**IST 664 – NLP Final Project**

**Text Classification – Email: Ham VS. Spam**

**6/23/2021**

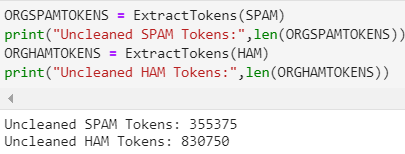
**David Doman**

**Introduction & Background**

This homework is focused on Classification and focuses on developing features for the task. In addition, various experiments are carried out exemplifying which sets of features are best suited for the data. Three possible options were provided for analysis but the chosen source for this assignment is the Enron SPAM emails. Two sets of emails are provided, 3,672 regular emails labeled as “ham” and 1,500 spam emails labeled as Spam. The following report contains the analysis and experiments performed on these emails.

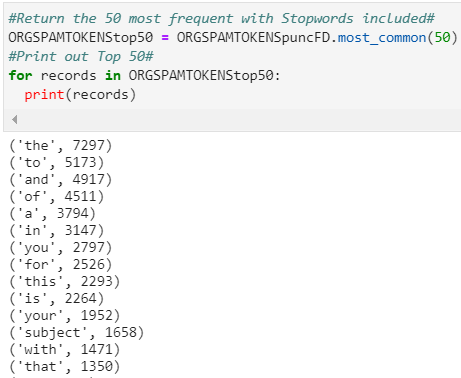
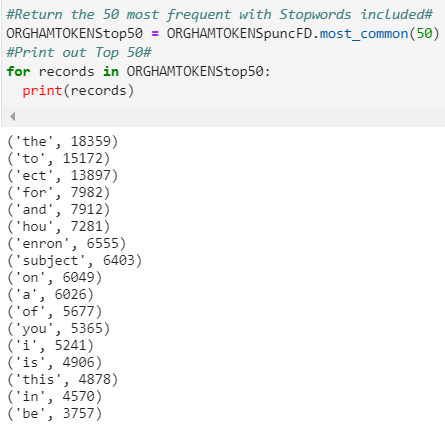
**Data Cleaning & Analysis**

To start, I imported the Spam and Ham emails using defined function. I then tokenized both the Spam and Ham e-mails and added a “Spam” or “Ham” label to each respective e-mail. Initially, the Spam e-mails contained 355,375 tokens in total and the Ham e-mails contained 830,750. The disparity in the number of tokens between Spam and Ham makes sense since there were drastically more Ham e-mails than there were Spam e-mails.



Viewing the first 100 or so tokens for each set of e-mails, it was clear that further cleaning needed to be done. I did view the Bigrams and Trigrams first before any cleaning was done but this did not provide any valuable information. Cleaning the tokens included removing punctuation and converting all tokens to lowercase. I next decided to look at the frequency distribution of the top 50 most common tokens in each set of e-mails. Below is a snippet of each:

***Spam Ham***

As was expected, there are many words in the top 50 most common that are stopwords such as “the”, “to”, “of”, etc. The nltk corpus stopwards were then called and removed from the tokens. After doing so, the number of tokens in for Spam was cut down to 179,807 and Ham was trimmed to 323,901. The frequency distributions were then viewed again to get a better idea of what the meaningful tokens. In terms of the objective of the analysis, which is to try to be able to best identify between Spam and Ham e-mails, I thought it would be beneficial to look at the most common words between the two sets of e-mails. Having a lot of very common words in each set of e-mails will make it more difficult to distinguish as to whether an e-mail is Spam or Ham. I did this by taking the 100 most common words in each set of e-mails and then added any words that were in both list to my stopwords. It was found that there were 27 such words, so these were added to the stopword list. These contained words such as “company”, “information”, “mail”, “net”, for example.

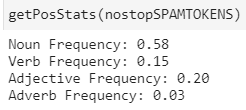
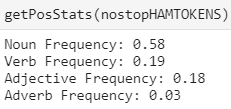
***Common words between Spam and Ham***



After adding these 27 words to the stopword list and removing these from the Spam and Ham e-mails, the total tokens for Spam was 168,557 and for Ham 292,733. Frequency lists were then viewed again to get a better understanding of the data.

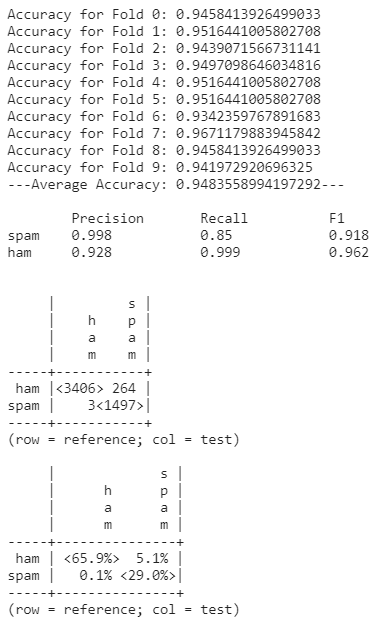
Now that the cleaning was complete, I thought it would be beneficial to look at the Bigram and Trigrams to see if any further value could be obtained. Top Bigrams for Spam included (“http”,”www”), (“looking”, “statements”), (“copy”, “past”) along with a lot of bigrams containing non-words that were just letters. This type of text and misspellings are frequent in Spam e-mails, so this makes was expected. For the Ham Bigrams, combos like (“hou”,”ect”), (“enron”, “cc”), (“north”, “America”) and a lot of bigrams containing first and last names. These bigrams make a lot of sense since they relate to Enron and its business, and the people constantly involved in sending and receiving e-mails. Trigrams were also viewed but they did not seem to provide any real additional value to the Bigrams. PMI Bigrams and Trigrams were also produced to determine if any further information could be obtained from this form of analysis.

