Indian General Elections-2019 -Analysis & Modelling with Supervised Learning & Sentiment Analysis

A project report submitted to ICT Academy of Kerala

in partial fulfillment of the requirements

for the certification of

CERTIFIED SPECIALIST

IN

DATA SCIENCE & ANALYTICS

submitted by

Arjun KB

Vallibhai E.S

Vasantha S

Venkitesh S

ICT ACADEMY OF KERALA

THIRUVANANTHAPURAM, KERALA, INDIA

May 2021

List of Figures

S.No

List of Abbreviations

1. NDA - National Democratic Alliance
2. GDP - Gross Domestic Product
3. BJP - Bharatiya Janata Party
4. AIADMK - All India Anna Dravida Munnetra Kazhagam
5. AI - Artificial Intelligence
6. IT - Information Technology
7. MP - Member of Parliament
8. MLA - Member of Legislative Assembly
9. k-NN - K Nearest Neighbours
10. SVM - Support Vector Machine
11. XGBoost - eXtreme Gradient Boosting
12. ROC, AUC - Receiver Operating Characteristic, Area Under Curve
13. TP, FP - True Positive, False Positive
14. LOOCV - Leave-One-Out Cross-Validation
15. NLP - Natural LAnguage Processing
16. API - Application Programming Interface
17. REST - Representational State Transfer
18. TFIDF - term frequency–inverse document frequency
19. Pos - Part of Speech
20. LSI - Latent Semantic Indexing
21. MI - Mutual Information
22. BOW - Bag of Words
23. LIWC - Linguistic Inquiry and Word Count
24. NLTK - Natural Language Toolkit
25. VADER - Valence Aware Dictionary Sentiment Reasoning
26. INC - Indian NAtional Congress
27. BSP -Bahujan Samaj Party
28. CPI(M) - Communist Party of India (Marxist)
29. CSV - Comma Separated Value
30. NaN - Not a Number
31. UP - Uttar Pradesh

Table of Contents

Abstract

Indian General Elections is the world's largest democratic exercise which elects the members to the House of People or the lower house of the parliament- Lok Sabha. It is conducted once every five years. In this project, we present an analysis of the general elections held in the year 2019. The performance of different national and state political parties is considered along with the voter turnout to give a complete picture of the elections. The analysis is done with the use of suitable visualizations to gain clarity on various factors which influenced the victory or defeat of both the incumbents and challengers. In addition to this, a brief study on the initiatives and innovations which made this election different from the earlier ones is also done.

We also intend to create a machine learning model which can predict the results of this election. This is compared with the actual results to observe the performance of the model. The model is created based on the sentimental analysis of tweets made before elections as well as on the results of the election. Different supervised learning approaches are compared to arrive at the best fit model.

The primary source of data is the general election statistics of the year 2019. In addition to this, a sample twitter dataset is used for sentiment analysis and modeling. Other secondary datasets are employed for analysis and visualizations.

1. Problem Definition

1.1 Overview

The general elections in India is one of the biggest voting exercises in the world and it decides who will rule the country for the next five years. The election process involves rigorous campaigning and every political party targets the voting population through various mediums like road shows, one-to-one meetings, mass gatherings etc. But, in recent years, the trend has slowly shifted to social media as well. Among them, twitter has received widespread popularity among both the contestants and the common man. It has given them a platform to exchange their views and opinions seamlessly. The tweets are representative of the political opinions of people and analysing them can derive useful insights. It can help to predict the result of an impending election.

This project deals with the analysis of results of the general elections in India held in the year 2019. In addition, we also aim to perform a prediction of results based on the sentiment analysis of social media posts. For this, we have considered a sample data of tweets made on one of the days before the elections. The prediction of election results based on this analysis is compared with the actual results to study the impact of social media in the voting pattern of the people.

1.2 Problem Statement

The results of general elections in the year 2019 are analysed to study the vote share of the major political parties, the polling percentages and few other useful insights. Sentimental analysis is made on tweets made before elections to study the impact of people's real opinions and hence the results. A classification is also done to predict the result based on the analysis. This is compared to the actual election results to evaluate the efficiency of the prediction.

1.3 Objectives

- To perform data analysis and visualization on the results of indian general election in 2019 using various libraries in the Python environment and Tableau tool.
- To predict the election results based on the sentimental analysis of election tweet samples(1 lakh samples used) and to evaluate it using the actual results.

1.4 About the Domain

This project comes under the domain of election and polling. The use of data analytics and machine learning are well identified in this domain as more and more political parties are employing these techniques to get a grasp of their vote share in future. It is already in place in many countries in the world including India. The recent elections in India have made use

of full fledged data analytics to gather information on the voter sentiment and to frame appropriate election agendas.

2. Introduction

India is the largest democracy in the world and it provides its countrymen the freedom to choose their government. There are two levels of elections, one at the Parliamentary level which elects the members to the Lok Sabha and the other at the Assembly or State level which decides on the respective state governments. The general elections are held every five years and it is very crucial for the country. The recent elections were held in 2019, where the Narendra Modi led NDA government came to power for the second time. This election was different from the previous ones with respect to the impact of digitalisation and data analytics. Different political parties and their strategists performed rigorous analysis to gather information on the political wind in the country and to boost communication with the voters. This helped them to introduce suitable campaign methodologies.

Social media was extensively used during each stage of elections. It was a neck-to-neck competition as every party tried to influence the voters in the best possible ways. To cite an example, Twitter was one among the popular platforms for both the contestants and voters. It appeared like a battlefield where there were exchanges of viewpoints, arguments, hate speech and more. There were instances of bots creating huge volumes of tweets in favour of one party over the other. In addition there were both positive, negative as well as neutral sentiments towards the incumbents and challengers' tweets.

The results of this election is worth an analysis as it represents the second term of the NDA government and can shed light into its popularity/disapproval compared to the previous election in terms of the vote share, impact of other national and state parties etc.

Our project is divided into five major sections. The first section focuses on the analysis and visualization of major aspects of the elections held in 2019. The second section deals with sentiment analysis based on the twitter data samples. The third section deals with the development of a supervised machine learning model on the basis of the election results and the fourth section involves the development of a supervised learning model based on sentiment analysis. The results of both the prediction models are compared to derive meaningful conclusions. The final section is the deployment of the model by developing a suitable web application.

3 The Indian Election System- Past & Present

3.1 History of Indian General Elections

The Indian parliament follows a biaxial system. It has two houses, namely the Rajya Sabha (Upper House) & the Loksabha (Lower House). The party (or a coalition) that gets a majority in the Loksabha gets to form the central government. The term of office is for a maximum period of 5 years or until such time the party (or a coalition) enjoys a majority in the Loksabha, whichever is earlier.

Lok Sabha is composed of representatives of the people chosen by direct election on the basis of the adult ballot. The maximum strength of the House envisaged by the Constitution is 552, which is made up by election of up to 530 members to represent the States, up to 20 members to represent the Union Territories and not more than two members of the Anglo-Indian Community to be nominated by the President, if, in his/ her opinion, that community is not adequately represented in the House.

Democracy took a big leap ahead with the first general election held in 1951-52 over a four-month period. These elections were the biggest test of democracy anywhere in the world. The elections were held based on universal adult franchise, with all those twenty-one years of age or older having the right to vote. There were over 173 million voters, most of them poor, illiterate, and rural, and having had no experience of elections. The big question at the time was how the people would respond to this opportunity. Many were questioning such an electorate being able to exercise its right to vote in a politically mature and responsible manner. Some said that democratic elections were not suited to a caste-ridden, multi-religious, illiterate and backward society like India's and that only a unilateral dictatorship could be effective politically in such a diverse landscape. India's electoral system was developed according to the directives of the Constitution. The Constitution made a provision for an Election Commission. It was to be headed by a Chief Election Commissioner, to conduct elections. It was to be independent of the executive or the parliament or the party in power.

3.2 How different were 2019 elections:

It was election season again in India in 2019. The South Asian nation conducted general elections in April/May 2019 in which about more than 900 million Indians – more than the population of all the countries in Europe combined – casted their votes to elect a new federal government.

Here are a few core aspects that went as feeders into INDIAN election:

- **World's largest election:** India's elections represent the world's largest democratic exercise. Voters were electing lawmakers for the 543-member lower house of parliament, or Lok Sabha. In 2014, the Election Commission of India deployed 3.7 million polling staff, 5,

50,000 security personnel, 56 helicopters and 570 special trains to conduct a five-week-long exercise in close to a million polling stations.In a multi-party democratic system, India had 1,709 registered political parties, of which 464 fielded their candidates in 2014. In the last election, the central government spent Rs 3870 crores so that 834 million electorates could choose their parliamentary representative from 8,251 candidates.Divided into 29 states and seven federally-administered territories, and with a population of more than 1.3 billion, India's poll took place in seven phases between April 11 and May 19, with results announced on May 23.

- **Woman power:** More and more women were coming out to vote. Women outnumbered men at polling booths in half of the states in 2019 in an election that created history with the highest ever female voter turnout of 65.63 percent.Additionally election commission authorised pink polling booths completely managed by women officials to motivate female voters.
- **New battlefield:** There were 15 million voters aged 18-19 years voting in 2019. They were better informed, more educated and tech savvy than older constituents and more willing to take a different direction from their family's established political leanings so that they became a decisive force to swing any outcome.
- **New technology:** 430 million Indians owned a smartphone, half a billion used the internet, 300 million into facebook, 200 million sent messages on Whatsapp and 30 million were on Twitter. It meant that political parties and candidates had to reinvent themselves to aggressively use new technology and social media to win the hearts and minds of young voters. However the possible misuse of these platforms remained a significant concern for the election commission.
- **Costs mind boggling money:** India's Centre for Media Studies estimated parties and candidates spent some $8.3 billion for the 2019 elections. Upon comparing it to $6.5bn that the US spent on the famously free-spending presidential and congressional elections in 2016, we realise how costly India's elections are.Financing of political parties in India continues to be opaque despite the fact that they are forced to declare their incomes. In the previous year before the election, the ruling government launched electoral bonds, which allowed businesses and individuals to donate to parties without their identities being disclosed. Donors gave away nearly $150m in these bonds - and the bulk of it, according to reports, has gone to the BJP.
- **Influence of Economy:** Under the ruling government, Asia's third-largest economy appeared to have lost some of its momentum. Farm incomes had stagnated because of a crop glut and declining commodity prices, leaving farmers saddled with debt and angry.The controversial 2016 currency ban - locally called 'demonetisation' - and a complex and badly executed new uniform goods and services tax hurt small and medium businesses and threw many out of their jobs in India's huge informal economy.Exports had dropped. Joblessness had risen, and Mr Modi's government was accused of hiding uncomfortable jobs data. To make matters worse, some of India's state-owned banks were drowning in bad loans. Yet, inflation was in check. Increased government spending in infrastructure and public works had kept the economy moving. Growth was expected to be 6.8% in 2019 fiscal year. But the

fact is that India's GDP needs to grow at a rate faster than 7% for the country to continue to pull millions out of poverty.

- **Parties banking on populism and nationalism:** When governments start putting cash in the pockets of the poor to cover up for the deficiencies of the state, it results in competitive populism.The ruling government had announced direct cash transfers to farmers and waivers of farm loans. It had also promised job quotas for the less well-to-do among the upper castes and other religions.Even the opposition congress had promised to guarantee a minimum income for the poor if their party wins the elections.

Additionally, as stated by the critics, BJP's muscular nationalism and the party's majoritarian politics have left India a deeply divided and anxious nation.Unfortunately the nationalist rhetoric has emboldened radical rightwing groups to take law into their own hands on various occasions. People critical of radical Hinduism have been labelled anti-nationals. Dissent is frowned upon.India's 170 million Muslims, many say, have become the "invisible" minority. The BJP has no Muslim MPs in the lower house - even though it fielded seven candidates in 2014 and all of them lost.

3.3 Outcome of 2019 election:

- **Margins of Victory:** Narendra Modi and his ruling BJP were swept back to power. The party won 303 seats in the Lok Sabha, the lower house of India's parliament, bettering the 282 seats they won in 2014 - a performance that not many thought was possible.
- **A heterogeneity of voting behaviour:** Heterogeneity of voting behaviour varied across India, with scores by the NDA ranging from less than 5% in the south to more than 70% in its strongholds of Western India. Yet in spite of the NDA progressing nationwide by more than 5%, the geography of its votes has only marginally changed over the last five years.Unlike in regional elections characterised by the “anti-incumbency” phenomenon – when voters express their dissatisfaction against the incumbent party by voting against it – the NDA retained the vast majority of seats obtained in 2014.The highest NDA scores remain concentrated in a few states of western and northern India, the coalition having in particular gained all the seats in a single regional block, stretching from Himachal Pradesh, Haryana and Uttarakhand in the northwest to Rajasthan and Gujarat.When studied through the lens of spatial analysis, the singular geographical impact of the NDA vote appears unmistakable.
- **Constituencies influencing each other:**The national index of spatial autocorrelation (Moran’s I,which measures the strength of similarities between adjacent observations), has reached 0.73: this means that the correlation between NDA votes in one constituency and those in the neighbouring constituencies is as high as 73%.This is a very strong level of spatial dependence compared to other social, religious or economic indicators, which confirms the pronounced geographic patterning of the NDA votes in 2019.This stable and regular distribution of voting behaviour contradicts the proverbial volatility of India’s regional politics last illustrated in 2018, when the BJP lost the local elections in Chhattisgarh, Madhya Pradesh and Rajasthan. This geographic structure also demonstrates that these spatial patterns owe less to the vagaries of local political coalitions and candidates than suggested by media reports.

- **Discontinuity:** NDA strength dropped significantly where the most pronounced discontinuity line corresponds to southern and eastern Karnataka, a state where the NDA recorded an almost flawless victory: its vote share abruptly falls from around 50% to less than 20% when one crosses the boundaries to Kerala, Andhra Pradesh and Telangana.The NDA share declines slightly less dramatically in Tamil Nadu, thanks to its local alliance with the local AIADMK. Similar steep declines in voting shares were seen in the northeast (Meghalaya, Mizoram and Sikkim) as well in the Muslim-dominated constituencies of Kashmir. The NDA percentage also drops by half when one enters Andhra Pradesh from Odisha or Punjab from neighbouring Rajasthan and Haryana.These cases of discontinuity contradict the otherwise high level of spatial autocorrelation highlighted earlier and points to the presence of strong regional parties that rejected any alliance with the BJP or to the presence of strong social heterogeneity along religious or ethnic lines.More broadly, this corresponds to vigorous regional political traditions away from the Hindi belt, the states where Hindi is used as lingua franca. In such areas, local parties, including the Congress and Communist parties fight against each other for local dominance and the BJP's nationalist and conservative agenda appears somewhat irrelevant to Hindu voters.
- **Eradication of a major opposition:** Congress was decimated to their lowest number of 44 wins after 2019 election results were announced.Yet, the Congress Party won around 22 percent of the vote in the 2019 elections, which, compared with the BJP's 38 percent share, is quite decent. There is clearly space in the Indian political spectrum for a center-left and non-Hindutva party, but whether Congress, or some new party can fulfill this role remains to be seen. Ideologically, and in terms of leadership, the Congress party turned out to be largely aimless. It has the structure, the history, and the presence in India to be competitive, but has yet to settle on how. The party's biggest dilemma is what to do about the Nehru-Gandhi family that has led it for four successive generations. On one hand, the Gandhi family holds the party back from becoming more meritocratic and open. Unlike Narendra Modi who rose from among the ranks of the BJP to rule India, a similar outcome for a young leader in Congress is impossible. Yet, on the other hand, the family is what holds the party together. Without such a center, and without the strong ideological commitment that the BJP's cadres have, it is quite conceivable that the Congress Party without splinter due to infighting. As it is, regional charismatic leaders have broken off from the Congress Party to form their own outfits, including some very successful ones in West Bengal and Andhra Pradesh.

3.4 Impact of social media in election campaigns and result prediction

Social media has become one of the deciding factors in the election scenarios all over the world. It has become a common affair for the political parties and the voters to express their opinions through social media. The social media data can be analysed to get these opinions in the form of sentiments in a post. This is termed as Sentiment Analysis.

3.4.1 Popularity of Twitter platform in election prediction

Twitter platform is a preferred choice in social media based sentiment analysis due to the restriction on the text content per tweet and hence an ease of analysis compared to other

platforms. Politics is increasingly becoming a hot topic of critical analysis in social media and Twitter hosts a huge collection of opinions in this domain. Sentiment Analysis plays a big role in these posts since it helps to decide the sentiment of people towards a specific political party or candidate and hence the success or failure probabilities. This is really helpful for the parties to gear up their election campaigns in a better way. This also helps people to grasp the sentiment which their fellow citizens feel for a party. Also, For instance, if anyone wants to predict any election results, the survey is conducted by going through a restricted group of individuals and asking their views on different electoral and their respective political parties. Further results are evaluated on the grounds of their opinions, and prediction of election results is done. All this process of the offline survey cost so much while considering manpower, time, and money. It can, therefore, be safely concluded that election prediction from social media platforms covers a varied and extensive population, and saves both time and men's efforts.

3.5 Role of Data Analytics & Machine Learning in Politics

As the role of technology becomes multi-fold in every sector, it generates huge amounts of information that can yield valuable insights about the field. This has led to a boom in the data industry in the last ten years.In an increasingly politically aware world, big data analytics is the "trump" card most organisations and movements have when planning election strategies. With great power comes great responsibility and data analytics is a powerful tool that can be utilised to harness and reap maximum benefits.

We often see politicians use big data analytics to optimise their campaigns. For example, experts and journalists have coined Trump's campaign a 'data machine' powered by AI capable of swinging voters, demonstrating the power of data and analytics systems.We also see politicians utilising such data to know where their possible voters could originate from, cross-reference them with the topics supported by the candidate and use the feedback to refine their policies.This is not a new phenomenon, since the 50's every party has used big data analytics to strategise their election campaigns.Let us look into some of the big data systems and applications that have allowed countries and groups to manipulate the results in their favour.

3.5.1 The first "big" data break:

The idea of using data in elections is not new. For example, the Kennedy administration used the "People Machine" to great success. It was, at the time, the largest such project ever conducted and it involved the use of massive data decades before "big data analytics" became a buzzword.It was during this time that the use of computer simulation, pattern detection and prediction for election campaigns began. Opinion poll data from the archives of pollsters, George Gallup and Elmo Roper, created a model of the US electorate. The information gathered was pivotal in creating relevant strategies and ensuring those votes were coming in. This was seen even further back in our history when data was collected to better understand the masses.

3.5.2 A little bit of data goes a long way

A British "global election management agency" gained global traction because it utilised advanced data analysis along with strategic communication during electoral processes, which proved to be successful. They started in 2013 as a satellite firm of the private intelligence company and self-described "global election management agency". They were essentially in the big data analytics business.This company used personal data to sway the outcome of the US 2016 presidential election and the UK Brexit referendum. But its reach extends well beyond the UK and US having supported more than 100 campaigns across five continents.

These data methods were used twice to help secure victory for Kenyan President Uhuru Kenyatta first in 2013, then again in 2017. Officially, the company's website boasts of doing in-depth research to uncover the issues driving voters in the country. They rebranded the entire party twice, wrote the manifesto, did research and finalised the messaging. They essentially created the platform, basing it on their findings derived from big data analytics to curate what the masses exactly wanted to see.

Jacinda Ardern , the Prime Minister in New Zealand utilised big data analytics to bridge the gap between her policies and her voters, which drew the constituents to her side and secured a second term. This highlights the importance of analytics in both elections and policy-making.

3.5.3 The future of big data analytics in elections

If current trends are any indication, future election campaigns will be further entrenched in data analysis methods so as to glean the best approaches, efforts and results. Data analytics will be used much after the campaigns as well, with it being an integral part of understanding and flagging problems plaguing different population sections. Data analytics has evolved itself to become the brain of every election campaign since the early 2000s. Data analytics helps the election campaign committee understand the voters better and adapt their policies to their sentiments, demonstrating the versatility of analytics platforms.

3.5.4 Big Data Analytics at Play in India's General Elections 2019

Political parties and their IT cells were leveraging big data analytics to reach out to the voters with appropriate messaging in General Elections 2019.In the general elections 2019 in India, technology made it easier for political parties to gauge the mood of the voters and put their best message forward. Political parties were leveraging big data analytics to gather insights into voter preferences based on their socio-economic status, caste, local issues, and various other parameters. Based on the voter sentiment and segments, customized election campaigns with the most relevant messages and videos were created and pitched to the specific target groups. Once the campaign was launched, data was then gathered to analyze its effectiveness and tweak it further according to the response it generates.

