
Python Assignment Report

Prathamesh Baviskar
220285
April 2024
baviskarp22@iitk.ac.in

1 Methodology

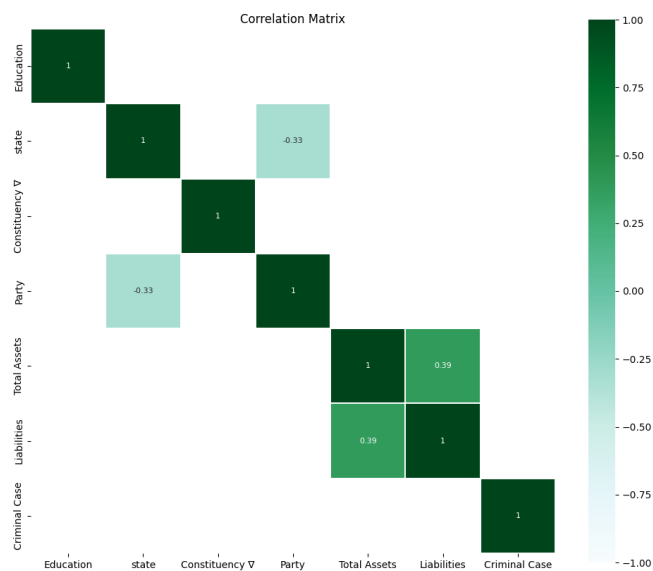
1.1 Data Preprocessing Steps

Basic Preprocessing^[1] was carried out on the data. This included steps like:

1. The Total Assets and Liabilities, present as categorical values with various suffixes (Thou+, Hund+, Lac+, Crore+), were converted to their numerical counterpart.
2. Other categorical values such as Party, state, Education, and Constituency were label encoded.
3. No duplicates were found. Also, there were no missing values in the train data.

1.2 Feature Engineering

1. Useless Features like ID and Candidate Names were dropped. Constituencies were also unique in most of the data but were retained in the feature vector.



2. The difference between the Total Assets and Liabilities was also incorporated as a new feature. This was done after a positive correlation of around 0.39, which suggested a moderate positive linear relation was revealed.

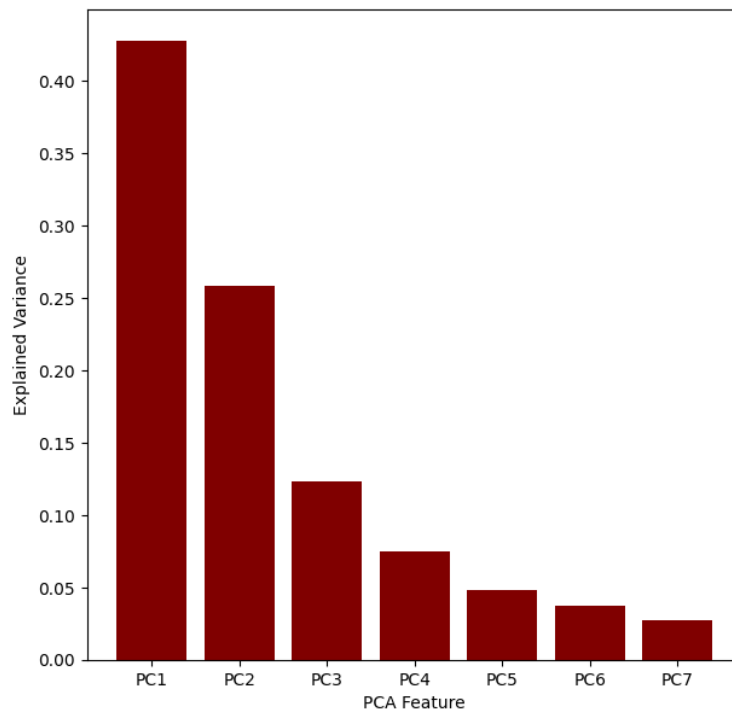
1.3 Identifying Outliers

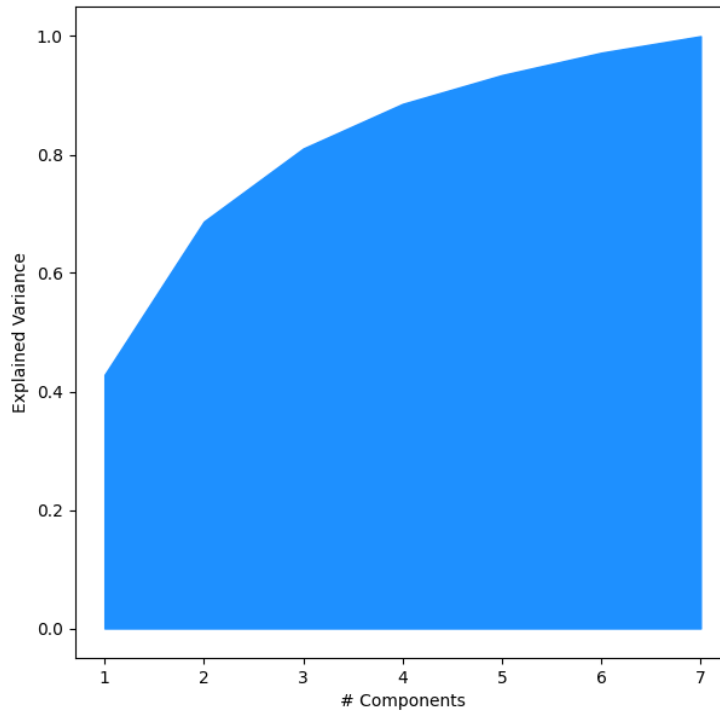
Z-Score Normalization was used to identify and remove outliers. The Z-score Normalization (which standardizes the data by subtracting the mean and dividing by the standard deviation) score for Criminal Records, Assets, and Liabilities was calculated and used to eliminate outliers.

