Twitter Sentiment Analysis using Unsupervised and Supervised Approach: A Case Study.

BY

SHERESH ZAHOOR

2K18/SPD/10

Department of Electronics and Communication Engineering

Delhi Technological University

New Delhi, Delhi 110042

Submitted for the

Major presentation in

Fourth semester for the degree

Of

Masters of Technology (2018-2020)

In

Signal Processing and Digital Design

Submitted By-

SHERESH ZAHOOR

2K18/SPD/10

Under the Guidance

of

Dr. Rajesh Rohilla

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY, DELHI

ACKNOWLEDGEMENT

I would like to express my gratitude towards my family for their unconditional support, my friends and my dearest Shams who made me believe that I can do a lot more than I think I can. Thank you for being my biggest support, my light in the dark times. I would also like to thank people who have contributed their precious time and effort to help me, without whom it would not have been possible for me to understand and complete the project.

I would like to thank Dr. Rajesh Rohilla, Department of Electronics and Communication Engineering, my Project guide, support, motivation and encouragement throughout the period this work was carried out. His readiness for consultation at all times, his educative comments, his concern and assistance even with practical things have been invaluable.

Sahrish

Sheresh Zahoor

(2K18/SPD/10)

Department of Electronics and Communication Engineering

Delhi Technological University

CERTIFICATE

I hereby certify that the Dissertation titled "**Twitter Sentiment Analysis Using Unsupervised and Supervised Approach: A Case Study**" which is submitted by **SHERESH ZAHOOR**, **2K18/SPD/10** [ECE Department], Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

APPROVAL FOR SUBMISSION

Dr. Rajesh Rohilla

Professor

Contents

List of Publications:

1) Twitter Sentiment Analysis Using Lexical or Rule Based Approach: A Case Study (Zahoor & Rohilla, 2020) ICRITO'20- 978-1-7281-7016-9/20/$31.00 ©2020 IEEE- paper accepted.
2) Twitter Sentiment Analysis Using Machine Learning Approach: A Case Study (ICACCM 2020) IEEE conference – paper accepted

ABSTRACT

Opinion analysis or sentiment analysis is one of the most sorted out technique these days in order to determine the sentiments or emotions of people regarding any event. This technique has come into existence because of extensive use of social media platforms like Facebook, Twitter etc by people to express their emotions regarding any event that has occurred or any event that is most likely to happen, be that the release of a movie or a political rally that is about to take place. People make sure to express their sentiments. Sentiment analysis proves very beneficial for any company selling a product to know how their product was received by people or by any political party to determine how people are reacting towards their running candidate. Different events occur worldwide so it is not very easy to determine the emotions and sentiments of people regarding these events; it results in huge amount of data and many steps to reach any conclusion about the sentiment.

In order to analyze these sentiments two approaches of machine learning can be used – unsupervised or supervised. Machine learning algorithms can be used to determine whether a series of words reflect a positive, negative or neutral meaning. Unsupervised learning involves a rule based or lexical approach and this can be done using the pre-built open source libraries like TextBlob, VADER. Unsupervised learning is a simple and efficient approach that has been used in the past so many years to determine the sentiments or opinions, over the years many libraries have been built to ease the task of analyzing the sentiments. This library is used to determine whether the sentiments are positive, negative or neutral effectively. Once the sentiments have been recorded and the data is converted to a more structured form, these can be fed to the machine learning algorithms like Naïve Bayes, SVM, LSTM and so on in order to predict the accuracy with which these algorithms can predict the sentiment.

CHAPTER 1

INTRODUCTION

Opinions have always been of utmost importance in any field, be that politics, automation, fashion etc. every aspect needs and works on opinions and sentiments of people related to these field or events that occur every day throughout the world. Before the advent of the social networking sites like Twitter, Facebook, Instagram etc., people relied on newspapers, television sets, magazines for news or any big event occurring in the world, but now because of the advent of social media people have all the information in their hands and are not dependent on a single source for their news or opinions. Social media has drastically changed the outlook of life of many people and has given them a platform to express their views related to any event or any product. Social media allows a person to express his like or dislike towards any product, person etc.

Researches indicate that using the social media sites is considered as the best way to grow a business in terms of money, time, effort and other resources [2]. However, the massive availability of opinions and the unstructured nature in which these opinions are present makes it very difficult to extract any useful information out of them, so over the years it has become very important to devise and develop techniques that could ease extracting information out of the unstructured opinions that are present on any Social networking site.

Sentiment analysis, also referred to as opinion mining is one of the most recent technique that has been developed in order to extract information that has been gathered from any social networking site. Sentiment analysis is an automated process of computationally identifying and classifying the opinions expressed in a piece of text to determine whether the sentiment of a person towards a particular event, product, person etc is positive, negative or neutral. Sentiment analysis is considered the most advance

way of determining opinions as it makes use of AI to classify a piece of text into positive, negative or neutral.

Twitter is one of the most growing social media sites and has become very famous over the last few years as people openly express their opinions in the form of tweets with a maximum length of 140 characters. As of 2019, twitter has around 330 million active users. Of these, more than 40 percent, or more specifically, 134 million, use the service on a daily basis [3]. This accounts for a large amount of unstructured data which can prove to be very useful to determine opinions or sentiments.

This is a case study of a number of events that have occurred throughout the world and involves determining how people perceived these events. It involves the implementation of Twitter sentiment analysis by utilizing APIs provided by the twitter itself. Tweets were collected related to events like Howdy Modi, a gathering that occurred in Houston, Haryana Assembly Elections, a movie release i.e. The sky is pink, Delhi odd even and an international conference UNGA. The data of all these events was scraped from twitter and was analyzed using an open source Python library called TextBlob and another in-built library NLTK-VADER.

CHAPTER 2

LITERATURE REVIEW

Sentiment analysis refers to the study of text on various levels. Document level [9] This is the first level where objects are classified according to their attributes. (Turney, 2002; Pang and Lee, 2004), The second level is the sentence level wherein [11] the sentences are classified on the basis of whether they represent negative, positive or neutral sentiments (Hu and Liu, 2004; Kim and Hovy, 2004). The next level is the phrase level [12] wherein it is determined whether the expression is unbiassed or polar, followed by the removal of uncertainty.

Pak and Paroubek (2010) [5] classified tasks as subjective and objective. For subjective data, the information was gathered from the user tweets such as text or image or symbols. For objective information on the other hand, the information was gathered from authentic sources such as newspapers.

Information which is taken for analysis is simply a sample of flowing tweets collected by using queries. In the past years there have been numerous documents observing the Twitter sentiment and buzz [5], [12] (Jansen et al. 2009; Pak and Paroubek 2010; O'Connor et al. 2010; Tumasjan et al. 2010; Bifet and Frank 2010; Barbosa and Feng 2010; Davidov, Tsur, and Rappoport 2010). Barbosa and Feng (2010) [4] analyzed the sentimental classification on Twitter data by determining the polarity of tweets on the basis of symbols, retweets, emoticons and even syntax of the Tweets. Kamps et al. (2002) analyzed the data by using the lexical database such as WordNet which is the description of lexemes. This contains the emotional content of a word. The distance metric of words is used to determine semantic polarity of adjectives.

It was Hearst [17] in 1992 and Kessler et al [18] in 1997 who initiated the classification of text based on the sentiments or emotions. There are two main approaches for classification of a piece of text, one is the lexical approach and the other is machine learning approach. Machine learning techniques gained interest because of the ability to extract features, capturing context [19]. These techniques are mostly used to detect the sentiment of an entire document and not just a few sentences.

In sentiment analysis, it is not always possible to detect the sentiment based on a single word, so the concept of n-gram extraction was introduced and found to be effective by Pederson in 2001 [20]. Other approaches involved selecting only a subset of the words detected using part-of-speech (POS) recognizer. Also, a step-wise approach of classification was developed later on which first involved the removal of objective sentences [9].

Over the years a lot of research has been done based on the tweets collected from twitter. It was Barbosa and Feng in 2010 who analyzed the Twitter data by determining the polarity of tweets on the basis of symbols, retweets, emoticons and even syntax of the Tweets. Initially, Naïve Bayes algorithm was used to analyze the sentiments of movie reviews [21]. It was Tong and Koller who later made use of Support Vector Machine in order to further improve the accuracy of prediction [22].

CHAPTER 3

SENTIMENT ANALYSIS

Sentiment analysis is one of the most recently developed techniques of determining opinions of people. It is a contextual mining of text which extracts information from a piece of text. Sentiment analysis or "opinion mining" refers to the use of Natural Language Processing to determine the opinions and emotions of a piece of text. Sentiments analysis is used to identify opinions, sentiments, attitude, and state of mind of people towards a product or service and classify them as positive, negative and neutral. Sentiment analysis can be done using two approaches:

Lexical or rule based approach-

In this method the text is represented as a bag of words and these words are then compared to the contents of dictionaries which contain words along with its associated semantic orientation. It is considered to be the unsupervised approach of sentiment analysis and is efficient in determining the polarity i.e. whether a given piece of text is positive, negative or neutral. The most commonly used library for sentiment analysis using lexical or rule based approach is TextBlob [4]. It is a python library for processing of textual data and simplest way of diving into common natural language processing tasks like sentiment analysis, classification. VADER is another such library; however, it has been specifically designed for analysis of text of social-media as the language of social-media is different from that of normal spoken or written language.

TextBlob

Textblob is an open source Python library that is based on NLTK- Natural Language Tool Kit and is used for analyzing a piece of text and assigning a polarity score to it. Polarity is a float value between [-1, 1] where 0 indicates a neutral sentiment, 1 indicates positive and -1 indicates negative sentiment. TextBlob analyses each word in a piece of text and assigns a semantic score to each word, then this score is

weighted i.e. a weighted average is computed to assign a score to the complete sentence, the score is based on the polarity of each word in the sentence. TextBlob computes another important factor, Subjectivity- it is a float value having a range [0, 1] where 0 indicates that the sentence is objective which means that it is based on factual data whereas 1 indicates that the sentence is subjective which means that it is based on emotions, feelings, opinions, desires, and allegations of a person. It is an efficient way of analyzing raw tweets or any piece of text. TextBlob has been designed to work on any piece of text and not just the data from social media. However, the advent of social media has made it popular for social media sentiment analysis because of its simplicity at handling raw and unstructured data. This library plays a very important role in ensuring that the unstructured data that is obtained from Twitter undergoes sufficient amount of preprocessing to be in a position for analysis.

VADER

Valence Aware Dictionary and sentiment Reasoner is another python library that can be used for lexicon based sentiment analysis, however the difference between VADER and TextBlob is that VADER was created mostly to analyze the social media texts i.e. it is able to comprehend and understand the social media slangs and language better than TextBlob and gives better result with social media texts than TextBlob. It is capable of analyzing repetitive vocabulary and the correlation coefficient shows that VADER ($r = 0.881$) performs as well as individual human raters ($r = 0.888$) at matching ground truth (aggregated group mean from 20 human raters for sentiment intensity of each tweet) [1]. Thus, VADER is considered to be highly efficient for social media text analysis. VADER breaks down the sentiment intensity score into a positive, negative and neutral component, these are further normalized to be within the range [-1, 1] as a compound score.

Machine learning based or supervised approach

This approach is based on machine learning algorithms like Naïve Bayes, SVM, Random Forest Classifier and LSTM to determine the polarity of a piece of text. This method has been introduced recently and is a very efficient approach to classify text based on its polarity. In this method the dataset is divided into training set and test set and then accuracy with which the algorithm can determine the polarity is predicted.

In this research the lexical or rule based approach has been used to record the sentiment for a number of events. For each of these events minimum of 5000 tweets were scraped from the twitter, all this data is in the raw for i.e. it is unlabeled, there are no output and input parameters. Using the lexical approach,

sentiments of all these tweets can be found out in a simple and efficient way. And using the lexical approach the unstructured data is converted into a structured form i.e. the sentiments are recorded. Now that we have a structured dataset, this data is fed to the supervised algorithms in order to determine how accurately the algorithm is used to predict the sentiment.

Naïve Bayes Algorithm

This is one of the supervised algorithm that is based on the probabilistic approach to classify the text to a particular class i.e. positive or negative [12]. This algorithm calculates the probability of all the words in the dataset and then classifies the tweets or text into particular categories. This algorithm is based on the Bayes rule, according to which

$$P(x|y) = \frac{P(y|x)\,P(x)}{P(y)}$$

Where

x, y = events

$P(x|y)$ = Probability of x given y is true

$P(y|x)$ = Probability of y given x is true

$P(x), P(y)$ = Independent probabilities of x and y

Using Bayes rule, we can find the probability of **x** happening, given that **y** has occurred. Here, **y** is the evidence and **x** is the hypothesis. It is assumed that the features or the predictors are independent of each other i.e. one feature does not have any impact on the presence of other features.

Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm that has been used for both classification and regression

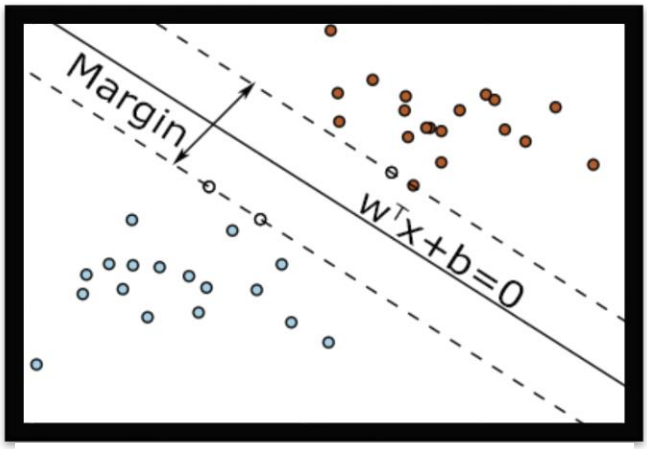


Figure 1: SVM depicting Hyperplane

problems. SVM classifies by determining a hyperplane to classify the data distributed in the n-dimensional space. The classification is done based on the mathematical functions called kernels and these kernels are used to determine a hyperplane. Two different classes exist on the opposite sides of the hyperplane and thus this plane could be considered a decision boundary which could help in simple classification of the data points available. The mathematical functions called kernels can be Linear, sigmoid, RBF. The SVM has several advantages which are very helpful in case of sentiment analysis like its potential to handle large features and also it is immensely robust when the features are linearly separable [13]. SVM involves a number of steps, first being training the machine by providing it a structured data. Now, the trained machine could be utilized to make predictions for the test data. Throughout this classification, one step that is very important is dividing the dataset into a test and training data. The idea behind this classification is to find a hyperplane that maximizes the margin around the hyperplane that is separating the data points available [14].

