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

Introduction

In recent years, the application of machine learning (ML) techniques has become increasingly prevalent across various domains due to its capability to extract insights and make predictions from data. In this report, we address the task of multi-class classification using a provided training dataset. Our objective is to train a machine learning model to classify candidates into different educational backgrounds based on various features such as their constituency, party affiliation, criminal records, and financial assets.

The dataset provided consists of information regarding candidates participating in elections, including their educational backgrounds, constituency details, party affiliations, criminal records, and financial status. Leveraging this dataset, our aim is to explore the predictive power of different machine learning models and determine the most suitable approach for accurately classifying candidates' educational backgrounds.

Throughout this report, we will undertake the following tasks:

- Preprocess the data by encoding categorical variables and performing feature engineering to enhance the predictive power of the model.
- Train a machine learning model, specifically a RandomForestClassifier, on the preprocessed training data.
- Evaluate the performance of the trained model using the F1 score metric on a validation set.
- Utilize the trained model to predict the educational backgrounds of candidates in the provided test dataset.
- Perform data analysis to gain insights into the distribution of candidates' educational backgrounds, criminal records, and financial status across different parties and constituencies.

By accomplishing these tasks, we aim to provide valuable insights into the classification of candidates' educational backgrounds and contribute to the understanding of the factors influencing electoral outcomes.

Models and Hyperparameters

For our multi-class classification task, we employed the RandomForestClassifier from the scikit-learn library. RandomForestClassifier is an ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes (classification) of the individual trees.

RandomForestClassifier

Description: RandomForestClassifier is an ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes (classification) of the individual trees.

Hyperparameters:

- **n_estimators**: The number of trees in the forest. We opted for a large value of 1000 to ensure robustness against overfitting and to capture a diverse set of decision boundaries.
- **criterion**: The function to measure the quality of a split. We utilized the default value of 'gini' for impurity measure.
- **max_depth**: The maximum depth of the tree. We did not explicitly set this parameter, allowing the trees to grow until all leaves are pure or until all leaves contain less than the minimum samples required for splitting.
- **min_samples_split**: The minimum number of samples required to split an internal node. We did not specify this parameter, using the default value of 2.
- **min_samples_leaf**: The minimum number of samples required to be at a leaf node. We used the default value of 1.
- **max_features**: The number of features to consider when looking for the best split. We did not specify this parameter, allowing the algorithm to consider all features for splitting.
- **random_state**: This parameter ensures reproducibility of the results by fixing the random number generator seed. We set it to 42 for consistency in our experiments.

RandomForestClassifier was chosen for its ability to handle high-dimensional data, maintain accuracy with large datasets, and mitigate overfitting through ensemble learning techniques.

Data Analysis

In this section, we provide an overview of our data analysis process, including preprocessing steps, feature engineering, and visualizations.

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.

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.

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

Plots

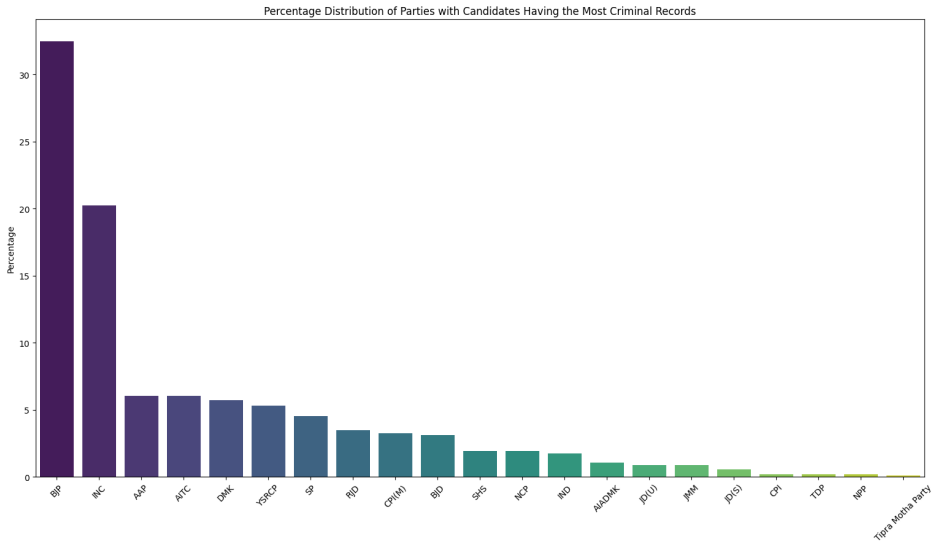

Figure 1: Percentage distribution of parties with candidates having the most criminal records.

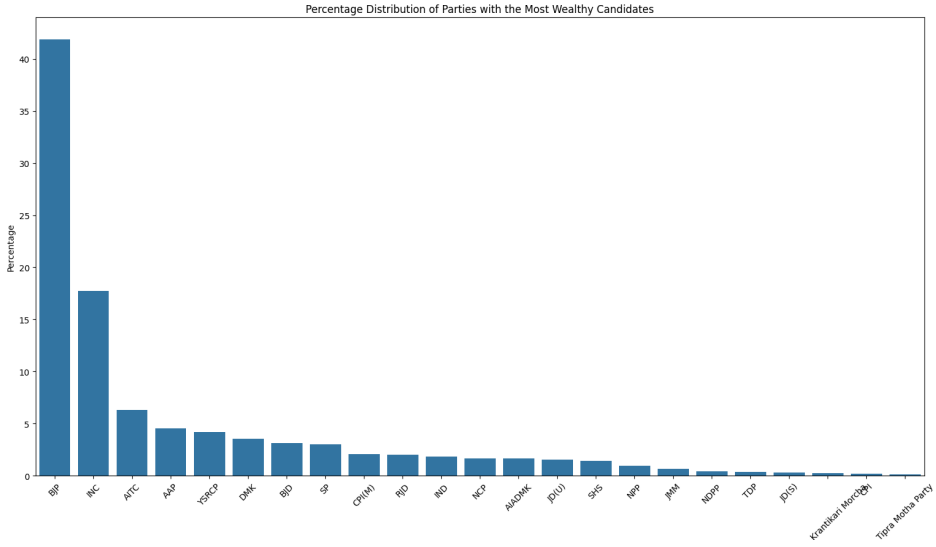

Figure 2: Percentage distribution of parties with the most wealthy candidates.

