ABSTRACT

Twitter is an online microblogging and social networking platform, which allows users to write short status, updates of maximum length 280 characters. These tweets reflect public sentiment about various topics and events happening. Analysing the public sentiment can help, firms trying to find out the response of their products in the market, predicting political elections and predicting socioeconomic phenomena like stock exchange.

System does sentiment classification of tweets using machine-learning approach in python. A web application developed using flask in python displays these results of sentiment analysis in the form of various graphs like line graphs, maps, pie chart etc. This project deals with analysing sentiments behind the tweets be it positive or negative. Focus is not only on classifying the tweets, but also on making this task faster and more accurate by removing the parts of the tweets not contributing to the sentiment analysis, with the help of data pre-processing techniques. These pre-processing techniques include case conversion, punctuation removal, stopword removal, URL removal etc.

Classifier based on supervised machine learning algorithm classifies the sentiment present in a tweet. The model obtained from this classifier is applied on the tweets of currently trending topic, on twitter. The application will help in understanding people's views and emotions regarding the trending topics. It helps in understanding the effect of a particular topic on people, whether the topic garners majority of positive reviews or negative reviews.

CHAPTER 1: INTRODUCTION

1.1. INTRODUCTION

The project relies heavily on techniques of “Natural Language Processing” in extracting significant patterns and features from the large data set of tweets and on “Machine Learning” techniques for accurately classifying individual unlabelled data samples (tweets) according to whichever pattern model best describes them.

The features that can be used for modelling patterns and classification can be divided into two main groups: formal language based and informal blogging based[1]. Language based features are those that deal with formal linguistics and include prior sentiment polarity of individual words and phrases, and parts of speech tagging of the sentence. Prior sentiment polarity means that some words and phrases have a natural innate tendency for expressing particular and specific sentiments in general[1]. For example, the word “excellent” has a strong positive connotation while the word “evil” possesses a strong negative connotation. Therefore, whenever a word with positive connotation is used in a sentence, chances are that the entire sentence would be expressing a positive sentiment.

Parts of Speech tagging, on the other hand, is a syntactical approach to the problem. It means to automatically identify which part of speech each individual word of a sentence belongs to: noun, pronoun, adverb, adjective, verb, interjection, etc. [1]. Patterns can be extracted from analysing the frequency distribution of these parts of speech (ether individually or collectively with some other part of speech) in a particular class of labelled tweets.

Twitter allows the users to express their opinion in 140 words, which is now extended to 280 words. These tweets include twitter hashtags, retweets, word capitalization, word lengthening, question marks, presence of URL in tweets, exclamation marks, internet emoticons and internet shorthand/slangs. There are many performance measures, which can be used to measure the performance of the classifier. Some of them are accuracy, precision, recall, true positive rate, false positive rate etc. A typical confusion matrix for the problem will be given as,

Table 1.1 Confusion Matrix

	Predicted positive	Predicted negative
Actual positive	TP	FN
Actual negative	FP	TN

1.2 KEY CONCEPTS

1.2.1 Sentiment Analysis:

Sentiment analysis (opinion mining or emotion AI) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information.

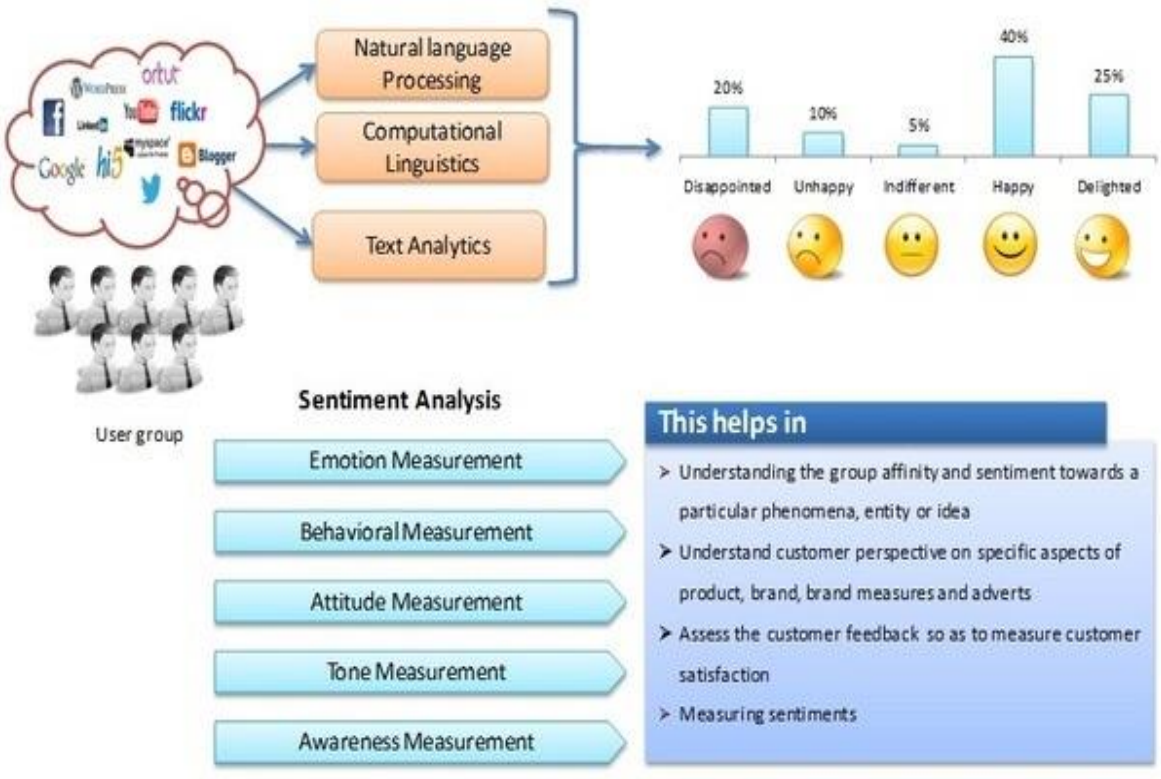


Fig 1.1 Sentiment Analysis[9]

Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

Sentiment analysis aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. The attitude may be a judgment, affective state (the emotional state of the author or speaker), or the intended emotional communication (the emotional effect intended by the author or interlocutor). The classification depends on polarity score of the sentence. It also depends on subjectivity and objectivity of a sentence. Using machine learning classifiers, the sentence can be classified in any one of the above three classes.

There are three main classification levels in sentiment analysis:

- **Document-Level Sentiment Analysis:**

It aims to classify an opinion document as expressing a positive or negative opinion or sentiment. It considers the whole document a basic information unit (talking about one topic).

- **Sentence-Level Sentiment Analysis:**

It aims to classify sentiment expressed in each sentence. The first step is to identify whether the sentence is subjective or objective. If the sentence is subjective, Sentence-level sentiment analysis will determine whether the sentence expresses positive or negative opinions.

- **Aspect Level Sentiment Analysis:**

It aims to classify the sentiment with respect to the specific aspects of entities. The first step is to identify the entities and their aspects.

1.2.2 Sentiment Analysis Approaches:

There are several approaches for sentiment analysis:

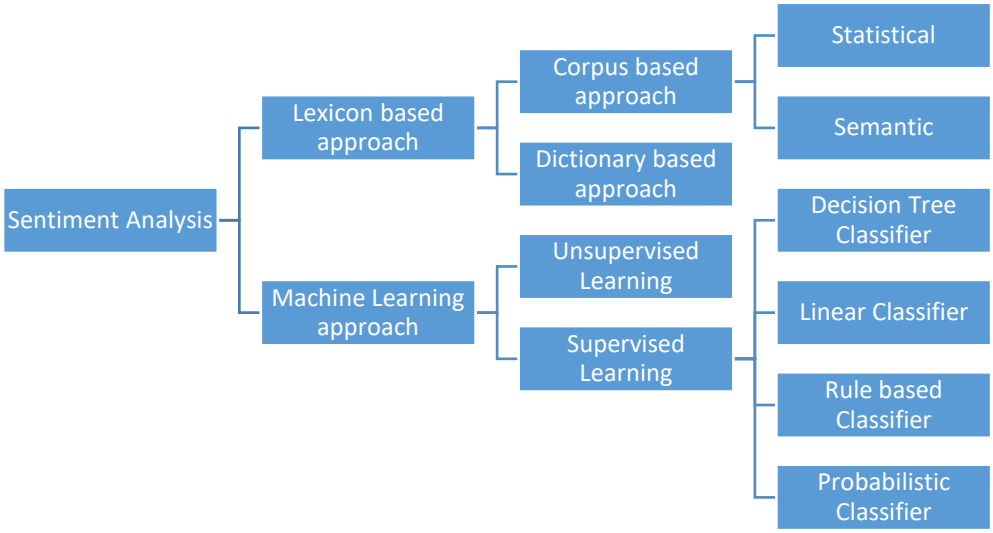


Fig 1.2 Sentiment Analysis Approaches

1.2.2.1 Machine learning based approach (ML):

It uses several machine-learning algorithms (supervised or unsupervised algorithms) to classify data. Here, two sets of documents are needed: training and a test set. A supervised learning classifier uses the training set to learn and train itself with respect to the differentiating attributes of text, and the performance of the classifier is tested using test dataset. Several machine-learning algorithms like Maximum Entropy (ME), Naive Bayes' (NB) and Support Vector Machines (SVM) are usually used for classification of text (tweets).

