# MACHINE LEARNING LAB 02

STEFANO FOCHESATTO

## 1. Introduction

In this lab we will explore the TreeNet and RandomForest modeling engines in the SPM software. To do so we will build the same email spam classification model from the previous lab. The aim of this lab is to become more familiar with the TreeNet and RandomForest engines, beyond that we will also build a workflow for exporting data and figures from SPM. Generally, we hope to gain a deeper understanding of how these ensemble models work by exploring their parameters and evaluating their performance.

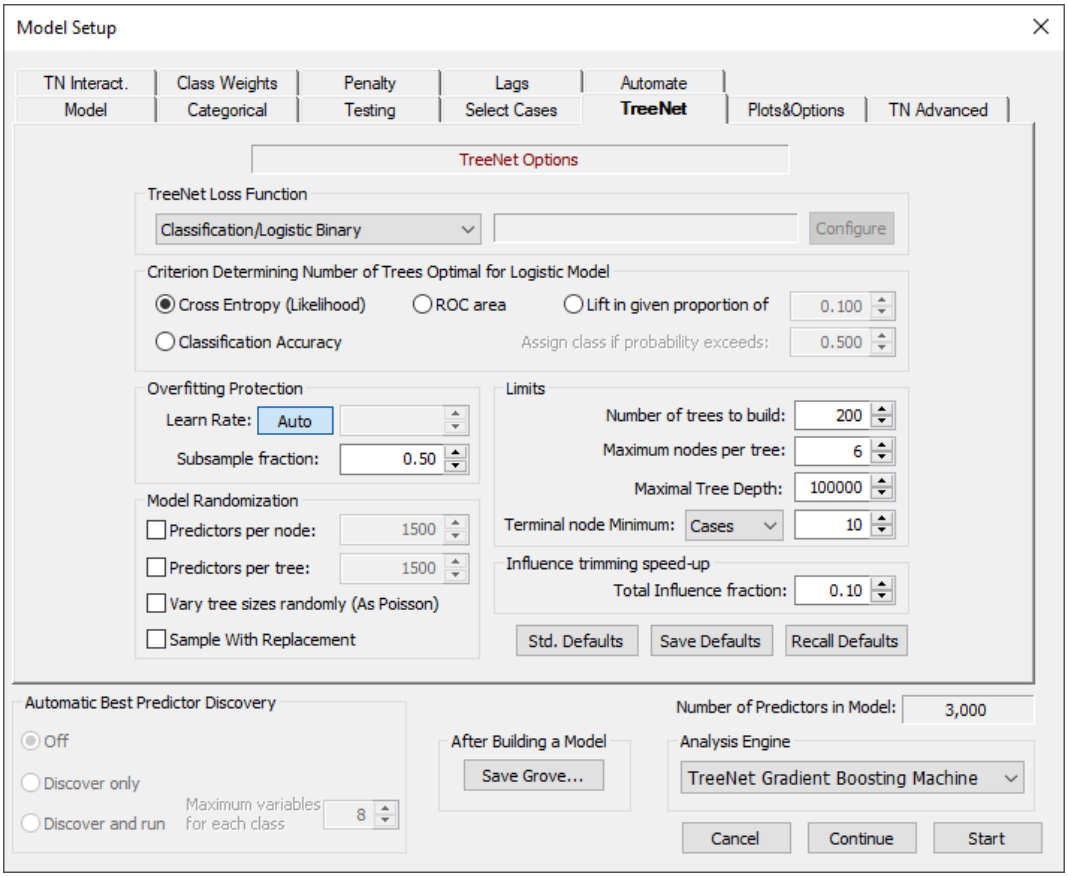
**Preliminary Analysis (Loading Data).** Recall that the workflow for performing basic exploratory analysis like computing correlations, and other univariate statistics, as well as data preparation in SPM was covered in lab #1. These tools are not entirely relevant to the subject of the current lab, however we should note that in lab #1 we concluded that most data preparation and basic exploratory analysis should be done outside of SPM for flexibility and workflow reasons. For this lab we will simply continue to the model menu.

## 2. SPM TreeNet Model

Choosing 'TreeNet Gradient Boosting Machine' we can begin tuning the parameters to our TreeNet model. Similarly to lab #1 we want to select all the parameters as predictors, with the exception of the 'prediction' parameter which will be used as the target. From here we'll move to the main settings for our TreeNet model in the TreeNet pane.

*Date*: October 26, 2021.

FIGURE 1. SPM TreeNet Pane




In this pane we mainly want to decide on the TreeNet Loss Function for computing the residual, the criteria for determining the optimal number of trees(similar to pruning with CARTs) , the limits for the total number of trees to build, the maximum number of nodes per tree, and the maximal depth for each tree.

