

Build Models without Code with the new SAS® Viya® Visual Interface

Jim Box, SAS Institute

ABSTRACT

SAS has made it even easier to build statistical and machine learning models with a new visual interface. See how to partition data, build and compare multiple models without writing any code. Learn how to export model and scoring code for future use.

INTRODUCTION

The new SAS Viya visual interface makes it very easy to build and compare several types of models. It is an excellent tool to use when investigating potential models and performing discovery on your endpoints of interest. There are multiple ways to create models: through tasks from SAS Studio, a model building interface that is very similar to SAS Enterprise Miner, and through a visual exploration of the data. For this paper, we will focus on the visual exploration approach. Figure 1 shows the SAS Home page with tiles for actions we can take; we'll use the Explore and Visualize Data button to get started.

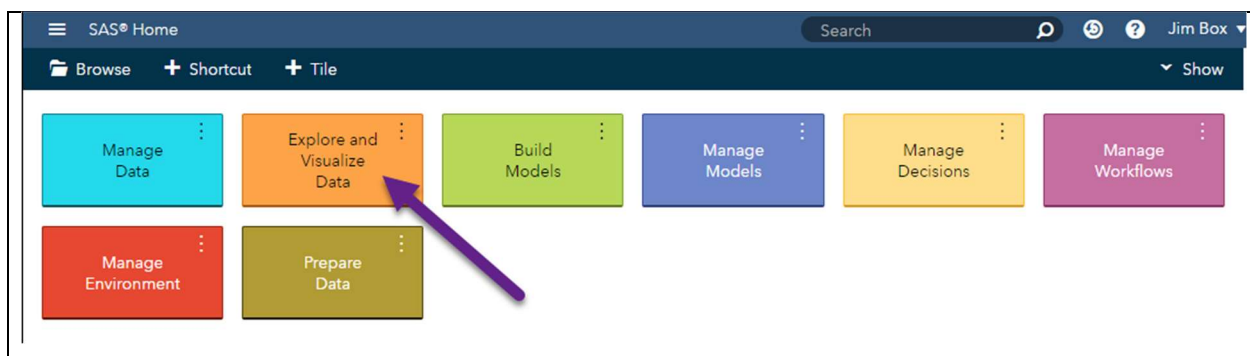


Figure 1: SAS Home Page

DATA PREPERATION

SELECTING AND EXPLORING DATA

Selecting the Explore and Visualize Data tile takes you to the data screen, where we add the data source. For this example, we will use the SASHELP.HEART data, which has results from the Framingham Heart Study. Figure 2 shows the data selection screen, and it gives us a count of the number of rows and columns. We can also get a snapshot of the data in a table to further look at the values.

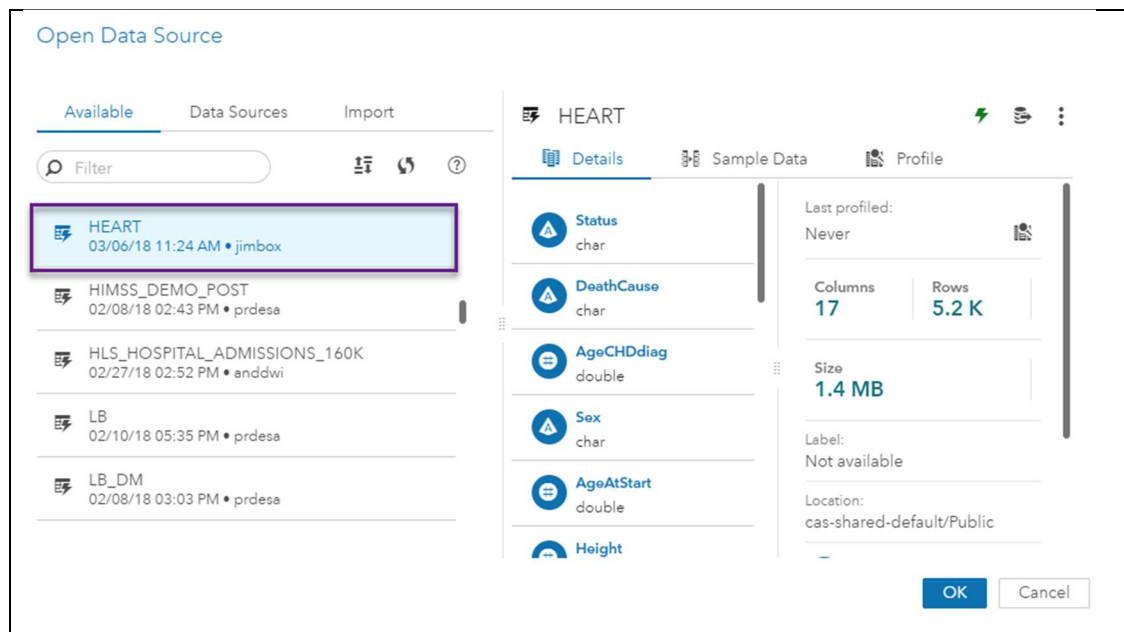


Figure 2: Data Selection

The response variable we are interested in is the Status variable – it is an indicator of survival. We'll use the exploration tools to get an understanding of this response. Figure 3 shows that we select the objects button from the left menu and select a bar chart, and then select the Status variable to get a distribution of the responses.

