# Analysis of Political Sentiments on Twitter using Deep Learning

MAJOR PROJECT REPORT

Submitted in in partial fulfilment for the award of the degree

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER ENGINEERING**


**UNDER THE SUPERVISION OF:**

Mr. Waseem Ahmed

(ASSISTANT PROFESSOR)

**SUBMITTED BY:**

Mohd Bilal Aziz (15BCS0024)

Himanshu Mehra(15BCS0020)

Karan Pratap Singh(15BCS0021)

**DEPARTMENT OF COMPUTER ENGINEERING**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**JAMIA MILLIA ISLAMIA (Year 2018-19)**

# CERTIFICATE

This is to certify that project entitled '**Analysis of Political Sentiments on Twitter using Deep Learning**' by Mohd Bilal Aziz (15BCS0024), Himanshu Mehra (15BCS0020) and Karan Pratap Singh (15BCS0021) is a record of bonafide work carried out by them, in the Department of Computer Engineering, Jamia Millia Islamia, New Delhi, under my supervision and guidance in partial fulfilment of requirement for the award of Bachelor in Technology in Computer Engineering, Jamia Millia Islamia in academic year 2018-19.


**Prof. Tanvir Ahmad**

**Head of Department**

Department of Computer Engineering

Faculty of Engineering and Technology

Jamia Millia Islamia

New Delhi

Session: 2018-19

**Mr. Waseem Ahmed**

**Assistant Professor**

Department of Computer Engineering

Faculty of Engineering and Technology

Jamia Millia Islamia

New Delhi

Session: 2018-19

# Acknowledgement

A very sincere and honest acknowledgement to **Mr. Waseem Ahmed,** Assistant Professor, Department of Computer Engineering, Jamia Millia Islamia, New Delhi for his valuable technical guidance and support, great innovative ideas and overwhelming moral support. We are also very grateful to the Head of Department of Computer Engineering **Prof. Tanvir Ahmad** for his valuable support throughout the project and also our gratitude to the Department of Computer Engineering and entire faculty members, for their teaching, guidance and encouragement.

A special thanks to our parents for their guidance and support in this wonderful journey.

We are also thankful to our classmates and friends for their valuable suggestions and support whenever required.

We regret any inadvertent omissions.

**Mohd Bilal Aziz** **Himanshu Mehra** **Karan Pratap Singh**

**(15BCS0024)** **(15BCS0020)** **(15BCS0021)**

**DEPARTMENT OF COMPUTER ENGINEERING**

**FACULTY OF ENGINEERING AND TECHOLOGY**

**JAMIA MILLIA ISLAMIA**

**NEW DELHI**

# ABSTRACT

In this project, we discuss a variety of issues related to opinion mining from micro posts, and the challenges they impose on an NLP system, along with an example application we have developed to determine political learnings from a set of tweets. While there are several sentiments analysis tools available which summarize positive, negative and neutral tweets about a given keyword or topic, these tools generally produce poor results, and operate in a simplistic way, using only the presence of certain positive and negative adjectives as indicators, or simple learning techniques which do not work well on short micro posts. Our methods make use of a variety of sophisticated **NLP** techniques in order to extract more meaningful and higher quality opinions and incorporate extra-linguistic contextual information. Our Study analyzes a set of approx. 4000 tweets to build a predictive model to infer the result of the 2019 Lok Sabha elections in India. We compare the performance of various classifiers such as SVM, ANN, LSTM, etc. coupled with the Count Vectorizer, TFIDF and Word2Vec feature selection technique to predict a post's sentiment. We find that a LSTM classifier coupled with the TFIDF technique can on average predict a post's sentiment 72.14% of the time.

# Contents

# 1. Introduction

## 1.1 About the project

Twitter is a popular website used for social networking and microblogging where users post messages or *tweets* that are 280 characters long. As of 2018, Twitter has more than 321 million monthly active users, of which 34.4 million were from India. As a result of this, posts are on a variety of topics, with news being one of the major ones. In 2016, Twitter was the largest source of breaking news during the Presidential Elections.

Twitter has had a huge impact in the political sphere in India. The success of a leader has always depended heavily on their ability to communicate with the masses. In recent times, this statement still holds true. All that's changed is the media used for communication. This phenomenon is very visible in the 2014 Lok Sabha elections. Though the number of twitter users compared to the total population is very small, many educated men, politicians, actors and other celebrities are a part of these numbers. These 'influencers' are capable of affecting the sentiments and opinions of their followers based on their tweets. Tides turn based on these sentiments.

The ability to judge sentiment would be helpful when applied to the vast number of opinions found in the growing number of news websites, tweets etc. It would allow for organization of information into groups and make it easier for users to find and react to similar or opposite opinions thus improving and simplifying the process of sharing and discussing opinions.

In this paper we investigate the different political leanings a user can have in relation to Indian electoral parties of 2019. We've harvested 3,896 tweets manually using *Tweepy* API and sorted them on the basis of 8 classifiers. Keeping in mind that time is an important factor, we've harvested tweets from a period of 3 months; from Jan 2019 – March 2019, because this is when sentiments and opinions will be at an all time high, as it is the time just preceding the 2019 Lok Sabha elections, which are held in April. However, it must be drawn to attention that the dataset used is extremely small when compared to the population size of the country and hence our result may not be completely indicative of the final outcome. We investigate the use of different machine learning techniques such Artificial Neural Network and Support Vector Machines. We wish to determine if such techniques can be applied in our domain problem.

We consider a vote for a party to be the best indicator of political preference of users. Given a user and a list of parties P, with $P_i \in P$, $P_0$ stands for the Bhartiya Janta Party (BJP), $P_1$ stands for Indian National Congress (INC), $P_2$ stands for Other parties.

## 1.2 Challenges

- **Sarcasm:** is one of the most difficult sentiments for automated tracking to interpret properly. Example: “It was awesome for the week that it worked.”
- **Navel gazing:** is when social media tracking turns up items related to your own promotional efforts and should be filtered out.
- **Neutral sentiment:** is like the concept of swing voters, and Frank recommended dividing it into specific themes to uncover more detailed opinions.
- **Relative sentiment:** is not a classic negative, but can be a negative, nonetheless. Example: “I bought an iPhone” is good for Apple, but not for Nokia.
- **Compound or multidimensional sentiment:** contain positives and negatives in the same phrase. Example: “I love *Mad Men* but hate the misleading episode trailers.”
- **Conditional sentiment:** includes actions that may happen in the future. Example: the customer isn’t angry now but says he will be if the company doesn’t call him back.
- **Positive feelings can be unrelated to the core issue.** For example, many comments about actors focus on their personal lives, not their acting skills.
- **Negative sentiment is not necessarily bad:** This relates to the classic PR dilemma regarding negative publicity. Example: Sarah Palin’s appearance on the *Today* show generated many negative comments but still drove ratings increases.
- **Ambiguous negative words:** Their context needs to be thoroughly understood and tagged accordingly. Example: “That backflip was so sick” is really a positive statement.

