MACHINE LEARNING LAB 01

STEFANO FOCHESATTO

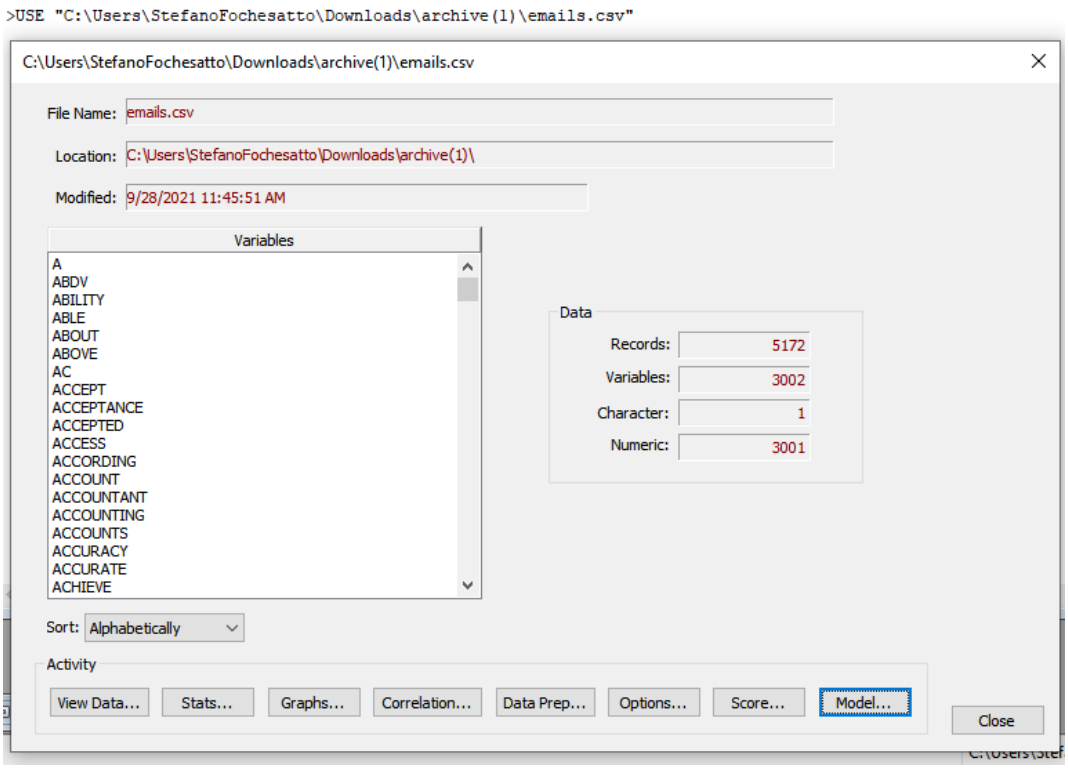
1. Introduction

In this lab we will build a CART model to detect whether or not an email is spam. In doing so we will explore SPM's CART Decision Tree Analysis Engine, and develop a data workflow with SPM. The data used throughout this lab can be found on Kaggle and is free to use for educational purposes under the Open Data Commons Open Database License. The data comes in the form of a '.csv' file with 5172 rows and 3002 columns. Each row represents data for a single email, the first column is used to number each email, the last column contains our prediction information (1 for spam, 0 for not spam) and the rest of the columns contain word counts for the 3000 most common words found across all emails.

Preliminary Analysis (Loading Data). Upon loading the data into SPM using either the file menu, or the USE"" command, we are met with a variety of options to perform exploratory analysis and data preparation in the 'Activity' pane. For the most part the most important tools for exploratory analysis are the View Data, Stats, Graphs, and Correlation options in this menu (these options can also be found under the View and Explore menus). With these options we can view histograms, evaluate correlations, and compute univariate statistics like the Mean, Min, Max and IQR. Not entirely relevant to our lab but usefully nonetheless.

Date: September 30, 2021.

