# FINANCIAL FRAUD DETECTION ANALYSIS

A project Report Submitted in partial fulfilment of the requirements for the course

Applied Machine Learning

Submitted by

Akash Parmar

Under the esteemed guidance of

Dr. Li Liao

Associate Professor

University Of Delaware

# ABSTRACT

With the advancement of technology and its presence in every field , the security issues have also increased parallel . Like all the fields , the field of finance is not untouched by technology . In today's world where technology is helping to open new horizons in this field , the financial frauds have become an increasing problem. Payment related frauds have become a major challenge and the solution to this problem is technology. Many researches have shown that machine learning can provide an optimal solution to this problem.

In this project I am going to apply various supervised machine learning techniques to detect payment fraud in the available dataset. Since payment fraud accounts for a very small portion of the payments which creates an imbalance therefore I will be exploring different techniques to overcome the imbalance problem.

Through exploratory data analysis I aim to differentiate between the fraudulent payments from the non-fraudulent payments . The analysis was carried out using

various classification algorithms such as Logistic Regression , BaggingClassifier and RandomForest.

# INDEX

| | |
|---|---|
| Title | i |
| Abstract | ii |
| Contents | ii |

## Contents

| **Heading** | **Pg.No** |
|---|---|
| Introduction | 4 |
| Literature review | 5 |
| Data Extraction | 6 |
| Exploratory Data Analysis | 8 |
| Experiment and Evaluation | 15 |
| Handling Class Imbalance | 17 |
| Results | 21 |
| Discussion | 24 |
| References | 28 |

# INTRODUCTION

The global digital payment market size was valued at USD 81.03 billion in 2022 and is expected to expand at a compound annual growth rate (CAGR) of 20.8% from 2023 to 2030. The global digital payments transactions were valued at over USD 8 trillion in 2022. The tremendous growth in the digital payment space has signaled its world wide acceptance but what also comes with this advancement is the threat of financial frauds in the digital payment space.

All the banks and financial institutions have recognized the vulnerability of digital payments and therefore are focusing towards making digital payments a more secure mode by utilizing the technology. They have  dedicated teams working in the domain of cyber-security and analysts working towards achieving the aim of more secure digital payments . In order to be better equipped to tackle the cybercrime cases, it is crucial to investigate the methods for resolving the issue of spotting fraudulent entries/transactions in vast amounts of data.

# Literature Review

Financial fraud detection is an important topic due to its relevance across the industries. Therefore there is a considerable amount of literature that shows the amount of work done in this field. Some of the references of the work are given below:

- Ali, A.; Abd Razak, S.; Othman, S.H.; Eisa, T.A.E.; Al-Dhaqm, A.; Nasser, M.; Elhassan, T.; Elshafie, H.; Saif, A. Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review. *Appl. Sci.* **2022**, *12*, 9637.
- F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," in IEEE Access, vol. 10, pp. 39700-39715, 2022, doi: 10.1109/ACCESS.2022.3166891.
- Albashrawi, Mousa. (2016). Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015. Journal of Data Science. 14. 553-570. 10.6339/JDS.201607_14(3).0010.
- Kuangyi Gu. 2022. Deep Learning Techniques in Financial Fraud Detection. In Proceedings of the 7th International Conference on Cyber Security and Information Engineering (ICCSIE '22).
- Leevy, J.L., Khoshgoftaar, T.M., Bauder, R.A. *et al.* A survey on addressing high-class imbalance in big data. *J Big Data* 5, 42 (2018).

Most of the literature available on fraud detection shows that there is now extensive usage of deep learning in order to create a better predictive model. Albashrawi et al., (2016) present a systematic review of the most used methods in financial fraud detection.

# Data Extraction

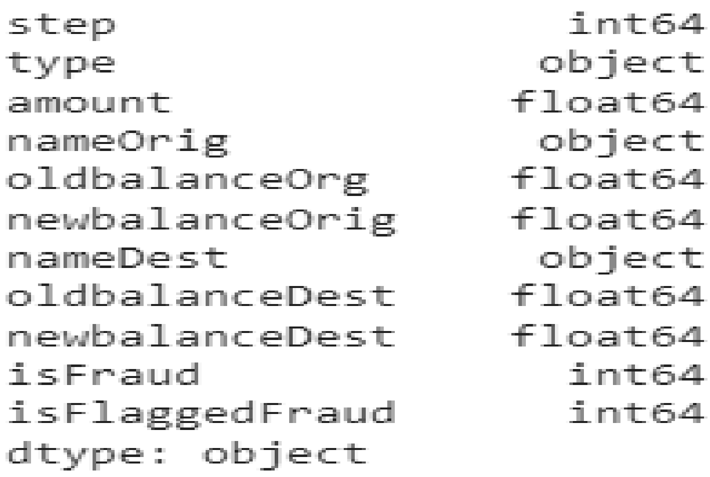
### Dataset Description:

The dataset used in this project is a synthetic dataset generated using the simulator called PaySim . PaySim uses aggregated data from the private dataset to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behavior to later evaluate the performance of fraud detection methods.

This dataset has over 2.5 million rows and contains 11 columns describing various features of the transactions. The dataset contains the isFraud column that is the target column.

| Columns | Detailed description of columns |
|---|---|
| step | It maps a unit of time in the real world. In this case 1 step is 1 hour of time. |
| type | It denotes different transaction types . There are 5 types in this dataset given by CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER. |
| amount | amount of the transaction in local currency. |
| nameOrig | customer who started the transaction |
| oldbalanceOrg | initial balance before the transaction |
| newbalanceOrig | new balance after the transaction |
| nameDest | customer who is the recipient of the transaction |
| oldbalanceDest | initial balance recipient before the transaction. |
| newbalanceDest | new balance recipient after the transaction. |
| isFraud | It denotes whether the transaction is fraud or not . It consists of 2 values 0 & 1 where 0 denotes a transaction is valid and 1 denotes a fraud transaction. |

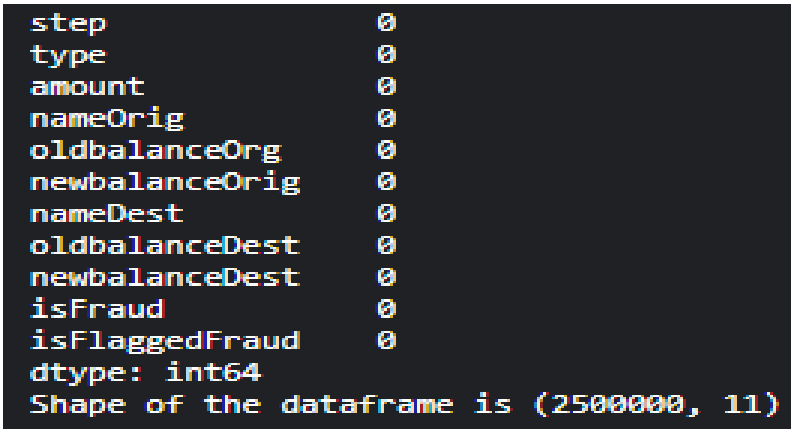
The dataset contains both numeric and categorical types of data as shown below .



```
step                 int64
type                object
amount             float64
nameOrig            object
oldbalanceOrg      float64
newbalanceOrig     float64
nameDest            object
oldbalanceDest     float64
newbalanceDest     float64
isFraud              int64
isFlaggedFraud       int64
dtype: object
```

## Data Cleaning :

The datasets usually have null values , inconsistent data and duplicate records therefore it becomes important for the data to undergo a preprocessing stage where the data is cleaned and pre-processed inorder to make data consistent and ready for the data analysis.

