CS5691: Pattern Recognition and Machine Learning

Assignment 3: Spam Classifier

Shrivarshan K, MM20B058

Dataset:

The dataset that is being used for training our algorithms is the Enron-Spam dataset, which comprises six Enron files, each containing around 5000-6000 emails of both spam and ham. We extracted the emails directly from the zip file. The zip files used for training have been provided in the submission file. The enron1 and enron2 datasets are used for training, and the trained algorithm is tested on the enron4 dataset.

Preprocessing:

Since all the datasets are in tar.gz format, they are all unzipped, and a dataframe is created using them. A "message" column plus a "class" column make up the dataframe. Each email's raw data is contained in the "message" column, and the "class" column corresponds to the classification (spam/ham). Now that each entry in the "message" column must be cleaned, a cleaning mechanism was added. The mechanism used for cleaning is as follows:

- All the spam/ham labels to a binary encoding, where 1 represents a spam email and 0 for a ham email.
- As the mails are in bytes, they are converted to a string by a decoding function.
- AL the words in the mail are converted to lowercase.
- Punctuation and escape characters such as '\r\n' present in the mails are removed.
- We substitute "http\https" with "url" and similarly, we substitute $ and ₹ with dollars and rupees.
- The string with '@' in it are considered as email address and are substituted with "emailid".
- Any numbers present in the emails are removed.
- We also implemented a lemmatization using the nltk library's WordNetLemmatizer(). This is a standard procedure in NLP (Natural Language Processing), which is done to convert all the different inflected forms of a word into the same base/root form of the word.

An example of a mail before and after preprocessing is shown below:

Before Preprocessing:

```
'Subject: meter 7268 nov allocation\r\nfyi .\r\n- - - - - - - - - - - - - - - - - - - - - - - - forwarded by lauri a allen / hou /
ect on 12 / 14 / 99 12 : 17\r\npm - - - - - - - - - - - - - - - - - - - - - - - - -\r\nkimberly vaughn\r\n12 / 10 / 99 02 :
54 pm\r\nto : lauri a allen / hou / ect @ ect\r\ncc : mary m smith / hou / ect @ ect\r\nsubject : meter 7268 nov allocation\r\n
lauri . . i have put this on strangas gas until i can get a contract from\r\ndaren .\r\n- - - - - - - - - - - - - - - - - - - -
- - forwarded by kimberly vaughn / hou / ect on 12 / 10 / 99 01 : 52\r\npm - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-\r\nlauri a allen\r\n12 / 09 / 99 01 : 20 pm\r\nto : kimberly vaughn / hou / ect @ ect , anita luong / hou / ect @ ect\r\ncc :
howard b camp / hou / ect @ ect , mary m smith / hou / ect @ ect\r\nsubject : meter 7268 nov allocation\r\nkim / anita -\r\na v
olume of 7247 mm shows to have been allocated to the reliant 201 contract\r\nfor november . there was no nomination for reliant
at this point in november\r\nand , therefore , there should be no volume allocated to their contract .\r\nplease make sure thes
e volumes are moved off the reliant contract prior to\r\nnovember close .\r\nthanks .'
```

After preprocessing:

```
'subject meter nov allocation fyi forwarded by lauri a allen hou ect on pm kimberly vaughn pm to lauri a allen hou ect ect cc m
ary m smith hou ect ect subject meter nov allocation lauri i have put this on strangas gas until i can get a contract from dare
n forwarded by kimberly vaughn hou ect on pm lauri a allen pm to kimberly vaughn hou ect ect anita luong hou ect ect cc howard
b camp hou ect ect mary m smith hou ect ect subject meter nov allocation kim anita a volume of mm show to have been allocated t
o the reliant contract for november there wa no nomination for reliant at this point in november and therefore there should be
no volume allocated to their contract please make sure these volume are moved off the reliant contract prior to november close
thanks'
```

Feature Extraction:

We must transform the preprocessed emails into an input format our system can utilize for training after obtaining them. As a train set, the enron and, enron2 have been employed. Every email will be converted into a vector count of certain dictionary terms. The frequency of each word in each class will essentially serve as our feature, which will be represented as a vector. We employed the Sklearn library's CountVectorizer() for this.