Random Forest Classifier

This is a classifier that is composed of several Decision Trees as a single Decision Tree suffers from several disadvantages like noise which can affect the overall results and performance. However, Random Forest classifier has several advantages over the Decision Tree like it is robust when compared to a Decision Tree and the reason being, a Random Forest Classifier uses the concept of Bootstrapping i.e. each tree is trained on a different training data as we divide the training data into subsets equivalent to the number of trees. This ensures that each tree has a different and its own training data. Also, its more accurate as it uses the concept of Bagging which ensures that the output of all the classifiers is averaged which in turn ensures a consistent result [15].

Random Forest Classifier deals with a number of hyperparameters like-

1) Number of Decision trees that a Random Forest Classifier is made of
2) Number of features that are used
3) The Longest path from a root to a leaf i.e. The Depth of each tree

LSTM

Long Short Term Memory Networks which are mostly referred to as LSTMs are special kind of Recurrent Neural Networks that have the capability to learn long-term dependencies. These networks are widely used and over the years have improvised tremendously. Earlier it was RNN that was widely used to understand the sentences based on the meaning of previous words, these form loops and thus allowing the information to be stored. This stored information allows the network to understand the complete sentence and helps in prediction. However, there are times when more context is required to understand and more information needs to be stored. Thus as the gap increases, RNN becomes obsolete and is not able to understand and predict the meaning. LSTM solves this issue as it is considered an advanced or extended version of RNN that is able to learn long-term dependencies. The structure of LSTM is chai-like similar to that of RNN. However, the repeating module is slightly different, instead of having a single neural network layer, it consists of four networks that interact in a unique way.

CHAPTER 4

METHODOLOGY

Lexical or rule based approach is based on a number of steps that are very critical in extracting the sentiment or opinion from the raw tweets obtained from twitter. These steps are:

- Data collection
- Data pre-processing

    - Tokenization
    - N-grams Extraction
    - Stemming and Lemmatization
    - Stop words removal
- Part of speech tagging (POS)
- Sentiment analysis using in-built dictionary
- Feeding the structured data to Machine Learning algorithms
- Visualization of results

Data collection

In order to be able to collect data from twitter it is important to form an account on twitter and then go to developer.twitter.com/en/apps to create an app that allows collection of tweets based on the query. However one needs permission from the twitter to become a developer and after getting the permission, customer API keys are provided which are to be kept safe for further use while scraping the tweets. For one-time collection of tweets REST API is used and in this case study in order to collect the tweets based on different queries REST API has been used. Also, the data has been stored in the .CSV files in this research. The .CSV file will have the following columns:

- Created (date)
- Text (the actual tweet)
- Retweet
- Hashtag
- Followers
- Friends

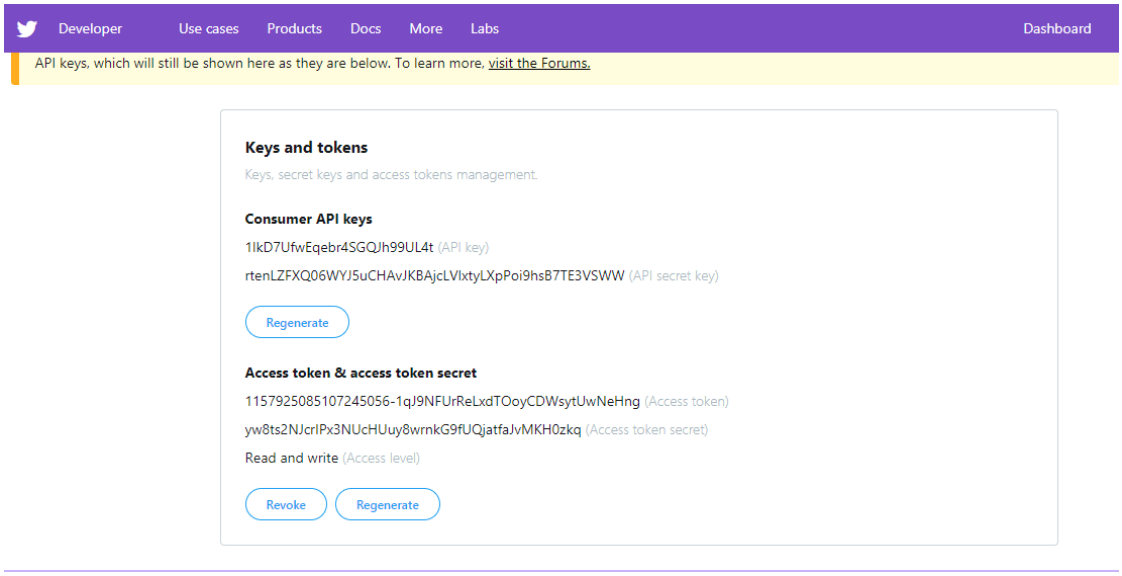


Figure 2: Developer page of Twitter

Data pre-processing

Before the actual sentiment analysis techniques are applied, a number of pre-processing steps are performed so as to make the tweet ready to be analyzed because the tweets obtained are in unstructured form and cannot be used as such without any processing. The typical pre-processing steps are as follows:

- **Tokenization or Bag-of-Words Creation:** Tokenization involves separation of words from URLs, hashtags, at-mentions. Thus it is a method of breaking a string, text into a list of tokens. It is very important in the sentiment analysis as this gives an idea about the words that constitute a tweet. It is done in order to:
  - Count the number of words in a piece of text
  - Count the frequency of that word i.e. how many that word was used or repeated in the sentence
- **N-grams Extraction:** In this technique accompanying words are grouped together into phrases called n-grams. It has been observed that when words are grouped together, it improves the quality of sentiment analysis; however, there is no solution to the size of n-gram to be used.
- **Stemming and Lemmatization:** In stemming words will be replaced by their stems or roots i.e. the words like write, writer and writing are mapped to a single word i.e. write and they are

counted together as a result of which the dimensionality of the bag of words is reduced. In case of lemmatization all the verbs are transformed to the infinite tense whereas all the nouns are transformed to the singular form.

- **Stop words removal:** Stop words i.e. the prepositions, articles have high frequency of occurrence but they do not affect the final sentiment of the text and thus should be removed from the text while pre-processing. For this purpose, the NLTK stop words in built library can be used.

Part-of-Speech Tagging (POS)

This is the process of automatically tagging each word in the text in terms of the part of speech it belongs to like noun, pronoun, adverb etc. Pak and Paroubek analysed the distribution of POS tagging specifically for Twitter messages and identified the following patterns [5]:

– Subjective texts (carrying the sentiment) often contain more pronouns, rather than common and proper nouns;

– Creators of subjective texts talk from the first person point of view (describing themselves) or from the second person (addressing the audience), while the authors of objectives texts normally speak in the third person;

– Subjective messages often use past simple tense;

– Subjective texts contain many verbs in a base form and many modal verbs;

– Authors of subjective texts use superlative adjectives to express their emotions, while comparative adjectives are mostly used for expressing facts in the objective messages;

– Adverbs are mostly used in subjective texts;

– Very common for expressing a positive opinion in the text is the usage of superlative adverbs, for example the most, the biggest, the best;

– In the negative sentences there is a high rate of using verbs in the past tense, probably because the authors tend to express their negative emotions about losses or unsatisfactory experiences in the past. For example: lost, tired, and bored.

Sentiment analysis using in-built dictionary

In the lexical approach, sentiment analysis relies on an in-built library or dictionary of words with pre-calculated polarity. This method is considered as a part of Machine learning i.e. Unsupervised approach where the unlabelled data is analyzed to determine the polarity or sentiment of a tweet. In this research two in-built libraries have been used to analyze the sentiment of particular events and the results obtained are compared with each other. However, VADER is a library specifically designed for the social media texts or messages, so it has a much better hold at analyzing data from social media platform. The idea behind the lexicon approach-

- Constructing a lexicon of words with polarities
- Forming a bag-of-words from the given text
- Pre-processing the data like the removal of stop words etc
- Comparing each of the word with the lexicon and assigning a polarity to it
- Calculating the whole sentiment score for the text by adding the sentiment score or polarity of each word in the text

CHAPTER 5

CASES

Haryana Assembly polls

Haryana assembly polls were held in Haryana, India in the month of October and the final turnout was 68.47% [6]. Tweets for this event were collected prior to the event and post the event to understand the overall sentiment of people towards the election and the leading party. Around 16,000 tweets with the hash tag #HaryanaAssemblyPolls were collected and analyzed using two in built libraries i.e. TextBlob and VADER.

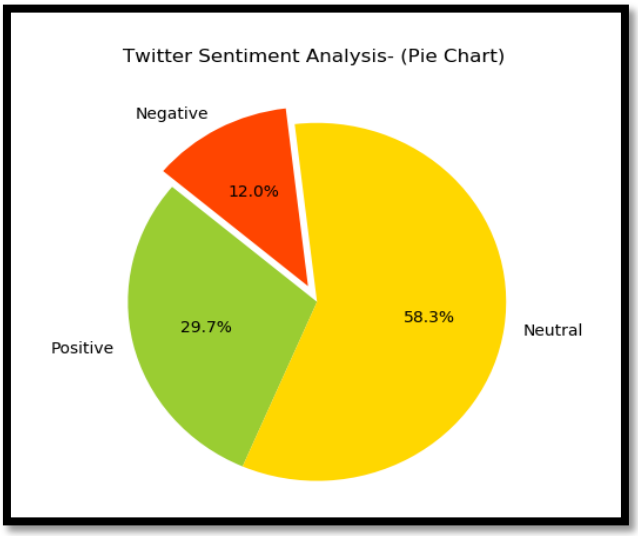


Figure 3: Haryana Assembly Polls using TextBlob

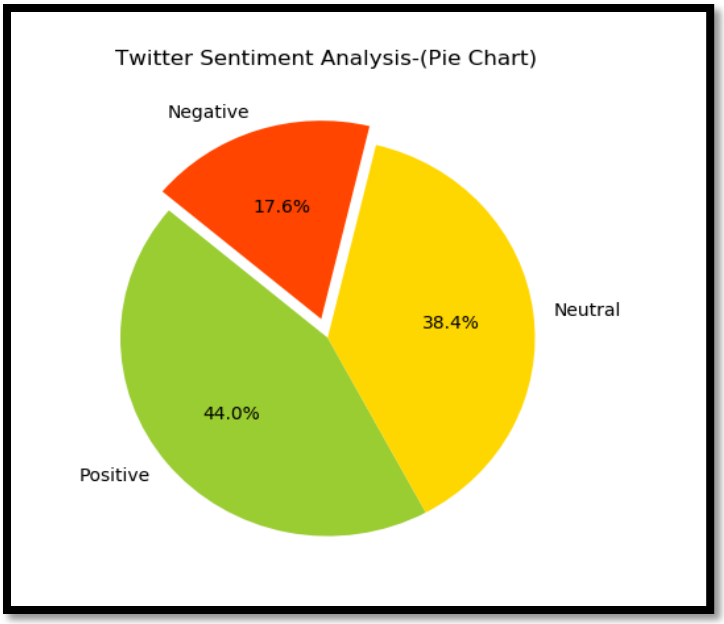


Figure 4: Sentiment Analysis using VADER

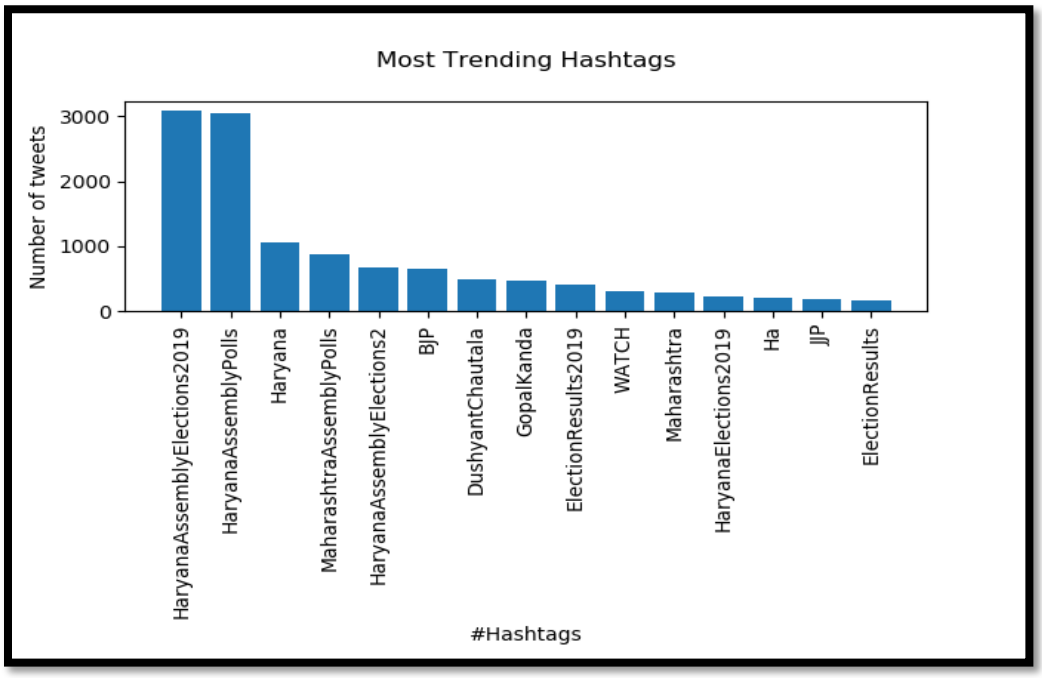


Figure 5: Trending Hashtags

It was noted that a single party was mentioned more than any other party i.e. BJP. So the tweets for that particular party and their candidate were scraped from the twitter to analyze the sentiment of people towards the BJP and their candidate i.e. ML Khattar. Around 4000 tweets for BJP and 2000 for Khattar were scraped and analyzed using TextBlob as well as VADER.