## 1.3 Tools used

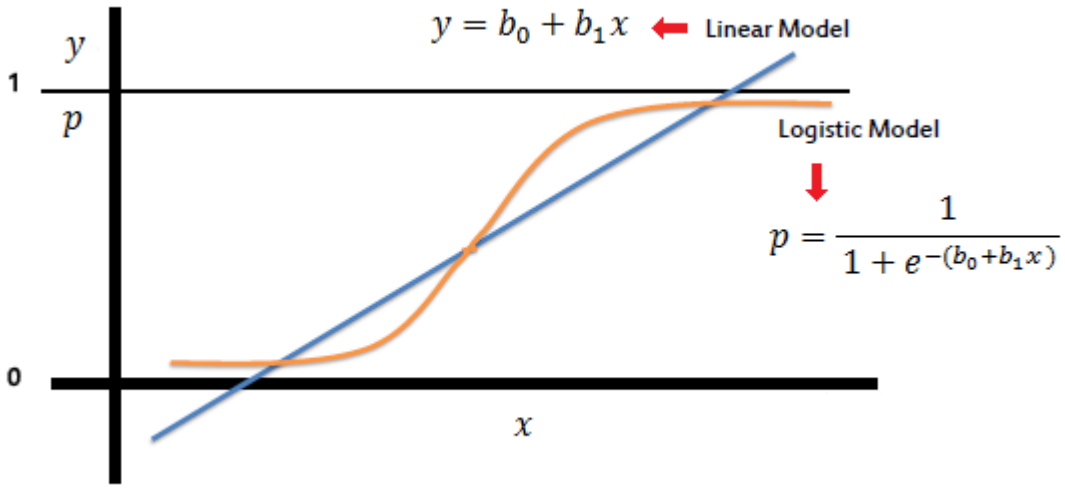
- **NLTK**- NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems.

- **Gensim**- Gensim provides the Word2Vec class for working with a Word2Vec model.

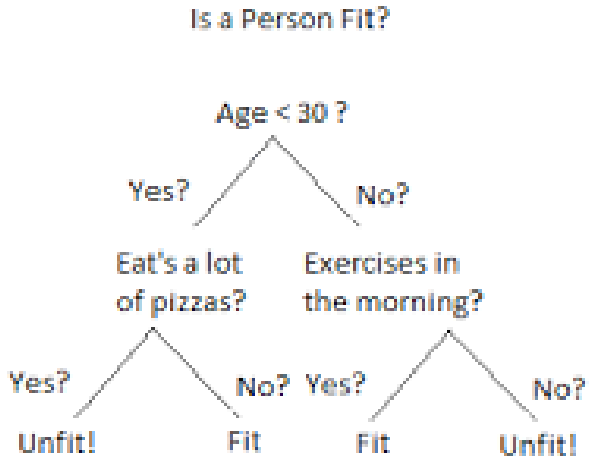
- **Matplotlib**- Matplotlib is a Python 2D plotting library which produces publication quality figures. Matplotlib tries to make easy things easy and hard things possible. We can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc.

- **Spyder**- Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language.

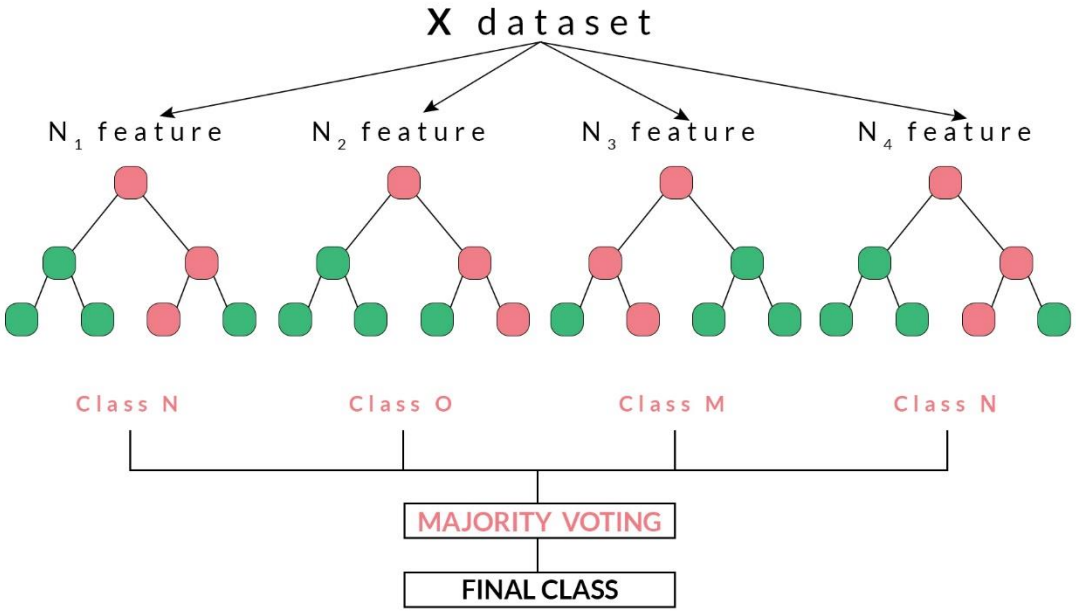
- **Logistic Regression**- Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.




- **Decision Tree**- A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

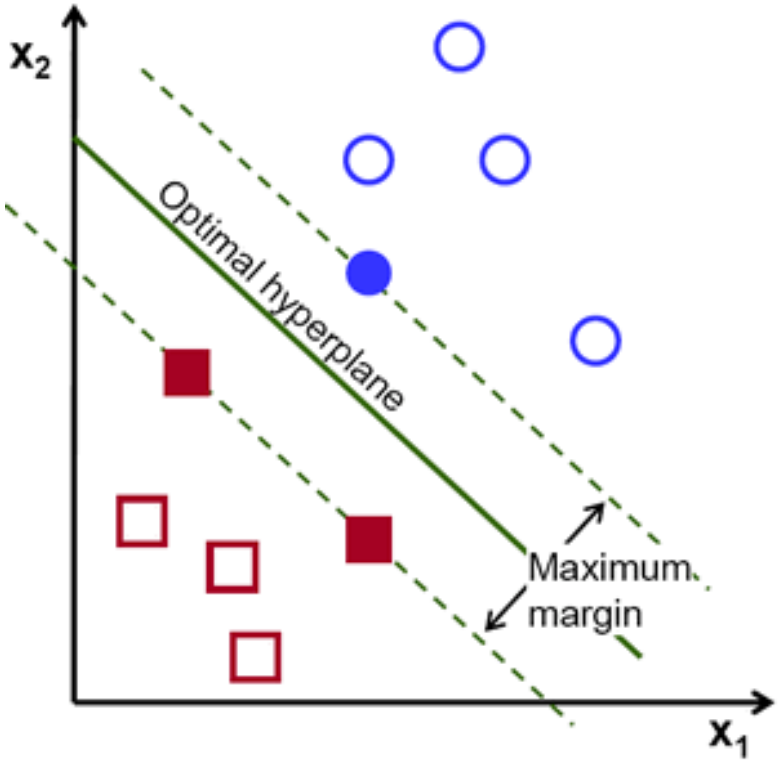


- **Random Forest**- Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes.

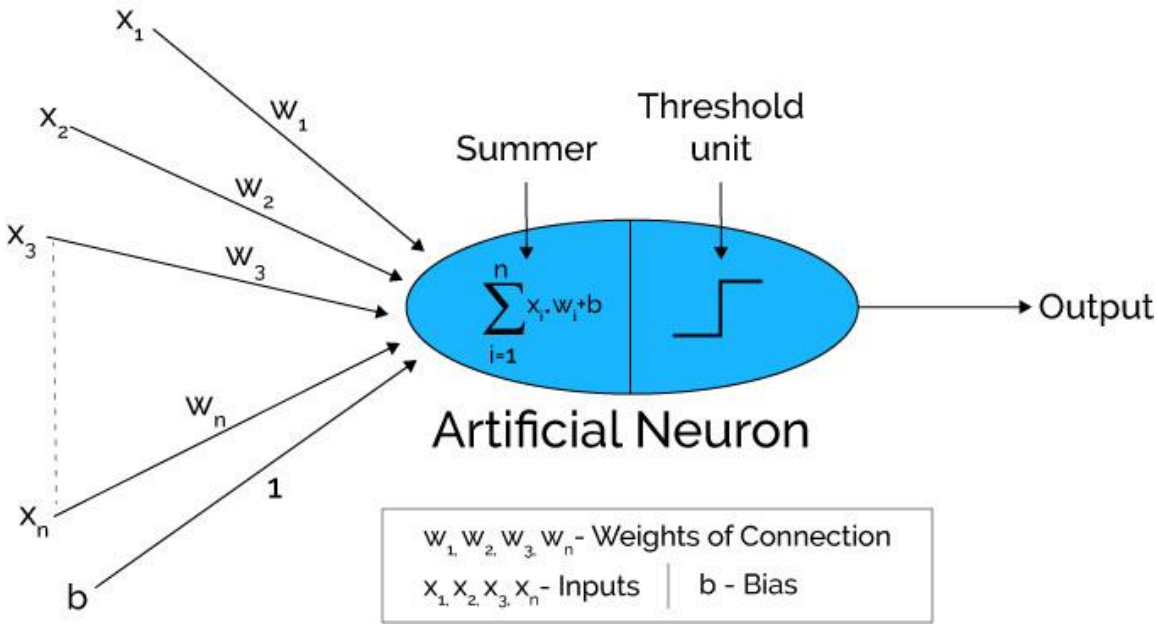



- **Support Vector Machine**- The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.

