ABSTRACT

Hashtag investor is a system that can analyze twitter data to generate useful information including some predictions. Machine learning techniques have been used for this research which falls into data mining to archive sentiment analysis to categorize and identify tweets based on the contents. Twitter has an enormous collection of data. If these data is converted into some useful information, accurate decisions can be made using this data. That is our main objective, which can be very helpful to users, and this system works with respect to four specific objectives. One objective is sentimental analysis of twitter data and finding false tweets. Supervised learning has been used and NLTK and also the naïve Bayes classifier has been used as techniques. The output will be display percentage wise, negative positive and neutral percentages of the given keyword. Twitter data is analyzed according to the given keyword. False tweets identification is done by analyzing user profile. If the user profile criteria does not match with our assumptions this profile is marked as a fake profile. Second objective is comparing two similar products and getting the popularity according to the time. The output is displayed by charts. Similar keywords will be grouped. Clustering algorithms has been used for grouping. Our forth objective is finding some latest ongoing events and the number of users who were active at certain time periods, ARIMA model has been used as the technique. Our final objective is to analyze retweets comments and tweets on particular two products. Output is displayed as a graph. Propagation topology is used as the technique for retweet analysis and exponential regression function is used for popularity prediction.

TABLE OF CONTENT

1. INTRODUCTION

1.1 Background

Twitter is an online news and social networking service where users post and interact with messages, "tweets," restricted to 140 characters. Registered users can post tweets, but those who are unregistered can only read them.

In the past few years, there has been a huge growth in the use of microblogged platforms such as Twitter. Spur by that growth, companies and media organizations are increasingly seeking ways to mine Twitter for information about what people think and feel about their products and services. Many times, these companies study user reactions and reply to users on microblogs. One challenge is to build technology to detect and summarize an overall sentiment

Twitter usage statistics shows that there are on average 350 million tweets per day, 140 million active users and 50 000 000 processed tweets. This is a massive data resource for big data analytics.

According to the brief research conducted, there is no proper means of analyzing or querying twitter to gain access to valuable statistical information regarding certain topics and accounts by a general user. The few that is available also do not provide *sentiment* analysis of the queried search criteria.

In this research, we look at build models for classifying “tweets” into positive, negative and neutral sentiment. We build models for two classification tasks: a binary task of classifying sentiment into positive and negative classes and a 3-way task of classifying sentiment into positive, negative and neutral classes.

This Research is mainly focused on analyzing data retrieved from twitter, and displaying relevant information in a comprehensive and visually appeasing manner to the user. This includes the analysis of individual tweets related to a certain topic, and the analysis of twitter accounts. The project also features the ability to compare two topics, analyze the tweet behavior with relation to geographical locations. Main reasons for getting our applications popular, first is the wide range of commercial and social service to Organizations.

1.2 Literature Review

Twitter is different to other forms of raw data, which are used for sentiment analysis as sentiments are conveyed in one or two sentence blurbs rather than paragraphs. Twitter is much more informal and less consistent in terms of language. Users cover a wide array of topics, which interest them and use many symbols such as emoticons to express their views on many aspects of their life. When using human generated status updates, sentiment is not always obvious; many tweets are ambiguous and can use humor to maximize the opinion to other human readers but deflect the opinion to a machine-learning algorithm[11]. Another consideration when using a dataset generated from Twitter is that a considerably large amount of tweets which convey no sentiment such as linking to a news article, which can lead to difficulties in data gathering, training and testing [12].Sentiment analysis provides a means of tracking opinions and attitudes on the web and determines if the public positively or negatively receives them.

According to Mejova (2009) [13] Sentiment analysis is usually conducted between two levels; a coarse level and a fine level. Coarse level sentiment analysis deals with determining the sentiment of an entire document and Fine level deals with attribute level sentiment analysis. Sentiment analysis in Twitter provides a dramatically different data set where multiple interesting challenges can arise.

Symbolic techniques and Machine Learning techniques are the two basic methodologies used in sentiment analysis from text [14].

The next two sections deal with these techniques in further detail.

A. Symbolic Techniques

Symbolic techniques in supervised classification models make use of available lexical resources. In his sentiment analysis he used bag-of-words approach. In this approach the document was treated as a collection of words where relationships between words are not considered important. To determine the overall sentiment, sentiments of every word are given a value and using aggregation functions, those values are combined. Where tuples are phrases having adjectives or adverbs which may be considered positive or negative, he found the polarity of a review was based on the average semantic orientation of tuples extracted from the review [15].

WordNet which is a database consists of words and their relative synonyms were used Jaap Kamps [16]. In this study a distance metric was developed on WordNet and the semantic orientation of adjectives was determined from this metric. In their study, Mr A Balahur (2012) introduced a conceptual representation of text, which stored the structure and the semantics of real events, in a system called EmotiNet. Emotinet was able to identify the emotional responses triggered by actions with the information it stored [17].

The difficulty with using a Knowledge base approach however that is it requires of a large lexical database. This has become harder and harder to provide as the language of social networks is so trend dependent and changeable that lexicon datasets cannot keep up. Therefore, Knowledge based approaches to sentiment analysis are not as popular as they used to be.

B. Machine Learning Techniques

In contrast to Knowledge based approaches Machine Learning techniques are not dependent on a lexicon dataset, instead the use of a training set and a test set in order to classify is employed. This allows the algorithm to remain dynamic in the face of ever changing social network language lexicons. In this methodology a classification model is developed using a training set, which tries to classify the input feature vectors into corresponding class labels. A test set is used to prove the model by predicting the class labels of unseen feature vectors as outlined in the training set.

A number of machine learning techniques like Naive Bayes (NB) and Support Vector Machines (SVM) are used to classify reviews into either positive or negative orientation. In their paper Naive Bayes works as a good classifier for certain problems as it results in highly dependent features[18].

A new model which was based on Bayesian algorithm was introduced in 2012 by Zhen Niu (2012). In this model weights of the classifier are adjusted by making use of representative feature (information that represents a class) and unique feature (information that helps in distinguishing classes). Using those weights, the researchers calculated the probability of each classification. This allowed for an improved Bayesian algorithm [19].

Alexander Pak and Patrick Paroubek created a twitter corpus by using a Twitter API which automatically collected tweets from Twitter as well as annotating those using emoticons. Using that corpus, they built a sentiment classifier which used N-gram and POS-tags as features based on the multinomial Naive Bayes classifier [20].

In combining various feature sets and classification techniques an Ensemble framework is created. Rui Xia [21] used an ensemble framework for sentiment classification in their paper were two types of feature sets and three base classifiers were used to form the ensemble framework. Part-of-speech information and Word-relations were used to create two types of feature sets [21]. Three base classifiers Naive Bayes, Maximum Entropy and Support Vector Machines were also selected. Fixed combination, weighted combination and Meta-classifier combination ensemble methods for sentiment classification were applied and measured so as to obtain better accuracy. When the classifiers were measured separately Naive Bayes was found to be the most accurate. [8]

In this paper, we construct a Twitter corpus using Twitter API, use R studio coding to prepossess the Twitter corpus, then using know ledge based methods we use an available lexical resources and apply it to the Twitter corpus. To compare the results from the knowledge based method to a machine learning technique we then use Naive Bayes classification models to the corpus which will split the corpus into positive and negative tweets as well as highlighting which tweets are classified. Naïve Bayes is used as it is often works well as a good first classifier in data analysis.

Throughout this whole survey regarding tweet analyzing we come across with some main technologies like sentimental analysis and big data which are used to analyze tweets. These technologies are used in some research done in past like

Sentiment Analysis for Product Rating-by Nivon Projects

This project proposed an advanced Sentiment Analysis for Product Rating system that detects hidden sentiments in comments and rates the product accordingly. The system uses sentiment analysis methodology in order to achieve desired functionality. This project is an E-Commerce web application where the registered user will view the product and product features and will comment about the product. System will analyze the comments of various users and will rank product. We use a database of sentiment based keywords along with positivity or negativity weight in database and then based on these sentiment keywords mined in user comment is ranked. Comment will be analyzed by comparing the comment with the keywords stored in database. The System takes comments of various users, based on the comment, system will specify whether the product is good, bad, or worst. Once user login the system he can view the product and product features. After viewing product user can comment about the product. User can also view comment

of other user's. The role of the admin is to add product to the system and to add keywords in database. User can easily find out correct product for his usage. This application also works as an advertisement which makes many people aware about the product. This system is also useful for the users who need review about a product. [9]

"AIDSvu"- by a team of researchers at Emory University's Rollins School of Public Health.

AIDSVu is an interactive map that illustrates the prevalence of HIV in the United States. The data is pulled from the U.S. Center for Disease Control's national HIV surveillance reports, which are collected at both state and county levels each year.

The site's massive database lets you view individual state profiles. It breaks HIV prevalence rates by race and estimates percentages of people living with HIV in certain areas, making it easy to find testing sites close to home. You can view the full country map or search for individual cities

The main data is updated once a year, in correlation with the reports, but other data testing locations that spring up throughout the year, for example are updated as soon as the information becomes available. [7]

MixedEmotions: Social Semantic Emotion Analysis for Innovative Multilingual Big Data Analytics Markets

MixedEmotions will develop innovative multilingual multi-modal Big Data analytics applications that will analyze a more complete emotional profile of user behavior using data from mixed input channels: multilingual text data sources, A/V signal input (multilingual speech, audio, video), social media (social network, comments), and structured data. Commercial applications (implemented as pilot projects) will be in Social TV, Brand Reputation Management and Call Centre Operations. Making sense of accumulated user interaction from different data sources, modalities and languages is challenging and has not yet been explored in fullness in an industrial context. Commercial solutions exist but do not address the multilingual aspect in a robust and large-scale setting and do not scale up to huge data volumes that need to be processed, or the

integration of emotion analysis observations across data sources and/or modalities on a meaningful level. mixedemotions will implement an integrated big linked data platform for emotion analysis across heterogeneous data sources, different languages and modalities, building on existing state of the art tools, services and approaches that will enable the tracking of emotional aspects of user interaction and feedback on an entity level. the mixedemotions platform will provide an integrated solution for: large-scale emotion analysis and fusion on heterogeneous, multilingual, text, speech, video and social media data streams, leveraging open access and proprietary data sources, and exploiting social context by leveraging social network graphs; semantic-level emotion information aggregation and integration through robust extraction of social semantic knowledge graphs for emotion analysis along multidimensional clusters. [10]

1.3 Research Gap

Throughout the literature survey we found some missing element in the existing research literature. Following chart shows how above existing products are familiar with the areas we are going to perform in our research.

	Sentiment Analysis for Product Rating	**AIDSvu**	**Social Semantic Emotion Analysis**	**Proposed solution**
Sentiment Analysis	✓	✗	✓	✓
Positive score	✗	✗	✗	✓
Neutral tweets	✗	✗	✗	✓
Vulgar Tweets	✗	✗	✗	✓
Analyzed Tweets	✗	✗	✓	✓
Shows most and least Tweet of above categories.	✗	✓	✗	✓
Specify whether the product is good, bad	✓	✓	✗	✓
Show review	✓	✓	✓	✓
View in map	✗	✓	✗	✓
Big Data Analytics	✗	✓	✓	✓
Topic comparison	✗	✗	✗	✓
Set up charts and graphs	✗	✗	✓	✓

1.4 Research Problem

Twitter is a free social networking Social media service that allows registered members to broadcast short posts called tweets. Twitter members can broadcast tweets and follow other users' tweests by using multiple platforms and devices.

According to the research conducted, there is no proper means of analyzing or querying twitter to gain access to valuable statistical information regarding certain topics and accounts by a general user. The few that is available also do not provide sentiment analysis of the queried search criteria

There’s no real way of analyzing this meta data in terms of “emotions”, “feelings”, “desire” and “health”, “rise and fall of market”,” interest” of modern day social network users. Even if that was somehow possible for the average internet user, advanced querying and visualization is not elaborative and comprehensive enough because of twitter data is so powerful, so valuable.

2. OBJECTIVES

2.1 Main objective

Twitter has an enormous collection of data. If these data is converted into some useful information, accurate decisions can be made using this data. That is our objective, which can be very help to users. This is more useful as the details of a particular location are displayed using Google maps for accurate decision making.

2.2 Specific Objectives

2.2.1 To analyze according to the user's topic the system will give particular location with relation to tweet behavior positive, negative, neutral

According to the geo location, users can get positive and negative classifications on a particular topic which they look for. Percentages will appear in graphs and also system will generate/ present similar topics to the user with respect to the topic that user looked for. Further, the system will identify and avoid false tweets generated by software tools

2.2.2 To identify and compare the topic with respect to the users desire and give analysis

System Compares two topics by analysing the tweets related to them in order to give an overall analysis about the given topics supplement with geographical categorization. In here system will be able to compare similar type of topics only because it's not possible to compare iPhone with laptop.

2.2.3 To analyze the time evolution of the tweets in topics

System analyses given topic in tweet change according to a given time period and geo locations. How well-liked are certain topic posts at different times in years. Impressions and views are always a nice metric to keep an eye on, and users will also probably be interested in how engaged your audience is with your sharing. This stat can be helpful in figuring out when your followers are most engaged during over the years

2.2.4 To analyze about specific events and analyze selected account and relation of followers geographical location of a given account

System mainly analyze particular event through our application and can identify which are the time periods users are mainly active in those events. We have planned to analyze tweets based on given event period to forecast and guide user's behaviors. Real time analysis on the tweet to predict and forecast current affairs.According to this tweet analyze broadcasters can get a clear idea about people's perception at the time period in the event also they can get sentimental analysis about those tweets. And we can predict through machine, learning how their next event will attract users actively through time periods

2.2.5 Create API base on requirement

We have created a API for users who are using our application. Using the API provided by us any user can obtain information they needed to analyze customers reviews about their products.

