**ABSTRACT**

Now-a-days social networking sites are at the boom, so large amount of data is generated. Millions of people are sharing their views daily on micro blogging sites, since they contain short and simple expressions. We shall discuss about a paradigm to extract the sentiment from a famous micro blogging service, Twitter, where users tweet their opinions about different things. In this project, we will discuss the sentiment analysis of twitter dataset with data mining approach. An approach is introduced that automatically classifies the sentiments of Tweets taken from Twitter dataset by using different classifier algorithms. These tweets are classified as positive or negative along with the confidence. In this procedure of sentiment analysis, we will take the user input keyword and fetch the Live Twitter Data and classify them as positive and negative tweets and save them locally. The training and testing data consists of tweets which can be plain-text, acronyms and abbreviations.

**TABLE OF CONTENTS**

**DECLARATION**

**ACKNOWLEDGEMENT**

**ABSTRACT**

**1. INTRODUCTION** 1

1.1 Existing System 1

1.2 Problem Statement 1

1.3 Proposed System 1

1.4 Why Twitter 1

**2. LITERATURE SURVEY** 3

2.1 Sentiment Analysis 3

2.2 Python 4

2.3 NLTK 5

**3. MACHINE LEARNING CLASSIFIER ALGORITHMS**  6

# 3.1 Text Classification 6

3.1.1The Bag of Words Model 6

3.1.2 Tokenization 8

3.1.3 Stop Words 8

3.1.4 Stemming and Lemmatization 9

3.2 Naive-Bayes (NB) Classifier 10

3.3 Multinomial Naive-Bayes Classifier 13

3.4 Bernoulli Naive-Bayes Classifier 15

3.5 Logistic Regression Classifier 16

3.6 SVC (Support Vector Classifier 16

3.6.1 LinearSVC Classifier 16

3.6.2 NuSVC Classifier 17

3.7 User Defined Classification (Voted Classifier) 18

**4. FEASIBILITY STUDY**  20

4.1 Operational feasibility 20

4.2 Technical feasibility 20

4.3 Economical feasibility 20

**5. SOFTWARE REQUIREMENTS AND SPECIFICATIONS** 21

5.1 Introduction 21

5.1.1Purpose 21

5.1.2Scope 22

5.1.3Objective 22

5.2. Functional Requirements 22

5.3 Non Functional Requirements 23

5.3.1 Usability 23

5.3.2 Reliability 23

5.3.3 Performance 23

5.3.4 Supportability 23

5.3.5 Implementation 23

5.3.6 Interface 23

5.3.7 Packaging 23

5.3.8 Legal 24

5.4 Design Requirements 24

5.4.1 Hardware Requirements 24

5.4.2Software Requirements 24

**6. OBJECT-ORIENTED ANALYSIS** 25

6.1 Use Case Diagram 25

6.2 Activity Diagram 26

6.3 Sequence Diagram 27

6.4Collaboration Diagram 28

**7. DESIGN** 29

7.1 Object-Oriented Design 29

7.1.1 Class Diagram 29

7.1.2 Deployment Diagram 30

**8. SAMPLE SCREENS** 31

**9. SAMPLE CODE** 35

**10. TESTING** 39

10.1. Introduction 39

10.1.1 Unit Testing 39

10.1.2 Functional Testing 39

10.1.3 Stress Testing 39

10.1.4 Performance Testing 39

10.2 Test Report 40

**11. CONCLUSION** 41

**12. BIBILOGRAPHY** 42

#### LIST OF FIGURES

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **FIGURE NO** | **CAPTION** | **PAGE NO** |
| 1 | **Figure 1** | **Use Case Diagram** | **25** |
| 2 | **Figure 2** | **Activity Diagram** | **26** |
| 3 | **Figure 3** | **Sequence Diagram** | **27** |
| 4 | **Figure 4** | **Collaboration Diagram** | **28** |
| 5 | **Figure 5** | **Class Diagram** | **29** |
| 8 | **Figure 6** | **Deployment Diagram** | **30** |

#### LIST OF TABLES

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **TABLE NO** | **TABLE CAPTION** | **PAGE NO** |
| 1 | **Table 1** | **Machine Learning Algorithm** | **7** |
| 2 | **Table 2** | **Example of tokenization.** | **8** |
| 3 | **Table 3** | **Example of stop word removal.** | **9** |
| 4 | **Table 4** | **Example of Porter Stemming.** | **9** |
| 5 | **Table 5** | **Example of Lemmatization.** | **10** |

**LIST OF SAMPLE SCREENS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE OF THE SCREEN** | **PAGE NO** |
| **Screen 1** | **Program started and loading is done** | **31** |
| **Screen 2** | **User is prompted to give filename where to save data offline and asks for search keyword to start live streaming of tweets and analyzing them.** | **32** |
| **Screen 3** | **Live streaming of tweets started and each tweet is given a sentiment values positive or negative** | **33** |
| **Screen 4** | **Live streaming and live graphing of sentiment analysis.** | **34** |

**INTRODUCTION**

**1. INTRODUCTION**

**1.1Existing System**

Present system in use performs the sentiment analysis on offline data. Also it is difficult for a common user to understand the end results of the sentiment. To overcome these difficulties, we came up with a project which is easy to use and produce simplified results for user understandability. Moreover we targeted the popular micro blogging site – Twitter for best analysis.

**1.2 Problem Statement**

Given a message, determine itspolarity i.e., identify whether the message is positive or negative message. For messages conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen.

For each classified tweet generate a confidence score to determine the accuracy of the system.

**1.3 Proposed System**

The present proposed system targets day-to-day live streaming data so as to improve the accuracy of the sentiment about updated data. In our case, we targeted Twitter as our dataset. We take the live streaming tweets from twitter data and perform sentiment analysis.

This system takes the user input (a keyword) and fetch the live streaming twitter data. It performs the sentiment analysis on each and every tweet and generates sentiment value.

**1.4 Why Twitter?**

1) Popular micro blogging site.

2) 240+ million active users.

3) 500 million tweets are generated every day.

1

4) Audience varies from common man to celebrities.

5) Tweets are smaller in length and hence unambiguous.

6) Users often discuss current affairs and share personal views on various subjects.

2

**LITERATURE SURVEY**

**2. LITERATURE STUDY**

**2.1Sentiment Analysis**

Sentiment Analysis is process of collecting and analyzing data based upon the person feelings, reviews and thoughts. Sentimental analysis often called as opinion mining as it mines the important feature from people opinions. Sentimental Analysis is done by using various machine learning techniques, statistical models and Natural Language Processing (NLP) for the extraction of feature from a large data. Sentiment Analysis can be done at document, phrase and sentence level. In document level, summary of the entire document is taken first and then it is analyze whether the sentiment is positive, negative or neutral. In phrase level, analysis of phrases in a sentence is taken in account to check the polarity.

