rohit-kumar-jupyter-notebbok

October 7, 2024

1 To Predict the chances of winning of a candidate

Importing Important Libraries

[144]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import r2_score, classification_report as cr ,␣
↪confusion_matrix as cm
import warnings
warnings.filterwarnings("ignore")
import matplotlib.pyplot as plt
import numpy as np
```

Importing Election dataset csv file

[146]:
```
z = pd.read_csv(r"C:\Users\skj_h\OneDrive\Desktop\dataset\Election dataset.csv")
```

[147]:
```
z
```

[147]:
```
               STATE     CONSTITUENCY                    NAME  WINNER PARTY  \
0          Telangana         ADILABAD          SOYAM BAPU RAO       1   BJP
1          Telangana         ADILABAD           Godam Nagesh       0   TRS
2          Telangana         ADILABAD           RATHOD RAMESH       0   INC
3          Telangana         ADILABAD                    NOTA       0  NOTA
4      Uttar Pradesh             AGRA  Satyapal Singh Baghel       1   BJP
…                …                …                      …      …     …
2258     Maharashtra  YAVATMAL-WASHIM     Anil Jayram Rathod       0   IND
2259       Telangana        ZAHIRABAD              B.B.PATIL       1   TRS
2260       Telangana        ZAHIRABAD        MADAN MOHAN RAO       0   INC
2261       Telangana        ZAHIRABAD     BANALA LAXMA REDDY       0   BJP
2262       Telangana        ZAHIRABAD                    NOTA       0  NOTA

     SYMBOL GENDER CRIMINAL\nCASES   AGE CATEGORY      EDUCATION  \
0     Lotus   MALE              52  52.0       ST      12th Pass
```

1	Car	MALE	0	54.0	ST	Post Graduate
2	Hand	MALE	3	52.0	ST	12th Pass
3	NaN	NaN	NaN	NaN	NaN	NaN
4	Lotus	MALE	5	58.0	SC	Doctorate
…	…	…	…	…	…	…
2258	SHIP	MALE	0	43.0	GENERAL	Post Graduate
2259	Car	MALE	18	63.0	GENERAL	Graduate
2260	Hand	MALE	0	49.0	GENERAL	Post Graduate
2261	Lotus	MALE	3	47.0	GENERAL	12th Pass
2262	NaN	NaN	NaN	NaN	NaN	NaN

	ASSETS	LIABILITIES \
0	Rs 30,99,414\n ~ 30 Lacs+	Rs 2,31,450\n ~ 2 Lacs+
1	Rs 1,84,77,888\n ~ 1 Crore+	Rs 8,47,000\n ~ 8 Lacs+
2	Rs 3,64,91,000\n ~ 3 Crore+	Rs 1,53,00,000\n ~ 1 Crore+
3	NaN	NaN
4	Rs 7,42,74,036\n ~ 7 Crore+	Rs 86,06,522\n ~ 86 Lacs+
…	…	…
2258	Rs 48,90,000\n ~ 48 Lacs+	Rs 10,20,000\n ~ 10 Lacs+
2259	Rs 1,28,78,51,556\n ~ 128 Crore+	Rs 1,15,35,000\n ~ 1 Crore+
2260	Rs 90,36,63,001\n ~ 90 Crore+	Rs 0\n ~
2261	Rs 5,85,77,327\n ~ 5 Crore+	Rs 52,50,000\n ~ 52 Lacs+
2262	NaN	NaN

	GENERAL\nVOTES	POSTAL\nVOTES	TOTAL\nVOTES \
0	376892	482	377374
1	318665	149	318814
2	314057	181	314238
3	13030	6	13036
4	644459	2416	646875
…	…	…	…
2258	14661	25	14686
2259	434066	178	434244
2260	427900	115	428015
2261	138731	216	138947
2262	11138	2	11140

	OVER TOTAL ELECTORS \nIN CONSTITUENCY \
0	25.330684
1	21.399929
2	21.092771
3	0.875023
4	33.383823
…	…
2258	0.766419
2259	28.975369
2260	28.559732

```
2261                                  9.271379
2262                                  0.743328

      OVER TOTAL VOTES POLLED \nIN CONSTITUENCY  TOTAL ELECTORS
0                                     35.468248         1489790
1                                     29.964370         1489790
2                                     29.534285         1489790
3                                      1.225214         1489790
4                                     56.464615         1937690
…                                           …               …
2258                                   1.250060         1916185
2259                                  41.574183         1498666
2260                                  40.977823         1498666
2261                                  13.302678         1498666
2262                                   1.066535         1498666

[2263 rows x 19 columns]
```

[148]:
```
z.shape
```

[148]: (2263, 19)

[149]:
```
z.size
```

[149]: 42997

[150]:
```
z.ndim
```

[150]: 2

[151]:
```
z.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
```

[152]:

```
z.describe()
```

[152]:

```
            WINNER          AGE  GENERAL\nVOTES  POSTAL\nVOTES  TOTAL\nVOTES  \
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

       OVER TOTAL ELECTORS \nIN CONSTITUENCY  \
count                            2263.000000
mean                               15.811412
std                                14.962861
min                                 0.097941
25%                                 1.296518
50%                                10.510553
75%                                29.468185
max                                51.951012

       OVER TOTAL VOTES POLLED \nIN CONSTITUENCY  TOTAL ELECTORS
count                                2263.000000    2.263000e+03
mean                                   23.190525    1.658016e+06
std                                    21.564758    3.145187e+05
min                                     1.000039    5.518900e+04
25%                                     1.899502    1.530014e+06
50%                                    16.221721    1.679030e+06
```