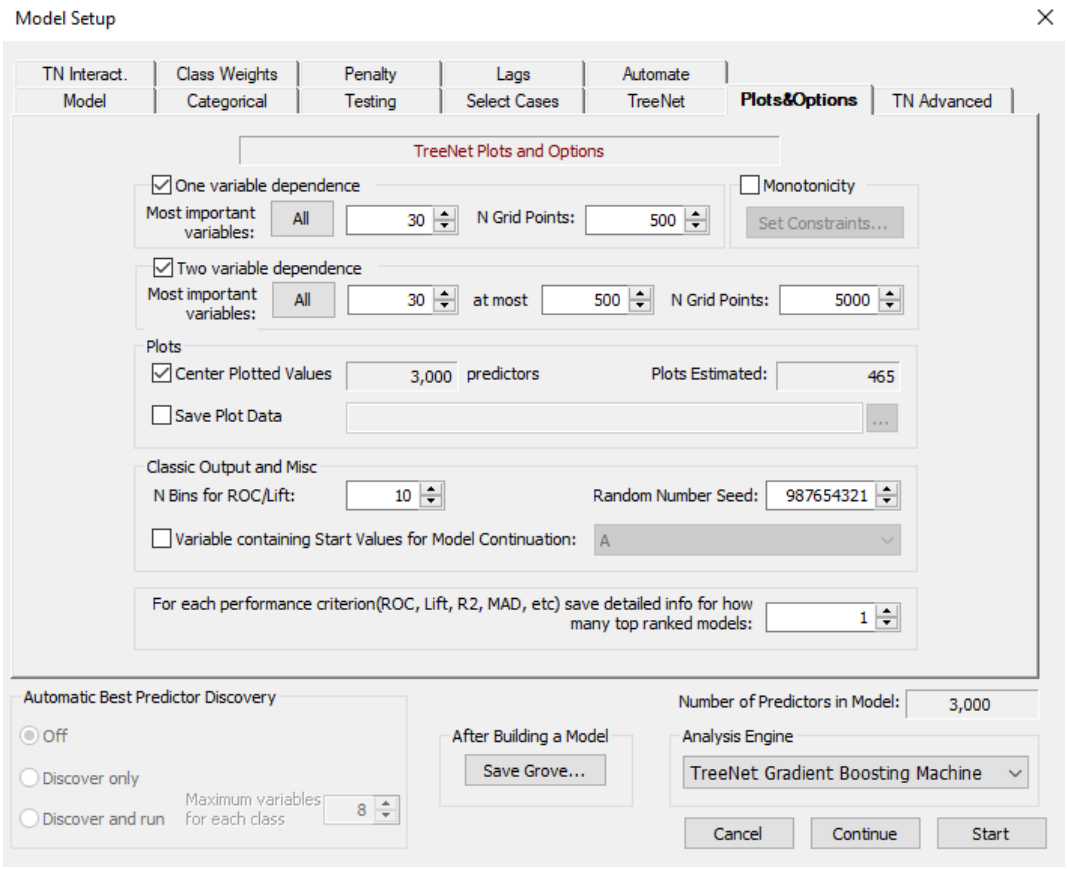
Other parameters like Learn Rate and Model Randomization can be explored. From the SPM documentation we know that adjusting the Learn Rate can increase the speed of convergence of the TreeNet model at the cost of overfitting. This can be useful for massive data and low computational power. Predictors per node and Predictors per tree allow us to select a subset of predictors to evaluate at each node and for each tree respectively, similarly to RandomForest. Varying tree size allows us to randomize the maximum nodes per tree at each iteration (documentation states that this is done by a poisson distribution with mean 'Maximum nodes per tree'). The 'Sample with replacement' options allow you to sample

residuals at each iteration with replacement. For our model we will leave these settings at default.

The Testing menu is again similar to the basic CARTs testing menu. For our model we will use the same V-fold cross validation testing scheme we used in lab #1.

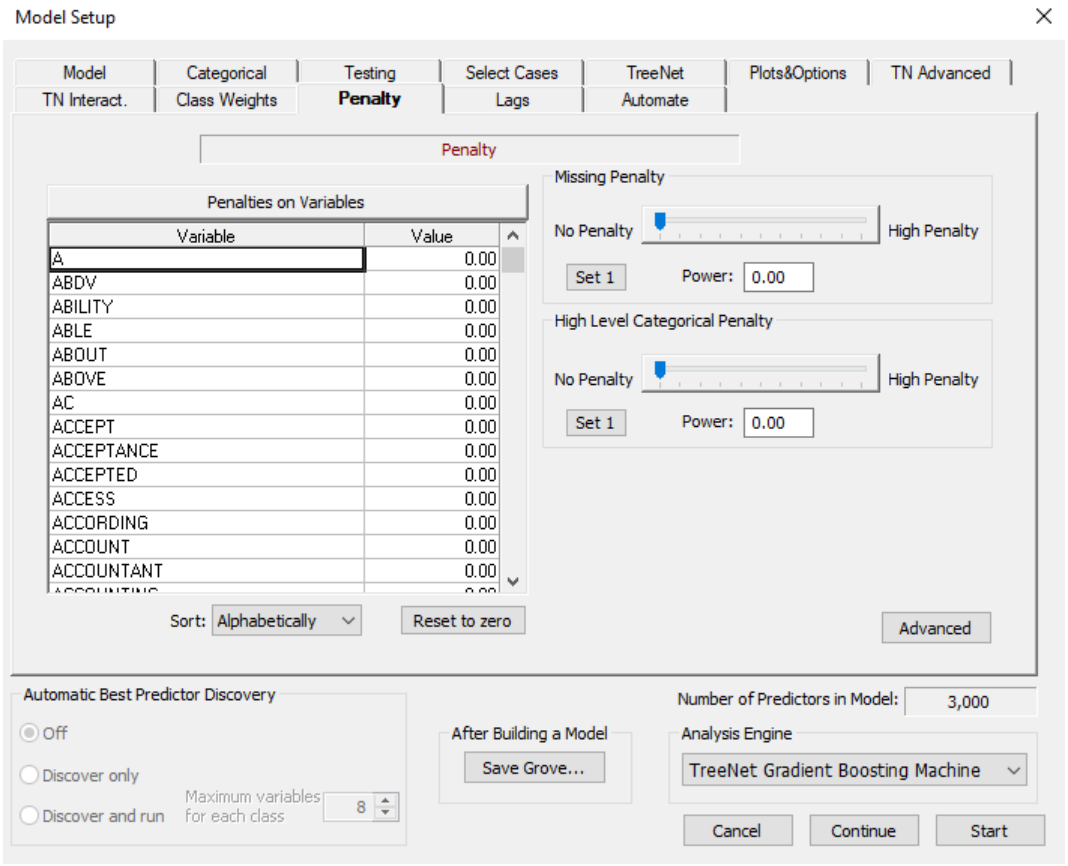
The Plots and Options menu is of particular importance to us, as we will be exploring methods for exporting visualizations/data outside of SPM later on. Here we can decide which Dependence plots SPM will generate. It seems that the default option is to generate the one and two-variable dependence plots for the 30 most important predictors. We will leave this on the default settings for our model.

FIGURE 2. SPM Plot & Options Pane



In the case that our data has missing values or high level categorical predictors we would use the penalty pane to account for that. Our data has no missing values or categorical predictors so we will leave these settings at default.

FIGURE 3. SPM Penalty Pane




The Categorical menu allows us to add labels to our categorical data values, for convenience we can use this to label our target data as spam, and non-spam.