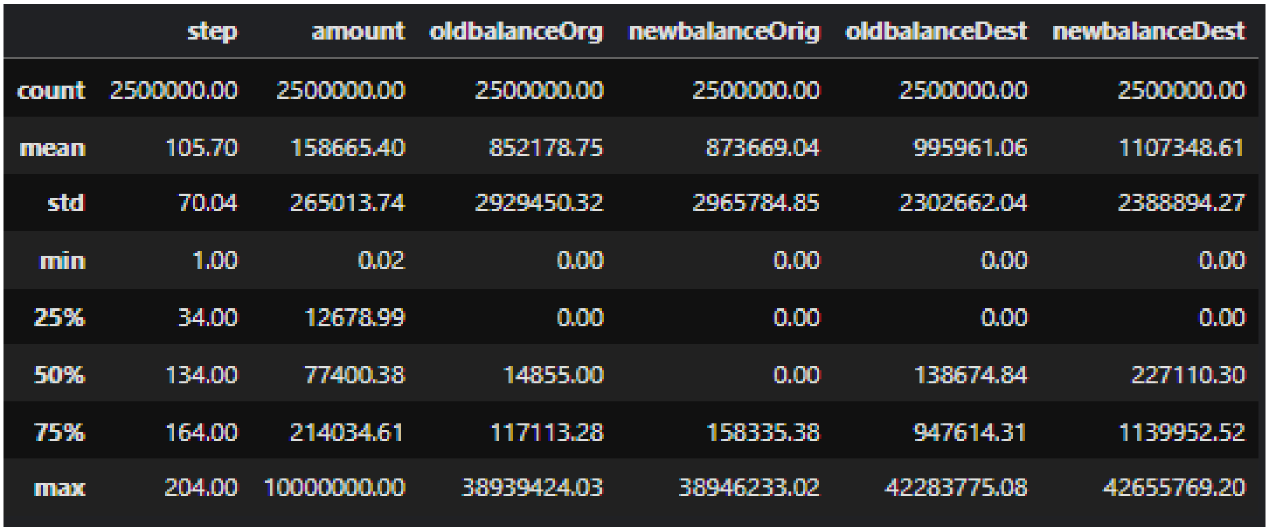


```
step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
Shape of the dataframe is (2500000, 11)
```

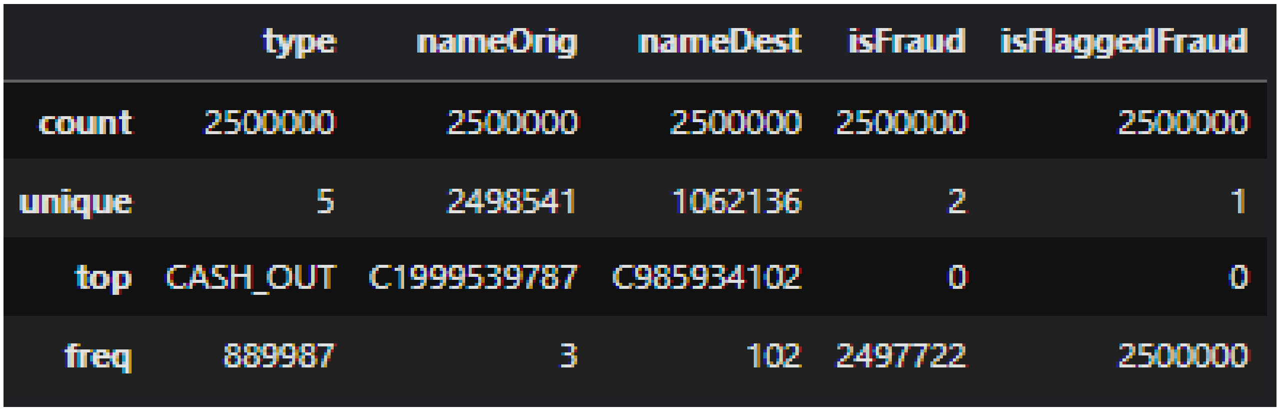
# Exploratory Data Analysis

**Summary of data** :

Under this section along with count , mean and standard deviation , a 5 point summary of the dataset is presented .The summary for the numerical columns and the categorical columns is presented separately.



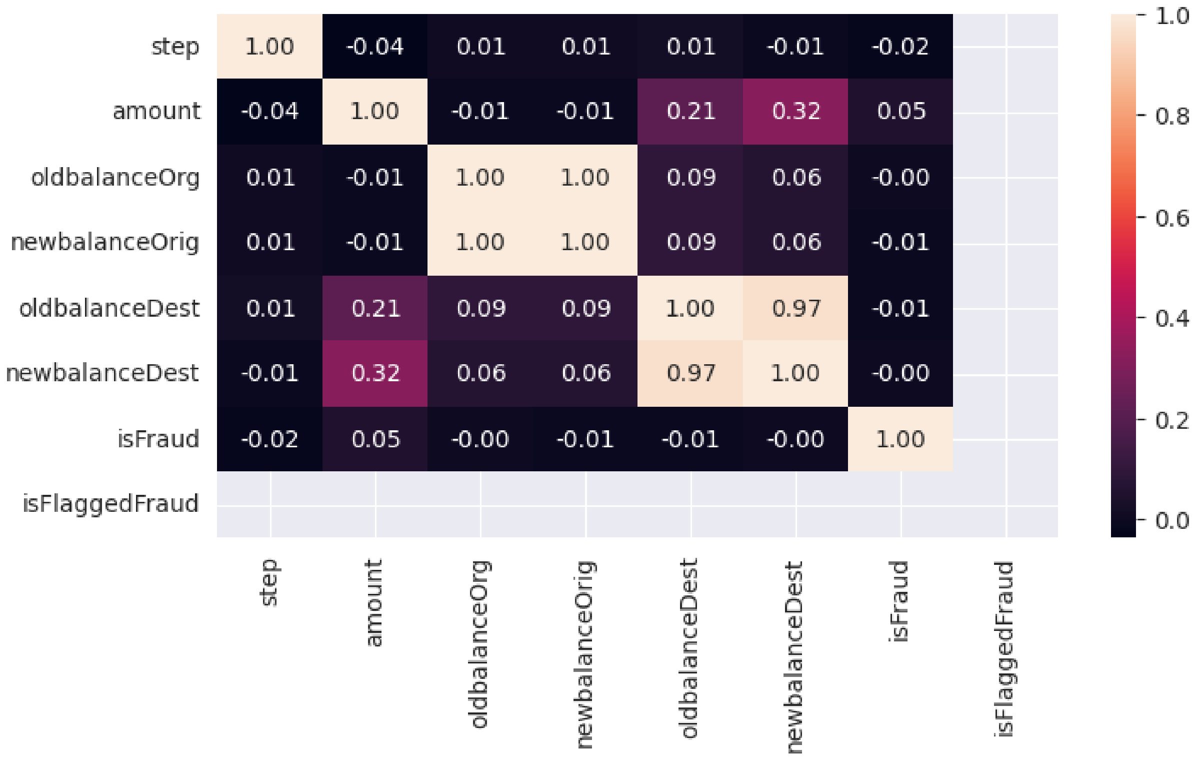
| | step | amount | oldbalanceOrg | newbalanceOrig | oldbalanceDest | newbalanceDest |
|---|---|---|---|---|---|---|
| count | 2500000.00 | 2500000.00 | 2500000.00 | 2500000.00 | 2500000.00 | 2500000.00 |
| mean | 105.70 | 158665.40 | 852178.75 | 873669.04 | 995961.06 | 1107348.61 |
| std | 70.04 | 265013.74 | 2929450.32 | 2965784.85 | 2302662.04 | 2388894.27 |
| min | 1.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 |
| 25% | 34.00 | 12678.99 | 0.00 | 0.00 | 0.00 | 0.00 |
| 50% | 134.00 | 77400.38 | 14855.00 | 0.00 | 138674.84 | 227110.30 |
| 75% | 164.00 | 214034.61 | 117113.28 | 158335.38 | 947614.31 | 1139952.52 |
| max | 204.00 | 10000000.00 | 38939424.03 | 38946233.02 | 42283775.08 | 42655769.20 |



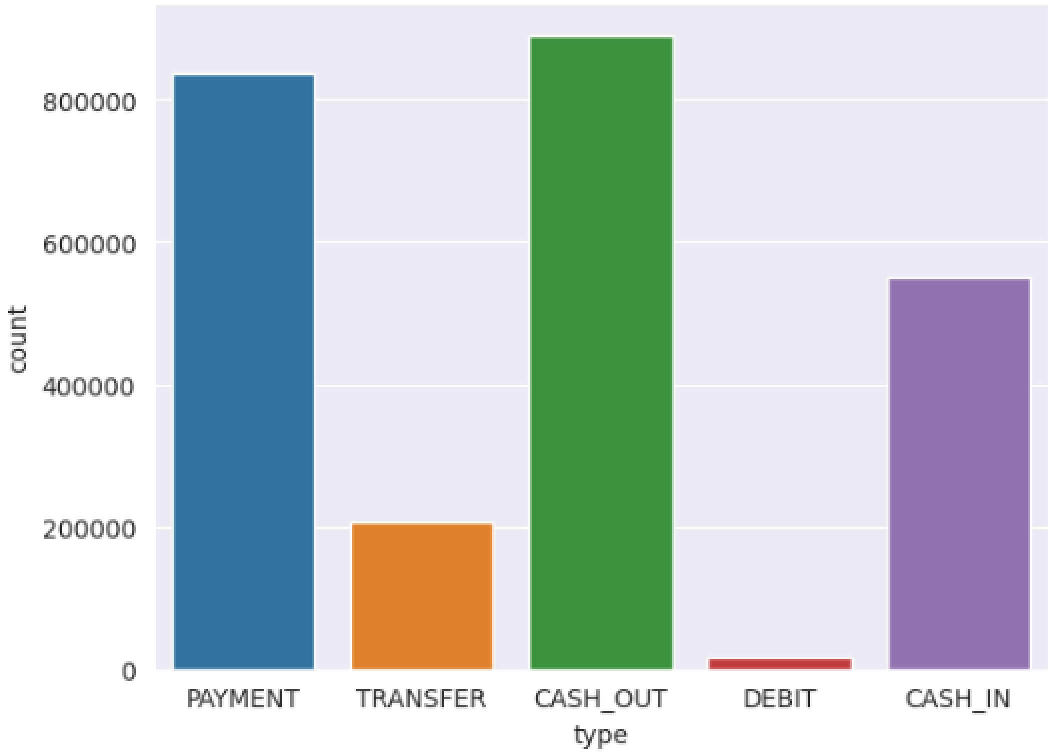
| | type | nameOrig | nameDest | isFraud | isFlaggedFraud |
|---|---|---|---|---|---|
| count | 2500000 | 2500000 | 2500000 | 2500000 | 2500000 |
| unique | 5 | 2498541 | 1062136 | 2 | 1 |
| top | CASH_OUT | C1999539787 | C985934102 | 0 | 0 |
| freq | 889987 | 3 | 102 | 2497722 | 2500000 |

