22-07-2023

FRAUD DETECTION

THE GOAL IS TO DEVELOP A MACHINE LEARNING MODEL THAT CAN EFFECTIVELY DISTINGUISH BETWEEN LEGITIMATE AND FRAUDULENT TRANSACTIONS.

Hrishikesh Tonge
230340325015
DBDA Mumbai

TABLE OF CONTENTS

0 1

DATA UNDERSTANDING

We first understand the given dataset through various parameters and then proceed for handling the data, outliers, null values, scaling and so on.

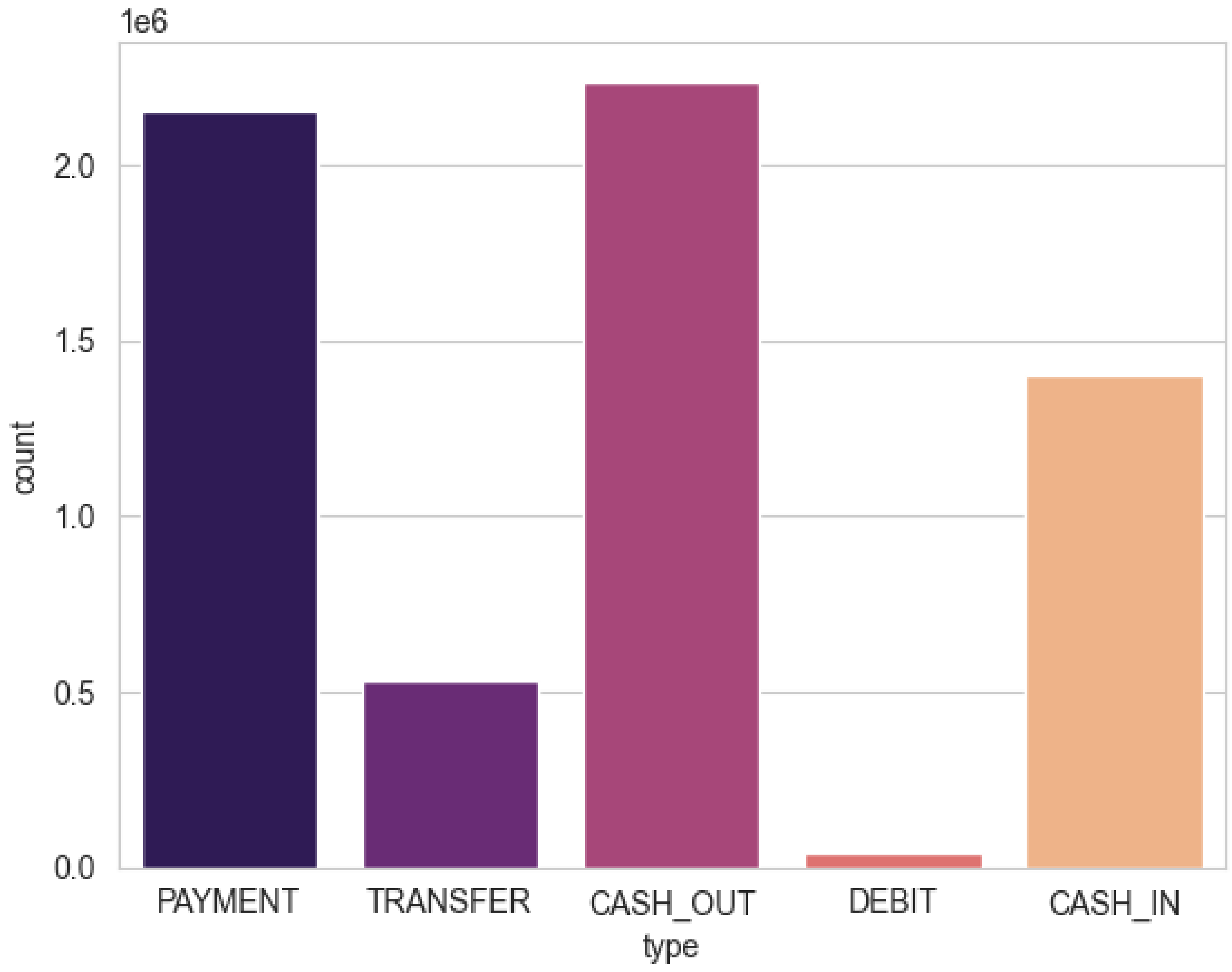
Size of data	**(6362620, 11)**
Null Values	**0**
Data Type	**Numerical + Categorical**
Target Column	**isFraud**