FIGURE 4. SPM Categorical Pane

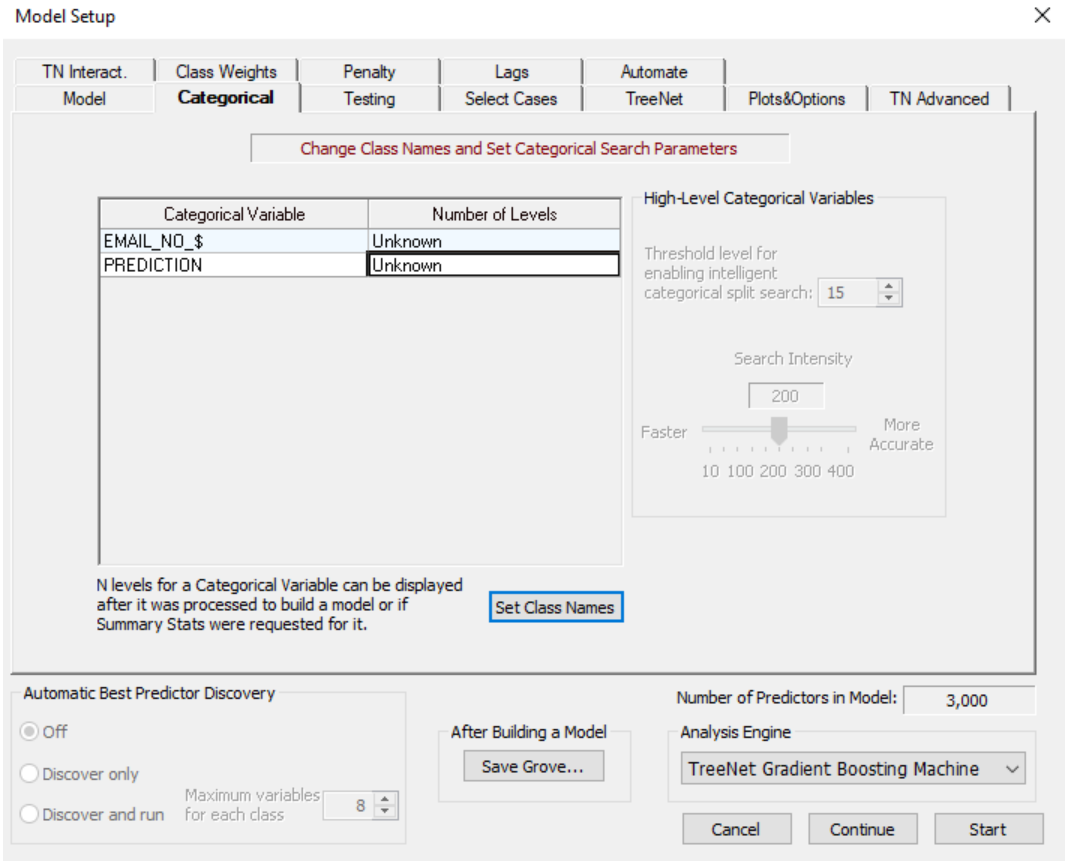
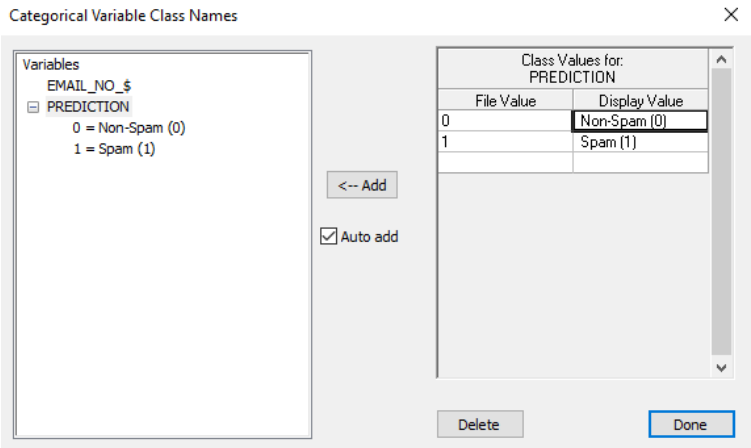


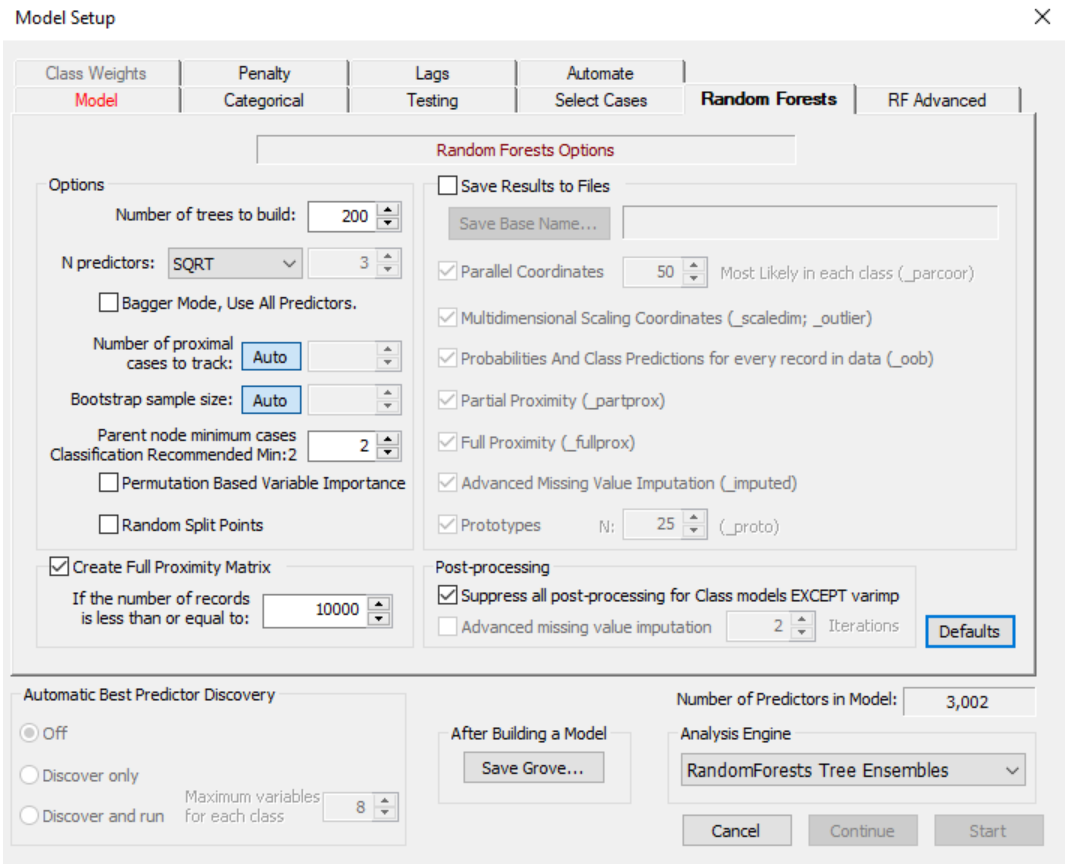

FIGURE 5. Renaming Data



## 3. SPM Random Forest Model

Selecting 'Random Forest Tree Ensembles' in the model pane we can being to tune the parameters for our RandomForest Model. Selecting our predictors and response variable is the same as it is for CARTs and TreeNet. We get to the main parameters of the Random Forest model under the 'Random Forest' tab.

