

A Short Technical Report towards - A8518 - Python for Data Analytics

Course End Project

Indian Elections Data Analysis

Submitted in the Partial Fulfillment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted

By

Poojitha	22881A1287
Sahithi Reddy	22881A12B0
Vineesha	22881A12A5
Ifra	22881A12B6

Under the Esteemed Guidance of

Ms.Ch .Dhanalaxmi

Assistant Professor



DEPARTMENT OF INFORMATION TECHNOLOGY

VARDHAMAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified
Kacharam, Shamshabad, Hyderabad - 501218, Telangana, India.

2023- 24

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express my deep sense of gratitude to **Ms. Ch DhanaLaxmi**, Assistant Professor, for her able guidance and useful suggestions, which helped us in completing the design and implementation part of project in time.

We particularly thankful to **Dr.G Suryanaryana**, Associate Professor & Head, Department of information technology for his guidance, intense support and encouragement, which helped us to mold our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R. Ravindra**, for having provided all the facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartfelt thanks to **Dr. TeegalaVijender Reddy**, Chairman and **Sri TeegalaUpender Reddy**, Secretary of VCE, for providing congenial atmosphere to complete this project successfully.

We also thank all the **Teaching and Non-Teaching** members for their valuable support and generous advice. Finally, thanks to all our friends and family members for their continuous support and enthusiastic help.



VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD

An Autonomous Institute affiliated to JNTUH

Department of Information Technology

CERTIFICATE

This is to certify that the short technical report work entitled “**Indian Election Data Analysis**” carried out by **Poojitha 22881A1287, Vineesha 22881A12A5, R.Sahithi 22881A12B0, Ifra 22881A12B6, A8518 - Python for Data Analytics** course and submitted to the Department of Information technology, in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology in Information technology** during the year 2023-24.

Name & Signature of the Instructor

Ms. Ch Dhanalaxmi
Assistant Professor

Name & Signature of the HOD

Dr.G Suryanaryana
HOD, IT

Abstract

This project aims to analyze Indian election data to uncover patterns, trends, and insights that can inform future political strategies and decision-making. Utilizing various data analysis and visualization techniques, this study examines voter demographics, party performance, and election outcomes over multiple years. The project leverages Python and relevant libraries to process and analyze large datasets, ultimately presenting findings in a clear and accessible manner.

CHAPTER 1

INTRODUCTION

1.1 Background

Elections in India are a significant democratic exercise that involves millions of voters and numerous political parties. Understanding the intricacies of election data can provide valuable insights into voter behavior, party performance, and election outcomes. This project focuses on analyzing election data from multiple past elections in India to identify trends and patterns.

1.2 Objectives

The primary objectives of this project are:

- To analyze historical election data from India.
- To identify patterns in voter turnout and party performance.
- To visualize election data to reveal insights.
- To develop predictive models for future election outcomes.

1.3 Scope

This project covers election data from the national level, focusing on the Lok Sabha (House of the People) elections. The analysis includes data from multiple election years, voter demographics, party-wise performance, and geographical voting patterns.

CHAPTER 2

PROBLEM STATEMENT

2.1 Problem Definition

Understanding election data is complex due to the vast and varied nature of the data involved. This project addresses the following problems:

- Lack of consolidated and accessible election data.
- Difficulty in identifying trends and patterns in voter behavior and election outcomes.
- Challenges in predicting future election results based on historical data.

2.2 Research Questions

The project seeks to answer the following questions:

- What are the trends in voter turnout over the years?
- How do demographic factors influence election outcomes?
- What are the key determinants of party performance?
- Can historical data be used to predict future election outcomes?

CHAPTER 3

ARCHITECTURE

3.1 Data Collection

Data for this project is sourced from the Election Commission of India, which provides detailed election results, voter demographics, and party performance data.

3.2 Data Processing

The data processing pipeline includes:

- Data cleaning: Handling missing values and inconsistent data.
- Data transformation: Normalizing and structuring data for analysis.
- Feature extraction: Identifying key features relevant to the analysis.

3.3 Data Analysis and Visualization

Tools and techniques used for analysis:

Python libraries: pandas, numpy, matplotlib, seaborn, and scikit-learn. Visualization tools: matplotlib, seaborn, and Plotly.

3.4 Predictive Modeling

Machine learning models are employed to predict future election outcomes. Techniques include:

- ❖ Regression analysis for turnout prediction.
- ❖ Classification algorithms for predicting winning parties.

Election Sentiment Analysis by Analyzing Tweets

Steps to be followed:

1. Importing necessary libraries
2. Loading datasets
3. Data preprocessing
4. Exploratory Data Analysis
5. Sentiment Analysis

Dataset features:

created_at: Date and time of tweet posted

tweet_id: Tweet's unique ID

tweet: Full tweet text

likes: Number of likes

retweet_count: Number of retweets

source: Utility used to post the tweet

user_id: User ID of tweet creator

user_name: Username of tweet creator

user_screen_name: Screen name of tweet creator

user_description: Self-description by tweet creator

user_join_date: Join date of tweet creator

user_followers_count: Followers count on tweet creator

user_location: Address was given on tweeter's profile

lat: Latitude parsed from user_location

long: Longitude parsed from user_location

city: City parsed from user_location

country: Country parsed from user_location

state: State parsed from user_location

state_code: State code parsed from user_location

collected_at: Date and time tweet data was mined from Twitter

Steps to be followed:

5. Importing necessary libraries
6. Loading datasets
7. Data preprocessing
8. Exploratory Data Analysis
9. Sentiment Analysis

Dataset features:

created_at: Date and time of tweet posted

tweet_id: Tweet's unique ID

tweet: Full tweet text

likes: Number of likes

retweet_count: Number of retweets

source: Utility used to post the tweet

user_id: User ID of tweet creator

user_name: Username of tweet creator

user_screen_name: Screen name of tweet creator

user_description: Self-description by tweet creator

user_join_date: Join date of tweet creator

user_followers_count: Followers count on tweet creator

user_location: Address was given on tweeter's profile

lat: Latitude parsed from user_location

long: Longitude parsed from user_location

city: City parsed from user_location

country: Country parsed from user_location

state: State parsed from user_location

state_code: State code parsed from user_location

collected_at: Date and time tweet data was mined from Twitter

CHAPTER 4

SOURCE CODE WITH OUTPUT

```
import pandas as pd  
  
raw_election_data = pd.read_csv('./indian-candidates-for-general-election-2019/LS_2.0.csv')  
  
raw_election_data
```

OUTPUT:

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL CASES	AGE	CATEGORY	EDUCATION	ASSETS	LIABILITIES	GENERAL VOTES	POSTAL VOTES	TOTAL VOTES	OVER TOTAL ELECTORS IN CONSTITUENCY	OVER TOTAL VOTES POLLED IN CONSTITUENCY	TOTAL ELECTORS
0	Telangana	ADILABAD	SOYAM BAPU RAO	1	BJP	Lotus	MALE	52	52.0	ST	12th Pass	Rs 30,99,414\n ~ 30 Lacs+	Rs 2,31,450\n ~ 2 Lacs+	376892	482	377374	25.330684	35.468248	1489790
1	Telangana	ADILABAD	Godam Nagesh	0	TRS	Car	MALE	0	54.0	ST	Post Graduate	Rs 1,84,77,888\n ~ 1 Crore+	Rs 8,47,000\n ~ 8 Lacs+	318665	149	318814	21.399929	29.964370	1489790
2	Telangana	ADILABAD	RATHOD RAMESH	0	INC	Hand	MALE	3	52.0	ST	12th Pass	Rs 3,64,91,000\n ~ 3 Crore+	Rs 1,53,00,000\n ~ 1 Crore+	314057	181	314238	21.092771	29.534285	1489790
3	Telangana	ADILABAD	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	13030	6	13036	0.875023	1.225214	1489790
4	Uttar Pradesh	AGRA	Satyapal Singh Baghel	1	BJP	Lotus	MALE	5	58.0	SC	Doctorate	Rs 7,42,74,036\n ~ 7 Crore+	Rs 86,06,522\n ~ 86 Lacs+	644459	2416	646875	33.383823	56.464615	1937690
...
2258	Maharashtra	YAVATMAL-WASHIM	Anil Jayram Rathod	0	IND	SHIP	MALE	0	43.0	GENERAL	Post Graduate	Rs 48,90,000\n ~ 48 Lacs+	Rs 10,20,000\n ~ 10 Lacs+	14661	25	14686	0.766419	1.250060	1916185

2263 rows x 19 columns

```
def convert(x):
    """
    Extract the numeric value from the passed string and return it as float
    """
    if str(x)[0] == 'R':
        # this is to ensure only valid values (and not NaN values) are converted
        return float(str(x).split()[1].replace(",", ""))
    return 0.0
# default 0

raw_election_data.ASSETS = raw_election_data.ASSETS.apply(convert)
raw_election_data.LIABILITIES = raw_election_data.LIABILITIES.apply(convert)
# convert the ASSETS and LIABILITIES to numeric data
# the above can also be done using lambda function

raw_election_data.sample(5)
# check if the applied operations were successful

raw_election_data.at[raw_election_data.EDUCATION == "Post Graduate\n", "EDUCATION"] = "Post Graduate"
raw_election_data.at[raw_election_data.EDUCATION == "Graduate Professional", "EDUCATION"] =
"Graduate\nProfessional"

raw_election_data.at[192, "WINNER"] = 1
raw_election_data.at[702, "WINNER"] = 1
raw_election_data.at[951, "WINNER"] = 1
raw_election_data.at[1132, "WINNER"] = 1
raw_election_data.at[172, "WINNER"] = 0
```

```
raw_election_data.columns
```

```
OUTPUT: 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')
```

```
candidates_df = raw_election_data.drop(['SYMBOL', 'GENERAL\nVOTES', 'POSTAL\nVOTES',
'OVER TOTAL ELECTORS \nIN CONSTITUENCY', 'OVER TOTAL VOTES POLLED \nIN CONSTITUENCY'],
axis=1)
# take out the unnecessary columns
```

```
candidates_df.rename(columns = {"CRIMINAL\nCASES": "CRIMINAL CASES", "TOTAL\nVOTES": "TOTAL VOTES"},
inplace = True)
candidates_df.sort_values(["STATE", "CONSTITUENCY"], inplace = True)
# rename some of the columns and sort the data with respect to State and Constituency columns
```

```
candidates_df.info()
```

OUTPUT :