2.2.6 Create database using twitter data

There is a huge amount of data collected gradually with the time. These data is difficult to collect at once .we stored this data in the DB we have created in mongo .the redundant and unnecessary data will be eliminated here the data can be processed efficiently in that way.

3. METHODOLOGY

Our goal was to build a system that can analyze twitter data to generate useful information including some predictions. As a solution, we have done a research on giving knowledge to the world using twitter data. In our point of view this project has considerable amount of research areas. This project includes big data analyzes, sentimental analysis, machine learnings.

If we talk about machine learning. Machine learning is a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed. The process of machine learning is similar to that of data mining. We can divide this process in to two parts. Supervised learning and unsupervised learning.

Supervised and unsupervised in the sense means, Suppose you are providing solution to your kids for each and every situation in their life, it is called your kids are supervised. But, if your kids take their decisions out of their own understanding, it is called your kids are unsupervised.

In machine learning, such solutions are called target or output and situations are called input or unleveled data. Situation and solution in combination it is called leveled data.

If we talk about technique wise supervised in the sense means, if you are training your machine learning task for every input with corresponding target, it is called supervised learning and if you are training your machine learning task only with a set of inputs, it is called unsupervised learning, which will be able to find the structure or relationships between different inputs.

So in this research we have used supervised learning. Each and every input we have programed an output. Going deeper, in supervised learning we have done Sentiment analysis.

Sentimental in the sense means the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study effective states and subjective information.in here we have used some techniques to identify sentences.

Unigram, bigram and trigram are the techniques which we have used. In unigram It can regard words one by one.in bigram It can regard words two at a time. Each two adjacent words create a bigram. In trigram it can regard words three at a time. Each three adjacent words create a trigram.

Big data analytics is the process of examining large data sets to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information [1]. Twitter is consist with enormous database and it grows day by day

Twitter has an enormous collection of data. If these data is converted into some useful information, accurate decisions can be made using this data.

To twitter analysis we have used data mining analytics.do Data Mining is extraction of knowledge hidden from large volumes of raw data. Knowledge discovery varies from traditional information retrieval from databases. In a traditional DBMS, database records are returned in response to a query; while in knowledge discovery, what is retrieved is implicit patterns [2]. The process of discovering such patterns is termed data mining.

Two main reasons to use data mining:

• Large amounts of data that can't be handled by individuals

• Need of making significant extrapolations from large sets of data

Mainly we have used data mining techniques in our research are Regression Analysis techniques and Classification Analysis [2].

Classification Analysis This analysis is used to retrieve important and relevant information about data, and metadata. It is used to classify different data in different classes. Classification is similar to clustering in a way that it also segments data records into different segments called classes. So, in our research classification analysis we have applied algorithms to decide how new data should be classified Regression Analysis In statistical terms, a regression analysis is the process of identifying and analysing the relationship among variables. It can help understand the characteristic value of the dependent variable changes, if any one of the independent variables is varied. This means one variable is dependent on another, but it is not vice versa. It is generally used for prediction and forecasting. There for we can use these algorithms in our research to predict on time based an event based twitter analysis [1, 2, and 3]

In this research, we are using python as the programing language and mongo DB as the database. Our first step is to collect data from twitter which is not an easy task. We have used the Twitter API to collect tweets and other data from their site. The REST API, the Search API, and the

Streaming API are the three methods we found to get this data. The Search API allows you to search old tweets the REST API allows you to collect user profiles, friends, and followers, and the Streaming API collects tweets in real time as they happen. There were some restrictions when getting data through API. So, that we have used our own database.

First, we had to get API keys from twitter developers' site. The authentication requires that we get an API key from the Twitter developers' site. The authentication gives permission to our program to make API calls.

JSON files are the data structure that Twitter returns. These are rather comprehensive with the amount of data, but hard to use without them being parsed first. We are using NoSQL database like MongoDB to store and query your tweets.

If we talk about what MongoDB does, it is a document-based database that uses documents instead of tuples in tables to store data. These documents looks like just like JSON objects using key-value

MongoDB represents JSON documents in binary-encoded format called BSON. BSON extends the JSON model to provide additional data types, ordered fields, and to be efficient for encoding and decoding within different languages.

Some of the functionalities in our application required to show various data sets in maps (world maps). The functionalities which use this are Analyze Topic, Compare Topics and Analyze Accounts.

When comes to the geo locations. It shows the geo location using Google maps according to Twitter accounts (profiles) .In Analyzing Twitter Accounts, shows the location of the Tweeter followers as mentioned in the profile by the profile owner.

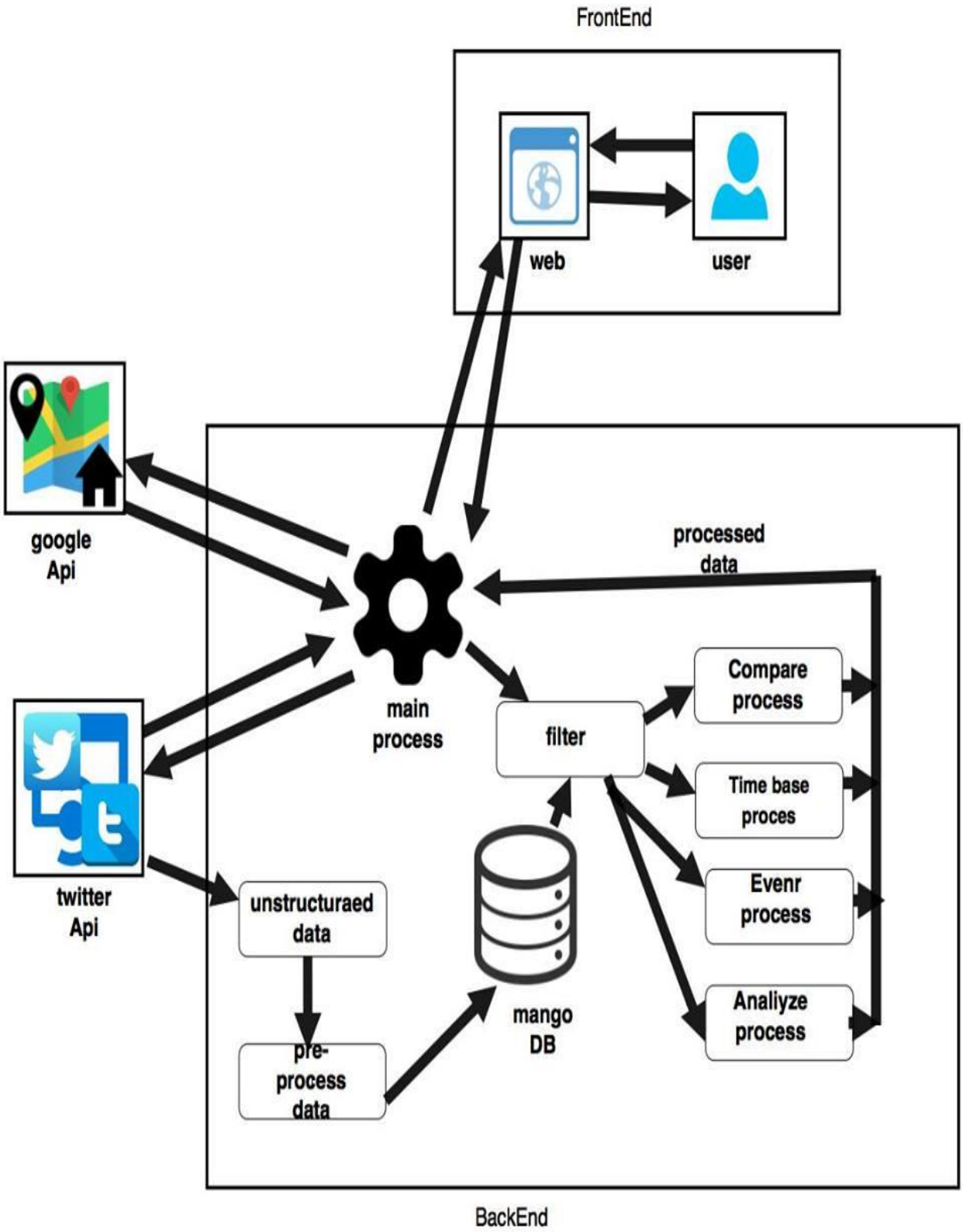

Figure 1-System Diagram

3.1 Text Analysis

The input of this Topic analysis component is a keyword according to the user's choice. Then the system will be analyze that keyword and find tweets which contains that keyword. Then the system analyze that particular tweets and finds out whether it is a positive negative or neutral and also the system detects the false tweets.

Tweets in English language were selected for our experiments. This can be done by sending request to the twitter site requesting only English language tweets. Then the system will start to analyze the twitter details. User details, geolocation text etc. when it comes to the text analysis system needs to be identified if the particular tweet is a positive negative or a neutral one. For that we have come up with an algorithm. And this algorithm has been created using NLTK and naïve bays classifier which is already in build algorithm in NLTK.

NLTK is a leading platform for building Python programs to work with human language data which provide many facilities. Text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

The greatest challenge in this component is to find a data set and train that data set.so and accurate data set was found and it has been labeled by an expert whether it is positive negative and neutral. Hence we received this data and remove unwanted data and are store them in an arrays.

Examples:-

Good
Awesome
Excellent
love

Positive review Array

person
mobile
ok
tell

neutral review Array

Bad
Hate
Kill
sad

Negative review array

Then we obtain the word from the array and create our own vocabulary. The initial part in this is to match that twitter text with our vocabulary and to find out the matching words and parse them to naïve bays classifier.

Examples:-

Good
Awesome
Bad
Hate
Person
ok

Vocabulary list

Then the word are got from the word in each arrays and a list is created called training _data. Each words is labeled weather it a positive negative or neutral.

Examples:-

good	positive
hate	negative
person	neutral
love	positive

Trained data set

Now we come to the probability calculating part. For this we used naïve bays algorithm. This algorithm is in build functioned in the NLTK.using naïve bays classifier this algorithm. Can be used

Naïve bays algorithm uses bays theorem, bays theorem stared that "It works on conditional **probability**. Conditional probability is the probability that something will happen, *given that something else* **has already occurred**. Using the conditional probability, we can calculate the probability of an event using its prior knowledge" .this theorem used in naïve bays classifier. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is

considered as the most likely class. Naive Bayes classifier assumes that all the features are **unrelated** to each other. Presence or absence of a feature does not influence the presence or absence of any other feature. We can use Wikipedia example for explaining the logic

"A fruit may be considered to be an apple if it is red, round, and about 4″ in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple."[1]

This classifier takes tweets text as an input and compare with the trained data set, then the output will state weather this is text is a positive negative or a neutral one. Here it is shown how whether the particular words occurrence are according to the probability. Trained data set will act as the evidence of a problem. According to this evidence that the output will be given. If it's possible to lactate a big data set the output will be very much accurate.

```
def extract_features(self, review):
    review_words = set(review)
    features = {}
    for word in self.vocabulary:
        features[word] = (word in review_words)
    return features
```

Figure. 1. (Lines of codes of word extraction)

```
def naive_bayes_sentiment_calculator(self, review):
    problem_instance = review.split()
    problem_features = self.extract_features(problem_instance)
    return self.trained_NB_Classifier.classify(problem_features)
```

Figure. 2. (Lines of codes of execution of the methods)

In figure 1 the lines of codes check the given text of the tweets words with our vocabulary and the matching words will be taken. Figure 2 the lines of codes show execution of figure 1 code and get the output of that code and execute the trained classifier which compared the matching words with the trained data set. Then the main output will send

3.1.2. Emoji analysis

When it comes to the emoji analysing part, first the system will process the tweet text and check if the particular tweet text contains some emojis, for that we had used emoji library. Figure 4 will show the way of doing it. If any emoji's are found those emojis will be sent to main algorithm.

```
def extract_emojis(self, text):
    temp = []
    for c in text:
        if c in emoji.UNICODE_EMOJI:
            temp.append(c)
    return temp
```

Figure. 3. (Lines of codes of emoji identification)

Initial part of the main algoritham system will open a sample emoji data set and get the emoji face, positive score, negative score, neutral score and the total score. Those details will be stored in our own emoji list. Then these emojis are checked with previously found emojies. if any matching emojis are identifiend, the scores of the emojies will get and adding to previous score. Then the sytem will check which score will be the highest. The highest score will be the result. Figure 6 will show the way.

```
conclusion = ""
if pr_positive > pr_neutral and pr_positive > pr_negative:
    # positive
    conclusion = "positive"
elif pr_neutral >= pr_positive and pr_neutral >= pr_negative:
    # negative
    conclusion = "neutral"
elif pr_negative > pr_positive and pr_negative > pr_neutral:
    # neutral
    conclusion = "negative"

# return self.trained_NB_Classifier.classify(problem_features)
return conclusion
```

Figure. 4. (Lines of codes of conclusion identification)

Example :-

```
Emoji,Unicode codepoint,Occurrences,Position,Negative,Neutral,Positive,Unicode name,Unicode block
😂,0x1f602,14622,0.805100583,3614,4163,6845,FACE WITH TEARS OF JOY,Emoticons
❤,0x2764,8050,0.746943086,355,1334,6361,HEAVY BLACK HEART,Dingbats
♥,0x2665,7144,0.753806008,252,1942,4950,BLACK HEART SUIT,Miscellaneous Symbols
😍,0x1f60d,6359,0.765292366,329,1390,4640,SMILING FACE WITH HEART-SHAPED EYES,Emoticons
😭,0x1f62d,5526,0.803351976,2412,1218,1896,LOUDLY CRYING FACE,Emoticons
😘,0x1f618,3648,0.854480172,193,702,2753,FACE THROWING A KISS,Emoticons
😊,0x1f60a,3186,0.813302436,189,754,2243,SMILING FACE WITH SMILING EYES,Emoticons
👌,0x1f44c,2925,0.805222883,274,728,1923,OK HAND SIGN,Miscellaneous Symbols and Pictographs
💕,0x1f495,2400,0.765725889,99,683,1618,TWO HEARTS,Miscellaneous Symbols and Pictographs
```

Sample emoji data set

3.1.3 False tweet analysis

False tweets finding part is done using user profile analysis. System checks our predefined conditions with the user account. If any user account matches with the conditions, then it will give a rating to a particular account. The conditions are if the users' bio data is an empty one, users account creation time period is less than 24 hours, users' number of followers count is less than 10, users tweet amount is greater than 10000 and finally user's friend's count is less than 10. Those are the conditions that we have used.

