# MSDS 7333 Fall 2020: Case Study 3

Jayson Barker, Brandon Croom, Shane Winestock

## Business Understanding

Spam email is unsolicited and unwanted junk email sent out in bulk to an indiscriminate recipient list. Spam is typically send out for commercial purposes, but may also be leveraged for nefarious purposes. These emails are typically send out in bulk by botnets, networks of infected computers. According to multiple internet sites spam email accounts for 14.5 billion email message globally per day. This is approximately 45% of all emails. This magnitude of spam emails mixes in with regular emails that users need to address for business or personal purposes, thus making email more difficult to use.

To overcome the magnitude of spam emails and the interference with regular emails, being able to classify an email as spam or not-spam (aka ham) becomes a necessity. This problem has been solved in multiple ways by various technology vendors such as Microsoft and Google. For the purposes of this report, the project team will setup to classify emails as spam or not-spam through tree based approaches.

A tree provides a visual representation of a course of action or statistical probability. When visualized, the tree forms the outline of its physical namesake. Trees consist of branches and leaves (or decision nodes and terminal nodes). Each branch contains a decision point which represents a test on a feature. Each leaf node represents the decision made after computing all features. Leaves have no further decisions and are the furthest points on the tree. The path through the tree from the root node (or first node) to an individual leaf node represents a classification path. Building out this classification path will be what the project team will use to determine if an email is spam or not-spam.

## Data Acquisition/Cleaning

In order to determine if an email is spam or not-spam a corpus of emails needs to be collected and attributes on that corpus defined. For this report, the corpus of emails has been provided. At a high level the corpus contains 9,348 records with 30 different features. The data summary table below provides a good overview of the data set. The following summary of insights is apparent in the table:

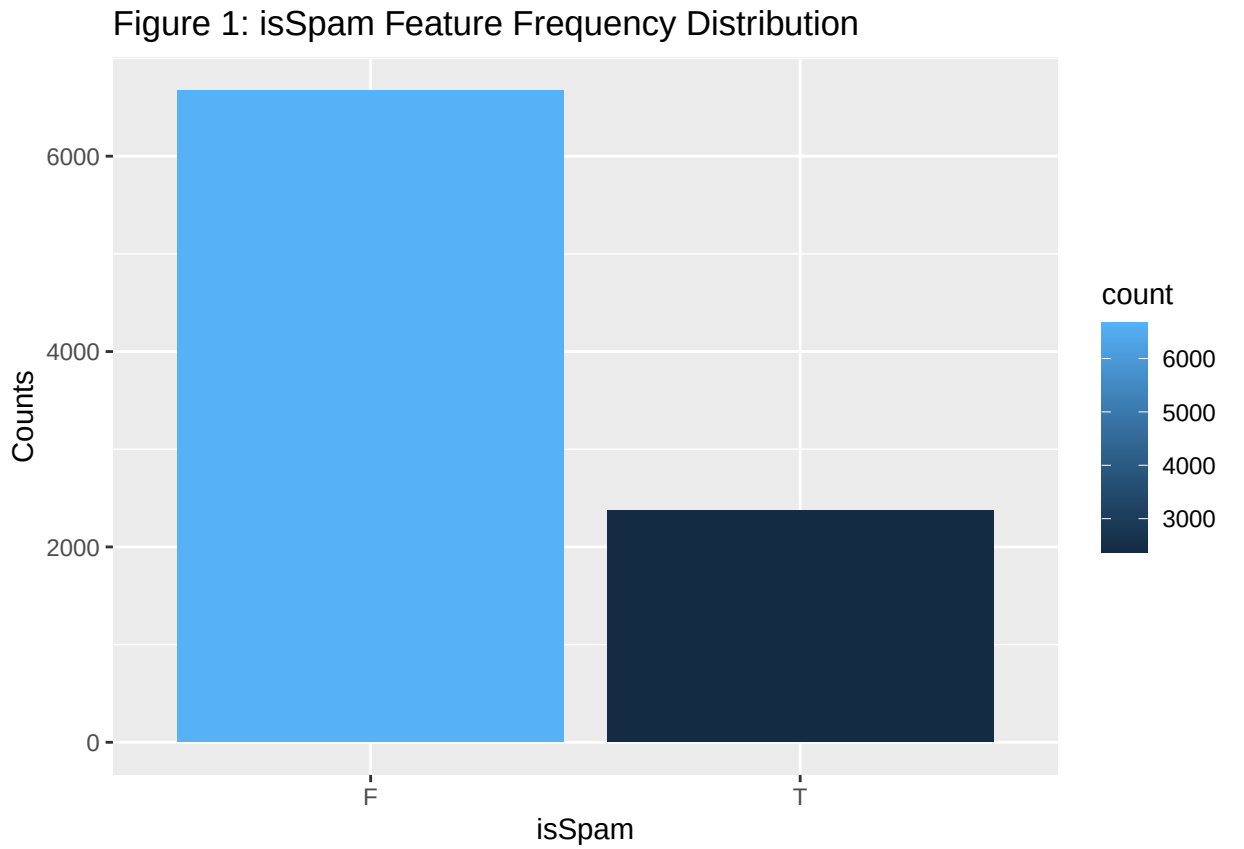
- There are 17 categorical variables in the data set
- There are 13 continuous variables in the data set
- At least four features (subExcCt, subQuestCt, numRec, and subBlanks) have missing values that need to be addressed

With respect to the missing values, for the purposes of this analysis the rows with missing values will be dropped. This will bring our working record count to 9045.

