ST 4007 Statistics in Practice II

A PREDICTIVE MODEL TO DETECT FRAUD IN ONLINE PAYMENTS

Project Report

Prepared By

S. M. Shashiprabha Senanayake
s15659

Abstract

Online payments have emerged as the primary mode of transactions globally, thanks to their speed and ease of use. However, the proliferation of digital payment methods has brought about a rise in payment fraud, creating significant challenges for users, including financial institutions and consumers. Fraudulent transactions lead to financial losses and erode the integrity and trust in online payment systems, thereby posing substantial risks to these platforms.

This project developed a predictive model for detecting fraudulent transactions in online payment systems. By analyzing characteristics such as transaction type, transaction amount, and customer account balance, the model was trained to identify and flag potential frauds among transactions.

Key attributes that significantly impact fraud detection were selected for the model-building process, while redundant and unnecessary attributes were discarded. A thorough exploration of the dataset was conducted, and insights gained from exploratory data analysis guided the subsequent model-building phase. Since this is a binary classification problem, various supervised learning algorithms were employed. Appropriate preprocessing techniques, including handling imbalanced data using SMOTE and feature scaling, were applied to enhance model performance.

Initially, a logistic regression model was used, achieving an accuracy of 0.82. Then the decision tree classifier led to an accuracy of 0.99. As the decision trees inherently have a tendency to overfit on the training data as they can create complex decision boundaries that perfectly fit the training data, leading to poor generalization on unseen data, regularization techniques were used. Then the accuracy was 0.96. The effectiveness of each model was evaluated and compared based on accuracy and other relevant metrics.

Online payments have become the predominant method of payment worldwide, and parallel to that, it has led to an increase in payment fraud that presents significant challenges for the users of these services such as financial institutions and consumers. So this project holds significant importance in enhancing the security and reliability of financial transactions.

Contents

List of Figures and Tables

Introduction

In this digital age, online payments have revolutionized, providing unparalleled speed and convenience and also as the way of financial transactions are conducted. Because of this tendency, it has made as the preferred mode of transaction globally. However, with the proliferation of these online comes the heightened risk of payment fraud, presenting significant challenges for both financial institutions and consumers. Fraudulent transactions not only lead to financial losses but also decreases the trust and integrity of online payment systems.

This project addresses the critical issue of payment fraud by developing a predictive model, that designed to detect fraudulent transactions in online payment systems. By analysing the transaction characteristics such as the time step that transaction happening, transaction type, transaction amount, and customer account balances before and after the transactions, the model aims to accurately identify and flag potentially fraudulent activities. The dataset itself consist of flagging system, but that system has not flagged any fraudulent transaction. So developing a new model is crucial.

The model-building process involved a careful selection of key attributes that significantly impact fraud detection, while discarding unnecessary features. Comprehensive exploratory data analysis was conducted to derive valuable insights like, determining the overall percentage of fraud in online transactions, identify the types of transactions that are most susceptible to fraud, and analysing the amount of ranges where fraud is most likely to occur, will be provided a great understanding for the subsequent phases of model development. However, the main objective is to “build a predictive model to detect frauds in online payments”.

As overall, conducting this study basically aims to enhance the detection and prevention of fraudulent activities in online payments. By identifying fraudulent transactions, financial institutions and customers can improve their confidence in online payment platforms. And also the service providers can also implement more robust security measures, and reduce financial losses.

As online payments continue to dominate the global financial landscape, addressing the associated risks of payment fraud becomes increasingly critical. This project plays a pivotal role in enhancing the security and reliability of financial transactions.

Literature Review

Online payment fraud detection is crucial for financial institutions due to the increasing volume of fraudulent transactions. This literature review explores various methodologies and advancements in the field by analysing three recent studies.

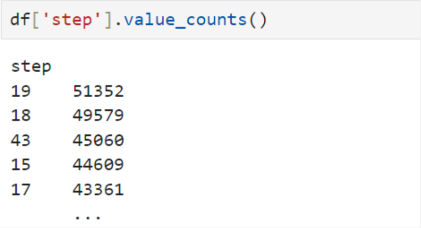
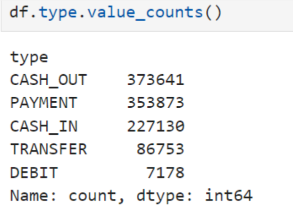
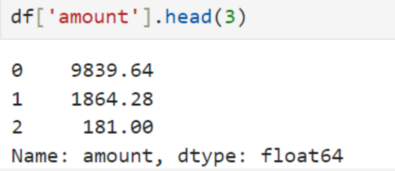
1. Shaohui, D., Qiu, G., Mai, H. and Yu, H., 2021, January. Customer transaction fraud detection using random forest. In 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE) (pp. 144-147). IEEE. This particular research project has published in 2021. In here they have proposed a model by outperforming traditional models like logistic regression and support vector machines, achieving an accuracy of 97.4% and an AUC ROC score of 92.7%. This research project highlights the potential of machine learning techniques in enhancing the robustness of fraud detection systems in online financial transactions, that are common today.
2. Mishra, K.N. and Pandey, S.C., 2021. Fraud prediction in smart societies using logistic regression and k-fold machine learning techniques. Wireless Personal Communications, 119(2), pp.1341-1367. This research paper has published on 2021. Their approach involves creating multiple folds of bank transaction data before applying logistic regression and machine learning techniques. That method encompass registration, classification, clustering, dimensionality reduction, deep learning, training, and reinforcement learning. The study has utilized intelligent machine learning tools such as ROC curves, confusion matrices, mean-recall scores, and precision-recall curves. The results indicate that their methodology is efficient, accurate, and reliable for detecting fraud, demonstrating the potential of advanced machine learning techniques in enhancing fraud detection in dynamic and complex digital environments.
3. Nami, S. and Shajari, M., 2018. Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors. Expert Systems with Applications, 110, pp.381-392. This research paper has published on 2018. Their methods involve, in the first stage, a new similarity measure based on transaction time is introduced, assigning greater weight to recent transactions to account for the dynamic nature of spending behaviors. In the second stage, a dynamic random forest algorithm is utilized for initial detection, coupled with a minimum risk model for cost-sensitive detection. Testing on a real transactional dataset the method demonstrated that recent cardholder behaviors significantly influence the evaluation of transactions as fraudulent or legitimate. Though this is not directly about online payment fraud, the methods, and techniques are same as the things in online payment fraud case. In this also they use a advanced version of random forest.

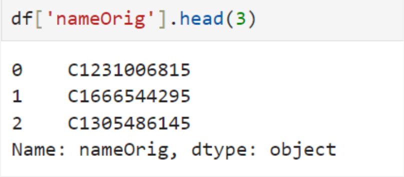
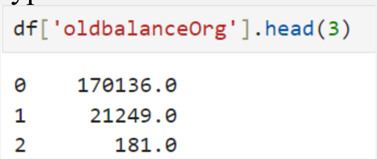
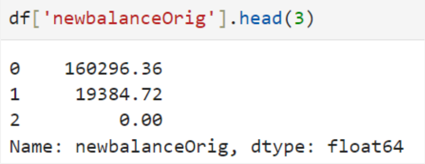
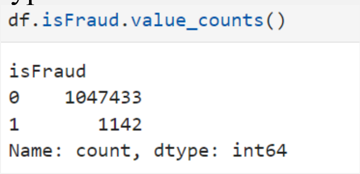
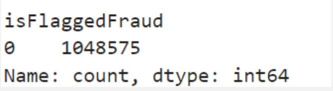
Data

The dataset consists of 11 variables. 10 are exploratory variables and the other is a binary-type response variable. And 1 048 575 unique observations are there in the dataset.

i. Variable Description

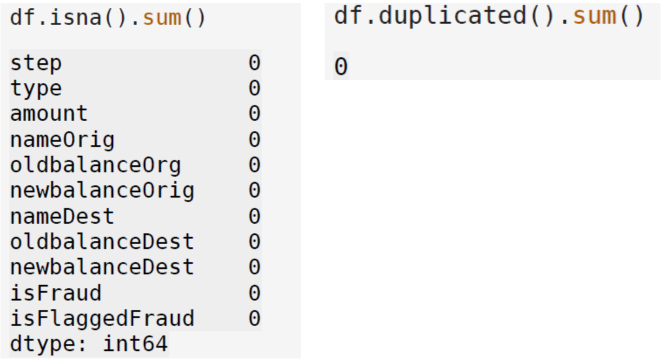
Table 1

Variable name	Description
Step	Represents a unit of time where 1 step equals 1 hour. This variable indicates a time step within a time duration that gathers data. One step is one hour. For a one-step (eg: step 1) there can be multiple observations happen. There are 1 to 95 steps. This is an int-type variable.  df['step'].value_counts() step 19 51352 18 49579 43 45060 15 44609 17 43361 ... *Figure 1*
Type	Type of online transaction (payment, transfer, cash-out, cash-in, debit). An object type variable. The value counts are as follows.  df.type.value_counts() type CASH_OUT 373641 PAYMENT 353873 CASH_IN 227130 TRANSFER 86753 DEBIT 7178 Name: count, dtype: int64 *Figure 2*
Amount	The amount of the transaction. A float type variable.  df['amount'].head(3) 0 9839.64 1 1864.28 2 181.00 Name: amount, dtype: float64 *Figure 3*
Nameorig	Identifier for the customer initiating the transaction. Object type variable.

	 df['nameOrig'].head(3) 0 C1231006815 1 C1666544295 2 C1305486145 Name: nameOrig, dtype: object *Figure 4*
Oldbalanceorg	Balance of the customer before the transaction. A float type variable.  df['oldbalanceOrg'].head(3) 0 170136.0 1 21249.0 2 181.0 *Figure 5*
Newbalanceorig	Balance of the customer after the transaction. A float type variable.  df['newbalanceOrig'].head(3) 0 160296.36 1 19384.72 2 0.00 Name: newbalanceOrig, dtype: float64
Namedest	Identifier for the transaction recipient. An object type variable.
Oldbalancedest	Initial balance of the recipient before the transaction. A float type variable.
Newbalancedest	Balance of the recipient after the transaction. A float type variable.
Isfraud	Indicator of whether the transaction is fraudulent. A binary variable. However, the type is considered as an int-type variable. This is the response variable.  df.isFraud.value_counts() isFraud 0 1047433 1 1142 Name: count, dtype: int64 *Figure 6*
Isflaggedfraud	Indicator of whether the transaction was flagged as fraudulent by the system. (will not be used for model)  isFlaggedFraud 0 1048575 Name: count, dtype: int64

ii. Data Preprocessing

- The dataset does not consist of any missing data or no duplicate values.

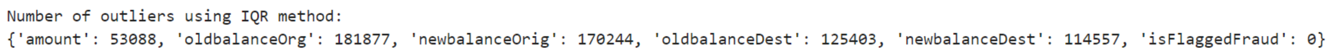


```
df.isna().sum()

step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

```
df.duplicated().sum()

0
```

Figure 7

- Handling outliers

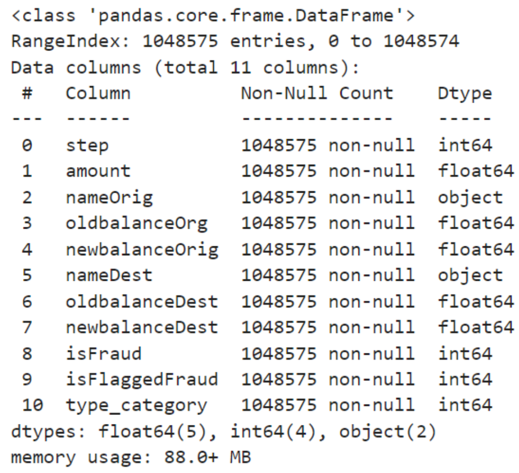


```
Number of outliers using IQR method:
{'amount': 53088, 'oldbalanceOrg': 181877, 'newbalanceOrig': 170244, 'oldbalanceDest': 125403, 'newbalanceDest': 114557, 'isFlaggedFraud': 0}
```

Figure 8

There are outliers in the data set that could be identified, by using the IQR method. However, the transaction amounts and available balances can be varied. Outliers in fraud data can often signify fraudulent transactions and mishandling them can result in a loss of valuable information or even inaccurate models. So, imputing them using median or mean without any knowledge, may cause a loss of valuable information. Therefore, machine learning algorithms that are robust to outliers, like decision trees will be used in future.

- Handling categorical variables.

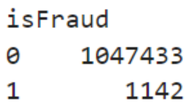


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 11 columns):
 #   Column          Non-Null Count    Dtype  
---  ------          --------------    -----  
 0   step            1048575 non-null  int64  
 1   amount          1048575 non-null  float64
 2   nameOrig        1048575 non-null  object 
 3   oldbalanceOrg   1048575 non-null  float64
 4   newbalanceOrig  1048575 non-null  float64
 5   nameDest        1048575 non-null  object 
 6   oldbalanceDest  1048575 non-null  float64
 7   newbalanceDest  1048575 non-null  float64
 8   isFraud         1048575 non-null  int64  
 9   isFlaggedFraud  1048575 non-null  int64  
 10  type_category   1048575 non-null  int64  
dtypes: float64(5), int64(4), object(2)
memory usage: 88.0+ MB
```

Figure 9

As some machine learning algorithms require numerical inputs only, the object variable ‘type’ can be converted to numerical categories. As we prefer use logistic regression model, then for the future purposes this conversion would be helpful.

- Handle the class imbalance



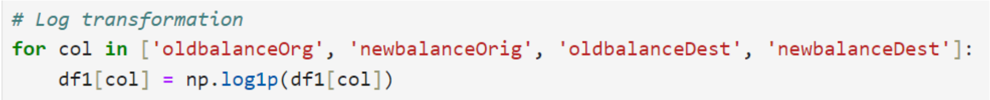
```
isFraud
0    1047433
1       1142
```

Figure 10

In fraud data it can be observed a huge class imbalance. So handling this class imbalance is particularly useful in scenarios like this, which can negatively impact the performance of many machine learning algorithms. So the SMOTE tomek (Synthetic Minority Over-sampling Technique) is a technique used to handle class imbalance in a dataset.

- Log transformation

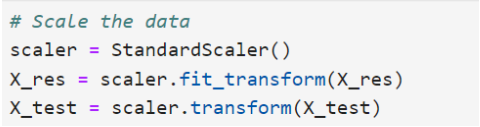
The features 'oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest', 'newbalanceDest', are highly skewed, the log transformation was performed to reduce skewness, stabilize variance, improve linearity, normalize the distribution, and reduce the impact of outliers, it helps to meet the assumptions of many machine learning algorithms, and enhance the performance.



```
# Log transformation
for col in ['oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest', 'newbalanceDest']:
    df1[col] = np.log1p(df1[col])
```

- Feature scaling

The data scaling involves transforming the data so that its features have comparable scales. This is particularly important for algorithms that are sensitive to the scale of the data. So this ensures that all features contribute equally to the final result, avoiding any domination by features with larger scales.

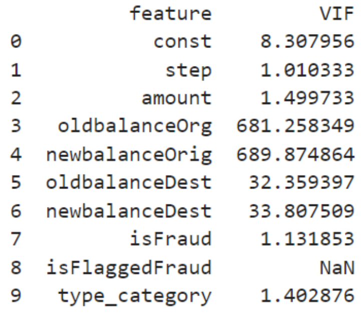


```
# Scale the data
scaler = StandardScaler()
X_res = scaler.fit_transform(X_res)
X_test = scaler.transform(X_test)
```

Figure 11

Using Z score normalization technique, the scaling was implemented to the data, before the model building.

- Dimensionality reduction.



```
          feature         VIF
0           const    8.307956
1            step    1.010333
2          amount    1.499733
3   oldbalanceOrg  681.258349
4  newbalanceOrig  689.874864
5  oldbalanceDest   32.359397
6  newbalanceDest   33.807509
7         isFraud    1.131853
8  isFlaggedFraud         NaN
9   type_category    1.402876
```

Figure 12 – VIF values

According to the Variance Inflation Factor, some variables indicate high multicollinearity. Therefore to address this issue, the Principal Component Analysis technique is used to transform the features into a set of linearly uncorrelated components.

Theory and Methodology

Fraud detection involves identifying fraud that happens in online payment platforms. The key challenges included in the dataset were, handling class imbalance, handling skewness, balancing the impact of different features that with different scales, choosing appropriate features, selecting suitable models, and avoiding overfitting etc. Here's a structured approach to tackle these challenges and techniques used:

1. Data Preprocessing

1.1 Handling Missing Values:

- Ensure the dataset has no missing values.

1.2 Exploratory data analysis:

- ➢ Non parametric methods like Mann-Whitney U test will be better performed on skewed data.

1.3 Log Transformation:

- The skewed features can be identified by the descriptive analysis and apply log transformation to those skewed features to reduce their skewness and stabilize the variance.

1.4 Scaling:

- Normalize features to a common scale (Standard Scaler). To ensure that all features contribute equally to the final result, avoiding any domination by features with larger scales.

1. 5Multicollinearity - Variance Inflation Factor (VIF):

- The occurrence of multicollinearity can be identified by the indication of high VIF values.

1.6 Principal Component Analysis (PCA):

- The multicollinearity can be addressed by using PCA to reduce dimensionality while retaining significant variance, which helps in speeding up the training process and reducing the complexity.

2. Handling Class Imbalance

2.1 SMOTE (Synthetic Minority Over-sampling Technique):

- SMOTE generates synthetic samples for the minority class by interpolating between existing minority instances, thereby balancing the class distribution.
-

2.2 Tomek Links:

- Tomek Links remove borderline examples that are likely to be misclassified, thereby cleaning the dataset after applying SMOTE.

2.3 SMOTE-Tomek:

- A combination of SMOTE and Tomek Links is used to oversample the minority class and clean the noisy instances, providing a balanced and clean dataset for training.

3. Model Building and Tuning

3.1 Choosing the Model:

- Start with a simple model like logistic regression that can address binary classification tasks. Further to handle complex datasets and their inherent feature importance measure, more robust models can be used (Decision trees, Random forest, XGBoost).

3.2 Regularization: To address the overfitting.

  - **Limiting Maximum Depth**: Restrict the depth of the Decision Tree, which controls how many splits it can make. This helps prevent the tree from growing too complex and overfitting.
  - **Pruning**: Pruning is a technique that removes nodes from the tree that do not provide significant predictive power, reducing the complexity and preventing overfitting.

- **Class Weight Adjustment**: Adjusting class_weight to balanced or balanced_subsample helps the model give more importance to the minority class.

4. Model Evaluation

4.1 Evaluation Metrics:

- Use appropriate metrics such as precision, recall, and F1-score to evaluate the model.

  Precision = TP / (TP +FP)

  Recall = TP / (TP +FN)

  F ratio = 2 * (Precision*Recall) / (Precision+Recall)

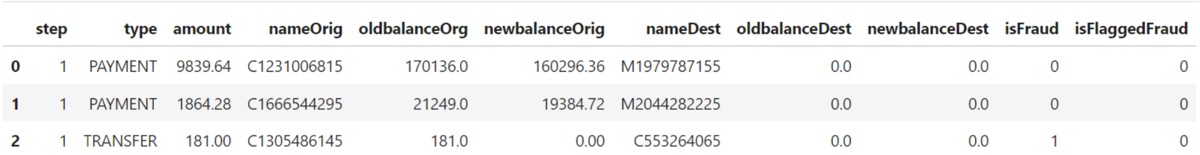
Where, TP = True Positives, FP = False Positives, FN = False Negatives

4.2 Confusion Matrix:

- Analyse the confusion matrix to understand the model’s performance on both the majority and minority classes.

Exploratory Data Analysis

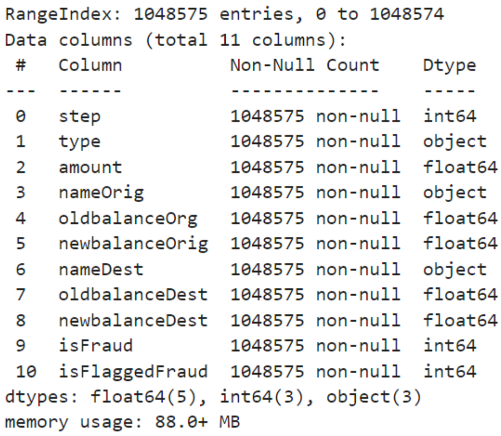
The dataset consist of 11 features, step, amount, nameOrig, OldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFraud and isFlaggedFraud. The response variable is 'isFraud'.



	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0

Figure 13 – Dataset preview(first 3 rows)

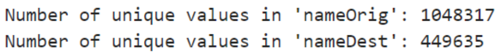
To successfully achieve the goal of 'building a predictive model to detect fraud in online payments', understanding and gain clear insights of data would be very useful.



```
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 11 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   step            1048575 non-null  int64
 1   type            1048575 non-null  object
 2   amount          1048575 non-null  float64
 3   nameOrig        1048575 non-null  object
 4   oldbalanceOrg   1048575 non-null  float64
 5   newbalanceOrig  1048575 non-null  float64
 6   nameDest        1048575 non-null  object
 7   oldbalanceDest  1048575 non-null  float64
 8   newbalanceDest  1048575 non-null  float64
 9   isFraud         1048575 non-null  int64
 10  isFlaggedFraud  1048575 non-null  int64
dtypes: float64(5), int64(3), object(3)
memory usage: 88.0+ MB
```

The dataset consist of total 1 048 575 entries. There are 3 int64 type variables, 5 float64 type variables and 3 object type variables. So in the process of model building categorical features can be converted to numerical and depending on the value each feature put on the model, it can be decided whether a particular feature is going to be used or dropped.

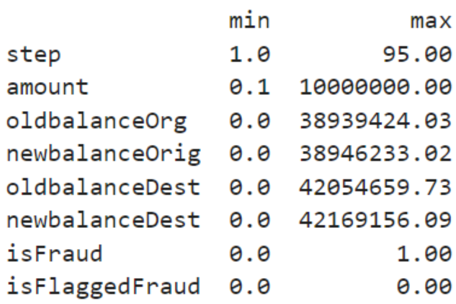
Figure 14 – Dataset Information



```
Number of unique values in 'nameOrig': 1048317
Number of unique values in 'nameDest': 449635
```

From the dataset total 1 048 575 entries, 1 048 317 are unique customers and 449 635 are unique recipients. From those unique customers, 0.11% have faced real fraud. And 0.25% of unique recipients have faced frauds.

When considering the minimum and maximum values of the data, the highest amount that any customer made on a transaction is $ 10,000 000. The maximum amount that any customer has as the balance before the transaction is $38, 939 424 and the maximum balance after the transaction is $38, 946 233.



```
                min          max
step            1.0        95.00
amount          0.1  10000000.00
oldbalanceOrg   0.0  38939424.03
newbalanceOrig  0.0  38946233.02
oldbalanceDest  0.0  42054659.73
newbalanceDest  0.0  42169156.09
isFraud         0.0         1.00
isFlaggedFraud  0.0         0.00
```

Figure 15 - Minimum and maximum values in numerical features

The data has been collected between 1 to 95 steps(i.e 1st hour to the 95th hour from the starting time). Among them 18th, 19th hours are the time intervals that highest number of transactions happened. However, in every time interval, at least one transaction has happened.

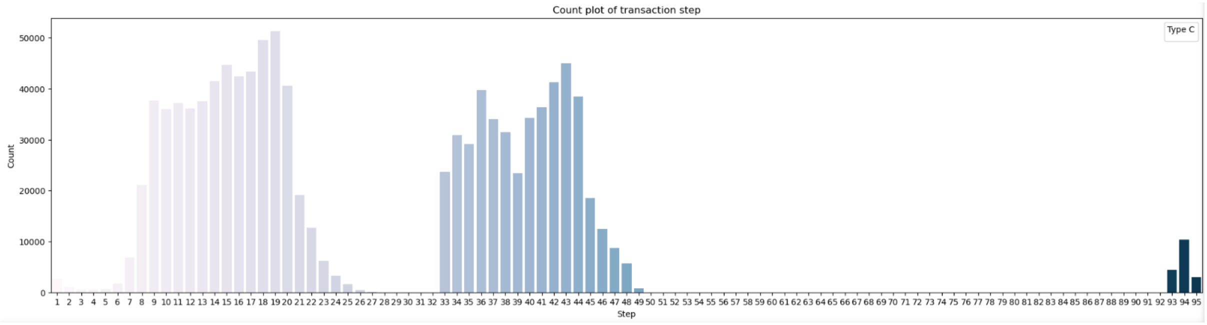


Figure 16

Among those transactions following is how fraudulent transactions have been distributed. Out of the total 1,142 frauds, this is how each fraudulent transaction occurred. It can be observed that at least one fraud has happened in each time step. However, the maximum number of fraudulent transactions has happened in the 66th time step. Overall, a roughly cyclic pattern, recurring every 20 to 25 steps, can be observed in the presence of fraud.

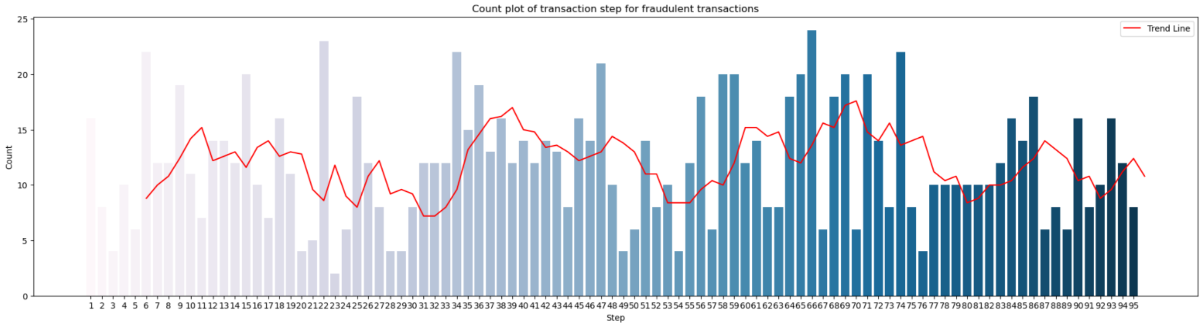
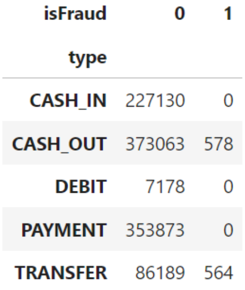


Figure 18

There are 5 types of transactions that has happened. Among them cash-outs are the most occurred transaction type and it is about 373 641 in total. 354 873 are payment transactions. Lowest of 7178 are debits.
Among them the fraud has happened only in cash outs and transfers. No fraud case has reported during cash in, debit or a payment.



isFraud	0	1
type		
CASH_IN	227130	0
CASH_OUT	373063	578
DEBIT	7178	0
PAYMENT	353873	0
TRANSFER	86189	564

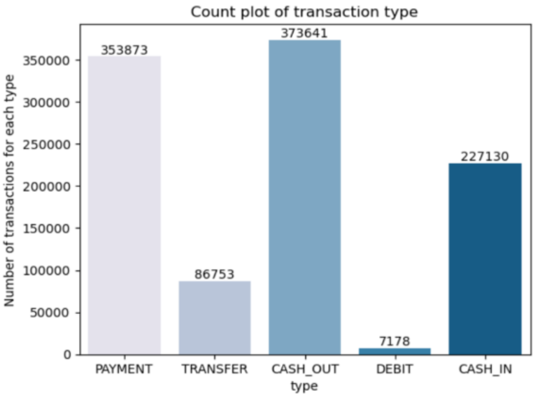
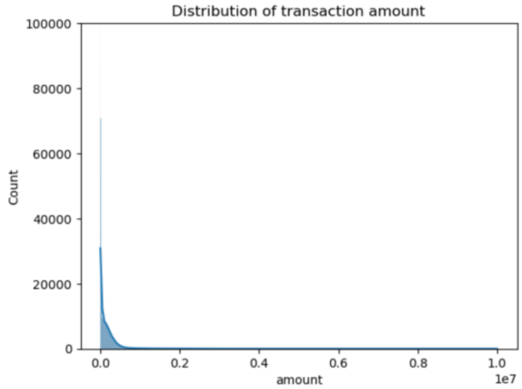
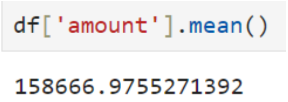


Figure 17

However, most transaction amounts are below $ 200,000. Transactions are $ 158 667 on average.

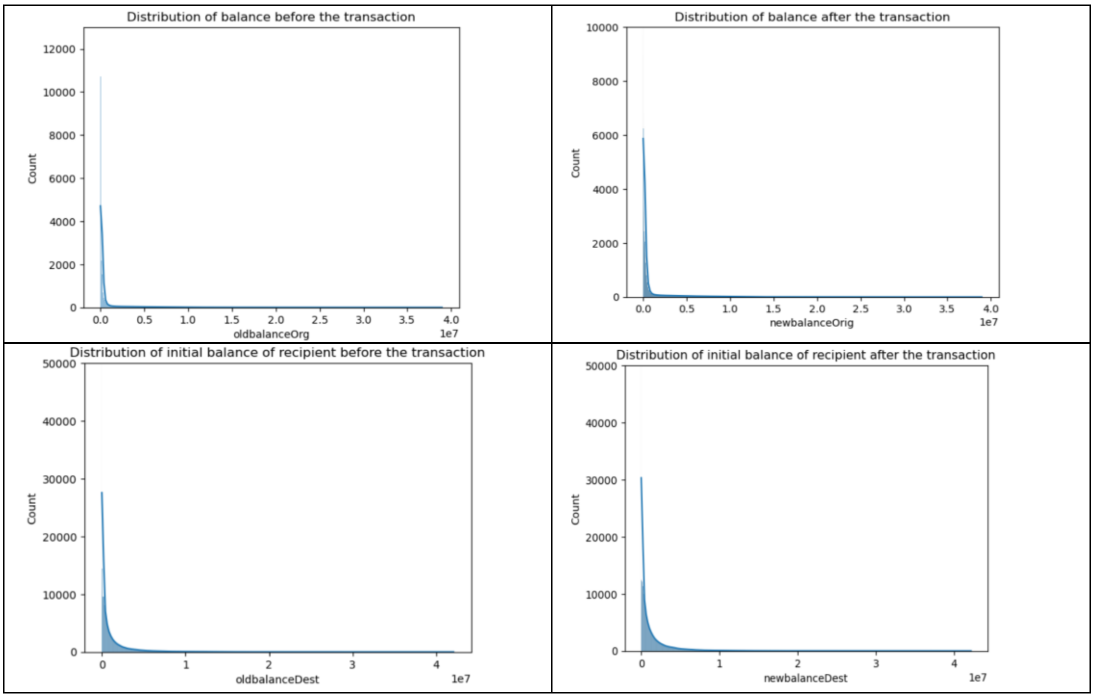




```
df['amount'].mean()

158666.9755271392
```

Figure 19

Most of the balances before and after transaction in customers' accounts and recipient accounts, are below $ 1, 000 000.



The data contains 1 048 575 total number of transaction. From that only 1142 are fraudulent. That is 0.1% from the whole, while rest 99.9% transactions are non-fraudulent.

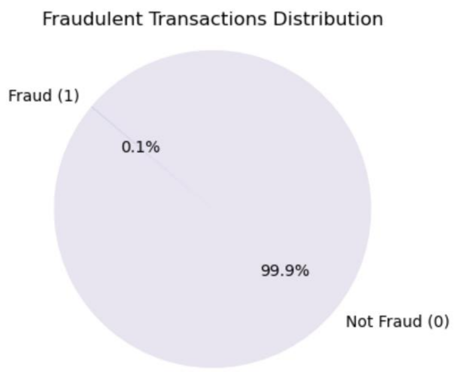
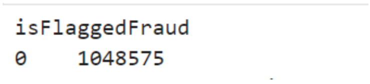


Figure 20



```
isFlaggedFraud
0    1048575
```

But the existing fraud detection model has not flagged any fraudulent transaction in advance.

When considering the time steps, it can clearly be observed that transactions occurring between the 28th and 32nd time steps, as well as between the 50th and 92nd time steps, are highly probable to be fraudulent. It is crucial to note that nearly all transactions during these time intervals are fraudulent.

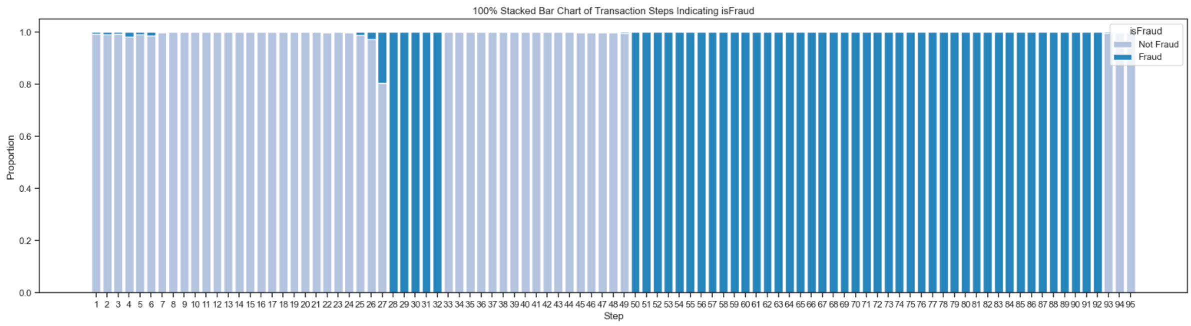
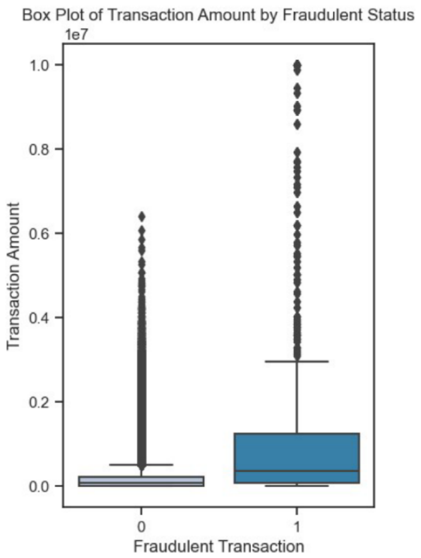


Figure 21

In most cases fraud has happened during transacting higher amounts through the platform.



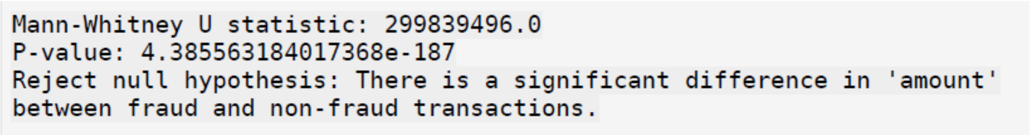
So, it indicates a tendency to occur a fraud when transaction is a very high amount. 75% of the fraudulent transactions are between $ 0 and $ 2000000. But in the case of non-fraud 75% of the transactions are below $ 1000000.

The highest amount that it is also a fraudulent transaction is, between $ 9 000 000 and $ 10 000 000.

The highest amount that it a non-fraudulent transaction is, between $ 6 000 000 and $ 7 000 000.

Figure 22

From the Mann-Whitney U statistics also, it is confirmed that there is a significance difference in amount between fraud and non-fraud transaction under the 5% significance level. In most cases, fraud has happened during the account balance of the customer, before the transaction is, higher.



```
Mann-Whitney U statistic: 299839496.0
P-value: 4.385563184017368e-187
Reject null hypothesis: There is a significant difference in 'amount'
between fraud and non-fraud transactions.
```

Furthermore, there is an indication of a tendency for fraud to occur when the account balance of the customer is very high. Seventy-five percent of the fraudulent transactions happen when the customer has an account balance between $0 and $2,500,000. In the fraudulent cases, customers' average account balance is also very high, while the maximum account balance of the non-fraudulent cases is still less than that average.

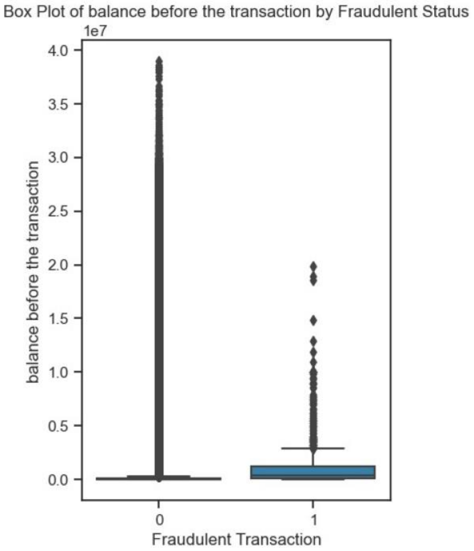
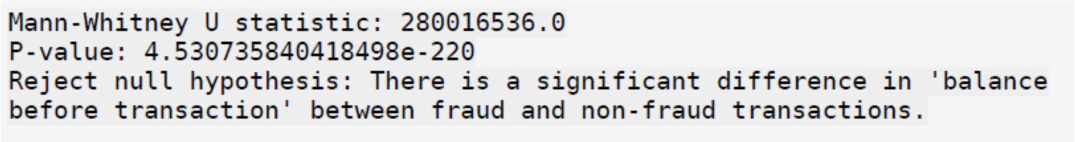


Figure 23



```
Mann-Whitney U statistic: 280016536.0
P-value: 4.530735840418498e-220
Reject null hypothesis: There is a significant difference in 'balance
before transaction' between fraud and non-fraud transactions.
```

As the statistical test confirms there is a significant difference in the account balance before transactions, between fraud and non-fraud transaction.

When considering the relationships between features,

There is a noticeable pattern between amount and oldbalanceOrg, and newbalanceOrig indicating transactions where the amount might be closely related to the original and new balances. amount vs. oldbalanceDest and newbalanceDest shows a scattered pattern, indicating no strong relationship. There is a strong linear relationship between oldbalanceOrg vs. newbalanceOrig and oldbalanceDest vs. newbalanceDest, indicating that most transactions directly transfer the balance without much alteration.

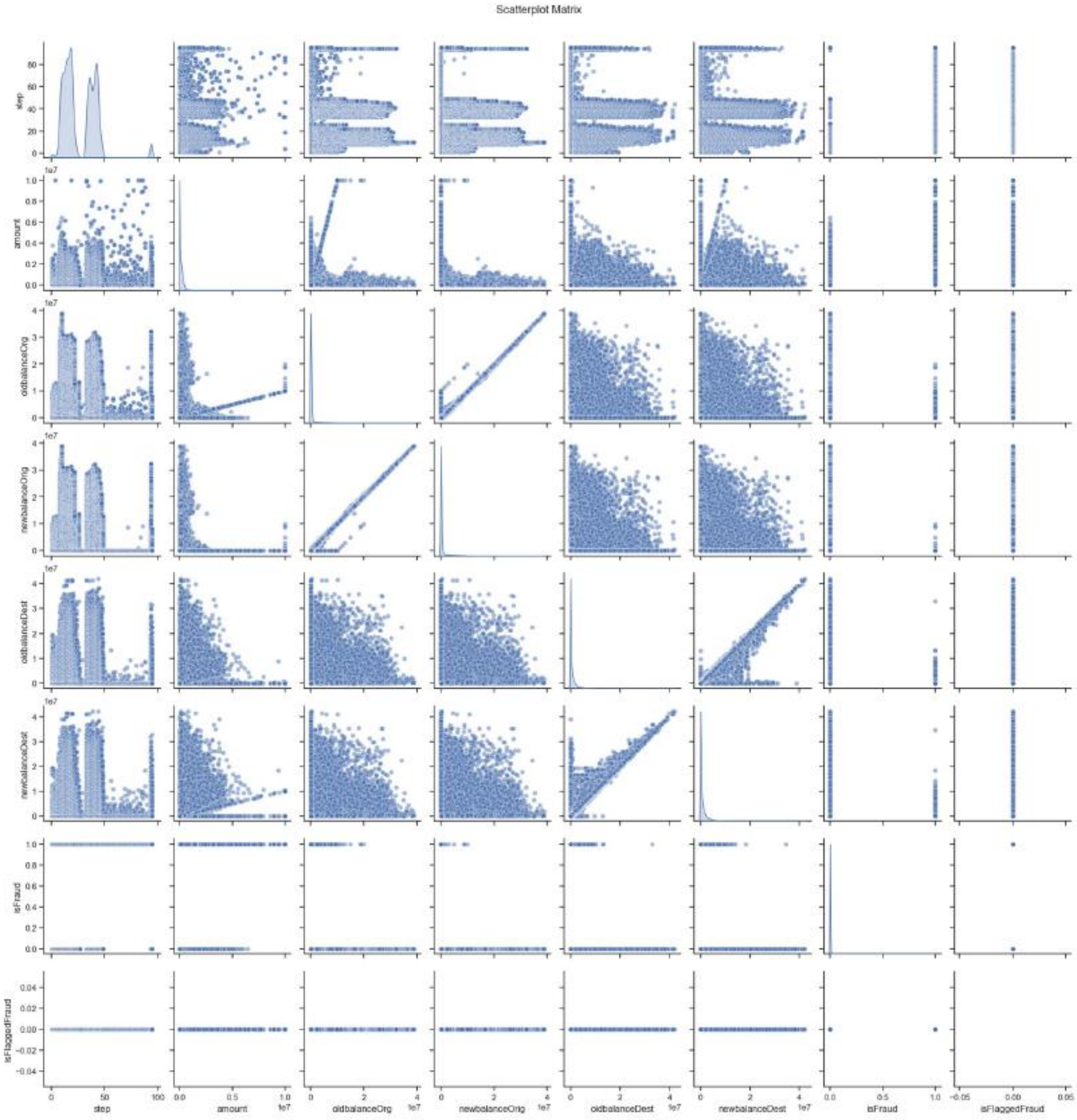
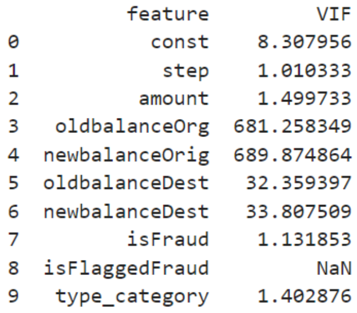


Figure 24



```
         feature         VIF
0          const    8.307956
1           step    1.010333
2         amount    1.499733
3  oldbalanceOrg  681.258349
4 newbalanceOrig  689.874864
5 oldbalanceDest   32.359397
6 newbalanceDest   33.807509
7        isFraud    1.131853
8 isFlaggedFraud         NaN
9  type_category    1.402876
```

According to the VIF values also it can be seen that VIFs are higher in oldbalanceOrg, newbalanceOrig, oldbalanceDest and newbalanceDest variables. So that could be an alarm of existence of relationships between those exploratory variables with others.

From the heatmap also there is a clear indication of strong correlation between oldbalanceOrg, newbalanceOrig, oldbalanceDest and newbalanceDest variables.

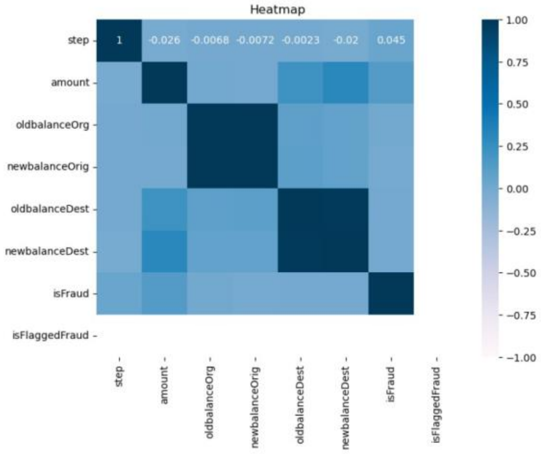


Figure 25

After a fraudlent transaction occurred, the balance change in the accounts of customers, are as follows. Most of the accounts has showed $ 500 000 balance change after a fraud. And this balance change ranges between $ 100 000 and $ 10 000 000.

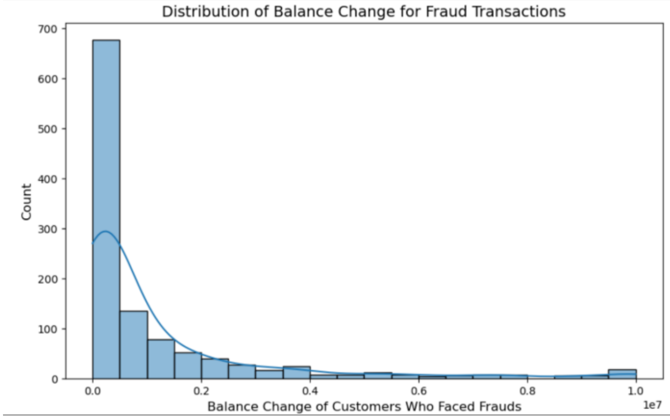


Figure 26

Advanced Analysis

As the main part of the project, the model building will be started. To stabilise the variance and make the data more normally distributed, log transformation can be applied to several key features: ‘amount’, 'oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest', and 'newbalanceDest' as the distributions are showing high skewness to the right. The log transformation helps in reducing skewness and making patterns in the data more evident.

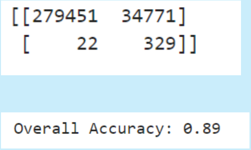
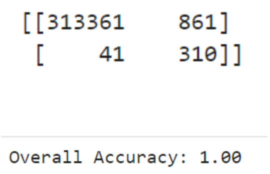
Then the features (X) and the target variable (y) can be separated from the dataset. 'isFraud’ as the target variable, while all other columns were used as features. To evaluate the model performance, the dataset was split into training and testing sets. That allows for assessing how well the model generalizes to unseen data.

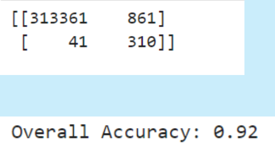
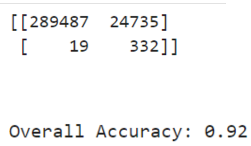
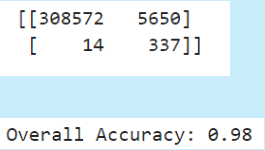
As there was a class imbalance in data, to address that, the SMOTETomek technique can be applied to the training set.

Then in order to ensure all features contribute equally to the model and to improve convergence during training, the data can be scaled using StandardScaler. This step standardizes the features by removing the mean and scaling to unit variance.

Later PCA can be applied to reduce the dimensionality of the data. This helps in reducing the computational complexity and removing multicollinearity.

Upon completion of above different machine learning algorithms (classification) can be initialize and use to fit the model.

Model	Reason to choose	Model evaluation (Confusion matrix and Accuracy)	Interpretation and limitations
Logistic Regression	The response variable is binary. Logistic regression is the simplest and most commonly used binary classification model.	 [[279451 34771] [22 329]] Overall Accuracy: 0.89	From 314 222 actual non-frauds, 279 451 has predicted as non-frauds. And 34 771 incorrectly predicted as frauds. From 351 actual frauds, 329 were predicted correctly. 22 of actual frauds were incorrectly predicted as non-fraud(So the type I error is high). Model accuracy is 89%
Decision Tree Classifier	More robust classification model.	 [[313361 861] [41 310]] Overall Accuracy: 1.00	From 314 222 actual non-frauds, 313 361 has predicted as non-frauds. And 861 were incorrectly predicted as frauds. From 351 of actual frauds, 310 have been predicted correctly. 41 of actual frauds were incorrectly predicted as non-

			fraud (Type I error is higher than the logistic model.). Model accuracy is 100%. This can be an indication of overfitting.
Decision Tree Classifier with Limited Maximum Tree Depth	As the decision tree model overfit the data, then it will be not accurate on unseen data. So regularization techniques can be used.	[[313361 861] [41 310]] Overall Accuracy: 0.92	From 314 222 actual non-frauds, 313 361 has predicted as non-frauds. And 861 were incorrectly predicted as frauds. From 351 of actual frauds, 310 have been predicted correctly. 41 of actual frauds were incorrectly predicted as non-fraud(Type I error is high). Model accuracy is 92%.
Decision Tree Classifier with Pruning Parameters	As the decision tree model overfit the data, then it will be not accurate on unseen data. So regularization techniques can be used.	[[289487 24735] [19 332]] Overall Accuracy: 0.92	From 314 222 actual non-frauds, 289 487 has predicted as non-frauds. And 24735 incorrectly predicted as frauds. From 351 of actual frauds, 332 have been predicted correctly. 19 of actual frauds were incorrectly predicted as non-fraud (Type I error has decreased). Model accuracy is 92%. Overall model accuracy should be improved.
Random Forest Classifier	After decision tree regularization, overall model accuracy decreased. As the random forest is more robust model to outliers and noise in the data and also a good model for reducing the risk to overfitting.	[[308572 5650] [14 337]] Overall Accuracy: 0.98	From 314 222 actual non-frauds, 308 572 has predicted as non-frauds. And 5650 were incorrectly predicted as frauds. From 351 actual frauds, 337 were predicted correctly. 14 of actual frauds were incorrectly predicted as non-fraud. Model accuracy is 98%. Type I error is comparatively lower. Overall model accuracy is good.

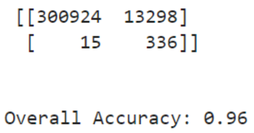
XGB Classifier	The in-built scale_pos_weight parameter in XGBoost can be used to adjust the balance of positive and negative weights, helping the model to pay more attention to the minority class. Using XGBoost on a balanced dataset (after SMOTE) can yield better results.	 [[300924 13298] [15 336]] Overall Accuracy: 0.96	From 314 222 actual non-frauds, 300 924 has predicted as non-frauds. And 13 298 incorrectly predicted as frauds. From 351 of actual frauds, 336 have been predicted correctly. 15 of the actual frauds were incorrectly predicted as non-fraud. Model accuracy is 96%. Type 1 and type 2 errors became larger.

Table 2

The random forest classifier is comparatively best in the performance. By averaging multiple decision trees, the random forest reduces the risk of overfitting, which is a common issue with individual decision trees. Random forests are less sensitive to noise in the data and can handle missing values and outliers well(As we considered every data point as a real transaction). As it is non-parametric, it does not assume a specific distribution of the data, making it flexible for various types of data. So to detect fraud in online transactions, random forest shows the best fit.

Discussion

- Key insights derived from the dataset can guide the development of an effective fraud detection model. Following are some notable observations that could observe from the data.
- The dataset, consisting of 1,048,575 individual transactions and 11 features:- step, amount, nameOrig, OldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFraud, and isFlaggedFraud.
- Only 0.11% of unique customers and 0.25% of unique recipients experienced fraud, indicating a highly imbalanced dataset.
- Out of 1,048,575 transactions, only 1,142 (0.1%) were fraudulent, highlighting the rarity of fraud cases in the dataset.
- Fraudulent transactions often involve higher amounts of online transactions. And the all the fraud transactions are either a transfer or a cash-out. No fraud reported in cash-ins, debits or payments.
- Customers' maximum account balance before a transaction is $38 939 424, and the maximum balance after is $38 946 233. Higher account balances before transactions are often associated with fraudulent activities.
- The average balance of fraudulent accounts is significantly higher compared to non-fraudulent ones.
- Transactions occur over 95 steps (hours), with peaks at the 18th and 19th hours. Fraudulent transactions show a cyclic pattern, recurring every 20 to 25 steps, with a peak at the 66th time step.
- Specific time intervals, such as the 28th to 32nd and 50th to 92nd steps, show a higher probability of fraudulent transactions.
- Random forest classifier is best fit to detect fraud in online transactions.

Conclusion

- The analysis of the dataset provides valuable insights for building a predictive model for online payment fraud detection.
- Key factors such as transaction amount, account balance, transaction type, and temporal patterns are crucial in distinguishing fraudulent transactions from legitimate ones.
- The imbalance in the dataset necessitates careful handling, potentially through techniques like SMOTE for oversampling. Feature selection and multicollinearity considerations are also important for model accuracy and interpretability.
- By leveraging these insights, a robust fraud detection model like random forest, can be developed, capable of identifying fraudulent transactions with high accuracy and minimizing false positives.
- After a thorough exploratory data analysis, a robust fraud detection model built with 98% present accuracy.
- This model will play a crucial role in enhancing the security and reliability of online payment systems.