CS253 Python Programming Assignment

Tanush Goel

April 2024

1 Methodology

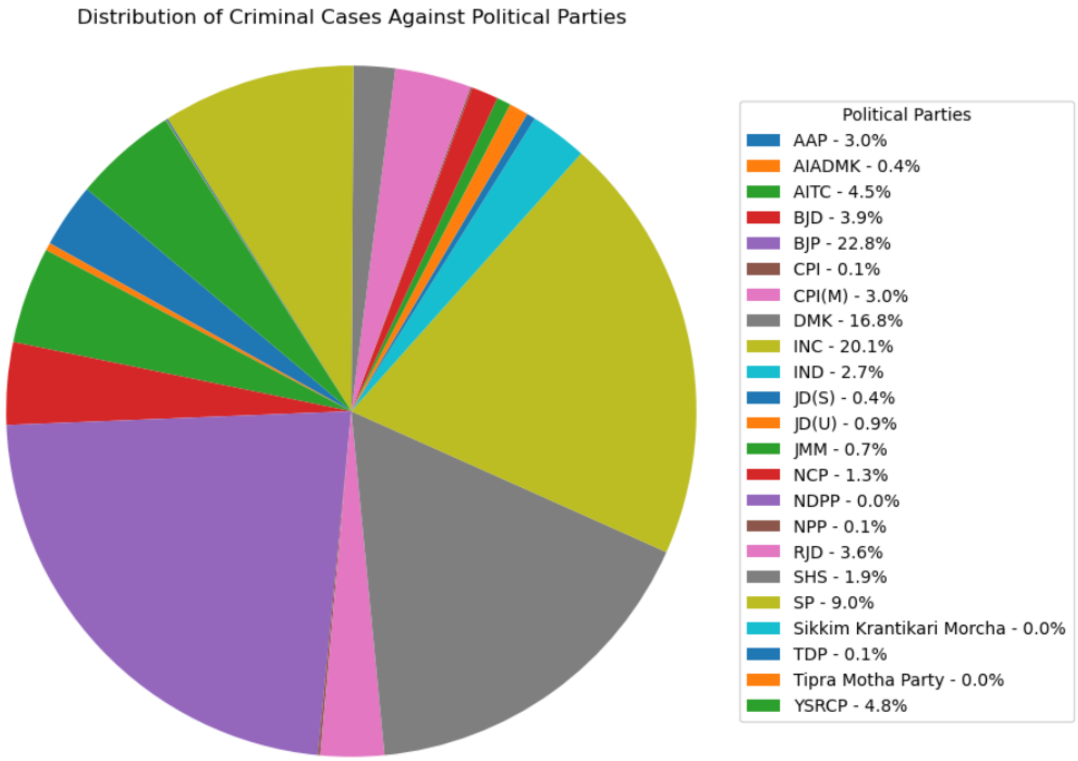
You can list the methodology used. For example:

1. **Data Preprocessing Steps:**
   Data was preprocessed by reading from CSV files and dropping unnecessary columns.
2. **Feature Engineering:**
   Asset and liability values were converted to lakhs. Categorical variables were encoded using one-hot encoding.
3. **Identifying Outliers:**
   Outlier detection techniques such as visualization or statistical methods like z-score or IQR could be applied.
4. **Dimensionality Reduction Techniques:**
   Although not explicitly used, Principal Component Analysis (PCA) could be applied.
5. **Normalization, Standardization, or Transformation Used:**
   Asset and liability values were converted to lakhs. Box-Cox transformation was applied for normalizing the data. Standardization was performed using StandardScaler.
6. **Others:**
   Hyperparameter tuning was conducted using GridSearchCV or RandomizedSearchCV. Model training and evaluation were performed using the Naive Bayes classifier (BernoulliNB). K-fold cross-validation was applied using KFold. Inverse label encoding was used for converting predictions back to original labels.