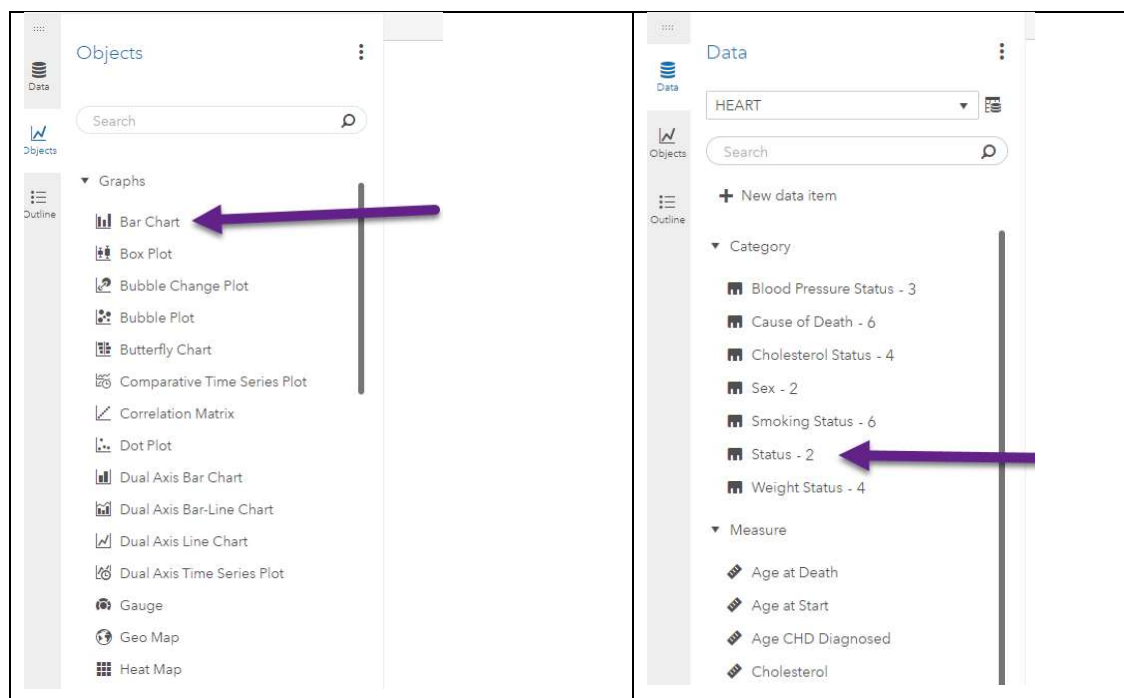


Figure 3: Create Bar Chart of Status

We can set up the bar chart a variety of ways by choosing different Options and Roles. Figure 4 shows the results of using vertical bars and showing Data Labels in the Options menu and selecting both Frequency and Frequency Percent as Measures in the Roles menu. As a result, we see that just over 38% of the records show a status of “Dead.” We would like to explore some different model options and fits a good model that predicts death as a status for future patients.

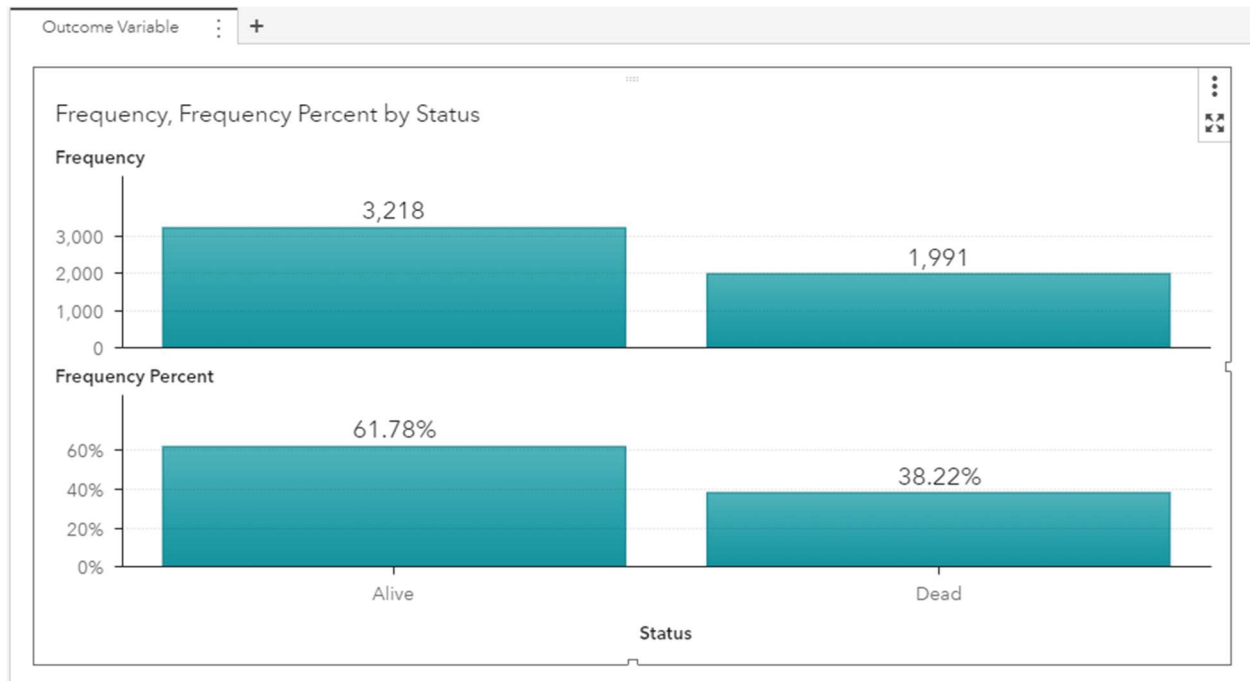


Figure 4: Endpoint Breakdown

PARTITIONING DATA

When developing models that will be used for prediction, it's important to evaluate their predictive power. It's also important to not overfit the data – we want the model to be useful at predicting outcomes for data that was not part of the construction of the model. To that end, it's a modelling best practice to partition the data into at least two segments – one to develop the model on (the training data) and one to use to evaluate the model's ability to predict outcomes for new observations (the validation data). People generally use 60-70% of the data to train and the remaining to validate.

If a partition variable exists in the dataset (usually coded as a 0 or 1, randomly assigned), we can identify the variable as a partition data type, but generally we'll need to make a partition variable. Figure 5 shows how we accomplish this. First, we click on the data table and select the “Add partition data item ...” option. Then we give the partition column a name, tell it that we want to use two partitions, and set the training partition size to 70%. If we wanted, we could identify the random seed to use to create this partition; here we will leave it blank.