```
75%                                     42.590233    1.816857e+06
max                                     74.411856    3.150313e+06
```

[153]:
```
z.dtypes
```

[153]:
```
STATE                                       object
CONSTITUENCY                                object
NAME                                        object
WINNER                                       int64
PARTY                                       object
SYMBOL                                      object
GENDER                                      object
CRIMINAL\nCASES                             object
AGE                                        float64
CATEGORY                                    object
EDUCATION                                   object
ASSETS                                      object
LIABILITIES                                 object
GENERAL\nVOTES                               int64
POSTAL\nVOTES                                int64
TOTAL\nVOTES                                 int64
OVER TOTAL ELECTORS \nIN CONSTITUENCY      float64
OVER TOTAL VOTES POLLED \nIN CONSTITUENCY    float64
TOTAL ELECTORS                               int64
dtype: object
```

Counting number of null values, NAN values and removing it

[155]:
```
z.isnull().sum()
```

[155]:
```
STATE                                        0
CONSTITUENCY                                 0
NAME                                         0
WINNER                                       0
PARTY                                        0
SYMBOL                                     245
GENDER                                     245
CRIMINAL\nCASES                            245
AGE                                        245
CATEGORY                                   245
EDUCATION                                  245
ASSETS                                     245
LIABILITIES                                245
GENERAL\nVOTES                               0
POSTAL\nVOTES                                0
TOTAL\nVOTES                                 0
OVER TOTAL ELECTORS \nIN CONSTITUENCY        0
```

```
OVER TOTAL VOTES POLLED \nIN CONSTITUENCY      0
TOTAL ELECTORS                                 0
dtype: int64
```

[ ]:

```
[156]: z = z[z["SYMBOL"].notna()]
```

```
[157]: z.isnull().sum().sum()
```

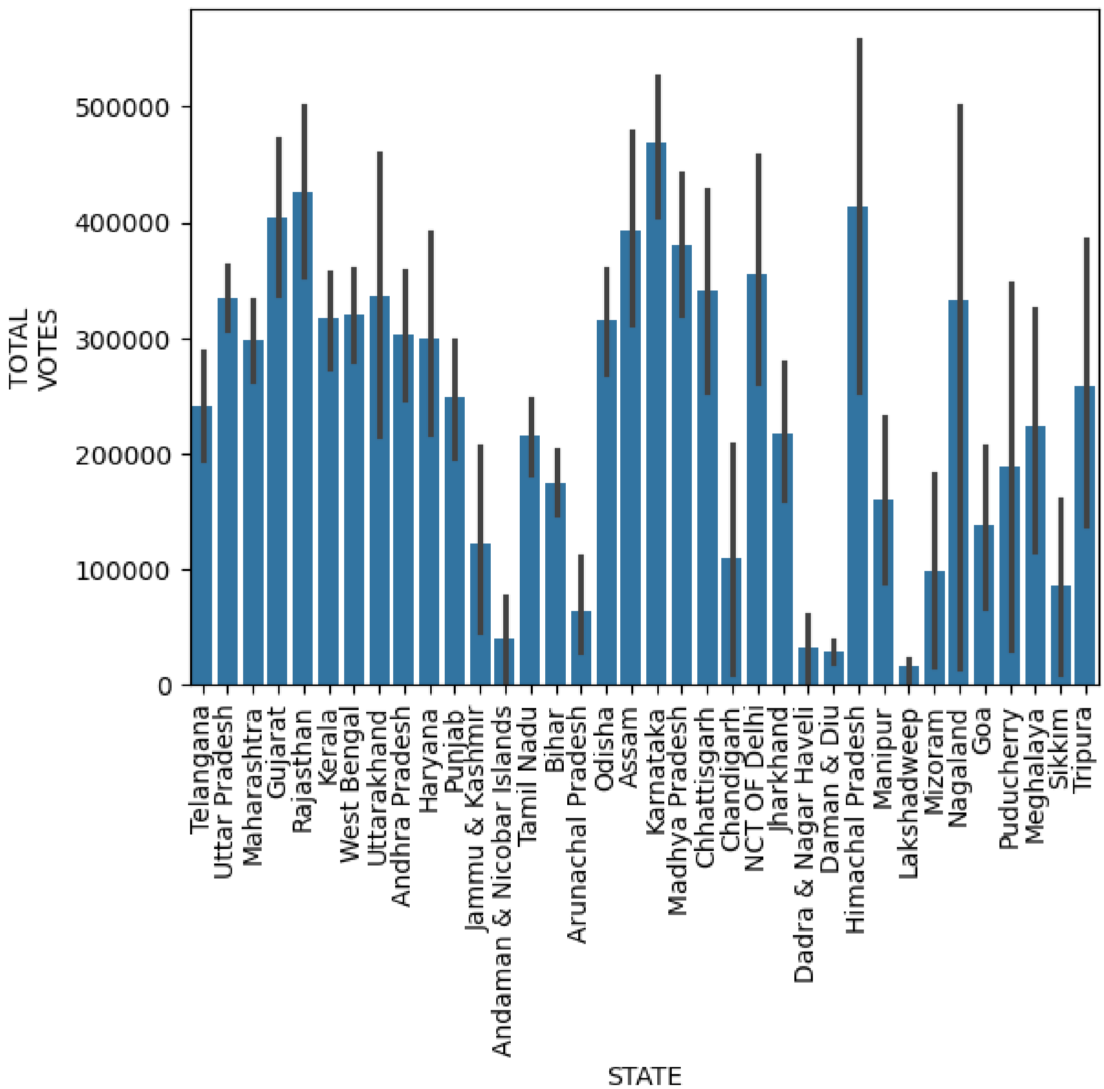
```
[157]: 0
```

Data visualization

