SAS® Visual Analytics 8.5: Working with SAS® Visual Data Mining and Machine Learning

About SAS Visual Data Mining and Machine Learning

What Is SAS Visual Data Mining and Machine Learning

SAS Visual Data Mining and Machine Learning is an add-on to SAS Visual Analytics and SAS Visual Statistics that enables you to develop and test models using the in-memory capabilities of SAS servers. SAS Visual Analytics enables you to explore, investigate, and visualize data sources to uncover relevant patterns. SAS Visual Data Mining and Machine Learning extends these capabilities by creating, testing, and comparing models based on the patterns discovered in SAS Visual Analytics. SAS Visual Data Mining and Machine Learning can export the score code or analytic store, before or after performing model comparison, for use with other SAS products and to put the model into production.

Benefits of Using SAS Visual Data Mining and Machine Learning

SAS Visual Data Mining and Machine Learning enables you to rapidly create powerful data mining and machine learning models in an easy-to-use, web-based interface. SAS Visual Statistics provides a model-comparison tool that works on SAS Visual Data Mining and Machine Learning objects. The model-comparison tool enables you to evaluate the relative performance of two or more models against each other and to choose a champion model. A wide variety of model-selection criteria is available. Regardless of whether you compare models, you can export the model. With the exported model, you can easily apply your model to new data.

Modeling Information

Available Models

The following models are available in SAS Visual Data Mining and Machine Learning:

- Bayesian network on page 16 creates a directed, acyclic graphical model in which the nodes represent random variables and the links between the nodes represent conditional dependency between two random variables.
- Factorization machine on page 22 implements a factorization model to effectively handle large, sparse data sets.
- Forest on page 25 is an ensemble model that contains a specified number of decision trees. Each tree is trained on only part of the available data.
- Gradient boosting on page 31 is an iterative approach that creates multiple trees. Each tree is based on an independent sample of the data.
- Neural network on page 37 creates a series of connections that are designed to mimic the biological structures of the human brain.
- Support vector machine on page 45 creates a set of hyperplanes that maximize the distance between two classes.

Overview of Common Information

SAS Visual Data Mining and Machine Learning uses the same basic structures as SAS Visual Statistics. For information about variables, validation data, missing values, filter variables, fit statistics, score code, registering models, integration with Model Studio, and predicted values, see the SAS Visual Statistics documentation.

Nondeterministic Behavior

Some SAS Visual Data Mining and Machine Learning models are created with a nondeterministic process. This means that you might experience different displayed results when you run a model, save that model, close the model, and re-open the report or print the report at a later time. This includes creating a model in SAS Visual Analytics and later viewing it in SAS Report Viewer.

When you create a pipeline for use in Model Studio, a stable version of the model is created.

Export and Score a Model

When you export a SAS Visual Data Mining and Machine Learning model, an analytic store table is created and saved in the Models library of your SAS Cloud Analytics Services (CAS) server. In addition, the SAS DATA step code that is needed to score the model is downloaded to your browser's `Downloads` folder.

Note: The **Neural network** object does not export an analytic store. It exports only SAS DATA step score code.

To export and score a model:

1. Right-click in the model canvas, and select **Export model**. In the Export Model window, specify a base name for the analytic store table. Click **OK**. The name of the analytic store table is case sensitive.
2. Click **Export** to download the SAS DATA step code to your browser's `Downloads` folder.
3. Open the downloaded code. Modify the macro variables as indicated by the code comments:
   - **SOURCE_HOST**: The host name of the CAS server where the analytic store table is saved.
   - **SOURCE_PORT**: The port of the CAS server where the analytic store table is saved.
   - **SOURCE_LIB**: The CAS library where the source data that you want to score is located.
   - **SOURCE_DATA**: The CAS table name of the source data that you want to score.
   - **DEST_LIB**: The CAS library where you want to save the scored data.
   - **DEST_DATA**: The CAS table name for the scored data.

   Note: See SAS Environment Manager for information about your CAS server:

   In the upper left corner of the window, click the ☰ icon and select **Manage Environment**.

   In the left pane, click to access the Servers window. The server name, host, and port are listed in this table.
4. Run the code to score your new data.

Autotuning

To create a good statistical model, you must decide which model and hyperparameters to use. You can try a trial-and-error approach or you can rely on experience and personal preference. However, neither of these guarantees that you will find the best model for your data. SAS Visual Data Mining and Machine Learning can automate the process of determining optimum model hyperparameters.

Autotuning is the process of automatically and algorithmically adjusting model hyperparameters to create a set of competing versions of one particular model. Those competing versions are then compared to determine which set of hyperparameters produces the best model.

When autotuning a model, you can specify the following options in the Autotune Hyperparameters window by clicking **Autotune** in the **Options** pane:

Maximum seconds
The maximum amount of time the model will run in seconds. You must specify an integer between 1 and 2,147,483,647. The autotuning algorithm always runs for a minimum of 60 seconds, even if you specify a value smaller than 60.

Maximum iterations
At each iteration, a set of models is created to be evaluated against each other. This property determines the number of sets of models that are created. You must specify an integer between 1 and 2,147,483,647.

Maximum evaluations
The maximum number of different models that are created for evaluation. You must specify an integer between 3 and 2,147,483,647.

Every model available for SAS Visual Data Mining and Machine Learning in SAS Visual Analytics can be autotuned. For all models, you can limit the maximum amount of time, training iterations, and model evaluations performed during autotuning. This prevents server resources from being monopolized by a single user or task. Autotuning does run a slight risk of overfitting a model, especially when no partitioning is used.

You might encounter unexpected behavior if you cancel an incomplete autotuning session. For example, when you cancel an autotune query, and then restart the autotune query, the restarted query will not finish. Also, the value of some autotuned options might update in the **Options** pane, although the model is still using the original values.

The following table describes the hyperparameters that you can tune:

Bayesian Network				
	Parameter	Minimum Value	Maximum Value	Notes
	Number of bins	2	100	Integers
	Prescreen predictors	not applicable	not applicable	Yes and No are evaluated
	Use variable selection	not applicable	not applicable	Yes and No are evaluated
	Independence test statistic	not applicable	not applicable	All possible test statistics are evaluated
	Significance level	0	1	Real Values
	Network structure	not applicable	not applicable	All possible network structures are evaluated
	Maximum number of parents	1	16	Integers
	Choose best structure and number of parents	not applicable	not applicable	Yes and No are evaluated
	Parenting method	not applicable	not applicable	All possible parenting methods are evaluated
Factorization Machine				
	Parameter	Minimum Value	Maximum Value	Notes

	Factor count	5	30	Only multiples of 5 are evaluated
	Maximum iterations	10	200	Only multiples of 10 are evaluated
	Learn step	0.000001	1	Only multiples of 10 are evaluated
Forest				
	Parameter	Minimum Value	Maximum Value	Notes
	Number of trees	20	150	Integers
	Bootstrap	0.1	0.9	Real values
	Number of predictors to split nodes	1	100	Chooses the lesser of 100 or maximum number of inputs
	Maximum levels	2	51	Integers
	Leaf size	1	2,147,483,647	Integers
	Predictor bins	2	500	Integers
Gradient Boosting				
	Parameter	Minimum Value	Maximum Value	Notes
	Number of trees	20	150	Integers
	Learning rate	0.01	1	Real values
	Subsample rate	0.1	1	Real values
	Lasso	0	10	Real values
	Ridge	0	10	Real values
	Number of predictors to split nodes	1	Number of inputs	Integers
	Maximum levels	2	51	Integers
	Leaf size	1	2,147,483,647	Integers
	Predictor bins	2	500	Integers
	Auto-stop method	not applicable	not applicable	All possible methods are evaluated. Available only when partitioning is used.
	Auto-stop iterations	1	10,000	Integers. Available only when partitioning is used.
	Tolerance value	0	1	Real values. Available only when partitioning is used.

Neural Network				
	Parameter	Minimum Value	Maximum Value	Notes
	Number of hidden layers	0	5	Maximum value is 2 when **Optimization method** is set to **LBFGS**.
	Number of neurons	1	100	Chooses the lesser of 100 and three times the number of inputs
	L1	0	10	Real values
	L2	0	10	Real values
	Learning rate	0.000001	0.1	Real values. Available only when **Optimization method** is set to **SGD**.
	Annealing rate	1.0e-13	0.01	Real values. Available only when **Optimization method** is set to **SGD**.
	Maximum iterations	10	200	Integers
	Auto-stop method	not applicable	not applicable	All possible methods are evaluated. Available only when partitioning is used.
	Auto-stop iterations	1	10,000	Integers. Available only when partitioning is used.
	Goal value	0.000001	1	Real values. Available only when partitioning is used.
Support Vector Machine				
	Parameter	Minimum Value	Maximum Value	Notes
	Kernel function	not applicable	not applicable	All possible kernel functions are evaluated
	Penalty value	1.0e-10	100	Real values
Decision Tree				
	Parameter	Minimum Value	Maximum Value	Notes
	Maximum levels	2	20	Integers
	Leaf size	1	2,147,483,647	Integers
	Predictor bins	2	1500	Integers

Getting Started with SAS Visual Data Mining and Machine Learning

Overview

This is a brief overview of using SAS Visual Data Mining and Machine Learning to derive a new variable, create three different models, and compare those models. This example uses the Framingham Heart Study data set, located at http://support.sas.com/documentation/onlinedoc/viya/examples.htm, to compare the performance of a forest, gradient boosting, and neural network. The goal is to predict a person's age of death based on a collection of health factors. These factors include gender, weight, height, whether the person is a smoker, blood pressure, and more. The focus of this example is how to use SAS Visual Data Mining and Machine Learning, not how to build the best model.

Download the Sample Data

1. In a web browser, navigate to http://support.sas.com/documentation/onlinedoc/viya/examples.htm.
2. Download the file **heart.csv** to your local machine.

Create the Report

This example assumes that you have already signed in to SAS Drive.

1. In SAS Drive, click ☰ and select **Explore and Visualize**. SAS Visual Analytics opens, and you can open a data source, create a new report, or load a report.
2. Click the **Start with Data** button in the home pane to load your data. The Choose Data window appears.
3. On the **Import** tab, click **Local files**, and then click **Local file**. Navigate to the location where you saved **heart.csv**, select **heart.csv**, and click **Open**.
4. In the Choose Data window, click **Import Item**. After the table is successfully imported, click **OK**.
5. Rename the project by saving it. By default, the report is named `Report 1`, which is displayed at the top of the page.

   In the upper right corner of the window, click ⋮, and then select **Save**. Navigate to a location where you have Write permission. In the **Name** field, enter `Heart Study`, and click **Save**.

   Typically, you can save your work in `My Folder`.

