Python Assignment Record

Srishti Chandra
Code is Uploaded here

April 14, 2024

1 Methodology

1.1 Data Preprocessing Steps

- **Converting Total Assets and Liabilities**: The code converts the 'Total Assets' and 'Liabilities' columns from string format (which includes units such as Crore+, Lac+, Thou+, etc.) to float values. This is crucial for numerical analysis as the values need to be in a consistent numeric format for machine learning algorithms to process.
- **Encoding Categorical Variables**: The code uses LabelEncoder from sklearn. preprocessing to encode categorical variables like 'Party', 'state', 'Candidate', and 'Constituency ' into numeric form. This is important because most machine learning algorithms cannot directly handle categorical data, so encoding them into numeric form enables the algorithm to operate on them.
- **Binary Feature Creation**: The code creates binary features like 'SC', 'ST', 'Doctor', and 'Advocate' based on certain conditions present in the data. This converts categorical information into a format that is more suitable for machine learning algorithms to understand.

1.2 Feature Engineering

- **Creating Binary Features**: Binary features like 'SC', 'ST', 'Doctor', and 'Advocate' are created based on certain conditions present in the data. This can help capture specific characteristics or attributes of the candidates which might influence their education level.

1.3 Identifying Outliers

- **Outlier Detection**: The function `detect_outliers()` identifies outliers in the dataset using the Interquartile Range (IQR) method. Outliers are data points that significantly differ from the rest of the data in a dataset.
- **Outlier Mitigation**: The function `mitigate_outliers()` replaces the identified outliers with the mode of the respective column. This approach helps in dealing with outliers by replacing them with more representative values, ensuring that they don't unduly influence the model.

These steps are important because they ensure that the data is appropriately processed and prepared for training machine learning models. Converting data into suitable formats, encoding categorical variables, and handling outliers are crucial for building robust and accurate predictive models. Additionally, feature engineering helps in creating new features that may capture important patterns or relationships in the data, potentially improving the model's performance.

2 Experiment Details

- **Final Model**: RandomForestClassifier
- **Hyperparameters**:

- `n-estimators`: 420
- `max-depth`: 50
- `min-samples-split`: 11
- `random-state`: 42

- **Explanation**:
  - `n-estimators`: The number of trees in the forest. A higher number of trees can improve the model's performance, and 420 is chosen to ensure a sufficiently large forest without making the model overfitting.
  - `max-depth`: The maximum depth of the trees. A depth of 50 allows the trees to capture complex patterns without overfitting to the training data.
  - `min-samples-split`: The minimum number of samples required to split an internal node. Setting this to 11 ensures that each split requires a sufficient number of samples, helping to prevent overfitting.
  - `random-state`: Ensures reproducibility of the results by fixing the random state.

2.1 Alternative Models and Possible Reasons for Underperformance

1. **Logistic Regression**:
   - **Possible Reason**: Logistic Regression did not perform well since the relationship between the independent variables and the target variable is non-linear, as it assumes a linear relationship between the independent and dependent variables.
2. **Gradient Boosting Machines**:
   - **Possible Reason**: Gradient Boosting Machines did not perform well since the dataset is noisy or contains a large number of outliers, as they are sensitive to outliers and can overfit to noisy data. But on removing the outlines its score was increasing.
3. **Naive Bayes**:
   - **Possible Reason**: Naive Bayes did not perform well since the assumption of independence between features is violated, as it assumes that features are conditionally independent given the class, which might not hold true in some datasets.
4. **K-Nearest Neighbors (KNN)**:
   - **Possible Reason**: KNN did not perform very well but could have if hyperparameters were adjusted more according to the dataset.
5. **Decision Trees**:
   - **Possible Reason**: Decision Trees did not perform well since the dataset is highly imbalanced, as they tend to favor majority classes and can struggle to accurately classify minority classes.

These models could have performed a lot better if I had invested more time in hyperparameterisation.