Now that we have the data selected and partitioned, we are ready to start building models.

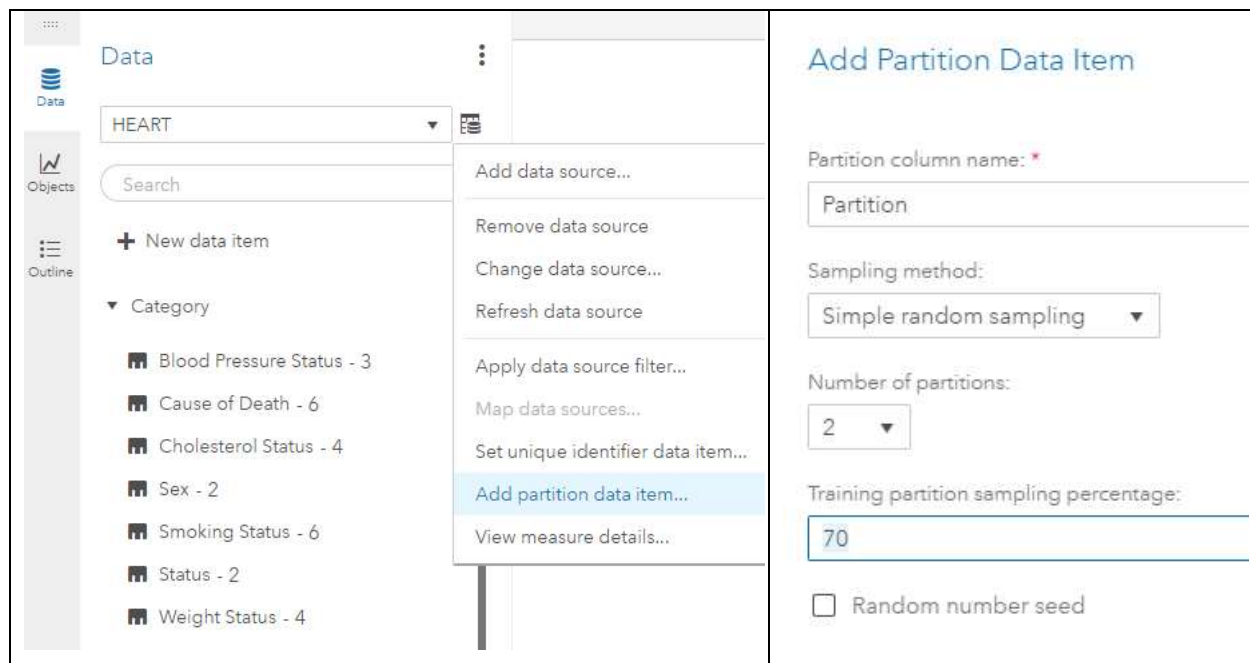


Figure 5: Partition Creation

MODEL BUILDING

There are several types of models we could build to predict the status variable. For this demonstration, we will make four different models with the training data, then have the system compare the four models and tell us which one did the best job correctly predicting the status of patients in the validation partition. There are two general types of models we might try out: statistical models and machine learning models.

STATISTICAL MODELS

The first approach is to look at some of the traditional statistical models that could be used. Figure 6 shows the different model types we could use. Since our response variable has a binary outcome, we will use the two main approaches for this type of data: a Decision Tree and a Logistic Regression.

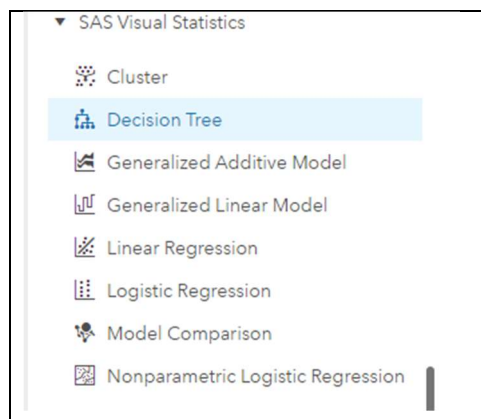


Figure 6: Statistical Models

Decision Tree

We start with a decision tree, as it is generally the easiest model to understand. To build the tree we'll add a new tab to our exploration, go into the objects menu and select the decision tree in the list we saw in Figure 6. Next, we go to the Roles tab on the left menu and set the Response variable to Status and the Partition ID to Partition. Finally, we will select the Predictors we want to consider. Decision trees can take both category and measure types of variables. It is a best practice to not select multiple variables that address the same thing. For example, in Figure 7 we are selecting Smoking (measured in number of cigarettes) but not Smoking Status, which is a category variable made from the smoking variable.

