CS253: Assignment 3

Python Assignment

Name :	Chayan Kumawat
Roll No. :	220309
Github Repo :	Click
Program:	B.Tech
Branch:	CSE
Batch:	2026
Date of Sub:	13.04.2024

1 Methodology

In this section, we outline the methodology adopted for the multi-class classification task using machine learning techniques. Our primary objective is to develop a model capable of accurately predicting the educational backgrounds of candidates based on various features extracted from the provided dataset.

1.1 Data Preprocessing Steps

We began by preprocessing the dataset to prepare it for model training. The following steps were undertaken:

1. **Data Loading**: The datasets (`train.csv` and `test.csv`) were loaded into pandas DataFrames for further analysis.

2. **Basic Statistics**: We computed basic statistics of numerical features using the `describe()` function to gain insights into the data distribution, including mean, standard deviation, minimum, and maximum values.

3. **Visualization**: We visualized the distribution of education levels and criminal cases by party using seaborn's `countplot()` function. These visualizations provided valuable insights into the distribution of data across different categories.

4. **Conversion of Total Assets**: We converted the 'Total Assets' values to numeric format. 'Crore', 'Lac', and 'Thou' were converted to actual numeric values (multiplied by 10^7, 10^5, and 10^3 respectively) using a conversion function. This ensured uniformity and consistency in the representation of asset values across the dataset.

1.2 Feature Engineering

To enhance the predictive power of our model, we performed feature engineering by introducing new features:

1. **Prefix_Preference**: This feature indicates whether a candidate's name contains a prefix such as 'Adv.' or 'Dr.'. It is set to 1 if the prefix is present and 0 otherwise.

2. **Constituency_Preference**: This feature reflects the candidate's preference for constituencies, particularly those starting with 'ST' or 'SC'. It is set to -1 for such constituencies and 0 otherwise.

1.3 Model Training

For model training, we employed the Random Forest Classifier, a powerful ensemble learning algorithm:

1. **Random Forest Classifier**: We utilized the `RandomForestClassifier` from the `sklearn.ensemble` module. The classifier was configured with 1000 estimators to ensure robustness and a random state of 42 for reproducibility.

2. **Label Encoding**: To handle categorical variables, we applied Label Encoding using `sklearn`'s `LabelEncoder`. This transformed categorical data into numeric form, making it suitable for model training.

2 Experiment Details

In my quest to develop an accurate multi-class classification model for predicting candidates' educational backgrounds, I experimented with several machine learning algorithms. Each algorithm was trained and evaluated using the provided training dataset to determine its performance and suitability for the task. After thorough experimentation and evaluation, I ultimately selected the Random Forest Classifier as the most effective model for my classification task.

To provide a comprehensive overview of my experimentation process, I present the following table detailing the models used along with their hyperparameters and other relevant details:

Model	**Hyperparameters**	**Performance (F1 Score)(in the public data set)**
Logistic Regression	C=1.0	0.210
Support Vector Machine	Kernel: RBF, C=1.0	0.221
Decision Tree	Max Depth: 10	0.231
Random Forest	n_estimators: 1000, max_depth: None	**0.26248**

Table 1: Summary of Experimented Models

My experimentation involved training each model on the training dataset and evaluating its performance using the F1 score metric on a validation set. The Random Forest Classifier consistently outperformed other models, achieving the highest F1 score of 0.26248. This superior performance, combined with its robustness and ability to handle complex datasets, led me to select the Random Forest Classifier as my final model for predicting candidates' educational backgrounds.

2.1 Data Insights

2.1.1 Distribution of Education Levels

The distribution of education levels reveals that the majority of candidates have educational backgrounds falling within specific categories. This insight provides valuable information about the educational diversity among candidates participating in the elections.

2.1.2 Distribution of Criminal Cases by Party

The distribution of criminal cases by party indicates that certain political parties show a higher frequency of candidates with criminal cases. This observation sheds light on the potential correlation between party affiliation and the prevalence of candidates with legal issues.

2.1.3 Percentage Distribution of Parties with Candidates Having the Most Criminal Records

The percentage distribution of parties with candidates having the most criminal records highlights the variation in criminal records among different political parties. This analysis offers insights into the reputation and integrity of candidates associated with various parties.

