1. **INTRODUCTION**

Sentiment analysis is a natural Language Processing and Information Extraction task that aims to obtain writer’s feelings expressed in positive or negative comments, questions and requests, by analyzing a large number of documents. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall tonality of a document.

**1.1 Need**

If you have 1 to 10 articles, the most effective way to measure sentiment is to simply read them. But what happens if you have 50,000? This is where sentiment analysis can provide some directional insight and set the tone for further analysis.

Sentiment analysis provide a simple, fast and efficient way to understand large amount of such data and help us to take business related decision quickly. This can provide companies a powerful tool to understand customers and their views and help them to provide better service/products.

**1.2 Applications**

Sentimental Analysis is considered to be the future of Ad optimization. Growing availability of opinion rich resources like online review sites, blogs, social networking sites have made this “decision-making process” easier for us. With explosion of Web 2.0 platforms consumers have a soapbox of unprecedented reach and power by which they can share opinions. Major companies have realized these consumer voices affect shaping voices of other consumers. Sentiment Analysis thus finds its use in Consumer Market for Product reviews, marketing for knowing consumer attitudes and trends, Social Media for finding general opinion about recent hot topics in town, Movie to find whether a recently released movie is a hit.

The future might see applications wherein a system gauges the human emotion through sensory means and then creates an environment that helps improve the human life in general.

This section describes a few of these applications that have been built or are possibilities in the near future.

**Applications to Review-related Websites**

Today Internet has an entire gamut of reviews and feedbacks on almost everything. These include movie review, product reviews, feedbacks on political issues etc. Thus there is a need for a sentiment engine that can extract sentiments about a particular entity. It will provide a consolidated feedback or rating for the given topic. Such applications would not themselves contain any opinions, but they would fetch the opinionated text from various resources and provide an elective polarity. This would serve the need of both the users and the vendors.

Another application of Sentiment Analysis is in automatic summarization of user reviews. Automatic summarization is the creation of a summary of the entire review using an automated program. In case of user reviews, it is difficult for a new user to look at all the reviews thoroughly and understand what aspect of the product is not appreciated. Thus, there is a need of a summarizing application that will briefly inform the user about the polarity of the reviews, for example, thumbs up or thumbs down for the topic.

**Applications as a Sub-component Technology**

A sentiment predictor system can be naturally considered to aid a recommender system. The recommender system will not recommend items that receive a lot of negative feedback.

While placing advertisements in sidebars it is important to understand the sensitivity of the users. A further improvement would be to detect the sentiment expressed in the page and thus bring up advertisements relevant to the sentiment. For example, on a positive review about a product an advertisement about a related product from the same manufacturer will improve the sales. Conversely, if a negative sentiment is detected then an advertisement from a competitor would be appreciated.

**Applications in Business Intelligence**

It has been observed that more and more people nowadays tend to look upon reviews of products online before buying them. And for many businesses the online opinion can make or break their product. Thus, sentiment analysis finds an important role in businesses. Businesses wish to understand the online reviews in order to improve their products and in turn their reputation.

**Applications across different Domains**

So far we have mentioned only applications pertaining to a business setting. But, Sentiment Analysis finds various applications in other fields. Studies in sociology and other fields have been aided by Sentiment Analysis systems that show trends in human emotions especially on social networks.

1. **REVIEW OF LITERATURE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***S.No.*** | ***Studies*** | ***Mining Techniques Used*** | ***Data Source*** | ***Performance***  ***(Accuracy)*** | ***Precision*** |
| 1. | *Kaiquan Xu(2011)* | *Multiclass SVM* | *Amazon reviews* | *61%* | *61.9%* |
| 2. | *Long Sheng*  *(2011)* | *Back Propagation Neural Network* | *Movie Review* | *64%* | *60%* |
| 3. | *Rui Xia (2011)* | *Naïve Bayes, Maximum Entropy, SVM* | *Movie Review, Multi Domain Dataset* | NB- 85.8  ME- 85.4  SVM- 86.4 | - |
| 4. | *Xue Bai (2011)* | *Naïve Bayes* | *Movie*  *review* | *92%* | *-* |
| 5. | *Ziqiong (2011)* | *Naïve Bayes, SVM* | *Cantonese reviews* | *93%* | *-* |
| 6. | *Gamgarn somprasti (2010)* | *Maximum Entropy* | *Amazon reviews* | *-* | *72.6%* |
| 7. | *Gang li (2010)* | *K-means Clustering* | *Movie Review* | *78%* | *-* |
| 8. | *Yulan He (2010)* | *Sentiment Lexicon* | *Movie Review* | *74.7%* | *-* |
| 9. | *Zhu Jian (2010)* | *Back propagation* | *Movie Review* | *86%* | *-* |
| 10. | *Melville (2009)* | *Bayesian classification* | *Blogs* | *91.2%* | *-* |
| 11. | Rudy (2009) | *ID3, SVM, Hybrid* | *Movie Review* | *89%* | *-* |
| 12. | *QingliangMiao (2009)* | *Lexical resource* | *Amazon reviews* | *87.6%* | *87.4%* |
| 13. | *Songho tan (2008)* | *Centroid classifier, K-Nearest, SVM* | *Consentir* | *90% (SVM)* | *-* |
| 14. | *Zhou and Chaovalit (2008)* | *ontology-supported polarity mining* | *Movie review* | *72.2%* | - |
| 15. | *Godbole et al. (2007)* | *Lexical approach* | *blog posts* | *82.7–95.7%* | - |
| 16. | *Kennedy and Inkpen (2006)* | *support vector machines* | *Movie review* | *86.2%* | *-* |

Table 2: Literature Survey [8]

Support vector machines (SVM), a discriminative classifier is considered the best text classification method (Rui Xia, 2011; Ziqiong, 2011; Songho tan, 2008 and Rudy Prabowo, 2009). The support vector machine is a statistical classification method proposed by Vapnik . Based on the structural risk minimization principle from the computational learning theory, SVM seeks a decision surface to separate the training data points into two classes and makes decisions based on the support vectors that are selected as the only effective elements in the training set. Multiple variants of SVM have been developed in which Multi class SVM is used for Sentiment classification (Kaiquan Xu, 2011).

Rui Xia (2011) made a comparative study of the effectiveness of ensemble technique for sentiment classification by efficiently integrating different feature sets and classification algorithms to synthesize a more accurate classification procedure. In his work, two types of feature sets are designed for sentiment classification, namely the part-of-speech based feature sets and the word-relation based feature sets. Then, three text classification algorithms, namely naive Bayes, maximum entropy and support vector machines, are employed as base-classifiers for each of the feature sets to predict classification scores.

Long-Sheng Chen (2011) proposed a neural network based approach, which combines the advantages of the machine learning techniques and the information retrieval techniques.

Naive Bayes is a simple but effective classification algorithm. The Naive Bayes algorithm is widely used algorithm for document classification (Melville et al., 2009; Rui Xia, 2011; Ziqiong, 2011; Songho tan, 2008 and Qiang Ye, 2009). The basic idea is to estimate the probabilities of categories given a test document by using the joint probabilities of words and categories.

In most of the comparative studies it is found that SVM outperforms other machine learning methods in sentiment classification. Ziqiong Zhang (2011) showed a contradiction in the performance of SVM. They focused their interest on proving that the chosen machine learning model could be able to draw its own conclusion from the distribution. Despite its unrealistic independence assumption, the naive Bayes classifier surprisingly achieves better performance than SVM.

Zhu Jian (2010) proposed an individual model based on Artificial neural networks to divide the movie review corpus into positive , negative and fuzzy tone which is based on the advanced recursive least squares back propagation training algorithm.

Gang Li & Fei Liu (2010) developed an approach based on the k-means clustering algorithm. The technique of TF-IDF (term frequency – inverse document frequency) weighting is applied on the raw data. Then, a voting mechanism is used to extract a more stable clustering result.

Rudy Prabowo (2009) described an extension by combining rule-based classification, supervised learning and machine learning into a new combined method. For each sample set, they carried out 10-fold cross validation. For each fold, the associated samples were divided into training and a test set.

Songho Tan (2008) presents an empirical study of sentiment categorization on Chinese documents. He investigated four feature selection methods (MI,IG, CHI and DF) and five learning methods (centroid classifier, K-nearest neighbor, winnow classifier, Naive Bayes and SVM) on a Chinese sentiment corpus. From the results he concludes that SVM exhibits the best.

Chaovalit and Zhou (2008) compared the Semantic Orientation approach with the N-gram model machine learning approach by applying to movie reviews. They confirmed from the results that the machine learning approach is more accurate but requires a significant amount of time to train the model.

1. **REPORT ON PRESENT INVESTIGATION**

**Unsupervised Sentiment Analysis with Emotional Signals:**

A traditional way to perform unsupervised sentiment analysis is the lexicon-based method. These methods employ a sentiment lexicon to determine overall sentiment polarity of a document. Lexicon-based unsupervised sentiment analysis becomes difficult due to the distinct features of social media data.

Abundant emotional signals are observed in social media. *Emotional signals are* any information that could be correlated with sentiment polarity of a document or the words in the document. For example, when communicating in the physical world, it is common for people to supplement vocal interaction with gestures and facial expressions. Similarly, in social media, users develop visual cues that are strongly associated with their emotional states. These cues, known as emoticons (or facial expressions), are widely used to show the emotion that a user’s post represents. When the authors use emoticons, they are effectively marking up the text with an emotional state.

**Automated Sentiment Analysis: Reality**

***The current report on Automated Sentimental analysis tools says:***Automated sentiment analysis is less accurate then flipping a coin when it comes to determining whether brand mentions in social media are positive or negative, according to a white paper from FreshMinds.Tests of a range of different social media monitoring tools conducted by the research consultancy found that comments were, on average, correctly categorized only 30% of the time.FreshMinds’ experiment involved tools from Alterian, Biz360, Brandwatch, Nielsen, Radian6, Scoutlabs and Sysomos. The products were tested on how well they assessed comments made about the coffee chain Starbucks, with the comments also having been manually coded.

On aggregate the results look good, said FreshMinds. Accuracy levels were between 60% and 80% when the automated tools were reporting whether a brand mention was positive, negative or neutral.

1. **AIM AND OBJECTIVE**

**Aim:**

The aim of our project is to analyze the sentiments about a particular topic of user’s interest which is discussed on twitter. We analyze it and display classified results of twitter tweets in terms of positive, negative and neutral along with the count represented by Google charts.

**Objective:**

* To use different machine learning classifiers and appropriate datasets to train them.
* To use OAuth and Twitter API to fetch live tweets.
* To develop convenient user interface of the twitter sentiment analyzer.
* To reach reasonable accuracy in results of all different classifiers.