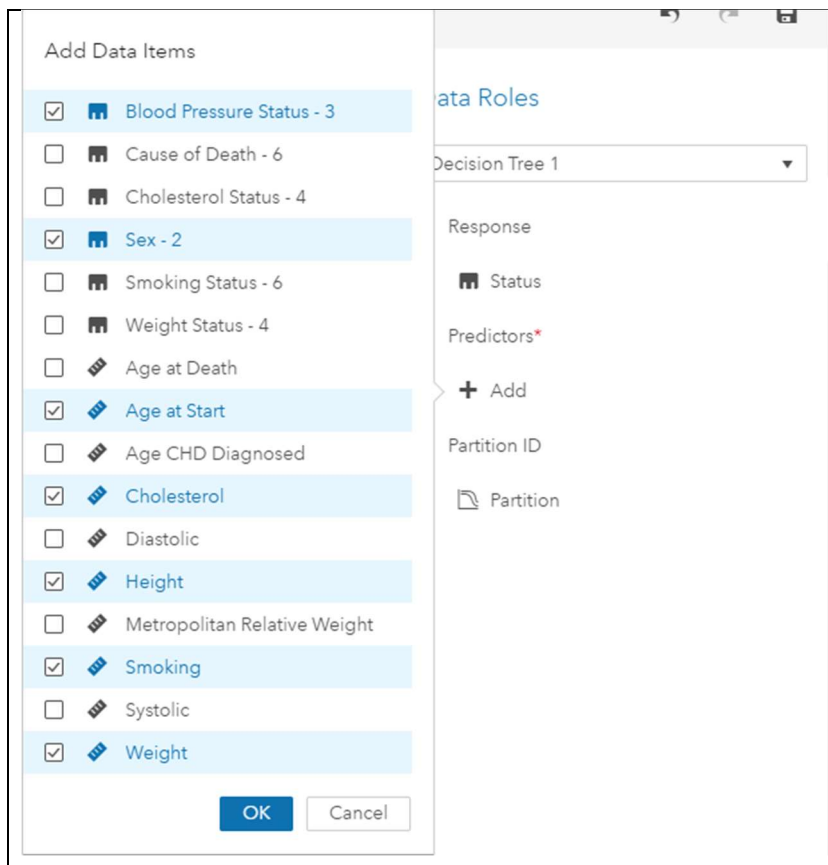


Figure 7: Variable Selection

As soon as the variables are selected, the decision tree is generated and output is produced in the tab.

Figure 8 shows the tree. The first split was on age, indicating that this variable was the most important in predicting the status. The data was cut on the value of age of 46.7, with 62% of patients older than that having a status of Dead (compared to the overall 38%). Note that we can choose the model comparison statistic we want to look at. Here we have chosen to look at the misclassification rate in the validation partition. This is telling us that the model was built on the training data and when it was used to predict the status of the validation subjects, it was wrong 27.96% of the time. This statistic will be useful when we are comparing all four models at the end of the process.

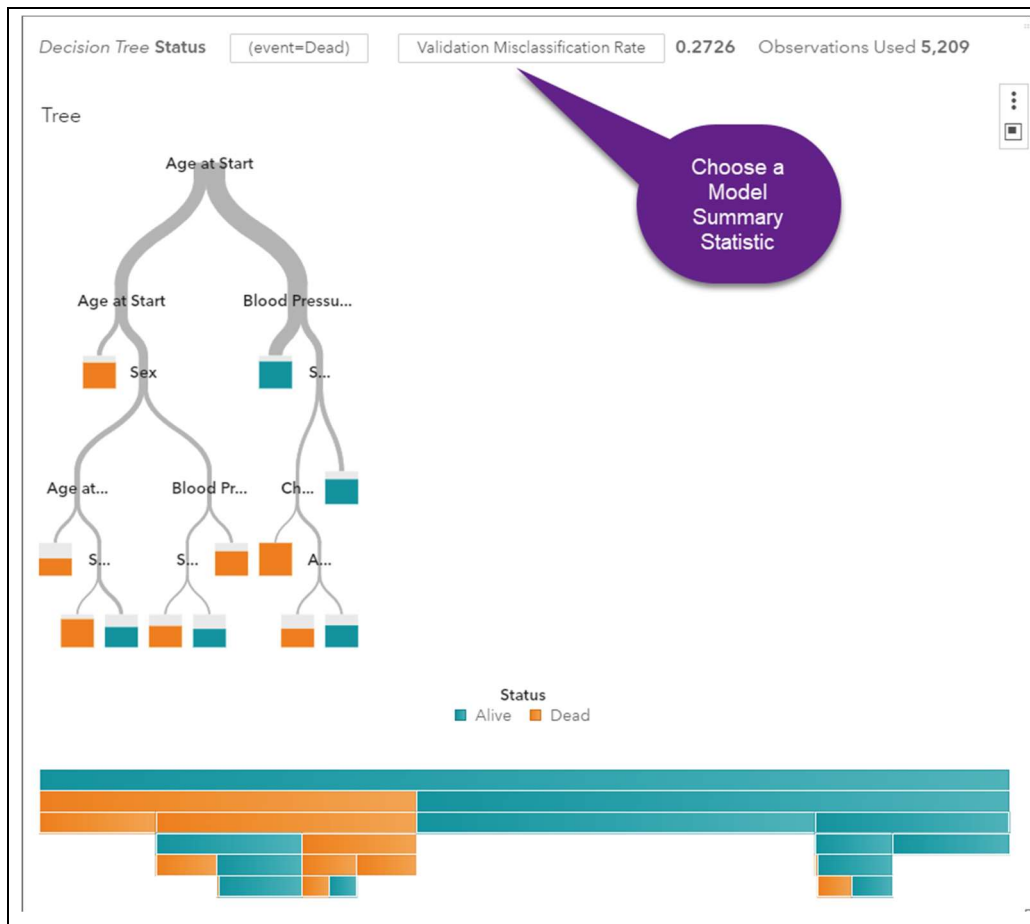


Figure 8: Decision Tree

Logistic Regression

The next model we will build will be a logistic regression. We could add a new tab and then assign roles like we did with the decision tree, but we have a shortcut available to us that makes it easier.

Figure 9 shows the way to do this; to get to that menu, we will click on the three vertical dots we see at the top right on the decision tree we see in Figure 8 (this three-dot menu is sometimes called the snowman menu). When the menu pops up, hold down the ALT key and the “Duplicate as” selection becomes “Duplicate on New Tab as”. Here we will select Logistic Regression, and all the Response, Predictors and Partition selections we made on the Decision Tree will be copied onto the Logistic Regression roles in a new tab. If we did not hold down the ALT key, Logistic Regression output would have been added below the Decision Tree, which makes it hard to see both models.

