.
5. Click **OK** when you finish modifying which columns to appear in the table on the **Model Templates** tab.

Results Tab

Overview of the Results Tab

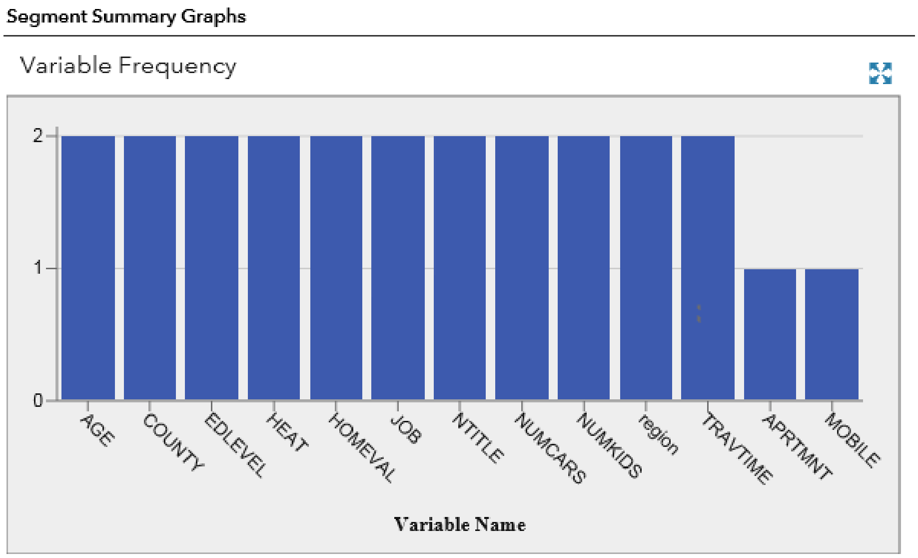
The **Results** tab contains graphical and tabular results for the models that you have run. You can click **Results** on the menu to open the **Results** tab after you open a project.

Note: You might need to run your models before you can examine results. For more information about running models, see “Run a Segment” on page 33.

For a segment on the **Results** tab, you can view graphical and tabular results, sort tabular results information, open segment models, run a segment, stop a segment from running, download score code for a segment, view the training history for a segment, and generate reports.

View Graphical and Tabular Results

After you click the **Results** tab, graphical and tabular results appear in the window.



You can expand your view of graphical output by clicking the Expand Report icon  next to the graphical output that you want to expand. To close the expanded report, click the Exit maximized view icon  in the window that contains the enlarged graphical output.

Tabular output contains information about segments, which model was the champion, and other statistical information.

	Segment ID ▲	Segment (region)	Champion	Override	Validate: Kolmogorov-Smirnov	Validate: Observations
☑	Segment 1	East	Neural Network template	No	0.5378	524
☐	Segment 2	Midwest	Neural Network template	No	0.5679	507
☐	Segment 3	Other	Baseline template	No	0.0000	25
☐	Segment 4	South	Decision Tree template	No	0.5032	414
☐	Segment 5	West	Neural Network template	No	0.4388	300

Tabular results also include the following status information for a segment run:

Status Icon	Segment Run Status
	Completed
	Running
	Scheduled
	Error

Status Icon	Segment Run Status
	Canceled
	Run Needed
	Failed
	Excluded

Sort Tabular Results Information

You can change your view of results information by filtering models by status on the **Results** tab. By default, your view of models by status might be set to **View All**. However, you can change your view by selecting a check box in the **Status** list next to the table of results.

Note: You might need to scroll down to see all of the options in the **Status** list.

Available options include whether a model completed, is running, or is scheduled to run; whether there was an error; whether a model was canceled; and whether a model needs to be run, or failed, or has been excluded.

Note: The model run status state of **Error** indicates that an error was found during a model's execution that stopped it from completing. This is different from the model run status of **Failed**, which means that the SAS code behind the model failed for statistical reasons. A model with a state of **Error** most likely encountered a configuration issue. For example, the project's directory was not available to the SAS server code on the workspace server, or a temporary resource situation occurred that caused its failure. This model can be rerun as part of a project or segment run or individually by selecting the model. However, a model that has a run status state of **Failed** can be rerun only by selecting the specific model.

Explore Model Output for a Segment

To explore model output for a segment on the **Results** tab:

1. Select the check box for the segment that you want to open.
2. Click the Open segment models icon  .

   Note:

   - You can also open segment models by selecting **Actions** ⇨ **Open** or just clicking the Segment ID.
   - To navigate back to the **Results** tab of your project, click the name of your project in the main menu at the top of the screen.

The models for the segment appear in a table, and the champion model is marked with the Champion icon .

		Name	Algorithm	Validate: Kolmogorov-Smirnov (KS)	Status
☑		Neural Network template	Neural Network	0.5378	Completed
☐		Baseline template	Baseline	0.0000	Completed
☐		Decision Tree template	Decision Tree	0.4172	Completed
☐		Random Forest template	Random Forest	0.2107	Completed
☐		Regression template	Regression	0.1328	Completed

Note: To change your view of the columns that are displayed in this table, click the Options icon , and select **Columns** from the menu that appears. The Columns window appears. Next, select the name for a column in the **Available columns** or the **Selected columns** list, and then click the Add icon , or the Remove icon to move the selected column. Finally, click **OK**.

Model graphs and additional tabular statistical results appear below the table.

3. Select the check box for the template that you want to see results for, or just click anywhere in the row.

   After you select a template, graphical output is displayed in the **Model Graphs** section under the table of templates.

   Note: You might need to scroll down to see all the graphical output for a template.

4. Modify your model for a segment.

   In this view, you can create a new model (New Model icon), add a model (Add Model icon), edit a model template (Edit Model Template icon), or perform one of the following in the window that appears when you click **Actions**.

   - Run a model
   - Stop a model from running
   - Compare multiple models
   - Delete a model
   - Duplicate a model
   - Mark a model as a champion
   - Unmark a model as a champion
   - Rename a model
   - Access a report
   - Access the log
   - Download score code

   Note: You might need to select the check box for one or more templates in the table before choosing an action. (Use **Ctrl** to select multiple templates.) For example, you might do this if you want to compare multiple models.

5. View segment properties.

You can view segment properties by clicking the Segment Properties icon  .

Properties for the segment appear in a new window. Click **Close** to exit the Segment Properties window.

Run a Segment

To run a segment on the **Results** tab:

Note: You can only run segments that have not been run yet or that have changes, such as a new model that has not been run.

1. Select the check box for a segment that you want to run in the table.
2. Click **Actions**.

   A menu appears.
3. Select **Run**.

Stop a Segment from Running

To stop a segment from running on the **Results** tab:

1. Select the check box for the segment that you want to stop.
2. Click **Actions**.

   A menu appears.
3. Select **Stop**.

Download Score Code for a Segment

To download score code for a segment on the **Results** tab:

1. Select the check box for a segment that you want to download score code for.
2. Click **Actions**.

   A menu appears.
3. Select **Download Score Code**.

   The Download Score Code window appears.
4. Specify whether you want to download all models or only champion models.
5. Click **Download**.

   Note: Depending on your browser and browser settings, a message might appear that asks whether you want to open or save the file.
6. If you are prompted for whether you want to open or save the file, specify whether you want to open or save the score code.

Note: You might be unable to open a downloaded score file using WinZip if the project display name, the segmentation variable values, or the localized model name contain characters that are not valid for a Windows file path. If you encounter trouble opening a downloaded score file using WinZip, try using 7-Zip or the Windows File Manager to access the content of the downloaded ZIP file.

View Training History

To view training history for a segment on the **Results** tab:

1. Select the check box for a segment that you want to view training history for.
2. Click **Actions**.

   A menu appears.
3. Select **History**.

   The View History window appears.
4. Click the Select Graph Statistic icon to open a menu with available statistics that you can select.
5. Click **Close** when you are finished viewing the history.

Generate Reports

To generate reports for a segment on the **Results** tab:

1. Select the check box for a segment that you want to generate reports for.
2. Click **Actions**.

   A menu appears.
3. Select **Report**.

   The Report window appears.
4. Specify whether you want the report to contain only champion models, or champion models and all model assessment statistics.
5. Specify whether you want a PDF or an RTF format.
6. Click **Generate**.

   The Report window appears.
7. Click **Yes** to view the report.

   A message appears that asks whether you want to open or save the report.
8. Specify whether you want to open or save the report.

Retrain a Project with a New Data Source

To retrain a project with a new data source:

1. Click the Retrain project with new data source icon .

   The Retrain window appears.
2. Select a data source from the menu. Alternatively, click **New data source** to specify a new data source to use.

Note: This action will retrain all segments in the project.

3. Click **Save**.

   This clears out existing models and rebuilds the segment profile with the specified data set.

4. Click **Run** to complete the retraining.

   Note: After you retrain a project with a new data source, a new entry is added to the training history.

Download Project Score Code

To download the project score code:

Note: This is the score code for all segments.

1. Click the Download Project Score Code icon .

   The Download Score Code window appears.

2. Specify whether you want to download all models or only the champion models.
3. Click **Download**.

   A message appears that asks whether you want to open or save the score code.

4. Specify whether you want to open or save the score code.

Edit Project Settings

To edit project settings:

1. Click the Edit Project Settings icon .

   The Edit Project Settings window appears.

2. Modify any settings that you are allowed to change.

   On the **Partition Data** tab, you can specify whether you want to use sample training data and partition the data. If you choose to use sample training data, you can select the sampling method, and indicate whether you want to use a percentage or a specific number of observations. If you choose to partition the data, you can specify the percentage of data to be used for training and validation.

   On the **Rules** tab, you can specify the minimum number of observations and the minimum event rate to use in your training data. Segments that do not meet both of these criteria will be excluded. Models are built only for segments that have at least the minimum number of observations and the minimum event rate that is specified. You can also specify the champion criterion.

   Note: The minimum event rate only applies to non-interval targets.

   On the **Logging** tab, you can specify whether to enable debug logging.

Note: Subsequent model runs will generate additional log information.

3. Click **Save**.

Build a Profile

After you have specified data configuration information, you can build the profile by clicking the Build Profile icon 

in the main menu.

Run All Models

After you create a project, specify a target and a segment variable on the **Data** tab, and build a segment profile, you can run all models by clicking the Run all models icon 

in the main menu.

Chapter 4
SAS Factory Miner Models

Overview of SAS Factory Miner Models

This section contains an overview of each of the following models, model requirements, and information about model properties to help you analyze data using SAS Factory Miner.

