NLP Programming Assignment #2 - Sentiment Analyzer.

Submitted by – Vaibhav Rawat

UIN – 626008171

**Steps to compile and run the code with Python 2.7:**

1. Open the terminal
2. Go to the folder NLPAssign2/python.
3. Enter the following commands:

python NaiveBayes.py ../data/imdb1

python NaiveBayes.py -f ../data/imdb1

python NaiveBayes.py -b ../data/imdb1

# for testing with other test set

python NaiveBayes.py ../data/imdb1 ../data/test\_data

**Results and Analysis:**

Following results are seen on running the program –

Vaibhavs-MacBook-Pro:python vrawat279$ python NaiveBayes.py ../data/imdb1

[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

Vaibhavs-MacBook-Pro:python vrawat279$ python NaiveBayes.py -f ../data/imdb1

[INFO] Fold 0 Accuracy: 0.760000

[INFO] Fold 1 Accuracy: 0.825000

[INFO] Fold 2 Accuracy: 0.825000

[INFO] Fold 3 Accuracy: 0.830000

[INFO] Fold 4 Accuracy: 0.800000

[INFO] Fold 5 Accuracy: 0.830000

[INFO] Fold 6 Accuracy: 0.830000

[INFO] Fold 7 Accuracy: 0.835000

[INFO] Fold 8 Accuracy: 0.755000

[INFO] Fold 9 Accuracy: 0.820000

[INFO] Accuracy: 0.811000

Vaibhavs-MacBook-Pro:python vrawat279$ python NaiveBayes.py -b ../data/imdb1

[INFO] Fold 0 Accuracy: 0.805000

[INFO] Fold 1 Accuracy: 0.840000

[INFO] Fold 2 Accuracy: 0.835000

[INFO] Fold 3 Accuracy: 0.825000

[INFO] Fold 4 Accuracy: 0.835000

[INFO] Fold 5 Accuracy: 0.825000

[INFO] Fold 6 Accuracy: 0.845000

[INFO] Fold 7 Accuracy: 0.835000

[INFO] Fold 8 Accuracy: 0.790000

[INFO] Fold 9 Accuracy: 0.855000

[INFO] Accuracy: 0.829000

To sum up:

Average Accuracy (with stop words): 0.8165

Average Accuracy (without stop words): 0.8110

Average Accuracy (with stop words): 0.8290

Binarized version of Naïve Bayes which relies on occurrence of word in a document rather than frequency, performs slightly better than general Naïve Bayes.

Removal of stop words doesn’t help in improving the accuracy.

**Other relevant features for Sentiment Analysis:**

Bigrams and Trigrams feature combined with Unigrams feature can help learn the context of the text. These features are not applied completely independent of bag of word features. They can be applied after removing the stops words. Maybe in that case they can help in better prediction for positive and negative reviews, since a lot of irrelevant word combinations formed by including the stop words are avoided and so scores are assigned to documents based on word combinations relevant to the context. These features can be tried with different machine learning methods, to see if some of them can benefit from the captured context.

Features can be identified which indicate whether sentences are on topic. New features can be developed such as average line polarity, average first line polarity etc. based on such analysis to represent sentence level polarity. Such features would be really helpful for reviews having statements with mixed sentiments, and rhetorical statements.

Other features such as POS-tagging, higher weights assigned to negation and elongated words (like goooooood), and exclamation marks, coupled with above features can further provide slight help to realize the correct sentiment of the review.

Any known bugs, problems, or limitations of your program

There are no known bugs.