Loksabha Election 2019 Data Analysis in India

Importing the Libraries:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.feature_selection import SelectKBest, chi2
import pandas.util.testing as tm
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestClassifier
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:12: FutureWarning: panda
  if sys.path[0] == '':
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files LS_2.0.csv

- **LS_2.0.csv**(application/vnd.ms-excel) - 393712 bytes, last modified: 1/21/2022 - 100% done

```
Saving LS_2.0.csv to LS_2.0 (8).csv
```

```
import io
df2 = pd.read_csv(io.BytesIO(uploaded['LS_2.0.csv']))
```

```
df2.head()
```

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL\nCASES
0	Telangana	ADILABAD	SOYAM BAPU RAO	1	BJP	Lotus	MALE	52
1	Telangana	ADILABAD	Godam Nagesh	0	TRS	Car	MALE	0

Loading the Files

Pradesh Baghel

Displaying the Data

```
df2.head()
```

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL\nCASES
0	Telangana	ADILABAD	SOYAM BAPU RAO	1	BJP	Lotus	MALE	52
1	Telangana	ADILABAD	Godam Nagesh	0	TRS	Car	MALE	0
2	Telangana	ADILABAD	RATHOD RAMESH	0	INC	Hand	MALE	3
3	Telangana	ADILABAD	NOTA	0	NOTA	NaN	NaN	NaN
4	Uttar Pradesh	AGRA	Satyapal Singh Baghel	1	BJP	Lotus	MALE	5

```
# rename invalid column names
df2 = df2.rename(columns={'CRIMINAL\nCASES': 'CRIMINAL_CASES', 'GENERAL\nVOTES': 'GENERAL_
df2.head()
```

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL_CASES
0	Telangana	ADILABAD	SOYAM BAPU RAO	1	BJP	Lotus	MALE	52
1	Telangana	ADILABAD	Godam Nagesh	0	TRS	Car	MALE	0
2	Telangana	ADILABAD	RATHOD RAMESH	0	INC	Hand	MALE	3
3	Telangana	ADILABAD	NOTA	0	NOTA	NaN	NaN	NaN
4	Uttar Pradesh	AGRA	Satyapal Singh Baghel	1	BJP	Lotus	MALE	5

Shape of the Dataset

```
df2.shape
```

```
(2263, 19)
```

Information about all the collumns in the Dataset

```
df2.info()
```

```
<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
```

Description of Dataset

```
df2.describe()
```

	WINNER	AGE	GENERAL_VOTES	POSTAL_VOTES	TOTAL_VOTES	OVER_TOTA
count	2263.000000	2018.000000	2.263000e+03	2263.000000	2.263000e+03	
mean	0.238179	52.273538	2.615991e+05	990.710561	2.625898e+05	
std	0.426064	11.869373	2.549906e+05	1602.839174	2.559822e+05	
min	0.000000	25.000000	1.339000e+03	0.000000	1.342000e+03	
25%	0.000000	43.250000	2.103450e+04	57.000000	2.116250e+04	
50%	0.000000	52.000000	1.539340e+05	316.000000	1.544890e+05	
75%	0.000000	61.000000	4.858040e+05	1385.000000	4.872315e+05	
max	1.000000	86.000000	1.066824e+06	19367.000000	1.068569e+06	

Checking the Null Value in the Dataset

```
df2.isnull().values.any()
df2.isna().sum()
```

```
STATE                                       0
CONSTITUENCY                                0
NAME                                        0
WINNER                                      0
PARTY                                       0
SYMBOL                                    245
GENDER                                    245
CRIMINAL_CASES                            245
AGE                                       245
CATEGORY                                  245
EDUCATION                                 245
ASSETS                                    245
LIABILITIES                               245
GENERAL_VOTES                               0
POSTAL_VOTES                                0
TOTAL_VOTES                                 0
OVER_TOTAL_ELECTORS_IN_CONSTITUENCY         0
```

```
OVER_TOTAL_VOTES_POLLED_IN_CONSTITUENCY      0
TOTAL_ELECTORS                               0
dtype: int64
```

Droping the columns which we found not relevant for our prediction model and have 'NA' values

```
df2
```

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL_CA
0	Telangana	ADILABAD	SOYAM BAPU RAO	1	BJP	Lotus	MALE	
1	Telangana	ADILABAD	Godam Nagesh	0	TRS	Car	MALE	
2	Telangana	ADILABAD	RATHOD RAMESH	0	INC	Hand	MALE	
3	Telangana	ADILABAD	NOTA	0	NOTA	NaN	NaN	I
4	Uttar Pradesh	AGRA	Satyapal Singh Baghel	1	BJP	Lotus	MALE	
...	...	...	...	...	...	...	...	
2258	Maharashtra	YAVATMAL-WASHIM	Anil Jayram Rathod	0	IND	SHIP	MALE	
2259	Telangana	ZAHIRABAD	B.B.PATIL	1	TRS	Car	MALE	
2260	Telangana	ZAHIRABAD	MADAN MOHAN RAO	0	INC	Hand	MALE	
2261	Telangana	ZAHIRABAD	BANALA LAXMA REDDY	0	BJP	Lotus	MALE	
2262	Telangana	ZAHIRABAD	NOTA	0	NOTA	NaN	NaN	I

