Detecting Spam Emails - Analysis Plan

Goal Statement: Investigate the categorization of an email as spam or not spam. Be able to predict, with 80% accuracy, whether or not an unseen email is spam based on the contents of the email.

Research Question: How accurately can we predict whether or not an email is spam based on the contents of the email? How do the accuracy results vary across different classifiers?

Modeling Approach: We will use email text data collected from Kaggle for this analysis. We plan to build our models using Random Forest and Logistic Regression, to determine which model is more accurate for email spam prediction. The downloaded dataset includes a variable indicating whether the email is spam or not spam, a text variable containing the subject line, and a text variable containing the body of the email. First, we will clean this dataset to remove the most common words across every observation (we suspect these will be words like is/and/hi/the, etc.). This will ensure that we have key words leftover to analyze. We will examine the words in the subsetted data set by whether or not the email is spam to determine the most common keywords used in spam emails and the most common keywords used in non-spam emails. We will create dummy variables representing the use of these words to add into our models. We will also create new numerical variables representing the number of words in the title, in the body of the email, etc. Again, we plan to create a Random Forest Model and Logistic Regression model for this analysis. We chose to use a Logistic Regression model for its simplicity and interpretability, and a RF model for its ability to handle a large number of input variables [3]. We will conduct PCA in order to reduce our number of variables so this model is less computationally expensive. We will also use cross validation to increase the accuracy and reliability of our models. Lastly, we will examine the evaluation metrics of each model and choose one.

Executive Summary:

In this document, we discuss the email data we are working with, the way we cleaned the data, and the modeling approach we plan to complete. The original data has 3 columns and 84 rows, but we created new columns such as word count, sentence count, and dummy variables representing whether or not certain words were used in each email. We plan on using these calculated variables in our Random Forest and Logistic Regression ML Models, in order to predict whether or not an unseen email is spam.

Data Set Establishment Details:

Each row in this dataset is an email. Each email has a title, a body, and a classification of whether or not it is a spam email. There are 84 emails (rows) total in this dataset. The data dictionary for this data is below.

Title	str	A string containing the subject line of the email.
Text	str	A string containing the body text of the email.
Type	str	Whether or not the email is spam - “spam” or “not spam” (response variable)

In our EDA, we intend to discover which words are most common in spam versus non-spam emails. We are currently hoping that these words are distinct words which will help us the most in our analysis, but it is possible that they are also filler words such as “so” and “of” or that there are not enough words that occur that many times in the data set at all. We also are curious about the potential correlation

between title length and/or length of the body of an email and whether or not it is spam. We suspect that these variables will be useful in our analysis, but this is another unknown at this time.

During exploratory data analysis, we asked the following questions:

1. What is the spread of the word count and sentence count for the title and body of each email? Do we see any trends? What is the mean for each of these counts?
2. Is there a relationship between word count and type (spam/not spam) for the title and body of emails in our dataset?
3. What classifies a word as "common" in an email and what are these common words? Do we see common words that are unique to "spam" or "not spam" email types?

To answer Question 1, we assembled a histogram to display the frequency of word and sentence counts for the title and body of the email. We found a rather large spread for word counts in the bodies of emails with counts ranging from 3 to 1088. The histogram is skewed-right with a mean of ~141 words, indicating email bodies tend to be shorter. Sentence counts in the bodies of emails shared the same trend with a minimum of 0, a maximum of 104, and a mean of ~11.

The title counts had less variance and could be fit to a normal distribution. We found a mean word count of ~6 for spam titles and ~7 for non-spam titles. Though they have similar means, we find that spam emails are shorter (0-2 words) more often than non-spam emails.

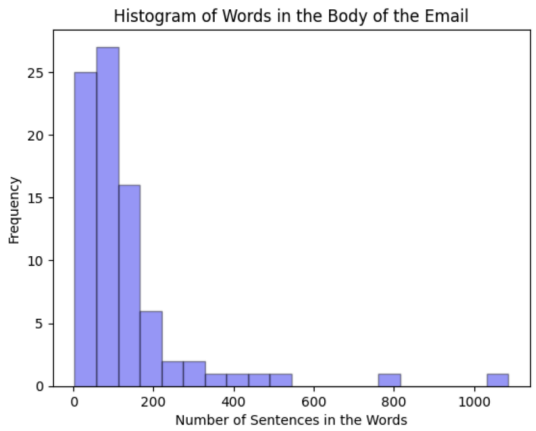
To answer Question 2, we used scipy.stats to perform two-sample T-tests on title word count and text word counts (spam and non-spam being the two samples). We found no statistical significance for the number of words in the body of a spam email versus a non-spam email, meaning we cannot conclude that their respective mean word counts are indicative of classification. We found a T-statistic of 1.7579 and a P-value of 0.0843 for the number of words in the title of a spam email versus a non-spam email (with alpha=0.1), indicating statistical significance. Thus, we conclude that the mean title word count can be indicative of classification. This is just one of many significant variables we explored during EDA.