3.5.5 Creating Trending Narratives

Social media and WhatsApp emerged as key influencers and political parties made good use of these tools to reach out to the voters. Informal conversations between online users were enabling parties to pick up positive narratives and convert them into trending stories. The IT cells also monitored negative narratives and pushed counter stories in a bid to minimize the harm these negative sentiments could cause. Online browsing patterns of users were being tracked to pitch relevant political messages that were further fine-tuned according to the user engagement with these pitches. Depending on the uptake of the messages at a particular location, political parties also created narratives for location-specific rallies and the key messages they must deliver.

3.5.6 Deciding Candidature

Not just the campaign, but poll strategy and fielding candidates had also become scientific with big data guiding the parties here as well. Analysis of data from numerous apps—that provide insights into voter perception of the MPs and MLAs, and review of their work, gave an insight into the popularity (or otherwise) of a candidate, enabling parties to make informed decisions .

4. An Overview on Data Analysis & Visualization

The word "data" has become extremely important in the present era. We often come across data in its raw form irrespective of domain of application. It remains as an overwhelming heap of disparate information unless suitable processing is not performed on it. Data analysis is defined as a process of cleaning, transforming, and modeling of this raw data to extract useful information. This information can be directly used for decision making.The basic steps involved in data analysis cycle are shown in Figure 1.

Figure 1

The need of data analysis is the stage where we decide the objective for which we try to extract some information out of the data. It gives the problem definition. In the second stage, the database required to address the problem is collected. The data is collected from various primary and secondary sources and cannot be employed directly into the analysis process. It requires removal of unwanted/ redundant information, filling of some missing information with suitable approaches and looking for large deviations (outliers) in the dataset. This process is termed as Data cleaning and this makes the data ready for analysis.

The analysis of data is done in many ways such as data mining, predictive analysis and data visualization. Data visualization is a preferred method for situations where various parameters in a data is to be compared with each other or their relationship is to be studied. It provides insights which are otherwise impossible to read from a pile of data. Data visualization is done using tools like Tableau, Power BI, Google charts etc. In addition to this, there are standard built-in libraries in Python like seaborn and matplotlib which help in creating visualizations using python coding. Here, we aim to provide analysis & visualizations through both python coding and Tableau tool.

5. Introduction to Machine Learning

Machine learning is the science of getting computers to act without being explicitly programmed. It is a part of Artificial Intelligence(AI) . A typical Machine Learning Life Cycle has the following steps.

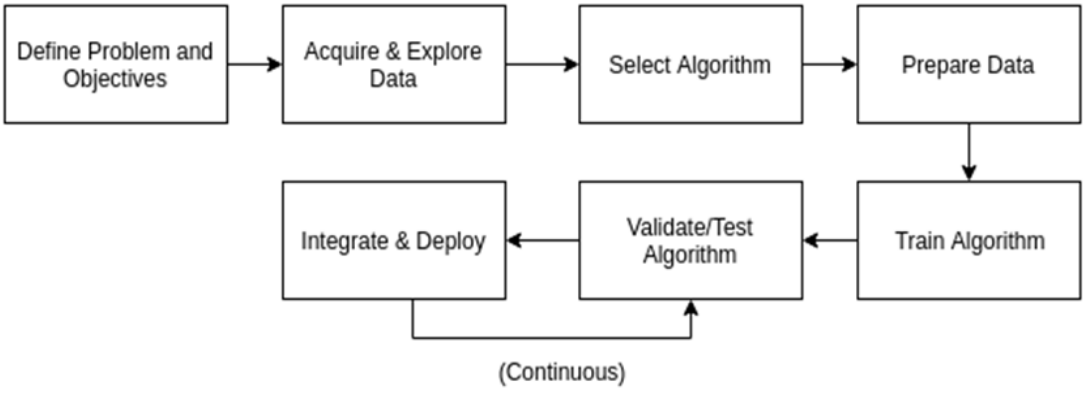


Figure 2 : Machine Learning Life Cycle

The problem definition involves stating the task at hand and the role of the machine learning model to solve it. The objectives involve stating the function of the model whether it is the regression or classification. Regression is used when we need to predict a continuous output variable and Classification is used to predict discrete valued outputs. Having decided the task of the project, a data source is identified or data is collected from surveys or repositories etc . Then, the algorithm is chosen. For example, if it is a Regression task, Linear Regression can be a good choice. For classification, we have algorithms like Logistic Regression, k-NN, Random Forest , k-means clustering etc. The datasets to be classified are often raw in nature and contain missing values, incorrect representations, outliers etc. Thus, suitable preprocessing and exploration is done so as to prepare the data for modelling. Once the data is cleaned, the necessary features are extracted and it is checked for a target variable. If there is a target variable and it is discrete we build a supervised classification model. Similarly there are unsupervised and reinforcement learning which do not involve target variables for training. Then the algorithm is trained and tested depending on the problem in hand. For small datasets, there comes an issue of overfitting when the model is trained on a given data for a large number of times. To avoid this, we use cross-validation, wherein, a part of training data is reserved for validation. Once the model is run on the training data, it is used to predict output based on test data as well as the validation data points. If the performance metrics are satisfactory in both cases, the model is said to be a better one. The various performance metrics are decided based on the type of problem in hand. If it is a Regression, we use Mean Squared Error and many forms of it. If it is a Classification, we go for Accuracy, Precision, F1-Score, Confusion Matrix etc.

5.1 Machine Learning Algorithms

Machine learning algorithms can be divided into three major types, which are supervised learning, unsupervised learning, and reinforcement learning.

5.1.1 Supervised Learning

It is a type of machine learning that will find a solution for the present problem based on data that we already have.To put it more technically, supervised learning is a type of machine learning algorithm in which a model or a function is developed based on input-output pairs in the training data to map the input from the test data to their respective output. In this, training data, also known as the training dataset, is a data bank of labelled data, and the test data, or the testing dataset, is the set of inputs with no labels.

Supervised learning can be further divided into two subtypes: regression and classification supervised learning. Regression refers to a model of supervised learning where all data is numerical (or continuous) values. In contrast, classification models use categorical values, which are usually binary values (0 or 1).Examples: Linear Regression, Logistic Regression, SVM, KNN, Naive Bayes.

5.1.2 Unsupervised Learning

It is a type of machine learning that uses inferences drawn about a dataset that is devoid of labels. The most common type of unsupervised learning is clustering which can be defined as the mechanism where hidden patterns or groupings in data are found during the process of exploratory data analysis. In other words, unsupervised learning algorithms can be thought of as a model that learns from the test data itself .Two main methods used in unsupervised learning include clustering and dimensionality reduction.A popular method of dimensionality reduction is called principal component analysis.

5.1.3 Reinforcement Learning

Reinforcement learning algorithms can be explained as the type of machine learning model where tasks are performed by an agent in a particular environment. In this model, the agent either receives a reward or punishment for each task performed. As the name suggests, it is a process of continuous improvements based on some rules. Unlike other machine learning approaches the algorithm is not told how to perform a task but learns by itself. Such models learn with experience and improve each time after committing an error. Examples: Markov Decision Process and Q Learning .

In this project we are performing “Classification”. Some of the popular Supervised Learning algorithms for Classification which are used in this project are explained below.

- *Logistic Regression Model*- In Logistic regression, it is used to model the probability of a finite number of outcomes, typically two. In essence, a logistic equation is created in such a way that the output values can only be between 0 and 1.
- *k Nearest Neighbors Model(kNN)*-K Nearest Neighbours is a basic algorithm that stores all the available and predicts the classification of unlabelled data based on a similarity measure.Like linear geometry when two parameters are plotted on the 2D Cartesian system and we identify the similarity measure by calculating the distance between the points, the same applies here, KNN algorithm works on the assumption that similar things exist in proximity, simply we can put into the same things stay close to each other.
- *Decision Tree Method*- This is an ensemble method. Decision Tree is a tree-like graph where sorting starts from the root node to the leaf node until the target is achieved. It is the most popular one for decision and classification based on supervised algorithms. It is constructed by recursive partitioning where each node acts as a test case for some attributes and each edge, deriving from the node, is a possible answer in the test case. Both the root and leaf nodes are two entities of the algorithm. Decision Tree Analysis is done via an algorithmic approach where a data set is split into subsets as per conditions. The name itself says it is a tree-like model in the form of if-then-else statements. The deeper is the tree and more are the nodes, the better is the model.
- *Random Forest Algorithm*-The random forest algorithm is based on supervised learning. It can be used for both regression and classification problems. It can be viewed as a collection of multiple decision trees algorithms with random sampling. Random forest is a combination of Breiman's "bagging" idea and random selection of features. The idea is to make the prediction precise by taking the average or mode of the output of multiple decision trees. The greater the number of decision trees is considered the more precise output will be. It also comes under ensemble methods.
- *Gradient Boosting Methods*- This is also an ensemble method which performs Boosting, a special type of Ensemble Learning technique that works by combining several predictors with poor accuracy into a model with strong accuracy . This works by each model paying attention to its predecessor's mistakes.There is an improved version to this which is the *XGBoost*
- *Naive-Bayes Classifier*- This is a probabilistic classifier which is used when each of the features are independent. The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

5.2 Evaluation Process & Prediction

The predicted values of output for the test data are obtained and the efficiency of classification is studied in terms of different performance metrics. The performance parameters to evaluate the classifier models are;

- Accuracy score- It is the ratio of number of correct predictions to the total number of input samples. It works well only if there are an equal number of samples belonging to each class.If you are working on a classification problem, the best score is 100% accuracy. If you are working on a regression problem, the best score is 0.0 error.

$$\text{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{(True Positive + False Positive + True Negative + False Negative)}}$$

- Precision score- It is the number of correct positive results divided by the number of positive results predicted by the classifier.In the simplest terms, Precision is the ratio between the True Positives and all the Positives.

$$Precision = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Positive(FP)}$$

- Recall score -It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Negative(FN)}$$

- F1 score- F1 Score is used to measure a test's accuracy. It is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).The greater the F1 Score, the better is the performance of our model.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- ROC AUC curve & score- A ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters ; True Positive Rate & False Positive Rate.

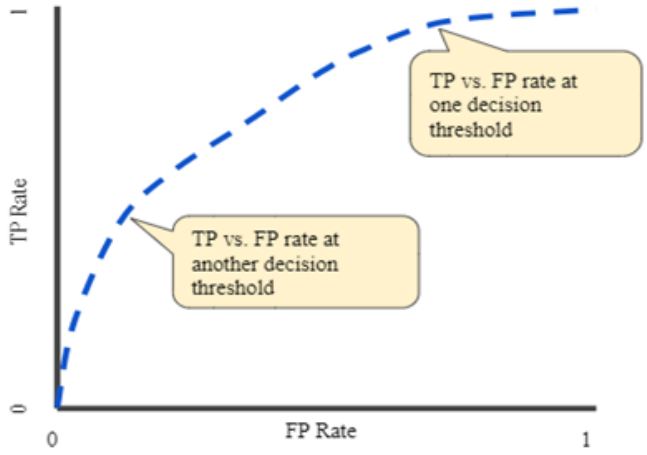


Figure 3 : ROC-AUC Curve

AUC stands for "Area under the ROC Curve." AUC measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1).AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. The curve plots the False positive rate versus True Positive Rate. A perfect classifier model is one whose ROC curve moves towards (0,1) coordinate in the plane.

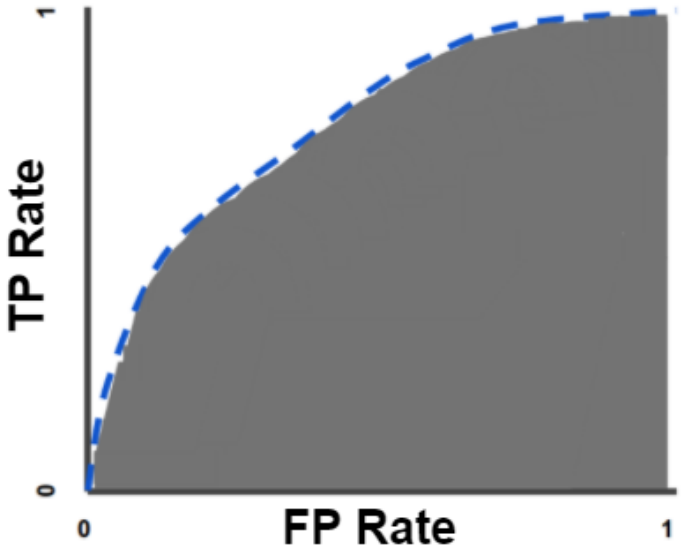


Figure 4 Area under ROC Curve

5.3 Cross Validation Techniques

Cross Validation is a technique which involves reserving a particular sample of a dataset on which is not used to train the model. Later, we test the model on this sample before finalizing it. The major steps involved in cross validation are

1. Reserve a sample data set
2. Train the model using the remaining part of the dataset
3. Use the reserve sample of the test (validation) set. If the model delivers a positive result on validation data, we can fix the current model.

There are many types of cross-validation techniques based on the size of dataset and application involved.

5.3.1 Validation set approach

In this approach, we reserve 50% of the dataset for validation and the remaining 50% for model training. A major drawback with this approach is that, since we are training a model on only 50% of the dataset, there is a huge possibility that we might miss out on some interesting information about the data leading to higher bias.

5.3.2 Leave one out cross validation (LOOCV)

In this approach, we reserve only one data point from the available dataset, and train the model on the rest of the data. This process iterates for each data point. Here, we make use of all data points, hence the bias will be low. Also, we repeat the cross validation process n times ,where n is the number of data points. This results in a higher execution time and is not suitable for large datasets.In addition to this, the effectiveness of the model relies on individual data points and it turns out to be an outlier, it can lead to a higher variation. Thus, this approach is desirable in datasets which are small, balanced and outliers have been removed.

5.3.3 k-fold cross validation

This is a better form of cross-validation in datasets which are small and medium sized. Here, we randomly split the entire dataset into k"folds".Then, for each k-fold in the dataset, train the model on k – 1 folds of the dataset. Then, test the model to check the effectiveness for *kth* fold. Record the error on each of the predictions and repeat this until each of the k-folds has served as the test set. The average of your k recorded errors is called the cross-validation error and will serve as the performance metric for the model. Here, the cross validation score serves as the important metric for the model.

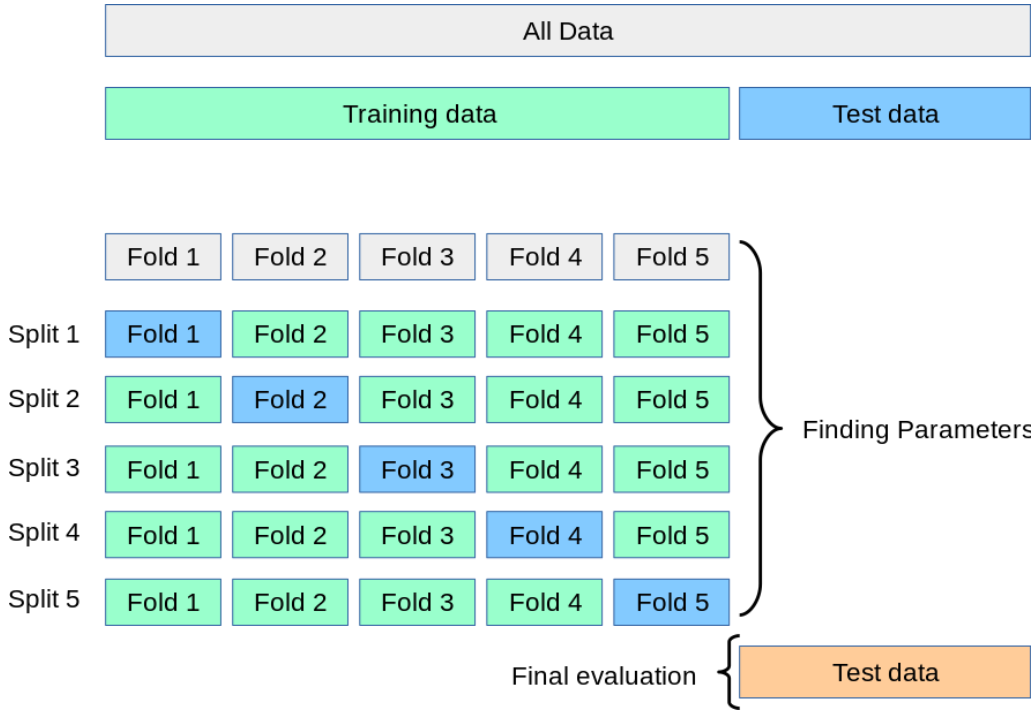


Figure 5: k-Fold Cross Validation

Once the model is validated, it is ready to be deployed. The validation process continues even after the deployment since the model is exposed to new data each time.

5.4 Deployment of Model

Deployment of machine learning models, or putting models into production, means making the models available to other systems within the organization or the web, so that they can receive data and return their predictions. Through the deployment of machine learning models, we can improve the model and expose it to newer data.

Deployment can be done using different dedicated platforms or developing a webpage by ourselves and deploying it using suitable Python web frameworks like Flask.

6. Introduction to Sentiment Analysis

6.1 An Overview

Sentiment analysis is the field of study that analyzes people's outlook in the form of their opinions, attitude, emotions etc towards any event or product. It was mainly used in market research,brand monitoring and hence to analyze the growth of businesses. It is closely related to opinion mining and emotional mining and is interchangeably used in academic fields.The term sentiment analysis is well defined in a simplified manner in [2].

The basic difference in sentiment and opinion mining is ; opinion mining considers the view point or judgement about a particular subject or event, whereas sentiment analysis considers the sentiment which created that view point. As an example, when a particular cosmetic brand gains popularity among its customers, their positive remarks towards it constitutes the opinions and the feeling such as happiness, satisfaction etc constitutes the sentiment towards it. In research or academic cases, we normally perform a sentiment analysis on a given event or product by classifying the opinions in written text as positive, negative or neutral sentiments. Refer to Figure 1 for an overview of sentiment analysis.

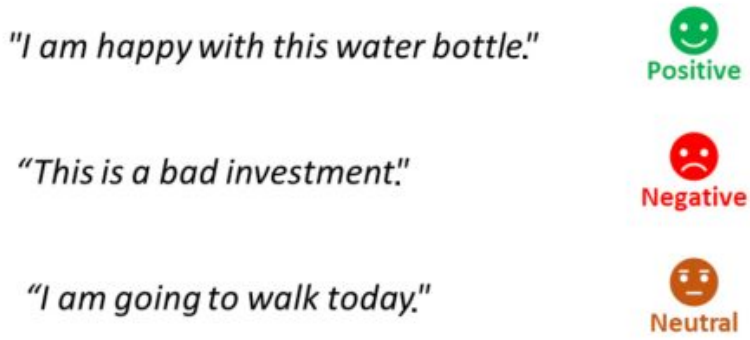


Figure 6 : Sentiment Polarity in text

In practice, a sentiment analysis system for text analysis combines natural language processing(NLP) and machine learning techniques. It assigns weighted sentiment scores to the different entities within a sentence or phrase and then these scores are aggregated to predict an effective score or sentiment for the statement. This field is very challenging since the sentiment scores given to each of the words decide the overall positivity or negativity of the statement.These words are manually added to the library of words and are updated with each synonym word appearing in subsequent texts. The choice of words is domain specific and hence requires a strong domain knowledge to decide on the words which describe a particular emotion. Also, opinionated data is appearing not just in business but through social media on diverse fields. These opinions often come with sentiments which require analysis to determine the

emotion behind it. Hence, opinion mining and sentiment analysis are strongly related to each other. The sentiment analysis and opinion mining are highly relevant as the scale of unstructured data is getting higher day by day. A deep coverage of sentiment analysis and opinion mining can be seen in [9].

6.2 Levels of Sentiment Analysis

The sentiment analysis of a given textual material can be performed at three levels as follows [4].

a) Document level Analysis
b) Sentence Level Analysis
c) Entity and Aspect Level Analysis

In Document level analysis, it is checked whether the whole opinion document expresses a positive or negative sentiment. This level of analysis assumes that each document expresses opinions on a single entity and is made by a single person. This does not take into account the minute insights or sentiments lying at a sentence level or a word level. Hence, this is not preferred in analysis where there are multiple and mixed sentiments in opinions. An example for document level analysis is shown in Figure 2 [5].