Example:-

```
# ratings for fake identification
FAKE_DATE = 0.40
FAKE_BIO = 0.10
FAKE_FOLLOWING = 0.25
FAKE_TWEETS_COUNT = 0.20
FAKE_FOLLOWERS_COUNT = 0.05
```

Rating conditions

First we equal the user account to 1.then the algorithm checks each condition one by one. First it will check if the user account is already a black listed one and if not then checks with other conditions. The conditions that we have specified previously are rated. When going down of the algorithm those conditions will be checked. If one or more conditions are satisfied the user account value will be reduced. This will happen until all the conditions are checked. Then algorithm checks the final score of the user account. If the value is less than 0.6 then it will be labelled as a fake user account. All the real user accounts and the fake accounts will be stored in separate arrays and will be sent as an output.

3.1.4 Topic comparison

In Topic comparison, system will analyze two topics given by the user and display Popularity of relevant topic according to number of followers, likes, comments and retweets. And also it shows the popularity of each keywords with geographical view. The input of this module are 2 keywords according to user choice. The system will analyses all tweets which includes the keywords given by user and view its popularity as a percentage. And also, it shows the popularity of the topics given by user in graph with given period of time. In order to do this I have use sentimental Analysis methods.

Sentiment analysis 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 [6]. To take popularity by comments we have to analyze comments and extract positive comments. Sentimental analysis is common to express this as a classification problem where a given text needs to be labeled as Positive, Negative or Neutral [7].

Retweet process is divided in to two parts

1.Retweet from original tweet

2.Retweet from retweeted tweet

Tweeter API only gives retweet from original tweet. Therefore, we have come up with a method to get count of Retweet from Retweeted tweet. To study the popularity of Topic from super nodes, we need to build the retweet propagation topology for a piece of Keyword, where nodes are connected by retweeting paths. However, as mentioned previously, Twitter API does not provide the previous parent retreaters of a user's retweeted, so it is difficult to identify the entire retweeting path of a retweet.

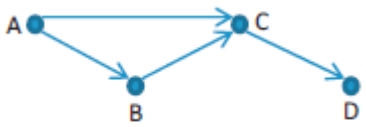

Figure 5-Retweet Path

For example, in Figure 1, user D receives the news from user C. User C can receive the news from user A, from B, or from both. Then, it is difficult to determine whether the retweet that user D received from user C is originally from A or B. Thus, it is a challenge to determine where the retweet is from for each retweeted for each tweet in our trace. We then build the retweet propagation topology indirectly

with two assumptions without the loss of generality listed below.

1. A user retweets a piece of news only when (s)he sees the tweet at the first time.

2. There is a time delay between when a user sees a piece of tweet and when the tweet was created/published.

The users that a user A follows are called user A's friends. For a given tweet, suppose V is a set of retweet nodes sorted by the retweeting time for a tweet, v0 is the source node (i.e., supernode)

of the tweet, Fvi is the set of friends of vi and Lvi is the set of retweet nodes retweeted before node vi. tvi is the time that vi retweeted the news since the publish time t0. We can get the values of all the mentioned parameters from our crawled data. We use ϑ to denote the users' response delay, defined as the time elapsed after a user sees a tweet and before (s)he retweets the tweet. We assume that the retweeter of a user is the user's friend, which is true in most cases. In order to find the retweet topology relationship for the topology construction of a given tweet, for each retweeter vi for the tweet, we need to find vi's parent retweeter in the topology that retweets to vi. Based on the above assumptions, we develop the following algorithm for this purpose:

1. For each vi, if Lvi $\cap$ Fvi= $\emptyset$, vi retweeted the news from v0 because none of vi's friends retweeted the tweet.

2. If Lvi $\cap$ Fvi$\neq \emptyset$, we sort the subset Lvi $\cap$Fvi $\cup$v0 by retweeting time and select the node with the latest retweeting time that is smaller than tvi$-\vartheta$ as the parent of node vi.

Recall our trace data includes all retweets for a tweet, by finding each retweet's parent using this algorithm, we can finally construct the retweet propagation topology of this tweet. In this model, the user's response delay ϑ is the main factor that might lead to an imprecise topology since the users' response delay may change in a relatively large range due to various reasons. Since the latest retweeting happens much earlier than their children's retweeting time in most situations, ϑ

is very relatively small and negligible. Therefore, this algorithm can help find precise retweeting path in most situations. [8]

After visualizing popularity of two keywords given by user system will show a popularity prediction for next hours. For that we used logistic regression Method to show a cumulative growth of prediction with 99% accuracy [9].

An explanation of logistic regression can begin with an explanation of the standard logistic function. The logistic function is useful because it can take any real input t, (t |R), whereas the output always takes values between zero and one [14] and hence is interpretable as a probability. The logistic function is defined as follows:

Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary value (0 or 1) rather than a numeric value.

Below is an example logistic regression equation:

y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

The actual representation of the model that you would store in memory or in a file are the coefficients in the equation (the beta value or b's) [9].

We are using graphs to represent all these data and predictions with the help of above methods. Because as a general data structure, graphs have become increasingly important in modeling event analyze structures and their interactions. Graph mining techniques and tools are then used to discover the required information for data visualization. Graph mining is the process of gathering and analyzing data represented in graphs.

Visualized the data using Zing chart library

We are using graphs to represent all these data and predictions with the help of above methods. Because as a general data structure, graphs have become increasingly important in modeling event

analyze structures and their interactions. Graph mining techniques and tools are then used to discover the required information for data visualization. Graph mining is the process of gathering and analyzing data represented in graphs

3.1.5 Event analysis and Account analysis

In event analyze function, system will mainly analyze particular event through our application and can identify which are the time periods users are mainly active in those events. Our system has capability to analyze tweets based on given event period to forecast and guide user's behaviors. Real time analysis on the tweet to predict and forecast current affairs, based can get locations and categories.

- **Threads creating**

```
GROUP_TIME_UNIT = 1140  # minutes (1 day)


def analyze_events(data):
    res = {}
    # threads
    threads = []
    now = datetime.now() + timedelta(days=1)
    for day_index in range(0, 10):
        crr_date = now - timedelta(days=day_index)
        t = threading.Thread(name="search_topic_" + str(day_index), target=worker_1, args=(crr_date, res, data))
        t.start()
        threads.append(t)

    for thread in threads:
        # join threads
        thread.join()

    # process tweets
    all_tweets = {}
    for key in res:
        for tweet in res[key]:
            if tweet.id_str not in all_tweets:
                # tweet not found
                all_tweets[tweet.id_str] = tweet
            # check on re-tweet
            if hasattr(tweet, 'retweeted_status'):
                if tweet.retweeted_status.id_str not in all_tweets:
                    # tweet not found
                    all_tweets[tweet.retweeted_status.id_str] = tweet.retweeted_status
```

Figure 6 -threads creation for 10 days

To download twitter data we are using multi-threading approach. Therefor we have selected 10 days in event analysis and one thread will create for one day. There for 10 threads will create for 10 days. Then we had to remove retweet data and junk tweets because it will help avoid redundancy and can get more accurate results.

❖ **Timestamp creating**

```
# make tweet time line
tweet_timeseries = dict()
for key in all_tweets:
    timestamp = all_tweets[key].created_at.timestamp()
    if timestamp not in tweet_timeseries:
        tweet_timeseries[timestamp] = []
    tweet_timeseries[int(timestamp)].append(key)

grouped_time_series = dict()
pre_time = 0
# get all time stamp keys
for timestamp in sorted(tweet_timeseries.keys()):
    date_diff = now - datetime.fromtimestamp(timestamp)
    if date_diff.days < 31:
        if pre_time == 0:
            pre_time = timestamp
        if int(round((timestamp - pre_time) / 60)) >= GROUP_TIME_UNIT:
            grouped_time_series[timestamp] = tweet_timeseries.get(timestamp)
            pre_time = timestamp
        else:
            if pre_time not in grouped_time_series:
                grouped_time_series[pre_time] = []
            for time_val in tweet_timeseries.get(timestamp):
                grouped_time_series[pre_time].append(time_val)
```

Figure 7 -Timestamp creation for 10 days

To get time of the tweet we have focused on tweet created time. From that we can get timestamp of the tweet. Those all crated timestamps are assign to time series Array and result Array. From those Arrays we removed retweet data and junk tweets. Also we have checked the tweet created time is more than 31 days or not. Because it will help avoid redundancy and can get more accurate results. Timestamp of the tweet and keywords will be stored in the database and from those data can analyze events.

❖ **ARIMA Model creation**

```
                squeeze=True)
# make ARIMA model
models = []
aic = []
count = 0
for p in range(0, 5):
    for d in range(0, 2):
        for q in range(0, 5):
            try:
                model = ARIMA(series, order=(p, d, q)).fit(disp=0)
                models.append(model)
                aic.append({
                    'id': count,
                    'aic': model.aic
                })
                count += 1
            except Exception as e:
                pass

forecast_data = {}
if len(aic) > 0:
    pre_aic = -9999
    pre_id = 0
    for aic_obj in aic:
        if pre_aic == -9999:
            pre_aic = aic_obj['aic']
            pre_id = aic_obj['id']
        if pre_aic > aic_obj['aic']:
            # lower AIC value found
            pre_aic = aic_obj['aic']
            pre_id = aic_obj['id']

    forecast = models[pre_id].forecast(steps=predict_day_count, exog=None, alpha=0.95)
    print(forecast[0])
    # forecasting results
    now = datetime.now()
    for i in range(1, predict_day_count + 1):
```

Figure 8 –ARIMA Model creation

In event analyze we have used Time series analysis and forecasting methods. In time series analysis we have used ARIMA model to analyze events. ARIMA means Autoregressive Integrated Moving Average (ARIMA) Model.

ARIMA is a general statistical model which is widely used in the field of time series analysis. General ARIMA model is denoted as the ARIMA(p,d,q) where p,d, and q are non negative integers. In the above notation p parameter basically refers to the autoregressive part ,d parameter refers to integrated part and the last parameter q refers to the moving average part.[8]

$$W_t = \mu + \frac{\theta(B)}{\phi(B)} a_t$$

The series Wt is computed by the IDENTIFY statement and is the series processed by the ESTIMATE statement. Thus,Wt is either the response series *Y*t or a difference of Yt specified by the differencing operators in the IDENTIFY statement.For simple (nonseasonal) differencing, Wt = (1 - B)dYt . For seasonal differencing.Wt = (1 - B)d(1 - Bs)DYt, where *d* is the degree of nonseasonal differencing, *D* is the degree of seasonal differencing, and *s* is the length of the seasonal cycle.[9]

❖ **Timestamp creation for accounts**

```
def find_users(data):
    api, conn_key = utility.create_api_connection()
    res = []
    users = api.search_users(q=data['user'])
    for user in users:
        res.append(user._json)

    # close connection
    utility.close_connection(conn_key)
    return res


def analyse_user_profile(data):
    res = {
        'recent_tweets': [],
        'hash_tags': {}
    }

    api, conn_key = utility.create_api_connection()
    tweets = api.user_timeline(id=data['id'])
    for tweet in tweets:
        res['recent_tweets'].append(tweet._json)
        # get hash tags filtered
        hash_tags = re.findall(r"#(\w+)", tweet.text)
        for tag in hash_tags:
            if tag not in res['hash_tags']:
                res['hash_tags'][tag] = 0
            res['hash_tags'][tag] += 1

    return res
```

Figure 9 -Timestamp creation for accounts

In Account analyze Users should be able to identify verified account so they can select correct account. In twitter, there are verified accounts and non-verified accounts and fake accounts as well. In order to select correct account, user should be able to identify verified accounts easily. To search correct account we have to pass the keyword to API and it will retrieve the similar named consists accounts. Then we have used timestamps to get most recent tweets and most popular hashtags account consist with. Those data will create Array list and from those lists we can analyze profiles.

3.1.5 Time evaluation in tweets

In Topic comparison with time evaluation system will analyse products and identify user preferences on those products. The input of this module is product based keyword according to user choice. Analysing of these data can be done in two topics. In here we consider topic regarding to mobile phones. We take mobile phones because there are huge data related to it therefore we can analyse and give accurate output. And, it's a famous tweeter topic among users. System will generate clusters according to user entered keywords and view the popularity in clusters. Clustering is an important data mining technique employed in dataset exploration where one wishes to partition these datasets into related groups. Therefore, to categorize tweets related to the given topic we use clustering. Among the algorithms that are typically used for clustering, k-means is arguably one of the most widely used and most effective clustering method. In our research project we have followed k-means algorithm to perform the data analysis part. Other than that, we have followed supervised and unsupervised clustering in this component.

Supervised clustering is applied on classified examples with the objective of identifying clusters that have a higher probability density to a single class. Unsupervised clustering is a learning framework using a specific object functions, for example a function that minimizes the distances inside a cluster to keep the cluster tight as possible. [11] In supervised clustering we used labelled data to generate clusters and in unsupervised clustering we have used unlabelled data. In supervised clustering we have only focused on two predefined labels. Those are product price and product features.

