Ethem Can Eröksüz

190101035

Fraud Detection Project Report

This project is a machine development study aimed at fraud detection.In the project, different machine development models were classified, their performances were evaluated and the best models were selected.

Reading Dataset & First Observations

Dataset Info: The binary classification goal is to predict if the transaction is fraud (variable Class).

The dataset contains transactions made by credit cards in September 2013 by European cardholders.

Dataset Link: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Notes: This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, dataset cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are Time and Amount.

Features:

1 - Time: Number of seconds elapsed between this transaction and the first transaction in the dataset

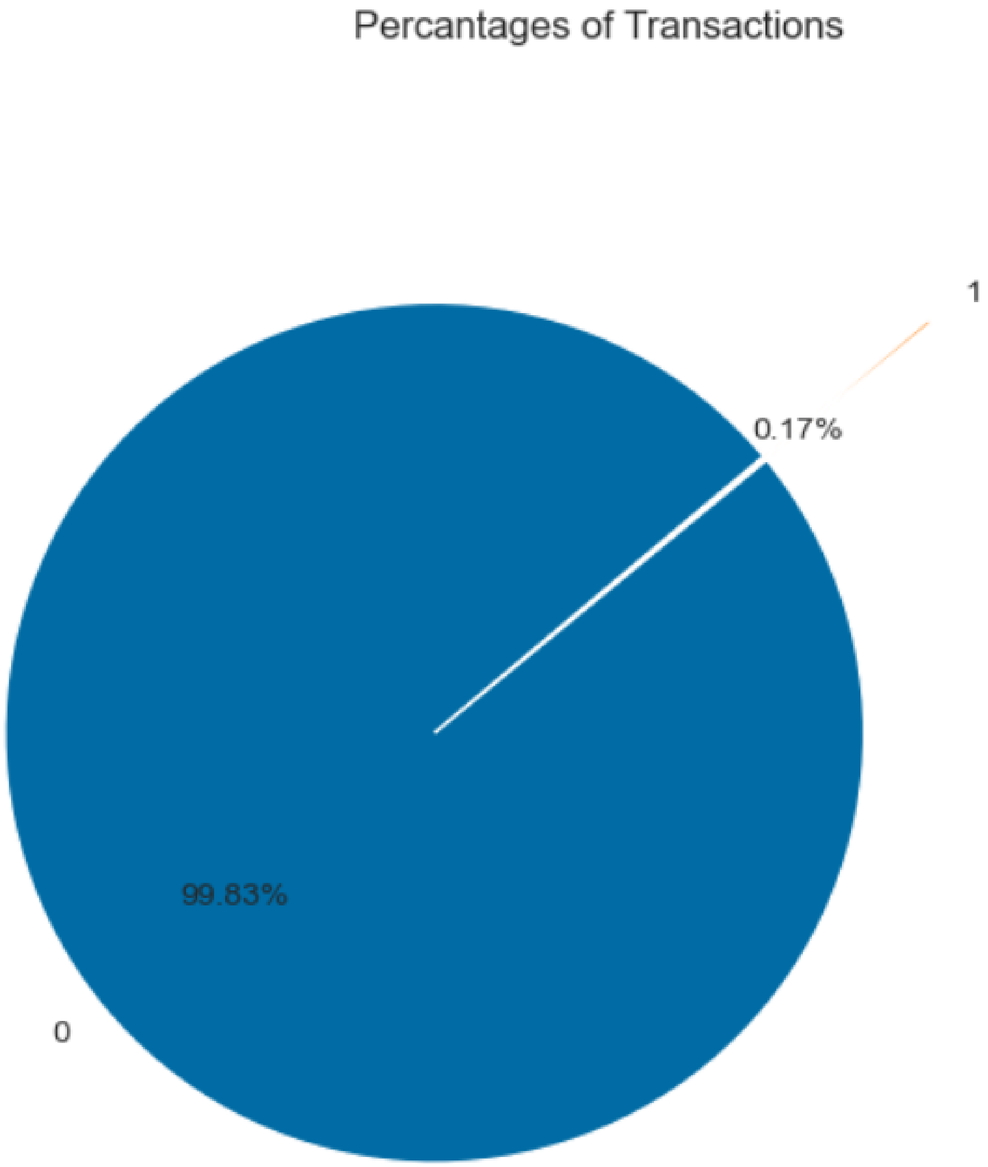
2 - V1,V2...V28: Those features are anonamised due to customer confidentiality, (all of them are numeric)