1.4 Dimensionality Reduction Techniques

Principal Component Analysis was used to find the principle components that explained most of the variance in the data.

PC1: Constituency
PC2: Party
PC3: Criminal Case
PC4: Total Assets
PC5: Liabilities
PC6: state
PC7: Wealth





1.5 Normalization, Standardization, or Transformation Used

Criminal Cases, Total Assets, and Liabilities values were highly right-skewed. Logarithmic transformation was applied to these features to reduce this skewness.

1.6 Other

Along with preprocessing, `StratifiedShuffleSplit` was used for splitting data into train and test sets while maintaining the same class distribution in both sets. This was done to handle the class imbalance. The oversampling technique ADASYN was tried although it did not provide an improvement in the F1-score.

2 Experiment Details

2.1 Models Used^[2]

In most of the Submissions, RandomForestClassifier was trained on the test data set. The best hyper-parameters for the model were found using GridSearchCV which was as follows:

max_depth	20
min_samples_leaf	3
min_samples_split	5
n_estimators	400

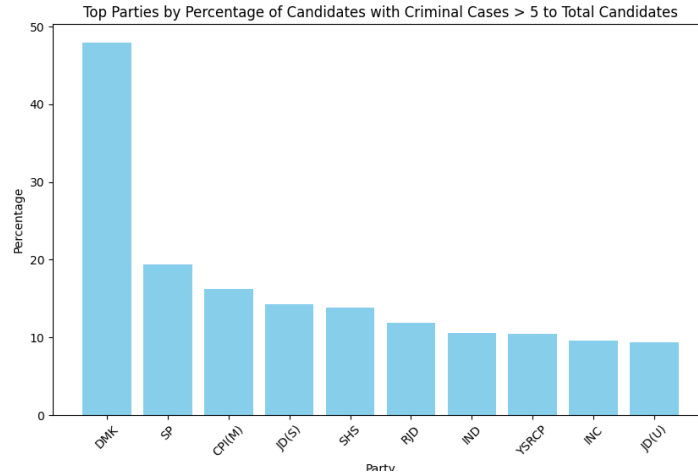
Some of the submissions also used other classifiers, such as XGBoost, SVC, and BernoulliNB, though their accuracy was less than that of RandomForest.

5-Fold-Cross-Validation was also used to measure the performance of the model.

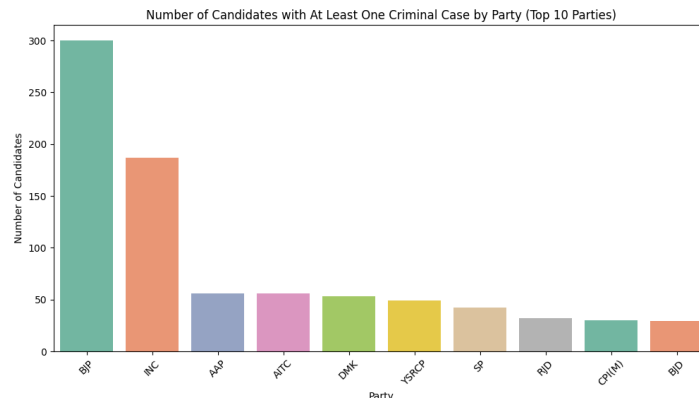
3 Data Insights

3.1 Parties and Criminal Cases

Here, I have plotted the percentage among the total candidates of a party who have more than 5 Criminal Cases against them. As we can see DNK has the highest percentage of such candidates.

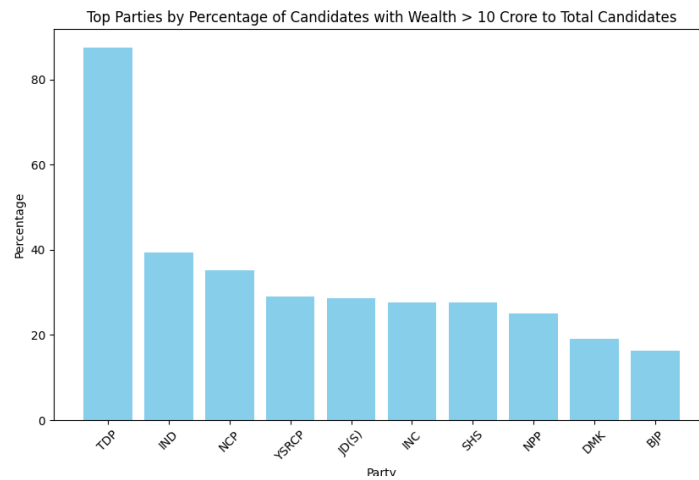


Next, I have plotted the number of candidates having criminal background present in the party. As we can see that BJP has the highest number of such candidates. This can also be attributed to the fact that BJP has the highest number of total Candidates.