In Sentence level, each sentence is classified in a particular class to provide the sentiment. Sentimental Analysis has various applications. It is used to generate opinions for people of social media by analyzing their feelings or thoughts which they provide in form of text. Sentiment Analysis is domain centered, i.e. results of one domain cannot be applied to other domain. Sentimental Analysis is used in many real life scenarios, to get reviews about any product or movies, to get the financial report of any company, for predictions or marketing.

Twitter is a micro blogging platform where anyone can read or write short form of message which is called tweets. The amount of data accumulated on twitter is very huge. This data is unstructured and written in natural language. Twitter Sentimental Analysis is the process of accessing tweets for a particular topic and predicts the sentiment of these tweets as positive or negative with the help of different machine learning algorithms.

1. With the advancement of web technology and its growth, there is a huge volume of data present in the web for internet users and a lot of data is generated too.
2. Internet has become a platform for online learning, exchanging ideas and sharing opinions. Social networking sites like Twitter, Facebook, Google+ are rapidly gaining
3. popularity as they allow people to share and express their views about topics, have

3

discussion with different communities, or post messages across the world.

1. There has been lot of work in the field of sentiment analysis of twitter data. This project focuses mainly on sentiment analysis of twitter data which is helpful to analyse the information in the tweets where opinions are highly unstructured, heterogeneous and are either positive or negative, or neutral in some cases.
2. Sentiment analysis can be defined as a process that automates mining of attitudes, opinions, views and emotions from text, speech, tweets and database sources through Natural Language Processing (NLP).
3. Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative" or "neutral". It is also referred as subjectivity analysis, opinion mining, and appraisal extraction.
4. The words opinion, sentiment, view and belief are used interchangeably but there are differences between them.

**Opinion:**A conclusion opens to dispute (because different experts have different opinions)

**View:** subjective opinion

**Belief:** deliberate acceptance and intellectual assent

**Sentiment:** opinion representing one's feelings

**2.2Python**

Python is a high level, dynamic programming language which is used for this thesis.Python3.4 version was used as it is a mature, versatile and robust programming version. It is an interpreted language which makes the testing and debugging extremely quickly as there is no compilation step. There are extensive open source libraries available for this version of python and a large community of users. Python is simple yet powerful, interpreted and dynamic programming language,which is well known for its functionality of processing natural language data, i.e.spoken English using NLTK. Other high level programming languages such as ‘R’and ‘Mat

4

lab’ were considered because they have many benefits such as ease of use but they do not offer the same flexibility and freedom that Python can deliver. Python (Python 3.4) language is used as a programming platform for this twitter sentiment analysis.

1. Python is chosen because it is the most popular open-source application with huge community support and it is best for engineers and programmers. Whereas as R-tool is best for scientists and researchers.
2. Moreover Python provides us with Natural Language Toolkit (NLTK) which makes our coding lot more easier on sentiment analysis.
3. Python can also be used for scripting a website to display the results on web pages.
4. Python is supported by vast number of modules and we can import any module we want which in turn reduces our tedious work.
5. Scipy, Matpotlib and Pygame modules are used so as to generate graphs and calculations.

**2.3 NLTK**

Natural Language Toolkit (NLTK) is library in Python, which provides a base for building programs and classification of data. NLTK is a collection of resources for Python that can be used for text processing, classification, tagging and tokenization. This toolbox plays a key role in transforming the text data in the tweets into a format that can be used to extract sentiment from them.NLTK provides various functions which are used in pre-processing of data so that data available from twitter become fit for mining and extracting features.

NLTK support various machine learning algorithms which are used for training classifier and to calculate the accuracy of different classifier. In our thesis we use Python as our base programming language which is used for writing code snippets. NLTK is a library of Python which plays a very important role in converting natural language text to a sentiment either positive or negative. NLTK also provides different sets of data which are used for training classifiers. These datasets are structured and stored in library of NLTK, which can be accessed easily with the help of Python.

5

**MACHINE LEARNING CLASSIFIER ALGORITHMS**

**3. MACHINE LEARNING CLASSIFIER ALGORITHMS**

Machine learning is a technique whose task is to deduce a function from tagged training samples. The training samples for supervised learningconsist of large set of examples for a particular topic. In supervised learning, every example training data comes in a pair of input (vector quantity) and output value (desired result). These algorithms analyze data and generate an output function, which is used to mapped new data sets to respective classes. Different machine learning classifiers which we are going to use to build our classifier are:

# Text Classification

* + - 1. Naive-Bayes Classifier
      2. MultinomialNaive-Bayes Classifier
      3. BernoulliNaive-Bayes Classifier
      4. Logistic Regression Classifier
      5. SVC (Support Vector Classifier): LinearSVC and NuSVC
      6. User Defined Classification Algorithm (Voted Classifier)

# 3.1Text Classification

### 3.1.1The Bag of Words Model

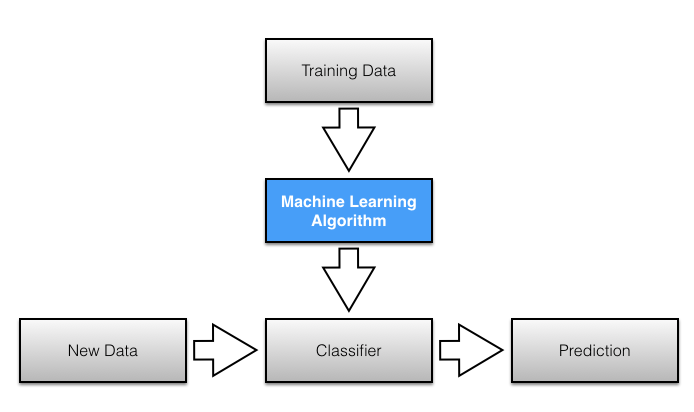
One of the most important sub-tasks in pattern classification arefeature extraction and selection; the three main criteria of good features are listed below:

1. **Salient.** The features are important and meaningful with respect to the problem domain.
2. **Invariant.** Invariance is often described in context of image classification: The features are insusceptible to distortion, scaling, orientation, etc. A nice example is given by C. Yao and others in Rotation-Invariant Features for Multi-Oriented Text Detection in Natural Images .
3. **Discriminatory.** The selected features bear enough information to distinguish well

6

between patterns when used to train the classifier.

**Table 1:** Machine Learning Algorithm



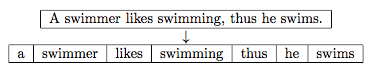
Prior to fitting the model and using machine learning algorithms for training, we need to think about how to best represent a text document as a feature vector. A commonly used model in Natural Language Processing is the so-called bag of words model. The idea behind this model really is as simple as it sounds. First comes the creation of the vocabulary — the collection of all different words that occur in the training set and each word is associated with a count of how it occurs. This vocabulary can be understood as a set of non-redundant items where the order doesn’t matter.

