**ELECTION**

*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

**ICT\_DS Batch 1, Group 3**

Sandra Anna Shaji

Bineeth Mathew

Basil Chacko Mathew

Neenu Sebastian



**ICT ACADEMY OF KERALA**

**THIRUVANANTHAPURAM, KERALA, INDIA**

**Feb 2022**

**List of Figures**

**List of Abbreviations**

**Table of Contents**

**Abstract**

We have picked a dataset related to our ELECTION: Indian Election 2019 for our data science project. In this dataset, we have 2018 candidates who are participated in 2019 Lok Sabha election from 543 parliamentary constituencies of India. Data set comprises the details of each candidate from 29 states the country. And it also includes there corresponding part’s symbol, candidates gender, age, category, education, assets, number of votes.

Here our target variable is “winner”. The information was gathered in order to create prediction models that would predict the winner from each constituency. For this prediction we use regression models.

The following is how our inquiry will proceed: gain a thorough understanding of the datasets, prepare and ranked data, analysis the data, model the data and provide conclusion. To better understand the data’s patterns, trends we want to employ descriptive analytics. We can also find the party which has won the most constituencies. We indent to apply a verity of machine learning techniques to discover the best model for our dataset.

**1. Problem Definition**

**1.1 Overview**

Everything about Indian general elections is colossal - the Economist magazine once compared it to a "lumbering elephant embarking on an epic trek".The number of voters is bigger than the population of Europe and Australia combined.India's Centre for Media Studies estimated parties and candidates spent some $5bn (£3.8bn) for the 2014 elections. A total of 10 lakh polling stations were set up in 2019 as compared to around nine lakh in 2014. Looking at all these stats it is only going to get bigger in future. Now let's just jump into the analysis of 2019 Loksabha election data.

The purpose of using this specific language is due to its versatility, vast libraries (Pandas, Numpy, Matplotlib, etc.), speed limitations, and ease of learning. We will be analyzing large election data sets in this project which can not be easily analyzed in other tools as compared to python. Python does not have it’s limitation to only data analytics but also in many other fields such as Artificial intelligence, Machine learning, and many more.

**1.2 Problem Statement**

* The Dataset is based on Indian LokSabha Election.
* 539 Constituency participation data in this dataset.
* There were 2263 candidates who contested 2019 LokSabha Election.
* Minimum age of the candidates was 25 whereas maximum age was 86.
* Average age of all the candidates who contested election was 52.
* 19367 postal votes were casted in the election.

**2 . Introduction**

Lok Sabha elections are held every five years in order to elect the next Prime Minister of India. People vote for their preferred party and the candidate of the winning party goes on to represent the Prime Minister of our country. The Lok Sabha elections are also called general elections. The Lok Sabha has a maximum strength of 552 members, with two nominations by the President of India from the Anglo-Indian community. Twenty of the members are from union territories while 530 members are represented from the state.  
  
Any party which manages at least 273 seats is considered victorious and is eligible to form the central government. The parties are at liberty to announce their prime ministerial candidate either before the elections or after the victory. The Prime Ministerial candidate is elected by the party officials.

The **Lok Sabha**, or **House of the People**, is the [lowerhouse](https://en.wikipedia.org/wiki/Lower_house) of [India](https://en.wikipedia.org/wiki/India)’s [bicameral](https://en.wikipedia.org/wiki/Bicameralism) [Parliament](https://en.wikipedia.org/wiki/Parliament_of_India), with the [upper house](https://en.wikipedia.org/wiki/Upper_house) being the [Rajya Sabha](https://en.wikipedia.org/wiki/Rajya_Sabha). [Members of the Lok Sabha](https://en.wikipedia.org/wiki/Member_of_Parliament,_Lok_Sabha) are elected by an adult [universal suffrage](https://en.wikipedia.org/wiki/Universal_suffrage) and a [first-past-the-post](https://en.wikipedia.org/wiki/First-past-the-post) system to represent their respective [constituencies](https://en.wikipedia.org/wiki/List_of_constituencies_of_the_Lok_Sabha), and they hold their seats for five years or until the body is dissolved by the [President](https://en.wikipedia.org/wiki/President_of_India) on the advice of the [council of ministers](https://en.wikipedia.org/wiki/Union_Council_of_Ministers). The house meets in the Lok Sabha Chambers of the [Sansad Bhavan](https://en.wikipedia.org/wiki/Parliament_House_(India)), [New Delhi](https://en.wikipedia.org/wiki/New_Delhi).

