Python Assignment Report

Animesh Madaan

April 13, 2024

1 Methodology

The methodology employed for data preprocessing and feature engineering is described below:

1.1 Data Preprocessing

1. **Loading Data:** The training and test data are loaded from CSV files using pandas, a Python data manipulation library.
2. **Label Encoding:** Categorical features such as 'Party', 'State', and 'Education' are encoded using LabelEncoder to convert them into numerical values suitable for machine learning algorithms. This step ensures that the categorical data can be effectively utilized in the training process.
3. **Modifying Numeric Columns:** Certain numeric columns, such as 'Total Assets' and 'Liabilities', contained non-numeric characters like 'Crore', 'Lac', etc. These characters were removed using regular expressions, and the columns were converted to integers. This ensures consistency and compatibility with numerical operations during subsequent analysis.

1.2 Feature Engineering

1. **Feature Selection:** Prior to model training, some columns deemed unnecessary for analysis, such as 'ID', 'Candidate', and 'Constituency', were dropped from the datasets. Additionally, columns 'Total Assets' and 'Liabilities' were dropped as they were found to contribute marginally to training accuracy. This step streamlines the dataset and focuses on relevant features for model learning.
2. **Standardization:** To ensure uniformity and comparability among features, the remaining features were standardized using StandardScaler. This transformation brings all features to the same scale by subtracting the mean and dividing by the standard deviation. Standardization facilitates effective model training and improves convergence during optimization.

2 Experiment Details

2.1 k-Nearest Neighbors

Hyper parameter	**Value**
n_neighbors	20, 40, 60, 80, 100
weights	uniform
algorithm	auto, brute
p	1

Table 1: Hyper-parameters used in the kNN classifier

The best Hyper-parameters found using random sampling and picking the ones that give highest f1-score on training set are – 'n_neighbors': 20, 'weights': 'uniform', 'algorithm': 'auto', 'p': 1.

2.2 Extra Trees Classifier

Hyper parameter	Value
n_estimators	180, 190, 200
max_depth	18, 20
min_samples_split	1, 2
min_samples_leaf	1, 2
max_features	None

Table 2: Hyper-parameters used in the Extra Trees Classifier

The best Hyper-parameters found using GridSearchCV that give highest f1-score on training set are – 'max_depth': 20, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 190.

2.3 Support Vector Classifier

Hyper parameter	Value
C	0.1, 1, 10, 100, 1000
kernel	rbf,linear
degree	1, 2

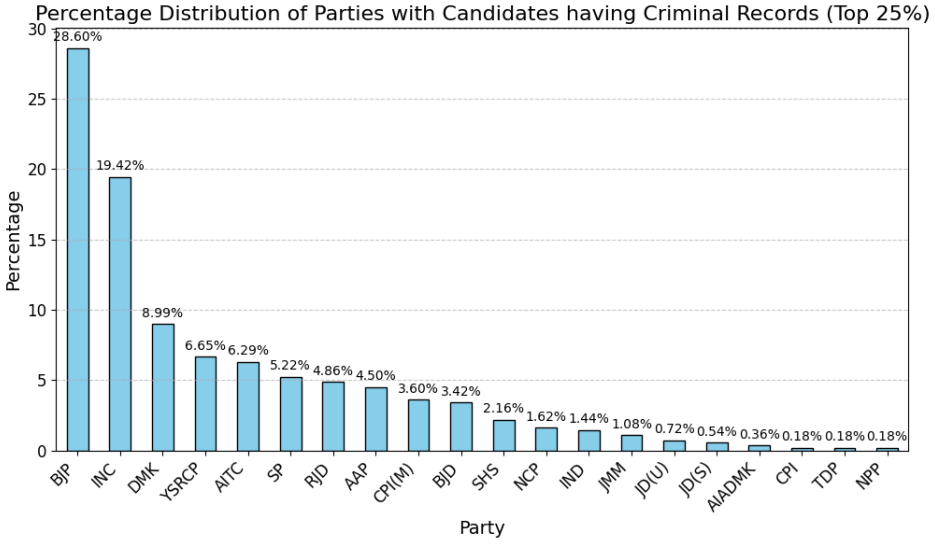
Table 3: Hyper-parameters used in the Support Vector Classifier

The best Hyper-parameters found using GridSearchCV that give highest f1-score on training set are – 'C': 1000, 'degree': 1, 'kernel': 'rbf'.

The best f1-score on test data (on kaggle) is given by k-Nearest Neighbors.

