**PROJECT REPORT**

**ON**

**FINANCIAL FRAUD DETECTION**

Logo

Description automatically generated

Submitted by-

Team 18

Amulya Rao Ravindra -001586084

Durga Bhavani Bangari - 001027500

**Abstract**

This report consists of the detailed findings of having applied various Machine Learning Algorithms to build a model that predicts the fraudulent money transactions based on a month-long mobile money transaction in an African Country.

The dataset for building the model has been obtained from Kaggle. The dataset has been cleaned, after which the Exploratory Data Analysis has been carried out to get visual insight into the dataset and the different features in it. Various Machine Learning Algorithms have been implemented to understand the model performance. This entire model building process is carried out in Jupyter Notebook (Anaconda 3).

After carefully going through the dataset, it becomes evident that it is a classification type problem with target variable ‘isFraud’ taking values of either 0 or 1. The different Machine Learning Algorithms used for modelling are Logistic Regression, Decision Tree Classifier, Random Forest Classifier and K-Nearest Neighbors.

The model built using different algorithms is tested for its accuracy and is compared to obtain the best model that can correctly classify the fraudulent transactions.

**Table of Contents**

* Introduction
  + Background
  + Motivation
  + Goal
* Methodologies and Algorithms
  + About the dataset
  + Software and Libraries used
  + Data Cleaning
  + Feature Selection
  + Exploratory Data Analysis
  + Methods used
    - Logistic Regression
    - Decision Tree
    - Random Forest
    - K-Nearest Neighbors
  + Feature Engineering
* Description of Dataset
* Analysis
* Data Preparation
* Modelling
* Conclusion
* Reference

**Introduction**

**Background:**

Financial Fraud Detection aims to predict a fraudulent money transaction which is based on sample of real transactions extracted from one month of financial logs from a mobile money service implementation in an African country. This fraud detection system has **‘isFraud’** as the target variable which is detected based on a number of features present in the dataset like the type of transaction made, the amount being transferred, the balance before and after transaction of the person transferring the money and the person who receives the money.

**Motivation:**

The financial services industry and the likes of those that involve financial transactions are seeing a rise in the number of financial crimes which is now becoming a matter of concern for the regulators. In the year 2016 US alone recorded 15.4 million people who encountered frauds. About $6 billion was stolen from the banks due to fraudulent transactions. With the technology growing in leaps and bounds, it paves the way for newer and easier ways of financial services distribution. It also breeds the environment for fraudsters. There is a growing need to develop systems that can detect these fraudulent transactions and prevent them from happening. As a result, we as a team decided to work on the dataset with many variables and find correlations between user behavior and likelihood of fraudulent actions and develop a system to predict if a particular transaction is fraudulent or not.

**Goal:**

The goal of this project is to get an analysis of the fraud transaction and to classify/predict the fraud detection as accurately as possible. We plan to achieve this by building a machine learning model that learns from the various features present in the dataset and predicts the target variable.

**Methodology & Algorithm**

**About the Dataset:**

The dataset consists of one month of financial log of mobile money service deployed in an African country. The dataset has a size of about 493MB.It has 6362620 rows and 11 columns.

**Software and Libraries used**

The dataset is downloaded from Kaggle. The Jupyter notebook is used with several Libraries

They are as follow:

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Metrics
* Mean Squared Error
* Scikitlearn Tree
* DecisionTreeClassifier
* train\_test\_split
* accuracy\_score
* sklearn. ensemble
* RandomForestClassifier
* sklearn.preprocessing
* LabelEncoder
* OneHotEncoder
* sklearn. pipeline
* GridSearchCV,
* RandomizedSearchCV
* sklearn. utils

**Data Cleaning:**

Before model development the data needs to be cleaned, which is to check if there are any missing values in the dataset. If there are any, it can hinder with the model development. Therefore, the missing value are either removed or replaced with the mean of the values in that respective column. At first glance the dataset has no null values.

**Feature Selection**

The dataset has in total 11 columns including the target variable. However, there are some features in the dataset that do not significantly contribute towards the target variable. The feature variables are ‘isFlaggedFraud’. This feature is removed before modelling. The final model is built with 9 columns excluding the target variable.

**Exploratory Data Analysis:**

Exploratory Data Analysis is done to get visual insights into the dataset using the library matplotlib and seaborn. We have plotted correlation plots called heatmaps using seaborn library. Then we have plotted various bar graphs and line plots to understand the number of fraud and non-fraud transactions and to understand at what time of the day usually the fraud attacks happen.

**Models Used**

Once the dataset is cleaned it will be divided into train and test data sets to build a model by applying different Machine Learning algorithms.

The prediction model that is bult is of classification type. In other words, it is built to predict if a particular transaction is fraud or not.

