INFO-I 590 BIG DATA APPLICATIONS AND ANALYTICS

PROJECT REPORT ON
MOVIE AND PRODUCT REVIEWS

Submitted by

Arpit Agarwal

(arpiagar@indiana.edu)

&

Raghuveer Raavi

(rraavi@indiana.edu)

ACKNOWLEDGEMENTS

We would like to thank our professor Dr. Geoffrey Charles Fox for giving us this opportunity and his support throughout the course and during the development of the project.

We would also like to thank our Associate instructor Kamalkanth Chowdary Adusumilli for his support and suggestions throughout the course.

MOVIE AND PRODUCT REVIEWS

Introduction:

The goal of the project is to do Sentiment Analysis and Trend Analysis on the reviews written by the users for movie/product. We intend to do category based analysis (on the basis of ratings given by user) on products and movies reviews text. We plan to use field “review/text” to do the sentiment analysis from data and mapping the corresponding rating given with the review itself.

We also want to find most useful features in the review text. After this we also do the trend analysis of certain famous products and movies. This is plotting popularity/rating (that is number of positive and negative reviews for the corresponding year) of certain products/movies over time. We intend to use dataset from SNAP repository.

Dataset Source:

Web Data: Amazon Movie Reviews from SNAP (Stanford Network Analysis Project) repository

http://snap.stanford.edu/data/web-Movies.html

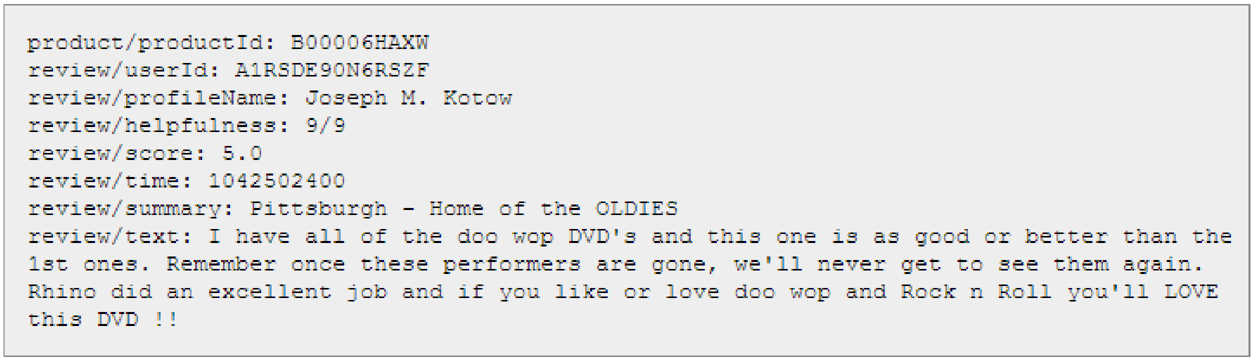
This dataset consists of movie reviews from amazon. The data span a period of more than 10 years, including all ~8 million reviews up to October 2012. Reviews include product and user information, ratings, and a plaintext review. We also have reviews from all other Amazon categories.

Dataset Statistics:

Number of reviews	:	7,911,684
Number of users	:	889,176
Number of products	:	253,059
Users with > 50 reviews	:	16,341
Median no. of words per review	:	101
Timespan	:	Aug 1997 - Oct 2012

Data format:

The format for the Dataset is as follows



```
product/productId: B00006HAXW
review/userId: A1RSDE90N6RSZF
review/profileName: Joseph M. Kotow
review/helpfulness: 9/9
review/score: 5.0
review/time: 1042502400
review/summary: Pittsburgh - Home of the OLDIES
review/text: I have all of the doo wop DVD's and this one is as good or better than the
1st ones. Remember once these performers are gone, we'll never get to see them again.
Rhino did an excellent job and if you like or love doo wop and Rock n Roll you'll LOVE
this DVD !!
```

It has 8 attributes in total and they are:

- product/productId : asin, e.g. amazon.com/dp/B00006HAXW
- review/userId : id of the user, e.g. A1RSDE90N6RSZF
- review/profileName : name of the user
- review/helpfulness : fraction of users who found the review helpful
- review/score : rating of the product
- review/time : time of the review (unix time)
- review/summary : review summary
- review/text : text of the review

Since the dataset has ~8 million reviews, the size of the file was close to 8GB to be approx. and there was no editor we could use to open the dataset. So we split up source into smaller pieces of size of 50MB each, and these files were used to extract the main required fields of the each review again.