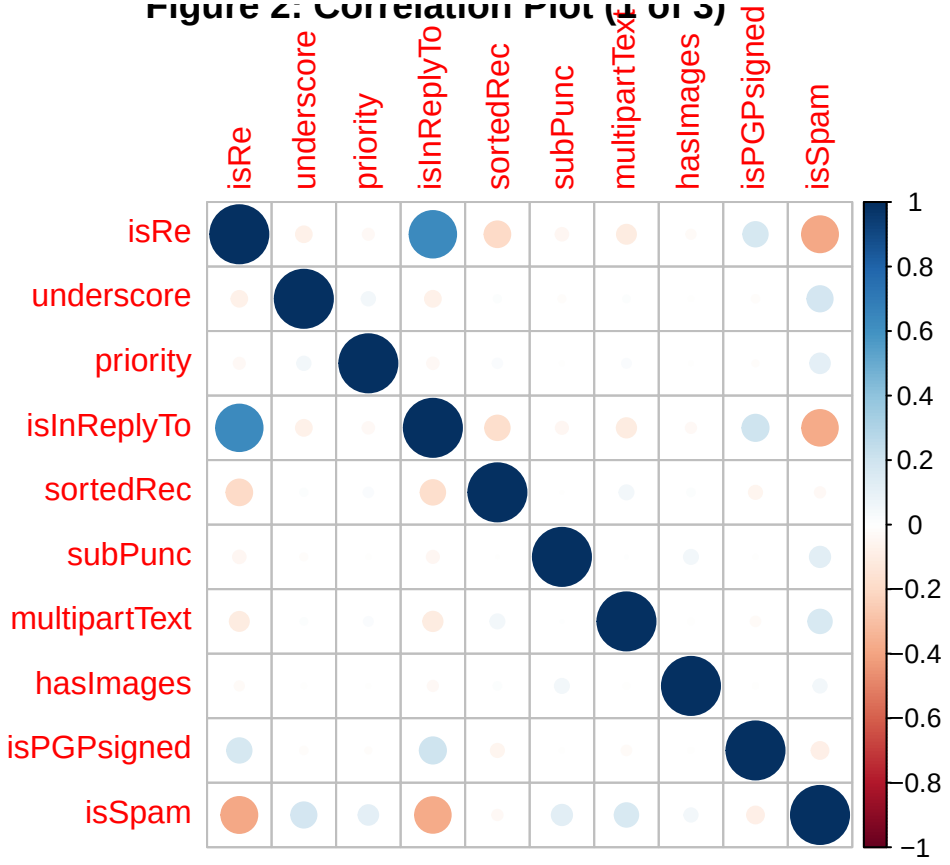
| skim_variable | n_missing | complete_rate | factor.ordered | factor.n_unique | factor.top_counts | numeric.mean | numeric.sd | numeric.p0 | numeric.p25 | numeric.p50 | numeric.p75 | numeric.p100 | numeric.hist |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| factor.isSpam | 0 | 1.00 | FALSE | 2 | F: 6951, T: 2397 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.isRe | 0 | 1.00 | FALSE | 2 | F: 6343, T: 3005 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.underscore | 0 | 1.00 | FALSE | 2 | F: 9222, T: 126 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.priority | 0 | 1.00 | FALSE | 2 | F: 9294, T: 54 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.isInReplyTo | 0 | 1.00 | FALSE | 2 | F: 6556, T: 2792 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.sortedRec | 0 | 1.00 | FALSE | 2 | T: 8400, F: 948 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.subPunc | 0 | 1.00 | FALSE | 2 | F: 9085, T: 263 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.multipartText | 0 | 1.00 | FALSE | 2 | F: 9020, T: 328 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.hasImage | 0 | 1.00 | FALSE | 2 | F: 9326, T: 22 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.isPGPsigned | 0 | 1.00 | FALSE | 2 | F: 9172, T: 176 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.subSpamWords | 7 | 0.99 | FALSE | 2 | F: 8697, T: 644 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.noHost | 1 | 0.99 | FALSE | 2 | F: 9318, T: 29 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.numEnd | 0 | 1.00 | FALSE | 2 | F: 8209, T: 1139 | NA | NA | NA | NA | NA | NA | NA | NA |