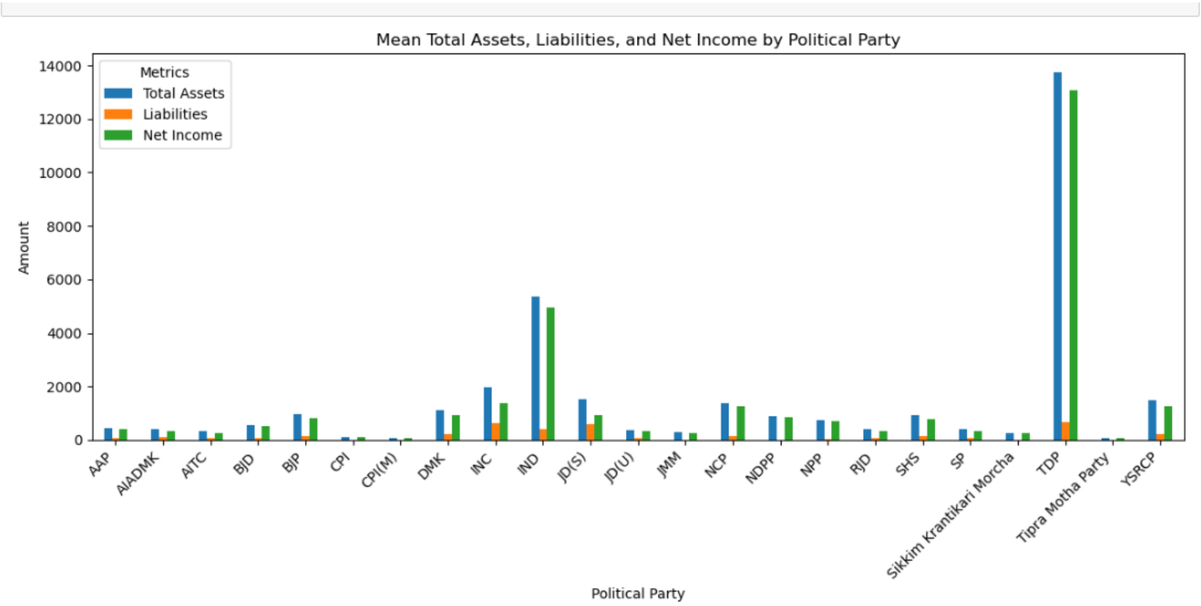
2 Experiment Details

2.1 Data Insights

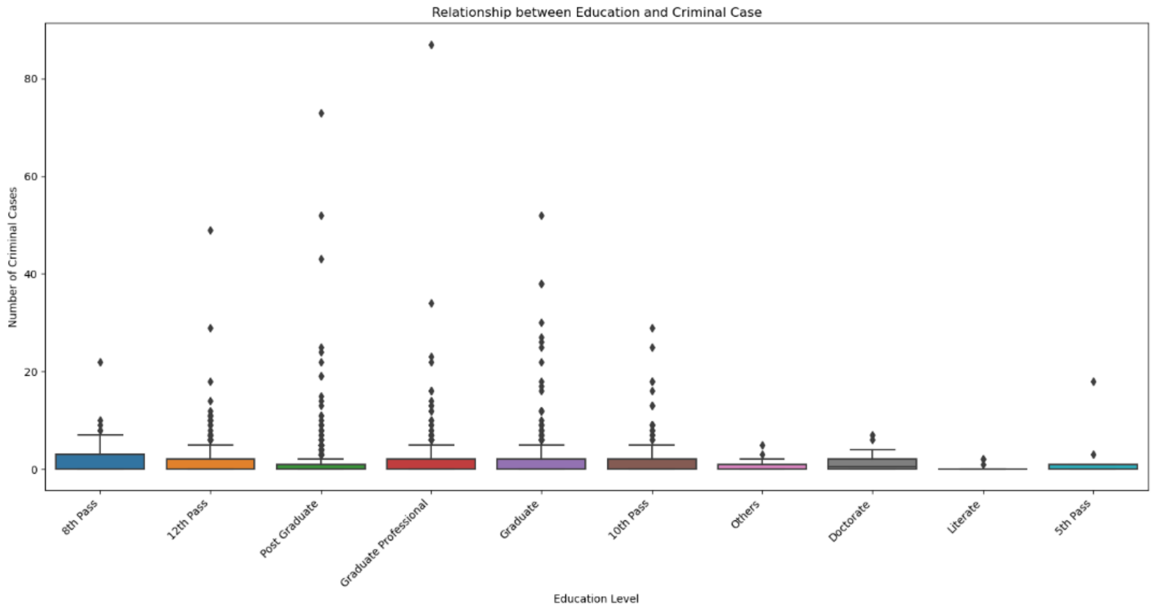
- **Percentage distribution of parties with criminal records**