- "Bayesian Network Model" on page 38
- "Decision Tree Model" on page 40
- "Generalized Linear Model" on page 42
- "Gradient Boosting Model" on page 45
- "Neural Network Model" on page 47
- "Random Forest Model" on page 49
- "Regression Model" on page 51
- "Support Vector Machine Model" on page 53
- "Common Properties" on page 54

Bayesian Network Model

About the Bayesian Network Model

A Bayesian network is a directed, acyclic graphical model in which nodes represent random variables and the connections between nodes represent the conditional dependency of the random variables. Because a Bayesian network provides a conditional independence structure and a conditional probability table at each node, it can be used as a predictive model with supervised data mining.

SAS Factory Miner can create naive Bayesian networks, tree-augmented naive Bayesian networks, Bayesian network-augmented naive Bayesian networks, and Markov blanket Bayesian networks.

The simplest Bayesian network is a naive Bayesian network. A naive Bayesian network includes only the target variable and some set of the input variables. All input variables are assumed to be conditionally independent of each other. Input variables that are conditionally independent of the target variable are dropped from the naive Bayesian network. Thus, you are left with a classification that contains only the target variable and the inputs on which it depends.

A tree-augmented naive (TAN) Bayesian network does not assume conditional independence of the input variables. Instead, it assumes that a tree structure exists between the input variables. In a TAN, each input variable can be conditionally dependent on the target variable and at most one other input variable. Thus, your classification can contain input variables that are conditionally independent of the target variable.

A Bayesian network-augmented naive (BAN) Bayesian network puts no restriction on the number of conditionally dependent input variables that are allowed for each input variable. That is, a BAN contains all the features of a TAN, but also includes a more robust set of connections between the input variables.

A Markov blanket Bayesian network permits conditional dependence on both the target variable and the input variables. That is, the target variable can be conditionally dependent on some input variables; some input variables can be conditionally dependent on other input variables; and some input variables can be conditionally dependent of the target variable.

Bayesian Network Model Requirements

The Bayesian Network model requires at least one segment and target variable.

Bayesian Network Properties

This section presents information about properties that you can use in a Bayesian Network model.

The following are the Bayesian Network Classifier properties for the Bayesian Network model:

- **Network model** — Specifies the Bayesian network type that is used for classification. The following network structures are available:
  - **Naive Bayes** — No relationship between input variables exists. All connections are between the target variable and the input variables. All input variables are assumed to be conditionally independent.
  - **TAN** — A tree-augmented naive Bayesian network (TAN) includes direct connections between the target variable and the input variables, and a tree structure between the input variables. Each input node can have at most two parent nodes.
  - **BAN** — A Bayesian network-augmented naive Bayesian network includes all the features of a TAN, but permits more than two parents for each child node.
  - **Bayesian Network** — Considers all possible Bayesian networks, including Markov blankets. A Markov blanket Bayesian network allows for some input variables to be parents of the target variable; some input variables to be children of the target variable; and some input variables to be parents of other input variables.
- **Variable selection** — Specifies whether variables selection is performed via a conditional independence test between the target variable and each input variable in the network.
- **Independence test statistics** — Specifies which test statistics are used during variable selection. The following options are available:
  - Chi-square
  - G-square

  - Chi and G-square
- **Significance level** — Specifies the significance level that is used in the independence tests.
- **Number of bins** — All interval input variables are binned into the number of levels specified here. Bucket binning is the only method available.
- **Use missing as level** — Specifies whether missing values are treated as their own measurement level. When this option is not selected, observations with missing values are excluded from the input data.

The following groups of properties are also available for use with the Bayesian Network model:

- "Score Properties" on page 55
- "Filter Properties" on page 55
- "Forest-Based Variable Selection Properties" on page 56
- "Impute Properties" on page 56
- "Principal Components Properties" on page 59
- "Transform Variables Properties" on page 60
- "Tree-Based Variable Selection Properties" on page 60
- "Variable Selection Properties" on page 61

For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

Decision Tree Model

About the Decision Tree Model

Decision trees produce a set of rules that can be used to generate predictions for a new data set. This information can then be used to drive business decisions. For example, in database marketing, decision trees can be used to develop customer profiles that help marketers target promotional mailings in order to generate a higher response rate.

You can create a Decision Tree model to do one of the following tasks:

- classify observations based on the values of nominal or binary targets
- predict outcomes for interval targets

An advantage of the Decision Tree model over other models, such as the Neural Network model, is that it produces output that describes the scoring model with interpretable Node Rules. The Decision Tree model also produces detailed score code output that describes the scoring algorithm in complete detail. For example, the Node Rules for a model might describe this rule: "If monthly mortgage-to-income ratio is less than 28% and months posted late is less than 1 and salary is greater than $30,000, then issue a gold card."

Another advantage of the Decision Tree model is the treatment of missing data. The search for a splitting rule uses the missing values of an input. Surrogate rules are available as backup when missing data prohibits the application of a splitting rule.

Note: Node Rules are useful for understanding the structure of a decision tree. But using the Node Rules set as the sole basis for a scoring algorithm is not recommended. The Node Rules output does not completely describe how the Decision Tree model handles missing and unknown data values in your scoring model.

The SAS Factory Miner implementation of decision trees finds two-way splits based on nominal and interval inputs. You choose the splitting criteria and other options that determine the method of tree construction. The options for a splitting criterion for a nominal target include CHAID, chi-square, entropy, Fast CHAID, Gini, and information gain ratio. For an interval target, the available splitting criteria are CHAID, F Test, and variance. You can specify the maximum depth for the tree as well as various pruning and subtree selection options, including cost-complexity, C4.5, and assessment measure pruning.

Decision Tree Model Requirements

The Decision Tree model requires at least one target variable and at least one input variable. The target variable must be a binary, nominal, or interval variable.

Decision Tree Model Properties

This section presents information about properties that you can use in a Decision Tree model.

The following are the Decision Tree properties for the Decision Tree model:

- **Nominal target splitting criterion** — specifies the method that you want to use to evaluate candidate splitting rules for nominal targets and to search for the best one. Choose from the following splitting criteria:
  - CHAID
  - Chi-square
  - Entropy
  - Fast CHAID
  - Gini
  - Information Gain Ratio
- **Interval target splitting criterion** — specifies the method that you want to use to evaluate candidate splitting rules for interval targets and to search for the best one. Choose from the following splitting criteria:
  - CHAID
  - F Test
  - Variance
- **Maximum depth** — specifies the maximum number of generations of nodes that you want to allow in your decision tree. The original node is the root node. Children of the root node are the first generation.
- **Maximum render depth** — specifies the maximum render depth. It is possible that a tree will require too many resources to render. The tree is only rendered if the maximum depth is less than this value.
- **Missing values** — specifies how splitting rules handle observations that contain missing values for a variable. Select from the following available missing value policies:

  - Popularity
  - Similarity
  - Branch
- **Surrogate rules** — specifies the maximum number of surrogate rules that the Decision Tree model seeks in each non-leaf node. The first surrogate rule is used when the main splitting rule relies on an input whose value is missing.
- **Subtree method** — specifies the method that you want to use to select a subtree from the fully grown tree for each possible number of leaves. The following subtree methods are available:
  - Assessment
  - C4.5
  - Cost-complexity
- **Nominal target assessment measure** — specifies the method that you want to use to select the best tree. The choice is based on the validation data when the **Subtree method** property is set to **Assessment**. If no validation data is available, training data is used. The available assessment measurements are as follows:
  - **Average Squared Error** — selects the tree that has the smallest average squared error.
  - **Misclassification** — selects the tree that has the smallest misclassification rate.
- **C4.5 confidence level** — Specifies the C4.5 confidence level to use to create subtrees based on the C4.5 algorithm.

The following groups of properties are also available for use with the Decision Tree model:

- "Score Properties" on page 55
- "Filter Properties" on page 55
- "Forest-Based Variable Selection Properties" on page 56
- "Impute Properties" on page 56
- "Principal Components Properties" on page 59
- "Transform Variables Properties" on page 60
- "Tree-Based Variable Selection Properties" on page 60
- "Variable Selection Properties" on page 61

For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

Generalized Linear Model

About the Generalized Linear Model

A Generalized Linear model is an extension of a traditional linear model that allows the population mean to depend on a linear predictor through a nonlinear link function. For example, a Generalized Linear model can be used to model traditional insurance measures such as claim frequency, severity, or pure premium. Claim frequency is typically modeled with a Poisson distribution and a logarithmic link function. Claim severity is typically modeled with a gamma distribution and a logarithmic link function. Pure premiums are modeled with the Tweedie distribution. The Generalized Linear model can fit models for standard distributions in the exponential family. The Generalized Linear model fits zero-inflated Poisson and negative binomial models for count data.

Generalized Linear Model Requirements

The Generalized Linear model requires at least one segment and target variable. This model does not support nominal and ordinal target variables. To build models using count data, your target variable must be an interval variable.

Generalized Linear Model Properties

This section presents information about properties that you can use in a Generalized Linear model. For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

The following properties are associated with the Generalized Linear model:

- **Two-factor interactions** — Enable the check box if you want to include all two-factor interactions for class variables that are used.
- **Polynomial terms** — Enable the check box if you want to include polynomial terms for interval variables that are used in the analysis. When the check box for this property is selected, you must specify an integer value for the **Polynomial degree** property.
- **Polynomial degree** — When the **Polynomial terms** check box is selected, use this property to specify the highest degree of polynomial terms (for interval variables that are used) to be included in the analysis. This property can be set to **2** or **3**.
- **Use missing as level** — Enable the check box for this property if missing values should be considered as their own classification level for class inputs.
- **Interval target probability distribution** — Use this property to determine the probability distribution function that is used in your model when your data contains an interval target variable. The binary distribution is used when your data contains a binary target variable. The available distributions are as follows:
  - Poisson
  - Negative binomial
  - Gamma
  - Normal
  - Inverse Gaussian
  - Tweedie
  - Zero-inflated negative binomial
  - Zero-inflated Poisson
- **Interval target link function** — Use this property to specify the link function that you want to use in your analysis when your data contains an interval target variable. Link functions link the response mean to the linear predictor. The available link functions are as follows:
  - Log

- Logit
- Log-log
- Complementary Log-log
- Identity
- Inverse
- Inverse squared
- Probit

- **Binary target link function** — Use this property to specify the link function that you want to use in your analysis when your data contains a binary target variable. Link functions link the response mean to the linear predictor. The available link functions are as follows:
  - Logit
  - Log-log
  - Complementary Log-log
  - Probit
- **Selection method** — Use this property to specify the model selection method that you want to use during training. You can choose from the following effect selection methods:
  - **None** — uses all inputs to fit the model. The **Stop criterion** property is not available when **None** is selected.
  - **Backward** — begins with all candidate effects in the model and removes effects until the **Stay significant level** or the **Stop criterion** property value is met.
  - **Forward** — begins with no candidate effects in the model and adds effects until the **Entry significant level** or the **Stop criterion** property value is met.
  - **Stepwise** — begins as in the forward model but might remove effects already in the model. Continues until the **Stay significant level** property value or the stepwise stopping criteria are met.
- **Stop criterion** — specifies the criterion that is used to stop the selection process. The following criteria are available:
  - **DEFAULT (SL)** — uses the significance levels of the effects.
  - **AIC** — Akaike's Information Criterion.
  - **AICC** — Corrected Akaike's Information Criterion.
  - **SBC** — Schwarz Bayesian Information Criterion.
  - **SL** — uses the significance levels of the effects.
  - **None** — No criterion for stopping selection is used. The selection process stops when no suitable addition or removal of candidate effects is found or if a size-based limit is reached.

