CS253 Python Assignment

Saagar K V
220927

April 13, 2024

Github repository containing the code:

1 Methodology

The dataset consists of 8 features - ID, Candidate Name, Constituency, Party, Number of criminal cases, Total Assets, Liabilities, and State. The target was to predict the Education level of the candidate.

1.1 Feature Selection

ID, Candidate Name, and Constituency are unique to each entry in the dataset. Thus, they play no role in training a model to predict the education level. Thus, these 3 features were removed before training the model.

1.2 Data Preprocessing

Number of criminal cases is an integer. However, the other four features were string-valued. In particular, Total Assets and Liabilities took values such as '2 Crore+', '29 Lac+', '60 Thou+' etc. These were converted into numbers using a function. The function used a simple if-else-if conditional statement to check the presence of substrings 'Crore+','Lac+','Thou+' and appropriately converted them to numbers.

The features 'party' and 'state' were label encoded before training the model. This was done using *sklearn.preprocessing.LabelEncoder* [1]. Education level (in the training set) was encoded using *sklearn.preprocessing.OrdinalEncoder* [2]. This is because there is a relative ordering among the education levels. The order used was:
0 - Literate
1 - 5th Pass
2 - 8th Pass
3 - 10th Pass
4 - 12th Pass
5 - Graduate

6 - Graduate Professional
7 - Post Graduate
8 - Doctorate
9 - Others

It was also expected that this ordering would enable us to solve this problem as a regression problem instead of a classification problem (by rounding the predicted values to the nearest integer). However, it was later found that regression algorithms did not give better results than classification algorithms. The encoding was not changed because this is as good as label encoding, for a classification problem.

1.3 Standardisation

In the case of using the *KNeighborsClassifier*, all parameters were standardised using *sklearn.preprocessing.StandardScaler* [3]. This is to ensure that the absolute values of the features do not introduce a bias in the algorithm since it is based on distance between points. Thus, all features are individually centered and scaled so that each of them now looks almost like standard normally distributed data (e.g. Gaussian with mean 0 and variance 1).

1.4 Feature Transformation, Dimensionality Reduction

Neighborhood Components Analysis (NCA):

NeighborhoodComponentsAnalysis [4] is a dimensionality reduction technique in scikit-learn that aims to improve the performance of classification algorithms by learning a linear transformation of the feature space. It is a supervised method that takes into account class labels (education levels) during the transformation, attempting to improve the discriminative power of the features for the classification problem. In one of the experiments, a pipeline was created combining NCA for feature transformation followed by kNN (k-Nearest Neighbours) for classification.

2 Experiment Details

2.1 Arriving at the final model

A *pairplot* [5] was plotted between the 5 features under interest along with Education level.

From this, it is clear that linear models won't do good. Trials with linear models gave poor results as expected. It can be seen that the data is not very easy to be trained with. This is because of various reasons like almost uniform distribution of points in education vs state (for some education levels) and party vs state. Liabilities do not vary much with education levels. There is a better scenario with total assets and criminal cases. However, there are points at all education levels which are associated with low values of criminal cases or total

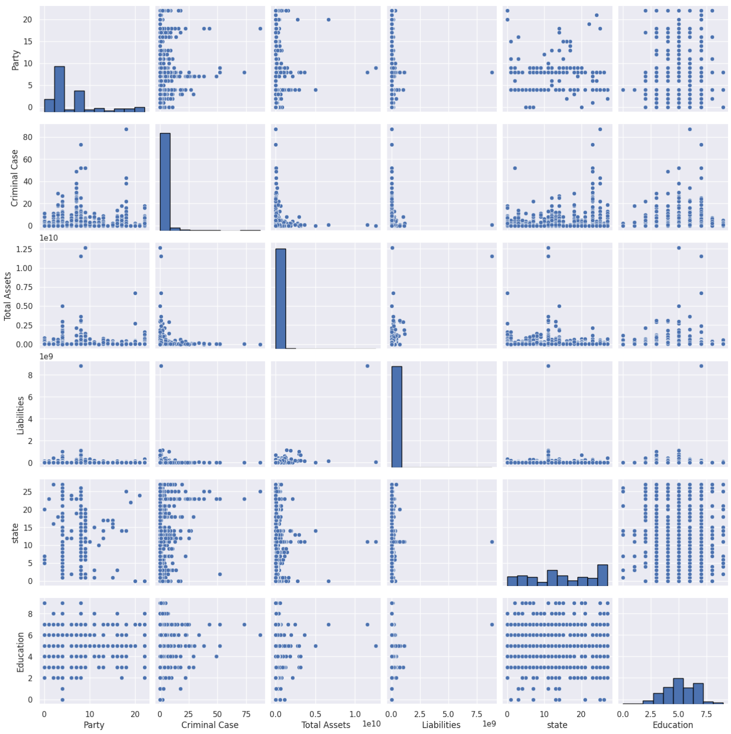

Figure 1: Pairplot

assets. In fact, the density of such points is high. The relation between two features excluding education level is not so clear.