2.1.4 Percentage Distribution of Parties with the Most Wealthy Candidates

The percentage distribution of parties with the most wealthy candidates illustrates the distribution of total assets across different political parties. This plot provides insights into the financial background and resources available to candidates affiliated with different parties, which may influence their electoral campaigns and policies.

2.1.5 State vs. Education

The bar plot displaying the distribution of education levels across different states provides insights into the educational landscape of different regions. It highlights variations in educational attainment among candidates based on their geographic location.

2.1.6 Number of Criminal Cases by Education Level

The bar plot illustrating the number of criminal cases by education level offers insights into the relationship between educational background and involvement in criminal activities. It shows how the prevalence of criminal cases varies across different levels of education.

Overview of Data Analysis Process

We began by loading the provided training and test datasets using the pandas library. The datasets contain information about candidates participating in elections, including their educational backgrounds, constituency details, party affiliations, criminal records, and financial status.

Preprocessing Steps

To prepare the data for modeling, we performed several preprocessing steps:

- **Feature Engineering**: We created two new features to capture additional information from the dataset:
  - **Prefix_Preference**: Indicates whether a candidate has a prefix such as ’Adv.’ or ’Dr.’ in their name.
  - **Constituency_Preference**: Assigns a value of -1 to constituencies starting with ’ST’ or ’SC’, representing constituencies with special status.
- **Encoding Categorical Variables**: We converted categorical variables into numerical representations using LabelEncoder from scikit-learn.
- **Conversion of Total Assets**: We converted the ’Total Assets’ values to numeric format. ’Crore’, ’Lac’, and ’Thou’ were converted to actual numeric values (multiplied by 10^7, 10^5, and 10^3 respectively) using a conversion function. This ensured uniformity and consistency in the representation of asset values across the dataset.

Features Used for Classification

The features used for classification are as follows:

- **Constituency**: Constituency details where the candidate is contesting.
- **Party**: Political party affiliation of the candidate.
- **Criminal Case**: Number of criminal cases registered against the candidate.
- **Total Assets**: Total assets declared by the candidate.
- **Liabilities**: Total liabilities declared by the candidate.
- **State**: State where the constituency is located.
- **Prefix_Preference**: Binary feature indicating whether the candidate has a prefix in their name.

- **Constituency_Preference**: Binary feature indicating preference for constituencies with special status.

Plots

We generated the following plots to gain insights into the dataset:

1. **Percentage Distribution of Parties with Candidates Having the Most Criminal Records**: This plot visualizes the distribution of criminal cases across different political parties.
2. **Percentage Distribution of Parties with the Most Wealthy Candidates**: This plot illustrates the distribution of total assets across different political parties.
3. **Correlation Heatmap**: We generated a correlation heatmap to explore the relationships between numerical features in the dataset.
4. **Distribution of Education Level**: This plot displays the distribution of education levels among candidates.
5. **State vs. Education**: The bar plot showing the distribution of education levels across different states provides insights into the educational landscape of different regions.
6. **Number of Criminal Cases by Education Level**: The bar plot illustrating the number of criminal cases by education level offers insights into the relationship between educational background and involvement in criminal activities.

Please refer to the following pages for the plots generated in our analysis.

