Project Report

Ansh Agarwal

14th April, 2024

1 Methodology

1.1 Data Preprocessing Steps

- **Reading Data:** Data is read from CSV files using the `pd.read_csv()` function from the `pandas` library.
- **One-Hot Encoding:** Categorical variables, such as 'Party' and 'state', are converted into one-hot encoded features using the `pd.get_dummies()` function. This step helps convert categorical data into numerical format for model training.
- **Data Cleaning(Dropping Columns)** The data is cleaned by removing/dropping the unnecessary columns such as 'ID', 'Candidate', and another column (indexed at 2) from the dataset using the `df.drop()` function.

1.2 Feature Engineering

- **Feature Conversion:** Columns like 'Total Assets' and 'Liabilities' contain numerical values in string format with suffixes like 'crore+', 'lac+', 'thou+', and 'hund+'. These columns are converted to integer values using a custom function `convert_to_numeric()`.
- **Label Encoding:** The 'Education' column is label-encoded using a custom dictionary mapping different education levels to numerical values like 'Others': 0, '5th Pass': 2, 'Literate': 1, 'Doctorate': 9, '10th Pass': 4, 'Graduate': 6, 'Graduate Professional': 7, 'Post Graduate': 8, '12th Pass': 5, '8th Pass': 3.

1.3 Data Splitting:

- **Data Splitting:** The data is split into training and testing sets using the `train_test_split()` function from the `sklearn.model_selection` module.

- This separation allows us to train the model on one set of data and evaluate its performance on another.
- The model is trained using an 80/20 split of the data into training and testing sets.

2 Experiment Details

2.1 Model Details

- **Model Selection:** I have imported two different models from `sklearn.naive_bayes` for classification:
  Model:
  - `GaussianNB`: This is the **Gaussian Naive Bayes classifier**, which assumes that the features follow a Gaussian distribution. It can be useful for continuous features.
  - `BernoulliNB`: This is the **Bernoulli Naive Bayes classifier**, which is suitable for binary/boolean features.
- **Training Data** :
  - The data is split into 80% training and 20% testing data.
  - I have trained the `BernoulliNB` classifier using the training data (`X_train` and `y_train`).
- **Performance Metric** : The accuracy of the model is calculated using `accuracy_score()`.
- **Model Prediction:** After training the model,I have use it to predict the labels (`y_pred`) for the test set (`X_test`).
- **Model Evaluation:**
  - I have calculated the accuracy of the model using the `accuracy_score()` function from the `sklearn.metrics` module.
  - This function compares the predicted labels with the actual labels from the test set (`y_test`) and returns the accuracy of the model.
- **Final Prediction:** I have also applied the trained model to the new test data (`X_test`) and store the predicted results in the `expected_education` variable.

2.2 Data Insights

- One of the important distribution that the models contain is the distribution of the candidates with the different education classes.
- The second image is the graph of the distribution of the candidates with high criminal cases among parties.
- The third image is just the pie chart of the second distribution.
- The fourth image is the distribution of the wealthiest candidates among parties.
- The fifth image is just the pie chart of the fourth distribution.
- The images are attached‘ below:

