Final Project

Jared A.

4/30/2021

## Introduction

In this project we will be using a dataset from GitHub containing data on spam emails to try and predict whether or not an email is spam. This data set contains 7 variables that look at certain aspects of an email and 6 of the 7 variables can be used to predict if an email is spam or not. Below is a listing and explanation of each of the variables in the data set.

**crl.tot**: total length of words in capitals

**dollar**: number of occurrences of the ‘$’ symbol

**bang**: number of occurrences of the ‘!’ symbol

**money**: number of occurrences of the word ‘money’

**n000**: number of occurrences of the string ‘000’

**make**: number of occurrences of the word ‘make’

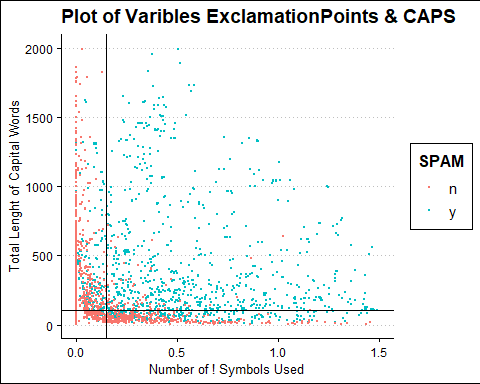
**yesno**: classification of the email that is either ‘y’ for spam or ‘n’ for not spam

Some of these variables are self explanatory by their name where others are not. Because of this we have renamed the variables in the dataset to make them easier to understand.

## Email\_No CAPS DollarSign ExclamationPoints Money TripleZeros Make SPAM  
## 1 1 278 0.000 0.778 0.00 0.00 0.00 y  
## 2 2 1028 0.180 0.372 0.43 0.43 0.21 y  
## 3 3 2259 0.184 0.276 0.06 1.16 0.06 y  
## 4 4 191 0.000 0.137 0.00 0.00 0.00 y  
## 5 5 191 0.000 0.135 0.00 0.00 0.00 y  
## 6 6 54 0.000 0.000 0.00 0.00 0.00 y

Above we are able to see the new variable names as well as examine the first 6 lines of data in the dataset. After looking at the table, we can take variables that we believe to be an important factor in predicting spam and visualize them to learn more. The two variables that we decided to look into further were CAPS and ExclamationPoints.

## Model 1 Visualization



Above we can see how the variables are distributed as well as whether they are classified as spam or not. We were able to make splits in the data that are represented by the lines seen in the plot. The splits appear to classify the data fairly accurate but we will now be creating a model based off of these splits to see how accurate they are.

## Model 1 Results

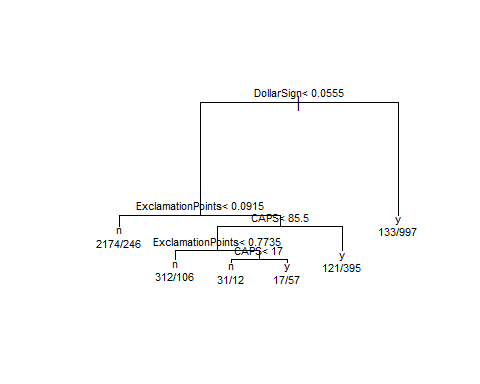
## Actual  
## Predict n y  
## n 2720 898  
## y 68 915

Here we are able to see how well our model predicted an email to be spam or not compared to the actual value. From this table we are able to see that our model correctly predicted an email not to be spam 2,720 times but also wrongly classified emails that were not spam 898 times. When it comes to predicting emails to be spam our model did this correctly 915 times and incorrectly 68 times. These numbers appear to be better than the classification of non-spam emails but we are going to take a look at the metrics to get a better understating of these results.