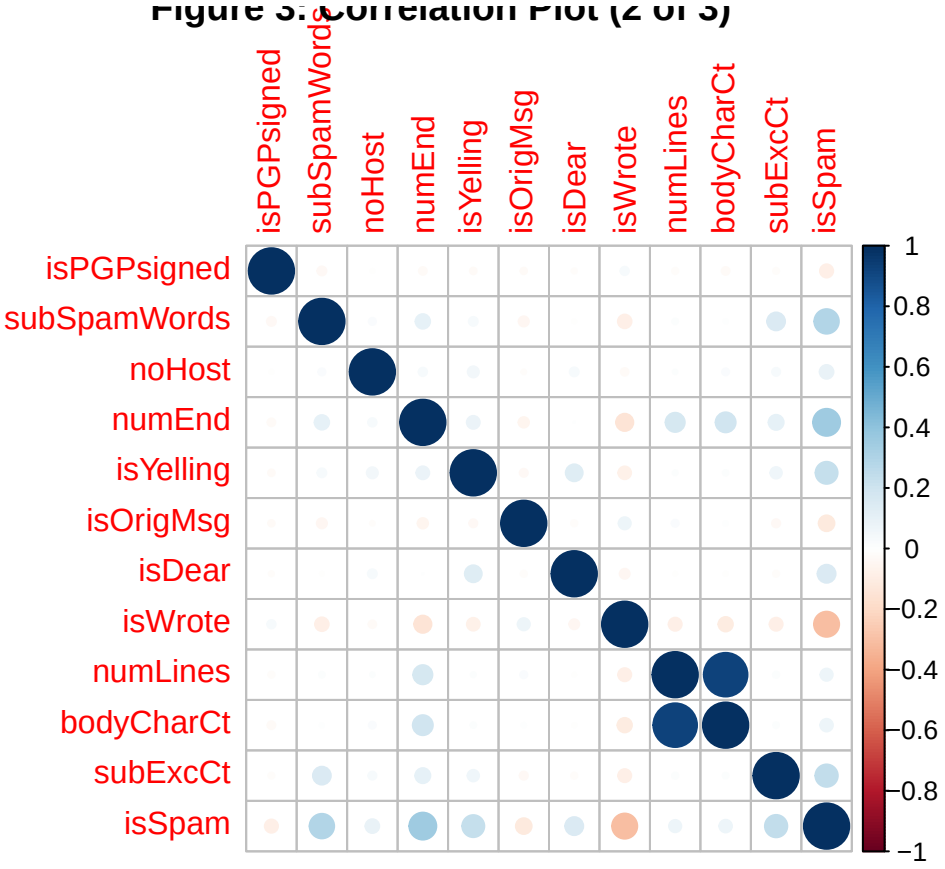
| skim_variable | n_missing | complete_rate | factor.ordered | factor.n_unique | factor.top_counts | numeric.mean | numeric.sd | numeric.p0 | numeric.p25 | numeric.p50 | numeric.p75 | numeric.p100 | numeric.hist |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| factor.isYelling | 7 | 0.999258 | FALSE | 2 | F: 9134, T: 207 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.isOrigMsg | 0 | 1.000000 | FALSE | 2 | F: 8988, T: 360 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.isDear | 0 | 1.000000 | FALSE | 2 | F: 9270, T: 78 | NA | NA | NA | NA | NA | NA | NA | NA |
| factor.isWrote | 0 | 1.000000 | FALSE | 2 | F: 7442, T: 1906 | NA | NA | NA | NA | NA | NA | NA | NA |
| numeric.maxLines | 0 | 1.000000 | NA | NA | NA | 66.90853 | 367.9553 | 2 | 8 | 19.00000 | 62.00000 | 6609.00000 | <U+2587><U+2581><U+2 |
| numeric.bodyCharCt | 0 | 1.000000 | NA | NA | NA | 2844.09 | 17633.36 | 6 | 587.00 | 1088.00 | 2102.00 | 885505.00000 | <U+2587><U+2581><U+2 |
| numeric.subExclCt | 20 | 0.997786 | NA | NA | NA | 0.13132 | 2.566156 | 0 | 0.00000 | 0.00000 | 0.00000 | 142.00000 | <U+2587><U+2581><U+2 |
| numeric.subQuestCt | 20 | 0.997786 | NA | NA | NA | 0.13775 | 1.307685 | 0 | 0.00000 | 0.00000 | 0.00000 | 12.00000 | <U+2587><U+2581><U+2 |
| numeric.maxAtt | 0 | 1.000000 | NA | NA | NA | 0.06578 | 0.324878 | 0 | 0.00000 | 0.00000 | 0.00000 | 18.00000 | <U+2587><U+2581><U+2 |
| numeric.maxRec | 282 | 0.969831 | NA | NA | NA | 1.92940 | 6.642396 | 0 | 1.00000 | 1.00000 | 1.00000 | 301.00000 | <U+2587><U+2581><U+2 |
| numeric.perCaps | 0 | 1.000000 | NA | NA | NA | 8.85037 | 9.683415 | 0 | 4.02553 | 6.19554 | 7.85939 | 99.00000 | <U+2587><U+2581><U+2 |
| numeric.hour | 0 | 1.000000 | NA | NA | NA | 12.21086 | 4.723932 | 0 | 8.00000 | 12.00000 | 16.00000 | 23.00000 | <U+2585><U+2587><U+2 |
| numeric.perHTML | 0 | 1.000000 | NA | NA | NA | 6.51708 | 21.13526 | 0 | 0.00000 | 0.00000 | 0.00000 | 100.00000 | <U+2587><U+2581><U+2 |
| numeric.subBlanks | 20 | 0.997786 | NA | NA | NA | 13.86697 | 342.1937 | 0 | 0.52631 | 3.62531 | 12.88687 | 27419.75000 | <U+2587><U+2581><U+2 |
| numeric.forwards | 0 | 1.000000 | NA | NA | NA | 10.44508 | 85.26357 | 0 | 0.00000 | 0.00000 | 0.00000 | 5384.00000 | <U+2587><U+2581><U+2 |
| numeric.avgWordLen | 0 | 1.000000 | NA | NA | NA | 4.48722 | 1.868582 | 1 | 3.63047 | 4.20825 | 4.75454 | 67.28571 | <U+2587><U+2581><U+2 |
| numeric.numDlr | 0 | 1.000000 | NA | NA | NA | 1.78155 | 36.38045 | 0 | 0.00000 | 0.00000 | 0.00000 | 1977.00000 | <U+2587><U+2581><U+2 |

