Samuel McMillan & Akhila Akkala
A02261822 & A02442064

Project 5 Naive Bayes Classifier

[Presentation Slides]
[Github Repo]

Introduction

Spam emails are something that are very invasive in the world today. Many of these spam emails are made to cheat whomever is receiving them out of money or other prized possessions. It is therefore imperative that one creates a model to filter out the spam emails from those that are not spam. Furthermore, this model needs to not label important emails as spam to ensure that the end user is receiving all the emails they want and not receiving the other emails. The model used to identify spam or not spam will be the Multinomial Naive Bayes. The particular variables the model will look at will be the different words in the email and their frequencies. Through proper manipulation of words to be fed to the model, one can see that this spam detecting model will become quite effective.

Dataset

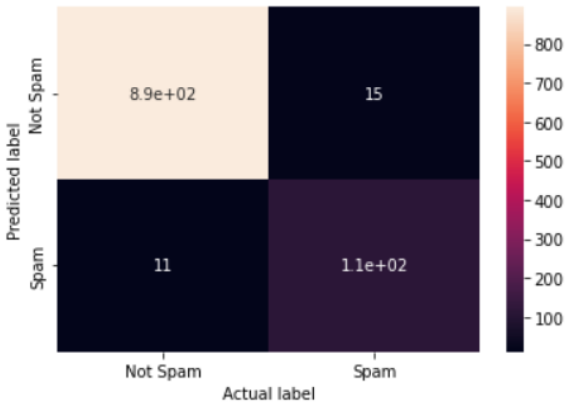
The dataset used in training the model contains 5000+ emails. Each email entry contains the email contents in one column and the label of spam or not in the other column. Due to the dataset's reasonably large pool of emails to look through and detect spam in, one can conclude that the model will have enough pieces of data to train on.

Analysis Technique

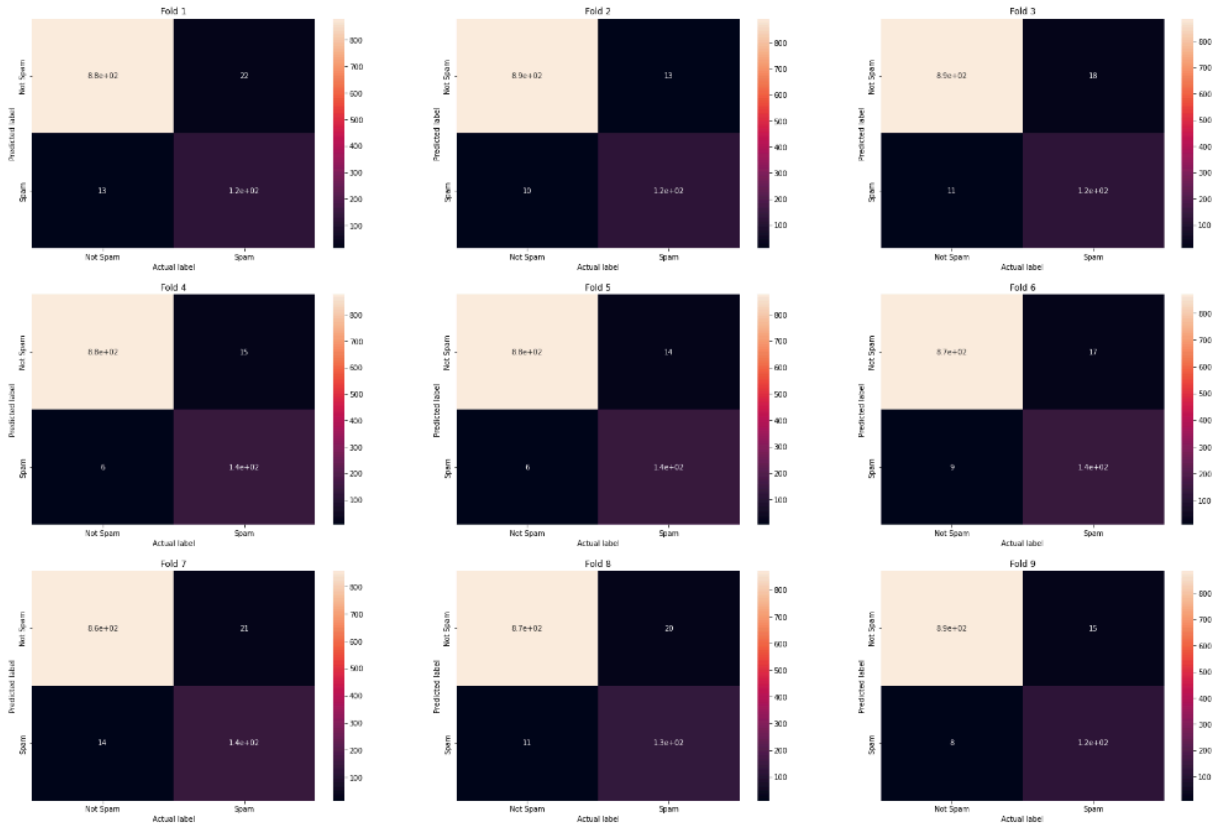
The classifier chosen for the dataset was the multinomial naive bayes. This is because the model detected spam based on the frequency of words in each email. Since frequency was a discrete variable instead of a continuous one, multinomial was chosen; the variation of naive bayes classifier Gaussian looks at continuous variables which weren't used in the model.

Results

The figure on the right displays the model's effectiveness at predicting spam. Paying attention to the few faulty predictions for an email that was spam but labeled as not spam or vice versa as compared to all the rest of the labels clearly shows the model is very good at labeling spam and not spam. This means that whomever would use this model as



their spam filter wouldn't be getting lots of spam in the inbox. In a similar way, whoever uses this model wouldn't be getting many important emails wrongly labeled as spam in the spam folder.



The screenshot of heat maps displayed above describes the being trained and tested with 9 different groups of data. The model for finding spam remained relatively the same for all, suggesting that the model can be very reliable no matter what emails it's trying to label as spam or not (i.e., maybe even your emails).

```
precision average score: 0.8766395125343684
recall average score: 0.9229214208833865
f1 average score: 0.8987212860288842
```

The above shows the precision, recall, and f1 average scores for the spam detecting model trained and tested with different data (i.e., as shown above with the 9 heat maps). The precision being close to 1 (i.e., 1 would suggest a perfect model), shows that the model is very good at not over labeling good emails as spam emails; therefore, if applied to someone's email, they wouldn't find a lot of good emails labeled as spam. The recall is also close to 1 showing that the

model is good at finding all the spam; therefore, if applied to someone's email, they would get most of their spam put in the spam folder. The f1 score being close to 1 suggests that the model would be good at putting all spam in the spam folder and not many good emails in the spam folder.

Technical

In order to use the dataset, duplicates were removed to avoid redundancy. The spam label column was transformed into binary to be used for the model later. The email messages were transformed by making words lowercase and removing punctuation. From there, getting each of the individual words and then removing all stop words (e.g., "and", "is", "the", etc.). From there each of the words was reduced to its base form (e.g., "swimming" became "swim" and "cats" became "cat"). The modified words were then put into a matrix format to be fed into the classifier. The classifier chosen for the dataset was the multinomial naive bayes. This is because the model detected spam based on the frequency of words in each email. Since frequency was a discrete variable instead of a continuous one, multinomial was chosen; the variation of naive bayes classifier Gaussian looks at continuous variables which weren't used in the model. In the future it might be nice to account for words in all caps because that might have been another indicator of spam (i.e., we made all words lowercase in our model for simplicity sake).