The target variable ‘isFraud’ has two values:

* isFraud 0- not a fraud transaction
* isFraud 1-fraud transaction

Below are the algorithms used in this project:

* Logistic Regression
* Decision Tree Classifier
* Random Forest Classifier
* K-Nearest Neighbors Classifier
* **Logistic Regression Classification:**

Logistic regression is a classification algorithm, used when the value of the target variable is clearly defined and discrete in nature It is a type which is used to predict the target variable. In our dataset the target variable ‘isFraud’ is also discrete in nature giving us a value of either 0 or 1 depending upon the transaction.

* **Decision Tree Classification:**

Decision Tree algorithm can be used for classification and regression type of problems. It predicts the target variable by applying decision rules on the training dataset and learning from it. The main benefit of the decision tree classifier is its ability to using different feature subsets and decision rules at different stages of classification. The dataset has features like the type of transactions cash\_in, cash\_out, payment, debit that it uses in the process to classify the transaction as fraud or not.

* **Random Forest Classification:**

The random forest algorithm contains several decisions trees. This type of algorithm uses bagging and feature randomness when building single tree and tries to create an uncorrelated forest of trees whose prediction is accurate than any individual tree. It provides higher accuracy through cross validation. This classification will also handle missing values and preserve the accuracy of a huge proportion of data. In our model building process, we use Hyperparameter tunning to attain a good accuracy score with Random Forest algorithm which is 77%

* **K-Nearest Neighbors:**

The K-Nearest Neighbors algorithm works on the assumption that similar things exist in close proximity to each other. In our model we have used the number of neighbors to be 5 and obtain an accuracy of 88%.

**Dataset**

**Description of Dataset:**

The dataset consists of 6362620 rows and 11 columns. It consists of integer data only. The various features of the dataset are as follows:

|  |  |
| --- | --- |
| **Feature Names** | **Feature Description** |
| 1.step | A timestamp / date variable that has been made arbitrary for data privacy. |
| 2.type | It indicates the type of transaction. Type is our only categorical independent variable. Categories include cash\_in, cash\_out, debit, payment, transfer. |
| 3.amount | It is the size of transaction. |
| 4. nameOrig | Customer who made the transaction |
| 5. oldbalanceOrg | Initial balance before the transaction |
| 6. newbalanceOrig | New balance after the transaction |
| 7. nameDest | Customer who is the recipient of the transaction, C = Client, M = Merchant |
| 8. oldbalanceDest | Initial balance of recipient before the transaction. |
| 9. newbalanceDest | New balance of recipient after the transaction. |
| 10. isFraud | It is the target variable. It tells whether the transaction was made by a fraudulent agent. 1 indicates fraudulent transaction, 0 indicates non fraudulent transaction |
| 11. isFlaggedFraud | It tells whether the transaction was flagged as fraud by the "business model". |

**Data Source**

The Dataset has been taken from Kaggle:

Synthetic Financial Fraud Detection Dataset

<https://www.kaggle.com/ntnu-testimon/paysim1>

**Analysis**

**Data Exploration**

As a first step of analysis, we have loaded the dataset into a data frame and see the columns and the data it has.

Table

Description automatically generated

The insight into the different columns and its datatypes in the data frame can be obtained using the command **df.info ()**

Table

Description automatically generated

The dataset chosen has 6362620 rows and 11 columns. Before starting to work with the dataset it needs to be cleaned i.e., it needs to be checked for null values and to see if there are any discrepancies in the dataset. We Have performed initial data cleaning steps to check if any of the 11 features have null values.