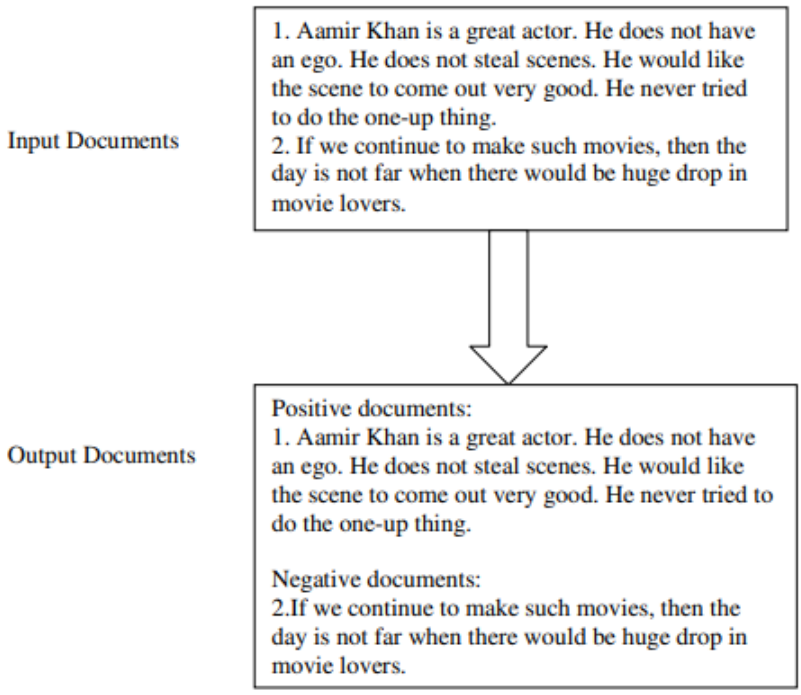


Figure 7 : Document Level Analysis

In sentence level analysis, the sentiment in each sentence is considered to decide whether it expresses any opinion or not and also on the polarity of those opinions. It thus relates to subjectivity classification. An example for sentence level analysis is as follows.

We like to choose this brand since it is affordable - Positive

I don't like this brand of clothing owing to its high price - Negative

I am not sure of the brand of clothing - Neutral

In entity and aspect level, the importance is given to the entities or aspects which affect the product. An example for this analysis and the polarity classification is shown in Figure 3. [6]

Opinion	Aspect	Sentiment
"It's so easy to use. It looks less than a week to understand where everything is in Drift"	UX-UI	Positive
"The mobile app can be really glitchy and is definitely not user friendly"	Mobile App	Negative
"Their customer success team is amazing and there's always someone available from thei support team on live chat to help you"	Customer Service	Positive

Figure 8 : Entity & Aspect Based Analysis

6.3 Sentiment Analysis Process and Algorithms

The various steps involved in the sentiment analysis process is summarised in the Figure below [7]. The different stages are

a) Data Collection and Preprocessing
b) Sentiment Identification
c) Feature Selection and Extraction
d) Sentiment Classification
e) Polarity Report

Data Collection & Preprocessing- The data from Twitter can be collected using the Application Programming Interface (API) that can be done in two ways:the first approach is to collect tweets at once which is termed as Representational State Transfer (REST) API. and the second approach is do a continuous collection of real-time tweets for a specific period, termed as streaming API. The data preprocessing is a crucial step in the sentiment analysis.Pre-processing involves cleaning and filtering of data. The data obtained from any social media platform include noisy information like misspelled words, slang words, user-generated abbreviations, and white spaces. Using the data without cleaning these unwanted elements, can degrade the accuracy of a sentiment classifier approach. The major processing done are as follows.[3] and is summarized in Figure 4.

- ***Stopwords Removal*** **-**Sentence may contain pronouns (he, she, It), nouns, verbs, adjectives, articles (a, the), prepositions (here, near) which don't make any sense. Hence these can be removed from the text.
- ***Stemming -*** It is the removal of prefix and suffix based on a given root word. For example 'playing', 'played' are modified words of the root word "play". These suffixes are not important for the analysis and can be replaced by the root word.
- ***Tokenization-*** It is also referred to as chopping. It is used to break down the sequence of text sentences to tokens, phrases or words.
- ***Negation Handling*** -Consider example "The brand is not so good" where, both the positive and negative words have occurred together but the polarity of sentence in total indicates negation. In some cases, the polarity of the whole sentence may need to be considered negative or only a part of the sentence which comes within the scope of the negation words. This is decided based on various algorithms.

Sentiment Identification - This step involves finding opinionated words or phrases to discover sentiments among the given data. This may also classify subjective and objective text to perform analysis of the subjective text only and discard the objective text because subjective sentences are rich in opinions and sentiment related words. In some cases, only subjective text is not enough for the sentiment analysis, there may be relevant objective text which can provide accurate insights.

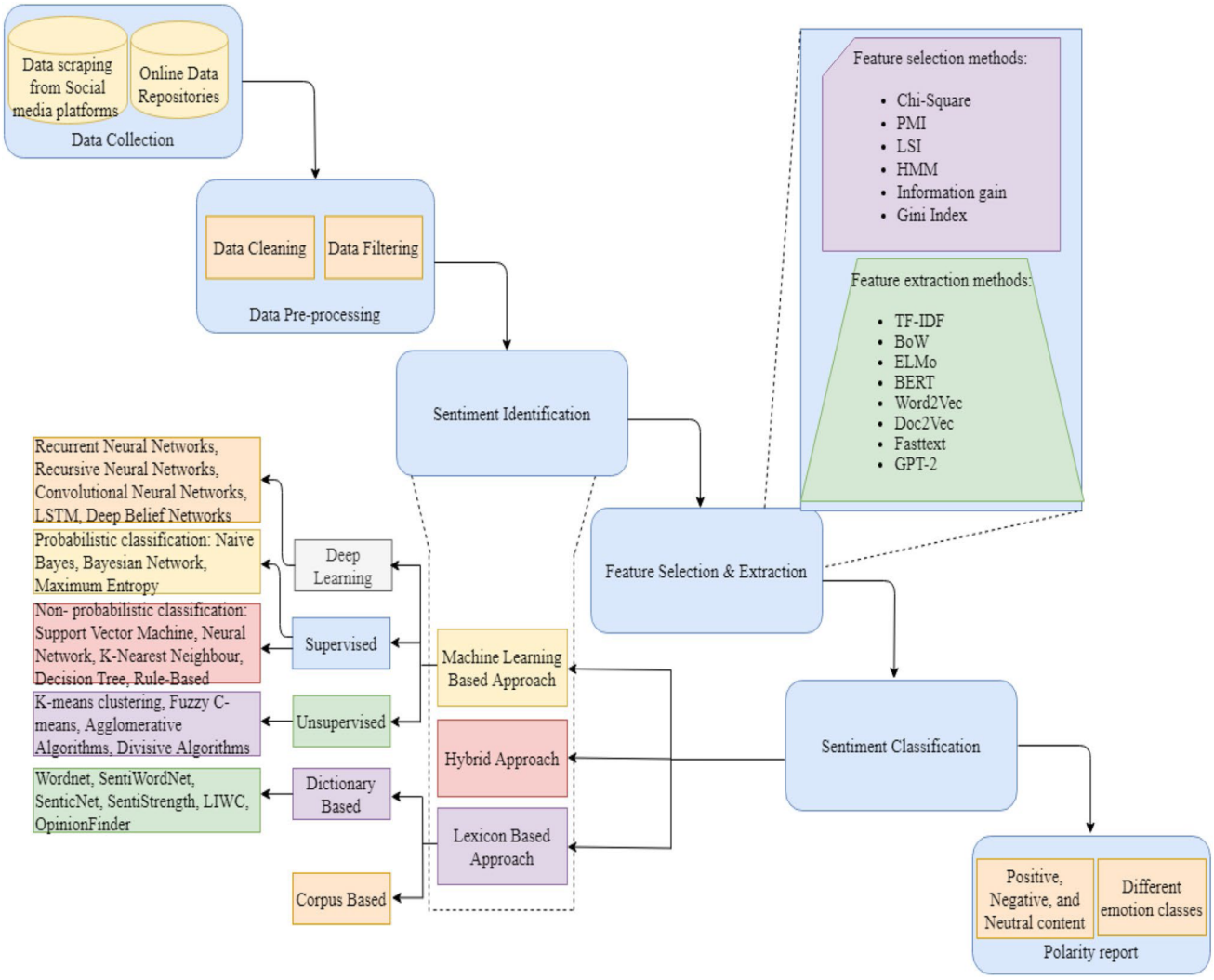


Figure 9 : Sentiment Analysis Stages

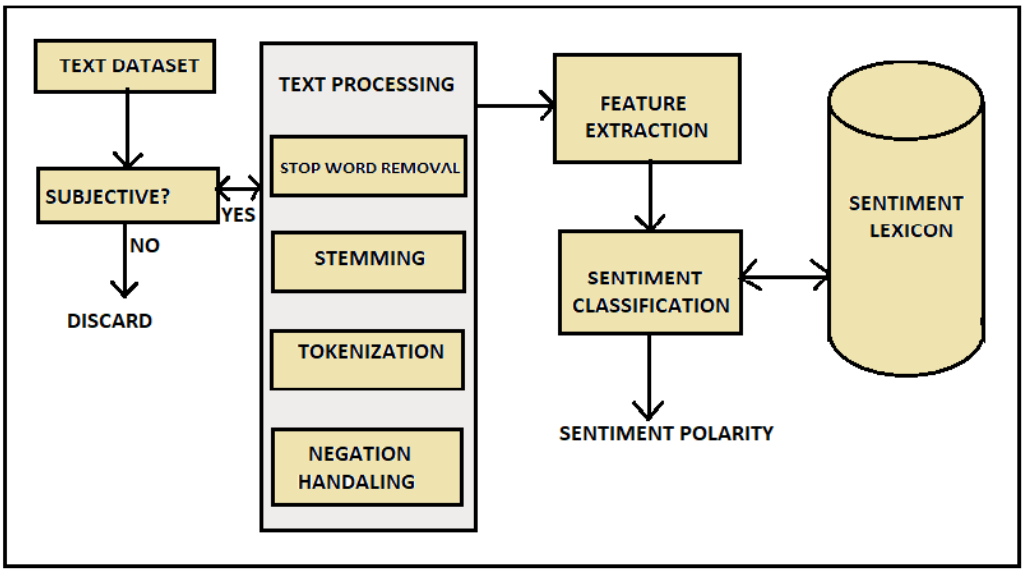


Figure 10 - Sentiment Analysis Stages-Simplified View

Feature Extraction and Simplification- A feature in text plays a very important role in sentiment classification and the performance of the model degrades if a crucial feature is eliminated. Textual features can be of two categories- syntactic features and semantic features. The most frequently used features such as unigrams, bigrams, n-grams, term's frequencies (TF-IDF), Part of Speech (PoS) tags (nouns, verbs, adjective, adverbs, etc.), and dependency trees are the syntactic features of a text. The features of a text that reveal positive or negative sentiments like opinion or sentiment words/phrases, semantic concept, and negation are termed as semantic features.

Features of the given text are selected using various techniques such as Chi square, frequency-based, Latent Semantic Indexing (LSI), Mutual Information (MI), Hidden Markov Model (HMM), Latent Dirichlet Allocation (LDA), Weight by correlation, Information gain, and Gini index etc. Feature extraction is a very crucial step in sentiment analysis since machine learning models can handle only numeric data. In this step, the given text is converted into a feature vector which comprises the most relevant features for testing and training the machine model. The vector representation of features is then used to classify the polarity of a text. The different feature extraction methods are count vectorizer, Term Frequency-Inverse Document Frequency (TF-IDF) ,Bag of Words (BOW) using n-grams, and Feature hashing. The method is chosen based on the application. Nowadays, deep learning is increasingly becoming popular in feature extraction.

Sentiment Classification

This stage involves deriving the sentiments out of the extracted features and classifying them as positive, negative or neutral categories. More labels can be introduced based on the application requirement. There are basically three approaches for sentiment classification which are discussed as follows [7].

a) Lexicon based approach
b) Machine Learning based approach
c) Hybrid approach

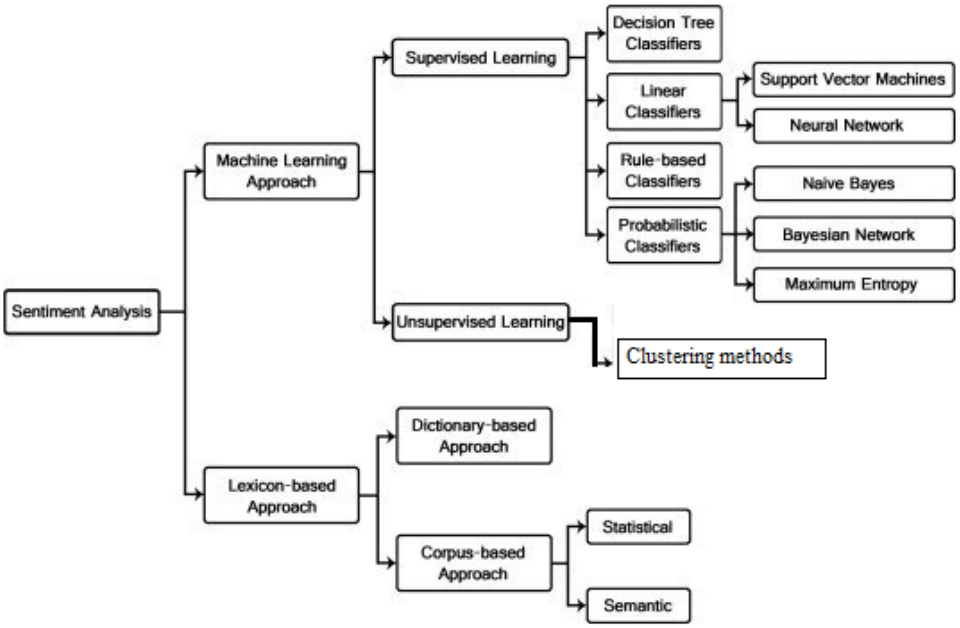


Figure 11 : Sentiment Classification Methods

Lexicon based approach- This is also known as rule based approach. Lexicon implies a complete set of meaningful units in a language also referred to as the vocabulary. Opinion lexicons are used to define the state of positive and negative text using a set of words. The polarity score of positive and negative words in a sentence or document is added to give the total sentiment score of that sentence or document. There are two methods under lexicon based approach as shown in Figure 6.

In dictionary based approach, the opinion words are formed using a blend of manual and automated practices using standard lexicon libraries which consist of words with their sentiment scores. Some popular libraries are Linguistic Inquiry and Word Count (LIWC) WordNet, SentiWordNet (Esuli et al. 2006), SenticNet,SentiStrength, OpinionFinder etc. The main disadvantage of the dictionary-based method is that it is domain and context-independent and it

is hard to use the dictionary-based methods to find the domain or context dependent orientations of sentiment words.

A corpus-based method is domain and context-specific. However, orientations of domain-specific sentiment words are useful, but sometimes it is not always correct because many words in the same domain can have different meanings in different contexts. sing the Lexicon method alone has practical limitations in terms of the number of words under one domain.

Machine Learning based approach- This approach uses a machine-learning technique and diverse features to construct a classifier that can identify text with sentiment. The features extracted from the text in numeric form are classified using the model. Nowadays, deep-learning methods are getting popular compared to traditional machine learning. The machine learning approach can be broadly categorized into supervised and unsupervised models, which contain respective models as shown in Figure 6 [9].

Hybrid Approach- This is a blend of both lexicon based and machine learning methods to perform sentiment analysis and classification to reap the benefits of both the approaches.

A detailed coverage of these methods can be seen in [7] and the associated references in [7].

Polarity Report

The result of sentiment analysis is in the form of polarity scores which are based on positive, negative and neutral sentiments. There have been many refinements in mining sentiments to obtain scores in many previous works.

6.4. Twitter Sentiment Analysis in Politics

The popularity of Twitter platform in political scenarios is evident from the exponential growth in the Twitter subscribers during elections. As an example in India, Twitter content related to Lok Sabha elections recorded around 583 per cent growth between 2014 and 2019. Although, majority of people sharing their opinions on Twitter are from urban backgrounds or in places where there is internet connectivity and people are indulging in social media, this represents the opinion of the whole country in many ways.

There are various approaches like the volumetric approach, sentiment approach, and social network analysis approach that can predict election results through social media.The volumetric approach, considers the volume of online followers, likes, and posts shared about an individual party or electoral for predicting election results. In the sentiment analysis approach, an aggregate of positive and negative sentiments of online posts about an electoral or political party is considered for election prediction. In the social network analysis approach, the networks of social media users who are supporting or discussing a couple of electoral or political parties are analyzed.

7. Python Libraries for Analysis and Modeling

Python is a preferred choice for data analytics, machine learning , sentiment analysis or mining owing to its extensive collection of libraries. The Python libraries used in this project are described as follows.

7.1 Python Libraries for Data Analytics & Machine Learning

7.1.1 Numpy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

We use numpy in many situations such as for statistical uses and array operations. It is one of the major libraries used in machine learning using python.

7.1.2 Pandas

Pandas is mainly used for data analysis. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL, Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features.

Mainly all the operations we do on data is using pandas. The pandas dataframe is used for our common machine learning and data analysis tasks.

7.1.3 Matplotlib

Matplotlib is a visualisation library we use for plotting graphs and explaining the data.

It has a large built in functions for plotting a variety a\of graphs including bar plots, pie charts, scatter plots, histograms etc.

7.1.4 Seaborn

Seaborn is also a powerful visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Seaborn uses fewer syntax and has stunning default themes and Matplotlib is more easily customizable through accessing the classes.

7.2 Python Libraries for Sentiment Analysis

7.2.1 NLTK

NLTK (Natural Language Toolkit) is a suite that contains libraries and programs for statistical language processing. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language.

NLTK contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.

The main modules we use from NLTK is:

- corpus : standardized interfaces to corpora and lexicons
- tokenize, stem : tokenizers, sentence tokenizers, stemmers

7.2.2 VADER

VADER (Valence Aware Dictionary for Sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It is available in the NLTK package and can be applied directly to unlabeled text data.

This approach relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores. The sentiment score of a text can be obtained by summing up the intensity of each word in the text.

VADER is intelligent enough to understand the basic context of the words. For example VADER can easily grasp the underlying emotions of capitalised words and punctuations used in a text.

VADER's `SentimentIntensityAnalyzer()` takes in a string and returns a dictionary of scores in each of four categories:

- Negative
- Neutral
- Positive

- Compound (Normalized scores of above)

7.2.3 Textblob

TextBlob is built on the shoulders of NLTK. It is easy to learn and offers a lot of features like sentiment analysis, pos-tagging, noun phrase extraction, etc.

Major NLP tasks using TextBlob

- Tokenization refers to dividing text or a sentence into a sequence of tokens, which roughly correspond to "words". This is one of the basic tasks of NLP.
- Sentiment analysis is basically the process of determining the attitude or the emotion of the writer, i.e., whether it is positive or negative or neutral.
- Spelling correction is a cool feature which TextBlob offers, which can be accessed using the correct( ) function.

8. Literature Survey

The literature survey covers the different work done in Indian election based data. In this project , we aim to perform sentiment analysis and prediction and compare its performance with conventional supervised learning. So, the literary works on sentiment analysis are considered and there is a specific focus on sentiment analysis in the Indian context.

There is extensive literature on the Twitter sentiment analysis in politics. The political tweets are challenging in nature since they may have sarcastic but negative comments which makes classification a daunting task. Here, we focus on three recent works done in Indian Elections as it is strikingly different from other countries on the count of parties, voting pattern etc.

Karthik Singal et al.,2015 performed a political sentiment analysis and result prediction using a combination of lexicon and machine learning approach. They followed an unsupervised learning model. Here, a hybrid approach enabled them to analyze words related to other words, thus giving overall sentiment of the sentence. For lexicon, SentiWordNet was used to give the sentiment scores of a word. A negative score signifies negative connotation and a positive score signifies positive connotation of the word. Tweets were manually downloaded from a time period of 28 February 2014 to 28 March 2014. Although a longer time period was chosen, only a small sample of around 256 tweets were used for analysis. This may not be a representative sample in all cases.

Ankita Sharma et al., 2020 performed to gain the opinion polarity of the folks concerning general elections held in India. Two candidates were considered for this study: Candidate-1 and Candidate-2. It was concluded that Candidate-1 is more liked compared to Candidate -2. In this paper, the tweets were collected over a period of Jan 2019 to March 2019 before the elections. Sentiment Analysis operator of the AYLIEN was used for performing opinion mining on tweets related to both the candidates which is a lexicon based approach. One good feature about the paper was, the opinions for Candidate-1(Modi) and Candidate-2 (Rahul Gandhi) were expressed through different words representing the emotions. They generated an estimate of positive,negative and neutral sentiments in terms of the pie charts. But this approach will not give good performance when there are confusing statements where the overall sentiment is different from the individual words and also there are practical limitations to the domain specific words. So, a hybrid approach is a better choice.

