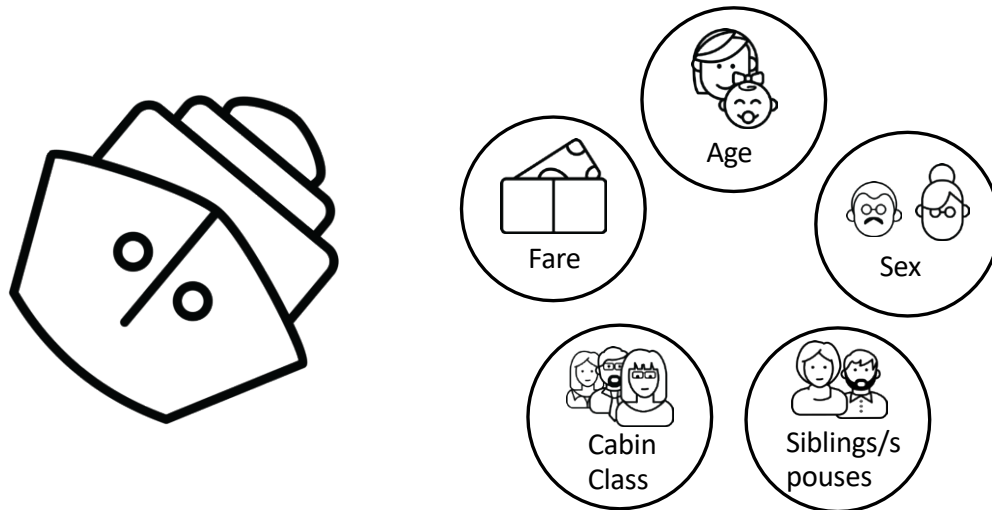# Interpreting Models in SAS®

Catherine Truxillo
Director, Analytical Education

Titanic Survival Data

The data to be used for analyses in the following parts of this course consist of survival status of individual passengers on the RMS Titanic maiden voyage. The data do not contain information for the crew, but they do contain names and ages for 80% of the passengers.

The data include the following:

A *target variable* indicates whether the passenger survived the sinking of the RMS Titanic.

Name	Description
survived	A binary target variable. Passengers coded with a 1 survived the sinking of the Titanic; passengers coded with a 0 perished in the Atlantic Ocean.

Interval valued inputs are numeric properties about passengers on the RMS Titanic.

Name	Description
age	Age of passenger in years
body	Body identification number
fare	Passenger fare in British pounds (pre-1970)

Categorical valued inputs summarize demographic and background information for passengers on the RMS Titanic.

Name	Description
boat	Lifeboat ID (if applicable)
cabin	Passenger cabin ID
embarked	Passenger Port of Embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)
home.dest	Passenger Home/Destination
name	Passenger name (unique ID)
parch	Passenger number of parents or children aboard
pclass	Passenger Class (1 = first, 2 = second, 3 = third)
sibsp	Passenger number of siblings or spouses aboard
ticket	Passenger ticket number
sex	Passenger sex

In this workshop, you fit predictive models to predict the survival status of passengers on the RMS Titanic. Although there are 13 possible inputs for our predictive model, the analysis will be restricted to include only 5 input variables.

You use **pclass**, **sibsp**, **sex**, **age**, and **fare** to predict the target, **survived**. The idea is that you are using demographic information to predict whether passengers of the RMS Titanic survived or perished at sea. The inputs not included either had too many missing values or were too specific to each passenger to be useful in a predictive model.