FIGURE 1. SPM Activity Pane

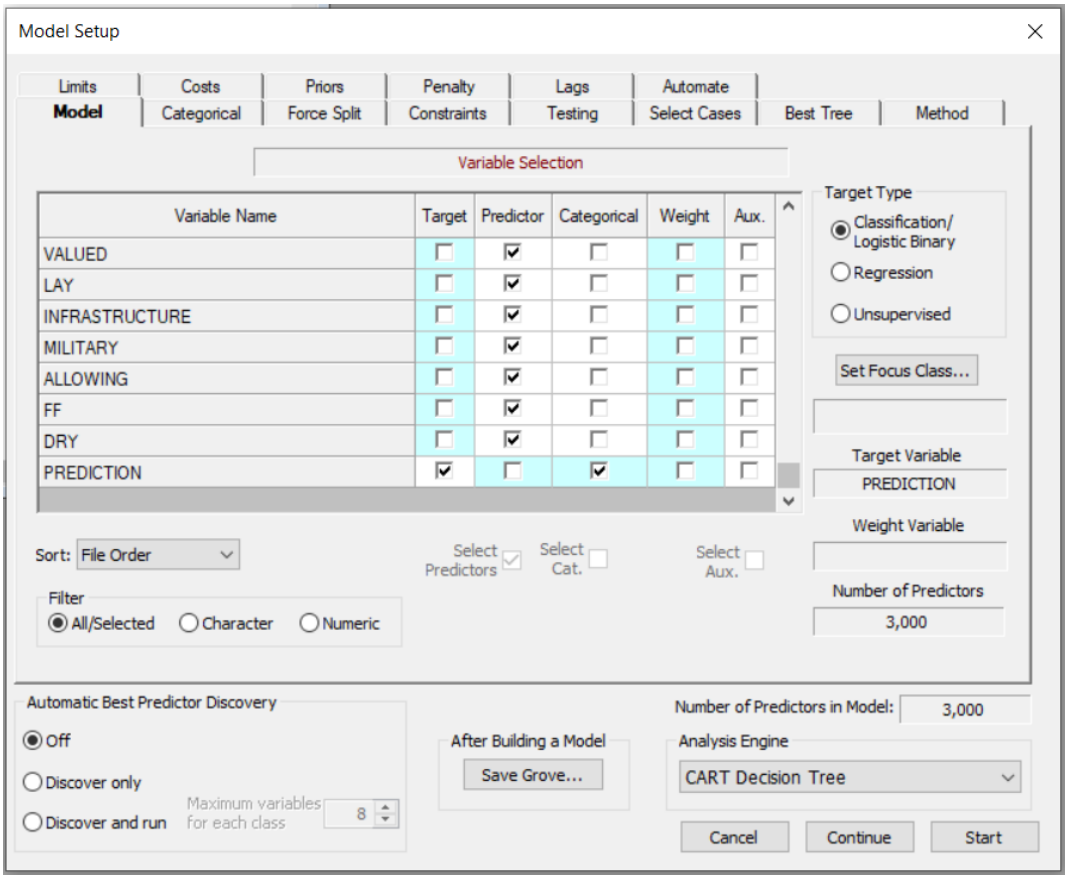


The Data Prep option allows us to do any preparation like handling missing data, duplicate data, unwanted outliers, or even fixing structural errors in our data using built in SPM commands. The Options menu gives us access to the content of our Model's text report, as well as the random number seeds for each SPM modeling engine. The Score menu allows us to test the data against a given model in the form of a grove file.

2. SPM CART Model

The Model menu is where we will be doing the majority of our work, tuning parameters to find the best model. In this menu we will decide which variables in our data will be our predictors and which variable is the response. We will also decide which modeling engine we will use, in our case we will explore the CART Decision Tree engine and since we are in need of a classification model we select the 'target type' to be "Classification/Logistic Binary".