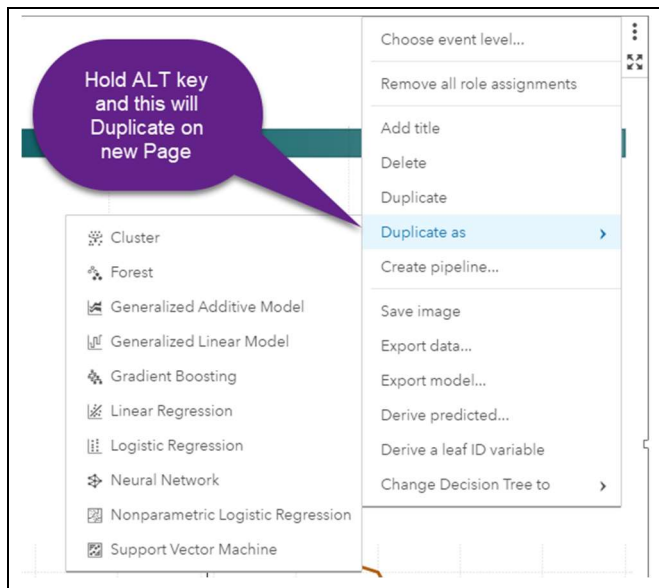


Figure 9: Duplicate Models

The Logistic Regression output is shown in Figure 10. We have selected the same validation misclassification rate to be the model statistic; it's a little better than the Decision Tree. Remember that a logistic regression model requires that there be no missing values in any of the selected predictors; any patient with a missing value is excluded. In this case, that meant we had to throw away 170 of the patients. That's a small number here, but it is important to consider when we are looking at other datasets.

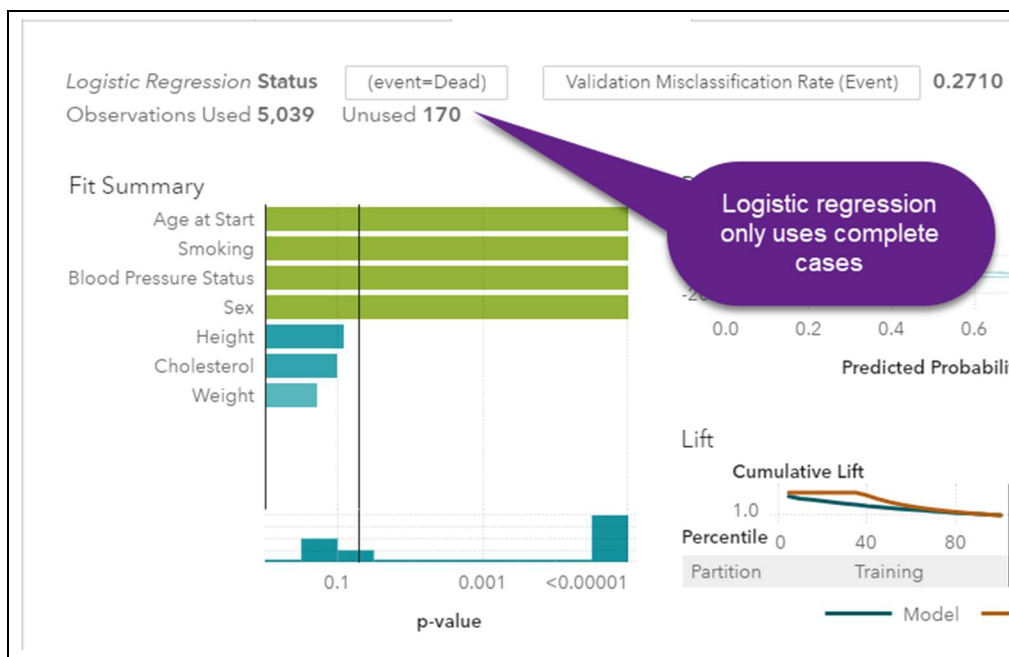


Figure 10: Logistic Regression

MACHINE LEARNING MODELS

In addition to the traditional statistical models, we also can use some of the more advanced machine learning models. Figure 11 shows the types of machine learning models that are available. They will often have less restrictive assumptions than do the statistical models, and it is helpful to read up on how they work before considering them as candidate models. Here we will use two of the most commonly talked about machine learning models: Gradient Boosting and Neural Networks.

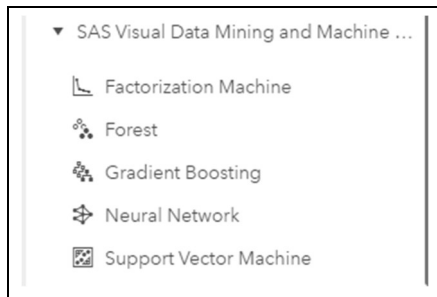


Figure 11: Machine Learning Models

Gradient Boosting

Gradient Boosting is a popular machine learning technique for this type of classification problem. It's a complex model, but for our purposes, we can think of it as a technique that uses a bunch of different decision trees to make a better, combined predictive model. We won't get into the discussion of how to tune parameters here, but the Options menu gives us a great deal of control over how the model is implemented. We'll just stick with the defaults and look at the outputs. Figure 12 shows that output; note that we see the same sort of variable importance that we have in the Decision Tree, and that we can still look at the Validation Misclassification Rate.