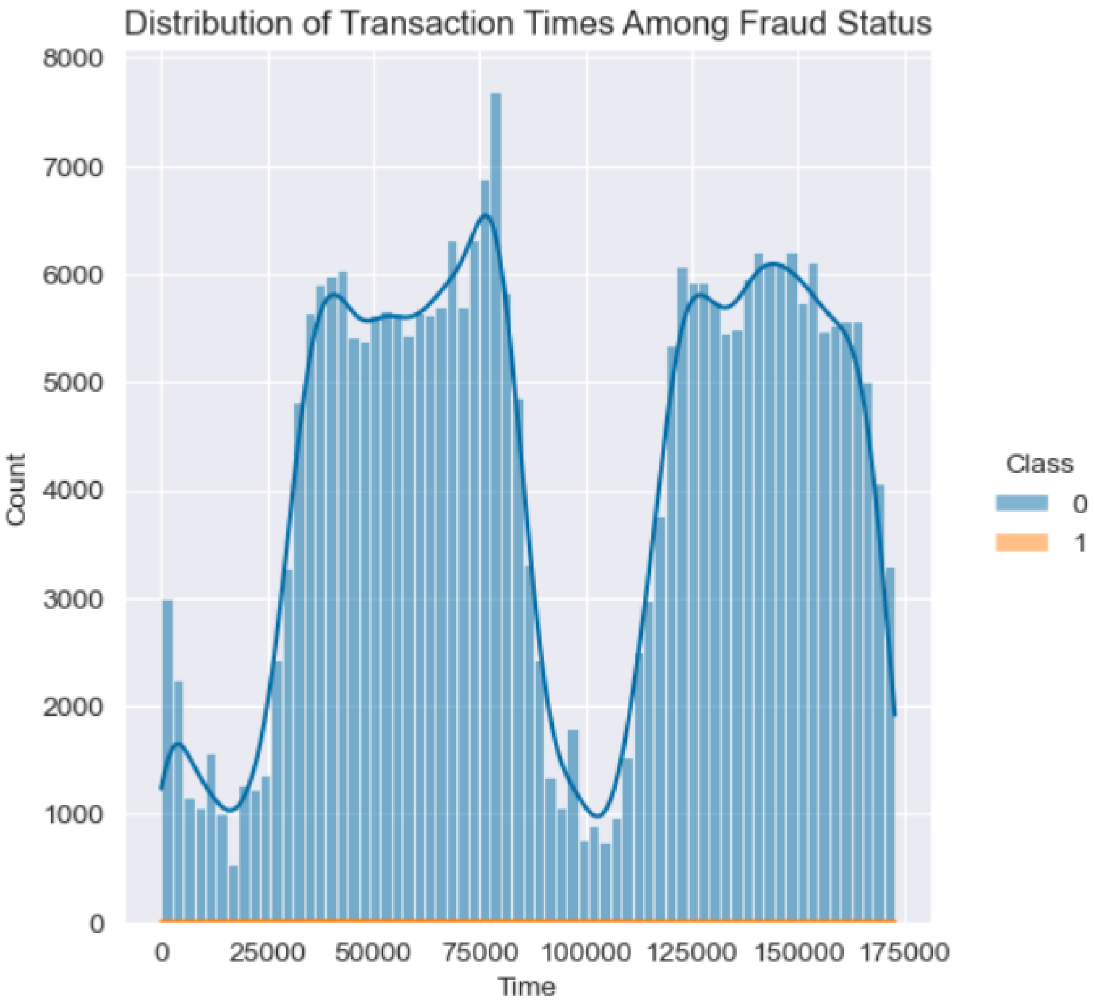
3- Amount: Transaction amount

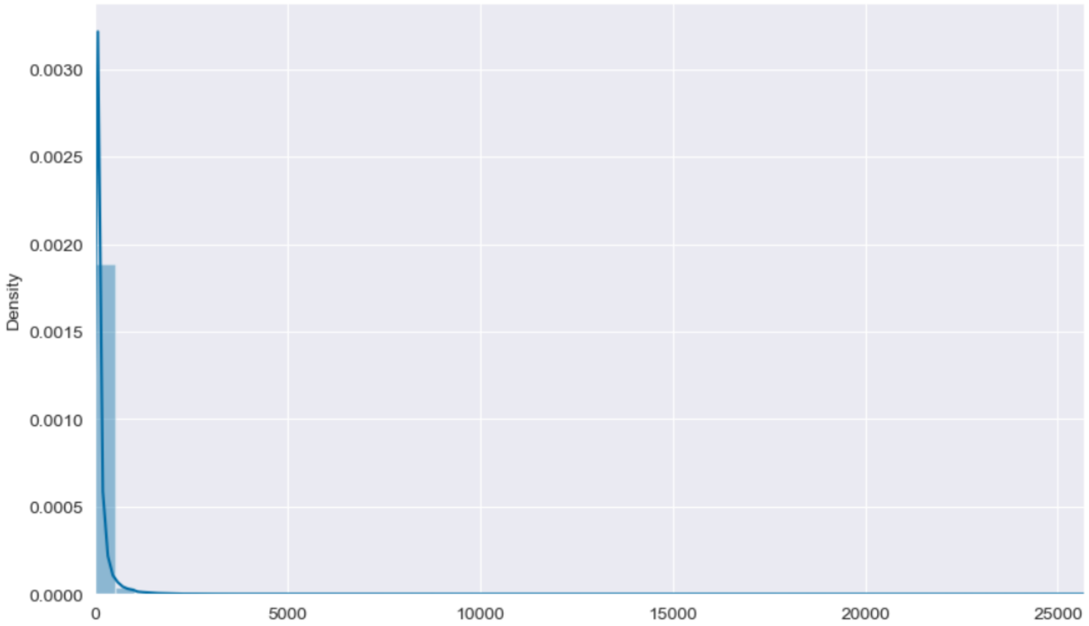
4- Class: Target variable, 1 for fraudulent transactions, 0 otherwise

EDA & Visualizations

Data distributions were examined by visualizing with various graphs;







Since we didn't have any knowledge or domain knowledge about anonymized columns, so we didn't have a chance to do feature engineering, I went directly to the pre-processing phase.

Preprocessing

I've decided to use RobustScaler because it is less prone to outliers.

I split the dataset into training and test sets.

To fix the imbalance I first used the Undersampler method.

