Naive Bayes Classifier

Mudit Chaturvedi	Ashna Swaika	Rohan Sachan
2018A7PS0248H	2018A7PS0027H	2018B3A70992H

1) Model Description and Implementation:

We have created a Bernoulli Event Model based Naive Bayes classifier. The model has been implemented in python from scratch:

1. The dataset is read line by line into a python list and further tokenized to extract out the label of the email for that particular line.
2. Next we split each line into words and remove any punctuation marks, symbols, numbers etc.
3. Further we have removed certain stop words from the word list so as to have only those words in our database which are more important for understanding the sentiment or the context of the email.
4. This final '**data**' is fed to the Naive Bayes classifier implemented with a 7 fold cross-validation. The number of folds for the cross validation can be modified as per requirement.
5. '*k_fold_train_test(data,nfolds)*' - This is the main function which randomly allots elements to different folds '*k_fold_split(data,nfolds)*'. Out of the 7 folds formed, 6 are used for forming the 'trainset' and one is used as 'testset'. Each fold is chosen to be a testset one time each i.e. the process of training and prediction repeats over 7 times with different combinations of folds.
6. '*spam_classifier_fit(trainset)*' - This function is called to train the model based on the training data set.
    a. Finds P(Spam) and P(NonSpam)
    b. Finds the unique words in the complete dataset, using the function call to '*get_unique_words(data)*'
    c. Finds P(Word|Spam), P(Word|NonSpam) along with using laplace smoothening and stores them in a dictionary. Uses the functions - '*prob_word_in_spam(word,data)*' and '*prob_word_in_nonspam(word,data)*'
7. '*spam_classifier_predict(testset)*' - This function is used to predict the spam/non-spam classification for the model generated in the previous step.
    a. Finds P(Spam|Word) and P(NonSpam|Word) for different words of the email.
    b. Finds P(Spam|word1,word2,word3) for a specific email.
    c. It returns a list of predicted values based on whether the above probability is >0.5 or not.
8. '*check_accuracy(testset,predicted)*' - This function is used to find the accuracy of the model by comparing the predicted values with the actual values.
9. The main function repeats over 7 times, calculating the accuracy over each fold each fold and then calculates the average accuracy.

2) Accuracy of the model over each fold & overall average accuracy:

```
#Calls the main Naive Bayes k-cross-validation training and testing function, with num_of_folds=7
k_fold_train_test(data,7)
```

- The model received the following accuracies over each fold:

```
Accuracy of the model over fold no. 1 = 0.8391608391608392
Accuracy of the model over fold no. 2 = 0.8251748251748252
Accuracy of the model over fold no. 3 = 0.8251748251748252
Accuracy of the model over fold no. 4 = 0.8111888111888111
Accuracy of the model over fold no. 5 = 0.7902097902097902
Accuracy of the model over fold no. 6 = 0.8251748251748252
Accuracy of the model over fold no. 7 = 0.795774647887324
```

- Overall average accuracy of the model:

```
Average accuracy of the model = 0.8159797948530343
```

3) Major limitations of the Naive Bayes classifier:

- It assumes that the predictor features are independent of each other. This can lead to highly imprecise probabilities. In practical applications, the assumption of independent predictors may not hold true.
- For cases of discrete variables, a new value in the test set can lead to the complete posterior probability becoming zero. Although the issue can be solved by ignoring the new words while predicting, or by making a model which maps any new word to a special token 'UNK', but for cases where a lot of new values are observed while prediction, then it can lead to imprecise probabilities.
- In case of discrete variables, the Naive Bayes classifier does not consider the frequency with which the items occur while calculating posterior probabilities.
- It loses the temporal structure of the data because the formula for calculating probability is independent of the sequence. This can hinder the performance of the model especially in case of Natural Language Processing tasks.