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2263 entries, 105 to 2171
Data columns (total 14 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 GENDER 2018 non-null object
6 CRIMINAL CASES 2018 non-null object
7 AGE 2018 non-null float64
8 CATEGORY 2018 non-null object
9 EDUCATION 2018 non-null object
10 ASSETS 2263 non-null float64
11 LIABILITIES 2263 non-null float64
12 TOTAL VOTES 2263 non-null int64
13 TOTAL ELECTORS 2263 non-null int64

```

dtypes: float64(3), int64(3), object(8) memory usage: 265.2+ KB

```

candidates_df["CRIMINAL CASES"] = pd.to_numeric(candidates_df["CRIMINAL CASES"], errors =
'coerce').convert_dtypes()

```

candidates_df

OUTPUT:

WINNER	PARTY	GENDER	CRIMINAL CASES	AGE	CATEGORY	EDUCATION	ASSETS	LIABILITIES	TOTAL VOTES	TOTAL ELECTORS				
105	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	KULDEEP RAJ SHARMA	1	INC	MALE	0	52.0	GENERAL	Graduate\nProfessional	132233012.0	80450870.0	95308	318471
106	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	VISHAL JOLLY	0	BJP	MALE	1	46.0	GENERAL	Graduate\nProfessional	85140573.0	4633914.0	93901	318471
107	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	PARITOSH KUMAR HALDAR	0	IND	MALE	0	61.0	GENERAL	Post Graduate	19916000.0	1311000.0	5341	318471
108	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	SANJAY MESHAK	0	AAP	MALE	0	49.0	GENERAL	12th Pass	99887782.0	1791076.0	2839	318471
109	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	PRAKASH MINJ	0	BSP	MALE	0	43.0	SC	Post Graduate	1250000.0	240000.0	2486	318471
...
2042	West Bengal	TAMLUK	Lakshman Chandra Sethi	0	INC	MALE	5	73.0	SC	Doctorate	2045953.0	0.0	16001	1694646

Indian Elections Data Analysis

2168	West Bengal	ULUBERIA	SAJDA AHMED	1	AITC	FEMALE	0	57.0	GENERAL	Graduate	28351569.0	5322615.0	694945	1614988
2169	West Bengal	ULUBERIA	JOY BANERJEE	0	BJP	MALE	2	56.0	GENERAL	Graduate	246933.0	0.0	479586	1614988
2170	West Bengal	ULUBERIA	MAKSUDA KHATUN	0	CPI(M)	FEMALE	0	58.0	GENERAL	Doctorate	12578736.0	2876353.0	81314	1614988
2171	West Bengal	ULUBERIA	Shoma Ranisree Roy	0	INC	FEMALE	0	40.0	GENERAL	Post Graduate	5400000.0	0.0	27568	1614988

2263 rows x 14 columns

```
candidates_personal_df = candidates_df[candidates_df.NAME != "NOTA"]

candidates_personal_df = candidates_personal_df.drop(["TOTAL VOTES", "TOTAL ELECTORS"], axis = 1)

candidates_personal_df

OUTPUT:
```

Indian Elections Data Analysis

STATE	CONSTITUENCY	NAME	WINNER	PARTY	GENDER	CRIMINAL CASES	AGE	CATEGORY	EDUCATION	ASSETS	LIABILITIES	
105	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	KULDEEP RAI SHARMA	1	INC	MALE	0	52.0	GENERAL	Graduate\nProfessional	132233012.0	80450870.0
106	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	VISHAL JOLLY	0	BJP	MALE	1	46.0	GENERAL	Graduate\nProfessional	85140573.0	4633914.0
107	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	PARITOSH KUMAR HALDAR	0	IND	MALE	0	61.0	GENERAL	Post Graduate	19916000.0	1311000.0
108	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	SANJAY MESHACK	0	AAP	MALE	0	49.0	GENERAL	12th Pass	99887782.0	1791076.0
109	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	PRAKASH MINJ	0	BSP	MALE	0	43.0	SC	Post Graduate	1250000.0	240000.0
...
2042	West Bengal	TAMLUK	Lakshman Chandra Seth	0	INC	MALE	5	73.0	SC	Doctorate	2045953.0	0.0
2168	West Bengal	ULUBERIA	SAJDA AHMED	1	AITC	FEMALE	0	57.0	GENERAL	Graduate	28351569.0	5322615.0
2169	West Bengal	ULUBERIA	JOY BANERJEE	0	BJP	MALE	2	56.0	GENERAL	Graduate	246933.0	0.0
2170	West Bengal	ULUBERIA	MAKSUDA KHATUN	0	CPI(M)	FEMALE	0	58.0	GENERAL	Doctorate	12578736.0	2876353.0
2171	West Bengal	ULUBERIA	Shoma Ranisree Roy	0	INC	FEMALE	0	40.0	GENERAL	Post Graduate	5400000.0	0.0

2018 rows × 12 columns

```
candidates_personal_df.describe()
```

```
# works on only numeric data
```

OUTPUT:

	WINNER	CRIMINAL CASES	AGE	ASSETS	LIABILITIES
count	2018.000000	1996.000000	2018.000000	2.018000e+03	2.018000e+03
mean	0.268583	1.469940	52.273538	1.315849e+08	1.973860e+07
std	0.443332	7.677432	11.869373	4.122697e+08	8.945292e+07
min	0.000000	0.000000	25.000000	0.000000e+00	0.000000e+00
25%	0.000000	0.000000	43.250000	6.272526e+06	0.000000e+00
50%	0.000000	0.000000	52.000000	2.404181e+07	9.039245e+05
75%	1.000000	1.000000	61.000000	9.152498e+07	6.097971e+06
max	1.000000	240.000000	86.000000	8.950179e+09	1.547511e+09

