CS5691: Pattern Recognition and Machine Learning
Programming Assignment 3
REPORT

Name: Chetan Reddy N
Roll No: ME19B093

Overview:

- Spam email classification is a popular machine learning problem statement.
- After data collection and thorough data analysis, the appropriate features are extracted.
- The algorithm used is a soft margin SVM classifier which achieves a test accuracy of about 97% after rigorous hyperparameter tuning.

Dataset:

- Numerous spam-ham datasets are available on the internet on open source websites like Kaggle and AWS Datasets.

- The dataset used is a combination of SMS messages (spam and ham), emails (spam and ham) and a few custom mails. The custom mails are derived from **smail** and a few emails which were written by me for professional purposes over the past two years.

- A **custom test dataset has been prepared with about 60 emails** (about 50 non-spam and 10 spam). The trained model is tested on this set for better-curated analysis. The training dataset and the custom test dataset have been included in the submission folder.

- The dataset (custom test dataset not included) has **5583 emails** of which 770 emails are spam and 4813 are non-spam emails.

Feature Extraction:

- The available data is text but a machine learning algorithm can only process numerical data. Therefore, each email should be vectorised and numerous methods are available to achieve this.

- Firstly, the number of stopwords is counted in each email. Stopwords are commonly used English words (like “the”, “a”, “an”, “I”) which may not hinder the vectorising process later. However, they are counted and the **number of stopwords** is stored as a feature before dropping them from each email.

- Secondly, the **number of contractions** is counted in each email. Contractions include words like I’m, we’ll, you’d etc. The contractions are split into words.

Contraction	Meaning
'aight	alright
ain't	is not
amn't	am not
aren't	are not
can't	cannot

- Finally, the number of special characters and the number of numerical characters in each email is counted and stored as a feature.

- The following statistic shows the **relevance of the features number_of_special_characters and number_of_numerical_characters** because they are clearly higher in spam emails and can be a useful feature

category	number_of_contractions	number_of_stopwords	number_of_special_characters	number_of_numerical_characters
0	0.312279	5.529192	1.023894	0.260544
1	0.072727	6.584416	3.110390	4.338961

N-gram Analysis

- N-gram analysis is done to look at the most common words, most common two-words, most common triple-words and so on in spam and non-spam emails.

```
Commonly used trigrams in Spam Emails
you have won              50
prize guaranteed call     21
1000 cash or              19
find out who              18
from land line            18
urgent your mobile        18
account statement for     16
to contact you            16
valid 12hrs only          16
selected to receive       15
```

Vectorisation

- Vectorisation is the process of converting a text into a vector for an ML algorithm to process. The vectorisation technique used is called **TF-IDF** (Term Frequency-Inverse Document Frequency)

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

i - word/feature
j - email
$tf_{i,j}$ - Number of Occurance of i in j
Df_i - Number of Documents containing i
N - Total Number of Documents

- The dimension of the vector is the top few words with the highest relevance for classification. It is a hyperparameter. The value used is 1000 (after performing hyperparameter tuning)

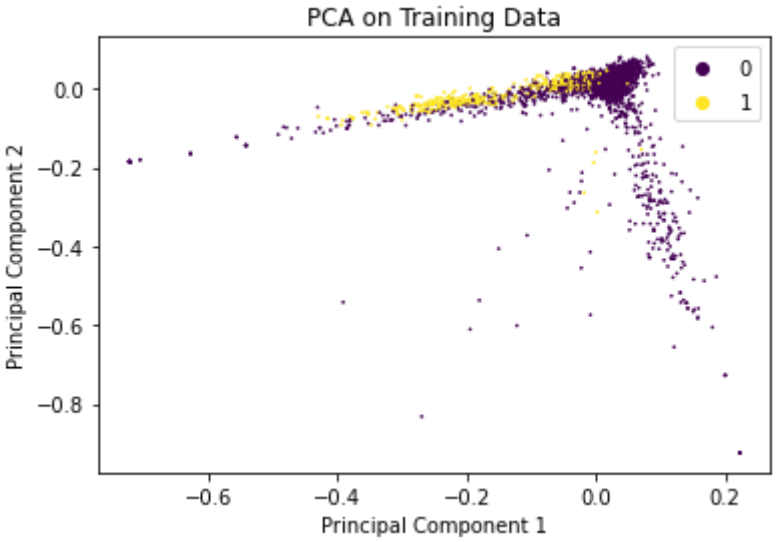
- The vectorisation is performed after processing the sentence:

  1. Converting it to lowercase
  2. Removing Stopwords
  3. Expanding Contractions
  4. Removing Special Characters
  5. Lemmatising (returning the base or dictionary form of a word)

- The features finally used are the output of the tf-idf vectoriser and the features listed above i.e number_of_special_characters, number_of_numerical_characters. The number_of_stopwords and number_of_contractions weren't used because of the lack of correlation with the label. The dimension of the vector is 1000+2 = 1002 i.e $\mathbf{x_i} \in \mathbf{R^{1002}}$ **and** $\mathbf{y_i} \in \mathbf{\{1,0\}}$

PCA

- PCA was performed on the 1002 dimensional data and the top two principal components (corresponding to the highest two eigenvalues of the covariance matrix) are visualised



- It looks promising as the data seems separable (atleast in the higher dimension)

Model

- The model used is a soft-margin SVM.
- The library sklearn is used to implement the model.
- The training matrix $X \in \mathbf{R}^{4187x1002}$ and most of the elements being zero makes it a sparse matrix which makes the training process slower. Therefore, the numpy matrix is converted in scipy's csr matrix which speeds up the training process of the SVM.

Hyperparameter Tuning:

- **Kernel:**
  The kernels tested are **'linear' and 'rbf' and 'poly'**.
- **C:**
  The regularisation parameter is chosen on a logarithm scale and tested on multiple values. A very high value brings it closer to hard-margin SVM
- **Gamma:**
  It is a coefficient in the kernel function of rbf and poly.

- After a rigorous hyperparameter tuning (by nested iterations or grid search), the following hyperparameters yielded the best test set accuracy:
  {Kernel: **"rbf"**, C: **3**, Gamma: '**auto**' or 1/1002 (n_features)}

Testing the Model on email#.txt

- The last section can be run to automatically read txt files and classify as spam/non-spam, it generates a vector of 0's and 1's corresponding to each email.
- Please note that the last cell assumes a trained model in memory, therefore all the cells above it should be executed before running this cell.
- Further, the files **custom_test_data.csv, english_contractions.csv, final_dataset.csv** and the folder **test** should be in the present working folder
  -- **final_dataset.csv**
  -- english_contractions.csv
  -- custom_test_data.csv
  -- test
  -- email1.txt
  -- email2.txt
  -- …
  -- email#.txt

Results

- For the hyperparameters mentioned above, the soft-margin SVM gives the following results
- Training Set Accuracy: `0.99`
- Test Set Accuracy: `0.97`
- Accuracy on the Custom Prepared Dataset: `0.96`
- Category of email1.txt : `0 (non-spam)`
- Category of email2.txt : `1 (spam)`