ECE 470

Project Report
Filtering Spam Emails

Mustafa Wasif - V00890184

Aqeel Mozumder - V00884880

Table of Contents

List of Figures and Tables

Introduction

Problem Description and Motivation

What is the problem?

- To identify spam emails
- To reduce or stop legit emails residing in spam folder in the respective mailing application

Why is it important to solve?

- Can protect people's privacy by storing phishing emails with other spam emails
- Saves time from checking unwanted emails, such as irrelevant ads, hams
- Can reduce missing out on important emails if they end up in spam folder

Furthermore, several people fall for spam emails which makes them lose their personal and credentials information, and this results in fraudulent transactions, blackmailing, cyberbullying etc. [1] Therefore to reduce the occurrence of cyberattacks, it is important to identify and report spam emails.

Related Work

- Mail providers have a separate spam folder consisting of all spam emails to help check out important mails first.
- Mobile and online applications can detect spam emails from non-spam emails.
- Softwares like SpamFilter can filter out spam emails for an organization [2]
- Several Security tool offers to block spam emails by having a secure email gateway [3]
- Articles to educate people on how to identify spam emails. [4]

Problem Formulation

Problem search space is huge and worldwide as most people have email identifications. Many people have multiple email accounts. Having our email in public is quite easy as we have signed up for many applications and services through our email accounts. If privacy settings are not followed and regularly monitored, it is quite easy to fall prey to phishing from spam emails.

Programming Language

- Programming Language: Python version 3.7.3
- Libraries: sklearn, pandas,matplotlib.pyplot
- Dataset (CSV file): Spam Mail Dataset by Kaggle

Methodology

Evaluation Approach:

The proposed project uses Naive Bayes algorithm that identifies spam emails and non-spam emails.The algorithm reads through a dataset (.csv file) and identifies all the spam emails present in that particular dataset. The dataset that will be used is Spam Mails Dataset by Kaggle which is provided to us by the professor.

The Dataset has three columns which are label, text and label_num. The label_num columns data is either 0 or 1. 0 identifies ham/real email and 1 identifies as spam email. Before training, the dataset is divided into training and validation. We used the Sklearn library to split the data into training and testing data in the ratio of 80:20.

Our raw dataset is the email messages. We can not feed such raw datasets to machine learning algorithms. Machine learning algorithms train models by doing computation, and the computation is possible with numerical values. So, let’s extract features from the raw dataset for training. For doing that, we transform all the email messages to the vectorized form using the CountVectorizer class.

Then to evaluate the training data and testing data we use two classifiers to predict the accuracy.

1. Bernoulli Classifier
2. Multinomial Classifier

In addition, each classifier provides a confusion matrix, ROC accuracy and a classification report to show average f1-score, precision and recall.

A confusion matrix is provided to evaluate the classification models as it categorizes the outcome into two or more categories.

ROC accuracy is to judge the accuracy of default probability models and The classification report is about key metrics in a classification problem.

The recall means "how many of this class you find over the whole number of elements of this class

"The precision will be "how many are correctly classified among that class"

The f1-score is the harmonic mean between precision & recall.

Thus the criteria for the algorithm results will be to match the results of the manual solution.

Our evaluation criteria is as follows:

Evaluation Criteria	Expected Results
Is there any non-spam email detected as spam?	No
Are there any null values?	No
Were there any Imputation Techniques used?	Yes
What's the Spam % before splitting the data?	Less than 50%
Was the Dataset clean?	Yes
If spam is detected, What is the Bernoulli Accuracy?	Greater than 50%
If spam is detected, What is the Multinomial Accuracy?	Greater than 50%
Both Classifier Accuracy greater than 85%?	No
Did the code able to produce any graphs/charts/reports	Yes

Table 1: Evaluation Criteria

Evaluation Validation

As mentioned earlier, our approach was based on 2 classifiers to predict and experiment what type of accuracy we would get for different classifiers. The Multinomial ROC accuracy is 99.79% and the Bernoulli accuracy is 89.6%. The difference between the two classifiers is because the binomial distribution generalises the Bernoulli distribution across the number of trials, the multinoulli distribution generalises it across the number of outcomes.