7

#### 3.1.2 Tokenization

Tokenization describes the general process of breaking down a text corpus into individual elements that serve as input for various natural language processing algorithms. Usually, tokenization is accompanied by other optional processing steps, such as the removal of stop words and punctuation characters, stemming or lemmatizing, and the construction of n-grams. Below is an example of a simple but typical tokenization step that splits a sentence into individual words, removes punctuation, and converts all letters to lowercase.

**Table 2.**Example of tokenization.

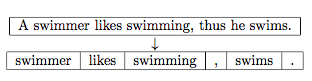


#### 3.1.3 Stop Words

Stop words are words that are particularly common in a text corpus and thus considered as rather un-informative (e.g., words such as so, and, or, the,”). One approach to stop word removal is to search against a language-specific stop word dictionary. An alternative approach is to create a stop list by sorting all words in the entire text corpus by frequency. The stop list — after conversion into a set of non-redundant words — is then used to remove all those words from the input documents that are ranked among the top n words in this stop list.

**8**

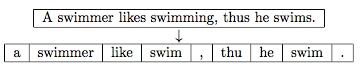
**Table 3.**Example of stop word removal.



### 3.1.4 Stemming and Lemmatization

Stemming describes the process of transforming a word into its root form. The original stemming algorithm was developed my Martin F. Porter in 1979 and is hence known as Porter stemmer .

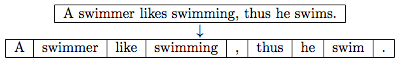
**Table 4.**Example of Porter Stemming.



Stemming can create non-real words, such as “thu” in the example above. In contrast to stemming, lemmatization aims to obtain the canonical (grammatically correct) forms of the words, the so-called lemmas. Lemmatization is computationally more difficult and expensive than stemming, and in practice, both stemming and lemmatization have little impact on the performance of text classification.

**9**

**Table 5.**Example of Lemmatization.



**3.2 Naive-Bayes Classifier**

It is a classification technique based on [Bayes’ Theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter.

Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

10

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/09/Bayes_rule-300x172.png)

**Advantages**

1. It is easy and fast to predict class of test data set. It also perform well in multi class prediction
2. When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
3. It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

**Disadvantages**

1. If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a

11

1. prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
2. On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.
3. Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

**Applications of Naive Bayes Algorithms**

1. **Real time Prediction:**Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
2. **Multi class Prediction:**This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
3. **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
4. **Recommendation System:**Naive Bayes Classifier and Collaborative together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

**Basic Model using Naive Bayes in Python**

Again, scikit learn (python library) will help here to build a Naive Bayes model in Python. There are three types of Naive Bayes model under scikit learn library:

[**Gaussian:**](http://scikit-learn.org/stable/modules/naive_bayes.html)It is used in classification and it assumes that features follow a normal distribution.

12

[**Multinomial**](http://scikit-learn.org/stable/modules/naive_bayes.html)**:**It is used for discrete counts. For example, let’s say, we have a text classification problem. Here we can consider Bernoulli trials which is one step further and instead of “word occurring in the document”, we have “count how often word occurs in the document”, you can think of it as “number of times outcome number x\_i is observed over the n trials”.

[**Bernoulli**](http://scikit-learn.org/stable/modules/naive_bayes.html)**:**The binomial model is useful if your feature vectors are binary (i.e. zeros and ones). One application would be text classification with ‘bag of words’ model where the 1s & 0s are “word occurs in the document” and “word does not occur in the document” respectively.

**3.3 Multinomial Naive-Bayes Classifier**

The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.It is also a Naive Bayes algorithm generally used in machine learning module.It comes within Scikit\_Learn module.

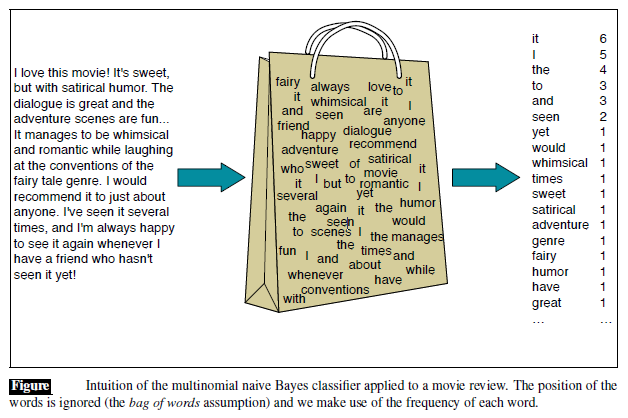
On testing it on a sample it gave 72% accuracy.

We introduce the multinomial naive Bayes classifier, so called because it is a Bayesian classifier that makes a simplifying (naive) assumption about how the features interact.

The intuition of the classifier is shown in Fig. We represent a text document as if it were a bag-of-words, that is, an unordered set of words with their position ignored, keeping only their frequency in the document.

In the example in the figure, instead of representing the word order in all the phrases like “I love this movie” and“I would recommend it”, we simply note that the word I occurred 5 times in theentire excerpt, the word it 6 times, the words love, recommend, and movie once, andso on.

13

****

[MultinomialNB](http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB) implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although tf-idf vectors are also known to work well in practice).

The distribution is parametrized by vectors for each class, where is the number of features (in text classification, the size of the vocabulary) and is the probability of feature appearing in a sample belonging to class.

14

The parameter is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting: It is simply an advanced version of Naïve-Bayes algorithm. It is based on the following formula.



where is the number of times feature appears in a sample of class’ y’ in the training set T , andis the total count of all features for class ‘y’.

The smoothing priors accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting is called Laplace smoothing, while is called Lidstone smoothing.

**3.4 Bernoulli Naive-Bayes Classifier**

BernoulliNB implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors; if handed any other kind of data, a BernoulliNB instance may binarize its input (depending on the binarize parameter).It comes with Scikit\_Learn module.

On testing it on a sample it gave 70% accuracy

P(x_i \mid y) = P(i \mid y) x_i + (1 - P(i \mid y)) (1 - x_i)

15

The decision rule for Bernoulli naive Bayes is based onwhich differs from multinomial NB’s rule in that it explicitly penalizes the non-occurrence of a feature ithat is an indicator for class y, where the multinomial variant would simply ignore a non-occurring feature.

In the case of text classification, word occurrence vectors (rather than word count vectors) may be used to train and use this classifier. BernoulliNB might perform better on some datasets, especially those with shorter documents.

**3.5Logistic Regression Classifier**

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the ‘multi\_class’ option is set to ‘ovr’, and uses the cross- entropy loss if the ‘multi\_class’ option is set to ‘multinomial’. It can be given with many customizable parameters, but we are using default parameters in our case.

It comes within Scikit\_Learn Linear-model module.

On testing it on a sample it gave 71% accuracy.