We do another processing technique which is the removal of stopwords. Stopwords are words that do not really affect the class probability. Words like the, and, if, etc, are very common words present almost everywhere. So we don't want these words to affect our classification.

The stopwords are found using an IDF (Inverse Document Frequency) measure. It measures how frequently the word is in a group of emails. The lower the value, the more times that word has appeared in our emails. So we removed the top 40 stopwords from being used in our dictionary. They are:

['subject', 'to', 'the', 'for', 'and', 'you', 'of', 'on', 'is', 'in', 'this', 'have', 'be', 'with', 'your', 'we', 'from', 'that', 'please', 'will', 'at', 'are', 'if', 'it', 'by', 'enron', 'as', 'me', 'not', 'or', 'am', 'cc', 'can', '2000', 'thanks', 'our', 're', 'com', 'any', 'know']

We now compute a frequency vector for the top 4000 words in the dictionary using the CountVectorizer function. This list has been updated to remove the stopwords. Our input data will be the feature vector for each email in our set. Hence Each row of the matrix 11029*4000 in X_train

represents one email, and each column represents the frequency of a term. The vector y_train has 11029 dimensions. After comparison, it was determined that the dictionary was 4000 words long.

Classifiers/Algorithms used:

1. Naive-Bayes Classifier:

This algorithm uses the Bayes rules to classify an email. Here A is the class, and B is the feature vector of an email. So to implement this, we need to estimate the parameters on the RHS.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

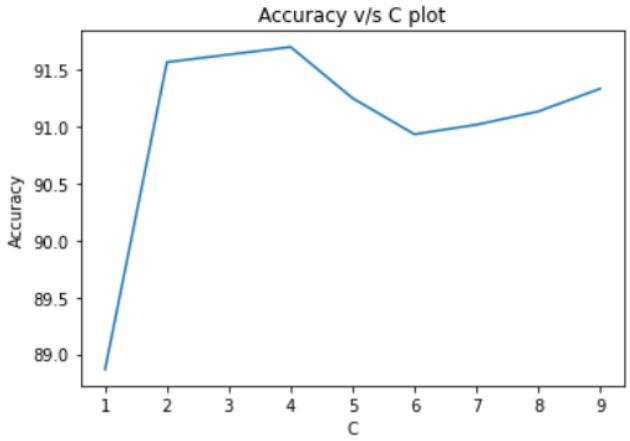
P(y = 1) = Fraction of the total number of spam mail
Probability of a word present in a spam mail:
(Number of times the word appeared in spam mail) / (Total number of spam mails)
Probability of a word present in a ham mail:
(Number of times the word appeared in ham) / (Total number of ham mails)

Next, we calculate the probability's Maximum Likelihood function for that class. We multiply all of the probabilities because we believe that each word is independent. We total the Log of probabilities rather than calculating the final probability for each email. The log value is then compared for whatever class it is greater and assigned to that class.

Laplace smoothing was applied to the training set after I manually created a feature vector that contained only 1s. This ensures that if a word appears in test data but not in the train set, the probability does not fall to 0. This smoothing resulted in noticeably superior outcomes.

2. Support Vector Machine (SVM):

We also trained a SVM model from sklearn directly. Hyperparameter tuning was performed by changing the C value from [1, 10] using grid search method. The highest accuracy for train set was obtained for C = 4.



Results:

We calculate the accuracy (a measure of performance) for each of the classifiers. Each of the models is trained with enron1 and enron2 dataset and the performance is tested on the enron 4 dataset. The accuracies are as follows:

Classifier used	Accuracy
Naive Bayes (developed from scratch)	94.88333333333333
BernoulliNB (imported from sklearn)	96.46666666666667
Support Vector Machines (from sklearn)	91.7

So the Naive Bayes appears to be performing well than SVM imported from sklearn.