**Insights from data** :

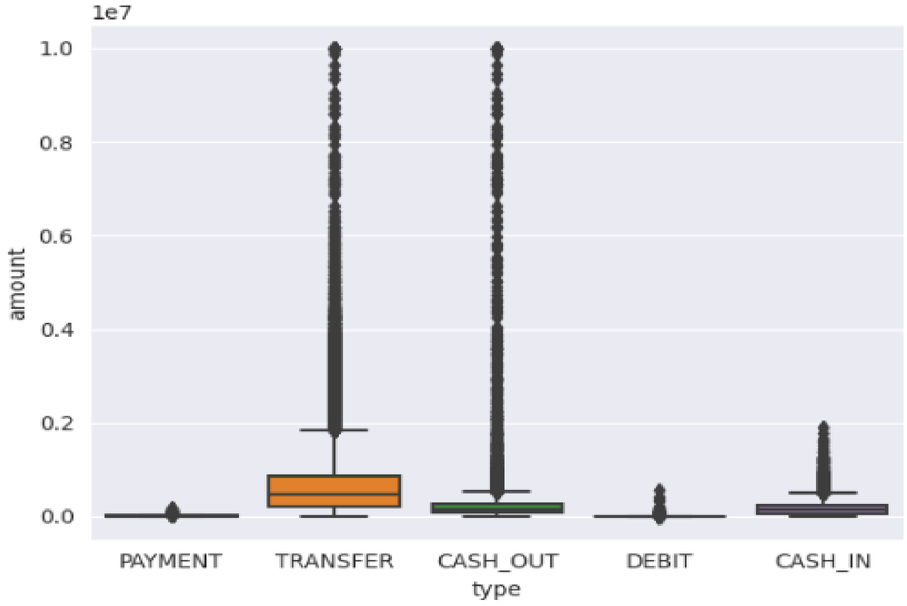
In this dataset I have used 2,500,000 rows for the analysis. I have evaluated the relations between the columns through the correlation matrix and found that there is no significant relation between any of the columns and the target column.



There are 5 modes of payments as described in the barchart . Among the types of payment , Cash_out type of transaction leads the usage among the customers followed by the payment type while the debit type of transaction is the least used by the customers as per the barchart . This is displayed in the figure below.

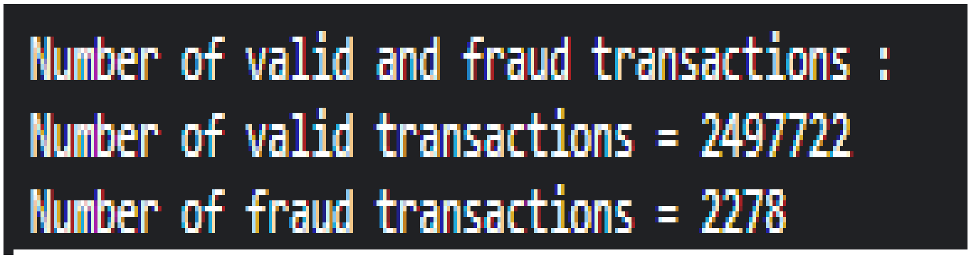


The boxplot given below describes the amount involved in the transactions in the given dataset. The transaction of type transfer involves the large amount followed by the cash out type. It can also be seen that in both the leading types the amount involved is very large , almost million dollars,  compared to the other types.

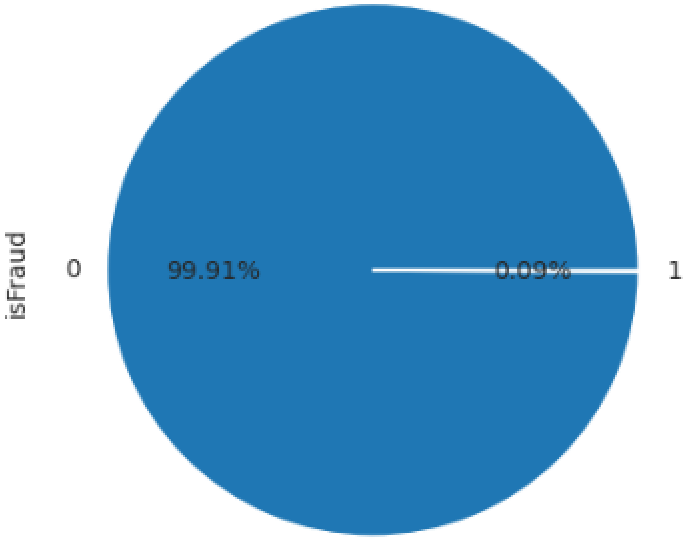


**Class Imbalance check**:

As we know the dataset contains the fraudulent and non-fraudulent transactions therefore a class imbalance check is a must. I am using 25,00,000 rows of transaction data . The image below describes the breakdown of the transactions into valid and fraudulent types.



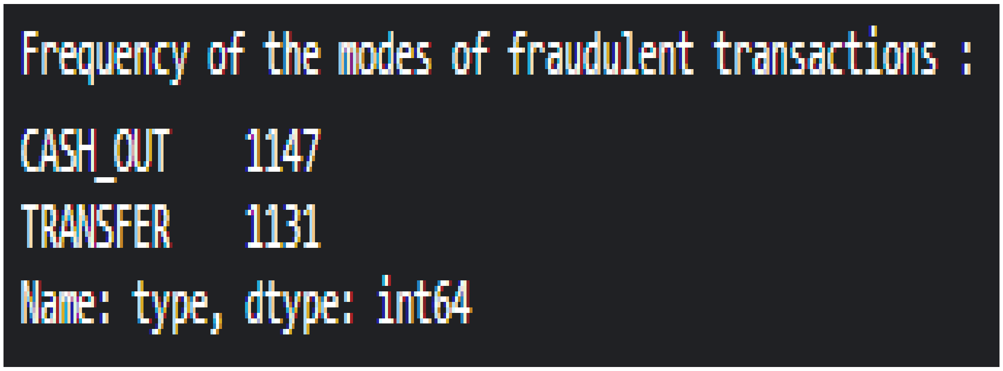
```
Number of valid and fraud transactions :
Number of valid transactions = 2497722
Number of fraud transactions = 2278
```