Thus, clustering algorithm (k-Nearest Neighbours (kNN)) was the first choice. Later, since no clear relation was observed in the dataset, decision trees and random forests were tried. Few other methods were also tried as shown in the table. [6] [7] [8] [9]

A total of 19 submissions were made. Only the major ones are mentioned here. Apart from this, most experiments were mainly done by splitting the train data into train and test sets.

Model	**Hyperparameters**	**F1 Score (Private)**	**F1 Score (Public)**
KNeighborsClassifier	n_neighbors=5	0.23577	0.21420
KNeighborsClassifier	n_neighbors=7	0.22587	0.20986
DecisionTreeClassifier	Default	0.17344	0.20772
NCA + kNN Pipeline	n_neighbors=6	0.20073	0.16783
BernoulliNB	Default	0.13231	0.11744
RandomForestClassifier	n_estimators=250	0.20092	0.20897
GaussianProcessClassifier	kernel=RBF with length scale = 1.0	0.15675	0.15977
GaussianProcessClassifier	kernel=Matern with length scale = 1.0, nu = 1.0	0.16276	0.16666
RandomForestClassifier	n_estimators = 200, min_samples_leaf = 5	0.24425	0.21082
RandomForestClassifier	n_estimators=300, min_samples_leaf = 7, bootstrap = False	**0.25050**	**0.23951**

2.2 Data Insights

2.2.1 Number of candidates (winners) from each party

This must be kept in mind while analyzing other parameters, as the number of candidates plays a major role in determining percentages and other statistics that are relative.

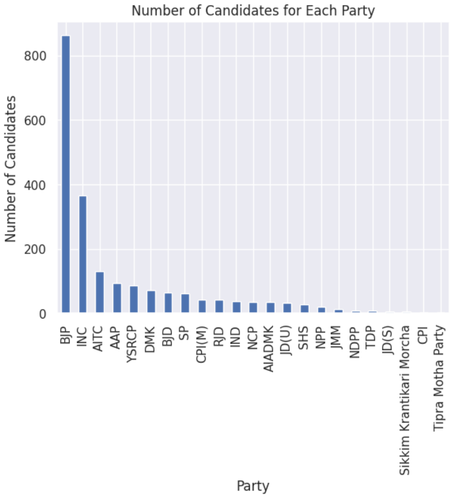

Figure 2: Number of candidates from each party

2.2.2 Analysis of the number of criminal cases of serious criminal winners

Considering the mean value of the number of criminal cases (1.78) and the standard deviation (4.76), as per the training set, let us call those candidates with atleast 7 criminal cases as serious criminals. Their data is analysed below:

The graphs obtained can lead to some interesting observations about the criminal cases on parties' candidates. It can be seen that DMK does not have the maximum average value of criminal cases but the percentage of criminal cases by DMK members is the highest. On the other hand, SP, although has the maximum average value, holds a comparatively lesser percentage contribution. This reflects the number of candidates in the party involved in crime. DMK has quite high number of winners who are involved in crime. The third indicates the presence of the most serious criminal in SP who has become a winner. However, a lesser number of SP candidates involved in crime have become winners, leading to lesser percentage contribution.