- As we dont have any null values we can proceed with the further analysis.
- We have recognised our target column as well, we will recheck this in the further steps.
- We have Categorical Data as well. Hence, at the later stage if the column is significant we will encode the categorical columns.
- The size of data is around ~ 480 mb with 6M rows and 11 columns.

0 2

UNIVARIATE ANALYSIS

1. OBSERVING **'TYPE'** COLUMN

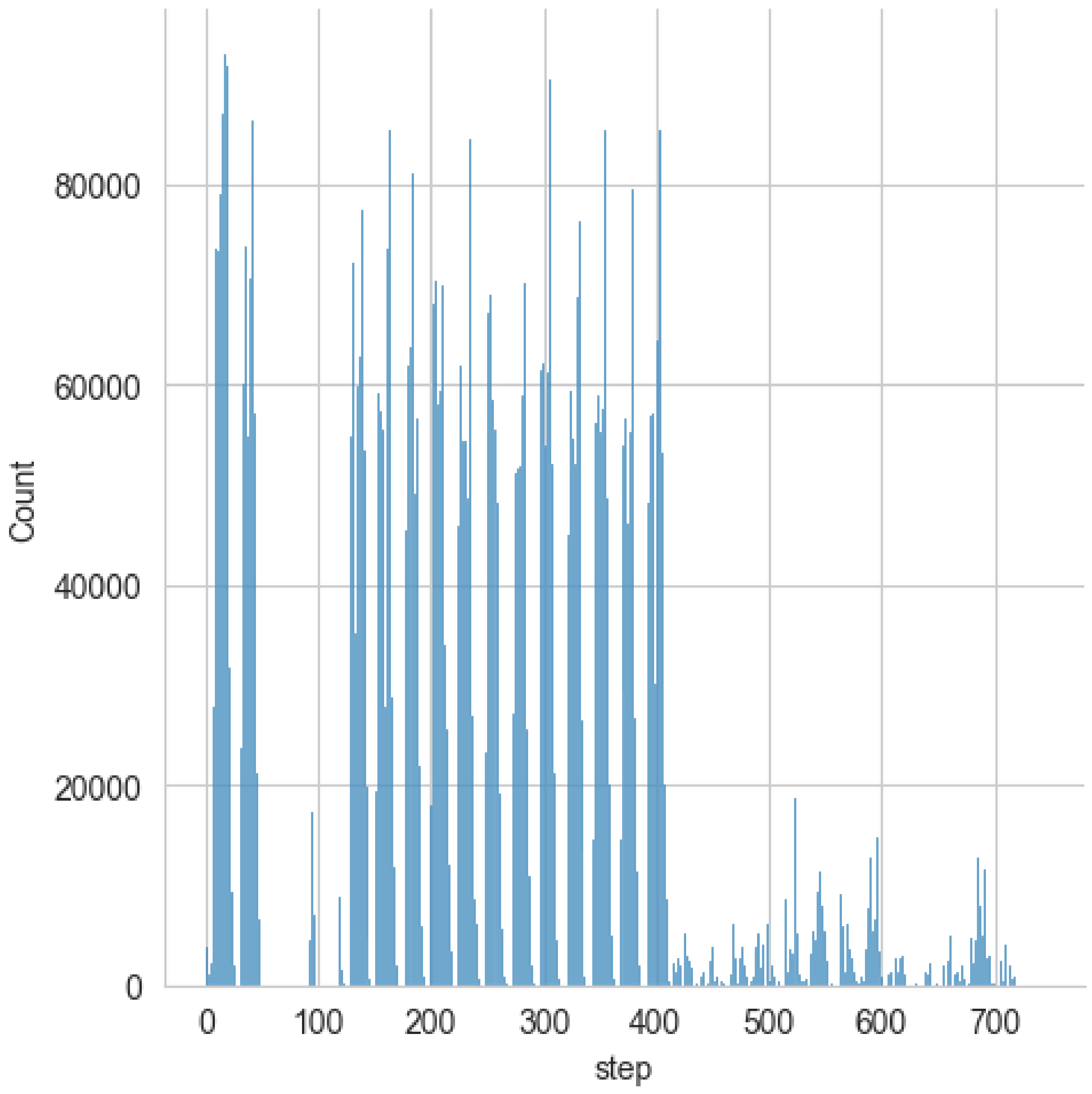


- Distribution of Payment and Cash out is maximum.
- People are preferring Payment type and Cashout type on larger basis as their Transaction medium.

0 2

UNIVARIATE ANALYSIS

2. OBSERVING DISTRIBUTION IN 'STEP' COLUMN