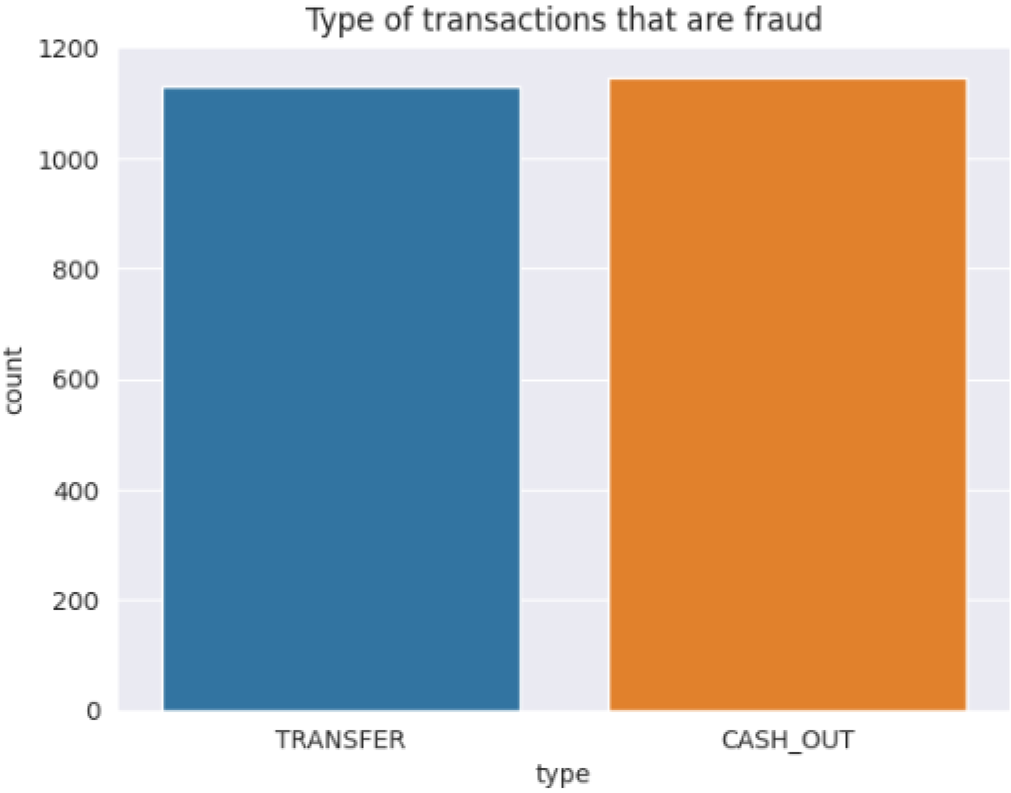
The non-fraudulent transactions account for 99.91% and the fraudulent transactions account for 0.09% of total transactions.

Types of Transactions:

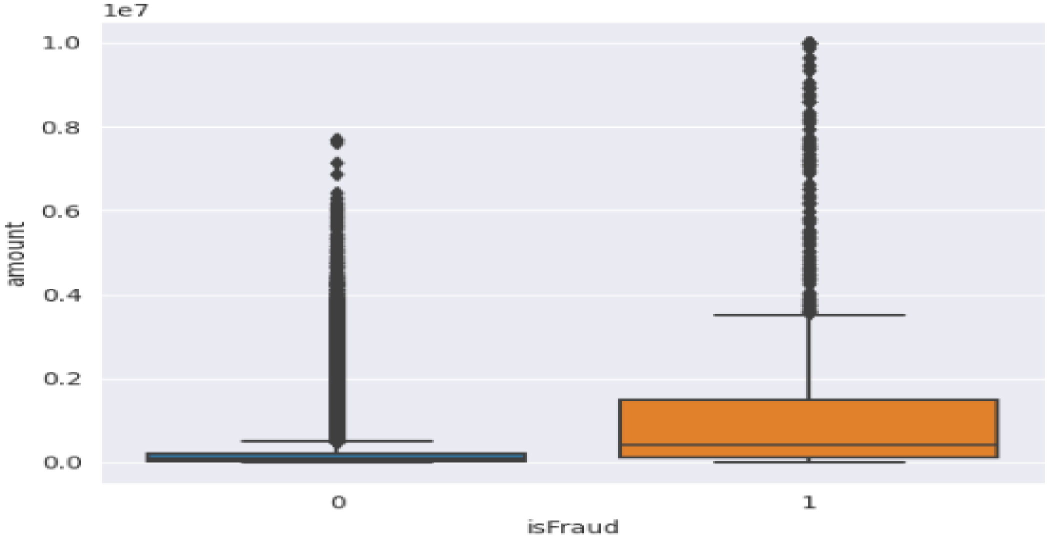
Among all transaction types, transfer and cash out modes of transactions are the only one contributing towards the fraudulent transactions. The cash out type is contributing more as compared to the transfer type. Below is the number of fraudulent transactions that both the types have

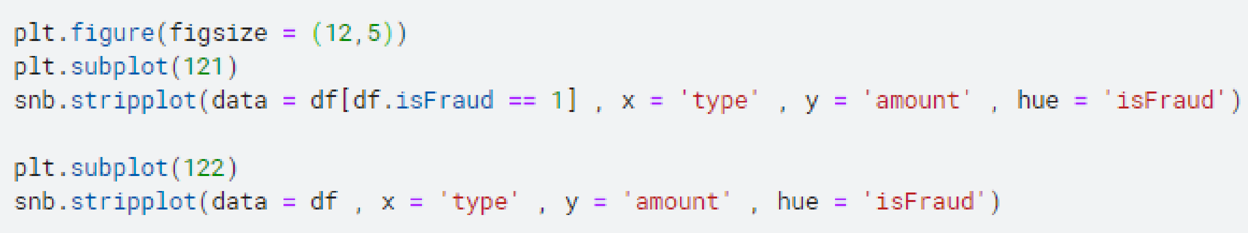


```
Frequency of the modes of fraudulent transactions :
CASH_OUT    1147
TRANSFER    1131
Name: type, dtype: int64
```



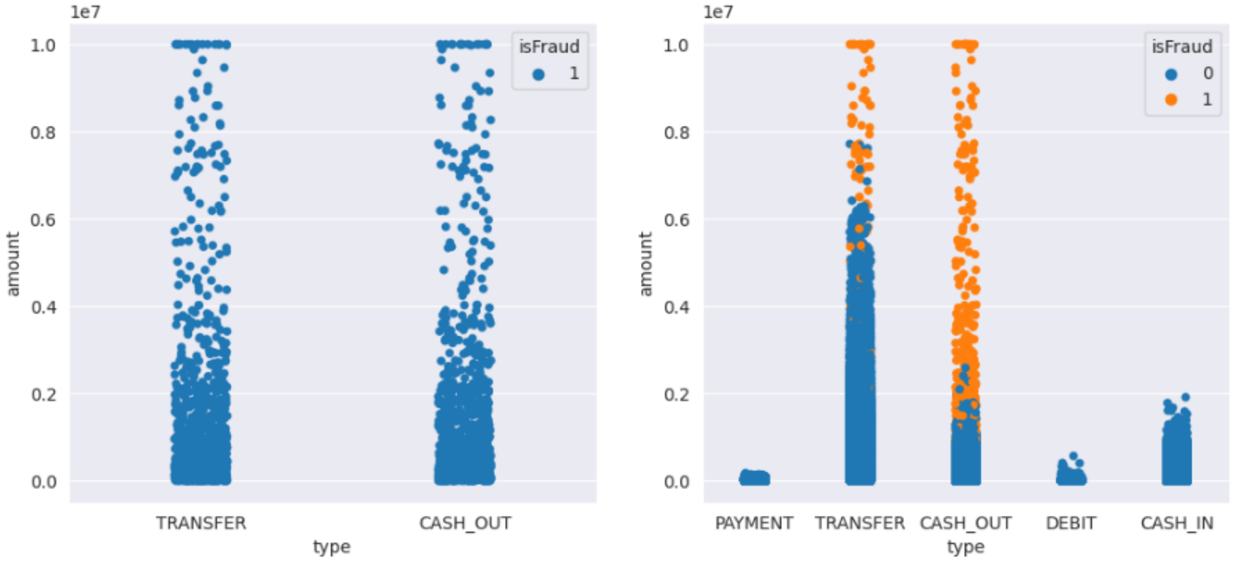
In terms of transactions it can be seen that only transfer and cashout types are involved so now we will look in terms of the amount involved in the transactions.





```
plt.figure(figsize = (12,5))
plt.subplot(121)
snb.stripplot(data = df[df.isFraud == 1] , x = 'type' , y = 'amount' , hue = 'isFraud')

plt.subplot(122)
snb.stripplot(data = df , x = 'type' , y = 'amount' , hue = 'isFraud')
```

