Team Runtime Terror
AI Syberthon

Data Analysis Report

Introduction:
The sole purpose of this analysis is to determine the stability and accuracy of the current existing system in the banks. We ran various data visualisation techniques and ML-Models to provide the most optimum solution.

Data Section:

1. Dataset: We were provided with a dataset PS_20174392719_1491204439457_log”

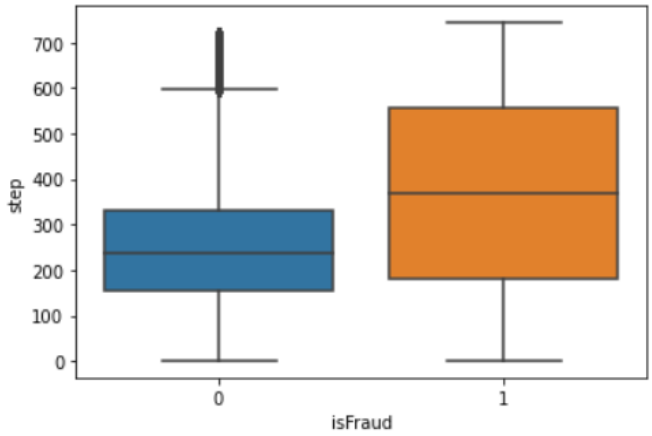
2. Testing and Training Sets: We are using the Scikit-Learn library for splitting data into training and testing sets. The proportion we’re setting is 4:6 (test_size = 0.4).

3. Data Cleaning: One of the most important tasks one has to do in data science analysis report is cleaning the raw data. The techniques we used are:
    a. Checking for NULL Values
    b. Checking for incorrect input in all integer columns
    c. Dropped columns which were not required
    d. Used Warning library

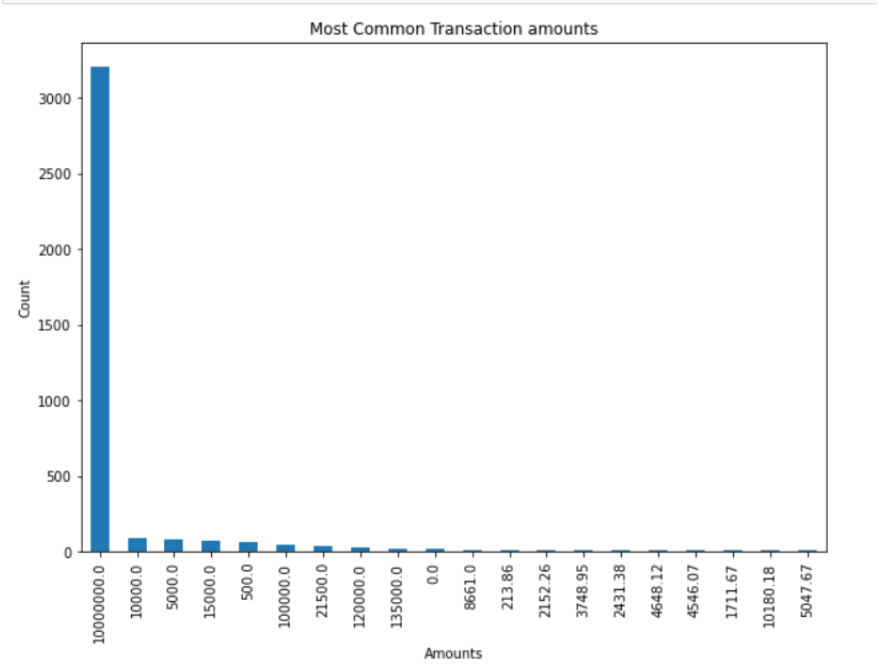
4. Transaction found to be Fraudulent: Frauds are identified in rows “ TRANSFER “ and “CASH_OUT”.

