PYTHON ASSIGNMENT(CS253)

1. Methodology:-

1.Data Preprocessing:-

**Converting Total Assets and Liabilities to Numerical Values**:

* + Initially, the 'Total Assets' and 'Liabilities' columns are read as strings (object datatype), containing additional text like 'Crore', 'Lac', etc.
  + To perform numerical operations, the text characters are removed using string manipulation (**str.replace()** function).
  + After cleaning, the columns are converted to numeric format to facilitate mathematical operations and analysis.
  + This ensures that the values are properly formatted for further processing without any text characters.

**Replacing Zero Entries in 'Total Assets' with Mean Value**:

* + Zero values in 'Total Assets' may be unrealistic or indicative of missing data rather than actual zero assets. Therefore, replacing them with meaningful values can prevent bias in the analysis..
  + Replacing them with meaningful values can prevent bias in subsequent analysis or modeling.
  + The mean of non-zero assets is calculated to represent a typical value for assets.
  + All zero entries in the 'Total Assets' column are replaced with this mean value using the **replace()** function.

2.Feature engineering:-  
 **Extracting information like candidate profession and constituency type**

* **extract\_constituency\_type** function identifies the type of constituency (SC, ST, or unreserved) based on the presence of specific keywords in the 'Constituency ∇' column.
* **extract\_extra\_info** function extracts information about the profession from the 'Candidate' column, categorizing candidates as Doctor (Dr.), Advocate (Adv.)
* These functions are applied to the respective columns using the **apply()** method, and the extracted information is stored in new columns 'Constituency\_Type' and 'Extra' in the DataFrame.

**One Hot encoding and label encoding**

* Categorical variables such as 'state' and 'Party' are one-hot encoded using scikit-learn's **OneHotEncoder**. This converts categorical variables into a numerical format suitable for machine learning algorithms.
* The 'Education' column is label encoded using scikit-learn's **LabelEncoder**, which assigns a unique numeric label to each category in the 'Education' column.

3. Dimensionality Reduction Techniques:-

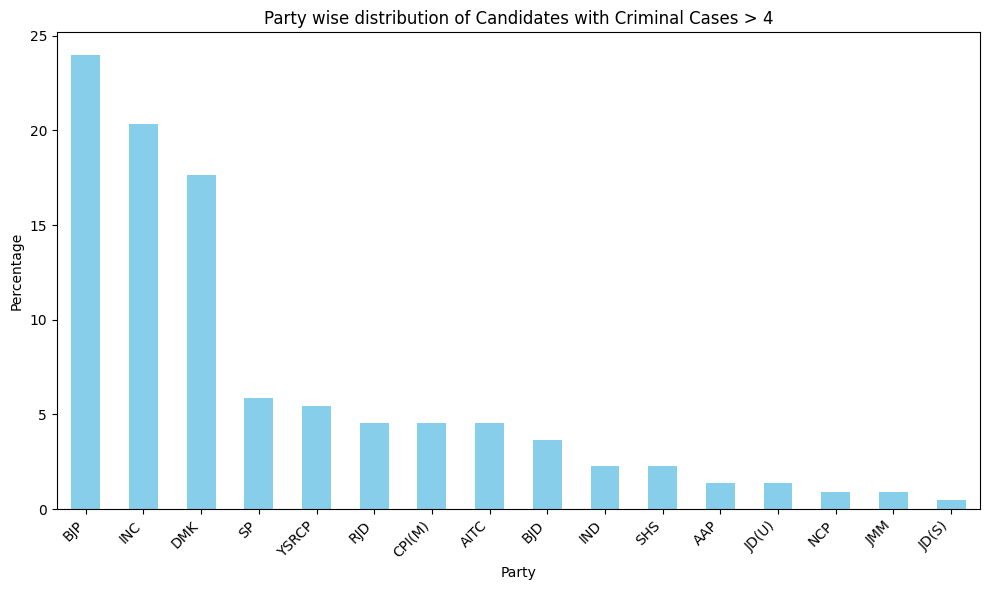
* Features such as ID, constituency, and candidate name may not directly contribute to predicting a candidate's education qualification.
* Including these features in the model could introduce noise or unnecessary complexity, which may hinder the model's performance.
* By removing these uninformative features, the model can focus on the features that are more directly related to the target variable (education qualification), potentially leading to better predictive accuracy.