2.2 Data Insights

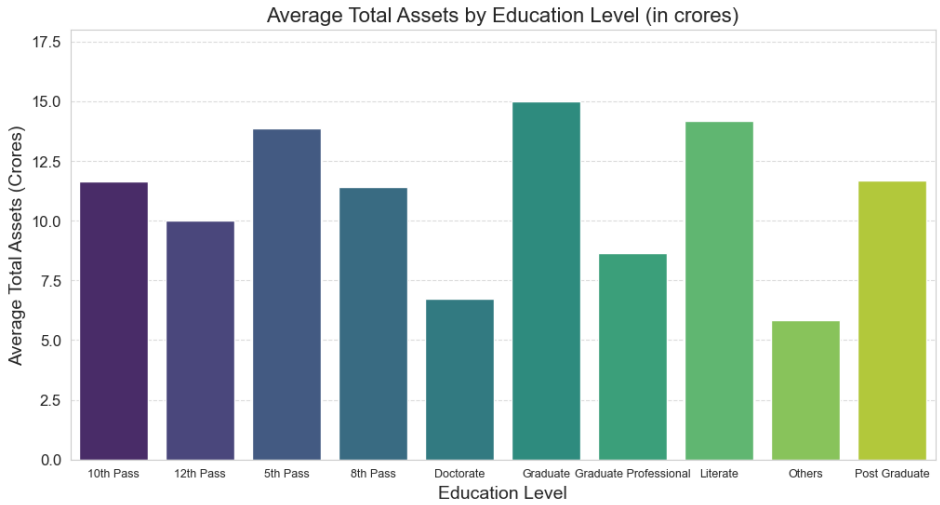

Figure 1: Assets Vs Education

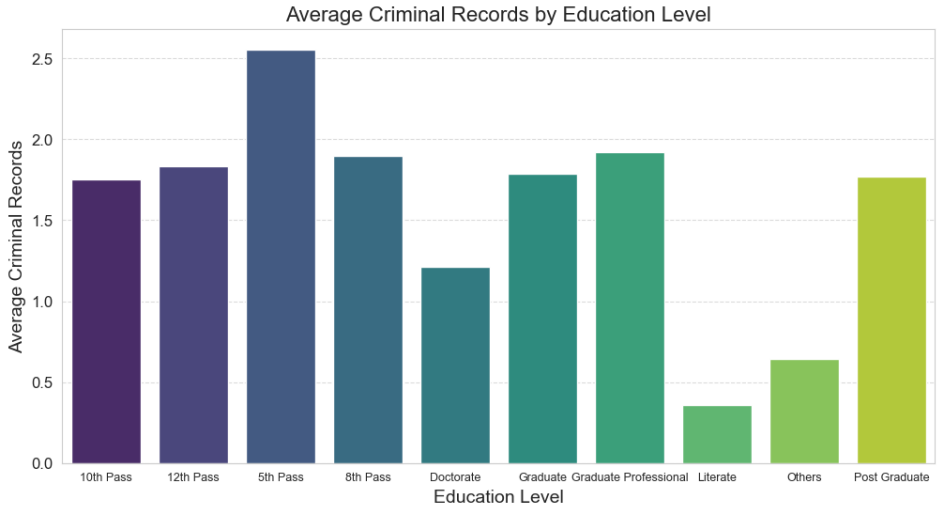

Figure 2: Crinimal Record Vs Education

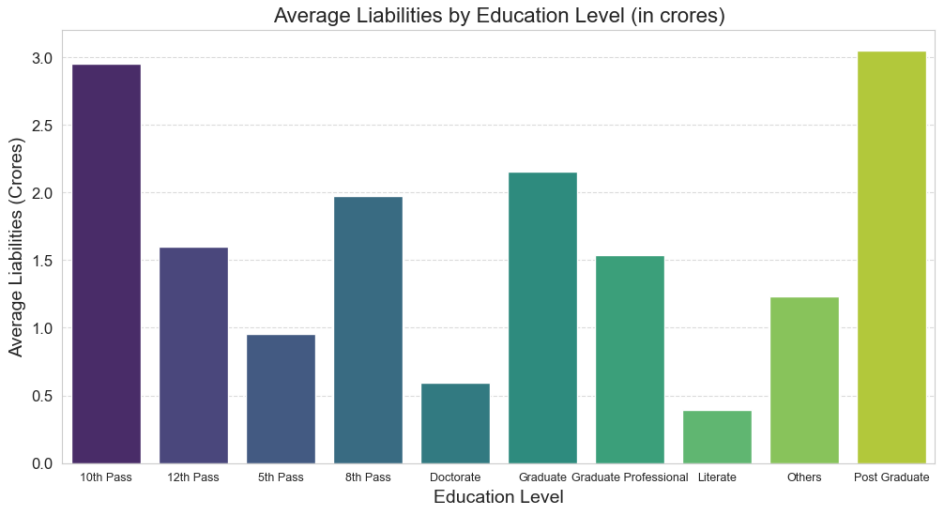

Figure 3: Liability Vs Education