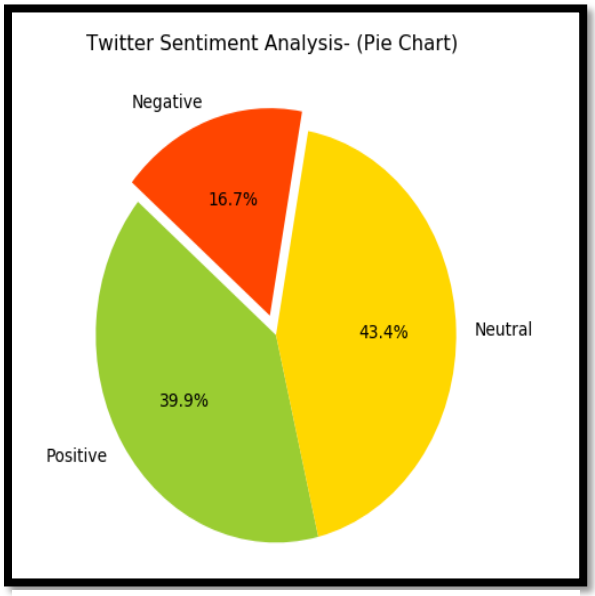


Figure 6: Sentiment analysis of BJP using TextBlob

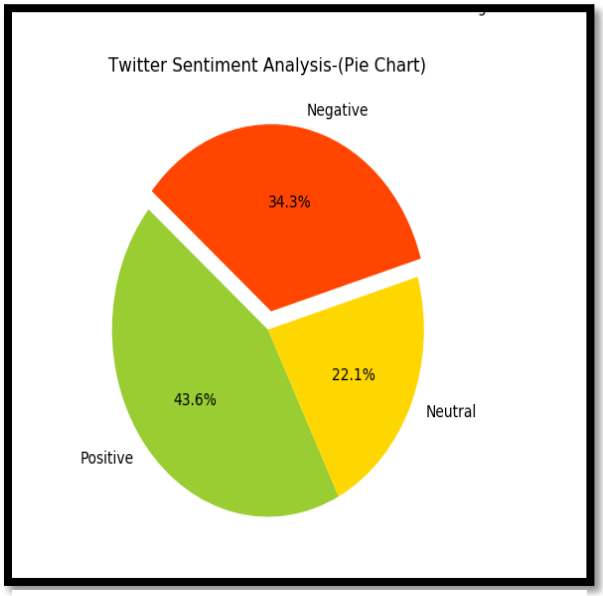


Figure 7: Sentiment analysis using VADER

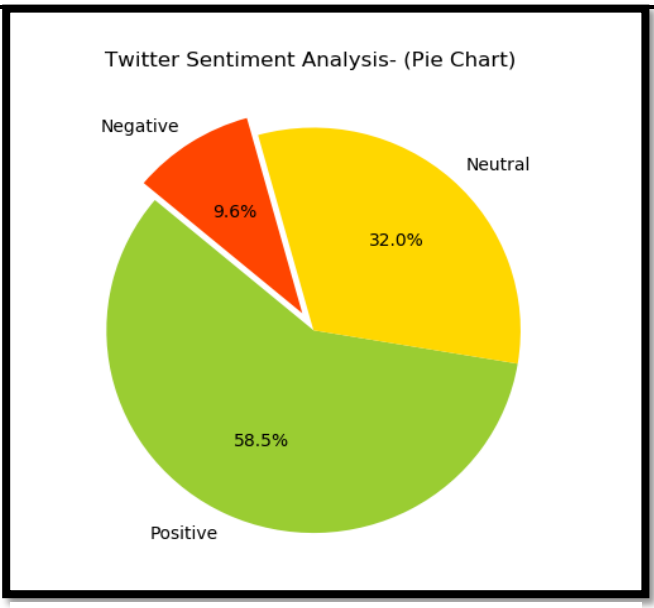


Figure 8: Sentiment analysis of ML Khattar using TextBlob

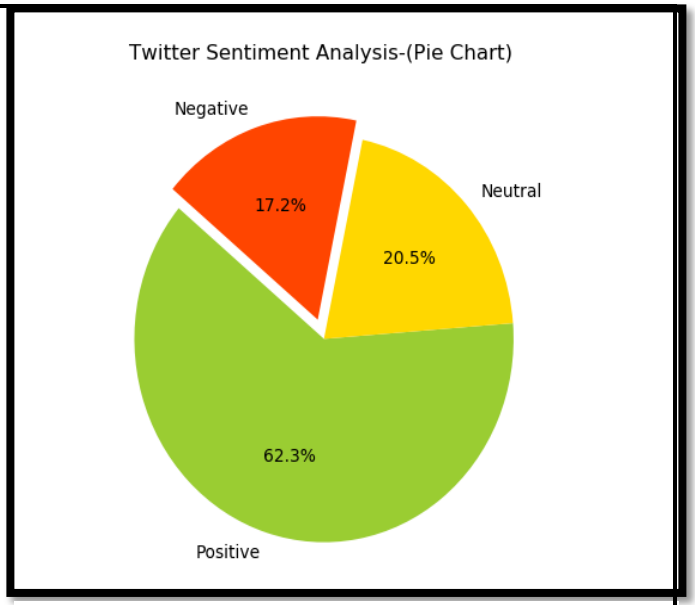


Figure 9: Sentiment analysis of ML Khattar using VADER

In case of TextBlob polarity and subjectivity of a tweets are calculated, subjectivity is float value where 0 indicates that a sentence is objective i.e. it is based on factual details rather on personal opinion whereas 1 indicates that the sentence is a personal opinion, emotion etc. whereas VADER breaks down sentiment intensity scores into positive, negative and neutral which are normalized in the range [-1, 1] which is called compound score.

It was noted that the overall sentiment towards BJPs candidate ML Khattar was positive and was obtained positive using both the in-built libraries. Also, the sentiment towards the party itself was obtained to be inclined towards a positive response, i.e. around 39.9% positive using TextBlob and 43.6% positive using VADER.

```
Performing Sentiment Analysis.....Total Tweets:  4042
Positive =  1612
Neutral=  1753
Negative=  677

b'"I urge Shiv Sena to find a way to join with BJP to form the govt," said senior BJP leader
@Swamy39.\n\n#Maharashtra\xe2\x80\xa6 https://t.co/7qLiG7xDfB'  =>  Sentiment(polarity=0.0,
subjectivity=0.0)

b'RT @ggiittiikkaa: Indian Media since a week -\nBJP + SS\nBJP + NCP\nSS + NCP + Congress\nBJP +
Congress\nCongress + NCP + AIMIM + SS\nFalana +Dhi\xe2\x80\xa6'  =>  Sentiment(polarity=0.0,
subjectivity=0.0)

b'RT @preadi: Hindus of South India are special.\n\nAt Sabarimala, they need BJP to stop CPM
atrocities.\n\nAt Tirupati, they need BJP to foil th\xe2\x80\xa6'  =>  Sentiment(polarity=0.0,
subjectivity=0.0)

b'RT @sankrant: The story of BJP everywhere. People vote it to fight the conversion mafia, to
preserve culture and traditions.\n\nAfter it come\xe2\x80\xa6'  =>  Sentiment(polarity=0.0,
subjectivity=0.0)

b'@dasraghubar @narendramodi @BJP4Jharkhand @AmitShah @JPNadda @nkishoreyadav @OmMathur_bjp @LaxmanMP
@idharampalsingh Phir se BJP sarkar!@!'  =>  Sentiment(polarity=0.0, subjectivity=0.0)
```

Figure 10: Random tweets using TextBlob

```
Performing sentiment analysis...
.....total 4042
negative 1388
positive 1762
neutral 892

b'#RT @TOIIndiaNews: Pakistan may have released poisonous gas to pollute air in India, says UP BJP
leader https://t.co/NtICLEIqtC'=> {'neg': 0.161, 'neu': 0.839, 'pos': 0.0, 'compound': -0.3182}

b'RT @preadi: Hindus of South India are special.\n\nAt Sabarimala, they need BJP to stop CPM
atrocities.\n\nAt Tirupati, they need BJP to foil th\xe2\x80\xa6'=> {'neg': 0.161, 'neu': 0.839,
'pos': 0.0, 'compound': -0.3182}

b'RT @ggiittiikkaa: Indian Media since a week -\nBJP + SS\nBJP + NCP\nSS + NCP + Congress\nBJP +
Congress\nCongress + NCP + AIMIM + SS\nFalana +Dhi\xe2\x80\xa6'=> {'neg': 0.161, 'neu': 0.839, 'pos':
0.0, 'compound': -0.3182}

b"RT @ashoswai: These guys are even capable of blaming Pakistan and China for India's population
growth! https://t.co/WhmQqZ67Ff via @ndtv"=> {'neg': 0.161, 'neu': 0.839, 'pos': 0.0, 'compound':
-0.3182}

b"RT @ashoswai: These guys are even capable of blaming Pakistan and China for India's population
growth! https://t.co/WhmQqZ67Ff via @ndtv"=> {'neg': 0.161, 'neu': 0.839, 'pos': 0.0, 'compound':
-0.3182}
```

Figure 11: Random tweets of BJP using VADER

```
Performing Sentiment Analysis.....Total Tweets:  2507
Positive =  1466
Neutral=  801
Negative=  240

b'@mlkhattar @cmohry @AmitShah @narendramodi sir once again reminding you the local issues of
Gurugram Vidhan Sabha 1\xe2\x80\xa6 https://t.co/P0zYy1ud9u'  =>  Sentiment(polarity=0.0,
subjectivity=0.0)

b'@DNivasi @LetMeBreathe_In @PMOIndia @PrakashJavdekar @ArvindKejriwal @SupremeCourtIND
@CPCB_OFFICIAL @PMOIndia\xe2\x80\xa6 https://t.co/zZTiv8ZoF1'  =>  Sentiment(polarity=0.0,
subjectivity=0.0)

b'RT @TajinderBagga: Wonderful work. I request @mlkhattar\nSahab, @capt_amarinder sahab to check this
\nhttps://t.co/1weXKtVO3M'  =>  Sentiment(polarity=1.0, subjectivity=1.0)

b'@mlkhattar : Sir, long queue at Gurgaon toll for commercial drivers on a daily. It not only wastes
time of passenge\xe2\x80\xa6 https://t.co/4LBKOvgc6h'  =>  Sentiment(polarity=-0.05,
subjectivity=0.27999999999999997)

b'RT @TajinderBagga: Wonderful work. I request @mlkhattar\nSahab, @capt_amarinder sahab to check this
\nhttps://t.co/1weXKtVO3M'  =>  Sentiment(polarity=1.0, subjectivity=1.0)
```

Figure 12: Random tweets of ML Khattar using TextBlob

```
Performing sentiment analysis...
.....total 2507
negative 430
positive 1563
neutral 514

b'RT @TajinderBagga: Wonderful work. I request @mlkhattar\nSahab, @capt_amarinder sahab to check this
\nhttps://t.co/1weXKtVO3M'=> {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}

b'@cmohry @cbseindia29 @mlkhattar Due to smog 2 days school off.. Now schools are doing as per their
wish. Tomorrow i\xe2\x80\xa6 https://t.co/n2v5l5YWe5'=> {'neg': 0.0, 'neu': 1.0, 'pos': 0.0,
'compound': 0.0}

b'@nsitharaman @PMOIndia @mlkhattar Being a startup we are running pillar to Post for one single
document from last 1\xe2\x80\xa6 https://t.co/Sc0zwFpPPk'=> {'neg': 0.0, 'neu': 1.0, 'pos': 0.0,
'compound': 0.0}

b'RT @AbrolPoonam: @MeraGurgaon @TrafficGGM @DC_Gurugram @IGtraffic_hry @gurgaonpolice
@police_haryana @sudhirsinglabjp @cmohry @mlkhattar Th\xe2\x80\xa6'=> {'neg': 0.0, 'neu': 1.0, 'pos':
0.0, 'compound': 0.0}

b'@ArvindKejriwal @capt_amarinder @mlkhattar you guys did nothing to stop this and we people suffer
\nAur quality is s\xe2\x80\xa6 https://t.co/u445zT3psv'=> {'neg': 0.0, 'neu': 1.0, 'pos': 0.0,
'compound': 0.0}
```

Figure 13: Random tweets of ML Khattar using VADER

Now, after obtaining a structured data, all the tweets along with their sentiments were fed to the Machine Learning Algorithms in order to compare the accuracy with which the sentiment can be predicted using these algorithms.