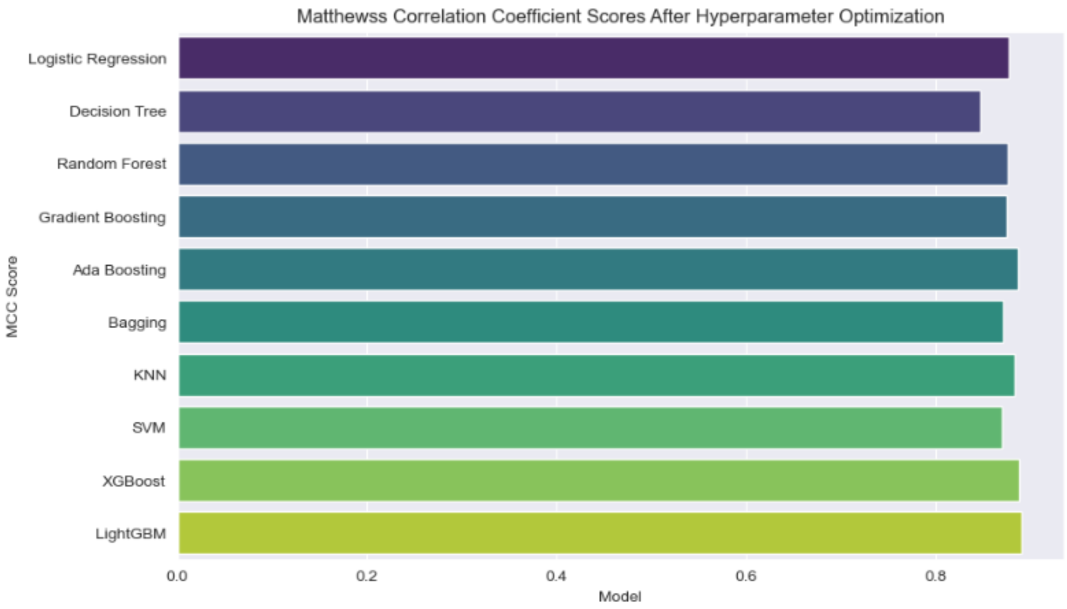
Undersampler

I used the undersampling method due to the imbalance in the dataset. This technique reduces the number of instances in the majority class to match the minority class, helping to balance the class distribution and improve the performance of the model on imbalanced data.

Model Selection & Hyper-parameter Optimization

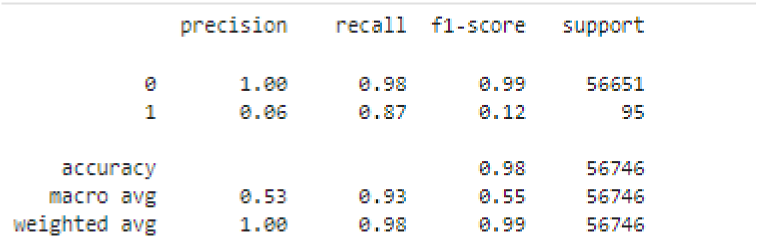
I used mcc(Matthews Correlation Coefficient) as the scoring metric because accuracy is not optimal performance metric for this project.

The hyperparameter optimization process identified optimal parameter combinations to achieve the best performance of each model. After optimization, the increase in each model's MCC score demonstrates how this process improves model performance. In this way, more accurate and reliable models were obtained in fraud detection.



Afterwards,I created a Voting Classifier by choosing the top five among the best models. Voting Classifier makes final predictions by combining the predictions of more than one model and thus aims to increase the overall model performance.

I made a prediction with the Voting Classifier Model and the results are as follows:



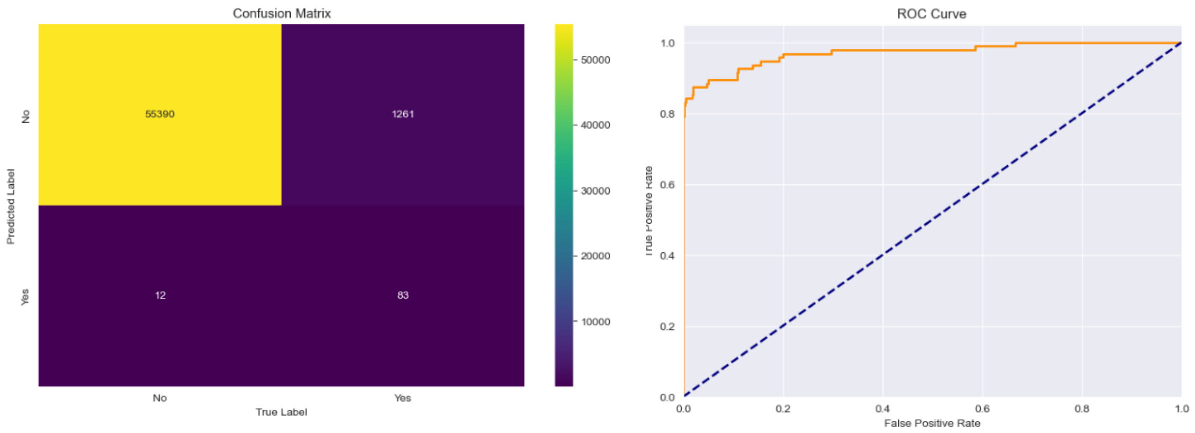
```
              precision    recall  f1-score   support

           0       1.00      0.98      0.99     56651
           1       0.06      0.87      0.12        95

    accuracy                           0.98     56746
   macro avg       0.53      0.93      0.55     56746
weighted avg       1.00      0.98      0.99     56746
```

As seen in the image, the recall value is high, but the precision value is low. This means that many innocent people are classified as fraudsters, but actual fraudsters are less likely to go undetected.