The maximum membership of the House allotted by the [Constitution of India](https://en.wikipedia.org/wiki/Constitution_of_India) is 552 (Initially, in 1950, it was 500). Currently, the house has 543 seats which are made up by the election of up to 543 elected members and at a maximum. Between 1952 and 2020, [2 additional members](https://en.wikipedia.org/wiki/Anglo-Indian_reserved_seats_in_the_Lok_Sabha) of the [Anglo-Indian](https://en.wikipedia.org/wiki/Anglo-Indian) community were also nominated by the President of India on the advice of the [Government of India](https://en.wikipedia.org/wiki/Government_of_India), which was abolished in January 2020 by the [104th Constitutional Amendment Act, 2019](https://en.wikipedia.org/wiki/One_Hundred_and_Fourth_Amendment_of_the_Constitution_of_India). The Lok Sabha has a seating capacity of 550.

A total of 131 seats (24.03%) are reserved for representatives of [Scheduled Castes (84) and Scheduled Tribes (47)](https://en.wikipedia.org/wiki/Scheduled_Castes_and_Scheduled_Tribes). The quorum for the House is 10% of the total membership. The Lok Sabha, unless sooner dissolved, continues to operate for five years for time being from the date appointed for its first meeting. However, while a [proclamation of emergency](https://en.wikipedia.org/wiki/State_of_Emergency_in_India) is in operation, this period may be extended by [Parliament](https://en.wikipedia.org/wiki/Parliament_of_India) by law or decree.

One of the most critical ways that individuals can influence governmental decision-making is through voting. We know that everyone has the right to vote in our country. But many people are not aware of politics. So by this project, we can learn about the different political parties, their background history as well as their recent success and failure in the Lok Sabha election 2019 in India. Unfortunately, we have found few candidates with criminal history also. So through this data analysis, we can aware of the candidate's history as well as the nature of the political party. We can learn about the winning party and their success in 2019.

We specifically want to mention that it was an unbiased analysis. Here we have not supported any specific party.

# Dataset:

This Dataset is based on the Lok Sabha 2019 in India. There are a total of 2263 rows and 21 columns in this dataset. By using this dataset this data analysis project is created.

Here we use google COLAB or Jupyter to run these codes and analysis the dataset but you can use other platforms also to run the code.

**3. Literature Survey**

**4.Exploratory data analysis:**

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns , to spot anomalies , to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

# Bar Graph of crime Count in different states:

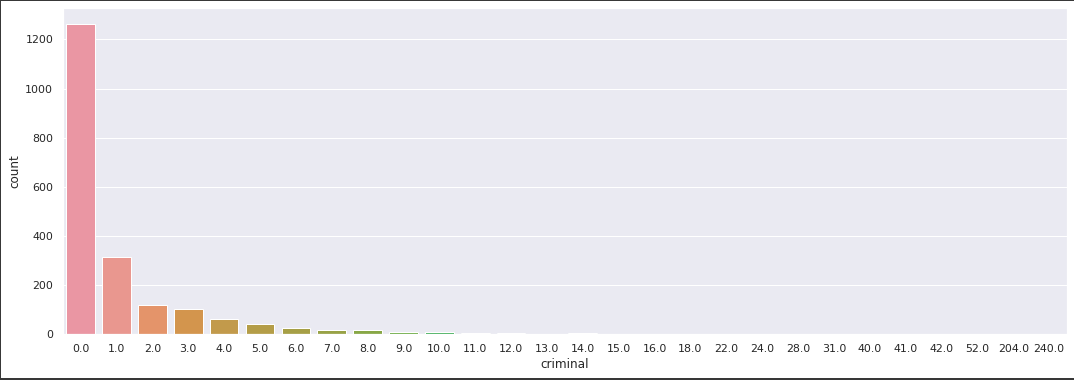


Fig 1: Graph of crime Count in different states

From the description given below, we can see that the mean of the crime among contestants is 1.45 where for the minimum crime , 25% and 50% of contestants did not make any crime but sadly in 75 % of total candidates the crime rate became 1.0. More surprisingly the maximum crime conceived by a person is 240, which’s huge.

