INFO 590
Big Data Applications and Analytics
Supervised Learning and Word Level Sentimental Analysis

Joydeep Ghatak - jghatak@indiana.edu
Hari Gudigundla - hgudigun@indiana.edu
Anudhriti Katanguri - anukatan@indiana.edu

December 18, 2015

Based on the Course: Big Data Applications and Analytics by Prof. Geoffrey Fox

1. **Objective**
   The main objective of the project is to train a Nave-Bayes algorithm with train data and perform the sentimental analysis for each word on the training data and predict the relative score of how the polarity of the word by applying the supervised machine learning techniques during training and testing the data.

2. **Dataset Description**
   The data provided along with the NLTK package was used. The dataset consisted of movie reviews which were segregated as positive movie reviews and negative movie reviews.
   The dataset is named as movie_reviews.
   The location is C : \Users \<user_name>\AppData\Roaming\nltk _data\corpora\movie_reviews.
   This dataset contains two types of data positive movie reviews and negative movie reviews.
   Each of these positive reviews and negative reviews folder contains 1000 files. Each file contains text data ranging from 1KB to 12 KB.

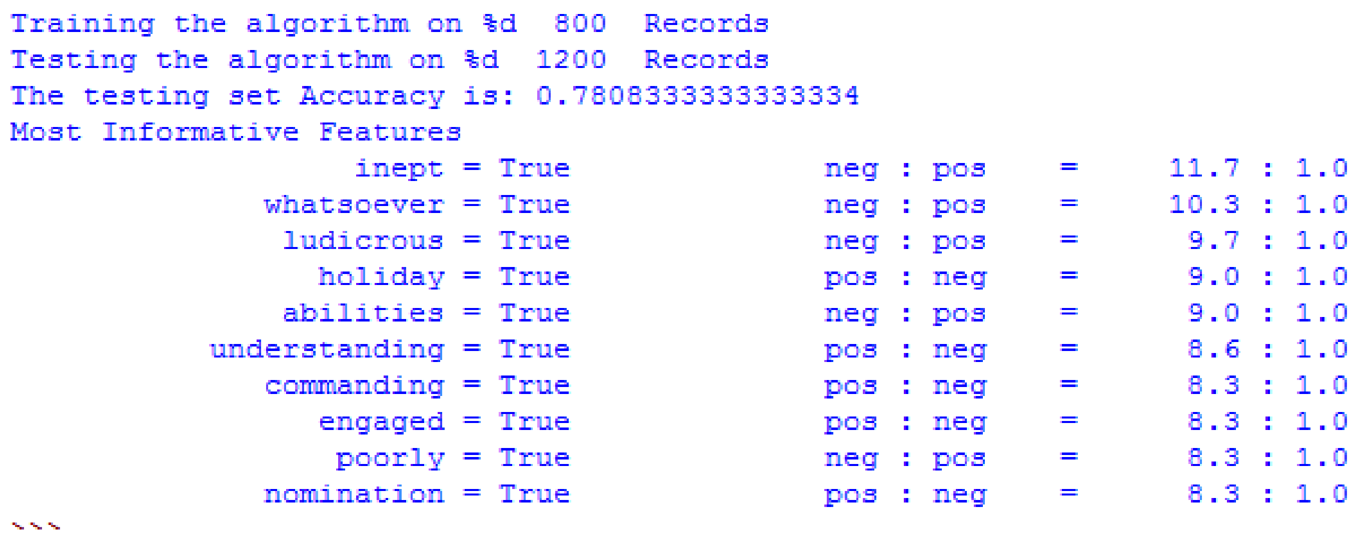
3. **Data Loading**
   The data which was used for the sentimental analysis came along with the NLTK package. It gets installed in the local machine, data is imported within the program using the from nltk.corpus import movie_reviews command.

4. **Data Preprocessing**
   No such pre-processing is required with the data as the data is present within the installed NLTK package. Importing the data within the code directly worked successfully. But within the code, as there are several text files (1000 positive text files and 1000 negative text files), the words has been tokenized to manipulate the data files for either positive or negative tag along with the file.