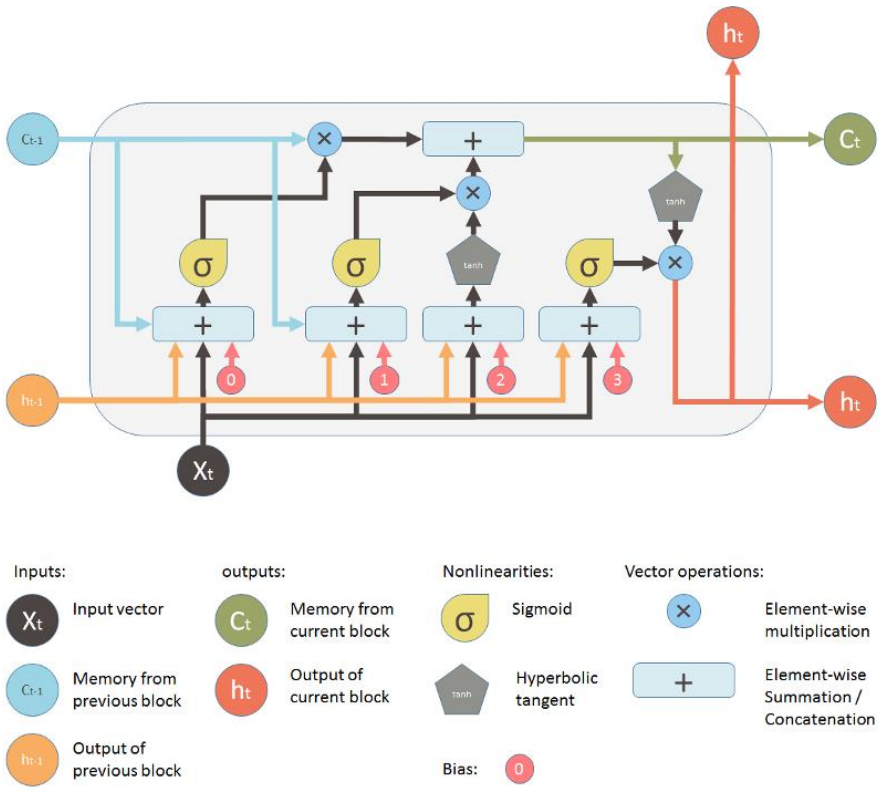
  To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.




- **Artificial Neural Networks**- Artificial Neural networks (ANN) or neural networks are computational algorithms. It intended to simulate the behaviour of biological systems composed of "neurons". ANNs are computational models inspired by an animal's central nervous systems. It is capable of machine learning as well as pattern recognition.



- **Long Short-Term Memory**- Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections that make it a "general purpose computer" (that is, it can compute anything that a Turing machine can).



# 2. Literature survey

## 2.1 Introduction

Social media provides a wealth of information about a user's behaviour and interests, from the explicit "John's interests are tennis, swimming and classical music", to the implicit "people who like skydiving tend to be big risk-takers", to the associative "people who buy Nike products also tend to buy Apple products". While information about individuals is not always useful on its own, finding defined clusters of interests and opinions can be interesting. For example, if many people talk on social media sites about fears in airline security, life insurance companies might consider opportunities to sell a new service. This kind of predictive analysis is all about understanding one's potential audience at a much deeper level, which can lead to improved advertising techniques, such as personalised advertisements to different groups. It is in the interests of large public knowledge institutions to be able to collect and retrieve all the information related to certain events and their development over time. In this new information age, where thoughts and opinions are shared through social networks, it is vital that, in order to make best use of this information, we can distinguish what is important, and be able to preserve it, in order to provide better understanding and a better snapshot of situations. Online social networks can also trigger a chain of reactions to such situations and events which ultimately lead to administrative, political and societal changes.

## 2.2 Related work

Social construction by users can be accredited with the increase in citizen participation in the electoral processes. Such participation occurs not only during the lead up to the main day, i.e. during the campaign, but also on election day itself. According to Wagner and Gainous (2013), this participation brings forth a new context where engagement in social networks may increase the citizen's role in the political process and strengthen democratic institutions on different levels.

An interest in analysing the enormous amount of data produced globally has been created by advances in social media mining and social network analysis. These data can be collected and analysed to investigate and discover patterns of behaviors and interactions and better understand related issues (Xi and Li 2013). This was also the approach used in the study Prati and Said-Hung (2017) where they studied the exchange of messages that had a defined ideological load, as well as the citizen participation on Twitter during the 24M Spanish elections. They analysed levels of segregation observed in homophilous and political orientations (Gruzd and Roy 2014) in messages posted digitally, and took into consideration the timeline of events (Elmer 2012).

Sentiment analysis is generally in a two-phase format, where in the first phase, relevant data (tweets) are gathered, and in the second phase, the actual sentiment is extracted. Tweets can be considered relevant if they contain words from a list of target keywords compiled either manually (Wang et al. 2012; O'Connor et al. 2010; Tumasjan et al.2010), or semi-automatically (Conover et. al. 2011a, b) by expanding a seed set. Subsequently, once the set of relevant tweets is compiled, various supervised or unsupervised methods are utilized to extract the sentiment from the tweet. Unsupervised methods rely on the so-called opinions of lexicons, a list of 'positive' and 'negative' keywords, estimating a sentiment based on the ratio of the occurrence of the two types of keywords with respect to one another (O'Conner et. al. 2010), or just the raw count of words (Choy et. al. 2011). More advanced approaches employ supervised learning techniques, and train prediction models either on manually labelled tweets (Conover et. al. 2011a, Wang et. al. 2012), or on tweets with an emotional context (Marchetti-Bowick and Chambers 2012), i.e., emoticons and hashtags, such as :-), #happy, #sad, etc.