# Line Graph of State vs Criminal Case:

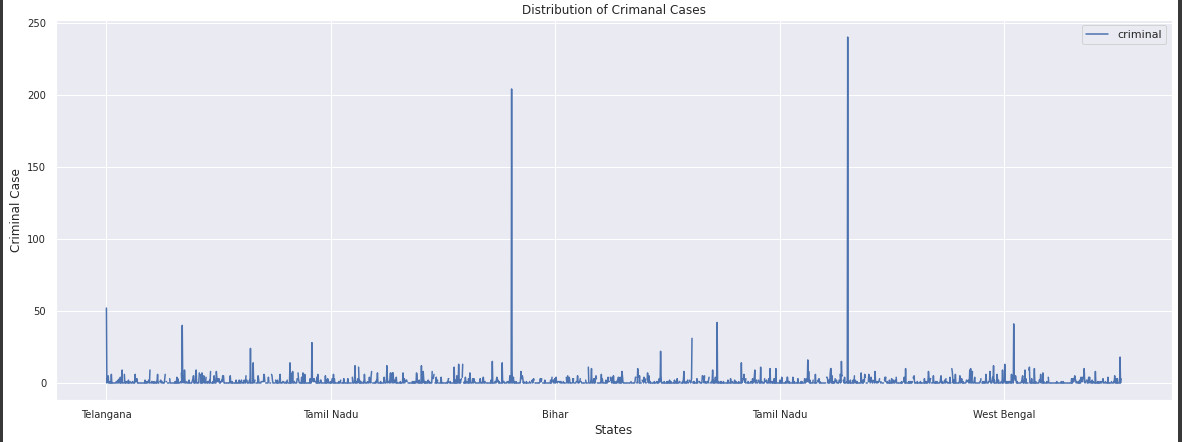


Fig 2: Graph of State vs Criminal Case

From this graph and the below description, we can see that the maximum no of criminal cases done by a single person is 240.

# The Educational Qualification of the Candidates:

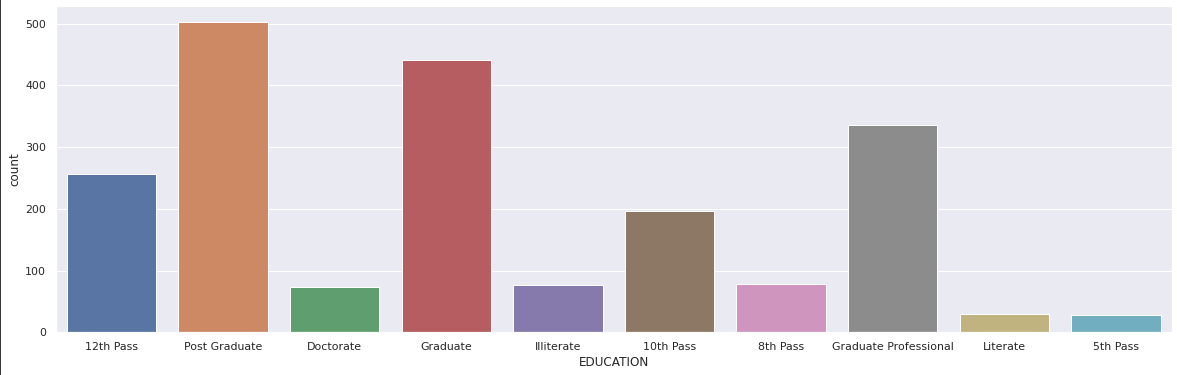


Fig 3: Graph of Educational Qualification of the Candidates

After analyzing the graph, we can see that there are two columns of class VIII pass and class V pass. But we believe the minimum qualification to be called literate is X pass. So we convert all V pass and VIII candidates as illiterate.

We can see that the number of postgraduate candidates in India is maximum(officially). So this is a positive site from the educational point of view.

# ****Education vs Crime Cases Bargraph:****

This graph represents the candidate's educational qualification vs criminal cases they have. Now we are aware of their previous criminal background with their educational qualification.

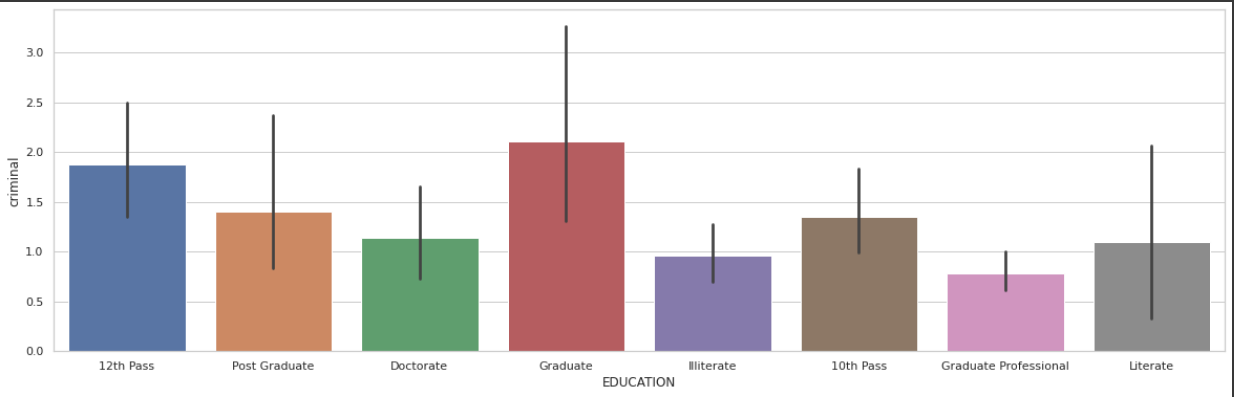


Fig 4: **Education vs Crime Cases Bargraph**

We can analyze from the graph that Graduate and 12th Pass criminal candidates are maximum. Especially we want to mention that a single graduate person has done 240 crimes.