1.2.2.2 Lexicon based approach:

It uses a dictionary containing positive and negative words to determine the sentiment polarity. It is based on finding the opinion lexicon for calculating the sentiment for a given text. It deals with counting the number of positive and negative words in the text. If the text consists of more positive words, the text is assigned a positive score. If there are more number of negative words, the text is assigned a negative score. If the text contains equal number of positive and negative words then it is assigned a neutral score. To determine whether a word is positive or negative an opinion lexicon (positive and negative opinion words) is built[2]. There are several approaches to compile and build an opinion lexicon:

- **Dictionary based approach:**

A small set of opinion words is collected manually with known orientations. Then, synonyms and antonyms of these words are searched in corpora like WordNet or thesaurus and added to the set[2]. The set gradually grows until no new words are found. This approach has a disadvantage that the strength of the sentiment classification depends on the size of the dictionary. As the size of the dictionary grows this approach becomes more erroneous.

- **Corpus based approach:**

They depend on large corpora for syntactic and semantic patterns of opinion words[2]. The words that are generated are context specific and may require a huge labelled dataset.

1.2.2.3 Hybrid approach:

It uses a combination of both ML and lexicon based approach for classification. The main advantage of this hybrid approach is that we can attain best of both world. The lexicon/learning combination has proven to improve accuracy. Lexicon based approach have high precision and low recall.

1.2.3 Polarity:

Polarity, also known as orientation is the emotion expressed in the sentence. It can be positive, negative or neutral. A positive sentence has polarity from 0 to 1. A negative sentence has polarity from -1 to 0. A neutral sentence has polarity 0.

1.2.4 Subjectivity and Objectivity:

The ability to perceive or describe something without being influenced by personal emotions or prejudices is known as objectivity. Objective sentences are often facts. Interpretation based on personal opinions or feelings rather than on external facts or evidence is called subjectivity. Subjective sentences are often opinions. In sentiment analysis, an objective statement does not contribute to the sentiment analysis. However, a subjective statement is used for analysis, because it poses an opinion of an individual, which can reflect a sentiment.

CHAPTER 2: LITERATURE SURVEY

Many researchers have carried out their research work in sentiment analysis using social media. Still there are many challenges in social media sentiment analysis, which needs to be overcome.

2.1 PRIOR ART

Sentiment analysis of micro-blogging website is a new research topic, so there is lot of scope for research in this area. Fair amount of related prior work has been done on sentiment analysis of reviews, documents, web blogs/articles and general phrase level sentiment analysis. The best results in sentiment analysis until obtained are obtained using supervised machine learning classifiers such as Naïve Bayes Algorithm and Support Vector Machine. The problem with sentiment analysis is the expensive manual labelling for supervised approach. Even though some work has been done on unsupervised and semi-supervised approaches, there is still, room for improvement. There is need of proper and efficient techniques for feature extraction and classification techniques.

2.2 IMPORTANCE OF SENTIMENT ANALYSIS

People share their sentiments regarding various topics on social media in the form of short messages. This can be important information if you know how to deal with it. If human behaviour is observed, it can be noticed that a person is greatly influenced by opinion of another person. From this we can conclude that, people can influence sentiments of other people on social media network. So sentiment analysis of social media becomes very important as social media now-a-days covers all aspects of life.

2.3 CHALLENGES IN SENTIMENT ANALYSIS

In sentiment analysis, the main aim is to extract positive or negative words from the text and classify the text as positive, negative or neutral based on the sentiment words. There are various challenges in sentiment analysis. Some of them are:

2 Implicit sentiment and sarcasm

A sentence may have an implicit sentiment even without the presence of any sentiment bearing words. Consider the following examples, “How can anyone elect this candidate?”,

"One should question the stability of mind of the writer who wrote this book." Both the sentences do not explicitly carry any negative sentiment bearing words although both are negative sentences. Thus identifying semantics is more important in sentiment analysis than syntax detection.

- **Domain Dependency**

There are many words whose polarity changes from domain to domain. Consider the following examples.

"The result was unpredictable."

"The steering of the car is unpredictable."

"Go read the book."

In the first example, the sentiment conveyed is positive whereas the sentiment conveyed in the second is negative. The third example has a positive sentiment in the book domain but a negative sentiment in the movie domain.

- **Thwarted Expectations**

Sometimes the author deliberately sets up context only to refute it at the end. Consider the following example:

"This film should be brilliant. It sounds like a great plot, the actors are first grade, the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However it can't hold up." Inspite of the presence of words that are positive in orientation the overall sentiment is negative because of the crucial last sentence. In traditional text classification this would have been classified as positive as term frequency is more important than the term presence.

- **World Knowledge**

Often world knowledge needs to be incorporated in the system for detecting sentiments. Consider the following example: "He is a Frankenstein. Just finished Doctor Zhivago for the first time and all I can say is Russia sucks." The first sentence depicts a negative sentiment whereas the second statement depicts a positive sentiment, but one has to know about Frankenstein and Doctor Zhivago to find out the sentiment.

- **Subjectivity Detection**

This is to differentiate between opinionated and non-opinionated text. This is used to enhance the performance of the system by including a subjectivity detection module to filter out objective facts. However, this is often difficult to do. Consider the following examples: I hate

love stories. I do not like the movie. The first example presents an objective fact whereas the second example depicts the opinion about a particular movie.

2.4 RELATED WORK

Application of Machine Learning Techniques to Sentiment Analysis

Author: Anuja P Jain, Asst. Prof Padma Dandannavar

Published Year: 2016

Published in: 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)

This paper explains the sentiment analysis approaches. It also explains the data pre-processing steps and some of the machine learning algorithms that are used for sentiment analysis. It concludes that Lexicon based approach faces a disadvantage that the strength of the sentiment classification depends on the size of the lexicon (dictionary). As the size of the lexicon increases, this approach becomes more erroneous and time consuming. It also concludes that Decision tree performs extremely well showing 100% accuracy, precision, recall and F1-Score. From this paper, we are taking the sentiment analysis approaches and methods out of which our proposed system is going to use the machine learning approach for sentiment analysis.

Comparative Analysis of Twitter Data using Supervised Classifiers

Author: Rohit Joshi, Rajkumar Tekchandani

Published Year: 2016

Published in: International Conference on Inventive Computation Technologies (ICICT)

This paper explains three machine learning algorithms like Support Vector Machine, Naïve Bayes, Maximum Entropy. It also does comparison between them based on the performance of classifier using unigram, bigram and hybrid (unigram + bigram) feature. Based on the comparison, effect of each machine learning algorithm on the system is estimated. It concludes that, SVM using hybrid feature outperforms all other classifiers and selection feature with accuracy of 84%. Maximum Entropy surpass Naïve Bayes with bigram feature. Maximum Entropy on some data sets gives better results than Naïve Bayes. It is concluded

that SVM gives better results than other classifiers. From this paper we understood which machine learning classifier works better for which kind of feature in which situation.

Twitter data clustering and visualisation

Author: Andrei Sechelea, Tein Do Huu

Published Year: 2016

Published in: 23rd International Conference on Telecommunications (ICT)

This paper explains how to store tweets in distributed cluster and how the data can be processed using algorithms implemented in a MapReduce framework. They proposed a visualisation method, which allows following the intensity of twitter activity at geographical location. They also presented a clustering algorithm capable of identifying hot topics of interest in a tweet data set.

Twitter data analysis and visualisation using the R language on top of the Hadoop Platform

Author: Martin Sarnovsky, Peter Butka

Published Year: 2017

Published in: SAMI 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics January 26-28, 2017 Herl'any, Slovakia

This paper proposes the design and implementation of the system for twitter data analysis and visualization in R environment using big data processing technologies. The system is developed using R and utilizes the big data processing technologies. Small-sized Hadoop cluster is deployed and enhanced with RHadoop packages to support the distributed processing of R functions. The authors developed a set of analytical methods using MapReduce framework from RHadoop package and designed a set of visualizations implemented as Shiny web applications. From the paper we are taking the various visualization techniques that can be used to visualize the results of sentiment classification.

CHAPTER 3: PROBLEM STATEMENT

3.1 PROBLEM STATEMENT

Sentiment analysis and visualization of twitter trends using machine learning approach.

3.2 OBJECTIVES

3 To build an efficient and accurate model for sentiment classification which classifies a tweet as positive or negative.

4 To download current trends and tweets related to them from twitter using an API and apply this model on those tweets.

5 To develop a web application in python which visualises the results of sentiment classification.

3.3 SCOPE

The scope of the project is to provide a user-friendly web based product that extracts people's sentiment feelings toward topics, which are currently trending on twitter. System will classify only English tweets. There are no provisions for classification of other language tweets. System will classify tweets of only top ten trends of that day and will generate graphs for the same, for analysis.

3.4 MOTIVATION

Twitter has resulted in hosting massive datasets of information. Thus its data is gaining increasing interest. People use Twitter to share experiences and emotions with their friends about movies, products, events etc., so a system that extracts sentiments through an online community may have many real-life applications such as recommendation systems. This enormously continuous stream of Twitter data posts reflects the user's opinions and reactions to phenomena from political events all over the world to consumer products. It is well pointed that Twitter posts relate to the user's behaviour and often convey substantial information about their emotional state.

Unlike other networks, user's posts in Twitter have some special characteristics. The short length that the posts are allowed to have, results in more expressive emotional statements.

Analysing tweets and recognizing their emotional content is a very interesting and challenging topic in the microblogging area. Recently many studies have analysed sentiment from documents or web related content, but when such applications are focused on microblogging, many challenges occur. The limited size of the messages, along with the wide range of subjects discussed, make sentiment extraction a difficult process. Concretely, researchers have used long-known machine learning algorithms in order to analyse sentiments. So the problem of sentiment extraction is transformed into a classification problem. Datasets of classified tweets are used to train classifier which in following are used to extract the sentiments of the messages.