Ferdin Joe John Joseph(2020) in his paper proposes a methodology to predict the outcome of the 2019 Indian general elections using the sentiment analysis of twitter data. Decision tree classifier is used to train and test data and the predicted outcome is found to be close to that of the actual outcome and most of the pre poll analysis done so far. The experiments reported in the paper are only on tweets in English language and having the most number of retweets by the users. Every day during a particular time, 5000 tweets each for the ruling and opposition parties' most famous twitter handles were extracted. Here, a hybrid approach is presented, where a Decision Tree

Classifier is used to implement the machine learning and Textblob is used in lexicon based sentiment classification.

In this project we present a hybrid approach where we try to evaluate the performance of the machine learning as well the lexicon methods. We aim to compare the popular algorithms and arrive at a suitable sentiment based classification method.

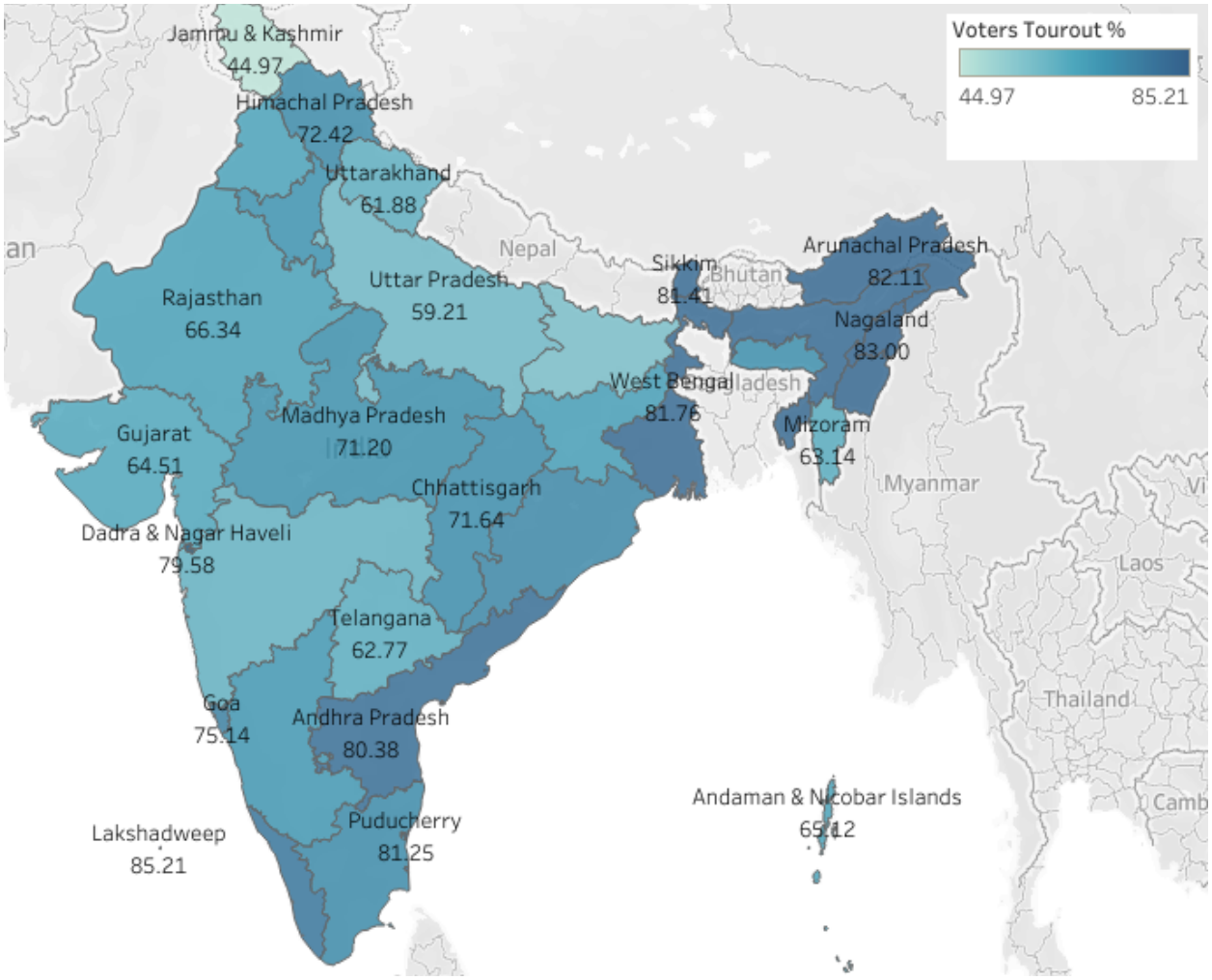
9. Indian General Elections 2019- Result Analysis

9. 1 General Overview

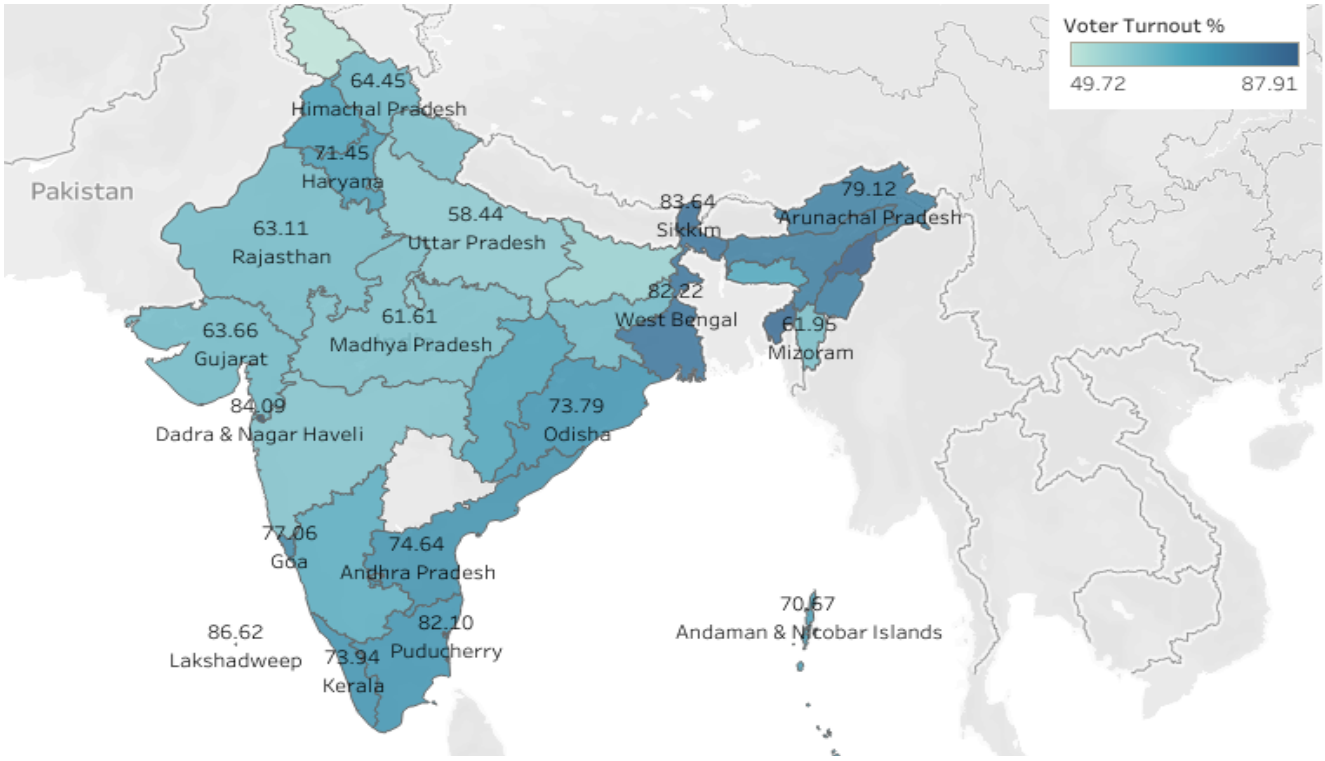
The general elections held in the year 2019 saw an increase in the voter turnout compared to the previous years. The main motto of the election commission was to ensure that "no voter is left behind". An overview of the elections is picturised below(Used from Election Commission reports)

Figure 12 : General Election 2019-Summary

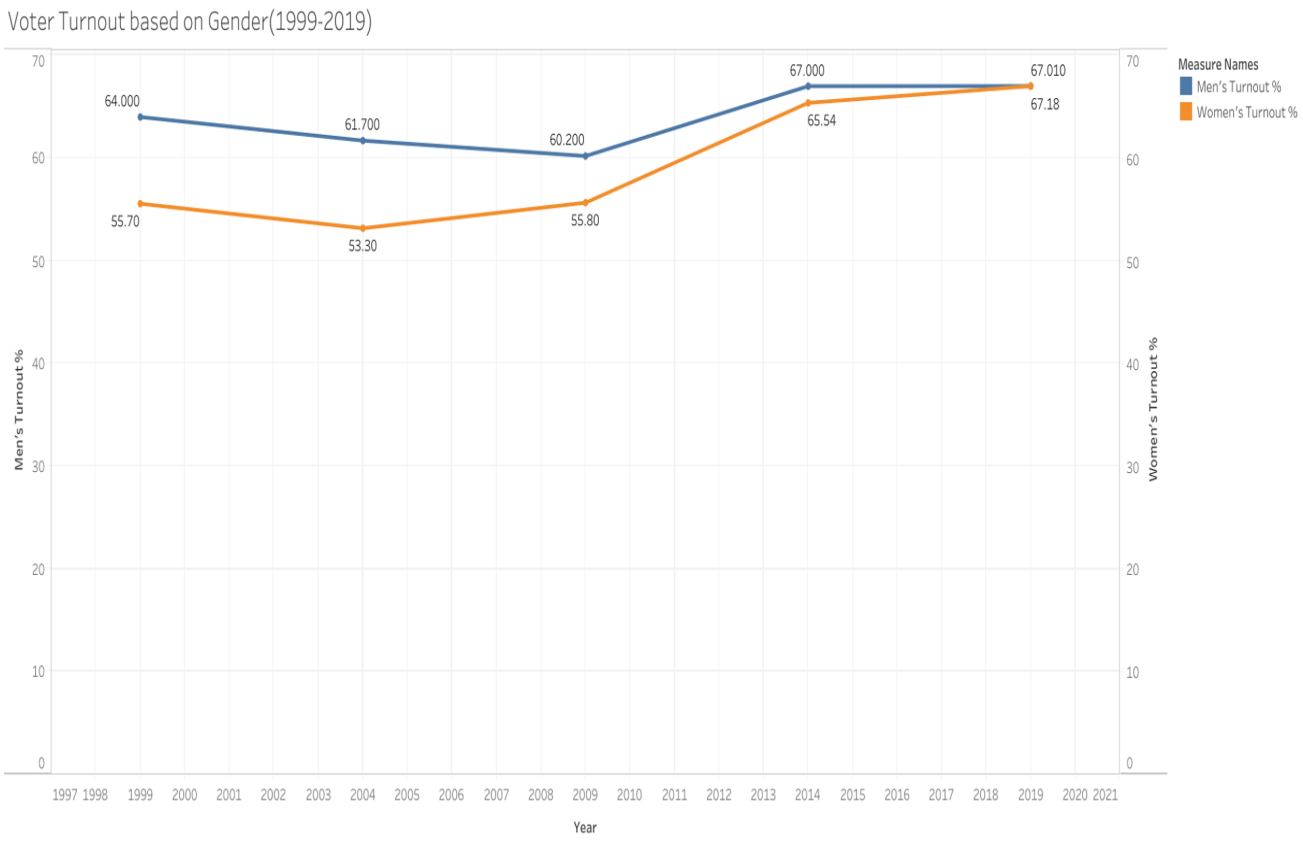
This was the 17th Lok Sabha election and it was held in seven phases. It became a historic one owing to the highest voter turnout over the previous years. It showed more than 1 % increase in the net voter turnout which is a great achievement as a democracy is concerned. A map of state wise percentage turnout is as depicted below in Figure 2. The light colored regions show lower voter turnout percentages and moving towards darker shade indicate the states with higher turnout percentages.



We can see that the eastern and southern states have shown high voter turnout percentages compared to the rest of the country. The lowest turnout percentage was reported as 44.9 % , in Jammu & Kashmir. The highest turnout percentage was reported in the Union Territory of Lakshadweep, 85.21 %. This is the first general election held in the newly formed state Telangana and it showed a turnout percentage of 62 % which is less than the national average score. The trend of polling has not changed very significantly from the previous general elections held in 2014 as observed in the figure below.

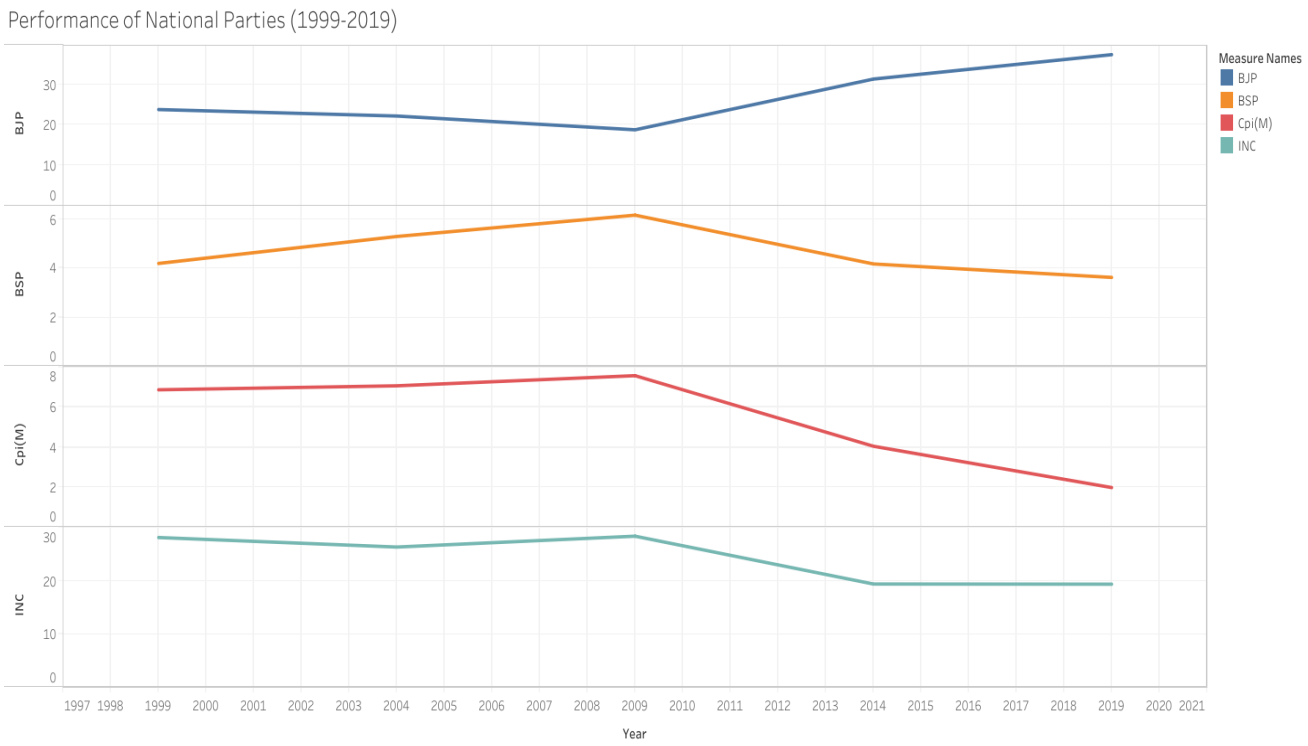


Another positive aspect was the increase in the women voter turnout. There has been a steady increase in the women voters casting their votes. The year 2019 marked the election with almost equal turnout among both genders. This can be observed from the gender wise voter turnout over the past two decades.

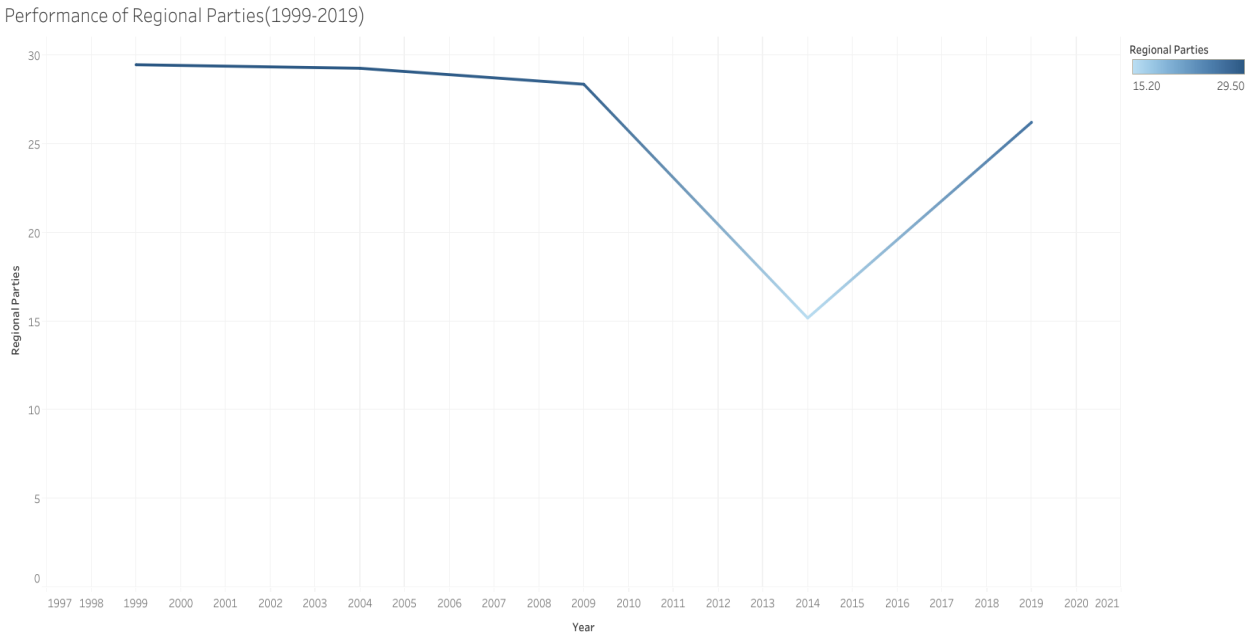


The next crucial factor to consider is the performance of national and regional parties in the past two decades. The elections in 2019 saw the Congress party being badly hit by the verdict. The steady decline of the Congress era and the emergence of many regional parties can be summarised in the Figures below.

Among the national parties, BJP attained popularity from the year 2009 and they remained unbeaten in the current elections. The other prominent national parties like Indian National Congress(INC), Bahujan Samajwadi Party(BSP) and CPI(M) faced a pathetic decline over the period as evident from the graph.



The role of regional parties got evident in recent elections as observed in the graph below. They showed a steady growth in performance especially from the year 2014, though there was a sharp decline before that period.



9.2 The Dataset

We have considered the dataset consisting of the results of general elections held in the year 2019. There are 18 columns in the dataset . It consists of 2263 entries corresponding to the candidates who contested in the elections from various constituencies throughout the country. It consists of the winners as well as the losers of the battle.

The data analysis process for the given election results can be performed using the following steps in the python environment.

9.2.1 Choice of platform and programming language

We have used the Google Colaboratory platform for executing the programming instructions for the dataset analysis. Google Colaboratory is a free online cloud-based Jupyter notebook environment that allows us to train our machine learning and deep learning models on CPUs, GPUs, and TPUs. It's an incredible online browser-based platform that does not involve any costs. It can work with large datasets and build complex models. It is possible to share the work seamlessly with others.

Python is used for coding the dataset. Python is a popular choice due to the ease in learning, numerous libraries inside it which offer a wide range of functionalities etc. There also exists a supportive community to offer clarifications in coding using python.

9.2.2 Importing Libraries

The major python libraries used in the dataset are numpy, pandas, matplotlib, seaborn and scikit learn. They are imported to the python environment. In addition to this, we have used ipwidgets library to enable interactive visualizations.