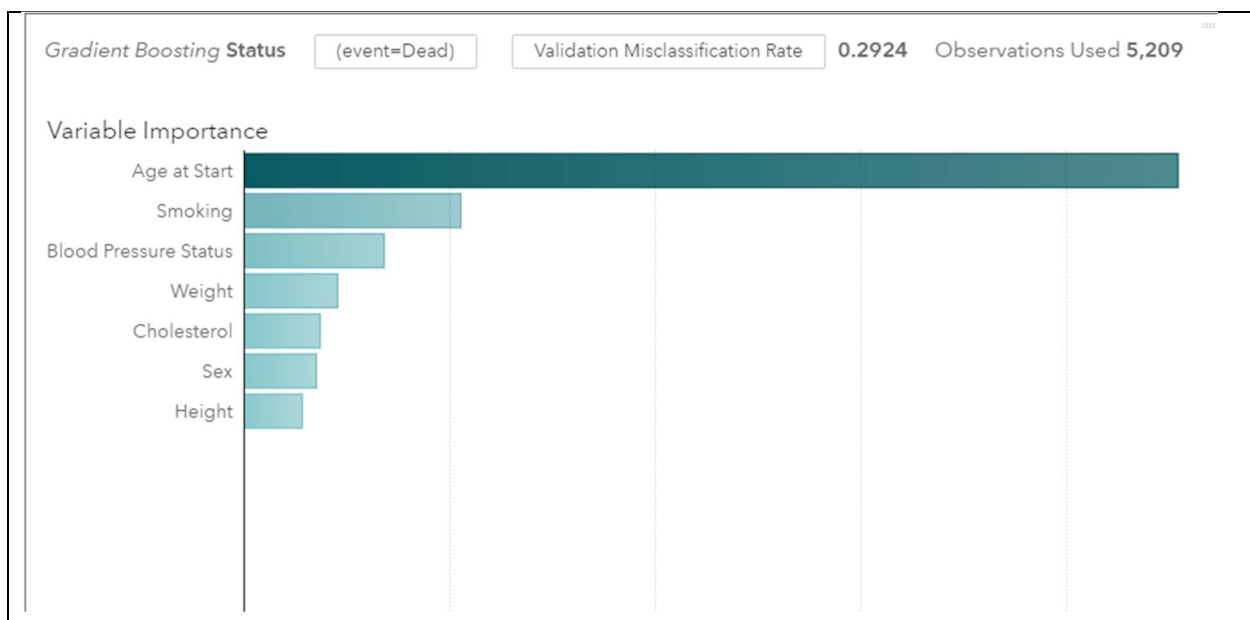


Figure 12: Gradient Boosting

Neural Networks

The final model we will add is probably the most complicated: the Neural Network. A high-level view of a Neural Network is that it is a collection of nested regressions to get to the predicted variable. Like the Gradient Boosting model, we can just use the model defaults for this exercise, but there are a variety of options we could adjust to try to fine-tune the network.

As before, we can see the Validation Misclassification Rate in Figure 13. Notice that this model also requires complete cases, and 170 observations were not used.

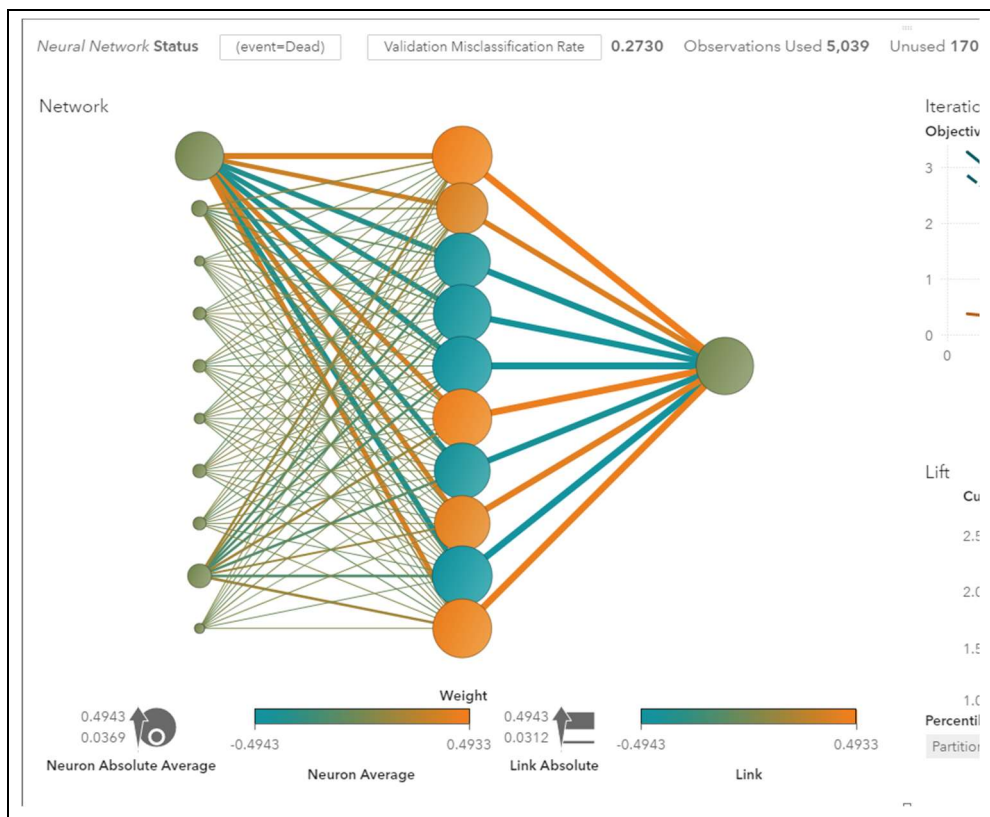


Figure 13: Neural Networks

MODEL SELECTION & SCORING

Now that we have created four candidate models, it's time to do a model comparison. Figure 14 shows the process. First, we add a new tab to the exploration, then select the Objects menu on the left. In the Visual Statistics menu, select Model Comparison, and a dialog box pops up. The data source, Partition, Response, Event Level and Group By selections are all made by default to match the first model. All models that match that criteria show up in the Available Models section (you cannot compare a model that has a validation partition with one that does not, for example). Here we selected all, and the results were added to the tab.