**3.6 SVC (Support Vector Classifier)**

**3.6.1 LinearSVC Classifier Algorithm**

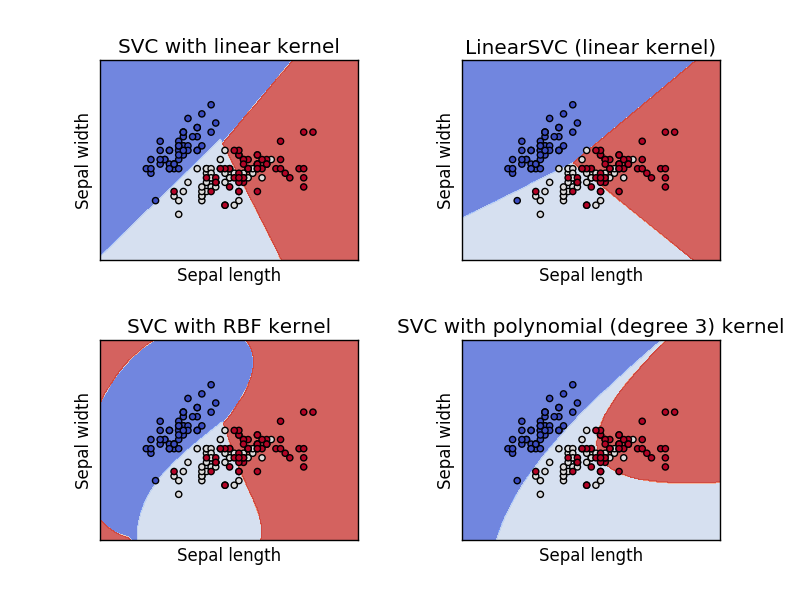
Linear Support Vector Classification.Similar to SVC with parameter kernel=’linear’, but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples.This class supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.

Default parameters are used.

It comes within Scikit\_Learn SVM module.

On testing it on a sample it gave 69% accuracy.

16



**3.6.2NuSVC Classifier Algorithm**

Nu-Support Vector Classification.Similar to SVC but uses a parameter to control the number of support vectors.Default parameters are used.It comes within Scikit\_Learn SVM module.

On testing it on a sample it gave 73% accuracy.

**Advantages**

1. It works really well with clear margin of separation
2. It is effective in high dimensional spaces.
3. It is effective in cases where number of dimensions is greater than the number of samples.
4. It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

**Disadvantages**

1. It doesn’t perform well, when we have large data set because the required training time is

17

1. higher
2. It also doesn’t perform very well, when the data set has more noise i.e. target classes are overlapping
3. SVM doesn’t directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library.

**3.7User Defined Classification Algorithm (Voted Classifier)**

This classifier is used to improve the accuracy of the classifications. What it basically does is, it takes all the accuracies of the above mentioned 5 pre-defined classifiers as input and constructs a voting system . Basic on this voting system, it’s accuracy is calculated. So that we get accurate classifications.A voted\_classifier class is defined to calculate the accuracy of this classifier.

This Class is Inherited from nltk.classify module.

Procedure of voting system:

1. Suppose take a word called “bad”. If all the 5 algorithms vote the word as negative word then the accuracy of the voted classifier will be 5/5 = 1 i.e. 100% accurate.
2. If we get a score of 3/5 the possibly its accuracy will be 60% accurate.
3. Similarly we get 20%, 40% and 80% accuracies.
4. However the words with accuracy below and equal to 80% are eliminated in the process.
5. This voted classifier is trained on 1900 random words and tested on 100 random words initially.
6. Size of the samples are varied every time we train the system.
7. More the system is trained the better is the accuracy.

18

1. Various training samples are changed during the training process.
2. Initially the system is trained on movie reviews offline data set.
3. This data set contains 10000 positive and 10000 negative sentences.
4. This classifier is trained on this data set for better accuracy.
5. Then it is further trained on offline twitter tweets data set.
6. Once this is done, it is applied for the Analysis on of LIVE TWEETS on TWITTER .

19

**FEASIBILITY STUDY**

**4. FEASIBILITY STUDY**

It is necessary and discreet to evaluate the feasibility of a project at the earliest possible time. There may be different ways of checking whether a system is feasible or not. The following feasibility studies were performed to gauge the feasibility of the system.Feasibility study can be divided into three basic forms as follows.

**4.1 Operational feasibility**

It refers to the feasibility of the product to be operational. Some products may work very well at design and implementation but may fail in the real time environment. It includes the study of additional human resource required and their technical expertise.

Our application is very easy to be operated. A common man with basic computer knowledge can use our application. User only has to give his input, nothing else.

**4.2 Technical feasibility**

This test includes a study of function, performance and constraints that may affect the ability to achieve an acceptable system. The technical feasibility of the project is explained as follows:

1. The project has been done with a very easy software technology.
2. It is user friendly and easy to understand and so it is technically feasible.

**4.3 Economical feasibility**

It refers to the benefits or outcomes we are deriving from the product as compared to the total cost we are spending for developing the product. User has to spend money only to buy our application, apart from that he need not buy anything else. However all the additional requirements for our project is completely free to install from internet.

20

**SOFTWARE REQUIREMENTS AND SPECIFICATIONS**

**5. SOFTWARE REQUIREMENTS AND SPECIFICATIONS**

**5.1 Introduction**

Now-a-days social networking sites are at the boom, so large amount of data is generated. Millions of people are sharing their views daily on micro blogging sites, since they contain short and simple expressions. We shall discuss about a paradigm to extract the sentiment from a famous micro blogging service, Twitter, where users tweet their opinions about different things. In this project, we will discuss the sentiment analysis of twitter dataset with data mining approach. An approach is introduced that automatically classifies the sentiments of Tweets taken from Twitter dataset. These messages or tweets are classified as positive, negative or neutral. In this procedure of sentiment analysis, we will take the Twitter dataset as input and train the system to understand the positive, neutral and negative tweets and cluster them as individual data chunks. The training data consists of tweets which can be plain-text, acronyms, emoticons and abbreviations. This is very useful for the companies who want to know the feedback about their product brands and customer purchase product details which in turn is helpful for them to develop and gain profits.

**5.1.1 Purpose**

The main purpose of our project is to identify the sentiment about a topic, person or an object with respect to Twitter micro blogging site.

Also to prove useful for the companies who want to know the feedback about their product brands and customer purchase product details which in turn is helpful for them to develop and gain profits.

Sentiment analysis (SA)tells user whether the information about the product is satisfactory or not before they buy it. Marketers and firms use this analysis data to understand about their products or services in such a way that it can be offered as per the user's requirements.

21

**5.1.2 Scope**

The main scope of our project is to analyze dataset of Twitter, so as to identify the valuable opinions of users regarding particular topic, person or organization.

Our project can be further expanded by including detailed analysis of tweets i.e identifying time, place and user details of each tweet.

And also our project’s scope can be expanded to various other social networking sites such as Facebook, Linkedin etc.