Softwares/Languages/API's used:

Python 2.7, MSWord, matplotlib, NLTK and Numpy, gSplit.

API description:

- **Matplotlib:** It is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB. SciPy makes use of matplotlib.

- **Natural Language Toolkit [NLTK]:** The Natural Language Toolkit, or more commonly NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

NLTK is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for the Python programming language. NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit plus a cookbook. NLTK is intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems.

Naïve Bayes Classifier provided by NLTK:

Naïve Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness and diameter features. For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

- **Numpy:** NumPy is an open source extension module for Python. The module NumPy provides fast precompiled functions for numerical routines. It adds support to Python for multi-dimensional arrays and matrices. The implementation is aiming at huge matrices and arrays. Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays. NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy address the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring (re)writing some code, mostly inner loops using NumPy. Thus any algorithm that can be expressed primarily as operations on arrays and matrices can run almost as quickly as the equivalent C code.

Steps followed in achieving the Goals:

Step – I: Cleaning and extracting the data:

The dataset we have is of size 8.9GB containing close to 8 million reviews approximately. Our systems cannot handle such large datasets and process them. So, initially we split up the Dataset into smaller pieces of size 50MB size each using gSplit software, which produces 179 smaller pieces. We then systematically search for the movies/products that have the highest reviews among all the 8 million reviews. This is done by searching productID field line by line and building up a dictionary for it with the frequency. We displayed most frequent 20 fields (Output Screen – I).

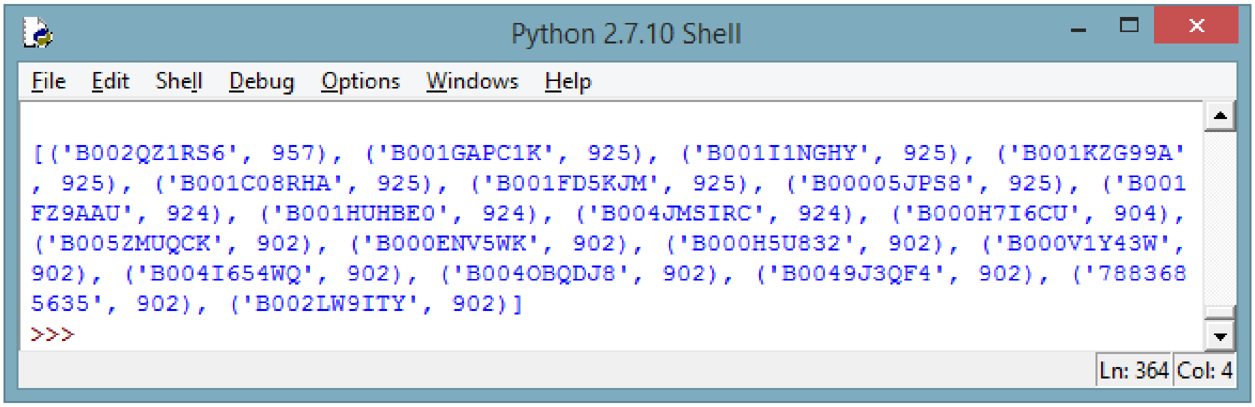
Now, that we have found out the titles/items with highest reviews among all others, we extract the data pertaining to those titles/reviews from the dataset into separate files. We do this by searching for those specific titles/items throughout the dataset and copying only those items that have the productID field into new text files.

The code for this process is in generateData.py file and the output are new separate text files.