```
winners_df = candidates_df[candidates_df.WINNER == 1].sort_values(["STATE", "CONSTITUENCY"]).reset_index()
```

```
# extract the list of winners
```

```
winners_df.drop(["index", "WINNER"], axis = 1, inplace = True)
```

```
winners_df
```

OUTPUT:

STATE	CONSTITUENCY	NAME	PARTY	GENDER	CRIMINAL CASES	AGE	CATEGORY	EDUCATION	ASSETS	LIABILITIES	TOTAL VOTES	TOTAL ELECTORS	
0	Andaman & Nicobar Islands	ANDAMAN & NICOBAR ISLANDS	KULDEEP RAI SHARMA	INC	MALE	0	52.0	GENERAL	Graduate\nProfessional	132233012.0	80450870.0	95308	318471

Indian Elections Data Analysis

1	Andhra Pradesh	AMALAPURAM	CHINTA ANURADHA	YSRCP	FEMALE	0	46.0	SC	Graduate	85993362.0	2431407.0	485958	1459556
2	Andhra Pradesh	ANAKAPALLI	Dr.Beesetti Venkata Satyavathi	YSRCP	FEMALE	0	52.0	GENERAL	Graduate\nProfessional	91526244.0	11139279.0	586226	1521363
3	Andhra Pradesh	ANANTAPUR	TALARI RANGAIAH	YSRCP	MALE	2	48.0	GENERAL	Doctorate	11773091.0	1135547.0	695208	1664160
4	Andhra Pradesh	ARUKU	GODDETI. MADHAVI	YSRCP	FEMALE	0	26.0	ST	Graduate\nProfessional	141179.0	0.0	562190	1451418
...
537	West Bengal	RAIGANJ	DEBASREE CHAUDHURI	BJP	FEMALE	2	48.0	GENERAL	Post Graduate	6139123.0	823416.0	511652	1601220
538	West Bengal	RANAGHAT	JAGANNATH SARKAR	BJP	MALE	1	56.0	SC	Graduate\nProfessional	17622476.0	0.0	783253	1762252
539	West Bengal	SRERAMPUR	KALYAN BANERJEE	AITC	MALE	0	62.0	GENERAL	Graduate\nProfessional	175968902.0	5522403.0	637707	1785472
540	West Bengal	TAMLUK	Adhikari Dibyendu	AITC	MALE	1	42.0	GENERAL	Graduate	37687867.0	8397501.0	724433	1694646
541	West Bengal	ULUBERIA	SAJDA AHMED	AITC	FEMALE	0	57.0	GENERAL	Graduate	28351569.0	5322615.0	694945	1614988

542 rows x 13 columns

```
print("Number of Parties which fielded at least 1 candidate: ", candidates_df.PARTY.unique().shape[0]-2)
# -2 : 1 for independent candidates and 1 for NOTA
```

OUTPUT:

Number of Parties which fielded at least 1 candidate: 131

```
print("Number of Independent Candidates who contested the elections: ", candidates_df[candidates_df.PARTY == 'IND'].shape[0])
```

OUTPUT:

Number of Independent Candidates who contested the elections: 201

```
print("Number of Parties which won at least 1 seat: ", winners_df.PARTY.unique().shape[0] - 1)
# -1 : for independent winners
```

OUTPUT:

Number of Parties which won at least 1 seat: 35

```
print("Number of Independent Winners: ", winners_df[winners_df.PARTY == 'IND'].shape[0])
```

OUTPUT:

Number of Independent Winners: 4

Exploratory Analysis and Visualization

```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (15, 10)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
import numpy as np
```

Seat Share

```
all_party_seats = winners_df.PARTY.value_counts().sort_values(ascending = False)
# frequen cy of each PARTY in the winner list
all_party_seats
```

OUTPUT:

BJP	303
INC	52
DMK	23
YSRCP	22
AITC	22
SHS	18
JD(U)	16
BJD	12
BSP	10
TRS	9
LJP	6
NCP	5
SP	5
CPI(M)	5

IND	4
IUML	3
JKN	3
TDP	3
SAD	2
AIMIM	2
ADAL	2
SKM	1
AJSUP	1
JMM	1
AIADMK	1
VCK	1
AAP	1
RLTP	1
KEC(M)	1
JD(S)	1
NDPP	1
MNF	1
RSP	1
NPF	1
AIUDF	1
NPEP	1

Name: PARTY, dtype: int64

