Python Assignment Report

Daksh kumar Singh
220322

April 2024

1 Methodology

1.1 Data Preprocessing Steps

- Loaded the training and test data from CSV files.
- Combined the training and test data to fit the `LabelEncoder` for categorical variables.
- Used `LabelEncoder` to convert categorical features to numeric values.

1.2 Feature Engineering

Features chosen for modeling included 'Party', 'Criminal Case', 'Total Assets', 'Liabilities', and 'state' based on their potential impact on the target variable ('Education').

1.3 Data Splitting

The training data was split into training and validation sets with a ratio of 80% for training and 20% for validation.

1.4 Model Training and Evaluation

Three classifiers were employed: `RandomForestClassifier`, `DecisionTreeClassifier`, and `KNeighborsClassifier`. These classifiers were combined into a `VotingClassifier`, using hard voting. Models were trained on the training data and evaluated on the validation set to check their performance. After evaluation, the final model was trained on the full training set to make predictions on the test data.

2 Experiment Details

Model	Hyperparameters
`RandomForestClassifier`	n_estimators=200, max_depth=15, min_samples_split=5, min_samples_leaf=2, random_state=42
`DecisionTreeClassifier`	max_depth=7, min_samples_split=5, criterion='gini', random_state=42
`KNeighborsClassifier`	n_neighbors=5, weights='distance'

Table 1: Models and their hyperparameters.

2.1 Voting Classifier

The voting method used was hard voting, which combines predictions from the three models.

3 Data Insights

Data preprocessing and model selection aimed to capture the relationships between features and the target variable. Exploratory data analysis led to the selection of the chosen features.

```
    corr=new_df.corr()
plt.figure(figsize=(10,5))

sns.heatmap(corr,annot=True)
```

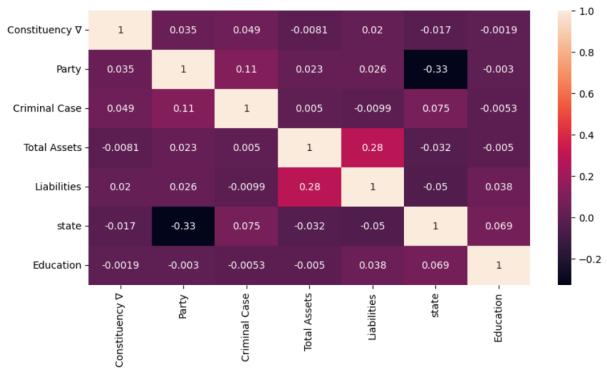

Figure 1: A descriptive caption for Figure 1. This figure shows the correlation between different variables in the dataset.

```
X
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
X_train
sns.histplot(y_train,bins=50)
```

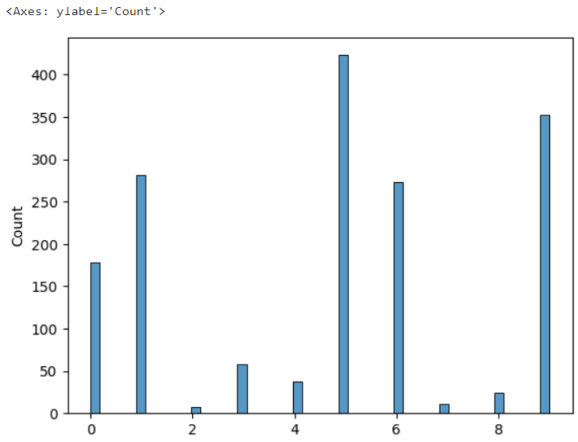

Figure 2: A descriptive caption for Figure 2. This figure shows the distribution of the target variable before oversampling.

```
from imblearn.over_sampling import SMOTE

oversample = SMOTE()
X_train, y_train = oversample.fit_resample(X_train, y_train)

sns.histplot(y_train,bins=50)
```

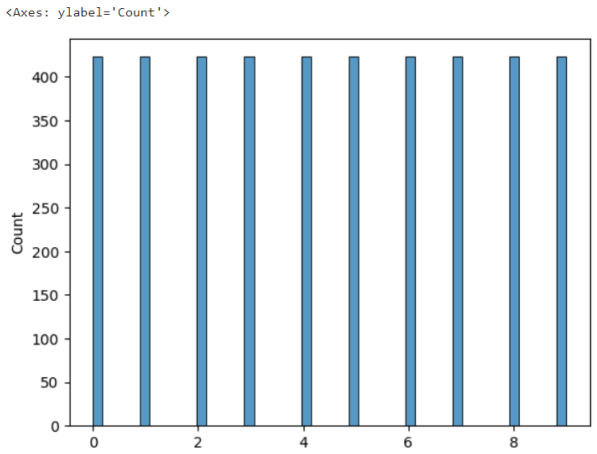

Figure 3: A descriptive caption for Figure 3. This figure shows the distribution of the target variable after oversampling.

4 Results

- Validation F1 Score: 0.1982 (Weighted average F1 score across classes).
- Public Leaderboard Rank: 112.
- Public F1 Score: 0.23638.
- Private Leaderboard Rank: 176.
- Private F1 Score: 0.21643.

5 References

- Data Sources: The data was loaded from `train.csv` and `test.csv`.
- Libraries Used: Used `pandas`, `scikit-learn`, and other libraries for data analysis and modeling.
- Used voting classifier and learned from the following links:
  1. Kaggle - Voting Classifier
  2. GeeksforGeeks - ML Voting Classifier using sklearn

6 GIT Repo Link

- github.com/daksh677/Machine-learning/tree/main