5. isFlaggedFraud: It is only activated when a transaction with a huge transfer amount. (For example, more than 200k)

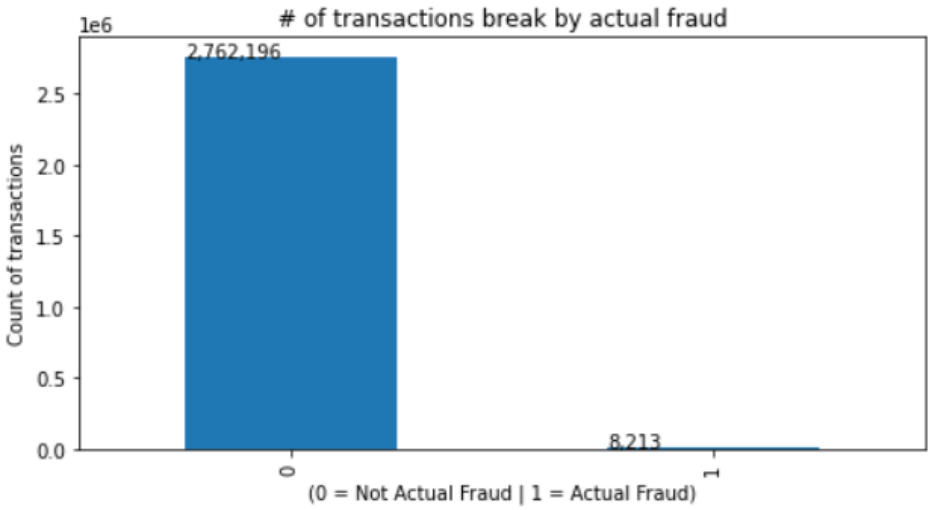
6. Data Visualization:



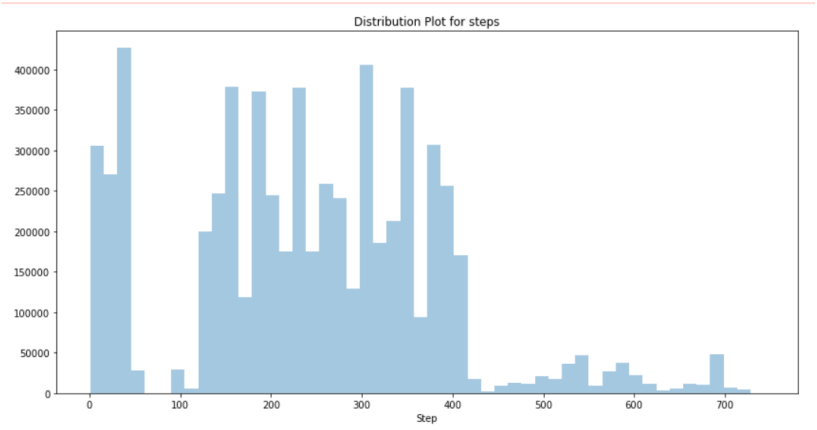
A graphical representation of isFraud vs step.



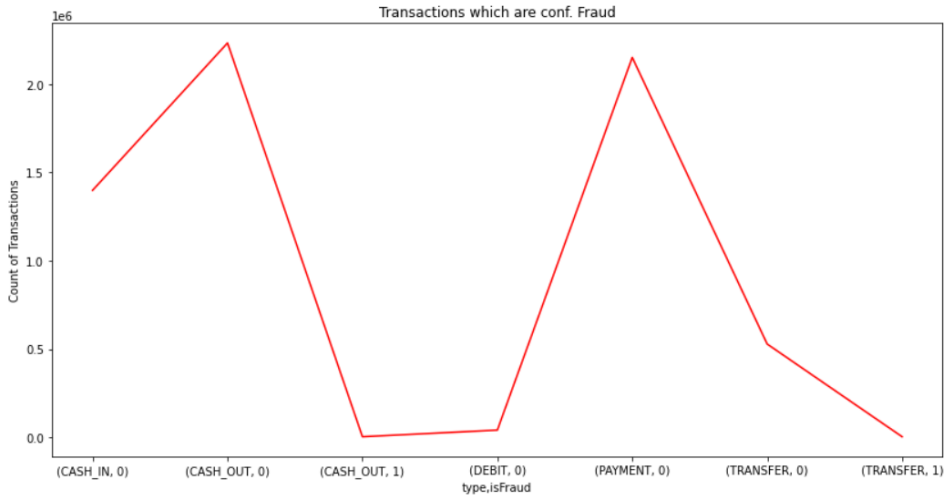
A graphical representation of most common transaction amounts