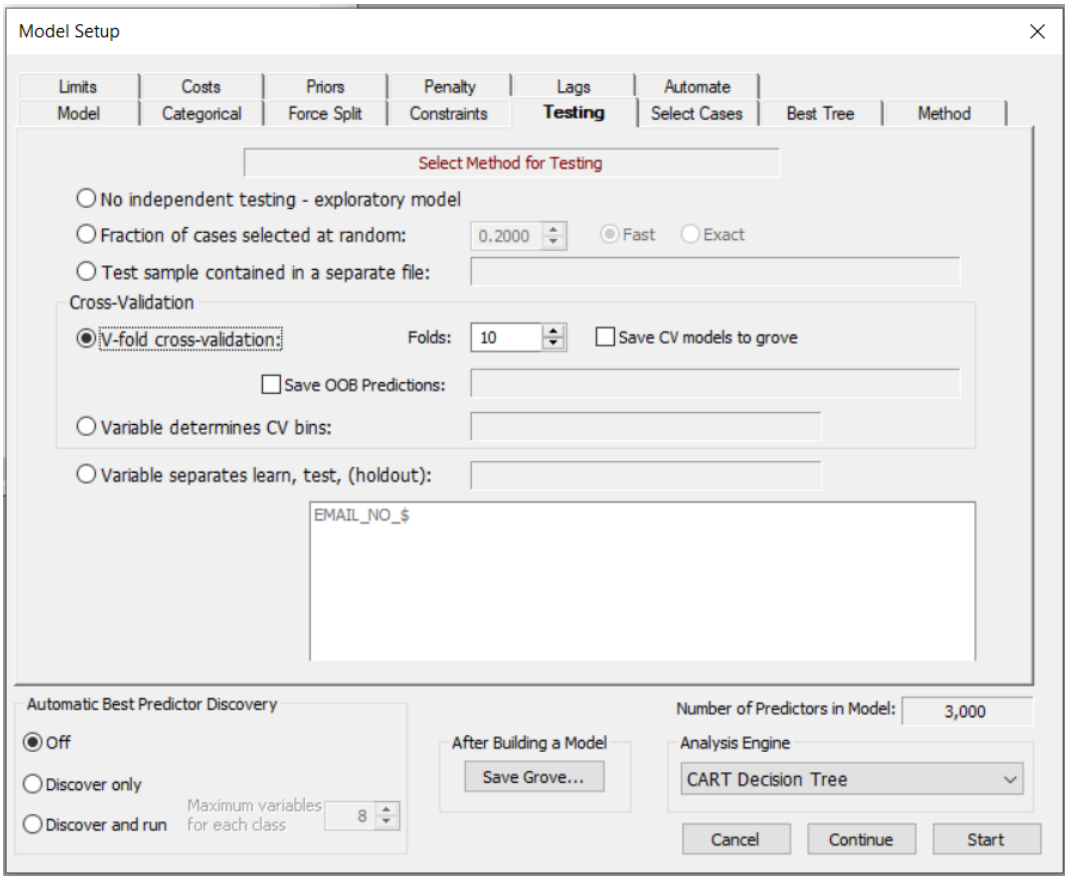
FIGURE 2. SPM Model Menu Pane



From here we want to explore how we can tune the basic parameters for a CART model. Parameters like Cost Function (Node-Impurity), Tree size (in terms of depth and number of nodes), our testing/training schema, and pruning parameters for selecting the best tree.

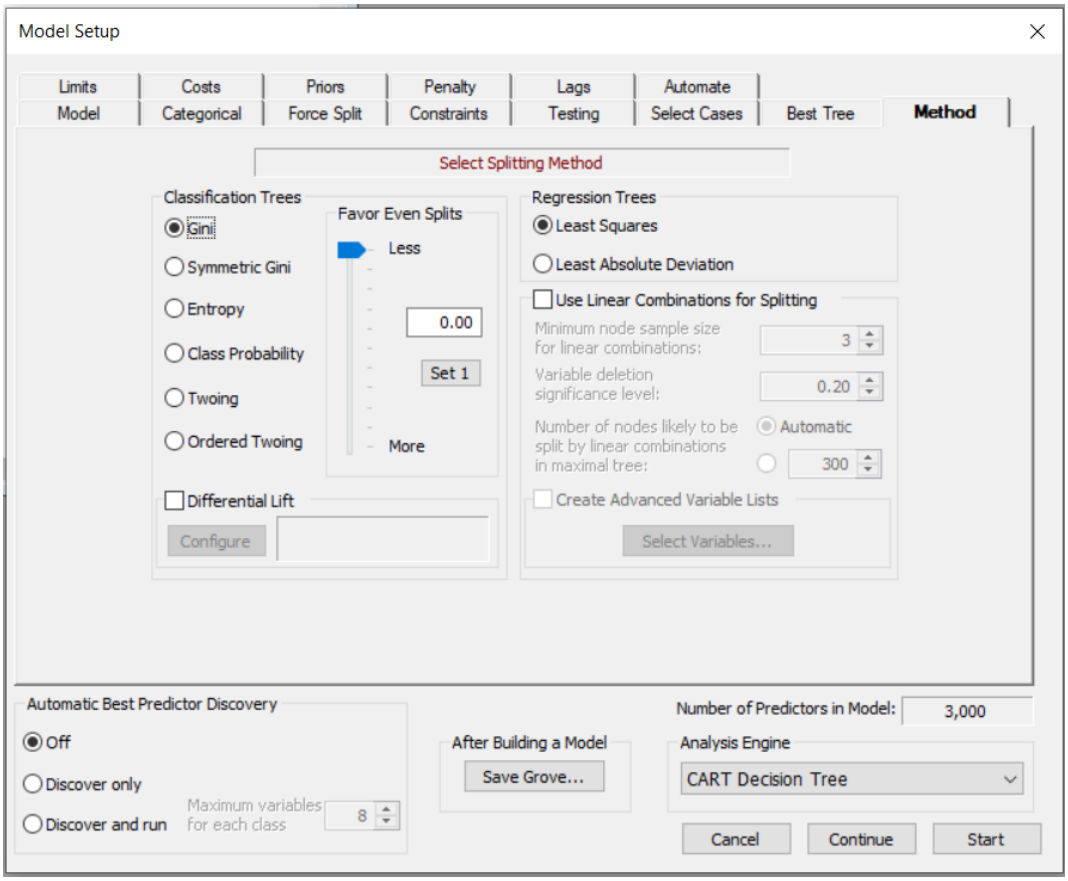
In the Testing menu we can choose from a variety of different testing schema. There is an option for no testing, partitioning via a percentage, selecting particular holdout data, and V-fold cross validation. For our model we will be using V-fold cross validation.