  Note: If you specify a criterion other than **SL** or **None**, then the selection process stops when a local extremum is found or if a size-based limit is reached. The determination of whether a local minimum is achieved is made on the basis of a stop horizon at the next three steps.
- **Entry significant level** — significance level for adding variables in forward and stepwise selection.

- **Stay significant level** — significance level for removing variables in backward and stepwise selection.
- **Maximum number of steps** — Maximum number of selection steps that are performed. The value of zero indicates that this option is ignored.

The following groups of properties are also available for use with the Generalized Linear model:

- "Score Properties" on page 55
- "Filter Properties" on page 55
- "Forest-Based Variable Selection Properties" on page 56
- "Impute Properties" on page 56
- "Principal Components Properties" on page 59
- "Transform Variables Properties" on page 60
- "Tree-Based Variable Selection Properties" on page 60
- "Variable Selection Properties" on page 61

For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

Gradient Boosting Model

About the Gradient Boosting Model

The Gradient Boosting model uses a partitioning algorithm described in Friedman (2001 and 2002). A partitioning algorithm searches for an optimal partition of the data, which is defined in terms of the values of a single variable. The optimality criterion depends on how another variable, the target, is distributed into the partition segments. The more similar the target values are within the segments, the greater the worth of the partition. Most partitioning algorithms further partition each segment in a process called recursive partitioning. The partitions are then combined to create a predictive model. The model is evaluated by goodness-of-fit statistics, which are defined in terms of the target variable. These statistics are different from the measure of worth of an individual partition. A good model might result from many mediocre partitions.

A gradient boosting resamples the analysis data set several times to generate results that form a weighted average of the resampled data set. Tree boosting creates a series of decision trees that together form a single predictive model. A tree in the series is fit to the residual of the prediction from the earlier trees in the series. The residual is defined in terms of the derivative of a loss function. For squared error loss with an interval target, the residual is the target value minus the predicted value. Each time, the data is used to grow a tree, and the accuracy of the tree is computed. The successive samples are adjusted to accommodate previously computed inaccuracies. Because each successive sample is weighted according to the classification accuracy of previous models, this approach is sometimes called stochastic gradient boosting. Boosting is defined for binary, nominal, and interval targets.

Like decision trees, boosting makes no assumptions about the distribution of the data. For an interval input, the model depends only on the ranks of the values. For an interval target, the influence of an extreme value theory depends on the loss function. The Gradient Boosting model offers a Huber M-estimate loss, which reduces the influence of extreme target values. Boosting is less prone to overfit the data than a single decision tree, and if a decision tree fits the data fairly well, then boosting often improves the fit.

Gradient Boosting Model Requirements

The Gradient Boosting model requires at least one segment and target variable.

Gradient Boosting Properties

This section presents information about properties that you can use in a Gradient Boosting model. For more information about how to configure properties for a model, see “View and Modify the Properties of a Model Template” on page 27.

The following properties are associated with the Gradient Boosting model:

- **N iterations** — specifies the number of terms in the boosting series. For interval and binary targets, the number of iterations equals the trees. For a nominal target, a separate tree is created for each target category in each iteration series.
- **Seed** — specifies the seed for generating random numbers. The **Train proportion** property uses this value to select a training sample at each iteration.
- **Shrinkage** — specifies how much to reduce the prediction of each tree.
- **Train proportion** — specifies the proportion of training observations to train a tree with. A different training sample is taken in each iteration. Trees trained in the same iteration have the same training data.
- **Maximum depth** — specifies the maximum depth of a node that will be created. The depth of a node equals the number of splitting rules needed to define the node. The root node has depth zero. The children of the root node are the first generation and have depth one.
- **Missing values** — specifies how splitting rules handle observations that contain missing values for a variable. If a surrogate rule can assign an observation to a branch, then it does. The missing value policy is ignored for the specific observation. Select from the following available missing value policies:
  - **Use in search** — uses missing values during the split search.
  - **Largest branch** — assigns the observations that contain missing values to the branch with the largest number of training observations.
  - **Most correlated branch** — assigns an observation with missing values to the most correlated branch.
- **Assessment measure** — specifies the method that you want to use to select the best tree, based on the validation data. If no validation data is available, training data is used. The available assessment measurements are as follows:
  - **Average Squared Error** — This method selects the tree that has the smallest average square error.
  - **Misclassification** — This method selects the tree that has the smallest misclassification rate.
- **Leaf Fraction** — specifies the smallest number of training observations that a new branch can have, expressed as the proportion of the number N of available training observations in the data. N can be less than the total number of observations in the data set because observations with a missing target value are excluded.

The following groups of properties are also available for use with the Gradient Boosting model:

- "Score Properties" on page 55
- "Filter Properties" on page 55
- "Forest-Based Variable Selection Properties" on page 56
- "Impute Properties" on page 56
- "Principal Components Properties" on page 59
- "Transform Variables Properties" on page 60
- "Tree-Based Variable Selection Properties" on page 60
- "Variable Selection Properties" on page 61

For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

Neural Network Model

About the Neural Network Model

The Neural Network model creates multilayer neural networks that pass information from one layer to the next in order to map an input to a specific category or predicted value. The Neural Network model enables this mapping to take place in a distributed computing environment. This enables you to build neural networks on massive data sets in a relatively short amount of time.

A neural network consists of units (neurons) and connections between those units. There are three types of units:

- Input Units — obtain the values of input variables and standardize those values.
- Hidden Units — perform internal computations and provide the nonlinearity that makes neural networks powerful.
- Output Units — compute predicted values and compare those values with the values of the target variables.

Units pass information to other units through connections. Connections are directional and indicate the flow of computation within the network. Connections cannot form loops, because the Neural Network model permits only feed-forward networks. The following restrictions apply to feed-forward networks:

- Input units can be connected to hidden units or to output units.
- Hidden units can be connected to other hidden units or to output units.
- Output units cannot be connected to other units.

Each unit produces a single computed value. For input and hidden units, this computed value is passed along the connections to other hidden or output units. For output units, the computed value is the predicted value. The predicted value is compared with the target value to compute the error function, which the training method attempts to minimize.

The Neural Network model was designed with two goals. First, the Neural Network model aims to perform efficient, high-speed training of neural networks. Second, the

Neural Network model attempts to create accurate, generalizable models in an easy to use manner. For this reason, most parameters for the neural network are selected automatically. This includes standardization of input and target variables, activation and error functions, and termination of model training.

Neural Network Model Requirements

The Neural Network model requires at least one segment and one binary, nominal, or interval target variable. This model also requires at least one input variable.

Neural Network Properties

This section presents information about properties that you can use in a Neural Network model. For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

The following properties are associated with the Neural Network model:

- **Architecture** — specifies the network architecture that you want to use during network training. Available options include the following:
  - Logistic
  - One layer
  - One layer with direct architecture
  - Two layers
  - Two layers with direct architecture
- **Number of hidden neurons** — specifies the value of hidden neurons. You can use the slider or type in the value in the text box that appears when you select the slider. For two-layer architectures, the number of hidden neurons is split equally across the hidden layers. If the number of hidden neurons is odd, the first hidden layer has the extra neuron.
- **Number of tries** — Specify the number of tries as an integer. You can use the slider or, you can enter the value in the text box that appears when you select the slider.
- **Maximum iterations** — specifies the maximum number of iterations that you want to allow during network training.
- **Use missing as level** — specifies whether missing values are assigned to their own level.

The following groups of properties are also available for use with the Neural Network model:

- "Score Properties" on page 55
- "Filter Properties" on page 55
- "Forest-Based Variable Selection Properties" on page 56
- "Impute Properties" on page 56
- "Principal Components Properties" on page 59
- "Transform Variables Properties" on page 60
- "Tree-Based Variable Selection Properties" on page 60
- "Variable Selection Properties" on page 61

For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

Random Forest Model

About the Random Forest Model

The Random Forest model is a predictive model that consists of several decision trees that differ from each other in two ways. First, the training data for a tree is a sample without replacement from all available observations. Second, the input variables that are considered for splitting a node are randomly selected from all available inputs. In other respects, trees in a forest are trained like standard trees.

The Random Forest model accepts interval and nominal target variables. For an interval target, the procedure averages the predictions of the individual trees to predict an observation. For a categorical target, the posterior probabilities in the forest are the averages of the posterior probabilities of the individual trees. The model makes a second prediction by voting: the forest predicts the target category that the individual trees predict most often.

The training data for an individual tree excludes some of the available data. The data that is withheld from training is called the out-of-bag sample. An individual tree uses only the out-of-bag sample to form predictions. These predictions are more reliable than predictions from training data.

Averaging over trees with different training samples reduces the dependence of the predictions on a particular training sample. Increasing the number of trees does not increase the risk of overfitting the data and can decrease it. However, if the predictions from different trees are correlated, then increasing the number of trees makes little or no improvement.

The Random Forest model overfits the training data when every tree overfits the training data. One way to mitigate overfitting in a tree without pruning is to require each leaf to contain many observations. The Random Forest model initializes the minimum leaf size to 0.1% of the available data and limits the number of leaves to one thousand. The three main training options for tree forests are the number of trees, the number of inputs, and the sampling strategy.

In each node of a decision tree, the Random Forest model randomly selects which input variables to consider for splitting the node, and ignores the rest of the available inputs. The selection ignores the predictive quality of the inputs. The intent is to insert random variation in the trees to reduce their correlation. If most inputs are predictive, then limiting the selection to a few of these can make sense. However, if most inputs are useless for prediction, then many inputs should be considered in order to include at least one that it is predictive. Unfortunately, the number of predictive inputs is rarely known before you perform the analysis.