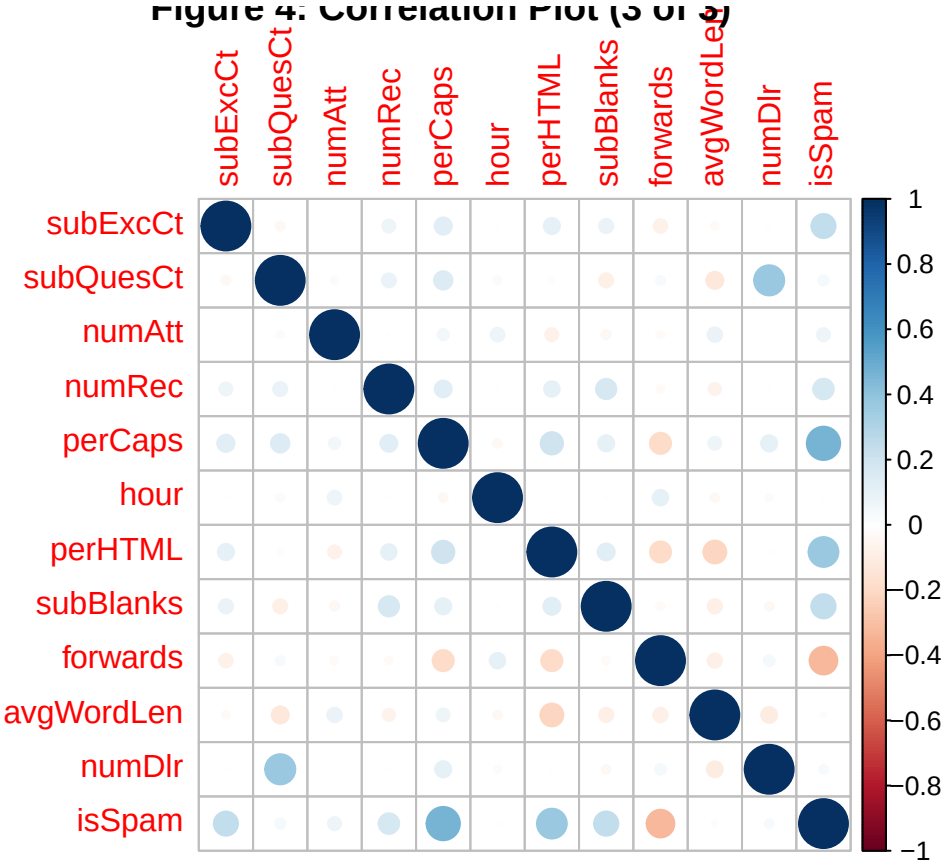
Continuing in the exploratory data analysis, the data set has a feature called isSpam. This feature is the original classification value for each email to determine whether it is spam or not. For the purposes of building out a classification model an evaluation needs to occur on this feature to see whether the data is balanced or unbalanced. Balanced data indicates that the feature values have approximately a 50/50 split in them. Unbalanced data indicates that the feature values skew in one direction or another. As Figure 1 below displays, the isSpam field is unbalanced in that there are more records in the data set classified with isSpam=F which indicates a not-spam email versus a spam email where isSpam=T. From a numeric perspective, 6674 of the 9045 records are classified as not spam, while 2371 are classified as spam.




Further exploring the data, the investigation of correlations between features needs to occur. Correlations are relationships between two or more features. Understanding which features in the dataset are correlated allow for data set simplification. If fields are highly correlated then similar enough information may be present that both features do not need to be carried forward into analysis. The correlation plots in Figures 2 - 4 show the correlations between all 30 features in the data set.


isRe
underscore
priority
isInReplyTo
sortedRec
subPunc
multipartText
hasImages
isPGPsigned
isSpam
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1


isPGPsigned
subSpamWords
noHost
numEnd
isYelling
isOrigMsg
isDear
isWrote
numLines
bodyCharCt
subExcCt
isSpam
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1




From the correlation plot thre are some features that show high correlation. In summary, highly correlated features are

- isInReplyTo and isRE
- numLines and bodyCharCt
- numDlr and subQuestCt

It should also be noted at this point that in order to achieve accurate correlations, the categorical factors were one hot encoded. One hot encoding simply changes categorical variables to numeric values for easier analysis. Specifically relating to this data set the categorical variables are primarily the values T and F representing True and False. The T values were converted to the numeric value 1 and F values were converted to the numeric value 0. This is a typical approach for handling True/False values.