**Data Cleaning**

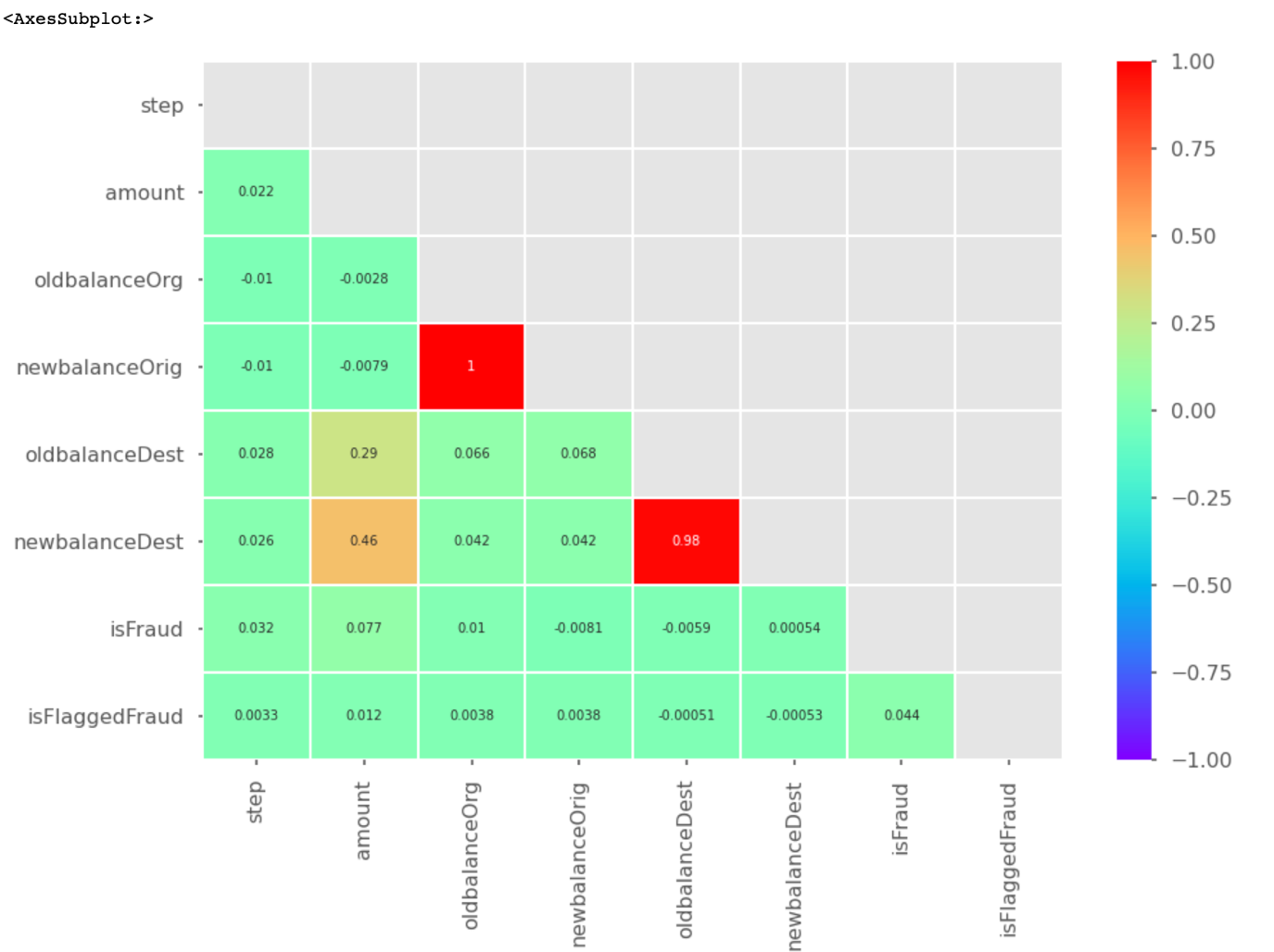
We have checked for null values in the columns using the command **df. isnull (). sum ()**

Table

Description automatically generated

From initial inspection we can see that none of the features have null values in them.

The correlation matrix is plotted to understand how the different features are related to each other.



From the above correlation matrix, we can see that the featured having a correlation of 1 or close to 1 are highly correlated and those having values close to 0 or 0 are not correlated.

For the five different types of transactions, we check if there are any missing or zero values in oldbalanceOrig and newbalanceOrig, when there is an amount >0 being transferred.

**PAYMENT** type of transaction

Table

Description automatically generated

**CASH\_IN** type of transaction



There are no zero entries in oldbalanceOrig and newbalanceOrig

**CASH\_OUT** type of transaction

Table

Description automatically generated

**TRANSFER** type of transaction

Table

Description automatically generated

**DEBIT** type of transaction

Table

Description automatically generated

From the above findings we see that except for CASH\_IN type of transactions all other types of transactions have zero values in oldbalanceOrig and newblanceOrig. Now this could be because of wrongly recorded values or missing values. To fix this issue we check if the difference of the newbalancedest and the amount of transaction equals to the oldbalanceDest. If the money adds up correctly then it means that there is a missing value in oldbalanceOrig.

Graphical user interface, text, application, email

Description automatically generated

We replace oldbalanceOrig with the amount of transaction if the money adds up.

For the rows where the money doesn’t add up, we drop them if it’s a non-fraud transaction.

Graphical user interface, text, application

Description automatically generated

Checking the correlation after dropping the rows

Chart

Description automatically generated with low confidence

Table

Description automatically generated

**Exploratory Data Analysis**

We start by visualizing the number of fraud and non-fraud transactions in the dataset