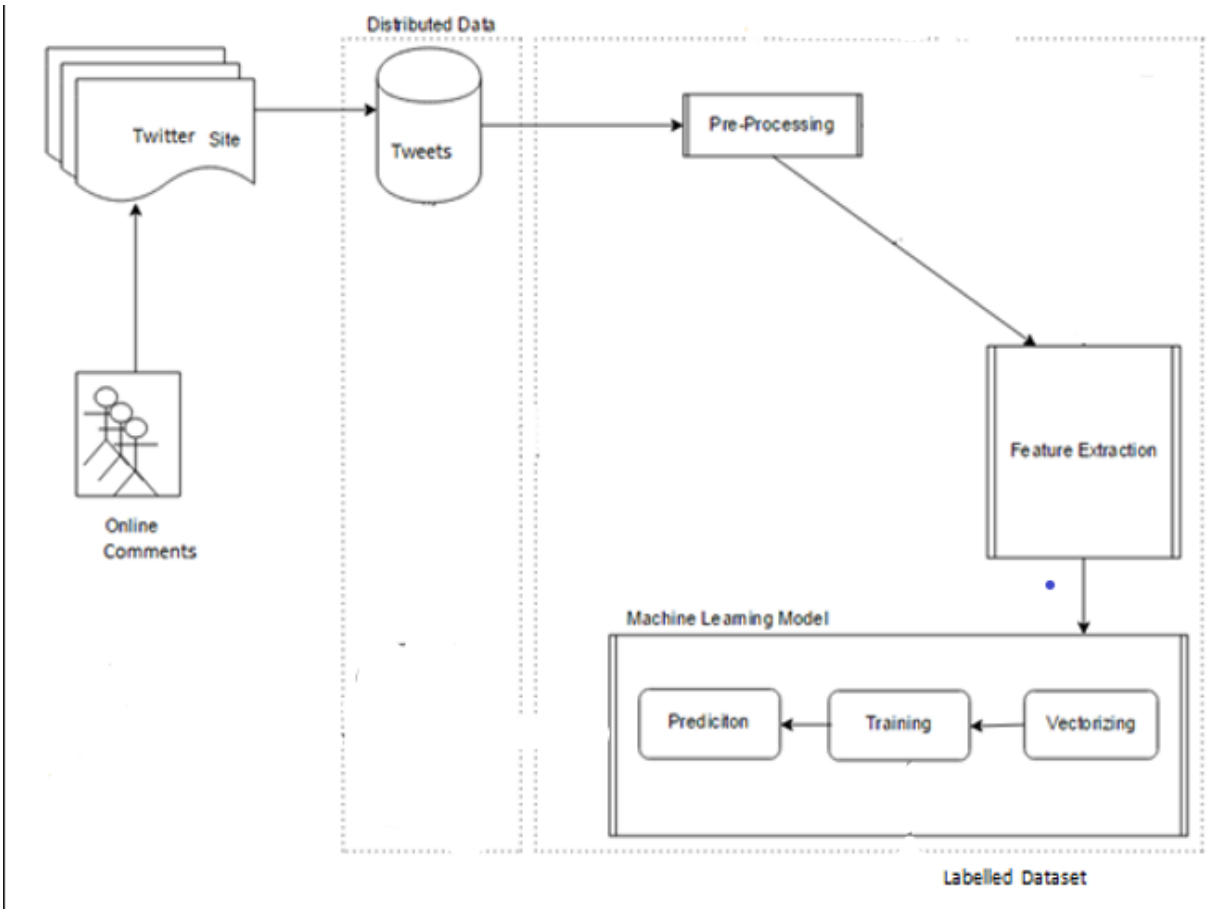
Research studies by A. Jain and P. Dandannavar (2017) focused on combining a lexical based approach with a machine learning based approach to form a hybrid approach to sentiment analysis. Other studies, such as the ones done by Koc-Michalska et. al. (2014) and Fominya (2014), focused on traditional media outlets, social movements, electoral campaigns and voter engagement. A paper put forward by M.D. Conover et. al. (2011) studied the partisanship of voters with respect to political leanings and how the two groups interact with each other; in the form of retweets or mentions. Deltell et. al (2013) and Deltell (2012) have studied the impact of Twitter in particular, and digital media in general, in both local and national elections in Spain. V. Sahayak et. al. (2015) focused on sentiments pertaining to a variable search query, with the intention of making it easier for companies to gather feedback about their products, and for customers who want to search the opinions of others regarding a product, before purchase.

# 3. Implementation

## 3.1 Introduction

- In the implementation, we have applied five methods to represent the words into vector form in the political comments.
- Methods are:
    1. Count Vectorizer
    2. TF-IDF (Unigrams)
    3. TF-IDF (Bigrams)
    4. TF-IDF (Trigrams)
    5. Word2Vec
- After that we have applied different classifiers such as Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Support Vector Machine, Artificial Neural Network and Long Short-Term Memory.

## 3.2 Proposed framework



**Phase 1: Data Scraping**

Twitter has provisions for the collection of data to be used by researchers. One must create an application to obtain credentials (i.e. API keys, Access tokens etc) which can be done from the twitter developer website. This application is then given an "OAuth access token" which makes sure that authorized calls are made to Twitter API. OAuth is the newer method for accessing the Twitter API through the tweepy library, with the previous method – Basic Authentication – now being discarded. It permits users to approve of third-party applications to access their account data, but shows only their public data, without sharing data like passwords. We extracted tweets based on the search queries 'Elections in India', 'Lok Sabha Elections' etc. Twitter data collected here is in the form of Json data. The fields selected are given below:

- ID: A unique ID given to each tweet. We use a numerical ID and not the username as an identifier because we wish to keep the users anonymous. All we want to do is get an idea of the number of people belonging in any of the camps.
- Text: Shows the original tweet posted by the user.
- Class: Three columns, two of them belonging to the two major parties (i.e. BJP and Congress) and the last column for any other party. These columns accept binary data based on whether the tweet is aligned towards any one, two or all of them or even none of them at all. The 3 columns are taken together, and the binary number formed is converted to decimal to serve as an easy classifier. It can be from the range of 0-7.

**Phase 2: Data Pre-processing**

Prior to performing the sentiment analysis, data must be primed or pre-processed. This entails bringing it into proper form and retrieving the relevant features required for successful sentiment classification. The text field, as mentioned before, contains the tweet as it was posted by the user. This includes any mentions, URLs, and other data that is irrelevant to the performance of analysis. All these things must be filtered out to reduce unnecessary complications in the process of finding out the sentiments. After relevant data is extracted from the tweet, the corpus is scanned for any occurrence of duplicate data. This too is filtered out, thereby assuring the uniqueness of each tweet. Every word in each tweet of the dataset is then converted to lowercase, to make the analysis easier to process. Following this, stop words are removed from the corpus. Stop words are frequently occurring words such as 'a', 'has', 'it' etc that inflate the data present in the corpus and do nothing in terms of easing the sentiment analysis process. The final two processes utilized in this phase are stemming and lemmatization, where stemming refers to the reduction of words such as 'playing', 'voters' etc. to their root words, 'play', 'voter' etc. to reduce the number of unique vectors and make classifications and predictions easier; Lemmatization, deals with the abstraction of inflectional terminuses only, and

returns the base or dictionary form of a word, which is termed as the lemma. E.g. Converting words like ‘are’, ‘am’, ‘is’ to ‘be’.

- Before pre-processing:

| | | | | |
|---|---|---|---|---|
| 781 | Jai hind...Modi again? | 1 | 0 | ( |
| 782 | Pappu has gone mad.nobody cares for his SICK SLOGAN? | 1 | 0 | ( |
| 783 | #MainBhiChowkidar | 1 | 0 | ( |
| 784 | Mai bhi Chowkidar? | 1 | 0 | ( |
| 785 | Main Bhi Chowkidar.? | 1 | 0 | ( |
| 786 | Modi is the best Pm? | 1 | 0 | ( |
| 787 | Main bhi Chowkidaar hu. ????????????? | 1 | 0 | ( |
| 788 | We don't Want this choukidar in 2019? | 0 | 1 | ( |
| 789 | Cong will be win? | 0 | 1 | ( |
| 790 | Bjp will win all major state all exitpoll wrongs this time? | 1 | 0 | ( |
| 791 | BJP only BJP? | 1 | 0 | ( |
| 792 | Congress is bad party in India? | 1 | 0 | ( |
| 793 | BJP ... backed media....zee news.? | 0 | 1 | ( |
| 794 | Narendra modi u r surely Goin to hell.... N why India people r look like potato...? | 0 | 1 | ( |
| 795 | Jai congress Vijay congress? | 0 | 1 | ( |
| 796 | Pappu will exit permanently.? | 1 | 0 | ( |
| 797 | we want bjp? | 1 | 0 | ( |
| 798 | Congress will be back in Madhya Pradesh, chhatisgarh, rajasthan,mizoram,telangana and at the the centre too. | 0 | 1 | ( |
| 799 | Rahul is the best political dynamic leader.old yrs gone lo New one come.? | 0 | 1 | ( |
| 800 | Pappu is comedian of India. ?????? His speeches are too funny and stress busting ??????? | 1 | 0 | ( |
| 801 | Always give support young generation,so next PM rahul? | 0 | 1 | ( |
| 802 | ONLY CONGRESS MUMBAI BHAI LOG CONGRESS ZINDABAD? | 0 | 1 | ( |