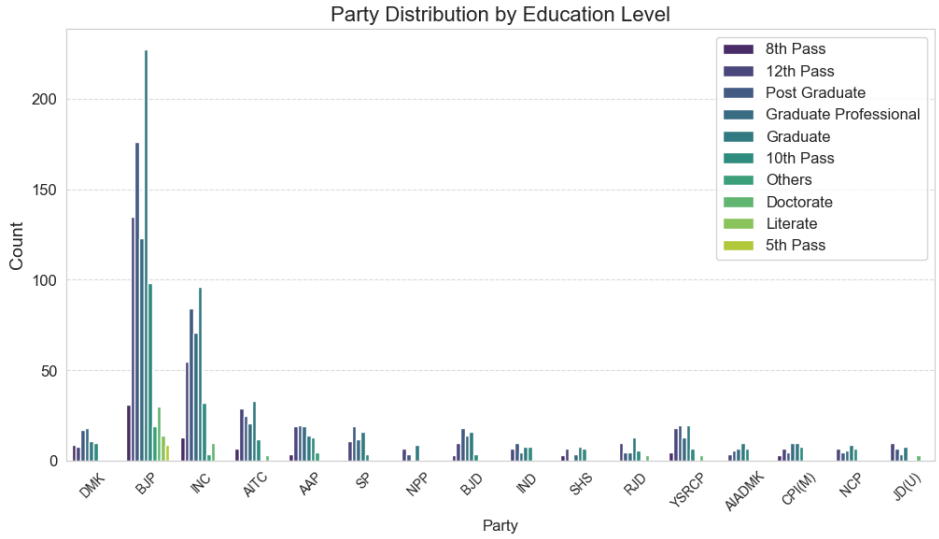

Figure 4: Party and Education

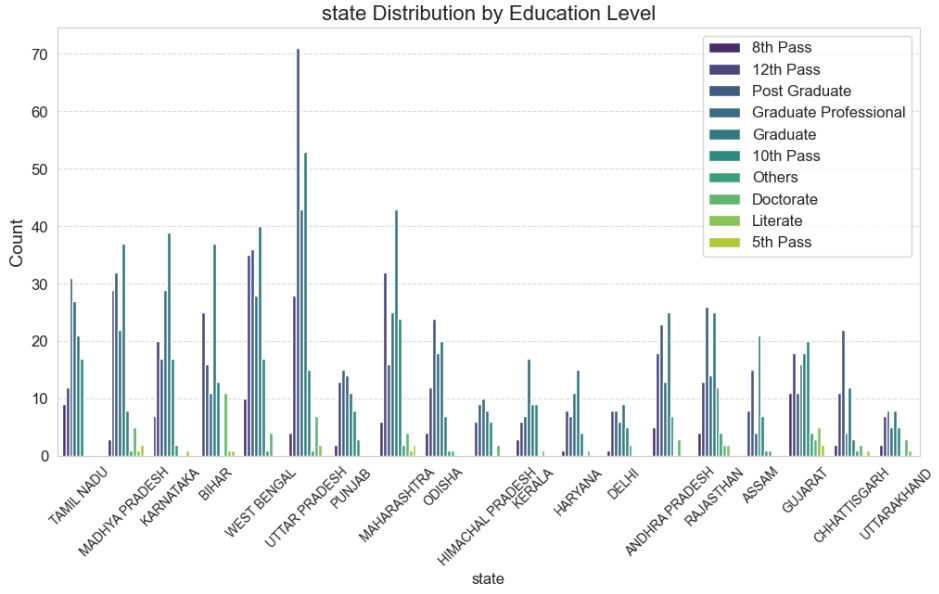

Figure 5: State and Education

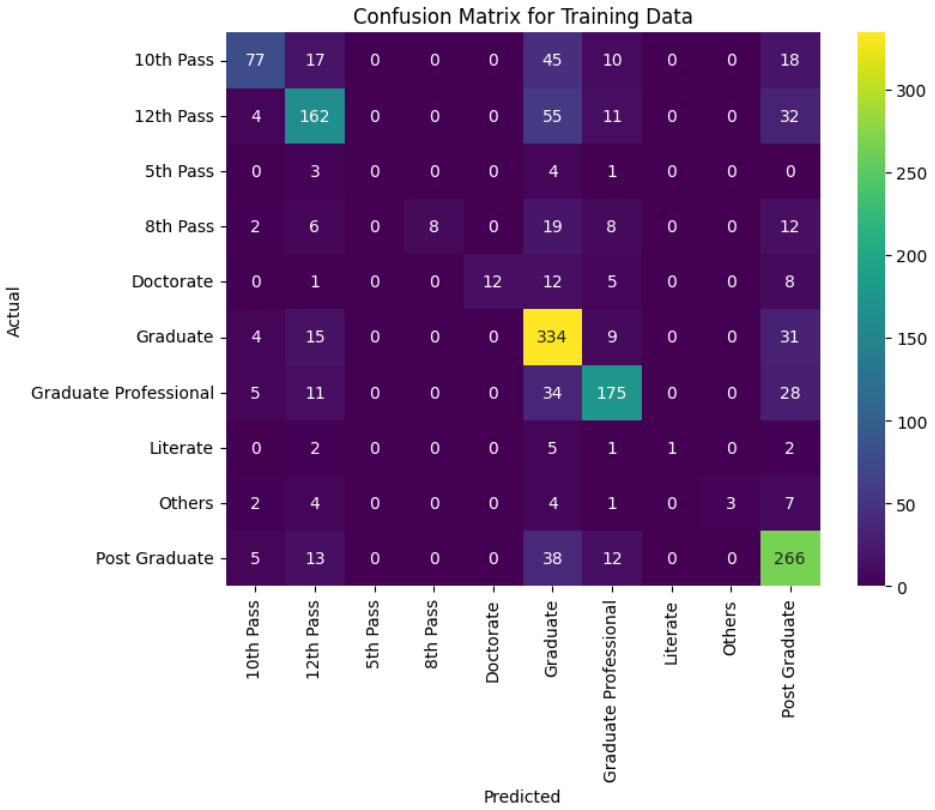

Figure 6: Confusion Matrix for training data