Naïve Bayes

```
In [10]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/ActualNaiveBayes.py',
wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
0.9750994548401356
Naive Bayes Accuracy: 97.51
0.968

Naive Bayes Accuracy with the Test Set: 96.80
```

Figure 14: Accuracy of Haryana Assembly Polls using Naive Bayes

```
In [11]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/ActualNaiveBayes.py',
wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
0.9646288209606987
Naive Bayes Accuracy: 96.46
0.96

Naive Bayes Accuracy with the Test Set: 96.00
```

Figure 15: Accuracy of BJP using Naive Bayes

```
In [12]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/ActualNaiveBayes.py',
wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
0.9786910197869102
Naive Bayes Accuracy: 97.87
0.984

Naive Bayes Accuracy with the Test Set: 98.40
```

Figure 16: Accuracy of ML Khattar using Naive Bayes

SVM

```
In [7]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/svmm.py', wdir='C:/
Users/Sahrish Zahoor/Desktop/Project code')
[nltk_data] Downloading package stopwords to C:\Users\Sahrish
[nltk_data]     Zahoor\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
positive    4831
negative    1955
Name: sentiment, dtype: int64
0.9605893186003683
```

Figure 17: Accuracy of Haryana Assembly Polls using SVM

```
In [12]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/svmm.py', wdir='C:/
Users/Sahrish Zahoor/Desktop/Project code')
[nltk_data] Downloading package stopwords to C:\Users\Sahrish
[nltk_data]     Zahoor\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
positive    1612
negative     677
Name: sentiment, dtype: int64
0.8973799126637555
```

Figure 18: Accuracy of BJP using SVM

```
In [15]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/svmm.py', wdir='C:/
Users/Sahrish Zahoor/Desktop/Project code')
[nltk_data] Downloading package stopwords to C:\Users\Sahrish
[nltk_data]     Zahoor\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
positive    529
negative    127
Name: sentiment, dtype: int64
0.9239543726235742
```

Figure 19: Accuracy of ML Khattar using SVM

Random Forest Classifier

```
In [116]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/Randomforestclassifier.py', wdir='C:/Users/Sahrish Zahoor/Desktop/
Project code')
Loading train and test data
Data loaded
Applying TF-IDF transformation
Training Random Forest Classifier
Predicting on train data
Training accuracy: 99.981577008106115%
Predicting on test data
Testing accuracy: 95.9499263622975%
```

Figure 20: Accuracy of Haryana Assembly Polls using Random Forest Classifier

```
In [121]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/Randomforestclassifier.py', wdir='C:/Users/Sahrish Zahoor/Desktop/
Project code')
Loading train and test data
Data loaded
Applying TF-IDF transformation
Training Random Forest Classifier
Predicting on train data
Training accuracy: 99.94538503549973%
Predicting on test data
Testing accuracy: 90.611353711790393%
```

Figure 21: Accuracy of BJP using Random Forest Classifier

```
In [126]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/Randomforestclassifier.py', wdir='C:/Users/Sahrish Zahoor/Desktop/
Project code')
Loading train and test data
Data loaded
Applying TF-IDF transformation
Training Random Forest Classifier
Predicting on train data
Training accuracy: 98.473282442748091%
Predicting on test data
Testing accuracy: 88.636363636363643%
```

Figure 22: Accuracy of ML Khattar using Random Forest Classifier

LSTM

```
In [115]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/lstmaccuracy.py', wdir='C:/
Users/Sahrish Zahoor/Desktop/Project code')
Loading train and test data
Data loaded
Tokenizing and padding data
Tokenizing and padding complete
Creating the LSTM model
Model: "sequential_14"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_13 (Embedding)     (None, 19, 128)           256000
_________________________________________________________________
spatial_dropout1d_13 (Spatia (None, 19, 128)           0
_________________________________________________________________
lstm_13 (LSTM)               (None, 256)               394240
_________________________________________________________________
dense_13 (Dense)             (None, 2)                 514
=================================================================
Total params: 650,754
Trainable params: 650,754
Non-trainable params: 0
_________________________________________________________________
Training the LSTM model
C:\Users\Sahrish Zahoor\Anaconda3\envs\venv\lib\site-packages\tensorflow_core\python\framework\indexed_slices.py:433: UserWarning:
Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
Train on 4342 samples, validate on 1086 samples
Epoch 1/10
4342/4342 [==============================] - 14s 3ms/step - loss: 0.4671 - accuracy: 0.7672 - val_loss: 0.1853 - val_accuracy: 0.9171
Epoch 2/10
4342/4342 [==============================] - 12s 3ms/step - loss: 0.1464 - accuracy: 0.9454 - val_loss: 0.1416 - val_accuracy: 0.9355
Epoch 3/10
4342/4342 [==============================] - 9s 2ms/step - loss: 0.0872 - accuracy: 0.9689 - val_loss: 0.1245 - val_accuracy: 0.9466
Epoch 4/10
4342/4342 [==============================] - 10s 2ms/step - loss: 0.0572 - accuracy: 0.9800 - val_loss: 0.1294 - val_accuracy: 0.9494
Epoch 5/10
4342/4342 [==============================] - 11s 2ms/step - loss: 0.0432 - accuracy: 0.9855 - val_loss: 0.1271 - val_accuracy: 0.9484
Epoch 6/10
4342/4342 [==============================] - 10s 2ms/step - loss: 0.0331 - accuracy: 0.9885 - val_loss: 0.1498 - val_accuracy: 0.9512
Epoch 7/10
4342/4342 [==============================] - 10s 2ms/step - loss: 0.0240 - accuracy: 0.9922 - val_loss: 0.1665 - val_accuracy: 0.9503
Epoch 8/10
4342/4342 [==============================] - 10s 2ms/step - loss: 0.0158 - accuracy: 0.9940 - val_loss: 0.1389 - val_accuracy: 0.9558
Epoch 9/10
4342/4342 [==============================] - 10s 2ms/step - loss: 0.0089 - accuracy: 0.9982 - val_loss: 0.1912 - val_accuracy: 0.9576
Epoch 10/10
4342/4342 [==============================] - 11s 2ms/step - loss: 0.0089 - accuracy: 0.9975 - val_loss: 0.2400 - val_accuracy: 0.9558
<keras.callbacks.callbacks.History object at 0x00000135D8E59008>
Testing the LSTM model
1358/1358 [==============================] - 1s 997us/step
Test accuracy: 0.9528718590736389
```

Figure 23: Accuracy of Haryana Assembly Polls using LSTM

```
In [120]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/lstmaccuracy.py', wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
Loading train and test data
Data loaded
Tokenizing and padding data
Tokenizing and padding complete
Creating the LSTM model
Model: "sequential_15"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_14 (Embedding)     (None, 23, 128)           256000
_________________________________________________________________
spatial_dropout1d_14 (Spatia (None, 23, 128)           0
_________________________________________________________________
lstm_14 (LSTM)               (None, 256)               394240
_________________________________________________________________
dense_14 (Dense)             (None, 2)                 514
=================================================================
Total params: 650,754
Trainable params: 650,754
Non-trainable params: 0
```

```
Training the LSTM model
C:\Users\Sahrish Zahoor\Anaconda3\envs\venv\lib\site-packages\tensorflow_core\python\framework\indexed_slices.py:433: UserWarning:
Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
Train on 1464 samples, validate on 367 samples
Epoch 1/10
1464/1464 [==============================] - 9s 6ms/step - loss: 0.6134 - accuracy: 0.6995 - val_loss: 0.5541 - val_accuracy: 0.6730
Epoch 2/10
1464/1464 [==============================] - 6s 4ms/step - loss: 0.4290 - accuracy: 0.7773 - val_loss: 0.3231 - val_accuracy: 0.8474
Epoch 3/10
1464/1464 [==============================] - 5s 4ms/step - loss: 0.2304 - accuracy: 0.9064 - val_loss: 0.2658 - val_accuracy: 0.8883
Epoch 4/10
1464/1464 [==============================] - 6s 4ms/step - loss: 0.1548 - accuracy: 0.9385 - val_loss: 0.2582 - val_accuracy: 0.8801
Epoch 5/10
1464/1464 [==============================] - 5s 4ms/step - loss: 0.0935 - accuracy: 0.9672 - val_loss: 0.2395 - val_accuracy: 0.8774
Epoch 6/10
1464/1464 [==============================] - 5s 4ms/step - loss: 0.0697 - accuracy: 0.9781 - val_loss: 0.2624 - val_accuracy: 0.8937
Epoch 7/10
1464/1464 [==============================] - 6s 4ms/step - loss: 0.0479 - accuracy: 0.9816 - val_loss: 0.2427 - val_accuracy: 0.8937
Epoch 8/10
1464/1464 [==============================] - 5s 3ms/step - loss: 0.0303 - accuracy: 0.9918 - val_loss: 0.2803 - val_accuracy: 0.8910
Epoch 9/10
1464/1464 [==============================] - 5s 3ms/step - loss: 0.0206 - accuracy: 0.9932 - val_loss: 0.2924 - val_accuracy: 0.9019
Epoch 10/10
1464/1464 [==============================] - 5s 3ms/step - loss: 0.0136 - accuracy: 0.9952 - val_loss: 0.3159 - val_accuracy: 0.9019
<keras.callbacks.callbacks.History object at 0x00000135C7AE1748>
Testing the LSTM model
458/458 [==============================] - 0s 988us/step
Test accuracy: 0.9279475808143616
```

Figure 24 Accuracy of BJP using LSTM

```
In [125]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/lstmaccuracy.py', wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
Loading train and test data
Data loaded
Tokenizing and padding data
Tokenizing and padding complete
Creating the LSTM model
Model: "sequential_16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_15 (Embedding)     (None, 19, 128)           256000
_________________________________________________________________
spatial_dropout1d_15 (Spatia (None, 19, 128)           0
_________________________________________________________________
lstm_15 (LSTM)               (None, 256)               394240
_________________________________________________________________
dense_15 (Dense)             (None, 2)                 514
=================================================================
Total params: 650,754
Trainable params: 650,754
Non-trainable params: 0
```

```
Training the LSTM model
C:\Users\Sahrish Zahoor\Anaconda3\envs\venv\lib\site-packages\tensorflow_core\python\framework\indexed_slices.py:433: UserWarning:
Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
Train on 419 samples, validate on 105 samples
Epoch 1/10
419/419 [==============================] - 5s 11ms/step - loss: 0.6685 - accuracy: 0.7064 - val_loss: 0.5635 - val_accuracy: 0.8381
Epoch 2/10
419/419 [==============================] - 2s 4ms/step - loss: 0.5019 - accuracy: 0.8043 - val_loss: 0.4917 - val_accuracy: 0.8381
Epoch 3/10
419/419 [==============================] - 1s 3ms/step - loss: 0.4631 - accuracy: 0.8043 - val_loss: 0.3951 - val_accuracy: 0.8857
Epoch 4/10
419/419 [==============================] - 1s 3ms/step - loss: 0.3969 - accuracy: 0.8807 - val_loss: 0.3440 - val_accuracy: 0.8857
Epoch 5/10
419/419 [==============================] - 1s 3ms/step - loss: 0.3072 - accuracy: 0.8950 - val_loss: 0.2201 - val_accuracy: 0.9238
Epoch 6/10
419/419 [==============================] - 1s 3ms/step - loss: 0.2158 - accuracy: 0.9045 - val_loss: 0.1606 - val_accuracy: 0.9333
Epoch 7/10
419/419 [==============================] - 1s 3ms/step - loss: 0.1556 - accuracy: 0.9332 - val_loss: 0.1911 - val_accuracy: 0.9048
Epoch 8/10
419/419 [==============================] - 1s 3ms/step - loss: 0.1133 - accuracy: 0.9618 - val_loss: 0.1229 - val_accuracy: 0.9524
Epoch 9/10
419/419 [==============================] - 1s 3ms/step - loss: 0.0825 - accuracy: 0.9690 - val_loss: 0.1083 - val_accuracy: 0.9714
Epoch 10/10
419/419 [==============================] - 1s 3ms/step - loss: 0.0625 - accuracy: 0.9833 - val_loss: 0.1016 - val_accuracy: 0.9714
<keras.callbacks.callbacks.History object at 0x00000135D5CBE5C8>
Testing the LSTM model
132/132 [==============================] - 0s 666us/step
Test accuracy: 0.8863636255264282
```

Figure 25: Accuracy of ML Khattar using LSTM

The Sky Is Pink

This movie starring Priyanka Chopra Jonas, Farhan Akhtar, Zaira Wasim and Rohit Suresh Saraf was released on 11 October. Around 8000 tweets for this particular movie were scraped using the hashtag #TheSkyIsPink. This movie gained popularity because it was considered to be a comeback movie of Priyanka and when hashtags related to this movie were scraped from twitter, it was seen that #PriyankaChopra was leading.

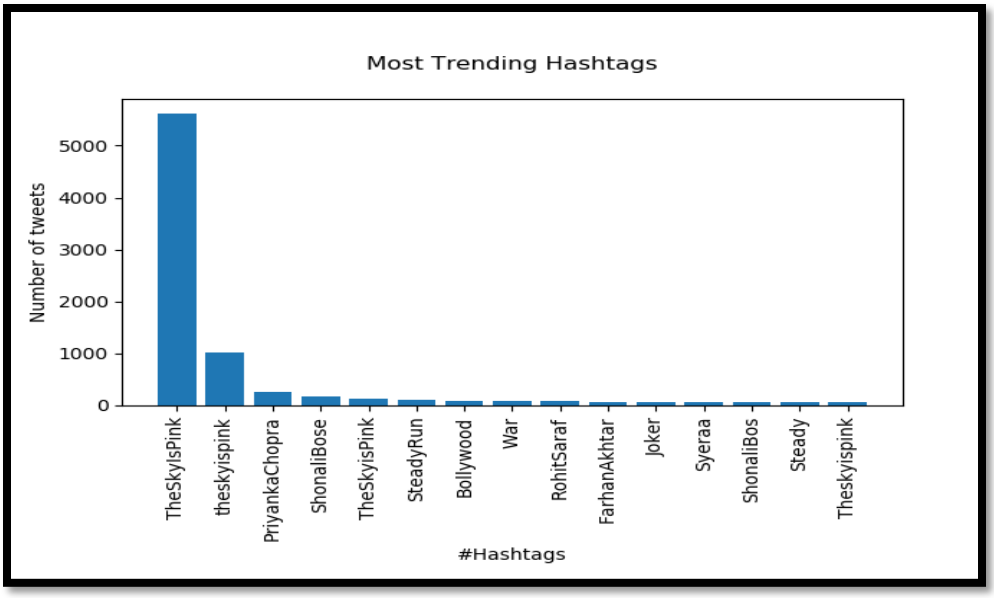


Figure 26: Trending Hashtags

The tweets scraped were analyzed using TextBlob and VADER to determine the sentiment and the overall positive sentiment detected using TextBlob was 64.1% and using VADER was 62.8%. thus the movie was received positively by the audience and it could be seen that the most common hashtag amongst all was that of #PriyankaChopra.

