Indian Institute of Technology, Kanpur

CS253: Python Programming Assignment

Assignment Report

Pathe Nevish Ashok (220757)

April 14, 2024

1 Methodology

1.1 Data Preprocessing

The dataset given to us had the columns `Total Assets` and `Total Liabilities`. These columns contained the amounts not in numerical format but in word format (e.g. `One Crore+` instead of `10000000`). Thus, these columns were preprocessed, to convert into numeric values.

Also, the columns `Party`, `State` and `Constituency` corresponded to categorical variables, so there were One-Hot Encoded. This was done using the `sklearn.preprocessing.OneHotEncoder` [Ped+11] class wrapped inside a `ColumnTransformer`.

Why One-Hot Encoding and not Label Encoding? Label Encoding would have assigned a numerical value to each category, which would have been misleading to the model. For example, if we had used Label Encoding on the `State` column, the model would have assumed that the states are ordinal, which is not the case. Although One-Hot Encoding increases the dimensionality of the dataset, it is a better choice in this case.

1.2 Feature Engineering

A new column `Wealth` was created by subtracting the `Liabilities` from the `Total Assets` column. This column was created to see if the wealth of a candidate has any effect on the criminal cases filed against them. Didn't give any significant improvement in the model, but was kept for analysis purposes. So was later dropped.

1.3 Identifying Outliers

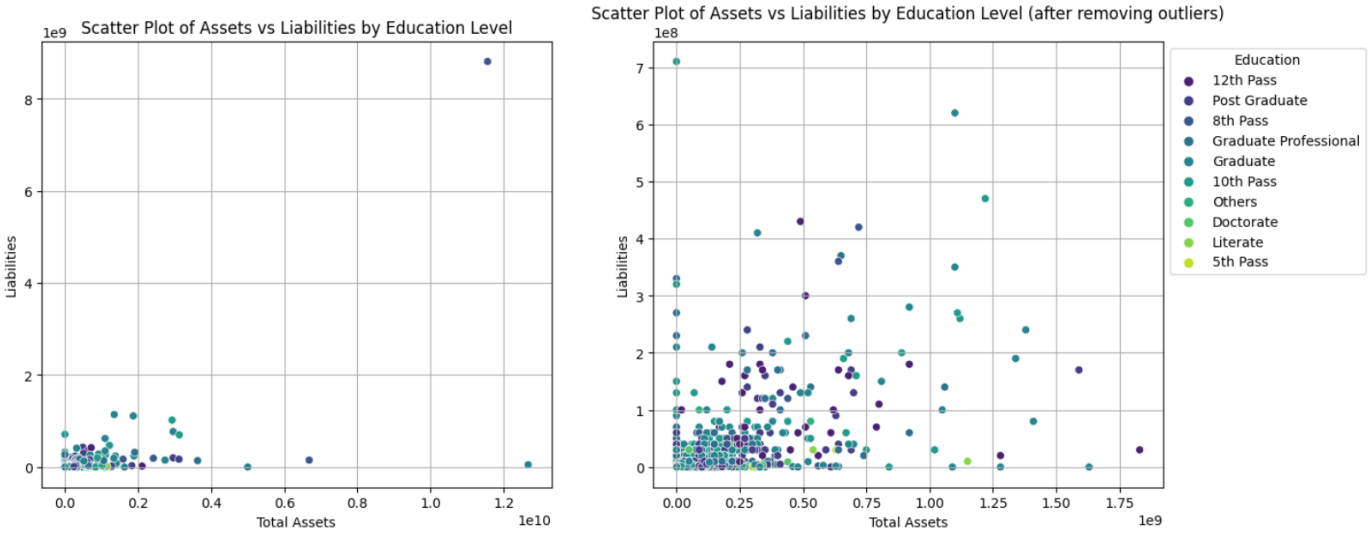

Figure 1: Scatter Plot of Assets vs Liabilities

Although very few, but there were some outliers in the dataset. These were identified using the `Z-Score` method and were removed from the dataset (`threshold = 3`).

1.4 Normalization, Standardization and Transformation

Like categorical variables, numerical variables were also preprocessed. The `Total Assets`, `Liabilities` and `Criminal Case` columns were normalized using the `StandardScaler` class from `sklearn.preprocessing`. [Ped+11]