Figure 6. SPM Random Forest Pane

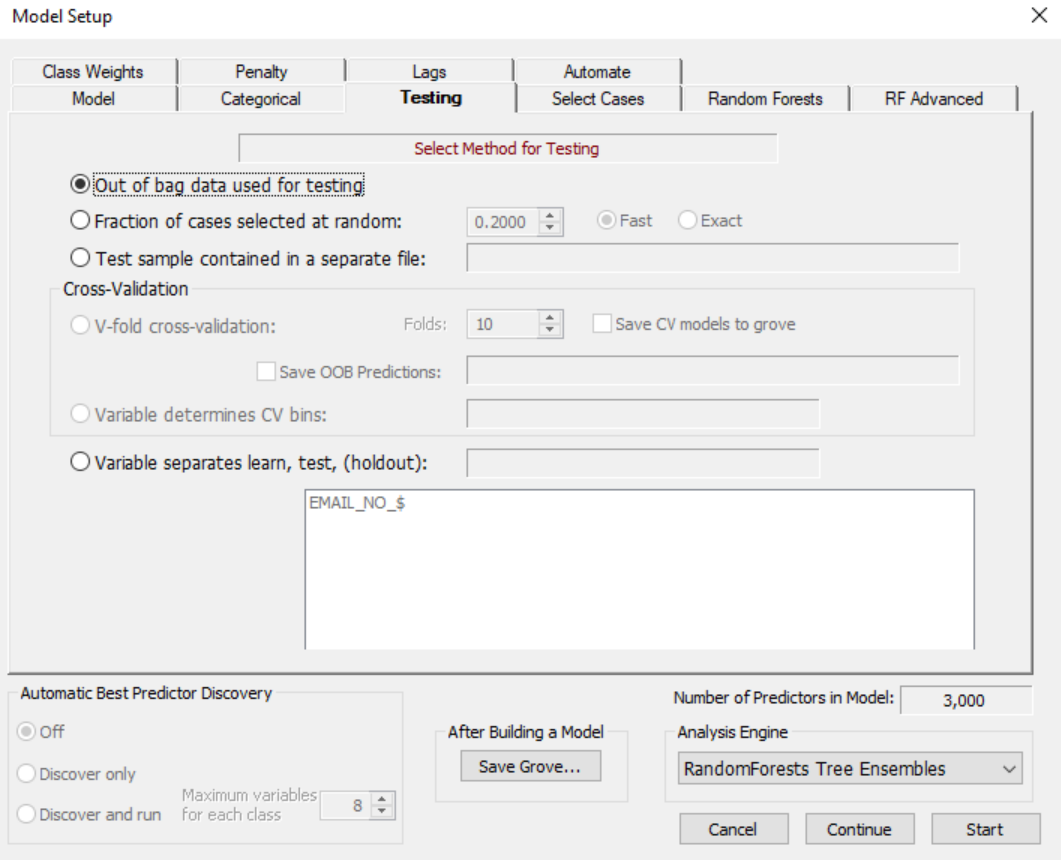



By default SPM will build 200 trees each with a sample the size of the square root of the total number of predictors. For our model we will use the default number of trees and predictors. The 'number of proximal cases to track' option will influence reports on clustering and outliers that the model produces. The 'Bootstrap sample size' option allows us to decide how big of a bootstrap sample of data we are taking for each forest. The parent node minimum option allows us to prune each tree in the model to a certain size, however it is recommended that each tree be grown to it's maximum size. The rest of the options

pertain to saving results in the form of proximity matrices, outlyiness measures and parallel coordinate plots. All of these will be available in the Random Forest plot toolbox but here we can save to a file.

In terms of testing, it is recommended to use out of bag testing for random forest models.

FIGURE 7. SPM Random Forest Testing Pane




## 4. PERFORMANCE ANALYSIS

Let's recall some of the performance benchmarks from the previous lab where we explored a basic Decision Tree model. The CART model achieved a ROC score of .957 on the test data and .987 on the learning data. From it's confusion matrix we saw that the CART model misclassified 87 spam emails and 218 non-spam emails in the testing data. Later we will discuss variable importance as an exercise in exporting data.