1. **PROBLEM STATEMENT**

Before buying a product or a service people want to check if it is value for money or not. But it is a rigorous task for a customer to go through all the reviews on a website about the particular service or a product before buying it. Some of the users may have positive reviews while some may be negative towards it but the buyer needs to know how well the product or service is overall, according to that one will decide to buy a product.

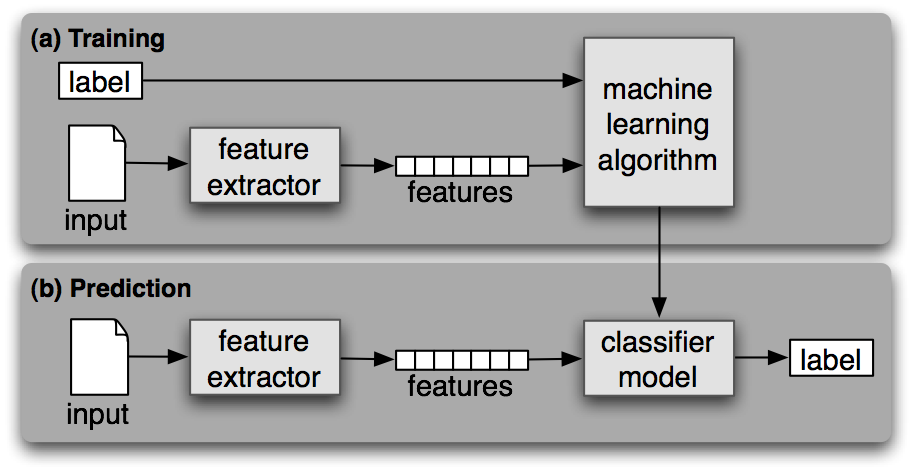
Our focus is to display the reviews and give the user a brief count on the number of reviews which are positive, negative or neutral about the product which will help him in decision making.

1. **PROPOSED SYSTEM FOR PROJECT**

A Sentiment Analyzer is what we need to solve the above mentioned problem. This system will contain a trained machine learning classifier. When a user enters particular keyword the system will retrieve the data about it from sources like twitter and the trained classifier will calculate the result about the queried keyword.

Our approach is to use different machine learning classifiers and feature extractors. The machine learning classifiers are Naive Bayes and Maximum Entropy. All of these classifiers require training data and hence these methods fall under the category of supervised classification.

The feature extractors are unigrams and unigrams with weighted positive and negative keywords. We build a framework that treats classifiers and feature extractors as two distinct components. This framework allows us to easily try out different combinations of classifiers and feature extractors.

****

**Fig 6.1. Supervised Classification [9]**

We can compare the accuracy of different classifiers. The accuracy also depends on the quality and amount of training dataset.

1. **REQUIREMENTS ANALYSIS (SRS)**

**Usability:**

The user of the system will find this system to be very easy to use since they just have to enter the keyword of their interest and the result will be shown to them in a matter of seconds.

**Reliability:**

This system is reliable because of the use of up to the mark NLP and Machine learning techniques. According to various researches, on applying appropriate NLP techniques the accuracy of Naive Bayes and Maximum entropy can cross 80%.

**Performance:**

The performance of this system will be fairly good in terms of speed. The only thing that will make it slow is the restriction of twitter to fetch the tweets.

**Implementation:**

Implementation is done by using python 2.7, NLP and Machine Learning approach. Machine Learning tasks will be handled by NLTK 3 and libsvm libraries. All these libraries have efficient classes which will carry out the tasks in minimum time. Also for interface we will use webpy, which is again a python library for plotting graphs of the data analyzed.

**Interface:**

The interface will consist of a text box which will take the input (i.e. a keyword) from the user and a button to submit that keyword. The Output will be shown in terms of graph which shows the changes in sentiments over the past few days or according to how we program it.

1. **SCOPE (FEASIBILITY OF PROJECT)**

**1. Technical Feasibility**

The technical requirements of the project are a bare minimal. It only requires Python and NLTK, with some basic hardware configuration. Python can be learnt easily and NLTK has a great documentation resource to learn from. Hence, the project is technically feasible at all levels.

**2. Financial Feasibility**

The project requires Python and NLTK both of which are freely available. Hence, the project has a great financial feasibility and can be implemented without any cost and in any environment.

**3. Operational Feasibility**

The project was made to solve the problem of large amount of time required to understand the views of the users. After understanding this problem, we made this project to do automation of this work. It will improve the operational time and cost of the things. Hence, the project is operational in many places.

1. **METHODOLOGY**

Our methodology includes the following steps:

1. **Gathering the data**

Gather the data for training from publicly available twitter datasets which contain positive, negative and neutral tweets.

1. **Pre-Processing the data**

The tweets are processed, which firstly involves converting all the textual data to lower case. The tweets also contain URL, hash tags, whitespace, usernames and punctuation marks which don’t provide any information regarding the sentiment. So we filter out that things and we are left with only words for analyzing.

1. **Feature Vector**

Feature vector is the most important concept in implementing a classifier. In tweets, we can use the presence/absence of words that appear in tweet as features. In the training data, consisting of positive, negative and neutral tweets, we can split each tweet into words and add each word to the feature vector. Some of the words might not have any say in indicating the sentiment of a tweet and hence we can filter them out. Adding individual (single) words to the feature vector is referred to as 'unigrams' approach. Some of the other feature vectors also add 'bi-grams' in combination with 'unigrams'. For example, 'not good' (bigram) completely changes the sentiment compared to adding 'not' and 'good' individually. Here, for simplicity, we will only consider the unigrams.

**Filtering Tweet Words**

1. *Stop words* - a, is, the, with etc. These words don't indicate any sentiment and can be removed.
2. *Repeating letters* - if we look at the tweets, sometimes people repeat letters to stress the emotion. E.g. hunggrryyy, huuuuuuungry for 'hungry'. We can look for 2 or more repetitive letters in words and replace them by 2 of the same.
3. *Punctuation* - we can remove punctuation such as comma, single/double quote, question marks at the start and end of each word. E.g. beautiful!!!!!! replaced with beautiful
4. *Words must start with an alphabet* - For simplicity sake, we can remove all those words which don't start with an alphabet. E.g. 15th, 5.34am.

After this words are classified as positive, negative or neutral we add these words to feature vector. In this way we can make a training set and store it in .csv file

1. **Feature Extraction & Feature List**

We will use Natural Language Toolkit (NLTK) and implement the two classifiers namely Naive Bayes and Maximum Entropy in Python 2.7. The feature vector obtained from previous steps are labeled and saved in a .csv file. Now we extract the new tweets and labels from the csv file and process it. The feature words extracted from the tweets are processed to convert it into feature vector. The feature vector is then used to train the subsequent classifier. With the help of NLTK we Bulk Extract the Features of the tweets and the respective classifiers.

1. **Naive Bayes Classifier**

In [machine learning](http://en.wikipedia.org/wiki/Machine_learning), Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying [Bayes' theorem](http://en.wikipedia.org/wiki/Bayes%27_theorem) with strong (naive) [independence](http://en.wikipedia.org/wiki/Statistical_independence) assumptions between the features.  A Naive Bayes classifier assumes that the value of a particular feature is unrelated to the presence or absence of any other feature, given the class variable.

1. **Maximum Entropy**

In Maximum Entropy classification, the probability that a document belongs to a particular class given a context must maximize the entropy of the classification system. By maximizing entropy, it is ensured that no biases are introduced into the system.

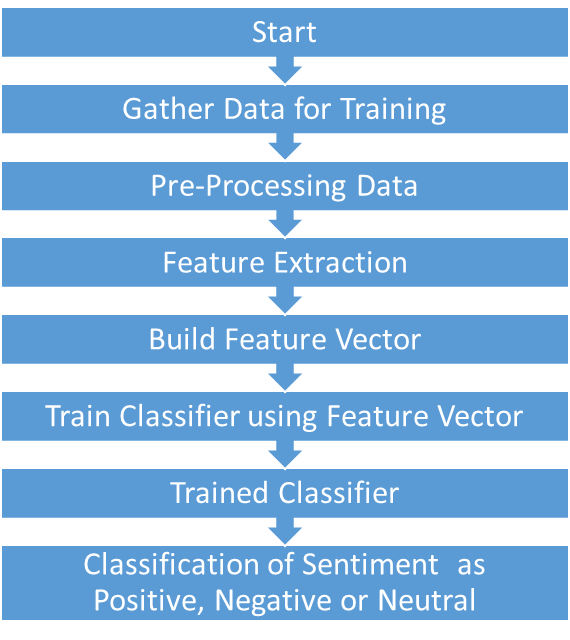
1. **Retrieving tweets for a particular topic (UI)**

When we build a twitter sentiment analyzer, the input to our system will be a user entered keyword. Hence, one of the building blocks of this system will be to fetch tweets based on the keyword within selected time duration. We use OAuth with Twitter API to fetch tweets from Twitter.

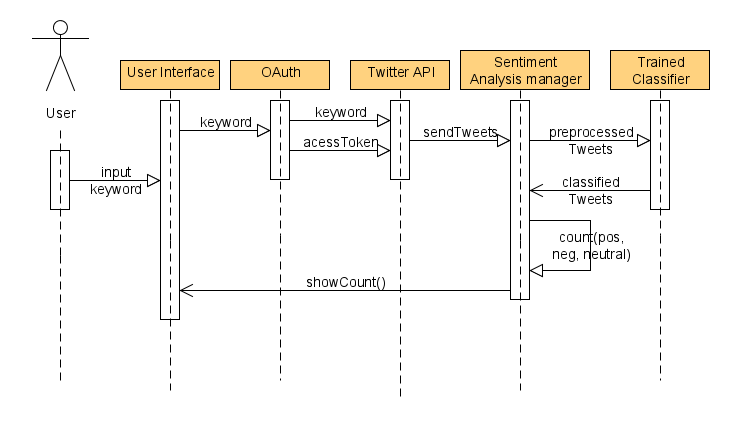
* 1. **Classification of Tweets :**

Finally,we pre-process the tweets gathered from the twitter API and classify each tweet as positive, negative or neutral by running our trained classifier on those tweets and show the result graphically (line chart or column chart using [Google Charts](https://developers.google.com/chart/)).