In the meantime, as data grows, cloud computing evolves[3]. Frameworks like Hadoop, Apache Spark, Apache Storm and distributed data storages like HDFS and HBase are becoming popular, as they are engineered in a way that makes the process of very large amounts of data almost effortless. Such systems evolve in many aspects, and as a result, libraries, like Spark's MLlib that make the use of Machine Learning techniques possible in the cloud, are introduced.

The sentiments of people indicate the nation's stability. People are the first pillar of democracy. It has been observed that, popular sentiments spread by social networking platforms influences politics a lot. Underlying changes in public opinion across generations highlight the profound impact this may have on drawing up the public policy priorities of the future. We chose political sentiment analysis because of its far-reaching consequences on India's political orientation and public policies. Using this analysis, political orientation of public can be predicted. Public's sentiments regarding decisions taken by government and its effectiveness can be understood. Similarly public's sentiment regarding opposition parties as well as government and their work, can be understood. Hence analysing public opinion and its underlying sentiment becomes imperative.

In this proposed system, we aim on creating a Sentiment Analysis tool of Twitter, which classifies tweets of trending topics using supervised learning techniques. System will visualize the classification of tweets based on the user's input (like according to geographical location) also it will generate report of the working model of the sentiment classifier for administrator.

CHAPTER 4: PROJECT REQUIREMENTS

4.1 HARDWARE REQUIREMENTS:

System Requirements:

RAM: 8 GB

Hard Disk: 1 TB

Processor: Intel core I7

4.2 SOFTWARE REQUIREMENTS:

Operating System: Windows 2010, Ubuntu

Software: JetBrains PyCharm Community Edition 2016.1.4

Anaconda 3

Oracle VM VirtualBox

Apache Hadoop

Apache Spark

Database: MongoDB

Language: Python

API: Tweepy

Web Framework : Flask

4.3 API REQUIREMENTS

Tweepy:

Tweepy is open-sourced, and enables Python to communicate with Twitter platform and use its API.

OAuth:

Tweepy supports accessing twitter via OAuth. Twitter uses OAuth to provide authorized access to its API.

Features of OAuth:

- **Secure** - Users are not required to share their account credentials with 3rd party applications, increasing account security.
- **Standard** - A wealth of client libraries and example code are compatible with Twitter's OAuth implementation.

Twitter API Authentication Model:

1. Application-only Authentication:

Application-only authentication is a form of authentication where an application makes API requests on its own behalf, without the user context[13]. This method is for developers that just need to access public information.

2. User Authentication

The user authentication method of authentication allows an app to act on behalf of the user, as the user.

Code for accessing twitter-using tweepy:

```
import tweepy

# Consumer keys and access tokens, used for OAuth
consumer_key = '7EyzTcAkINVS3T2pb165'
consumer_secret = 'a44R7WvbMW7L8I656Y4l'
access_token = 'z00Xy9AkHwp8vSTJ04L0'
access_token_secret = 'A1cK98w2NXXaCWMqMW6p'

# OAuth process, using the keys and tokens
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)
```

To access twitter data using API first the user needs to create an app on twitter, which provides the user with consumer_key, consumer_secret, access_token, access_token_secret. Tweepy.OAuthHandler creates an OAuthHandler instance and it takes consumer_key and consumer_secret as argument. Tweepy.API method creates an instance of API. While downloading tweets, if maximum tweets to be

downloaded in a particular time limit is crossed, then the application will sleep for specific period.

4.4 WEB FRAMEWORK REQUIREMENT

Flask

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, and upload handling, various open authentication technologies and several common framework related tools.

Features:

- Easy to use.
- Built in development server and debugger
- Integrated unit testing support
- RESTful request dispatching
- Uses Jinja2 templating
- Support for secure cookies (client side sessions)
- 100% WSGI 1.0 compliant
- Unicode based
- Extensively documented

4.5 USER CLASSES AND CHARACTERISTICS:

This part is to identify various user classes that we anticipate will use the web application. User classes will be differentiated based on the use, product functions and features, privilege levels and educational level. The solution is intended to be used by two main different user classes; system administrators and customers or regular users. No special knowledge or skills should be assumed for the part of the regular users. Users are not expected to learn or remember a set of commands in order to start using the application. The application will be only a web based. The following clearly describes a visionary role for each participant.

• **Users:**

- Users with no particular knowledge needed, users who are interested to use the tool looking for knowing people's thoughts about a desired topic.

• **System Administrators:**

- Develop and maintain installation and configuration procedures and operational requirements
- Perform weekly/monthly backup operations, ensuring all required files and data are successfully backed up
- Repair and recover from hardware or software failures
- View reports

4.6 INTERFACE REQUIREMENTS:

4.6.1 User Interfaces

User interface includes various forms and windows. The interface will visualize the features and functionalities listed in this document for this project as:

• Drop down menu for various option selection

• Selection list for filtering results

• Push buttons for user's feedback and reclassifying tweets

• Visual graphs to show results

• Help button

4.6.2 Communications Interfaces

Internet connection and a web browser are required in order to view the website, which contains various reports and statistical data.

4.6.3 Hardware Interfaces:

System: Pentium IV 2.4GHz

RAM: 1GB

HDD: 1TB

4.6.4 Software Interface:

Operating System: windows 8

Platform: Python

API: Tweepy

4.7 NON-FUNCTIONAL REQUIREMENTS

4.7.1 Performance Requirements

For the application we will keep on detecting if the system crashed, hanged or an operating system error occurred. Also detecting the performance of the system in terms of the efficiency of integration of the different components

4.7.2 Safety Requirements

For the safety requirements, nothing but an operation of weekly backups for the database will take place.

4.7.3 Security and Privacy Requirements

There are no specific security requirements, anyone can access and use the portal but only authorized persons who are allowed to use and access the database and web pages from administration side.

4.8 SOFTWARE QUALITY ATTRIBUTES:

- **Reliability**

The solution should provide reliability to the user that the product will run with all the features mentioned in this document are available and executing perfectly. It should be tested and debugged completely. All exceptions should be well handled.

- **Accuracy**

The solution should be able to reach the desired level of accuracy.

- **Availability:**

The system shall be available as long as operations are performed on application.

- **Updatability:**

The system shall allow addition and deletion of files based on access right provided.

- **Testability:**

New modules designed to be added to system must be tested to check whether they are integrated properly and compatible with input-output format of the system.

CHAPTER 5: SYSTEM ARCHITECTURE

5.1 PROPOSED SYSTEM ARCHITECTURE

"

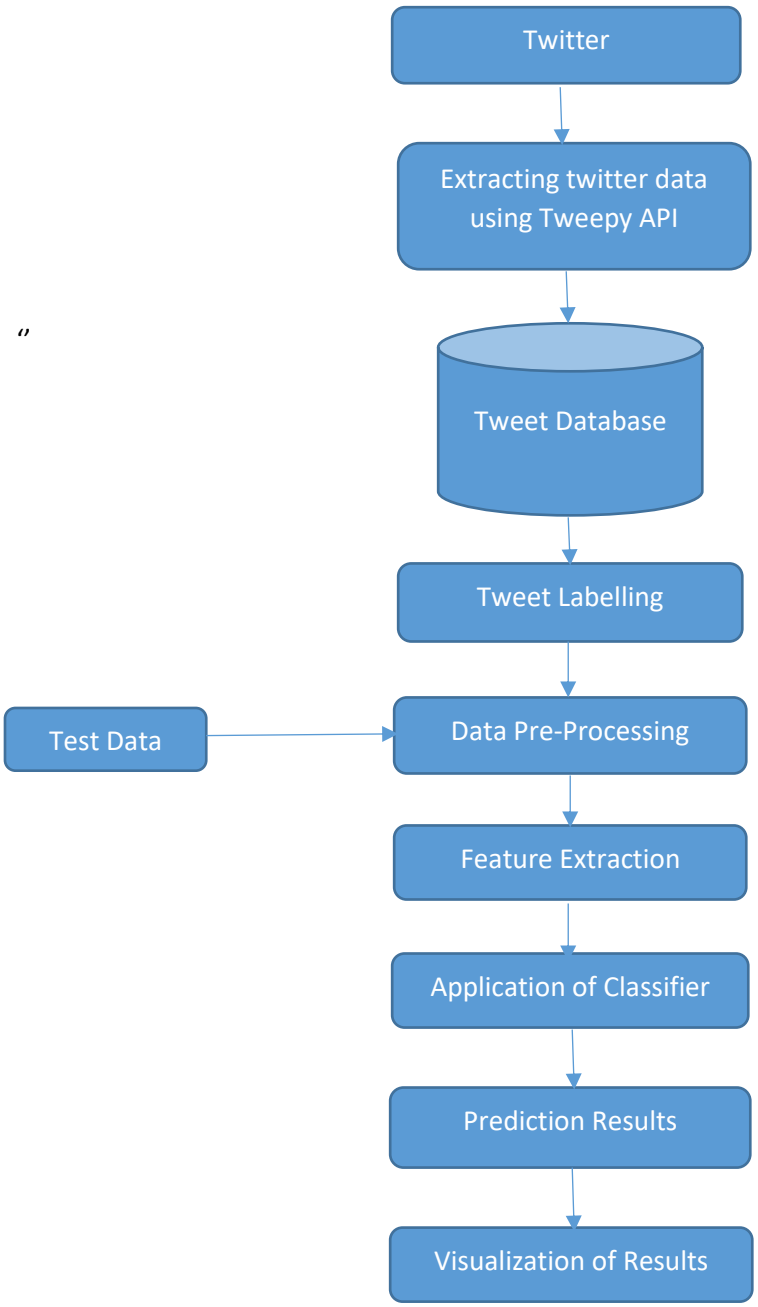


Fig 5.1 System Architecture

5.2 PROCEDURE

The process of designing a classifier for sentiment analysis involve following steps:

5.2.1 Data Acquisition

Data in the form of raw tweets is acquired by using the python library "tweepy" which provides a package for simple twitter streaming API. Tweepy supports accessing twitter via basic authentication and the newer method, OAuth. Tweepy provides access to the well-documented twitter API. With tweepy, it is possible to get any object and use any method that the official Twitter API offers. It can filter the delivered tweets according to three criteria:

1. Specific keyword(s) to track/search for in the tweets
2. Specific Twitter user(s) according to their user-id's
3. Tweets originating from specific location(s) (only for geo-tagged tweets).