FIGURE 3. SPM Model Testing Pane



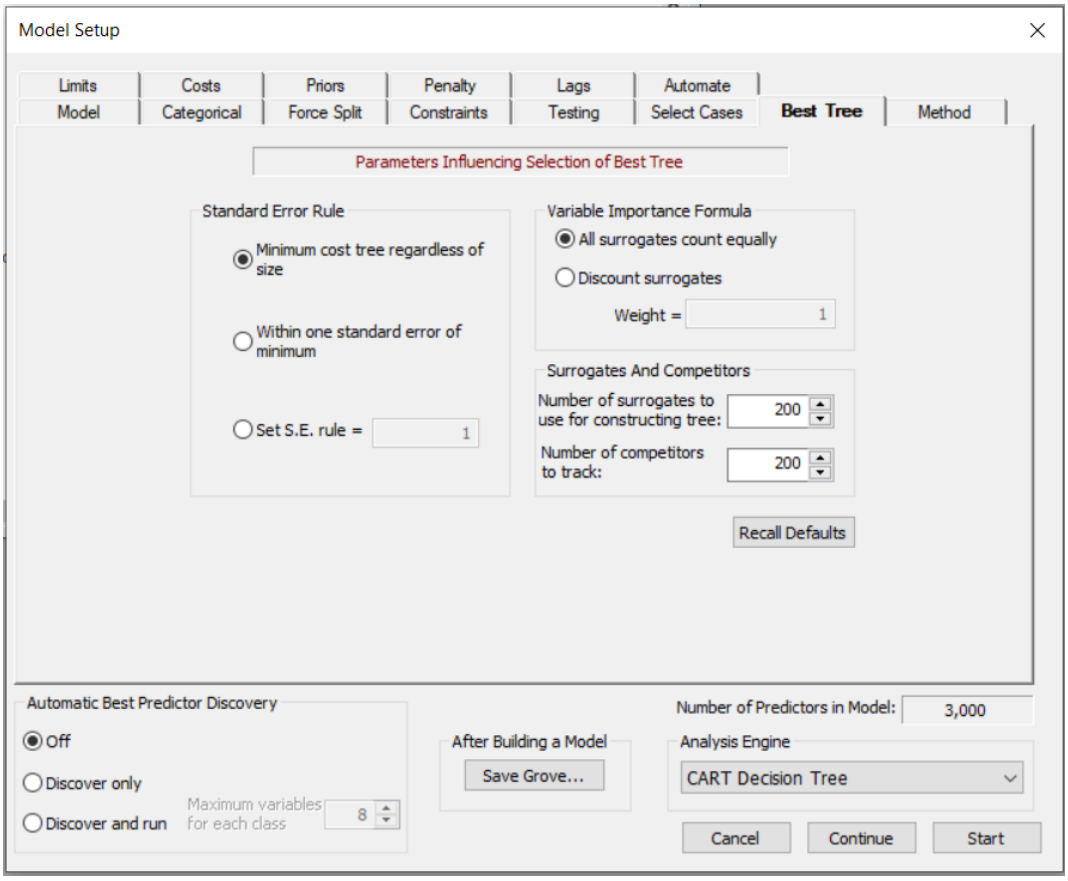
In the Method menu we can explore how CART model will evaluate node impurity. For our model we will use the Gini index for evaluating node impurity.