Insights from Data Extraction:

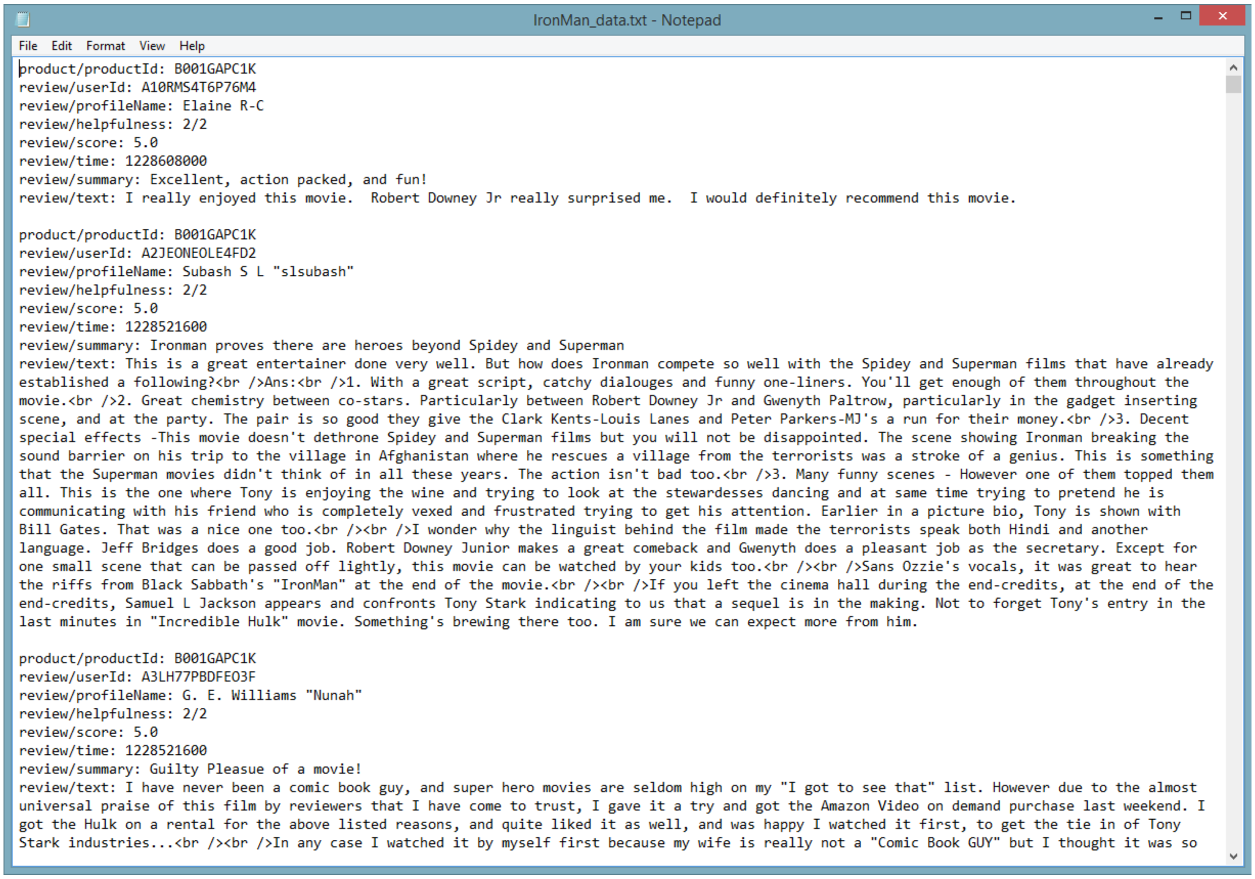
- After we split the source dataset into smaller pieces, we notice that there are some <br/> statements within the review field that needs to be cleaned
- We also notice that even though our dataset contains 8 million reviews approximately, we do not have much data for individual movie/products, that is 957 is highest amount of reviews we have on a single product.
- We can also check that the productID of a title/product, although different, they have the same reviews. For example, enter the productID B001GAPC1K and B001I1NGHY in amazon website. We can see that both the productID are for same movie title i.e. Iron Man containing same reviews, although different productID. So there is a lot of redundant and repetitive data present in the source data set.