A tweet acquired by this method has a lot of raw information in it, which we may or may not be useful for our particular application[2]. It comes in the form of the python "dictionary" data type with various key-value pairs. A list of some key-value pairs are given below:

- Whether a tweet has been favorited
- User ID
- Screen name of the user
- Original Text of the tweet
- Presence of hashtags
- Whether it is a re-tweet
- Language under which the twitter user has registered their account
- Geo-tag location of the tweet
- Date and time when the tweet was created

Since this is a lot of information, we only filter out the information that we need and discard the rest. The filtering criteria applied are stated below:

- Remove non-English tweets

5.2.2 Labelling

We label the tweets in three classes according to sentiments expressed/observed in the tweets positive, negative, neutral/objective. We are going to use following guidelines for labelling process:

- **Positive**: If the entire tweet has a positive/happy/excited/joyful attitude or if something is mentioned with positive connotations[2]. In addition, if more than one sentiment is expressed in the tweet but the positive sentiment is more dominant. Example: "*4 more years of being in shithole Australia then I move to the USA! :D*".
- **Negative**: If the entire tweet has a negative/sad/displeased attitude or if something is mentioned with negative connotations[2]. In addition, if more than one sentiment is expressed in the tweet but the negative sentiment is more dominant. Example: "*I want an android now this iPhone is boring :S*".
- **Neutral/Objective**: If the creator of tweet expresses no personal sentiment/opinion in the tweet and merely transmits information[2]. Advertisements of different products would be labelled under this category. Example: "*US House Speaker vows to stop Obama contraceptive rule... http://t.co/cyEWqKlE*".

5.2.3 Data Pre-processing

The data pre-processing can often have a significant impact on the performance of a supervised ML algorithm. The steps that are carried out in pre-processing of data are as follows:

- Case Conversion: All words are converted into either lower case or upper case in order to remove the difference between "Text" and "text" for further processing.
- Stop-words Removal: The commonly used words like a, an, the, has, have etc. which carry no meaning i.e. do not help in determining the sentiment of text while analysing should be removed from the input text[2].
- Punctuation Removal: Punctuation marks such as comma or colon often carry no meaning for the textual analysis hence they can be removed from input text.
- Stemming: Stemming usually refers to a simple process that chops off the ends of words to remove derivational affixes.
- Lemmatization: Deals with removal of inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma[2].

5.2.4 Feature Extraction

Once the tweets are pre-processed, we need to extract features relevant for sentiment analysis. There are multiple number of features that can extracted from a tweet.

Some of the features include:

1. Term presence and frequency: It usually consists of n-grams of words and their frequency counts
2. Parts of speech tagging: words in the text are tagged with their respective parts of speech in order to extract adjectives nouns verbs, which add meaning to the sentiment[2].
3. Opinion words and phrases: words or phrases that indicate opinion of the text.
4. Negation: presence of words like ‘not’, ’nor’, ’neither’ may reverse the sentiment of whole sentence. E.g. “not good“

5.2.5 Machine Learning Algorithms for Sentiment Classification

1. Naïve Bayes classifier
2. Support vector machines
3. Decision trees

5.2.6 Visualisation

The web application will have two modules:

1. Administrator
2. User

User can view the tweets, sentiment scores, graphs indicating impact of various parameters on people’s sentiments.

Administrator can view confusion matrix, algorithm being used, performance parameters of the model built, number of sample tweets obtained etc.

CHAPTER 6: PROJECT PLAN

Table 6.1: Semester I Project Plan

2017-18 Semester-I	
Duration	**Plan**
June 2017	
First Week	Discussion, Drafting and Study about carrying the TE Seminar Topic forward for implementation of BE project.
Second Week	Finalizing the problem statement, modules in project.
Third Week	Identifying proper input to the system and expected output. Deciding the software, equipment and tools to be used
Fourth Week	Submission of synopsis and preparation of introductory presentation
July 2017	
First Week	Basic Study of Machine Learning models.
Second Week	Discussion about paper topic, study for Paper.
Third Week	Preparation and submission of Review paper to IJCA.
Fourth Week	Basic Implementation of few machine-learning algorithms.
August 2017	
First Week	Designing the scope of the system and plan to start the project design.
Second Week	Preparation of the Project design models.
Third Week	Prepared the Presentation for Review- I.
Fourth Week	Data acquisition from twitter step done.
September 2017	
First Week	Discussion of the Project Review-II.
Second Week	Discussion about project report.
Third Week	Preparing of Semester-I report.
Fourth Week	Preparing Review-II presentation.
October 2017	
First Week	Submission of report
Second Week	Reviewing of work done for semester I
Third Week	Study of software to implement web interface in python
Fourth Week	Selecting final software for web interface implementation
November 2017	
First Week	Detailed study of python web interface software
Second Week	Simple coding for website implementation
Third Week	Tasks to be completed as per discussions in vacations
Fourth Week	Tasks to be completed as per discussions in vacations

Table 6.2 Semester II Project Plan

2017-18 Semester-II	
Duration	**Plan**
December 2017	
First Week	Labelling the data obtained using tweepy API
Second Week	Checking labels to find out errors
Third Week	Dividing the dataset into training and testing dataset
Fourth Week	Study of visualization techniques
January 2018	
First Week	Study of feature extraction methods
Second Week	Study of various features and their importance, which are needed for sentiment analysis
Third Week	Selecting the features which are important for sentiment analysis
Fourth Week	Discussion of project review III
February 2018	
First Week	Review III
Second Week	Implementing feature extraction step
Third Week	Implementing various machine learning algorithms on extracted features
Fourth Week	Extracting data from obtained results and dataset to visualise on web interface
March 2018	
First Week	Implementation of web interface
Second Week	Testing of web interface
Third Week	Participating in various project competitions
Fourth Week	Preparation of final report(black book) and presentation
April 2018	
First Week	Reviewing of entire project, presentation and report submission

CHAPTER 7: UML DESIGN

7.1 USECASE DIAGRAM:

A use case diagram at its simplest is a representation of a user's interaction with the system. It 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 in a system and their usecases. Other types of diagrams will often accompany it as well.

7.1.1 Basic Use Case Diagram Symbols and Notations

System

Draw your system's boundaries using a rectangle that contains use cases[12]. Place actors outside the system's boundaries.

System

Usecase

Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.

Use case

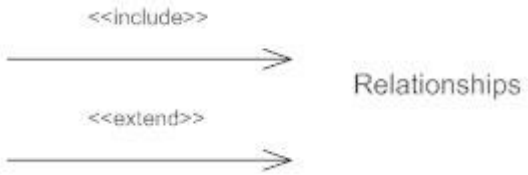
Actors

Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype[12].

Actor

Relationships

Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labelled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



7.1.2 System Use Case Diagram

This usecase diagram has three primary actors i.e. user, administrator and twitter API. This usecase scenario depicts interaction between user and web interface. The user can search a particular term to analyse its sentiment, can view reports and give feedback. The administrator can process search results, analyse data, store data. The function of twitter API is to download tweets from twitter.

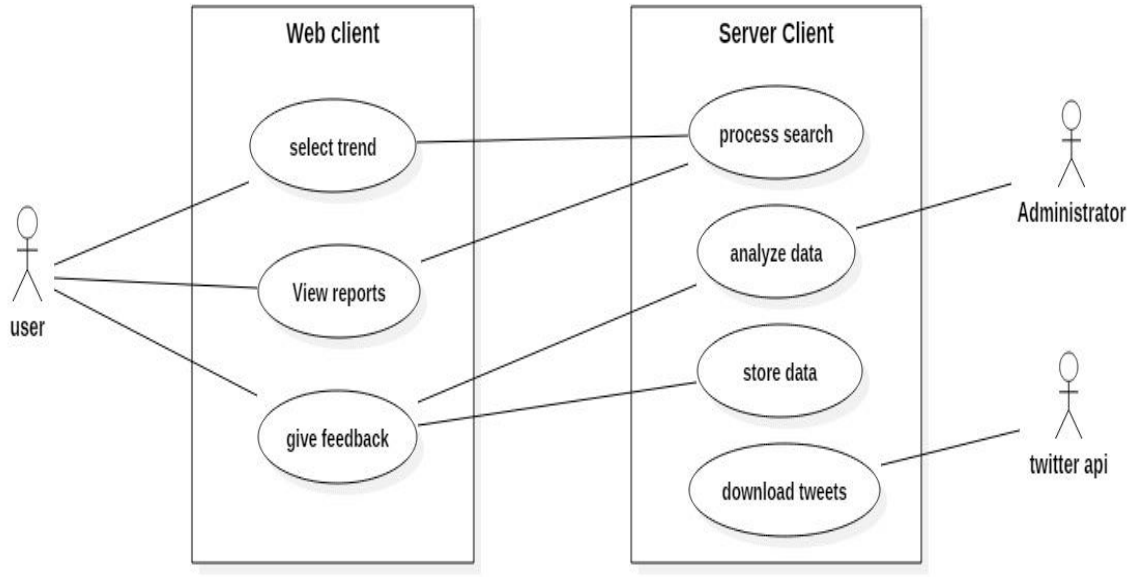


Fig 7.1 Usecase Diagram

CHAPTER 8: IMPLEMENTATION

For this project of sentiment analysis of twitter data, we are using machine-learning approach. The data is extracted from twitter, is processed and classifier is applied on it to get results which are then visualised. The steps for sentiment analysis are as follows:

8.1 DATA COLLECTION:

Twitter is a social micro-blogging website, which allows people to express their opinion in up to 280 words. For this project, we are downloading tweets from twitter using Tweepy API. The downloaded tweet is in JSON format which is then stored into a MongoDB collection. A tweet is in key-value pair format. There are numerous keys in a tweet, some of them not useful for our project. A list of some key-value pairs are given below:

- Whether a tweet has been favorited
- User ID
- Screen name of the user
- Original Text of the tweet
- Presence of hashtags
- Whether it is a re-tweet
- Language under which the twitter user has registered their account
- Geo-tag location of the tweet
- Date and time when the tweet was created

Out of all this attributes user ID, user name, text, geo-tag location, date and time is stored in a MongoDB collection for further processing.

8.2 DATA PRE-PROCESSING

The tweets stored in MongoDB collection are noisy in nature. If a machine-learning algorithm is applied on this tweets accuracy of prediction will be decreased. Therefore, to remove noise in the dataset, we use pre-processing methods, which remove the unwanted words which do not contribute to the sentiment analysis. The data pre-processing steps include URL removal, stopword removal, lowercase conversion, punctuation marks removal,

lemmatization, stemming etc. The final tweet obtained after data pre-processing is done contains only the words which contribute to sentiment analysis.

8.3 LABELLING

To use supervised machine learning approaches we need labelled dataset. In case of sentiment analysis the labels are positive, negative and neutral. We manually labelled all the tweets stored into the database. If the tweet contains words with positive/happy/joyful connotations then the tweet is labelled as positive. If the tweet contains words with negative/sad/displeased connotations then the tweet is labelled as negative. If the tweet does not have any sentiment words or it is just stated as a fact then it is labelled as neutral. The dataset is then divided into training and testing dataset. The ratio of training to testing dataset is 30:70.

8.4 FEATURE EXTRACTION

Features are the backbone of machine learning algorithms. Therefore feature selection and extraction is a very important step. There are number of features that can be extracted from the tweet for the purpose of sentiment analysis. Some of them are unigrams, bigrams, trigrams, term presence, term frequency, negation words etc. In this project, we extract unigrams, bigrams and trigrams from training dataset as features.

8.5 APPLICATION OF MACHINE LEARNING ALGORITHM

After the preparation of feature vector the next step is to apply machine learning classifier to it. In this project we have used Naïve Bayes, Support Vector Machine, Decision Tree classifiers.

1. Naïve Bayes classifier:

The Naive Bayes classifier is the simplest (as the name suggests) and most commonly used classifier. Naive Bayes classifier works very well for text classification as it computes the posterior probability of a class, based on the distribution of the words (features) in the document. It works using probabilistic model given below:

$$p(C_k \mid z1 \ldots zn)$$

For each of k possible outcomes or classes. However, if number of features is large, that is value of n is large then above formula does not work well, because probability table becomes too large and infeasible to handle. Therefore Bayes theorem is used, which decomposed the conditional probability as:

$$p(C_k/z) = \frac{p(C_k)\ p(z/C_k)}{p(z)}$$

Where C_k is class for each of k possible outcomes, and z are the instances to be classified.

```
CLASSIFIER = nltk.classify.NaiveBayesClassifier
classifier_tot = CLASSIFIER.train(v_train)
accuracy_tot = nltk.classify.accuracy(classifier_tot, v_test)
```

nltk.classify.NaiveBayesClassifier is used to get an instance of Naïve Bayes classifier into CLASSIFIER. CLASSIFIER.train is used to train the model based on training dataset. It returns trained classifier model. Nltk.classify.accuracy gives the accuracy of the model.

2. Support vector machines:

The main principle of SVMs is to find out linear separators or hyperplane in the search space, which can best separate the different classes. There can be several hyperplanes that separate the classes, but the one that is chosen is the hyperplane in which the normal distance of any of the data points is the largest, so that it depicts the maximum margin of separation.

3. Decision trees:

Here, the training data space is represented in a hierarchical form in which a condition on the attribute value is used to partition the data. The condition on attribute values is the presence or absence of one or more words. The partition of the data space is done recursively until the leaf nodes contain certain minimum numbers of records which are used for the purpose of classification.

8.6 VISUALISATION

In our web application, we provided the details of analysis and their visualizations on the processed twitter trends. For visualization purpose we utilized several Python provided packages such as Bokeh, MatPlotLib etc. We provided following group of data visualization-

- Visualization of tweets frequency and their locations
- WordCloud
- Pie chart displaying percentage of positive, negative and neutral tweets of the trend
- Bar graph displaying count of each trending topic
- Bar graph displaying percentage of positive, negative and neutral tweets of all trending topics

CHAPTER 9: RESULTS

The pie chart given below is the visualized result of sentiment analysis. It displays the percentage of count of tweets of each trending topic. The percentage of tweet count of top ten trending topics is displayed.

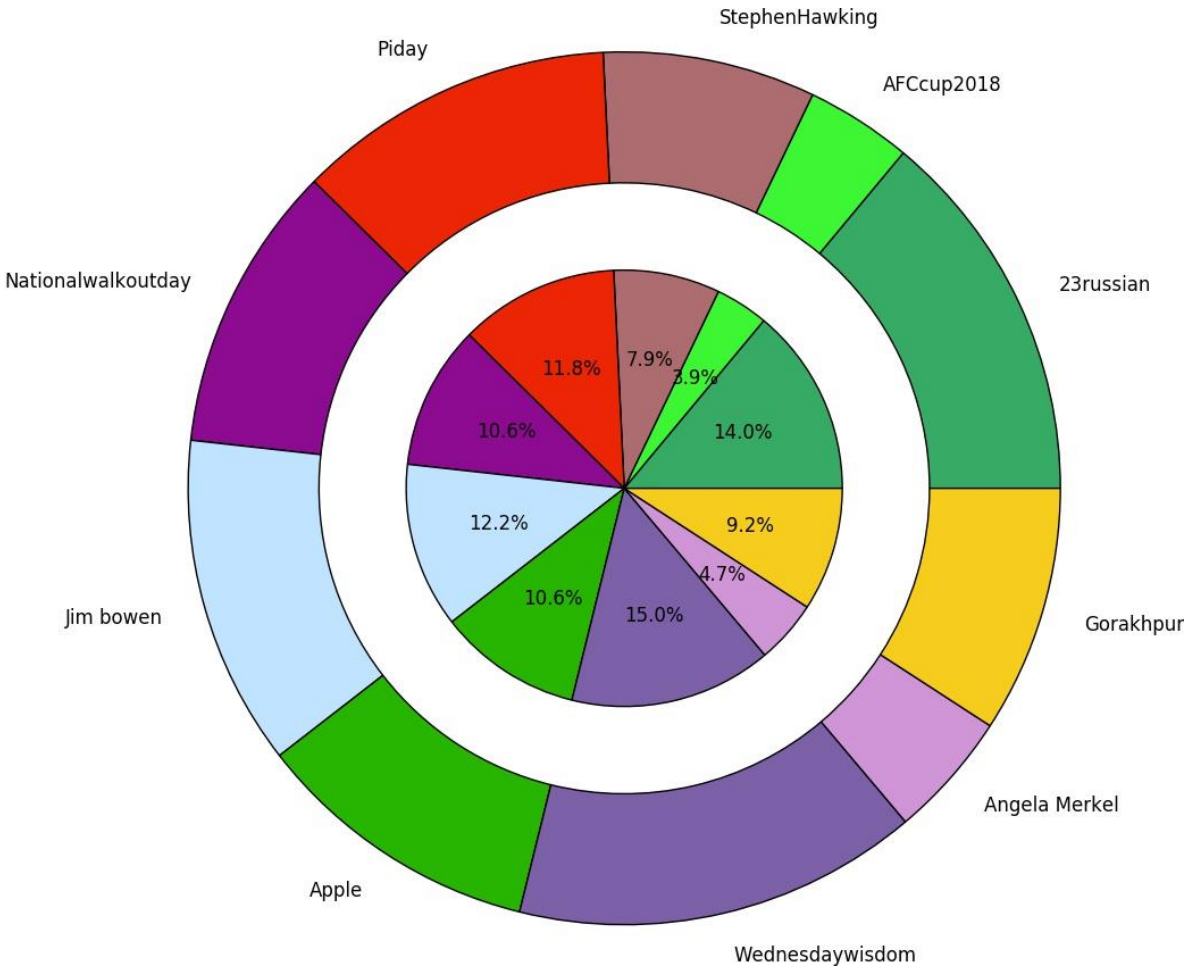


Fig 9.1 Pie Chart

This bar graph displays the count of positive, negative and neutral tweets related to each trending topic. The count is given in the metric of thousands.