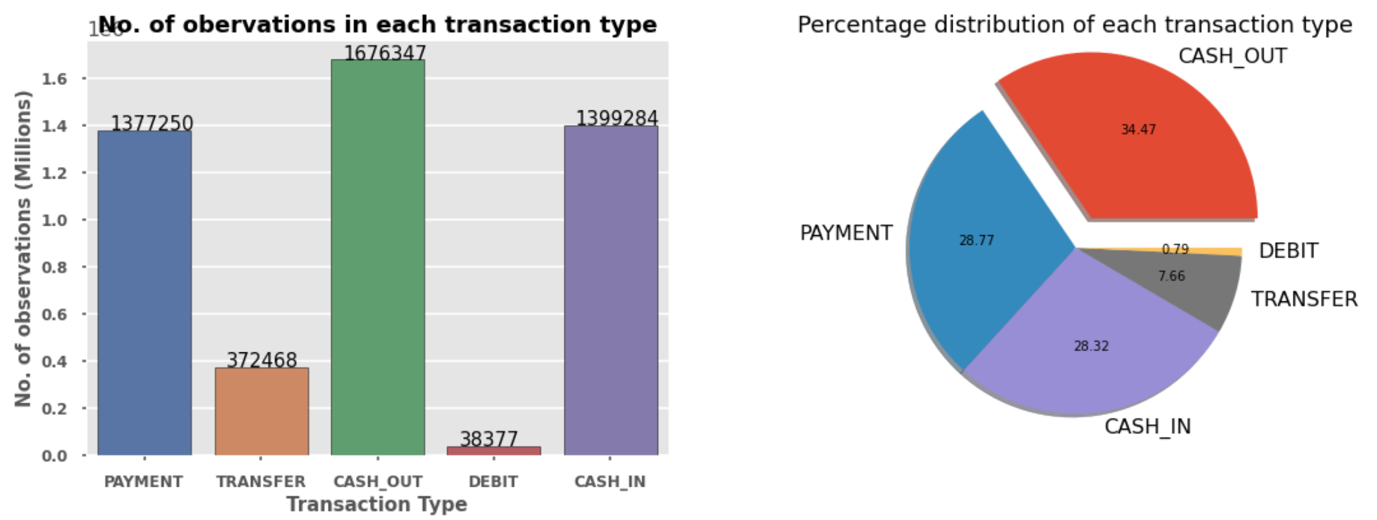
Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

From the graphs we see that the dataset is skewed or imbalanced. That is to say that the number of non-fraud transactions recorded in the dataset are very high in number in comparison to fraud transactions.



The above graphs depict the total number of transactions in each type.

Chart, line chart

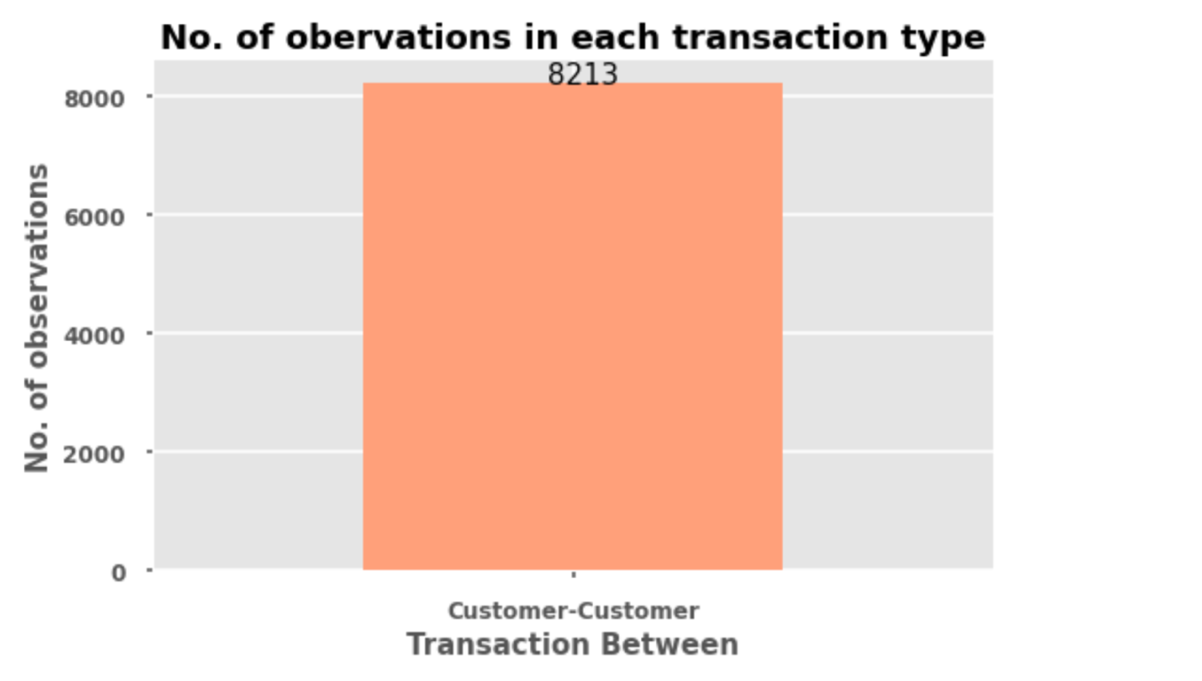
Description automatically generated

It shows the peak of Transfer and Cash\_out type of transactions during the day. The other types of transactions are stable.