The Random Forest model uses normalized, formatted values of categorical variables. It considers two categorical values to be the same if the normalized values are identical. Normalization removes any leading blank spaces from a value, converts lowercase characters to uppercase, and truncates all values to 32 bytes.

Random Forest Model Requirements

The Random Forest model requires at least one segment and target variable. If your input data set contains a frequency variable, then the frequency variable must be an interval variable. Frequency variable observations that are non-positive are ignored.

Random Forest Model Properties

This section presents information about properties that you can use in a Random Forest model. For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

The following properties are associated with the Random Forest model:

- **Maximum number of trees** — specifies the number of trees in the forest. The number of trees in the resulting forest can be less than the value specified when the Random Forest model fails to split the training data for a tree. The Random Forest model attempts to create up to twice the number of trees specified.
- **Proportion of observations in each sample** — specifies what percentage of observations is used for each tree.
- **Maximum depth** — specifies the maximum depth of a node in any tree that the Random Forest model creates. The root node has depth 0. The children of the root have depth 1, and so on.
- **Method to determine leaf size** — specifies the method used to determine the leaf size value. Select **Default** to let the Random Forest model determine the method that is used for each leaf. Select **Count** to use the number of observations that is specified in the **Leaf Size** property.
- **Leaf Size** — specifies the leaf size when you select **Count** as the value for the **Method to determine leaf size** property using the slider. Or you can click the slider and enter the value for the leaf size in the text box that appears.

The following groups of properties are also available for use with the Random Forest model:

- "Score Properties" on page 55
- "Filter Properties" on page 55
- "Forest-Based Variable Selection Properties" on page 56
- "Impute Properties" on page 56
- "Principal Components Properties" on page 59
- "Transform Variables Properties" on page 60
- "Tree-Based Variable Selection Properties" on page 60
- "Variable Selection Properties" on page 61

For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

Regression Model

About the Regression Model

The Regression model can be used for both linear and logistic regression models. Linear regression attempts to predict the value of an interval target as a linear function of one or more independent inputs. Logistic regression attempts to predict the probability that a binary or nominal target will acquire the event of interest as a function of one or more independent inputs.

The Regression model uses an identity link function and a normal distribution error function for linear regression. The Regression model uses either a logit, complementary log-log, or probit link function and a binomial distribution error function for a logistic regression analysis with a binary target. For a nominal or ordinal target, the GLOGIT link function and a multinomial distribution error function are used.

The Regression model supports binary, interval, nominal, and ordinal target variables. An example of a binary target variable is "purchase" or "no-purchase", which is often used for modeling customer profiles. An example of an interval target variable is value of purchase. Value of purchase is useful for modeling the best customers for particular products, catalogs, or sales campaigns. Note that ordinal targets are treated as nominal. Your input variables can be continuous (interval) or discrete (binary, nominal, or ordinal).

The Regression model supports forward, backward, and stepwise selection methods, and fast backward for non-interval targets, and forward, backward, stepwise, LAR, and LASSO selection methods for interval targets.

Regression Model Requirements

The Regression model requires at least one segment and target variable.

The input data should have the following data structure:

- One observation per customer.
- An interval, ordinal, nominal, or binary target (response) variable.
- Input variables, which can contain ID (identification) variables, demographic data, history of previous purchases, and so on.

Regression Properties

This section presents information about properties that you can use in a Regression model. For more information about how to configure properties for a model, see “View and Modify the Properties of a Model Template” on page 27.

The following properties are associated with the Regression model:

- **Two-factor interactions** — Enable the check box of this property if you want to include all two-factor interactions for class variables that are used.
- **Polynomial terms** — Enable the check box of this property if you want to include polynomial terms for interval variables that are used in the regression analysis. When

this property is selected, you must specify an integer value for the **Polynomial degree** property.

- **Polynomial degree** — When the **Polynomial terms** property is selected, use this property to specify the highest degree of polynomial terms (for interval variables that are used) to be included in the regression analysis. This property can be set to **2** or **3**.
- **Use missing as level** — specifies whether missing values are assigned to their own level.
- **Binary target link function** — specifies the link function that you want to use in your regression analysis. Link functions link the response mean to the linear predictor. In a linear regression, the identity link function g(M) = Xβ is used. For an ordinal or nominal target, the GLOGIT function is used. In a Logistic regression for a binary target, you can select one of the link functions:
  - Complementary log-log
  - Logit
  - log-log
  - Probit
- **Selection method** — specifies the model selection method that you want to use during training. You can choose from the following effect selection methods:
  - **None** — All inputs are used to fit the model.
  - **Backward** — begins with all candidate effects in the model. It then systematically removes effects that are not significantly associated with the target until no other effect in the model meets the **Stay significance level** value or until the **Stop criterion** value is met. This method is not recommended when the target is binary or ordinal and there are many candidate effects or many levels for some classification input variables.
  - **Fast Backward** — This method starts with all effects in the model and deletes effects without refitting the model. This is used only for a logistic regression model.
  - **Forward** — begins with no candidate effects in the model. It then systematically adds effects that are significantly associated with the target until none of the remaining effects meets the **Entry significant level** value or until the **Stop criterion** value is met.
  - **Stepwise** — Selection begins with no candidate effects in the model and then systematically adds effects that are significantly associated with the target. However, after an effect is added to the model, stepwise might remove any effect that is already in the model that is not significantly associated with the target. This stepwise process continues until one of the following occurs:
    - No other effect in the model meets the **Stay significance level** value.
    - The **Maximum number of steps** value is met. If you choose the stepwise selection method, then you can specify a maximum number of steps to permit before the effect selection process stops.
    - An effect added in one step is the only effect deleted in the next step.
  - **LAR** — The Least Angle Regression (LAR) algorithm is useful for selecting the best-fitting model. The LAR algorithm produces a sequence of regression models. One model parameter is added with each step. The sequence of models terminates at the full least squares solution after all parameters have entered the model. This is used only for linear regression models.

- **LASSO** — The LASSO (Least Absolute Shrinkage and Selection Operator) selection method arises from a constrained form of ordinary least squares, where the sum of the absolute values of the regression coefficients is constrained to be smaller than a specified parameter. This is used only for linear regression models.

- **Stop criterion** — specifies the stop criterion to use. Available choices include the following:
  - DEFAULT
  - AIC
  - AICC
  - SBC
  - Significance Level
  - None
- **Entry significant level** — specifies the entry significance level setting that you want to use to add variables in forward or stepwise regressions.
- **Stay significance level** — specifies the significance level that you want to use when removing variables during backward or stepwise regression.
- **Maximum number of steps** — specifies the maximum number of steps that you want to allow during the stepwise model effect selection process.

The following groups of properties are also available for use with the Regression model:

- "Score Properties" on page 55
- "Filter Properties" on page 55
- "Forest-Based Variable Selection Properties" on page 56
- "Impute Properties" on page 56
- "Principal Components Properties" on page 59
- "Transform Variables Properties" on page 60
- "Tree-Based Variable Selection Properties" on page 60
- "Variable Selection Properties" on page 61

For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

Support Vector Machine Model

About the Support Vector Machine Model

A Support Vector Machine (SVM) model is a supervised machine-learning method that is used to perform classification and regression analysis. Vapnik & Cortes (1995) developed the concept of SVM in terms of hard margin. Later, he and his colleague proposed the SVM with slack variables, which is a soft margin classifier. The standard SVM model solves binary classification problems that produce non-probability output (only sign +1/-1) by constructing a set of hyperplanes that maximize the margin between two classes. Most problems in a finite dimensional space are not linearly separable. In this case, the original space needs to be mapped into a much higher dimensional space or an infinite dimensional space, which makes the separation easier. SVM uses a kernel function to define the larger dimensional space.

The SVM model uses PROC HPSVM. The SVM model supports only binary classification problems, using interior point optimization with a linear kernel. The SVM model does not perform multi-class problems or support vector regression.

Support Vector Machine Model Requirements

The Support Vector Machine (SVM) model requires at least one segment and one target variable. The SVM model requires one or more input variables and exactly one binary target variable. The input variables can be binary, ordinal, nominal, or interval. Observations with missing values are ignored during model training.

Support Vector Machine Properties

This section presents information about properties that you can use in a Support Vector Machine model. For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

The following properties are associated with the Support Vector Machine model:

- **Penalty** — specifies the penalty value.
- **Maximum iteration** — specifies the maximum number of iterations that are allowed within each optimization.
- **Use missing as level** — specifies whether missing values are assigned to their own level.
- **Tolerance** — specifies a termination tolerance for the optimization.

The following groups of properties are also available for use with the Support Vector Machine model:

- "Score Properties" on page 55
- "Filter Properties" on page 55
- "Forest-Based Variable Selection Properties" on page 56
- "Impute Properties" on page 56
- "Principal Components Properties" on page 59
- "Transform Variables Properties" on page 60
- "Tree-Based Variable Selection Properties" on page 60
- "Variable Selection Properties" on page 61

For more information about how to configure properties for a model, see "View and Modify the Properties of a Model Template" on page 27.

Common Properties

Overview of Common Properties

This section contains information about properties that you can use in multiple models.

Score Properties

The following are the Score properties:

- **Optimize score code** — Disable the check box for this property if you do not want to optimize the score code that is generated.
- **Cutoff method** — specifies one of the following cutoff methods:
  - **Prior value** — uses the prior probability of the target event as the cutoff for assigning predictions based on the posterior probabilities for a binary target.
  - **User input** — uses the value specified in the **Classification cutoff** property for determining the predicted value of a binary target
- **Classification cutoff** — Use the slider to specify specifies a classification cutoff value when the **Cutoff method** property is set to **User input**. Alternatively, click the slider and enter a value in the check box that appears.

Filter Properties

The following are the Filter properties:

- **Keep missing interval variable values** — Disable the check box for this property if you want to filter out observations that contain missing values for interval variables.
- **Keep missing class variable values** — Disable the check box for this property if you want to filter out observations that contain missing values for class variables.
- **Interval default method** — specifies one of the following interval default methods:
  - Standard deviations from the mean
  - Trimmed
  - Winsorized
  - None
- **Class default method** — specifies the method that you want to use to filter class variables. Select one of the following options:
  - **Rare Values (Count)** — This option drops rare levels that have a count that is less than the level that you specify in the **Minimum frequency cutoff** property.
  - **Rare Values (Percentage)** — This option drops rare levels that occur in proportions lower than the percentage that you specify in the **Minimum cutoff for percentage** property.
  - **None** — This option specifies that no class variable filtering is performed.
- **Minimum frequency cutoff** — When you set the **Class default method** property to **Rare Values (Count)**, you must use this property to quantify the minimum count threshold. Class variable values that occur fewer times than the specified count are filtered.
- **Minimum cutoff for percentage** — When you set the **Class default method** property to **Rare Values (Percentage)**, you must use this property to specify the minimum percentage threshold. Observations with rare levels for a class variable are counted to calculate rare level proportions. Rare levels that do not meet the minimum percentage threshold are filtered.