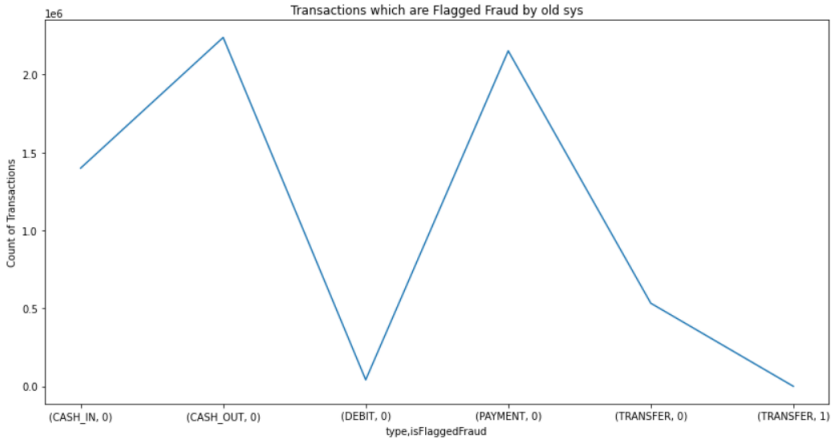
A graphical representation of transactions break by actual fraud



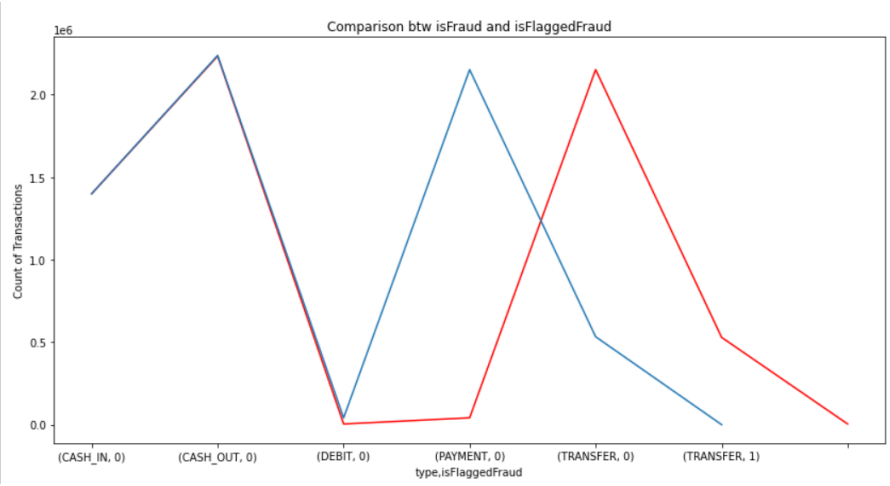
A distribution plot for steps



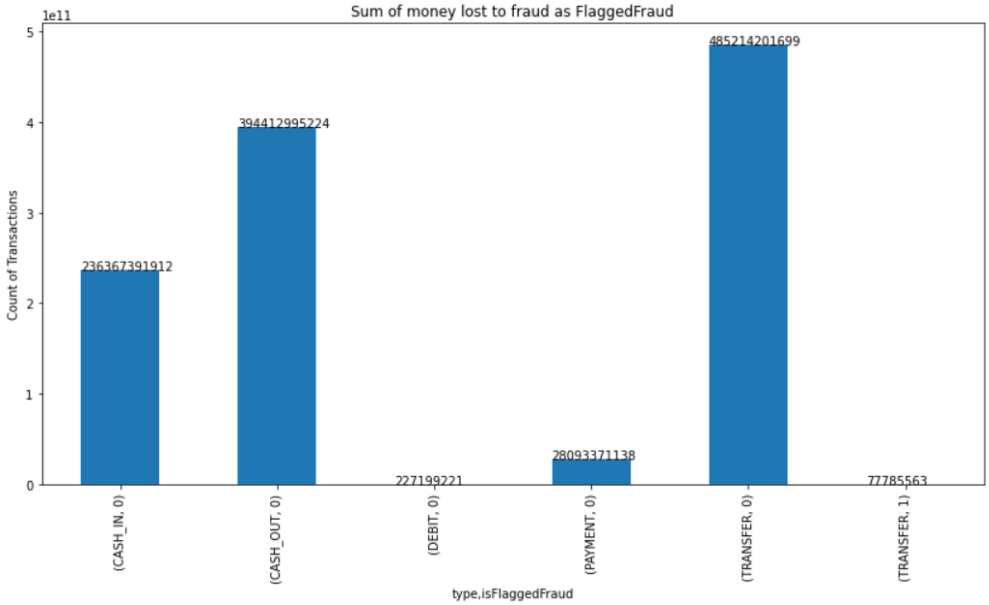