```
[159]: sns.barplot(x = z["STATE"], y = z["TOTAL\nVOTES"], data = z)
       plt.xticks(rotation = 90)
```

```
[159]: ([0,
  1,
  2,
  3,
  4,
  5,
  6,
  7,
  8,
  9,
  10,
  11,
  12,
  13,
  14,
  15,
  16,
  17,
  18,
  19,
  20,
  21,
  22,
  23,
  24,
  25,
  26,
  27,
  28,
  29,
```

```
 30,
 31,
 32,
 33,
 34,
 35],
[Text(0, 0, 'Telangana'),
 Text(1, 0, 'Uttar Pradesh'),
 Text(2, 0, 'Maharashtra'),
 Text(3, 0, 'Gujarat'),
 Text(4, 0, 'Rajasthan'),
 Text(5, 0, 'Kerala'),
 Text(6, 0, 'West Bengal'),
 Text(7, 0, 'Uttarakhand'),
 Text(8, 0, 'Andhra Pradesh'),
 Text(9, 0, 'Haryana'),
 Text(10, 0, 'Punjab'),
 Text(11, 0, 'Jammu & Kashmir'),
 Text(12, 0, 'Andaman & Nicobar Islands'),
 Text(13, 0, 'Tamil Nadu'),
 Text(14, 0, 'Bihar'),
 Text(15, 0, 'Arunachal Pradesh'),
 Text(16, 0, 'Odisha'),
 Text(17, 0, 'Assam'),
 Text(18, 0, 'Karnataka'),
 Text(19, 0, 'Madhya Pradesh'),
 Text(20, 0, 'Chhattisgarh'),
 Text(21, 0, 'Chandigarh'),
 Text(22, 0, 'NCT OF Delhi'),
 Text(23, 0, 'Jharkhand'),
 Text(24, 0, 'Dadra & Nagar Haveli'),
 Text(25, 0, 'Daman & Diu'),
 Text(26, 0, 'Himachal Pradesh'),
 Text(27, 0, 'Manipur'),
 Text(28, 0, 'Lakshadweep'),
 Text(29, 0, 'Mizoram'),
 Text(30, 0, 'Nagaland'),
 Text(31, 0, 'Goa'),
 Text(32, 0, 'Puducherry'),
 Text(33, 0, 'Meghalaya'),
 Text(34, 0, 'Sikkim'),
 Text(35, 0, 'Tripura')])
```



[160]:
```
z["CATEGORY"].value_counts()
```

[160]:
```
CATEGORY
GENERAL    1392
SC          383
ST          243