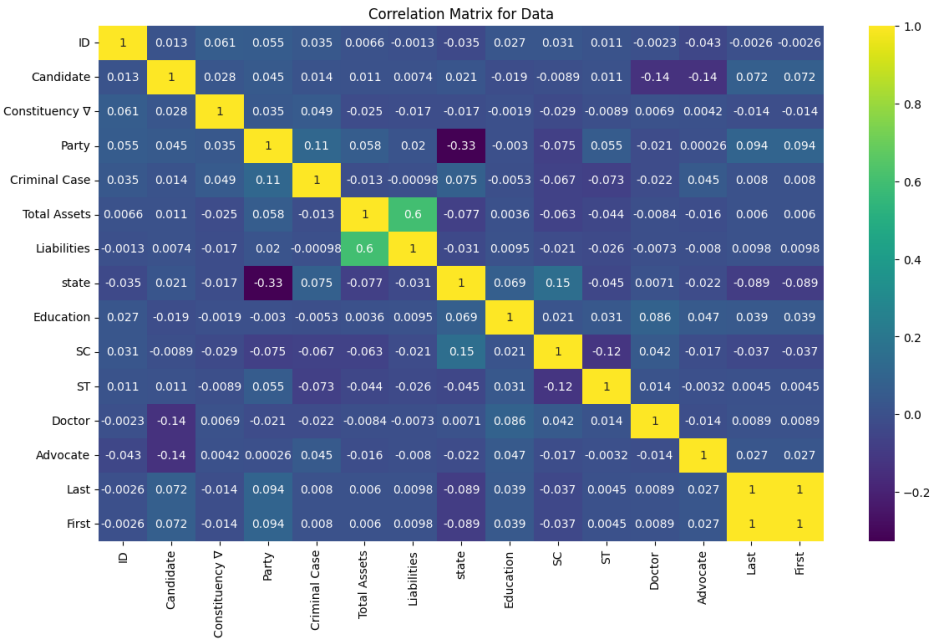

Figure 7: Correlation Matrix(First and Last stand for first name and last name)

2.3 Relevent Insights found

First name and Last name are not signification in predicting the education level of ministers .