**5.1.3 Objectives**

1. Ask for user input (a keyword to search).
2. Fetching Live Twitter Data basing on the user input.
3. Classification of each Live tweet Positive or Negative.
4. Generate a confidence of polarity for each tweet.
5. Plot the live graph for better understandability of sentiment analysis.
6. Save the tweets offline for day to day analysis.

**5.2. Functional Requirements**

Following functional modules are required:

* 1. NLTK moduleis required support natural language processing functions and classifiers.
  2. Scipy is required for scientific calculations in classifiers.
  3. Numpy is for numerical calculations in classifiers.
  4. Matplotlib for plotting graph basing on sentiments of tweets.
  5. Scikit\_learn module for various machine learning classifiers.
  6. Tweets for live twitter connection.

22

**5.3 Non Functional Requirements**

**5.3.1 Usability:**

The User can operate, give inputs and interpret outputs of this system. Any User who knows how to use a computer should be able to use this system.

**5.3.2 Reliability:**

This system has the ability to perform all the required functions under stated conditions for a specified period of time. Processing depends on the user’s pc specifications.

**5.3.3 Performance:**

This system can react quickly to a user input. The system can accomplish any work within a specified amount of time.

Performance highly depends on user’s electronic device specifications.

**5.3.4 Supportability:**

Thisapplication can be easily transferred from one hardware or software environment to another.

**5.3.5 Implementation:**

We require one computer which acts as the Server and any number of computers which acts as clients can connect to it with a Web Browser. The Sever machine should be installed with web server that supports python and NLTK platform. The client system should be equipped with a browser like the Internet Explorer.

**5.3.6 Interface:**

Simple command line interface with detailed description is maintained.

**5.3.7 Packaging:**

Application along with the User manual of installation and other requirements. Also includes supporting software and their download links.

23

**5.3.8 Legal:**

All the requirements concerned with licensing, regulation and certification issues of the system are met.

**5.4 Design Requirements**

It consists of hardware and software requirements.

**5.4.1 Hardware Requirements**

1. 1 GB ram or higher is preferable.

2. Stable internet connection.

3. Amd or Intel i3 processor or higher is preferable.

4. Considerable amount of hard disk space for storing tweets offline

**5.4.2 Software Requirements**

1. Python 3.2 or above ( 32bit / 64bit).

2. NLTK full toolkit.

3. Scipy, Matplotlib, Pygame modules.

4. Windows, MacOS or Linux operating system.

5. User should have a twitter account.

24

**OBJECT ORIENTED ANALYSIS**

**6. OBJECT-ORIENTED ANALYSIS**

Object-oriented analysis is concerned with modeling the application domain. The application domain represents all aspects of the user’s problem. This includes the physical environment, the users and other people, their work processes and so on.

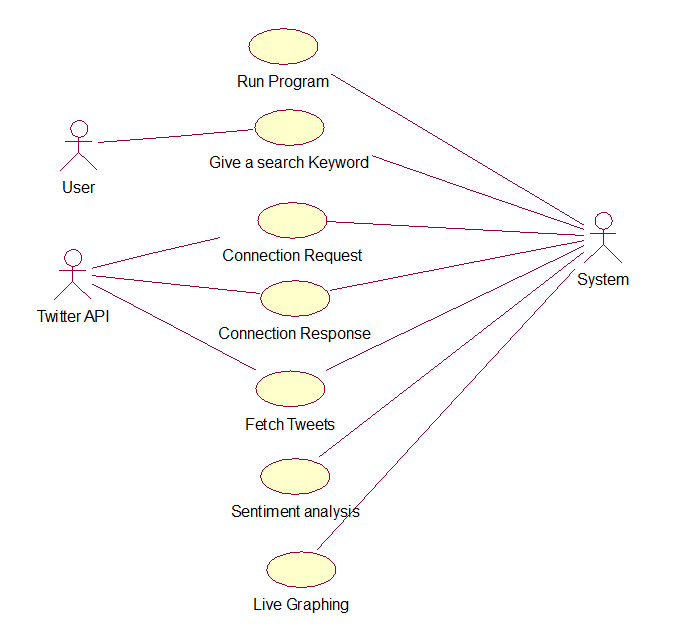
**6.1 Use Case Diagram**

Use case diagrams are used to gather the requirements of a system including internal and external influences. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

The purposes of use case diagrams can be as follows:

1. Used to gather requirements of a system.
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Show the interacting among the requirements are actors.

**Figure 1:**



25

**6.2 Activity Diagram**

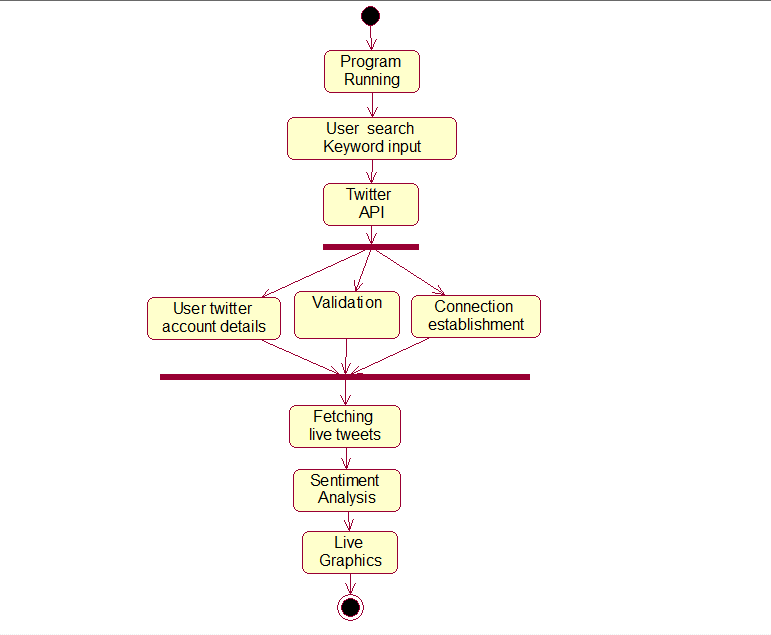
Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system.

So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of control by using different elements like fork, join etc.