FIGURE 4. SPM Model Method Pane



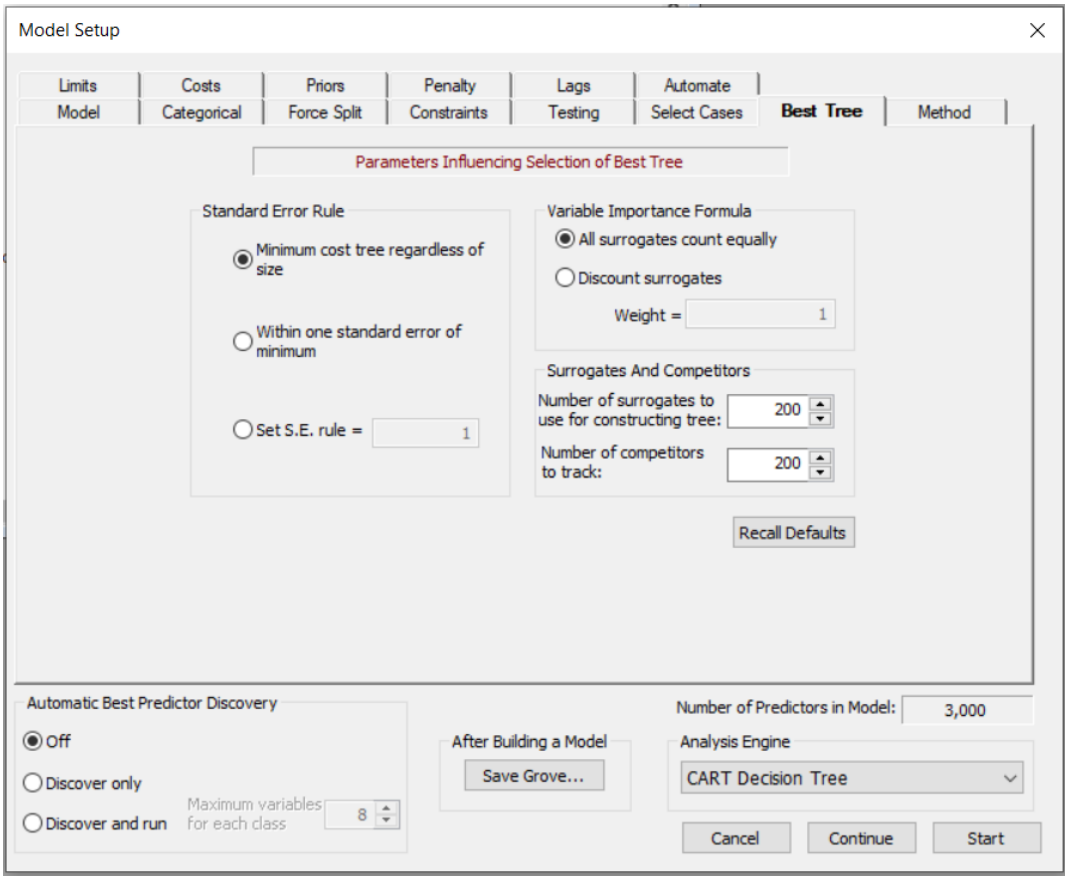
In the Best Tree menu, we can influence how SPM decides on the optimal pruning parameters to produce the best tree. For our model we will give us the minimum cost tree, but if we wanted we could prune our tree even further by finding the minimum cost within one standard deviation. In some cases this might be useful, regardless we will still be able to prune and grow our tree after the fact.

FIGURE 5. SPM Model Best Tree Pane



In the Limits menu we get access to our maximum tree depth and maximum number of nodes, minimum node sample sizes, and our learn and test sample sizes. These paramaters will have the biggest effect on out CART model, so we will let SPM find the most optimal values. With other software we would likely have to some sort of gridsearch to find the optimal paramaters.