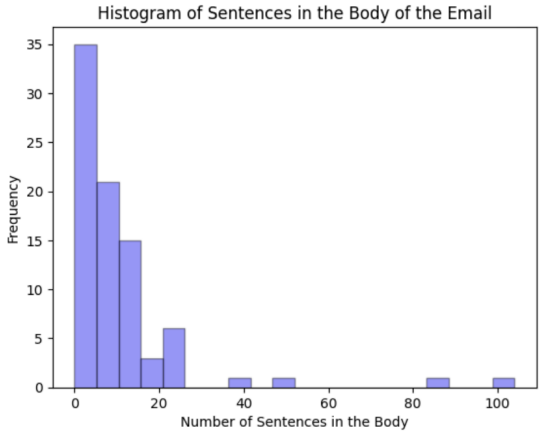
Answering Question 3, we defined a common word as one appearing more than 10 times in the dataset and removed ambiguous words such as "and," "hi," etc.. Although there is overlap in the words "from" and "it," we found many common words unique to both "spam" and "not spam" emails, which can be seen graphically below.

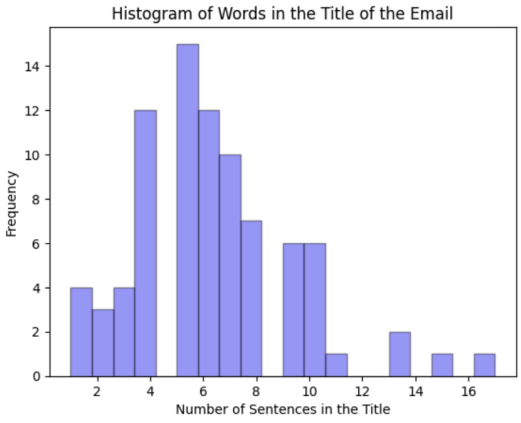
EDA:

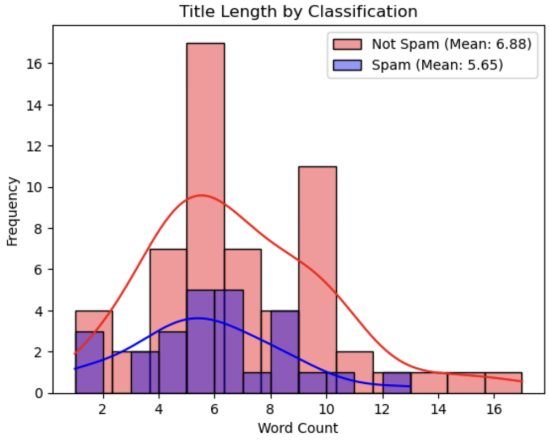
Classification	**Count**
Not Spam	58
Spam	26

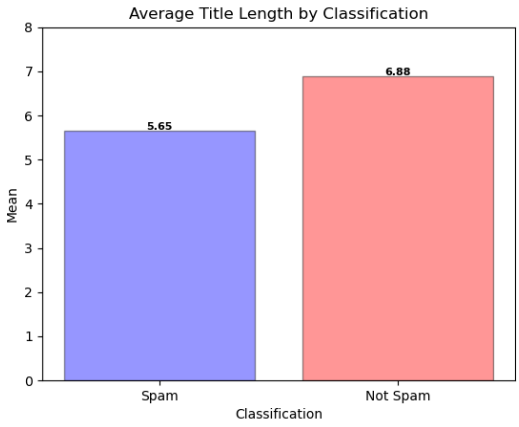
*__note__: Above, we can see that 58 emails are non-spam, and 28 are spam. While there are more non-spam emails than there are spam, 32.5% of the emails are spam, so class-imbalance should not be a large issue here for model building.