# Pie chart of Male vs Female candidates:

This graph represents the male and female candidates who participated in Lok Sabha 2019.

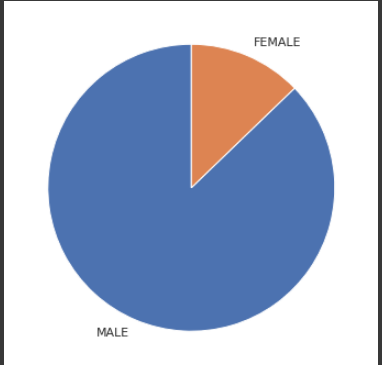


Fig 5: Pie chart of Male vs Female candidates

From this pie chart, we can see that the number of male candidates is much greater than the number of female candidates.

# State-wise Candidates with Crime Cases:

This is the bar graph of state-wise criminal case contestants and state-wise criminal case winners. The number of candidates with criminal cases is maximum in Bihar, Kerala, Maharashtra, West Bengal, Uttar Pradesh states.

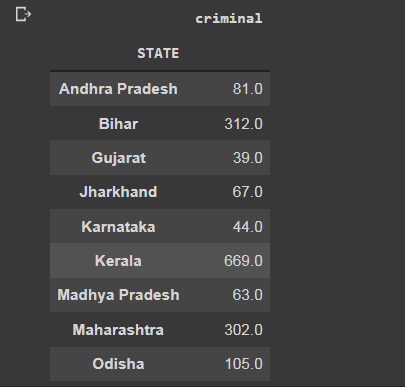
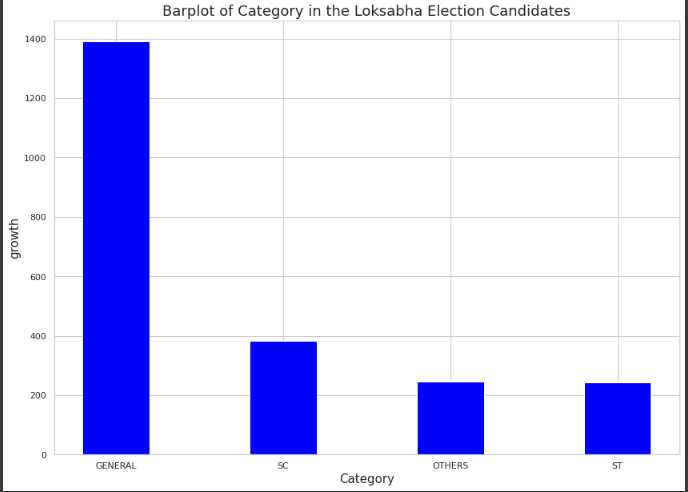


Fig 6: State-wise Candidates with Crime Cases

Here we can see the crime case across the states of candidates and winners. Here the maximum height of the bar graph is showing in the state of Kerala but West- Bengal, Uttar Pradesh, and Telangana are not far behind.

# Bar Graph of category Growth:

Here we calculating the number of SC, ST, and GENERAL candidates in the Lok Sabha election 2019.

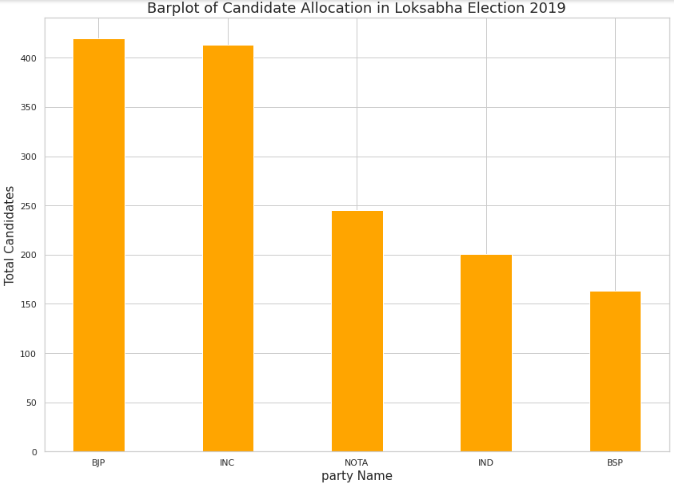
****

**Fig 7:** Bar Graph of category Growth

From the graph, we can see that the number of general candidates is maximum in India. The difference between general and other categories is very high.