## Data Analysis

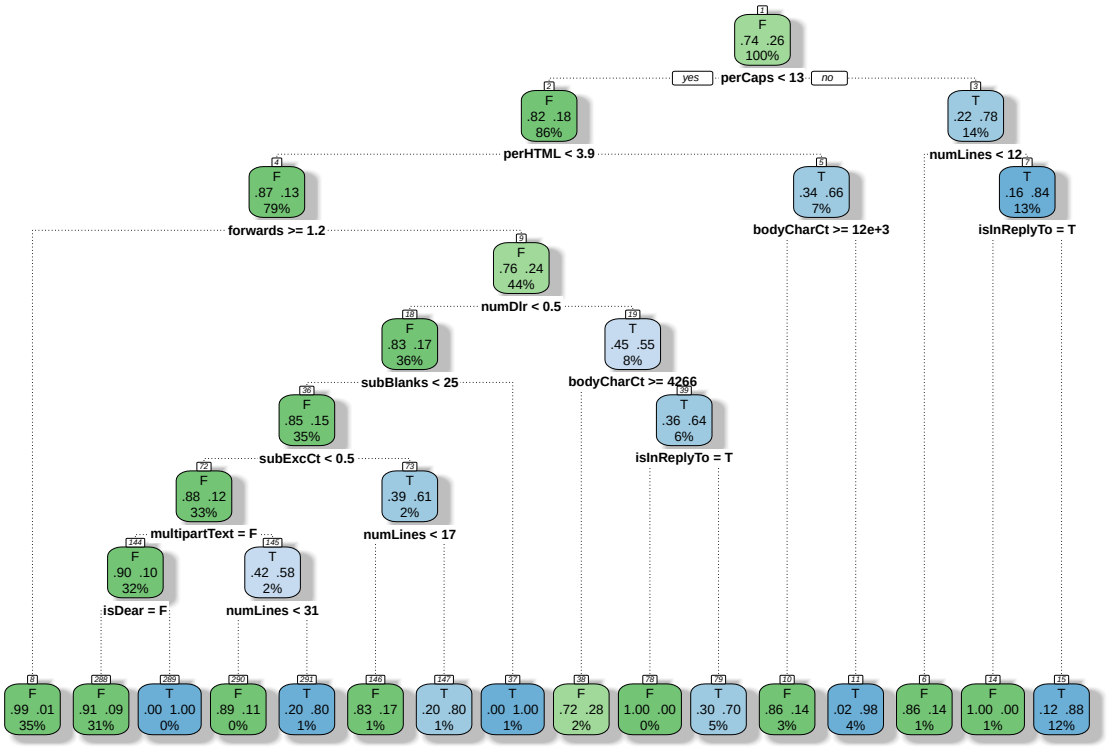
Moving into the data analysis phase where classification of an email as spam or not-spam can begin to occur, the first step that needs to occur is building out training and test datasets. Breaking the data out into a training and test dataset allows for model building, using the training data set, and then model evaluation can occur on the test dataset. Building out two data sets in this way assists in managing model overfitting.

```
## [1] 7237   30
```

```
## [1] 1808   30
```

## Basic Tree Model

Starting off the classification of email message the first step will be to begin with a basic tree classification model. This will allow for a quick understanding of classification and provide a baseline for further comparison.




Initial Tree

As shown above the tree is relatively large but does provide an idea about how an email message may be classified as spam or not-spam. Using this base tree the overall accuracy of the model can be evaluated. Again this will help determine a baseline for analysis as future trees are developed.

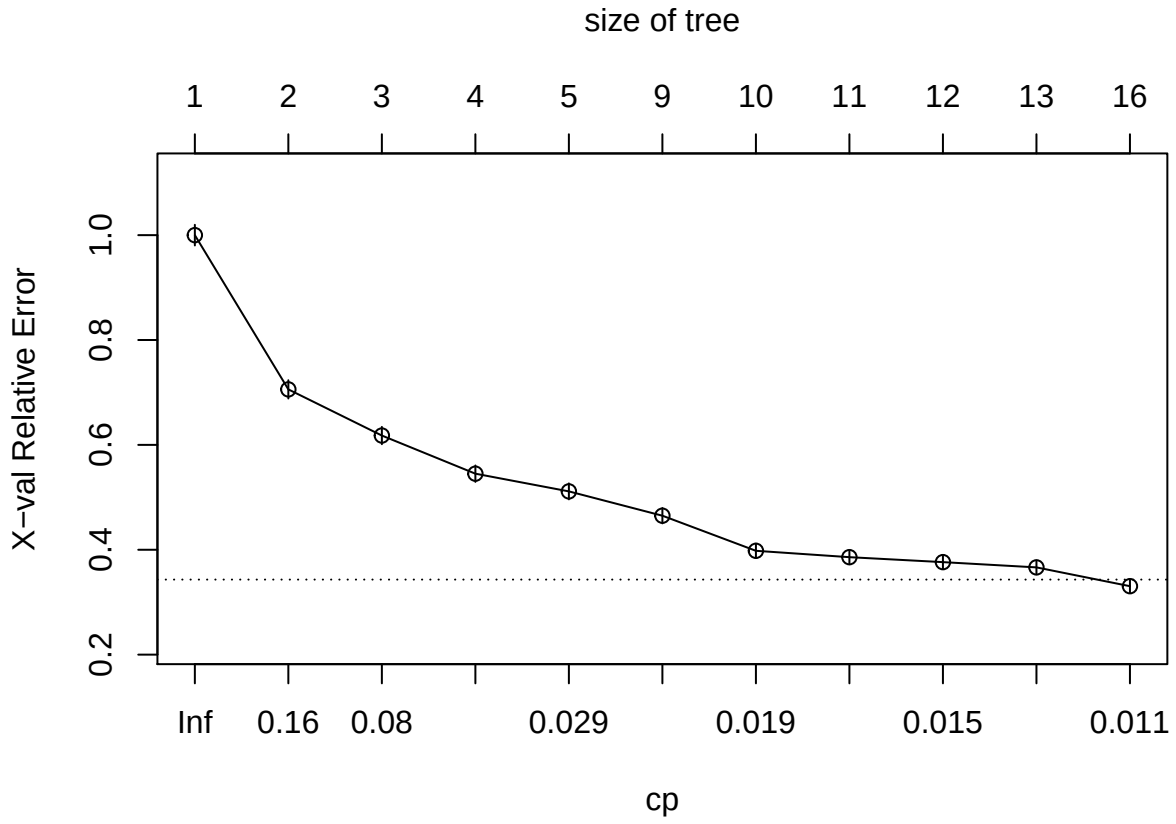
In evaluating this tree, the first node starts with the perCaps feature. If perCaps is less than 1.3 the values move to the left of the tree. If they are greater they move to the right of the tree. Following the right side of the tree, the next test is the numLines feature. If the numLines is less than 1.2 the data moves to the left node and those values greater move to the right. Continuing right from numLines, the isInReplyTo feature is tested. If this feature is true the email is classified as spam. If it is false then the email is classified as not-spam. Following a simlar process as outlined above can be done for all nodes in the tree and will provide an indication on how the new emails may be classified.