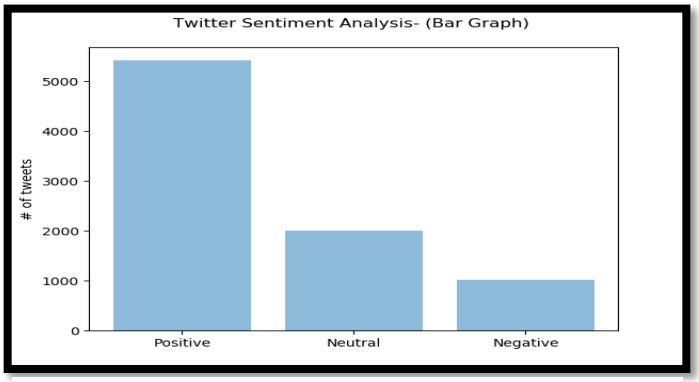
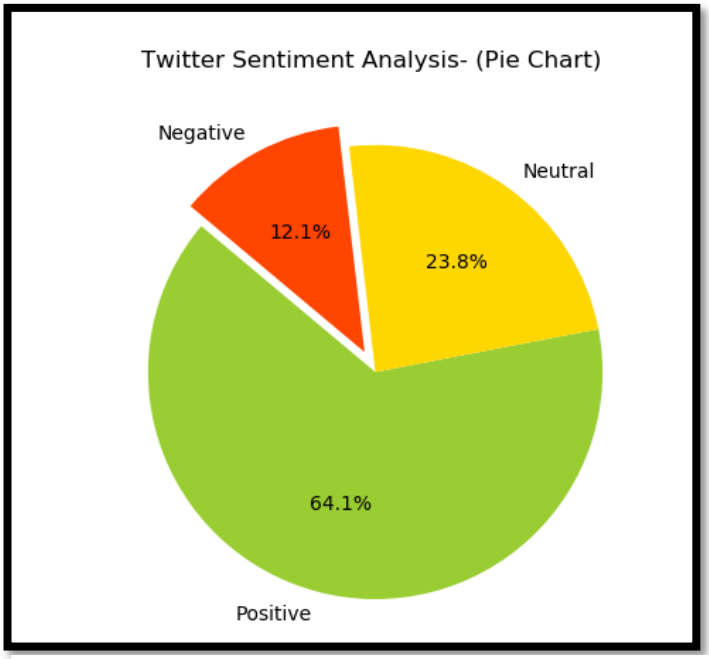


Figure 27: Bar Graph



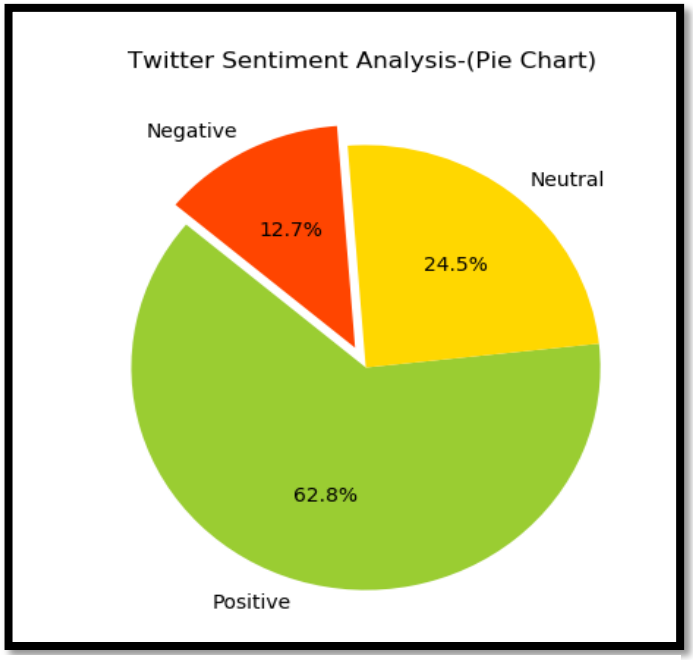


Figure 28: Sentiment Analysis of The Sky Is Pink using TextBlob and VADER

```
Performing Sentiment Analysis.....Total Tweets:  8447
Positive =  5418
Neutral=  2011
Negative=  1018

b'RT @yasiruvismini_: moose is ART \xf0\x9f\x8d\xa5 #theskyispink https://t.co/8fsgoEsAic'  =>
Sentiment(polarity=0.0, subjectivity=0.0)

b'RT @PritamFC: Weekend is here - Go to @ZeeMusicCompany YouTube channel and watch/Listen all
#TheSkyIsPink full song videos now \n\n#DilHiTohH\xe2\x80\xa6'  =>  Sentiment(polarity=0.35,
subjectivity=0.55)

b'RT @yasiruvismini_: I LOVE YOU 4000 @priyankachopra #theskyispink https://t.co/tMMuLv6ymx'  =>
Sentiment(polarity=0.5, subjectivity=0.6)

b'RT @jayantadas100: @PrabhasRaju #TheSkyIsPink maintain its Ranking (Highest Net Gross 2019).
@priyankachopra \xe2\xac\x86\xef\xb8\x8f'  =>  Sentiment(polarity=0.0, subjectivity=0.0)

b'RT @rayofsun28: \xf0\x9f\x91\x8f\xf0\x9f\x91\x8f\xf0\x9f\x91\x8f\xf0\x9f\x91\x8f can we know the
figure please...also \xf0\x9f\x96\x95to people who called it flop..#theskyispink #PriyankaChopra
https://t.co/J6ra0OA\xe2\x80\xa6'  =>  Sentiment(polarity=0.0, subjectivity=0.0)
```

Figure 29: Random tweets using TextBlob

```
Performing sentiment analysis...
.....total 8447
negative 1069
positive 5307
neutral 2071

b'RT @guptairesh: my little something on #TheSkyIsPink by Shonali Bose. @priyankachopra @FarOutAkhtar
@ZairaWasimmm @RSVPMovies @roykapurfilm\xe2\x80\xa6'=> {'neg': 0.079, 'neu': 0.558, 'pos': 0.363,
'compound': 0.9042}

b'RT @PRDMovieReviews: #TheSkyIsPink\xc2\xa0\xc2\xa0Week5 in a few screens!! 24cr+ at BO! #SteadyRun
\n\nWeek1 17.52cr\nWeek2 5.60cr\nWeek3 0.83cr\nWeek4 0.36cr\xe2\x80\xa6'=> {'neg': 0.079, 'neu':
0.558, 'pos': 0.363, 'compound': 0.9042}

b'RT @priyankachopra: Thank you so much for loving our film \xf0\x9f\x92\x95 #TheSkyIsPink \nIn
cinemas tomorrow!\n\n@FarOutAkhtar @ZairaWasimmm #RohitSaraf #S\xe2\x80\xa6'=> {'neg': 0.079, 'neu':
0.558, 'pos': 0.363, 'compound': 0.9042}

b'@priyankachopra @TeamPriyanka #TheSkyIsPink 4th week updated. Yes, this @priyankachopra  movie is
still running at\xe2\x80\xa6 https://t.co/a2ST4WNt7r'=> {'neg': 0.079, 'neu': 0.558, 'pos': 0.363,
'compound': 0.9042}

b'RT @shayPClove: @PRDMovieReviews @priyankachopra @FarOutAkhtar @ZairaWasimmm @QDCompagnia
@BagriFoundation @river2riverfiff @TeamPriyanka @\xe2\x80\xa6'=> {'neg': 0.079, 'neu': 0.558, 'pos':
0.363, 'compound': 0.9042}
```

Figure 30: Random tweets using VADER

Naïve Bayes

```
In [7]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/ActualNaiveBayes.py',
wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
0.9812024234892031
Naive Bayes Accuracy: 98.12
0.9

Naive Bayes Accuracy with the Test Set: 90.00
```

Figure 31: Accuracy of The Sky Is Pink using Naive Bayes

SVM

```
In [4]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/svmm.py', wdir='C:/
Users/Sahrish Zahoor/Desktop/Project code')
[nltk_data] Downloading package stopwords to C:\Users\Sahrish
[nltk_data]     Zahoor\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
positive    5418
negative    1018
Name: sentiment, dtype: int64
0.9829126213592233
```

Figure 32: Accuracy of The Sky Is Pink using SVM

Random Forest Classifier

```
In [96]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/
Randomforestclassifier.py', wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
Loading train and test data
Data loaded
Applying TF-IDF transformation
Training Random Forest Classifier
Predicting on train data
Training accuracy: 100.0%
Predicting on test data
Testing accuracy: 98.214285714285714%
```

Figure 33: Accuracy of The Sky Is Pink using Random Forest Classifier

LSTM

```
In [95]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/lstmaccuracy.py', wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
Loading train and test data
Data loaded
Tokenizing and padding data
Tokenizing and padding complete
Creating the LSTM model
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_9 (Embedding)      (None, 29, 128)           256000
_________________________________________________________________
spatial_dropout1d_9 (Spatial (None, 29, 128)           0
_________________________________________________________________
lstm_9 (LSTM)                (None, 256)               394240
_________________________________________________________________
dense_9 (Dense)              (None, 2)                 514
=================================================================
Total params: 650,754
Trainable params: 650,754
Non-trainable params: 0
```

```
Training the LSTM model
C:\Users\Sahrish Zahoor\Anaconda3\envs\venv\lib\site-packages\tensorflow_core\python\framework\indexed_slices.py:433: UserWarning:
Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
Train on 4118 samples, validate on 1030 samples
Epoch 1/10
4118/4118 [==============================] - 18s 4ms/step - loss: 0.4217 - accuracy: 0.8368 - val_loss: 0.2208 - val_accuracy: 0.8864
Epoch 2/10
4118/4118 [==============================] - 17s 4ms/step - loss: 0.1011 - accuracy: 0.9604 - val_loss: 0.0885 - val_accuracy: 0.9699
Epoch 3/10
4118/4118 [==============================] - 15s 4ms/step - loss: 0.0361 - accuracy: 0.9879 - val_loss: 0.0795 - val_accuracy: 0.9748
Epoch 4/10
4118/4118 [==============================] - 14s 3ms/step - loss: 0.0239 - accuracy: 0.9942 - val_loss: 0.0711 - val_accuracy: 0.9796
Epoch 5/10
4118/4118 [==============================] - 14s 3ms/step - loss: 0.0152 - accuracy: 0.9947 - val_loss: 0.0715 - val_accuracy: 0.9796
Epoch 6/10
4118/4118 [==============================] - 15s 4ms/step - loss: 0.0092 - accuracy: 0.9973 - val_loss: 0.0761 - val_accuracy: 0.9816
Epoch 7/10
4118/4118 [==============================] - 14s 3ms/step - loss: 0.0067 - accuracy: 0.9978 - val_loss: 0.0757 - val_accuracy: 0.9835
Epoch 8/10
4118/4118 [==============================] - 14s 4ms/step - loss: 0.0038 - accuracy: 0.9995 - val_loss: 0.0911 - val_accuracy: 0.9825
Epoch 9/10
4118/4118 [==============================] - 13s 3ms/step - loss: 0.0029 - accuracy: 0.9990 - val_loss: 0.1042 - val_accuracy: 0.9816
Epoch 10/10
4118/4118 [==============================] - 14s 3ms/step - loss: 0.0076 - accuracy: 0.9983 - val_loss: 0.0800 - val_accuracy: 0.9825
<keras.callbacks.callbacks.History object at 0x00000135E1B3B348>
Testing the LSTM model
1288/1288 [==============================] - 1s 922us/step
Test accuracy: 0.97826087474823
```

Figure 34: Accuracy of The Sky is Pink using LSTM

Delhi Odd-Even

Towards the end of year, Delhi- the capital of India goes through terrible pollution. The situation becomes so grim almost every year that schools are shut and people cannot leave their places without a mask on. This happens mostly because of the number of vehicles present on the roads and to control this, the government has devised a scheme from the past couple of years i.e. odd-even. This restricts the movement of cars on the roads; however it causes a lot of trouble as well. Not everyone among the public supports this decision as it restricts their movement and people are allowed to use their vehicles according to their number plates. This causes a lot trouble but according to the government plays a huge role in reducing the pollution levels in the state. Around 10,000 tweets related to the odd-even were scrapped during the month of November when it was imposed, the most common hashtags found were #AAPBJPchokeDelhi

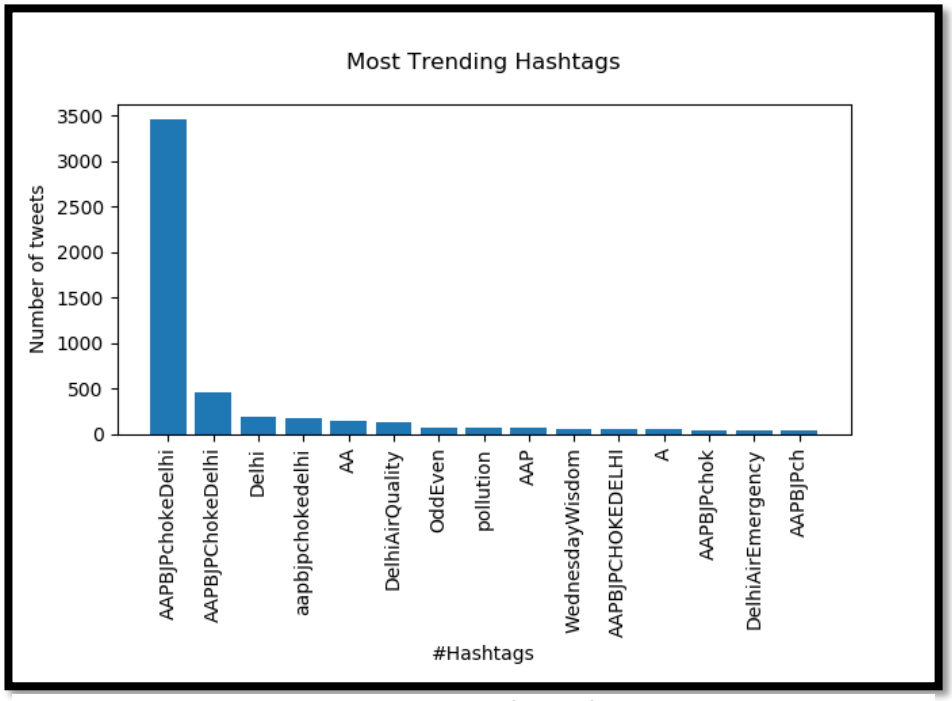


Figure 35: Trending Hashtags

The tweets collected were analyzed using VADER and TextBlob so as to determine the sentiment of people towards this rule of odd-even introduced by the government and other steps taken towards reducing the pollution in the capital. It was found out that people were immensely upset with the government and the measures taken by them to reduce the pollution as they were causing inconvenience to the common man of Delhi. Using TextBlob around 37.6% negative tweets were found and the negative tweets were comparatively higher than the positive and neutral tweets. Using VADER around 43.0% negative tweets were found and thus it could be concluded that the overall sentiment was negative as the highest number of tweets were categorized as negative.

