mirzaozeer@gmail.com
/Dodger22
/mirzazer
MİRZA ÖZER

CREDIT CARD FRAUD DETECTION

INTRODUCTION

Credit card fraud happens when consumers give their credit card number to unfamiliar individuals, when cards are lost or stolen, when mail is diverted from the intended recipient and taken by criminals, or when employees of a business copy the cards or card numbers of a cardholder

In recent years credit card usage is predominant in modern day society and credit card fraud is keep on growing. Financial losses due to fraud affect not only merchants and banks (e.g. reimbursements), but also individual clients. If the bank loses money, customers eventually pay as well through higher interest rates, higher membership fees, etc. Fraud may also affect the reputation and image of a merchant causing non-financial losses that, though difficult to quantify in the short term, may become visible in the long period.

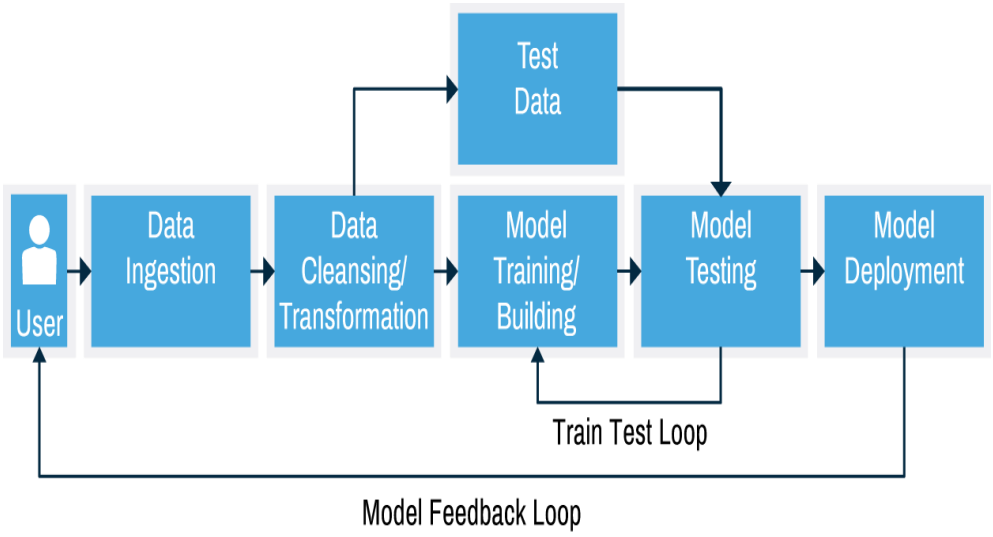
A Fraud Detection System (FDS) should not only detect fraud cases efficiently, but also be cost-effective in the sense that the cost invested in transaction screening should not be higher than the loss due to frauds . The predictive model scores each transaction with high or low risk of fraud and those with high risk generate alerts. Investigators check these alerts and provide a feedback for each alert, i.e. true positive (fraud) or false positive (genuine).

Most banks considers huge transactions, among which very few is fraudulent, often less than 0.1% . Also, only a limited number of transactions can be checked by fraud investigators, i.e. we cannot ask a human person to check all transactions one by one if it is fraudulent or not.

Alternatively, with Machine Learning (ML) techniques we can efficiently discover fraudulent patterns and predict transactions that are probably to be fraudulent. ML techniques consist in inferring a prediction model on the basis of a set of examples. The model is in most cases a parametric function, which allows predicting the likelihood of a transaction to be fraud, given a set of features describing the transaction.

METHODOLOGY

Fraud detection is a binary classification task in which any transaction will be predicted and labeled as a fraud or legit. In this Notebook state of the art classification techniques were tried for this task and their performances were compared.



-Logistic Regression
-Linear Discriminant Analysis
-KNeighbors Classifier
-RandomForest Classifier
-Decision Tree Classifier
-XGB Classifier
-GaussianNB
-Gradient Boosting Classifier
-LGBM Classifier

The dataset contains transactions made by credit cards in September 2013 by European cardholders. 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, we 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'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset.