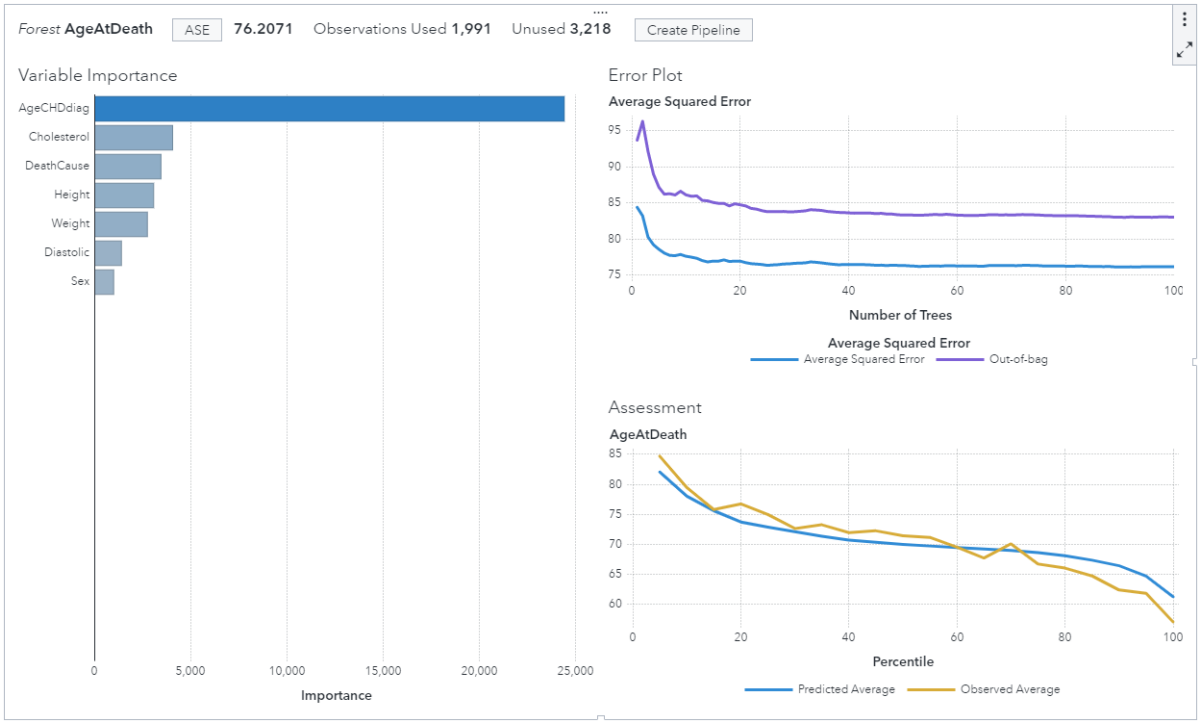
For more information about reports, see *SAS Visual Analytics: Designing Reports*

Create a Forest

1 In the left pane, click to select an object. Drag the icon onto the canvas to create a forest.

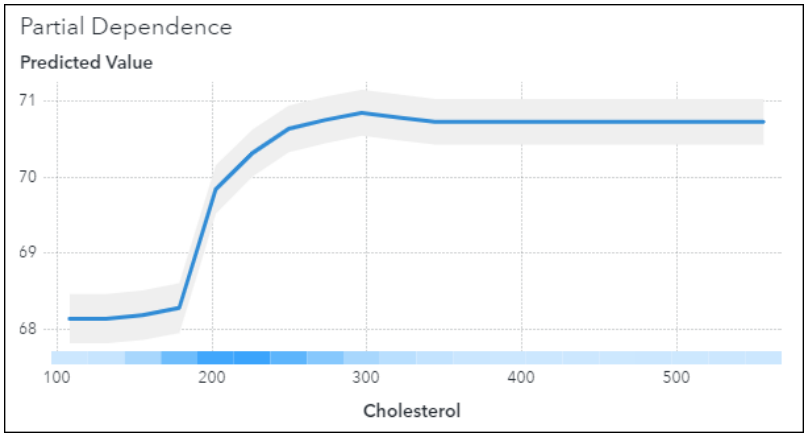
2 Click the **Assign Data** button in the middle of the report canvas. For **Response**, click **Add**, and select **AgeAtDeath**.

3 For **Predictors**, click **Add**, and select **DeathCause**, **Sex**, **AgeCHDdiag**, **Cholesterol**, **Diastolic**, **Height**, and **Weight**. Click **OK**, and then click **Close**. The forest automatically updates.



4 Right-click in the Error plot, and then select **Partial dependence**.

5 Right-click in the Partial Dependence plot, and then select **Cholesterol**.



Notice how as the cholesterol value increases, the predicted value of the age of death also increases. Once the value of cholesterol reaches 300, increases in cholesterol have little additional impact on the model prediction.

The heat map at the bottom of the plot indicates that the majority of observations that were sampled to create the plot have cholesterol values in the range 175–250.

6 Click ↙↗ to enter maximize mode.

In the details tables, you can view more detailed information about your model. This includes the variable importance values and error metric results.

7 Click ↘↖ to exit maximize mode.

8 Click to save the report.

Create a Gradient Boosting Model

1 Click + to add a new page.

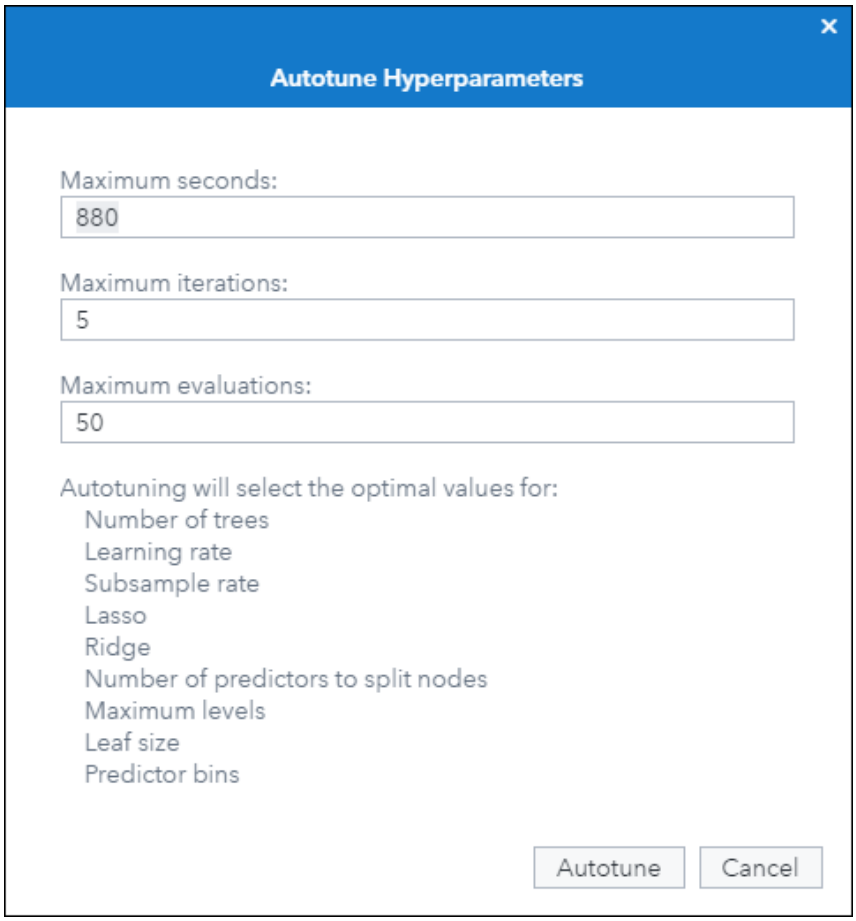
2 In the left pane, click to select an object. Drag the icon onto the canvas to create a gradient boosting model.

3 In this example, the variable of interest is **AgeAtDeath**.

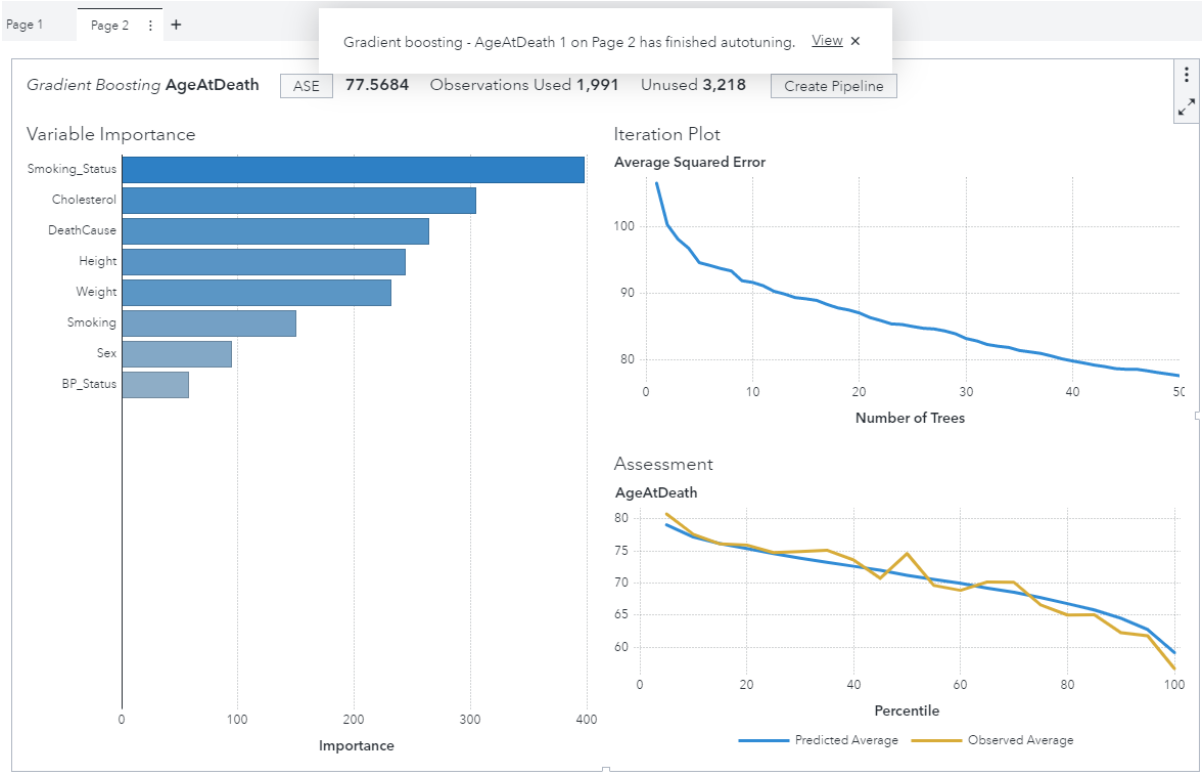
Click the **Assign Data** button in the middle of the report canvas. For **Response**, click **Add**, and select **AgeAtDeath**.

4 For **Predictors**, click **Add**, and select **BP_Status**, **DeathCause**, **Sex**, **Smoking_Status**, **Cholesterol**, **Height**, **Smoking**, and **Weight**. Click **OK**, and then click **Close**. The gradient boosting model automatically updates.

5 In the right pane, click , and then click **Autotune**.



When you autotune your model, SAS Visual Data Mining and Machine Learning automatically and algorithmically determines the optimal values for the specified properties. In the Autotune Hyperparameters window, click **Autotune**.



The Assessment plot indicates that the observed average and predicted average vary slightly across the bins.

6 Save the report.

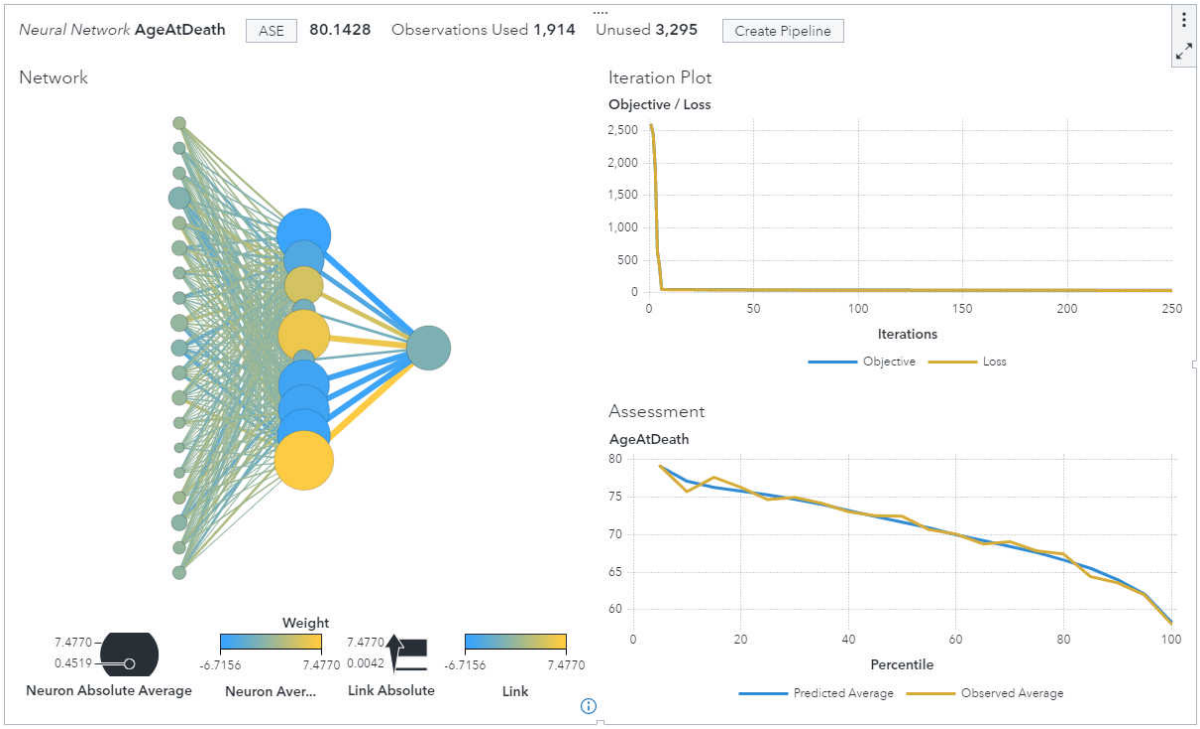
Create a Neural Network

1 Click + to add a new page.

2 In the left pane, click to select an object. Drag the icon onto the canvas to create a neural network.

3 Click the **Assign Data** button in the middle of the report canvas. For **Response**, click **Add**, and select **AgeAtDeath**.