9.2.3 Loading the data

The dataset is in comma separated value (CSV) format and is loaded into the python environment using appropriate commands in the pandas library. Once the dataset is loaded into the python environment, it is converted into a dataframe using the pandas library command. It is a two-dimensional ,size mutable heterogenous data structure which allows for various manipulations in the dataset easily. There are 18 columns in the dataset. The columns are as follows. This can be observed using the .columns command in python.

```
Index(['STATE', 'CONSTITUENCY', 'NAME', 'WINNER', 'PARTY', 'SYMBOL',
'GENDER','CRIMINAL\nCASES', 'AGE', 'CATEGORY', 'EDUCATION', 'ASSETS',
'LIABILITIES',  'GENERAL\nVOTES',  'POSTAL\nVOTES',  'TOTAL\nVOTES',
'OVER TOTAL ELECTORS \nIN CONSTITUENCY','OVER TOTAL VOTES POLLED \nIN
CONSTITUENCY', 'TOTAL ELECTORS'],dtype='object')
```

It can be seen that most of the columns have values having object data type. The WINNER column represents the target variable in the dataset and the remaining variables are predictors. Now, the dataset needs to be explored to find out features which are relevant. Also some basic preprocessing is also required, which can be explained as follows.

9.2.4 Data Preprocessing

The data preprocessing in this dataset for generating analysis and visualizations involved the following steps.

a) renaming columns into proper and readable names
b) dropping entries in GENDER column with null values
c) replacing unavailable or NaN values in ASSETS, LIABILITIES and CRIMINAL_CASES using 0.
d) changing the datatype of ASSETS, LIABILITIES and CRIMINAL_CASES to numeric type.

Once the preprocessing stage is over, the exploratory data analysis is performed.

9.2.5 Exploratory Data Analysis

The first few entries of the dataset after preprocessing appears as follows. This is generated using the .describe( ) function in python.

	CRIMINAL_CASES	AGE	ASSETS	LIABILITIES	GENERAL_VOTES	POSTAL_VOTES	TOTAL_VOTES
0	52	52.0	3099414.0	231450.0	376892	482	377374
1	0	54.0	18477888.0	847000.0	318665	149	318814
2	3	52.0	36491000.0	15300000.0	314057	181	314238
4	5	58.0	74274036.0	8606522.0	644459	2416	646875
5	0	47.0	133784385.0	22251891.0	434199	1130	435329

OVER_TOTAL_ELECTORS_IN_CONSTITUENCY	OVER_TOTAL_VOTES_POLLED_IN_CONSTITUENCY	TOTAL_ELECTORS	WINNER
25.330684	35.468248	1489790	1
21.399929	29.964370	1489790	0
21.092771	29.534285	1489790	0
33.383823	56.464615	1937690	1
22.466390	37.999125	1937690	0

The total number of rows and columns can be found from the .shape( ) function which gives the output as follows.

```
(2263, 19)
```

The statistical description of the dataset can be obtained using the .describe( ) function as below.

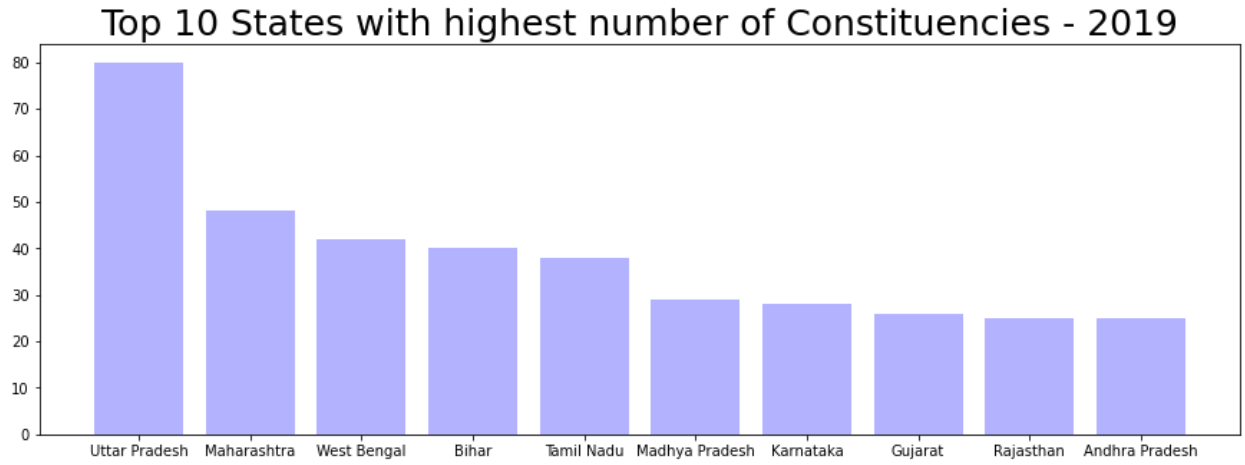
	CRIMINAL_CASES	AGE	ASSETS	LIABILITIES	GENERAL_VOTES	POSTAL_VOTES	TOTAL_VOTES	OVER_TOTAL_ELECTORS_IN_CONSTITUENCY
count	2018.000000	2018.000000	2.018000e+03	2.018000e+03	2.018000e+03	2018.000000	2.018000e+03	2018.000000
mean	1.453915	52.273538	1.315849e+08	1.973860e+07	2.911903e+05	1105.111001	2.922954e+05	17.596810
std	7.636973	11.869373	4.122697e+08	8.945292e+07	2.545964e+05	1661.283371	2.555874e+05	14.886247
min	0.000000	25.000000	0.000000e+00	0.000000e+00	1.339000e+03	0.000000	1.342000e+03	0.097941
25%	0.000000	43.250000	6.272526e+06	0.000000e+00	3.047625e+04	97.000000	3.074375e+04	1.953617
50%	0.000000	52.000000	2.404181e+07	9.039245e+05	2.846300e+05	463.000000	2.855250e+05	18.036861
75%	1.000000	61.000000	9.152498e+07	6.097971e+06	5.058620e+05	1545.500000	5.076175e+05	30.708115
max	240.000000	86.000000	8.950179e+09	1.547511e+09	1.066824e+06	19367.000000	1.068569e+06	51.951012

Now, it is always a good approach to study the type and count of the columns in the dataset. This helps in understanding the number of null-values in large proportion, if the data is categorical or numerical or object type due to the presence of NaN values. This can be done in one glance using the .info( ) function of the pandas library. The info about the dataset is summarised as follows.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2263 entries, 0 to 2262
Data columns (total 19 columns):
 #   Column                                    Non-Null Count  Dtype
---  ------                                    --------------  -----
 0   STATE                                     2263 non-null   object
 1   CONSTITUENCY                              2263 non-null   object
 2   NAME                                      2263 non-null   object
 3   WINNER                                    2263 non-null   int64
 4   PARTY                                     2263 non-null   object
 5   SYMBOL                                    2018 non-null   object
 6   GENDER                                    2018 non-null   object
 7   CRIMINAL
CASES                            2018 non-null   object
 8   AGE                                       2018 non-null   float64
 9   CATEGORY                                  2018 non-null   object
 10  EDUCATION                                 2018 non-null   object
 11  ASSETS                                    2018 non-null   object
 12  LIABILITIES                               2018 non-null   object
 13  GENERAL
VOTES                             2263 non-null   int64
 14  POSTAL
VOTES                              2263 non-null   int64
 15  TOTAL
VOTES                               2263 non-null   int64
 16  OVER TOTAL ELECTORS
IN CONSTITUENCY      2263 non-null   float64
 17  OVER TOTAL VOTES POLLED
IN CONSTITUENCY  2263 non-null   float64
 18  TOTAL ELECTORS                            2263 non-null   int64
dtypes: float64(3), int64(5), object(11)
memory usage: 336.0+ KB
```

9.3. 2019 Election ; Some Visual Exploration and Insights

1. States With Most Number of Constituencies

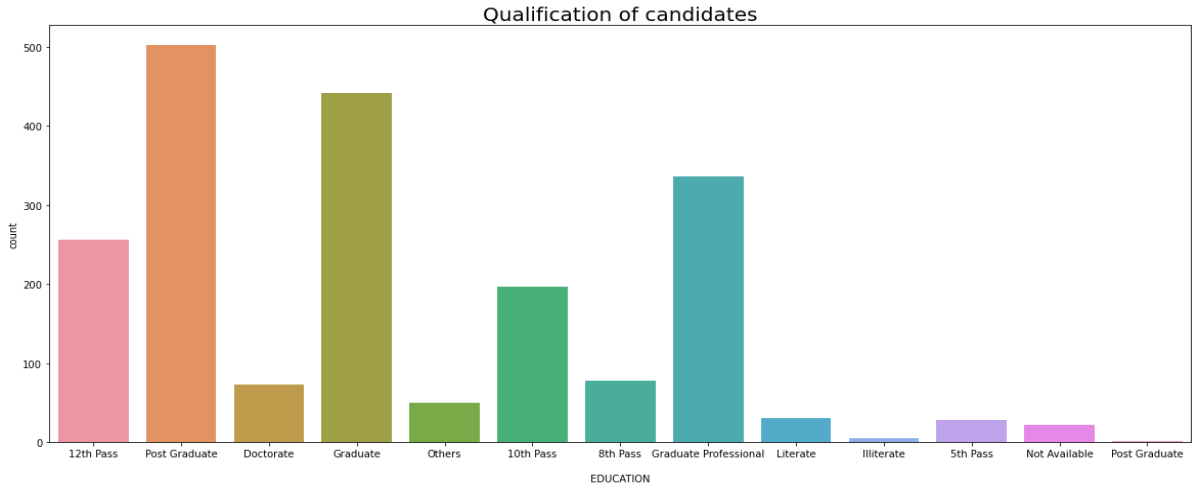


Observations :

1. Uttar Pradesh has the highest number of constituencies in India
2. Even Though Rajasthan is the largest state in India, it comes at 8th position.
3. UP has nearly double the number of constituencies when compared the second highest

2. Qualification of Candidates

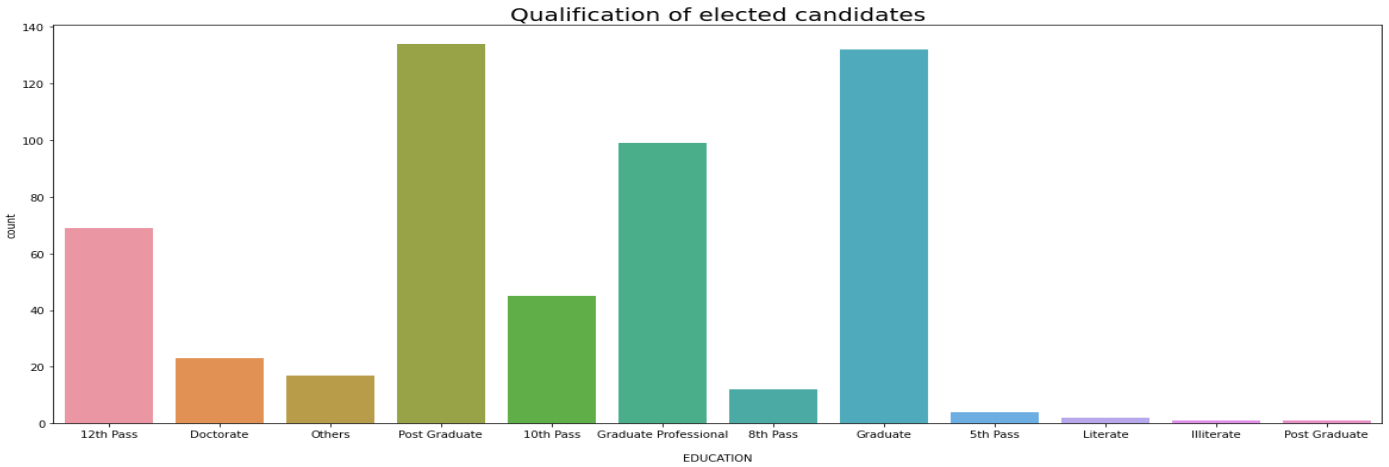
i) Qualification of Every Candidate



Observations :

1. Most of the candidates have basic education
2. Candidates having Post Graduation and Graduation as higher when compared

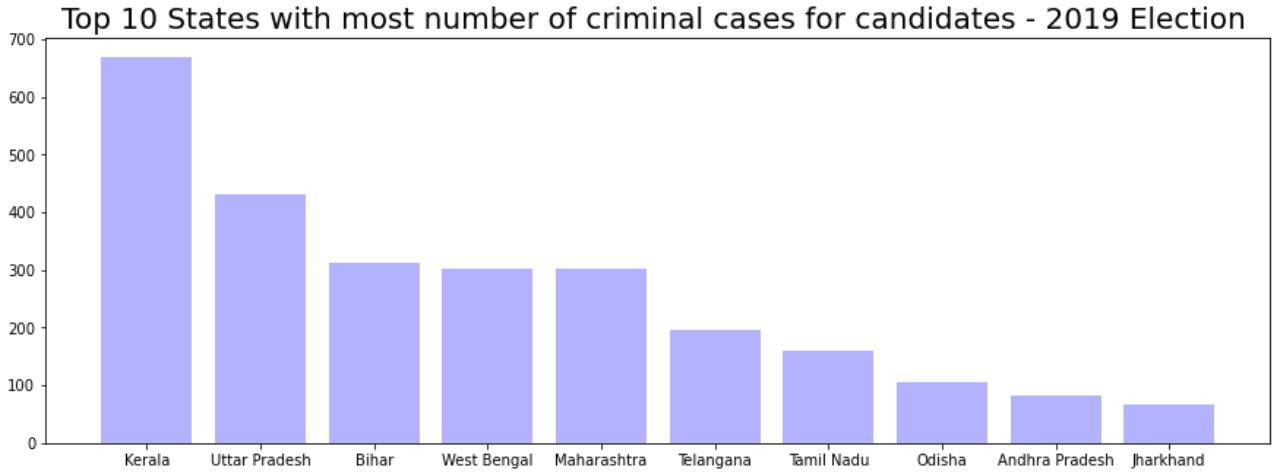
ii) Qualification of Elected Candidates



Observations :

1. About 90% of the elected candidates have primary education.
2. Majority of MPs have Post Graduation and Graduation
3. Educational qualification also plays a not-so-small role in choosing the right candidate

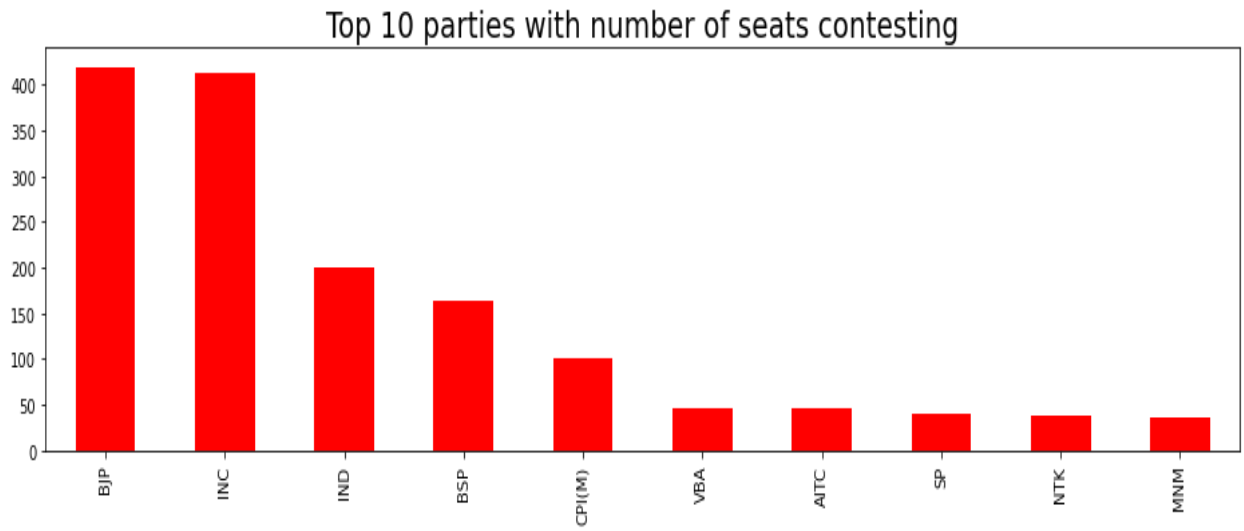
3. States with Most Criminal Cases



Observations :

1. It is shocking to see that even with lesser number of constituencies and candidates Kerala has the highest number of criminal cases

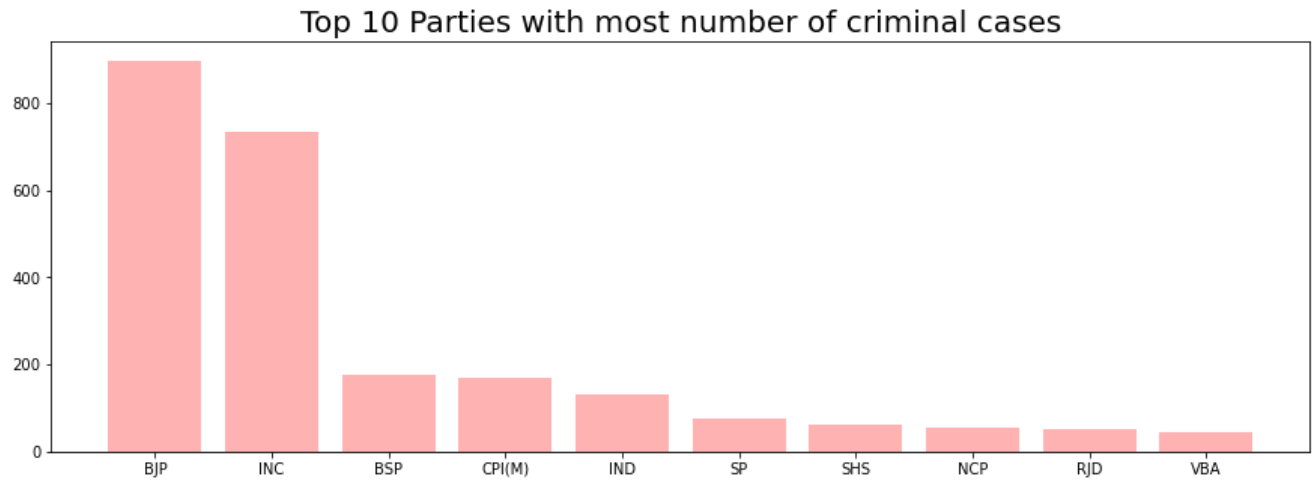
4. Number of Seats Each Parties Contesting



Observations :

1. BJP and INC are the real competitors in 2019 Elections
2. Majority of these parties will decide who will govern India for the next 5 years

5. Criminal Cases in Parties



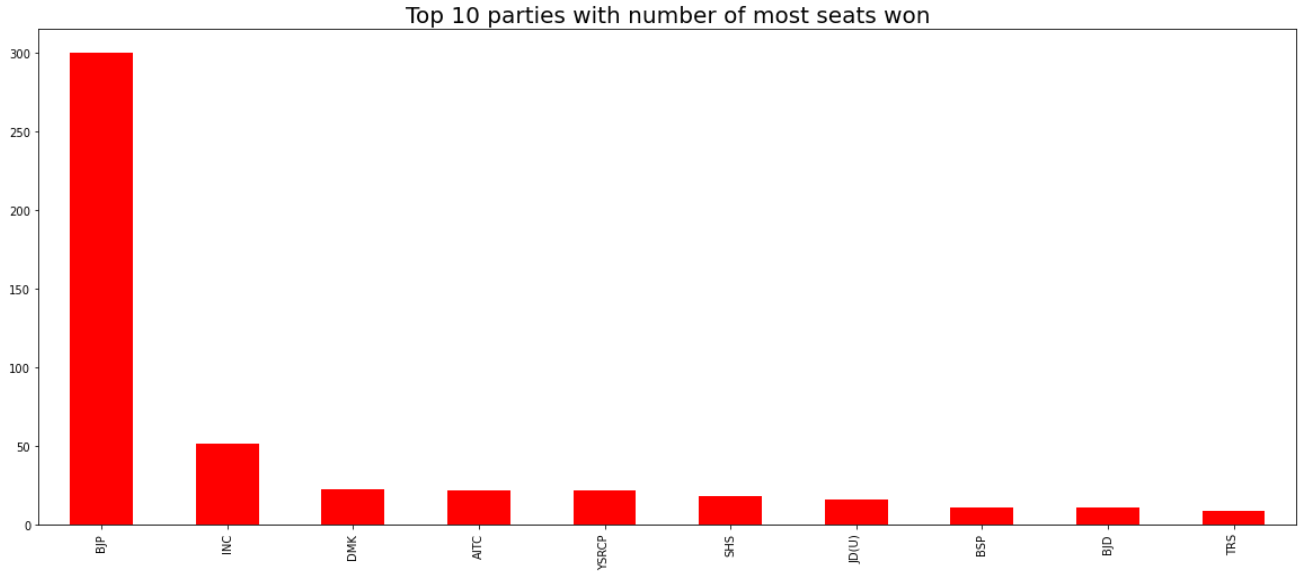
Observations :

1.BJP has the most number of criminal cases followed by INC

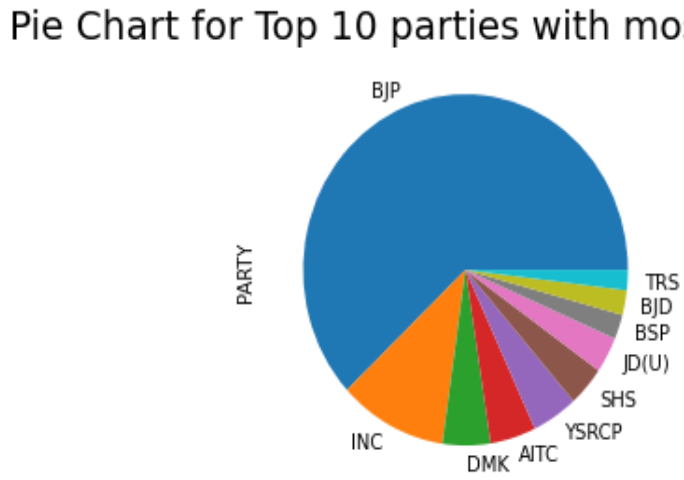
2.This huge difference between BJP & INC and other parties are because BJP and INC are contesting in almost every constituency, hence higher number of candidates.

6. Total Number of Seats Won

i) Bar Graph



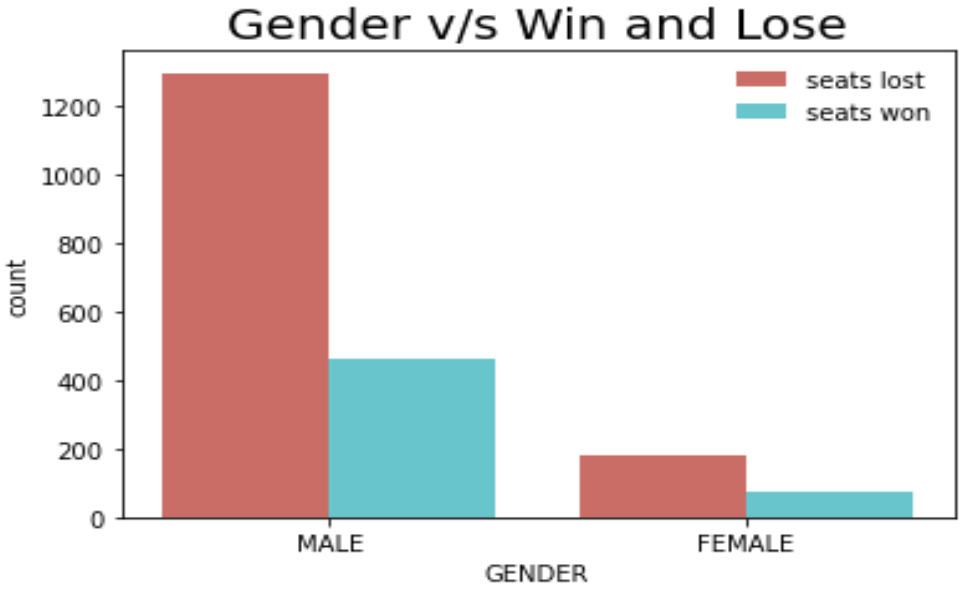
ii) Pie Chart



Observations :

1. Out of total seats contested BJP has secured almost 300 seats
2. INC had a huge defeat.

7. Gender Role in Elected Candidates



Observations :

1. Most of the elected candidates are male
2. There is a huge gap in the gender ratio

Having obtained a detailed analysis and visualization of the dataset, the next step is towards a supervised classification of the data to predict the winner or loser. The WINNER column is the target variable.

10. Supervised Learning Model

In our work, we employed the supervised learning approach to develop a classifier to predict if the given candidate/party is a winner.Irrespective of the approaches used, the basic cycle of machine learning model development consists of the following stages[10].

Problem Definition : In this project, we aim to develop a machine learning model which can classify win/ loss in the election based on the polling data. Different supervised models are employed and the best model is chosen. Proper validation is used to refine the performance of the model. Since this is a classification problem, Here, we are training the model with the existing set of results. The election dataset of 2019 is considered here. The WINNER is the target variable.

Preparing Data & Exploratory Analysis : The necessary columns of the election dataset are used for modeling.

The following steps are performed in preparing data for modeling.

- `'CRIMINAL_CASES','AGE','ASSETS','LIABILITIES','TOTAL_ELECTORS','SYMBOL'` are not deciding factors of a candidate's success as inferred from the domain knowledge and analysis. So, these columns are dropped from the dataset while creating a model. The remaining columns can be observed using the .info ( ) command in python as below.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2018 entries, 0 to 2261
Data columns (total 13 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   STATE                                   2018 non-null   object
 1   CONSTITUENCY                            2018 non-null   object
 2   NAME                                    2018 non-null   object
 3   PARTY                                   2018 non-null   object
 4   GENDER                                  2018 non-null   object
 5   CATEGORY                                2018 non-null   object
 6   EDUCATION                               2018 non-null   object
 7   GENERAL_VOTES                           2018 non-null   int64
 8   POSTAL_VOTES                            2018 non-null   int64
 9   TOTAL_VOTES                             2018 non-null   int64
 10  OVER_TOTAL_ELECTORS_IN_CONSTITUENCY     2018 non-null   float64
 11  OVER_TOTAL_VOTES_POLLED_IN_CONSTITUENCY 2018 non-null   float64
 12  WINNER                                  2018 non-null   int64
dtypes: float64(2), int64(4), object(7)
memory usage: 300.7+ KB
```