In addition to that, time evaluation is shown in relation to clusters. We have used graphs for the visualization of these event twitter data in linear model. As a general data structure, graphs have become increasingly important in modelling event analyse structures and their interactions. [12]

Graph mining techniques and tools are then used to discover the required information for data visualization. Graph mining is the process of gathering and analysing data represented in graphs.

The input of this module are 2 keywords according to user choice. The system will analyses all tweets which includes the keywords given by user and view its popularity as a percentage. And

also, it shows the popularity of the topics given by user within days as a graph. And it show the popularity of the selected categories.

The system process is divided into three parts

1. Retrieve tweet according to selected categories
2. put the data into the timestamp and sort
3. Visualized the data using singchart library.

❖ **Retrieve tweet according to selected categories**

This study was based on the following aspects.

First user has to type topic they need to search, then they have to choose one or any category from the given 5 categories. For one topic we give 5 categories to search by that user can choose and see the popularity of categories they choose. Then they can see how their searched topic is spread during a time period. This contains another part which include two topic evaluation with time.In that user has to type two topic to get the time evaluation. By that user can compare two topics with time evolution. Tweeter has huge data set it take time to process therefore we used multi-threading to reduce loading time. System made threads according to the no of categories select by the user and retrieve all data. Like this we take all positive data related to topics.

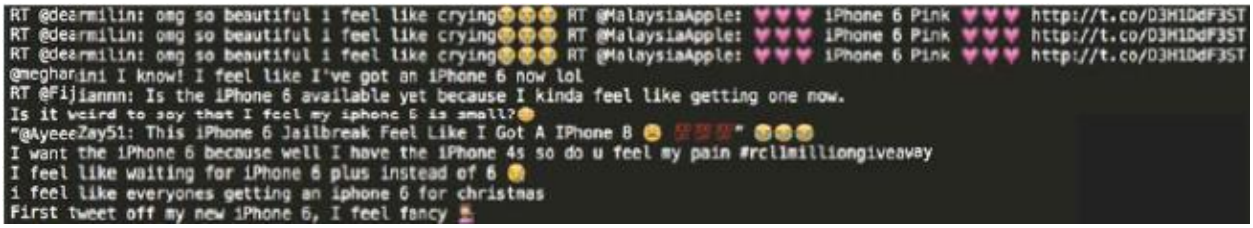

Figure 10 – Example of Data retrieve

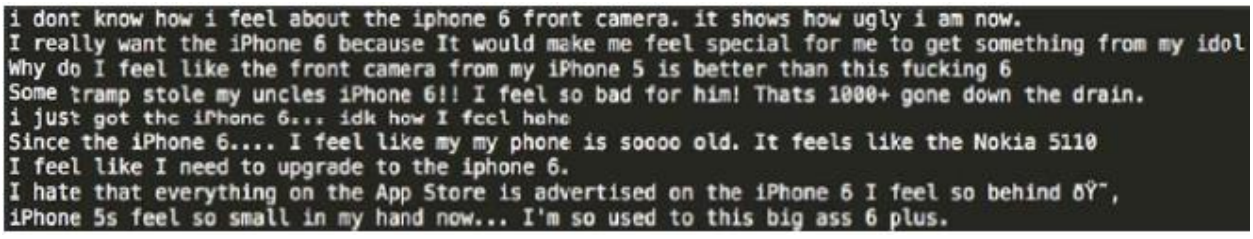

Figure 11 – Example2 of Data retrieve

❖ **Clustering**

Clustering can be considered the most important learning problem of machine learning. It is the task of making a set of objects in such a way that the objects in the same group are called cluster [5]. The clusters are more similar to each other than to those in other clusters.

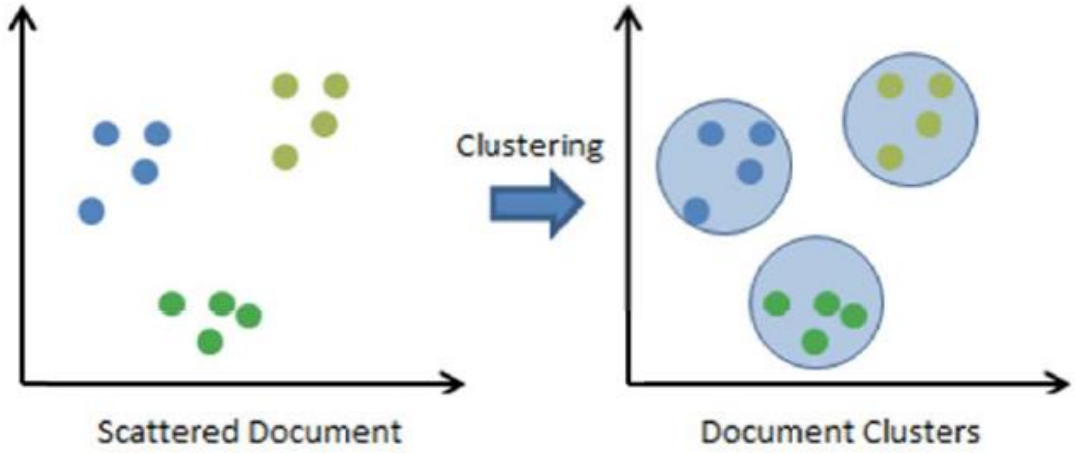

Figure 12 clustering [10]

In Figure 12. we can easily separate data to 3 clusters. Distance is an important point to know because each object should belong to a cluster. Two or more objects belong to the same cluster if they are close, according to the distance. Unfortunately, the negative side to using this method is that to much time is taken for data collection. Furthermore, this method using clustering too much, which adds more to the amount of time used. Most of the time, clustering time consumption is focused on finding a good center point. Therefore, less clustering is usually a betterchoice when trying to save time.

❖ **K-means Algorithm**

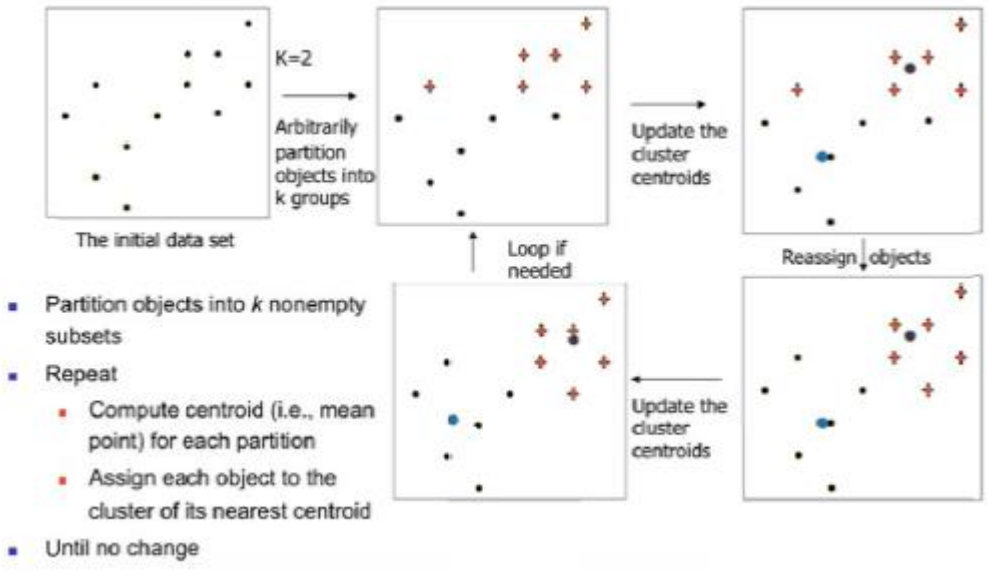

Figure 13 K-means Algorithm [9]

The k-mean clustering algorithm is known to be efficient in clustering large data set, and is one of the simplest and the best known matching learning algorithm that slow the clustering problem shows the four steps of the k-means algorithm. the first steep divides items into k nonempty subgroups. In the second, the compute seed points to the centroids of the clusters of the current divisions. The centroid is at the center, which means the middle point of the cluster group. The third step is when each object is assigned to the cluster with the nearest seed point. The fourth and last step goes back to Step 2 and stops when the assignment does not change [11].

K-means process has some weaknesses. First, there is a problem with comparing the quality of the clusters. Second, because there is a fixed number of cluster, it can be hard to find out what K should be. Third, k-means only work well with circular cluster shape. Fourth, when the original partitions are not the same, this may cause final clusters that are also different. It is useful to run the program again by like and unlike K values, so that we use the center point as a selected value, then we can make cluster's without any weakness. compare the outcomes gained [9].

3.2 Implementation and Testing

3.2.1 Implementation

In this research our approach is to give meaningful real-time results and predictions analysing twitter data. To accomplish this approach we divided our system into four major parts. First of all we had to make a connection with the twitter website. Secret keys were gathered from twitter developer site. Streaming API had been used for the API, using this API author can request more tweets. Desktop application was chosen because in this approach author had to display lots of charts graphs and small data.

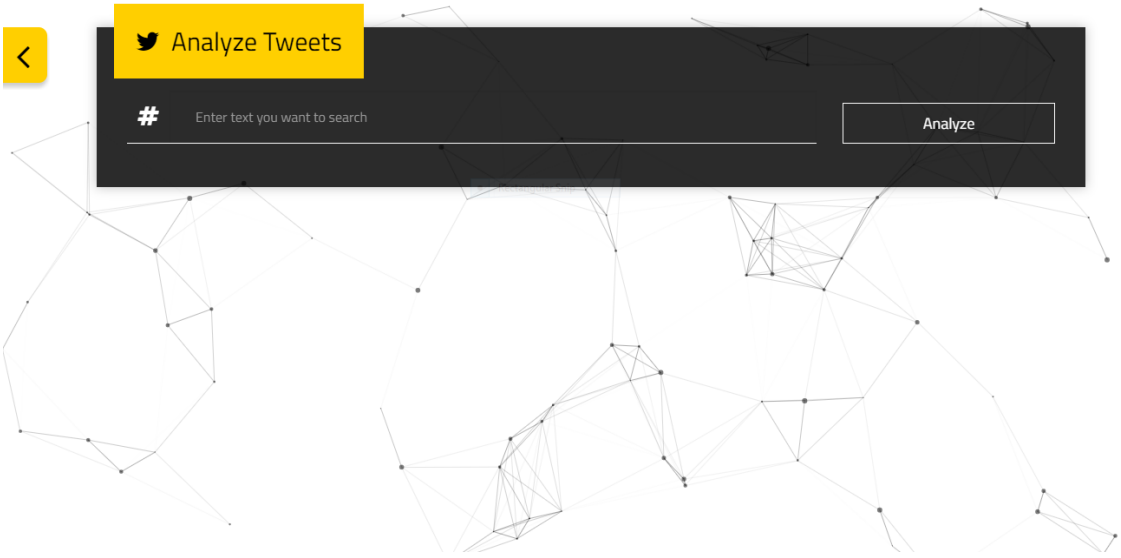

Figure 15 -Interface for sentential analyse and false tweets analyse

Author's first approach is to get an input from a user and analyse that keyword using twitter data and analyse false tweets. When user gives an input, the system makes a connection with twitter. Secret keys are required to accomplish this. The system requests tweets which contain the user given input, language of the tweets and the count of the tweets. For this approach one secret key is enough because system requests only one time. As soon as the response comes analysing parts

will execute. For this approach sample data sets and NLTK which contains naïve bays classifier are used. After execution of the algorithm each tweet will be stored in an array and ladled which one is positive, negative or neutral. Those arrays will be sent as respond to the frontend. Frontend will calculate the percentage and display it using pie charts. False tweet identification is done using the tweets which came for the user given input. Those tweets will be checked with system's predefined conditions and rated as a fake tweet or a real tweet. The respond will be sent to the frontend. Frontend will display the tweets according to user's requirement. According to the result user can find out the impact of the given input and how many fake tweets is there.

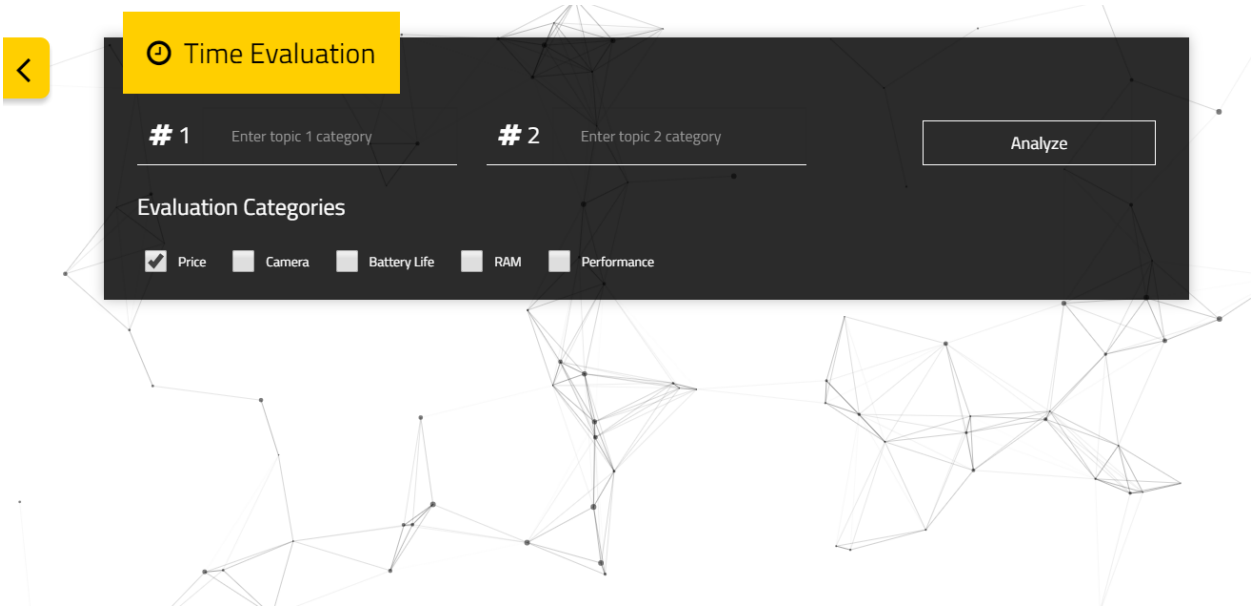

Figure 16 - interface for product analyse according to time

Author's second approach is to find the popularity of the given two products with the given features according to time. User has to input two similar type products. The system will make a connection with twitter. For this approach author had used multithreading. Thread count will depend on user's choice. If user asks for three features six threads will be created. For this approach clustering algorithms had been used to group the product tweets with the features. The tweets will be analysed by the system and respond will be send to frontend. Frontend will display the result using charts with respect to a particular time frame and to the most popular product. Using this result user can identify which product is best in features wise and ultimately which product is the most popular one.