- After pre-processing:

| | | | |
|---|---|---|---|
| 780 | str | 1 | jai hind...modi again? |
| 781 | str | 1 | pappu gone mad.nobodi care sick slogan? |
| 782 | str | 1 | #mainbhichowkidar |
| 783 | str | 1 | mai bhi chowkidar? |
| 784 | str | 1 | main bhi chowkidar.? |
| 785 | str | 1 | modi best pm? |
| 786 | str | 1 | main bhi chowkidaar hu. ???????????? |
| 787 | str | 1 | want choukidar 2019? |
| 788 | str | 1 | cong win? |
| 789 | str | 1 | bjp win major state exitpol wrong time? |
| 790 | str | 1 | bjp bjp? |
| 791 | str | 1 | congress bad parti india? |
| 792 | str | 1 | bjp ... back media....ze news.? |
| 793 | str | 1 | narendra modi u r sure goin hell.... n india peopl r look like potato. ... |
| 794 | str | 1 | jai congress vijay congress? |
| 795 | str | 1 | pappu exit permanently.? |
| 796 | str | 1 | want bjp? |
| 797 | str | 1 | congress back madhya pradesh, chhatisgarh, rajasthan,mizoram,telangana ... |
| 798 | str | 1 | rahul best polit dynam leader.old yr gone lo new one come.? |
| 799 | str | 1 | pappu comedian india. ?????? speech funni stress bust ??????? |

**Phase 3: Vector representation of words**

- **Method 1: Count Vectorizer** - Count Vectorizer provides a way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary. 500 most frequent words was extracted in this method.
    - Sample of training dataset using Count Vectorizer:

| | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 261 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 262 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 263 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 264 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 265 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 266 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 267 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 268 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 269 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 270 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 271 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 272 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 273 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 274 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 275 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 276 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 277 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 278 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 279 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Method 2: TFIDF Vectorizer (Unigrams)** - In information retrieval, **tf–idf** or **TFIDF,** short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. An n-gram of size 1 is referred to as a "**unigram**".

  In other words, it assigned weightage to words on the basis of rarity in a document. TF-IDF formula is expressed below:

  For a term *i* in document *j*:

  $$w_{i,j} = tf \times \log\left(\frac{N}{df_i}\right)$$

  $tf_{i,j}$ = number of occurrences of *i* in *j*.

$df_i$ = number of documents containing *i*

*N* = total number of documents

- Sample of training dataset using TFIDF (Unigram):

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 0 | 0 | 0.853892 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 27 | 0 | 0 | 0.448188 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 30 | 0 | 0.343624 | 0.19737 | 0 | 0 | 0.36513 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 33 | 0 | 0 | 0.393353 | 0.713908 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.267223 | 0 | 0 | 0 | 0 |
| 35 | 0 | 0 | 0.180351 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 36 | 0 | 0 | 0 | 0 | 0 | 0.489586 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Method 3: TFIDF Vectorizer (Bigrams)** - An n-gram of size 2 is referred to as a "**bigram**". We will choose a pair of words instead of just a single word. We are dealing with 200 most frequent words.
    - Sample of training dataset using TFIDF (Bigram):

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0.328732 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.828963 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0.466082 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.533007 | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.700301 | 0 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0.853892 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

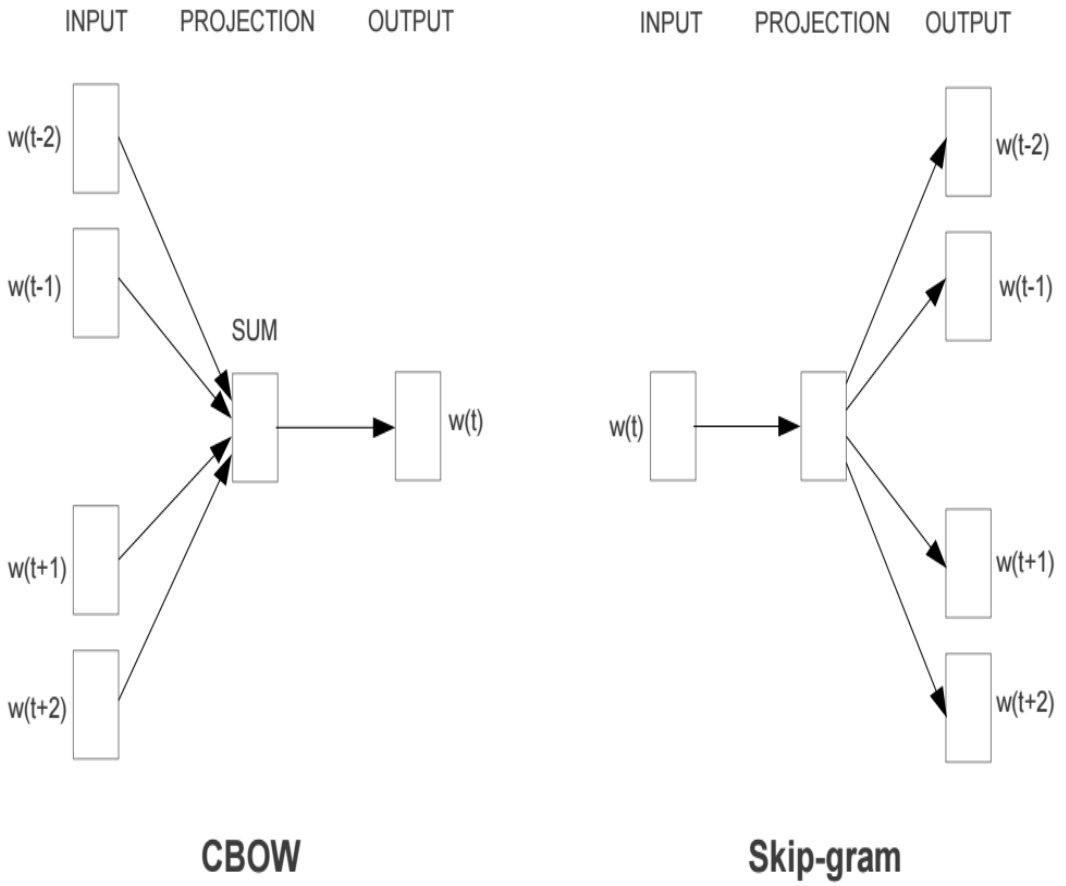
- **Method 4: TFIDF Vectorizer (Trigrams)** - An n-gram of size 3 is referred to as a "**trigram**". We will choose three words which appear in a line consecutively instead of a pair of word. We are dealing with 200 most frequent words.
    - Sample of training dataset using TFIDF (Bigram):

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 210 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 211 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 212 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 213 | 0 | 0 | 0.360735 | 0.654709 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 214 | 0 | 0 | 0.220629 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 215 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 216 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 217 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 218 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 219 | 0 | 0 | 0.364812 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 220 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 221 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 222 | 0 | 0 | 0.803493 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 223 | 0 | 0 | 0.678318 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 224 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 225 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.408481 | 0 |
| 226 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 227 | 0 | 0 | 0.481096 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 228 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Method 5: Word2Vec** - Word2Vec is a two-layer neural net that processes text. Its input is a text corpus and its output are a set of vectors: feature vectors for words in that corpus. While Word2Vec is not a deep neural network, it turns text into a numerical form that deep nets can understand. It can be obtained using two methods (both involving Neural Networks): Skip Gram and Common Bag of Words (CBOW).