Here is the final validation results from the Evaluation Criteria:

Evaluation Criteria	Expected Results	Actual Results
Is there any non-spam email detected as spam?	No	No
Are there any null values?	No	No
Were there any Imputation Techniques used?	Yes	Yes
What's the Spam % before splitting the data?	Less than 50%	29%
Was the Dataset clean?	Yes	No
If spam is detected, What is the Bernoulli Accuracy?	Greater than 50%	89.6%
If spam is detected, What is the Multinomial Accuracy?	Greater than 50%	98.4%
Both Classifier Accuracy greater than 85%?	No	Yes
Did the code able to produce any graphs/charts/reports	Yes	Yes

Table 2: Evaluation Criteria

Results & Discussion

Comparisons

```
PS C:\Users\aqeel\OneDrive\Desktop\ECE 470\Project_AI> python Spam_detector.py

Start Report:

Imputation Technique to check Null values:
Unnamed: 0    0
label         0
text          0
label_num     0
dtype: int64

Spam is  28.98859021465867 %

Bernoulli Confusion Matrix:  [[199 105]
 [  3 728]]

 Bernoulli Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.65      0.79       304
           1       0.87      1.00      0.93       731

    accuracy                           0.90      1035
   macro avg       0.93      0.83      0.86      1035
weighted avg       0.91      0.90      0.89      1035


Accuracy of Bernoulli:  89.56521739130436 %

Bernoulli ROC Accuracy:  89.68968248254014 %

Confusion Matrix:  [[293  11]
 [  6 725]]

 Multinomial Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.96      0.97       304
           1       0.99      0.99      0.99       731

    accuracy                           0.98      1035
   macro avg       0.98      0.98      0.98      1035
weighted avg       0.98      0.98      0.98      1035


Accuracy of Multinomial:  98.35748792270532 %

Multinomial ROC Accuracy:  99.78895168838649 %
```

Figure 1: Report

From the above report which is generated by the code, notice that the imputation technique was used to check if there are any null values. The report shows there are no null values and the spam value before splitting the data is 29%.
After the Data is Splitted:

The confusion matrix for Bernoulli classifier is :

199	105
3	728

The Bernoulli confusion matrix shows that out of the 304 instances of not spam, the classifier predicted correctly 199 of them
And out of 731 instances of spam, the classifier correctly predicted 728 of them. In total, the classifier predicted 927 out of 1035.

The confusion matrix for Multinomial classifier is :

293	11
6	725

The Multinomial confusion matrix shows that out of the 304 instances of not spam, the classifier predicted correctly 293 of them
And out of 731 instances of spam, the classifier correctly predicted 725 of them. In total, the classifier predicted 1018 out of 1035.

From the Bernoulli classification report we found that:

- Average Recall: 90%
- Average Precision: 91%
- Average F1-score: 89%

From the Multinomial classification report we found that:

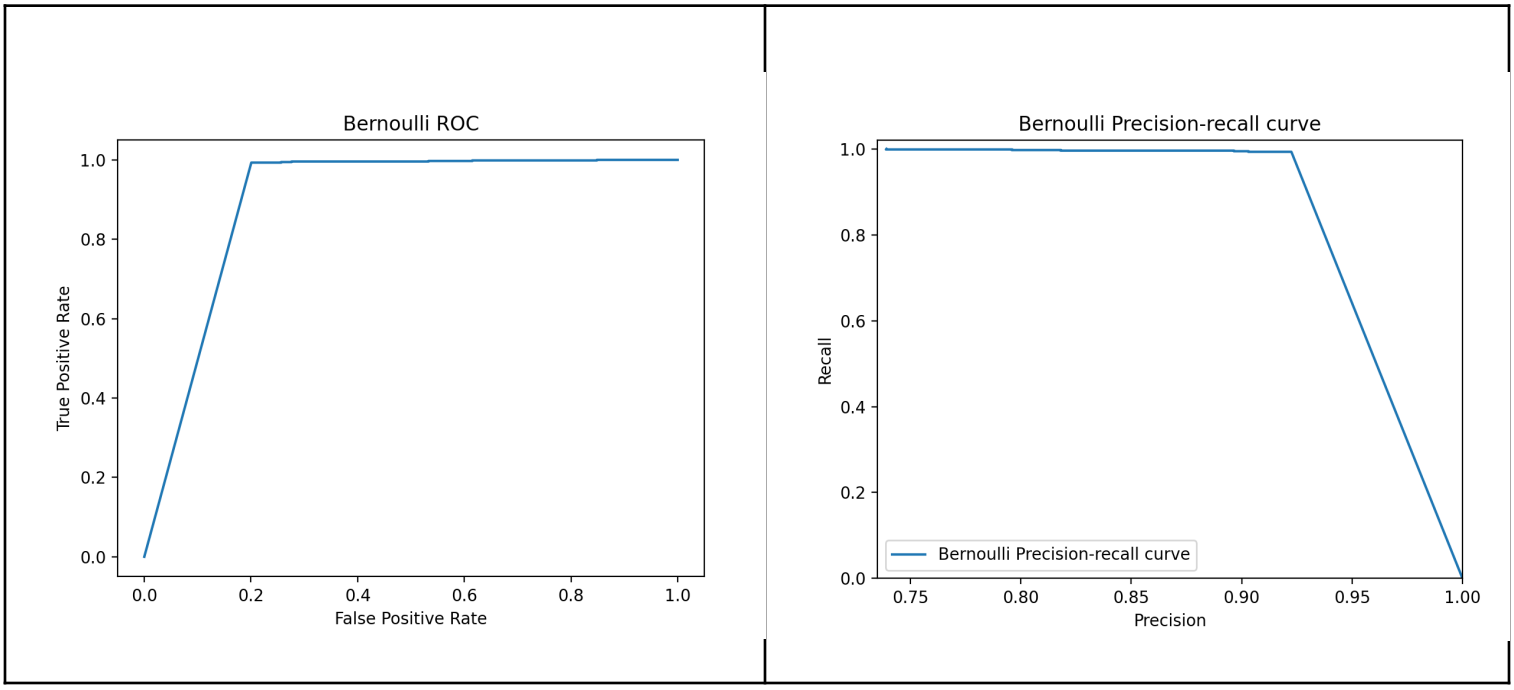
- Average Recall: 98%
- Average Precision: 98%
- Average F1-score: 98%

Overall:

- Multinomial prediction accuracy: 98.4%
- Bernoulli prediction accuracy: 89.6%
- Multinomial ROC accuracy: 99.8%
- Bernoulli ROC accuracy:89.7%

Graphs

Bernoulli Graph Analysis:



Graph 1: Bernoulli ROC & Bernoulli Precision-Recall curve

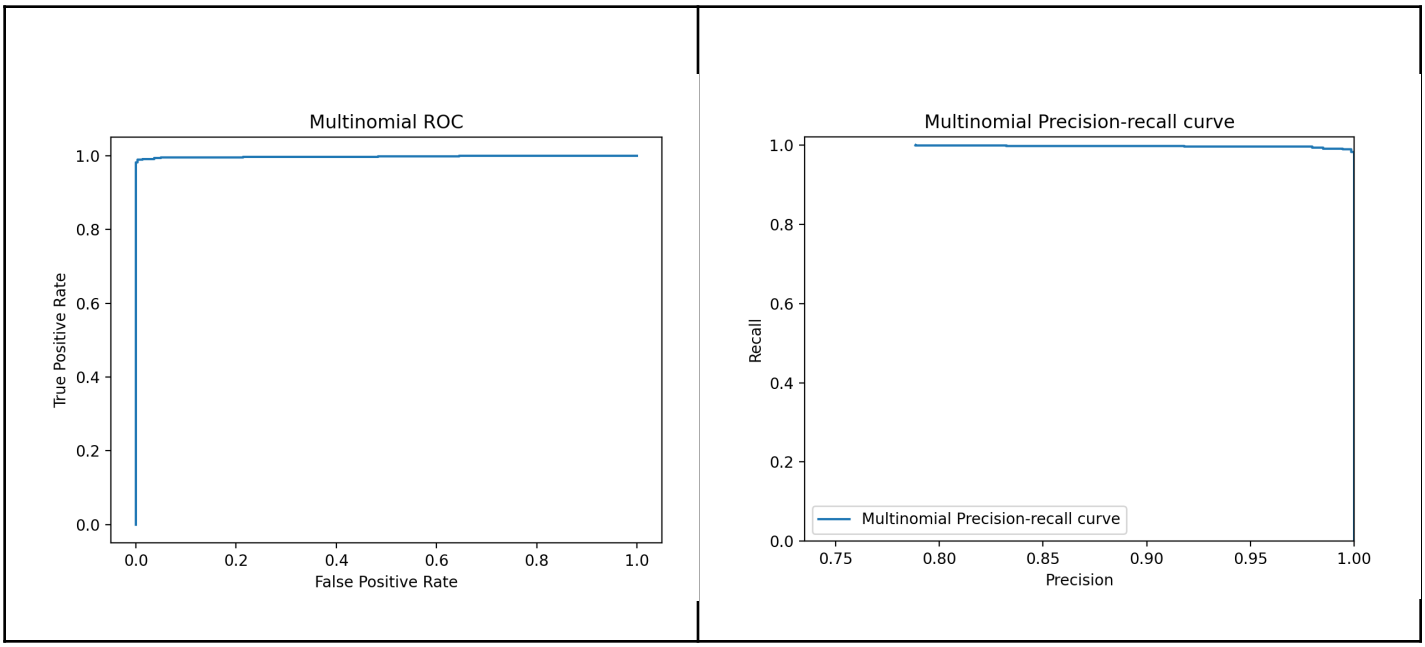
The ROC Graph shows:

- When False Positive Rate is 0 and True Positive rate is 0
  - No email is classified as Spam
- When False Positive Rate is 1 and True Positive rate is 1
  - No spam emails are detected as real emails
  - No real emails are classified as real emails
- The rest of the curve shows the results of the model as the curve approaches the corner of the plot
  - True positive rate is 1 and False positive rate is 0.2
  - Shows that some spam emails were classified as real
  - Shows that some real emails were classified as spam

The Precision - Recall Graph shows:

- The precision starts from roughly 0.7 as there are not many real emails classified as spam
- Precision is roughly 91%

Multinomial Graph Analysis:



Graph 2 : Multinomial ROC & Multinomial Precision-Recall curve

The ROC Graph shows:

- When False Positive Rate is 0 and True Positive rate is 0
  - No email is classified as Spam
- When False Positive Rate is 1 and True Positive rate is 1
  - No spam emails are detected as real emails
  - No real emails are classified as real emails
- The rest of the curve shows the results of the model as the curve approaches the corner of the plot
  - True positive rate is 1 and False positive rate is 0
  - Shows that no spam emails were classified as real
  - Shows that noreal emails were classified as spam

The Precision - Recall Graph shows:

- The precision starts from roughly 0.8 as there are not many real emails classified as spam
- Precision is roughly 98%

Justification

The results from the report and graphs justify that the algorithm can detect spam emails majority of the time. The reason Naive Bayes was chosen is because when assumption of independent predictors holds true, a Naive Bayes classifier performs better as compared to other models.

Moreover, as we had a small amount of data in the dataset, Naive Bayes requires a small amount of training data to estimate the test data. So, the training period is less.

Challenges

The most challenging problem was to clean the dataset. For example, removing punctuations, and other variables from the email content. This problem was not able to be solved due to compilation errors and therefore, we will think of a better solution in our future work. In addition, a lack of enough dataset that would meet our criteria was scarce.

From our research, only a few datasets were found that had both the subject and body fields for the emails. Furthermore, from those limited numbers of datasets, some were private, so those datasets could not be used to train the model.

Therefore, Naive Bayes algorithm was preferred as it works with minimum data and since only a limited datasets were available.

Conclusion & Future Work

The spam email filtration model is trained to distinguish between two types of email, spam and ham, by reading a unique dataset that requires both the subject and the body of the email be present.

This approach has led to better accuracy as the model can have more information about the context of the email, thereby making a better decision in determining the type of email.

In future, as more such datasets become public, our aim is to train our model with more and bigger datasets which can lead to better accuracy for spam email detection.
In addition, more investigation can be performed to clean the dataset to feed the model with easier to read data.

References:

[1] *Jack Schofield, (2019, January 17), I got a phishing email that tried to blackmail me - what should i do?, The Guardian.* *https://www.theguardian.com/technology/askjack/2019/jan/17/phishing-email-blackmail-sextortion-webcam*

[2] *SpamTitan, TitanHQ, spamtitan-anti-spam-solution-new (titanhq.com)*

[3] *Expert Insights, (2021, January 01), The Top Email Spam Filtering Solutions, Expert Insights* *https://expertinsights.com/insights/the-top-email-anti-spam-filtering-solutions/*

[4] *Scott Nelson, (2020, August 03), How to Recognize Spam, wikiHow, 3 Ways to Recognize Spam - wikiHow*