1.5 Data Augmentation

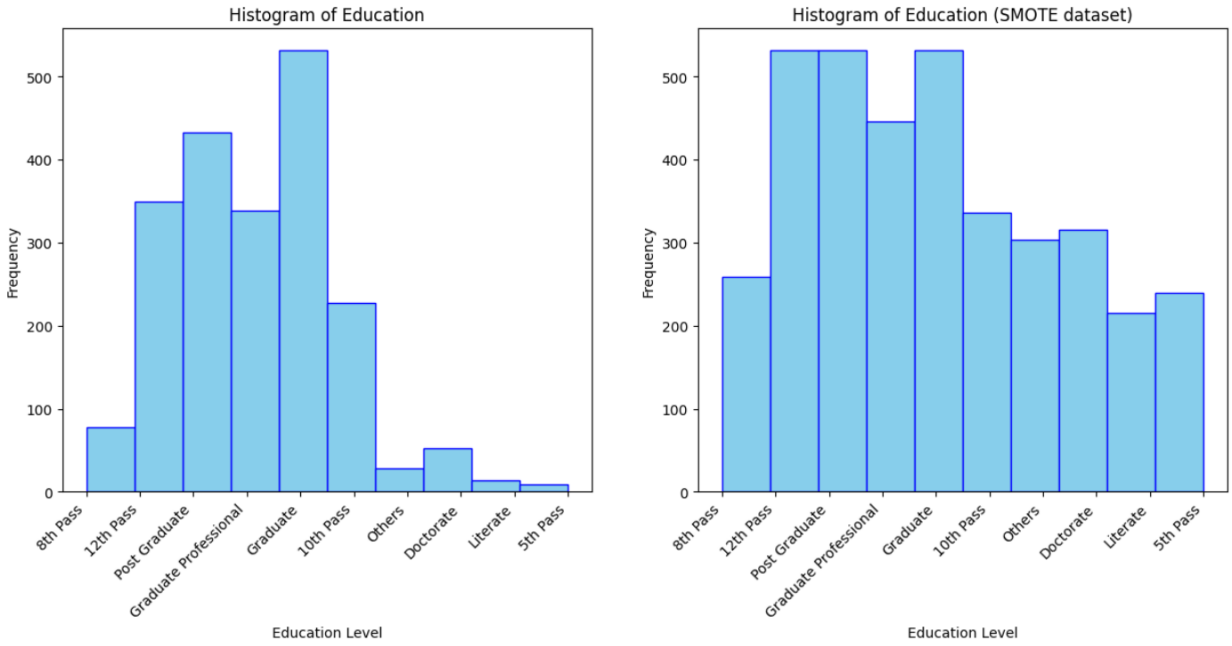

Figure 2: Histogram of Education

Why the need for data augmentation? We can clearly see that the dataset is imbalanced. The classes (`Others, Doctorate, Literate, 5th Pass, 8th Pass`) have very few samples, while the class `Graduate` has the most samples. The dataset was augmented using the `SMOTE` class from `imblearn.over_sampling` module. [LNA17] Number of samples were increased from 2059 to 3707. [Cha+02]

The ADASYN [He+08] method was also tried, but it did not give better results than SMOTE.

2 Experiment Details

The jupyter notebook containing code and graphs is available at this Github Repo.

2.1 Models Used

The train dataset was split into 80% training and 20% validation dataset. The following models were used to train the dataset:

Model (Best)	Parameters
Random Forest Classifier	`n_estimators=100`
SV Classifier	`kernel='rbf', C=7`
KNN Classifier	`n_neighbors=10`
Decision Tree Classifier	`max_depth=15, criterion='gini', splitter='best'`

Table 1: Unique Values in each Column

Out of these models, the Random Forest and Support Vector Classifier gave the best results. Decision Tree models are prone to overfitting, so the Decision Tree Classifier was not used in the final model. The SVM model was trained for final predictions.

The optimal value of `C` for out SVC Model was grid-searched over different combinations of `C` and `kernel`

A Bernoulli Naive Bayes model was also tried, but didn't give significant improvement, so was later dropped.

2.2 Data Insights

(We are doing the analysis on the original `train.csv` training dataset)

Column	Unique Values
ID	2059
Candidate	2039
Constituency	2037
Party	23
Criminal Case	35
Total Assets	210
Liabilities	170
State	28
Education	10

Table 2: Unique Values in each Column

Note 1: The Constituency column has 2037 unique values. But even after considering Constituency as a categorical variable and One-Hot Encoding it the model reported an anomalous increase in F1 score, which might be due to increased dimensionality. Thus, the Constituency column was dropped.

