**Case #3**

Using the “emailDFrp” dataset:

1. Build and evaluate a tree-based model for predicting “spam”

**Data preparation :**The data is generated into a raw text file which we used RdaReader in Python to read in the dataset called “emailDFrp.” To get an understanding of how the data is organized. The data set includes 30 attributes and 9,348 data points. Within the attributes there are a number of categorical attributes, including the response variable. These categorical variables help determine if the email is spam or not. Other types of variables in the data set are comprised of metadata or detail for each email, for example average word length or number of lines in the body of the email.

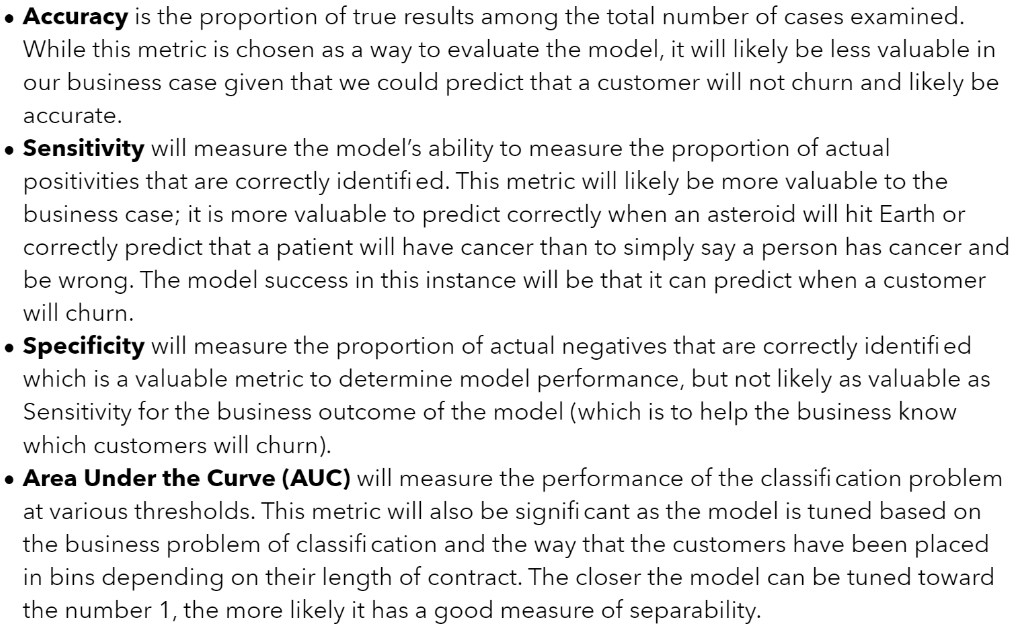
And there are systemic null values with the original data, and we decided to treat them as ‘False’ or ‘Zero’. For instance, there is a feature called ‘isYelling’ to indicate if an email writer used Uppercase letters with email subject. For some of emails, the Subject element itself is missing, so this data is null. We decided to make this as ‘False (Not Yelling)’.

**Model selection and evaluation metric:**

We took Random Forest Classifier, one of the decision-tree based classification algorithm to predict spam and no-spam emails. The classifier provides interpretable or explainable model such as feature importance while providing good training and testing performance

Considering the imbalanced nature of the target variable, we decided to use ‘F1 Score’ as a primary metric to balance between sensitivity and specificity while using the accuracy and AUC as secondary reference metric.

* F1 Score: The F1 score is the harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall).