- There were outliers in the `POSTAL_VOTES` column. These were removed from the dataset. Even after removing the outliers once using the quantile range to be 75%, there were still a small number of outliers which did not seem to create a major impact.
- The next step in preparation of data was to encode the categorical features. This was performed using the `OrdinalEncoder()` in the preprocessing section of sklearn library. This converts all the categorical features into their equivalent numeric representations as the model handles only numeric features.
- The correlation matrix is studied to observe the dependency of the target variable on the various features using the .corr( ) function.From the heatmap, it can be observed that there are a lot of features which are not related with the target variable. These can be removed and a classification model can be built from the remaining features. Another approach used was to evaluate the feature scores based on the `SelectKBest` function. This generated the following importance scores for the features listed below in the order of the column names as in the column list.

```
Index(['STATE',    'CONSTITUENCY',    'NAME',    'PARTY',    'GENDER',
'CATEGORY','EDUCATION','GENERAL_VOTES','POSTAL_VOTES','TOTAL_VOTES','
OVER_TOTAL_ELECTORS_IN_CONSTITUENCY','OVER_TOTAL_VOTES_POLLED_IN_CONS
TITUENCY', 'WINNER'], dtype='object')
```

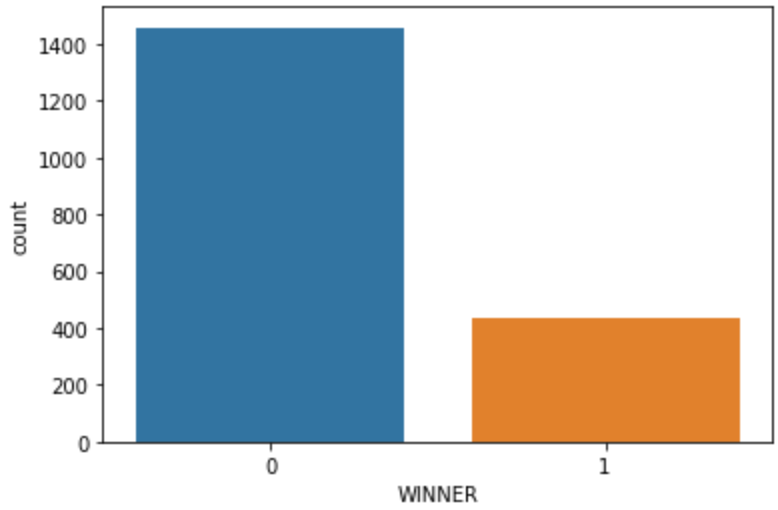
```
Feature 0: 10.923026
Feature 1: 10.763989
Feature 2: 20.159420
Feature 3: 963.739853
Feature 4: 0.268248
Feature 5: 4.778464
Feature 6: 0.003780
Feature 7: 169703673.845435
Feature 8: 377216.538663
Feature 9: 170074025.101606
Feature 10: 9886.748683
Feature 11: 14802.027147
```

The features which showed good importance were used to create the model and they are as follows(target is shown, but not to be treated as one of the features).

```
Index(['GENERAL_VOTES','POSTAL_VOTES','TOTAL_VOTES','OVER_TOTAL_ELECT
ORS_IN_CONSTITUENCY','OVER_TOTAL_VOTES_POLLED_IN_CONSTITUENCY','WINNE
R'],
dtype='object')
```

- The value counts of class or target variable was found to see if there is an imbalance in the class variable. This is helpful in deciding the cross-validation technique to be used. It was found using the .value_counts function.

```
0    1479
1     539
Name: WINNER, dtype: int64
```



Here, the StratifiedKFold technique can be preferred for cross-validation of the model.

Training of the model : This process involves splitting the data into train and test data using the `train_test_split` function of the sklearn library. 80 % of data is assigned for training and 20 % is assigned for testing the performance of the model. Validation is not performed initially. But, the stratify option is given so that there is a splitting of data without any bias. The random_state is chosen to be 1 as it was observed to give better metrics compared to 42.

Model Generation : Here we perform different classification algorithms on our training data and then it is used to predict the outcome of test data. The various models used were LogisticRegression, RandomForestClassifier, DecisionTree, Gradient Boosting methods and Naive Bayes Classifier.

Testing of Algorithms: Once a model is fitted on to the training data, it is then evaluated based on the test data. The performance metrics for the classifier is estimated and it gives the best model for the given dataset. No hyperparameter tuning is done at this stage.

The performance metrics for the different algorithms are summarised below.

Logistic Regression The performance of the model is expressed as a summary using the Classification Report which is obtained using the `classification_report()` function.

```
              precision    recall  f1-score   support

           0       0.73      0.45      0.56       292
           1       0.20      0.45      0.27        87
```

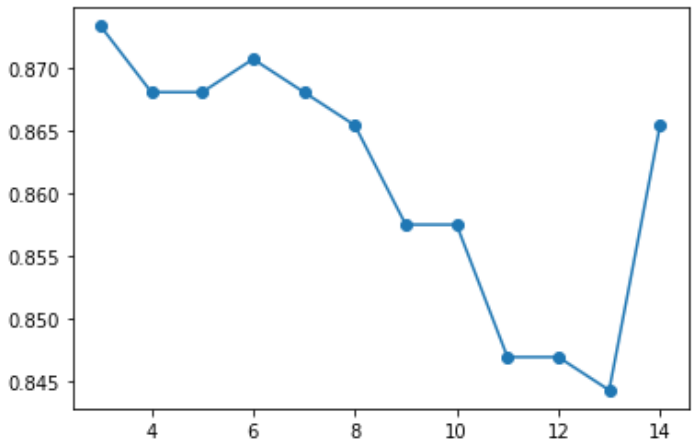
accuracy			0.45	379
macro avg	0.46	0.45	0.41	379
weighted avg	0.61	0.45	0.49	379

The classification accuracy for class 1 is very low. To address this, the data was scaled to see if there is any improvement. StandardScaler ( ) function was used to perform scaling. It converts the data points to follow a normal distribution. The performance after standard scaling is as shown below.

	precision	recall	f1-score	support
0	0.97	0.98	0.97	292
1	0.93	0.89	0.91	87
accuracy			0.96	379
macro avg	0.95	0.93	0.94	379
weighted avg	0.96	0.96	0.96	379

The performance has significantly improved with scaling. MinMax Scaling was also performed, but it was not suitable for this model.

k-NN Model The optimum number of neighbors are chosen first and then the model is implemented. For this, the accuracy of the model for different numbers of neighbors is implemented and then the model is built using the optimum value. The number of neighbors for which the model performs well was obtained as k=5 as at this value, the accuracy is found to be maximum as shown in the figure.



The classification metrics for this model is summarised below.

	precision	recall	f1-score	support
0	0.91	0.93	0.92	292
1	0.74	0.69	0.71	87
accuracy			0.87	379
macro avg	0.83	0.81	0.82	379
weighted avg	0.87	0.87	0.87	379

Since this is a distance based method, scaling can play a significant role. The performance was improved by scaling the data and the resulting scores are as follows.

	precision	recall	f1-score	support
0	0.95	0.98	0.96	292
1	0.91	0.84	0.87	87
accuracy			0.94	379
macro avg	0.93	0.91	0.92	379
weighted avg	0.94	0.94	0.94	379

The scores have improved a lot after standard scaling. MinMax scaling was found to be slightly more effective in kNN.

	precision	recall	f1-score	support
0	0.96	0.98	0.97	292
1	0.94	0.85	0.89	87
accuracy			0.95	379
macro avg	0.95	0.92	0.93	379
weighted avg	0.95	0.95	0.95	379

SVM Model The execution time for the SVM model on the unscaled data was found to be longer and hence we considered the model on scaled data. The results on the standard scaled data are as follows.

	precision	recall	f1-score	support
0	0.96	0.98	0.97	292
1	0.93	0.87	0.90	87
accuracy			0.96	379
macro avg	0.94	0.93	0.94	379
weighted avg	0.95	0.96	0.95	379

There was a bias in classification with MinMax scaling on the data. So it is not considered.

Decision Tree Model

In this model, the factors considered are the maximum depth of the tree, criterion for splitting and weight of class labels. Since the class labels are imbalanced,

```
precision    recall  f1-score   support

           0       0.94      0.97      0.95       292
           1       0.88      0.80      0.84        87

    accuracy                           0.93       379
   macro avg       0.91      0.89      0.90       379
weighted avg       0.93      0.93      0.93       379
```

Random Forest Algorithm

The performance metrics for Random Forest classification is as below. It does not require scaling. The parameters were optimised to get the final set of parameters as

```
RandomForestClassifier(n_estimators=200,max_depth=5,random_state=1,cri
terion='entropy')
```

```
precision    recall  f1-score   support

           0       0.96      0.98      0.97       438
           1       0.93      0.85      0.89       130

    accuracy                           0.95       568
   macro avg       0.95      0.92      0.93       568
weighted avg       0.95      0.95      0.95       568
```

Gradient Boost Algorithm

The performance metrics for the algorithm are as follows.

```
                precision    recall  f1-score   support

           0       0.96      0.98      0.97       438
           1       0.93      0.85      0.89       130

    accuracy                           0.95       568
   macro avg       0.94      0.91      0.93       568
weighted avg       0.95      0.95      0.95       568
```

There was no significant change in the performance with hyperparameter tuning

XGBoost Model

The summary of performance metrics for the algorithm is as follows.

	precision	recall	f1-score	support
0	0.96	0.98	0.97	438
1	0.93	0.85	0.89	130
accuracy			0.95	568
macro avg	0.94	0.92	0.93	568
weighted avg	0.95	0.95	0.95	568

Naive-Bayes Model

The summary of the model is given below. It doesn

	precision	recall	f1-score	support
0	0.98	0.82	0.89	438
1	0.60	0.95	0.74	130
accuracy			0.85	568
macro avg	0.79	0.88	0.81	568
weighted avg	0.89	0.85	0.86	568

It didn't offer a satisfactory performance and the log loss was also found to be high

Cross-Validation : The dataset used in this project for analysis is a small one having less than 3k rows. In this case, there is every chance that we will face an overfitting issue. When there is a situation of overfitting, the performance of the model will not be satisfactory. Our dataset is small and hence *k*-fold cross-validation or Leave One Out Cross-Validation(LOOCV) can be performed. Among them, *k*-fold is good enough to perform the validation and here the maximum value of k=10 is chosen. A less value of k can result in a biased estimate. We have used StratifiedkFold validation here.

11. Sentiment Analysis & Modeling

In this section, we will deal with the sentiment analysis of the Twitter data related to the Indian General Elections 2019.

11.1 Problem Definition :

Here, we aim to find the sentiments of the voters towards the ruling party(NDA) on the basis of a twitter campaign raised by the party. In Twitter sentiment analysis, analysing data on the days of some campaigns or movements is of great significance as more and more opinions are generated in a single day and it can be a useful estimate of the sentiments. This can even prove effective compared to analyzing tweets over a month or two if the objective is to find the polarity of scores and a generalized prediction.

11.2 Data Gathering :

We have used a Twitter dataset based on the MainBhiChowkidaar campaign by the Bharatiya Janatha Party. This took place before the elections (April 19,2019) and was a relevant campaign since this invited a lot of positive and negative opinions. The data traffic was much more compared to a usual day and it is an obvious occurrence. Under standard procedures, data is accessed through Twitter API. We have considered a dataset from GitHub which was available in CSV format. It consists of 1 lakh rows and more than 25 features. The most relevant of them is the tweet content in the form of text. We are considering only text content.

11.3 Importing the libraries and dataset :

The major libraries include pandas, NLTK, TextBlob with many other packages. The libraries are imported and the dataset is loaded into the environment. The loading time is more due to the size of the dataset.

11.4 Exploratory Analysis & Visualization

To understand the data well, we do an exploration of the various features of the dataset.

- The columns of the dataset can be views using the .columns() as follows.

```
Index(['user_id', 'status_id', 'created_at', 'screen_name', 'text',
'source','display_text_width','reply_to_status_id', reply_to_user_id',

'reply_to_screen_name', 'is_quote', 'is_retweet', 'favorite_count',
'retweet_count', 'hashtags', 'symbols', 'urls_url', 'urls_t.co',
'urls_expanded_url', 'media_url', 'media_t.co', 'media_expanded_url',
```

```
'media_type','ext_media_url','ext_media_t.co',
‘ext_media_expanded_url',     'ext_media_type',     'mentions_user_id',
'mentions_screen_name', 'lang', 'quoted_status_id', 'quoted_text',
 'quoted_created_at', 'quoted_source', 'quoted_favorite_count',
 ‘quoted_retweet_count', 'quoted_user_id', 'quoted_screen_name',
'quoted_name', 'quoted_followers_count', 'quoted_friends_count',
'quoted_statuses_count', 'quoted_location', 'quoted_description',
'quoted_verified', 'retweet_status_id', 'retweet_text',
 'retweet_created_at', 'retweet_source', 'retweet_favorite_count',
‘retweet_retweet_count', 'retweet_user_id', 'retweet_screen_name',
 'retweet_name', 'retweet_followers_count', 'retweet_friends_count',
 'retweet_statuses_count', 'retweet_location', 'retweet_description',
 'retweet_verified', 'place_url', 'place_name', 'place_full_name',
'place_type','country', 'country_code', 'geo_coords', 'coords_coords',
'bbox_coords', 'status_url', 'name', 'location', 'description', 'url',
 'protected', 'followers_count', 'friends_count', 'listed_count',
'statuses_count','favourites_count', 'account_created_at', 'verified',
 'profile_url', 'profile_expanded_url', 'account_lang',
'profile_banner_url', 'profile_background_url', 'profile_image_url'],
      dtype='object')
```

It can be observed that, there are a large number of features, but many of them have missing values as well as they don't contribute to analysis or modeling. They are dropped before the analysis.

- The tweets for a duration of 24 hours were considered for analysis and this is specific to the Chowkidaar campaign. The data of start and end of tweet can be found using the min and max function in data column as follows.

```
Start - '2019-03-19 14:07:46'  End-  2019-03-20 22:06:56'
```

- The mean length of a tweet is important as it gives an idea about the data to be processed from each tweet and hence the total text content. It is obtained using the mean( ) function.

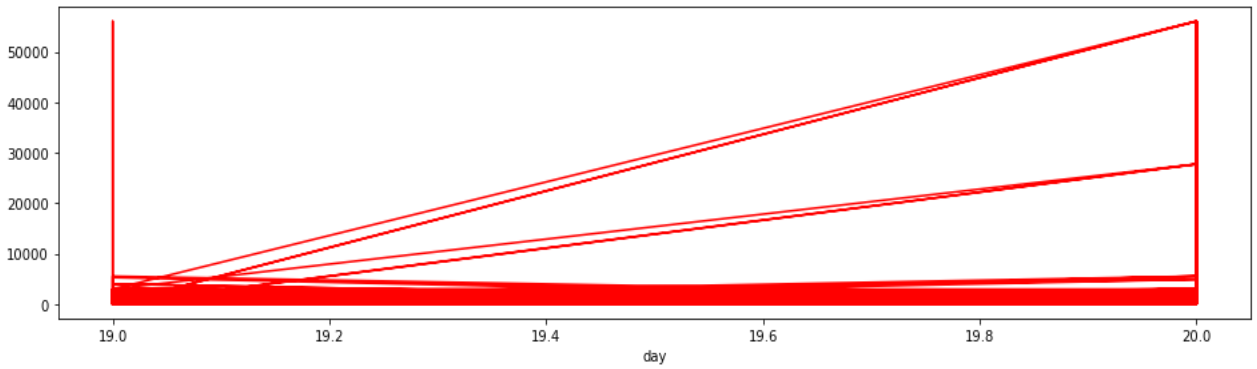
```
The mean length of the tweets:  134.89383891523414
```

- Another interesting aspect is the most liked tweet and the number of rewteets made. This can be obtained as follows.