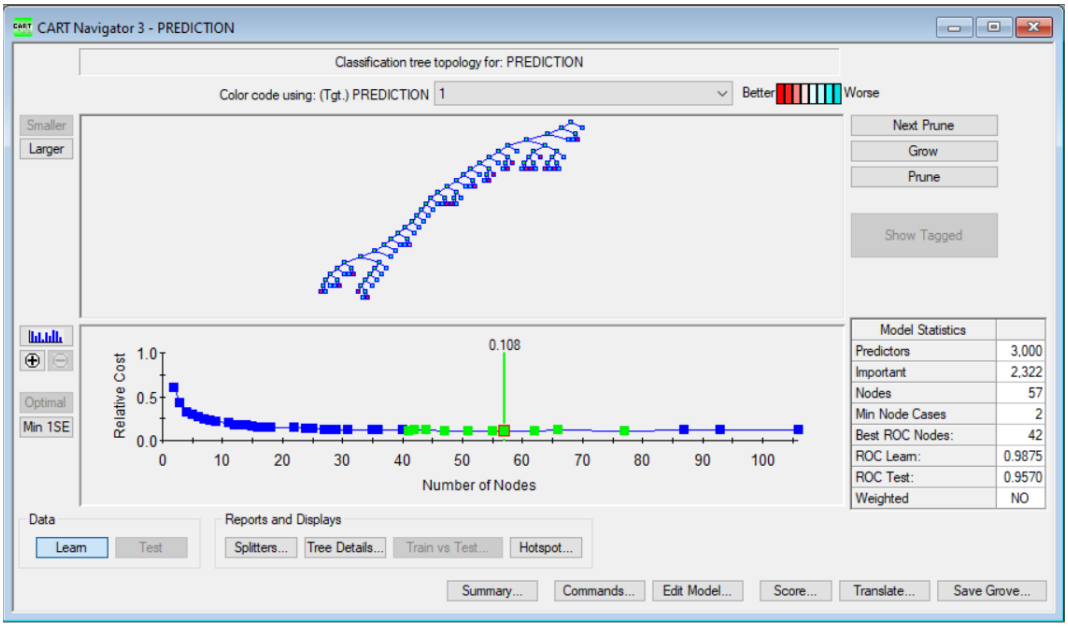
FIGURE 6. SPM Model Limits Pane



3. PERFORMANCE ANALYSIS

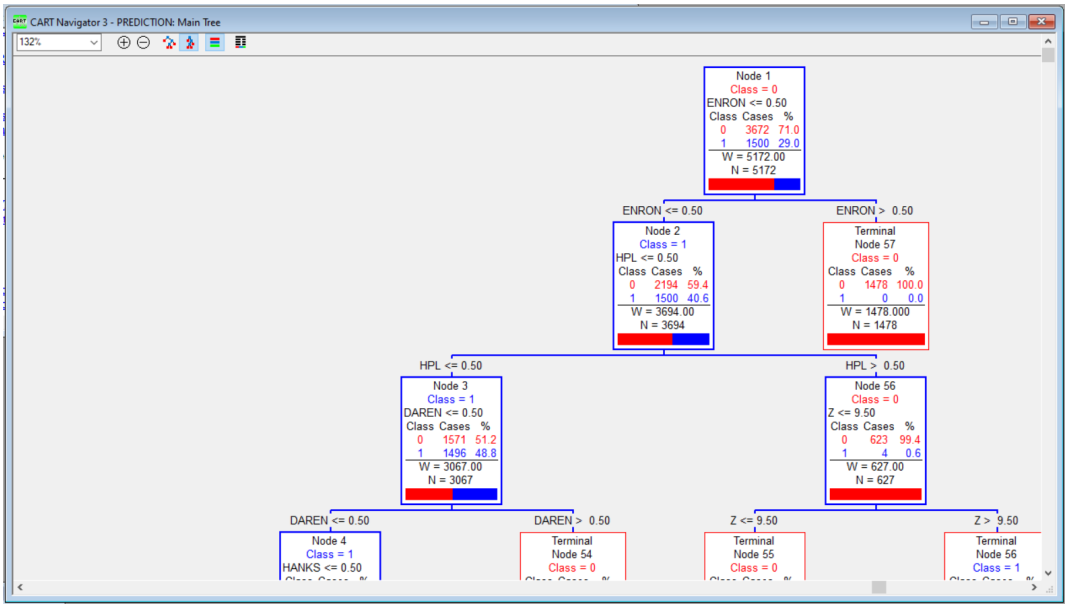
With all other parameters set to the SPM default we can generate our model. Doing so we arrive at the following screen.

FIGURE 7. SPM Model Results Pane



From here we can press on the button to the left of our Cost vs Number of Nodes graph to display a Standard Error interval, and tapping on the Min 1SE button we can view the minimum CART model where the relative cost is within one standard error. For our analysis we will continue with the minimal cost model. For specific details of each node we can look through the Splitters, and Tree Details menu, which give us a visual representation of the CART model with index and splitting criteria,

FIGURE 8. SPM Model Tree Details Pane



Finally we looking at the Summary menu we can get a measure of our model's performance. We can see that our model achieved a ROC score of .957 on the test data and a .987 on the learn data. In this menu we are given a confidence interval for the ROC score, as well as the variance should we need further analysis. Moving into the Gains/ROC menu we can actually get a plot of the ROC curve,