## Metric Value  
## 1 Accuracy 0.790  
## 2 Sensitivity 0.505  
## 3 Specificity 0.976

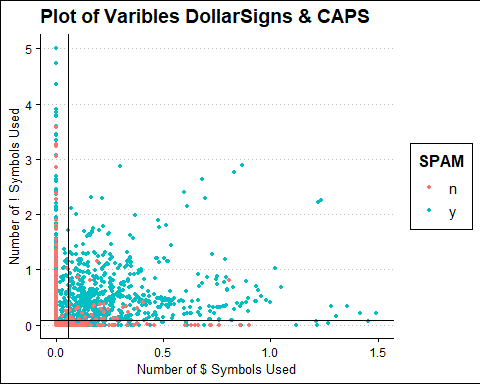
Above we are able to see our overall accuracy of the model as well as specific accuracies for each of the two classifications. We are able to see that our overall accuracy with this model is 79% providing valuable insights. The next metric of sensitivity is much lower at 50% and this was seen in the previous table showing that we incorrectly predicted an email that was not spam as spam numerous times. However, the specificity accuracy is much higher at 97% showing that our model predicts spam emails correctly very well.

## Model 2 Visualization 1

In our next model we will be attempting to improve the accuracy score using the rpart function to assist in splitting the data. This function looks at each of the variables and builds a decision tree based off of that data. The decision tree made for this dataset can be seen below.

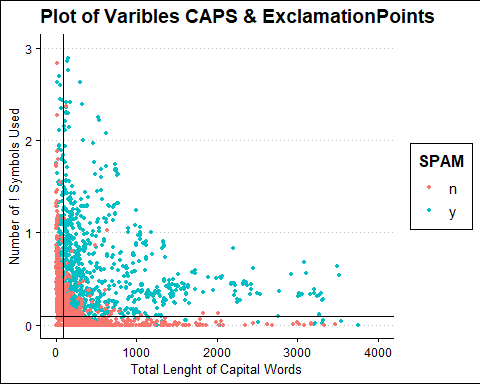
In the decision tree we are able to see the variables that were used to make splits as well as the specific point value used in that split. From this we will be creating a 3 variable model and will be visualizing the 3 variables before hand to better understand where the splits are made.

## Model 2 Visualization 2



The chart above shows the emails plotted by the number of ‘!’ and ‘$’ symbols which are the first two variables used in the generated decision tree. The lines on the chart represent the splits that the rpart function generated for each of the variables. We can see that these splits appear to do a pretty good job with the data but it is clear that more variables are needed to get better results.

## Model 2 Visualization 3



This chart visualizes the next split in the decision tree which branches off of the ExclamationPoints variable and adds the CAPS variable. We are able to see how these are plotted above as well as the splits determined by the function. Now that these splits have been visualized we need to create a model to see how well a spam email can be predicted.

## Model 2 Results

## Actual  
## Predict n y  
## n 2534 421  
## y 254 1392

Looking at the table above on model 2, we can see that at first glance there are some significant differences compared to the first model. We can see that this model greatly reduces the number of times that a non-spam email was classified as spam. However, it does appear that this model may perform worse in correctly predicting an email to be spam. To get a better understanding of this and to compare directly to the first model, we are going to take a look at the metrics.

## Metric Value  
## 1 Accuracy 0.853  
## 2 Sensitivity 0.768  
## 3 Specificity 0.909

In the metrics above we can see that we have a higher overall accuracy percentage. This higher accuracy came from a much higher sensitivity but did result in a lower specificity compared to the first model.

## Conclusion

In this project we were able to take a look at two different models that helped in predicting whether an email was spam or not. Our first model was able to help in the classification but lacked a high overall accuracy score, making it the lesser of the two models. Our second model was able to provide a much higher overall accuracy and this would be the recommended model for this data set. This model does sacrifice some accuracy in classifying spam emails, but due to its ability to make less errors in classifying than the first model, it is still the preferred model. There are great applications for this model as it could be added to an email system to automatically flag anything that satisfy the conditions of spam seen in this report. Using this model would improve the lives of email users and reduce the success of spam that is in our world today.