Assignment-2 Report

Course: DA5400
Name: Shishir Kulkarni
Roll No: ED21B037

---

1 Training Data

- The training data was a csv file downloaded from Kaggle
- The dataset contains 5172 emails with 3000 words out of which 1500 were spam and other ham.
- Each column has the frequency of a word in all the emails, and a row contains the frequency of all the words in a particular email.
- The last column labelled 'Prediction' contains information about the mail being spam or ham.
- The $[i][j]$ element in the Excel sheet has the frequency of the j^{th} word in the i^{th} email.

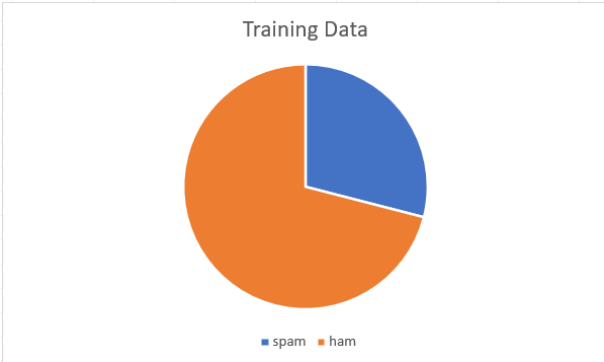

Figure 1: Visualisation of Training Data

2 Test Data

- The test data was a collection of .txt files downloaded from internet.
- The dataset contains 55 emails divided into 25 spam and 30 ham.

3 Algorithms

3.1 Naive-Bayes classifier

- In Naive Bayes algorithm, we use a probabilistic model based on Bayes' theorem, assuming that the presence of each word in an email is independent of the others. Despite this simplifying assumption, Naive Bayes has proven effective for spam classification due to its efficiency and good performance on high-dimensional data such as text.
- The algorithm classifies an email as spam or ham based on probabilities. The conditional probability of each word is given by: $\text{P(word} \mid \text{label)} = \frac{\text{Count(word in class)}+\alpha}{\text{Total words in class}+\alpha\times\text{Vocabulary size}}$

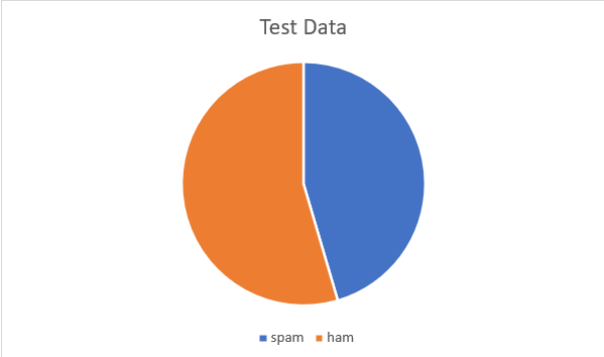

Figure 2: Visualisation of Test Data

- α is the Laplace smoothing parameter, which takes care of the cases when a word is not present in the train data but is present in the test data by assigning a small probability to it.
- The words from the training data were saved from being later used as a vocabulary for the test data while predicting individual word probabilities.
- The test data is a set of Txt files, so for extracting features from this, I have tokenised the text in the email and stored the frequencies of each word just like the training data. The algorithm then works the same on the test data as well.

3.2 Support Vector Machine

- SVM is a supervised learning algorithm that seeks to find an optimal hyperplane that separates the data points maximally by maximising the margin distance and keeping only the support vectors in the dataset.
- We have used a linear kernel which assigns a weight to each word frequency and finds the boundary that best classifies emails as spam as ham.
- A linear kernel is used because it is computationally efficient and also effective on text data.

4 Hyperparameter Tuning

4.1 Smoothing Parameter α

- Different values of Laplace smoothing parameter α gave different accuracy scores.
- Values in the range of 1 to 15 were tried for α which gave accuracy score in the ranges of 0.79 to 0.83 on the test data.
- The problem with high value of α is that it can cause underfitting due to excessive smoothing.
- Higher α value can help in the case of a smaller dataset.
- The value of alpha used in the code was 15

4.2 Vocabulary Size

- This is an indirect hyperparameter restricted by the dataset.
- The more the number of words in the vocabulary the better the model. In our case this parameter value is 3000

5 Model Evaluation and Results

5.1 Naive-Bayes classifier

- This model achieved an accuracy of 93.6% on the cross-validation data which had almost 29 % spam mails and 71 % ham emails.
- The model achieved accuracy of 83.6 % on the test dataset which had 42 % spam emails and 58 % ham emails.
- We can see that the accuracy dropped as the proportion of spam emails increased in the test data.
- This is because the train data has less number of spam emails, which makes prior biased towards ham, so the Naive Bayes classifier is biased towards predicting ham email frequently.
- This problem can be solved by training the model on less skewed data.

5.2 Support Vector Machine

- This model achieved an accuracy of 100% on the cross-validation data which had almost 29 % spam mails and 71 % ham emails.
- The model achieved accuracy of 70.9 % on the test dataset which had 42 % spam emails and 58 % ham emails.
- We can see that the accuracy dropped as the proportion of spam emails increased in the test data.
- This is because the train data has less number of spam emails, which makes prior biased towards ham, so the Naive Bayes classifier is biased towards predicting ham email frequently.
- This problem can be solved by training the model on less skewed data.

6 Conclusion

Upon comparing the accuracy scores of SVM and Naive Bayes algorithms, we can conclude that Naive Bayes is a more suitable algorithm for email spam classification