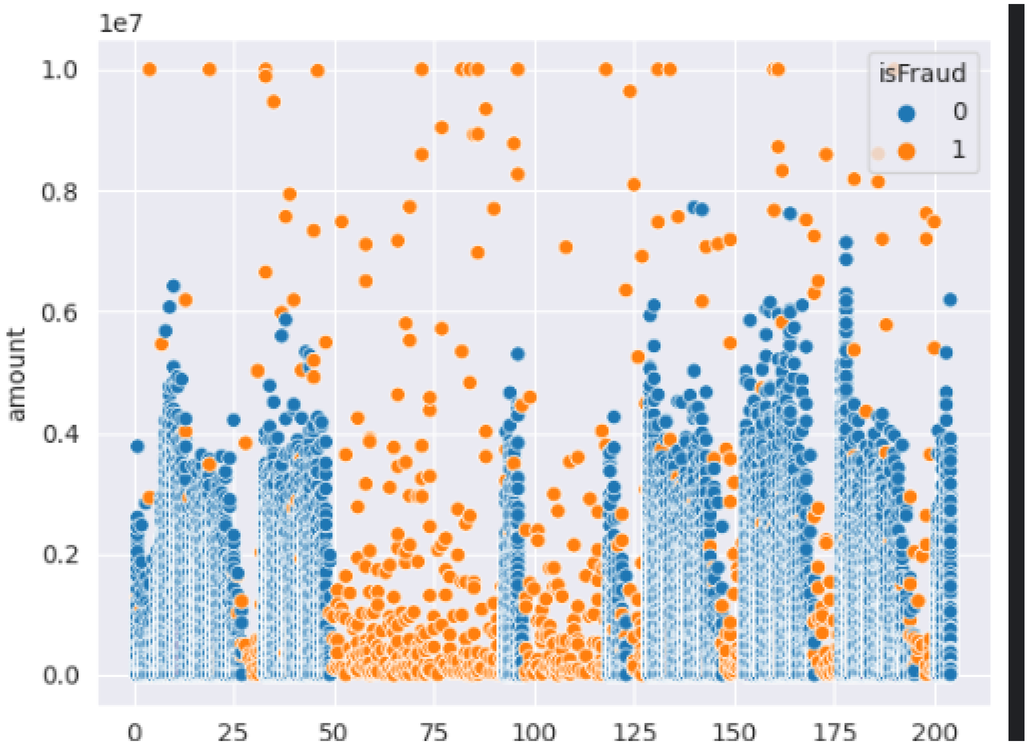


**Only Fraudulent transactions**

**Both fraudulent and**

**non-fraudulent transactions**

In the above plots it can be seen that the amount involved in the fraudulent transactions is higher as compared to the non-fraudulent transactions when the type is either transfer or cash out. There cannot be a boundary point identified in the amount that can be used to differentiate between the fraudulent and non-fraudulent transactions.

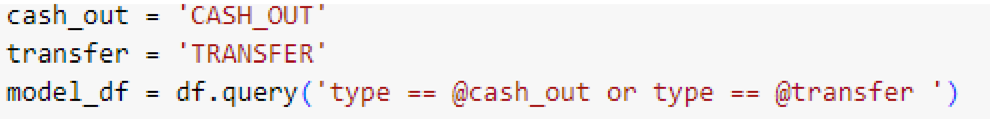


The above plot is between steps and the transaction amount and it is clearly visible that non-fraudulent transactions follow some pattern where such transactions are repeated after a particular step but for fraudulent transactions no such pattern is being followed. Such transactions are spread throughout the plot and definitely have higher amounts involved.

# Experiment and Evaluation

### Preparing dataset for modeling:

From the above analysis it was drawn that fraudulent transactions are limited to the transfer and cash out types therefore our modeling part will include all the transactions related to these two types.



```
cash_out = 'CASH_OUT'
transfer = 'TRANSFER'
model_df = df.query('type == @cash_out or type == @transfer ')
```

The new dataset has **1097217** rows and **11** columns .

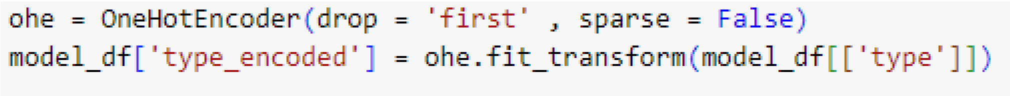
The columns in the dataset are given below:

```
Index(['step', 'type', 'amount', 'nameOrig', 'oldbalanceOrg',
'newbalanceOrig', 'nameDest', 'oldbalanceDest', 'newbalanceDest',
'isFraud', 'isFlaggedFraud'], dtype='object')
```

Out of all the columns 'nameOrig' and 'nameDest' are dropped.

### Encoding categorical data:

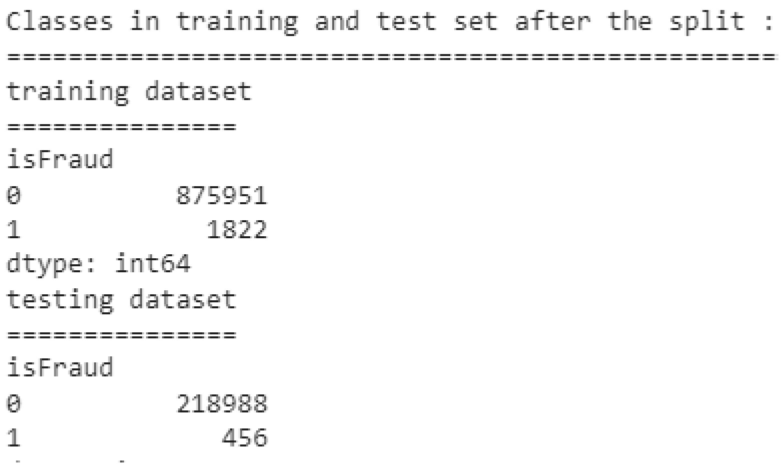
In the  dataset column type is a categorical column therefore One hot encoding is used to encode the categorical column.



```
ohe = OneHotEncoder(drop = 'first' , sparse = False)
model_df['type_encoded'] = ohe.fit_transform(model_df[['type']])
```

### Train and Test split:

For all the models, the train and test split are 80% and 20% for training and testing respectively. Even the stratification is also applied in the split which means that the ratio of both the classes in the target variable is same for both train and test data. After following the above plan of data split we get the data distribution described below.



```
Classes in training and test set after the split :
==================================================
training dataset
===============
isFraud
0         875951
1           1822
dtype: int64
testing dataset
===============
isFraud
0         218988
1            456
```

The models used here are Logistic Regression , Random Forest Classifier, Bagging Classifier. The Models are trained using data which are both randomly over-sampled and under-sampled with an 80:20 split of the data. The models are evaluated on , precision, recall , f1-score and confusion matrix.

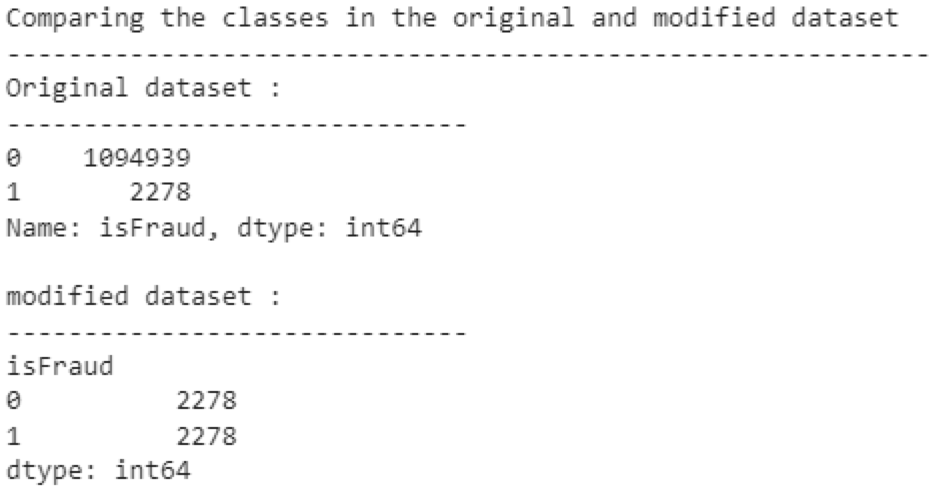
# Handling Class Imbalance

Class Imbalance can be handled in the following ways:

### A. Under Sampling

Under sampling refers to a technique of reducing the number of observations in the majority class to balance the class distribution with the minority class. It involves removing some of the data points in the majority class until the number of observations in both classes becomes comparable.