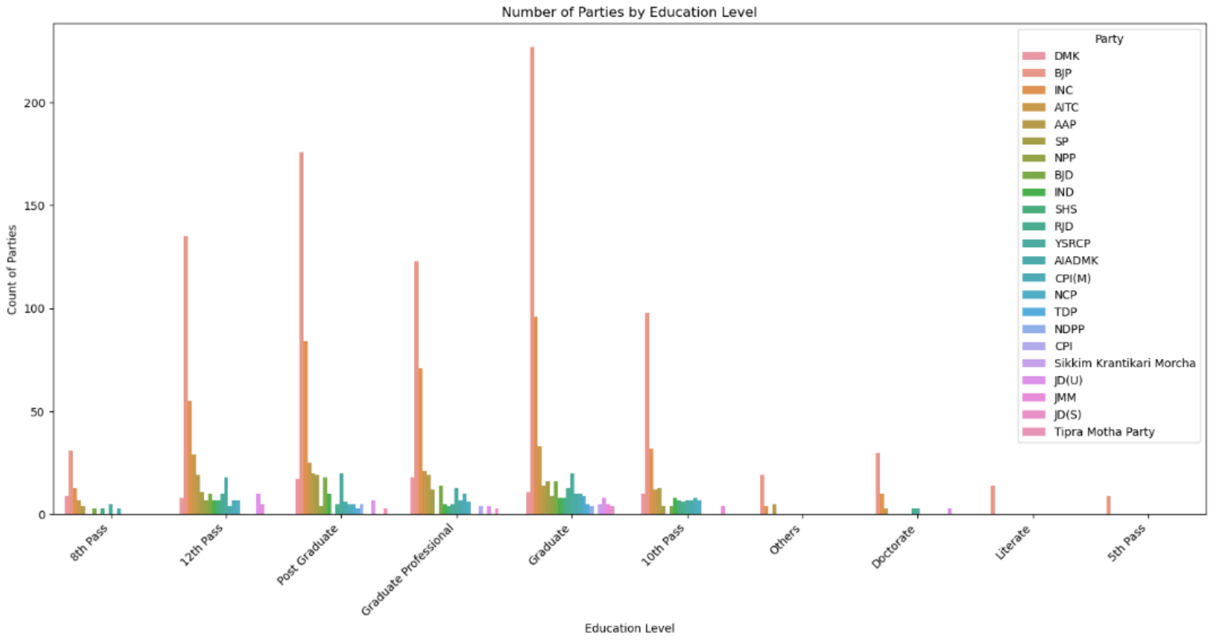
- **The percentage distribution of parties with the net income, assets and liabilities**



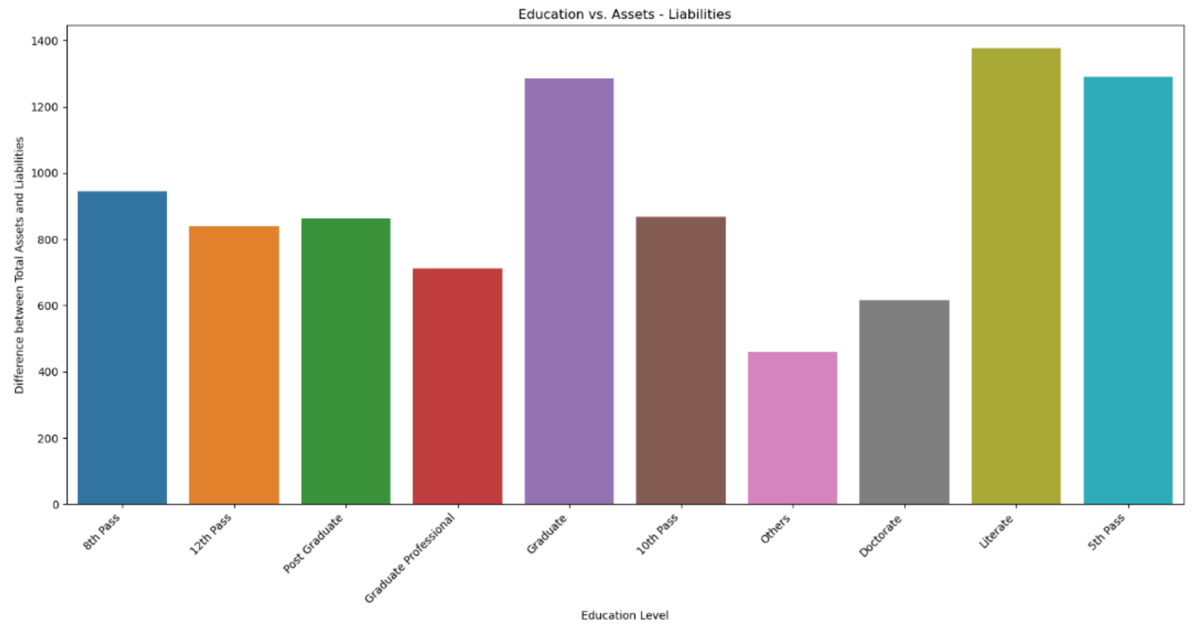
- **Relationship between Education and Criminal Case**



- **Relationship between Education and Political Parties**



- **Relationship between Education and Criminal Cases**



2.2 Insights

1. **Feature Engineering:** Conversion of asset and liability values to lakhs allows for easier interpretation and analysis of financial data.