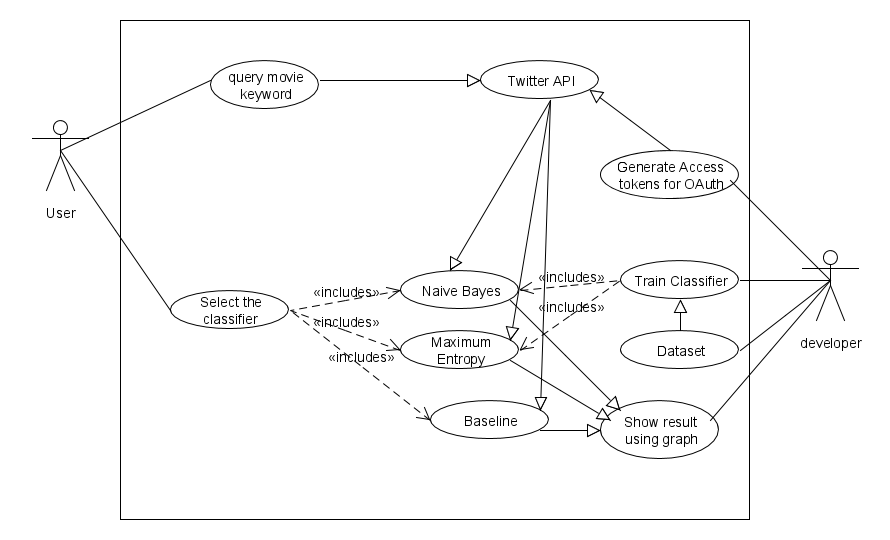
1. **DESIGN DETAILS**

****

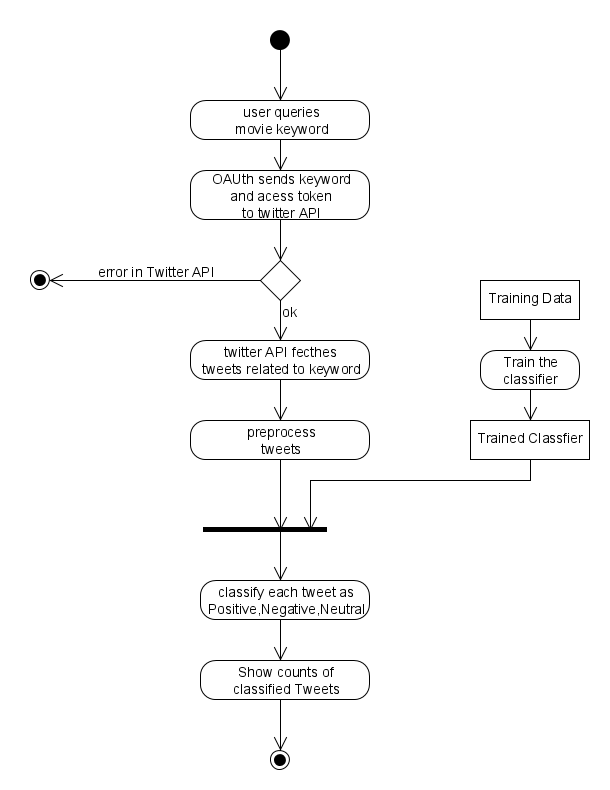
**Fig 10.1 Control Flow Diagram of System**



**Fig 10.2 Sequence Diagram**



**Fig. 10.3 Use Case Diagram**



**Fig 10.4 Activity Diagram**

**11. IMPLEMENTATION PLAN**

**11.1 H/w and S/w requirements**

For implementation of our application we need various software and hardware configuration.

* **Hardware Requirement:**

1. CPU: 34 bit / 64-bit
2. Processor: Intel i5 (4th Generation)
3. RAM: 8 GB or more
4. Hard Disk: 500 GB or more

* **Software Requirement:**

1. Operating System: Ubuntu 12.04
2. Software: Python 2.7
3. Natural Language Toolkit (NLTK)
4. Twitter REST API
5. OAuth 2.0
6. Google Charts

**11.2Data Set**

For accurate classification we need to train the classifier with as much data as possible. Since we are developing our sentiment classifier based on twitter, we need twitter dataset. To get fairly good results one needs at least 10,000 tweets. Using a dataset which is too large is also time consuming to train hence we have made a dataset of more than 13,000 tweets .We gathered 5000 positive and 5000 negative tweets from the Cornell University’s website.

Link: **http://www.cs.cornell.edu/people/pabo/movie-review-data/**

This dataset is of movie reviews which were collected by researchers at Cornell for their research in NLP.

There were no neutral classified dataset which were available, hence we ourselves gathered around 3000 tweets from twitter and hand classified it.

**11.3 Gantt Chart**

The Gantt Chart Shows planned and actual progress for a number of tasks displayed against a horizontal time scale. It is effective and easy-to-read method of indicating the actual current status for each of set of tasks compared to planned progress for each activity of the set. Gantt Charts provide a clear picture of the current state of the project.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | August | | | September | | | October | | | November | | | December | | | January | | | February | | | March | | |
|  | 1 | 15 | 30 | 1 | 15 | 30 | 1 | 15 | 30 | 1 | 15 | 30 | 1 | 15 | 30 | 1 | 15 | 30 | 1 | 15 | 30 | 1 | 15 | 30 |
| Study existing systems |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Planning |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Problem Definition |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studying the System Requirements |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Learning Python and NLTK |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Study of Various NLP techniques |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Gathering Training Dataset |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creating a Prototype in Python |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Individual algorithm and System Testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analysis for improvement |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Improving system by fixing bugs |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Final Product Deployment |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

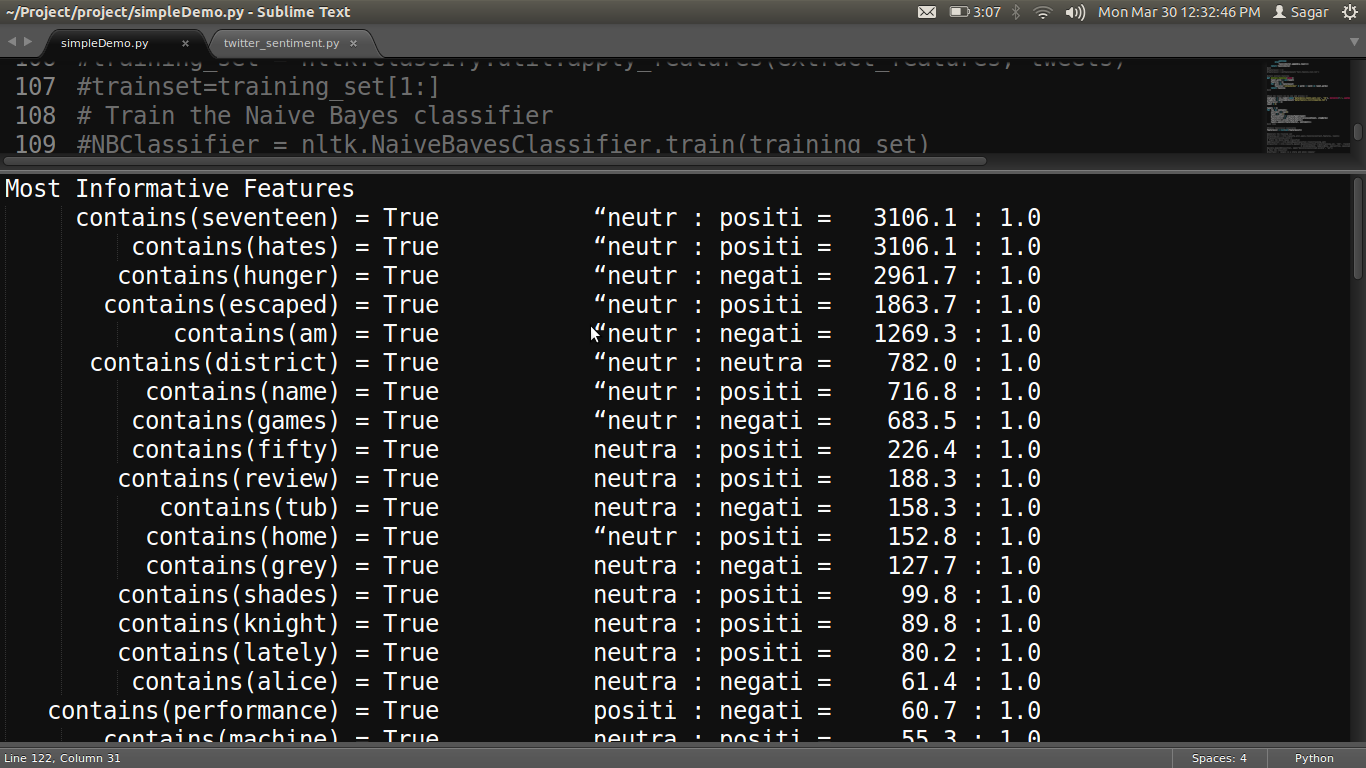
Fig 11.3 Gantt chart

* + 1. **TESTING**

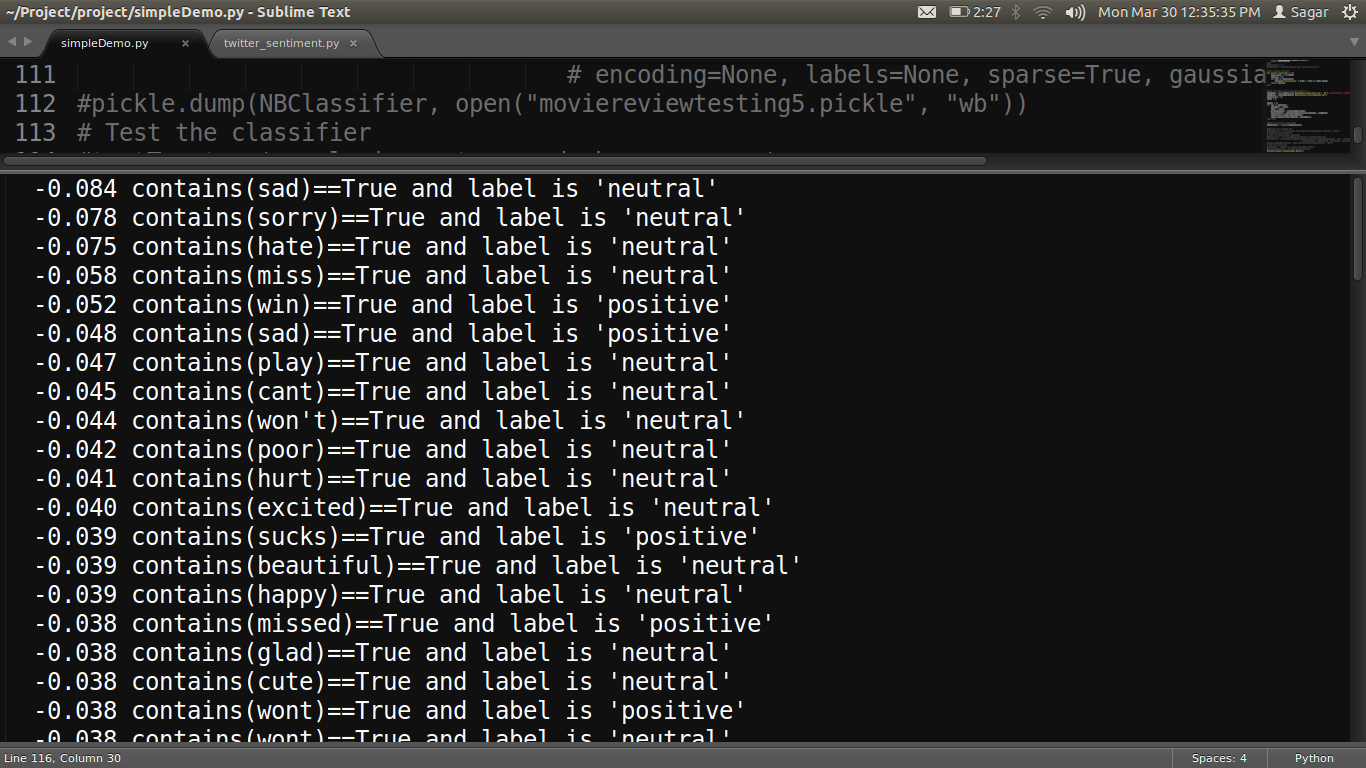
**12.1 MOST INFORMATIVE FEATURES**

We have used various algorithms for classifying the sentiments. But before analyzing the sentiment, in order to know how well the algorithm is trained, we have a function in NLTK library called the show\_most\_informative\_features(). This function is very unique and quite useful. After we have trained the classifier, we can use this function to see all the important features of the trained classifier. We have to just pass the trained classifier as a parameter and then the output of the classifier is displayed. Since baseline classifier is a simple algorithm which works on dictionary and frequency distribution, we cannot see its most informative features. Hence we can see the most informative features only for Naive Bayes and Maximum Entropy classifier.