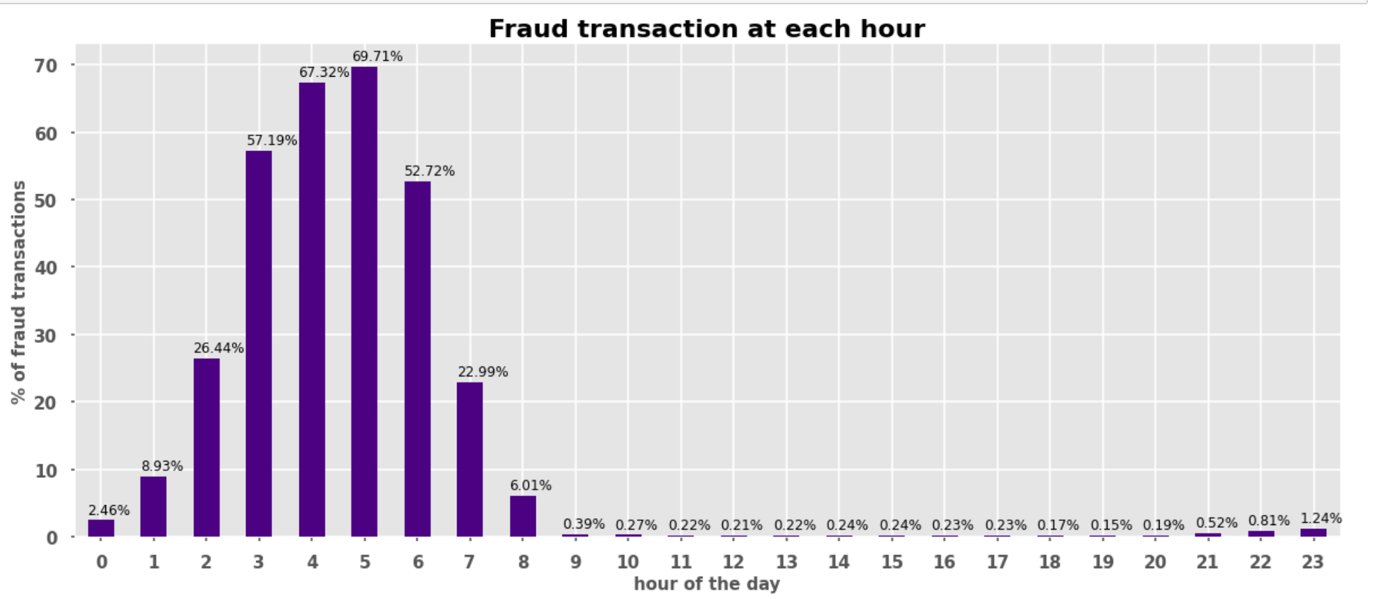
Chart, bar chart

Description automatically generated

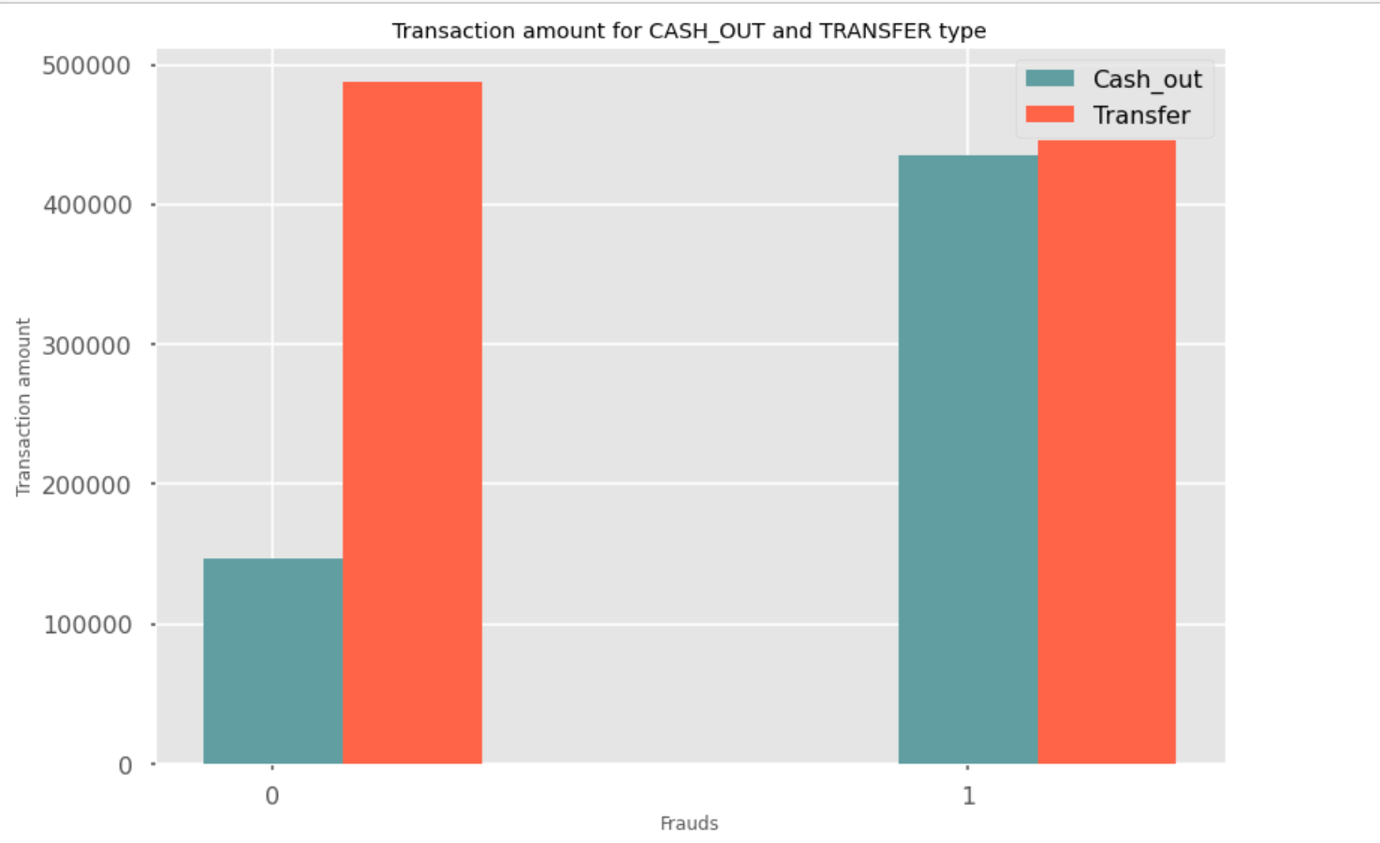
From the graph above we see that the fraud transactions are only of the transaction type CASH\_OUT and TRANSFER.



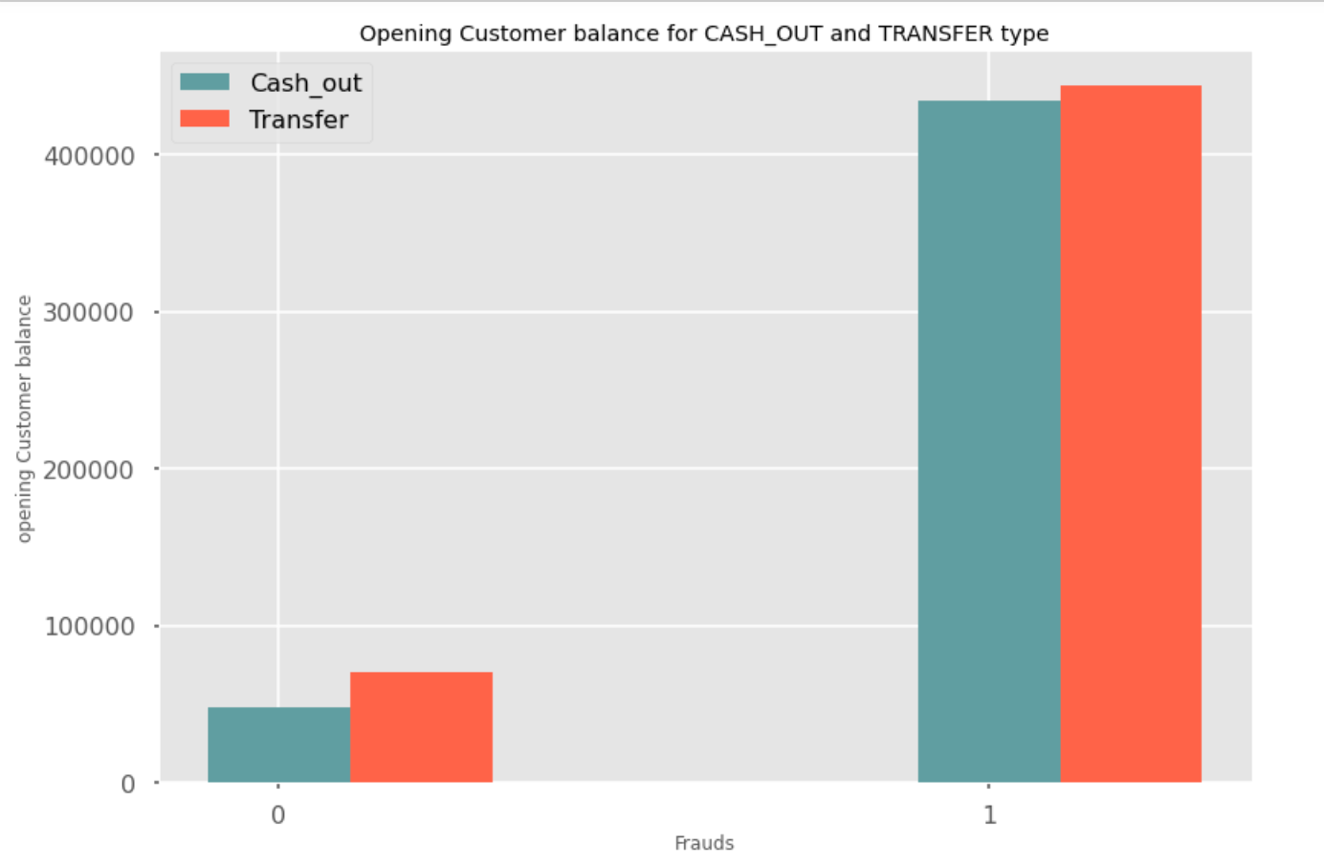
All the fraudulent transactions are between customer to customer only