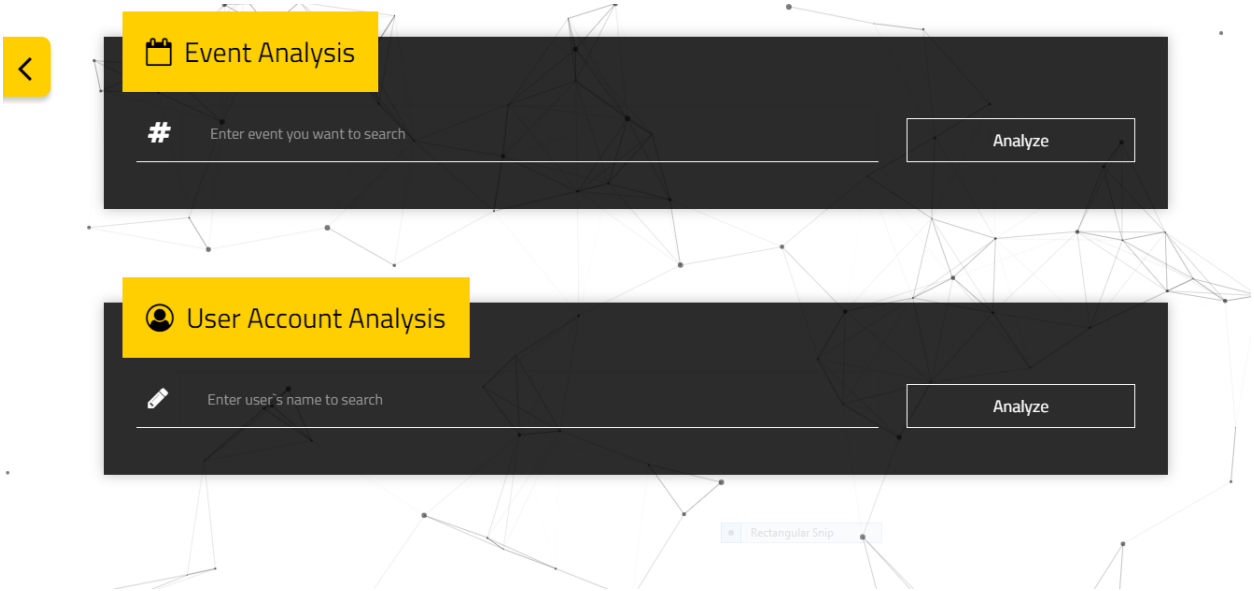

Figure 17- interface for event analize and user accout analyse

Author's third approach is to analyse events and give predictions regarding those events and to analyse a user account. To accomplish this approach author had to use ten threads. Because previous ten days tweets will be required for this approach. After gathering the tweets system analyses the tweets. ARIMA model had been used to give predictions. System finds out how many users were active at certain time periods and predicts what will happen in the future to those events.in the account analyse part system will show details of a certain user account. Final result

will be displayed using charts and predictions. According to the result user can get an idea of events which they wish to focus on.

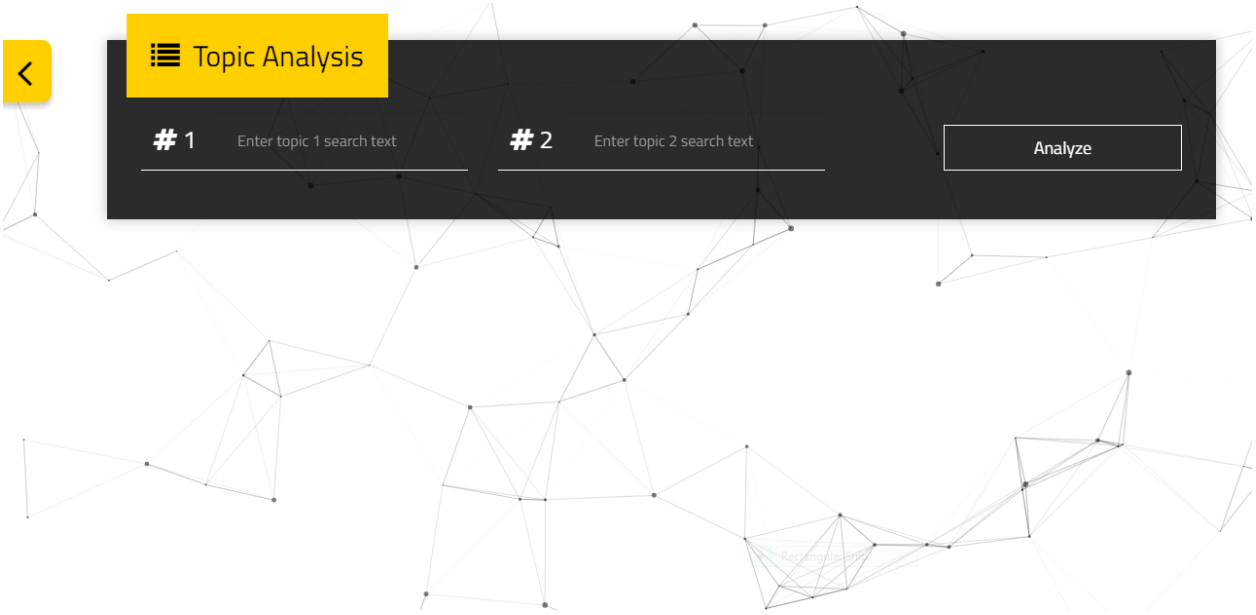

Figure 18- interface for retweet comments and tweets analyse

Author's final approach is to analyse retweets, comments and tweets on a particular product. Two threads will be used for this approach. After gathering the tweets system analyses it using the algoritham.in here system finds the number of retweets comments and tweets on the given two topics. Propagation topology for the retweet analysis and exponential regression function for popularity prediction are used as techniques. Using final result user can identify how many users are following their product and most importantly user can compare with their rival product with their own product.

Technologies used

- Mongo DB - MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas.
- Python 3.5 - Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991.
- AngularJS - AngularJS is a JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications.
- GIT - Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people.

Software

- Pycharm community edition - PyCharm is an Integrated Development Environment used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains
- Robomongo - **Robomongo** with a single version of MongoDB shell and it supports your version of MongoDB.
- SourcetTree - A free Git client for Windows and Mac. **Sourcetree** simplifies how you interact with your Git repositories so you can focus on coding. Visualize and manage your repositories through **Sourcetree's** simple Git GUI.

Hardware

- ADSL Router
- PC windows 10 operating system, 8gb RAM, core i7

External Libraries.

- NLTK - NLTK is a leading platform for building Python programs to work with human language data which provide many facilities. Text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

- Bower - Bower is a package manager for the web. It offers a generic, unopinionated solution to the problem of front-end package management, while exposing the package dependency model via an API that can be consumed by a more opinionated build stack.
- Emoji- The entire set of Emoji codes as defined by the unicode consortium is supported in addition to a bunch of aliases. By default only the official list is enabled but doing emoji.emojize(use_aliases=True) enables both the full list and aliases.
- Corpus - NLTK includes a diverse set of corpora which can be read using the nltk.**corpus** package. Each **corpus**is accessed by means of a "**corpus** reader" object from nltk.
- Statemodel - Statsmodels is a Python package that provides a complement to scipy for statistical computations including descriptive statistics and estimation and inference for statistical models.
- Tweepy - Very active developer community creates many libraries which extend the language and make it easier to use various services. One of those libraries is **tweepy**. **Tweepy** is open-sourced, hosted on GitHub and enables **Python** to communicate with Twitter platform and use its API.

3.3 Testing

- **Unit Testing**

  Unit testing will be carried out at the initial stages of the software development process. Developer himself will test the individual units to identify bugs and will correct them at that point. We tested our system with following scenarios.

  - ❖ Test system without giving any values
  - ❖ Test System giving one topic
  - ❖ Test System by giving same topic for both
  - ❖ Test System by giving topic not relevant to mobile phones
  - ❖ Test System by giving Two Topics (Relevant To Mobile)

- **Integration Testing**

  Once, more than one modules are completed, those modules will be integrated and tested to find bugs. This way, when a module completes, it will be integrated with the other modules and will test for finding bugs. We integrate parts one function by one and check one by one while integrating. After integrating all functions, we recheck our system with all functions to check all are functioning properly after intergradation.

- **Socializing concept testing**

  The system is fully tested with the developers of the team and the issues were identified and noted. Also, some outsiders will give the opportunity to use our system and their feedback will be noted as well. To test it, we show our system to a friend who doesn't have IT knowledge. He confused what to choose as best because we only show graphical view of data in a graph. As a solution we shows most popular topic in a chart.

- **System Testing**

  Testing the whole system against the specifications is called as system testing. Once the unit and the integration testing is done this system testing is started to make sure the

compatibility and the integration of the units are done properly. Waterfall model in testing can only be done by dividing coded program into various manageable units. After that in integration phase those units are integrated into complete system. After that we have to test to verify if every modules are coordinate with each other and the system as a whole behaves as per the specification.

3.4 API unit testing

Table 1: Test case 01- Check all the text boxes, radio buttons, buttons etc in the interface.

Test Case ID	01
Description	Check all the text boxes, radio buttons, buttons etc
Pre-Condition	Interface
Steps	1. Click on Radio buttons, buttons, and drop down list 2. type on text box
Inputs	keyword
Expected output	-UI should be perfect -Text boxes and button should be aligned
Actual output	Perfect UI with proper alignment

Table 2: Test case 02- **Check page load on slow connections**

Test Case ID	02
Description	Check page load on slow connections
Pre-Condition	Good Network Connection
Steps	1. Connect to Router 2. place device in area where good signals comes. 3. remove connectivity of other connection types apart from existing connection. 4. Refresh
Inputs	WIFI username and password
Expected output	Page load within 30 seconds
Actual output	Page load after 1 minutes

Table 3: Test case 03- **Download twitter data through the API and store in database**

Test Case ID	03
Description	Download twitter data through the API and store in database
Pre-Condition	API server should up and running
Steps	Type topic in the UI Wait until data downloaded into the database
Inputs	Twitter Topic
Expected output	Twitter data should store in database properly
Actual output	Twitter data properly inserted in to the database

Table 4: Test case 04- **Analyses topic related keyword using twitter data**

Test Case ID	04
Description	Analyses topic related keyword using twitter data and displays those data in percentage wise
Pre-Condition	API server should up and running
Steps	Type topic related keyword Click analyse button
Inputs	Topic related keyword
Expected output	Tweets will be displayed according to categories and as a percentage wise pie chart
Actual output	Tweets will be displayed according to categories and display a percentage wise pie chart

Table 5: Test case 05- **Test case for Displaying positive tweets only**

Test Case ID	05
Description	After clicking the selection name positive on the pie chart the positive tweets will be displayed
Pre-Condition	API server should up and running
Steps	Type topic related keyword Click on the analyse button Click on the positive section on the pie chart
Inputs	Topic related keyword
Expected output	Tweets will be displayed under the positive category
Actual output	Tweets will be displayed under the positive category

Table 6: Test case 06- **Test case for Displaying negative tweets only**

Test Case ID	06
Description	After clicking the selection name negative on the pie chart the negative tweets will be displayed
Pre-Condition	API server should up and running
Steps	Type topic related keyword Click on the analyse button Click on the negative section on the pie chart
Inputs	Topic related keyword
Expected output	Tweets will be displayed under the negative category
Actual output	Tweets will be displayed under the negative category

Table 7: Test case 07- **Test case for Displaying neutral tweets only**

Test Case ID	07
Description	After clicking the selection name neutral on the pie chart the neutral tweets will be displayed
Pre-Condition	API server should up and running
Steps	Type topic related keyword Click on the analyse button Click on the neutral section on the pie chart
Inputs	Topic related keyword
Expected output	Tweets will be displayed under the neutral category
Actual output	Tweets will be displayed under the neutral category

***Table 8: Test case 08*-Test case for displaying real and false user accounts**

Test Case ID	05
Description	After clicking on the analyse false tweet button the tweets will be categorised to two separate sections called real users and fake users
Pre-Condition	API server should up and running
Steps	Type topic related keyword Click on the analyse button Click on the analyse false tweet button
Inputs	Topic related keyword
Expected output	Twitter user accounts will be categorised and the count of the user account will be displayed under the relevant category
Actual output	Twitter user accounts will be categorised and the count of the user account will be displayed under the relevant category

Table 9: Test case 09- Check UI with empty value

Test case ID	09
Description	Check UI with empty value
Testing steps	1. Log in to the system 2. Select the topic analyse tab 3. Tick the categories box 4. Click analyzed
Inputs	Null
Expected output	UI should not be load
Actual output	Not load the UI

Table 10:***Test case*** 10- Check with two values (not relevant)

Test case ID	10
Description	Check with two values (not relevant)
Testing steps	1. Log in to the system 2. Select the time evaluation tab 3. Tick the categories box 4. Enter topic one and topic two 5. Click analyzed
Inputs	Laptop , mobile
Expected output	UI should not be load Should Give an error message
Actual output	Not load the UI View error message