The purposes can be described as:

1. Draw the activity flow of a system.
2. Describe the sequence from one activity to another.
3. Describe the parallel, branched and concurrent flow of the system

**Figure 2:**



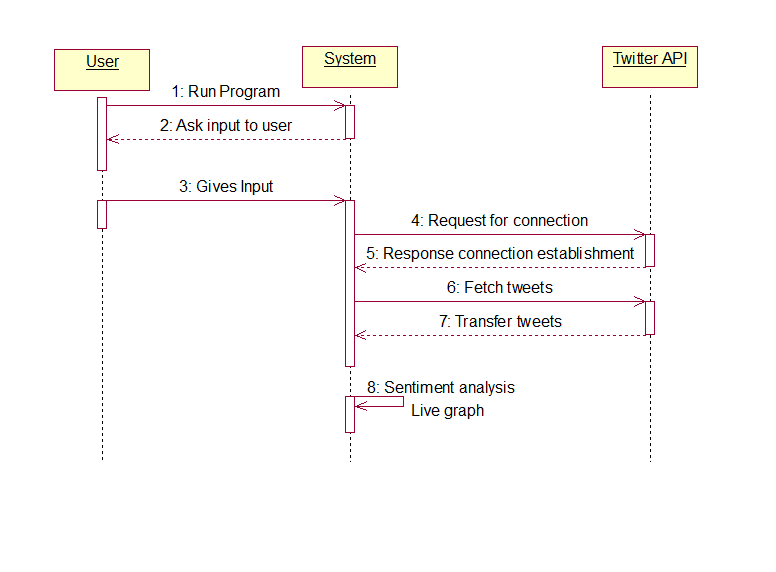
26

**6.3 Sequence Diagram**

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order.

A Sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

**Figure 3:** 

27

**Description for sequence diagram**

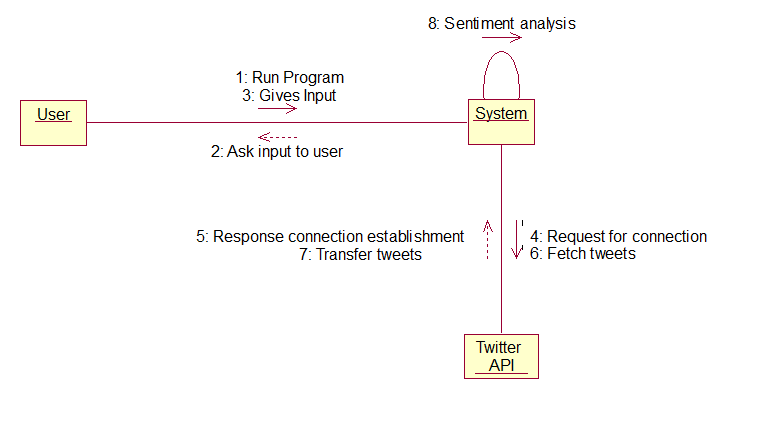
1. Training data send reviews with result for extraction.
2. Extraction sends review to classifier.
3. Testing data send new review with result for extraction.
4. Extraction sends new review to classifier.
5. Classifier sends opinion type of review (positive or negative or neutral) to feature extraction and training data.

**6.4 Collaboration Diagram**

It shows the object organization as shown below. Here in collaboration diagram the method call sequence is indicated by some numbering technique as shown .The number indicates how the methods are called one after another.

The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

**Figure 4:**

****

28

**DESIGN**

**7. DESIGN**

**7.1 Object-Oriented Design**

Object-oriented design is concerned with modeling the solution domain. The solution domain is a modeling space of all possible systems. Modeling the solution domain represents the system design and object design activities of the development process.

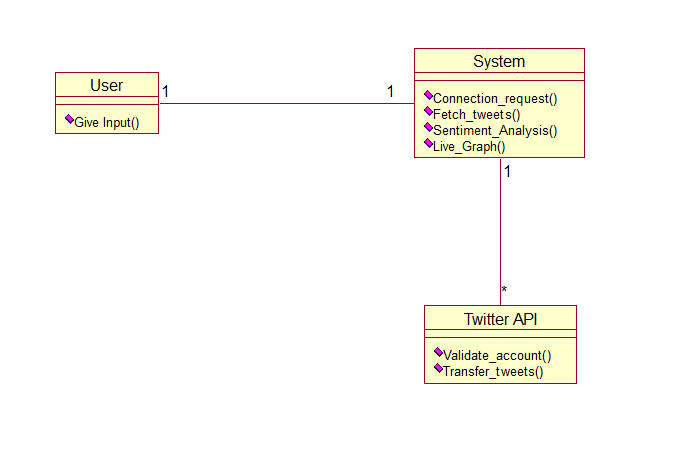
**7.1.1 Class Diagram**

Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system.

The purpose of the class diagram can be summarized as:

1. Analysis and design of the static view of an application.
2. Describe responsibilities of a system.
3. Base for component and deployment diagrams.
4. Forward and reverse engineering.

**Figure 5:**



29

**7.1.2 Deployment Diagram**

The UML deployment diagram shows the physical architecture of a computer-based system. It can depict the computers, show their connections with one another, and show the software that sits on each machine. Each computer is represented as a cube, with interconnections between computers drawn as lines connecting the cubes.

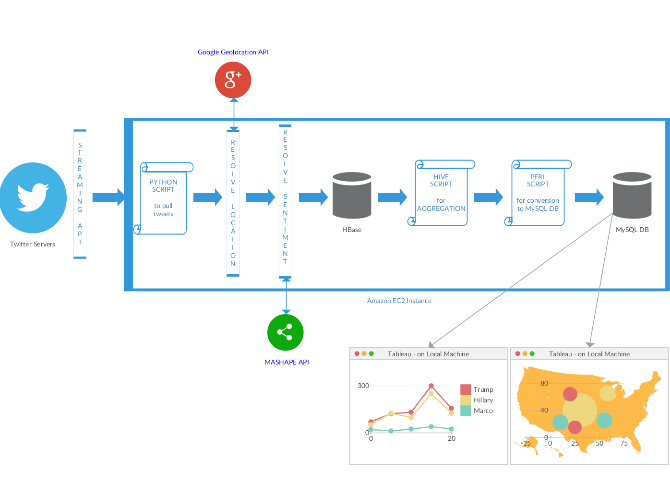
Deployment diagrams are used to depict the relation among run-time components and hardware nodes. Components are self-contained entities that provide services to other components or actors.

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

1. The purpose of deployment diagrams can be described as:
2. Visualize hardware topology of a system.
3. Describe the hardware components used to deploy software components.