  ***CBOW Model:*** This method takes the context of each word as the input and tries to predict the word corresponding to the context.

  ***Skip Gram Model:*** The Skip Gram model architecture usually tries to achieve the reverse of what the CBOW model does. It tries to predict the source context words (surrounding words) given a target word (the centre word).
  CBOW is faster and has better representations for more
  frequent words. We used CBOW model.

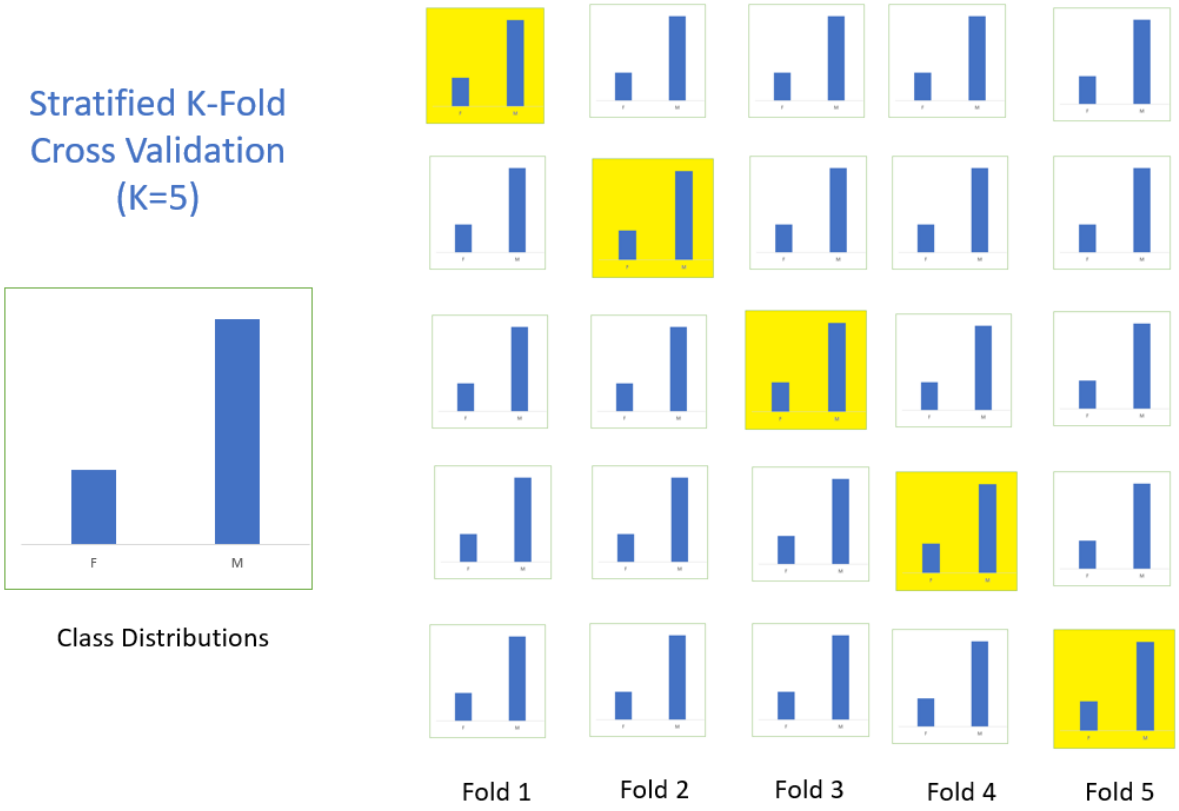


- Sample of training dataset using Word2Vec: Each word is represented using 50 dimensions.

| Index | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2943 | 145747 | 0.257686 | 0.32916 | 0.0635675 | 0.485204 | 0.103703 | -0.0700773 | 0.313047 | 0.372663 | 0.311399 | 0.117103 | 0.319066 | -0.0 |
| 3722 | 52155 | 0.904414 | -0.507035 | -0.208156 | -0.588073 | -0.237457 | 0.312889 | -1.15634 | -0.970405 | 1.06762 | -0.0737752 | 1.13119 | 0.41 |
| 1865 | 9605469 | 0.10866 | 0.251132 | 0.218653 | 0.0848256 | -0.412062 | 0.0175475 | -0.0722837 | 0.212098 | 0.384166 | -0.231225 | 0.327267 | 0.16 |
| 3213 | 134358 | -0.0981644 | 0.292753 | 0.343312 | 0.168787 | -0.252452 | -0.0964573 | 0.154318 | 0.457703 | 0.240763 | -0.113004 | 0.287249 | -0.1 |
| 701 | 407642 | -0.245949 | 0.475201 | -0.677199 | 0.458553 | 0.0258104 | 0.380127 | 0.030496 | 0.150129 | 0.917487 | 0.249444 | 0.155547 | -1.1 |
| 2561 | 184509 | -0.298738 | 0.449301 | 0.202123 | 0.367407 | -0.140682 | -0.204015 | -0.01377 | 0.395185 | 0.445846 | 0.0276459 | 0.310263 | -0.2 |
| 3433 | 52155 | 0.904414 | -0.507035 | -0.208156 | -0.588073 | -0.237457 | 0.312889 | -1.15634 | -0.970405 | 1.06762 | -0.0737752 | 1.13119 | 0.41 |
| 3388 | 9963401 | -0.0799112 | 0.254391 | 0.213508 | 0.224291 | -0.157382 | -0.0253348 | 0.211279 | 0.304155 | 0.254052 | -0.0716688 | 0.22322 | -0.1 |
| 1236 | 3712 | -0.273467 | 0.075597 | 0.285628 | 0.393153 | -0.180683 | 0.0558804 | 0.0651813 | 0.218276 | 0.382591 | -0.0344939 | 0.0721377 | 0.26 |
| 3320 | 110724 | -0.0540824 | 0.182681 | 0.346598 | 0.377711 | -0.11182 | -0.0399628 | 0.376556 | 0.300736 | 0.299966 | -0.269532 | 0.250893 | 0.16 |
| 2961 | 9819118 | -0.465225 | 0.150177 | 0.420868 | -0.145536 | -0.714594 | 0.360483 | 0.541762 | 0.39382 | -0.125988 | -0.420138 | 0.220499 | -0.5 |
| 2456 | 9819118 | -0.465225 | 0.150177 | 0.420868 | -0.145536 | -0.714594 | 0.360483 | 0.541762 | 0.39382 | -0.125988 | -0.420138 | 0.220499 | -0.5 |
| 3885 | 226495 | -0.0223608 | 0.262528 | 0.122595 | 0.209722 | -0.144994 | 0.0470969 | 0.240808 | 0.201121 | 0.308599 | -0.094327 | 0.310437 | -0.2 |
| 1587 | 9939883 | 0.477007 | -0.108515 | -0.0799695 | 0.0300297 | -0.273083 | 0.0711237 | -0.23035 | 0.311062 | 0.193338 | -0.0603644 | 0.171966 | -0.1 |
| 3777 | 204817 | 0.0686283 | 0.251089 | 0.124333 | 0.242395 | -0.0466692 | 0.0096193 | 0.133098 | 0.240747 | 0.247614 | -0.0281799 | 0.350073 | -0.2 |
| 2714 | 28344 | 0.457429 | 0.27852 | -0.14048 | 0.434192 | 0.501052 | -0.575975 | -0.209694 | 0.578766 | 0.300025 | 0.369491 | 0.254614 | -0.2 |
| 2155 | 514678 | 0.465737 | 0.279941 | -0.376273 | 0.981006 | 0.229735 | -0.079523 | 0.55026 | -0.0365256 | 1.03008 | -0.0643246 | 0.303662 | -0.1 |
| 1590 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 422 | 52155 | 0.904414 | -0.507035 | -0.208156 | -0.588073 | -0.237457 | 0.312889 | -1.15634 | -0.970405 | 1.06762 | -0.0737752 | 1.13119 | 0.41 |
| 3027 | 52155 | 0.904414 | -0.507035 | -0.208156 | -0.588073 | -0.237457 | 0.312889 | -1.15634 | -0.970405 | 1.06762 | -0.0737752 | 1.13119 | 0.41 |

**Phase 4: Splitting of data into training and test set.**

- **Stratified K-fold**- Stratified K Fold algorithm is used to generate the training fold/set, where k is set as 10. This algorithm seeks to ensure that each fold is representative of all strata of data. This means that each class is (approximately) equally represented across each test fold. The excess data is then used as the test set.