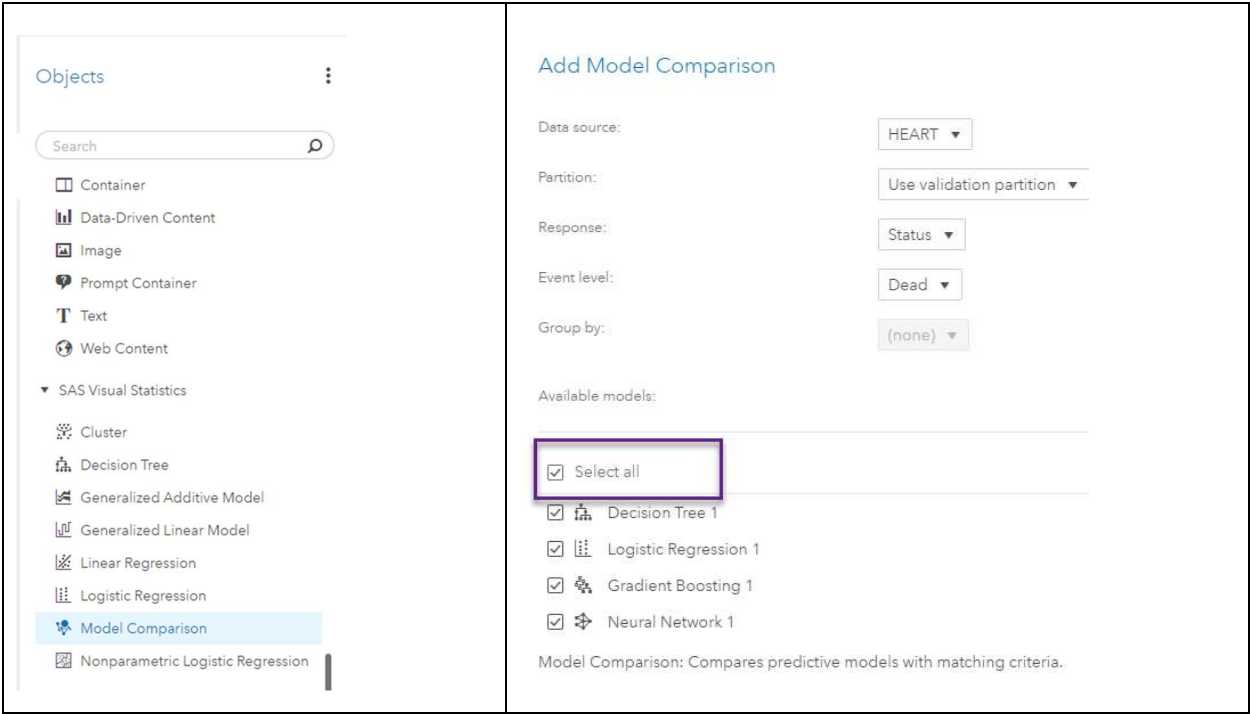


Figure 14: Model Comparison Setup

The model comparison results are shown in Figure 15. Also note that by right-clicking on any of the model bars, we can change the statistic we use to compare the models. There are several available, on both the training and the validation partition. We've chosen the Validation Misclassification Rate, since an accurate prediction is how we want to select the final model.

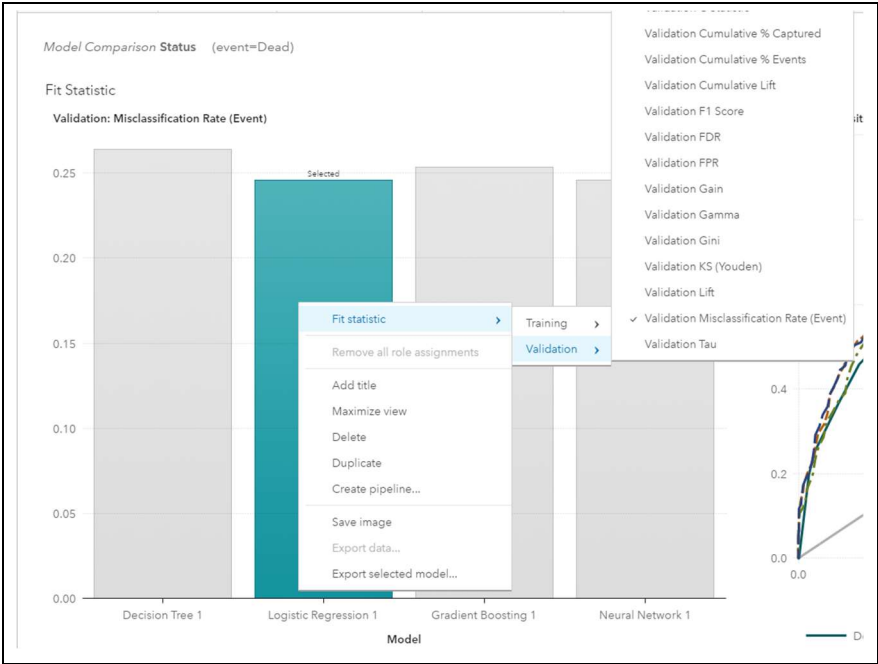


Figure 15: Model Comparison Results

Based on our model selection criteria, the Logistic Regression is the winner. Note that there is not a lot of difference between these selections, so if we had a valid reason, we may choose to implement a different model. The two statistical models are much easier to explain and implement than the two machine learning models, so that may be an important factor. Decision Trees (and Gradient Boosting) allow for missing values, so if we have a concern about future data missing some values, we may make that a key component of the choice. The model selection decision should be a combination of statistical and practical decisions. Once we have made that decision, we can export the score code and set up a process to implement it for future observations. As Figure 16 shows, right-click on the bar and select Export Specific Model.