Output Screens:



```
[('B002QZ1RS6', 957), ('B001GAPC1K', 925), ('B001I1NGHY', 925), ('B001KZG99A'
, 925), ('B001C08RHA', 925), ('B001FD5KJM', 925), ('B00005JPS8', 925), ('B001
FZ9AAU', 924), ('B001HUHBE0', 924), ('B004JMSIRC', 924), ('B000H7I6CU', 904),
('B005ZMUQCK', 902), ('B000ENV5WK', 902), ('B000H5U832', 902), ('B000V1Y43W',
902), ('B004I654WQ', 902), ('B004OBQDJ8', 902), ('B0049J3QF4', 902), ('788368
5635', 902), ('B002LW9ITY', 902)]
>>>
```

Output Screen - I: Snip showing dictionary of most common products with their corresponding ID fields and their respective number of reviews.



```
product/productId: B001GAPC1K
review/userId: A10RMS4T6P76M4
review/profileName: Elaine R-C
review/helpfulness: 2/2
review/score: 5.0
review/time: 1228608000
review/summary: Excellent, action packed, and fun!
review/text: I really enjoyed this movie.  Robert Downey Jr really surprised me.  I would definitely recommend this movie.

product/productId: B001GAPC1K
review/userId: A2JEONEOLE4FD2
review/profileName: Subash S L "slsubash"
review/helpfulness: 2/2
review/score: 5.0
review/time: 1228521600
review/summary: Ironman proves there are heroes beyond Spidey and Superman
review/text: This is a great entertainer done very well. But how does Ironman compete so well with the Spidey and Superman films that have already
established a following?<br />Ans:<br />1. With a great script, catchy dialouges and funny one-liners. You'll get enough of them throughout the
movie.<br />2. Great chemistry between co-stars. Particularly between Robert Downey Jr and Gwenyth Paltrow, particularly in the gadget inserting
scene, and at the party. The pair is so good they give the Clark Kents-Louis Lanes and Peter Parkers-MJ's a run for their money.<br />3. Decent
special effects -This movie doesn't dethrone Spidey and Superman films but you will not be disappointed. The scene showing Ironman breaking the
sound barrier on his trip to the village in Afghanistan where he rescues a village from the terrorists was a stroke of a genius. This is something
that the Superman movies didn't think of in all these years. The action isn't bad too.<br />3. Many funny scenes - However one of them topped them
all. This is the one where Tony is enjoying the wine and trying to look at the stewardesses dancing and at same time trying to pretend he is
communicating with his friend who is completely vexed and frustrated trying to get his attention. Earlier in a picture bio, Tony is shown with
Bill Gates. That was a nice one too.<br /><br />I wonder why the linguist behind the film made the terrorists speak both Hindi and another
language. Jeff Bridges does a good job. Robert Downey Junior makes a great comeback and Gwenyth does a pleasant job as the secretary. Except for
one small scene that can be passed off lightly, this movie can be watched by your kids too.<br /><br />Sans Ozzie's vocals, it was great to hear
the riffs from Black Sabbath's "IronMan" at the end of the movie.<br /><br />If you left the cinema hall during the end-credits, at the end of the
end-credits, Samuel L Jackson appears and confronts Tony Stark indicating to us that a sequel is in the making. Not to forget Tony's entry in the
last minutes in "Incredible Hulk" movie. Something's brewing there too. I am sure we can expect more from him.

product/productId: B001GAPC1K
review/userId: A3LH77PBDFEO3F
review/profileName: G. E. Williams "Nunah"
review/helpfulness: 2/2
review/score: 5.0
review/time: 1228521600
review/summary: Guilty Pleasue of a movie!
review/text: I have never been a comic book guy, and super hero movies are seldom high on my "I got to see that" list. However due to the almost
universal praise of this film by reviewers that I have come to trust, I gave it a try and got the Amazon Video on demand purchase last weekend. I
got the Hulk on a rental for the above listed reasons, and quite liked it as well, and was happy I watched it first, to get the tie in of Tony
Stark industries...<br /><br />In any case I watched it by myself first because my wife is really not a "Comic Book GUY" but I thought it was so
```

Output Screen – II: Snip showing extracted data of specific productID from the original movie dataset.