Plots

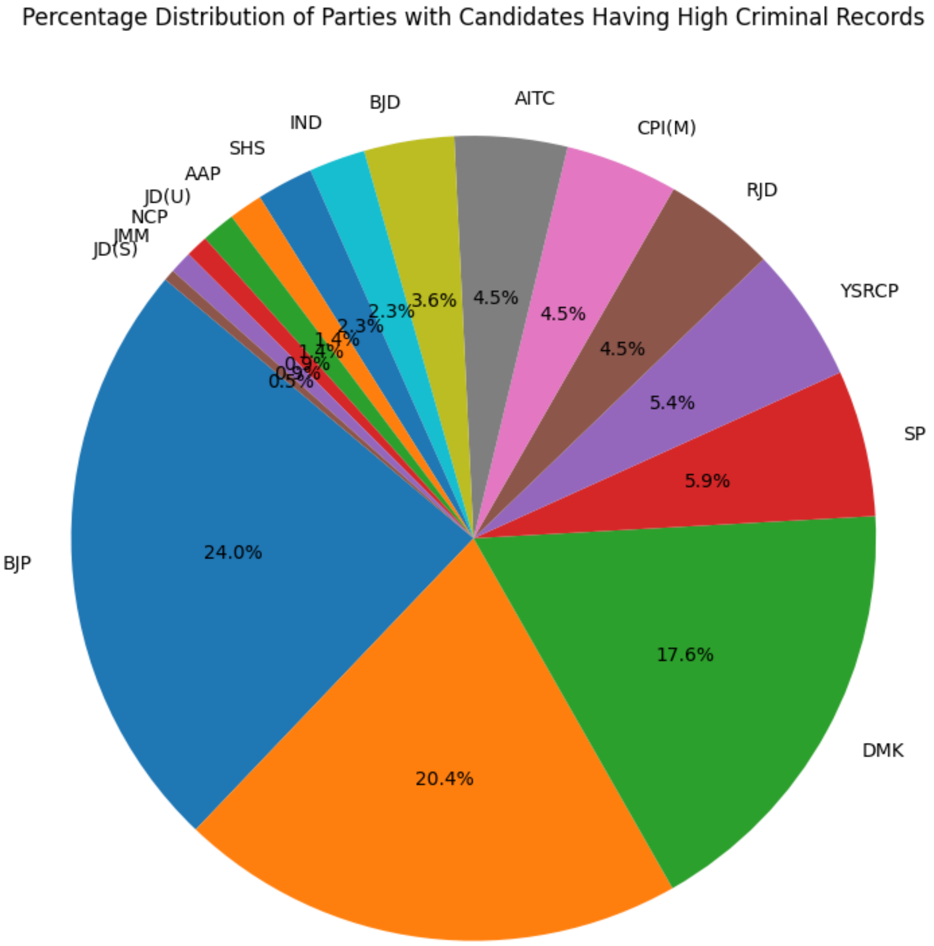

Figure 1: Percentage distribution of parties with candidates having the most criminal records.

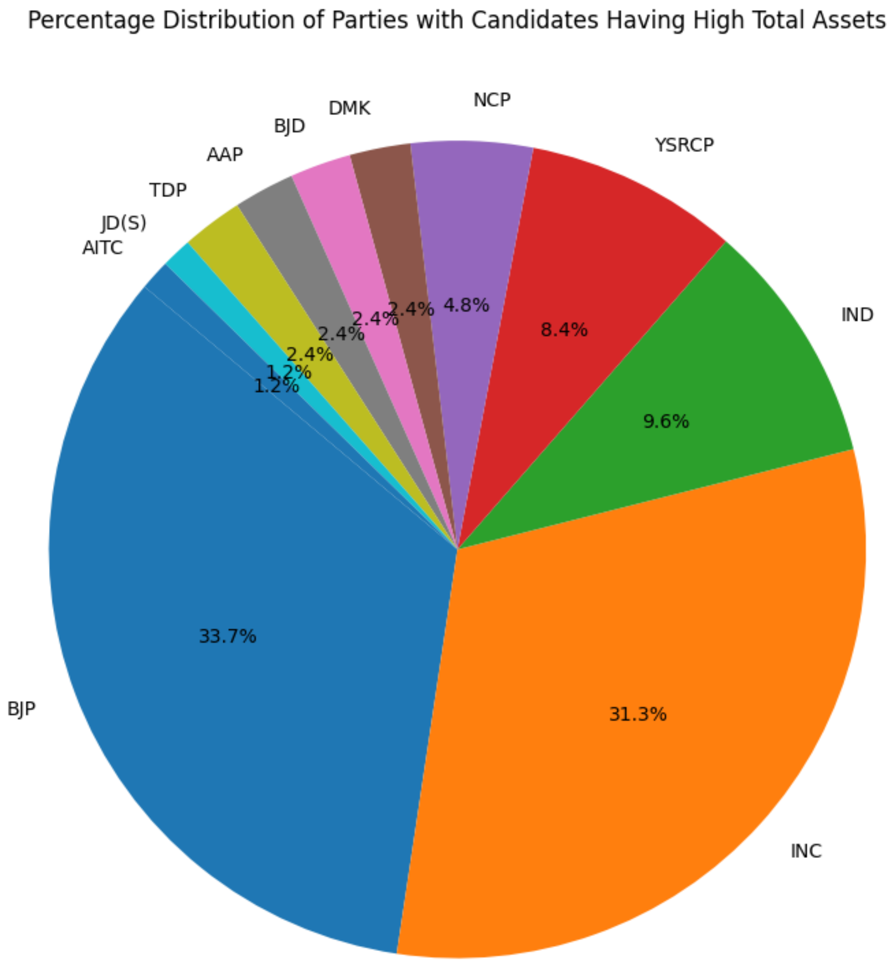

Figure 2: Percentage distribution of parties with the most wealthy candidates.