# Bar Graph of Candidate Allocation in Loksabha Election 2019:

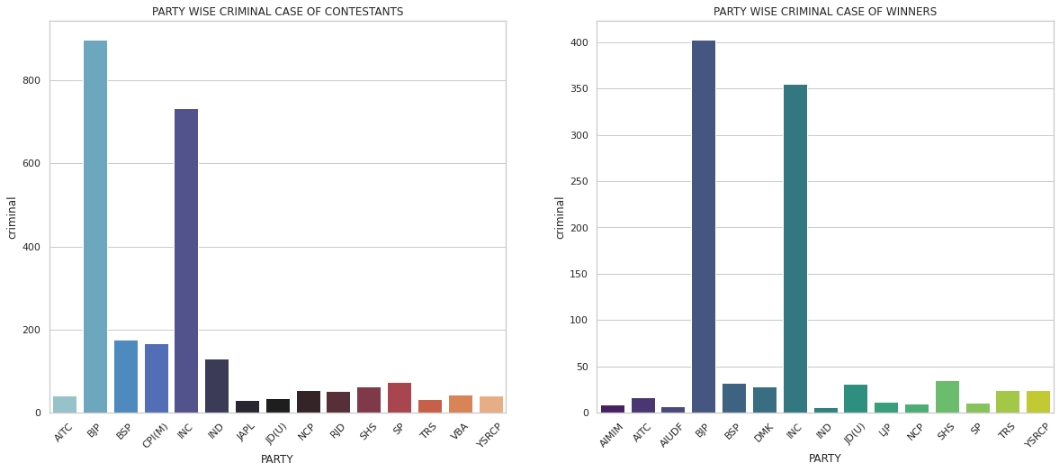
Here we are counting the total number of allocation of candidates for different parties in different constituencies in India.

****

**FIG 8:** Bar Graph of Candidate Allocation in Loksabha Election.

# Bar Graph of Party vs Candidates with Crime Case:

Here we are calculating the criminal case candidates in different parties. From that knowledge, we can aware of the criminal cases of the different parties.

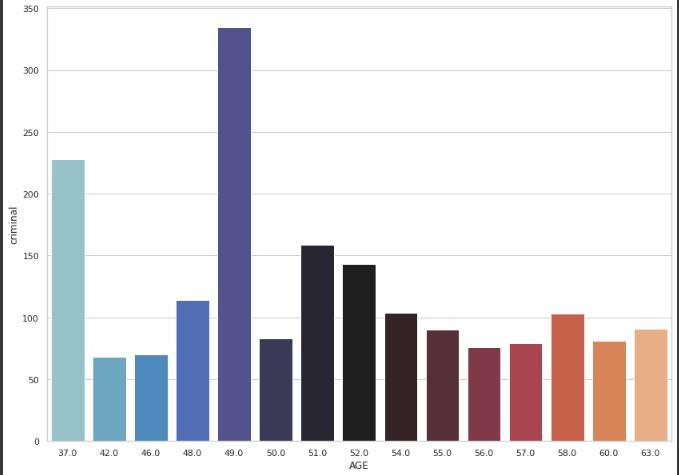


**Fig 9:** Bar Graph of Party vs Candidates with Crime Case

From the above diagram, we can see that the BJP and Congress parties have the maximum number of criminal cases in India. This is because that, these two parties are all India-based whereas most of the other parties are regional parties.

# Bar Graph of Age vs Crime Cases:

From this bar graph can know about the criminal cases of the candidates of different age group.

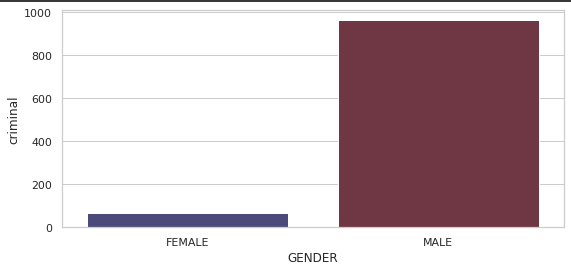
****

**Fig 10:** Bar Graph of Age vs Crime Cases

From the graph, we can notice that the criminal cases history is maximum at the age of 49,37, and 51.