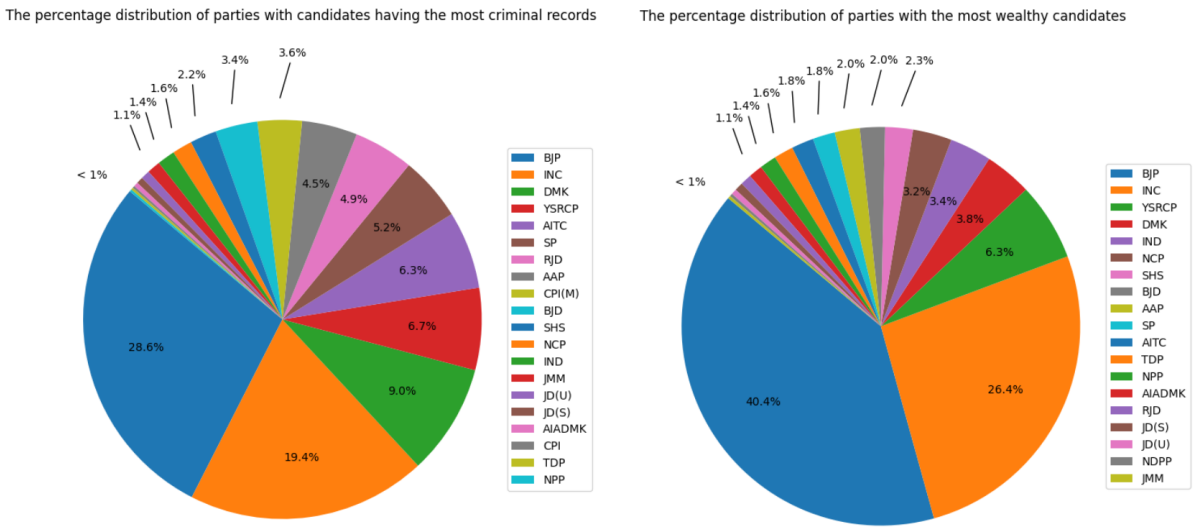

Figure 3: Percentage Distribution of Parties according to most Crime Cases, Wealth

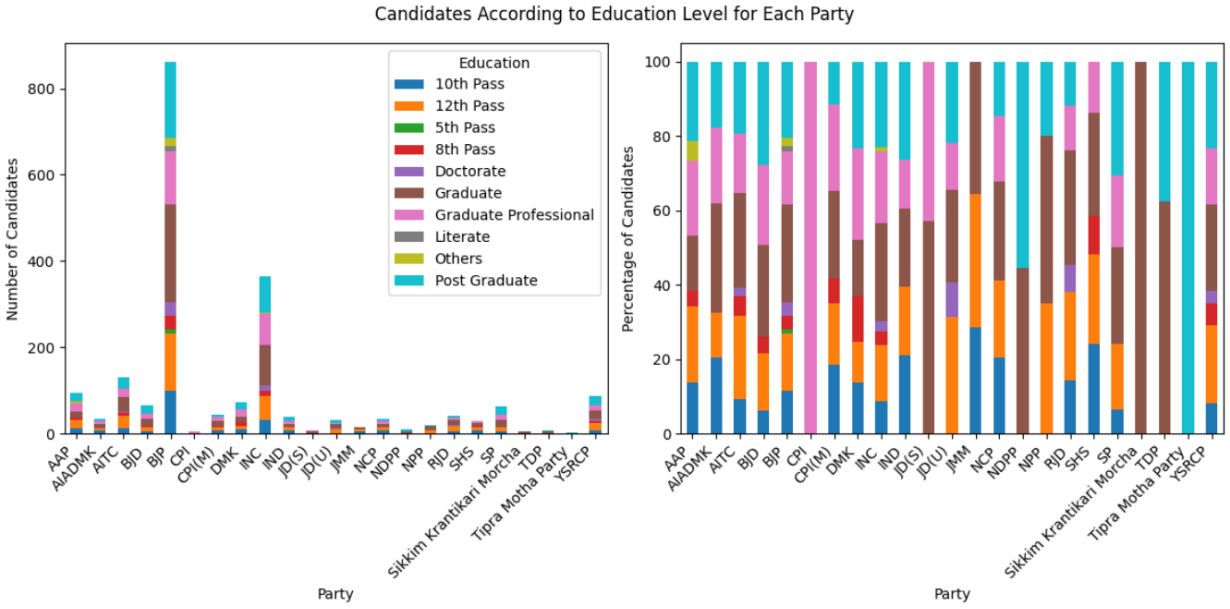

Figure 4: Candidates' Education Distribution according to Party

Figure 5: Correlation Matrix

Here, we can see that the `Total Assets` and `Liabilities` columns are highly correlated. Note that we have removed the `Constituency` column, so it is not present in the correlation matrix.

3 Results

Leaderboard	Score	Rank*
Public	`0.26277`	32
Private	`0.25377`	35

Table 3: Leaderboard Scores and Ranks

Final F1 Score : `0.25377`
Personal Best F1 score on the private dataset was `0.26156` (not evaluated, Apparent Rank 19), while that on public dataset was `0.26277`.

4 References

[Cha+02] N. V. Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (June 2002), pp. 321–357. ISSN: 1076-9757. DOI: `10.1613/jair.953`. URL: `http://dx.doi.org/10.1613/jair.953` (cit. on p. 2).

[He+08] Haibo He et al. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning". In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 2008, pp. 1322–1328. DOI: `10.1109/IJCNN.2008.4633969` (cit. on p. 2).

[LNA17] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning". In: *Journal of Machine Learning Research* 18.17 (2017), pp. 1–5. URL: `http://jmlr.org/papers/v18/16-365.html` (cit. on p. 2).

[Ped+11] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on pp. 1, 2).