Election Result Analysis

Ishita Chakravarthy 2018B4AA0670G

Nupur Funkwal *2018A7PS0624G*

Shivansh Rustagi 2018A8PS0745P

The following report focuses on the project on predictive analysis of election results in India pursued as a part of Practice School - I internship at PASS Consulting, Hyderabad. The project aimed to build a model to predict the election outcomes by analyzing data on demographic parameters and the political background of contesting candidates. The report focuses on the three urban constituencies assigned- Bangalore, Hyderabad and Secunderabad.

Data collection

The first step to building a predictive model was to collect necessary data. Data regarding the candidates who have contested for elections, demographics of the constituency, infrastructure and various other factors which might play a role in the elections was collected and analysed.

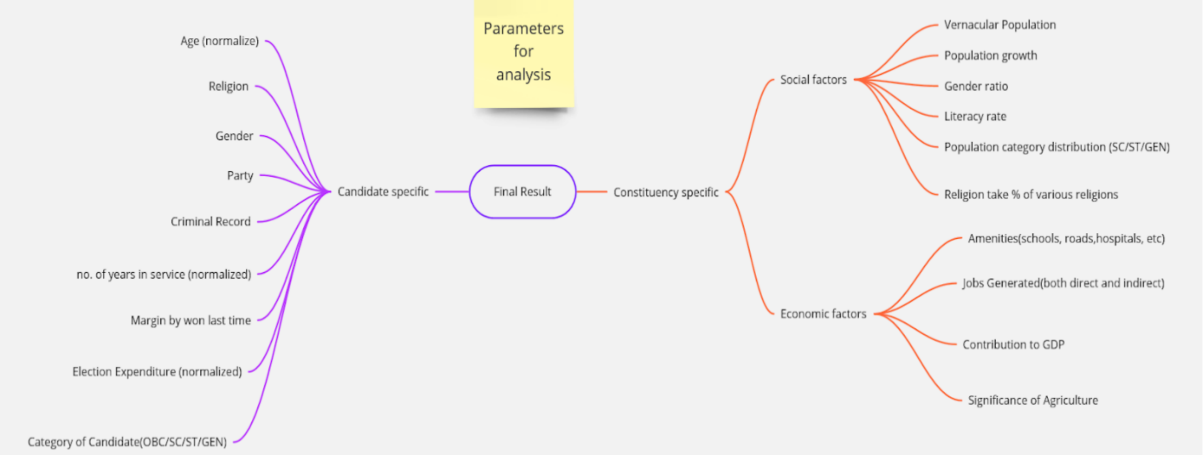
Initially, demographic data was collected from Census 2011 and electoral data was collected using ECI website and MyNeta. Shrug Repository was also used for this purpose as it had both demographic & electoral data cleaned and formatted properly in csv. This, however, led to a problem where Shrug doesn't map demographics data to constituencies which have overlapping borders with multiple district/ mandal/ villages (mostly for Urban Constituencies) [Shrug Codebook: pg 12,last para]. Hence, the data for Hyderabad & Secunderabad was missing completely. The Election data for Hyderabad and Secunderabad was obtained from Lok Dhaba, due to its unavailability in Shrug.

A major problem faced was, for urban constituencies, the census divisions (geographic) and the constituencies were not the same, resulting in no or poor mapping of the census data to constituency (one of the example). One workaround was approximating the subdistrict and clubbing their data together for using in Lok Sabha constituency election. Downside to this was the inaccuracy that can arise due to accumulation of a lot of data points and loss of useful micro-information (ward/ subdistrict wise info) rendering this method unusable for Assembly Constituencies.

Moreover, though the data regarding other parameters was available, it was often not in an appropriate format to club it with the election result data. The data was present in the form of pdfs and images of old government documents. Using various online OCR tools, the data was converted into CSV format.

Data visualization and analysis

After collecting all the necessary data, the next step was to understand the data collected and draw conclusions from it. Graphs such as scatter plots, bar charts, line plots etc. were plotted initially to figure out some general trends in the data against time. This would also help to determine the factors suitable to include in the predictive model. With the help of these plots, various trends regarding the voter population, the winning party, the margin of votes between the winner and runner-up, and demographics of each constituency were analysed. The significant factors identified are given below.