Name: count, dtype: int64
```

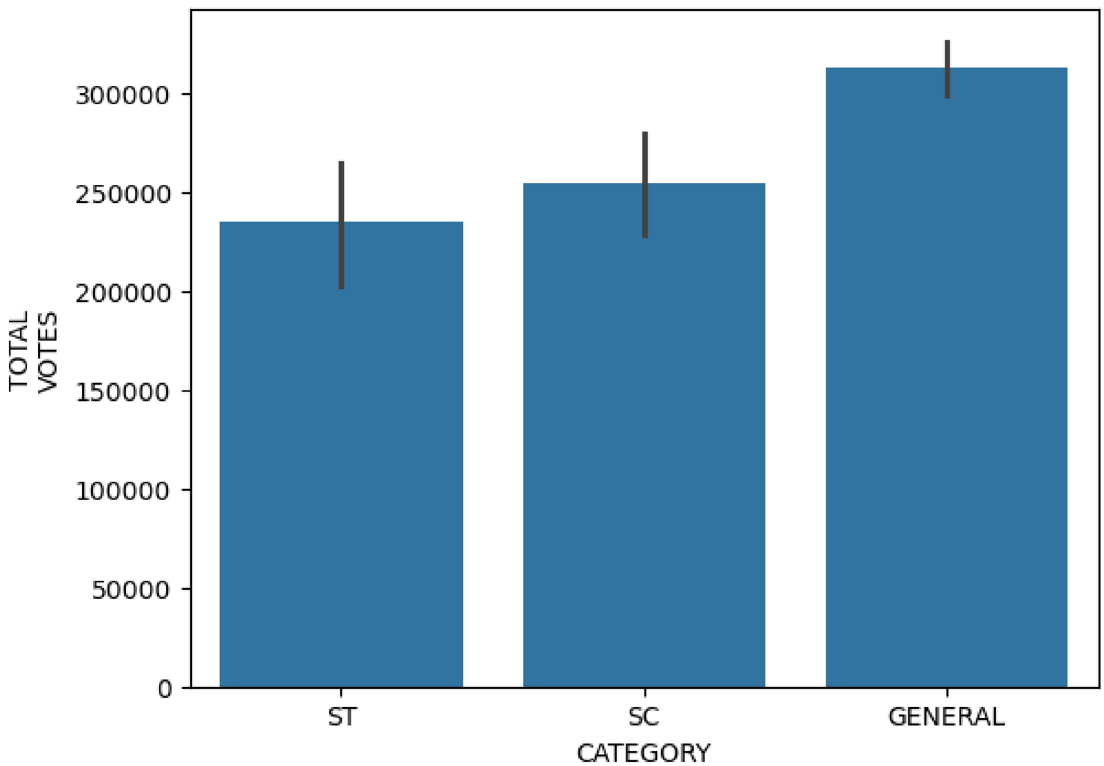
[161]:
```
z.columns
```

[161]:
```
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')
```

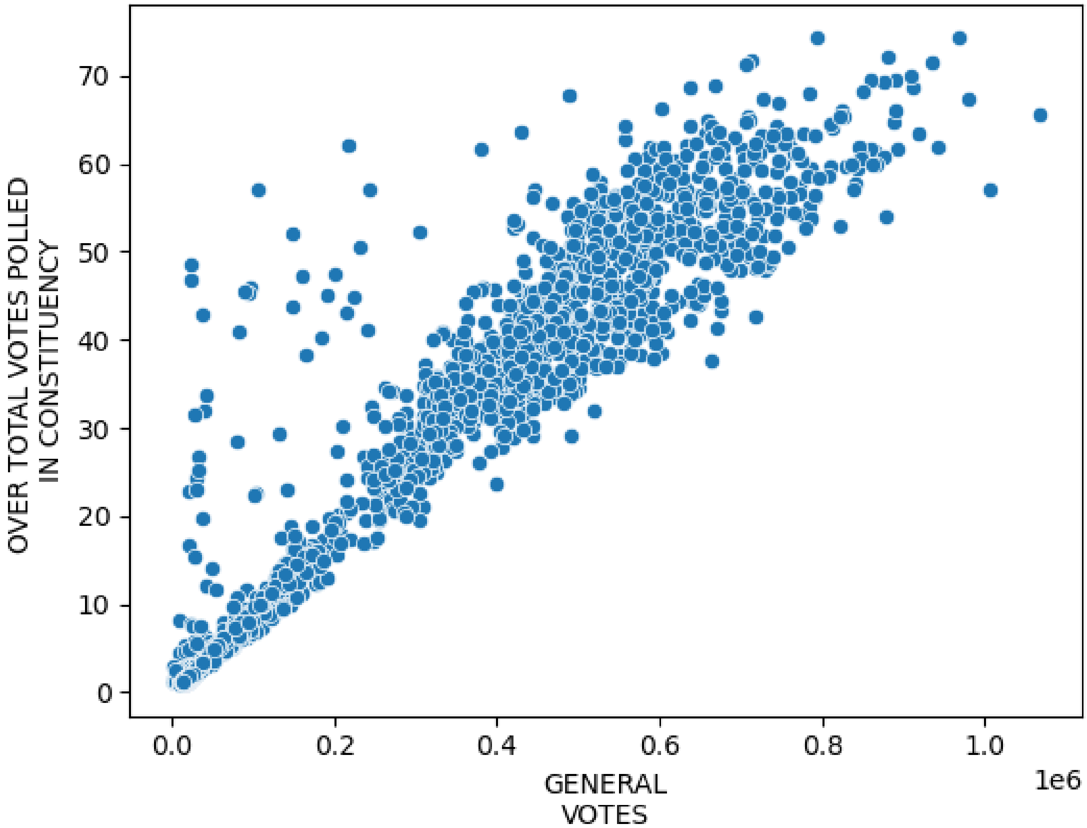
[162]:
```
sns.barplot(x = z["CATEGORY"], y = z["TOTAL\nVOTES"], data = z)
```

[162]:
```
<Axes: xlabel='CATEGORY', ylabel='TOTAL\nVOTES'>
```



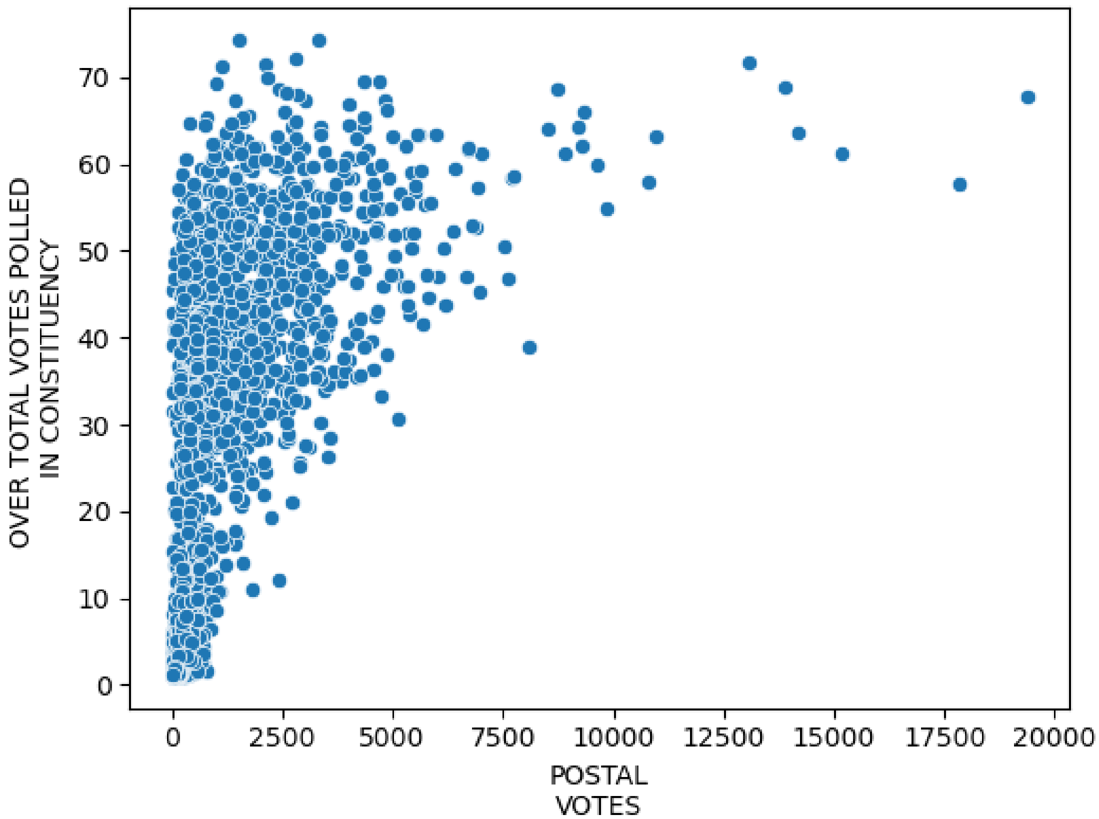
[163]:
```
sns.scatterplot(x = z["GENERAL\nVOTES"], y= z["OVER TOTAL VOTES POLLED \nIN␣
↪CONSTITUENCY"], data = z)
```

[163]:
```
<Axes: xlabel='GENERAL\nVOTES', ylabel='OVER TOTAL VOTES POLLED \nIN
CONSTITUENCY'>
```



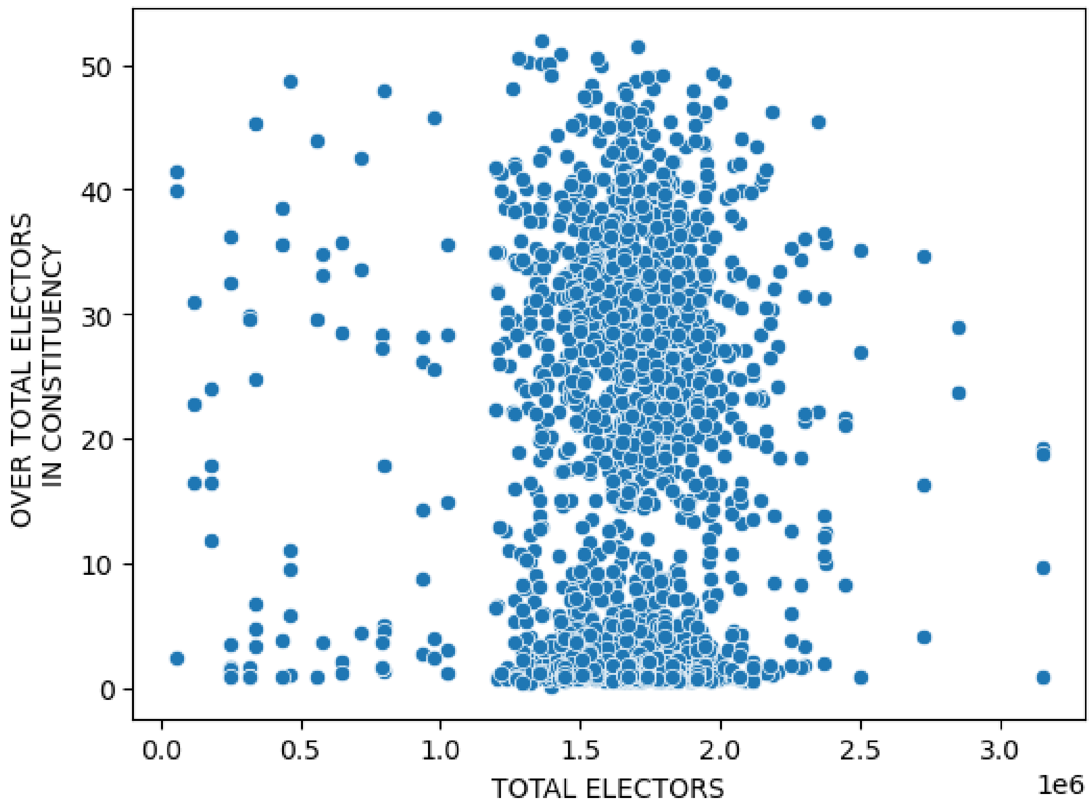
[164]:
```
sns.scatterplot(x = z["POSTAL\nVOTES"], y= z["OVER TOTAL VOTES POLLED \nIN␣