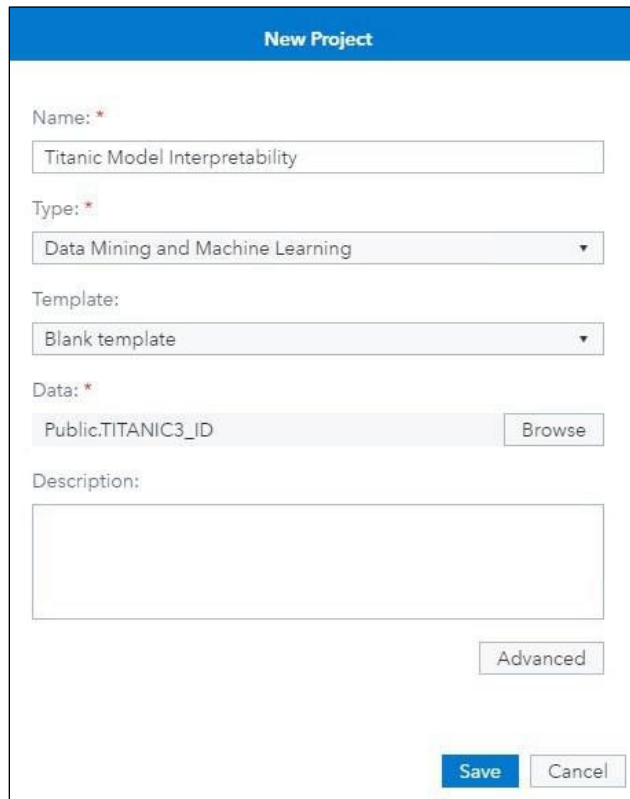## Building a Modeling Pipeline and Exploring Interpretable Models

1. Follow the instructions given by your instructor to access the virtual machine.

2. Select the **Google Chrome** shortcut on your virtual machine.

3. Select **SAS Drive** from the shortcuts bar.

4. Sign in to SAS using the user name **student** and the password **Metadata0**.

5. Click **No** when asked whether you want to opt in to your assumable groups.

Load the Modeling Data

1. Open the **Show Applications** menu ☰ and select **Build Models** to open the Model Studio visual interface.

2. In the Model Studio application, select **New Project**.

3. Name your project **Titanic Model Interpretability**. Leave **Type (Data Mining and Machine Learning)** and **Template (Blank)** at their default values.

4. Under **Data**, select **Browse** to access the Choose Data window.

5. From the Choose Data window, click the **Import** tab.

6. Select **Local files > Local file** to import a file from the local client machine.

7. Select **titanic3_ID.xlsx**.

8. Select **Replace file** under **If target table name exists**. Click **Import Item** to import the data to the SAS server. When the import is successful, select the **Titanic** table and click **OK** to use this data source for your project.

9. Click **Advanced** in the New Project window. Select **Partition Data**.

10. Change the partition distribution to **70%** Training, **30%** Validation, and **0%** Test. It should look like this:

11. Click **Save** in the New Project window to save the new project settings.

The New Project window should look like this:

12. Click **Save**.

Build a Modeling Pipeline

1. After the project is created, it opens and brings you to the Data tab. The software has already tried to make some decisions about how to treat the input variables, but some changes are needed.

2. Select **survived** and change the role from **Input** to **Target**. Deselect **survived**.

3. Select **key_ID** and change the role from **ID** to **Key**. This will be used at the end of the workshop so that you can reference individual observations by their key value. Although the name is a unique ID (and thus treated as an ID variable), it is useful to have a numeric key value for each passenger when you try to explain individual observations.

4. Deselect **key_ID**.

5. Reject some input variables. Some have already been rejected by the metadata advisor. Select **embarked** and **parch** and change their role from **Input** to **Rejected**.

6. Deselect **embarked** and **parch**.

At the end, your project metadata should look like this:

	Variable Name ↑	Label	Type	Role	Assess for Bias	Level
☐	age		Numeric	Input		Interval
☐	boat		Character	Rejected		Nominal
☐	body		Numeric	Rejected		Interval
☐	cabin		Character	Rejected		Nominal
☐	embarked		Character	Rejected		Nominal
☐	fare		Numeric	Input		Interval
☐	home.dest		Character	ID		Nominal
☐	key_ID		Numeric	Key		Interval
☐	name		Character	Text		Nominal
☐	parch		Numeric	Rejected		Nominal
☐	pclass		Numeric	Input		Nominal
☐	sex		Character	Input		Binary
☐	sibsp		Numeric	Input		Nominal
☐	survived		Numeric	Target		Binary
☐	ticket		Character	ID		Nominal

7. Select **Pipelines** next to **Data** to move to the Pipelines tab to create a predictive modeling pipeline.

8. Right-click the **Data** node and select **Add Child Node** > **Supervised Learning** > **Decision Tree** to add a decision tree to the pipeline.

9. Right-click the **Data** node again and select **Add Child Node** > **Supervised Learning** > **Gradient Boosting** to add a gradient boosting model to the pipeline.