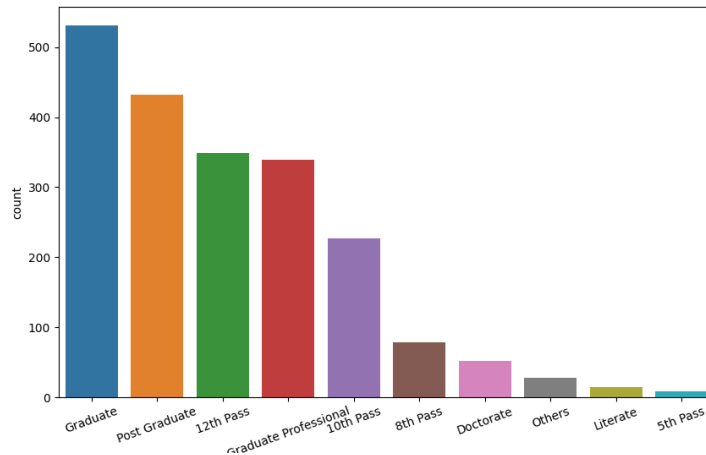
3.2 Wealth and Parties

Wealth been defined as assets minus the Liabilities, we get the following plot, number of Candidates having a wealth of more than 10 Crore versus their Political Party.

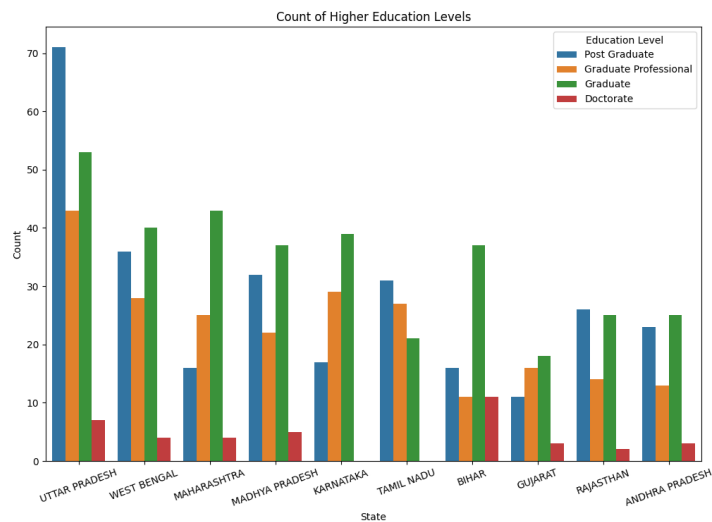


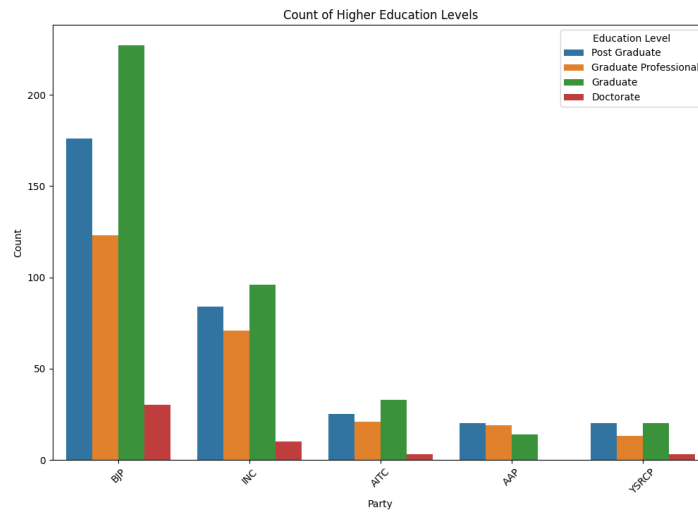
3.3 Additional Informative Plots

The highest Number of Candidates are Graduates followed by Post Graduates.



Plots showing the distribution of Education Level in different States and Parties can also be plotted. This helps us understand what frequency of a particular label is found in a state/Party and can help better predict the Education for unseen data.





4 Results

The final F1 Score (public) is 0.24303 and private is 0.23175. The public leaderboard Rank is 87, and the private leaderboard rank is 114.

5 References

- [1] 6.3. Preprocessing data — scikit-learn 1.4.2 documentation
 - [2] 1.12. Multiclass and multioutput algorithms
 - [3] Multi-Class Classification
- Kaggle Feature Engineering