- **Maximum number of levels for cutoff** — determines whether a class variable in a data set will be considered for filtering. Only class variables that have fewer levels than the cutoff value are considered for filtering.
- **Cutoff for standard deviation** — quantifies n, the threshold for number of standard deviations from the mean. Observations with values more than n standard deviations from the mean are filtered.
- **Cutoff for robust methods** — specifies the cutoff for robust methods. Use the slider to select a value, or specify a value in the text box that appears when you click the slider.

Forest-Based Variable Selection Properties

The following are the Forest-Based Variable Selection properties:

- **Maximum number of trees** — specifies the number of trees in the forest. The number of trees in the resulting forest can be less than the value specified here when the model fails to split the training data for a tree. The model attempts to create up to twice the number of trees specified.
- **Proportion of observations in each sample** — specifies what percentage of observations is used for each tree.
- **Maximum depth** — specifies the maximum depth of a node in any tree that the model creates. The root node has a depth of 0. The children of the root have a depth of 1, and so on.
- **Variable importance method** — specifies the variable importance method. Select one of the following:
  - Loss reduction
  - Random branch assignments
- **Number of variables to consider for random branch assignments** — specifies the number of variables to consider for random branch assignment.
- **Cutoff fraction for random branch assignments** — specifies the cutoff fraction for random branch assignments. You can use the slider to specify a value. Alternatively, click the slider, and enter the decimal fraction in the text box that appears.
- **Method to determine leaf size** — specifies the method used to determine the leaf size value. Select **Default** to let the model determine the method that is used for each leaf. Select **Count** to use the number of observations specified in the **Leaf size** property.
- **Leaf size** — specifies the leaf size for the model to use when the **Method to determine leaf size** property is set to **Count**. You can use the slider to specify a value. Alternatively, click the slider, and enter a value in the text box that appears.

Impute Properties

The following are the Impute properties:

- **Non-missing variables** — Enable the check box of this property if you want to create score code for variables that did not have missing values during training. If the check box for this property is not checked, this means that imputation score code is not created for variables that did not have missing values in the training data.

- **Missing cutoff** — specifies the maximum percent of missing allowed for a variable to be imputed. Variables whose percentage of missing exceeds this cutoff are ignored.
- **Default Class Input Method** — specifies the imputation statistic that you want to use to replace missing class variables. The choices are as follows:
  - **Count** — replaces missing class variable values with the most frequently occurring class variable value.
  - **Default constant value** — replaces missing class variable values with the value that you enter in the **Default character value** property.
  - **Distribution** — replaces missing class variable values with replacement values that are calculated based on the random percentiles of the variable's distribution. In this case, the assignment of values is based on the probability distribution of the nonmissing observations. The distribution imputation method typically does not change the distribution of the data very much.
  - **None** — Missing class variable values are not imputed under this option.
- **Default target method** — specifies the imputation statistic that you want to use to replace missing class target variables. The choices are as follows:
  - **Count** — replaces missing target variable values with the most frequently occurring target variable value.
  - **Default constant value** — replaces missing target variable values with the value that you enter in the **Default character value** property.
  - **Distribution** — replaces missing target variable values with replacement values that are calculated based on the random percentiles of the variable's distribution. In this case, the assignment of values is based on the probability distribution of the nonmissing observations. The distribution imputation method typically does not change the distribution of the data very much.
  - **None** — Missing target variable values are not imputed under this option.
- **Distribution cutoff** — specifies the distribution cutoff value. You can use the slider to specify the value. Alternatively, you can select the slider, and enter the value in the text box that appears.
- **Statistics** — specifies the impute statistics that are used. You can select one of the following options:
  - Data
  - Trimmed
  - Winsorized
- **Default interval input method** — specifies the imputation statistic that you want to use to replace missing interval variables. The choices are as follows:
  - **Mean** — replaces missing interval variable values with the arithmetic average, calculated as the sum of all values divided by the number of observations. The mean is the most common measure of a variable's central tendency. It is an unbiased estimate of the population mean. The mean is the preferred statistic to use to replace missing values if the variable values are at least approximately symmetric (for example, a bell-shaped normal distribution).
  - **Maximum** — replaces missing interval variable values with the maximum value for the variable.

- **Minimum** — replaces missing interval variable values with the minimum value for the variable.
- **Midrange** — replaces missing interval variable values with the maximum value for the variable plus the minimum value for the variable divided by two. The midrange is a rough measure of central tendency that is easy to calculate.
- **Median** — replaces missing interval variable values with the 50th percentile. The 50th percentile is either the middle value or the arithmetic mean of the two middle values for a set of numbers arranged in ascending order. The mean and median are equal for a symmetric distribution. The median is less sensitive to extreme values than the mean or midrange. Therefore, the median is preferable when you want to impute missing values for variables that have skewed distributions. The median is also useful for ordinal data.
- **Default constant value** — replaces missing interval variable values with the value that you enter in the **Default character value** property.
- **None** — Use this option if you do not want to replace missing interval variable values.

- **Default target method** — specifies the imputation statistic that you want to use to replace missing target variables. The choices are as follows:
  - **Mean** — replaces missing target variable values with the arithmetic average, calculated as the sum of all values divided by the number of observations. The mean is the most common measure of a variable's central tendency. It is an unbiased estimate of the population mean. The mean is the preferred statistic to use to replace missing values if the variable values are at least approximately symmetric (for example, a bell-shaped normal distribution).
  - **Maximum** — replaces missing target variable values with the maximum value for the variable.
  - **Minimum** — replaces missing target variable values with the minimum value for the variable.
  - **Midrange** — replaces missing target variable values with the maximum value for the variable plus the minimum value for the variable divided by two. The midrange is a rough measure of central tendency that is easy to calculate.
  - **Median** — replaces missing target variable values with the 50th percentile. The 50th percentile is either the middle value or the arithmetic mean of the two middle values for a set of numbers arranged in ascending order. The mean and median are equal for a symmetric distribution. The median is less sensitive to extreme values than the mean or midrange. Therefore, the median is preferable when you want to impute missing values for variables that have skewed distributions. The median is also useful for ordinal data.
  - **Default constant value** — replaces missing target variable values with the value that you enter in the **Default character value** property.
  - **None** — Use this option if you do not want to replace missing target variable values.
- **Default character value** — specifies the character string or value that you want to use during constant character imputation.
- **Default number value** — specifies the numeric value that you want to use as a constant value for numeric imputation.
- **Type** — indicates whether you want to create indicator variables to flag the imputed observations for each variable. You can choose from the following settings:

- **Single** — A single indicator variable is created to indicate that one or more variables were imputed.
- **Unique** — Unique binary indicator variables are created for every imputed variable.
- **None** — Do not create an indicator variable.

- **Source** — specifies the role that you want to assign to the created indicator variables. You can choose between **Imputed Variables** and **Missing Variables**.

Note: Information is picked up by templates that include the **Impute** component. If something is set in the **Impute** column on the **Data** tab, it overrides what is set for the various properties in the **Impute** component (**Default Class Input Method**, **Default target method**, **Default interval input method**, and **Default target method**).

Principal Components Properties

The following are the Principal Components properties:

- **Eigenvalue source** — specifies the type of source matrix that you want to use to calculate eigenvalues and eigenvectors. You can choose from the following:
  - **Covariance** — calculates eigenvalues and eigenvectors using the covariance matrix.
  - **Correlation** — calculates eigenvalues and eigenvectors using the correlation matrix.
- **Include class variables** — Enable the check box to specify that you want to include class variables.
- **Use missing as level** — specifies whether missing values are assigned to their own level.
- **Cumulative** — specifies the cutoff criterion of the cumulative proportion of the total variance that is attributable to principal components. Principal components that have a cumulative proportional variance greater than the cutoff value are not passed to successor nodes.
- **Increment** — specifies the cutoff criterion of the proportional increment for the total variance that is attributable to principal components. After the cumulative proportional variance reaches 0.9, the principal components that have a proportional increment less than the cutoff value are not passed on to successor nodes.
- **Apply maximum number** — indicates whether to apply the upper threshold for the maximum number of principal components that are passed to successor nodes. When the check box for this property is selected, you must specify the threshold value in the **Maximum number** property.
- **Maximum number** — specifies a value for the maximum number of principal components to be used when the **Apply maximum number** property is selected.
- **Applies a fixed number** — indicates whether to use a fixed number of principal components. When the check box for this property is selected, you must specify the fixed number in the **Fixed number** property.
- **Fixed number** — specifies a value for the fixed number of principal components to be used when the **Applies a fixed number** property is selected.

Transform Variables Properties

The following are the Transform Variables properties:

- **Interval input** — specifies the default transformation method that you want to apply to interval input variables. The available methods are as follows:
  - **Log** — transformed using the logarithm of the variable.
  - **Log 10** — transformed using the base-10 logarithm of the variable.
  - **Square root** — transformed using the square root of the variable.
  - **Inverse** — transformed using the inverse of the variable.
  - **Square** — transformed using the square of the variable.
  - **Exponential** — transformed using the exponential logarithm of the variable.
  - **Centering** — centers variable values by subtracting the mean from each variable.
  - **Standardize** — standardizes the variable by subtracting the mean and dividing by the standard deviation.
  - **Range** — transformed using a scaled value of a variable equal to (x - min) / (max - min), where x is the current variable value, min is the minimum value for that variable, and max is the maximum value for that variable.
  - **Bucket** — Buckets are created by dividing the data into evenly spaced intervals based on the difference between the maximum and minimum values.
  - **Pseudo-Quantile** — divides data into groups with approximately the same frequency in groups.
  - **Optimal binning** — bins data in order to maximize the relationship to the target.
  - **None** — No transformation is performed.
- **Number of bins** — specifies the number of bins to use when performing optimal binning transformations.
- **Missing values** — specifies how to handle missing values when you use an optimal binning transformation. Select from any of the available missing value policies:
  - **Ignore** — ignores missing values.
  - **First** — assigns the observations that contain missing values to the first branch.
  - **Separate** — assigns missing values to its own separate branch.
- **Reject** — specifies whether the model role of the original variables should be changed to **Rejected** or not.