4 For **Predictors**, click **Add**, and select **BP_Status**, **DeathCause**, **Sex**, **Smoking_Status**, **Cholesterol**, **Height**, **Smoking**, and **Weight**. Click **OK**, and then click **Close**. The neural network automatically updates.



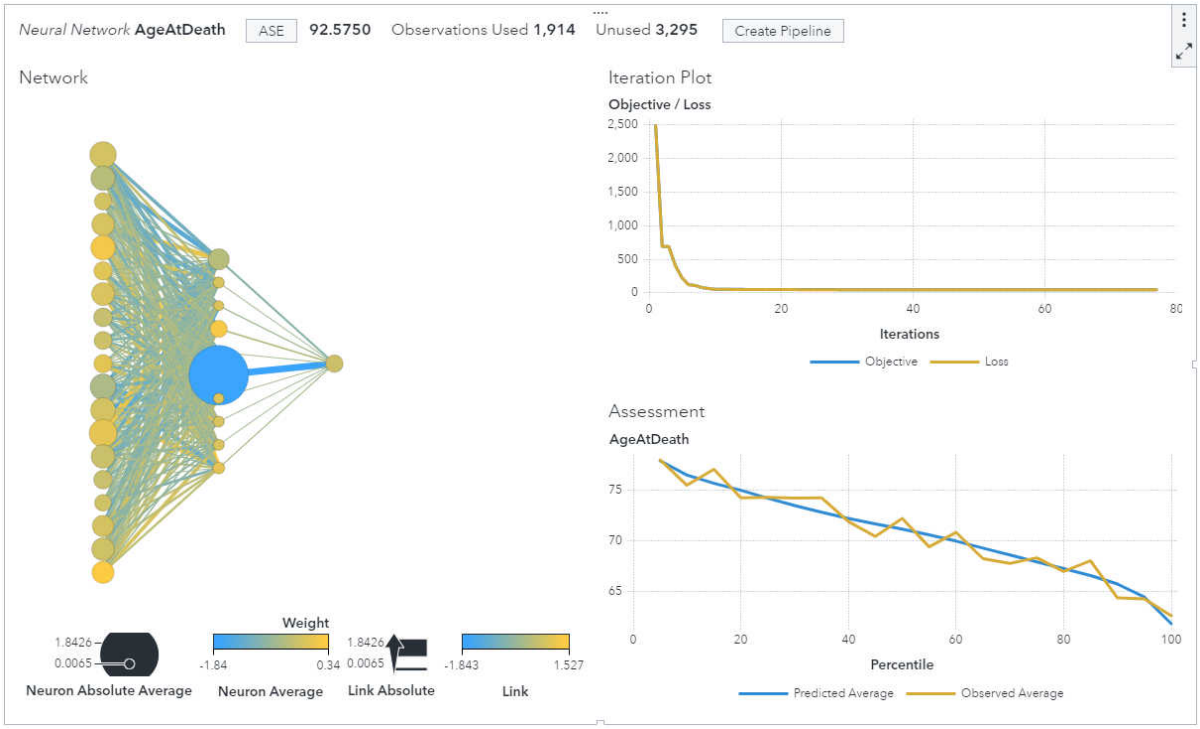
5 In the right pane, click . The **Distribution** option enables you to specify the distribution of the response variable and to build a model based on that distribution. The default distribution is **Normal**.

To determine whether the normal distribution applies to the response variable, click + to add a new page. From the **Data** pane, drag and drop **AgeAtDeath** onto the canvas. A histogram is automatically created.

6 Notice that **AgeAtDeath** is not normally distributed and is slightly skewed left. Click **Page 3** to navigate back to the neural network.

7 Although the distribution is not exactly Poisson, use the Poisson distribution for this example. For the **Distribution** option, select **Poisson**.

Note: You are encouraged to repeat this example with different distributions and standardization techniques and compare their performances and to familiarize yourself with SAS Visual Data Mining and Machine Learning.

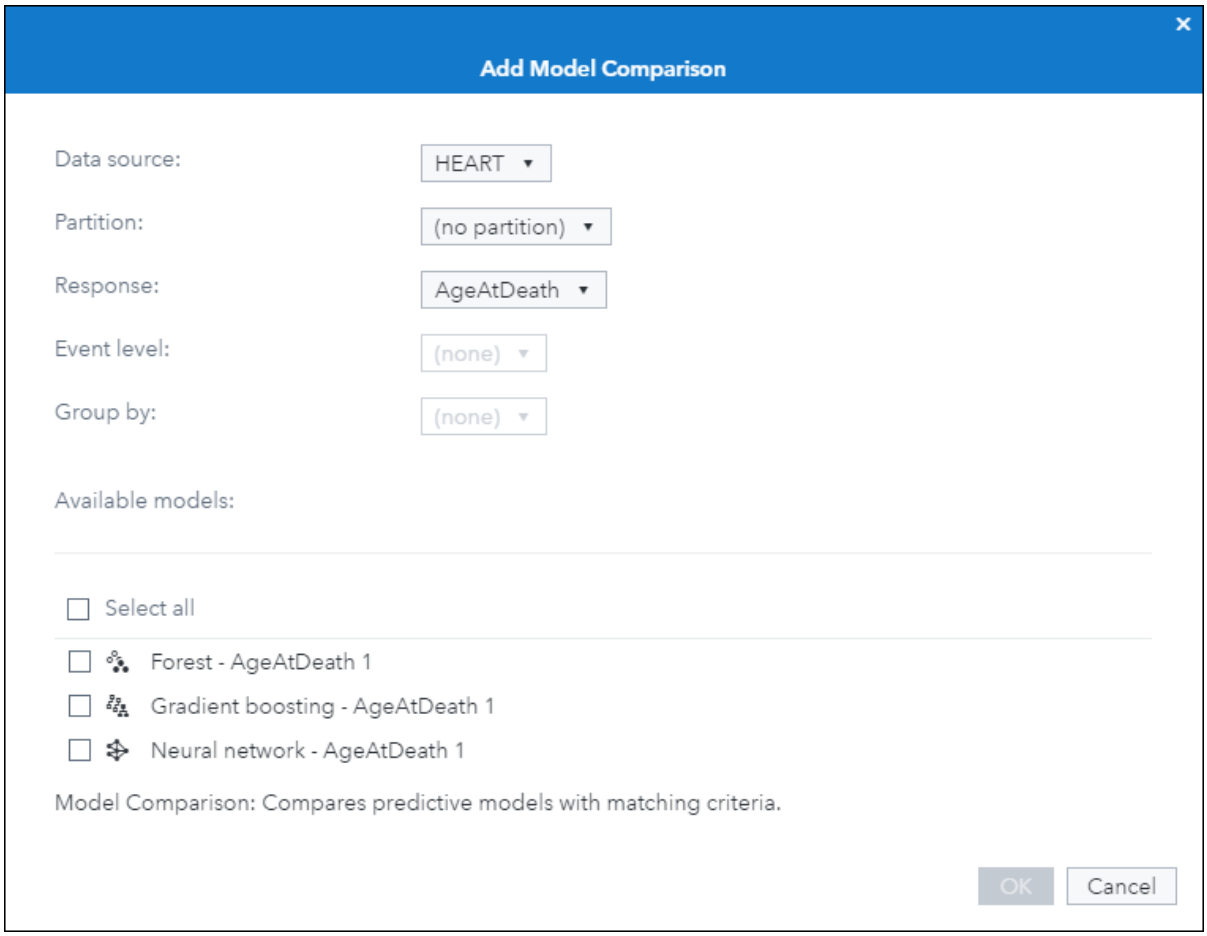


8 Save the report.

Perform a Model Comparison

1 Click + to add a new page.

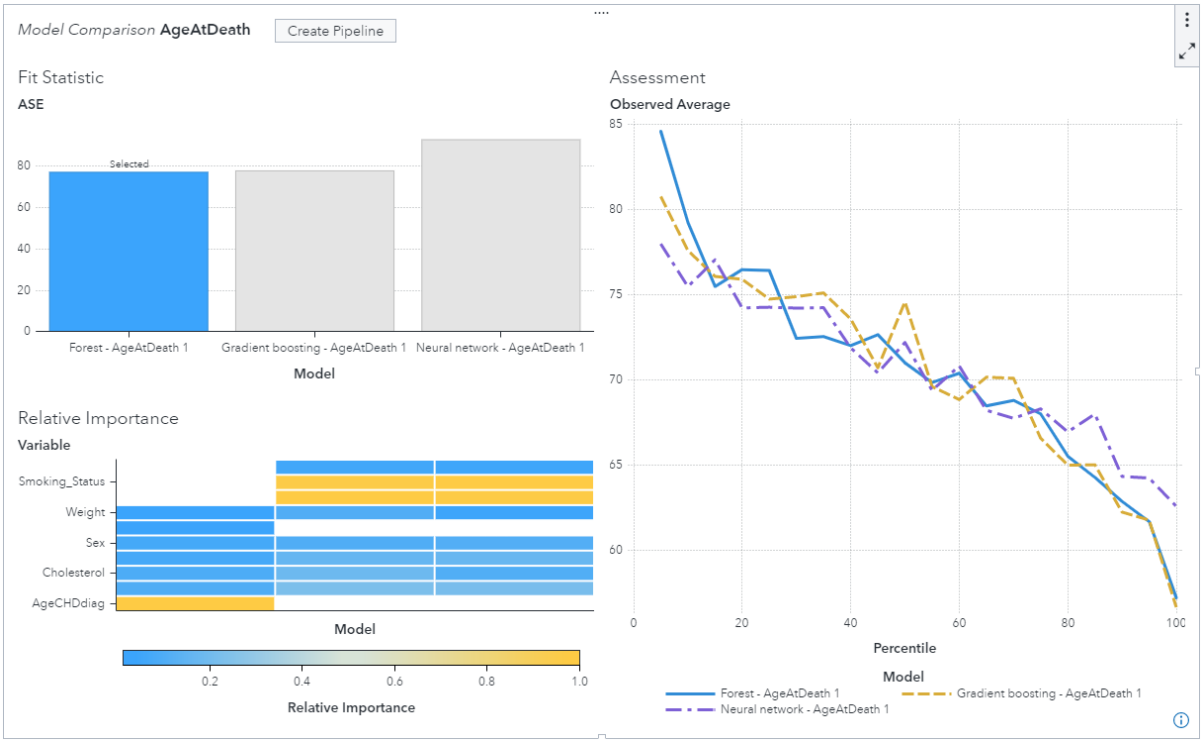
2 In the left pane, click to select an object. Drag the icon onto the canvas to create a model comparison.



The **Response** variable is already set to **AgeAtDeath**, and **Event level** and **Group by** are unavailable.

With these settings, the available models are **Forest – AgeAtDeath 1**, **Gradient boosting – AgeAtDeath 1**, and **Neural network – AgeAtDeath 1**.

3 Select **Select all**, and click **OK**.



By default, the fit statistic average square error, **ASE**, is used to compare the models. The **Fit statistic** option enables you to change the fit statistic that is used to compare the models. The other available fit statistics are **SSE** and **Observed Average**. Because smaller values are preferred, the forest model is chosen as the champion when **ASE** or **SSE** is the criterion. The models are very similar.

When the fit statistic is **Observed Average**, the **Percentile** slider is available. This slider specifies the percentile where the observed average and predicted average are compared. The champion model varies by percentile.

If you view the Assessment plot, both the Observed Average plot and the Predicted Average plot show that in some cases the models are significantly different.