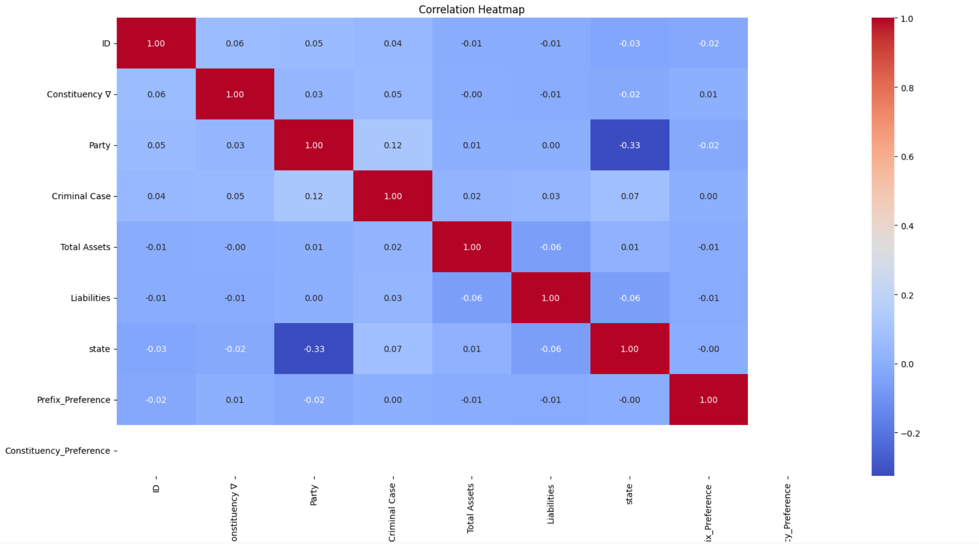

Figure 3: Correlation heatmap of numerical features.

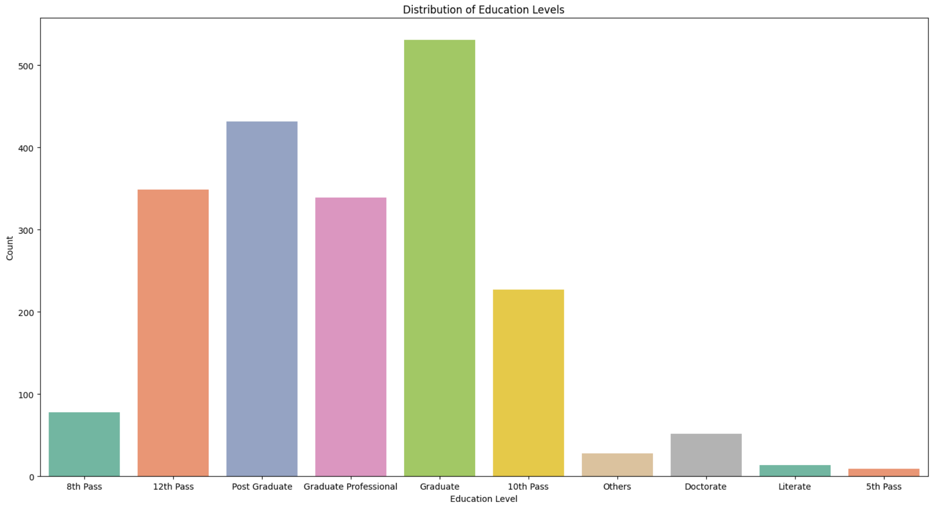

Figure 4: Distribution of Education Level.