2. **Categorical Encoding:** One-hot encoding of categorical variables (Party and state) ensures compatibility with machine learning algorithms that require numerical input.

3. **Model Selection:** The Bernoulli Naive Bayes classifier was chosen for its simplicity and ability to handle binary features, which may be suitable for the given dataset.

4. **Hyperparameter Tuning:** The hyperparameters of the Bernoulli Naive Bayes classifier (alpha, binarize, fit_prior) were tuned using K-fold cross-validation to optimize model performance.

5. **Prediction:** Predictions were made on the test dataset using the trained Bernoulli Naive Bayes classifier to determine the education level of individuals.

2.3 Models Used

Table 1: Bernoulli Naive Bayes (BernoulliNB)

Hyperparameter	**Value**
Alpha	1.0
Binarize	0.0
Fit Prior	True
Class Prior	None
Details	
Model Description	Predicts education level based on features
Assumptions	Features are binary after one-hot encoding
Hyperparameter Tuning	K-fold cross-validation

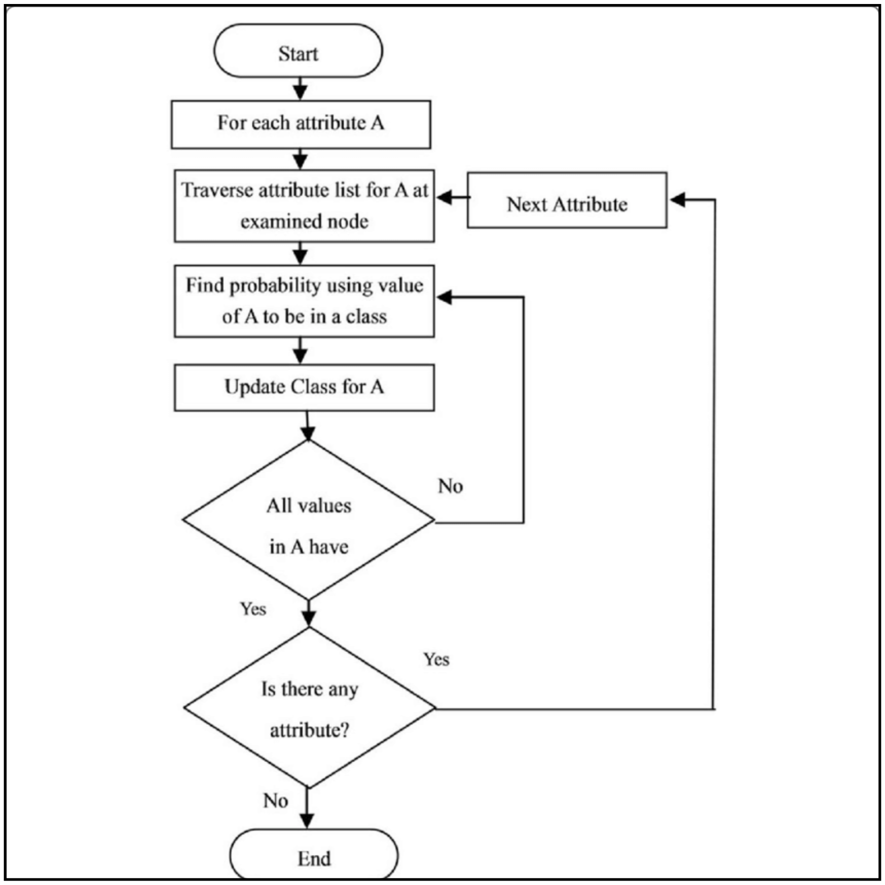


Figure 1: Bernoulli Naive Bayes

3 Results

- **Public F1 Score:** 0.2403
- **Private F1 Score:** 0.25655
- **Public Leaderboard Rank:** 93
- **Private Leaderboard Rank:** 27

4 References

1. GeeksforGeeks - Bernoulli Naive Bayes
   This article provides a detailed explanation of the Bernoulli Naive Bayes classifier.
2. Scikit-Learn Documentation - LabelEncoder
   The official documentation for the LabelEncoder class in Scikit-Learn, used for encoding categorical variables.
3. Kaggle - Intro to Machine Learning
   Kaggle's introductory course to machine learning, providing comprehensive tutorials and exercises.
4. Bernoulli Naive Bayes Video Tutorial
   This YouTube video provides a detailed explanation of the Bernoulli Naive Bayes classifier.

5 GitHub Repository

The code related to the above problem statement can be found in the GitHub repository linked below. The repository includes the two best submission files along with the ‘train.csv‘ and ‘test.csv‘ datasets.The repository link is provided below: **GitHub Repository**