4 Now that you have a champion model, you can export the model score code for that model to score new data.

   a Right-click in the model canvas, and select **Export selected model**.

   b In the Export Model window, specify a base name for the analytic store table. Click **OK**. A window appears that contains the SAS code to export.

   c Click **Export**. The code is downloaded to your local machine.

   The model is stored as an analytic store. For more details, see Export and Score a Model on page 3.

5 Save the report.

Working with Bayesian Networks

Overview of Bayesian Networks

A Bayesian network is a directed, acyclic graphical model in which the nodes represent random variables and the links between the nodes represent conditional dependency between two random variables. The structure of a Bayesian network is based on the conditional dependency between the variables. As a result, a conditional probability table is created for each node in the network. Because of these features, Bayesian networks can be effective predictive models during supervised data mining. The following Bayesian network structures are available:

- Naive is a Bayesian network that connects the target variable to each input variable. There are no other connections between the variables because the input variables are assumed to be conditionally independent of each other.
- Tree-augmented naive is a Bayesian network that connects the target variable to each input variable and connects the input variables in a tree structure. The tree that connects the input variables is based on the maximum spanning tree algorithm. In a tree-augmented naive network, each node can have at most two parents, one of which must be the target variable.
- Parent-child is a Bayesian network that connects the target variable to each input variable. However, input variables can be either a parent of the target variable or a child of the target variable. The Bayesian information criterion (BIC) is used to determine whether an input variable is a parent of the target variable or a child of the target variable.
- Markov blanket is a Bayesian network that creates a set of connections between the target variable and the input variables. It also permits connections between certain input variables. Only the children of the target variable can have an additional parent node. All other variables are conditionally independent of the target variable. Therefore, they do not affect the classification model. BIC is used to determine whether an input variable is a parent of the target variable or a child of the target variable.

Create a Bayesian Network

To create a Bayesian network, complete the following steps:

1. In the left pane, click to select an object. Drag the icon onto the canvas to create a Bayesian network.
2. Click in the right pane. Specify a single category variable as the **Response** variable.
3. Specify at least one variable for **Predictors**. You can specify only measure or category variables for **Predictors**.
4. (Optional) Specify **Partition ID**.

Bayesian Network Options

The following options are available for the Bayesian network:

General

Event level
enables you to choose the event level of interest. When your category response variable contains more than two levels, SAS Visual Data Mining and Machine Learning treats all observations in the level of interest as an event and all other observations as nonevents.

Autotune
enables you to specify the constraints that control the autotuning algorithm. The constraints determine how long the algorithm can run, how many times the algorithm can run, and how many model evaluations are allowed.

The autotuning algorithm selects the values for the following parameters that produce the best model:

- **Number of bins**
- **Prescreen predictors**
- **Use variable selection**
- **Independence test statistic**
- **Significance level**
- **Network structure**
- **Maximum number of parents**
- **Choose best structure and number of parents**
- **Parenting method**

Include missing
specifies whether missing values are included in the model. Missing category values are treated as their own measurement level. Missing measure variables are imputed to the mean value for that variable.

For a Bayesian network that performs variable selection and excludes missing values, the number of observations used can differ based on the presence of a partition data item. For models that are partitioned, the nonmissing values of all predictors are used to train the model. However, the validation action performed by the CAS server rejects the missing values only in the selected predictors.

Number of bins
specifies the number of bins for measure variables. Possible values range from 2 to 100.

Prescreen predictors
specifies whether to prescreen variables using independence tests between the target and each input variable. By default, this option is selected.

Use variable selection
specifies whether to select variables using conditional independence tests between the target and each input variable based on the network. By default, this option is deselected.

Independence test statistic
specifies the statistic used for the independence test. Possible values are **Chi-square**, **G-square**, and **Chi and G-square**. Select **Chi and G-square** to use both independence test statistics. Both statistics must be satisfied for the independence test. The default value is **G-square**.

Significance level
specifies the significance level (*p*-value) for independence testing using Chi-square and G-square. The smaller the value you specify, the fewer variables selected. The default value is 0.2.

Network structure
specifies the network structure. You can select multiple network structures. The following are available:

- **Naive**
- **Tree-augmented naive**
- **Parent-child**
- **Markov blanket**

If you want to choose the best structure among these structures, you can select multiple values in any combination. For more details about these structures, see the overview of this chapter. By default, all structures are selected.

Maximum number of parents
specifies the maximum number of parents that are allowed for each node in the network structure. Possible values are in the range 1–16. The default value is 5.

Choose best structure and number of parents
specifies whether to choose the best number of parents starting with 1 and ending with the value of the **Maximum number of parents**. By default, this option is selected. When multiple network structures are selected, this option is always enabled.

Parenting method
specifies the parenting method of each node. Possible values are **One parent** and **Set of parents**. Select **One parent** to add the best possible candidate as a parent of the node. Select **Set of parents** to test a set of variables among the possible candidates to be the parents of each node. Then, the best set of variables are added as the parents of the node. The default value is **Set of parents**.

Partial Dependence

Number of observations
specifies the maximum number of observations to sample from the input data. These observations are used to generate the model predictions that are displayed in the Partial Dependence plot.

Number of bins
specifies the number of bins to use to categorize measure variables in the Partial Dependence plot.

Assessment

Number of bins
specifies the number of bins to use in the assessment. You must specify an integer value in the range 5–100.

Prediction cutoff
specifies the value at which a computed probability is considered an event.

Statistic percentile
specifies the depth for the percentile bins that are used to calculate the observed average, lift, cumulative lift, cumulative percentage captured, cumulative percentage events, and gain.

Tolerance
specifies the tolerance value that is used to determine the convergence of the iterative algorithm that estimates the percentiles. Specify a smaller value to increase the algorithmic precision.

Bayesian Network Model Display Options

The following display options are available for Bayesian networks:

General

Plot layout
specifies how objects are displayed on the canvas. **Fit** aligns all of the objects on the canvas automatically. **Stack** displays the objects as if they are in a slide deck. Only one object is displayed at a time. When **Stack** is specified, a control bar instead of a scroll bar lets you move between objects.

Statistic to show
specifies which assessment statistic to display in the model. If you are using a partition variable, the following fit statistics are available for each partition. The object toolbar contains submenus for each partition type that is available (training, validation, and test).

- **C Statistic**
- **Cumulative % Captured**
- **Cumulative % Events**
- **Cumulative Lift**
- **F1 Score**
- **FDR**
- **FPR**
- **Gain**
- **Gamma**
- **Gini**
- **KS (Youden)**
- **Lift**
- **Misclassification Rate**
- **Misclassification Rate (Event)**
- **Tau**

See Fit Statistics for more information about the fit statistics that are available.

Variables in Network / Partial Dependence Plot

Plot to show
specifies whether the Variables in Network plot or the Partial Dependence plot is displayed.

X axis
specifies which variable is displayed in the Partial Dependence plot.

Assessment Plots

Display test partition
specifies whether to display the assessment plot for the test partition. This option is available only when you specify a partition ID with a test partition.

Plot to show
specifies which assessment plot is displayed. Select **Confusion matrix**, **Lift**, **ROC**, or **Misclassification**.

Y axis
specifies whether a standard Lift plot or a Cumulative Lift plot is displayed.

Legend visibility
specifies whether the legend is displayed in the confusion matrix, Lift plot, ROC plot, or Misclassification plot.

Bayesian Network Results

Network Plot

The Network plot displays the Bayesian network with the best misclassification rate. The selected Bayesian network is indicated in the Model Selection plot with a ★ icon.

Variables in Network Plot

The Variables in Network plot shows the variable used to create the model, sorted by its BIC score.

Partial Dependence Plot

The Partial Dependence (PD) plot helps you understand how the model's predictions depend on the value of a given predictor. This plot helps you determine how the value of the response changes as the value of a given predictor changes. On the Y axis, categorical response models display the predicted probability of the event, and measure response models display the predicted response value. In both cases, the values are the average of the observations in that bin. For multinomial models, you can modify the event level to examine other relationships. The X axis displays the values of the specified predictor. The predictor that is shown by default is the predictor with the largest importance value. The plot displays a line that shows the point estimate and a 95% confidence band. The heat map at the bottom of the plot indicates the density of the observations that were sampled. Some bins might have zero observations. The Partial Dependence plots are generated by using a sample of all observations, even if you are using a partition variable.

Model Selection Plot

The Model Selection plot displays how the misclassification rate of the Bayesian network changes as the value of **Maximum number of parents** changes. Misclassification rates are grouped by network structure, and the best model is indicated with a ★ icon. The best model is displayed in the Network plot.

By definition, **Tree-augmented naive** networks contain exactly two parents. This section of the Model Selection plot will always contain exactly one plotted value.

Assessment Plot

The confusion matrix displays the classification results for categorical response models. After a model is created, each observation has an observed value and a predicted value. The total number of each observed-predicted pair is calculated. The confusion matrix displays how many observations fall into each pair. A perfect model always predicts the observed value, and all values lie on the diagonal of the matrix. Any off-diagonal values represent a misclassification. Cells are shaded based on the proportion of the value in each cell to the number of observed values for that level. Darker shaded

cells show the concentration of the predictions for the observed level. For a binary response, the information in the confusion matrix is identical to the Misclassification plot.

Lift is the ratio of the percent of captured responses within each percentile bin to the average percent of responses for the model. Similarly, *cumulative lift* is calculated by using all of the data up to and including the current percentile bin.

A receiver operating characteristic (ROC) chart displays the ability of a model to avoid false positive and false negative classifications. A false positive classification means that an observation has been identified as an event when it is actually a nonevent (also referred to as a Type I error). A false negative classification means that an observation has been identified as a nonevent when it is actually an event (also referred to as a Type II error).

The *specificity* of a model is the true negative rate. To derive the false positive rate, subtract the specificity from 1. The false positive rate, labeled **1 – Specificity**, is the X axis of the ROC chart. The *sensitivity* of a model is the true positive rate. This is the Y axis of the ROC chart. Therefore, the ROC chart plots how the true positive rate changes as the false positive rate changes.

A good ROC chart has a very steep initial slope and levels off quickly. That is, for each misclassification of an observation, significantly more observations are correctly classified. For a perfect model, one with no false positives and no false negatives, the ROC chart would start at (0,0), continue vertically to (0,1), and then horizontally to (1,1). In this instance, the model would correctly classify every observation before a single misclassification could occur.

The ROC chart includes two lines to help you interpret it. The first line is a baseline model that has a slope of 1. This line mimics a model that correctly classifies observations at the same rate it incorrectly classifies them. An ideal ROC chart maximizes the distance between the baseline model and the ROC chart. A model that classifies more observations incorrectly than correctly would fall below the baseline model. The second line is a vertical line at the false positive rate where the difference between the Kolmogorov-Smirnov values for the ROC chart and baseline models is maximized.

The Misclassification plot displays how many observations were correctly and incorrectly classified for each value of the response variable. When the response variable is not binary, the Bayesian network model considers all levels that are not events as equal. A significant number of misclassifications could indicate that your model does not fit the data.

Details Table

When you click ↙↗ in the object toolbar of the canvas, the details table is displayed at the bottom of the canvas. The details table contains the following information:

Model Information
Gives an overview of the model.

Number of Observations
A summary of the number of observations used for various model-building tasks.

Fit Statistics
Lists all of the fit statistics.

Model Selection
Provides a summary of the model selection results based on various modeling parameters.

Variable Selection
Provides a summary of the variable selection results at each step in the selection process.

Variable Levels
Provides a mapping of the categorical variable levels to the index values used in the model.

Variable Order
Lists the variables used in the model, sorted by their BIC scores.

Confusion Matrix
Provides a summary of the correct and incorrect classifications for the model that is used to generate the confusion matrix.

Lift
Lists the binned assessment results that are used to generate the Lift plot.

ROC
Lists the results that are used to generate the ROC plot.

Misclassification
Provides a summary of the correct and incorrect classifications for the model.

Assessment Statistics
Provides the value of any assessment statistic computed for the model.

Working with Factorization Machines

Overview of Factorization Machines

A factorization machine is a predictive model that creates a factorization model. By modeling all variable interactions with factorized parameters, factorization machines are able to handle large, very sparse data and can be trained in linear time.