3 Data Insights

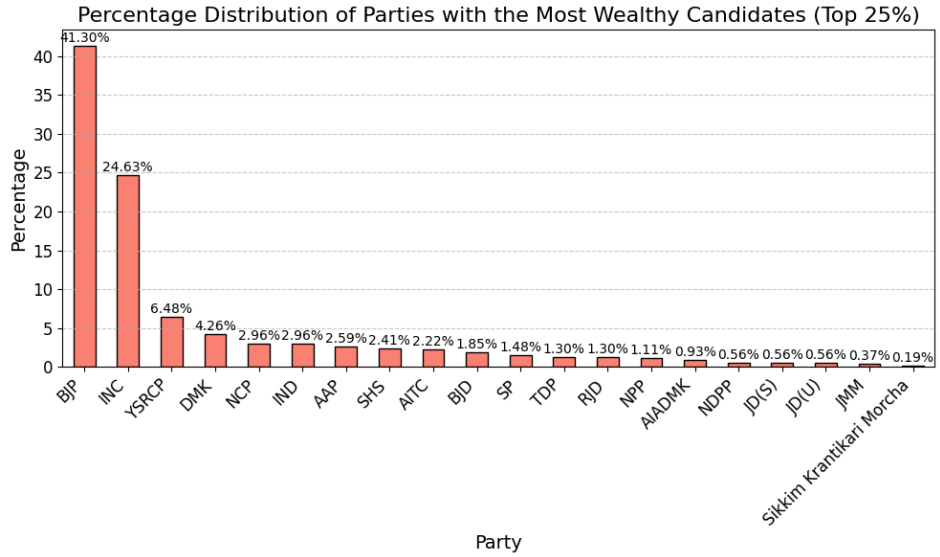
3.1 Percentage distribution of parties with candidates having the most criminal records



- The analysis reveals that among political parties, BJP, INC, and DMK are ranked as the top three parties in terms of the percentage of candidates with criminal records above the 75^{th} percentile.

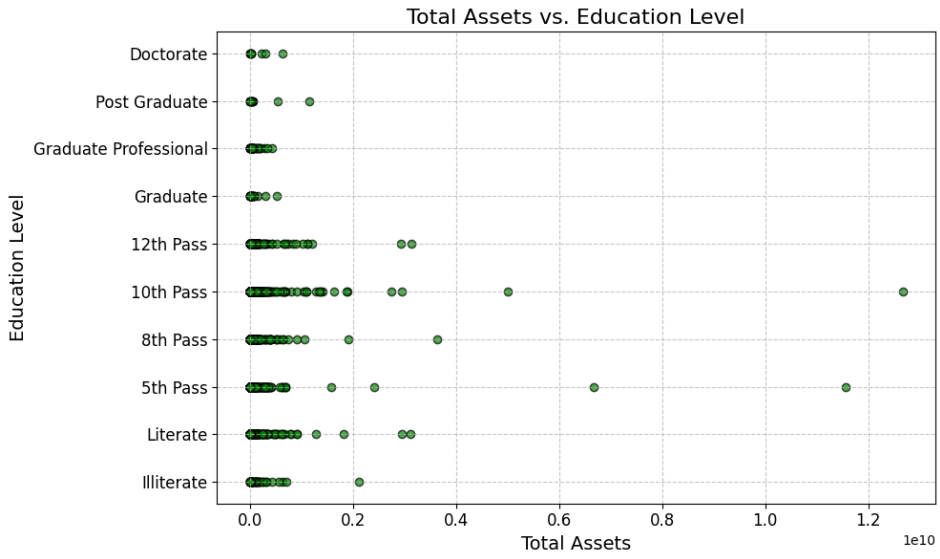
- This trend sheds light on the prevalence of candidates with criminal backgrounds within these parties, which may influence public perception and voter behavior.

3.2 Percentage distribution of parties with the most wealthy candidates



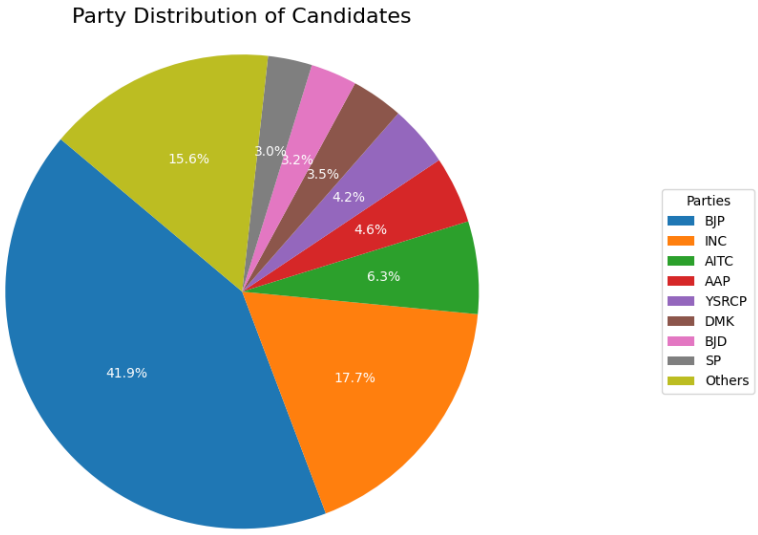
- The examination indicates that BJP and INC emerge as the leading parties in terms of the percentage of candidates with wealth over the 75^{th} percentile.
- This observation underscores the financial strength of candidates affiliated with these parties, which could impact their campaign strategies and resource allocation during elections.