    After you add the models, the pipeline should look like this:
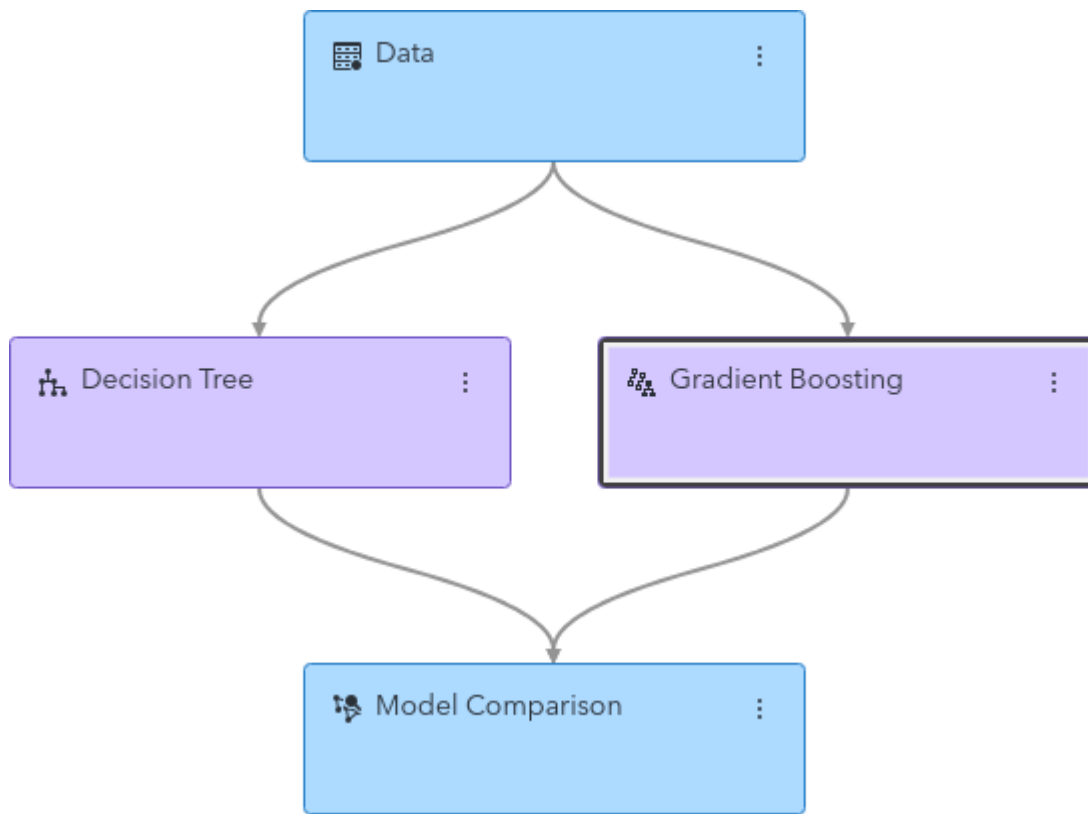
10. Click the **Decision Tree** node and scroll down to the bottom of the options on the right. Under **Post-training Properties**, select **Model Interpretability**. Below that, open **Global Interpretability** and **Local Interpretability**.

11. Select **Variable importance** and **PD plots** under **Global Interpretability**. This turns on a Variable Importance table as well as the partial dependence plots discussed in the next section.

12. Select **ICE plots** and turn on **LIME** and **HyperSHAP** under **Local Interpretability**. This turns on the individual conditional expectation plots and the local model-agnostic explanations discussed later in the workshop.

13. Repeat the same settings for the Gradient Boosting model, selecting all Global and Local interpretability plots.

14. In the top right of the pipeline, select **Run Pipeline**. This trains the models and creates the model interpretability plots specified.

15. When the pipeline has finished running, right-click the **Decision Tree** node and select **Results**.

    The pruning error plot reveals that the best decision tree on the validation data had 7 leaves and a validation misclassification rate of 16.79%.