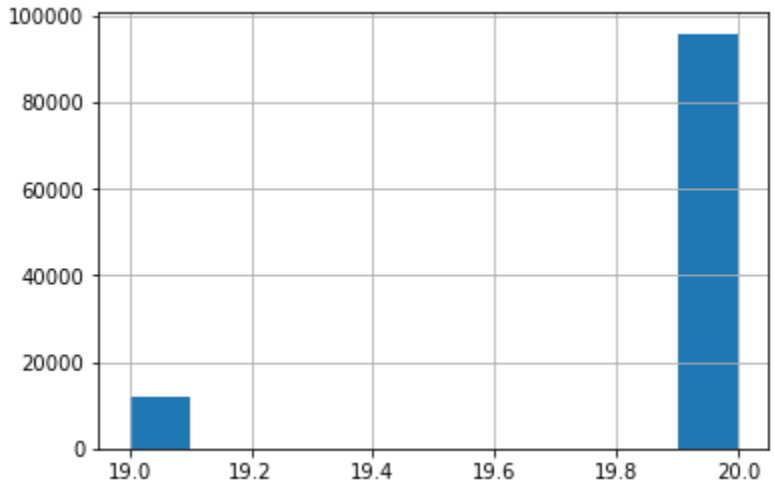
```
The most liked tweet is: Your Chowkidar is standing firm ; serving the
nation.

But, I am not alone.

Everyone who is fighting corruption, dirt, social evils is a Chowkidar.
```

```
Everyone working hard for the progress of India is a Chowkidar.

Today, every Indian is saying-#MainBhiChowkidar


Number of retweets: 56030
------------------
```

- Also, the distribution of the retweet count during the day is obtained as shown below.



- The tweets are distributed over two days constituting the entire 24 hours span as shown below



- The trending hashtag was also found from the dataset using the mode function as shown below.

```
0    MainBhiChowkidar
```

```
dtype: object
```

- Also, the number of times the word Chowkidaar was mentioned in the tweets were also found as below.

```
chowkidar comes 105899 times
```

- The major hashtags and their respective counts were also estimated to get an idea of the data.

```
MainBhiChowkidar                                                                16991
NiravModi                                                                        5530
Chowkidar                                                                        2316
Lootidar NiravModi Chowkidar                                                     1261
NiravModi NiravModi                                                              1043
                                                                                 ...
ChowkidarPhirSe ChowkidarNarendraModi Chowkidar                                     1
sms WeWantChowkidar                                                                 1
NaMoAgain2019 NaMoAgain 2019Elections Hindus tamilhindus MainBhiChowkidar hinduism  1
reservation SaveTheUnreserved                                                       1
MainBhiChowkidar IssbaarbhiModiSarkar                                               1
Name: hashtags, Length: 5208, dtype: int64
```

- The most favorite tweet among the given tweets and the count of favorite counts were found as below.

```
#NiravModi arrested in London but Chowkidar  chor hai aur  jo
Nirav Modi ke saath cocktail pe rahe the paak saaf hai? Will
Rahul accept my challenge ; deny he did not meet Nirav Modi when
loans were being given to Mama Bhanja?

11827
```

- Most favorite twitter handle was found as shown below. It was decided based on the number of followers.

```
Screen Name : timesofindia

No of followers : 11702921
```

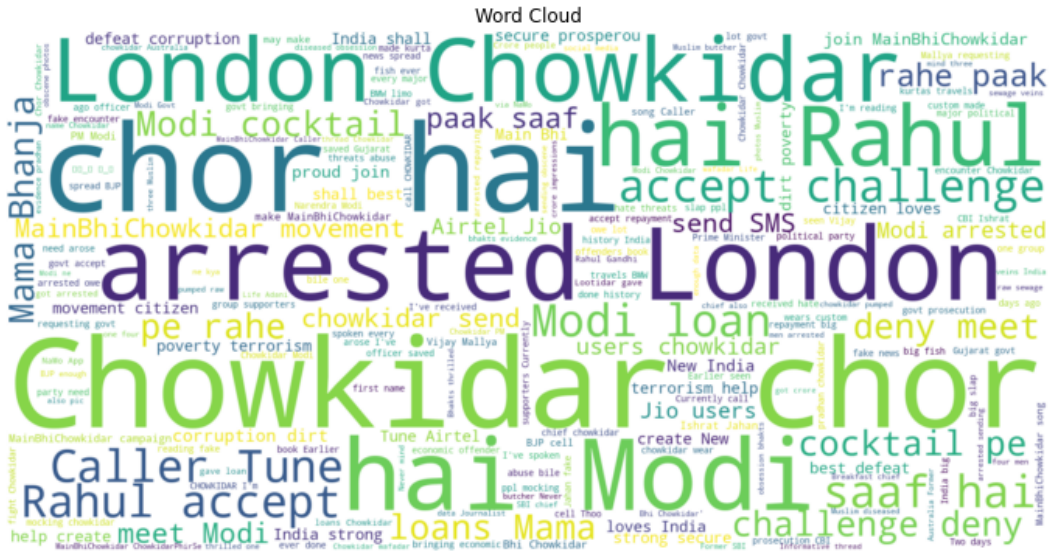
11.4 Preprocessing the tweet text

- The tweet text is the only relevant column when it comes to extracting sentiments. So that column is selected for analysis.
- The tweet text involve a lot of special characters like @,#, “” etc and these do not contribute to sentiment analysis. So, they need to be removed. They are replaced by blank space.

- **Tokenizing the tweet:** Here the entire sentence is split into words and stemming is performed to express every word in terms of its root word, so as to reduce unwanted features.
- **Stop Words Removal** : This consists of removing unwanted words which do not represent any information , like ‘and’, ‘for’,’has’, etc. There is a stopwords library which contains the stopwords in English language. Along with this, we can also add our list of stop words to be removed from the tweet.

11.5 : Word Cloud

This represents the essence of the data. It displays the most repeated words in the tweet such that the word with the highest count appears in larger font. Less popular words are shown in small fonts. In this data, the wordcloud has been obtained as follows. There can be different combinations of wordclouds based on the stopwords.



Observation

There are both supporting as well as opposing tweets towards the Chowkidar campaign and the most highlighted word is Chowkidar and Chowkidar Chor hai.

Observations:

- It can be seen that there are words from regional languages in addition to English. We have not created an automated system to address this issue. Rather, we have picked up words which are not relevant in the present context. This approach is more realistic compared to the automated one.
- From the above word cloud, it can be understood that, Chowkidaar campaign is a suitable choice for sentiment analysis before elections.
- The word cloud does not give the exact sentiment. So, for this, we perform the sentiment analysis to find the polarity scores towards the specific campaign.

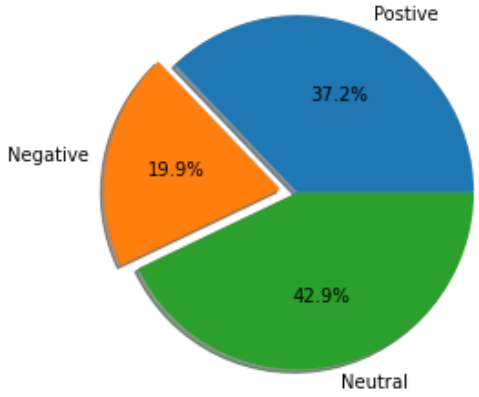
11.6 Sentiment Polarity Scores

We obtained the sentiment polarity based on two libraries

1) Textblob
2) Valence Aware Dictionary for Sentiment Reasoning(VADER)

11.6.1 *Textblob*

The tweets after tokenization are ready for sentiment extraction. Textblob is built on top of NLTK and the polarity scores were obtained in three levels - positive, negative and neutral. Since we are considering a campaign, we can consider the sentiment polarities towards this campaign. It can be calculated by the `.sentiment.polarity` function of the Textblob package. The resulting polarity counts are plotted in a pie chart for better visualization as shown below.



It can be observed that the positive sentiment score is more compared to the negative score. This shows a favour towards the campaign and hence to the Bharatiya Janata Party . This

can be further reinforced by the occurrence of for and against texts on this campaign as below.

```
chowkidar 51448
#mainbhichowkidar 32695
modi 26581
nirav 14203
chor 12120
india 10757
arrested 10540
pm 10479
#chowkidar 9805
govt 8848
```

Here, the texts favouring the party are more than that against it justifying the polarity diagram.

11.6.2 VADER

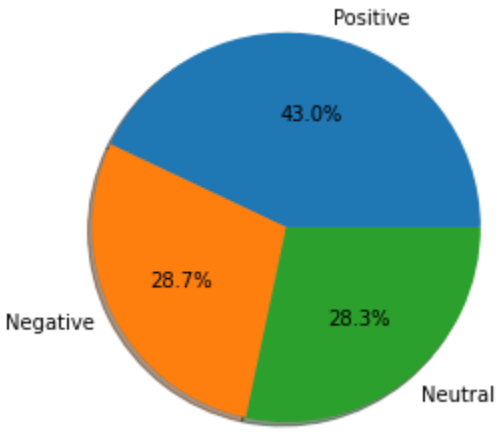
The sentiment scores from this analyzer consists of four parts- positive, negative , neutral and compound. The compound score is a good estimate when it comes to classification of sentiments. The polarity of the sentiments can be estimated from the compound score using the relation

Positive Sentiment: compound score >= 0.05

Neutral Sentiment: compound score > -0.05 and < 0.05

Negative Sentiment: compound score <= -0.05

Based on this, the compound scores and the corresponding polarity scores were calculated and the pie diagram representing the sentiments is as follows.



By comparing this result with that of TextBlob, it can be seen that it has suggested more positive sentiments and been able to interpret semantic opinions well. This can be justified by using a sample from the dataset as well from a random text.

a) Sample text from the dataset

See one #Chowkidar thrashed Pappu supporter mi...	0	-0.1531	-1

The first value represents the polarity score assigned by Textblob and the second and third represents the compound and sentiment score given by VADER . It can be observed that the sentence is negative as per human intelligence. And it is classified into neutral sentiment by the Textblob algorithm, whereas VADER classifies it correctly.

b) Random Text- Modi is not a good ruler for the country though he has done many reforms

The Textblob classifies this as positive as it does negation handling whereas the VADER classifies this as neutral as it has both positive and negative aspects and the text is conveying a neutral sentiment.

In this way , VADER can offer a better classification and having more positive comments indicates a supporting tendency for the Chowkidar Campaign and hence a positive sentiment for the Bharatiya Janata Party.

11.6.3 Machine Learning Model

The tweet along with the sentiment scores based on both the lexicon methods were used to develop machine learning models which can predict the sentiment polarity for a given text. This involved the following steps.

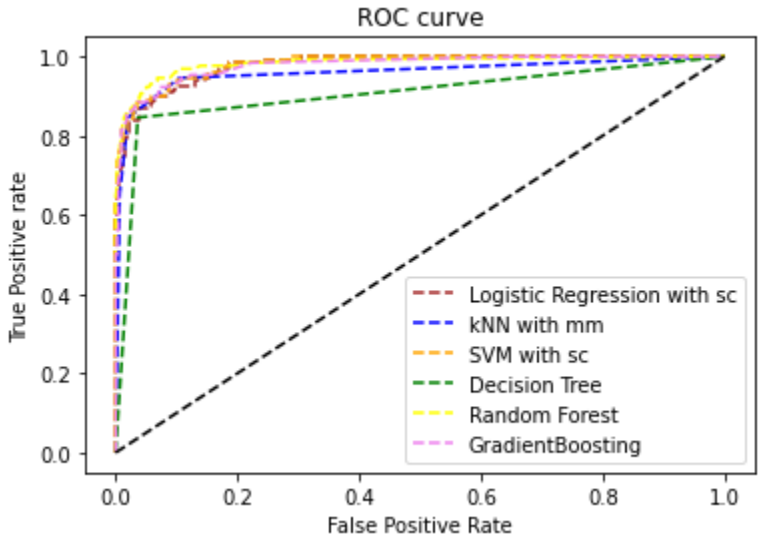
1. **Conversion of text to numeric data -** We used the bag of words(BoW) approach to convert the text into numerical data . For this CountVectorizer( ) function was used.
2. **Converting bow matrix to array-** The bag of words gives the output as sparse matrix and it needs to be converted to array as some of the machine learning models can not handle this without conversion.
3. **Splitting into train,test and validation data -** Here, since the dataset is large, there is no requirement to have a k-fold or similar cross-validation techniques.Here we split the dataset into test and train data first , followed by train and validation set with 35 percent allotted for test data and 20% for validation.

12. Results

12.1 Supervised Learning Model

We analyzed the performance of the five popular supervised learning models in classifying the WINNER to 0 or 1. Our dataset was a small one having less than 3000 entries and there was a concern of overfitting of data since there was an imbalance in the class labels.

- 30% of the sample belonged to Class label 1 and the remaining was in Class 0.
- At first, we implemented the conventional train_test_split and recorded the performance metrics. Since this was an imbalance classification, we preferred to give importance to the Log Loss metric as it offers a penalty to the algorithm for each incorrect classification for the minority label .The other major parameters for classification algorithms like precision, F1 score etc were also studied for a comparison.(presented in Section 10) The AUC-ROC curve was also plotted which gave a graphical interpretation of the classification efficiency.



It can be observed that all the models are giving a satisfactory performance as their curves are moving towards (0,1) point. The decision tree is comparatively giving a poor classification. The best models are the Random Forest classifier, along with Logistic Regression and Support Vector Machine(SVM). But, the limitation with the SVM model and Logistic Regression is its dependence on scaling. In our case, both these algorithms showed very poor performance without scaling, with SVM showing a very long execution time without scaling.

- To get more clarity about the best model, the StratifiedKFold cross-validation split was performed on the data as it is effective in classes with imbalance in labels. The cross_val_scores for all the models are summarised below.

	Model	Accuracy	Precision	Recall	F1score	ROCAUC
0	Logistic Regression	0.946647	0.893110	0.873256	0.882647	0.981410
1	Random Forest	0.945592	0.917889	0.838689	0.875943	0.980504
2	SVM	0.945065	0.900463	0.857135	0.877590	0.981313
3	Gaussian Naive Bayes	0.857354	0.628318	0.944767	0.753669	0.958165
4	Decision Tree	0.923409	0.836544	0.829387	0.831952	0.890367

Observation : Logistic Regression, Random Forest and SVM are showing good precision, F1 and AUC-ROC scores compared to the remaining models. The log-loss parameter is also evaluated for the predicted outcome and it is shown below.

	model_names	log_loss
0	Logistic Regression	1.520207
1	Logistic Regression Tuned	1.824256
2	KNN	1.763436
3	SVM	1.641822
4	Decision Tree	2.189100
5	Random Forest	1.641819
6	Random Forest Tuned	1.581014
7	GradientDescent	1.702627
8	GaussianNB	5.351192

Observation : The log loss score is the least for the Logistic Regression followed by Random Forest and SVM models. This suggests that the Logistic Regression is good in terms of classification in classes with label imbalance, if scaled data is available. Random Forest can be chosen if there is unscaled data. SVM also performs well, but only with scaled

data. So Random Forest model and Logistic Regression are equally well for the given dataset.

12.2 Sentiment Analysis & Prediction Model

In our project, we have attempted to perform sentiment analysis on Twitter data based on the Chowkidaar campaign. Ours was a hybrid approach, where we tried to use a combination of the lexicon model and the machine learning model. Based on the results of the sentiment analysis,

- VADER package provided a better sentiment classification
- Random Forest Model was used for modeling the tweet - Here we had three labels 0,1 and -1 representing neutral, positive and negative sentiments. The model was run for both the sentiment scores of both the lexicon models.

TextBlob

	precision	recall	f1-score	support
-1	0.97	0.90	0.93	9337
0	0.89	0.95	0.92	9149
1	0.94	0.94	0.94	13905
accuracy			0.93	32391
macro avg	0.93	0.93	0.93	32391
weighted avg	0.93	0.93	0.93	32391

```
Out[175]: array([[ 8389,   435,   513],
                 [  112,  8664,   373],
                 [  158,   641, 13106]])
```

VADER

```
                   precision    recall  f1-score   support

               -1       0.98      0.90      0.94      6230
                0       0.93      0.97      0.95     14121
                1       0.95      0.94      0.94     12040

         accuracy                           0.95     32391
        macro avg       0.95      0.94      0.94     32391
     weighted avg       0.95      0.95      0.94     32391
```

Out[173]:
```
array([[ 5578,   392,   260],
       [   54, 13761,   306],
       [   63,   705, 11272]])
```

VADER gives a better prediction than TextBlob

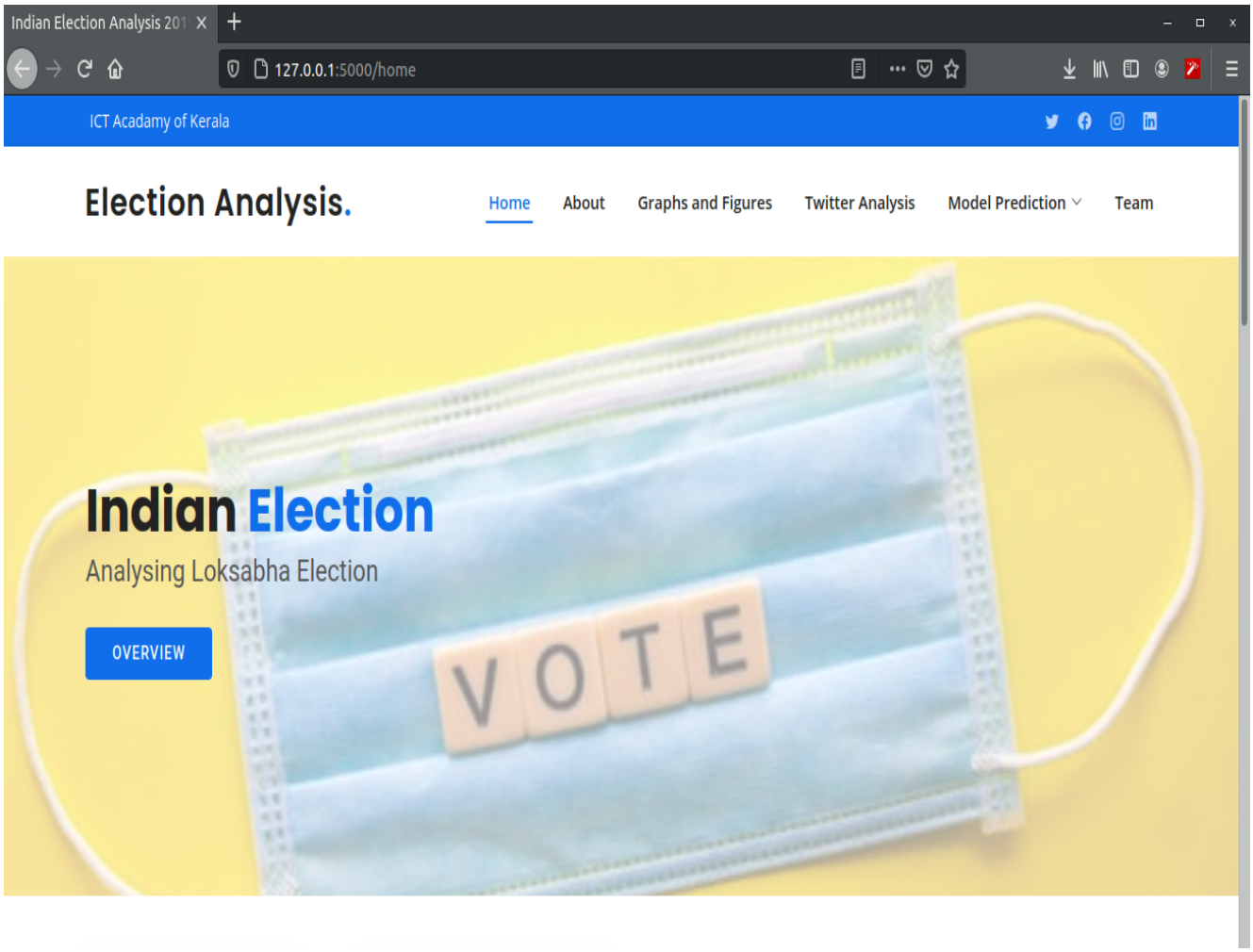
13. Deployment of Model

We have used the Heroku platform to deploy our model . Model has been hosted at :