Table 11: Test case 11- Check with one value

Test case ID	11
Description	Check with one value
Testing steps	1. Log in to the system 2. Select the time topic analyse tab 3. Tick the categories box 4. Enter the one topic 5. Click analyzed
Inputs	Dell
Expected output	UI should not be load Should Give an error message
Actual output	Not load the UI View error message

Table 12: Test case 12- Check with two values

Test case ID	12
Description	Check with two values
Testing steps	1. Log in to the system 2. Select the Topic analyse tab 3. Tick the categories box 4. Enter topic one and topic two 5. Click analyzed
Inputs	Iphone6, Samsung s6
Expected output	UI should be load Should view the time evaluation Should view the categories Should view the consolation
Actual output	UI load correctly Should view the time evaluation Should view the categories Should view the consolation

Table 13: Test case 13- Analyse events from tweets

Test Case ID	13
Description	Analyse events from tweets and give predictions regarding those events
Pre-Condition	API server should up and running
Steps	Type Event topic related keyword Click on the analyse button
Inputs	Twitter Event topic keyword
Expected output	UI should be load Should view the events evaluation Should view predictions of the event Should view most popular event categories
Actual output	UI load correctly Should view the events evaluation Should view predictions of the event Should view most popular event categories

Table 14: Test case 14- Analyse Twitter accounts

Test Case ID	14
Description	Analyse Twitter accounts
Pre-Condition	API server should up and running
Steps	Type accounts related keyword Select correct account Click on the analyse button
Inputs	Twitter accounts keyword
Expected output	UI should be load Should view the accounts name consist with keyword Should view accounts analysis data Should view most popular accounts categories
Actual output	UI load correctly Should view the accounts name consist with keyword Should view accounts analysis data Should view most popular accounts categories

3.4 Research Findings

The main purpose of Hash-Tag-Invenstor is to analyse twitter data to generate useful information including some predictions. Twitter has an enormous collection of data. If these data is converted into some useful information, accurate decisions can be made using this data. Twitter data extract can be done by a twitter API call, for that author used twitter streaming API which can get large amount of tweets at a time rather than using other API's for data saving part we used mongo db. One advantage of this data, over previously used data-sets, is that the tweets are collected in a streaming fashion and therefore represent a true sample of actual tweets in terms of language use and content. Our new data set is available to other researchers. Tweets sometimes express opinions about different topics. Information available from social networks is beneficial for analysis of user opinion, for example measuring the feedback on a recently released product, looking at the response to policy change or the enjoyment of an ongoing event. Manually identify this data is difficult and potentially expensive.

Build of models for classifying "tweets" into positive, negative and neutral sentiment, identify time evolution in products from tweets, analyse events through tweets, analyse twitter accounts, analyse likes comments retweets on tweets and analyse geo relations with these tweets. Supervised learning has been used and NLTK and the naïve Bayes classifier has been used as techniques in classifying tweet into positive and negative. The output will be display percentage wise, negative positive and neutral percentages of the given keyword. Twitter data is analysed according to the given keyword. False tweets identification is done by analysing user profile. If the user profile criteria do not match with our assumptions this profile is marked as a fake profile. Output of Getting the tweeter popularity according to the time is displayed by charts. Similar keywords will be grouped. Clustering algorithms has been used for grouping. Finding some latest ongoing events and the number of users who were active at certain time periods, ARIMA model has been used as the technique. The popularity of retweets, likes and comment collection of two topics is displayed in graph for 24 hours and prediction is shows in another chart for next 24 hours. Propagation topology is used as the technique for retweet analysis and exponential regression function is used for popularity prediction.

Example

In the current context, author found a way to predict whether the given text is a positive, negative or a neutral and the given user account is fake or real. Author had to find a sample data set and using that data set author created the trained data model. After completion of this experiment author had found the perception of the topic name cricket, 53% of the tweet population rate cricket as a positive thing and 34% saw it as neutral thing and 14% saw it as a negative thing. After gathering those details author could say cricket is a positive thing to the world. Because most of them saw cricket as positive. As another example author found the perception of the topic North Korea, 14% of the tweet population rate North Korea as a positive thing and 17% saw it as neutral thing and 69% saw it as a negative thing. According to these information we can say the world see North Korea as a negative subject.

4. RESULTS AND DISCUSSION

4.1.Results

4.1.1Tweet Analysis

Integrated the whole system after completing each and every individual component and test the system.

The figure 9 shows the result for the input name cricket. Altogether there were 300 tweets about the topic name cricket and 34% from the twitter population consider cricket as positive, 14% as negative and 53% as neutral

The figure 10 shows the fake and real users. There were 299 real users and 1 fake user profile.

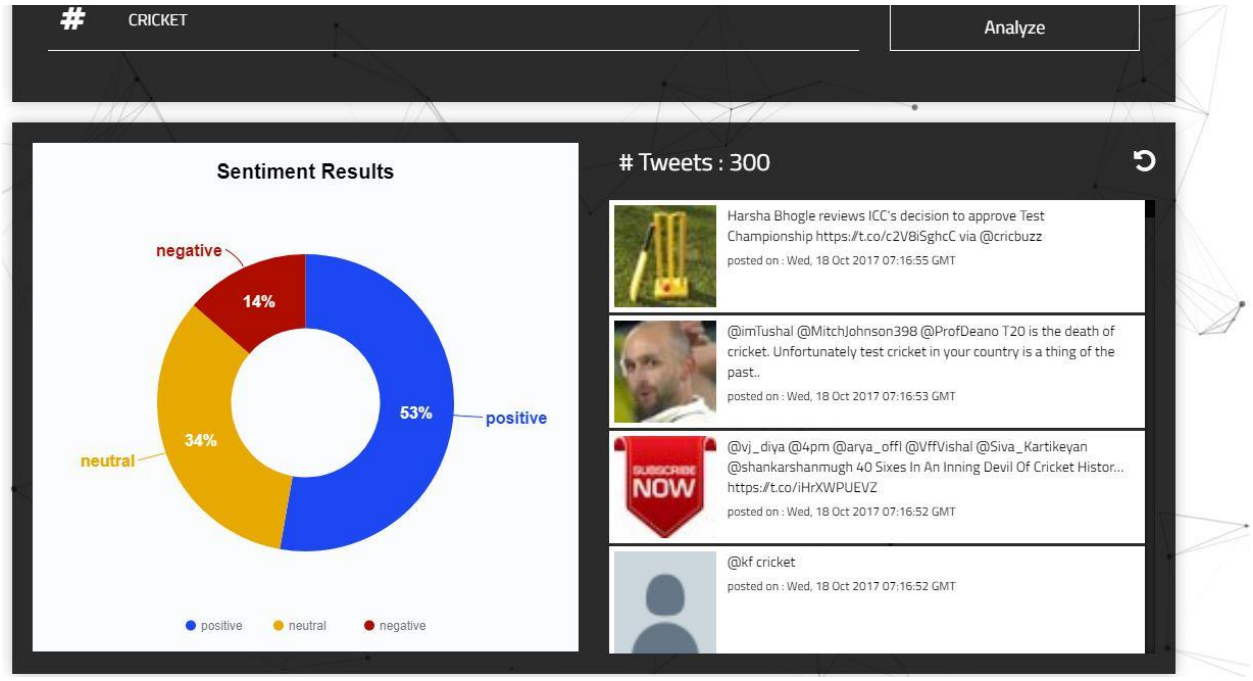

Figure 19-Generated output of the perception analysis

In this interface the user have searched cricket as the topic. The output shows percentage wise analysis and the number of count of the topic cricket.

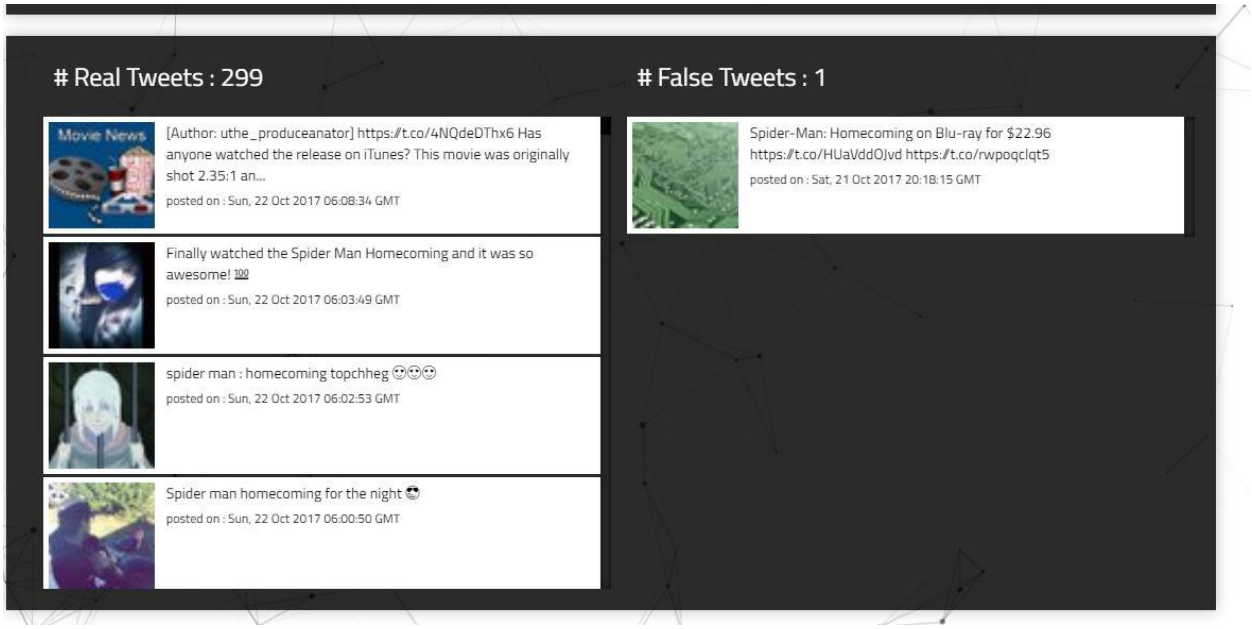

Figure 20 - Generated output of the false tweet analysis

This interface shows the real tweet count and the fake tweet count and lists the tweets under those categories.

4.1.2 Topic Analysis

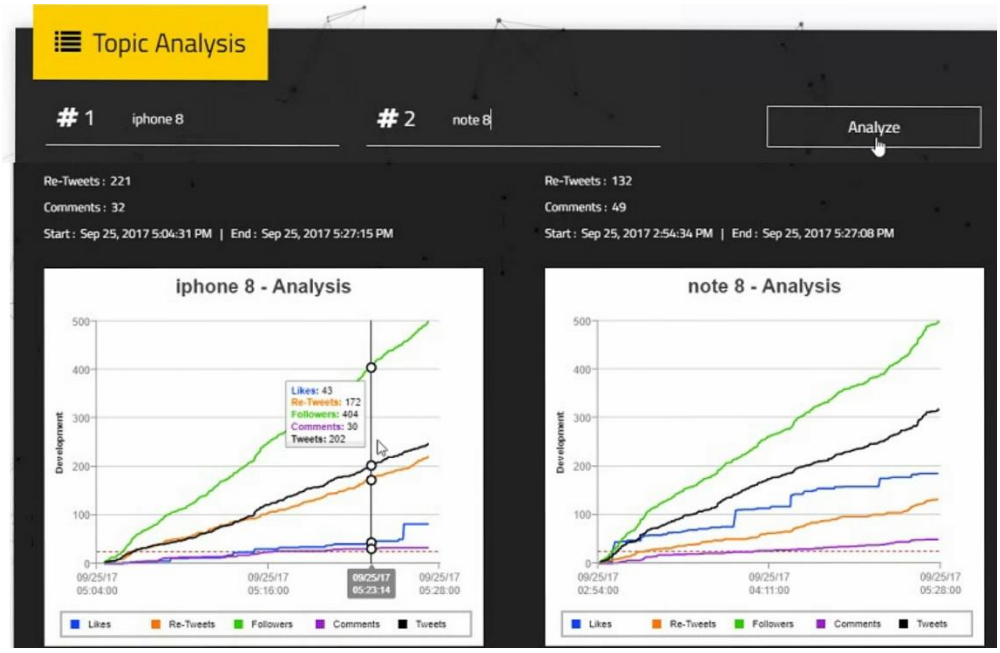

Figure 21-Interface (Analyze within 24 hrs.)

In this interface user can type two key words and click on analyze button to see the comparison. System will show popularity of retweet, likes, comments count with respective to given two topics. And also, it shows graphical evolution of those within an hour in a Chart. In here x axis is represent time and y axis represent the development of likes, retweet, followers and comments. Different colors in lines are represent the growth of like, retweet followers and comments.

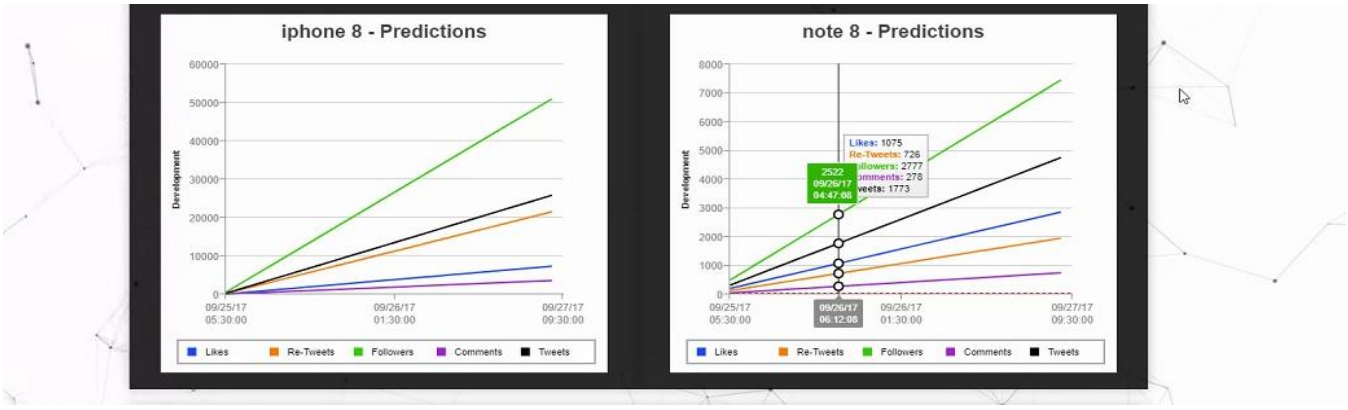

Figure 22-Interface (Prediction chart for next 24 hrs.)

