REPORT - NLP (CSCE-689, Programming Assignment #1 SpamLord)
Name - Navneet Gupta
UIN - 226000691

1. Compile and Execution

It is developed with python 2.7 . The original training data and results are in the zip itself, along with the source code.

Steps -

1. Unzip the file.
2. Cd into that folder and then cd into python.
3. python NaiveBayes.py ../data/imdb1/ (Use python 2.7) (-f, -b flags supported)
4. python Perceptron.py ../data/imdb1/ 10
5. Output will get printed on the terminal

2. Results and Analysis

Results -

Q1. Naive Bayes Classification -

Output -

[INFO] Fold 0 Accuracy: 0.765000
[INFO] Fold 1 Accuracy: 0.850000
[INFO] Fold 2 Accuracy: 0.835000
[INFO] Fold 3 Accuracy: 0.825000
[INFO] Fold 4 Accuracy: 0.815000
[INFO] Fold 5 Accuracy: 0.820000
[INFO] Fold 6 Accuracy: 0.835000
[INFO] Fold 7 Accuracy: 0.825000
[INFO] Fold 8 Accuracy: 0.755000
[INFO] Fold 9 Accuracy: 0.840000
[INFO] Accuracy: 0.816500

Analysis - This version of Naive Bayes gives the most accuracy.

Q2. Stop Words Removed.

Output -

[INFO] Fold 0 Accuracy: 0.765000
[INFO] Fold 1 Accuracy: 0.825000
[INFO] Fold 2 Accuracy: 0.815000
[INFO] Fold 3 Accuracy: 0.830000
[INFO] Fold 4 Accuracy: 0.795000
[INFO] Fold 5 Accuracy: 0.830000
[INFO] Fold 6 Accuracy: 0.835000
[INFO] Fold 7 Accuracy: 0.835000

[INFO] Fold 8 Accuracy: 0.760000
[INFO] Fold 9 Accuracy: 0.820000
[INFO] Accuracy: 0.811000

Analysis - Here, the accuracy gets reduced as compared to the previous version. This may be due to the fact that the stop words were adding to the classification substantially in some documents. I believe this will not happen always.

Q3. Binary Naive Bayes Model.
Output -
[INFO] Fold 0 Accuracy: 0.690000
[INFO] Fold 1 Accuracy: 0.670000
[INFO] Fold 2 Accuracy: 0.755000
[INFO] Fold 3 Accuracy: 0.735000
[INFO] Fold 4 Accuracy: 0.695000
[INFO] Fold 5 Accuracy: 0.735000
[INFO] Fold 6 Accuracy: 0.755000
[INFO] Fold 7 Accuracy: 0.700000
[INFO] Fold 8 Accuracy: 0.730000
[INFO] Fold 9 Accuracy: 0.715000
[INFO] Accuracy: 0.718000

Analysis - Binary Naive bayes models performs badly in comparison due to the simplification in the model assumptions.

Q4. Perceptron Algorithm
Output -
1 iteration -
[INFO] Fold 0 Accuracy: 0.510000
[INFO] Fold 1 Accuracy: 0.485000
[INFO] Fold 2 Accuracy: 0.465000
[INFO] Fold 3 Accuracy: 0.480000
[INFO] Fold 4 Accuracy: 0.510000
[INFO] Fold 5 Accuracy: 0.490000
[INFO] Fold 6 Accuracy: 0.530000
[INFO] Fold 7 Accuracy: 0.470000
[INFO] Fold 8 Accuracy: 0.500000
[INFO] Fold 9 Accuracy: 0.510000
[INFO] Accuracy: 0.495000

10 iterations -
[INFO] Fold 0 Accuracy: 0.595000
[INFO] Fold 1 Accuracy: 0.575000

[INFO] Fold 2 Accuracy: 0.690000
[INFO] Fold 3 Accuracy: 0.615000
[INFO] Fold 4 Accuracy: 0.655000
[INFO] Fold 5 Accuracy: 0.580000
[INFO] Fold 6 Accuracy: 0.650000
[INFO] Fold 7 Accuracy: 0.655000
[INFO] Fold 8 Accuracy: 0.660000
[INFO] Fold 9 Accuracy: 0.690000
[INFO] Accuracy: 0.636500

50 iterations -
[INFO] Fold 0 Accuracy: 0.770000
[INFO] Fold 1 Accuracy: 0.770000
[INFO] Fold 2 Accuracy: 0.805000
[INFO] Fold 3 Accuracy: 0.760000
[INFO] Fold 4 Accuracy: 0.770000
[INFO] Fold 5 Accuracy: 0.775000
[INFO] Fold 6 Accuracy: 0.785000
[INFO] Fold 7 Accuracy: 0.795000
[INFO] Fold 8 Accuracy: 0.760000
[INFO] Fold 9 Accuracy: 0.780000
[INFO] Accuracy: 0.777000

100 iterations -
[INFO] Fold 0 Accuracy: 0.805000
[INFO] Fold 1 Accuracy: 0.815000
[INFO] Fold 2 Accuracy: 0.840000
[INFO] Fold 3 Accuracy: 0.810000
[INFO] Fold 4 Accuracy: 0.810000
[INFO] Fold 5 Accuracy: 0.790000
[INFO] Fold 6 Accuracy: 0.860000
[INFO] Fold 7 Accuracy: 0.850000
[INFO] Fold 8 Accuracy: 0.805000
[INFO] Fold 9 Accuracy: 0.870000
[INFO] Accuracy: 0.825500

1000 iterations -
[INFO] Fold 0 Accuracy: 0.810000
[INFO] Fold 1 Accuracy: 0.845000
[INFO] Fold 2 Accuracy: 0.845000
[INFO] Fold 3 Accuracy: 0.840000
[INFO] Fold 4 Accuracy: 0.820000
[INFO] Fold 5 Accuracy: 0.835000

[INFO] Fold 6 Accuracy: 0.880000
[INFO] Fold 7 Accuracy: 0.855000
[INFO] Fold 8 Accuracy: 0.830000
[INFO] Fold 9 Accuracy: 0.885000
[INFO] Accuracy: 0.844500

Analysis -
In case of Perceptron, we try to align weights according to the training we perform. Here, we can see from the results that as we increase the number of iterations, its efficiency increases.

3. Problems and Limitations -
Naive Bayes is based on bag of words model. So, it doesn't classify correctly in some cases. Moreover, in binomial version of Naive Bayes model, we calculate probability in terms of number of documents, which may be super simplified assumption in some of the documents.

In case of Perceptron, it takes around 15-20 minutes for 1000 iterations. For all other cases, it calculates in a timely manner.