```
others = all_party_seats[all_party_seats<10].sum()
# simplifying the output for visualization purposes
seat_distribution = all_party_seats[all_party_seats>=10].append(pd.Series({"Others":others}))
seat_distribution
```

OUTPUT:

BJP	303
INC	52
DMK	23
YSRCP	22
AITC	22
SHS	18
JD(U)	1 ;
BJD	12
BSP	10
Others	64

dtype: int64

Plot the distribution as a pie chart.

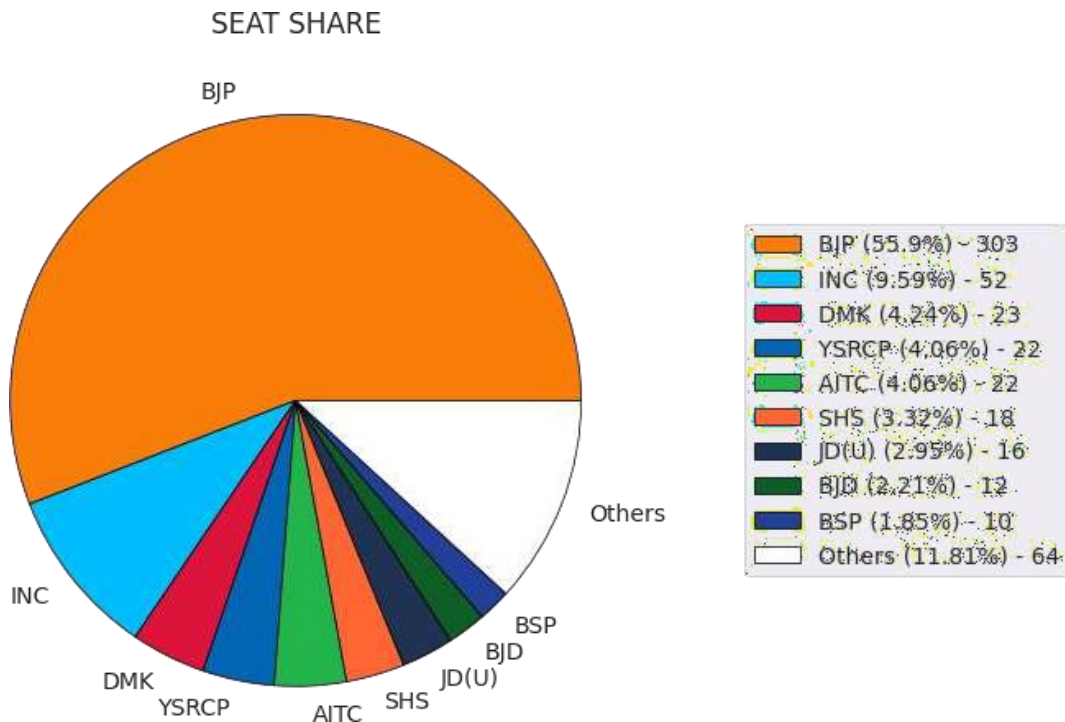
```
plt.figure(figsize = (10,8))
plt.title("SEAT SHARE")
# basic details

plt.pie(seat_distribution, labels = seat_distribution.index,
        colors = ['#f97d09', '#00bdfe', '#dc143c', '#0266b4', '#24b44c', '#ff6634',
                  '#203354', '#105e27', '#22409a', '#FFFFFF'],
        wedgeprops = {'edgecolor': 'black', 'linewidth': 0.75, 'antialiased': True})
# pie chart created using the Data, labels, colors, and wedge border properties
# colors are customised according to each party's colors

seat_percent = round((seat_distribution/seat_distribution.sum())*100,2)
```

```
legend = seat_percent.index + " (" + seat_percent.values.astype(str) + "%" ) - " + seat_distribution.values.astype(str)
# the legend would the percentage seat share of each party (& others)
```

```
plt.legend(legend, loc = "right", bbox_to_anchor = (1.6,0.5));
# legend is placed outside the main chart accordingly
```



```
plt.figure(figsize = (20,10))
plt.title("Age of Candidates Contested and Won", fontsize=20)
plt.xlabel("Age", fontsize=17)
plt.ylabel("Number of candidates", fontsize=17)
# put the basic labelling
```

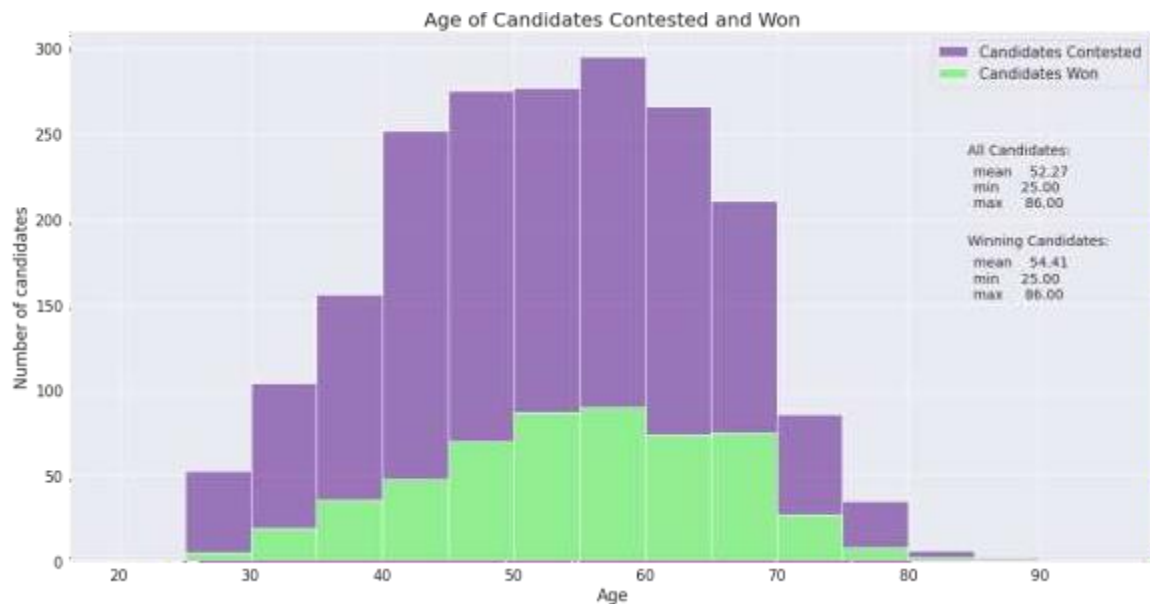
```
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
# axes ticks size
```