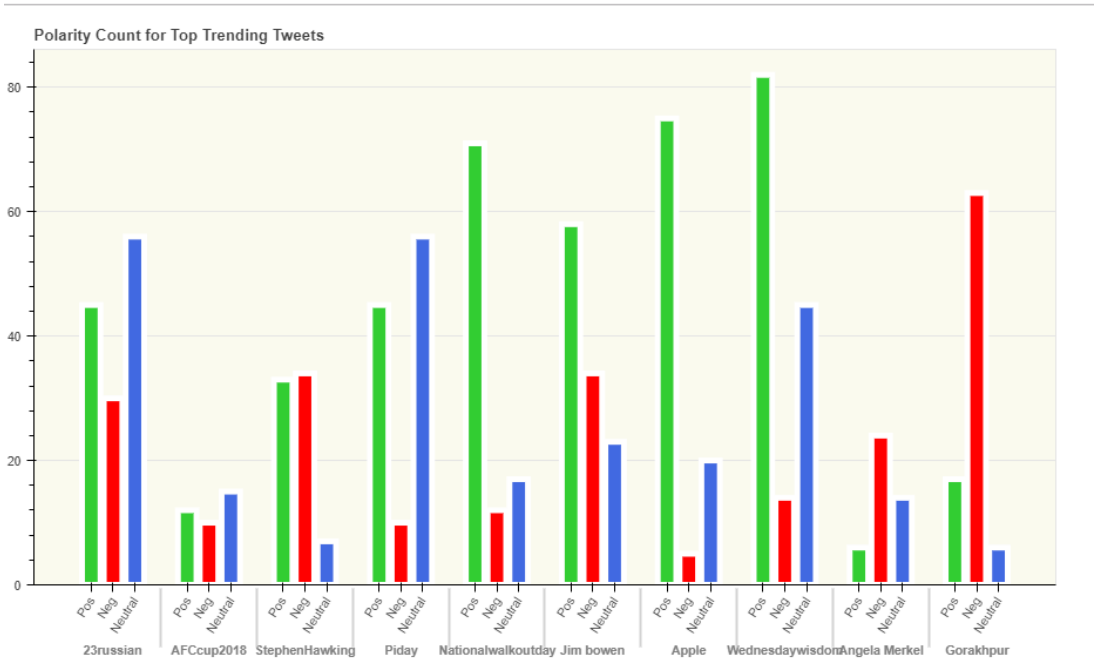


Fig 9.2 Bar Graph I

This bar graph displays the count of tweets related to each trending topic in the metric of thousands.

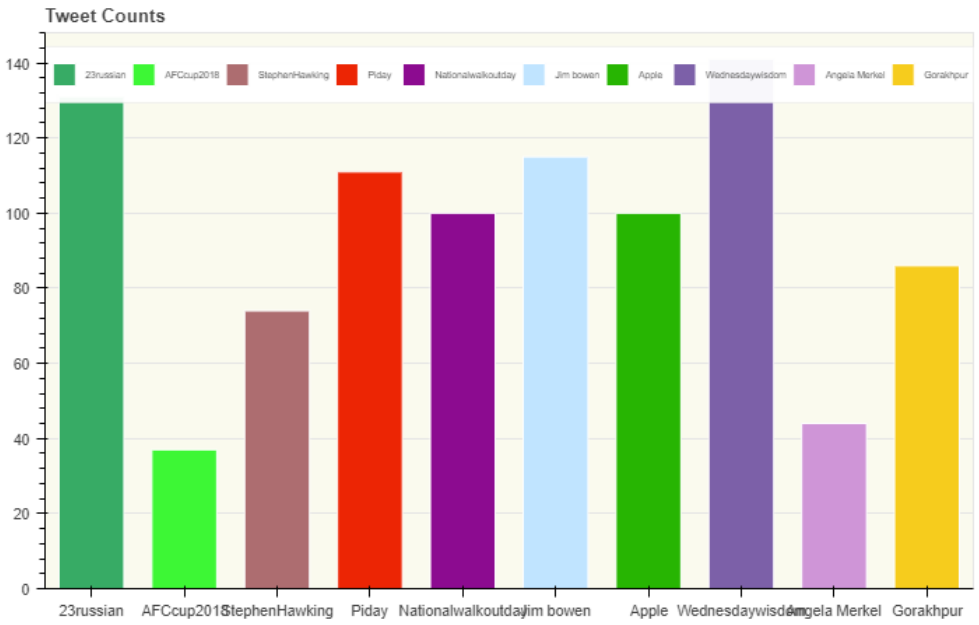


Fig 9.3 Bar Graph II

The map of India displays the location of positive, negative and neutral tweet related to the individual trending topic. The green color displays positive tweet location, the red color displays negative tweet location while blue color displays neutral tweet location.

