Alonso Salcido, Chris Haub

Lab 3, Email Filtering

**Introduction**

For this project we developed a model that could automatically filter and categorize incoming emails based on their content as either spam or not spam. Benefits of email filtering include improved productivity as spam emails consume time and unnecessary energy. Proper spam filtering also improves security as often spam emails contain malicious links and attachments. Finally, spam filtering will result in reduced costs as there is a large cost associated with storing and processing large volumes of unwanted emails.

**Methods**

5 folders containing over 9 thousand emails were provided for review. Two of the folders contained emails that were spam messages and three were folders that contained messages that were not spam. Spam emails made up over 25% of the dataset. Initially extensions that were not relevant to our search were excluded to focus only on the email files in each folder. Since emails were formatted different, they were read in using the import email function, import os, and beautiful soup. Next the formatting was cleaned up so that emails could be turned into bags of words and avoid irrelevant characters or strings. The “\n”’s were replaced with blank spaces. The URLs were replaced with the word “URL” to fit all URLs under the same word in the dictionary. We also removed unwanted characters and numbers.

The Count Vectorizer function from scikit-learn was used in the text analysis for natural language processing (NLP). This was done to convert the email text into arrays. This allowed for counting the frequency of each word in the document as well as filtering to remove stop or common words.

A reverse dictionary was created to translate back and forth between the vector and text. This step was necessary to double check the dictionary that was created using the count vectorizer function.

The final model was created using Multinomial Naïve Bayes algorithm with an alpha of 0.1. The alpha parameter in this context specifies the smoothing parameter for the model. It is used to handle the case where some features or words in the test data are not present in the training data. Without smoothing, the model would assign a probability of zero to any word that is not present in the training data.

**Results**

The model was trained and tested using 10-fold cross-validation with a Naive Bayes classifier. The metrics used to score the model were accuracy, precision, and specificity.

Accuracy (0.9753 (+/- 0.0786) is the proportion of correct predictions made by the model compared to the total number of predictions. The value of 0.9753 means that the model correctly classified 97.53% of the emails. The "+/- 0.0786" in the accuracy score represents the standard deviation of the accuracy scores across the 10 folds of the cross-validation. This gives an idea of how consistent the model's performance is across different samples of the dataset.

Precision (0.9592) is a measure of how accurate the positive predictions made by the model are. A precision of 0.9592 indicates that when the model classified an email as spam, it is correct about 95.92% of the time.

Specificity (0.9620) is the proportion of negative instances that are correctly classified by the model. A specificity of 0.9620 means that the model correctly identifies 96.20% of the non-spam emails.

**Conclusion**

Based on the evaluation metrics of accuracy, precision, and specificity, it can be concluded that the Naive Bayes classifier used in this study is performing well with a high level of accuracy. The model achieved an accuracy of 97.53%, with a precision of 95.92% and a specificity of 96.20%. This indicates that the model can correctly predict the class of the majority of instances in the dataset, both for positive and negative classes.

Moreover, the standard deviation of the accuracy scores across the 10 folds of cross-validation is relatively low, suggesting that the model's performance is consistent across different samples of the dataset.

The results suggest that the Multinomial Naïve Bayes classifier is a good choice for this particular problem and has the potential to generalize well to unseen data.

**Appendix AText

Description automatically generated** **Text

Description automatically generated** Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated