ML Project

Online Payments Fraud Detection

This project represents a classification task for determining a transaction is suspected as Fraud or not. Different ML models tested for improving the quality of detection the positive class/ the fraudulent transaction (the recall rate). The ideal case is to achieve accuracy, precision and recall equals to 1.0. In this work, the winner ML model increases the ability to detect fraudulent samples from 0.002 to 0.9957 recall rate.

The dataset:

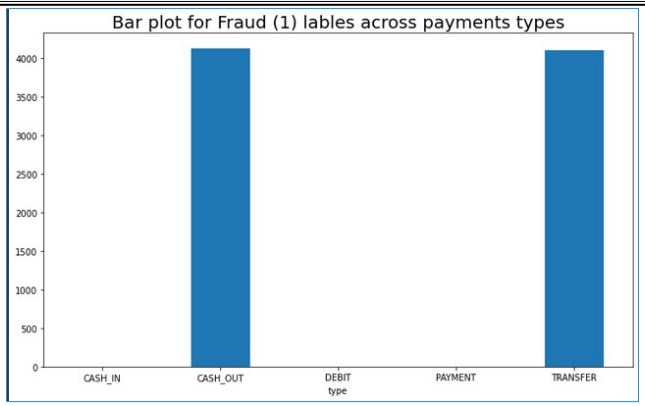
The dataset contains samples of online payments with Fraud/ not Fraud labels. To enhance detecting Fraudulent transactions, different ML Models implemented. The raw data show that only 16 out of 8213 fraud transactions was correctly detected (0.002 recall rate). This work objective to improve the ability for detection (higher recall rate).

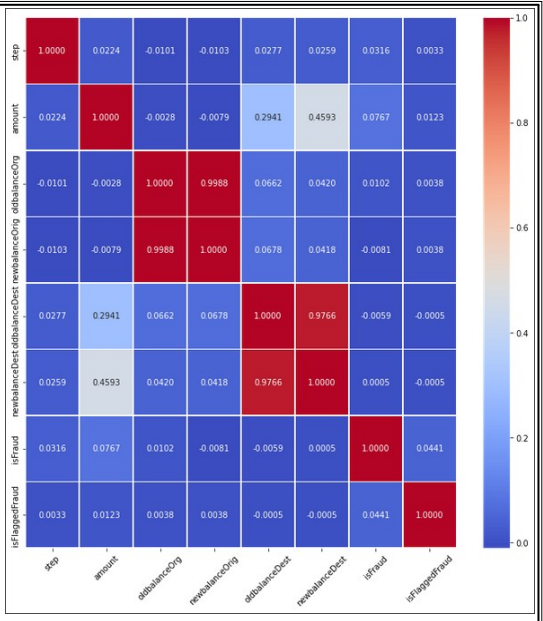
The provided variables as follows:

1. step: represents a unit of time where 1 step equals 1 hour (int).
2. type: categories of online transaction (PAYMENT, TRANSFER, CASH_OUT, CASH_IN, DEBIT).
3. amount: the amount of the transaction (float).
4. nameOrig: customer starting the transaction (str).
5. oldbalanceOrg: sender's balance before the transaction (float).
6. newbalanceOrig: sender's balance after the transaction (float).
7. nameDest: recipient of the transaction (str).
8. oldbalanceDest: initial balance of recipient before the transaction (float).
9. newbalanceDest: the new balance of recipient after the transaction (float).
10. isFraud: fraud transaction (1) or not (0) - target label (int).
11. isFraudFlagged: fraud transaction detected (1) or not (0) (int).

Exploratory Data Analysis (EDA):

- The dataset is extremely imbalanced to the 0 class (more than 99%). About 0.13% of data belong to the positive class.
- The features "nameOrig" and "nameDest" are anonymized as so they are irrelevant.
- The dataset has no missing values to substitute.
- All fraudulent transactions belong to two types out of five: CASH_OUT and TRANSFER. Majority of samples (69%) belongs to these two types as shown in the bar plot below. This helps ML models to capture fraud transactions.





- Pearson's correlation between the variables showed that the target variable has no linear relations with any of the variables. This hints to the relationship to be a nonlinear as shown in the heat map.
- All categorical variables have significant relationship with the target variable based on Chi^2 test.
- Except for "newbalanceDest" variable, numerical variables in the dataset do make difference in the mean between the two groups; negative and positive samples, based on the t-test.
- Mutual Information test tells that all predictors are independent from the target variable with very low values.
- Since there are small number of features, all features involved in the tests is taken into account.

- Preprocessing:

- Ordinal Encoder was implemented for encoding "type" feature categories.
- Robust encoder was applied on numerical features since those features contains a lot of outliers upon box-and-whisker plot charts.

- Metrics:

The accuracy measure considered useless as the dataset is very unbalanced. Instead, Precision and Recall are considered for examining the quality of the predictions. Visually, Precision-Recall (PR) and ROC curves are plotted for getting an easy assessment of each model.

- Modeling and Results:

- Logistic regression was able to bring high results on one class only. The precision value is almost double the recall value **(0.9 vs 0.477)**. This is due to nonlinear boundary between the two classes. Neither reducing the number of features nor creating new by feature engineering was helpful for bringing better results in the case of linear regression.
- Using LASOO for feature importance does not help about which features are more contributing in the decision.

- KNN -with K=3 - achieved better results for both the precision (91.22) and the recall (75.00) metrics on testing data. Though, KNN model overfits. K =3 assigned to the model after running grid search function. Increasing K could reveal less overfitting model.
- KNN model could not hit higher performance because the unbalance between the two classes. Later, borderline SMOTE was included in the pipeline for oversampling label 1. Yet, it did not help. It led to more bias to the positive (1) class by which larger positive predictions was the result of that. So, the precision got largely down to smaller value (0.5961) in favor of small increase on the recall **(0.8290)** and AUC score.
- Random Forest (RF) Classifier was applied on the dataset. The Random Forest Classifier has generalized well on validation and testing data with precision and recall values equal to **98.95 and 75.68** respectively because of the nature of ensemble learning in RF algorithm. The probability threshold tuned to 0.01 in order to increase the recall to **99.38.**
- "newbalanceDest", "oldbalanceOrg" and "amount" have considerable contribution in decision trees (i.e estimators) that were run by RFC according to the importance map. The other five features have marginal contribution - less than 0.1.
- Also, XGBoost was chosen to make prediction and achieved recall equals to (**0.6741**). The model overfits a little on the recall on testing data, which could be from such Boosting kind of models, contrary to RFC model. RFC model follows bagging method for computing the outcome which has more success to overcome overfitting than Boosting.
- Then SMOTE technique was included in the pipeline. The technique helped XGBoost to bring near perfect recall **(0.9957)**. ''oldbalanceOrg', 'amount' features are the top features according to the feature map by which XGBoost model relied on in predicting.