2263 rows × 19 columns

Imputing Age

```
df2['AGE'].fillna(df2['AGE'].median(),inplace=True)
```

```
df2
```

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL_CA
0	Telangana	ADILABAD	SOYAM BAPU RAO	1	BJP	Lotus	MALE	
1	Telangana	ADILABAD	Godam Nagesh	0	TRS	Car	MALE	
2	Telangana	ADILABAD	RATHOD RAMESH	0	INC	Hand	MALE	
3	Telangana	ADILABAD	NOTA	0	NOTA	NaN	NaN	I
4	Uttar Pradesh	AGRA	Satyapal Singh Baghel	1	BJP	Lotus	MALE	
...	...	...	...	...	...	...	...	
2258	Maharashtra	YAVATMAL-WASHIM	Anil Jayram Rathod	0	IND	SHIP	MALE	
2259	Telangana	ZAHIRABAD	B.B.PATIL	1	TRS	Car	MALE	
2260	Telangana	ZAHIRABAD	MADAN MOHAN RAO	0	INC	Hand	MALE	
2261	Telangana	ZAHIRABAD	BANALA LAXMA REDDY	0	BJP	Lotus	MALE	
2262	Telangana	ZAHIRABAD	NOTA	0	NOTA	NaN	NaN	I

2263 rows × 19 columns

ENCODING EDUCATION COLUMN for numeric values

```
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.feature_selection import SelectKBest, chi2


#encode education column
encoded_edu = []
# iterate through each row in the dataset
for row in df2.itertuples():
    education = row.EDUCATION
    if  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)
df2['EDUCATION'] = encoded_edu
df2
```

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL_CA
0	Telangana	ADILABAD	SOYAM BAPU RAO	1	BJP	Lotus	MALE	
1	Telangana	ADILABAD	Godam Nagesh	0	TRS	Car	MALE	
2	Telangana	ADILABAD	RATHOD RAMESH	0	INC	Hand	MALE	
3	Telangana	ADILABAD	NOTA	0	NOTA	NaN	NaN	I
4	Uttar Pradesh	AGRA	Satyapal Singh Baghel	1	BJP	Lotus	MALE	
...	...	...	...	...	...	...	...	
2258	Maharashtra	YAVATMAL-WASHIM	Anil Jayram Rathod	0	IND	SHIP	MALE	
2259	Telangana	ZAHIRABAD	B.B.PATIL	1	TRS	Car	MALE	

DATA PROCESSING

data display

REDDY

```
df2.drop(["ASSETS", "LIABILITIES"], axis = 1, inplace = True)
df2
```

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL_CA
0	Telangana	ADILABAD	SOYAM BAPU RAO	1	BJP	Lotus	MALE	
1	Telangana	ADILABAD	Godam Nagesh	0	TRS	Car	MALE	
2	Telangana	ADILABAD	RATHOD RAMESH	0	INC	Hand	MALE	
3	Telangana	ADILABAD	NOTA	0	NOTA	NaN	NaN	I
4	Uttar Pradesh	AGRA	Satyapal Singh Baghel	1	BJP	Lotus	MALE	
...	...	...	...	...	...	...	...	

```
df2['PARTY'].value_counts()

# 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 frequent
df2.loc[~df2["PARTY"].isin(['BJP','INC','IND','BSP', 'CPI(M)', 'AITC', 'MNM', 'TDP', 'VSRC
df2['PARTY'].value_counts()
```

```
Other     775
BJP       420
INC       413
IND       201
BSP       163
CPI(M)    100
AITC       47
SP         39
MNM        36
TDP        25
DMK        23
BJD        21
Name: PARTY, dtype: int64
```