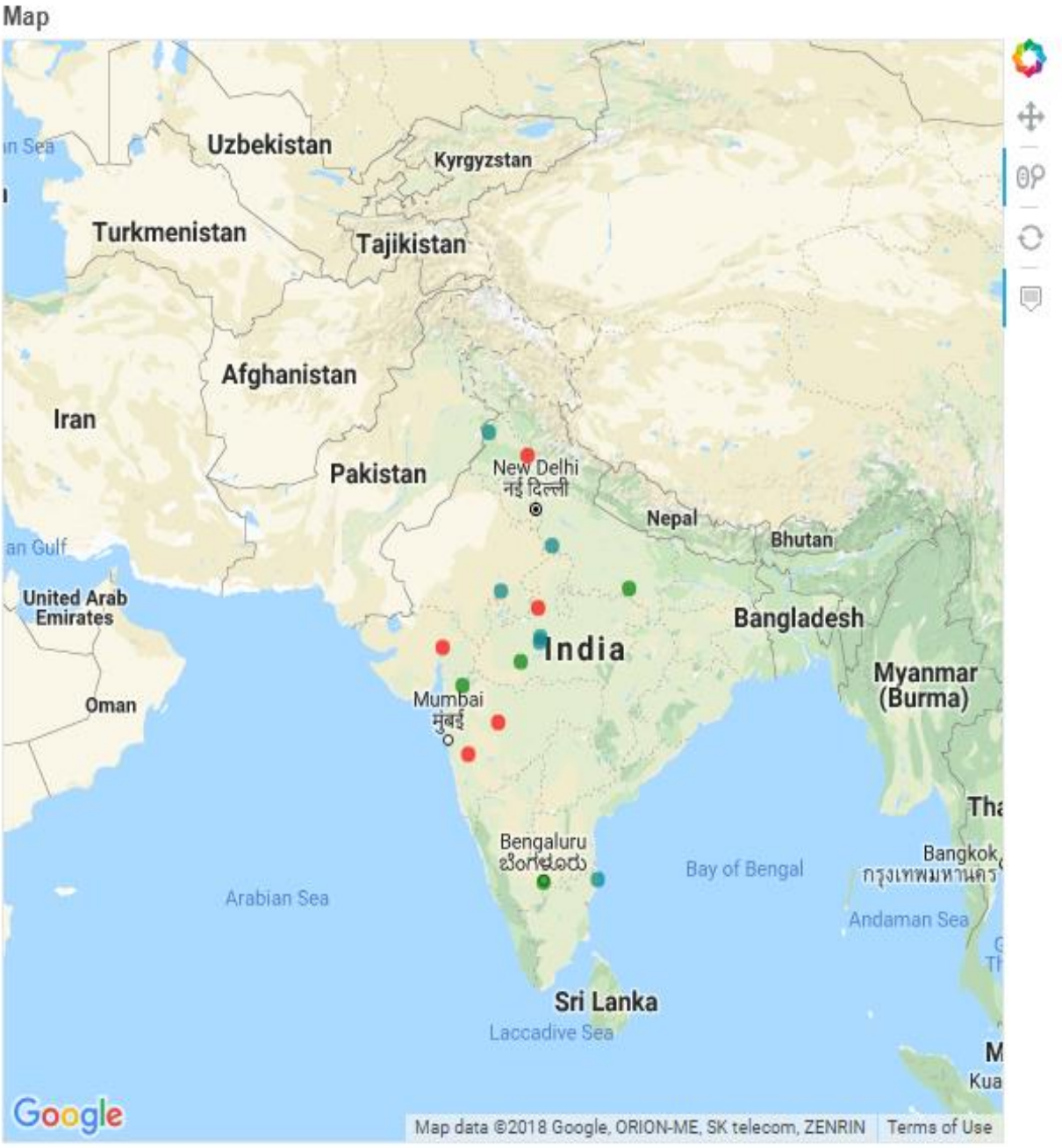


Fig 9.4 Map of India

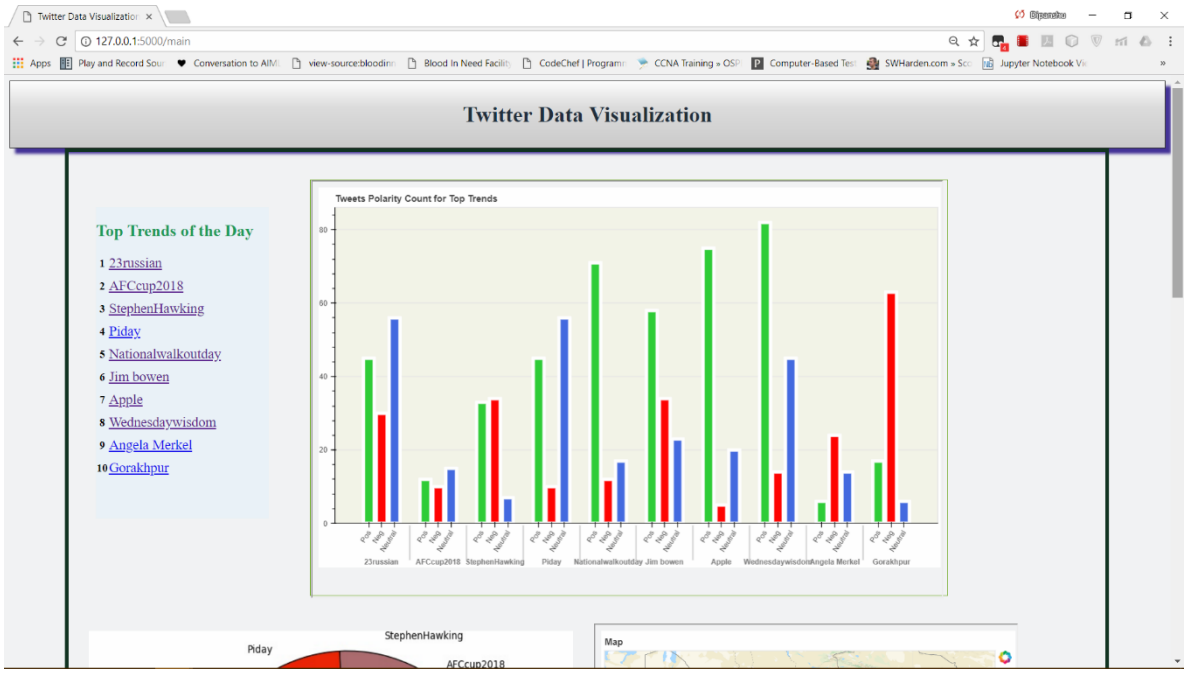


Fig 9.5 Homepage

CHAPTER 10: TESTING

10.1 TESTING

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process. Basics of software testing: There are two basics of software testing: black-box testing and white-box testing.

1. Black-box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

2. White-box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

10.2 TYPES OF TESTING

1. Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

2. Integration Testing

Integration testing is testing in which a group of components is combined to produce output. In addition, the interaction between software and hardware is tested approaches testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

3. Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

4. System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

5. Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavourable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

6. 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. It falls under the class of black box testing.

7. Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly. How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

8. Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

9. Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

10. Beta Testing

Beta testing is the testing which end users, a team outside development, do or publicly releasing full pre-version of the product which is known as beta version. The aim of beta testing is to cover unexpected errors. It falls under the class of black box testing.

10.3 AUTOMATION TESTING

PyUnit

PyUnit is designed to work with any standard Python version 1.5.2 and higher. Unittest supports test automation, sharing of setup and shutdown code for tests, aggregation of tests into collections, and independence of the tests from the reporting framework. The unittest module provides classes that make it easy to support these qualities for a set of tests[11]. To achieve this, unittest supports some important concepts:

- **test fixture**

A *test fixture* represents the preparation needed to perform one or more tests, and any associate cleanup actions. This may involve, for example, creating temporary or proxy databases, directories, or starting a server process.

- **test case**

A *test case* is the smallest unit of testing. It checks for a specific response to a particular set of inputs. unittest provides a base class, TestCase, which may be used to create new test cases.

- **test suite**

A *test suite* is a collection of test cases, test suites, or both. It is used to aggregate tests that should be executed together.

- **test runner**

A *test runner* is a component which orchestrates the execution of tests and provides the outcome to the user. The runner may use a graphical interface, a textual interface, or return a special value to indicate the results of executing the tests.

The test case and test fixture concepts are supported through the TestCase and FunctionTestCase classes; the former should be used when creating new tests, and the latter can be used when integrating existing test code with a unittest-driven framework. When building test fixtures using TestCase, the setUp() and tearDown() methods can be overridden to provide initialization and cleanup for the fixture[11]. With FunctionTestCase, existing functions can be passed to the constructor for these purposes. When the test is run, the fixture initialization is run first; if it succeeds, the cleanup method is run after the test has been executed, regardless of the

outcome of the test. Each instance of the TestCase will only be used to run a single test method, so a new fixture is created for each test. Test suites are implemented by the TestSuite class. This class allows individual tests and test suites to be aggregated; when the suite is executed, all tests added directly to the suite and in "child" test suites are run[11].

A test runner is an object that provides a single method, run(), which accepts a TestCase or TestSuite object as a parameter, and returns a result object. The class TestResult is provided for use as the result object. unittest provides the TextTestRunner as an example test runner which reports test results on the standard error stream by default. Alternate runners can be implemented for other environments (such as graphical environments) without any need to derive from a specific class.

Table 10.1 Manual Testcases

Test Case ID	Description	Expected Results	Actual Results	Status
Test case 01	Click on hyperlink of topic name to view report	On clicking the link a page containing graphs should be opened	A new page containing graphs is opened	Pass
Test case 02	User gives rating	The rating should be between 0 to 5	User entered rating between 0 to 5	Pass
Test case 03	Enter comment in the given text field then click submit	Feedback submitted message should be displayed	Feedback submitted message	Pass

CHAPTER 11: APPLICATIONS

Sentiment analysis is extremely vital, particularly in the internet, as the sheer volume of the opinionated text on blogs, social networking sites etc. is increasing day by day. There is no practical way of efficiently analysing this huge data manually. However, sentiment analysis is important as human tend to depend on others opinions before making their own decisions, ranging from which product to buy to which song to listen. This human habit of depending on others is what makes sentiment analysis extremely important. Some of the important applications of sentiment analysis are as follows:

1 Electoral Predictions

Sentiment analysis can be used to predict election results. The probable winner of the elections can be predicted using social media data. Use of sentiment analysis in electoral predictions has increased now a day.

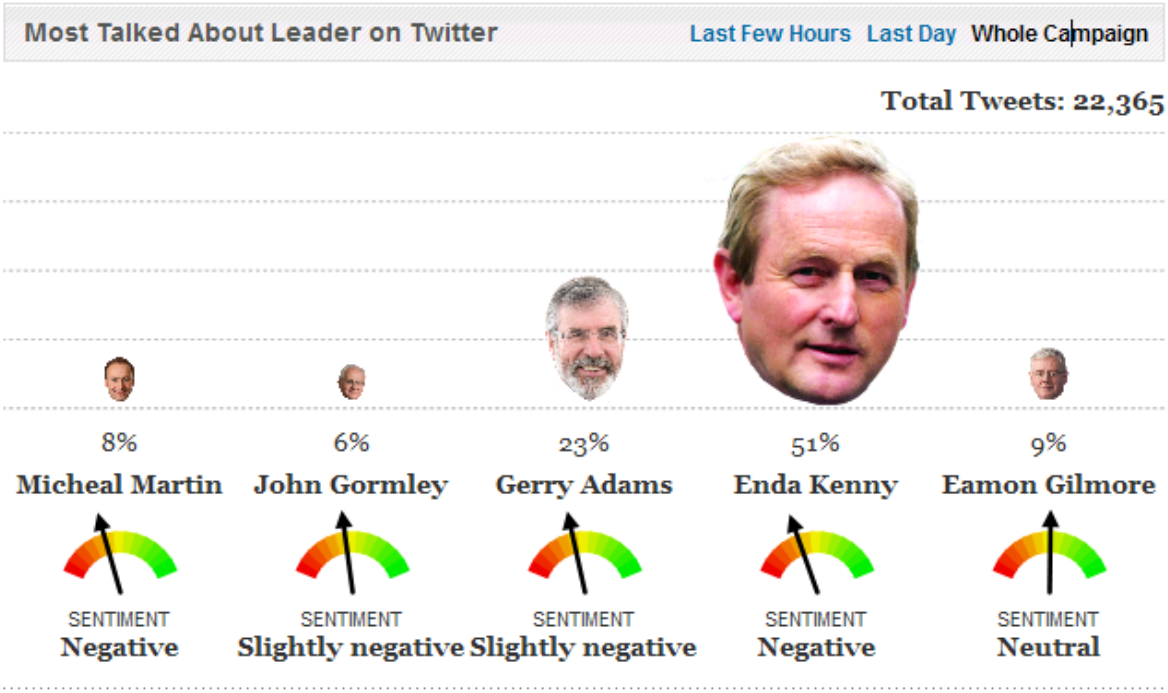


Fig 11.1 Electoral Prediction

2 Personality Traits

A person can be categorized into different personalities according to the traits he exhibits.

3 Stock Market Movements Prediction

Stock market prediction is one of the most widely researched topic today. Sentiment analysis can be used to analyse peoples sentiment towards a specific company, using which we can predict that company's stock market progress.