```
sns.histplot(data = candidates_personal_df, x = 'AGE', bins = np.arange(20,100,5), color = 'indigo', alpha = 0.5)
sns.histplot(data = winners_df, x = 'AGE', bins = np.arange(20,100,5), color = 'lightgreen', alpha = 1)
# two histograms plotted, Won over Contested to show the relative percentage
```

```
plt.legend(["Candidates Contested", "Candidates Won"], fontsize = 15)
# legend to the plot
```

```
plt.text(84.5, 238, "All Candidates:")
plt.figtext(0.77, 0.63, round(candidates_personal_df.describe().AGE[['mean', 'min', 'max']], 2).to_string())
```

```
plt.text(84.5, 185, "Winning Candidates:")
plt.figtext(0.77, 0.5, round(winners_df.describe().AGE[['mean', 'min', 'max']], 2).to_string());
# basic stats printed
```



```
print("Youngest Member of the House:")
winners_df[(winners_df.AGE == 25)][["NAME", "PARTY", "STATE", "CONSTITUENCY"]].reset_index(drop = True)
```

OUTPUT:

Youngest Member of the House:

	NAME	PARTY	STATE	CONSTITUENCY
0	Chandrani Murmu	BJD	Odisha	KEONJHAR



Chandrani Murmu

```
print("Oldest Member of the House:")
winners_df[(winners_df.AGE == 86)][["NAME", "PARTY", "STATE", "CONSTITUENCY"]].reset_index(drop = True)
```

OUTPUT:

Oldest Member of the House:

	NAME	PARTY	STATE	CONSTITUENCY
0	DR. SHAFIQR REHMAN BARQ	SP	Uttar Pradesh	SAMBHAL

Seat Category

```
seat_category = winners_df.CATEGORY.value_counts()
# winners_df has 1 constituency only 1 time, so analysing its CATEGORY column will give the correct result
```

OUTPUT:

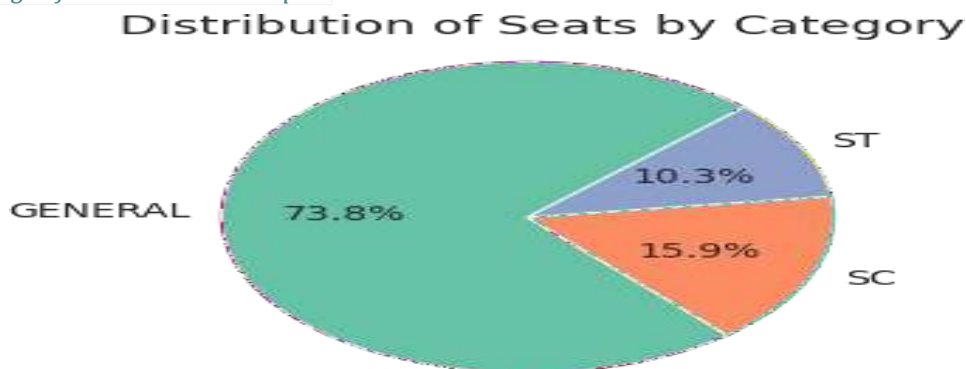
```
pd.DataFrame(seat_category)
```

CATEGORY	
GENERAL	400
SC	86
ST	56

Plot the distribution as a Pie chart:

```
sns.set_palette(sns.color_palette('Set2'))
plt.figure(figsize=(8,5))
plt.title("Distribution of Seats by Category", size=18, x = 0.52, y=0.95)

plt.pie(seat_category, labels = seat_category.index, autopct = '%1.1f%%', startangle = 45);
# percentage of seats shown on the plot
```



Gender

```
gender_group = candidates_personal_df.groupby(["GENDER", "WINNER"]).size()
gender_group = gender_group.unstack()
gender_group = gender_group[[1,0]]
# a2a from stack overflow

# gender with winning condition is extracted as a dataframe

sns.set_palette(sns.color_palette("icefire"))
# color palette set
gender_group.plot(kind = 'barh', figsize = (15,6), title = "Gender Comparison of Contesting and Winning Candidates")
# horizontal bar plot created with Pandas