A common application of factorization machines is for recommendation engines. A factorization machine can consider all items that a user has rated, and then predict ratings for other items.

Create a Factorization Machine

1. In the left pane, click to select an object. Drag the icon onto the canvas to create a factorization machine.
2. Click in the right pane. Specify a single measure variable as the **Response** variable.
3. Specify at least two category variables and any number of measure variables for **Predictors**.

Factorization Machine Options

The following options are available for the factorization machine:

General

Autotune
enables you to specify the constraints that control the autotuning algorithm. The constraints determine how long the algorithm can run, how many times the algorithm can run, and how many model evaluations are allowed.

The autotuning algorithm selects the **Factor count**, **Maximum iterations**, and **Learn step** values that produce the best model.

Factor count
specifies the number of factors to use for each predictor level during the fitting process. You must specify an integer value in the range 1–100.

Nonnegative factorization
forces nonnegative factors with bias terms to be set to zero. This can improve performance when the training data is very sparse.

Maximum iterations
specifies the maximum number of optimization iterations that are used before model training stops.

Learn step
specifies the minimum change in the loss function for model estimation to proceed.

Assessment

Number of bins
specifies the number of bins to use in the assessment. You must specify an integer value in the range 5–100.

Statistic percentile
specifies the depth for the percentile bins that are used to calculate the observed average.

Tolerance
specifies the minimum change in the optimization function value for optimization to proceed.

Factorization Machine Model Display Options

The following display properties are available for the factorization machine:

General

Plot layout
specifies how the subplots within objects are displayed on the canvas. **Fit** aligns all of the subplots on the canvas automatically. **Stack** displays the subplots as if they are in a slide deck where only one plot is displayed at a time. When **Stack** is specified, a control bar instead of a scroll bar lets you move between the subplots.

Statistic to show
specifies which assessment statistic to display in the model. The following statistics are available:

- **ASE**
- **Observed Average**
- **SSE**

See Fit Statistics for more information about the fit statistics that are available.

Recommendations Plot

Show recommendations for
For each category variable, the factorization machine shows either the most recommended or least recommended categories. When **Nonnegative factorization** is enabled, the recommendation is sorted by the sum of all factors.

Count
specifies how many recommended categories are displayed.

Scored Response / Relative Importance

Plot to show
specifies which assessment plot is displayed.

Use histogram
specifies whether the Scored Response plot is a histogram.

Legend visibility
specifies whether the legend is displayed in the Scored Response plot or the Relative Importance plot.

Assessment Plot

Legend visibility
specifies whether the legend is displayed in the Assessment plot

Factorization Machine Results

Iteration Plot

The Iteration plot displays the change in objective function value at each step of the model creation process. The vertical lines in the plot represent the first inner iteration of each performance iteration. The objective function value might increase at each vertical line, but it should always decrease within a performance iteration.

Scored Response

The Scored Response plot plots the computed responses against the true, observed values. You can view this information as a histogram.

Relative Importance

The Relative Importance plot plots the importance value of each input variable. The variables are ranked using their first-split log worth when applied to the scored training data. This plot can be empty if no variables are determined to be important. Beginning in SAS Visual Data Mining and Machine Learning 8.3, a new relative importance calculation is used so that all values are between 0 and 1. Due to this change, relative importance might differ from older versions.

Assessment Plot

For a factorization machine, the Assessment plot plots the predicted average and observed average response values against the binned data. Use this plot to reveal any strong biases in your model. Large differences in the predicted average and observed average values can indicate a bias.

Recommendations

The Top Recommendations plot and the Bottom Recommendations plot display the top-ranked and bottom-ranked event levels for category variables.

Details Table

When you click ↙↗ in the object toolbar of the canvas, the details table is displayed at the bottom of the canvas. The details table contains the following information:

Model Information
Gives an overview of the model.

Iteration History
Provides the loss function iteration results. This tab shows the value of the loss function at each iteration.

Measures
Provides summary statistics for the input measure variables.

Categories
Provides information about the input category variables.

Final Exact Loss
Provides error statistics for the model.

Factors
Provides the bias and factor weights for each observation in the input data.

Assessment
Lists the binned assessment results that are used to generate the Assessment plot.

Assessment Statistics
Provides the value of any assessment statistic computed for the model.

Working with Forests

Overview of Forests

A forest is an ensemble model that contains a specific number of decision trees. To ensure that a forest does not overfit the data, two key steps are taken. First, each tree in the forest is built on a different sample of the training data. Second, when splitting each node, a set of candidate inputs for the split are selected at random, and the best split is selected from those. Other than these two steps, the trees in a forest are trained like standard trees.

The training data for each tree in the forest excludes some of the available data. This data is called the out-of-bag sample. For each leaf in a tree, the prediction for an interval target is the average of the target values. The posterior probability for a target category of a nominal target is the proportion of the target category among observations in the leaf. These values can be based on training some of

the out-of-bag sample. To predict the target value of a new observation, that observation is fed through each tree, and the predicted value is determined by the **Vote** option.

By default, in the SAS procedures that implement forests, the training sample is used.

Create a Forest

1. In the left pane, click to select an object. Drag the icon onto the canvas to create a forest.
2. Click in the right pane. Specify a single variable as the **Response** variable.
3. Specify at least one measure variable or category variable for **Predictors**.
4. (Optional) Specify **Partition ID**.

Forest Options

The following options are available for the forest:

General

Event level
enables you to choose the event level of interest. When your category response variable contains more than two levels, SAS Visual Data Mining and Machine Learning treats all observations in the level of interest as an event and all other observations as nonevents when calculating assessment statistics.

Autotune
enables you to specify the constraints that control the autotuning algorithm. The constraints determine how long the algorithm can run, how many times the algorithm can run, and how many model evaluations are allowed.

The autotuning algorithm selects the values for the following parameters that produce the best model:

- **Number of trees**
- **Bootstrap**
- **Number of predictors to split nodes**
- **Maximum levels**
- **Leaf size**
- **Predictor bins**

Number of trees
specifies the number of trees in the forest. You must specify an integer value in the range 1–10,000.

Bootstrap
specifies the bootstrap value. This is the percentage of data used to grow each tree in the forest.

Vote
specifies the method used to determine the predicted value for each observation when you specify a classification response. For **Majority** voting, the event level that is predicted most often is selected. For **Probability** voting, the average of the probabilities predicted by each tree is computed and compared against the **Prediction cutoff** value.

Measure responses always use the average across all trees as the predicted value.

Splitting criterion
specifies the splitting criterion that is used to create branches. For a category response, the available criteria are **Entropy**, **Information gain ratio**, **Chi-square**, **Gini index**, and **CHAID**. For a measure response, the available criteria are **CHAID**, **FTest**, and **RSS/Variance**.

Set fixed number of predictors to split nodes
specifies whether the default number of predictors are used to split nodes in each decision tree.

Number of predictors to split nodes
specifies the number of predictors used in each decision tree.

Decision Tree

Missing assignment
specifies how observations with missing values are included in the model:

- **None**: Observations with missing values are excluded from the model.
- **Use in search**: If the number of observations with missing values is greater than or equal to **Minimum value**, then missing values are considered a unique measurement level and are included in the model.
- **As machine smallest**: Missing interval values are set to the smallest possible machine value such that the observations are always in the split with the lower variable values. Missing category values are treated as a unique measurement level.

Minimum value
specifies the minimum number of observations allowed to have missing values before missing values are treated as a distinct category level. This option is used only when **Missing assignment** is set to **Use in search**.

Maximum branches
specifies the maximum number of branches allowed when splitting a node.

Maximum levels
specifies the maximum depth of the decision tree.

Leaf size
specifies the minimum number of observations allowed in a leaf node.

Predictor bins
specifies the number of bins used to categorize a predictor that is a measure variable.

Bin method
specifies the method that is used to bin the measure predictors. Select **Bucket** to divide the measure predictors into evenly spaced intervals based on the difference between maximum and minimum values. Select **Quantile** to divide the measure predictors into approximately equally sized groups. The default value is **Quantile**.

Partial Dependence

Number of observations
specifies the maximum number of observations to sample from the input data. These observations are used to generate the model predictions that are displayed in the Partial Dependence plot.

Number of bins
specifies the number of bins to use to categorize measure variables in the Partial Dependence plot.

Assessment

Number of bins
specifies the number of bins to use in the assessment. You must specify an integer value in the range 5–100.

Prediction cutoff
specifies the value at which a computed probability is considered an event.

Statistic percentile
specifies the depth for the percentile bins that are used to calculate the observed average, lift, cumulative lift, cumulative percentage captured, cumulative percentage events, and gain.

Tolerance
specifies the tolerance value that is used to determine the convergence of the iterative algorithm that estimates the percentiles. Specify a smaller value to increase the algorithmic precision.

Forest Model Display Options

The following display options are available for forests:

General

Plot layout
specifies how the subplots within objects are displayed on the canvas. **Fit** aligns all of the subplots on the canvas automatically. **Stack** displays the subplots as if they are in a slide deck where only one plot is displayed at a time. When **Stack** is specified, a control bar instead of a scroll bar lets you move between the subplots.

Statistic to show
specifies which assessment statistic to display in the model. If you are using a partition variable, the following fit statistics (with the exception of SSE) are available for each partition. The object toolbar contains submenus for each partition type that is available (training, validation, and test).

- **ASE**
- **Observed Average**
- **SSE**
- **C Statistic**
- **Cumulative % Captured**
- **Cumulative % Events**
- **Cumulative Lift**
- **F1 Score**
- **FDR**
- **FPR**
- **Gain**
- **Gamma**
- **Gini**
- **KS (Youden)**
- **Lift**
- **Misclassification Rate**
- **Misclassification Rate (Event)**
- **Tau**

See Fit Statistics for more information about the fit statistics that are available.

Note: **ASE**, **Observed Average**, and **SSE** are available only for measure responses. The remaining fit statistics are available only for category responses.

Error Plot / Partial Dependence Plot

Plot to show
specifies whether the Error plot or the Partial Dependence plot is displayed.

X axis
specifies which variable is displayed in the Partial Dependence plot.

Legend visibility
specifies whether the legend is displayed in the Error plot.

Assessment Plots

Display test partition
specifies whether to display the assessment plot for the test partition. This option is available only when you specify a partition ID with a test partition.

Plot to show
specifies which assessment plot is displayed. Select **Confusion matrix**, **Lift**, **ROC**, or **Misclassification**.

Y axis
specifies whether a standard Lift plot or a Cumulative Lift plot is displayed.

Legend visibility
specifies whether the legend is displayed in the confusion matrix, Lift plot, ROC plot, or Misclassification plot.

Forest Results

Variable Importance

The Variable Importance plot displays the importance of each variable as measured by its loss reduction variable importance value.

Error Plot

The Error plot compares the model and out-of-bag misclassification rates for a category response. You can use the out-of-bag misclassification rates as a validation measurement to ensure that the model is not overfitting the input data. The Error Plot displays the misclassification rate across the number of trees for the entire model. The Assessment plot displays the assessment results for just a single target level. For a measure response, the Error Plot compares the average squared error.

Partial Dependence Plot

The Partial Dependence (PD) plot helps you understand how the model's predictions depend on the value of a given predictor. This plot helps you determine how the value of the response changes as the value of a given predictor changes. On the Y axis, categorical response models display the predicted probability of the event, and measure response models display the predicted response value. In both cases, the values are the average of the observations in that bin. For multinomial models, you can modify the event level to examine other relationships. The X axis displays the values of the specified predictor. The predictor that is shown by default is the predictor with the largest importance value. The plot displays a line that shows the point estimate and a 95% confidence band. The heat map at the bottom of the plot indicates the density of the observations that were sampled.

Some bins might have zero observations. The Partial Dependence plots are generated by using a sample of all observations, even if you are using a partition variable.

Assessment Plot

The confusion matrix displays the classification results for categorical response models. After a model is created, each observation has an observed value and a predicted value. The total number of each observed-predicted pair is calculated. The confusion matrix displays how many observations fall into each pair. A perfect model always predicts the observed value, and all values lie on the diagonal of the matrix. Any off-diagonal values represent a misclassification. Cells are shaded based on the proportion of the value in each cell to the number of observed values for that level. Darker shaded cells show the concentration of the predictions for the observed level. For a binary response, the information in the confusion matrix is identical to the Misclassification plot.