We consider Naive Bayes classifier in this case. Below is a picture of the output of the most informative features.



It shows the word along with its presence and ratio of the sentiments. Naive Bayes works on the probability of the words and finds the probability of each word. So if the word occurs quite frequently with respect to particular sentiment, then its probability increases next time the same word occurs in the sentence. So according to the abovescreenshot, the first word that occurs in the result has the highest probability of being neutral as its neutral to positive ratio is 3106 to 1. Hence, anytime this word occurs it has very high probability of the sentence being neutral. We take all the words after filtering, calculate the probability of each word, add all the probabilities and get the final sentiment as the output.



Similarly, for maximum entropy the above screenshot shows good details about the words in the dataset used for training. As it can been seen from the screenshot, the word is displayed along with its value. The value displayed is that of the logarithm of the probability. To show the final sentiment, all the values are added and the final sentiment is displayed.

For the project we have consistently tested these features so thatthe best possible results could be obtained. These results are highly dependent on dataset being used and the words contained in the dataset. Hence, to improve the accuracy we have kept improving the dataset and always tested for the features.

**12.2 Accuracy**

We tested the accuracy of the algorithms after we trained them. And we found that Naïve Bayes performed the best as it provided the best results. Maximum Entropy was also showing very good accuracy and was close in comparison to Naïve Bayes. For testing the accuracy we have to follow a certain method which includes:

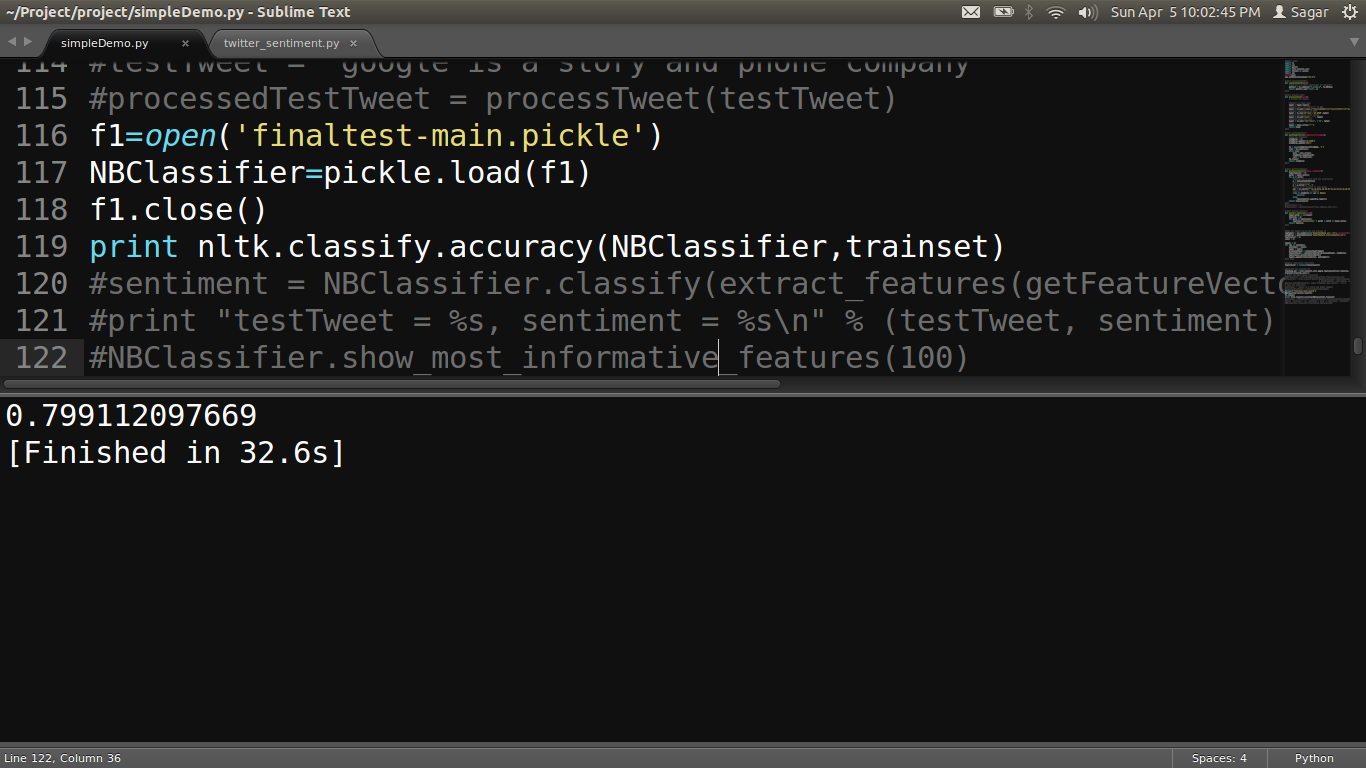
**1.** Train the classifier properly with the desired dataset. We trained it with approximately 11,000 tweets.

**2.** Make a test data-set of around 600 tweets. We made a totally new test data-set. The tweets used in training the classifier and in the test data-set should be different. Overlapping of tweets will show superficial accuracy.

**3.**Extract the features from the test set. Now use the accuracy function of NLTK and find the accuracy of the test training data-set against the already trained classifier.

We will consider the case of Naive Bayes and its accuracy calculation:

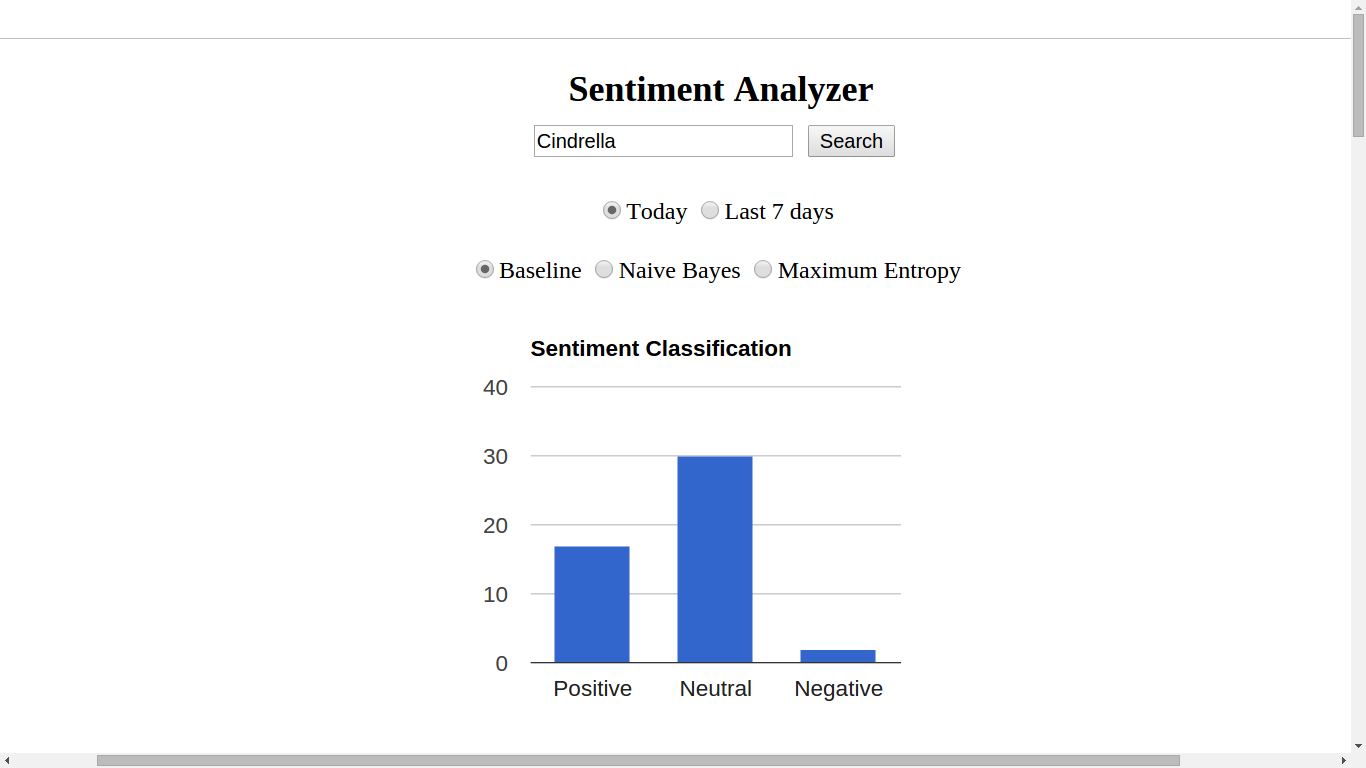
**Naive Bayes**: The Naive Bayes provides a mathematical model for finding the results and hence it is quite accurate. Below is the screenshot of the result for finding the accuracy of Naive Bayes.