FIGURE 8. SPM CART Summary Pane

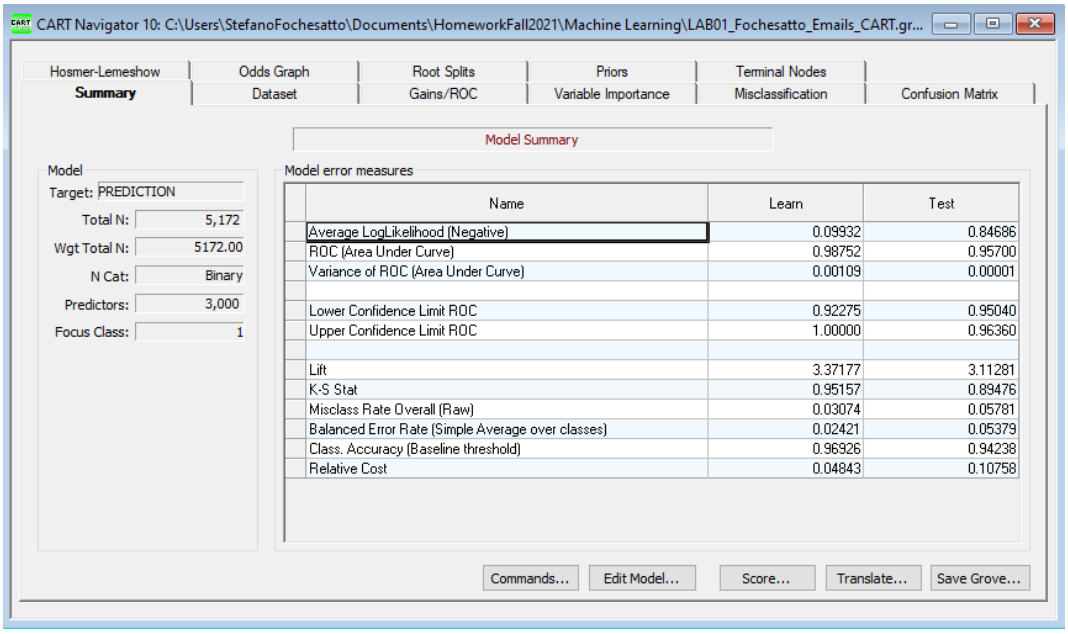
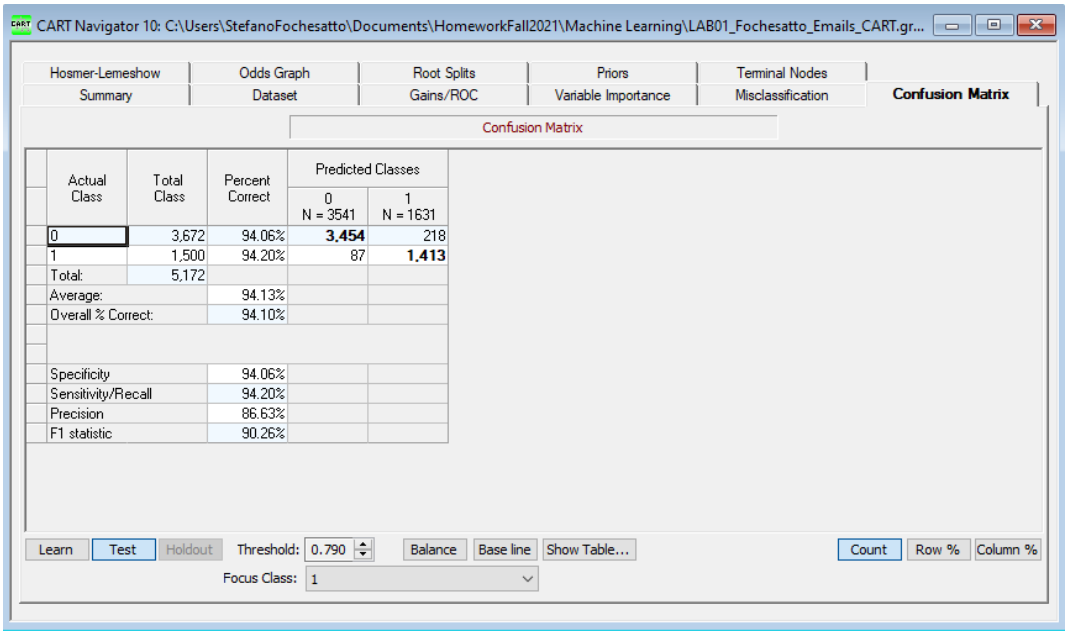



FIGURE 9. SPM CARt Confusion Matrix Pane




4.1. **TreeNet Performance.** Our default TreeNet Model performed better than the CART model. The TreeNet model achieved a ROC score of .99605 on the test data and .99890 on the learning data. We can also see from the confusion matrix that the TreeNet model

