Assignment 2

Naive Bayes Classifier Implementation on dataset

Sri Harsha Ganji - 200050041

Part A)

—----------

Preprocessing :

—-------------------------------

I have used the glob function in python which recursively takes in all the training files separated into respective classes and makes a list of all file names in sorted order.
After that I have made corresponding vocabulary and document vectors for each of the following models (i.e Binary Bag of Words(bbow) and Count Bag of Words(cbow))

For this I used a library called CountVectorizer which returns a vector of length vocabulary of the training set for each document passed as a list of words. This vector returned for each document is of length vocab and the ith index contains frequency of the ith element in the vocabulary in the document.

Since the data is really sparse we have to use only the non zero values of each doc_vec and store it in a json file as a dictionary of dictionaries with respective format.

Just some processing of these doc_vecs, replacing all elements greater than1 with 1 will give us BBoW json file.
Used json.dump to dump the respective dictionaries into respective json files.

Script to be run : script.py
Files Generated : bbow_vecs.json , cbow_vecs.json

Part B)

—----------

Naive Bayes Classification - Multinomial :

—----------

In this we have implemented the naive bayes classification from scratch by creating custom functions.

Main Implementation:

—----------

First we calculate the laplace_smoothinvg function of a word belonging to a class . Then we multiply the probabilities of all words in the test file belonging to the particular class and we do this for all the classes . Then we find the maximum of these probabilities generated and produce the label according to the index.

Note ::
This process does not take time if we are classifying a long list of files. We are storing the values of certain word and label pairs to be reused again.

There is this preprocessing which has to be done to the test input file. Basically in this preprocessing we take the input file path read the file append the words into a list cross reference with the vocab of training set and assign word_labels and then process them according to the functions defined above

Script to be run	:	bah.py
Files Generated	:	None
Input	:	Absolute path of the file
HyperParameter	:	k = 1

Naive Bayes Classification - Poisson :

—----------

Here the framework remains the same just we replace the laplace_smoothing function with an equivalent of the poisson laplace_smoothing implemented as in the research paper