Spam Detection Using Naïve Bayes

– Mohammad Mirzanjad

Description of program steps

1- **loadInputFiles:** first sms_spamcsv and stopwordscsv files are read and thenthe dataset is shuffled 80 percent of the dataset is allocated to trainingData and the rest to testData.
2- **removeSpaceFromStopWords:** To match the program method, spaces at the beginning of StopWords are removed.
3- **prepareDataFormat**: This function is executed for both training data and test data The purpose of this function is to break the words in each training example into its words and put the words in a vector so that stopwords and numbers are deleted Also words are converted to all lowercase letters.
4- **removeStopWords**: This function removes the words in thestopwords file, numbers, symbols, and single-letter words
5- **generateVocabulary**: Creates a vector consisting of unique words in the training data of trainingData.
6- **generateVocabularyForClass**: This function has two features:vocabForSpam and vocabForHam. It makes ham and spam for two classes In this way, for the example of ,education belonging to a class, if the words of that example are not available in the vector add them and enter the number 1 in the second column (the column corresponding to the frequency of each word) and if the word is available, the corresponding value in the column it adds one to the second. ,Finally, for all the words in the examples of a class their frequency is obtained and can be extracted and used from the second column of this matrix.
7- **calculatePrior**: The probability ofPrior for each class is obtained by dividing the number of labeled samples with the name of that class by the total number of training samples
8- **calculateNumberOfWordsInClass**: Calculates the total number of words in a class.
9- **calculateProbsOfWordsOfVocab:** Calculates the probability of each word in each class and puts the matrices related to the words in each class (vocabForSpam and vocabForHam) in the fourth column. The method of calculating the probability according to Bayes' law and Laplace's estimation is equal to the number of repetitions of the word+ 1 divided by the total size of the dictionary containing the words and the total number of words in the desired class
10- **testForPredict**: For each test data, this function calculates the probability of each word using the calculateWordProb function for both ham and spam classes and multiplies the

resulting probabilities for each class, and finally multiplies it by the Prior probability value of the corresponding class. The larger probability will determine the type of class for the test data This application is performed on all the test data, and in the meantime, the values related to theConfusion matrix.

11- **calculateWordProb**: Searches the word in vocabForHam for the ham class and in vocabForSpam for the spam class and extracts its probability values from the fourth column If the desired value is not present in those feature vectors and according to Laplace estimation r eturns the value of 1 divided by the size of the dictionary containing the total words + the total number of words in the desired class

At the end, the accuracy of the algorithm is reported according to the number of correct detections of the test data class. Due to shuffling the data at the beginning of the algorithm , the accuracy is somewhat different each time the program is executed and varies from 96.3 to 98.78 percent. It is worth noting that the number of test data is 1115 i.e. 20% ofthe data .

The confusion matrix for96.68 accuracy ٪ rate is as follows

	Ham	spam
Ham	919	16
Spam	21	159

True Positive 919

The number of samples that were ham and the program recognized them as ham

True negative 159

The number of samples that were spam and the program recognized them as spam

False Positive 16

The number of samples that were spam and the program recognized them as ham

False Negative 21

The number of samples that were ham and the program recognized them as spam