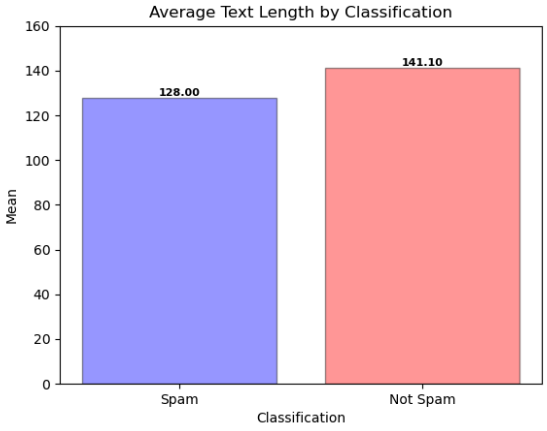

Histogram of Words in the Body of the Email
Frequency
Number of Sentences in the Words

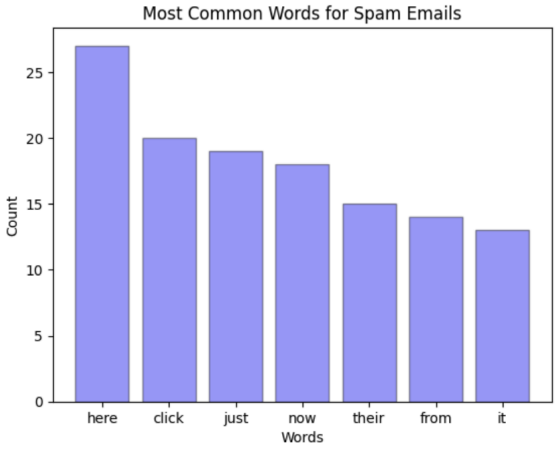

Histogram of Sentences in the Body of the Email
Frequency
Number of Sentences in the Body

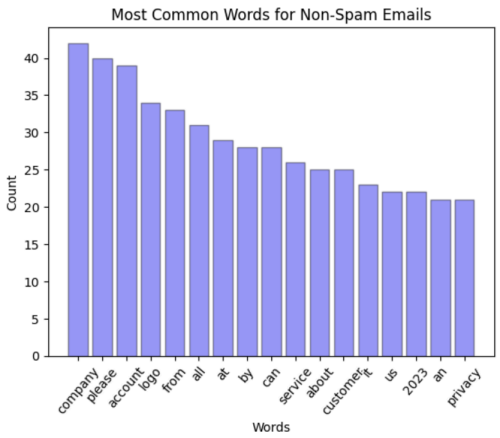

Histogram of Words in the Title of the Email
Frequency
Number of Sentences in the Title


Title Length by Classification
Not Spam (Mean: 6.88)
Spam (Mean: 5.65)
Frequency
Word Count


Average Title Length by Classification
5.65
6.88
Mean
Spam
Not Spam
Classification


Average Text Length by Classification
128.00
141.10
Mean
Spam
Not Spam
Classification

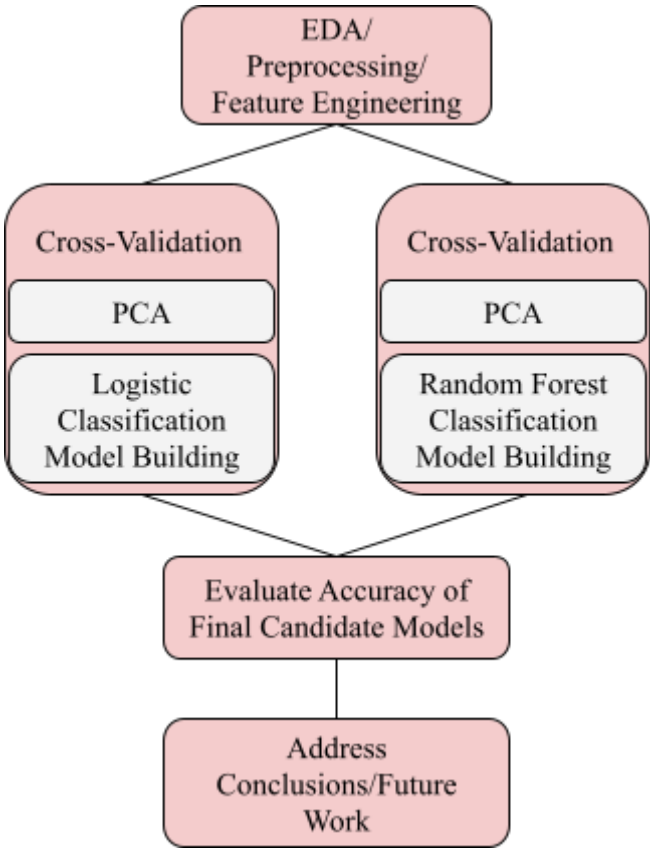




* **note**: there are more words than what is shown above that have a count of 10+ for Non-Spam emails. We have just included the words with a 20+ count here.