The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

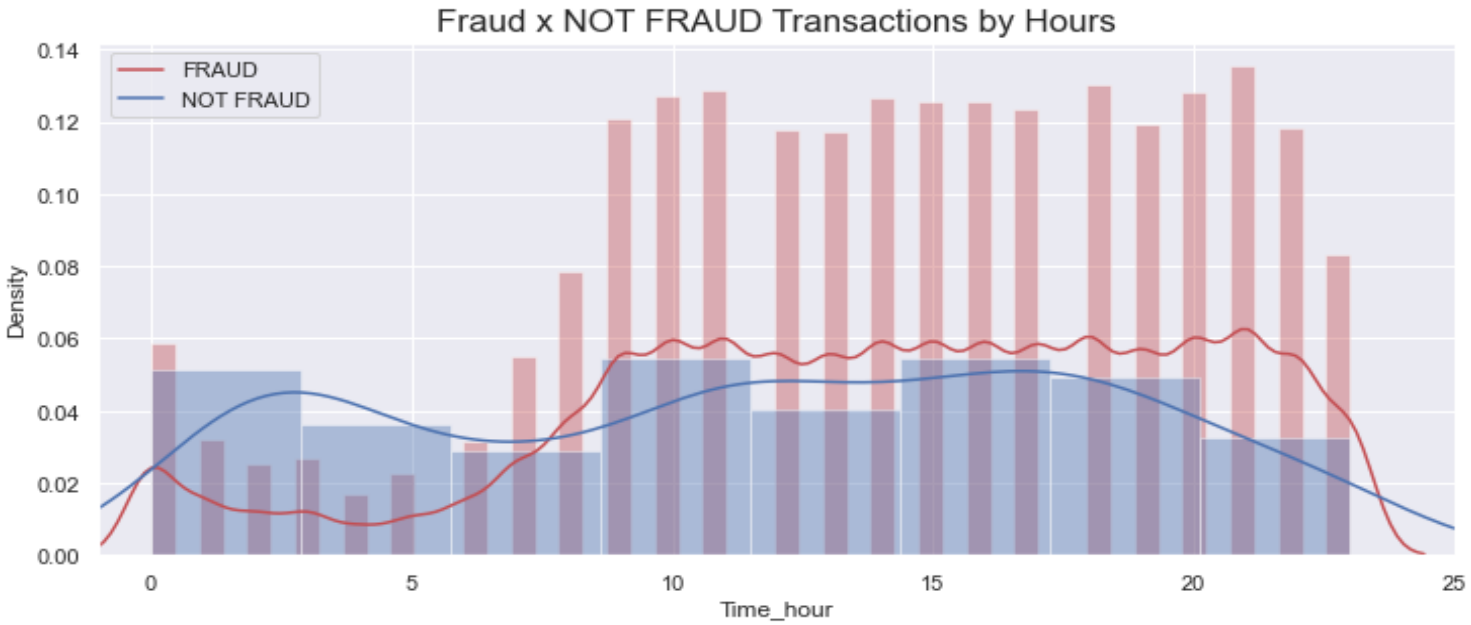
```
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

There are not any null variable. The data set contains 284,807 transactions. The mean value of all transactions is 88.35 USD while the largest transaction recorded in this data set amounts to 25,691 USD.

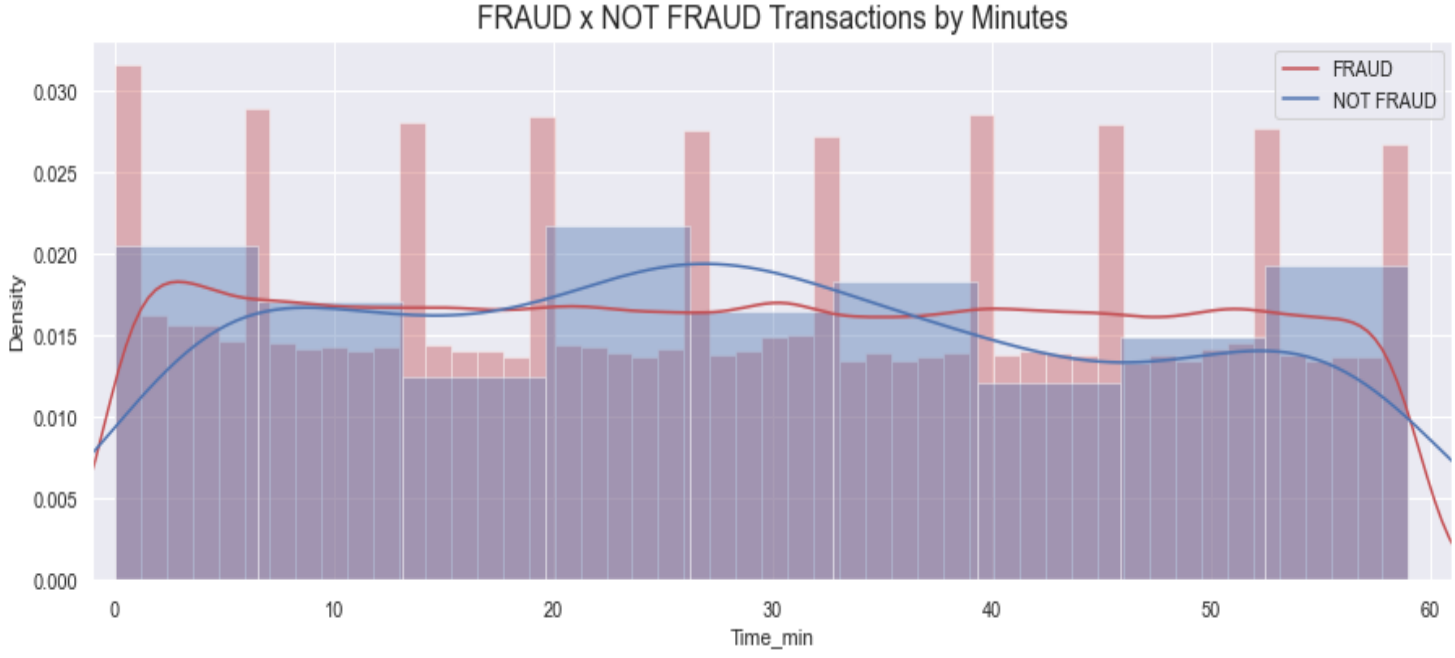
However, as you might be guessing right now based on the mean and maximum, the distribution of the monetary value of all transactions is heavily right-skewed. The vast majority of transactions are relatively small and only a tiny fraction of transactions comes even close to the maximum.

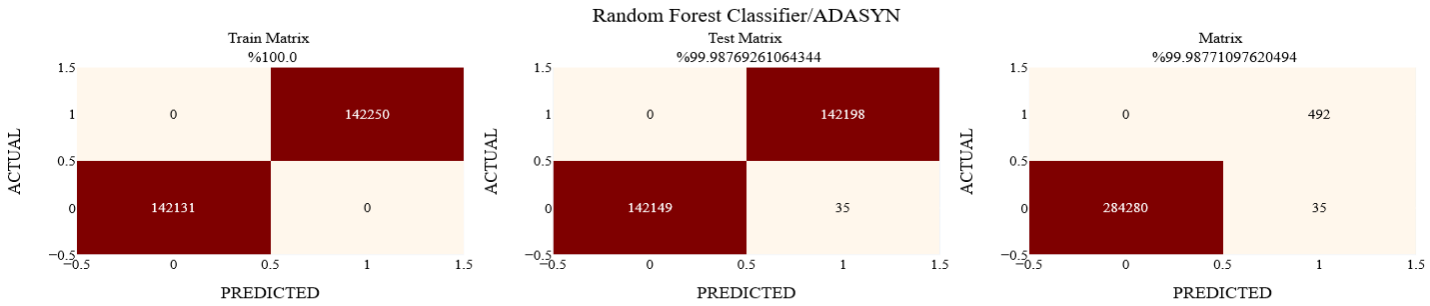
```
NOT FRAUD % 99.82725143693798
FRAUD % 0.1727485630620034
-------------------------------------------------
NOT FRAUD AMOUNT % 99.87508652641638
FRAUD % AMOUNT 0.12491347358363826
```

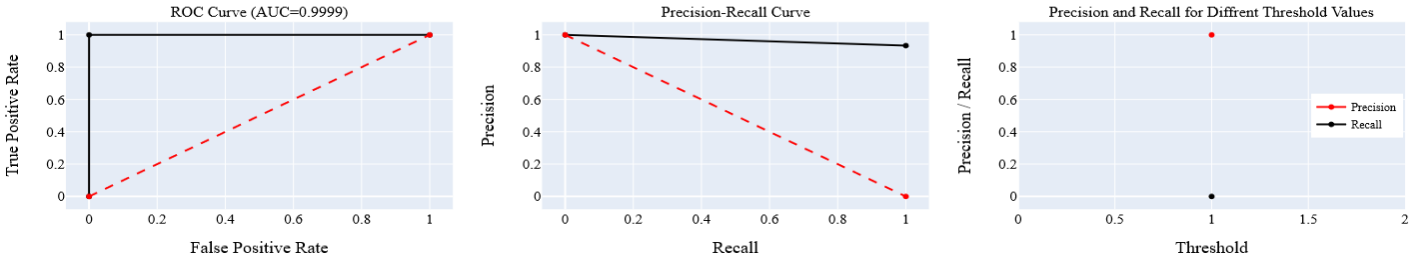
As you can see, there are 284315 "Not Fraud" transaction and 492 "Fraud" transaction . Only %0.17 transaction is "Fraud". "Not Fraud " transaction prediction might be very easy but "Not Frauds" transactions are very low according to "Not Fraud" transaction so "Not Fraud" transaction predicting is hard.