Step – II: Sentiment Analysis using Naïve Bayes Classifier provided by NLTK:

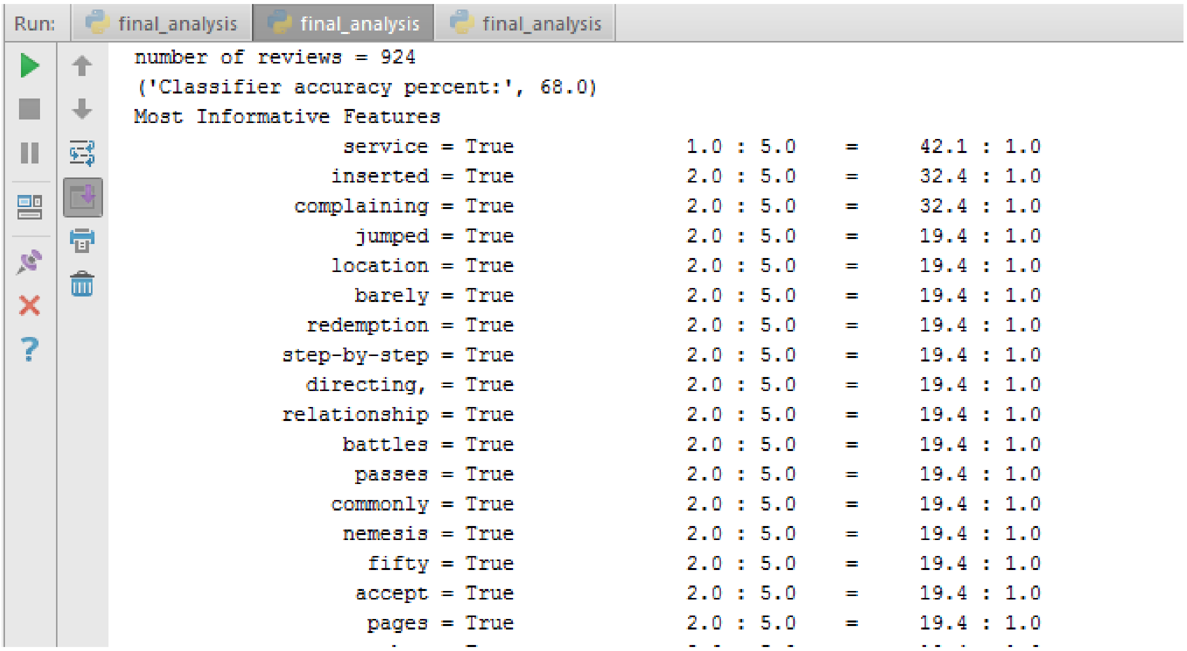
For sentiment analysis, we trained our classifier with the review text data and corresponding rating of the review. We are using 90% of the reviews for training and 10% for testing the classifier. We then got the accuracy ranging from 64- 75% approx. on the basis of dataset. We also found most informative features of the review text, for each review/score category ranging from 1.0 to 5.0 (with 1.0 being least favorite and 5.0 being most favorite). Refer to the output screens attached below.

Insights:

- If we are using 80% of the data for training and 20% of data for testing, the accuracy of the classifier falls below 55%, the reason for this is that we split the data into 5 categories which makes very less data in each category for classifier to learn.
- There are certain most commonly used words like ‘Good’ that are present in reviews of all category ranges. For example, consider a review with rating category 1.0 that has text as ‘not good’ and hence reduces the accuracy of classifier. Since we used unigrams in our bag of words