Lable Encoding for all non-numeric Coloumns

```
# label encode categorical columns

lblEncoder_state = LabelEncoder()
lblEncoder_state.fit(df2['STATE'])
df2['STATE'] = lblEncoder_state.transform(df2['STATE'])

lblEncoder_cons = LabelEncoder()
lblEncoder_cons.fit(df2['CONSTITUENCY'])
df2['CONSTITUENCY'] = lblEncoder_cons.transform(df2['CONSTITUENCY'])
```

```
lblEncoder_name = LabelEncoder()
lblEncoder_name.fit(df2['NAME'])
df2['NAME'] = lblEncoder_name.transform(df2['NAME'])

lblEncoder_party = LabelEncoder()
lblEncoder_party.fit(df2['PARTY'])
df2['PARTY'] = lblEncoder_party.transform(df2['PARTY'])

lblEncoder_symbol = LabelEncoder()
lblEncoder_symbol.fit(df2['SYMBOL'])
df2['SYMBOL'] = lblEncoder_symbol.transform(df2['SYMBOL'])

lblEncoder_gender = LabelEncoder()
lblEncoder_gender.fit(df2['GENDER'])
df2['GENDER'] = lblEncoder_gender.transform(df2['GENDER'])

lblEncoder_category = LabelEncoder()
lblEncoder_category.fit(df2['CATEGORY'])
df2['CATEGORY'] = lblEncoder_category.transform(df2['CATEGORY'])
```

```
df2['CRIMINAL_CASES'] = df2['CRIMINAL_CASES'].replace(['Not Available'], '0')
df2['CRIMINAL_CASES'] = df2['CRIMINAL_CASES'].astype(object).astype(float)
df2['CRIMINAL_CASES'].fillna(df2['CRIMINAL_CASES'].median(),inplace=True)
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2263 entries, 0 to 2262
Data columns (total 17 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   STATE                                   2263 non-null   int64
 1   CONSTITUENCY                            2263 non-null   int64
 2   NAME                                    2263 non-null   int64
 3   WINNER                                  2263 non-null   int64
 4   PARTY                                   2263 non-null   int64
 5   SYMBOL                                  2263 non-null   int64
 6   GENDER                                  2263 non-null   int64
 7   CRIMINAL_CASES                          2263 non-null   float64
 8   AGE                                     2263 non-null   float64
 9   CATEGORY                                2263 non-null   int64
 10  EDUCATION                               2263 non-null   int64
 11  GENERAL_VOTES                           2263 non-null   int64
 12  POSTAL_VOTES                            2263 non-null   int64
 13  TOTAL_VOTES                             2263 non-null   int64
 14  OVER_TOTAL_ELECTORS_IN_CONSTITUENCY     2263 non-null   float64
 15  OVER_TOTAL_VOTES_POLLED_IN_CONSTITUENCY 2263 non-null   float64
 16  TOTAL_ELECTORS                          2263 non-null   int64
dtypes: float64(4), int64(13)
memory usage: 300.7 KB
```

```
from statsmodels.stats.outliers_influence import variance_inflation_factor

def calc_vif(X):

    # Calculating VIF
```

```
    vif = pd.DataFrame()
    vif["variables"] = X.columns
    vif["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

    return(vif)
```

```
X = df2.iloc[:,:-1]
calc_vif(X)
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/stats/outliers_influence.py:185: F
  vif = 1. / (1. - r_squared_i)
```

	variables	VIF
0	STATE	4.190632
1	CONSTITUENCY	3.828570
2	NAME	4.264500
3	WINNER	3.265551
4	PARTY	6.525269
5	SYMBOL	8.230275
6	GENDER	9.683688
7	CRIMINAL_CASES	1.047166
8	AGE	17.898292
9	CATEGORY	2.664178
10	EDUCATION	9.926901
11	GENERAL_VOTES	inf
12	POSTAL_VOTES	inf
13	TOTAL_VOTES	inf
14	OVER_TOTAL_ELECTORS_IN_CONSTITUENCY	56.777492
15	OVER_TOTAL_VOTES_POLLED_IN_CONSTITUENCY	62.605639

```
df2.drop(["GENERAL_VOTES", "POSTAL_VOTES", "TOTAL_VOTES", "OVER_TOTAL_ELECTORS_IN_CONSTITU
df2
```

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL_CASES	AGE
0	31	0	1713	1	2	80	1	52.0	52.0
1	31	0	700	0	9	32	1	0.0	54.0
2	31	0	1498	0	6	66	1	3.0	52.0
3	31	0	1203	0	9	126	2	0.0	52.0
4	33	1	1789	1	2	80	1	5.0	58.0
...	...	...	...	...	...	...	...	...	...

```
scaler = MinMaxScaler(feature_range=(0, 1))
features = [
    'STATE', 'CONSTITUENCY', 'NAME', 'PARTY', 'SYMBOL', 'GENDER', 'CRIMINAL_CASES', 'AGE',
df2[features] = scaler.fit_transform(df2[features])
```

2261	31	538	249	0	2	80	1	3.0	47.0

```
# separate train features and label
y = df2["WINNER"]
X = df2.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, s
# 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:  75.27593818984548 %
```

✓ 0s completed at 4:08 PM