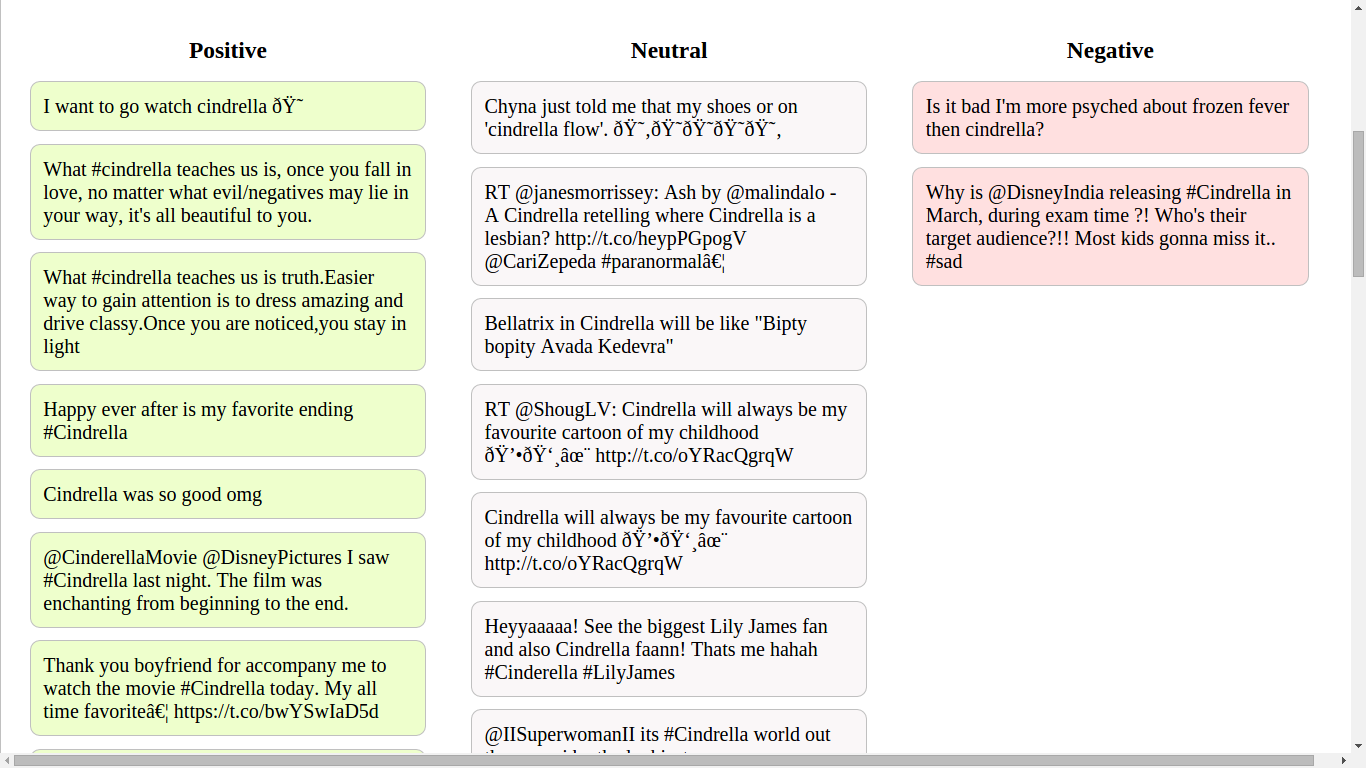


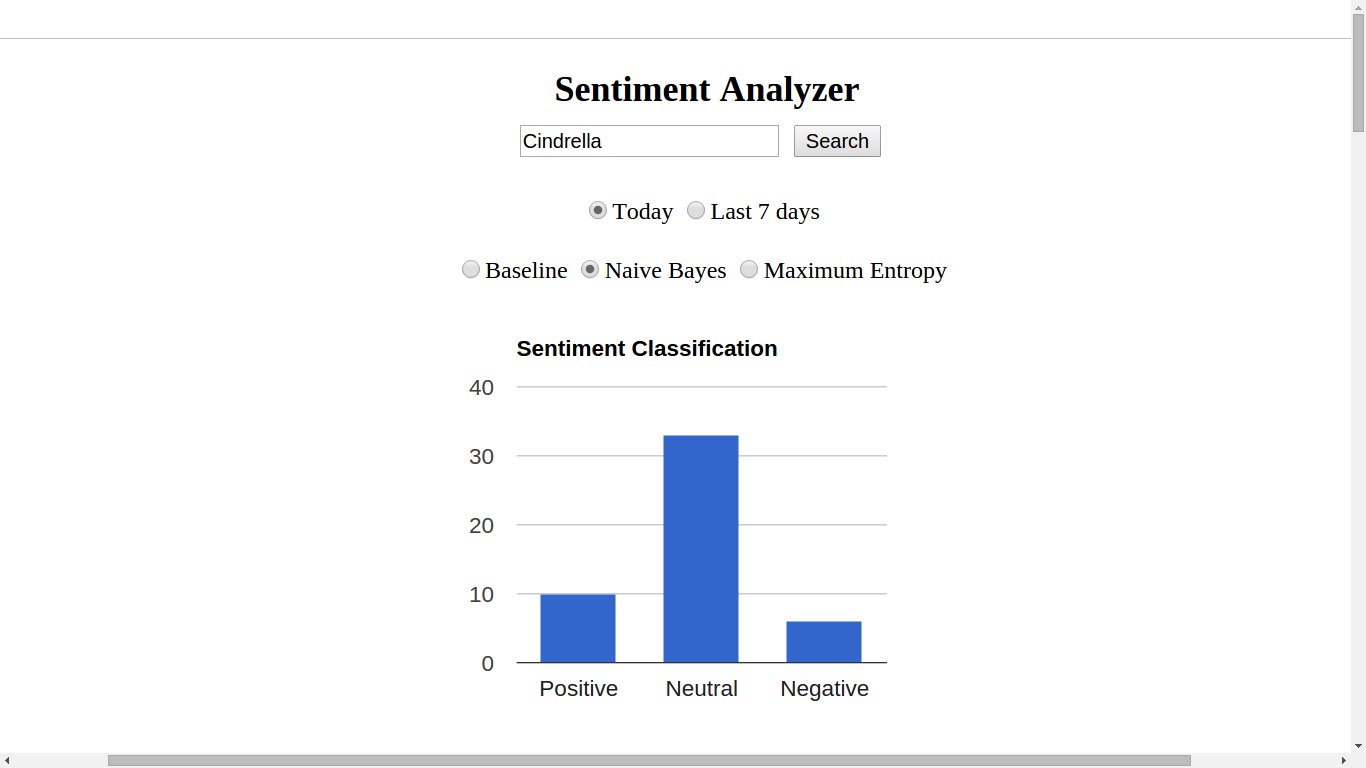
As we can see that the accuracy is approximately 80% which is very good. The accuracy is consistent with the results we got as the output. Hence it is more reliable, accurate and dependable than other algorithms used. It gives a clear idea of the sentiments and can be put to practical use without much effort.