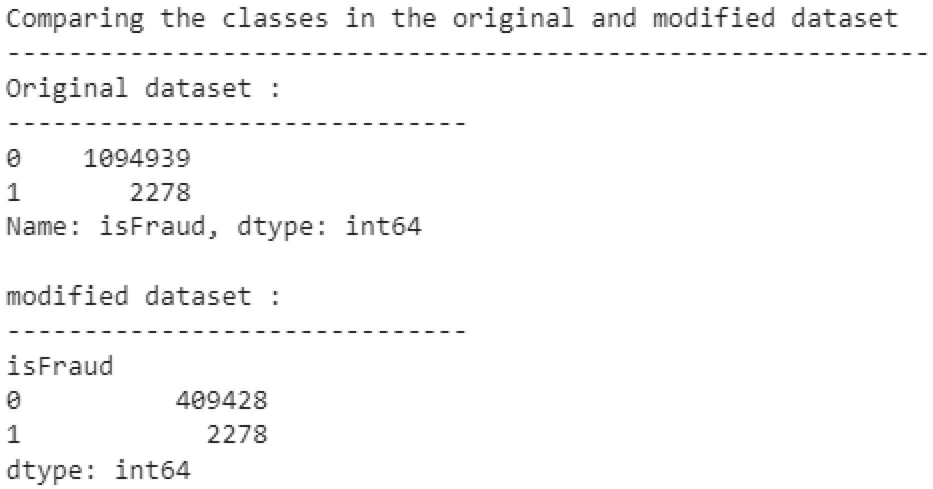
- **Random Under sampling :** From the imblearn library , RandomUndersampler is used to generate the under-sampled values. It reduces the number of majority class records to the number of minority class records.



```
Comparing the classes in the original and modified dataset
------------------------------------------------------------
Original dataset :
------------------------------
0    1094939
1       2278
Name: isFraud, dtype: int64

modified dataset :
------------------------------
isFraud
0          2278
1          2278
dtype: int64
```

- **OneSided Selection for Undersampling** : It is an undersampling technique that combines two other techniques Tomek Links and the Condensed Nearest Neighbor (CNN) Rule. Tomek Links are ambiguous points on the class boundary and are identified and removed in the majority class. The CNN method is then used to remove redundant examples from the majority class that are far from the decision boundary. In other words CNN for short seeks for a subset of a collection of samples that results in no loss in model performance, also referred to as a minimum consistent set.



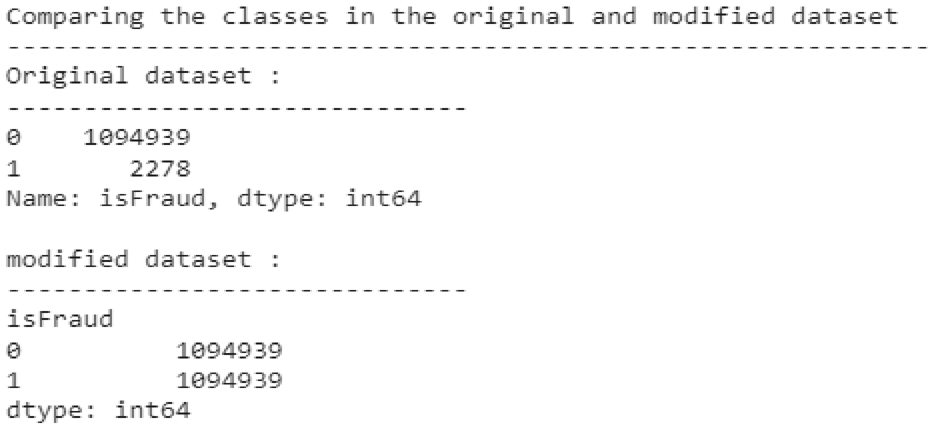
```
Comparing the classes in the original and modified dataset
------------------------------------------------------------
Original dataset :
------------------------------
0    1094939
1       2278
Name: isFraud, dtype: int64

modified dataset :
------------------------------
isFraud
0          409428
1            2278
dtype: int64
```

**B. Over Sampling**

Over sampling is a method of increasing the number of instances in the minority class by creating additional copies of the existing data points. It is a suitable approach in cases where there is insufficient data available to build a model that can accurately capture the minority class.

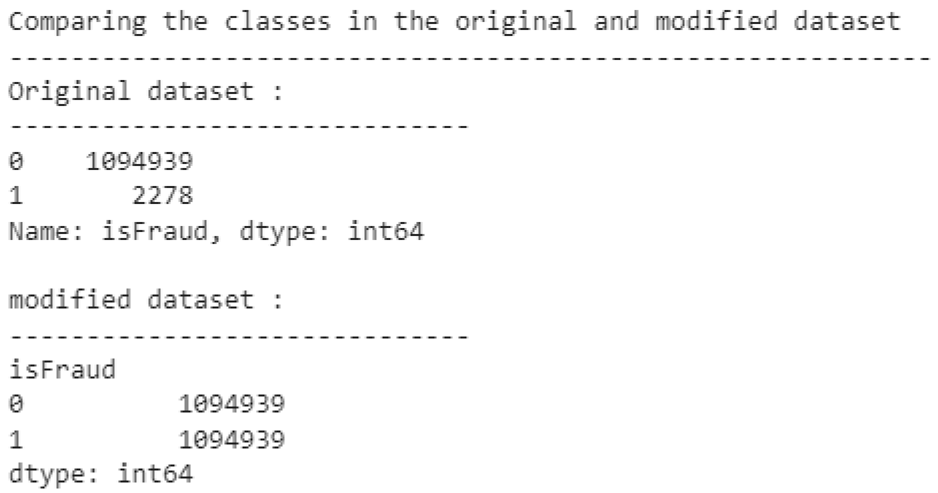
- **Random Under Sampling** : From the imblearn library , RandomOversampler is used to generate the sample records . It increases the number of minority class  records to the number of majority class records.



```
Comparing the classes in the original and modified dataset
------------------------------------------------------------
Original dataset :
------------------------------
0    1094939
1       2278
Name: isFraud, dtype: int64

modified dataset :
------------------------------
isFraud
0          1094939
1          1094939
dtype: int64
```

- **SMOTE :** SMOTE is the most popular over sampling method that is used widely**.** SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample as a point along that line.



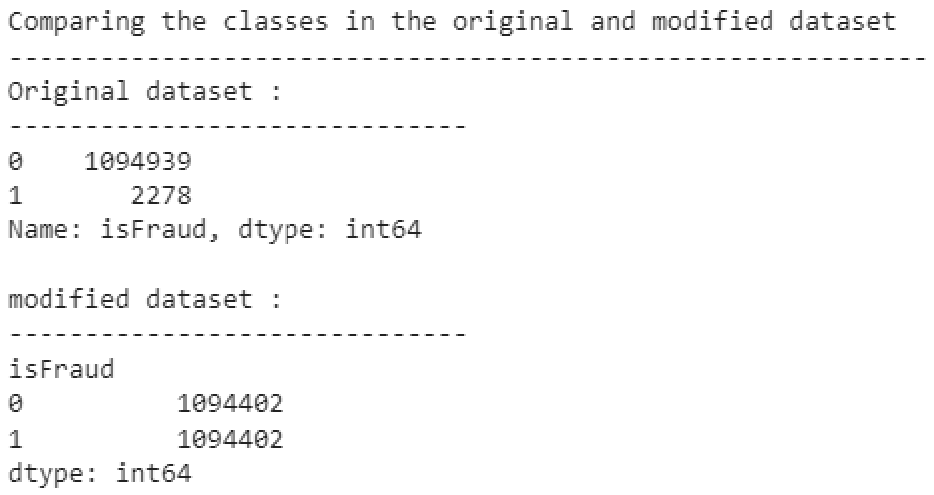
```
Comparing the classes in the original and modified dataset
------------------------------------------------------------
Original dataset :
------------------------------
0    1094939
1       2278
Name: isFraud, dtype: int64

modified dataset :
------------------------------
isFraud
0          1094939
1          1094939
dtype: int64
```

## C. Combining Oversampling and Undersampling

Apart from the oversampling and undersampling the other method to tackle the problem of the dataset imbalance is to combine both oversampling and the undersampling techniques.

- **Combining oversampling and undersampling using SMOTE and Tomek links :** SMOTE is an oversampling technique where instances of minority classes are increased to match the number of the majority class. Tomek Links are ambiguous points on the class boundary and are identified and removed in the majority class. It is an undersampling technique . Combining both can help to balance the dataset.



```
Comparing the classes in the original and modified dataset
------------------------------------------------------------
Original dataset :
------------------------------
0    1094939
1       2278
Name: isFraud, dtype: int64

modified dataset :
------------------------------
isFraud
0          1094402
1          1094402
dtype: int64
```

# RESULTS

By taking into consideration the imbalance problem , 3 different techniques were used namely , under sampling , oversampling and combination of both . For classification purposes the 3 different models were trained with the datasets on which above mentioned techniques were applied. The results obtained are presented below:

- **Evaluation for Imbalanced Dataset:**

| **Metrics** | **Logistic Regression** | **Random Forest Classifier** | **Bagging Classifier** |
|---|---|---|---|
| **Precision** | 0.46710526315789475 | 0.7478070175438597 | 0.7631578947368421 |
| **Recall** | 0.6068376068376068 | 0.9941690962099126 | 0.9329758713136729 |
| **F1-score** | 0.5278810408921932 | 0.853566958698373 | 0.8395657418576598 |

- **Evaluation for Undersampled data set:**
  - **Random Under Sampling**

| **Metrics** | **Logistic Regression** | **Random Forest Classifier** | **Bagging Classifier** |
|---|---|---|---|
| **Precision** | 0.9605263157894737 | 0.9736842105263158 | 0.9868421052631579 |
| **Recall** | 0.9319148936170213 | 0.9801324503311258 | 0.9782608695652174 |
| **F1-score** | 0.9460043196544277 | 0.976897689768977 | 0.982532751091703 |

- **One Sided Selection method:**

| **Metrics** | **Logistic Regression** | **Random Forest Classifier** | **Bagging Classifier** |
|---|---|---|---|
| **Precision** | 0.36622807017543857 | 0.743421052631579 | 0.7456140350877193 |
| **Recall** | 0.6162361623616236 | 0.9854651162790697 | 0.918918918918919 |
| **F1-score** | 0.45942228335625857 | 0.8475 | 0.8232445520581113 |

- **Evaluation of Over Sampled Dataset:**
    - **Random Over Sampling**

| **Metrics** | **Logistic Regression** | **Random Forest Classifier** | **Bagging Classifier** |
|---|---|---|---|
| **Precision** | 0.9611942206878916 | 1.0 | 1.0 |
| **Recall** | 0.9270318597010455 | 0.9999360736431631 | 0.9997397794973636 |
| **F1-score** | 0.9438040022777919 | 0.9999680357999041 | 0.9998698728178015 |

- **SMOTE :**

| **Metrics** | **Logistic Regression** | **Random Forest Classifier** | **Bagging Classifier** |
|---|---|---|---|
| **Precision** | 0.933197252817506 | 0.9998675726523828 | 0.999566186275047 |
| **Recall** | 0.9416988078945306 | 0.9989187758900709 | 0.9984127056526836 |
| **F1-score** | 0.9374287555705403 | 0.9993929490807515 | 0.9989891129979988 |

- **Evaluation of dataset where data is balanced by combination of undersampling and oversampling**

| Metrics | Logistic Regression | Random Forest Classifier | Bagging Classifier |
|---|---|---|---|
| **Precision** | 0.9332209460632476 | 0.999894922471058 | 0.9995431411785131 |
| **Recall** | 0.9422262608743784 | 0.9988681535842564 | 0.9983709267464623 |
| **F1-score** | 0.9377019831068675 | 0.9993812743009657 | 0.9989566900821636 |

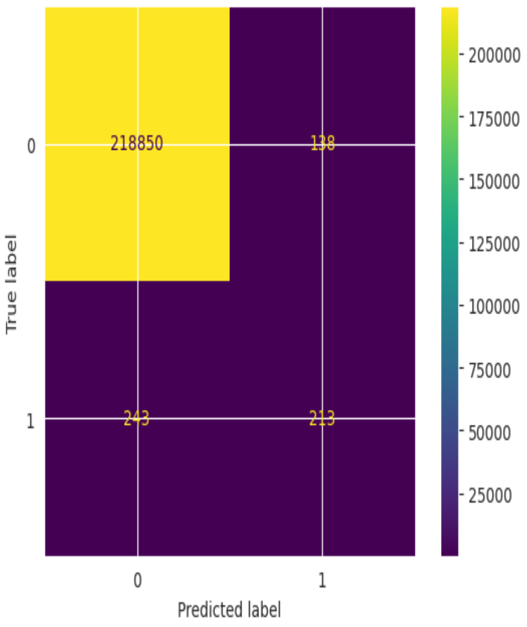
# DISCUSSION

In this project I have analyzed the dataset with the financial transactions and built a machine learning model to determine whether the transaction was fraudulent or not. Over the course of creating a predictive model I have conducted analysis of the dataset. The analysis included the data cleaning and conducting exploratory data analysis after which insights were presented in the report.

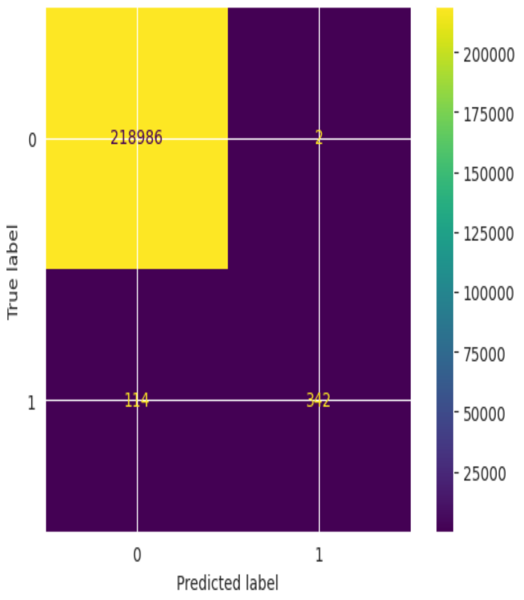
In predictive modeling I have used 3 machine learning algorithms which are Logistic Regression , Random Forest Classifier and Bagging Classifier. To evaluate the performance of each model , metrics were used and results for which are recorded in the result section. The models were given different datasets which included imbalanced , undersampled , oversampled and datasets which are treated with combinations of the undersampling and oversampling for training and testing purposes . In the imbalance dataset it was found that the recall was higher in all the 3 models as compared to the precision value , this was mainly due to low false negatives and very high false positives that can be identified from the confusion matrices .After conducting all experiments the techniques that are most suitable for an imbalanced dataset are oversampling and the combination of oversampling and undersampling because in the undersampling huge amounts of data is lost and even the model is trained on insufficient data that affects the performance of the model. The oversampling and the combination of oversampling and undersampling techniques have outstanding results as these can be seen in the results section . The reason for considering SMOTE is that , in case of oversampling, the records are just duplicated from the original records for the minority class but in SMOTE , synthetic records are generated based upon the minority class . All the 3 models were trained on the different datasets and Random Forest Classifier has performed well compared to the other 2 models in both oversampling and combination of oversampling and undersampling techniques.