Lift is the ratio of the percent of captured responses within each percentile bin to the average percent of responses for the model. Similarly, *cumulative lift* is calculated by using all of the data up to and including the current percentile bin.

A receiver operating characteristic (ROC) chart displays the ability of a model to avoid false positive and false negative classifications. A false positive classification means that an observation has been identified as an event when it is actually a nonevent (also referred to as a Type I error). A false negative classification means that an observation has been identified as a nonevent when it is actually an event (also referred to as a Type II error).

The *specificity* of a model is the true negative rate. To derive the false positive rate, subtract the specificity from 1. The false positive rate, labeled **1 – Specificity**, is the X axis of the ROC chart. The *sensitivity* of a model is the true positive rate. This is the Y axis of the ROC chart. Therefore, the ROC chart plots how the true positive rate changes as the false positive rate changes.

A good ROC chart has a very steep initial slope and levels off quickly. That is, for each misclassification of an observation, significantly more observations are correctly classified. For a perfect model, one with no false positives and no false negatives, the ROC chart would start at (0,0), continue vertically to (0,1), and then horizontally to (1,1). In this instance, the model would correctly classify every observation before a single misclassification could occur.

The ROC chart includes two lines to help you interpret it. The first line is a baseline model that has a slope of 1. This line mimics a model that correctly classifies observations at the same rate it incorrectly classifies them. An ideal ROC chart maximizes the distance between the baseline model and the ROC chart. A model that classifies more observations incorrectly than correctly would fall below the baseline model. The second line is a vertical line at the false positive rate where the difference between the Kolmogorov-Smirnov values for the ROC chart and baseline models is maximized.

The Misclassification plot displays how many observations were correctly and incorrectly classified for each value of the response variable. When the response variable is not binary, the forest considers all levels that are not events as equal. A significant number of misclassifications could indicate that your model does not fit the data.

For a measure response, the Assessment plot plots the average predicted and average observed response values against the binned data. Use this plot to reveal any strong biases in your model. Large differences in the average predicted and average observed values can indicate a bias.

Details Table

When you click ⤢ in the object toolbar of the canvas, the details table is displayed at the bottom of the canvas. The details table contains the following information:

Variable Importance
Provides the importance information and standard deviation for each variable in the model.

Error Metric
Provides the calculated error rate for every forest model sorted by the number of trees in the forest.

Confusion Matrix
Provides a summary of the correct and incorrect classifications for the model that is used to generate the confusion matrix.

Lift
Lists the binned assessment results that are used to generate the Lift plot.

ROC
Lists the results that are used to generate the ROC plot.

Misclassification
Provides a summary of the correct and incorrect classifications for the model.

Assessment
Provides the binned assessment results that are used to generate the Assessment plot.

Assessment Statistics
Provides the value of any assessment statistic computed for the model.

Note: The **Assessment** detail table is available only for measure responses. The **Lift**, **ROC**, and **Misclassification** detail tables are available only for category responses.

Working with Gradient Boosting Models

Overview of Gradient Boosting Models

Gradient boosting is an iterative approach that creates multiple trees where each tree is typically based on an independent sample without replacement of the data. A gradient boosting model hones its predictions by minimizing a specified loss function, such as average square error. The first step creates a baseline tree. Each subsequent tree is fit to the residuals of the previous tree, and the loss function is minimized. This process is repeated a specific number of times. The final model is a single function, which is an aggregation of the series of trees that can be used to predict the target value of a new observation.

The phrase “stochastic gradient boosting” refers to training each new tree based on a subsample of the data. This typically results in a better model. For gradient boosting models, each new observation is fed through a sequence of trees that are created to predict the target value of each new observation.

Create a Gradient Boosting Model

1 In the left pane, click to select an object. Drag the icon onto the canvas to create a gradient boosting model.

2 Click in the right pane. Specify either a single measure variable or a single category variable as the **Response** variable.

3 Specify at least one or more measure or category variables for **Predictors**.

4 (Optional) Specify **Partition ID**.

Gradient Boosting Options

The following options are available for the gradient boosting model:

General

Event level
enables you to choose the event level of interest. When your category response variable contains more than two levels, SAS Visual Data Mining and Machine Learning treats all observations in the level of interest as an event and all other observations as nonevents when calculating assessment statistics.

Autotune
enables you to specify the constraints that control the autotuning algorithm. The constraints determine how long the algorithm can run, how many times the algorithm can run, and how many model evaluations are allowed.

The autotuning algorithm selects the values for the following parameters that produce the best model:

- **Number of trees**
- **Learning rate**
- **Subsample rate**
- **Lasso**
- **Ridge**
- **Number of predictors to split nodes**
- **Maximum levels**
- **Leaf size**
- **Predictor bins**

If data partitioning is applied, then the following parameters are also autotuned:

- **Auto-stop method**
- **Auto-stop iterations**
- **Tolerance value**

Auto-stop method
specifies the method that controls early termination of the model-building algorithm. For **Stagnation**, the algorithm terminates when there is no improvement in a specified number of successive validation error calculations. For **Tolerance**, the algorithm terminates when the magnitude of the relative change in validation error is less than a specified value for a specified number of calculations. This option is available only when you specify a partition ID.

Auto-stop iterations
specifies the number of successive validation error calculations required to trigger the **Stagnation** or **Tolerance** auto-stop method. You must specify an integer value in the range 1–10,000. This option is available only when you specify a partition ID.

Tolerance value
specifies the minimum change required in the validation error required to trigger the **Tolerance** auto-stop method. You must specify a value in the range 0–1. This option is available only when you specify a partition ID.

Number of trees
specifies the number of trees in the gradient boosting model.

Learning rate
specifies the learning rate used to update the gradient boosting model.

Subsample rate
specifies the subsample rate to create each tree in the gradient boosting model.

Lasso
specifies the LASSO (Least Absolute Shrinkage and Selection Operator) parameter that is used to select the regression coefficients.

Ridge
specifies the ridge value that is used to update the gradient boosting model.

Set fixed number of predictors to split nodes
specifies whether the default number of predictors are used to split nodes in each decision tree.

Number of predictors to split nodes
specifies the number of predictors used in each decision tree.

Decision Tree

Missing assignment
specifies how observations with missing values are included in the model:

- **None**: Observations with missing values are excluded from the model.
- **Use in search**: If the number of observations with missing values is greater than or equal to **Minimum value**, then missing values are considered a unique measurement level and are included in the model.
- **As machine smallest**: Missing interval values are set to the smallest possible machine value such that the observations will always be in the split with the lower variable values. Missing category values are treated as a unique measurement level.

Minimum value
specifies the minimum number of observations allowed to have missing values before missing values are treated as a distinct category level. This option is used only when **Missing assignment** is set to **Use in search**.

Maximum branches
specifies the maximum number of branches allowed when splitting a node.

Maximum levels
specifies the maximum depth of the decision tree.

Leaf size
specifies the minimum number of observations allowed in a leaf node.

Predictor bins
specifies the number of bins used to categorize a predictor that is a measure variable.

Bin method
Specifies the method that is used to bin the measure predictors. Select **Bucket** to divide the measure predictors into evenly spaced intervals based on the difference between maximum and minimum values. Select **Quantile** to divide the measure predictors into approximately equally sized groups. The default value is **Quantile**.

Partial Dependence

Number of observations
specifies the maximum number of observations to sample from the input data. These observations are used to generate the model predictions that are displayed in the Partial Dependence plot.

Number of bins
specifies the number of bins to use to categorize measure variables in the Partial Dependence plot.

Assessment

Number of bins
specifies the number of bins to use in the assessment. You must specify an integer value in the range 5–100.

Prediction cutoff
specifies the value at which a computed probability is considered an event.

Statistic percentile
specifies the depth for the percentile bins that are used to calculate the observed average, lift, cumulative lift, cumulative percentage captured, cumulative percentage events, and gain.

Tolerance
specifies the tolerance value that is used to determine the convergence of the iterative algorithm that estimates the percentiles. Specify a smaller value to increase the algorithmic precision.

Gradient Boosting Model Display Options

The following display options are available for the gradient boosting model:

General

Plot layout
specifies how the subplots within objects are displayed on the canvas. **Fit** aligns all of the subplots on the canvas automatically. **Stack** displays the subplots as if they are in a slide deck where only one plot is displayed at a time. When **Stack** is specified, a control bar instead of a scroll bar lets you move between the subplots.

Statistics to show
specifies which assessment statistic to display in the model. If you are using a partition variable, the following fit statistics (with the exception of SSE) are available for each partition. The object toolbar contains submenus for each partition type that is available (training, validation, and test).

- **ASE**
- **Observed Average**
- **SSE**
- **C Statistic**
- **Cumulative % Captured**
- **Cumulative % Events**
- **Cumulative Lift**
- **F1 Score**
- **FDR**
- **FPR**
- **Gain**

- **Gamma**
- **Gini**
- **KS (Youden)**
- **Lift**
- **Misclassification Rate**
- **Misclassification Rate (Event)**
- **Tau**

See Fit Statistics for more information about the fit statistics that are available.

> **Note:** **ASE**, **Observed Average**, and **SSE** are available only for measure responses. The remaining fit statistics are available only for category responses.

Iteration Plot / Partial Dependence Plot

Plot to show
specifies whether the Error plot or the Partial Dependence plot is displayed.

X axis
specifies which variable is displayed in the Partial Dependence plot.

Legend visibility
specifies whether the legend is displayed in the Iteration plot.

Assessment Plots

Display test partition
specifies whether to display the assessment plot for the test partition. This option is available only when you specify a partition ID with a test partition.

Plot to show
specifies which assessment plot is displayed. Select **Confusion matrix**, **Lift**, **ROC**, or **Misclassification**.

Y axis
specifies whether a standard Lift plot or a Cumulative Lift plot is displayed.

Legend visibility
specifies whether the legend is displayed in the confusion matrix, Lift plot, ROC plot, or Misclassification plot.

Gradient Boosting Results

Variable Importance

The Variable Importance plot displays the importance of each variable as measured by its contribution to the change in the residual sum of squared errors value.

Iteration Plot

The Iteration plot displays either the misclassification rate for a category response or the average squared error for a measure response across the number of trees for the entire model.

Partial Dependence Plot

The Partial Dependence (PD) plot helps you understand how the model's predictions depend on the value of a given predictor. This plot helps you determine how the value of the response changes as the value of a given predictor changes. On the Y axis, categorical response models display the predicted probability of the event, and measure response models display the predicted response value. In both cases, the values are the average of the observations in that bin. For multinomial models, you can modify the event level to examine other relationships. The X axis displays the values of the specified predictor. The predictor that is shown by default is the predictor with the largest importance value. The plot displays a line that shows the point estimate and a 95% confidence band. The heat map at the bottom of the plot indicates the density of the observations that were sampled. Some bins might have zero observations. The Partial Dependence plots are generated by using a sample of all observations, even if you are using a partition variable.

Assessment Plot

The confusion matrix displays the classification results for categorical response models. After a model is created, each observation has an observed value and a predicted value. The total number of each observed-predicted pair is calculated. The confusion matrix displays how many observations fall into each pair. A perfect model always predicts the observed value, and all values lie on the diagonal of the matrix. Any off-diagonal values represent a misclassification. Cells are shaded based on the proportion of the value in each cell to the number of observed values for that level. Darker shaded cells show the concentration of the predictions for the observed level. For a binary response, the information in the confusion matrix is identical to the Misclassification plot.

Lift is the ratio of the percent of captured responses within each percentile bin to the average percent of responses for the model. Similarly, *cumulative lift* is calculated by using all of the data up to and including the current percentile bin.

A receiver operating characteristic (ROC) chart displays the ability of a model to avoid false positive and false negative classifications. A false positive classification means that an observation has been identified as an event when it is actually a nonevent (also referred to as a Type I error). A false negative classification means that an observation has been identified as a nonevent when it is actually an event (also referred to as a Type II error).

The *specificity* of a model is the true negative rate. To derive the false positive rate, subtract the specificity from 1. The false positive rate, labeled **1 – Specificity**, is the X axis of the ROC chart. The *sensitivity* of a model is the true positive rate. This is the Y axis of the ROC chart. Therefore, the ROC chart plots how the true positive rate changes as the false positive rate changes.