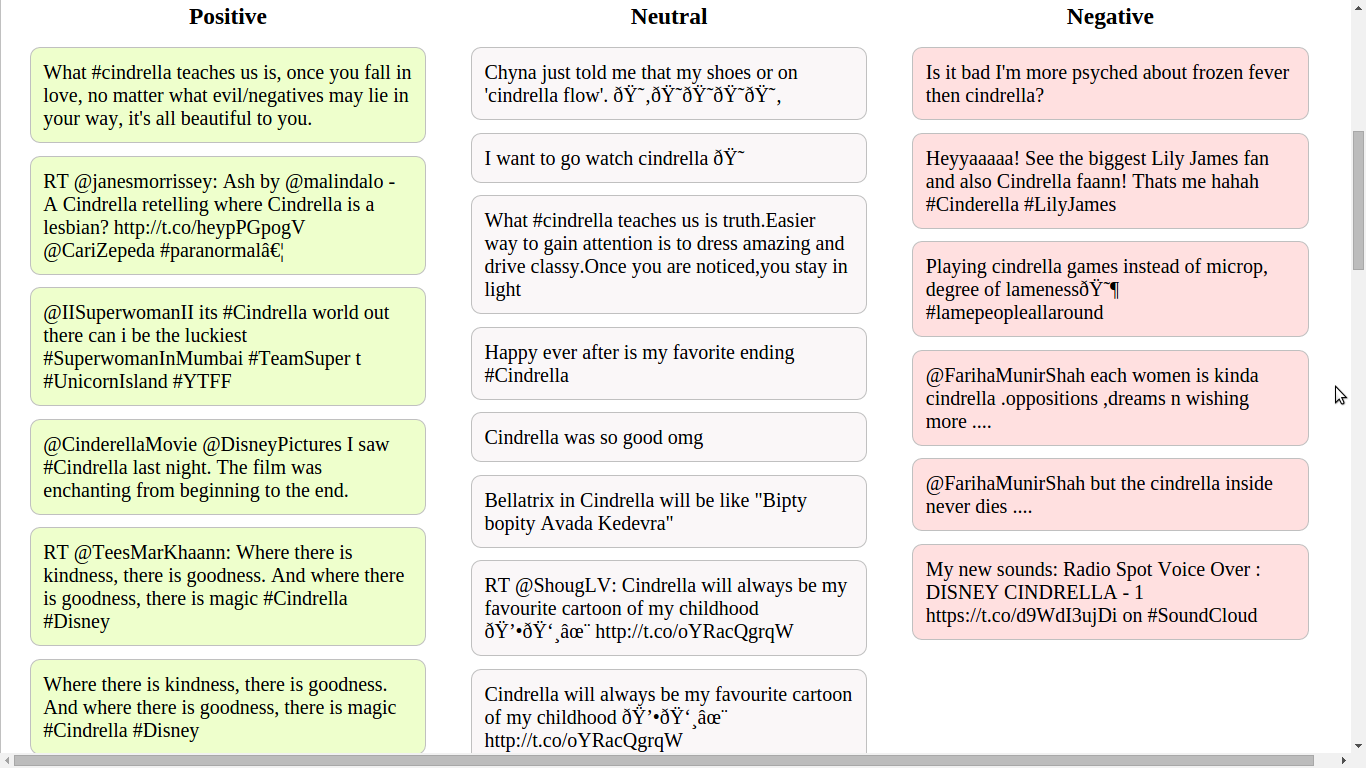
* + 1. **RESULT AND ANALYLSIS**

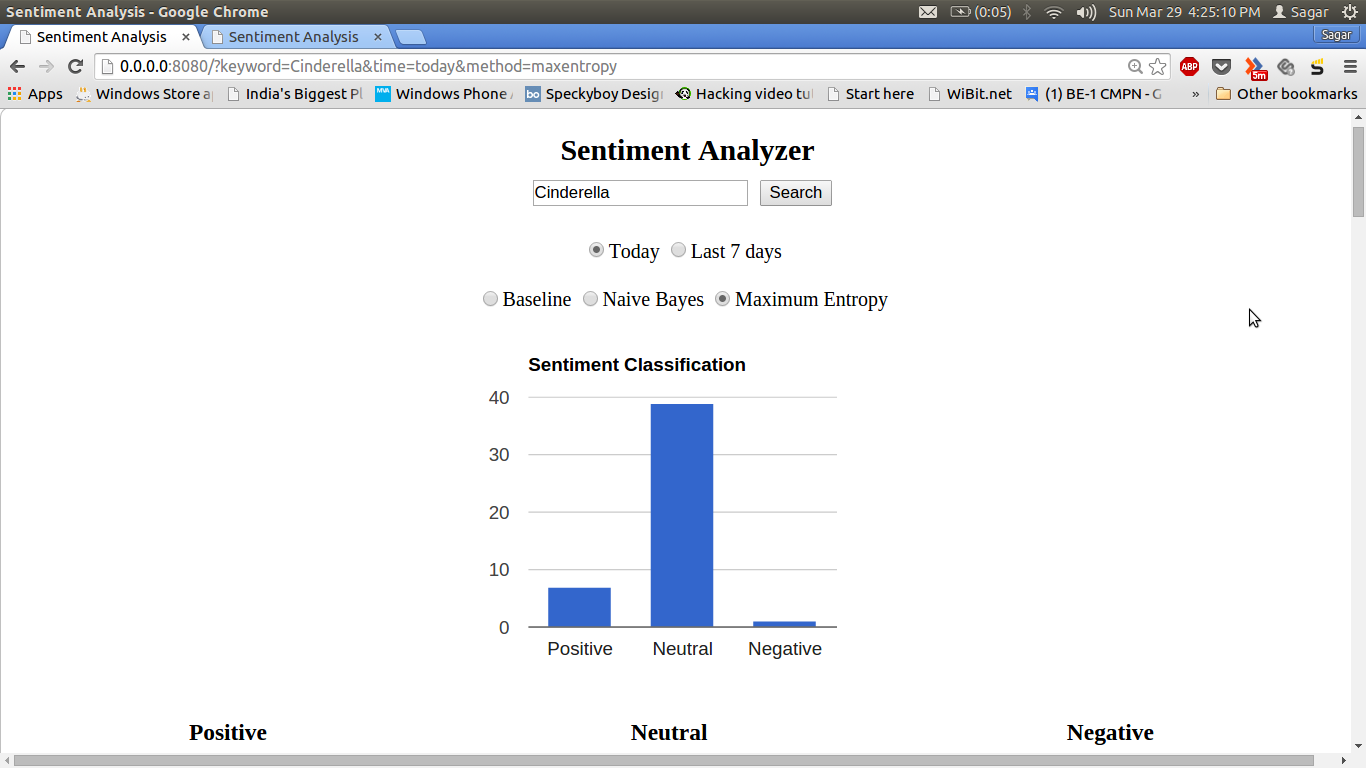
**Result for a particular day:**

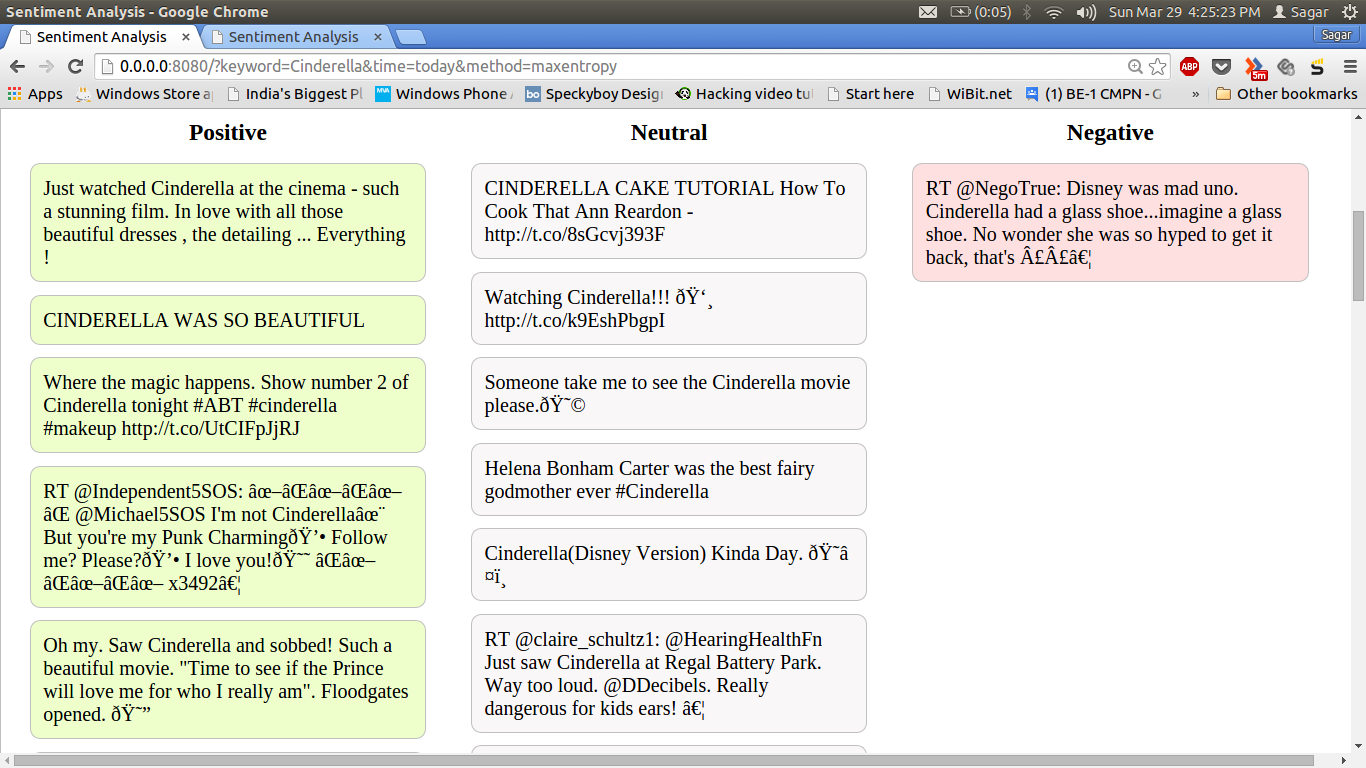
****

****

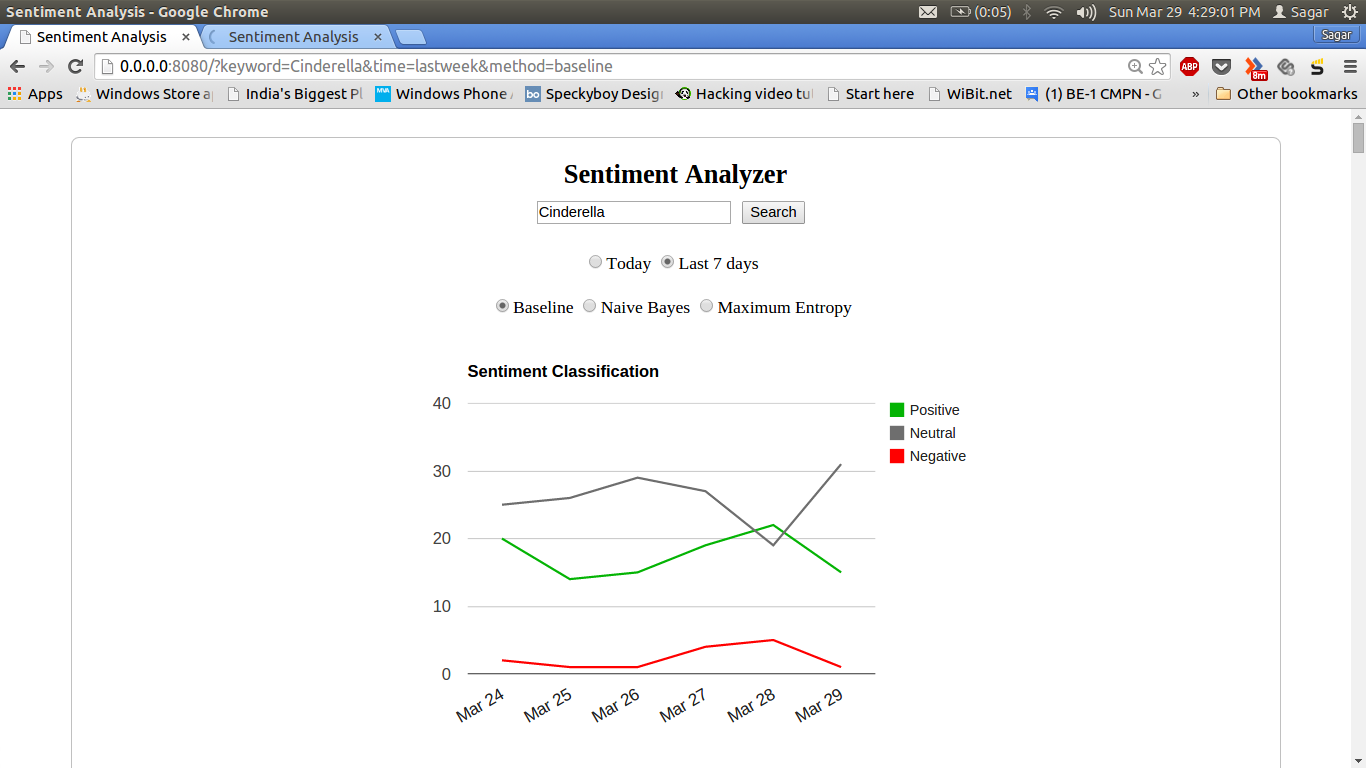
****

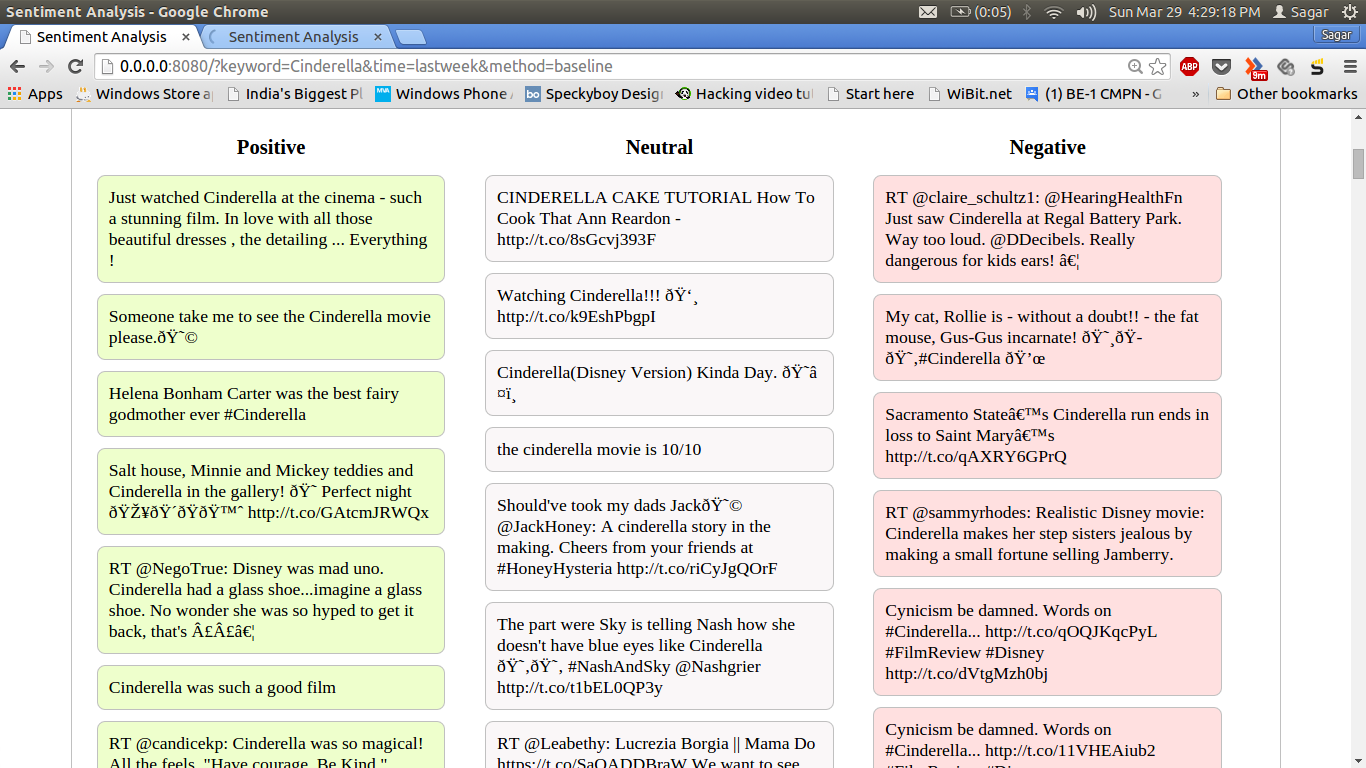
****

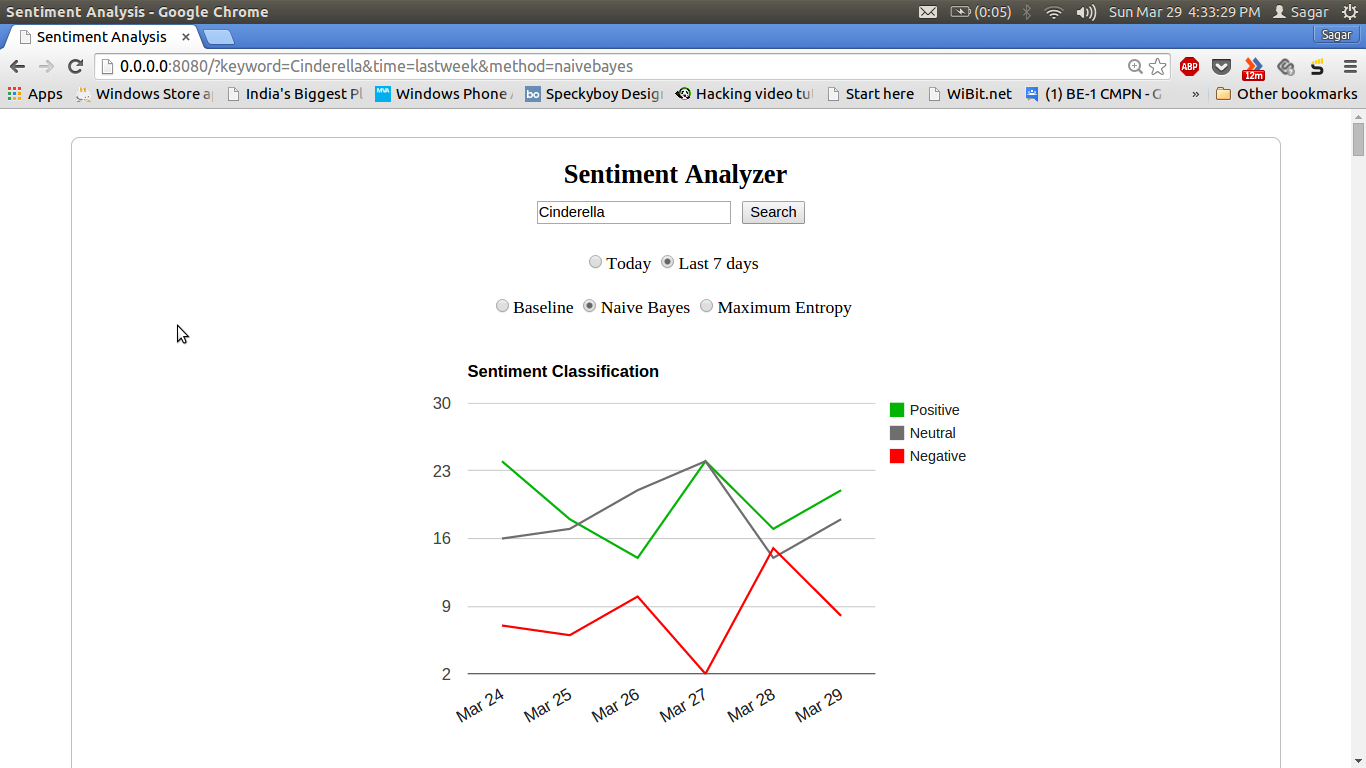
****

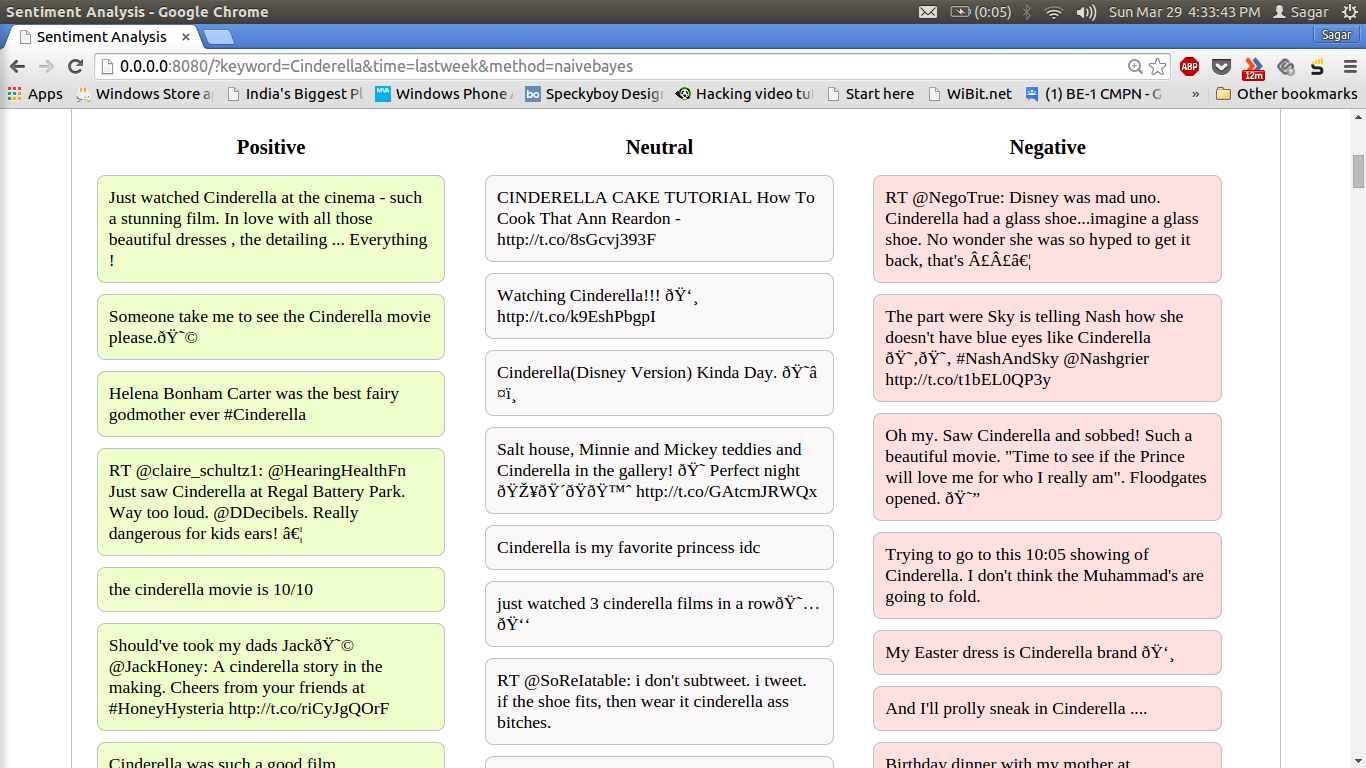
****

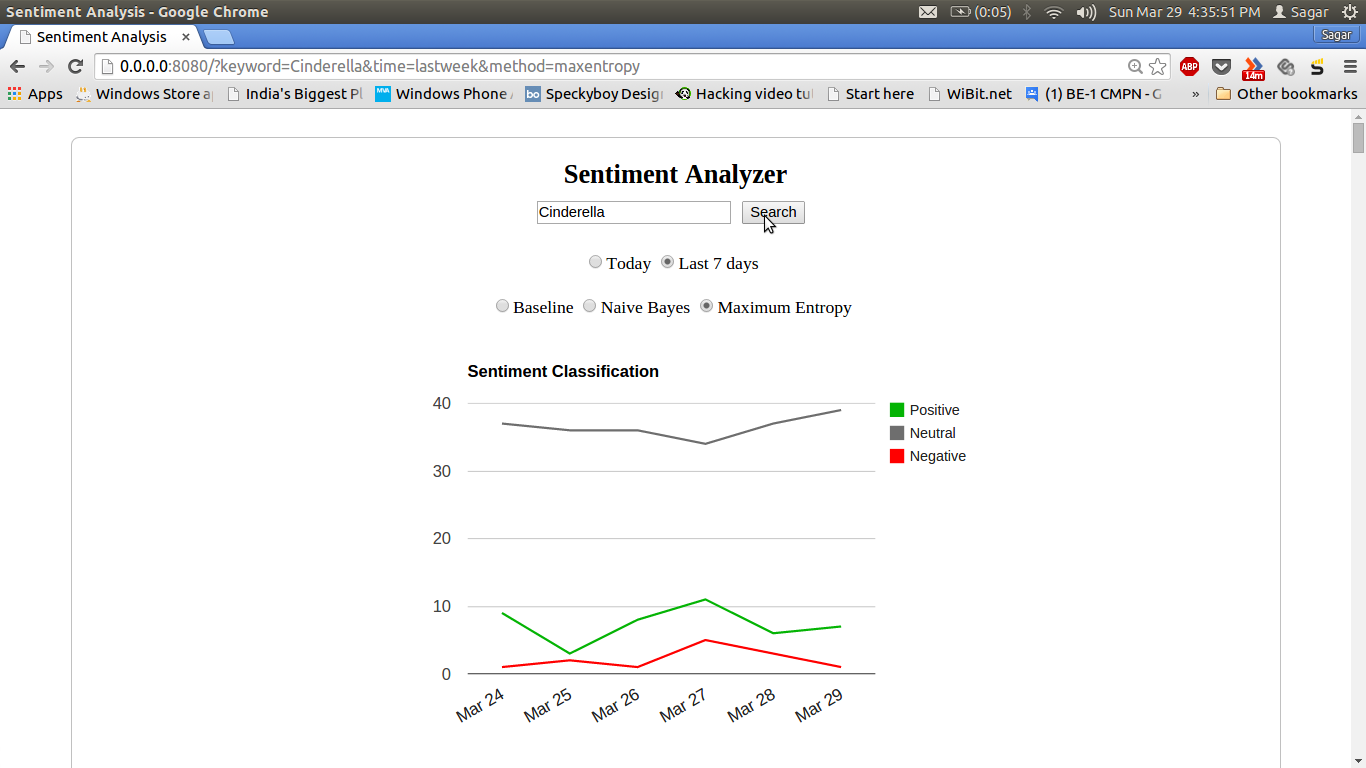
**Result for one week:**

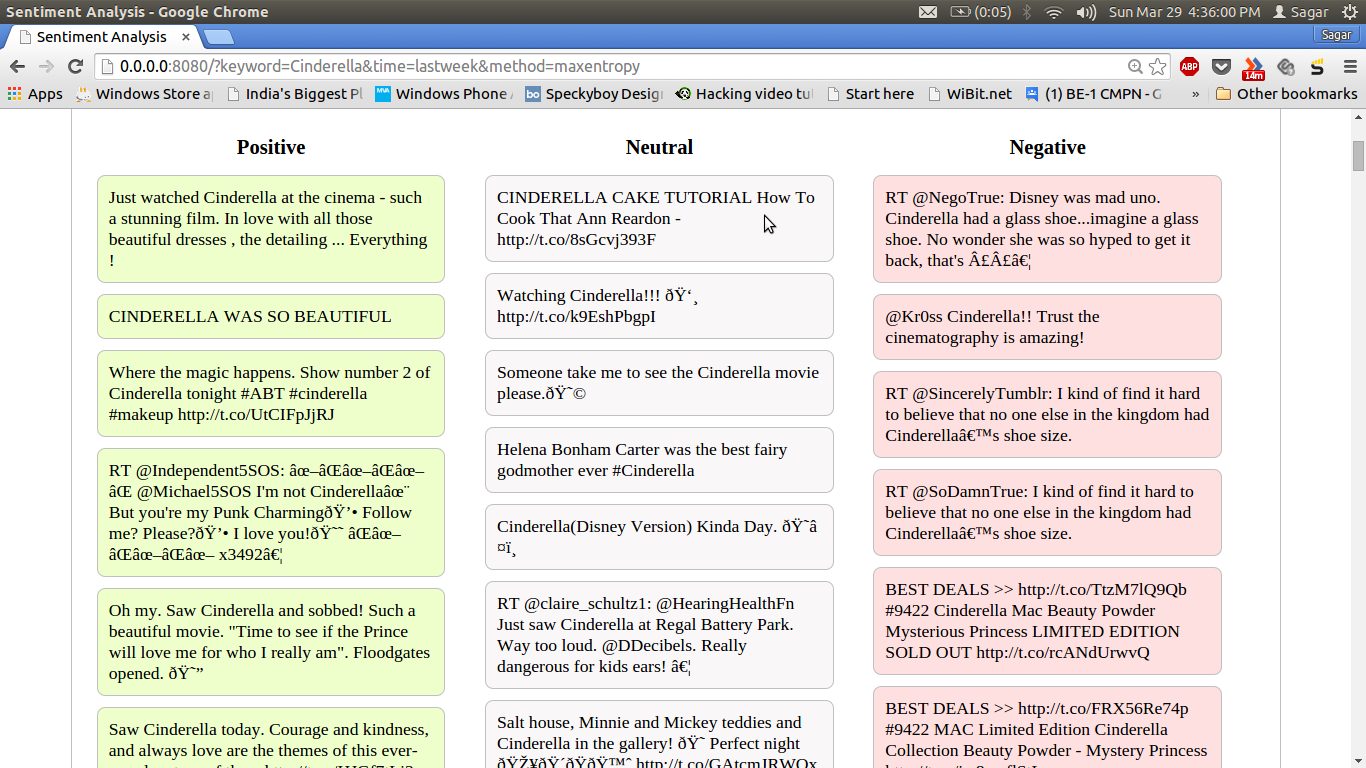
****

****

****

****

****

****

* + 1. **CONCLUSION**

We achieve our aim of analyzing sentiment of text data by using Naïve Bayes and Maximum Entropy classification algorithms over the twitter data. In the part of *natural language processing* we use unigram approach along with feature vector method, using whichwe get an accuracy of around 80%. This is really good considering the simplicity of the system. We have also included an extra class of *neutral* along with *positive* and *negative*; this is because the neutral class will classify the sentence which doesn’t have a particular opinion on the given subject, hence increasing accuracy of the system. The regular classification of positive, negative and neutral can be extended to include more emotions like happy, sad, angry, surprised, etc. This will be more helpful in understanding the actual emotion of the person/sentence. Our project can be further used in the fields of Artificial Intelligence and Speech Recognition which will help in future development of these fields and improve the accuracy of these fields.

1. **APPENDIX**

Sentiment Analysis

Sentiment Analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation, affective state (that is to say, the emotional state of the author when writing), or the intended emotional communication (that is to say, the emotional effect the author wishes to have on the reader).

Machine Learning

Machine learning is a [scientific discipline](http://en.wikipedia.org/wiki/Academic_disciplines) that explores the construction and study of [algorithms](http://en.wikipedia.org/wiki/Algorithm) that can [learn](http://en.wikipedia.org/wiki/Learning) from data. Such algorithms operate by building a [model](http://en.wikipedia.org/wiki/Mathematical_model) from example inputs and using that to make predictions or decisions, rather than following strictly static program instructions. Machine learning is closely related to and often overlaps with [computational statistics](http://en.wikipedia.org/wiki/Computational_statistics); a discipline that also specializes in prediction-making.