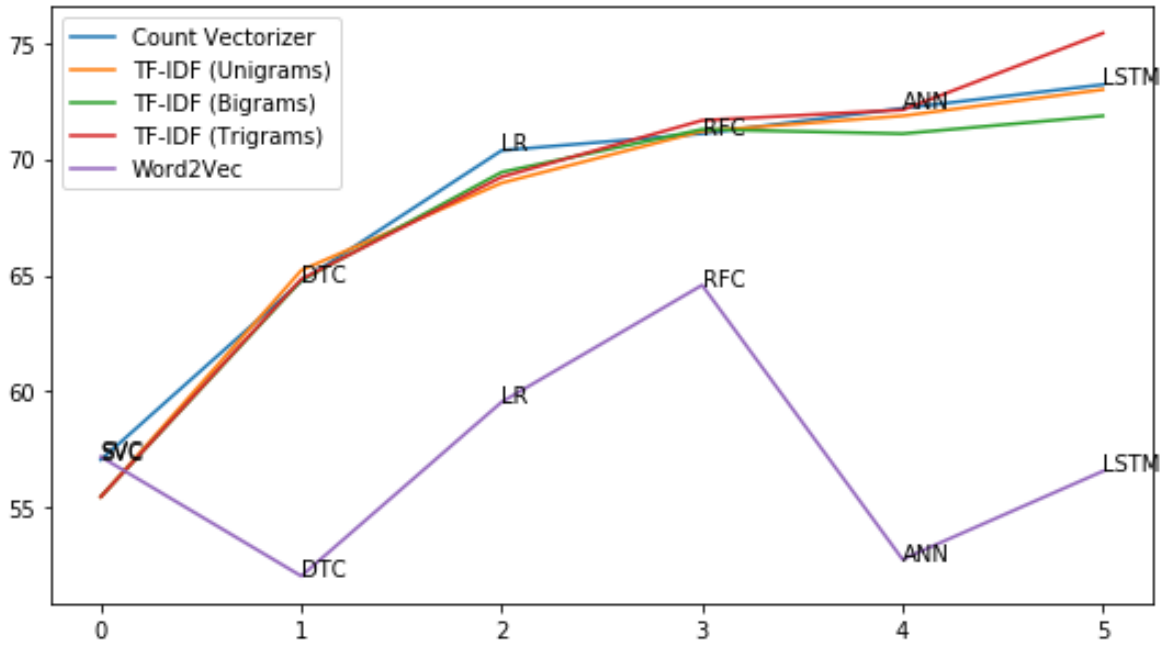

**Phase 5: Applying different classifiers**

- **Method 1: Count Vectorizer** – LSTM gave the highest accuracy, i.e., 73.21 percent.
- **Method 2: TFIDF Vectorizer (Unigrams)** – LSTM gave the highest accuracy, i.e., 73 percent.
- **Method 3: TFIDF Vectorizer (Bigrams)** – LSTM gave the highest accuracy, i.e., 71.86 percent.
- **Method 4: TFIDF Vectorizer (Trigrams)** – LSTM gave the highest accuracy, i.e., 75.43 percent.
- **Method 5: Word2Vec** – Random Forest Classifier gave the highest accuracy, i.e., 64.57 percent.

# 4. Experimental results

## 4.1 Experimental results

- Accuracies given by different classifiers using different feature extraction method:




- It was found that the LSTM classifier when used along with TF-IDF (Trigrams) approach gave the most accuracy, i.e. 75.43% as compared to the other models. According to our result, sentiments were recorded greatly in the BJP's favour and a prediction can be made that they can win the 2019 Lok Sabha Elections.

- On the other hand, LSTM classifier when used along with Word2Vec approach showed poor accuracy, i.e. 56.57% as compared to other models.

## 4.2 Word Cloud

- It is also useful to have an idea of the words most commonly floating around in these tweets. This information is shown in the form of a word cloud.
- Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analysing data from social network websites.

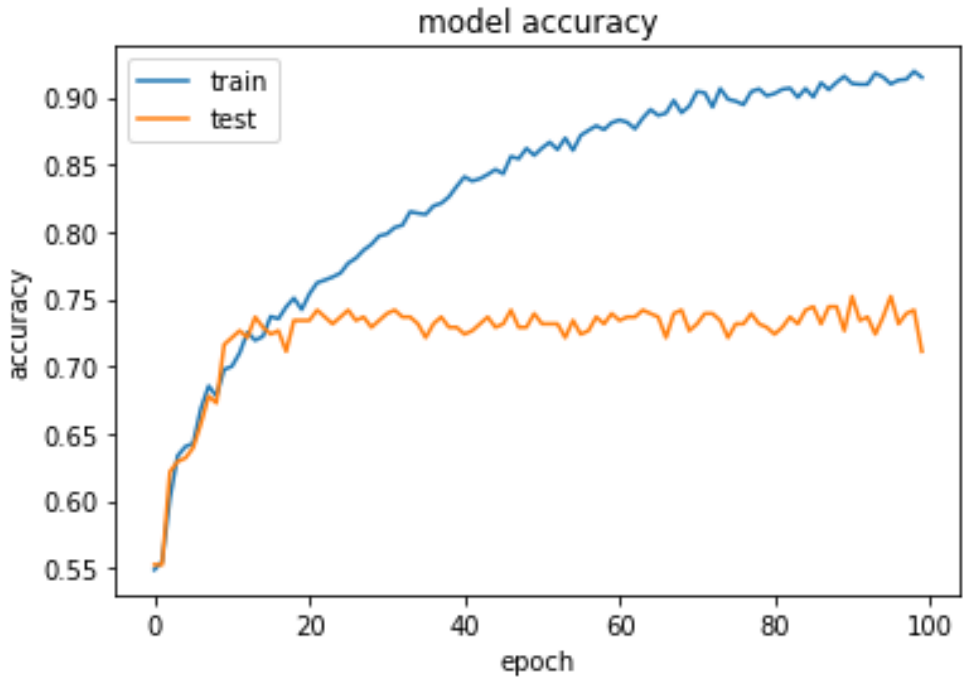
- According to the word cloud, the term *Modi* shows up most often in these tweets, with *BJP* and *Congress* coming in on second and third place respectively.