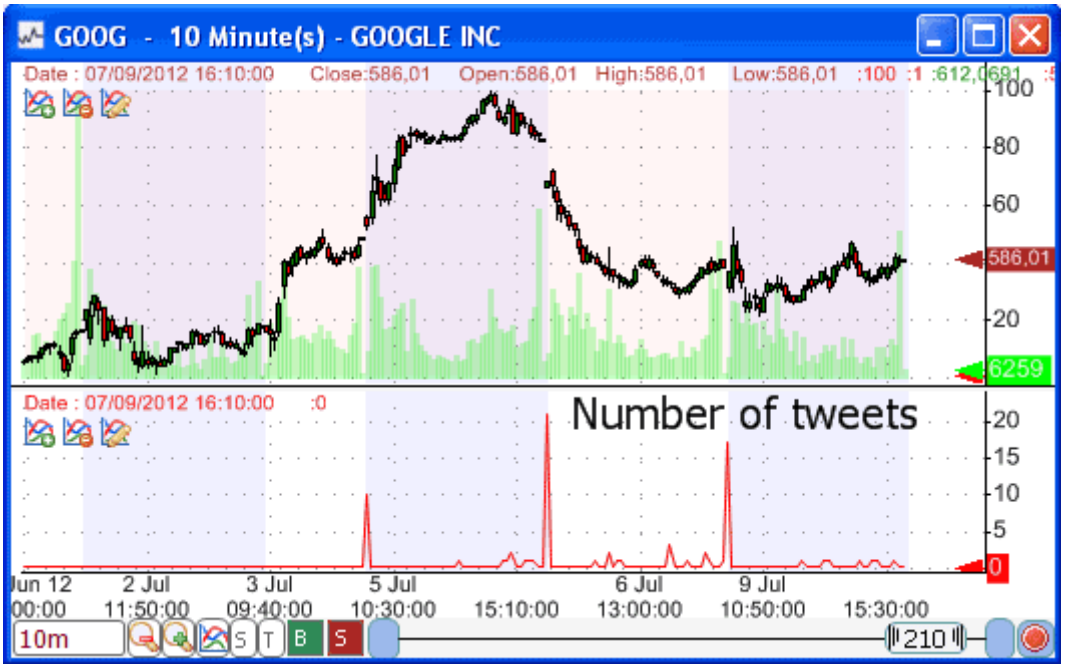


Fig 11.2 Stock Market Prediction

4 Business Intelligence Build up

Having insights-rich information eliminates the guesswork and execution of timely decisions. With the sentiment data about the established and the new products, it's easier to estimate customer retention rate[10]. Based on the reviews generated through sentiment analysis in business, one can always adjust to the present market situation and satisfy customers in a better way. Overall, one can make immediate decisions with automated insights. Business intelligence is all about staying dynamic throughout. Having the sentiments data gives that liberty. If one develops a big idea, they can test it before bringing life to it. This is known as concept testing[10]. Whether it is a new product, campaign or a new logo, just put it to concept testing and analyse the sentiments attached to it.

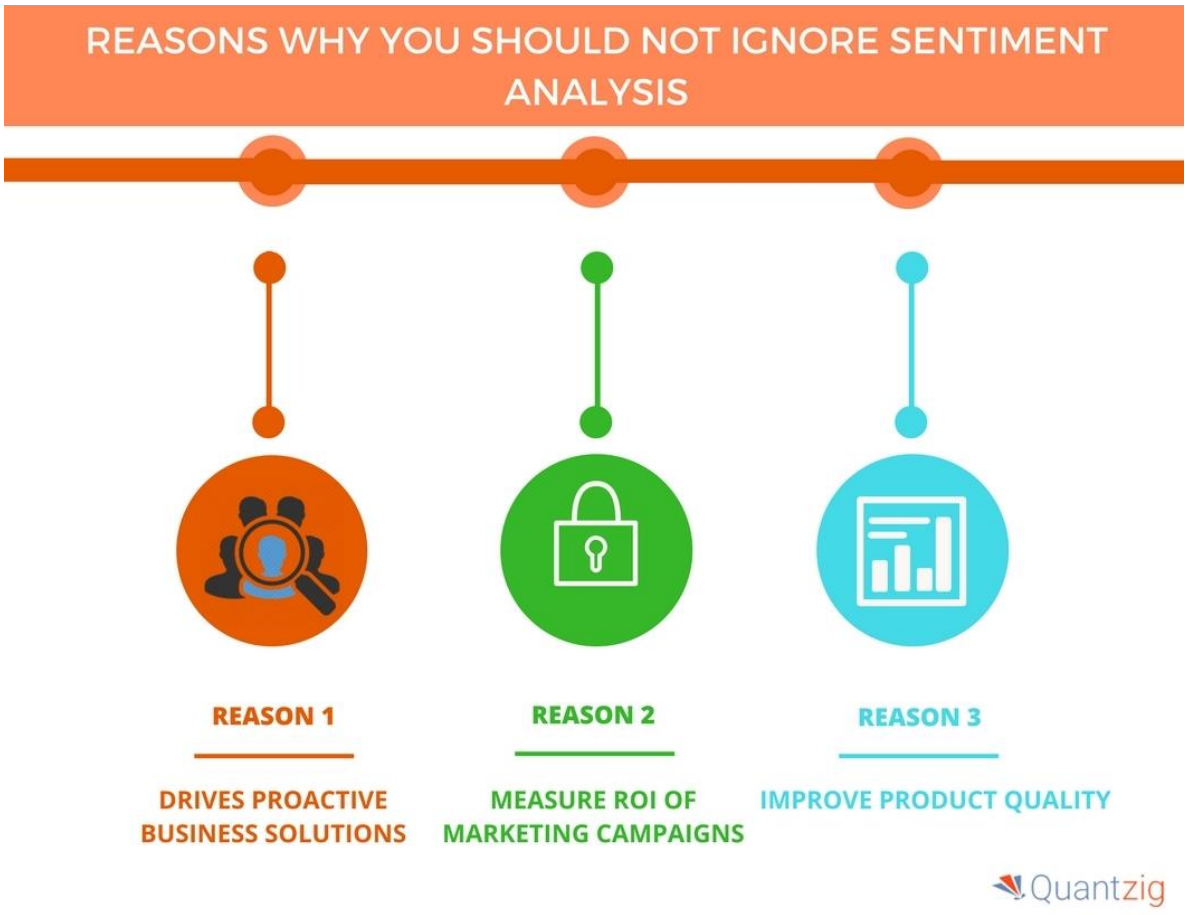


Fig 11.3 Applications of Sentiment Analysis in Business

5 Competitive Advantage in business

Getting x% negative or positive reviews on a certain product doesn't make much sense if one doesn't have a y% metric to compare it with[10]. Knowing the sentiment data of the competitors gives the opportunity as well as the incentive to perk up the performance. Sentiment analysis in businesses can be very helpful in predicting the customer trends. Once we are acquainted with the current customer trends, strategies can easily be developed to capitalize on them[10]. In addition, eventually, gain a leading edge in the competition.

6 Enhancing the Customer Experience

A business breathes on the gratification of its customers. The experience of the customers can be either positive, negative or neutral. Owing to the internet savvy era, this experience becomes the text of their social posting and online feedback[10]. The tone and temperament of this data can be detected and then categorized according to the sentiments attached. This

helps to know what is being properly implemented about products, services and customer support and what needs improvement. Getting a positive response to product is not always enough. The customer support system of company should always be impeccable no matter how phenomenal services are[10].

7 Brand Brisking

The product it manufactures or the services it provides do not define a brand. The name and fame that build a brand majorly depend on their online marketing, social campaigning, and content marketing and customer support services[10]. Sentiment analysis in business helps in quantifying the perception of the present and the potential customers regarding all these factors. Keeping the negative sentiments in knowledge, one can develop more appealing branding techniques and marketing strategies to switch from torpid to terrific brand status[10]. Sentiment analysis in business can majorly help us to make a quick transition.

CHAPTER 12: CONCLUSION

In this project, we studied various methods and approaches of sentiment classification.
After analysing all the methods, their advantages and disadvantages we concluded that machine learning approach is more efficient among all other methods. After experimentation, we found out that pre-processing tweet, to remove unwanted words, which do not express any sentiment, increases the accuracy. Unigram gives good result than bigram and trigram.

Future Work

- We conclude that the sentiment prediction accuracy will increase as the dataset increases.
- In this project, we focus on sentence level sentiment classification, but as a future work, it can be expanded to include feature/aspect level classification, which is useful in product review and recommendation system.
- The number of sentiment classes can be increased to get more refined sentiment prediction.
- Emoticons can be used as features to get more accurate sentiment prediction
- Processing of trends can be done runtime where the results are updated as soon as new trend arises during the day or on specific time intervals.
- This project works only with English language, but it can be extended to include other languages as well.
- In this project, labelling is done manually which increases time and cost. In future ways can be found to do labelling automatically.

REFERENCES

1. Afroze Ibrahim Baqapuri, "Twitter Sentiment Analysis",Department of Electrical Engineering, School of Electrical Engineering & Computer Science, National University of Sciences & Technology, Islamabad, Pakistan 2012

2. Anuja P Jain, Asst. Prof Padma Dandannavar, "Application of Machine Learning Techniques to Sentiment Analysis", proceedings of the *2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*

3. Martin Sarnovsky, Peter Butka, Andrea Huzvarova, "Twitter data analysis and visualizations using the R language on top of the Hadoop platform", *SAMI 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics January 26-28, 2017 Herl'any, Slovakia*

4. Andrei Sechelea, Tien Do Huu, Evangelos Zimos, and Nikos Deligiannis, "Twitter Data Clustering and Visualization", proceedings of the *2016 23rd International Conference on Telecommunications (ICT)*

5. Rohit Joshi, Rajkumar Tekchandani, "Comparative Analysis of Twitter Data Using Supervised Classifiers", proceedings of the *2016 International Conference on Inventive Computation Technologies (ICICT)*

6. Mr. Abhijit Janardan Patankar, Dr. Kshama V. Kulhalli, Dr. Kotrappa Sirbi "Emotweet: Sentiment Analysis tool for twitter" proceedings of the *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*

7. Walaa Medhat , Ahmed Hassan , Hoda Korashy , "Sentiment analysis algorithm and applications: A survey", Ain Shams Engineering Journal Volume 5, Issue 4, December 2014, Pages 1093-1113

8. www.tutorialspoint.com

9. www.quora.com

10. www.blog.paralleldots.com

11. www.docs.python.org

12. www.smartdraw.com

13. www.docs.tweepy.org