Liabilities also do not play a major role in predicting the education level.

3 Results

The result for different dataset are as follows:

Best F1 score for public test data: 0.25103 Rank:59

Best F1 score for private test data: 0.25844 Rank:(Sir I had already mailed that I had selected two of the scores during my initial submissions but when forgot to update those selected when my F1 score had highly improved therefore my rank is very low 206) I have attached the screen shots to show my highest F1 score which was not considered because of this mistake .

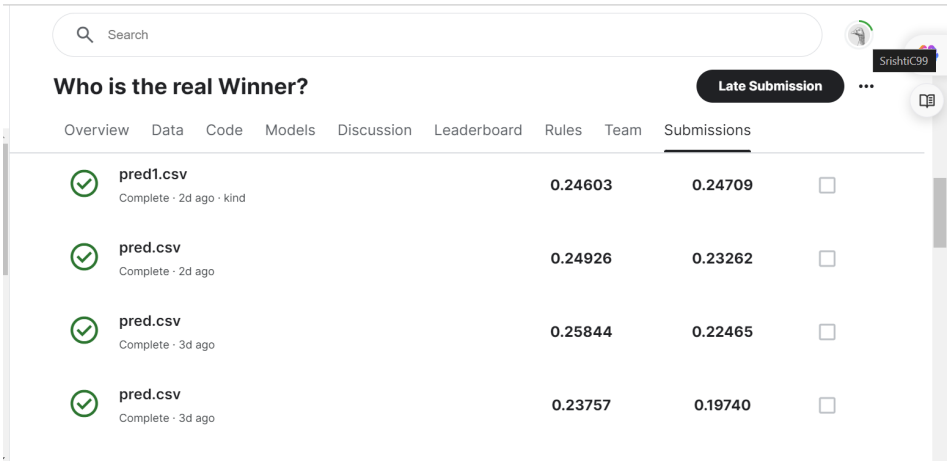

Figure 8: Screen Shot of my highest F1 score

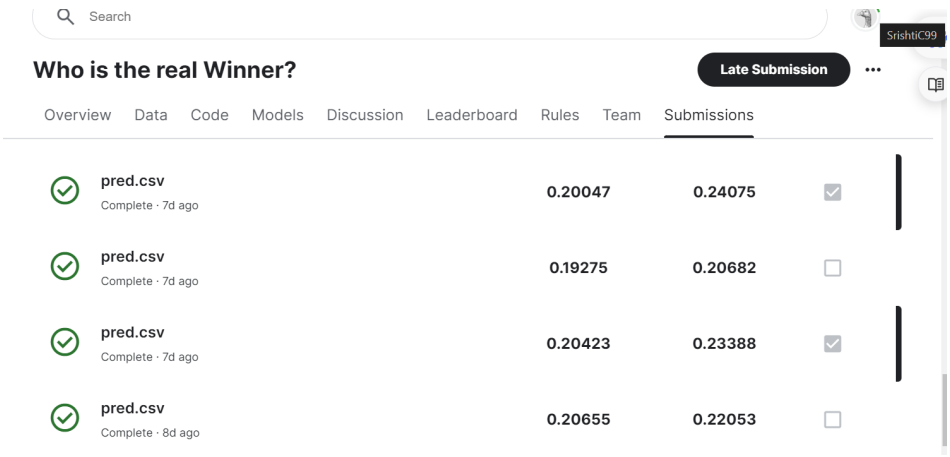

Figure 9: Screen Shot of submissions I had selected in the initial days of the competition which I forgot to remove

4 References

https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
https://scikit-learn.org/stable/modules/multiclass.html