FIGURE 9. SPM Model Summary Pane

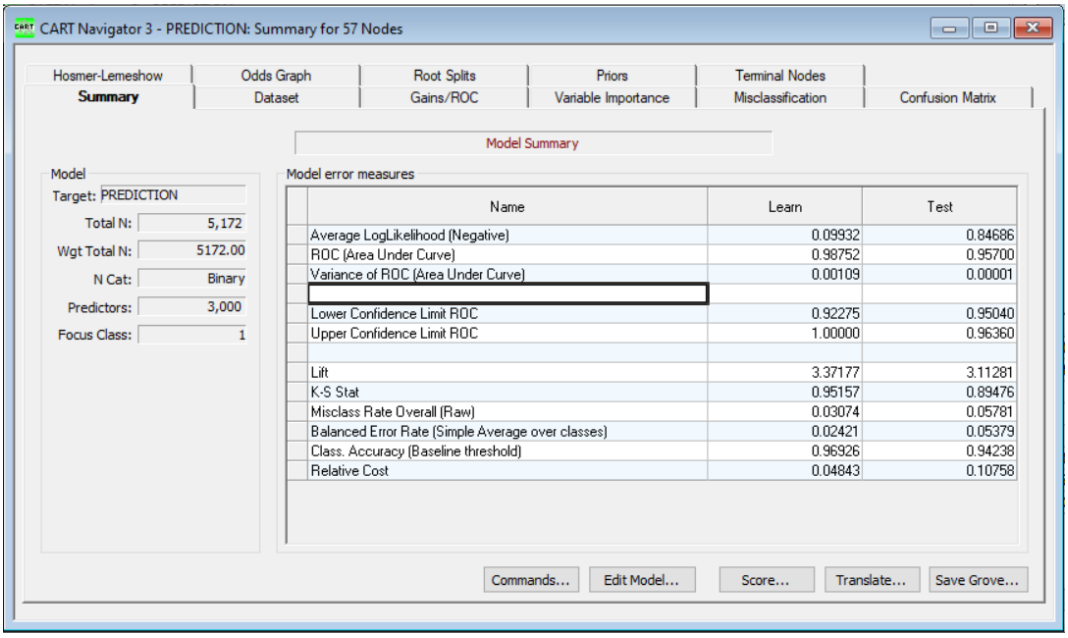
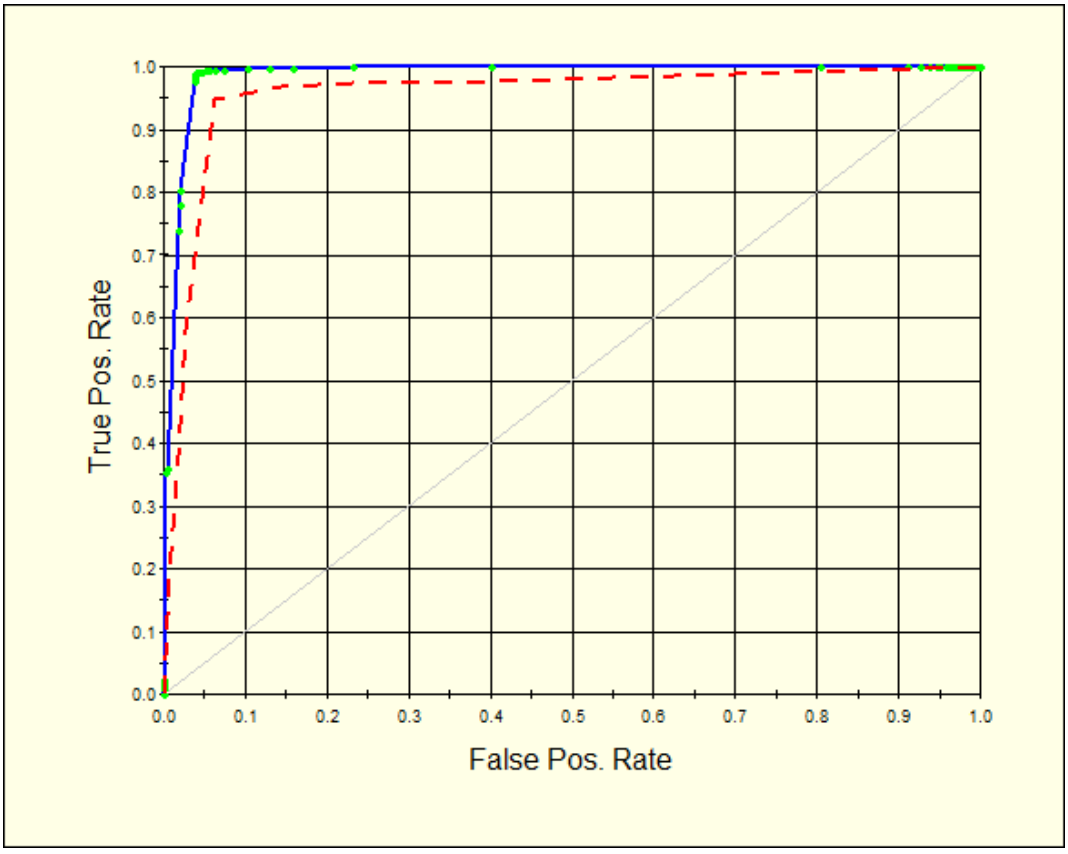
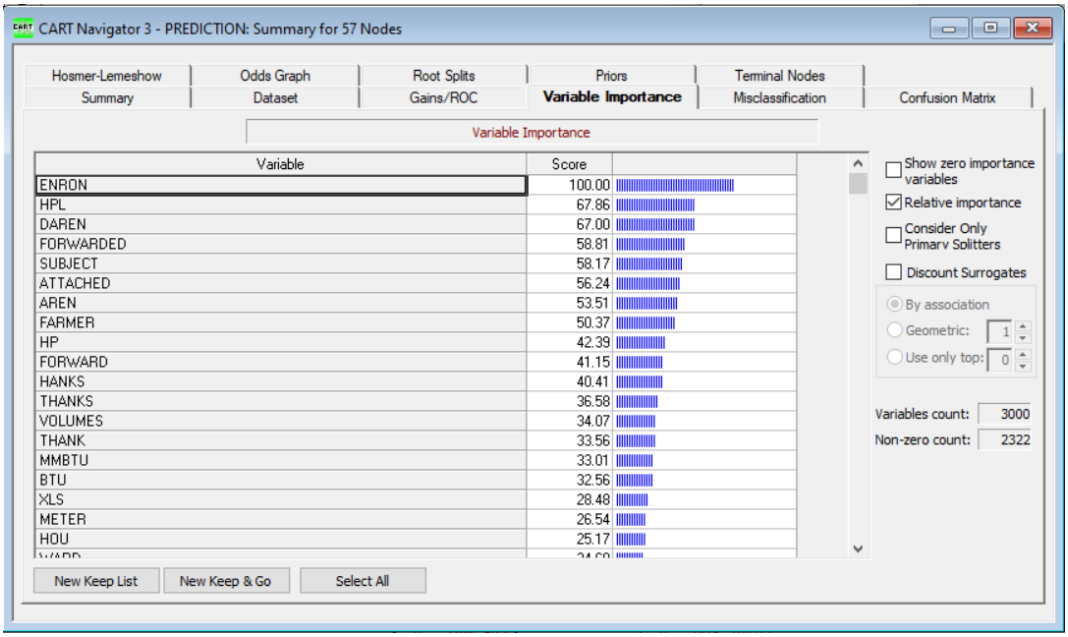