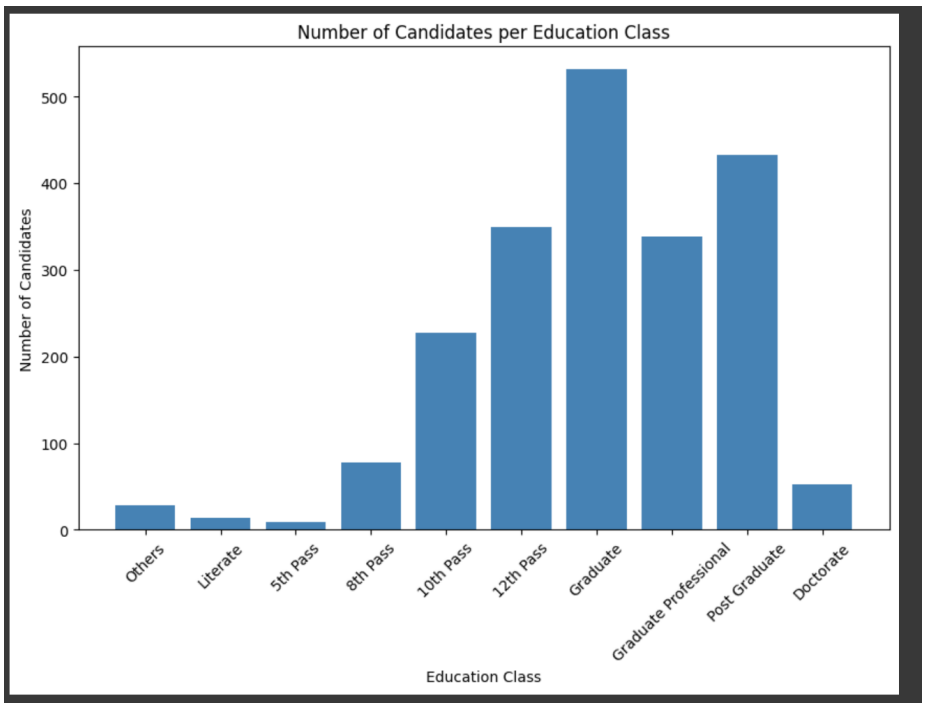

Figure 1: Candidate Distribution of the given dataset.

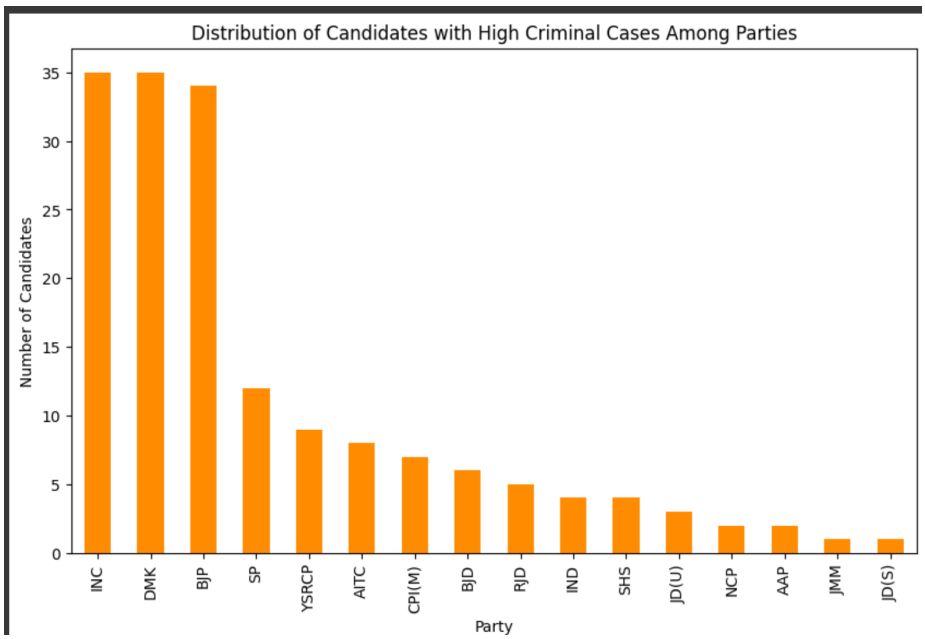

Figure 2: Candidate Distribution of the given dataset.