When we scale the transactions hourly, we can see that between 6 and 24 frauds have increased and the density has increased. When we do the same scaling in minutes, we can see that the distribution is equal, but the density of fraud transactions is still dominant. However, we should not forget that only 0.17% of transactions are fraudulent transactions.







Classification Report

Classification Report	precision	recall	f1-score	support
0	1	0.9998768971035648	0.9999384447629684	284315
1	0.9335863377609108	1	0.9656526005888125	492
accuracy	0.9998771097620494	0.9998771097620494	0.9998771097620494	0.9998771097620494
macro avg	0.9667931688804554	0.9999384485517824	0.9827955226758904	284807
weighted avg	0.9998852713528O5	0.9998771097620494	0.9998792164598238	284807

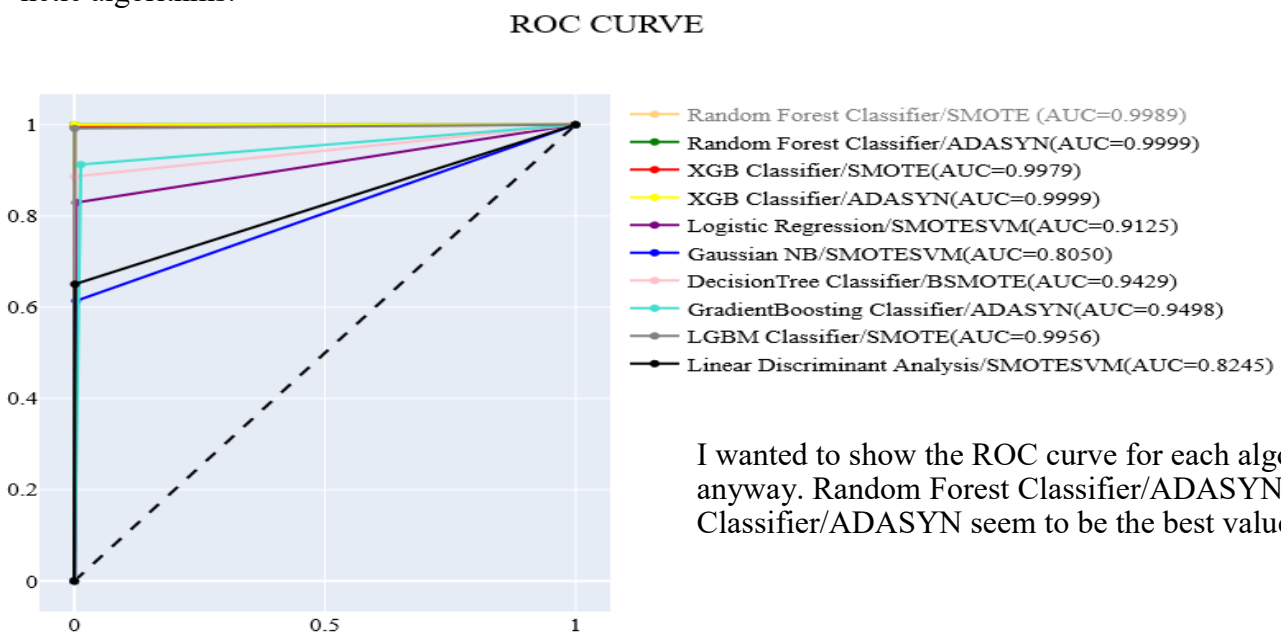
NOT FRAUD= 0 / FRAUD= 1

CONCLUSION

After using all the methods we determined, I decided that the method with the Classification table the most logical and efficient confusion matrix results is Random Forest Classifier/ADASYN. But you can also examine the tables of other methods from my kaggle or github account.

I think Random Forest Classifier/ADASYN method is better. When we look the charts, ı see the results are beter on the Random Forest Classifier/ADASYN charts. But don't forget, did not use Future Selection methods and when I got data , the data was PCA format so see the outiler data or noisy data is very hard. I would not wanted to incorrect predict. The data already was İnbalanced.

There are 35 False Positive in Forest Classifier/ADASYN method. This means, per 284772 transactions will have 35 wrong predict. When you looking first it can be looks good. But there are doing millions transactions by customers every day in the bank. This means, the banks might lock up hundered of customers account unnecessary and this would reduce bank confidence.This method can be developed with different methods and implement feature selection or Cross Validation methods. The data reviewing again with Neural Network or using Genetic algorithms.



I wanted to show the ROC curve for each algorithm anyway. Random Forest Classifier/ADASYN and XGB Classifier/ADASYN seem to be the best values.