    The misclassification rate does not tell the full story about how the decision tree classifies whether a passenger survives or dies. To better understand the model, you can interpret the decision tree itself.
    The Tree Diagram gives a graphical representation of the list of rules generated by the decision tree.
    This diagram is fundamentally interpretable because each leaf node can be reached by following a list of rules. For example, following the branches on the left, you see that for passengers who had a **sex** value of *male* and an **age** value greater than or equal to 2 or missing, 83.98% died. Thus, you would predict that any male passengers who are 2 or more years old will likely die on the Titanic.

16. Exit the Tree Diagram and close the Decision Tree node to return to the pipeline.

Although Decision Tree models with only a few leaves are generally easy to interpret, many other models are not so straightforward. In the sections that follow, you explore the model interpretability plots activated earlier in this demonstration for the Gradient Boosting model.

End of Demonstration

Partial Dependence (PD) Plots

Partial dependence plots reveal how the target prediction changes as the values of the inputs are changed. This technique uses the model to score a collection of modified data sets and plots the results to see how the average target prediction changes when the inputs are changed.

1. Choose an input to generate a partial dependence (PD) plot, and in the original data set, replace its values with X-axis values for the PD plot to generate a modified data set.

2. Use the model to make predictions on the modified data set, leaving all other inputs unchanged. This generates a target prediction for each observation. Average these target predictions to get the Y-axis value for the PD plot.

3. Iterate this process for each value on the X axis of the PD plot, replacing all values of the chosen input with the X-axis value at each iteration.

4. The end of this process is a table of X-axis values corresponding to different values of the selected input, and Y-axis values corresponding to the average target prediction for each of the X-axis values. These points are plotted to create the partial dependence plot.

Exploring Partial Dependence Plots

This demonstration proceeds from the end of the previous one. You are exploring the results of the Gradient Boosting node, noticing that this node does not provide any type of interpretable tree diagram. Instead, you will use the model interpretability tools.

1. Right-click the Gradient Boosting node and select **Results**.

2. From the Gradient Boosting results, click the **Model Interpretability** tab.

Node	Assessment	Model Interpretability	Fairness and Bias

The Model Variable Importance table tells you which variables were most useful in predicting the target. By default, the software creates partial dependence plots for only the top five most predictive variables. In this case, you have only five input variables, so you have PD plots for all five inputs.

3. Maximize the partial dependence plot by clicking the **Expand** button in the top right of the plot. The plot is on the left, and an automatically generated description of the contents of the plot is available on the right.

    This plot reveals that female passengers on the Titanic were much more likely to survive according to our model. The average probability of survival for female passengers was 0.654, whereas the average probability of survival for male passengers was 0.225.

4. From the menu next to **View chart**, change the input variable from **sex** to **age**. This shows another partial dependence plot, this time for the interval input **age**.
    The probability of survival drops precipitously after age 14, once again revealing the "women and children first" standard used for evacuating ships in the 19th and early 20th centuries. The probability of survival is relatively constant over age 15.

5. Explore the partial dependence plots for **fare**, **pclass**, and **sibsp** to see how these inputs relate to the target. The plots for **fare** and **pclass** reveal that socioeconomic class influenced the survival rate. (Wealthier passengers were more likely to survive the sinking of the Titanic.) The plot for **sibsp** show that having more family members on board reduced the probability of survival. However, the Model Variable Importance table shows that **sibsp** has the lowest relative importance in the model.

6. Close the partial dependence plot to return to the Model Interpretability tab in the decision tree results.

End of Demonstration

Individual Conditional Expectation (ICE) Overlay Plots

Individual conditional expectation (ICE) plots reveal similar information as partial dependence plots but can reveal input-target relationships that are averaged away by PD plots. These plots are created using the same procedure as PD plots.

ICE plots are constructed by performing the same procedure as described in the previous section on partial dependence plots.

Plotting the input-target relationship for a single observation, we generate a collection of fake observations based on the single observation of interest. These fake observations preserve the values of all variables other than the one we are interested in exploring. With all other variables held constant, we change the X-axis values of the variable of interest in the plot and score the observation of interest repeatedly using the model, generating target predictions for each value on the X axis. We plot these target predictions on the Y axis to see how this observation would be scored if it had different values of the variable of interest.

This is useful when trying to explain a specific observation but is less useful for exploring the overall input-target relationships because it is impossible to plot ICE curves for every observation on a single plot.

Five random observations are selected by default. However, you can specify up to five observations based on their key.