The next step was to get into some experimentation using POS Tagging, Naïve Bayes, Cross Validation, etc. Initially, I used to POS Tagging to look at the distribution of Nouns, Verbs, Adjective, and Adverbs contained in the Spam and Ham e-mails and if there was a substantial difference. They were very similar in their distribution of POS usage. Results for both are shown below:

As shown, both sets of e-mails use Nouns most frequently with similar usage with slight variation between verbs and adjective usage.

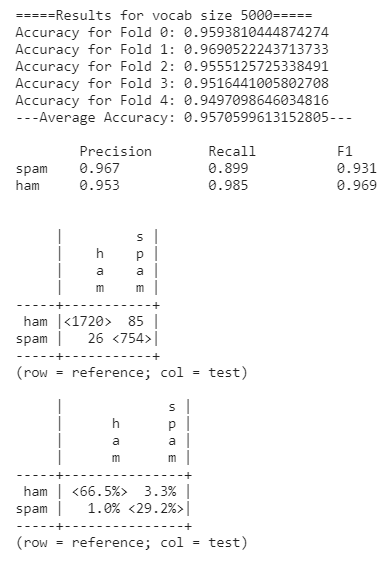
The first experiment run was to provide a baseline prediction model using test and train set. Below are the results of this experiment using 10 folds.



As shown, the precision, recall and F1 are all in the 80-95% range which are all very good, so this was model performed well.

Next, a model was run to look at the prediction accuracy by looking at different levels of vocabulary size. Iterations of size 100, 500, 2000, 5000, 7000, 10000 and 15000 were run, each with 5 folds. Results showed that Vocab size of 100 had an Average Accuracy of .877, 500 vocab size had an average accuracy of .915, 2000 vocab size with an average accuracy of .949, 5000 vocab size with average accuracy of .957, 7000 vocab size with average accuracy of .952 and 10000 with average accuracy of .943. So, from the results there is an accuracy increase as vocab size increases until it reaches a vocab size of 5000. As the vocab size increases above 5000, the average accuracy begins to decrease the higher we go. So, from this model, the peak accuracy of .957 occurs at the 5000-vocab size model.

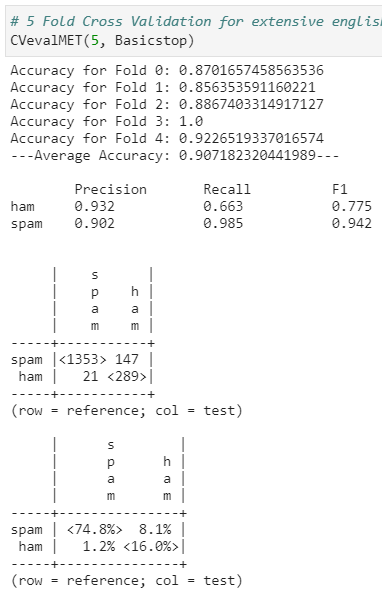
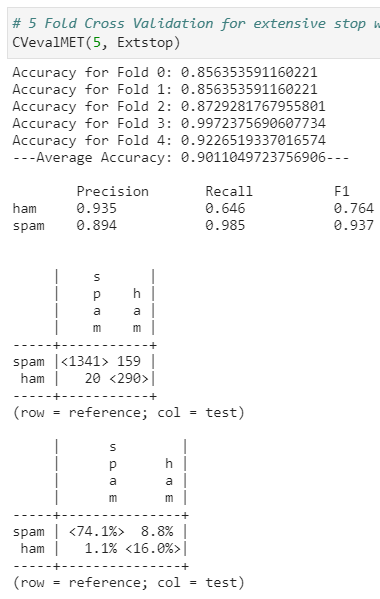
***Optimal Model***



The same model was run as previously described but this time using Bigrams instead. Using Bigrams in this type of model proved to be much less accurate than the prior model explained with average accuracies in the 70-85% range. A peak for this model occurred at 500 Bigrams with an average accuracy of .821 and an average PMI of .720. The Bigram model produced a much higher percentage of incorrectly labeling e-mails as Ham when they were actually SPAM.

The last two models involved looking at accuracy when the basic nltk corpus stopwords were removed and then the last model analyzed the accuracy with the full list of stopwords including the ones that I had added in addition to the nltk corpus stop words. With the basic stopwords removed the model produced an average accuracy of .9072 with the Recall and F1 for Ham being fairly poor. The final prediction model excluding the total list of stopwords resulted in very similar results with an average accuracy of .901 with the Recall and F1 for Ham being on the poorer side compared to Spam Recall and F1.

***Basic Stopwords Removed Total List of Stopwords Removed***

In conclusion, all of these models produced fairly accurate results. I believe running additional types of analyses along with improving the stopword list could improve the prediction accuracy in the models. Furthermore, some sentiment analysis, although may be more difficult than would be in analyzing tweets, could be beneficial and very interesting.