Describe runtime processing nodes

**Figure 6:**

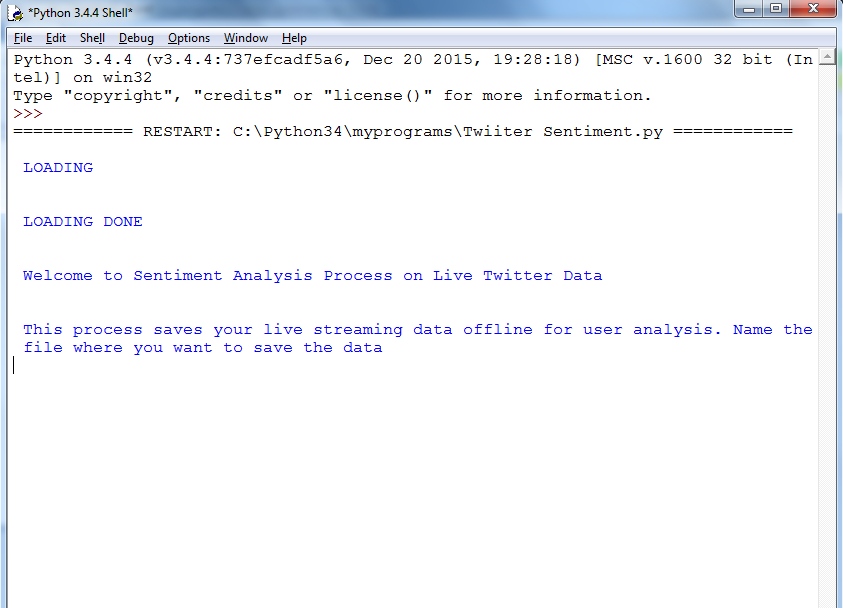
****

30

**SAMPLE SCREENS**

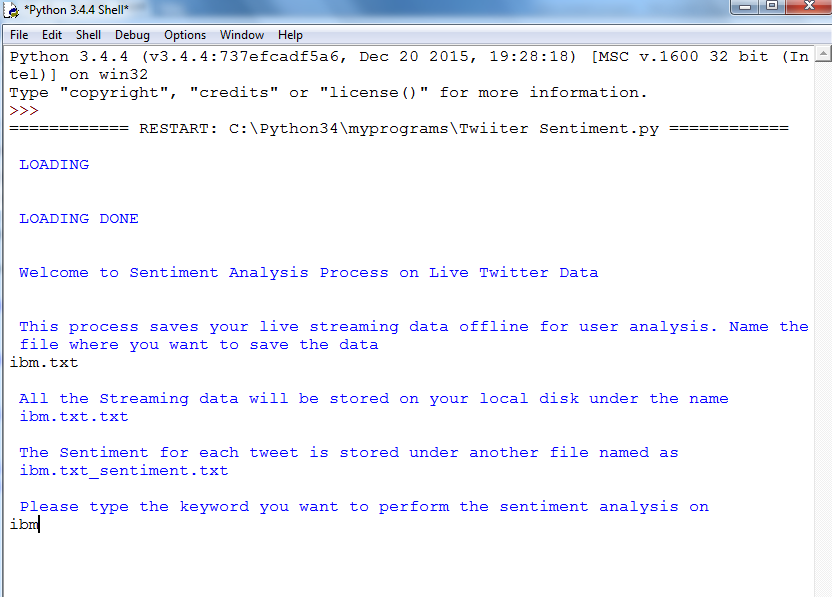
**8. SAMPLE SCREENS**

**1.Program started and loading is done:**



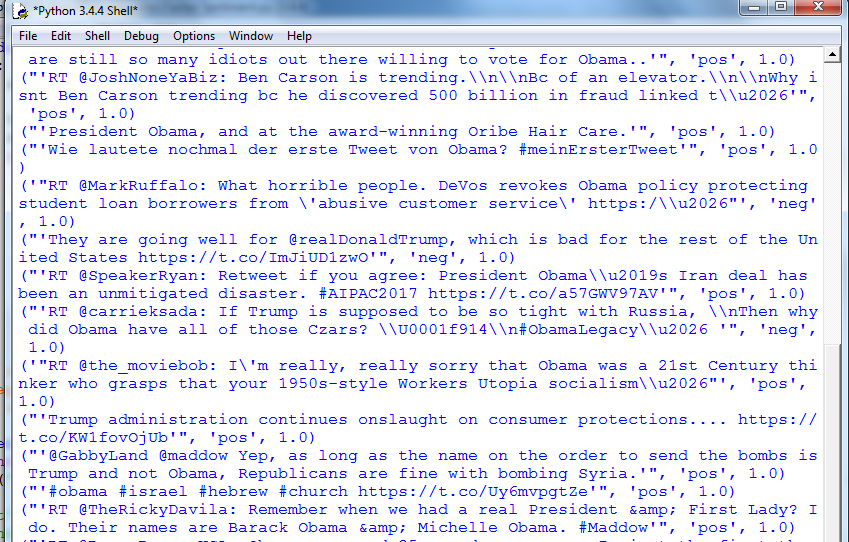
31

**2) User is prompted to give filename where to save data offline and asks for search keyword to start live streaming of tweets and analyzing them.**

