Enron Email Spam Detection

John O’Rourke – IST 664 – Prof. Ying Lin

## 1 Introduction

One common approach to spam detection within emails is to use machine learning algorithms. We can see this through the current Outlook Protection® suite as well as Google’s Spam protection system for Gmail. Naïve Bayes is one key algorithm in classification of documents or items within a given dataset. This algorithm is used to calculate the probability of an email being spam based on the content within a given email.

In this project, we will investigate the overall effectiveness of the Naïve Bayes algorithm as well as the VADER sentiment lexicon in detecting spam within the Enron Email Corpus. This dataset contains a large number of emails (both legitimate and spam) that were sent and received by Enron employees. The goal of this project is to identify the most effective method and approach to detecting spam emails and to provide insights for improving our current spam detection methods.

## 2 The Problem

Spam in email is a constant and persistent issue many people encounter throughout their personal and professional lives. To combat this widespread issue, tool sets, and machine learning models have been implemented to counteract spam from reaching a user’s inbox.

This is primarily done through the tracking of sentiment and word usage within the email or by using key indicators within a networking or DNS instance. This analysis focuses on the former and attempts to determine spam based on the sentiment of a given email.

## 3 Dataset Information

### 3.1 Source

The data for this analysis has been sourced from the Enron Spam dataset which is a public email corpus gathered from the company’s court case. In total, 1,500 spam files and 3,672 “ham” files were used.

This only accounts for 1 of 3 spam groupings and 1 of 6 legitimate groupings. Additionally, spam emails were injected into the already classified spam emails in order to ensure a proper sample for analysis. This was completed by the dataset maintainers.

### 3.2 Transformations

The corpus for both ham and spam categories were transformed in the following ways:

1. Label column added to both tuples.
2. Merged and shuffled into a single tuple.
3. Column “length” added.
4. Removal of all:
   1. Stop words.
   2. Punctuation
   3. The phrase “Subject: “.

This preprocessing was done to ensure that all data would be viewed on equal ground and provide a clean foundation for analysis.

### 3.3 Descriptive Statistics

A word cloud was generated for each tuple of emails to determine the commonality between them as well as to understand future results.

The ham dataset showcased a high amount of corporate information including employee names, the word “Enron” and general corporate jargon such as “see attached” or “please let me know”.

The spam dataset while remaining corporate, showcased a higher amount of marketing lingo designed to lure in sales of a product or service with no specific information to Enron.

## 4 Model Development

All model development was completed with the use of Naïve Bayes as the classification method. The train and test split were set to 0.7/0.3 with a random state of 17 provided for each model.

Each model utilized scikit-learn’s “train\_test\_split” function to achieve randomized training and testing data.

### 4.1 Naïve Bayes without Sentiment Lexicon

With this model, the Bag of Words feature matrix was generated with the CountVectorizer algorithm. This algorithm accounts for the frequency of a given word within an individual document. Naïve Bayes was utilized to classify the documents based on their values. This model achieved an accuracy of 97.74% and correctly identified 1,517 out of 1,552 used documents.

As well as a high model accuracy, the cross-validation scores, recall scores and F-measure scores are all high values. The cross-validation scores showcased a high-level of consistency with scores ranging from 0.966 to 0.985. The recall score was 0.965 which indicates that the model could effectively identify positive sentiment values within each document. Lastly, the F-measure score was 0.966 which results in the model showcasing a good balance between precision and recall.

Overall, this model performed effectively and was highly accurate and consistent in classifying the documents.

### 4.2 Gaussian Naïve Bayes with Sentiment Lexicon

For our second model, the Bag of Words feature matrix was generated with the CountVectorizer algorithm. Gaussian Naïve Bayes was utilized to classify the documents based on their values. The inclusion of the VADER (Valence Aware Dictionary and sEntiment Reasoner) lexicon in this model allows for an increased number of features for the model to use for classification. VADER assigns a positive, negative, or neutral score to each word in each document and calculates it to the column “composite” on a scale from -1 to 1. This model achieved an accuracy of 95.36% and correctly identified 1,480 out of 1,552 documents.

A majority of inaccurate ratings were due to false negatives wherein the model mistakenly classified a large number of “ham” as “spam”. The cross-validation scores of this model we high and consistent, ranging from 0.948 to 0.968. As for recall and F-measure scores, the model resulted in 0.890 and 0.923 respectively. These scores showcase that while not as accurate as the model without the sentiment lexicon, the model was highly accurate and with minor improvements, could become more refined and result in fewer false negatives.