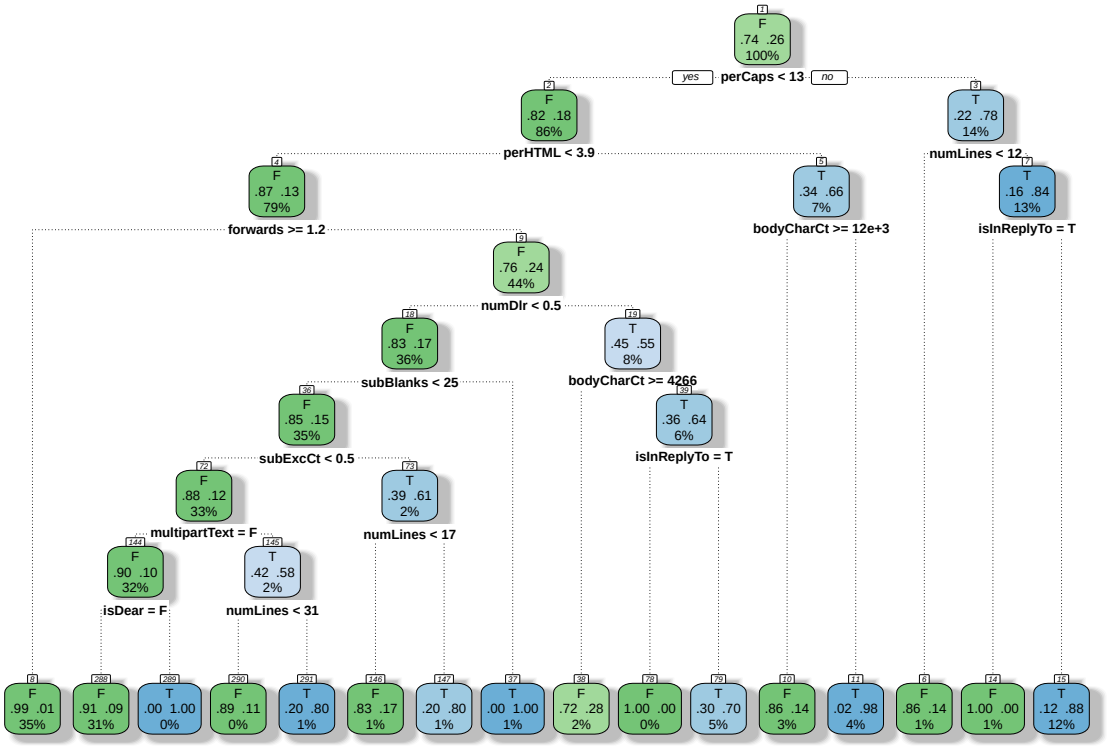
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    F    T
##          F 1271   98
##          T   63  376
##
##                Accuracy : 0.911
##                  95% CI : (0.8969, 0.9237)
##     No Information Rate : 0.7378
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7642
##
##  Mcnemar's Test P-Value : 0.007372
##
##             Sensitivity : 0.9528
##             Specificity : 0.7932
##          Pos Pred Value : 0.9284
##          Neg Pred Value : 0.8565
##              Prevalence : 0.7378
##          Detection Rate : 0.7030
##    Detection Prevalence : 0.7572
##       Balanced Accuracy : 0.8730
##
##        'Positive' Class : F
##
```

From the confusion matrix above the overall accuracy of the first tree configuration is approximately 91%. Of the 1809 records in the test data set, 1271 were accurately classified as not-spam and 376 were accurately classified as spam.

To help with any possible over fitting the tree can be pruned. Pruning trees attempts to get to a more optimal tree by removing nodes in the tree and stopping at what the modeler deems is an acceptable stopping criteria. To determine what that stopping criteria could be, investigation of the error in the original tree is helpful. The plot below shows the cross validation error for each split that can be used to prune the tree. From the plot the cp equal to approximately 0.014 would be a good spot to prune the tree.