Analysis Plan:

Analysis Plan Roadmap:



Preprocessing:

There was a lot of data cleaning required in order to perform our desired analysis. First, we focused on the 'text' variable in the data set, which includes the body of the emails. We removed all instances of '\n' from reading in data with multiple paragraphs. We then counted all instances of

punctuation marks that indicate the end of a sentence, and stored it as a new variable 'text_sentences' for the number of sentences in the body of the email [1]. We then removed all punctuation and made the 'text' variable all lowercase so we were left with just the desired words. We split each email on spaces to obtain a list of the words for each email. The number of words in each email became another new variable, 'text_words'. Next, we needed to find the most frequent key words for all emails. We counted up occurrences of each individual word in the 'text' variable across the entire data set, and removed words that occurred more than 50 times. Based on a line graph of the word counts, 50 seemed to be where it tapered off from extremely common words such as "the," "hi," "and," etc. to just words that were meaningful in the emails. With all words included less than 50 times, we determined which words were the most commonly occurring. We saved 'spam_word_counts' as a series which shows the words occurring more than 10 times in the spam emails (not including words that occurred more than 50 times in all emails). We then repeated this with the non-spam emails to obtain the series 'not_word_counts'. We created these two series separately for the purpose of EDA and ensuring that both classes were represented in this subset, but the threshold was the same for each class. With the two resulting series, we created dummy variables for each word in the original data set, to show whether or not a given email included each of those words.

Next, we worked on the 'title' variable in the data set, and conducted similar preprocessing steps. We obtained a word count variable called 'title_words' in a similar way as we obtained 'text_words' in the previous steps. We did not create a variable for the number of sentences since typically email titles do not include multiple sentences. We then obtained a list of every word in the 'title' variable, and subsetted to only include words that occurred less than 10 times. The thresholds for too common and for the later key words in the title data were smaller since the titles are generally shorter than the bodies of the emails. With the subsetted data, we obtained the series 'spam_title_counts' and 'not_title_counts' which include all words used more than 2 times in the titles of emails (not including words that occurred more than 10 times in all email titles). We then created dummy variables in the same way to add even more indicator variables into the original data set. Lastly, we dropped the original 'text' and 'title' columns from the data set, since we have now extracted all of the information we need from them.

For a final step, we standardized all predictors in the data set to be centered at 0. The reason for this is that we plan to perform PCA, and we would like all variables to be on the same scale so they are evaluated appropriately. Before standardizing the variables, the word count and sentence count variables were much larger numerically than the dummy variables, which would cause problems had we not standardized. The final size of the data set after preprocessing is 84 rows and 104 columns.

Methodology:

We will use 5-fold cross-validation to train each of our candidate models. 5 was chosen for the value of k due to the size of our data, but we will consider another number of folds if our results are unsuccessful. In 5-fold cross-validation, we will split the data into 5 folds, train the model on 4 of the 5, and then test for prediction accuracy on the remaining fold [2]. We will repeat this process 5 times, calculating the accuracy each time, so that each fold is the test set once. Within this cross-validation, we will perform PCA on the training set to reduce the number of variables we have to work with in the models we intend to fit. Currently, our data set (after adding the calculated and dummy variables) contains 104 variables, which would be very computationally expensive to fit. Conducting PCA will allow us to identify the most impactful features to continue with a smaller set of variables. We will use 0.8 as our value for the number of components, as we want 80% of the variability to remain within our predictors. This process of cross-validation will allow us to maximize our accuracy by averaging all of these accuracies together from each fold. This will be particularly helpful in training our model, since our data set is relatively small with only 84 observations. We will complete this process to build both a Logistic

Regression model and a Random Forest model. These models were selected due to their appropriateness for classification and the size of our data set. Lastly, we will evaluate the final two models to determine which one will be our final model.

Evaluation:

After training both the Logistic and Random Forest Models using 5-fold cross-validation, we will evaluate the performance of our classifiers using the following metrics: Accuracy, Precision, Recall, ROC-AUC, and F1 Score. We will also make a confusion matrix to better understand the false-positive and false-negative rate for our classification.

References:

[1] A. Khan, "Email Spam Detection with Machine Learning: A Comprehensive Guide," Medium, Mar. 22, 2024. https://medium.com/@azimkhan8018/email-spam-detection-with-machine-learning-a-comprehensive-guide-b65c6936678b

[2] GeeksforGeeks, "CrossValidation vs. Bootstrapping," GeeksforGeeks, Jun. 27, 2024. https://www.geeksforgeeks.org/cross-validation-vs-bootstrapping/