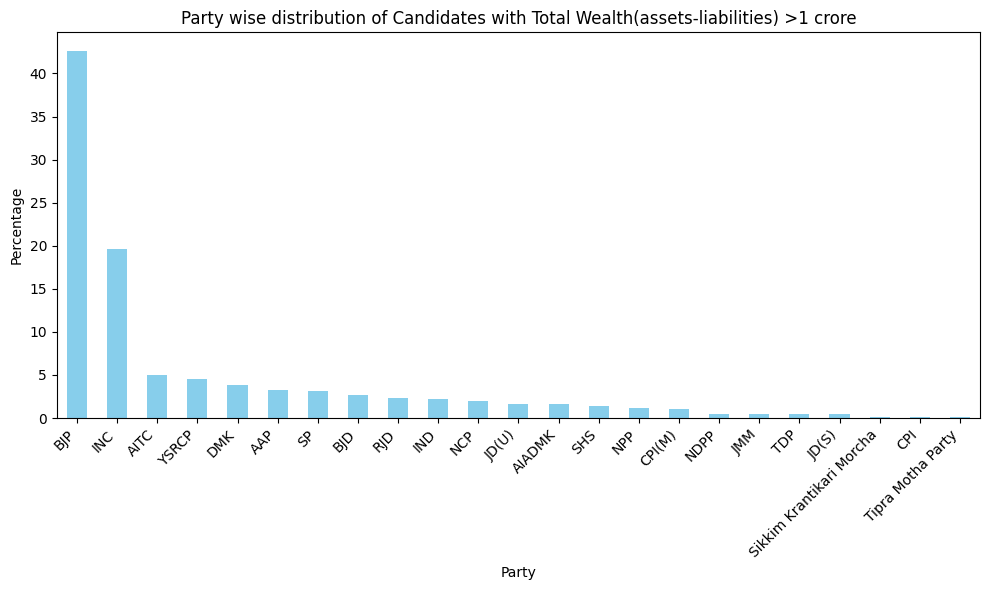
2.Experimental Details:-

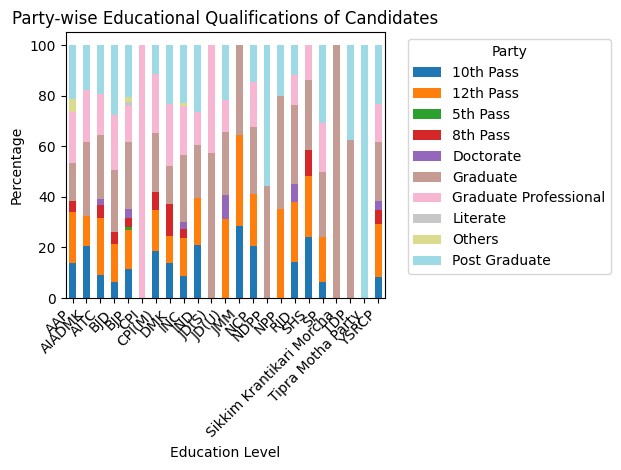
Models tried and tested:-

|  |  |  |
| --- | --- | --- |
| ML Model tested | Hyperparameters tuned | F1 score |
| Random forest | n\_estimators=150,random\_state=50 | 0.22422 |
| Bernoulli Naive Bayes | alpha=0.40, binarize=0.0 | 0.26505 |
| KNN | K-value=5 | 0.19426 |
| Support vector machine | Kernel=rbf, gamma=0.5, C=1.0 | 0.21582 |

GRAPHS:-



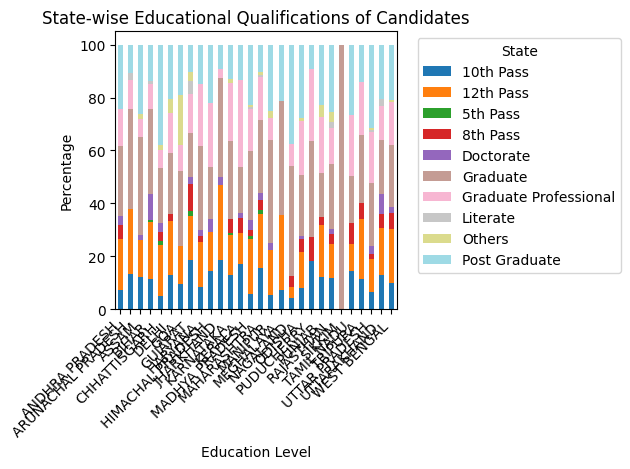




The party-wise plot showing different percentages of educational qualifications highlights the importance of using party as a parameter to predict educational qualifications:

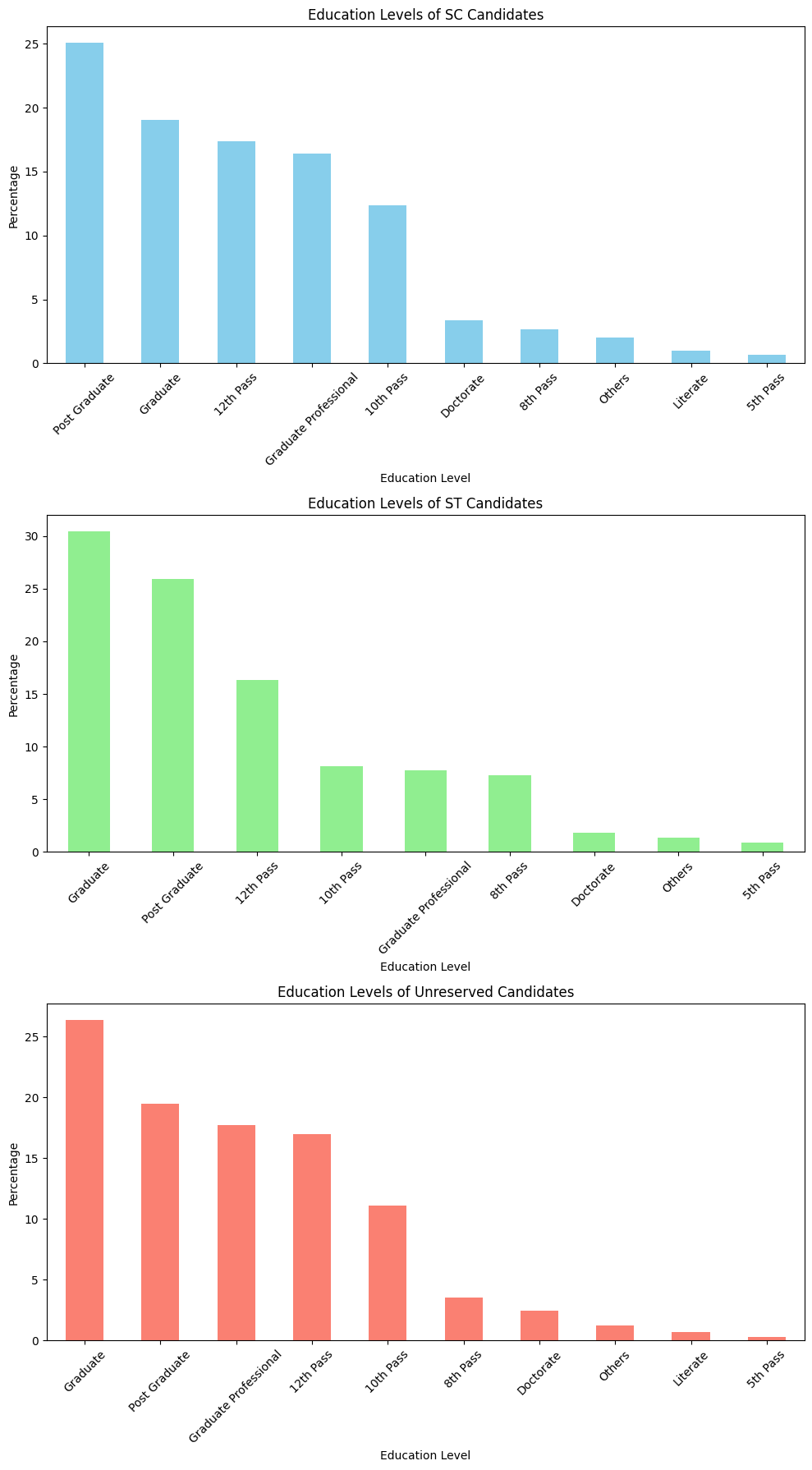
1. **Variation in Education Levels Across Parties:** The plot illustrates that different political parties attract candidates with varying educational backgrounds. For example, some parties may have a higher percentage of candidates with advanced degrees, while others may have more candidates with lower educational qualifications.

2. **Predictive Value of Party Affiliation:** By incorporating party affiliation as a predictor variable in the model, we can capture the influence of party-specific factors on candidates' educational qualifications. This can improve the accuracy of the prediction model by accounting for the varying distributions of education levels across different parties.



In a state-wise analysis, the variation in educational qualifications across different states provides insight into the need to use state as a parameter to predict educational qualifications.

1. **Regional Educational Disparities:** States often exhibit distinct educational landscapes due to factors such as historical policies, economic development, and cultural influences.
2. **Contextualizing Educational Data at the State Level:** By considering state affiliation alongside educational qualifications, predictive models can better capture the regional nuances in candidate profiles. Incorporating state as a predictor variable allows the model to account for state-specific factors that influence educational attainment, thereby improving the accuracy and relevance of predictions.



The educational plot for SC (Scheduled Caste), ST (Scheduled Tribe), and unreserved candidates reveals variations in educational qualifications among these demographic groups, emphasizing the importance of using SC, ST, and unreserved status as parameters to predict educational qualifications.Thus, incorporating SC, ST, and unreserved status as parameters in predicting educational qualifications acknowledges the diverse educational backgrounds and experiences of candidates from different demographic groups. By considering these factors, predictive models can better capture the nuanced relationships between demographic characteristics and educational attainment, leading to more accurate and equitable predictions.

3.Final F1 score :-  
 Best final f1 score- 0.26505

Leaderboard rank-23rd(including the deleted entries)

Github repo:- <https://github.com/arshitnarang123/CS253_ML_ass>

4.References:-

a)Sci-kit documentation on multiclass-classification:- <https://scikit-learn.org/stable/modules/multiclass.html>

b)Gfgs tutorial on naïve bayes bernoulli model  
<https://www.geeksforgeeks.org/bernoulli-naive-bayes/>

c)Gfgs tutorial on random forest  
<https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/>

d)Feature engineering tutorial

<https://www.javatpoint.com/feature-engineering-for-machine-learning>

e)Chatgpt for all miscellaneous doubts