FIGURE 10. SPM Model ROC Curve



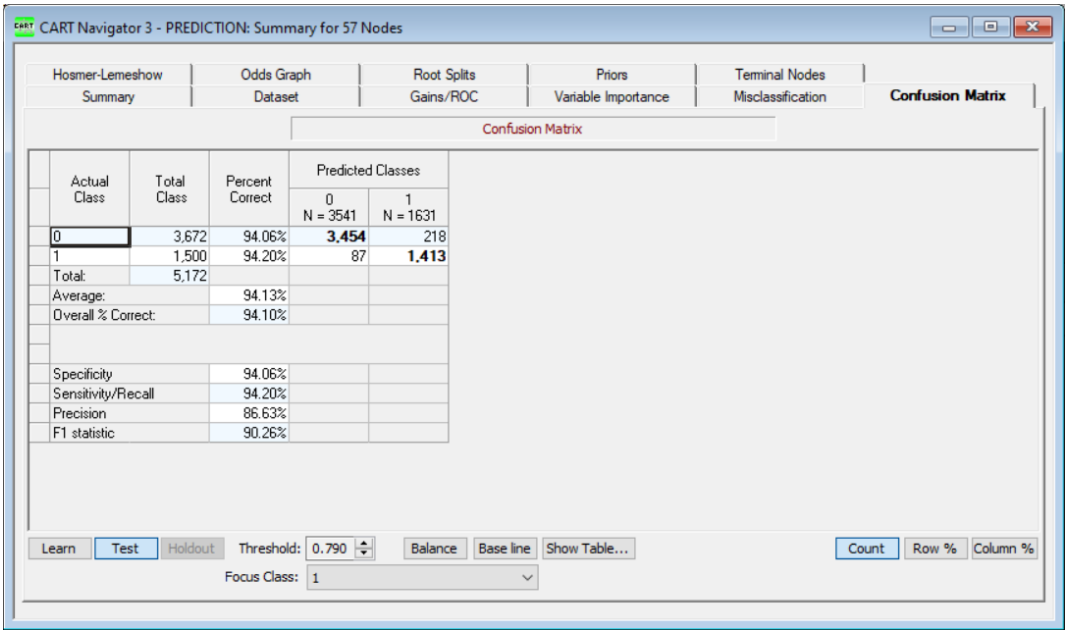
We can view the importance of each variable in the Variable Importance menu. This menu shows us that the word "Enron" had the greatest importance when determining whether an email was spam or not. Looking back at the tree diagram we can see that if there was any presence of the word "Enron" the CART model would correctly classify 1400+ emails as not spam.

FIGURE 11. SPM Model Variable Importance Pane



We can view the Confusion Matrix for the test and learning data in the Confusion Matrix menu. From the training data our model misclassified 87 spam emails as non-spam and 218 non-spam emails as spam.

FIGURE 12. SPM Model Confusion Matrix Pane



4. CONCLUSION

SPM is an incredibly robust and intelligent piece of software that allows the user a better mental image of the model as it's being built and tuned. Compared to other platforms, I've found the iterative process of making a model and tuning it for better performance is much faster and simpler using SPM. However it does seem as data manipulation and scrapping is a lot more robust and integrated on other platforms, machine learning with python. For further analysis access to the '.grv' file and data can be found in here.