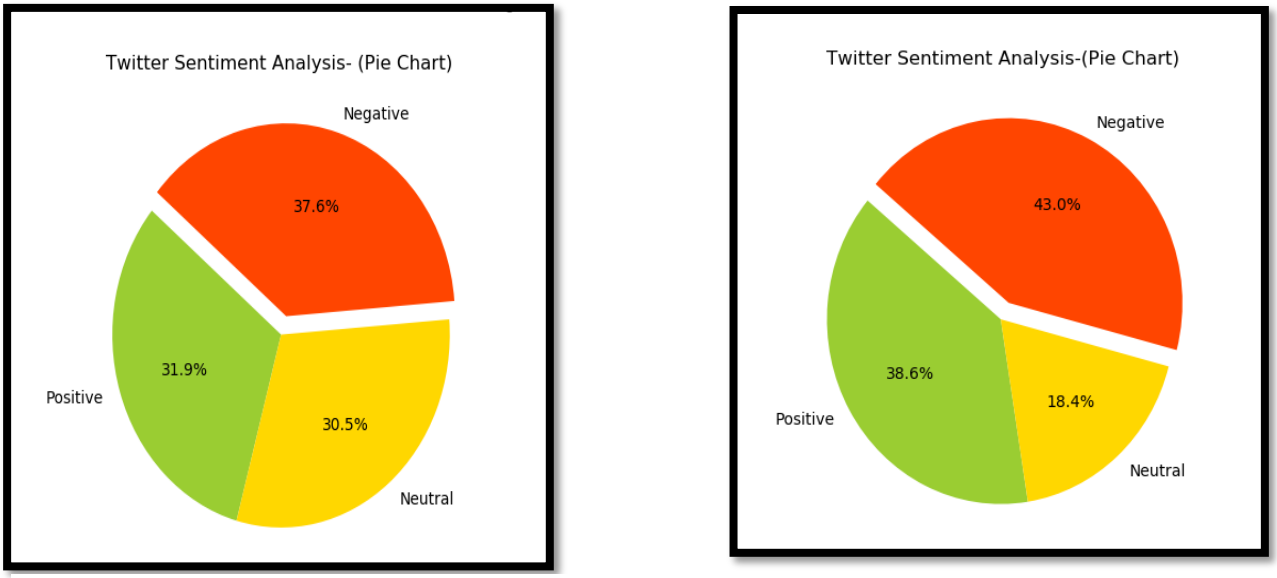


Figure 36: Sentiment analysis of Delhi-odd even using TextBlob and VADER

Some random tweets were analyzed using VADER and TextBlob to determine the sentiment intensity scores and also the polarity, subjectivity and compound score.

```
Performing Sentiment Analysis.....Total Tweets:  10865
Positive =  3461
Neutral=  3316
Negative=  4088

b'Fires in d last 1 month @capt_amarinder @CMOPb @INCPunjab @mlkhattar @cmohry @BJP4Haryana
@PMOIndia u &amp;amp; our belove\xe2\x80\xa6 https://t.co/3VTZLw0tUI'  =>  Sentiment(polarity=0.0,
subjectivity=0.06666666666666667)

b'RT @devendrayadvinc: Pollution breaks all previous records as failing to check &amp;amp; ensure ban on
demolition, construction &amp;amp; garbage burning,\xe2\x80\xa6'  =>
Sentiment(polarity=-0.16666666666666666, subjectivity=0.16666666666666666)

b'RT @devendrayadvinc: Delhi chokes from the apathy of Government while the BJP leaders advise people
to eat carrots and listen to music!\n#AA\xe2\x80\xa6'  =>  Sentiment(polarity=0.0, subjectivity=0.0)

b'RT @devendrayadvinc: Delhi chokes from the apathy of Government while the BJP leaders advise people
to eat carrots and listen to music!\n#AA\xe2\x80\xa6'  =>  Sentiment(polarity=0.0, subjectivity=0.0)

b'RT @INCIndia: Congress govt in Delhi ensured to increase the tree cover to reduce pollution which
now stands lower than it was in 2011. Ins\xe2\x80\xa6'  =>  Sentiment(polarity=0.0, subjectivity=0.0)
```

Figure 37: Random tweets using TextBlob

```
Performing sentiment analysis...
.....total 10865
negative 4672
positive 4195
neutral 1998

b'Please give a minute of time for below tweet.\nShe does not any money from anyone, just her own
money\xe2\x80\xa6 https://t.co/cnRkUFjuiS'=> {'neg': 0.172, 'neu': 0.682, 'pos': 0.146, 'compound':
-0.2023}

b'RT @kukk44: Prakash Javadekar was to help tackle Delhi\xe2\x80\x99s air pollution. He went
campaigning in Maharashtra instead.\n\nhttps://t.co/DpV28hOvR\xe2\x80\xa6'=> {'neg': 0.172, 'neu':
0.682, 'pos': 0.146, 'compound': -0.2023}

b'RT @devendrayadvinc: Pollution breaks all previous records as failing to check &amp;amp; ensure ban on
demolition, construction &amp;amp; garbage burning,\xe2\x80\xa6'=> {'neg': 0.172, 'neu': 0.682, 'pos':
0.146, 'compound': -0.2023}

b'RT @devendrayadvinc: Pollution breaks all previous records as failing to check &amp;amp; ensure ban on
demolition, construction &amp;amp; garbage burning,\xe2\x80\xa6'=> {'neg': 0.172, 'neu': 0.682, 'pos':
0.146, 'compound': -0.2023}

b'RT @PumpiNsui: Congress gave clean and green Delhi.\n\nMoAAP gave choked the Delhi with toxic smog
by playing with sowing cycle without study\xe2\x80\xa6'=> {'neg': 0.172, 'neu': 0.682, 'pos': 0.146,
'compound': -0.2023}
```

Figure 38: Random tweets using VADER

Naïve Bayes

```
In [6]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/ActualNaiveBayes.py',
wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
0.9952317880794702
Naive Bayes Accuracy: 99.52
0.988

Naive Bayes Accuracy with the Test Set: 98.80
```

Figure 39: Accuracy of Delhi-odd even using Naive Bayes

SVM

```
In [22]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/svmm.py', wdir='C:/
Users/Sahrish Zahoor/Desktop/Project code')
[nltk_data] Downloading package stopwords to C:\Users\Sahrish
[nltk_data]     Zahoor\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
negative    4088
positive    3461
Name: sentiment, dtype: int64
0.9956953642384105
```

Figure 40: Accuracy of Delhi-odd even using SVM

Random Forest Classifier

```
In [111]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/Randomforestclassifier.py',
wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
Loading train and test data
Data loaded
Applying TF-IDF transformation
Training Random Forest Classifier
Predicting on train data
Training accuracy: 99.98344096704753%
Predicting on test data
Testing accuracy: 99.73509933774835%
```

Figure 41: Accuracy of Delhi-odd even using Random Forest Classifier

LSTM

```
In [110]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/lstmaccuracy.py', wdir='C:/Users/Sahrish
Zahoor/Desktop/Project code')
Loading train and test data
Data loaded
Tokenizing and padding data
Tokenizing and padding complete
Creating the LSTM model
Model: "sequential_13"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_12 (Embedding)     (None, 20, 128)           256000
_________________________________________________________________
spatial_dropout1d_12 (Spatia (None, 20, 128)           0
_________________________________________________________________
lstm_12 (LSTM)               (None, 256)               394240
_________________________________________________________________
dense_12 (Dense)             (None, 2)                 514
=================================================================
Total params: 650,754
Trainable params: 650,754
Non-trainable params: 0
```

```
C:\Users\Sahrish Zahoor\Anaconda3\envs\venv\lib\site-packages\tensorflow_core\python\framework\indexed_slices.py:433: UserWarning:
Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
Train on 4831 samples, validate on 1208 samples
Epoch 1/10
4831/4831 [==============================] - 18s 4ms/step - loss: 0.2816 - accuracy: 0.8634 - val_loss: 0.0305 - val_accuracy: 0.9859
Epoch 2/10
4831/4831 [==============================] - 14s 3ms/step - loss: 0.0206 - accuracy: 0.9930 - val_loss: 0.0121 - val_accuracy: 0.9950
Epoch 3/10
4831/4831 [==============================] - 13s 3ms/step - loss: 0.0055 - accuracy: 0.9994 - val_loss: 0.0140 - val_accuracy: 0.9950
Epoch 4/10
4831/4831 [==============================] - 13s 3ms/step - loss: 0.0028 - accuracy: 0.9996 - val_loss: 0.0166 - val_accuracy: 0.9950
Epoch 5/10
4831/4831 [==============================] - 12s 2ms/step - loss: 7.2467e-04 - accuracy: 1.0000 - val_loss: 0.0200 - val_accuracy:
0.9950
Epoch 6/10
4831/4831 [==============================] - 12s 2ms/step - loss: 0.0020 - accuracy: 0.9998 - val_loss: 0.0178 - val_accuracy: 0.9950
Epoch 7/10
4831/4831 [==============================] - 12s 2ms/step - loss: 0.0023 - accuracy: 0.9994 - val_loss: 0.0173 - val_accuracy: 0.9959
Epoch 8/10
4831/4831 [==============================] - 11s 2ms/step - loss: 0.0015 - accuracy: 0.9998 - val_loss: 0.0157 - val_accuracy: 0.9950
Epoch 9/10
4831/4831 [==============================] - 11s 2ms/step - loss: 0.0013 - accuracy: 0.9998 - val_loss: 0.0162 - val_accuracy: 0.9959
Epoch 10/10
4831/4831 [==============================] - 11s 2ms/step - loss: 6.6622e-04 - accuracy: 0.9998 - val_loss: 0.0177 - val_accuracy:
0.9959
<keras.callbacks.callbacks.History object at 0x00000135E2880488>
Testing the LSTM model
1510/1510 [==============================] - 1s 639us/step
Test accuracy: 0.9940397143363953
```

Figure 42: Accuracy of Delhi-odd even using LSTM

HowdyModi

This was a gathering that took place in Houston, USA on September 22, 2019. This event was organized by the Indians settled in the USA for the PM Modi. This event became very popular on the social media with the hashtag #HowdyModi and was trending for a number of days. It was attended by a huge crowd and also the President of USA Donald Trump. It turned out to be a huge success even after being shadowed by the hurricane and various other political factors. In order to determine how people all over the world perceived this event, around 4000 tweets were scraped for this event and the most trending hashtag found was #HowdyModi. Also these tweets were analyzed using TextBlob and Vader to determine the exact sentiment of people.

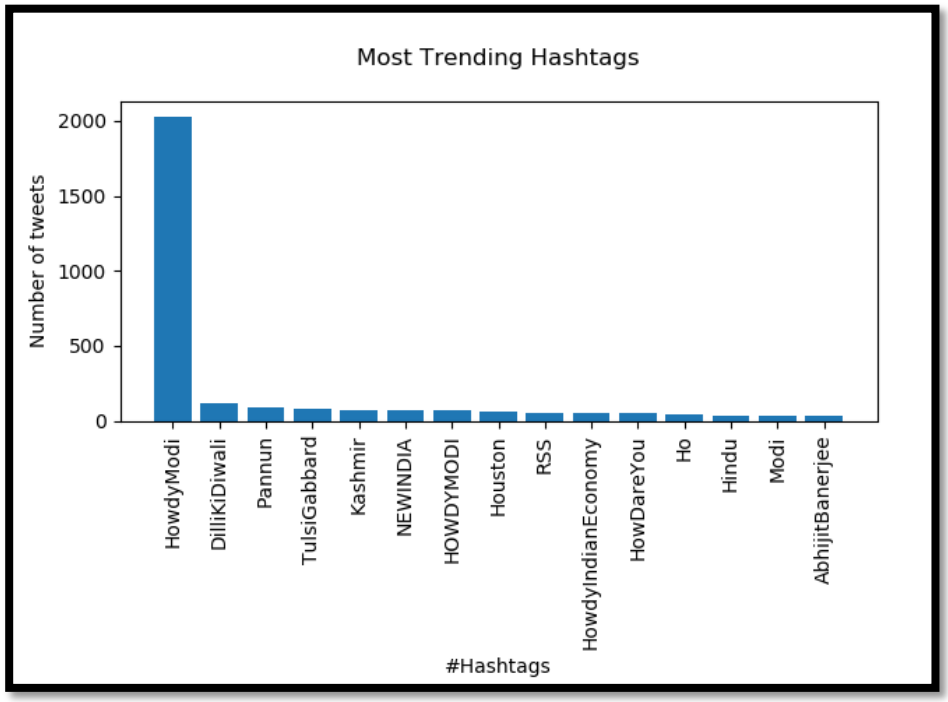
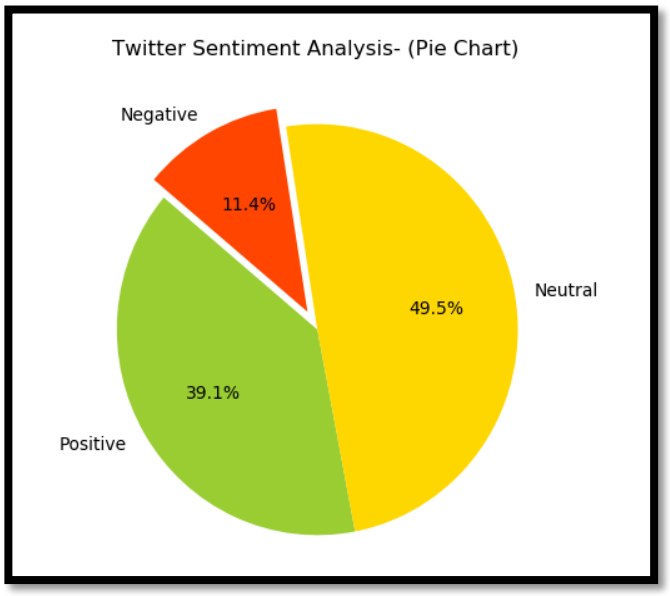
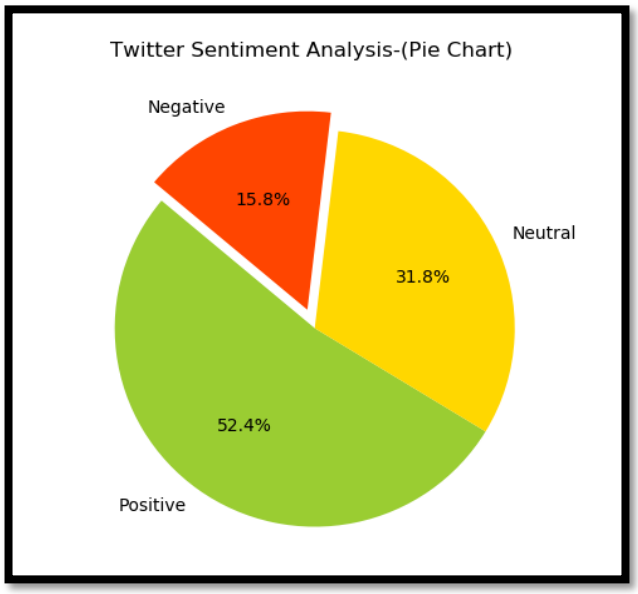