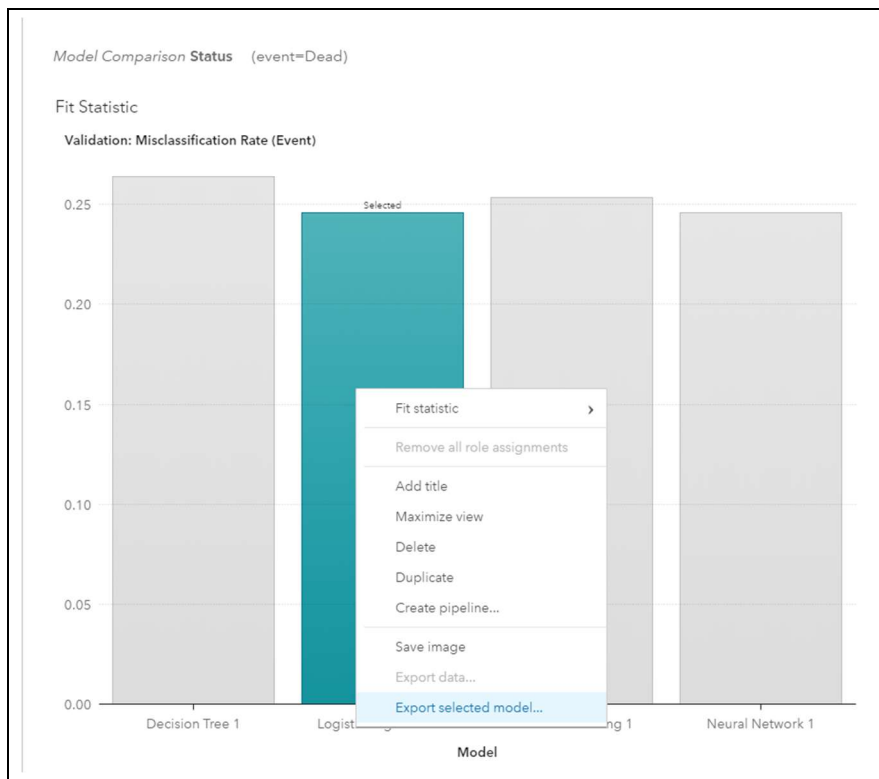


Figure 16: Exporting Models

Once the export has been selected, the box in Figure 17 appears. This contains the SAS code that would be applied to new observations and would generate a predicted value for Status. We would use this to evaluate new patients who were being seen and evaluated for risk of death. Once we have observed several more outcomes, we could redo the entire process with the old and new data and refresh our models.

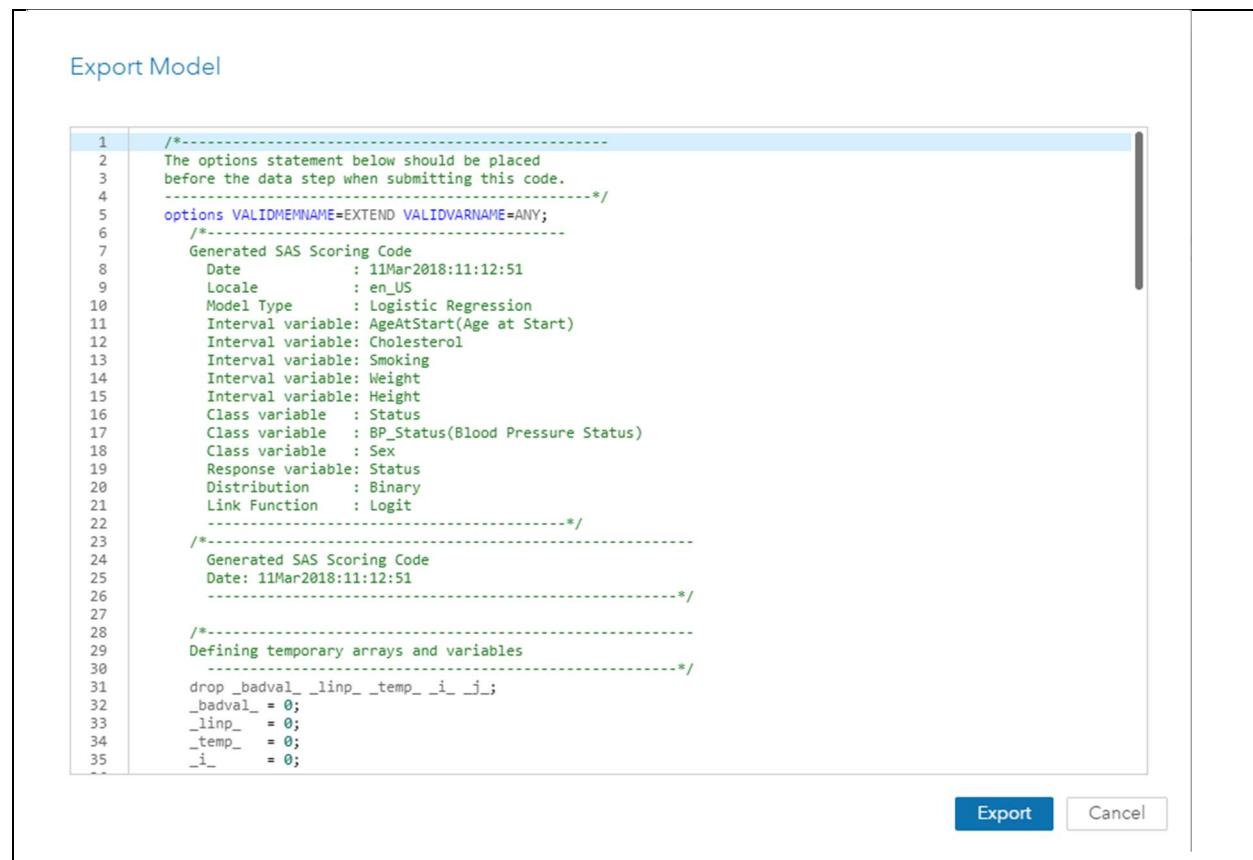


Figure 17: Exporting Code

CONCLUSION

The new SAS Viya interface makes it very easy to produce, evaluate and export models. There are options for traditional statistical methods and for newer, more complex machine learning models. Users can have a lot of control over the model options, or they could accept the defaults provided. There are other ways to build models without code in this interface (see the Build Models tab on Figure 1), but the Visual Exploration method is the easiest way to interact with complicated models.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jim Box
 SAS Institute
 Jim.Box@sas.com

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.