Note: In order for a value that is set for the **Transform** column on the **Data** tab to be used (for an interval input), the template must include the **Transform Variables** component. The option that is selected for the **Interval input** property in the **Transform Variables** component is used for all interval inputs for which a value has not been set on the **Data** tab. Otherwise, the value on the **Data** tab is used.

Tree-Based Variable Selection Properties

The following are the Tree-Based Variable Selection properties:

- **Nominal target criterion** — specifies the method that you want to use to evaluate candidate splitting rules for nominal variables and to search for the best one. Choose from the following splitting criteria:
  - Entropy
  - Fast CHAID
  - GINI
  - Chi-square
- **Interval target criterion** — specifies the method that you want to use to evaluate candidate splitting rules for interval variables and to search for the best one. Choose from the following splitting criteria:
  - Variance
  - F test
- **Minimum categorical size** — specifies the minimum number of training observations that a categorical value must have before the category can be used in a split search.
- **Maximum branch** — specifies the maximum number of branches that you want a splitting rule to produce. The minimum value of 2 results in binary splits.
- **Interval bins** — specifies the number of interval bins to use.
- **Leaf size** — specifies the minimum number of training observations that are allowed in a leaf node.
- **Maximum selected** — specifies the maximum selected.
- **Relative importance cutoff** — specifies the relative importance cutoff.

Variable Selection Properties

The following are the Variable Selection properties:

- **Target model** — specifies the variable selection method that you want to use. Select one of the following:
  - Unsupervised selection
  - Supervised selection
  - Sequential selection
- **Unsupervised maximum steps** — specifies the unsupervised maximum steps.
- **Cumulative variance cutoff** — specifies the cumulative variance cutoff.
- **Incremental variance cutoff** — specifies the incremental variance cutoff.
- **Selection method** — specifies the selection method. Select one of the following:
  - Fast selection
  - LAR
  - LASSO
- **Supervised maximum steps** — specifies the supervised maximum steps.
- **Stop criterion** — specifies one of the following stop criterion methods:
  - SBC

- AIC
- AICC
- No stop criterion

References

- Cortes, C., and Vladimir Vapnik. 1995. “Support-Vector Networks.” Machine Learning, 20:3:, pp. 273-297.
- Friedman, Jerome H. 2001. “Greedy function approximation: A gradient boosting machine.” Annals of Statistics, 29:5, pp. 1189-1232.
- Friedman, Jerome H. 2002. “Stochastic Gradient Boosting.” Computational Statistics Data Analysis, 38:4, pp. 367-378.

Chapter 5

Getting Started with SAS Factory Miner

Overview

This section provides an example of how you can create a project and a model, and examine results using SAS Factory Miner. This example is split into the following sections:

- "Scenario" on page 63
- "Data Set" on page 63
- "Example" on page 64

Scenario

Suppose that you work for a retailer that wants to identify characteristics of customers that are most likely to respond to online offers. This information will help you target future promotions to the customers that are most likely to respond. Suppose that you want to build separate models for each of the regions in which your clients live.

In this scenario, you will use an online offer data set that includes historical data on retail clients. Assume that customer information was collected from partial participation in a prior promotion. In addition to purchase information, the data that you have to work with also includes demographic and household information about each client.

Data Set

In this example, you use a data set called **PURCHASES**.

In the **PURCHASES** data set, the **purchase** variable should be set as the target (`yes` means purchase, and `no` means did not buy). The **region** variable (five regions) should be set as the segment variable.

Note: The following section describes the steps for identifying a target and a segment variable for this example.

region	Frequency	Percent	Cumulative Frequency	Cumulative Percent
East	1743	29.55	1743	29.55
Midwest	1689	28.64	3432	58.19
Other	87	1.48	3519	59.66
South	1380	23.40	4899	83.06
West	999	16.94	5898	100.00

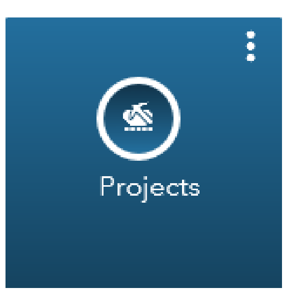
Example

The following example illustrates how you could use the SAS Factory Miner application to develop a model to help you identify characteristics of customers that are most likely to respond to online offers:

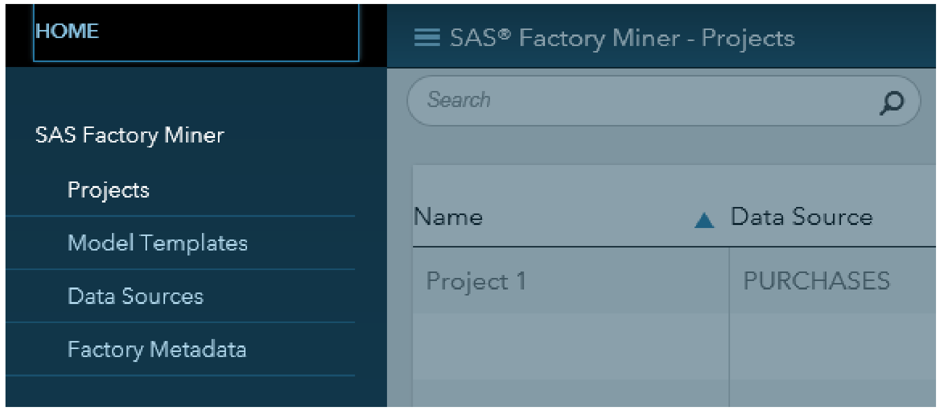
1. Log on to SAS Factory Miner.
   a. Navigate to the URL that you use to access SAS Factory Miner.
   b. Provide your user ID.
   c. Provide your password.
   d. Click **SIGN IN**.

   SAS Factory Miner appears and opens in the **Projects** workspace.

   Note: If your default view is the SAS Home page for the SAS Visual Analytics Hub, click the **Projects** button to navigate to the **Projects** workspace.

   If your default view after logging in to SAS Factory Miner is not the **Projects** workspace, then click the Side menu icon , and select **Projects** in the menu that appears under the **SAS Factory Miner** heading to navigate to the **Projects** workspace.



2. Create a new project.

   a. Click the Create a new Factory Miner project icon 

      The New Project window appears.

   b. Specify a name for your project.

   c. Specify a location to store information about your project.

      Note: If a path is already provided, you can use the default path.

   d. Select **PURCHASES** as the data source from the **Data source** menu.

      Note: If **PURCHASES** is not available from the **Data source** menu, you can click **New data source** to specify the **PURCHASES** data source. If you need help with this process or with registering your data source, contact your administrator, or see *SAS Factory Miner 14.1: Administration and Configuration* for more information or for details about registering a data source.

   e. Verify that a check box is selected to partition the data.

      Note: The default setting is to partition the data.

   f. Click **Save**.

      Your project appears and opens on the **Data** tab.

      Note: You might see warnings that notify you to define a segment variable and a target variable.

3. Verify project settings.

   a. Click the Edit Project Settings icon 

      The Edit Project Settings window appears.

   b. On the **Partition Data** tab, you can specify whether you want to use sample training data and partition the data. If you choose to use sample training data, you can select the sampling method, and indicate whether you want to use a percentage or a specific number of observations. If you choose to partition the data, you can specify the percentage of data to be used for training and validation.

In this example, you will use the default settings. Verify that the **PURCHASES** data source will be partitioned into training and validation samples with 70% training and 30% validation.

c. Click the **Rules** tab.

On the **Rules** tab, you can specify the minimum number of observations and the minimum event rate to use in your training data. Segments that do not meet both of these criteria will be excluded. Models are built only for segments that have at least the minimum number of observations and the minimum event rate that is specified.

In this example, you will use the default settings. Verify that the minimum number of observations is set to *100*, and the minimum event rate is set to *0.1*.

d. Click **Save**.

4. Configure variables for the **PURCHASES** data set.

a. Select the check box for the **purchase** variable to highlight it.

b. Click the Edit icon 

A window appears for the selected variable.

c. Select **Target** as the role for the **purchase** variable.

Note: The **purchase** variable's level is **Binary**, and it has a value of **`Yes`** or **`No`**.

d. Click **Save**.

The role of target appears for the **purchase** variable in the table.

e. Select the check box for the **region** variable to highlight it.

f. Click the Edit icon  .

A window appears for the selected variable.

g. Select **Segment** as the role for the **region** variable.

h. Click **Save**.

The role of segment appears for the **region** variable in the table.

Note: The **region** variable has values for 5 regions.

i. Select the check box for the **COUNTY** variable to highlight it.

j. Click the Edit icon  .

A window appears for the selected variable.

k. Select **Nominal** as the level for the **COUNTY** variable.

l. Click **Save**.

The level of nominal appears for the **COUNTY** variable in the table.

m. Select the check box for the **NUMCARS** and the **NUMKIDS** variables to highlight them.

Note: Use **Ctrl** to select both.

n. Click the Edit icon  .

A window appears for the selected variables.

o. Select **Interval** as the level for the selected variables.

p. Click **Save**.

The level of interval appears for the **NUMCARS** and the **NUMKIDS** variables in the table.

5. Click **Build Profile**.

After the profile is built, your view opens to the **Profile** tab. The **Profile** tab contains a table with information about each segment.

	Segment ID ▲	region	Train: Observations	Train: Event Rate	Run
☐	Segment 1	East	1219	0.5004	Include
☐	Segment 2	Midwest	1182	0.5025	Include
☐	Segment 3	Other	62	0.6290	Exclude
☐	Segment 4	South	966	0.5279	Include
☐	Segment 5	West	699	0.4935	Include

SAS Factory Miner tallies the number of observations and the event rate in the validation sample for each segment. You can sort the values of these columns to quickly identify smaller versus larger segments. Notice that Segment 3, with a region of **Other**, has only 62 training observations. Therefore, only a baseline model will be created for that segment.