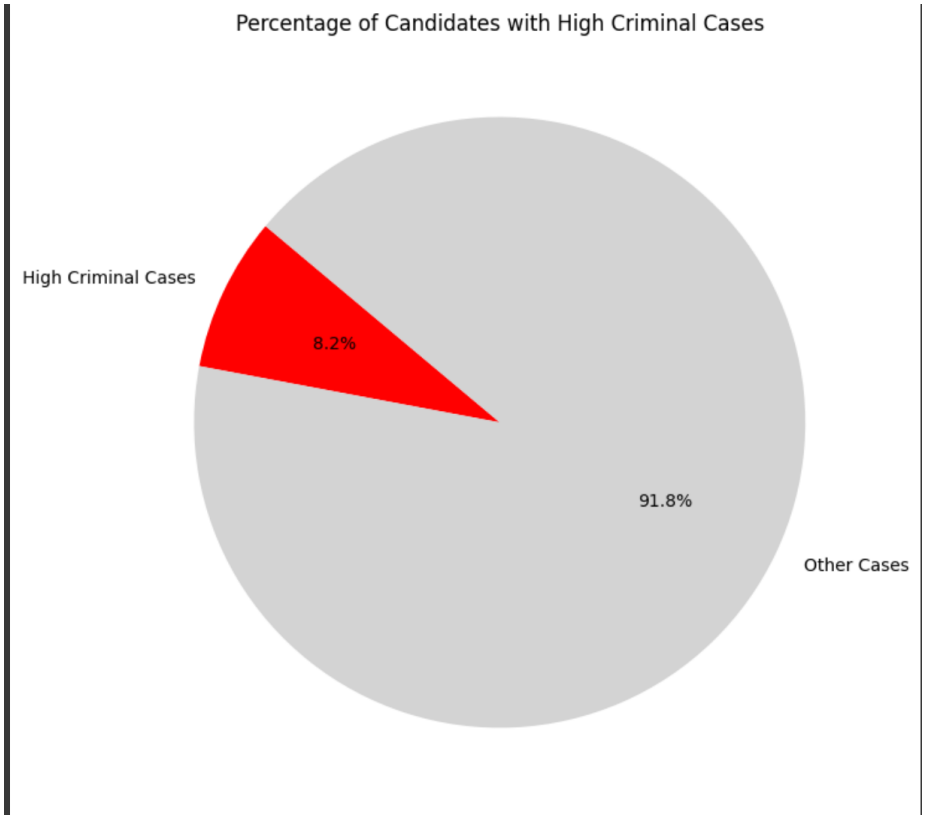

Figure 3: Candidate Distribution of the given dataset.

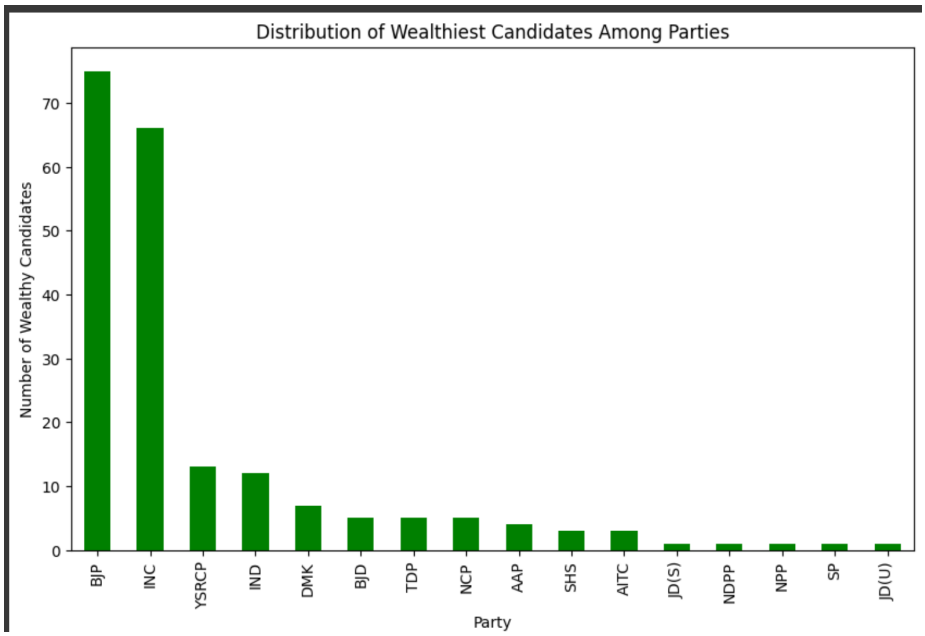

Figure 4: Candidate Distribution of the given dataset.