### 4.3 Gaussian Naïve Bayes with Sentiment Lexicon and TF-IDF Vectorization

Unlike models 1 and 2, this model utilizes the TF-IDF vectorizer algorithm to generate the Bag of Words feature matrix as an input to the Gaussian Naïve Bayes classification algorithm. TF-IDF (Term Frequency-Inverse Document Frequency) vectorization algorithm is similar to CountVectorizer but provides a weighted value based on a given words frequency in a document as well as the frequency within the entire corpus. This assists in identifying the key words that are more important for larger classification problems.

The model achieved an accuracy of 94.65%, correctly identifying 1,469 out of 1,552 documents. Compared to our previous models, this is the least accurate of the bunch. Inaccurate classifications are evenly distributed between both false positives and false negatives showcasing that while inaccurate, it did not have any imbalance with regards to its classification. Cross-validation scores for this model averaged 0.954 with recall and F-measure scores at 0.909 and 0.917 respectively. Like the previous model, this model has a high degree of accuracy and a higher level of precision.

## 5 Evaluation

Within each model, are strengths and weaknesses as can be seen with the varying accuracy of each individual model. In evaluating these models, we can see the impact of including a sentiment lexicon as well as the impact the vectorization algorithm choice holds in proper classification.

The first model, which used Naïve Bayes (Multinomial) with the Bag of Words feature matrix generated by CountVectorizer, achieved the highest accuracy of 97.74%, with a high level of consistency and reliability in cross-validation scores, recall and F-measure score. However, the limitations of CountVectorizer in accounting for word frequency across the entire corpus should be taken into consideration.

The second model utilized Gaussian Naïve Bayes with a B.o.W. feature matrix also generated by CountVectorizer and the introduction of VADER shows a slightly lower accuracy of 95.36%. This model also shows bias in that it favors misclassification of documents as false negatives. This model’s thresholds and parameters will need to be adjusted for further improvement.

The third model, which utilizes the same structure as the second model with the substitution of the vectorization algorithm to TF-IDF has the lowest accuracy at 94.65%. Unlike the second model, this model has a balanced distribution of misclassifications between both false positives and false negatives. This suggests that the model has better accuracy classifying sentiment but overall struggled classifying documents.

With this evaluation complete, it can be determined that each combination of vectorization algorithm and sentiment lexicon is key to the success of each model. It can also be seen that certain methods and parameters may introduce bias and display limitations of their respective algorithms.

## 6 Results

The overall results of this project showcase that a high level of accuracy can be seen across each iteration of the models. Each model had increasing numbers of errors and introduced bias in certain cases. Cross-validation, recall and F-measure scores remained high for all three model iterations and therefore, each model can be seen as an effective method of accurately identifying spam within emails.

## 7 Conclusion

With all things considered, this projects’ results are those which I am pleased with. Each model had a high degree of accuracy given the provided feature sets and allows for further research into spam detection through the use of machine learning.

Ideally, further research would be done using other classification algorithms and sentiment lexicons. Specific models that will be tested moving forward are Support Vector Machines, Random Forest, and Gradient Boosting. Gradient Boosting in particular would be influenced by VADER and should result in higher overall accuracy and better classification.

One additional area of interest with regard to this dataset is utilizing the subject line of each email to determine if spam identification can be completed without the body of the given email. With the provided corpus and text files, this would only require additional pre-processing that would prove to be fairly simple.

Overall, the models developed allow for accurate and effective spam detection and can be further improved with time and further training optimizations.

## 8 References

Androutsopoulos, I. (n.d.). The Enron-spam datasets. Retrieved March 21, 2023, from https://www2.aueb.gr/users/ion/data/enron-spam/

Metsis, V., Paliouras, G., & Androutsopoulos, I. (n.d.). *Athens University of Economics and Business*. Retrieved March 21, 2023, from https://www2.aueb.gr/users/ion/docs/ceas2006\_paper.pdf

Pang, B., Lee, L., & Vaithyanathan, S. (n.d.). *Thumbs up? sentiment classification using Machine Learning Techniques*. ACL Anthology. Retrieved March 21, 2023, from https://aclanthology.org/W02-1011/

*Resources and dataset descriptions¶*. Resources and Dataset Descriptions - VaderSentiment 3.3.1 documentation. (n.d.). Retrieved March 21, 2023, from https://vadersentiment.readthedocs.io/en/latest/pages/resource\_description.html