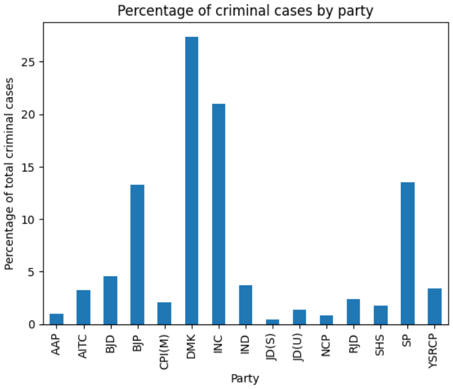

Figure 3: Percentage of criminal cases by party

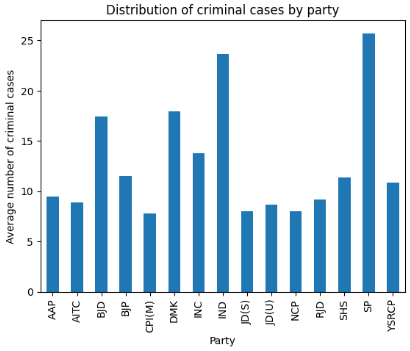

Figure 4: Average number of criminal cases

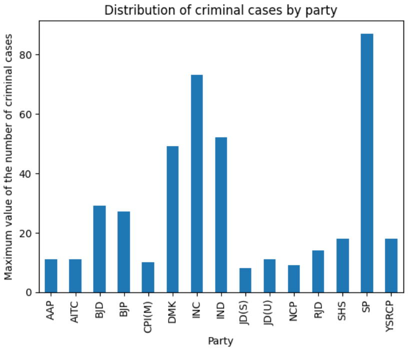

Figure 5: Maximum value of the number of criminal cases

2.2.3 Analysis of the wealth of very wealthy candidates

We define wealth of a candidate as total assets of the candidate minus the liabilities of the candidate.

Considering the mean wealth and the standard deviation of the candidates in the training set, we define very wealthy candidates as those whose wealth (assets - liabilities) is more than 65 crore. The data of such candidates is analysed below:

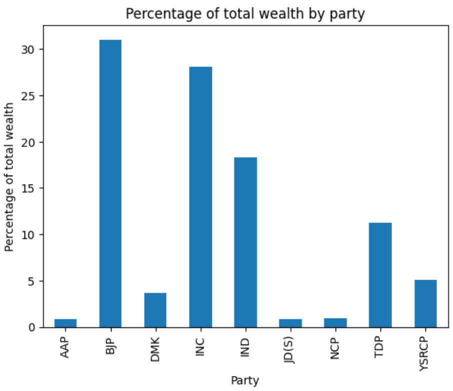

Figure 6: Percentage of total wealth (assets - liabilities) by party

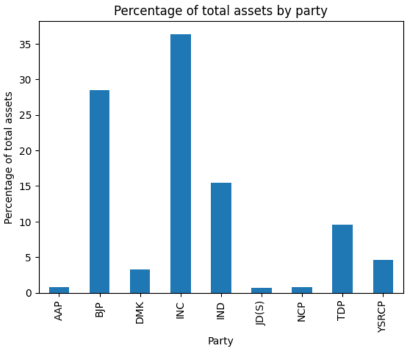

Figure 7: Total assets by party

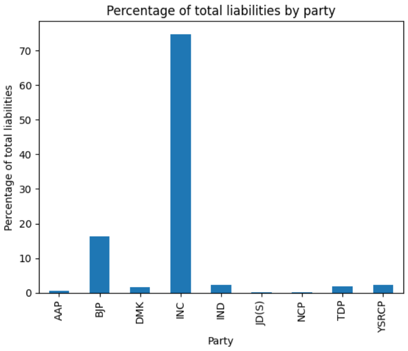

Figure 8: Total Liabilities by party

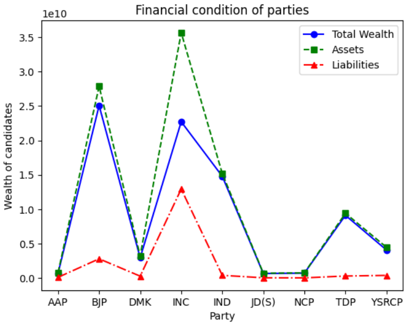

Figure 9: Financial condition of party's candidates

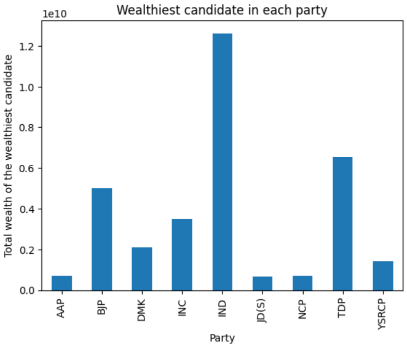

Figure 10: Wealth (assets - liabilities) of the wealthiest candidate in the party

One can observe how well the candidates of a party maintain their financial status. Some parties like INC possess large amount of assets but they also hold significant liabilities. While we have parties like BJP that have higher assets but relatively lower liabilities. However, the number of candidates belonging to a given party must always be kept in mind in all of these analyses.

3 Results

1. Final F1 Score (Private) - 0.25050
2. Final F1 Score (Public) - 0.24321
3. Final Leaderboard Rank (Private) - 48 (35 if we exclude deleted submissions above me in the leaderboard)
4. Final Leaderboard Rank (Public) - 86 (62 if we exclude deleted submissions above me in the leaderboard)

References

[1] Label Encoder

[2] Ordinal Encoder

[3] Standard Scaler

[4] Neighborhood Components Analysis

[5] Pairplot

[6] k-Nearest Neighbors

[7] Decision Tree Classifier

[8] Random Forest Classifier

[9] Gaussian Process Classifier