# Bar Graph of Gender vs Crime:

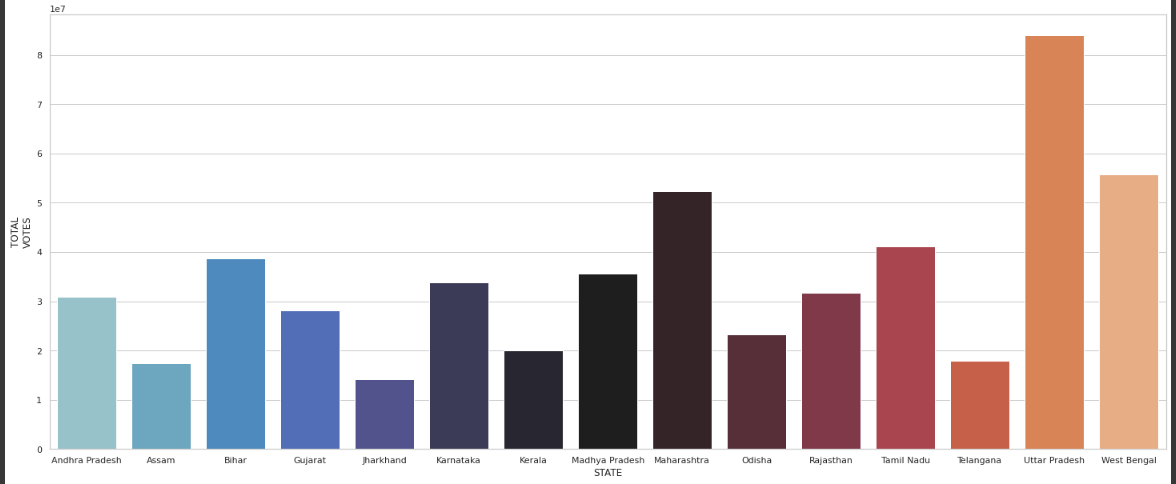
This is the bar plot of gender vs Crime from which we can know that the number of female candidates is maximum or the number of male candidates is maximum in India.

****

**Fig 11:** Bar Graph of Gender vs Crime

# Bar Graph of State vs Total Votes:

From this bar graph, we can get the knowledge about the no of votes in different states.

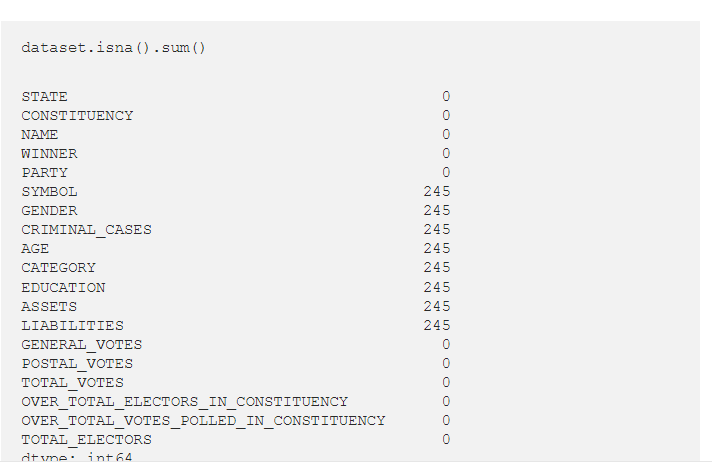
****

**Fig 12:** Bar Graph of State vs Total Votes

From the barplot, it is transparent that the total number of votes in Maharastra, Uttar Pradesh, and West Bengal are very much higher than the remaining states in India and Uttar Pradesh holds first place in the total number of votes.

**Data preprocessing**

* **Checking missing values**

****

You can see that 10% of the row values are missing. There are multiple ways to treat missing values like deleting them, using back-fill or forward-fill, constant value imputation, mean/ median or mode imputation, etc. However, I just delete these rows here for the simplicity (only 10% is missing), but always remember that removing values will make the prediction model less accurate. You should try to impute missing values as much as possible.

# drop rows with NA values  
dataset = dataset[dataset['GENDER'].notna()]

Let’s convert ASSETS, LIABILITIES and CRIMINAL\_CASES columns numeric because they represent money and count, and numeric will make sense to the models. In order to do that, we have to remove the ‘Rs’ sign, ‘\n’ character and commas in each value field. But also those columns contain ‘Nil’ and ‘Not Available’ as values. So before making them numeric, we have to replace those values with some meaningful value (for the moment I replaced them with 0).