This graph gives us an insight into what hour of the day do frauds mostly occur. It usually occurs in the early hours of the day.



We see that for both Cash\_out and Transfer type of transactions the amount being taken out is very high for frauds.



The fraudsters tend to attack accounts with very high customer balance.

**Data Preparation**

The features that don’t add value to the target variable are dropped. From EDA we have seen that transaction types of Cash\_in, Payment, Debit do not cause frauds. So, the rows containing these attributes are dropped. Similarly, the features hour and transaction between also aren’t of significance in determining the target.

Graphical user interface, text

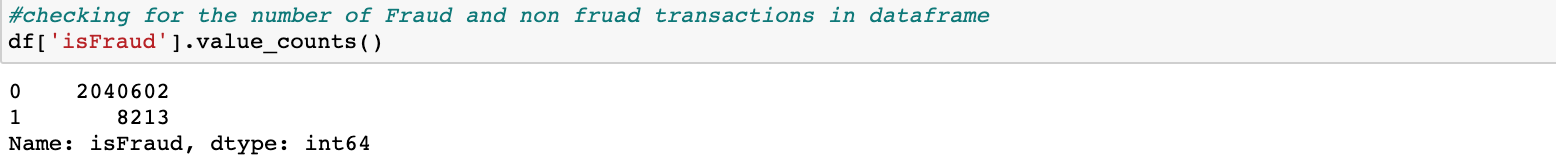
Description automatically generated

Checking the correlation after dropping the columns

Table

Description automatically generated

Checking for number of fraud and non-fraud transactions after dropping the columns



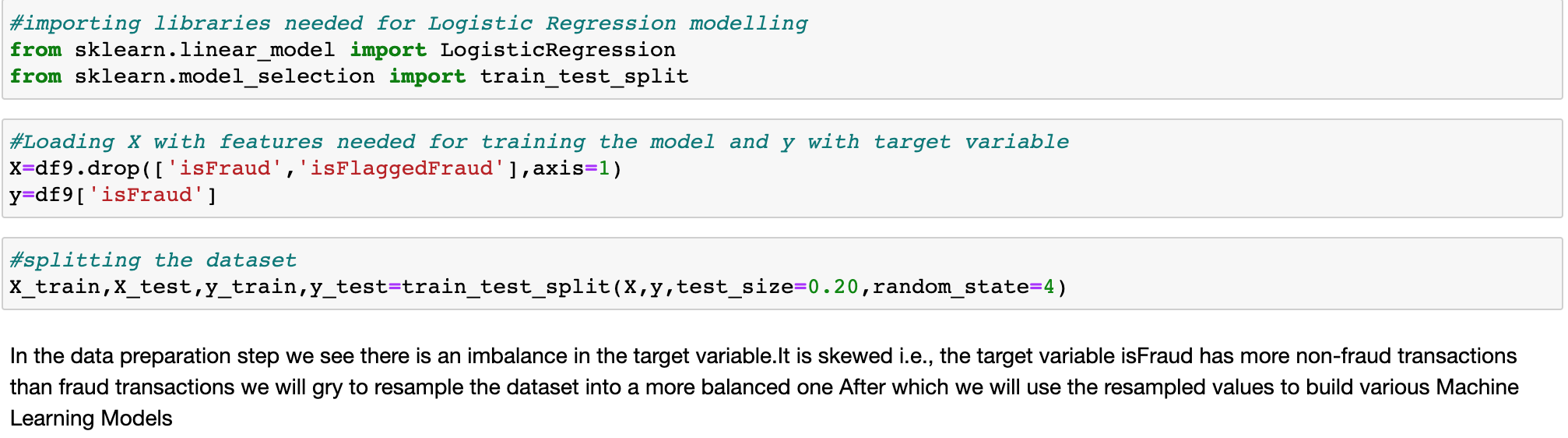
Encoding the categorical columns type, nameOrig, nameDest

Table

Description automatically generated

**Modelling**

This step involves splitting the dataset into train and test sets in the ratio 80:20 respectively.