A graphical representation of transactions that are confirmed frauds



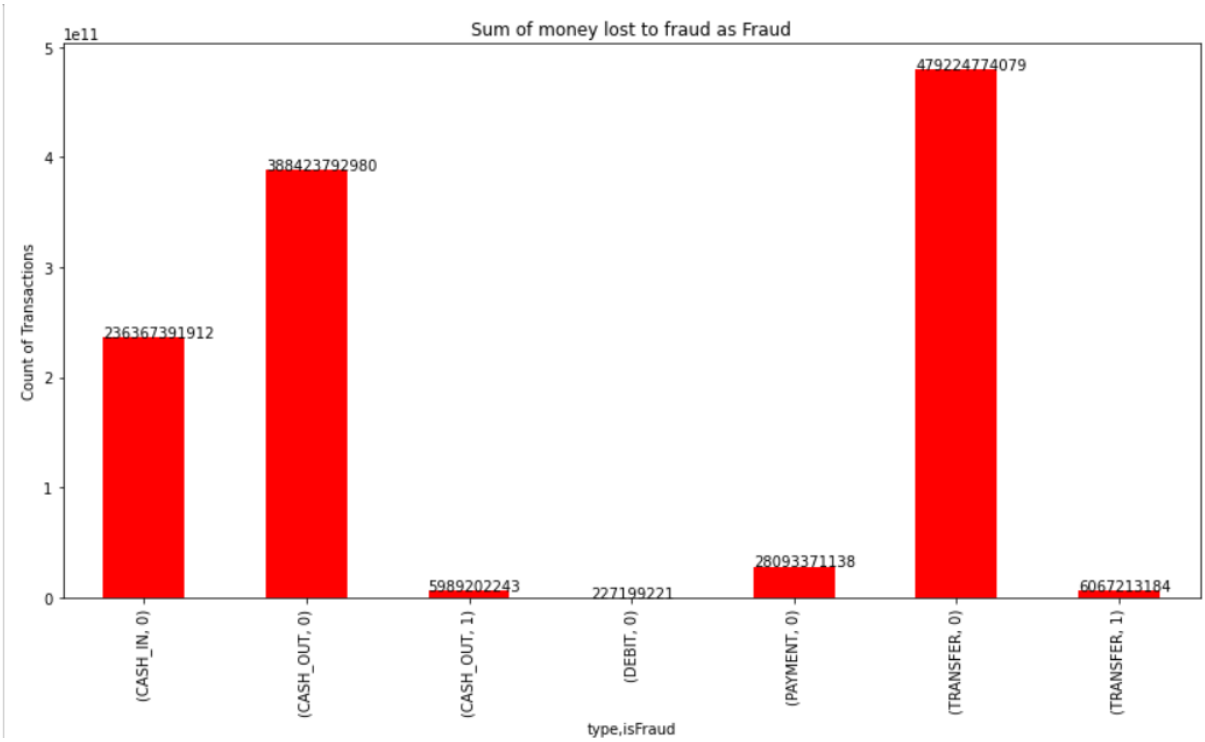
A graphical representation of transactions which are flagged fraud by the old system.