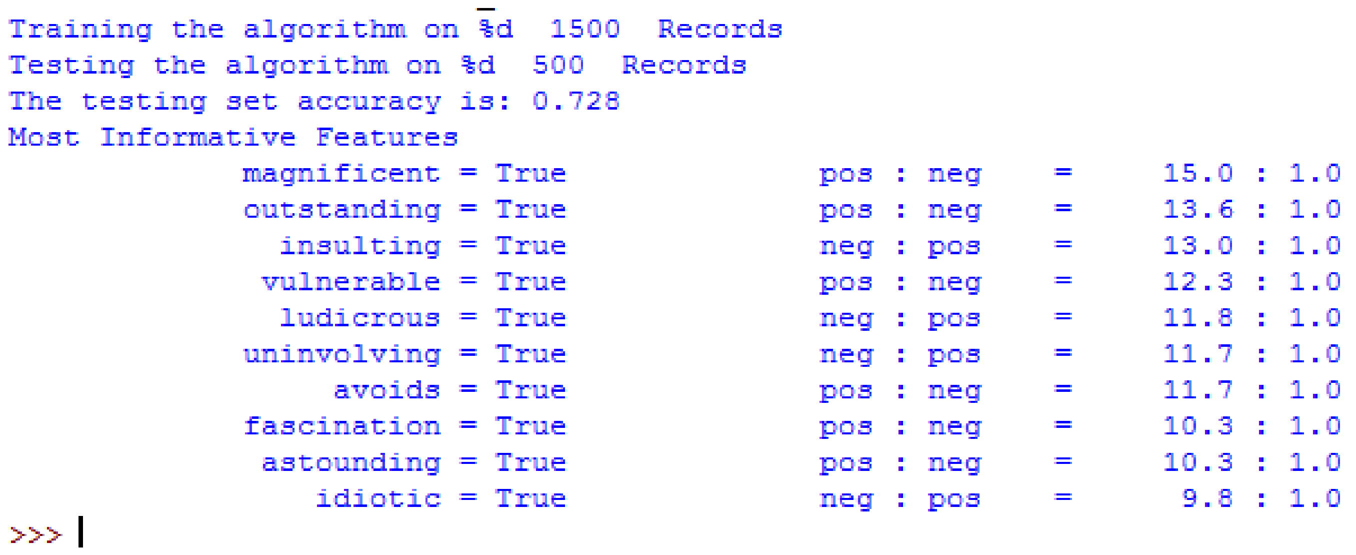
5. **Sentiment Analysis Tool**
   The sentiment analysis tool used for this project is NLTK toolkit which takes the raw input and provides the value by analyzing the input provided. It provides a negative value and a positive value for each word. If the negative value greater than positive value it indicates that word has negative sentiment while if positive value dominates the negative value it indicates positive sentiment. The application is created using python 3.5, in windows environment, which is compatible with NLTK 3.0 package and its functionality. Current code is written using Python 3.5 which has different syntax from python 2.X and hence will not be compatible with 2.x environment.
   Changing the number of negative values and number of positive values within the program, the amount of training data and amount of testing data can vary. Based on the number of training and testing data, the accuracy of the final output varies as seen from the following section.

6. **Results**
   The result of this application seems satisfactory as accuracy of the output seems similar to the human accuracy approximately 68 percent to 81 percent. The results may be a bit biased as already present and pre-processed files are used to train and test the data, but the overall performance by changing the train and test percentage, revolves within human tolerance level. The final output varies based on the number of training and testing set data as shown by the screen shots in Results section.