**Table 15.1 Information on Keywords**

|  |  |
| --- | --- |
| NLP | Natural Language Processing isconcerned with the interactions between [computers](http://en.wikipedia.org/wiki/Computer) and[human (natural) languages](http://en.wikipedia.org/wiki/Natural_language). As such, NLP is related to the area of [human–computer interaction](http://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction) and helps in understanding the languages and their meanings. |
| NLTK | Natural Language Toolkit 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. |
| Feature Vector | Feature Vector is an n-dimensional vector of numerical features that represent some object. Many algorithms in machine learning require a numerical representation of objects, since such representations facilitate processing and statistical analysis. |
| OAuth | OAuth provides client applications a 'secure delegated access' to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials |
| Twitter REST API | The REST APIs provide programmatic access to read and write Twitter data. Author a new Tweet, read author profile and follower data, and more. The REST API identifies Twitter applications and users using OAuth; responses are available in JSON. |
| Naive Bayes Classifier | Naive Bayes Classifier are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. |
| Maximum Entropy Classifier | Maximum Entropy Classifier is a [probability distribution](http://en.wikipedia.org/wiki/Probability_distribution) whose [entropy](http://en.wikipedia.org/wiki/Information_entropy) is at least as great as that of all other members of a specified class of distributions. |

* + 1. **REFERENCES**

1. Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002.
2. Sebastiani, Fabrizio. "Machine learning in automated text categorization." *ACM computing surveys (CSUR)* 34.1 (2002): 1-47.
3. Go, Alec, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision." *CS224N Project Report, Stanford* (2009): 1-12.
4. Colas, Fabrice, and Pavel Brazdil. "Comparison of SVM and some older classification algorithms in text classification tasks." *Artificial Intelligence in Theory and Practice*. Springer US, 2006. 169-178.
5. Joachims, Thorsten. *Text categorization with support vector machines: Learning with many relevant features*. Springer Berlin Heidelberg, 1998.