Exploring Individual Conditional Expectation Plots

This demonstration proceeds from the end of the previous one.

1. Close the results of the Gradient Boosting node to go back to the pipeline.

2. In the options for the Gradient Boosting node, scroll down to **Post-training properties**. Under **Local Interpretability**, find **Specify instances to explain**. Currently, **Random** is selected, which selects five random observations. In the drop-down list, change **Random** to **Specify up to 5.** This requires us to specify the "key" value for the observations of interest. In this case, the "key" variable is just a list from 1 to 1309 of passengers on the Titanic. Under **Individual observation 1**, enter **1**. This observation corresponds to the passenger Miss Elisabeth Walton Allen, a 29-year-old woman who survived. Under **Individual observation 2**, enter **1309**. This observation corresponds to the passenger Mr. Leo Zimmerman, a 29-year-old man who did not survive.

Post-training Properties

Changing these properties will not retrain the model.

∨ Model Interpretability

  ∨ Global Interpretability

    ☑ Variable importance

    ☑ PD plots

  ∨ Local Interpretability

    ☑ ICE plots

    ☑ LIME

    ☑ Kernel SHAP

Maximum number of Kernel SHAP variables:

    20

  1   34   67   100

Specify instances to explain:

  | Specify up to 5 ▾ |

Individual observation 1:

  | 1 |

Individual observation 2:

  | 1309 |

Individual observation 3:

  | Enter a string value |

3. Run the pipeline.

4. When the pipeline has run successfully, open the **Gradient Boosting** node's results and click the **Model Interpretability** tab.

5. Scroll to the **PD and ICE Overlay Plot**. Click the **Expand** button to make the plot full screen.

6. Hover on the Group 1 predictions. The PD and ICE Overlay plot shows how the prediction for Miss Allen would change if you changed different inputs corresponding to her.
    If Miss Allen were male instead of female, the predicted probability for survival generated by the gradient boosting model would drop from 0.944 to 0.563.

7. From the menu next to **View chart**, change the input variable from **sex** to **age**. This generates another individual conditional expectation plot, this time for the interval input **age**.
    There are three lines, one for each individual observation and one for the average predicted probability for the PD plot (in blue). Compared to the PD plot, you see a different relationship between the input **age** and the probability of survival between the two observations (labeled **Group** in the plots). For Group 1309, you see the same relationship as in the PD plot (Group PD). That is, passengers under the age of 13 have a much higher predicted probability of survival than those over the age of 13. For Group 1, you see that age does not have a large effect on the predicted probability of survival. The behavior seen for Group 1 was not visible in the PD plot. It required the ICE plot to reveal.

Hover on the plot. The PD/ICE plot reveals how the prediction for Miss Allen's survival would change if she were different ages. Evidently, Miss Allen's predicted probability for survival would be highest if she were 48 (she was 29 on the Titanic), and lowest if she were a baby. One thing to notice is that, regardless of age, the predicted probability for Miss Allen is always above 0.90, likely due to her being a woman in first class.

8.   Explore the individual conditional expectation plots for **fare** and **pclass** to look for differences between the groups. In some cases, the results for the ICE plots and the PD plots reveal the same information.
Although the predicted probabilities of survival for Group 1 and 1309 are very different in the ICE plot for **fare**, the relationship between **fare** and the target seems to be the similar for both observations and compared to the PD. (That is, increasing the fare increases the predicted probability of survival before leveling off.) This ICE plot reveals the same relationship shown in the PD plot for **fare.**
In the PD and ICE overlay plot, we can see that if Mr. Zimmerman was in first class rather than third class the predicted probability for survival generated by the gradient boosting model would increase from 0.132 to 0.322.

9.   Close the individual conditional expectation plot to return to the Model Interpretability tab in the gradient boosting results.

End of Demonstration

Local Interpretable Model-Agnostic Explanations (LIME)

1. Fit an uninterpretable model.

    - An example is a neural network or a gradient boosting model.

2. Choose an observation (or a set of observations).

    - A local explanation is generated around this observation.

3. Create perturbed samples around the chosen observation (or set of observations).

    - These samples are used to map out the decision boundary around the observation of interest.

4. Score the perturbed samples using the original model.

    - This creates a new data set that represents the decision boundary of the uninterpretable model in the local region around the selected observation.