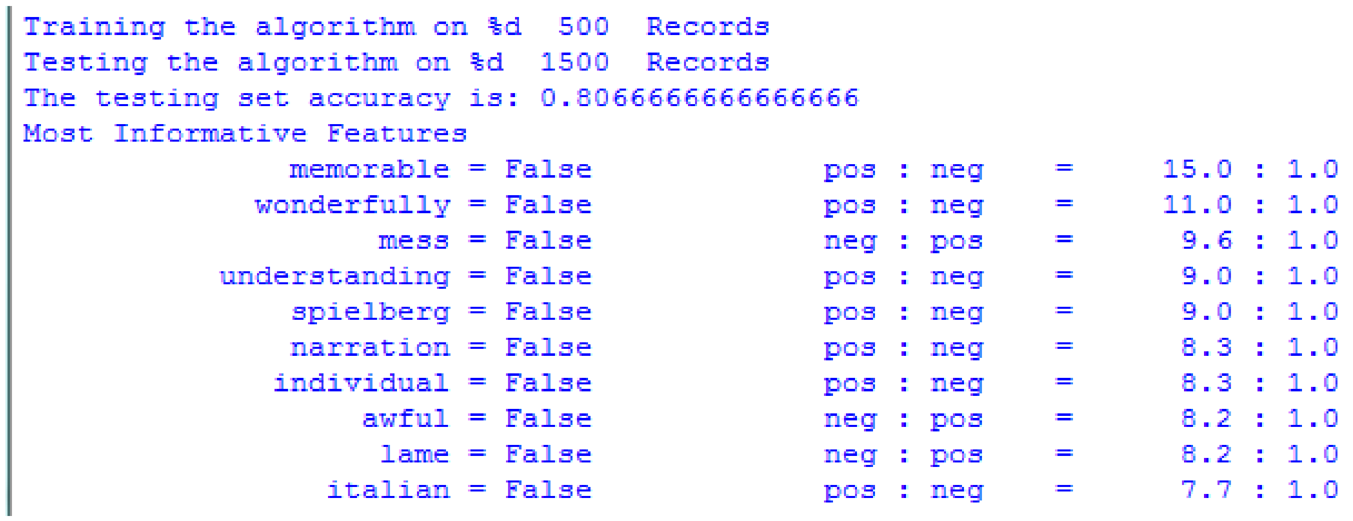
Output Screenshots



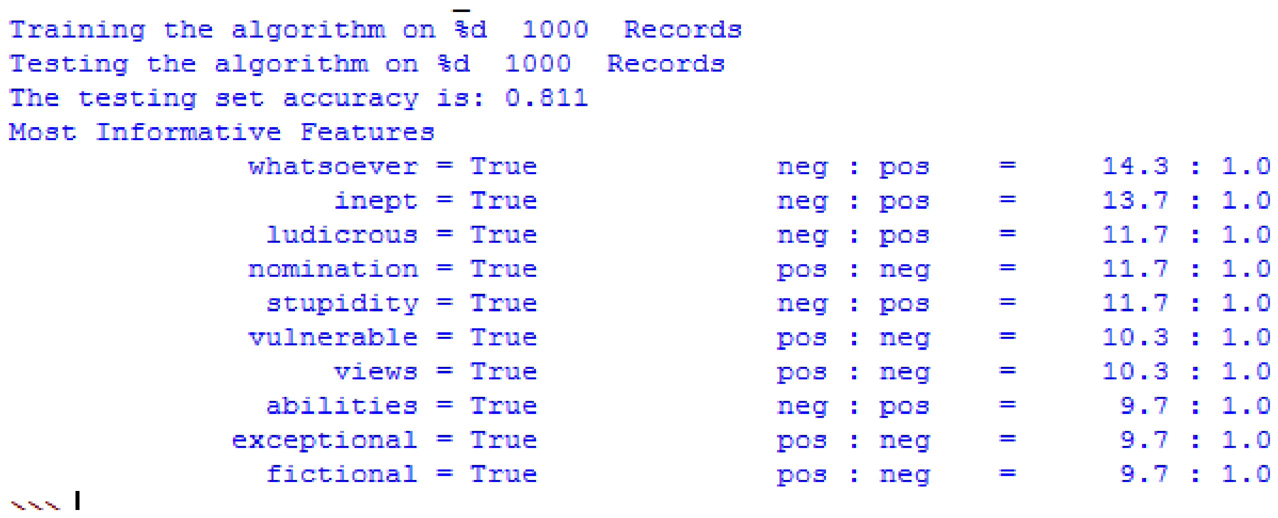
~~~
Training the algorithm on %d  800  Records
Testing the algorithm on %d  1200  Records
The testing set Accuracy is: 0.7808333333333334
Most Informative Features
                   inept = True              neg : pos    =     11.7 : 1.0
              whatsoever = True              neg : pos    =     10.3 : 1.0
               ludicrous = True              neg : pos    =      9.7 : 1.0
                 holiday = True              pos : neg    =      9.0 : 1.0
               abilities = True              neg : pos    =      9.0 : 1.0
           understanding = True              pos : neg    =      8.6 : 1.0
              commanding = True              pos : neg    =      8.3 : 1.0
                 engaged = True              pos : neg    =      8.3 : 1.0
                  poorly = True              neg : pos    =      8.3 : 1.0
              nomination = True              pos : neg    =      8.3 : 1.0
```
~~~