6. Dumais, Susan. "Using SVMs for text categorization." *IEEE Intelligent Systems*13.4 (1998): 21-23.
7. Nigam, Kamal, John Lafferty, and Andrew McCallum. "Using maximum entropy for text classification." *IJCAI-99 workshop on machine learning for information filtering*. Vol. 1. 1999.
8. G.Vinodhini and RM.Chandrasekaran. “Sentiment Analysis and Opinion Mining: A Survey” *IJARCSSE*Volume 2, Issue 6, June 2012.
9. http://www.nltk.org/book.
10. http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/.
11. http://www.cs.cornell.edu/people/pabo/movie-review-data.
12. http://www.cs.cornell.edu/people/pabo/movie-review-data/.

**16**. **ACKNOWLEDGEMENT**

On this occasion, of completing our project, we wish to express our immense gratitude to the range of people who provided invaluable support in the completion of our project **‘SENTIMENT ANALYSIS USING MACHINE LEARNING’**. Their guidance and encouragement has helped in making this project a success.

This work was influenced by countless individuals whom we were fortunate enough to meet during the project duration. While space does not permit us to acknowledge them all, we would remiss if we do not acknowledge the following individuals whose guidance, support and wisdom so greatly influenced this body of work. We are thankful to **Prof. Amruta Sankhe** for helping us, giving us good ideas for improving our work. Also, we are eager and glad to express our gratitude to the Head of the Computer Department **Prof. Mahendra Patil**, for his approval of this project. We are also thankful to him for providing us the needed assistance, detailed suggestion and also encouragement to do the project. We express our thanks to senior friends for extending their support.

We would like to deeply express our sincere gratitude to our respected principal **Dr. Shrikant Kallurkar** and the Atharva College of Engineering for providing such an ideal atmosphere to build up this project with well-equipped library and top class labs equipped with the latest technology.

Finally, we are extremely grateful to God for letting us get this idea for our project and making it possible for us to achieve its completion in time.

**Submitted by:**

Parikshit Hegde Gaurav Chavan Sagar Manjare

**17. PAPER PUBLICATION DETAILS**

*Paper Published in International Journal of Trends and Technology (IJETT) - Volume:15 No:6 - Sept, 2014*

*IJETT Journal Impact Factor = 2.695 Index Copernicus value = 4.15*

*“A Survey of Various Machine Learning Techniques for Text Classification”*