6. Review the model information.

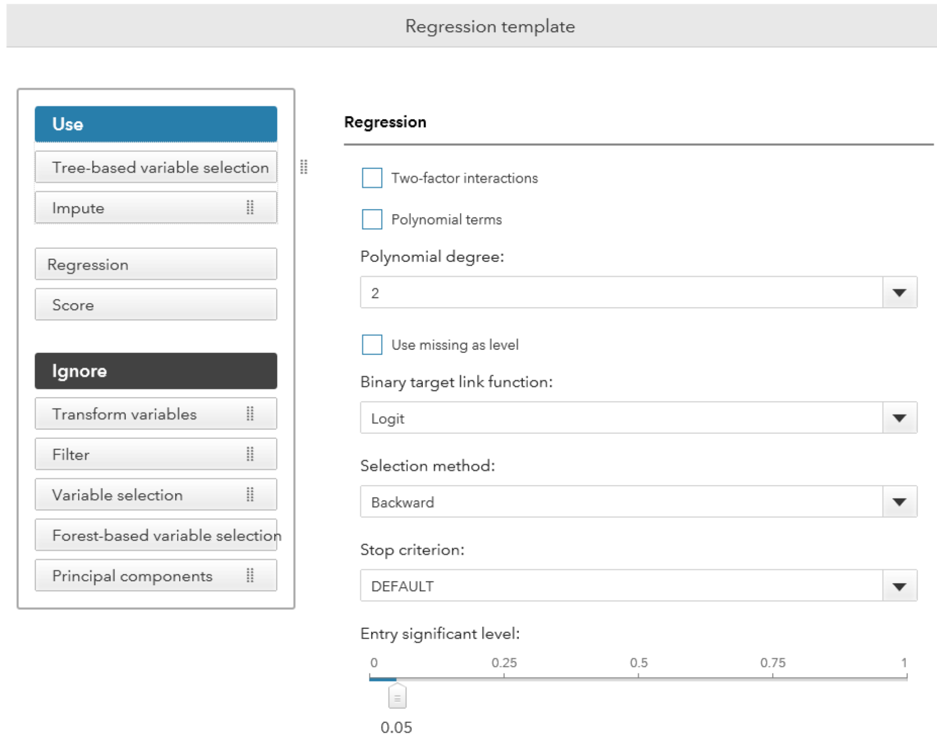
a. Click the **Model Templates** tab.

Several modeling algorithms can be used, including Decision Tree, Neural Network, Random Forest, and Regression. A baseline model is also provided.

b. Select the check box for the Regression template.

c. Click the Edit Properties icon  .

The Regression template window appears. Click the **Regression** button in the **Use** list.



d. On the left side of the screen, examine the **Use** list. You can see that this template performs tree-based variable selection and imputation before building a regression model and scoring the data. The actions on the **Use** list are performed in order of appearance.

   Examine the **Regression** properties. After you select a tab in the **Use** list, you can view and modify property values.

   Categories listed in the **Ignore** list are available, but are not currently selected to be used. For more information about modifying model properties, see "View and Modify the Properties of a Model Template" on page 27.

e. Click **Done** to exit the Regression template window.

7. Click the Run all models icon  .

   This runs each of the modeling algorithms for all segments. When the modeling algorithms have finished running, your view opens to the **Results** tab.

8. Examine the results.

   a. Examine the results in the segment table.

      The segment table lists each of the segments and shows its status. After a few moments, models are built for each of the segments except for segment 3, which had too few observations to use for model building.

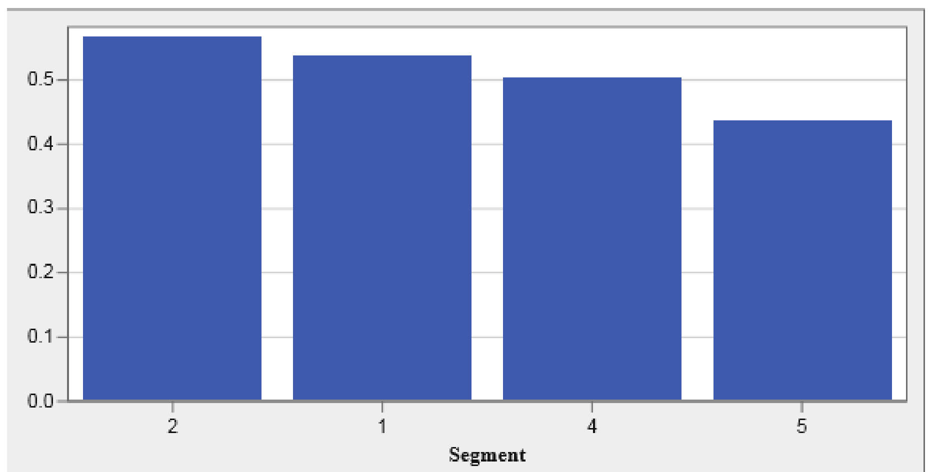
      The table also lists the champion model for each segment and KS value for each champion. For example, you can see that the champion (Neural Network) for the Midwest region had the highest KS value. But the champion for the West region (also Neural Network) had the lowest KS value among the champion models for each region that performed model building.

	Segment ID	Segment (region)	Champion	Override	Validate: Kolmogorov-Smirnov	Validate: Observations	
☑	Segment 1	East	Neural Network template	No	0.5378	524	5
☐	Segment 2	Midwest	Neural Network template	No	0.5679	507	5
☐	Segment 3	Other	Baseline template	No	0.0000	25	1
☐	Segment 4	South	Decision Tree template	No	0.5032	414	5
☐	Segment 5	West	Neural Network template	No	0.4388	300	5

The KS statistic is a measure of the performance of a classification model. That is, the KS statistic is the model's ability to discriminate purchasers from nonpurchasers. KS values range from 0 to 1. Higher values of the KS statistic reflect better models: A value of 0 corresponds to a random model, but a value of 1 represents an ideal model.

b. Review the graphical output.

The KS bar chart accompanies the segment table.



c. Click **Segment 2** in the table.

Note: To open a segment, you can also select the check box for that segment in the table, and then click **Open** on the **Actions** menu.

In the window that appears, a variety of tabular and graphical output is available for the selected model. The default view shows output using the champion model for the segment that you opened. To view output results using another template, you could select the check box for that template in the table.

		Name	Algorithm	Validate: Kolmogorov-Smirnov (KS)	Status
☑	★	Neural Network template	Neural Network	0.5679	Completed
☐		Baseline template	Baseline	0.0000	Completed
☐		Decision Tree template	Decision Tree	0.3559	Completed
☐		Random Forest template	Random Forest	0.2523	Completed
☐		Regression template	Regression	0.1417	Completed

In the above table, you can see that **Neural Network** was the champion model. The champion model has the Champion icon next to it in the table. The champion **Neural Network** model had a Kolmogorov-Smirnov (KS) value of

0.5679. The next best model for segment 2 was the **Decision Tree** model, which had a KS value of 0.3559.

d. Click the Segment Properties icon .

Information about this segment appears in a window.

Segment Properties

Target	purchase
Segment (region)	Midwest
Number of observations	1182

This segment represents the Midwest region, and contains 1182 observations.

e. Click **Close**.

f. Scroll down the graphical output for the **Neural Network** model to see the relative importance values in the Scoring Inputs table. These values illustrate the best drivers of purchases.

g. Click the Expand Report icon for the Scoring Inputs table.

Scoring Inputs

Variable Name	Rank	Relative Importance
COUNTY	1	1.0000
JOB	2	0.3271
AGE	3	0.2689
HOMEVAL	4	0.2645
NUMKIDS	5	0.2510
HEAT	6	0.2134
NUMCARS	7	0.2051
TRAVTIME	8	0.2021
MOBILE	9	0.1225
SNGLMOM	10	0.1068
NTITLE	11	0.0794
GENDER	12	0.0359
MARITAL	13	0.0354
EDLEVEL	14	0.0121

The **COUNTY** variable has a relative importance value of 1.0. For this segment, the variables **JOB** and **AGE** had the next highest relative importance scores with values of 0.3271 and 0.2689, respectively. The variables with the lowest relative importance score for this segment were **MARITAL** and **EDLEVEL** with scores of 0.0354 and 0.0121, respectively.

h. Click the Exit maximized view icon to close the Scoring Inputs window.

Below the Scoring Inputs table is the Fit Statistics table.

i. Click the Expand Report icon for the Fit Statistics table.

Fit statistics for the **purchase** target variable appear in the Fit Statistics window.

Fit Statistics

Target	Statistics Label	Train	Validate
purchase	Classification Cutoff	0.5026	0.5026
purchase	CR Cutoff	0.8979	0.7714
purchase	CR Sample Depth	34.9407	34.5167
purchase	Cumulative Gain (10)	98.9898	82.9176
purchase	Cumulative Gain (15)	98.9898	82.9176
purchase	Cumulative Gain (20)	98.9898	86.8941
purchase	Cumulative Gain (5)	98.9898	82.9176
purchase	Cumulative Lift (10)	1.9898	1.8291
purchase	Cumulative Lift (15)	1.9898	1.8291
purchase	Cumulative Lift (20)	1.9898	1.8689
purchase	Cumulative Lift (5)	1.9898	1.8291
purchase	Cumulative Percentage of Events	71.0437	62.7450

j. Click the Exit maximized view icon to close the Fit Statistics window.

k. Scroll up to see the table of modeling algorithms for this segment.

l. Select another algorithm in the table, such as **Decision Tree**.

Notice how the model graphs and tabular information changes for the selected algorithm.

After you select the check box for one or more models, you can click the **Actions** menu to perform one of the actions listed on the menu that appears. See the list below.

Note: To select multiple models, use **Ctrl**. Some of the following actions, such as accessing the log, are not supported for multiple selection. Other actions, such as comparing multiple models, require that multiple models be selected.

- Run a model
- Stop a model from running
- Compare multiple models
- Delete a model
- Duplicate a model
- Mark a model as a champion
- Unmark a model as a champion
- Rename a model
- Access a report
- Access the log
- Download score code

For more information about exploring model output for a segment, see "Explore Model Output for a Segment" on page 31

m. Click **Project 1**, or whatever you named your project, in the main menu to return to the **Results** tab.

Project 1 > Segment 2

9. Improve a model.

Assume that you want to improve the model for segment 5.

a. Click **Segment 5** in the table.

Information for this segment, which represents the West region, appears in a table.

		Name	Algorithm	Validate: Kolmogorov-Smirnov (KS)	Status
☑	★	Neural Network template	Neural Network	0.4388	Completed
☐		Baseline template	Baseline	0.0000	Completed
☐		Decision Tree template	Decision Tree	0.3187	Completed
☐		Random Forest template	Random Forest	0.2634	Completed
☐		Regression template	Regression	0.1567	Completed

b. Click the Add Model icon + .

The Add Model Template window appears.

c. Select the check box for the **Neural Network template**, and then click **Save**.

d. Click the Edit Model Template icon .

The Neural Network template window appears.

e. Click the **Neural Network** button in the **Use** list.

The properties for the Neural Network template appear in the window.

f. Select **Two layers** as the value for the **Architecture** property.

g. Specify `20` as the value for the **Number of hidden neurons** property.

h. Click **Save**.

i. Select **Run** from the **Actions** menu.