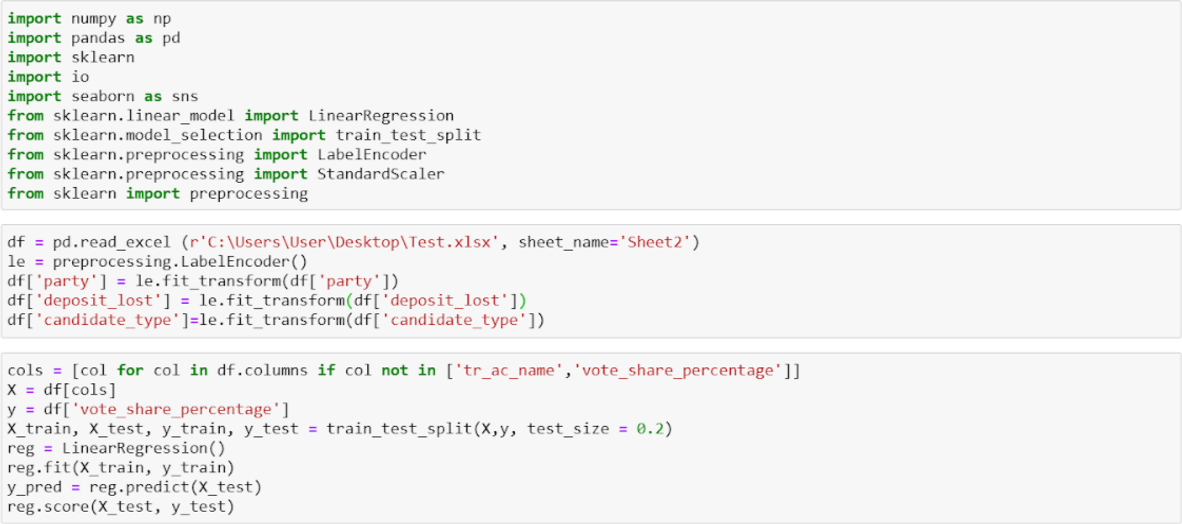

Parameters for analysis
Age (normalize)
Religion
Gender
Party
Criminal Record
no. of years in service (normalized)
Margin by won last time
Election Expenditure (normalized)
Category of Candidate(OBC/SC/ST/GEN)
Candidate specific
Final Result
Constituency specific
Social factors
Vernacular Population
Population growth
Gender ratio
Literacy rate
Population category distribution (SC/ST/GEN)
Religion take % of various religions
Economic factors
Amenities(schools, roads,hospitals, etc)
Jobs Generated(both direct and indirect)
Contribution to GDP
Significance of Agriculture

Model building

The predictive model which was built was based on Linear regression. For various parameters like demographics, candidate history, previous election results, the model was built by assigning coefficients obtained.

Demographic data was included in the Regression Model, but it gave muddy result with very low accuracy, hence the demographics data such as literacy rate, religious population distribution etc was dropped from the model (we ran this for Sivaganga - Lok Sabha Election, here data was taken from Census 2011 and 2001 and the points were extrapolated using a simple linear population growth formula). Another reason for lack of inclusion of the demographic data was that Shrug had the data for district(which includes parts of multiple LS constituencies) and it was difficult to relate this data of district level to smaller Assembly Constituencies. Hence, in the final model ran on candidate data and previous election results for the assembly constituencies.

For the implementation of the model, Python libraries such as Sklearn, Matplotlib, Seaborn, Pandas and Numpy were used. First, the data file was imported into a dataframe and the required factors (columns) were filtered out. The preditive model, determines the percentage vote share of the winner for an election. The predictive factors were enop, n_cand, party, turnout_percentage, candidate_type, deposit_lost and vote_share_percentage was predicted. The data was split into training and testing data. Through this, we determined the coefficients of each predictive factor and analysed which factors were more significant than others. The accuracy of this model for urban constituencies turned out to be 81% for Hyderabad, 87% for Secunderabad and 89% for Bangalore south. The weights mentioned below are for Bangalore South.



```
import numpy as np
import pandas as pd
import sklearn
import io
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn import preprocessing
```

```
df = pd.read_excel (r'C:\Users\User\Desktop\Test.xlsx', sheet_name='Sheet2')
le = preprocessing.LabelEncoder()
df['party'] = le.fit_transform(df['party'])
df['deposit_lost'] = le.fit_transform(df['deposit_lost'])
df['candidate_type']=le.fit_transform(df['candidate_type'])
```

```
cols = [col for col in df.columns if col not in ['tr_ac_name','vote_share_percentage']]
X = df[cols]
y = df['vote_share_percentage']
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2)
reg = LinearRegression()
reg.fit(X_train, y_train)
y_pred = reg.predict(X_test)
reg.score(X_test, y_test)
```

Weight	Feature
2.1034 ± 0.7719	deposit_lost
0.0015 ± 0.0083	turnout_percentage
0.0010 ± 0.0059	candidate_type
-0.0009 ± 0.0101	party
-0.0014 ± 0.0054	n_cand
-0.0103 ± 0.0166	enop

Future Prospects of the model:

By training the model with more urban constituencies, the accuracy of the model can be improved. With a higher accuracy, the model can act as an aid political parties in effectively choosing between prospective candidates depending on the input parameters of the candidate.