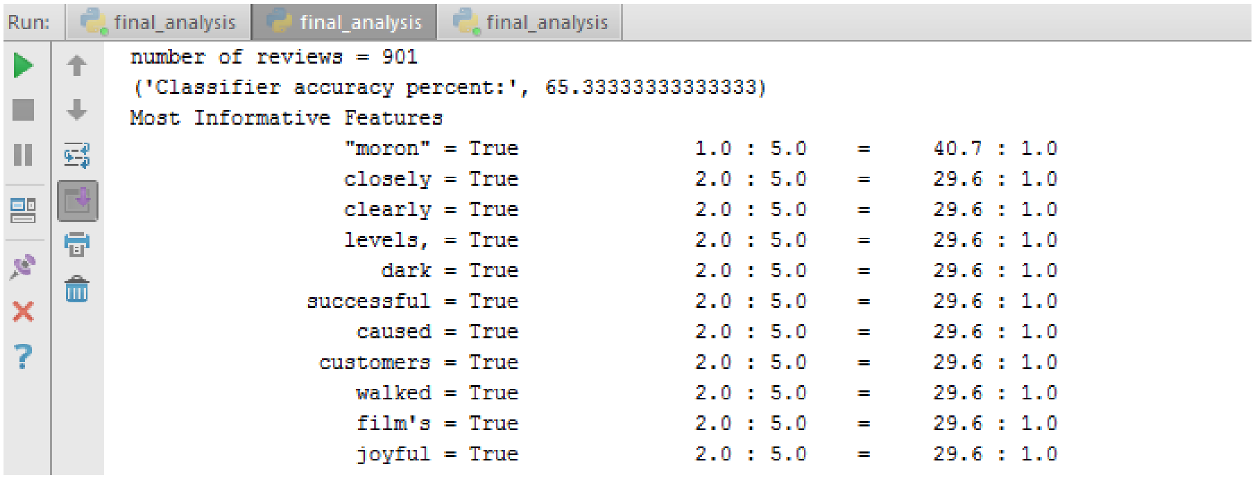
- Most Informative feature output is giving some good result. For example, words like “moron” are 40.7 times more likely to be present in a review with rating 1.0 w.r.t. review with rating 5.0

Output Screens:



```
number of reviews = 924
('Classifier accuracy percent:', 68.0)
Most Informative Features
                 service = True              1.0 : 5.0    =     42.1 : 1.0
                inserted = True              2.0 : 5.0    =     32.4 : 1.0
             complaining = True              2.0 : 5.0    =     32.4 : 1.0
                  jumped = True              2.0 : 5.0    =     19.4 : 1.0
                location = True              2.0 : 5.0    =     19.4 : 1.0
                  barely = True              2.0 : 5.0    =     19.4 : 1.0
              redemption = True              2.0 : 5.0    =     19.4 : 1.0
            step-by-step = True              2.0 : 5.0    =     19.4 : 1.0
              directing, = True              2.0 : 5.0    =     19.4 : 1.0
            relationship = True              2.0 : 5.0    =     19.4 : 1.0
                 battles = True              2.0 : 5.0    =     19.4 : 1.0
                  passes = True              2.0 : 5.0    =     19.4 : 1.0
                commonly = True              2.0 : 5.0    =     19.4 : 1.0
                 nemesis = True              2.0 : 5.0    =     19.4 : 1.0
                   fifty = True              2.0 : 5.0    =     19.4 : 1.0
                  accept = True              2.0 : 5.0    =     19.4 : 1.0
                   pages = True              2.0 : 5.0    =     19.4 : 1.0
```

Output Screen – I: The above is the Sentiment Analysis for the movie Iron Man, showing the classifier accuracy and the most informative features.