Comparison between isFraud and isFlaggedFraud



A graphical representation of sum of money lost to fraud as FlaggedFraud



A graphical representation of the sum of money lost to fraud as Fraud.

7. Fraud detection using ML Models:
   We looked upon various ML Models, like Logistic Regression, Decision Tree classifier, Random Forest Classifier. Each of them had a very comfortable accuracy score.
   We are cleaning, undersampling the unbalanced area and then feeding it into our models for better outcome.

In [49]:
```
model = LogisticRegression()
model.fit(x_train, y_train)
predictions= model.predict(x_test)
print(accuracy_score(y_test, predictions))
print(confusion_matrix(y_test, predictions))
print(classification_report(y_test, predictions))
```

```
0.8944805194805194
[[1909  158]
 [ 232 1397]]
              precision    recall  f1-score   support

           0       0.89      0.92      0.91      2067
           1       0.90      0.86      0.88      1629

    accuracy                           0.89      3696
   macro avg       0.90      0.89      0.89      3696
weighted avg       0.89      0.89      0.89      3696
```

In [50]:
```
y_pred = model.predict(x_test)
print(classification_report(y_test, y_pred))
print('Accuracy', accuracy_score(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.92      0.91      2067
           1       0.90      0.86      0.88      1629

    accuracy                           0.89      3696
   macro avg       0.90      0.89      0.89      3696
weighted avg       0.89      0.89      0.89      3696

Accuracy 0.8944805194805194
```

In [51]:
```
y_pred = model.predict(X)
print(classification_report(y, y_pred))
print('Accuracy:', accuracy_score(y, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.93      0.96   6354407
           1       0.02      0.88      0.03      8213

    accuracy                           0.93   6362620
   macro avg       0.51      0.90      0.50   6362620
weighted avg       1.00      0.93      0.96   6362620

Accuracy: 0.9272379931537637
```

In [53]:
```
cart=DecisionTreeClassifier()
cart.fit(x_train, y_train)
predictions=cart.predict(x_test)
print(accuracy_score(y_test, predictions))
print(confusion_matrix(y_test, predictions))
print(classification_report(y_test, predictions))
```

```
0.9878246753246753
[[2043   24]
 [  21 1608]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2067
           1       0.99      0.99      0.99      1629

    accuracy                           0.99      3696
   macro avg       0.99      0.99      0.99      3696
weighted avg       0.99      0.99      0.99      3696
```

In [55]:
```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=15)
if True:
    probabilities = clf.fit(x_train, y_train.values.ravel()).predict(x_test)
```

In [57]:
```
from sklearn.metrics import average_precision_score
if True:
    print(average_precision_score(y_test,probabilities))
```

```
0.9734942823498323
```

We preferred the Decision Tree Classifier as it had the best accuracy of 0.987824.
The most important features of a ML Model are:

1. Gathering Data
2. Cleaning Data
3. Defining the Model
4. Training, Testing and Predictions

Conclusion:
Existing System is not capable of detection of all the fraud transactions. Machine learning can be extensively used for the detection of fraud transactions.Predictive models produce good precision scores and are capable of detection of fraud transactions.