****

Since the dataset is skewed, we resample the target variable ‘isFraud’ so that both fraud and non-fraud transactions are of equal instances.

**Graphical user interface, text, application, email

Description automatically generated**

**Text

Description automatically generated with low confidence**

**Logistic Regression Classifier:**

**Graphical user interface, text, application, email

Description automatically generated**

**Decision Tree Classifier:**

**Graphical user interface, text, application, email

Description automatically generated**

Plotting the tree

Diagram

Description automatically generated

**Random Forest Classifier:**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**K-Nearest Neighbors Classifier:**

**Graphical user interface, text, application, email

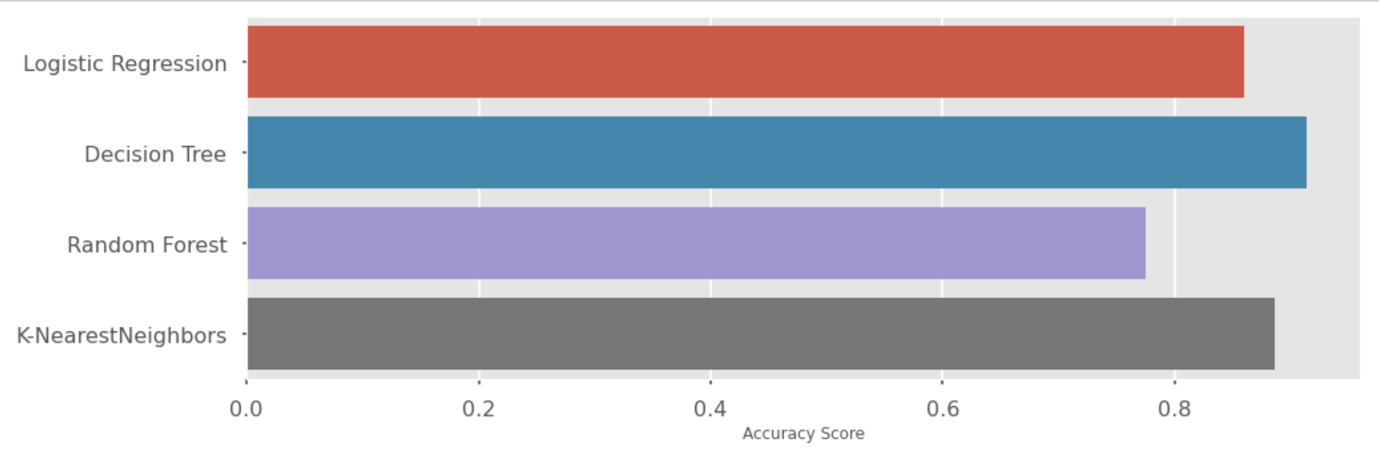
Description automatically generated**

**Representing all the accuracies in a data frame:**

**Graphical user interface, text, application

Description automatically generated**

Plotting all the accuracies in a bar graph



**Conclusion**

* From the above models used we see that the accuracy score of Decision Tree Classifier is best with the accuracy of 91% in predicting the target variable ‘isFraud’.
* From the analysis carried out we see that fraudulent transactions occur only due to Cash\_out and Transfer type of transactions.
* Fraudulent transactions occur between Customer-to-Customer transactions.

**References**

* Fraud Detection: How Machine Learning Systems Help Reveal Scams in Fintech, Healthcare, and eCommerce[: https://www.altexsoft.com/whitepapers/fraud-detection-how-machine-learning-systems-help-reveal-scams-in-fintech-healthcare-and-ecommerce/](NULL)
* [https://matplotlib.org/stable/api/\_as\_gen/matplotlib.pyplot.legend.html](•%09https:/matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.legend.html)
* [https://heartbeat.fritz.ai/resampling-to-properly-handle-imbalanced-datasets-in-machine-learning-64d82c16ceaa](•%09https:/heartbeat.fritz.ai/resampling-to-properly-handle-imbalanced-datasets-in-machine-learning-64d82c16ceaa)
* [https://towardsdatascience.com/optimizing-hyperparameters-in-random-forest-classification-ec7741f9d3f6](•%09https:/towardsdatascience.com/optimizing-hyperparameters-in-random-forest-classification-ec7741f9d3f6)
* [https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28](•%09https:/towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28)
* [https://medium.com/@ODSC/transforming-skewed-data-for-machine-learning-90e6cc364b0](•%09https:/medium.com/@ODSC/transforming-skewed-data-for-machine-learning-90e6cc364b0)
* [https://www.kaggle.com/ntnu-testimon/paysim1](•%09https:/www.kaggle.com/ntnu-testimon/paysim1)