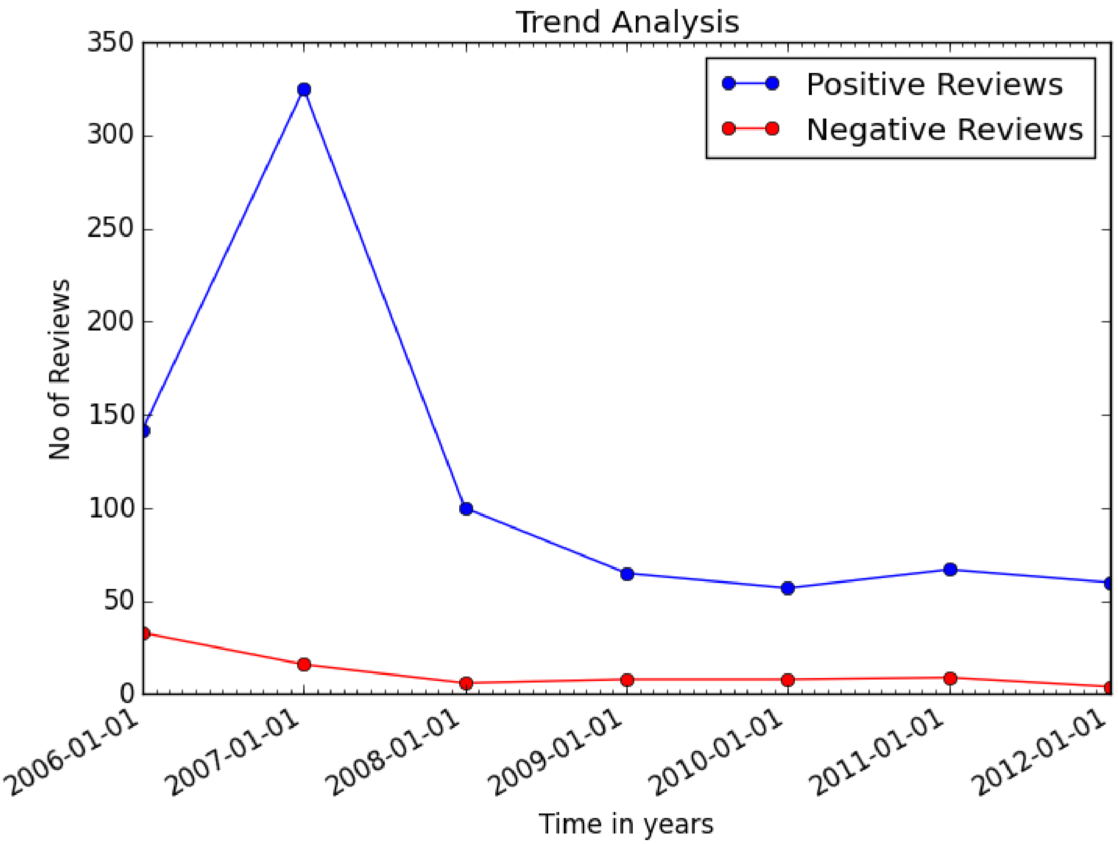
```
number of reviews = 901
('Classifier accuracy percent:', 65.33333333333333)
Most Informative Features
                 "moron" = True              1.0 : 5.0    =     40.7 : 1.0
                 closely = True              2.0 : 5.0    =     29.6 : 1.0
                 clearly = True              2.0 : 5.0    =     29.6 : 1.0
                 levels, = True              2.0 : 5.0    =     29.6 : 1.0
                    dark = True              2.0 : 5.0    =     29.6 : 1.0
              successful = True              2.0 : 5.0    =     29.6 : 1.0
                  caused = True              2.0 : 5.0    =     29.6 : 1.0
               customers = True              2.0 : 5.0    =     29.6 : 1.0
                  walked = True              2.0 : 5.0    =     29.6 : 1.0
                  film's = True              2.0 : 5.0    =     29.6 : 1.0
                  joyful = True              2.0 : 5.0    =     29.6 : 1.0
```

Output Screen – II: The above is the Sentiment Analysis for the movie Iron Man, showing the classifier accuracy and the most informative features.