Figure 43: Trending Hashtags





Using VADER around 52.4% tweets were categorized as positive and 31.8% as neutral whereas using TextBlob around 39.1% tweets were categorized as positive and 49.5% as neutral. thus it could be concluded that the event received positive or neutral response and the overall negative response was minimum around 11.4% using TextBlob and 15.8% using VADER.

```
Performing Sentiment Analysis.....Total Tweets:  3931
Positive =  1538
Neutral=  1945
Negative=  448

b'RT @AltafHussain_90: Pakistan must stop Army actions against Mohajirs, Balochs, Pashtuns &amp;
oppressed nations. Speeches of President @realDo\xe2\x80\xa6'  =>  Sentiment(polarity=0.0,
subjectivity=0.0)

b'RT @dalitdiva: #Queer #Desis across the Country are also saying #AdiosModi. There is no
#Queerliberation in #HinduFascism. This why #Queer\xe2\x80\xa6'  =>  Sentiment(polarity=0.0,
subjectivity=0.0)

b'RT @Celesstrikes: To sumup #HowdyModi #ModiInHouston event in two lines\n\nTrump : Radical Islamic
Terrorism \nModi : Decisive War Against Ter\xe2\x80\xa6'  =>  Sentiment(polarity=0.0,
subjectivity=0.0)

b'RT @PMadhwaraj: But the effect of global recession of 2008 was better managed by Dr Manmohan Singh
Govt and the situation then was not as b\xe2\x80\xa6'  =>  Sentiment(polarity=0.25,
subjectivity=0.25)

b'RT @Iffidel: Amit : you forgot to add " Vote Trump in next election " so that we can get more
chances for organizing  #howdymodi kind of ev\xe2\x80\xa6'  =>
Sentiment(polarity=0.3666666666666667, subjectivity=0.4666666666666666)
```

Figure 44: Random tweets using TextBlob

```
Performing sentiment analysis...
.....total 3931
negative 620
positive 2060
neutral 1251

b'RT @i_theindian: NRIs who were chanting "Modi, Modi" and "Bharat Mata Ki Jai" at #HowdyModi event
should give up their US citizenship and r\xe2\x80\xa6'=> {'neg': 0.067, 'neu': 0.706, 'pos': 0.227,
'compound': 0.7351}

b"@Seek_Err @Chohan1954 @HarbirSinghSuri Imaginary K's? What kind of knowledge surfer are you?
\xf0\x9f\xa4\xa3 just last month jiha\xe2\x80\xa6 https://t.co/CpgYZRr8vQ"=> {'neg': 0.067, 'neu':
0.706, 'pos': 0.227, 'compound': 0.7351}

b"RT @SUD_0107: It's not #HowdyModi , it's for people of Delhi ...Yes it's #DilliKiDiwali . https://
t.co/pvUs1KeQhb"=> {'neg': 0.067, 'neu': 0.706, 'pos': 0.227, 'compound': 0.7351}

b'RT @hey_amish: PM MODI speech at Houston be like:\n\n#HowdyModi\n#ModiInHouston https://t.co/
Jo5nXNgF04'=> {'neg': 0.067, 'neu': 0.706, 'pos': 0.227, 'compound': 0.7351}
```

Figure 45: Random tweets using VADER

Naïve Bayes

```
In [9]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/ActualNaiveBayes.py',
wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
0.9879214896829391
Naive Bayes Accuracy: 98.79
0.98

Naive Bayes Accuracy with the Test Set: 98.00
```

Figure 46: Accuracy of Howdy Modi using Naïve Bayes

SVM

```
In [20]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/svmm.py', wdir='C:/
Users/Sahrish Zahoor/Desktop/Project code')
[nltk_data] Downloading package stopwords to C:\Users\Sahrish
[nltk_data]     Zahoor\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
positive    1538
negative     448
Name: sentiment, dtype: int64
0.969811320754717
```

Figure 47: Accuracy of Howdy Modi using SVM

Random Forest Classifier

```
In [101]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/Randomforestclassifier.py', wdir='C:/Users/Sahrish
Zahoor/Desktop/Project code')
Loading train and test data
Data loaded
Applying TF-IDF transformation
Training Random Forest Classifier
Predicting on train data
Training accuracy: 99.87405541561712%
Predicting on test data
Testing accuracy: 98.24120603015075%
```

Figure 48: Accuracy of Howdy Modi using Random Forest Classifier

LSTM

```
In [100]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/lstmaccuracy.py', wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
Loading train and test data
Data loaded
Tokenizing and padding data
Tokenizing and padding complete
Creating the LSTM model
Model: "sequential_11"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 19, 128)           256000
_________________________________________________________________
spatial_dropout1d_10 (Spatia (None, 19, 128)           0
_________________________________________________________________
lstm_10 (LSTM)               (None, 256)               394240
_________________________________________________________________
dense_10 (Dense)             (None, 2)                 514
=================================================================
Total params: 650,754
Trainable params: 650,754
Non-trainable params: 0
```

```
Training the LSTM model
C:\Users\Sahrish Zahoor\Anaconda3\envs\venv\lib\site-packages\tensorflow_core\python\framework\indexed_slices.py:433: UserWarning:
Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
Train on 1270 samples, validate on 318 samples
Epoch 1/10
1270/1270 [==============================] - 7s 6ms/step - loss: 0.5898 - accuracy: 0.7417 - val_loss: 0.4904 - val_accuracy: 0.7673
Epoch 2/10
1270/1270 [==============================] - 4s 3ms/step - loss: 0.4190 - accuracy: 0.7874 - val_loss: 0.3307 - val_accuracy: 0.8459
Epoch 3/10
1270/1270 [==============================] - 3s 3ms/step - loss: 0.2007 - accuracy: 0.9087 - val_loss: 0.1402 - val_accuracy: 0.9497
Epoch 4/10
1270/1270 [==============================] - 3s 2ms/step - loss: 0.0998 - accuracy: 0.9819 - val_loss: 0.1339 - val_accuracy: 0.9245
Epoch 5/10
1270/1270 [==============================] - 3s 2ms/step - loss: 0.0543 - accuracy: 0.9858 - val_loss: 0.1069 - val_accuracy: 0.9654
Epoch 6/10
1270/1270 [==============================] - 3s 2ms/step - loss: 0.0351 - accuracy: 0.9906 - val_loss: 0.0869 - val_accuracy: 0.9686
Epoch 7/10
1270/1270 [==============================] - 3s 2ms/step - loss: 0.0226 - accuracy: 0.9929 - val_loss: 0.0783 - val_accuracy: 0.9686
Epoch 8/10
1270/1270 [==============================] - 3s 2ms/step - loss: 0.0158 - accuracy: 0.9945 - val_loss: 0.0848 - val_accuracy: 0.9654
Epoch 9/10
1270/1270 [==============================] - 3s 2ms/step - loss: 0.0104 - accuracy: 0.9961 - val_loss: 0.0785 - val_accuracy: 0.9654
Epoch 10/10
1270/1270 [==============================] - 3s 2ms/step - loss: 0.0091 - accuracy: 0.9961 - val_loss: 0.0790 - val_accuracy: 0.9686
<keras.callbacks.callbacks.History object at 0x00000135DC4D8C08>
Testing the LSTM model
398/398 [==============================] - 0s 653us/step
Test accuracy: 0.9597989916801453
```

Figure 49: Accuracy of Howdy Modi using LSTM

UNGA

74th session of United Nations General Assembly started from 17th September 2019 and concluded on 24th September. Many political leaders delivered speeches addressing a number of issues faced by the world presently like the economic slowdown, poverty, climate change, terrorism and much more. Many political issues were also brought to the table like that of Kashmir by the PM of Pakistan. Issues like sanitation and climate change were discussed by the PM of India. Many such issues of utmost importance were discussed by the world leaders. The trending hashtag was #UNGA. Around 10,000 tweets were collected and analyzed using the in-built libraries in order to determine the positive, negative or neutral sentiment of people towards this assembly.

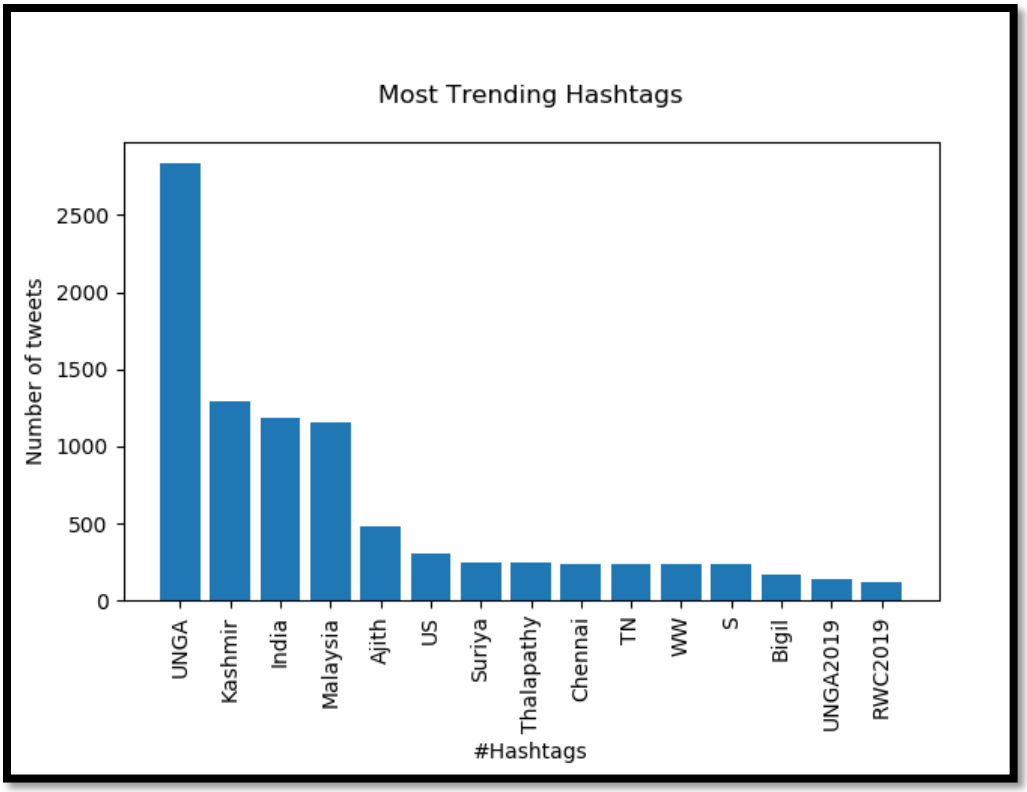
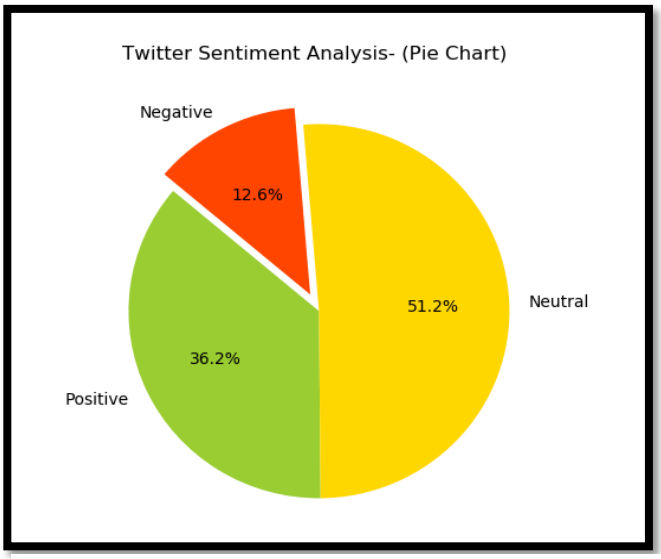


Figure 50: Trending Hashtags

It was analyzed that using TextBlob the around 12.6% tweets were categorized as negative whereas using VADER around 40.1% tweets were categorized as negative. This case confirms the fact that the social media language somewhat affects the exact analysis, this is because of the use of abbreviations and slang in the social media text.