****

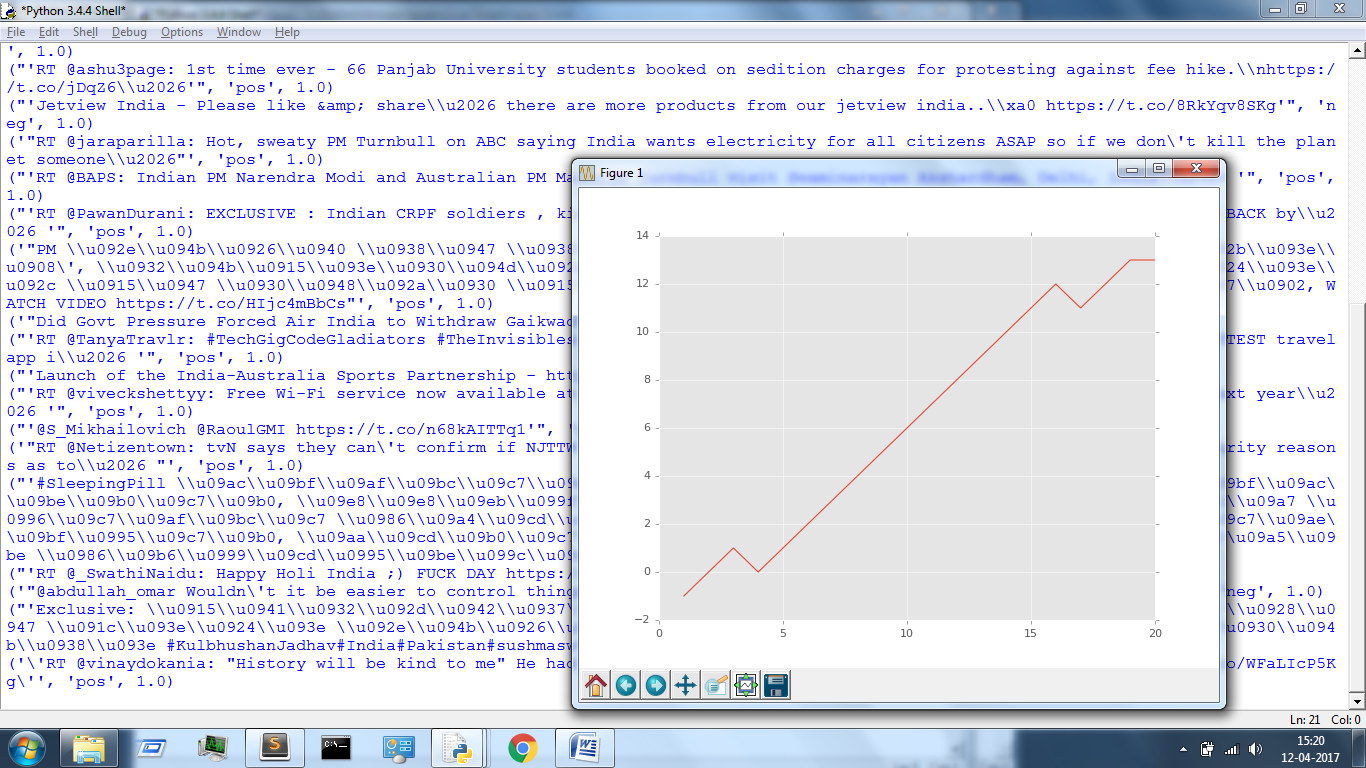
32

**3) Live streaming of tweets started and each tweet is given a sentiment values positive or negative**

****

33

**4) Live streaming and live graphing of sentiment analysis.**

****

34

**SAMPLE CODE**

**9. SAMPLE CODE**

**1. Sample code twitter API connection**

**Code:**

fromtweepy import Stream

fromtweepy import OAuthHandler

fromtweepy.streaming import StreamListener

importjson

importsentiment\_mod as s

#consumer key, consumer secret, access token, access secret.

ckey= "4AXdoe6af8pbNdOj2biewWOAW"

csecret= "Emm0Vp6Wwj7fUo0qL5QAc83DZPar8p1HKIZNbIQg290D5SYQpT"

atoken= "758300058450141184-Zj90kbnZiPvFnkJ2azIUNquQfXwaUZQ"

asecret= "7M3RAxETmyUjRw6IkwO6wX7UxMGlDtRf4ddKgrMjjHvXK"

print("\n Welcome to Sentiment Analysis Process on Live Twitter Data \n")

class listener(StreamListener):

defon\_data(self, data):

try:

all\_data = json.loads(data)

tweet = acai(all\_data["text"])

sentiment\_value, confidence = s.sentiment(tweet)

print((tweet, sentiment\_value, confidence))

35

if confidence\*100 >= 80:

output = open(fof1, "a")

outputfile = open(fof1, "a")

output. Write(sentiment value)

outputfile.write(tweet)

output. Write("\n")

outputfile.write("\n")

output.close()

outputfile.close()

return True

except:

return True

defon\_error(self, status):

print (status)

fof = input("\n This process saves your live streaming data offline for user analysis. Name the file where you want to save the data\n")

fof1=fof

fof += '.txt'

fof1 +='\_sentiment.txt'

print("\n All the Streaming data will be stored on your local disk under the name \n",fof)

print("\n The Sentiment for each tweet is stored under another file named as \n",fof1)

search = input("\n Please type the keyword you want to perform the sentiment analysis on \n")

auth = OAuthHandler(ckey, csecret)

36

auth.set\_access\_token(atoken, asecret)

twitterStream = Stream(auth, listener())

twitterStream.filter(track=[search])

**2.Live Graphing**

**Code:**

importmatplotlib.pyplot as plt

importmatplotlib.animation as animation

frommatplotlib import style

import time

##style.use('fivethirtyeight')

style.use("ggplot")

fig = plt.figure()

ax1 = fig.add\_subplot(1,1,1)

def animate(i):

pullData = open('twitter-out.txt','r').read()

lines = pullData.split('\n')

xar = []

yar = []

x = 0

37

y = 0

for l in lines:

x += 1

if "pos" in l:

y += 1

elif "neg" in l:

y -= 1

xar.append(x)

yar.append(y)

ax1.clear()

ax1.plot(xar, yar)

ani = animation.FuncAnimation(fig, animate, interval=1000)

plt.show()

38

**TESTING**

**10. TESTING**

**10.1 Types of testing**

There are many types of testing like

* Unit Testing
* Functional Testing
* Stress Testing
* Performance Testing

**10.1.1 Unit Testing**

Unit testing is the testing of an individual unit or group of related units.

Each and every unit of our application is tested.

Units like sentiment\_analysis unit, twitter\_connection unit, live\_graphing units are thoroughly tested.

**10.1.2 Functional Testing**

Functional testing is the testing to ensure that the specified functionality required in the system requirements works.

Each module is thoroughly tested for it’s functionality.

**10.1.3 Stress Testing**

Stress testing is the testing to evaluate how system behaves under unfavorable conditions.

On stress testing, we realized that our present system is able to stream 1500 tweets on an average.

**10.1.4 Performance Testing**

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements.

Our present system with 4GB RAM capacity is able to give decent performance of streaming live data. However it undergoes degradation after streaming average of 1300 tweets.

39

**10.2 Test Report**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Test Case** | **Input** | **Expected Output** | **Output** | **Result** |
| 1. | Positive sentiment test | Positive sentence | The sentiment value should return as pos | Sentiment value pos is returned | Pass |
| 2. | Negative sentiment test | Negative sentence | The sentiment value should return as neg | Sentiment value neg is returned | Pass |
| 3. | Looking for the confidence value | Few sentences are passed for analysis | Confidence value for each sentence should be generated. | Confidence value is generated for each tweet. | Pass |
| 4. | Live plotting of the graph for sentiment values | A file having a set of sentiment values | Plot the pos and neg graph accordingly | Pos and neg graph is plotted | Pass |

40

**CONCLUSION**

**11. CONCLUSION**

We conclude by saying that, our project can be used by any company or industry to conduct a survey about their products so as to determine the statistics of their business.

Customer analysis can be done to improvise their business.

Can be applicable to movie Review-related websites such as movie reviews, product reviews etc.

Individual user can also use it to know the sentiment about a particular product, topic or politics.

The usage of latest and updated NLTK classifiers improve our accuracy and more over we improve the training set to get more accurate results.

**FUTURE SCOPE**

We look forward to use bigger dataset to improve the accuracy.

We look forward to consider emoticons, expressions and internet slangs in our training data.

We look forward to classify tweets as neutral tweets along with positive and negative tweets.

41

**BIBILOGRAPHY**

**12. BIBILOGRAPHY**

**Paper References**

IEEE reference paper “Sentiment analysis of Twitter - April 2016” by Vishal A. Kharde and S.S. Sonawane.

IEEE reference paper “Sentiment analysis of Twitter Data” - March 2015 by Apoorv Agarwal and Rebecca Passonneau.

**Web References**

**YouTube channel**

Sentiment analysis (NLTK) tutorials by sentdex:

<https://www.youtube.com/watch?v=imPpT2Qo2sk&list=PLQVvvaa0QuDf2JswnfiGkliBInZnIC4HL&index=5>

**Wikipedia**

https://en.wikipedia.org/wiki/Naive\_Bayes\_classifier

<https://en.wikipedia.org/wiki/Machine>\_Learning \_classifiers

https://en.wikipedia.org/wiki/Naive\_Bayes\_classifier

42