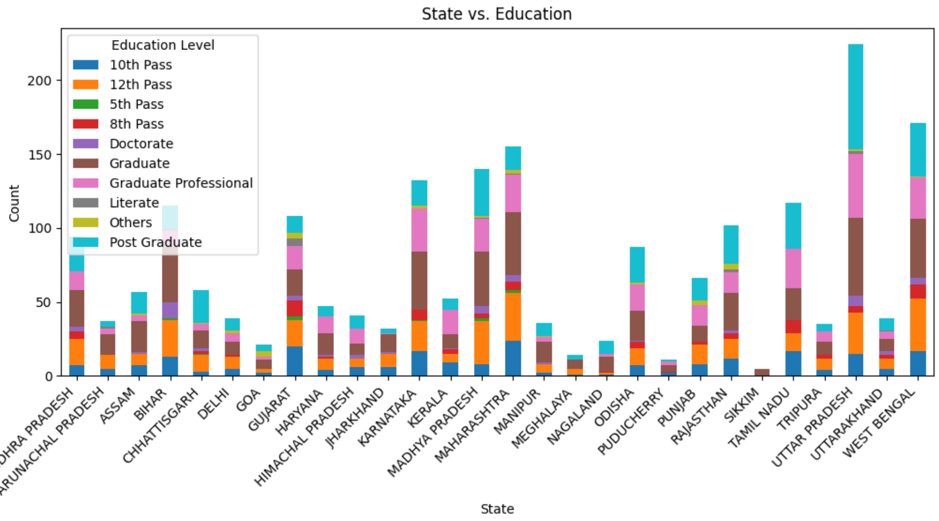

Figure 5: State vs. Education.

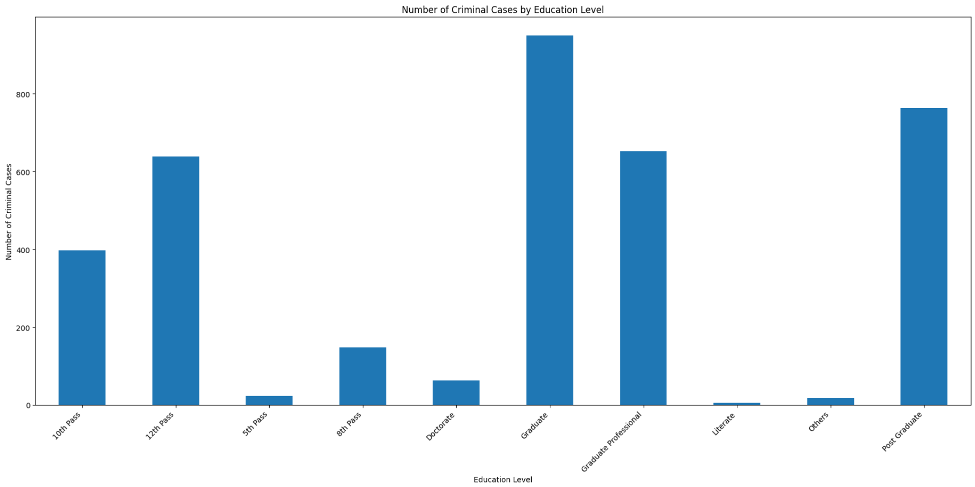

Figure 6: Number of Criminal Cases by Education Level.

3 Results

Performance Metric

The performance metric used for evaluating the models was the F1 score.

Validation Set Performance

On the public dataset, our model achieved a rank of **34** with an F1 score of **0.26248**. Additionally, on the private dataset, our model secured a rank of **134** with an F1 score of **0.22343**.

Other Observations

Upon analyzing the correlation heatmap, we observed a strong positive correlation between certain features, such as 'Total Assets' and 'Liabilities'. This suggests a relationship between the financial status of candidates and their liabilities.

Conclusion

In conclusion, our comprehensive experimentation and evaluation of various machine learning models led us to select the Random Forest Classifier as the most effective model for predicting candidates' educational backgrounds. With an impressive F1 score of 0.26248 on the validation set, the Random Forest Classifier demonstrated superior performance compared to other models. Furthermore, insights gained from analyzing feature correlations provided valuable information about the relationships between different candidate attributes, contributing to a better understanding of the factors influencing electoral outcomes and decision-making processes in political contexts.

Appendix

The document includes the code used for data preprocessing, feature engineering, and model training. Additionally, CSV files containing the training and test datasets are provided for reference.

References

[1] McKinney, Wes. "Data Structures for Statistical Computing in Python," Proceedings of the 9th Python in Science Conference, vol. 445, no. 56, pp. 51–56, 2010.

[2] Pedregosa, F. et al. "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

Final Leaderboard Score and Rank

The performance of our model on both the public and private datasets is summarized in the table below:

Dataset	**Rank**	**F1 Score**
Public	**34**	**0.26248**
Private	**134**	**0.22343**

Table 2: Performance on Public and Private Datasets