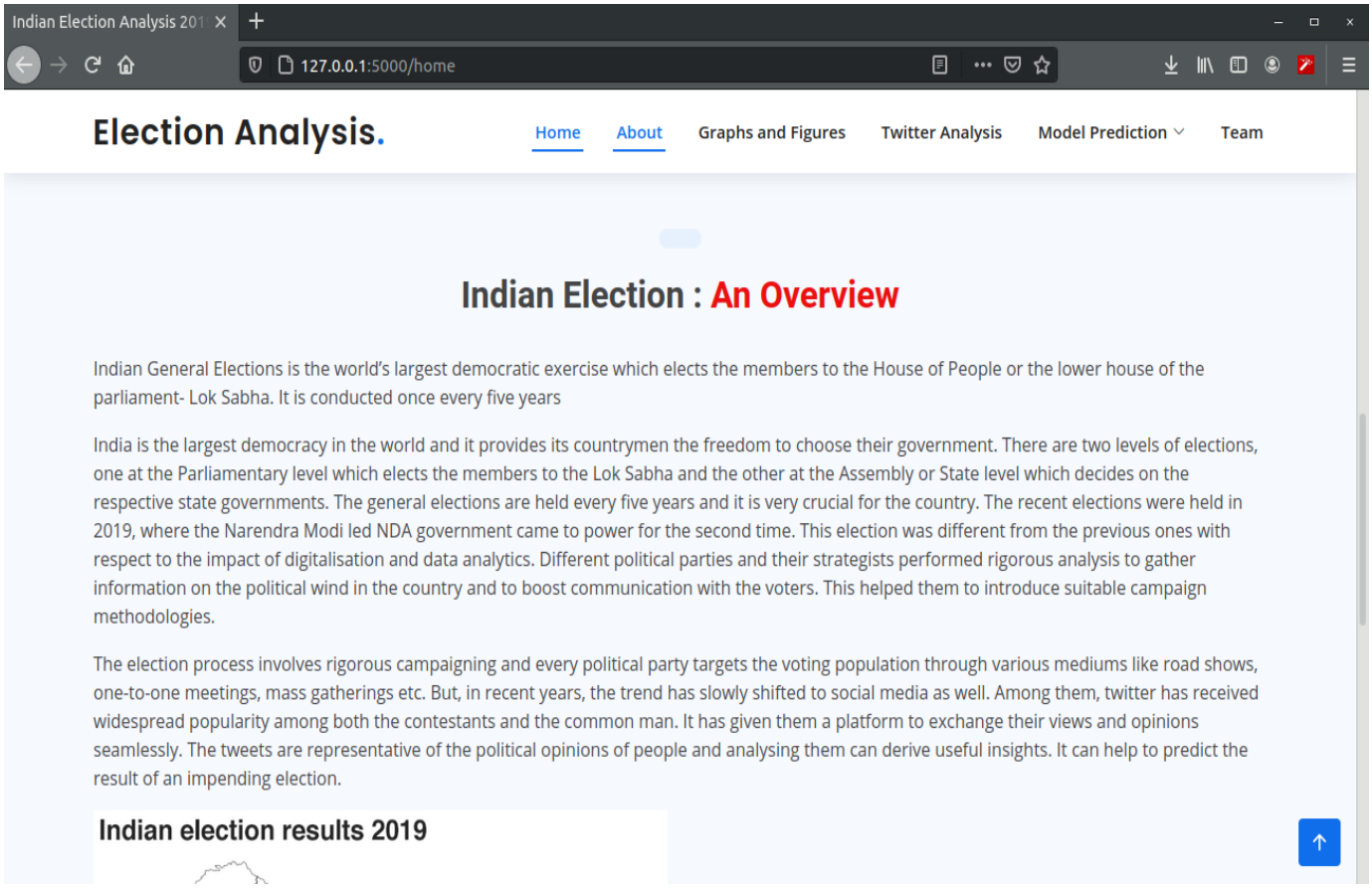
https://electionanalysis.herokuapp.com

Screenshots of Website

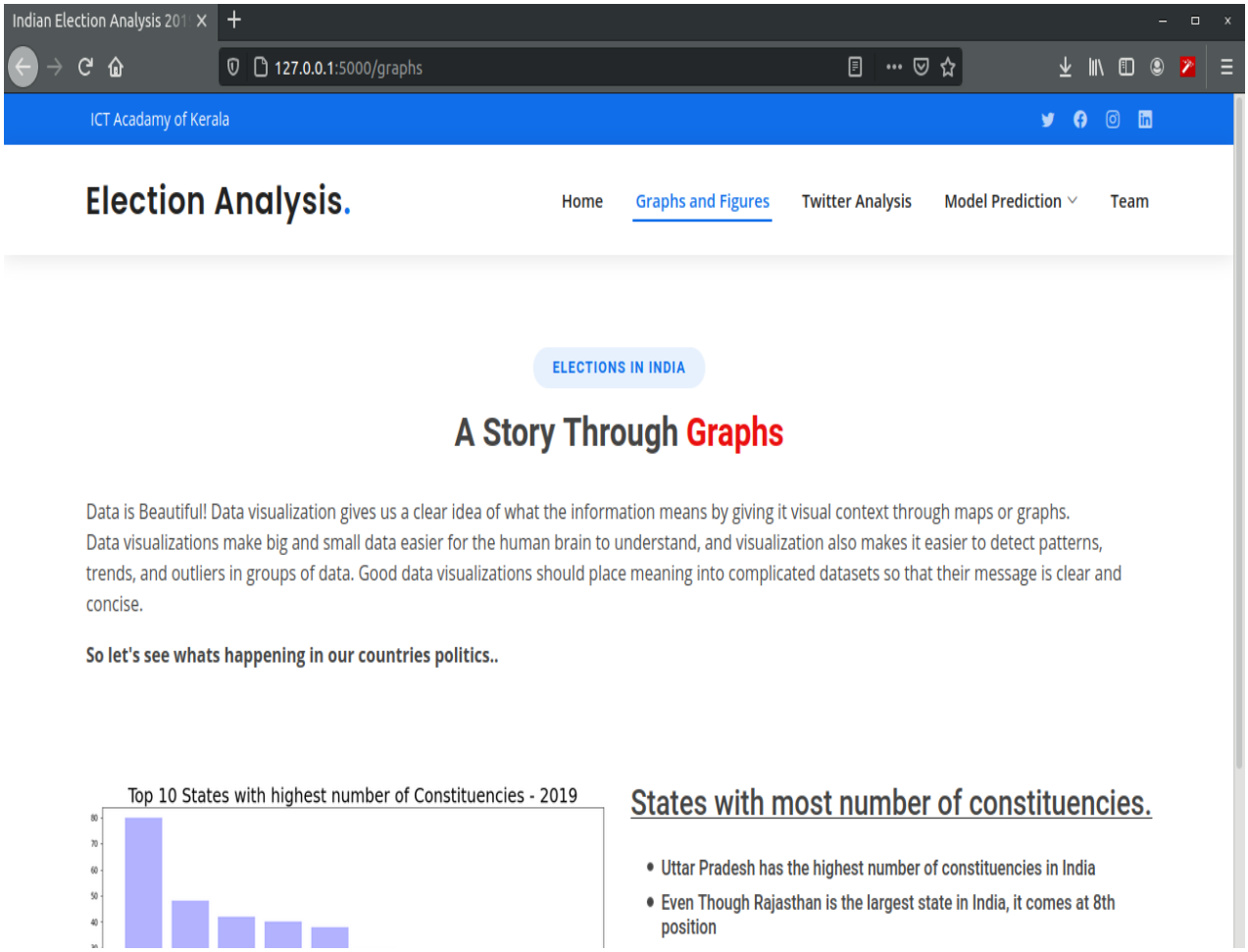
1.Home page



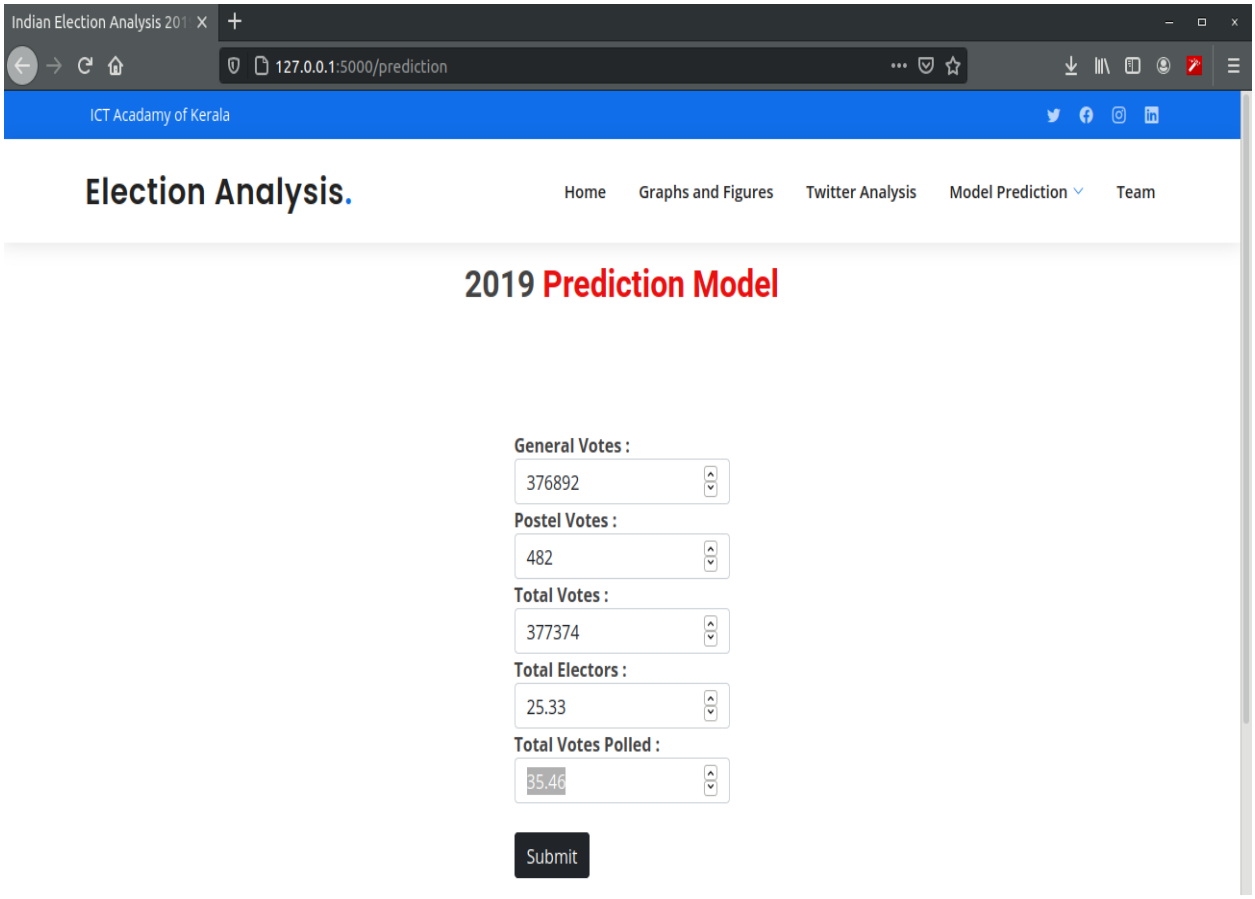
2.About


Indian Election Analysis 201
127.0.0.1:5000/home
Election Analysis.
Home
About
Graphs and Figures
Twitter Analysis
Model Prediction
Team
Indian Election : An Overview
Indian General Elections is the world's largest democratic exercise which elects the members to the House of People or the lower house of the parliament- Lok Sabha. It is conducted once every five years
India is the largest democracy in the world and it provides its countrymen the freedom to choose their government. There are two levels of elections, one at the Parliamentary level which elects the members to the Lok Sabha and the other at the Assembly or State level which decides on the respective state governments. The general elections are held every five years and it is very crucial for the country. The recent elections were held in 2019, where the Narendra Modi led NDA government came to power for the second time. This election was different from the previous ones with respect to the impact of digitalisation and data analytics. Different political parties and their strategists performed rigorous analysis to gather information on the political wind in the country and to boost communication with the voters. This helped them to introduce suitable campaign methodologies.
The election process involves rigorous campaigning and every political party targets the voting population through various mediums like road shows, one-to-one meetings, mass gatherings etc. But, in recent years, the trend has slowly shifted to social media as well. Among them, twitter has received widespread popularity among both the contestants and the common man. It has given them a platform to exchange their views and opinions seamlessly. The tweets are representative of the political opinions of people and analysing them can derive useful insights. It can help to predict the result of an impending election.
Indian election results 2019

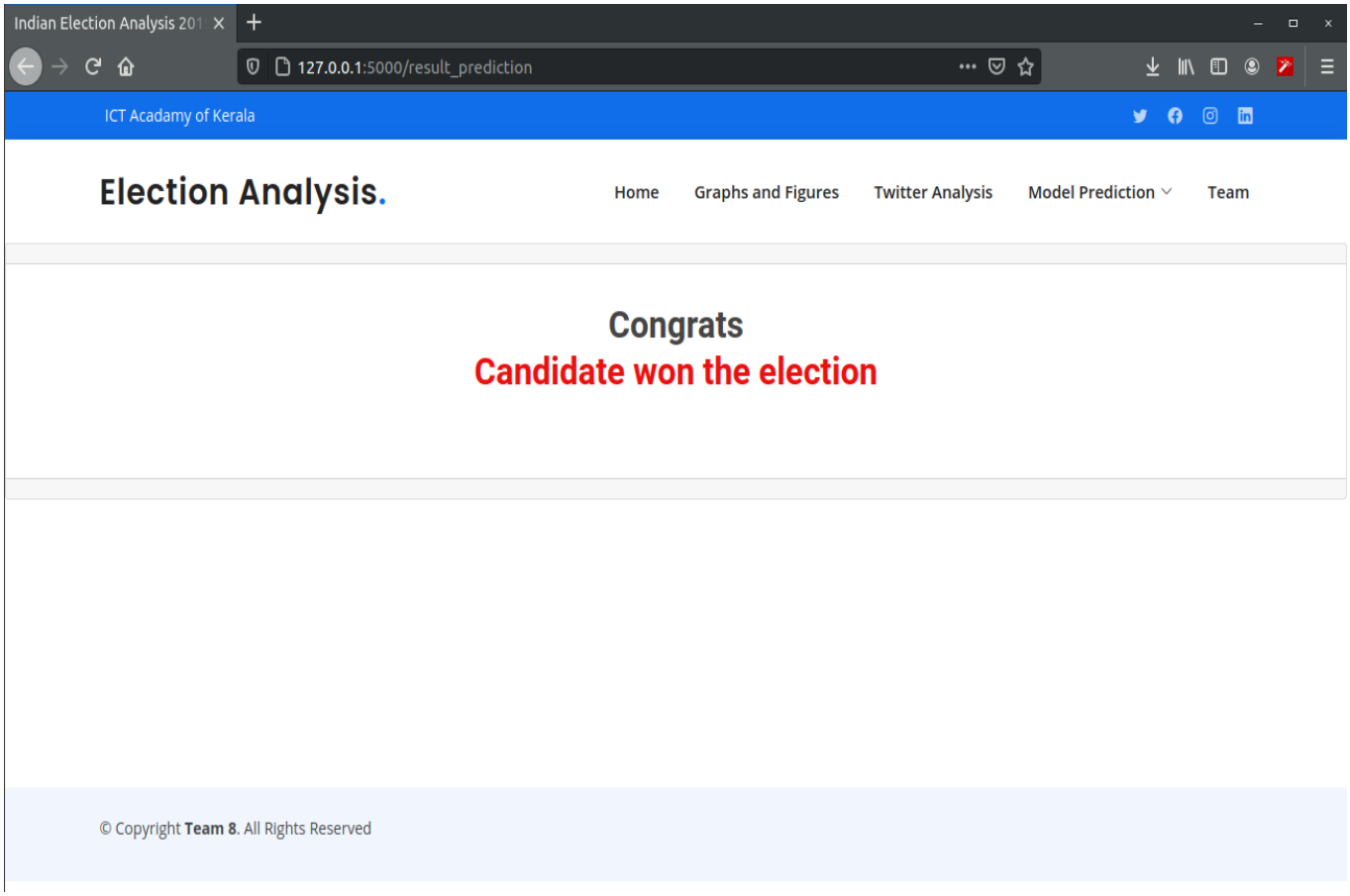
3.Graphs and Figures



4.Prediction



5. Result


Indian Election Analysis 201
127.0.0.1:5000/result_prediction
ICT Acadamy of Kerala
Election Analysis.
Home
Graphs and Figures
Twitter Analysis
Model Prediction
Team
Congrats
Candidate won the election
© Copyright Team 8. All Rights Reserved

14. Advantages & Limitations

Advantages

1. Sentiment Analysis on single day data was equally effective to data collected over a long period as in various research works. This was because of the choice of the tweets based on popular campaigns
2. Better Classification results were achieved on both validation and test data

Limitations

1. Tweets in regional languages written in English were not analysed .
2. Classification algorithms can be optimized further

15. Conclusion & Future Scope

The result data analysis and modeling was done on the Indian general election held in the year 2019 . The analysis and visualization showed a strong hold of the National Democratic Alliance (NDA) in the Indian politics. The modeling of the result was a binary classification problem and it was done using various Supervised Learning algorithms and we arrived at the best algorithm. In addition to this, Sentiment Analysis was done on another dataset with Twitter data to grasp an idea about the popularity of parties. Modeling was also done based on the sentiment scores. Both these approaches were found to agree with the actual results.

As future work, more optimized classifiers can be generated. Also, regional language tweets both written in English and otherwise have to be analysed for better performance.

16. References

[1] Ferdin Joe John Joseph, Faculty of Information Technology "*Twitter Based Outcome Predictions of 2019 Indian General Elections Using Decision Tree*",2019 4th International Conference on Information Technology (InCIT), Bangkok, Thailand

[2] Tetsuya Nasukawa, Jeonghee Yi, "*Sentiment Analysis --Capturing favorability using Natural Language Processing*", Conference Paper · January 2003, IBM Research Center

[3] Deshpande, P., Joshi, P., Madekar, D., Pawar, P., & Salunke, M. (2020). "*A Survey On: Classification of Twitter data Using Sentiment Analysis"*. Asian Journal For Convergence In Technology (AJCT), 5(3), 34-37.

[4] Venkateswarlu Bonta, Nandhini Kumaresh, N. Janardhan, *"A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis"* Asian Journal of Computer Science and Technology ISSN: 2249-0701 Vol.8 No.S2, 2019, pp. 1-6

[5] Richa Sharma ,Shweta Nigam, Rekha Jain, *"Opinion Mining of Movie Reviews at Document Level"*, August 2014, International Journal on Information Theory 3(3) DOI:10.5121/ijit.2014.3302,

[6] https://monkeylearn.com/blog/sentiment-analysis-examples/

[7] Priyavrat Chauhan, Nonita Sharma, Geeta Sikka, "*The emergence of social media data and sentiment analysis in election prediction*" February 2021 Journal of Ambient Intelligence and Humanized Computing 12(2) , DOI:10.1007/s12652-020-02423-y

[8] Singhal K., Agrawal B., Mittal N. (2015) , *Modeling Indian General Elections: Sentiment Analysis of Political Twitter Data.* In: Mandal J., Satapathy S., Kumar Sanyal M., Sarkar P., Mukhopadhyay A. (eds) Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing, vol 339. Springer, New Delhi. https://doi.org/10.1007/978-81-322-2250-7_46

[9] Bing Liu. Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012.

[10] Walaa Medhat, Ahmed Hassan , Hoda Korashy "Sentiment analysis algorithms and applications: A survey", May 2014 in Shams Engineering Journal 5(4) DOI:10.1016/j.asej.2014.04.011

[11] Michael Bironneau, Toby Coleman , "*Machine Learning with Go Quick Start Guide*" , May 2019

[12]https://cyfuture.com/blog/howbig-data-has-transformed-the-election-scenario-in-india

[13] https://seleritysas.com/blog/2020/10/22/big-data-analytics-and-its-role-in-elections/

[14] https://www.mathworks.com/discovery/sentiment-analysis.html