~~~
Training the algorithm on %d  1500  Records
Testing the algorithm on %d  500  Records
The testing set accuracy is: 0.728
Most Informative Features
             magnificent = True              pos : neg    =     15.0 : 1.0
             outstanding = True              pos : neg    =     13.6 : 1.0
               insulting = True              neg : pos    =     13.0 : 1.0
              vulnerable = True              pos : neg    =     12.3 : 1.0
               ludicrous = True              neg : pos    =     11.8 : 1.0
             uninvolving = True              neg : pos    =     11.7 : 1.0
                  avoids = True              pos : neg    =     11.7 : 1.0
             fascination = True              pos : neg    =     10.3 : 1.0
              astounding = True              pos : neg    =     10.3 : 1.0
                 idiotic = True              neg : pos    =      9.8 : 1.0
>>> |
~~~



~~~
Training the algorithm on %d  500  Records
Testing the algorithm on %d  1500  Records
The testing set accuracy is: 0.8066666666666666
Most Informative Features
              memorable = False             pos : neg    =     15.0 : 1.0
            wonderfully = False             pos : neg    =     11.0 : 1.0
                   mess = False             neg : pos    =      9.6 : 1.0
          understanding = False             pos : neg    =      9.0 : 1.0
              spielberg = False             pos : neg    =      9.0 : 1.0
              narration = False             pos : neg    =      8.3 : 1.0
             individual = False             pos : neg    =      8.3 : 1.0
                  awful = False             neg : pos    =      8.2 : 1.0
                   lame = False             neg : pos    =      8.2 : 1.0
                italian = False             pos : neg    =      7.7 : 1.0
~~~



```
Training the algorithm on %d  1000  Records
Testing the algorithm on %d  1000  Records
The testing set accuracy is: 0.811
Most Informative Features
             whatsoever = True              neg : pos    =     14.3 : 1.0
                  inept = True              neg : pos    =     13.7 : 1.0
              ludicrous = True              neg : pos    =     11.7 : 1.0
             nomination = True              pos : neg    =     11.7 : 1.0
              stupidity = True              neg : pos    =     11.7 : 1.0
             vulnerable = True              pos : neg    =     10.3 : 1.0
                  views = True              pos : neg    =     10.3 : 1.0
              abilities = True              neg : pos    =      9.7 : 1.0
            exceptional = True              pos : neg    =      9.7 : 1.0
              fictional = True              pos : neg    =      9.7 : 1.0
```

7. **Improvements**
The application outputs the sentiments for single word (positive connotation or negative connotation) by learning the words from positive and negative files. In the next version, the application can be improved to,

1. Analyze sentence level sentiments.
2. Able to test sentiments with static Twitter data.
3. Able to test live twitter stream using Twitter API.
4. Group similar sentiments for similar topic within a certain geographic location and which can be visually shown.
5. Should be able to take any word file and generate the output with proper accuracy.
6. Other algorithms can be used to verify the output accuracy.

8. **Inferences** The algorithm used is Nave-Bayes algorithm and imported data has been fed as a training-set and when tested on test set, it seems the results are satisfactory in nature. Different datasets of similar nature needs to be used to reduce the data bias if any, otherwise the output looks good. Also the variation of the number of training and test data shows variations in the output accuracy, which signifies that proper choice of training-testing data combination has effect on the overall algorithm output.

9. **Technologies Used**
1. Operating System: Windows 7
2. Natural Language Toolkit (NLTK) 3.0
3. Python 3.5.1 software
4. Python 3.5 IDLE as IDE
5. Numpy and Scipy has also been installed as part of the project.

10. **Citation**
http : //www.cs.cornell.edu/people/pabo/movie − review − data/
http : //scikit − learn.org/stable/tutorial/text_analytics/working_with_text_data.html
http : //www.lfd.uci.edu/ gohlke/pythonlibs/

11. **References**

1) *https://www.python.org/*
2) *http://www.nltk.org/*
3)INFO590 Big Data Applications and Analytics by Prof. Geoffery Fox:
http://openedx.scholargrid.org/courses/SoIC/INFO590/FALL_2015/courseware/84a9548d8aaf41d89
4) *https://docs.python.org/3.5/tutorial/index.html*
5) *https://docs.python.org/3.5/index.html*