↪CONSTITUENCY"], data = z)
```

[164]: <Axes: xlabel='POSTAL\nVOTES', ylabel='OVER TOTAL VOTES POLLED \nIN
CONSTITUENCY'>



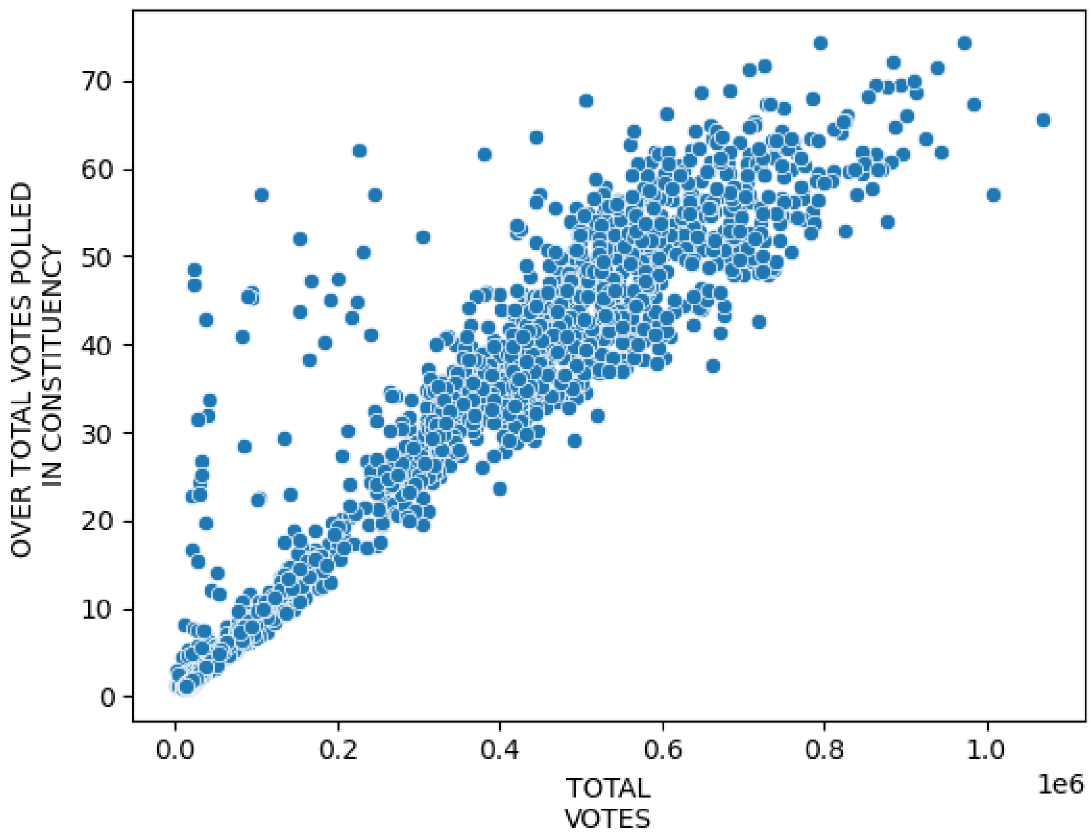
[165]:
```
sns.scatterplot(x = z["TOTAL ELECTORS"], y= z["OVER TOTAL ELECTORS \nIN␣
↪CONSTITUENCY"], data = z)
```

[165]: `<Axes: xlabel='TOTAL ELECTORS', ylabel='OVER TOTAL ELECTORS \nIN CONSTITUENCY'>`



[166]:
```
sns.scatterplot(x = z["TOTAL\nVOTES"], y= z["OVER TOTAL VOTES POLLED \nIN␣
↪CONSTITUENCY"], data = z)
```

[166]:
```
<Axes: xlabel='TOTAL\nVOTES', ylabel='OVER TOTAL VOTES POLLED \nIN
CONSTITUENCY'>
```

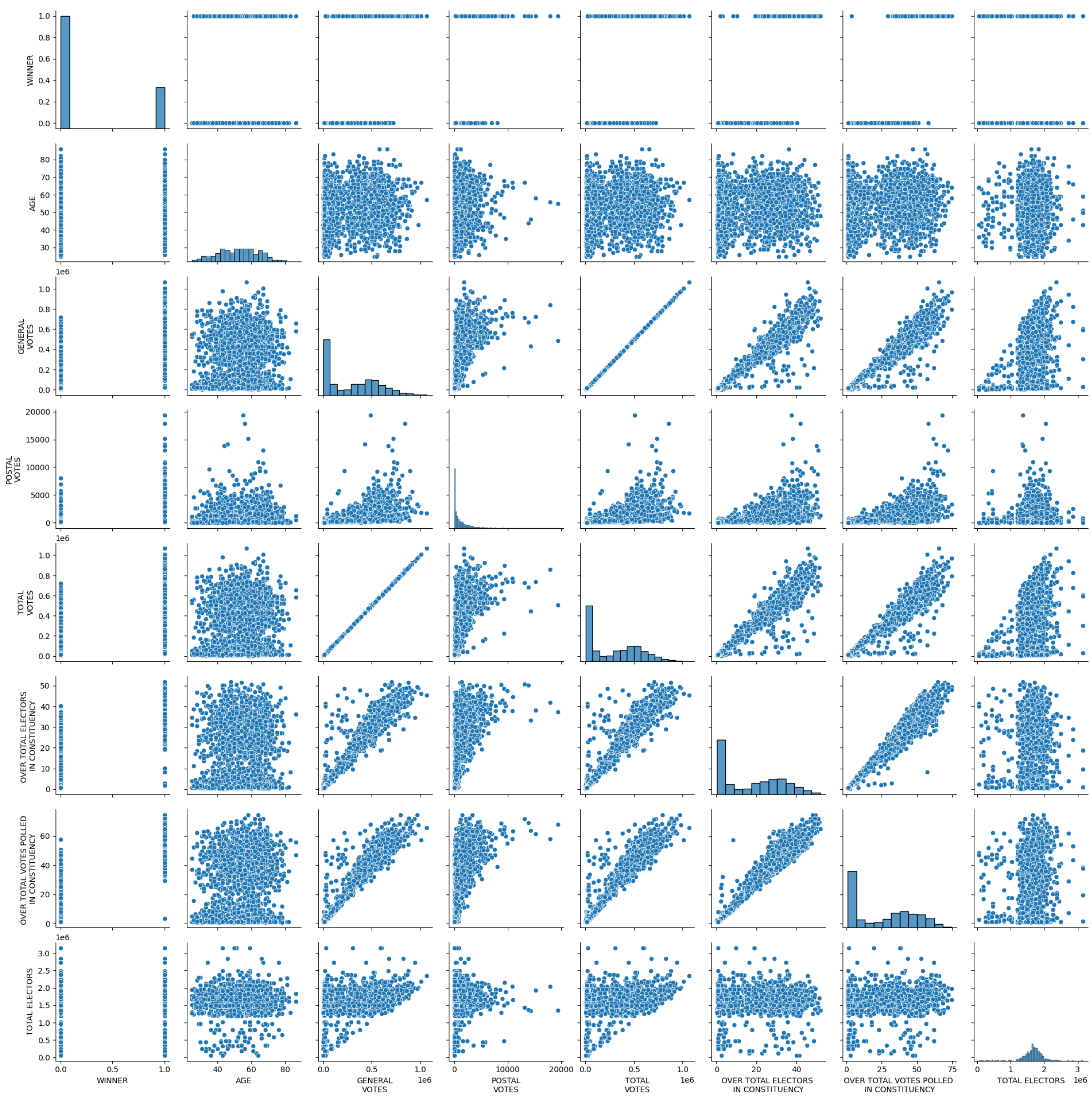


[167]:
```python
z.columns
```

[167]:
```
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')