misclassified only 39 spam email and 100 non-spam email. Comparing the test data, the TreeNet model performed on average 3.27% better than the CART model. We can also see that the confidence interval for the ROC score is an order of magnitude smaller than the CART model.

FIGURE 10. SPM TreeNet Summary Pane

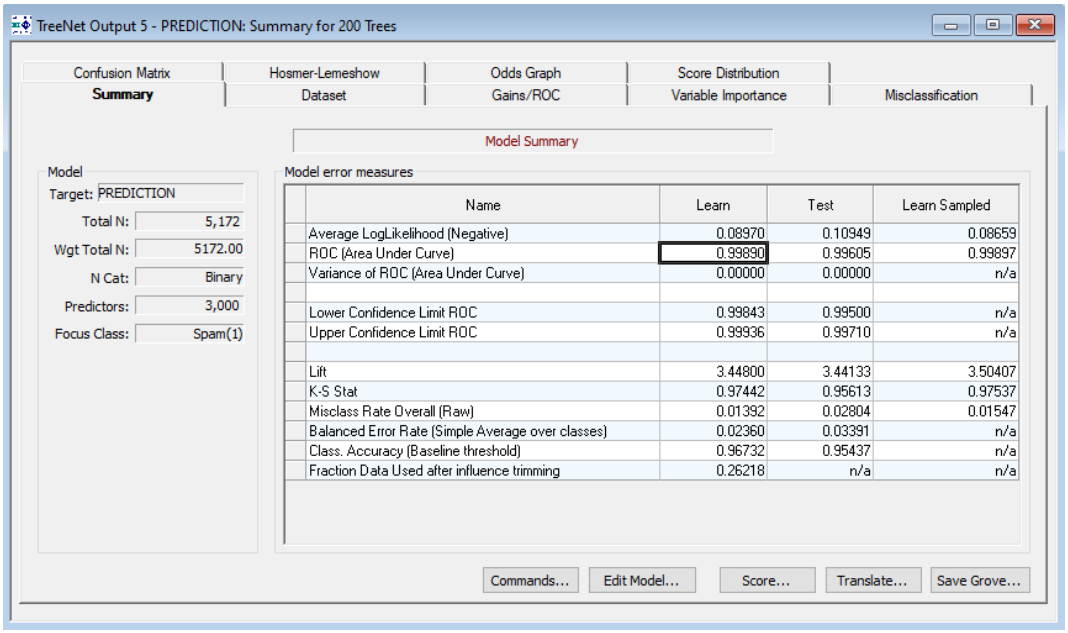
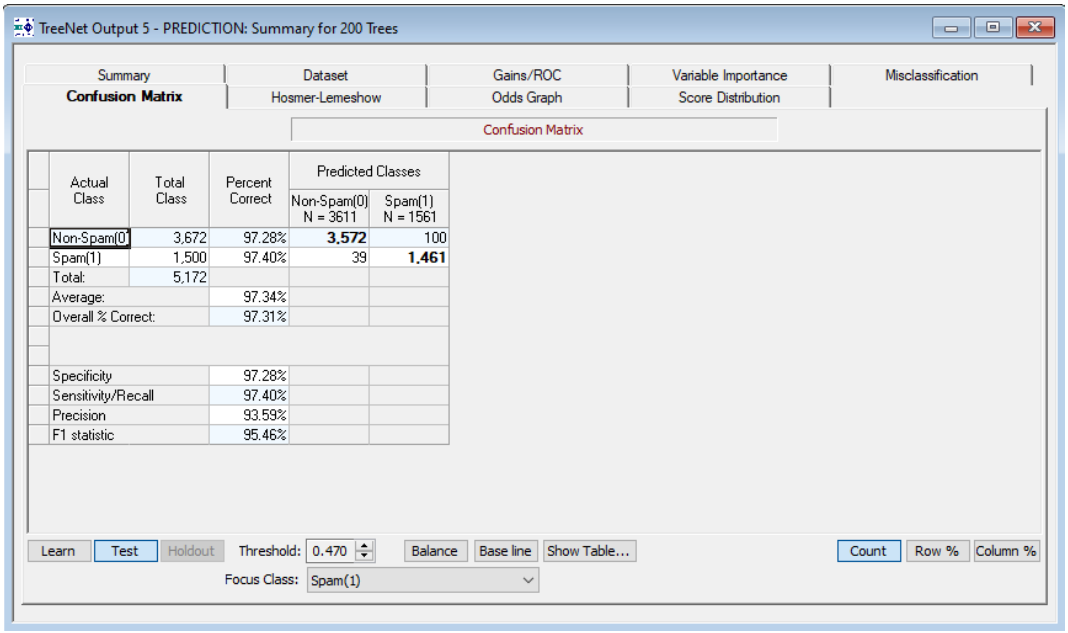
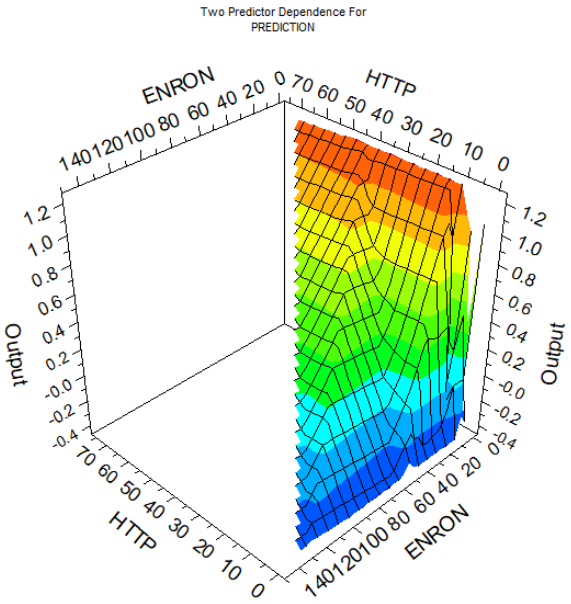