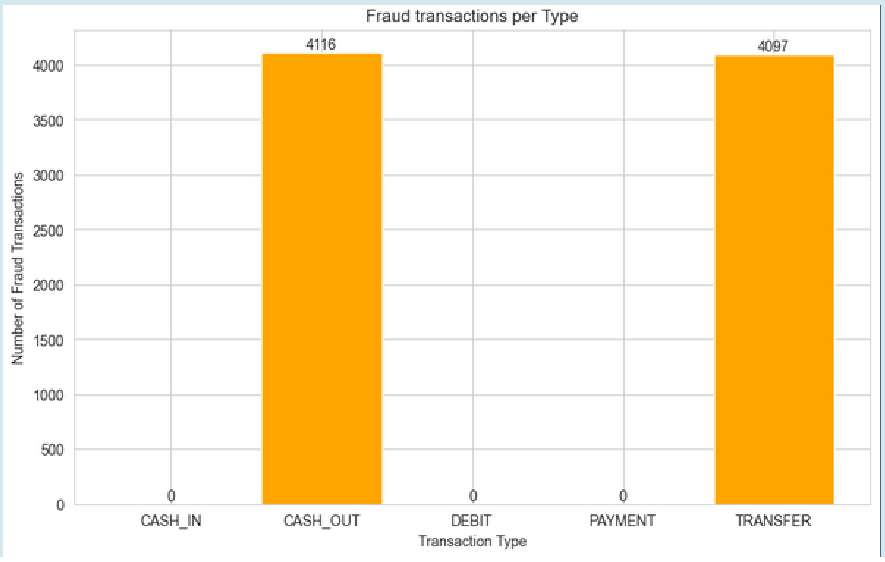
- STEP maps a unit of time in the real world. In this case 1 step is 1 hour of time.
- Thus we can observe maximum values lie between approx 0-30. And futher more between 150 to 400.

UNIVARIATE ANALYSIS

3. OBSERVING TARGET COLUMN

Number of Legit transactions	**6354407**	**99.8709 %**
Number of Fraudulent transactions	**8213**	**0.1291 %**

- As we can see the difference between the Fraud values varies too much. Thus data is imbalanced.
- These results prove that this is a highly unbalanced data as Percentage of Legit transactions= 99.87 % and Percentage of Fraud transactions= 0.13 %. SO **DECISION TREES AND RANDOM FORESTS ARE GOOD METHODS FOR IMBALANCED DATA.**