# replace Nil values with 0  
dataset['ASSETS'] = dataset['ASSETS'].replace(['Nil', '`', 'Not Available'], '0')  
dataset['LIABILITIES'] = dataset['LIABILITIES'].replace(['NIL', '`', 'Not Available'], '0')  
dataset['CRIMINAL\_CASES'] = dataset['CRIMINAL\_CASES'].replace(['Not Available'], '0')  
  
# clean ASSETS and LIABILITIES column values  
dataset['ASSETS'] = dataset['ASSETS'].map(lambda x: x.lstrip('Rs ').split('**\n**')[0].replace(',', ''))  
dataset['LIABILITIES'] = dataset['LIABILITIES'].map(lambda x: x.lstrip('Rs ').split('**\n**')[0].replace(',', ''))  
  
# convert ASSETS, LIABILITIES and CRIMINAL\_CASES column values into numeric  
dataset['ASSETS'] = dataset['ASSETS'].astype(str).astype(float)  
dataset['LIABILITIES'] = dataset['LIABILITIES'].astype(str).astype(float)  
dataset['CRIMINAL\_CASES'] = dataset['CRIMINAL\_CASES'].astype(str).astype(int)

# ****Encode Existing Features more Meaningfully****

If we consider the EDUCATION column, it contains 11 categorical values which a particular candidate can have.

If we think thoroughly, each of these value represents a particular level of education. By label encoding, we just assign some random integer for each of these values without thinking about their hierarchical level. However, if we assign that integer meaningfully according to the educational qualification, hopefully, the model will perform better. Note that there is a field value named as “Other”, which we don’t know the hierarchical position. I just assign a median value from the range for it.

dataset['EDUCATION'].value\_counts()Post Graduate 502  
Graduate 441  
Graduate Professional 336  
12th Pass 256  
10th Pass 196  
8th Pass 78  
Doctorate 73  
Others 50  
Literate 30  
5th Pass 28  
Not Available 22  
Illiterate 5  
Post Graduate\n 1  
Name: EDUCATION, dtype: int64  
# encode education column  
encoded\_edu = []# iterate through each row in the dataset  
for row in dataset.itertuples():  
 education = row.EDUCATIONif education == "Illiterate":  
 encoded\_edu.append(0)  
 elif education == "Literate":  
 encoded\_edu.append(1)  
 elif education == "5th Pass":  
 encoded\_edu.append(2)  
 elif education == "8th Pass":  
 encoded\_edu.append(3)  
 elif education == "10th Pass":  
 encoded\_edu.append(4)  
 elif education == "12th Pass":  
 encoded\_edu.append(7)  
 elif education == "Graduate":  
 encoded\_edu.append(8)  
 elif education == "Post Graduate":  
 encoded\_edu.append(9)  
 elif education == "Graduate Professional":  
 encoded\_edu.append(10)  
 elif education == "Doctorate":  
 encoded\_edu.append(11)  
 else:  
 encoded\_edu.append(5)dataset['EDUCATION'] = encoded\_edu

let’s have some analysis of the PARTY column. If we list the number of candidates in front of each party, we can see that only a few parties have a significant number of candidates. The whole purpose of including the PARTY column is to have the impact of party of the candidate for the winning of the election. If the party doesn’t have a significant number of candidates, impact of that party for the winning of the candidate is low. So we can encode all of them into one common category (I encoded them as “other”).

dataset['PARTY'].value\_counts()BJP 420  
INC 413  
IND 201  
BSP 163  
CPI(M) 100  
 ...   
AINRC 1  
SKM 1  
ANC 1  
YKP 1  
AJSUP 1  
Name: PARTY, Length: 132, dtype: int64  
# change party of the less frequent parties as Other  
# 'BJP','INC','IND','BSP', 'CPI(M)', 'AITC', 'MNM': high frequent  
# 'TDP', 'VSRCP', 'SP', 'DMK', 'BJD': medium frequentdataset.loc[~dataset["PARTY"].isin(['BJP','INC','IND','BSP', 'CPI(M)', 'AITC', 'MNM', 'TDP', 'VSRCP', 'SP', 'DMK', 'BJD']), "PARTY"] = "Other"  
dataset['PARTY'].value\_counts()

* **LabelEncoder**

Now let’s make non-numeric columns numeric for better performance. Note that some model types cannot perform with non-numerical data. Here I’m focusing on a classification algorithm, which certainly cannot train on non-numerical data. So I label encoded those non-numerical columns using sklearn LabelEncoder.

# label encode categorical columns  
  
lblEncoder\_state = LabelEncoder()  
lblEncoder\_state.fit(dataset['STATE'])  
dataset['STATE'] = lblEncoder\_state.transform(dataset['STATE'])  
  
lblEncoder\_cons = LabelEncoder()  
lblEncoder\_cons.fit(dataset['CONSTITUENCY'])  
dataset['CONSTITUENCY'] = lblEncoder\_cons.transform(dataset['CONSTITUENCY'])  
  
lblEncoder\_name = LabelEncoder()  
lblEncoder\_name.fit(dataset['NAME'])  
dataset['NAME'] = lblEncoder\_name.transform(dataset['NAME'])  
  