In this interface prediction for next 24 hours is show in a graph. In here x axis is represent time and y axis represent the development of likes, retweet, followers and comments. Different colors in lines are represent the predicted growth of like, retweet followers and comments.

Figure 23-User Locations

In here a map is load with respective to the locations of tweeter users when they are liking, commenting, following and retweeting a tweet. Different colors represent likes, comments, followers and retweets.

4.1.3 Event Analysis

Figure 24- interface for output of event analize

This interface will load after user clicked on the analyse button. This will show tweets time line in the chart and it is consist with the event details in past 10 days. User can see which are the most popular days and time slots of the events and which are the lowest popular days and timeslots in the event.

In the above Figure14 user have searched American Idol as his event and those are the event details displayed after analysing.

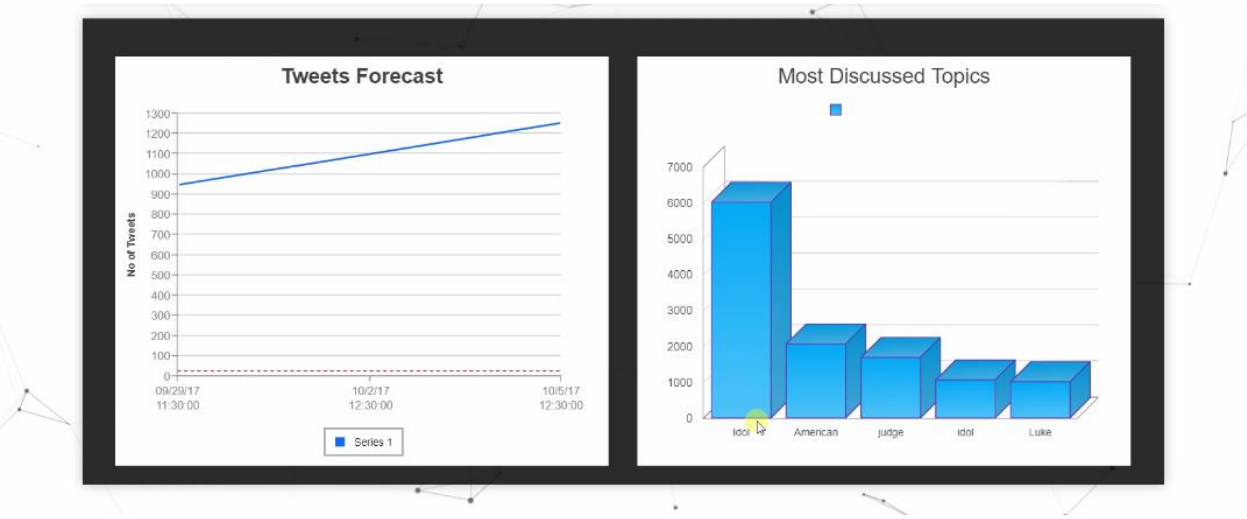

Figure 25- interface for output of event analize and predictions

This interface will load after user clicked on the analyse button. In here user can see the prediction for the event. System will show forecasting for 7 upcoming days and user can see which are the most popular days and time slots of the events and which are the lowest popular days and timeslots in the event.

4.1.5 Account Analysis

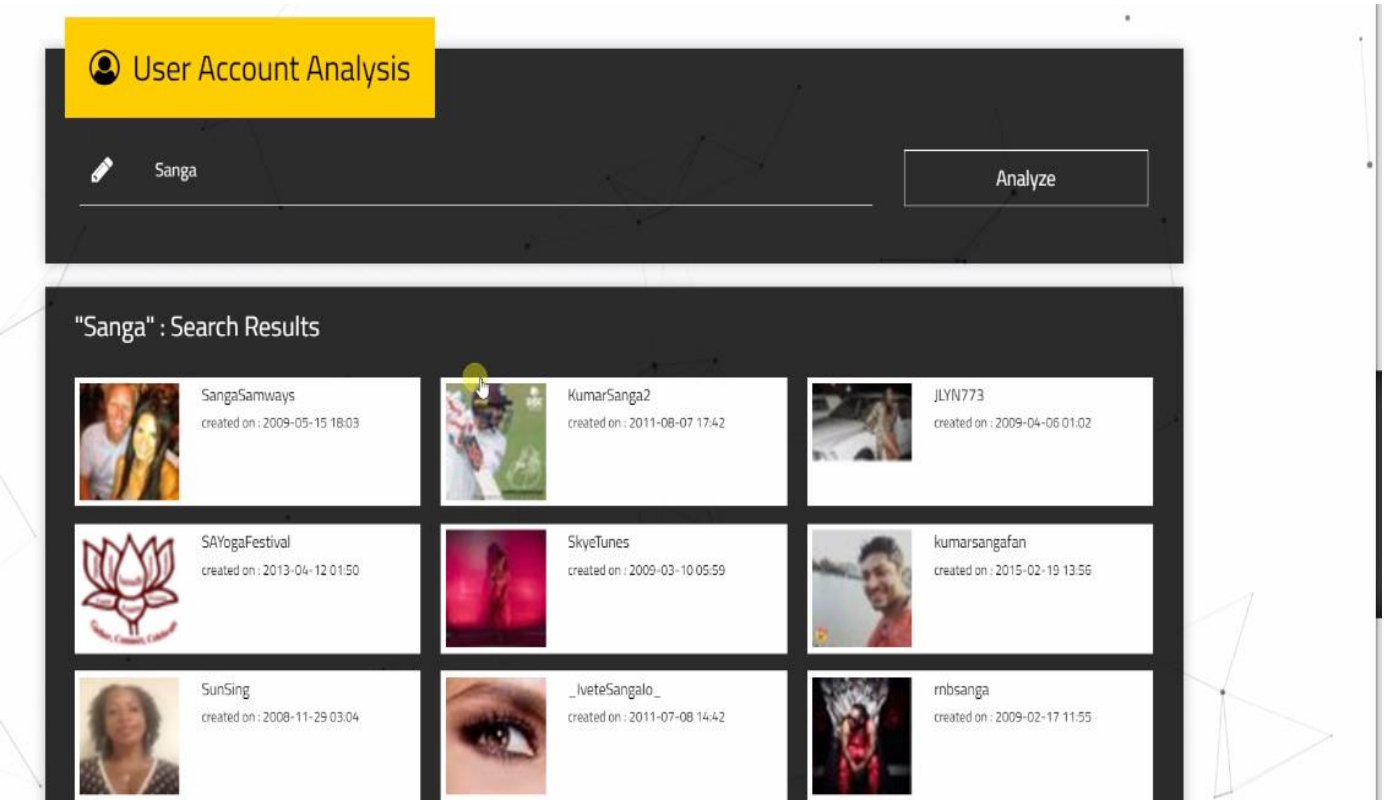

Figure 26-interface for output of account analize

In our system Users should be able to identify verified account so they can select correct account. In twitter, there are verified accounts and non-verified accounts and fake accounts as well. In order to select correct account, user should be able to identify verified accounts easily.

In this interface user can search account name and then click analyse button then it will display what are the similar accounts to search account name. Then user can select verified account and click on the account.

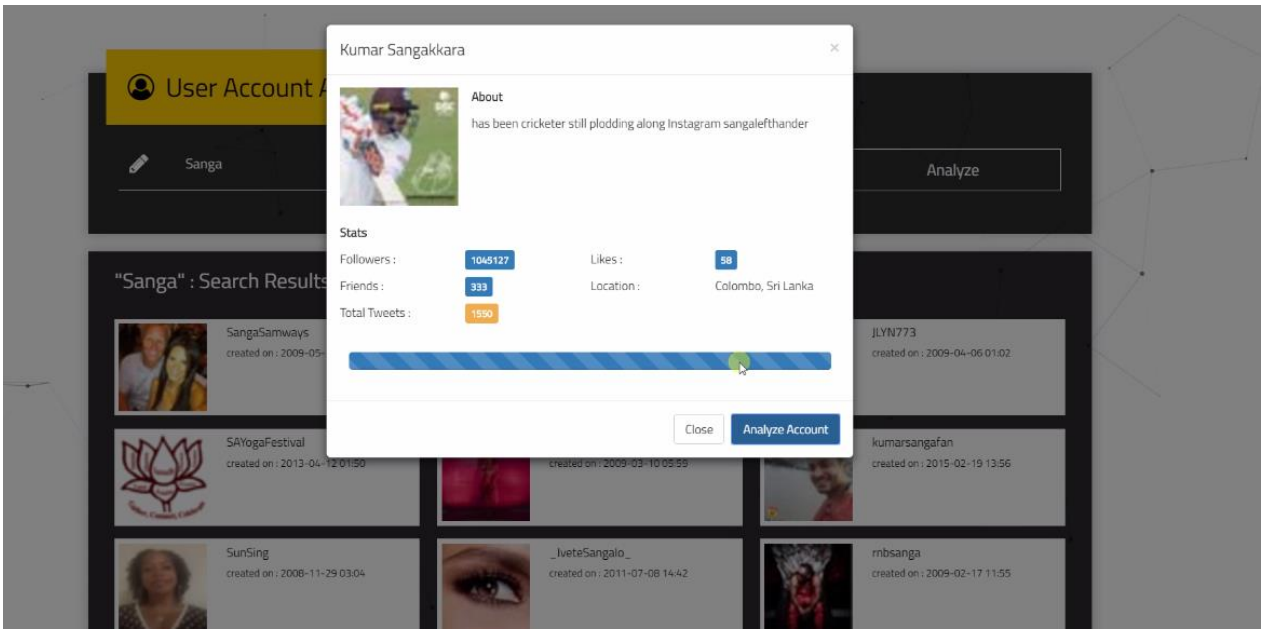

Figure 27-interface for output of account analize

This interface will show the basic details of the account user have searched. Such as Number of followers, likes, friends, etc. Then user can analyze further more account by click Analyze Account button.

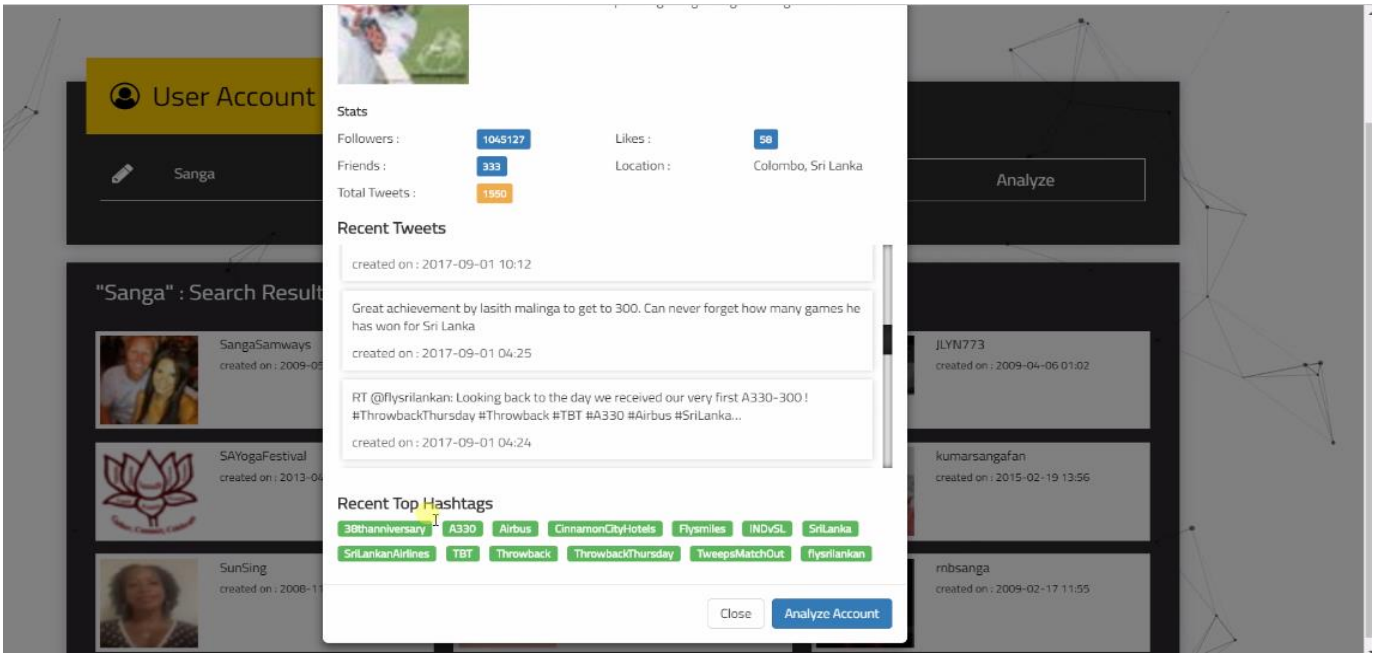

Figure 28-interface for output of account analize

This interface will display the further anaysis of the account.Those are which are the recent top hah tags,recent tweets etc.