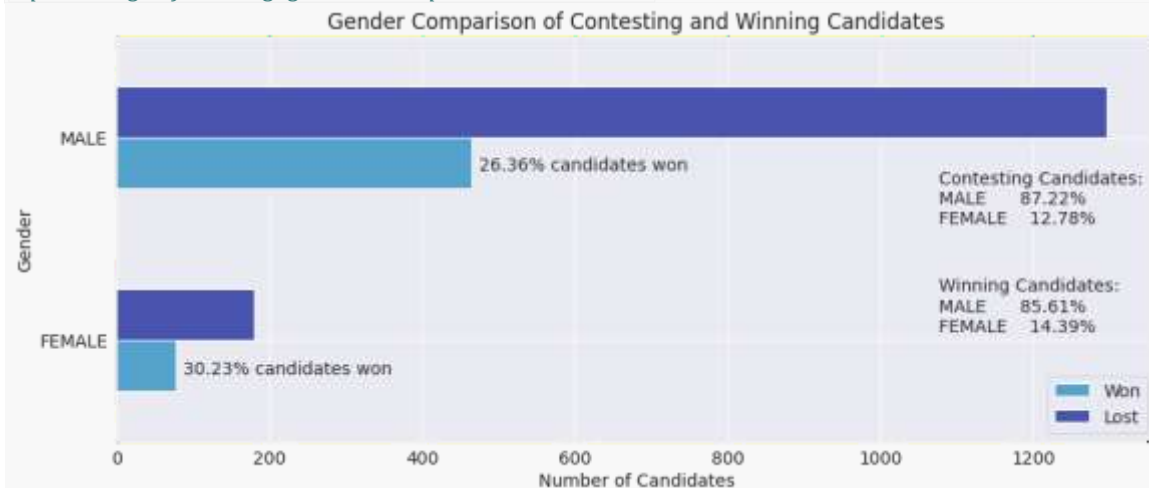
plt.legend(["Won", "Lost"])
plt.xlabel("Number of Candidates")
plt.ylabel("Gender")
# legend and labels set
```

```
plt.figtext(0.738,0.53, "Contesting Candidates:\n" +
            round((candidates_personal_df.GENDER.value_counts(normalize=True)*100),2).to_string().replace("\n",
            "%\n")+ "%")

plt.figtext(0.738,0.33, "Winning Candidates:\n" +
            round((winners_df.GENDER.value_counts(normalize=True)*100),2).to_string().replace("\n", "%\n")+ "%")

# Total candidates statistics (percentages) printed on the chart, with some applied String formatting to give the look

win_percent = round((winners_df.GENDER.value_counts()/candidates_personal_df.GENDER.value_counts())*100,2)
plt.figtext(0.395, 0.63, str(round(win_percent.MALE,2)) + "% candidates won")
plt.figtext(0.175, 0.25, str(round(win_percent.FEMALE,2))+ "% candidates won");
# percentage of winning, gender-wise printed on the chart
```



```
print("No. of male MPs: ", winners_df.GENDER.value_counts()["MALE"])
print("No. of female MPs: ", winners_df.GENDER.value_counts()["FEMALE"])
```

OUTPUT:

No. of male MPs: 464 No. of female MPs: 78

```
winners_df["EDUCATION"].unique()
```

OUTPUT:

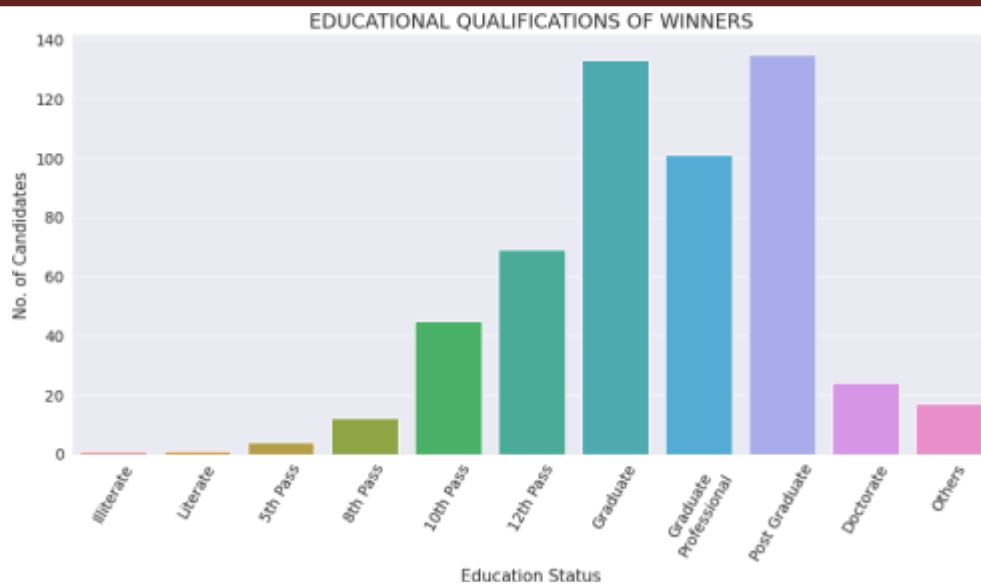
```
array(['Graduate\nProfessional', 'Graduate', 'Doctorate', '8th Pass',
      'Post Graduate', '12th Pass', '10th Pass', 'Others', '5th Pass',
      'Illiterate', 'Literate'], dtype=object)
```

Plot the data as a Bar Chart.