## 4.3 Model Training History

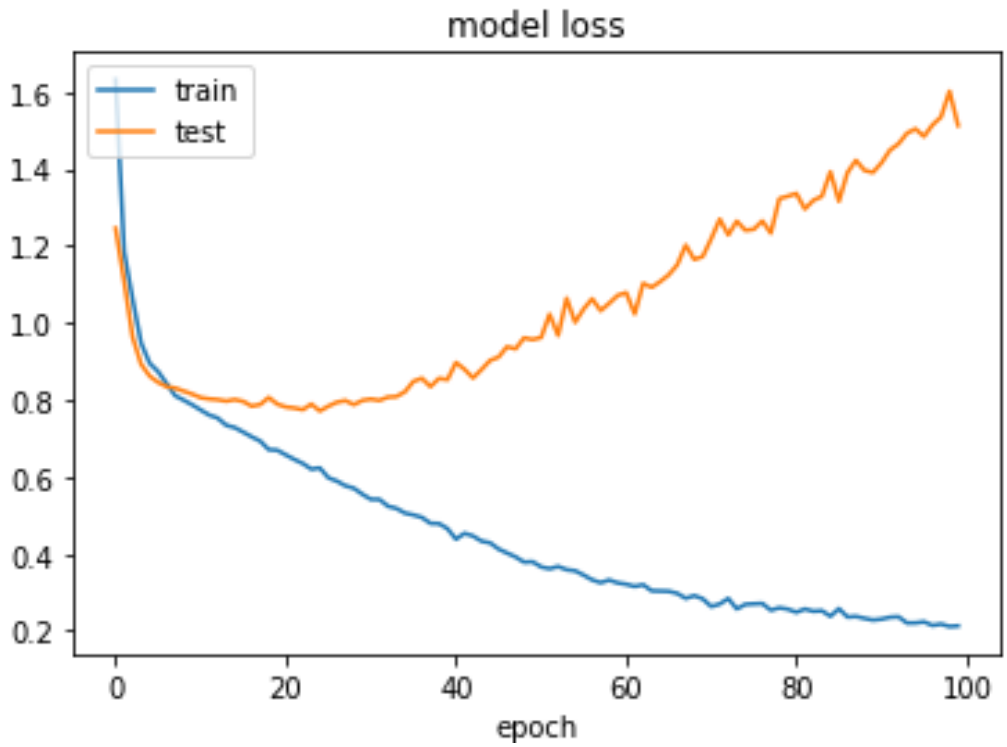
The plots can provide an indication of useful things about the training of the model, such as:

- It's speed of convergence over epochs (slope).
- Whether the model may have already converged (plateau of the line).
- Whether the mode may be over-learning the training data (inflection for validation line).
- How many epochs to choose?
- Loss is the penalty for a bad prediction. That is, loss is a number indicating how bad the model's prediction was on a single example. The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples.

1. Model Accuracy for LSTM when used with TFIDF (Trigrams): A plot of accuracy on the training and validation datasets over training epochs.

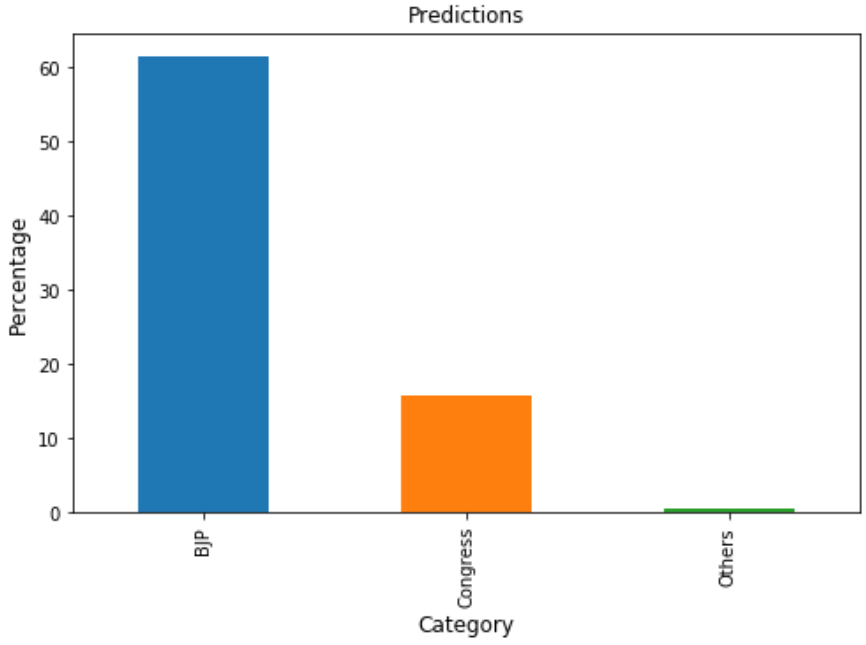



2. Model Loss for LSTM when used with TFIDF (Trigrams): A plot of loss on the training and validation datasets over training epochs.




## 4.4 Results

- The graph tells that there are a greater number of texts from BJP category.
- It can be seen that around 67% of tweets are in support of the BJP, as compared to the 15% (approx.) for Congress and Others. Tweets in which users were supporting 2 or more parties were not included in this graph.
- According to this BJP can win 2019 Lok Sabha elections.



- The highest accuracy was for the LSTM classifier when applied to TF-IDF (Trigrams) method for feature selection. The data for this classifier is shown in Table.

Table. Performance of LSTM using TF-IDF (Trigrams) along with stratified 10-fold cross validation technique.

| Classes | Precision | Recall | F1 Score |
|---|---|---|---|
| Unrelated | 0.88 | 0.7 | 0.78 |
| Others | 0.33 | 0.12 | 0.18 |
| Congress | 0.73 | 0.77 | 0.75 |
| Congress or Others | 0.72 | 0.39 | 0.5 |
| BJP | 0.78 | 0.92 | 0.85 |
| BJP or Others | 0.81 | 0.61 | 0.7 |
| BJP or Congress | 0 | 0 | 0 |
| Any of the three parties | 0 | 0 | 0 |

# 5. Conclusion

A dataset of around 4000 tweets was acquired and processed for this experiment. The corpus was divided into training and test sets using a stratified 10-fold cross validation technique. A variety of machine learning techniques were used, and their performances were evaluated. It was found that the LSTM classifier when used along with TF-IDF (Trigrams) approach gave the most accuracy, i.e. 75.43% as compared to the other models. According to our result, sentiments were recorded greatly in the BJP's favour and a prediction can be made that they can win the 2019 Lok Sabha Elections. Typically, opinion mining looks at social media content to analyse people's explicit opinions about an organisation, product or service. However, this backwards looking approach often aims primarily at dealing with problems, e.g., unflattering comments, while a forwards-looking approach aims at looking ahead to understanding potential new needs from consumers. This is achieved by trying to understand people's needs and interests in a more general way, e.g. drawing conclusions from their opinions about other products, services and interests. It is not sufficient, therefore, to look at specific comments in isolation: non-specific sentiment is also an important part of the overall picture.

## References


[1] Countries with the most twitter users 2019 | Statistics [Online]. Available: https://twitter-users-in-selected-countries/

[2] India: Number of twitter users 2019 | Statistics [Online]. Available: https://www.statista.com/statistics/381832/twitter-users-india/

[3] Wagner KM, Gainous J (2013) Digital uprising: the internet revolution in the Middle East. J Inf Technol Politics. doi:10.1080/19331681.2013.778802

[4] Koc-Michalska K et al (2014) Poland's 2011 online election campaign: new tools, new professionalism, new ways to win votes. J InfTechnol Politics. doi:10.1080/19331681.2014.899176

[5] Fominaya CF (2014) Social movements and globalization: how protests, occupations and uprisings are changing the world. Palgrave Macmillan, New York.

[6] Conover M, Gonc¸alves B, Ratkiewicz J, Flammini A, Menczer F. (2011a). Predicting the political alignment of Twitter users. In: Proceedings of SocialCom/PASSAT Conference, pp 192–199

[7] Conover, M., Ratkiewicz, J., Francisco, M., Goncalves, B., Menczer, F., & Flammini, A. (2011b). Political Polarization on Twitter. In International AAAI Conference on Web and Social Media. Retrieved from https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2847

[8] Prati R.C, Said-Hung E. (2017). Predicting the ideological orientation during the Spanish 24M elections in Twitter using

machine learning. doi 10.1007/s00146-017-0761-0

[9] Gruzd A, Roy J. (2014). Investigating political polarization on Twitter: a Canadian perspective. Policy Internet. doi:10.1002/1944-2866

[10] Elmer G (2012) Live research: Twittering an election debate. New Media Soc. doi:10.1177/1461444812457328