A good ROC chart has a very steep initial slope and levels off quickly. That is, for each misclassification of an observation, significantly more observations are correctly classified. For a perfect model, one with no false positives and no false negatives, the ROC chart would start at (0,0), continue vertically to (0,1), and then horizontally to (1,1). In this instance, the model would correctly classify every observation before a single misclassification could occur.

The ROC chart includes two lines to help you interpret it. The first line is a baseline model that has a slope of 1. This line mimics a model that correctly classifies observations at the same rate it incorrectly classifies them. An ideal ROC chart maximizes the distance between the baseline model and the ROC chart. A model that classifies more observations incorrectly than correctly would fall below the baseline model. The second line is a vertical line at the false positive rate where the difference between the Kolmogorov-Smirnov values for the ROC chart and baseline models is maximized.

The Misclassification plot displays how many observations were correctly and incorrectly classified for each value of the response variable. When the response variable is not binary, the gradient boosting

model considers all levels that are not events as equal. A significant number of misclassifications could indicate that your model does not fit the data.

For a measure response, the Assessment plot plots the average predicted and average observed response values against the binned data. Use this plot to reveal any strong biases in your model. Large differences in the average predicted and average observed values can indicate a bias.

Details Table

When you click ↙↗ in the object toolbar of the canvas, the details table is displayed at the bottom of the canvas. The details table contains the following information:

Variable Importance
Provides the importance information and standard deviation for each variable in the model.

Iteration History
Provides the error function iteration results. This tab shows the value of the error function at each iteration.

Confusion Matrix
Provides a summary of the correct and incorrect classifications for the model that is used to generate the confusion matrix.

Lift
Lists the binned assessment results that are used to generate the Lift plot.

ROC
Lists the results that are used to generate the ROC plot.

Misclassification
Provides a summary of the correct and incorrect classifications for the model.

Assessment
Provides the binned assessment results that are used to generate the Assessment plot.

Assessment Statistics
Provides the value of any assessment statistic computed for the model.

Note: The **Assessment** detail table is available only for measure responses. The **Lift**, **ROC**, and **Misclassification** detail tables are available only for category responses.

Working with Neural Networks

Overview of Neural Networks

A neural network is a statistical model that is designed to mimic the biological structures of the human brain. Neural networks consist of predictors (input variables), hidden layers, an output layer, and the connections between each of those. Predictors can be connected directly to the output layer. The creation of those connections is determined by the default activation function. You can specify an activation function for each hidden layer.

Create a Neural Network

To create a neural network, complete the following steps:

1. In the left pane, click to select an object. Drag the icon onto the canvas to create a neural network.
2. Click in the right pane. Specify a single variable as the **Response** variable.
3. Specify at least one measure variable or category variable for **Predictors**.
4. (Optional) Specify **Partition ID** or a **Weight** variable.

Neural Network Options

The following options are available for the neural network:

General

Event level
enables you to choose the event level of interest. When your category response variable contains more than two levels, SAS Visual Data Mining and Machine Learning treats all observations in the level of interest as an event and all other observations as nonevents when calculating assessment statistics.

Autotune
enables you to specify the constraints that control the autotuning algorithm. The constraints determine how long the algorithm can run, how many times the algorithm can run, and how many model evaluations are allowed.

The autotuning algorithm selects the values for the following parameters that produce the best model:

- **Number of hidden layers**
- **Number of neurons**
- **L1**
- **L2**
- **Learning rate**
- **Annealing rate**
- **Maximum iterations**

If data partitioning is applied, then the following parameters are also autotuned:

- **Auto-stop method**
- **Auto-stop iterations**
- **Goal value**

Auto-stop method
specifies the method that controls early termination of the model-building algorithm. For **Stagnation**, the algorithm terminates when there is no improvement in a specified number of successive validation error calculations. For **Goal**, the algorithm terminates when the validation

error is less than the specified **Goal value**. This option is available only when you specify a partition ID.

Auto-stop iterations
specifies the number of successive validation error calculations required to trigger the **Stagnation** auto-stop method. You must specify an integer value in the range 1–10,000. This option is available only when you specify a partition ID.

Goal value
specifies the maximum validation error acceptable to trigger the **Goal** auto-stop method. When the validation error is less than this value, the model-building algorithm terminates. This option is available only when you specify a partition ID.

Include missing
specifies whether missing values are included in the model. Missing category values are treated as their own measurement level. Missing measure variables are imputed to the mean value for that variable.

Distribution
specifies the distribution used to model the response variable. This option is available only when the response is a measure variable. Possible values are **Normal**, **Gamma**, and **Poisson**.

Output activation function
specifies the activation function that is used to create the output layer when using a measure response. The exponential activation function is used with a Gamma or Poisson distribution. This option is available only when you specify a measure response and the **Normal** distribution. Possible values are **Hyperbolic tangent**, **Identity**, and **Sine**.

Standardization
specifies the method that is used to standardize the measure predictors. Here are the possible values:

- **None**: No standardization is applied.
- **Standard deviation**: Transforms the effect variables so that they have a mean of zero and a standard deviation of 1.
- **Midrange**: Linearly transforms the effect values to the range [0, 1].

Maximum iterations
specifies the maximum number of optimization iterations that are used before model training stops.

Maximum time (sec)
specifies the maximum time-out value.

Optimization method
specifies the optimization method used to train the neural network. You can specify either the **SGD** or **LBFGS** method.

Learning rate
specifies the learning rate used in stochastic gradient descent optimization. You must specify a value that is greater than 0, up to and including 100. This option is available only when **Optimization method** is set to **SGD**.

Annealing rate
specifies the annealing rate parameter used in stochastic gradient descent optimization. You must specify a value in the range 0–100. This option is available only when **Optimization method** is set to **SGD**.

L1
specifies the L1 regularization parameter.

L2
specifies the L2 regularization parameter.

Hidden Layers

Number of hidden layers
specifies the number of hidden layers in the model. The maximum value allowed is 2 if the **Optimization method** is LBFGS. The maximum value is 5 if the **Optimization method** is SGD.

Allow direct connections between input and target neurons
specifies that each input neuron is connected to each output neuron in the network.

Neurons
specifies the number of neurons in the hidden layer.

Activation function
specifies the activation function used for each neuron's output based on the weighted sum of its inputs. This can be changed for each hidden layer. You can specify either a **Hyperbolic Tangent**, **Identity**, **Sine**, **Exponential**, **Logistic**, **Rectifier**, or **Softplus** function as the activation function.

Partial Dependence

Number of observations
specifies the maximum number of observations to sample from the input data. These observations are used to generate the model predictions that are displayed in the Partial Dependence plot.

Number of bins
specifies the number of bins to use to categorize measure variables in the Partial Dependence plot.

Assessment

Number of bins
specifies the number of bins to use in the assessment. You must specify an integer value in the range 5–100.

Prediction cutoff
specifies the value at which a computed probability is considered an event.

Statistic percentile
specifies the depth for the percentile bins that are used to calculate the observed average, lift, cumulative lift, cumulative percentage captured, cumulative percentage events, and gain.

Tolerance
specifies the tolerance value that is used to determine the convergence of the iterative algorithm that estimates the percentiles. Specify a smaller value to increase the algorithmic precision.

Neural Network Model Display Properties

The following display properties are available for neural networks:

General

Plot layout
specifies how the subplots within objects are displayed on the canvas. **Fit** aligns all of the subplots on the canvas automatically. **Stack** displays the subplots as if they are in a slide deck where only one plot is displayed at a time. When **Stack** is specified, a control bar instead of a scroll bar lets you move between the subplots.

Statistic to show
specifies which assessment statistic to display in the model. If you are using a partition variable, the following fit statistics (with the exception of SSE) are available for each partition. The object toolbar contains submenus for each partition type that is available (training, validation, and test).

- **ASE**
- **Observed Average**
- **SSE**
- **C Statistic**
- **Cumulative % Captured**
- **Cumulative % Events**
- **Cumulative Lift**
- **F1 Score**
- **FDR**
- **FPR**
- **Gain**
- **Gamma**
- **Gini**
- **KS (Youden)**
- **Lift**
- **Misclassification Rate**
- **Misclassification Rate (Event)**
- **Tau**

See Fit Statistics for more information about the fit statistics that are available.

> **Note:** **ASE**, **Observed Average**, and **SSE** are available only for measure responses. The remaining fit statistics are available only for category responses.

Network Diagram

Neuron labels
specifies whether the neurons in the Network diagram are labeled.

Neuron layout
specifies how the neurons are plotted within layers in the Network diagram.

Horizontal spacing
specifies the amount of horizontal space that exists between nodes when **Horizontal layout** is set to **Fixed**. When **Horizontal layout** is set to **Proportional**, this option specifies the ratio of horizontal spacing to vertical spacing.

Vertical spacing
specifies the amount of vertical space that exists between nodes.

Percentage of links to display
specifies how many connecting links are displayed in the Network diagram.

Horizontal layout
specifies whether the width between neuron layers is fixed or is proportional to the height of the Network diagram.

Number of neurons to display
specifies the maximum number of neurons that are displayed within each layer in the Network diagram.

Legend visibility
specifies whether the legend is displayed in the Network diagram.

Iteration Plot / Relative Importance Plot

Plot to show
specifies whether the Iteration plot, the Relative Importance plot, the Validation Error plot, or the Partial Dependence plot is displayed.

X axis
specifies which variable are displayed in the Partial Dependence plot.

Legend visibility
specifies whether the legend is displayed in the Iteration plot, the Relative Importance plot, or the Validation Error plot.

Assessment Plots

Display test partition
specifies whether to display the assessment plot for the test partition. This option is available only when you specify a **Partition ID** with a test partition.

Plot to show
specifies which assessment plot is displayed. Select **Confusion matrix**, **Lift**, **ROC**, or **Misclassification**.

Y axis
specifies whether a standard Lift plot or a Cumulative Lift plot is displayed.

Legend visibility
specifies whether the legend is displayed in the confusion matrix, Lift plot, ROC plot, or Misclassification plot.

Neural Network Results

Network Plot

The Network plot displays the input nodes, hidden nodes, connections, and output nodes of a neural network. Nodes are represented as circles and links between the nodes are lines connecting two circles. The size of the circle represents the absolute value at that node, relative to the model. The color indicates whether that absolute value is positive or negative. Similarly, the size of the line between two nodes indicates the strength of the link and the color indicates whether that value is positive or negative.

To modify the neural network, right-click in the Network plot, and select one of the following options:

- **Add a hidden layer** inserts a new hidden layer into the neural network and rebuilds the model.
- **Edit a hidden layer** specifies the hidden layer that you want to modify. In the Edit Hidden Layer window, specify the number of **Neurons** and the **Activation function**.
- **Remove a hidden layer** removes a hidden layer from the neural network and rebuilds the model.

Iteration Plot

The Iteration plot displays the change in objective function value at each step of the model creation process. The vertical lines in the plot represent the first inner iteration of each performance iteration. The objective function value might increase at each vertical line, but it should always decrease within a performance iteration.

Relative Importance Plot

The Relative Importance plot plots the importance value of each input variable. The variables are ranked using their first-split log worth when applied to the scored training data. This plot can be empty if no variables are determined to be important. SAS Visual Data Mining and Machine Learning 8.3 included a new relative importance calculation so that all values are between 0 and 1. Due to this change, relative importance might differ from older versions.

Partial Dependence Plot

The Partial Dependence (PD) plot helps you understand how the model's predictions depend on the value of a given predictor. This plot helps you determine how the value of the response changes as the value of a given predictor changes. On the Y axis, categorical response models display the predicted probability of the event, and measure response models display the predicted response value. In both cases, the values are the average of the observations in that bin. For multinomial models, you can modify the event level to examine other relationships. The X axis displays the values of the specified predictor. The predictor that is shown by default is the predictor with the largest importance value. The plot displays a line that shows the point estimate and a 95% confidence band. The heat map at the bottom of the plot indicates the density of the observations that were sampled. Some bins might have zero observations. The Partial Dependence plots are generated by using a sample of all observations, even if you are using a partition variable.