Pruned Tree

The pruned tree looks very similar to the initial tree. This is also confirmed by the confusion matrix results shown below. No change is seen between the initial tree and the pruned tree. The overall accuracy of the pruned tree configuration is approximately 91%. Of the 1809 records in the test data set, 1271 were accurately classified as not-spam and 376 were accurately classified as spam.

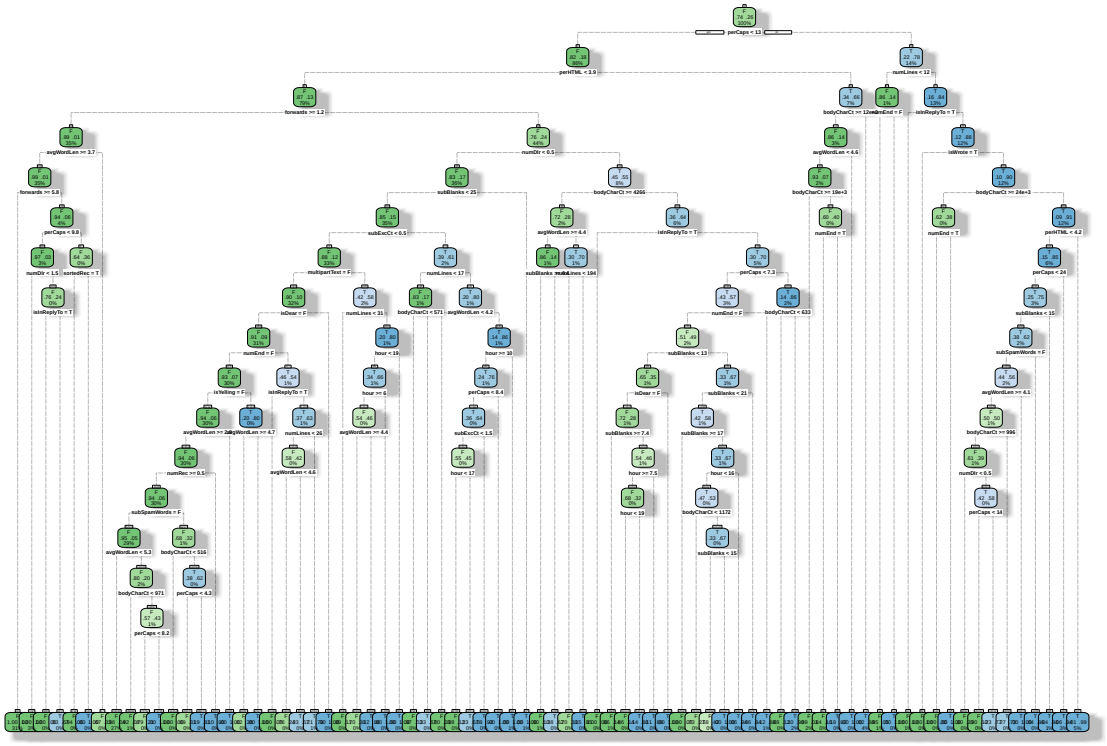
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    F    T
##          F 1271   98
##          T   63  376
##
##                Accuracy : 0.911
##                  95% CI : (0.8969, 0.9237)
##     No Information Rate : 0.7378
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7642
##
##  Mcnemar's Test P-Value : 0.007372
##
##             Sensitivity : 0.9528
##             Specificity : 0.7932
##          Pos Pred Value : 0.9284
##          Neg Pred Value : 0.8565
##              Prevalence : 0.7378
```

```
##          Detection Rate : 0.7030
##    Detection Prevalence : 0.7572
##       Balanced Accuracy : 0.8730
##
##        'Positive' Class : F
##
```