```

[168]:
```python
sns.pairplot(z)
```

[168]: <seaborn.axisgrid.PairGrid at 0x1e506b0aa20>

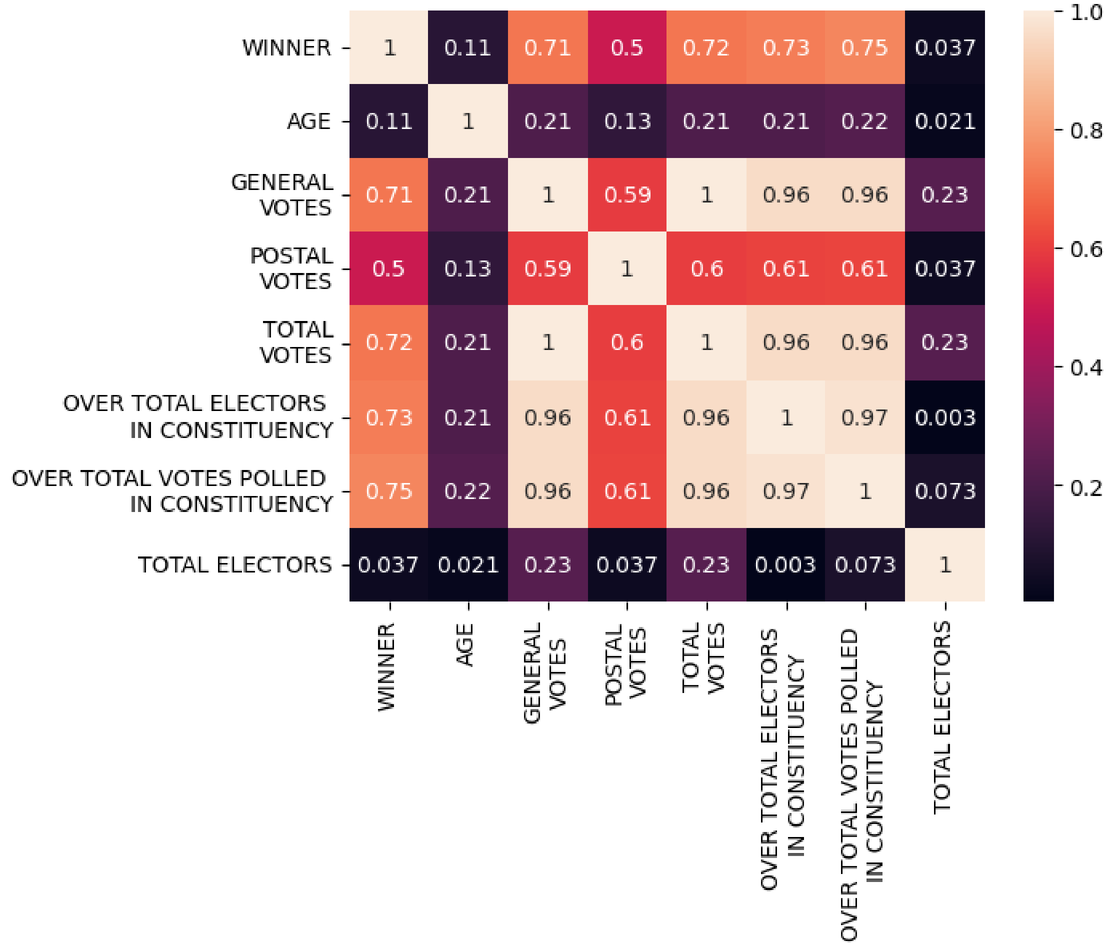


Modelling

[170]:
```
b = z.copy()
for i in b:
    if(b[i].dtype == "object"):
        b.drop([i], axis = 1, inplace = True)
```

[171]:
```
sns.heatmap(b.corr(), annot = True)
```

[171]: <Axes: >



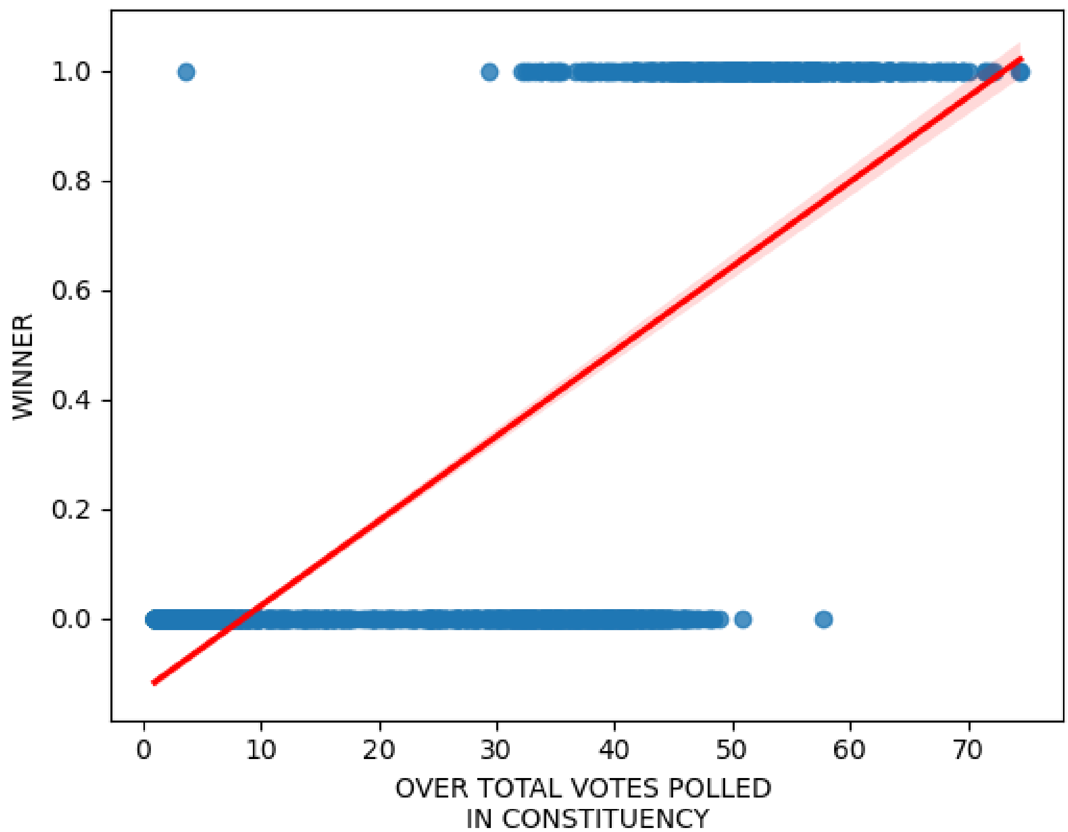
[172]:
```
z.columns
```

[172]:
```
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')
```

Regression Analysis

[174]:
```
sns.regplot(x = z["OVER TOTAL VOTES POLLED \nIN CONSTITUENCY"], y =␣
↪z["WINNER"], data = z, line_kws = {"color" : "red"})
```

[174]:
```
<Axes: xlabel='OVER TOTAL VOTES POLLED \nIN CONSTITUENCY', ylabel='WINNER'>
```