FIGURE 11. SPM TreeNet Confusion Matrix Pane



Interestingly the models produced differing variable importance plots, we will expand on this in the next section. For know let's look at the partial dependence plot for the two most important predictors in the TreeNet model, 'Enron' and 'HTTP'

FIGURE 12. SPM TreeNet Partial Dependence Plot




This plot shows us the inverse relationship between the two most important predictors. With increase mentions of 'Enron' the less likely the email will be spam, while with increasing mentions of 'HTTP' the more likely an email is going to be spam. This makes sense since most of the time 'HTTP' appears in a url, emails like product advertisements and even phishing attacks will contain multiple urls.

4.2. **RandomForest Performance.** When experimenting with the parameters of the RandomForest model, I found that using more than the default (200) number of trees resulted in marginal gains. In particular using 500 trees we saw a gain in only .0004 in the ROC curve evaluation. Doubling the number of predictors produces a similar marginal result, while significantly increasing the computation time. Our RandomForest model achieved a ROC score of .99591 on OOB samples. The model misclassified 40 spam email and 98 non-spam

email, with an average accuracy of 97.33. In terms of average accuracy our RandomForest model performed .01% worse than our TreeNet model.

FIGURE 13. SPM RandomForest Summary Pane

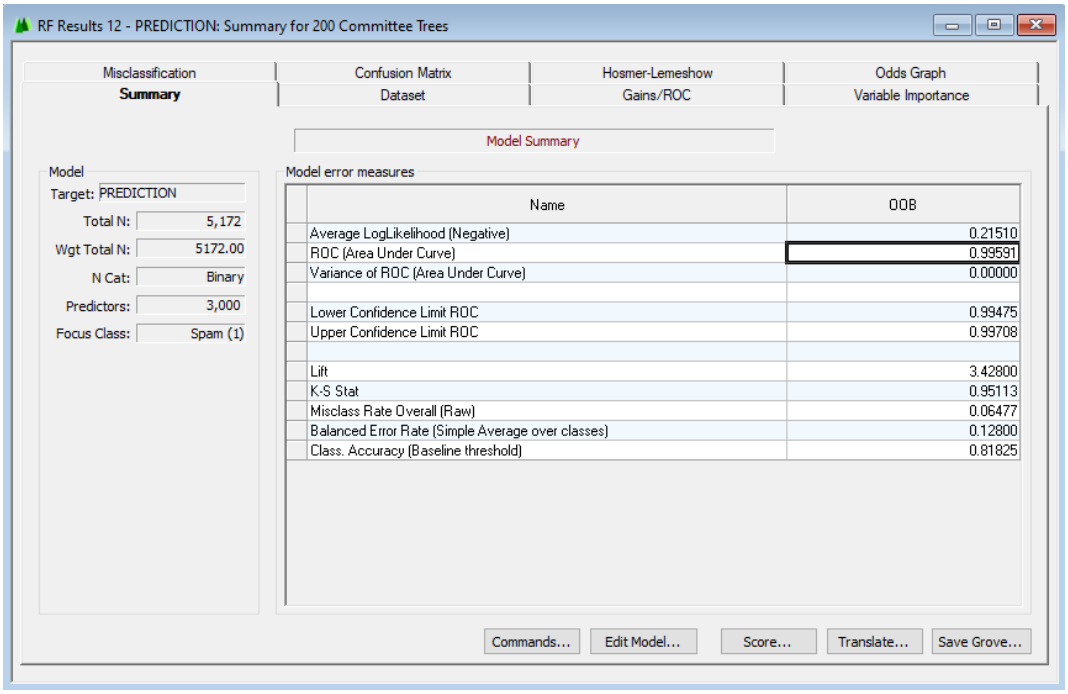
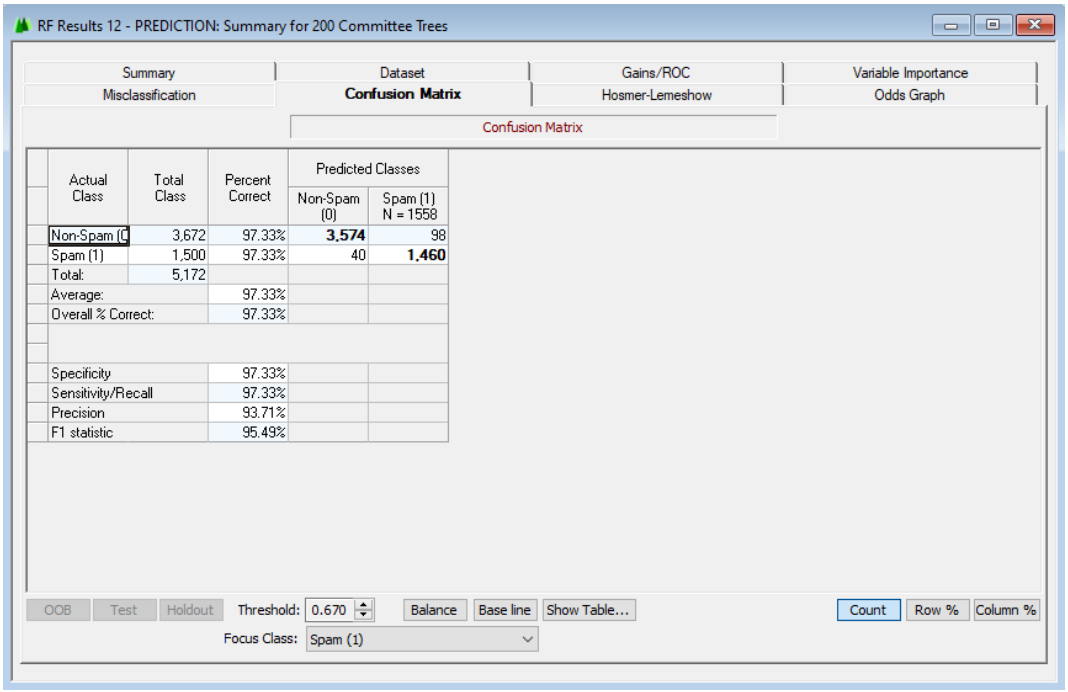
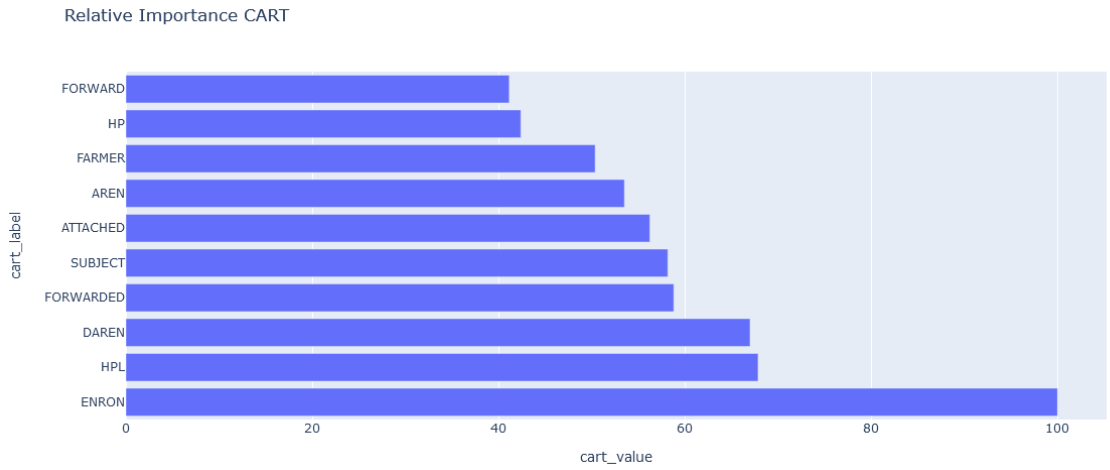
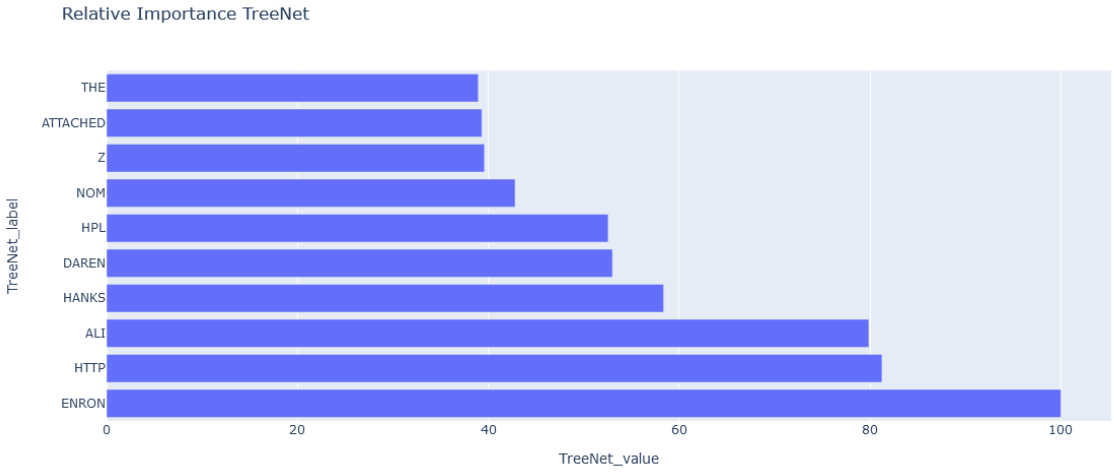