Assessment Plot

The confusion matrix displays the classification results for categorical response models. After a model is created, each observation has an observed value and a predicted value. The total number of each observed-predicted pair is calculated. The confusion matrix displays how many observations fall into each pair. A perfect model always predicts the observed value, and all values lie on the diagonal of the matrix. Any off-diagonal values represent a misclassification. Cells are shaded based on the proportion of the value in each cell to the number of observed values for that level. Darker shaded cells show the concentration of the predictions for the observed level. For a binary response, the information in the confusion matrix is identical to the Misclassification plot.

Lift is the ratio of the percent of captured responses within each percentile bin to the average percent of responses for the model. Similarly, *cumulative lift* is calculated by using all of the data up to and including the current percentile bin.

A receiver operating characteristic (ROC) chart displays the ability of a model to avoid false positive and false negative classifications. A false positive classification means that an observation has been identified as an event when it is actually a nonevent (also referred to as a Type I error). A false negative classification means that an observation has been identified as a nonevent when it is actually an event (also referred to as a Type II error).

The *specificity* of a model is the true negative rate. To derive the false positive rate, subtract the specificity from 1. The false positive rate, labeled **1 – Specificity**, is the X axis of the ROC chart. The *sensitivity* of a model is the true positive rate. This is the Y axis of the ROC chart. Therefore, the ROC chart plots how the true positive rate changes as the false positive rate changes.

A good ROC chart has a very steep initial slope and levels off quickly. That is, for each misclassification of an observation, significantly more observations are correctly classified. For a perfect model, one with no false positives and no false negatives, the ROC chart would start at (0,0), continue vertically to (0,1), and then horizontally to (1,1). In this instance, the model would correctly classify every observation before a single misclassification could occur.

The ROC chart includes two lines to help you interpret it. The first line is a baseline model that has a slope of 1. This line mimics a model that correctly classifies observations at the same rate it

incorrectly classifies them. An ideal ROC chart maximizes the distance between the baseline model and the ROC chart. A model that classifies more observations incorrectly than correctly would fall below the baseline model. The second line is a vertical line at the false positive rate where the difference between the Kolmogorov-Smirnov values for the ROC chart and baseline models is maximized.

The Misclassification plot displays how many observations were correctly and incorrectly classified for each value of the response variable. When the response variable is not binary, the neural network model considers all levels that are not events as equal. A significant number of misclassifications could indicate that your model does not fit the data.

For a measure response, the Assessment plot plots the average predicted and average observed response values against the binned data. Use this plot to reveal any strong biases in your model. Large differences in the average predicted and average observed values can indicate a bias.

Validation Error

For a neural network, the Validation Error plot displays the change in validation error at each iteration of the training process.

Details Table

When you click ↙↗ in the object toolbar of the canvas, the details table is displayed at the bottom of the canvas. The details table contains the following information:

Model Information
Gives an overview of the model.

Iteration History
Provides the objective function and loss function iteration results. This tab shows the value of the objective function and loss function at each iteration.

Convergence
Provides the reason for convergence.

Confusion Matrix
Provides a summary of the correct and incorrect classifications for the model that is used to generate the confusion matrix.

Lift
Lists the binned assessment results that are used to generate the Lift plot.

ROC
Lists the results that are used to generate the ROC plot.

Misclassification
Provides a summary of the correct and incorrect classifications for the model.

Assessment
Provides the binned assessment results that are used to generate the Assessment plot.

Assessment Statistics
Provides the value of any assessment statistic computed for the model.

Note: The **Assessment** detail table is available only for measure responses. The **Lift**, **ROC**, and **Misclassification** detail tables are available only for category responses.

Working with Support Vector Machines

Overview of Support Vector Machines

A support vector machine is a machine learning model that is used to perform classification by constructing a set of hyperplanes that maximizes the margin between two classes. SAS Visual Data Mining and Machine Learning assigns a continuous probability output to each observation based on its distance to the boundary and which side of the boundary it is on.

Create a Support Vector Machine

To create a support vector machine, complete the following steps:

1. In the left pane, click to select an object. Drag the icon onto the canvas to create a support vector machine.
2. Click in the right pane. Specify a single category variable as the **Response** variable.
3. Specify at least one measure variable or category variable for **Predictors**.
4. (Optional) Specify **Partition ID**.

Support Vector Machine Options

The following options are available for the support vector machine:

General

Event level
enables you to choose the event level of interest. When your category response variable contains more than two levels, SAS Visual Data Mining and Machine Learning treats all observations in the level of interest as an event and all other observations as nonevents.

Autotune
enables you to specify the constraints that control the autotuning algorithm. The constraints determine how long the algorithm can run, how many times the algorithm can run, and how many model evaluations are allowed.

The autotuning algorithm selects the **Kernel function** and **Penalty value** values that produce the best model.

Kernel function
specifies the kernel function used for spatial classification.

- **Linear** $K(u, v) = u^T v$.
- **Quadratic** $K(u, v) = (u^T v + 1)^2$. The 1 is added in order to avoid zero-value entries in the Hessian matrix.

- **Cubic** $K(u, v) = (u^T v + 1)^3$. The 1 is added in order to avoid zero-value entries in the Hessian matrix.

Include missing
specifies whether missing values are included in the model. Missing category values are treated as their own measurement level. Missing measure variables are imputed to the mean value for that variable.

Penalty value
specifies the penalty value. The penalty value balances model complexity and training error. A larger penalty value creates a more robust model at the risk of overfitting the training data.

Tolerance value
specifies a custom tolerance value for model training. The tolerance value balances the number of support vectors and model accuracy. A tolerance value that is too large creates too few support vectors, and a value that is too small overfits the training data.

Maximum iterations
specifies the maximum number of optimization iterations that are used before model training stops.

Standardize measure predictors
specifies whether interval variables are scaled.

Partial Dependence

Number of observations
specifies the maximum number of observations to sample from the input data. These observations are used to generate the model predictions that are displayed in the Partial Dependence plot.

Number of bins
specifies the number of bins to use to categorize measure variables in the Partial Dependence plot.

Assessment

Number of bins
specifies the number of bins to use in the assessment. You must specify an integer value in the range 5–100.

Prediction cutoff
specifies the value at which a computed probability is considered an event.

Statistic percentile
specifies the depth for the percentile bins that are used to calculate the observed average, lift, cumulative lift, cumulative percentage captured, cumulative percentage events, and gain.

Tolerance
specifies the tolerance value that is used to determine the convergence of the iterative algorithm that estimates the percentiles. Specify a smaller value to increase the algorithmic precision.

Support Vector Machine Model Display Options

The following display options are available for the support vector machine:

General

Plot layout
specifies how the subplots within objects are displayed on the canvas. **Fit** aligns all of the subplots on the canvas automatically. **Stack** displays the subplots as if they are in a slide deck where only one plot is displayed at a time. When **Stack** is specified, a control bar instead of a scroll bar lets you move between the subplots.

Statistic to show
specifies which assessment statistic to display in the model. If you are using a partition variable, the following fit statistics are available for each partition. The object toolbar contains submenus for each partition type that is available (training, validation, and test).

- **C Statistic**
- **Cumulative % Captured**
- **Cumulative % Events**
- **Cumulative Lift**
- **F1 Score**
- **FDR**
- **FPR**
- **Gain**
- **Gamma**
- **Gini**
- **KS (Youden)**
- **Lift**
- **Misclassification Rate (Event)**
- **Tau**

See Fit Statistics for more information about the fit statistics that are available.

Partial Dependence Plot

Show partial dependence
specifies whether Partial Dependence plot is displayed.

X axis
specifies which variable is displayed in the Partial Dependence plot.

Assessment Plots

Display test partition
specifies whether to display the assessment plot for the test partition. This option is available only when you specify a partition ID with a test partition.

Plot to show
specifies which assessment plot is displayed. Select **Confusion matrix**, **Lift**, **ROC**, or **Misclassification**.

Y axis
specifies whether a standard Lift plot or a Cumulative Lift plot is displayed.

Legend visibility
specifies whether the legend is displayed in the confusion matrix, Lift plot, ROC plot, or Misclassification plot.

Support Vector Machine Results

Relative Importance Plot

The Relative Importance plot plots the importance value of each input variable. The variables are ranked using their first-split log worth when applied to the scored training data. This plot can be empty if no variables are determined to be important. Beginning in SAS Visual Data Mining and Machine Learning 8.3, a new relative importance calculation is used so that all values are between 0 and 1. Due to this change, relative importance might differ from older versions.

Partial Dependence Plot

The Partial Dependence (PD) plot helps you understand how the model's predictions depend on the value of a given predictor. This plot helps you determine how the value of the response changes as the value of a given predictor changes. On the Y axis, categorical response models display the predicted probability of the event, and measure response models display the predicted response value. In both cases, the values are the average of the observations in that bin. For multinomial models, you can modify the event level to examine other relationships. The X axis displays the values of the specified predictor. The predictor that is shown by default is the predictor with the largest importance value. The plot displays a line that shows the point estimate and a 95% confidence band. The heat map at the bottom of the plot indicates the density of the observations that were sampled. Some bins might have zero observations. The Partial Dependence plots are generated by using a sample of all observations, even if you are using a partition variable.

Assessment Plot

The confusion matrix displays the classification results for categorical response models. After a model is created, each observation has an observed value and a predicted value. The total number of each observed-predicted pair is calculated. The confusion matrix displays how many observations fall into each pair. A perfect model always predicts the observed value, and all values lie on the diagonal of the matrix. Any off-diagonal values represent a misclassification. Cells are shaded based on the proportion of the value in each cell to the number of observed values for that level. Darker shaded cells show the concentration of the predictions for the observed level. For a binary response, the information in the confusion matrix is identical to the Misclassification plot.

Lift is the ratio of the percent of captured responses within each percentile bin to the average percent of responses for the model. Similarly, *cumulative lift* is calculated by using all of the data up to and including the current percentile bin.

A receiver operating characteristic (ROC) chart displays the ability of a model to avoid false positive and false negative classifications. A false positive classification means that an observation has been identified as an event when it is actually a nonevent (also referred to as a Type I error). A false negative classification means that an observation has been identified as a nonevent when it is actually an event (also referred to as a Type II error).

The *specificity* of a model is the true negative rate. To derive the false positive rate, subtract the specificity from 1. The false positive rate, labeled **1 – Specificity**, is the X axis of the ROC chart. The *sensitivity* of a model is the true positive rate. This is the Y axis of the ROC chart. Therefore, the ROC chart plots how the true positive rate changes as the false positive rate changes.

A good ROC chart has a very steep initial slope and levels off quickly. That is, for each misclassification of an observation, significantly more observations are correctly classified. For a perfect model, one with no false positives and no false negatives, the ROC chart would start at (0,0),

continue vertically to (0,1), and then horizontally to (1,1). In this instance, the model would correctly classify every observation before a single misclassification could occur.

The ROC chart includes two lines to help you interpret it. The first line is a baseline model that has a slope of 1. This line mimics a model that correctly classifies observations at the same rate it incorrectly classifies them. An ideal ROC chart maximizes the distance between the baseline model and the ROC chart. A model that classifies more observations incorrectly than correctly would fall below the baseline model. The second line is a vertical line at the false positive rate where the difference between the Kolmogorov-Smirnov values for the ROC chart and baseline models is maximized.

The Misclassification plot displays how many observations were correctly and incorrectly classified for each value of the response variable. When the response variable is not binary, the support vector machine considers all levels that are not events as equal. A significant number of misclassifications could indicate that your model does not fit the data.

Details Table

When you click ⤢ in the object toolbar of the canvas, the details table is displayed at the bottom of the canvas. The details table contains the following information:

Model Information
Gives an overview of the model.

Iteration History
Provides the complementarity and feasibility statistics for each iteration.

Training
Provides a brief description of the components of the model.

Fit Statistics
Lists all of the fit statistics.

Confusion Matrix
Provides a summary of the correct and incorrect classifications for the model that is used to generate the confusion matrix.

Lift
Lists the binned assessment results that are used to generate the Lift plot.

ROC
Lists the results that are used to generate the ROC plot.

Misclassification
Provides a summary of the correct and incorrect classifications for the model.

Assessment Statistics
Provides the value of any assessment statistic computed for the model.

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. Copyright © 2019, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. November 2019 8.5-P1:vaobjdmml