# CONFUSION MATRICES FOR THE MODELS

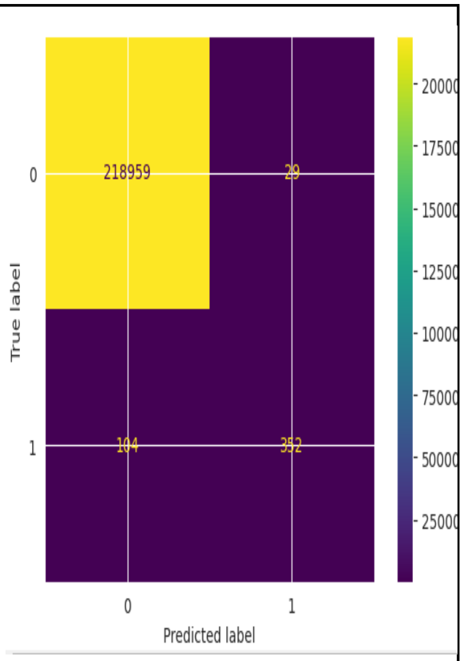
## Imbalanced Dataset



**Logistic Regression**

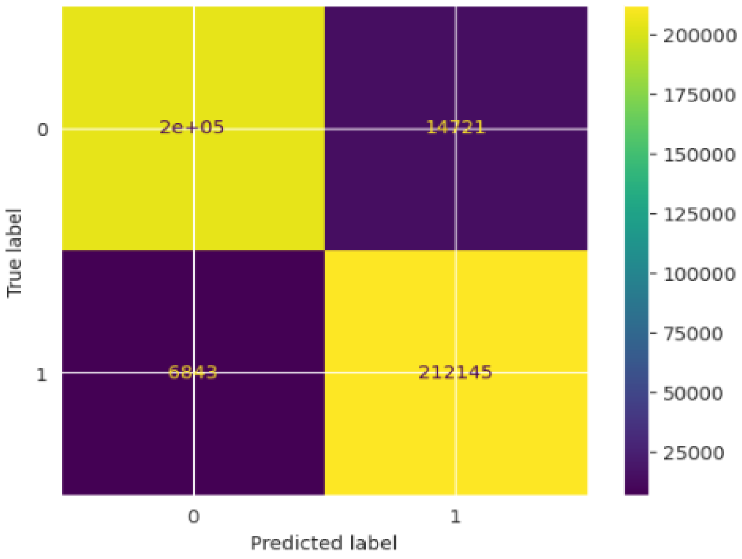


**Random Forest Classifier**

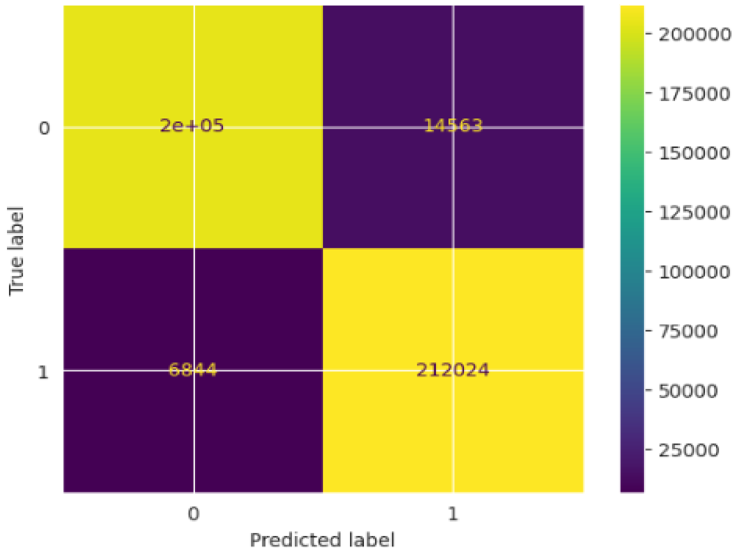


**Bagging Classifier**

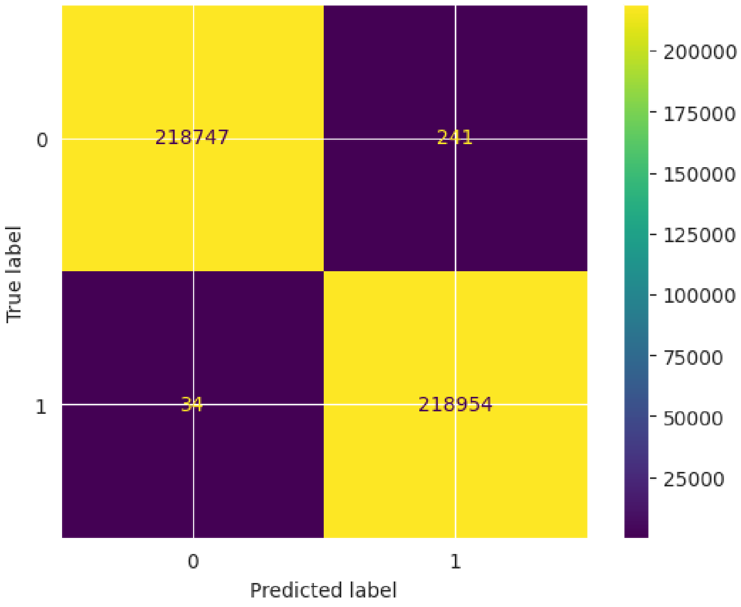
| SMOTE | Combination of Undersampling and Oversampling |
|---|---|



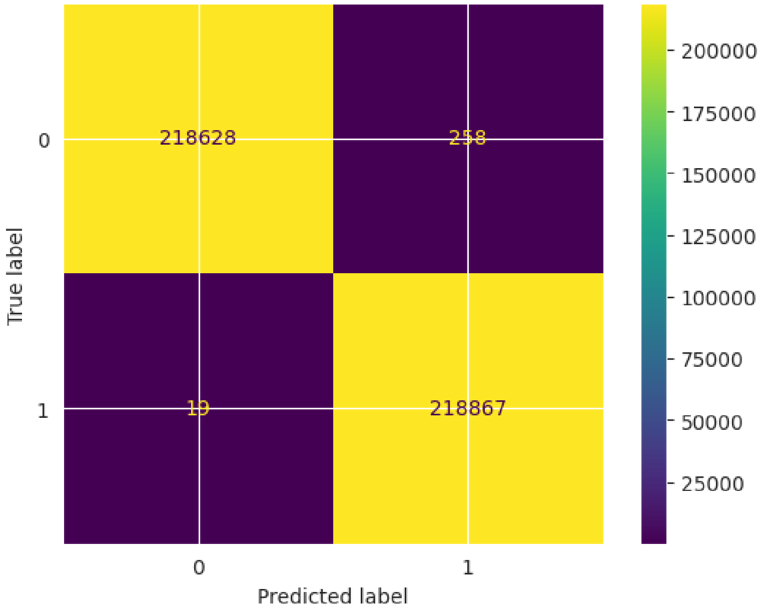
**Logistic Regression**



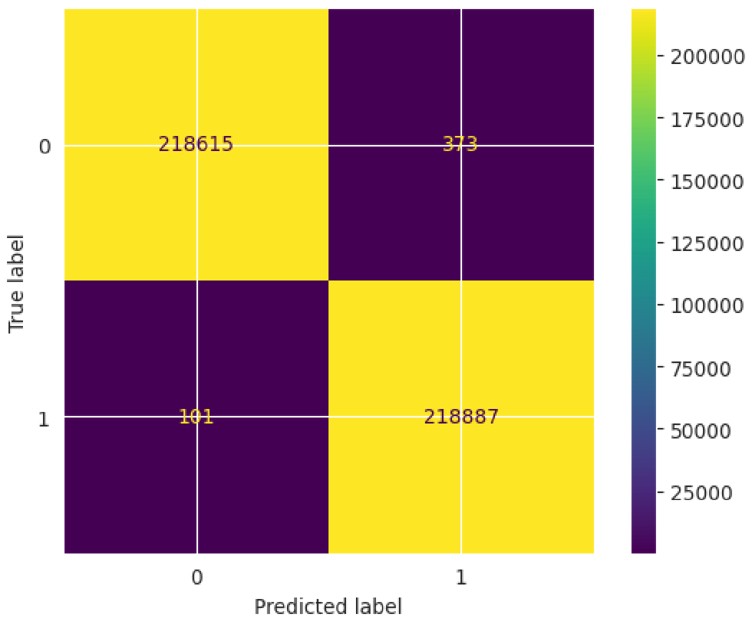
**Logistic Regression**



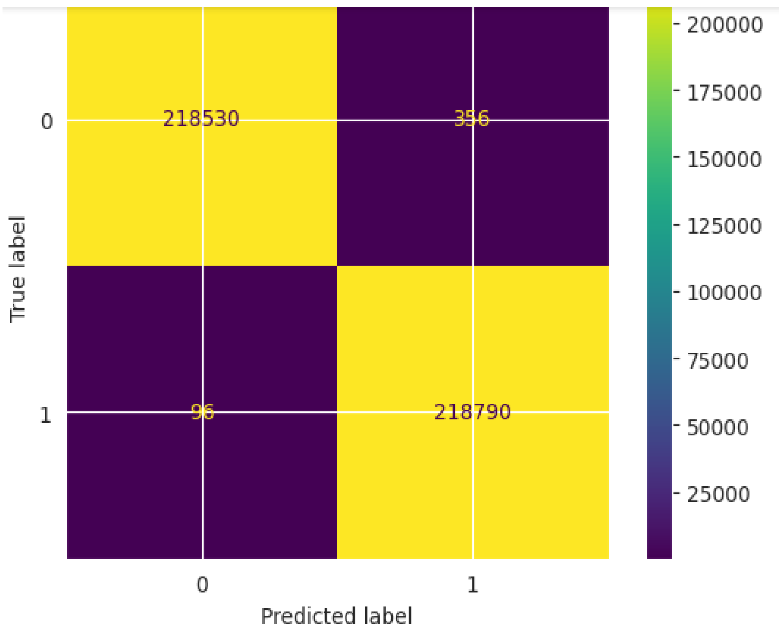
**Random Forest Classifier**



**Random Forest Classifier**



**Bagging Classifier**



**Bagging Classifier**

Along with the result section it can also be seen through the confusion matrices that the random forest classifier has low false negative , false positives and high true positives as compared to logistic regression and bagging classifier which is highly important from the security and business perspective. High false negative indicates that the model is unable to identify the fraudulent transactions thus not fulfilling the aim of the secure means . High false positive means that non-fraudulent transactions are misclassified as fraudulent transactions that can negatively hit the business. Thus a trade off between the false positives and false negatives is important . Since the motive of this project is to classify the fraudulent and non-fraudulent transactions therefore false negatives will be considered over the false positives and Random Forest Classifier outperforms other models on this parameter.

# REFERENCES

- Albashrawi, Mousa. (2016). Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015. Journal of Data Science. 14. 553-570. 10.6339/JDS.201607_14(3).0010.
- https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/
- https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/
- https://stripe.com/guides/primer-on-machine-learning-for-fraud-protection#:~:text=In%20machine%20learning%20parlance%2C%20a,example%2C%20blocking%20a%20legitimate%20customer.
- F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," in IEEE Access, vol. 10, pp. 39700-39715, 2022, doi: 10.1109/ACCESS.2022.3166891.
- Ali, A.; Abd Razak, S.; Othman, S.H.; Eisa, T.A.E.; Al-Dhaqm, A.; Nasser, M.; Elhassan, T.; Elshafie, H.; Saif, A. Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review. *Appl. Sci.* **2022**, *12*, 9637. https://doi.org/10.3390/app12199637