4.4 Time analysis in tweets

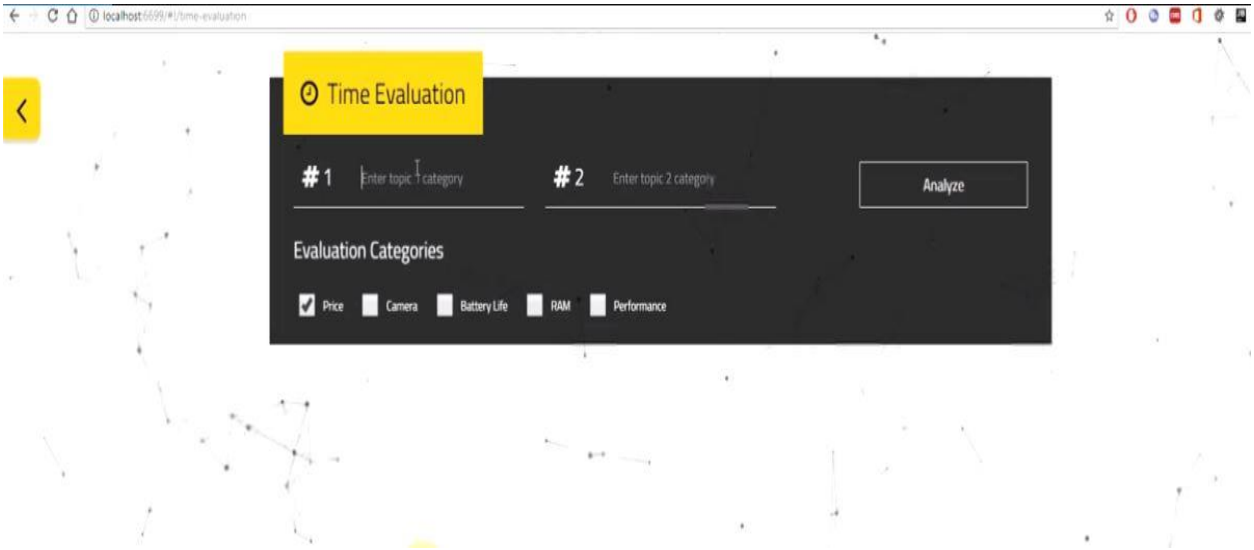

Figure 29 - Topic entering interface

In this interface user have to type two products (mobile) and select relevant category

Figure 30 - Time Evaluation interface

This interface shows the evaluated information of relevant topic user type and choose in a chart. In this chart x axis is Time and y axis is Data count. Different colours show the categories and when user moves the mouse it shows data count of relevant categories. When user click on one colour system ignore other colours and show only selected one.

Figure 31 - Category Overview Interface

In this interface system shows the popularity of selected categories with respective to given two topics

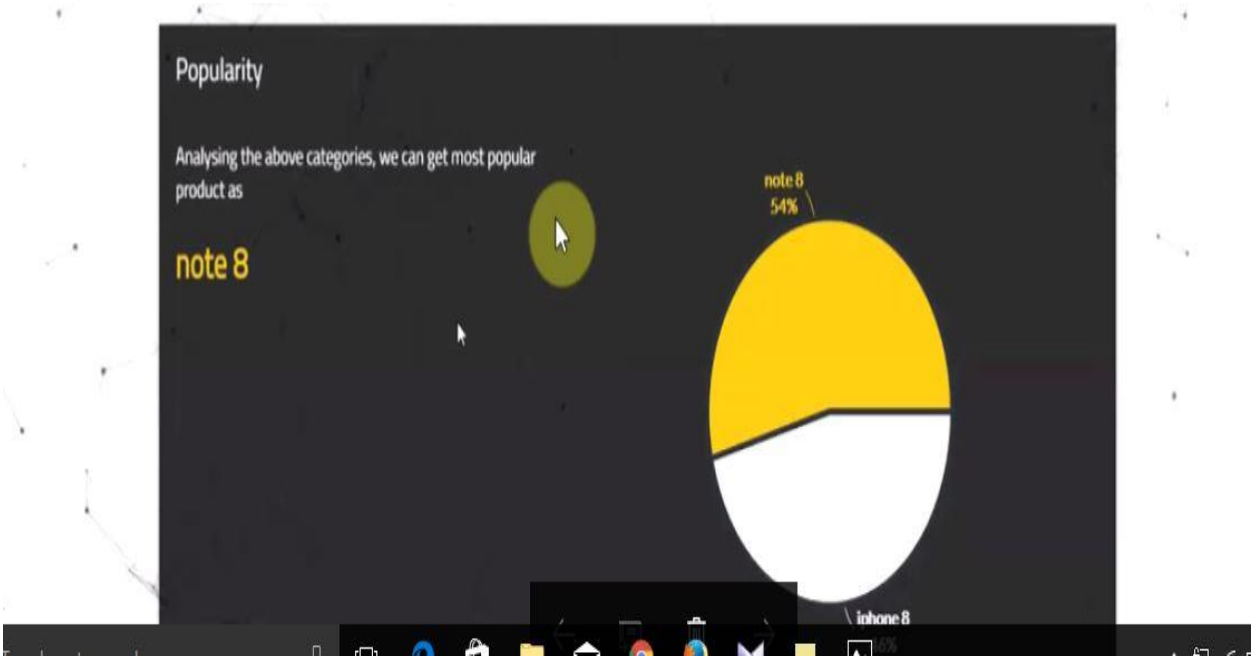

Figure 32 - Summary Interface

This interface shows the final conclusion of popularity in a pie chart by counting all values in categories.

5. COMMERCIALIZATION OF THE PRODUCT

Twitter usage statistics shows that there are on average 350 million tweets per day, 140 million active users and 50 000 000 processed tweets. This is a massive data resource for big data analytics.

If these data is converted into some useful information, accurate decisions can be made using this data for organizations and normal people as well. That is our objective, which can be very help to users. This is more useful as the details of a particular products they can get clear idea from our tool. Further, the system will identify and avoid false tweets generated by software tools and this will be good advantage for users. Analyse particular event through our application and can identify which are the time periods users are mainly active in those events. Analyse tweets based on given event period to forecast and guide user's behaviours. Real time analysis on the tweet to predict and forecast current affairs, based can get locations and categories. According to this tweet analyse broadcasters can get a clear idea about people's perception at the time period in the event also they can get sentimental analysis about those tweets. There is a huge amount of data collected gradually with the time. These data is difficult to collect at once .we stored this data in the DB we have created in mongo .the redundant and unnecessary data will be eliminated here the data can be processed efficiently in that way. This will be huge benefit for our customers. We have created a API for users who are using our application. Using the API provided by us any user can obtain information they needed to analyse customer's reviews about their products. This will be huge benefit for our customers.

6.CONCLUSION

This paper has discussed our experiments on twitter perception analysis show that Tweet keywords features may be useful for perception analysis in the microblogging domain. More research is needed to determine whether the Tweet keywords features are just of good quality for the perception analysis in this domain. This tool is more realistic and useful than the other available software such as financial market prediction tools. The system itself is a unique approach towards research conducted in order to enhance the experience in Twitter analyses. Using Mongo DB to collect training data did prove useful, as did using data collected based on positive and negative emotions. However, which method produces the better Training data and whether the two sources of training data are complementary may depend on the type of features used.

Due to too many API requests the response time of this system is too slow. so that in future we will develop a new way to do that. Since we have used only English, we are trying to expand our system then it will support other languages as well. This system can only analyse some predefined set of events. in future we hope to give user authority to give their own events to analyse

Many Researchers have also begun to investigate various ways of automatically collecting training data. Several researchers rely on emoticons for defining their training data [22]. It exploit existing Twitter sentiment sites for collecting training data. They also use hashtags for creating training data, but they limit their experiments to sentiment/non-sentiment classification [23]. But in ours rather than trying negative positives we are trying to get emoji classify neutral tweet removing, key word popularity and event handling and many more improvement.

Tweeter is already gives a Tweeter analyser application to users who have tweeter account can logon to their tweeter account through it. But they can only check positive negative tweets of their own account, rather than analysing all tweet accounts, as we do.

The authors have proposed a solution to use these huge set of data in an appropriate way to get a commercial value like tokenising positive negative tweets, event analysing, popularity through time evolution and popularity through comparison. with these function authors come up with a

valuable output which has a great commercial value.This system can be further developed and used in other social media but to that it need permissions to accesses those API.

This approach has many commercial benefits as well. This system This system can be further developed to any other social media but at the moment authors use tweeter because it has no restrictions. This system is especially valuable for company owners as they can see their product popularity behaviours, event popularity, positive comments how customers like their product and how they can compare their product with rival companies. And also with customer side they can compare brands and product and choose the best from the final outcome of comparisons.

The final goal of this research was to propose a way to use enormous collection of data in tweeter. If these data are converted into some useful information, accurate decisions can be made using this data. All the functionalities mentioned above were combined in order to use these huge set of data for user helpful manner as mention in commercial Value.

7. REFERENCES

[1]. Software Requirements Specification [Online]. Available: http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf [Accessed 28/04/2017]

[2]. Software Requirements Specification [Online]. Available: https://en.wikipedia.org/wiki/Software_requirements_specification [Accessed 28/04/2017]

[3]. Yan Huang. Support Vector Machines for Text Categorization Based on Latent Semantic Indexing. Johns Hopkins University.

[4]. Durgesh Sivastava K. and Lekha Bhambhu. Data Classification using support vector machine. 2005-2009, JATIT.

[5]. Software Requirements Quality Specification. Available: https://en.wikipedia.org/wiki/List_of_system_quality_attributes. [Accessed 28/04/2017]

[6]. Concepts and Methods of Sentiment Analysis on Big Data [Online]. Available:

https://www.ijirset.com/upload/2016/september/102_Concepts.pdf [Accessed 12/04/2017]

[7].Big-SoSA:Social Sentiment Analysis and Data Visualization on Big Data [Online]. Available:

http://www.ijarcce.com/upload/2015/april-15/IJARCCE%2069.pdf [Accessed 14/04/2017]

[8]. W. K. Chen, Linear Networks and Systems. Belmont, CA: Wadsworth Press, 2003

[9]. K. E. Elliott and C. M. Greene, “A local adaptive protocol,” Argonne National Laboratory, Argonne, France, Tech. Report. 916-1010-BB, 7 Apr. 2007

[10]. Big Data Analytics and Knowledge Discovery [Online]. Available:

http://www.dexa.org/previous/dexa2016/sites/default/files/calls/DaWaK%2716%20CfP.pdf [Accessed 10/04/2017]

[11] Sentiment Analysis of Twitter Data - Apoorv Agarwal Boyi Xie Ilia Vovsha Owen Rambow Rebecca Passonneau, Department of Computer Science,Columbia University,New York, NY 10027 USA

[12] Sentiment Analysis of User-Generated Twitter Updates using Various Classification Techniques[Online]. Available

https://nlp.stanford.edu/courses/cs224n/2009/fp/19.pdf [Accessed 11/04/2017]

[13] Sentiment Analysis: An Overview [Online]. Available

https://www.researchgate.net/publication/264840229_Sentiment_Analysis_An_Overview [Accessed 10/02/2017]

[14] Improving Blog Polarity Classification via Topic Analysis and Adaptive Methods[Online]. Available

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.308.5987&rep=rep1&type=pdf [Accessed 11/07/2017]

[15] More than Bag-of-Words:Sentence-based Document Representation for Sentiment Analysis[Online]. Available

http://www.aclweb.org/anthology/R13-1072 [Accessed 10/04/2017]

[16] Using WordNet to Measure Semantic Orientations of Adjectives[Online]. Available http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.409.2489&rep=rep1&type=pdf [Accessed 11/02/2017]

[17] Building and Exploiting EmotiNet, a Knowledge Base for Emotion Detection Based on the Appraisal Theory Model [Online]. Available

http://ieeexplore.ieee.org/abstract/document/6042854/ [Accessed 10/04/2017]

[18] A Sentiment model for Swedish with automatically created training data and handlers for language specific traits[Online]. Available

http://www8.cs.umu.se/~johanna/sltc2016/abstracts/SLTC_2016_paper_12.pdf [Accessed 11/03/2017]

[19] MapReduced - Based Bayesian Automatic text Classifier [Online]. Available

https://books.google.lk/books?id=S4y5BQAAQBAJ&pg=PA121&lpg=PA121&dq=Bayesian+algorithm+was+introduced+in+2012+by+Zhen+Niu+et+al.+(2012).&source=bl&ots=iez88ANH8-&sig=OEndwdUgCeBL3jXHGIrv0QmxSbE&hl=en&sa=X&ved=0ahUKEwjxxtX51fnWAhUGuY8KHSrZByIQ6AEIJjAA#v=onepage&q=Bayesian%20algorithm%20was%20introduced%20in%202012%20by%20Zhen%20Niu%20et%20al.%20(2012).&f=false [Accessed 11/07/2017]

[20] Twitter as a Corpus for Sentiment Analysis and Opinion Mining [Online]. Available

http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=3AE85D468A292EEB228F4F8111A5C5C3?doi=10.1.1.455.1616&rep=rep1&type=pdf [Accessed 11/06/2017]

[21] Ensemble of feature sets and classification algorithms for sentiment classification [Online]. Available

http://www.sciencedirect.com/science/article/pii/S0020025510005682 [Accessed 10/06/2017]

[22] Twitter Sentiment Analysis:The Good the Bad and the OMG! By Efthymios Kouloumpis∗

i-sieve Technologies, Athens, Greece and Theresa Wilson∗HLT Center of Excellence, Johns Hopkins University, Baltimore, MD, USA and Johanna Moore School of Informatics, University of Edinburgh

Edinburgh, UK

[23] Enhanced Sentiment Learning Using Twitter Hashtags and Smileys [Online]. Available

http://www.aclweb.org/anthology/C10-2028 [Accessed 11/07/2017]