You can view results for your new custom Neural Network model when it has finished running.

		Name	Algorithm	Validate: Kolmogorov-Smirnov (KS)	Status
☑	★	Neural Network template	Neural Network	0.7583	Completed
☐		Baseline template	Baseline	0.0000	Completed
☐		Decision Tree template	Decision Tree	0.3187	Completed
☐		Neural Network template	Neural Network	0.4388	Completed
☐		Random Forest template	Random Forest	0.2634	Completed
☐		Regression template	Regression	0.1567	Completed

The new custom Neural Network template is the champion model for this segment, and has a KS value of 0.7583.

j. Scroll down to see the corresponding relative importance values and fit statistics for this model.

Scoring Inputs

Variable Name	Rank	Relative Importance
COUNTY	1	1.0000
HOMEVAL	2	0.5131
NTITLE	3	0.4589
GENDER	4	0.4507
TRAVTIME	5	0.4243
JOB	6	0.3958
MARITAL	7	0.3817

Fit Statistics

Target	Statistics Label	Train	Validate
purchase	Classification Cutoff	0.4924	0.4924
purchase	CR Cutoff	0.8026	0.6186
purchase	CR Sample Depth	45.0643	45.0000
purchase	Cumulative Gain (10)	102.6086	104.0816
purchase	Cumulative Gain (15)	102.6086	104.0816
purchase	Cumulative Gain (20)	102.6086	104.0816
purchase	Cumulative Gain (5)	102.6086	104.0816

k. Click **Project 1**, or whatever you named your project, in the main menu to return to the **Results** tab.

10. Generate a report.

a. Select the check box for **Segment 5** if it is not already enabled.

b. Select **Report** from the **Actions** menu.

The Report window appears.

c. Select **Champion models** as the content that you want for your report.

d. Select **PDF** as the format of your report.

e. Click **Generate**.

After the report finishes generating, a window appears and prompts you about whether you want to display the report.

f. Click **Yes**.

Note: Depending on your browser settings, you might be prompted about whether you want to open or save the file. If this happens, click **Open** to open the file.

A PDF report appears in a new window.

The following are some examples of tabular and graphical output. You might need to scroll down in the report in order to see them.

Note: The output in your report includes more tables and graphs than are shown below. The following tables and graphs might appear on different pages.

You can see a table that displays the order of components that ran in your model.

Segment ID = 5
region = West

Model Flow

Order	Model Neural Network template
1	TreeVarSel
2	Impute
3	NeuralNetwork
4	Score

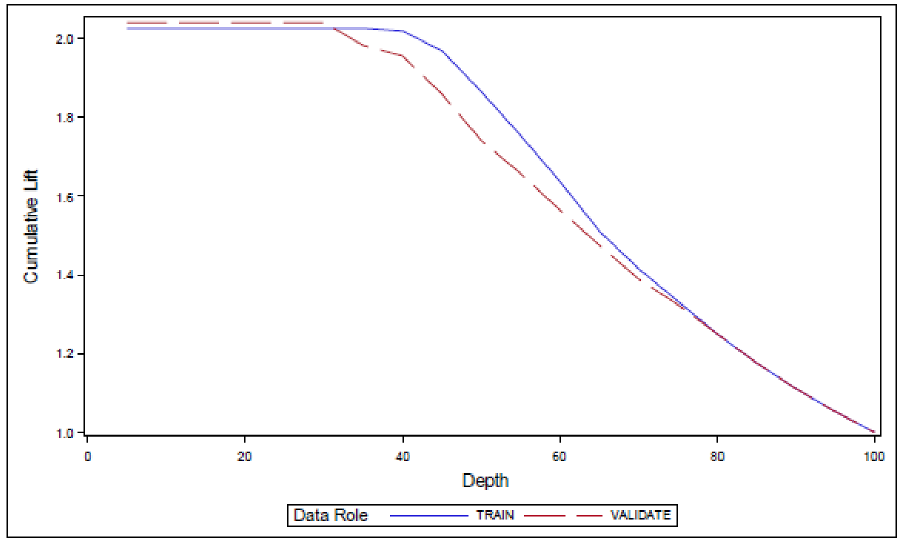
The report also includes the relative importance score for scoring inputs.

Scoring Inputs

Rank	Variable Name	Label	Measurement Level	Relative Importance
1	COUNTY	County Code	NOMINAL	1.00000
2	HOMEVAL	Home Value	INTERVAL	0.51315
3	NTITLE	Name Prefix	NOMINAL	0.45900
4	GENDER	Gender	BINARY	0.45072
5	TRAVTIME		INTERVAL	0.42430
6	JOB		NOMINAL	0.39580
7	MARITAL	Married (y/n)	BINARY	0.38176
8	NUMKIDS		INTERVAL	0.36769
9	AGE	Age	INTERVAL	0.36071
10	HEAT		NOMINAL	0.35078
11	NUMCARS		INTERVAL	0.34477
12	APRTMNT	Rents Apartment	BINARY	0.30187
13	EDLEVEL		NOMINAL	0.23178
14	MOBILE	Occupied <1 yr	BINARY	0.15771
15	SNGLMOM	Single Mom	BINARY	0.00477

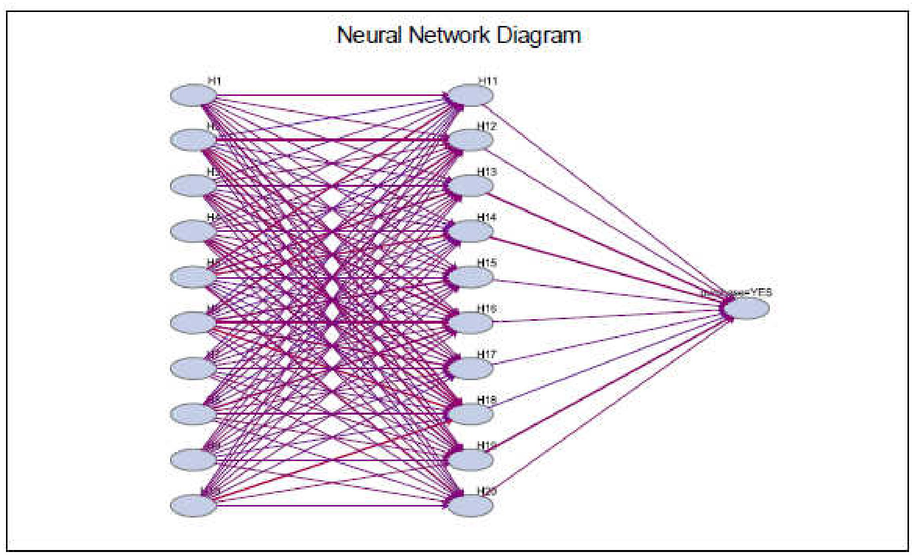
In addition to tables, you can also see graphical output, such as the following cumulative lift assessment plot.

Cumulative Lift Assessment Plot



The following is an example of a neural network diagram.

Neural Network Diagram



g. Return to the SAS Factory Miner application by clicking its browser window.

11. Download the score code for a segment.

    The score code can be used to identify potential purchasers in future data sets.

    a. Select the segment in the table whose score code you want to download.

    b. Select **Download Score Code** from the **Actions** menu.

       The Download Score Code window appears.

    c. Specify that you want to download only champion models.

    d. Click **Download**.

       Note: If your browser prompts you about whether you want to open or save the file, click **Save**. A message appears and states that the download has completed successfully. Click **Open** to view the score code for the champion model for the selected segment.

If you want to download the score code for the entire project, instead of just the selected segment, click the Download Project Score Code icon  in the main menu. In the Download Score Code window that appears, specify that you want to use the champion models only, and then click **Download**.

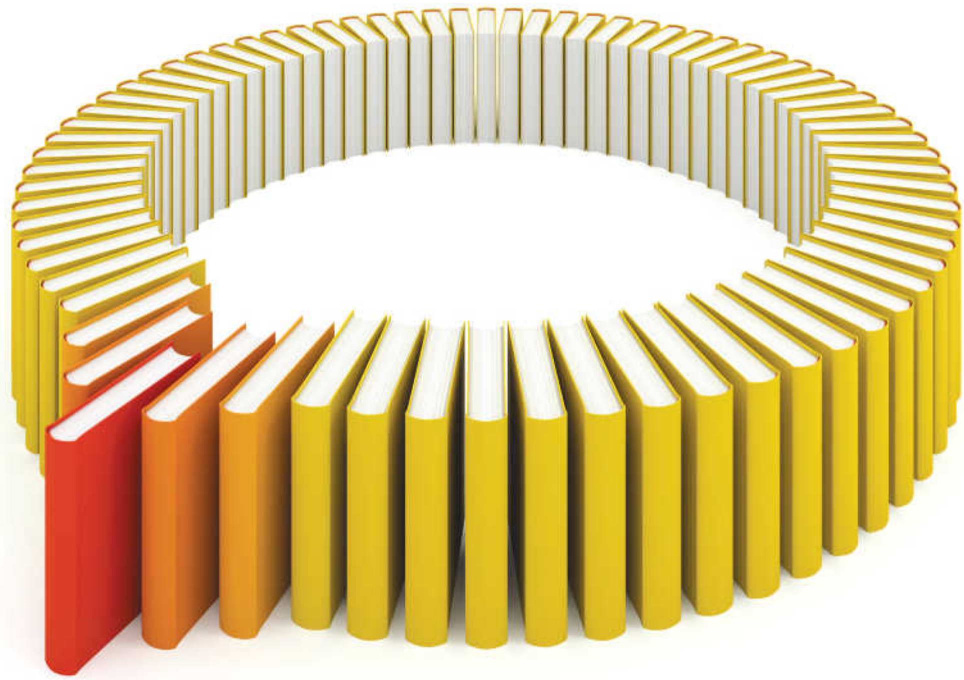
Note: If your browser prompts you about whether you want to open or save the file, click **Save**. A message appears and states that the download has completed successfully. Click **Open** to view the score code for the champion models for each segment.

e. Return to the SAS Factory Miner application by clicking its browser window.

When you have updated the data, you can retrain your models and examine their performance within SAS Factory Miner. You can also register your project to SAS Model Manager, where you can monitor your models' performance and publish a model to a database for scoring. For more information about how to register a project, see “Register a Project” on page 11. For more information about SAS Model Manager, see the *SAS Model Manager: User's Guide*.

For more information about model output and actions that you can perform on the **Results** tab, see “Results Tab” on page 29.

Index



Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore
for additional books and resources.

THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613