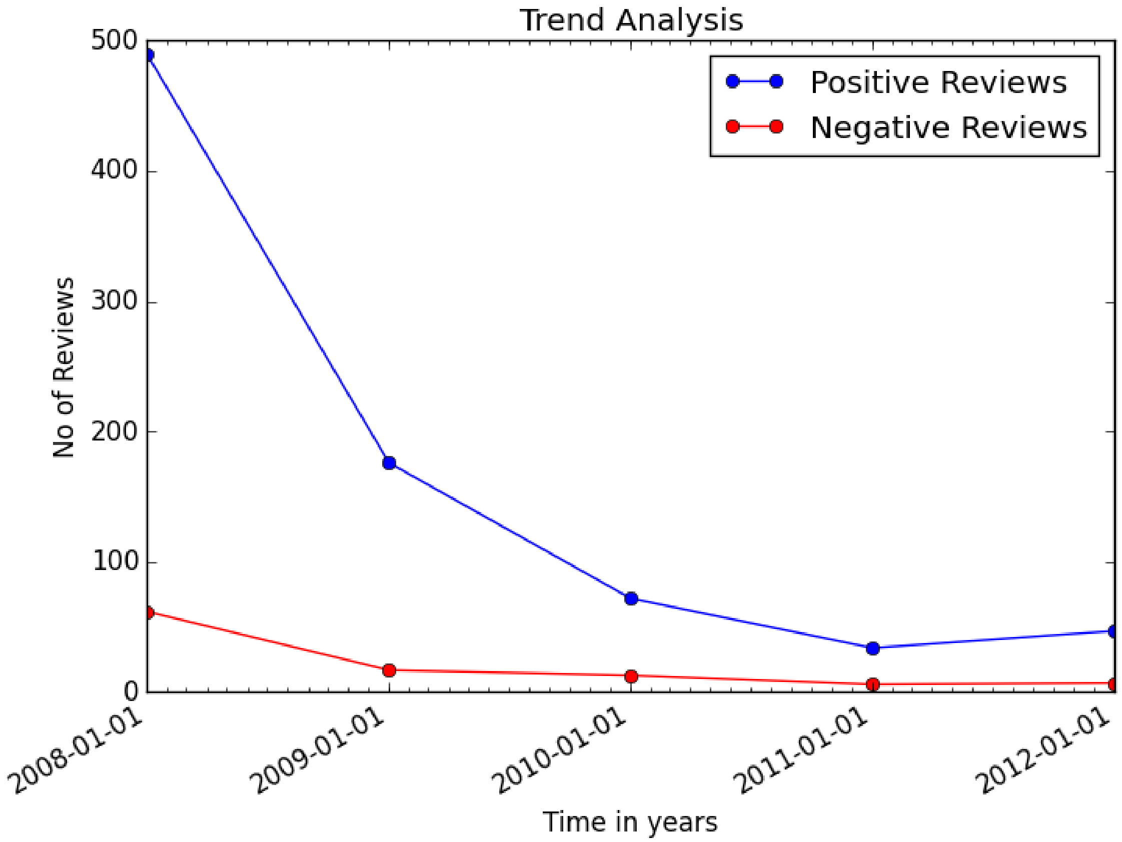
Trend Analysis:

We marked the reviews with rating above 3.0 as positive and the ones with rating 3.0 or less as negative. We then count the number of positive and negative reviews each year using review/time field, which is the UNIX time stamp given with the review. For the trend analysis we plot number of positive and negative reviews in each year using line graph.

Output Screens:



Output Screen – I: The above graph shows the trend of positive and negative reviews given for the movie Cars over the years.



Output Screen – II: The above graph shows the trend of positive and negative reviews given for the movie Iron Man over the years.

Conclusion:

By looking at the most informative features derived from sentiment analysis, one can clearly notice the kind of words and how frequently that they are used to give a review with in that review category.

Looking at the trend analysis one can clearly say for a product, how the positive and negative reviews progress over the time.

Limitations and Future scope:

1. The classifier accuracy percentage we achieved is relatively less since we do not have proper large datasets for individual product/movie.
2. Better trends can be drawn out in future provided we have more and better data.
3. We can implement naïve based classifier using SVM to get better results.

References:

[1] Stanford Network Analysis Project Repository, http://snap.stanford.edu/data/web-Movies.html

[2] Matplotlib, https://en.wikipedia.org/wiki/Matplotlib , http://matplotlib.org/

[3] Natural Language Toolkit, https://en.wikipedia.org/wiki/Natural_Language_Toolkit, http://www.nltk.org/

[4] Numpy, https://en.wikipedia.org/wiki/NumPy