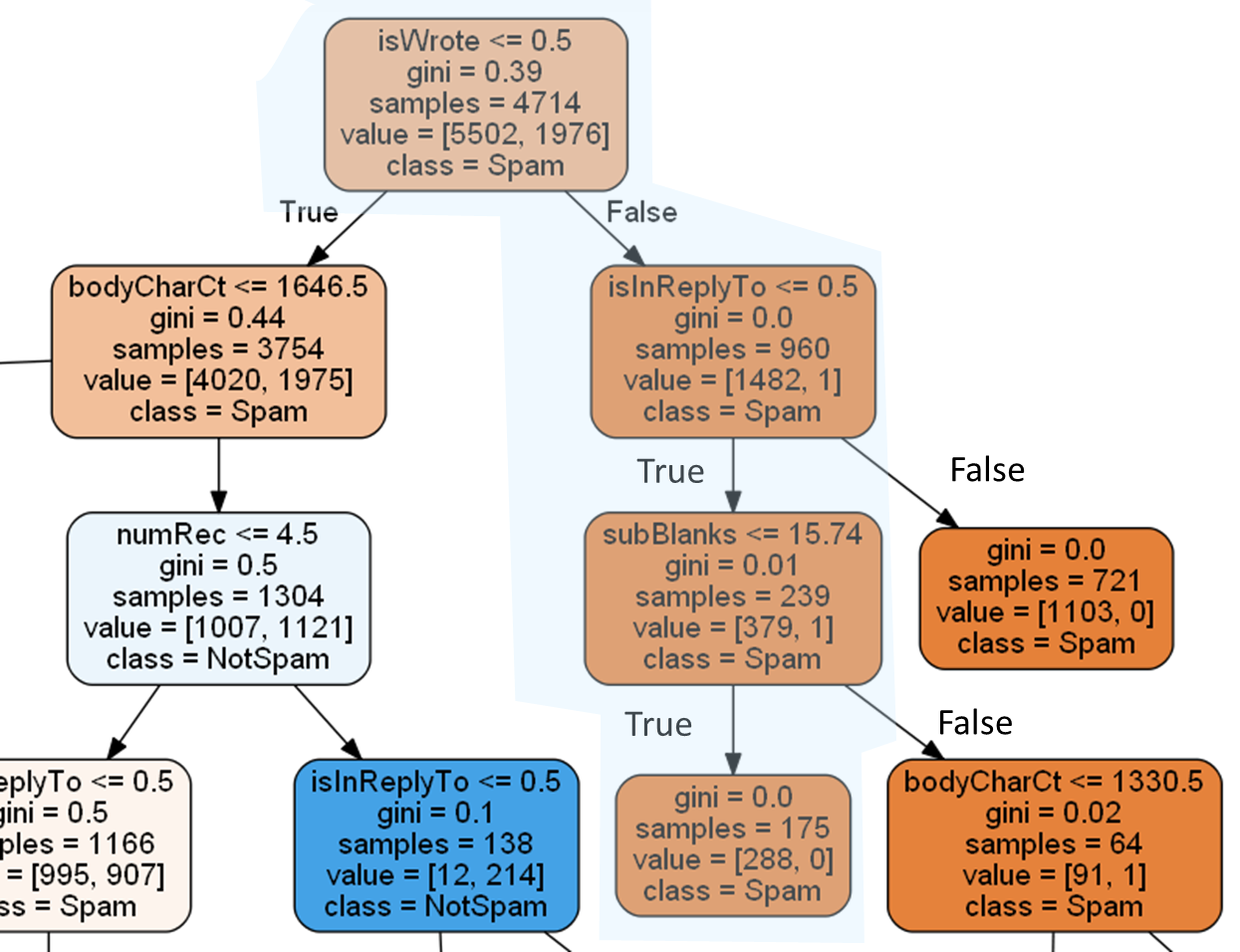


*(To Lance, I couldn’t open the ML1 HW 2 PDF but was only able to see it with Github as image. Adding the section as a screen capture, but hope you add it as text. – Shawn)*

1. Plot and analyze the paths through one (or many) of your trees

We selected one of the paths that identifies spam. It appears that the path makes sense. An email sent without any previous conversation, and includes sentences such as ‘As a price of distance country, I wrote this letter to get your help to pull my money from swiss bank’.

* Step1 – isWrote <=0.5   
  🡪 False (A message includes ‘wrote’)
* Step2 – isInReplyTo <= 0.5   
  🡪 True (A message header does not contains ‘In-Reply-To’ keyword)
* Step3 – subBlanks <= 15.74  
  🡪 True (A message subject has blanks character lower than 15.74%)



1. Explain the parameters involved in “tuning” your model

We decided to tune the model with following parameters and used the grid search method.

-. Criterion: gini & entropy (to decide which criterion gives us good discrimination on target variables)

-. Max depth: from 4 to 10 (to decide optimal tree depth that gives us good performance while avoiding over fitting)

-. Minimal samples per leaf: 2 to 10 (to avoid overfitting)

Iterating through options, 5-fold cross-validation was used to evaluate the predictive qualities of the hyperparameter combination by partitioning the original data into a training and test data set. The ‘validation’ dataset was set a side for independent performance evaluation.

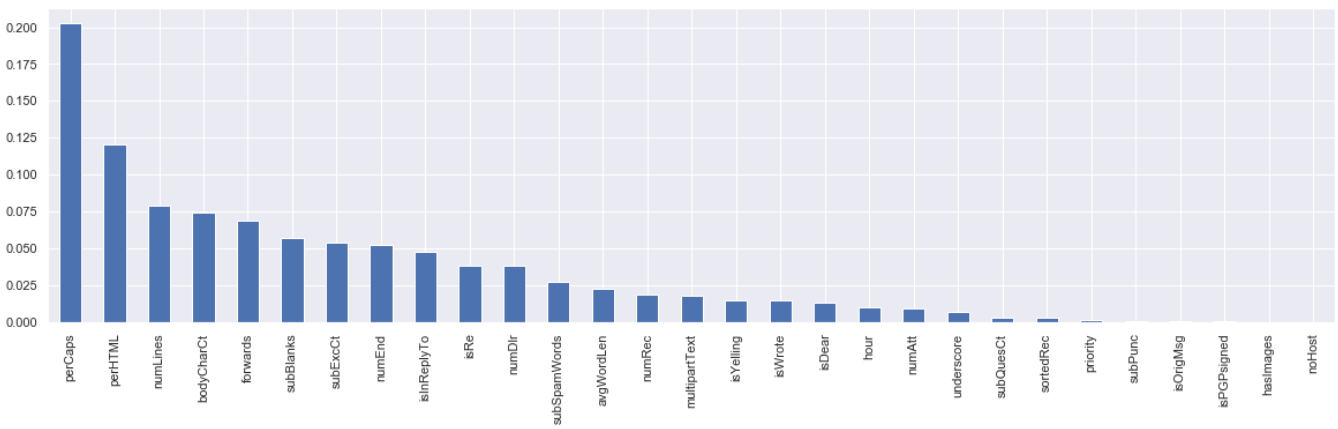
1. Which variables were “most” important?

It turns out that top three important variables are as following. From our experience of getting ‘CHEAP RAYBAN SUNGLASSES’ emails with html-laden message bodies, the result makes sense.

-. perCaps(percentage of capital characters)

-. perHTML(percentage of chracters in HTML tags)

-. numLines (numer of lines in a message)



1. How did you evaluate the “performance” of your model?

By comparing the grid search results with f1-score, we found that ‘gini’ criterion with maximum tree depth 10 and minimal sample leaf 2 showed the best f1-score of 0.90.

When applying the parameters to a split validation data, we could get nice and balanced result.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F1-Score |
| Not-Spam (0) | 0.97 | 0.98 | 0.97 |
| Spam (1) | 0.95 | 0.90 | 0.92 |
| Accuracy  (weighted average) | 0.96 | 0.96 | 0.96 |

Specificity: 0.98, sensitivity: 0.90

The ROC AUC value is also looks fine: 0.94