The lack of change between the original tree and pruned tree could be due to the imbalanced classes as related to the isSpam feature. As noted previously, there are more non-spam emails than spam emails in the data set. This could be impact the overall model accuracy. In order to overcome this possible over fitting evaluating the data leveraging a cross fold validation method may be called for. The cross fold validation method will also assist in dealing with the class inequality in the spam and not-spam values.

In cross fold validation, the dataset is broken up into a number of groups. Each of these groups is called a fold, denoted by the value 'k'. Each fold is then used as a testing and training data set. For purposes of this exercise k will be equal to 5 and this will be repeated 5 times.

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



CV Tree − Initial

Using this new tree development approach resulted in a much larger tree than the first tree build. Evaluating the confusion matrix, shown below, of this new approach will provide guidance on whether the predictions are getting better or worse.

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction    F    T
##          F 1296   73
##          T   38  401
##
##                Accuracy : 0.9386
##                  95% CI : (0.9265, 0.9492)
##     No Information Rate : 0.7378
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8374
##
##  Mcnemar's Test P-Value : 0.00125
##
##             Sensitivity : 0.9715
##             Specificity : 0.8460
##          Pos Pred Value : 0.9467
##          Neg Pred Value : 0.9134
##              Prevalence : 0.7378
##          Detection Rate : 0.7168
##    Detection Prevalence : 0.7572
##       Balanced Accuracy : 0.9088
##
##        'Positive' Class : F
##
```

The confusion matrix for the cross fold validation approach resulted in a higher accuracy of approximately 93%. Of the 1809 records in the test data set, 1296 were accurately classified as not-spam and 401 were accurately classified as spam. Taking this approach has resulted in better accuracy, however there are still other methods that will provide better results with respect to other metrics that will provide more confidence in classifying emails as spam or not-spam.

## Random Forest

Another approach that can be leveraged for classification is Random Forest. Using Random Forest multiple trees are built that are compared together to evaluate which tree is the best. Additionally, with Random Forest the most important variables that impact the tree creation can be evaluated. For the analysis of Random Forest 500 trees will be generated and the default for classification of using the square root of the number of features will be used when evaluating the number of features used in the construction of each tree. The model summary is presented below.

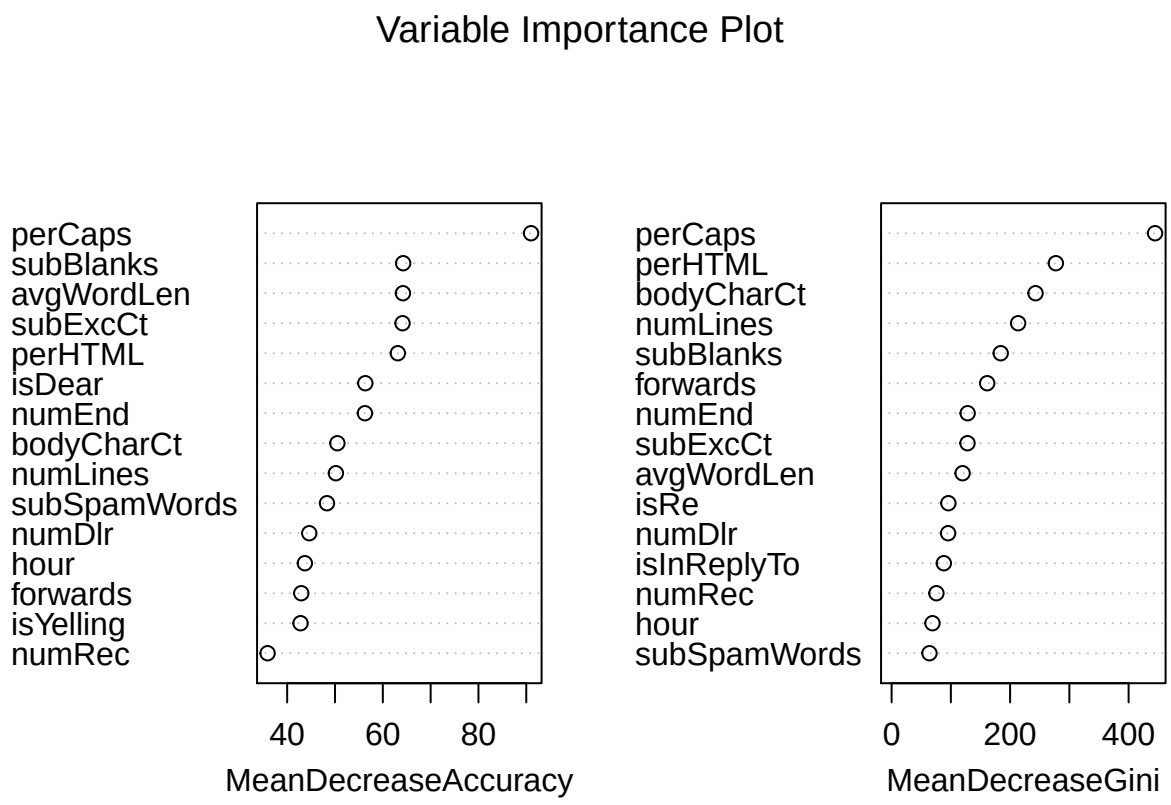
```
##
## Call:
##  randomForest(formula = isSpam ~ . - isSpam, data = train_df,      ntree = Ntrees, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 2.28%
## Confusion matrix:
##      F    T class.error
## F 5281   59  0.01104869
## T  106 1791  0.05587770
```

From the model summary on the training data it can be seen that 500 trees were build with 5 variables attempted at each split. The random forest was able to classify 5281 of the non-spam emails correctly, and 1791 of the spam emails correctly.

Continuing to dig into the random forest models, the important variables can be determined. Variable importance is presented two different ways in the plots below. The first plot represents the top 15 features in descending importance based on the mean decrease accuracy metric. The mean decrease accuracy metric describes the impact each feature has on the model if it is removed from the model. Similarly, the second plot represents the top 15 features in descending importance based on the mean decrease gini. The mean decrease gini evaluates the model based on how each feature impacts the gini index that is calculated.

In looking across both feature importance plots a few key observations are present:

- The perCaps feature is the most important
- Both plots contain a lot of similarities in important features




Now that there is an understanding of how accurate the Random Forest model is on the training data set and what features are important to the model, the random forest model can now be run on the test data set to determine how accurate the model is. The confusion matrix below indicates the results.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    F    T
##          F 1321   31
##          T   13  443
##
```

```
##                Accuracy : 0.9757
##                  95% CI : (0.9675, 0.9823)
##     No Information Rate : 0.7378
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9363
##
##  Mcnemar's Test P-Value : 0.01038
##
##             Sensitivity : 0.9903
##             Specificity : 0.9346
##          Pos Pred Value : 0.9771
##          Neg Pred Value : 0.9715
##              Prevalence : 0.7378
##          Detection Rate : 0.7306
##    Detection Prevalence : 0.7478
##       Balanced Accuracy : 0.9624
##
##        'Positive' Class : F
##
```

Utilizing random forest the model accuracy has increased to 97%. The model successfully classified 1321 non-spam emails as spam and 443 spam emails as spam. Indications seems to indicate that this model is better than the initial tree based models that were built. To a degree this is expected since a random forest classifier is building multiple trees versus the single trees in the initial outset.

# Conclusion