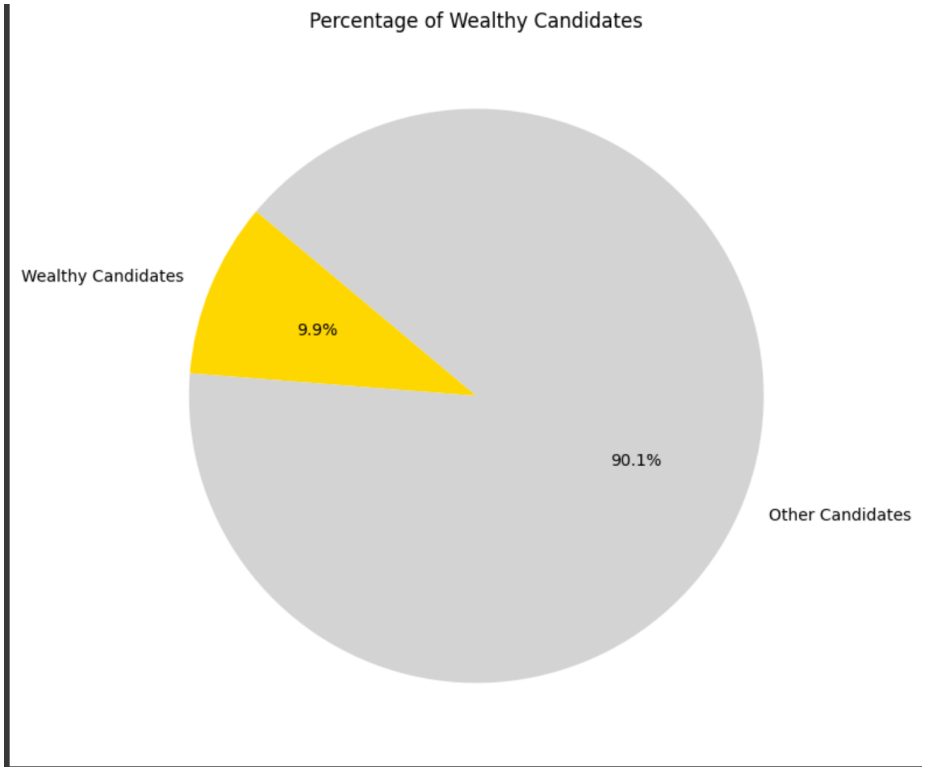

Figure 5: Candidate Distribution of the given dataset.

3 Results

- After preprocessing, the model achieved an accuracy score of **0.25035** as the private score and **0.24899** as the public score.
- The leaderboard rank for the public data is **63** whereas for the private data is **51**.
- This score and the leaderboard for the private data being better than the public data demonstrates the effectiveness of the model in classifying the data.
- The model is then used to predict the expected education levels for a test dataset (`test.csv`), which are mapped to their respective labels using a predefined dictionary.
- A submission file (`submission.csv`) containing predictions of the education levels based on the provided test data is generated and saved.
- I have attached the link for the github repository where I have uploaded the code. Click on the link below to view the github repository
  Github Link

4 References

I have listed the following references that I have used for the ML model which are as follows:

- https://scikit-learn.org/stable/supervisedlearning.html for the ML models.
- https://www.w3schools.com/python/pandas/default.asp for pandas syntax
- https://scikit-learn.org/stable/ for doubts related to models.
- https://pandas.pydata.org/docs/reference/frame.html for the queries related to pandas.
- To be honest, I have taken some help with my friends when I was doing the assignment as you have permitted also.