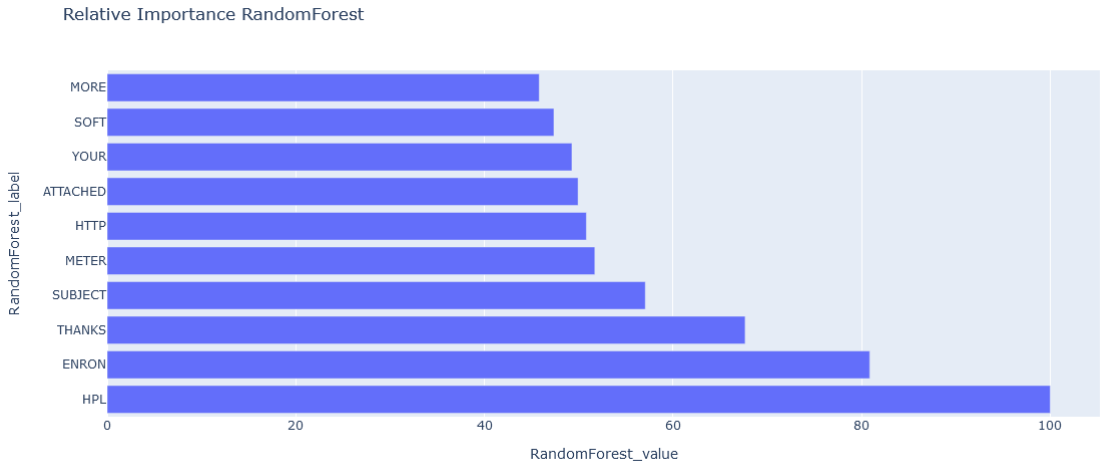

FIGURE 14. SPM TreeNet Confusion Matrix Pane



4.3. **Exporting Data.** I have found that all of the figures in the summary panes are reproducible by copying the data and importing it into a suitable plotting tool. A good example of this workflow is the variable importance plot. Below are are the relative variable importance plots for the three models. The plots were produced using the Plotly python library, the point we mean to illustrate here is that most data needed to document the performance of the models can be directly copied to other visualization tools.

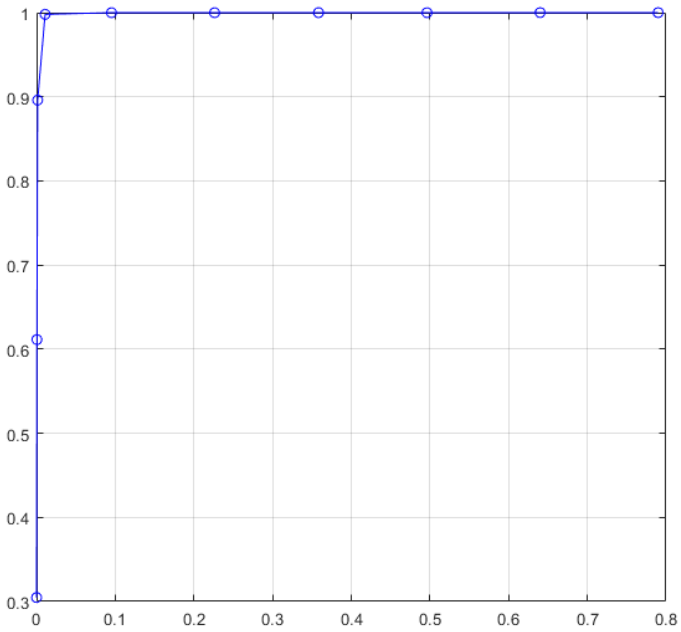




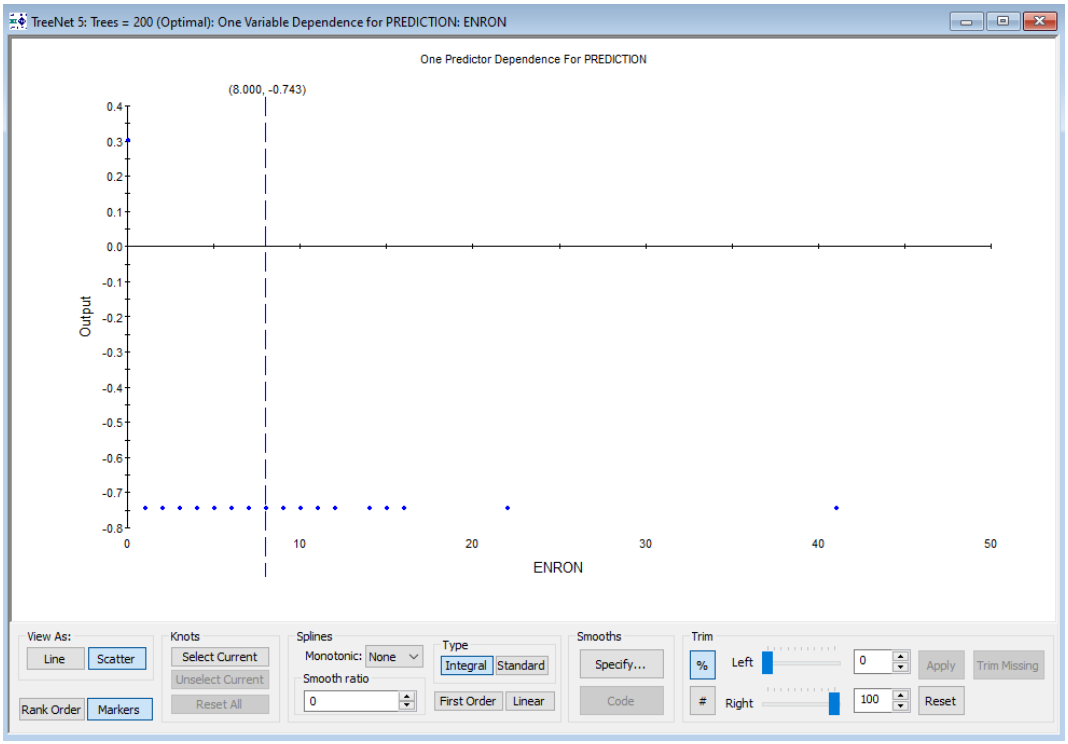

The following is another example were I was able to recreate the ROC curve for the RandomForest model in Matlab by simply exporting the Cum % Focus class data for each Spam and Non-Spam class,

FIGURE 15. ROC curve for RandomForest Model



Reproducing variable dependence plots seems possible, yet painstaking. A possible workflow involves viewing the individual dependence plots as a scatter plot, copying each point one by one, then reproducing them in another plotting tool.

FIGURE 16. Scatter View of One-Variable Dependence Plot




## 5. CONCLUSION

Of the models we explored TreeNet produced the best prediction results, followed closely by RandomForest and then finally CART. Again the SPM software allowed for quick testing when exploring the parameters for both models, and the documentation was a good resource for interpreting results and identifying best practices. Although we were able to export some data to visualize outside of SPM, exporting things like one and two-variable dependence plots isn't really an option. For further analysis all grove files and code can be found **here**.