5. Fit a linear/logistic model to the perturbed samples.

    - An interpretable model is fit locally around the observation of interest, and the explanation generated by the interpretable model can be used to explain the decision boundary of the original uninterpretable model around the observation of interest.

The *perturbed samples* are new input data points selected randomly around the observation of interest. The local model is fit using these observations, and the observations closer to the observation of interest are weighted more heavily in this local model. In the plot, the sizes of the perturbed samples are selected so that the samples closer to the observation of interest are larger, indicating that they are more heavily weighted in the local model.

The local model is fit on the perturbed samples, and the error function is weighted so that the samples closer to the observation of interest contribute more strongly to the local model.

This approach can be used to fit local models around individual observations.

Exploring Local Interpretable Model-Agnostic Explanations

This demonstration proceeds from the end of the previous one. You are continuing to explore the model interpretability results from the Gradient Boosting node.

1. Scroll to find the LIME Explanations plot and make it full screen. Use the menu to switch between the model for **Local Instance 1** and **1309**. The LIME plot reveals that having a **sex** value of *female* and a **pclass** of 1 (first class) contribute strongly to the model predicting that Miss Allen will survive.

    For **Local Instance 1309**, Mr. Zimmerman, you see that being a male passenger in third class decreases the probability of survival. Hover on the plot elements to open a tooltip that displays the value of the nominal variables. In both cases, you find that the estimate associated with **sex** is the largest magnitude parameter estimate for the logistic regression.

2. Close the LIME Explanations plot.

End of Demonstration

Kernel Shapley Additive Explanations (SHAP)

Shapley additive explanations (SHAP) is based on the game theoretically optimal Shapley values. Shapley values help you determine the relative importance of each variable to a given observation's prediction. In the feature space, Shapley values help you determine where you are, how you got there, and how influential each variable is for that observation.

This contrasts with LIME values that help you determine how changes in a variable's value affect the model's prediction. However, LIME and Shapley values should not be directly compared because they measure different behaviors. The specification of the local regression model for Kernel SHAP follows the paper by Lundberg and Lee. In contrast to LIME, Kernel SHAP provides a guarantee of consistency and local accuracy. The SHAP value represents the contribution of a variable to the difference between the actual prediction and the mean prediction.

Specifically, SHAP values are the parameter estimates of a weighted linear regression on the predicted probability of the selected observation. One important note is that the Shapley values of all inputs sum to the predicted value. For each individual observation, an input's Shapley value is the contribution of the observed value of the input to the predicted probability of the event. These values indicate the most influential variables for a selected observation compared against a reference data set. The reference data set is typically, though not necessarily, the training data.

Exploring Kernel SHAP values

This demonstration proceeds from the end of the previous one. You are continuing to explore the model interpretability results from the Gradient Boosting node.

1. Scroll to find the Hyper SHAP Values plot and make it full screen. Use the menu to switch between the model for **Local Instance 1** and **1309**. For each individual observation, an input's Shapley value is the contribution of the observed value of the input to the predicted probability of the event "1" for the target survived. The Shapley values of all inputs sum to the predicted value. The inputs are displayed in the chart ordered by importance according to the absolute Hyper SHAP values.

    Notice that for Miss Allen, **sex** and **pclass** were the most important contributors to her predicted probability of survival.

    Similar features are shown as being top contributors to Mr. Zimmerman's predicted probability of survival. However, these inputs contributed negatively to the probability following along with our previous results that show being a male and not in first class was detrimental to survival.

End of Demonstration

References

- "Interpret model predictions with partial dependence and individual conditional expectation plots," https://blogs.sas.com/content/subconsciousmusings/2018/06/12/interpret-model-predictions-with-partial-dependence-and-individual-conditional-expectation-plots/

- "Improving model interpretability with LIME," https://blogs.sas.com/content/subconsciousmusings/2018/10/31/improving-model-interpretability-with-lime/

- "Interpreting Black-Box Machine Learning Models Using Partial Dependence and Individual Conditional Expectation Plots," https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/1950-2018.pdf

- "Why Should I Trust You? Explaining the Predictions of Any Classifier" by Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, Proceedings of KDD '16, https://arxiv.org/abs/1602.04938