```
[175]: x = z[["OVER TOTAL VOTES POLLED \nIN CONSTITUENCY", "WINNER"]]
```

```
[176]: X = x
       Y = x["WINNER"]
```

```
[177]: x_train, x_test, y_train, y_test = train_test_split(X,Y, train_size = 0.7,␣
        ↪test_size = 0.3)
```

```
[178]: x_train = x_train.drop(["WINNER"], axis = 1)
```

```
[179]: y_train = np.array(y_train).reshape(-1, 1)
```

Fitting training dataset in Logistic Regression model

```
[181]: n = LogisticRegression()
       n.fit(x_train, y_train)
```

```
[181]: LogisticRegression()
```

Evaluating training dataset

[183]:
```
y_predict_train = n.predict(x_train)
r2_train =  r2_score(y_true = y_train, y_pred = y_predict_train)
```

[184]:
```
round((r2_train), 2)*100
```

[184]: 73.0

Classification report for Training dataset

[186]:
```
print(cr(y_true = y_train, y_pred = y_predict_train))
```

```
              precision    recall  f1-score   support

           0       0.96      0.97      0.97      1050
           1       0.90      0.89      0.90       362

    accuracy                           0.95      1412
   macro avg       0.93      0.93      0.93      1412
weighted avg       0.95      0.95      0.95      1412
```

[187]:
```
x_test = x_test.drop(["WINNER"], axis = 1)
y_test = np.array(y_test).reshape(-1, 1)
```

Fitting testing dataset into Logistic Regression model

[189]:
```
n = LogisticRegression()
n.fit(x_test, y_test)
```

[189]: LogisticRegression()

Evaluating testing dataset

[192]:
```
y_predict_test = n.predict(x_test)
r2_test = r2_score(y_true = y_test,y_pred = y_predict_test)
```

[194]:
```
round((r2_test), 2)*100
```

[194]: 64.0

Classification report for Testing dataset

[196]:
```
print(cr(y_true = y_test, y_pred = y_predict_test))
```

```
              precision    recall  f1-score   support

           0       0.95      0.95      0.95       429
           1       0.88      0.87      0.87       177
```

```
    accuracy                           0.93       606
   macro avg       0.91      0.91      0.91       606
weighted avg       0.93      0.93      0.93       606
```

The winner candidate is

```
[ ]: z["NAME"][z["OVER TOTAL VOTES POLLED \nIN CONSTITUENCY"] == max(z["OVER TOTAL␣
     ↪VOTES POLLED \nIN CONSTITUENCY"])]
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```