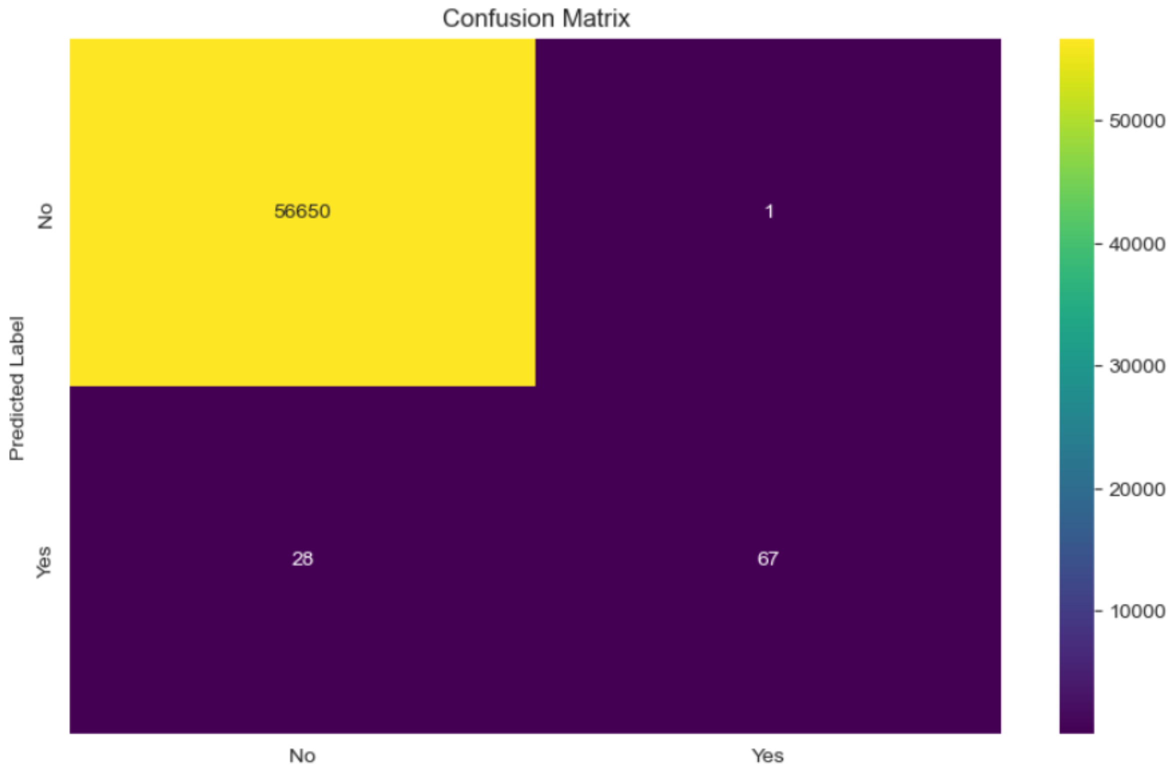
Subsequently, the confusion matrix and ROC curve generated are as follows:



Adjusting Class Weights

After using the undersampling method, I adjusted the class weights to better address the class imbalance. This process ensures that the model places more importance on the minority class (fraud), aiming to improve overall model performance.

The confusion matrix in this method I use is as follows.



As seen, in the undersampler method, when we prioritize the recall value, 1261 innocent individuals were classified as fraudsters, whereas in this method, since we prioritize the precision value, this number dropped to 1. Additionally, in the undersampler method, 12 fraudsters were missed. In this method, however, 28 fraudsters were missed.