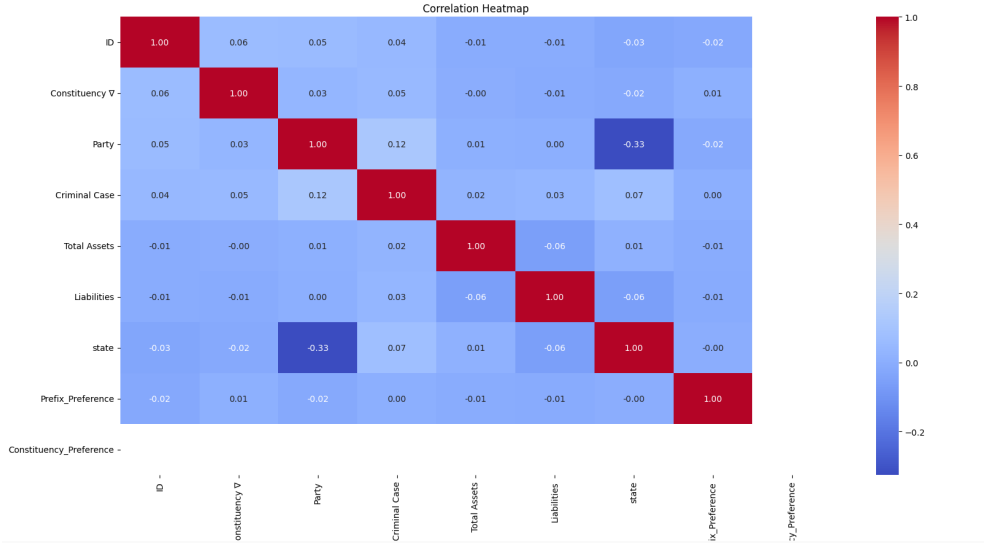

Figure 3: Correlation heatmap of numerical features.

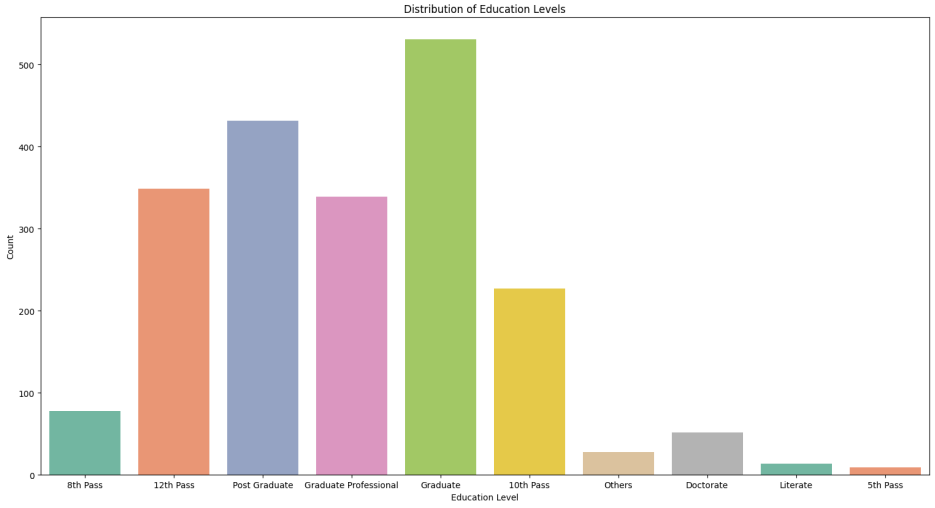

Figure 4: Distribution of Education Level.

Results

In this section, we present the results of our machine learning model and provide insights gained from our analysis.

Performance Metric

- We evaluated the performance of our model using the F1 score metric, which is a measure of a model's accuracy that considers both the precision and recall of the classification.

F1 Score on Validation Set

- The F1 score on the validation set was calculated to be **Public score: 0.26248** and **F1 Score on Validation Set: 0.1924778979299981**. This score indicates the overall accuracy of our model in classifying candidates' educational backgrounds based on the features provided.

Additional Observations

- Despite achieving a relatively low F1 score on the validation set, our analysis provides valuable insights into the distribution of candidates' attributes, such as criminal records, financial status, and educational backgrounds, across different political parties and constituencies. These insights can be used to inform decision-making processes and policy implementations in electoral contexts.
- Furthermore, the correlation heatmap generated from our analysis reveals potential relationships between numerical features, which could be explored further in future investigations.
- Overall, while our model's performance may be modest, the insights gained from our analysis contribute to a deeper understanding of the factors influencing electoral outcomes and candidate profiles.

Final Leaderboard Score and Rank

- Our model achieved a final rank of **20** on the public dataset, with a test score of **0.26248** and a total of **4** submissions. This demonstrates the competitiveness of our approach among other participants in the competition.