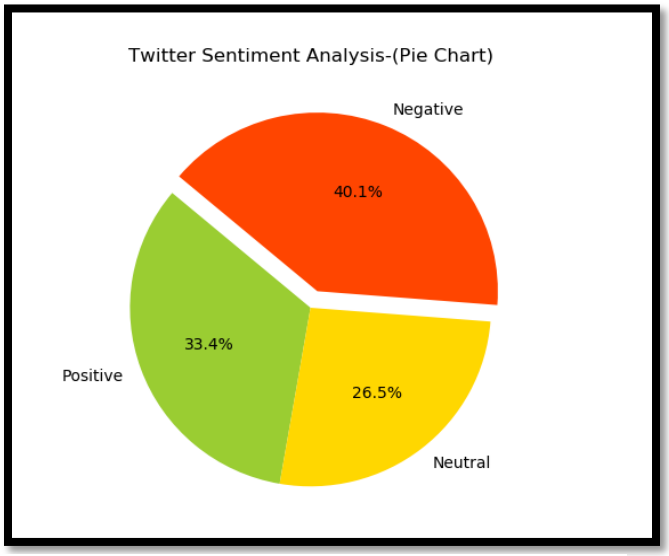


Figure 51: Sentiment analysis of UNGA using TextBlob and VADER

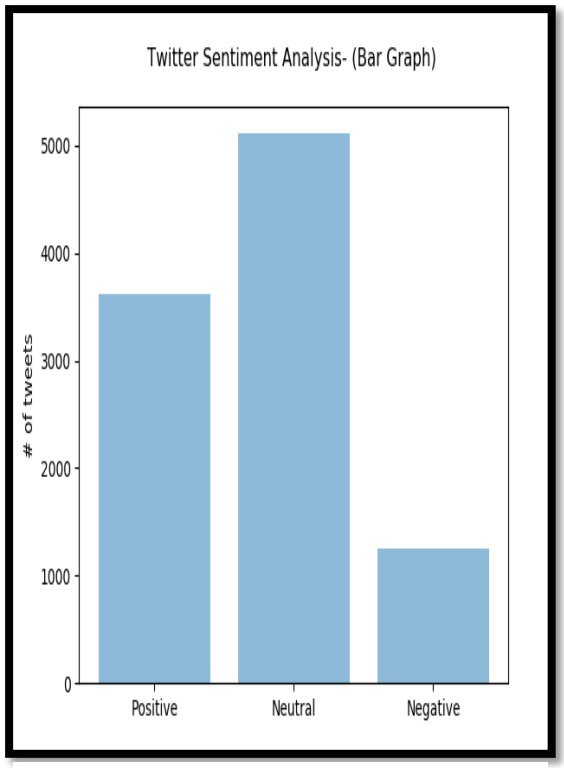


Figure 52: Bar Graph

```
Performing Sentiment Analysis.....Total Tweets:  9977
Positive =  3616
Neutral=  5107
Negative=  1254

b'RT @epwing_official: PM Imran Khan will hold several bilateral meetings with
his counterparts from various regions and participate in high-\xe2\x80\xa6'  =>
Sentiment(polarity=0.0, subjectivity=0.25)

b'RT @MFAestonia: Estonia is preparing to become an elected member of UN
Security Council in 2020-21. We want our whole society to be include
\xe2\x80\xa6'  =>  Sentiment(polarity=0.2, subjectivity=0.4)

b'@NLatUN @unGa @SwedenUN @USUN @GermanyUN @UKUN_NewYork @RussiaUN @ItalyUN_NY
@washingtonpost @cnni @BBCWorld @RFI\xe2\x80\xa6 https://t.co/tumDnnVJUP'  =>
Sentiment(polarity=0.0, subjectivity=0.0)

b'RT @epwing_official: PM Imran Khan to emphasize centrality of the Jammu and
Kashmir dispute through myriad engagements.\n#UNGA #PMIKatUNGA #\xe2\x80\xa6'
=>  Sentiment(polarity=0.0, subjectivity=0.0)

b'RT @RobertAlai: @DCI_Kenya Test and measure. Maybe you are destroying unga
ngano.'  =>  Sentiment(polarity=-0.2, subjectivity=0.0)
```

Figure 53: Random tweets using TextBlob

```
Performing sentiment analysis...
.....total 9977
negative 4003
positive 3330
neutral 2644

b'RT @HonOscarSudi: What happened to our electoral laws?... Why is @IEBCKenya and @NPSC_KE not
responsible?... Imran Okoth is donating Unga p\xe2\x80\xa6'=> {'neg': 0.274, 'neu': 0.726, 'pos':
0.0, 'compound': -0.5256}

b'RT @zlj517: Translation: 66:23! The result of a discussion on human rights in 3rd Committee of
UNGA. 23 countries led by US tried to blame\xe2\x80\xa6'=> {'neg': 0.274, 'neu': 0.726, 'pos': 0.0,
'compound': -0.5256}

b'RT @HonOscarSudi: What happened to our electoral laws?... Why is @IEBCKenya and @NPSC_KE not
responsible?... Imran Okoth is donating Unga p\xe2\x80\xa6'=> {'neg': 0.274, 'neu': 0.726, 'pos':
0.0, 'compound': -0.5256}

b'@Meneer_Mann @IRR_SouthAfrica @Lesufi @Lesufi unga ba thathi kancane laba, the last time
@Meneer_Mann and his\xe2\x80\xa6 https://t.co/qWjwJxUn7I'=> {'neg': 0.274, 'neu': 0.726, 'pos': 0.0,
'compound': -0.5256}

b'@iam_arjundas In Kaithi everyone acted well. Unga performance was mindblowing. \xf0\x9f\x91\x8f
\xf0\x9f\x91\x8f'=> {'neg': 0.274, 'neu': 0.726, 'pos': 0.0, 'compound': -0.5256}
```

Figure 54: Random tweets using VADER

Naïve Bayes

```
In [8]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/ActualNaiveBayes.py',
wdir='C:/Users/Sahrish Zahoor/Desktop/Project code')
0.9833709710531718
Naive Bayes Accuracy: 98.34
0.98

Naive Bayes Accuracy with the Test Set: 98.00
```

Figure 55: Accuracy of UNGA using Naive Bayes

SVM

```
In [17]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/svmm.py', wdir='C:/
Users/Sahrish Zahoor/Desktop/Project code')
[nltk_data] Downloading package stopwords to C:\Users\Sahrish
[nltk_data]     Zahoor\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
positive    3616
negative    1254
Name: sentiment, dtype: int64
0.9419917864476386
```

Figure 56:Accuracy of UNGA using SVM

Random Forest Classifier

```
In [106]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/Randomforestclassifier.py', wdir='C:/Users/Sahrish Zahoor/Desktop/
Project code')
Loading train and test data
Data loaded
Applying TF-IDF transformation
Training Random Forest Classifier
Predicting on train data
Training accuracy: 99.871663244353195
Predicting on test data
Testing accuracy: 94.66119096509245
```

Figure 57: Accuracy of UNGA using Random Forest Classifier

LSTM

```
In [105]: runfile('C:/Users/Sahrish Zahoor/Desktop/Project code/lstmaccuracy.py', wdir='C:/Users/Sahrish Zahoor/
Desktop/Project code')
Loading train and test data
Data loaded
Tokenizing and padding data
Tokenizing and padding complete
Creating the LSTM model
Model: "sequential_12"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_11 (Embedding)     (None, 20, 128)           256000
_________________________________________________________________
spatial_dropout1d_11 (Spatia (None, 20, 128)           0
_________________________________________________________________
lstm_11 (LSTM)               (None, 256)               394240
_________________________________________________________________
dense_11 (Dense)             (None, 2)                 514
=================================================================
Total params: 650,754
Trainable params: 650,754
Non-trainable params: 0
_________________________________________________________________
```

```
Training the LSTM model
C:\Users\Sahrish Zahoor\Anaconda3\envs\venv\lib\site-packages\tensorflow_core\python\framework\indexed_slices.py:433: UserWarning:
Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
Train on 3116 samples, validate on 780 samples
Epoch 1/10
3116/3116 [==============================] - 11s 4ms/step - loss: 0.5210 - accuracy: 0.7593 - val_loss: 0.3614 - val_accuracy: 0.8154
Epoch 2/10
3116/3116 [==============================] - 9s 3ms/step - loss: 0.2202 - accuracy: 0.9050 - val_loss: 0.1742 - val_accuracy: 0.9192
Epoch 3/10
3116/3116 [==============================] - 9s 3ms/step - loss: 0.1118 - accuracy: 0.9551 - val_loss: 0.1485 - val_accuracy: 0.9333
Epoch 4/10
3116/3116 [==============================] - 9s 3ms/step - loss: 0.0698 - accuracy: 0.9740 - val_loss: 0.1395 - val_accuracy: 0.9410
Epoch 5/10
3116/3116 [==============================] - 9s 3ms/step - loss: 0.0407 - accuracy: 0.9859 - val_loss: 0.1724 - val_accuracy: 0.9385
Epoch 6/10
3116/3116 [==============================] - 9s 3ms/step - loss: 0.0278 - accuracy: 0.9913 - val_loss: 0.1699 - val_accuracy: 0.9474
Epoch 7/10
3116/3116 [==============================] - 7s 2ms/step - loss: 0.0229 - accuracy: 0.9933 - val_loss: 0.1544 - val_accuracy: 0.9372
Epoch 8/10
3116/3116 [==============================] - 7s 2ms/step - loss: 0.0188 - accuracy: 0.9933 - val_loss: 0.1939 - val_accuracy: 0.9449
Epoch 9/10
3116/3116 [==============================] - 8s 3ms/step - loss: 0.0120 - accuracy: 0.9978 - val_loss: 0.1812 - val_accuracy: 0.9449
Epoch 10/10
3116/3116 [==============================] - 8s 2ms/step - loss: 0.0073 - accuracy: 0.9984 - val_loss: 0.1924 - val_accuracy: 0.9436
<keras.callbacks.callbacks.History object at 0x00000135E1E3C7C8>
Testing the LSTM model
974/974 [==============================] - 1s 802us/step
Test accuracy: 0.9404517412185669
```

Figure 58: Accuracy of UNGA using LSTM

CHAPTER 6

RESULT

The results obtained using the two in-built libraries are tabulated as:

Haryana Assembly Polls:

IN-BUILT LIBRARY	POSITIVE	NEGATIVE	NEUTRAL
TEXTBLOB	29.7%	12.0%	58.3%
VADER	44.0%	17.6%	38.4%

BJP:

IN-BUILT LIBRARY	POSITIVE	NEGATIVE	NEUTRAL
TEXTBLOB	39.9%	16.7%	43.4%
VADER	43.6%	34.3%	22.1%

ML Khattar:

IN-BUILT LIBRARY	POSITIVE	NEGATIVE	NEUTRAL
TEXTBLOB	58.5%	9.6%	32.0%
VADER	62.3%	17.2%	20.5%

The Sky Is Pink:

IN-BUILT LIBRARY	POSITIVE	NEGATIVE	NEUTRAL

TEXTBLOB	64.1%	12.1%	23.8%
VADER	62.8%	12.7%	24.5%

UNGA:

IN-BUILT LIBRARY	POSITIVE	NEGATIVE	NEUTRAL
TEXTBLOB	36.2%	12.6%	51.2%
VADER	33.4%	40.1%	26.5%

Delhi Odd-Even:

IN-BUILT LIBRARY	POSITIVE	NEGATIVE	NEUTRAL
TEXTBLOB	31.9%	37.6%	30.5%
VADER	38.6%	43.0%	18.4%

HowdyModi:

IN-BUILT LIBRARY	POSITIVE	NEGATIVE	NEUTRAL
TEXTBLOB	39.1%	11.4%	49.5%
VADER	52.4%	15.8%	31.8%

The results obtained using the Machine Learning Algorithms:

Machine Learning Algorithm	Haryana Assembly Polls	BJP	ML Khattar	The Sky is Pink	UNGA	Howdy Modi	Delhi odd-even
Naïve Bayes	96.8%	96%	98.4%	90%	98%	98%	98%
SVM	96.0%	89%	92.39%	98.29%	94.19%	96.8%	99.5%
Random Forest Classifier	95.94%	90.66%	88.63%	98.2%	94.66%	98.24%	99.7%
LSTM	95.28%	92.7%	88.36%	97.8%	94%	95.97%	99.4%

CHAPTER 7

Conclusion and Future scope

It can be concluded that the results and the accuracies obtained using the lexical or rule based approach (unsupervised learning) and the supervised approach could be further improvised by changing the parameters of that particular algorithm. However, it could be noted that these four algorithms that have been selected mostly give out the same result. Only in a few cases the results seem to be derailed and that could easily be attributed to the kind of language, emoji's and words that have been used in the tweets. Supervised Machine learning algorithms like Naïve Bayes, SVM train give a more exact and accurate result. Once the data is structured, it is divided into training and test set and then fed to the model for the classification. The supervised method is considered to be more specialized as compared to the unsupervised methods which make use of the in-built libraries.

References

[1] C.J. Hutto Eric Gilbert, VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text
[2] Edosomwan, Simeon, et al. "The history of social media and its impact on business." Journal of Applied Management and entrepreneurship 16.3 (2011): 79-91.
[3] https://www.oberlo.in/blog/twitter-statistics
[4] https://textblob.readthedocs.io/en/dev/
[5] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10) (N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, eds.), (Valletta, Malta), European Language Resources Association (ELRA), may 2010.
[6] https://en.wikipedia.org/wiki/2019_Haryana_Legislative_Assembly_election
[7] https://www.geeksforgeeks.org/twitter-sentiment-analysis-usingpython." Twitter Sentimental analysis for Realdonaldtrump" Devaki P, Ilakiya, J, Indumathi, R and Arul Priya, M
[8] Jansen,B.J.; Zhang,M.; Sobel,K.; and Chowdury,A. (2009), "Twitterpower: Tweets as electronic word of mouth", Journal of the American Society for Information Science and Technology 60(11):2169–2188.
[9] Pang, B., and Lee, L. (2008), "Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval" 2(1-2):1– 135.
[10]https://pdfs.semanticscholar.org/1b4d/954895c98be43528e9bf3df81a0db84ead14.pdf?_ga=2.254045646.736775005.1574603568-285830699.1554124917
[11] . T. T. P. Souza, O. Kolchyna, P. C. Treleaven, and T. Aste, "Twitter sentiment analysis applied to finance: A case study in the retail industry," 2015
[12] Sentiment Analysis Using Naïve Bayes Classifier Kavya Suppala, Narasinga Rao IJITEE 2019
[13] Sentiment Analysis Using Support Vector Machine Nurulhuda Zainuddin, Ali Selamat IEEE 2014
[14] Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20(3):273-297, 1995.
[15] Sentiment Mining of movie reviews using Random Forest using tuned hyperparameters Hitesh Parmar, Sanjay Bhandari, Glory Shah.
[16] Edosomwan, Simeon, et al. "The history of social media and its impact on business." Journal of Applied Management and entrepreneurship 16.3 (2011): 79-91.
[17] Hearst M A (1992). Direction-based text interpretation as an information access refinement. In: Jacobs P, ed., Text-based intelligent systems: current research and practice in information extraction and retrieval. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, pp. 257–274.
[18] Kessler B, Nunberg G and Sch¨utze H (1997). Automatic detection of text genre. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Somerset, New Jersey, pp. 32–38.
[19] Polanyi L and Zaenen A (2006). Contextual valence shifters. In: Shanahan J, Qu Y and Wiebe J, eds., Computing attitude and affect in text: Theory and applications. Springer, pp. 1–10.
[20] Pedersen T (2001). A decision tree of bigrams is an accurate predictor of word sense. In: Proceedings of the Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics. pp. 79–86.

[21] Boiy, E., Moens, M. A machine learning approach to sentiment analysis in multilingual Web texts. Inf Retrieval 12, 526–558 (2009).
[22] Tong S and Koller D (2002). Support vector machine active learning with applications to text classification. Journal of Machine Learning Research, 2:45–66.
[23] Sentiment Analysis Using Naïve Bayes Classifier Kavya Suppala, Narasinga Rao IJITEE 2019