3.3 Total Assets vs. Education Level



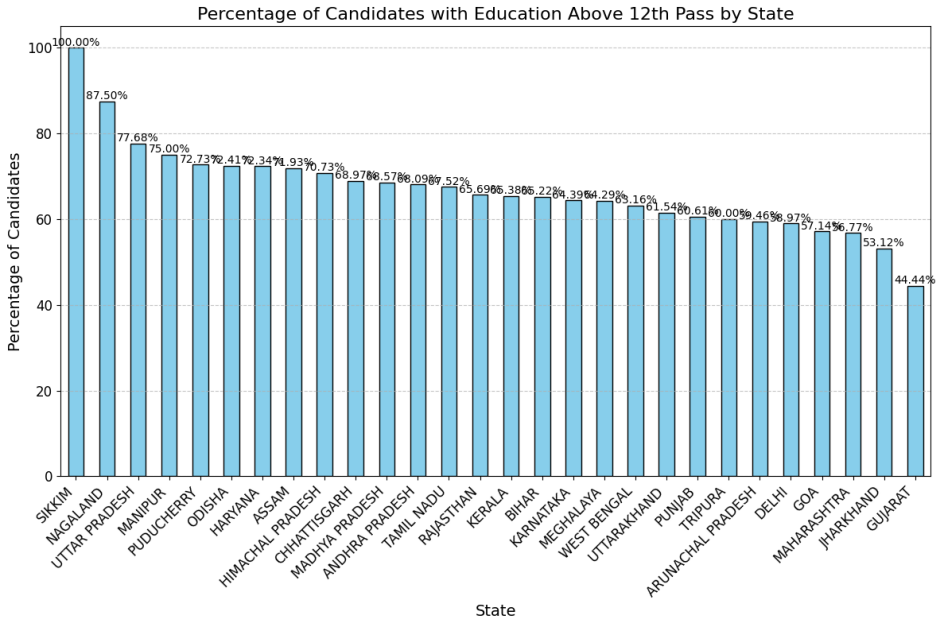
- The scatter plot demonstrates a negative correlation between education level and total assets among political candidates in India.
- This trend suggests that candidates with higher education levels tend to have lesser wealth, indicating a trend towards a middle-class background in political representation.

3.4 Party Distribution



- The pie chart illustrates that BJP and INC are the two leading parties, with the highest number of candidates among all political parties.
- This observation highlights the dominance of these two parties in terms of candidate representation, indicating their strong presence in the political landscape of India.

3.5 Percentage of candidates above 12^{th} Pass in each state



- The analysis reveals significant variation in the percentage of candidates above 12^{th} pass across states, with Sikkim exhibiting the highest percentage and Gujarat the lowest.
- This disparity underscores the importance of regional dynamics and educational policies in shaping the educational attainment levels of political candidates in different states of India.

4 Results

Final F1 Score:

My model achieved a final Public F1 score of **0.26369**. This score is obtained using the k-Nearest Neighbors model.

Leaderboard Ranks:

- **Public Leaderboard Rank: 28**
- **Private Leaderboard Rank:** Not Released

5 References

During the completion of this assignment, I found the following resources particularly helpful:

- **scikit-learn Documentation:** The scikit-learn library provided invaluable tools and algorithms that greatly contributed to the success of this assignment. Without its comprehensive documentation, navigating the intricacies of machine learning would have been a daunting task. I owe a great deal of gratitude to the scikit-learn team for their outstanding work. Link to documentation here.
- **pandas Documentation:** The pandas library played a crucial role in data manipulation and preprocessing, facilitating efficient analysis and visualization. The documentation can be accessed here.
- **Improving Class Imbalance with Class Weights in Machine Learning:** This insightful Medium article provided valuable insights into addressing class imbalance. I used this in SVC. Link to the article here.

6 GitHub Link

The link to the code base containing the models and scripts for data plotting can be found here.