lblEncoder\_party = LabelEncoder()  
lblEncoder\_party.fit(dataset['PARTY'])  
dataset['PARTY'] = lblEncoder\_party.transform(dataset['PARTY'])  
  
lblEncoder\_symbol = LabelEncoder()  
lblEncoder\_symbol.fit(dataset['SYMBOL'])  
dataset['SYMBOL'] = lblEncoder\_symbol.transform(dataset['SYMBOL'])  
  
lblEncoder\_gender = LabelEncoder()  
lblEncoder\_gender.fit(dataset['GENDER'])  
dataset['GENDER'] = lblEncoder\_gender.transform(dataset['GENDER'])  
  
lblEncoder\_category = LabelEncoder()  
lblEncoder\_category.fit(dataset['CATEGORY'])  
dataset['CATEGORY'] = lblEncoder\_category.transform(dataset['CATEGORY'])  
  
lblEncoder\_edu = LabelEncoder()  
lblEncoder\_edu.fit(dataset['EDUCATION'])  
dataset['EDUCATION'] = lblEncoder\_edu.transform(dataset['EDUCATION'])

Now let’s train a K-Nearest Neighbors model and see the accuracy. KNN is a supervised machine learning model which is categorized under classification algorithms. The algorithm works by taking a data point and finding out the k closest data points.

*# separate train features and label*  
y = dataset["WINNER"]  
X = dataset.drop(labels=["WINNER"], axis=1)*# split dataset into train and test data*  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1, stratify=y)# train and test knn model  
knn = KNeighborsClassifier()  
knn.fit(X\_train, y\_train)knn.predict(X\_test)  
print("Testing Accuracy is: ", knn.score(X\_test, y\_test)\*100, "%")**Testing Accuracy is: 70.79207920792079 %**

The model has achieved 70% accuracy without much effect. Let’s normalize the dataset and see how the accuracy improves. I have used MinMaxScaler from the scikit-learn library to scale down all the values into the 0–1 range.

As you can see below, accuracy has improved well by just normalizing the column values. However, we could further improve accuracy by applying a few other feature engineering techniques.

**Testing Accuracy is: 90.5940594059406 %**

# ****Feature Importance****

Removing irrelevant features or less important features will improve the accuracy of a model. So far we have trained our latest model on 21 . However, some of these features may not contribute a lot to derive conclusions based on the training data. Removing those features will not decrease the accuracy of the model, hopefully, will increase it, because irrelevant features may draw invalid inferences for the model.

First, we can remove some features by just analyzing their contribution. The NAME column won’t make any useful inferences for the model, because ideally, the name should be unique. However, the surname can be important in some cases, because some family names might have an impact on winning an election. Also the PARTY and SYMBOL both variables will represent the same feature, and we will be able to remove one of them without any impact on the accuracy. TOTAL\_VOTES column contains the sum of GENERAL\_VOTES column and POSTAL\_VOTES column. So we can remove those two as well. If we plot the heat map representing the correlation matrix, we will see that those 3 features will be highly correlated.

# remove unnecessary columnsX.drop(labels=["NAME"], axis=1, inplace=True)  
X.drop(labels=["SYMBOL"], axis=1, inplace=True)  
X.drop(labels=["POSTAL\_VOTES"], axis=1, inplace=True)  
X.drop(labels=["GENERAL\_VOTES"], axis=1, inplace=True)

Outer detection

An outlier is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error. The analysis of outlier data is referred to as outlier analysis or outlier mining.

Why outlier analysis?

Most data mining methods discard outliers noise or exceptions, however, in some applications such as fraud detection, the rare events can be more interesting than the more regularly occurring one and hence, the outlier analysis becomes important in such case.

## 1. Box plots

Box plots are a visual method to identify outliers. Box plots is one of the many ways to visualize data distribution. Box plot plots the q1 (25th percentile), q2 (50th percentile or median) and q3 (75th percentile) of the data along with (q1–1.5\*(q3-q1)) and (q3+1.5\*(q3-q1)). Outliers, if any, are plotted as points above and below the plot.

## 2. IQR method

IQR method is used by box plot to highlight outliers. IQR stands for interquartile range, which is the difference between q3 (75th percentile) and q1 (25th percentile). The IQR method computes lower bound and upper bound to identify outliers.

Lower Bound = q1–1.5\*IQR

Upper Bound = q3+1.5\*IQR

Any value below the lower bound and above the upper bound are considered to be outliers. Below is the implementation of IQR method in Python.

## 3. Z-score method

Z-score method is another method for detecting outliers. This method is generally used when a variable’ distribution looks close to Gaussian. Z-score is the number of standard deviations a value of a variable is away from the variable’ mean.

Z-Score = (X-mean) / Standard deviation

**7. Result**

**8. Conclusion**

**References**