```
plt.figure(figsize=(15,7))
plt.xticks(rotation = 60);
# plot detailing

plt.xlabel("Education Status", fontsize = 15)
plt.ylabel("No. of Candidates", fontsize = 15)
plt.title("EDUCATIONAL QUALIFICATIONS OF WINNERS", fontsize = 18)
# labels and title

sns.barplot(x = education.index, y = education.values);
# plotting the barplot
```



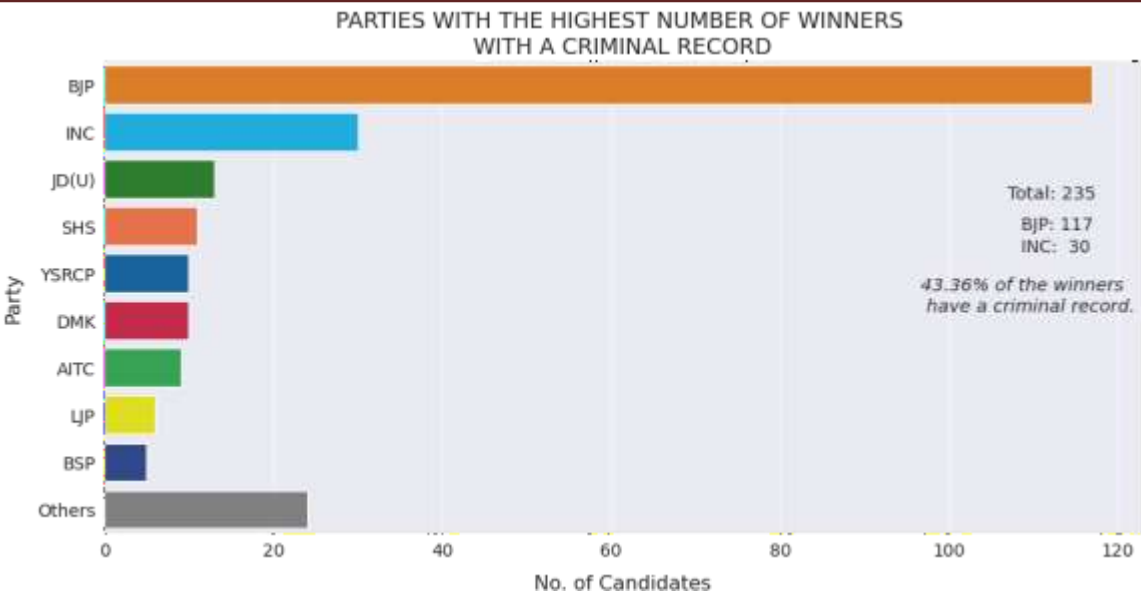
Plot the data as a horizontal bar chart, with the appropriate color grading.

```
plt.figure(figsize = (15,7))

palette = ['#f97d09', '#00bdfe', '#228b22', '#ff6634', '#0266b4', '#dc143c', '#24b44c',
           'yellow', '#22409a', 'grey']
# color palette is customized to correspond to each party's colors
sns.barplot(x = crime.values, y = crime.index, palette = sns.set_palette(palette, 10))
# the data is plotted as a horizontal bar plot

plt.title("PARTIES WITH THE HIGHEST NUMBER OF WINNERS \nWITH A CRIMINAL RECORD", fontsize = 18)
plt.ylabel("Party", fontsize = 16, labelpad = 12)
plt.xlabel("No. of Candidates", fontsize = 16, labelpad = 12)
plt.tick_params(labelsize = 14) # increase label size
# plot detailing

plt.figtext(0.8, 0.66, "Total: " + str(crime.sum()), fontsize = 14.5)
plt.figtext(0.81, 0.61, "BJP: " + str(crime["BJP"]), fontsize = 14)
plt.figtext(0.81, 0.575, "INC: " + str(crime["INC"]), fontsize = 14)
plt.figtext(0.735, 0.48, str(round(crime.sum()/winners_df.shape[0]*100,2)) + "% of the winners\n have a criminal
record.",
           fontsize = 14.5, style = 'oblique');
# printing details on the graph
```

5.1 Data Insights

CHAPTER 5

5.1 Data Insights

RESULTS

- ❖ Voter Turnout Trends: Analysis reveals a general increase in voter turnout over the years, with significant spikes during particular elections.
- ❖ Demographic Influence: Age, gender, and urban/rural divide have noticeable effects on voting patterns and outcomes.
- ❖ Party Performance: Historical data shows fluctuating fortunes for major parties, with regional parties gaining ground in certain states.

5.2 Predictive Model Performance

The predictive models demonstrated reasonable accuracy in forecasting election outcomes based on historical data. Key performance metrics include:

- Accuracy
- Precision
- Recall

5.3 Visualizations

Graphs and charts illustrating voter turnout, party performance over years, and demographic influences on voting patterns.

CHAPTER 6

CONCLUSION

6.1 Summary

The analysis of Indian election data provided valuable insights into voter behavior, demographic influences, and party performance. The predictive models, though preliminary, showed promise in forecasting election results.

6.1 Future Work

Future work could involve:

- Incorporating more granular data at the constituency level.
- Enhancing predictive models with more sophisticated algorithms.
- Exploring the impact of social media and other external factors on election outcomes.

6.2 Final Thoughts

This project underscores the importance of data analysis in understanding and predicting electoral trends. By leveraging modern data science techniques, stakeholders can gain deeper insights and make more informed decisions.

CHAPTER 7

REFERENCES

1. <https://www.geeksforgeeks.org/data-analyst-projects>.
2. 2019 Indian General Elections Data Analysis , sh007online, Jovian.
About
3. realtime sentiment analysis of india elections 2019, <https://github.com/v-adhithyan/india-elections>
4. Results related to General Assembly (Lok Sabha) elections 2019 in India, <https://github.com/pratapvardhan/Elections-India-2019>.
5. Election System in India, <https://www.geeksforgeeks.org>