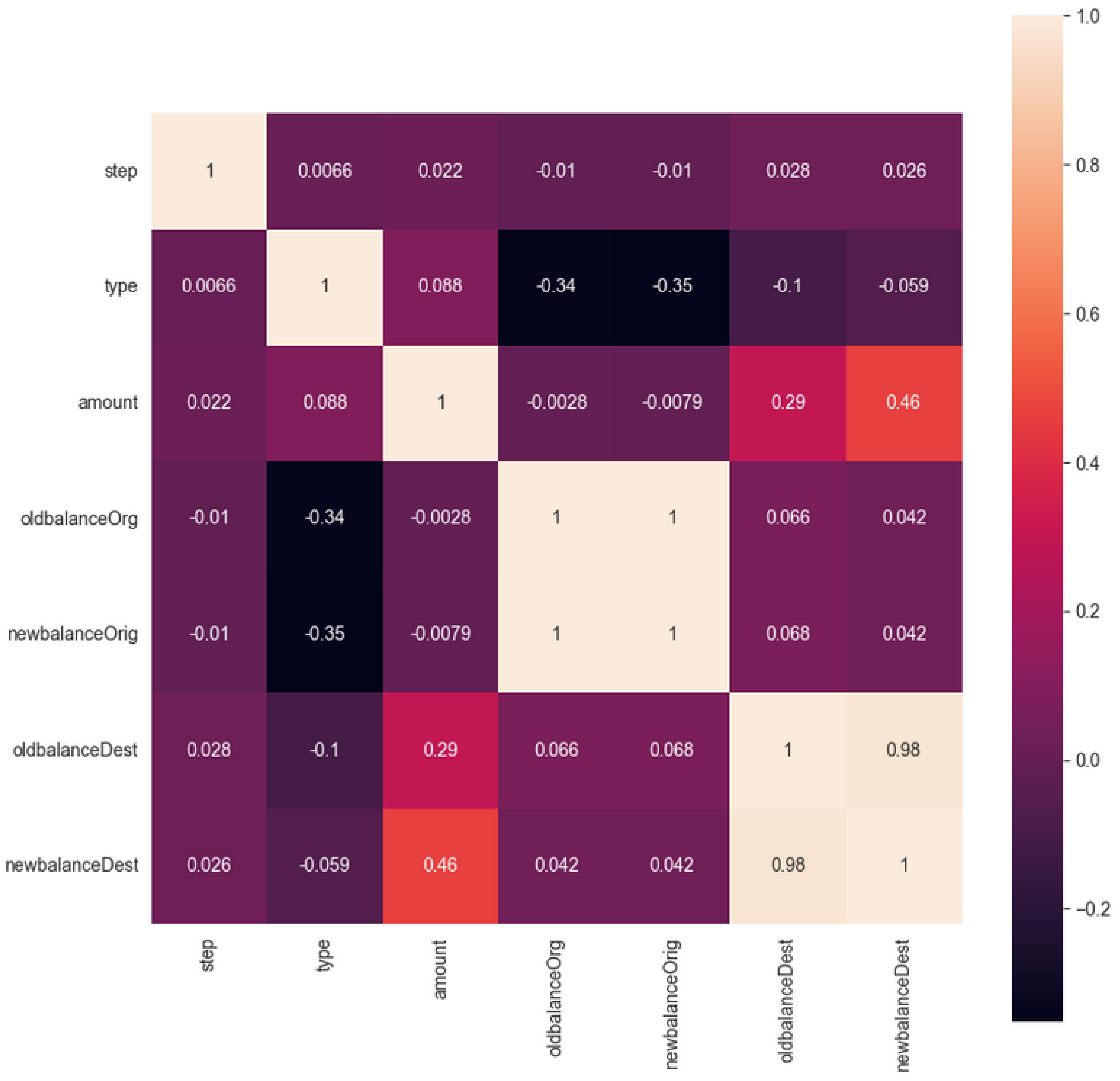


0 3

SCALING DATA

WE ARE USING STANDARD SCALAR TO SCALE THE DATA.

HEATMAP AFTER SCALING DATA

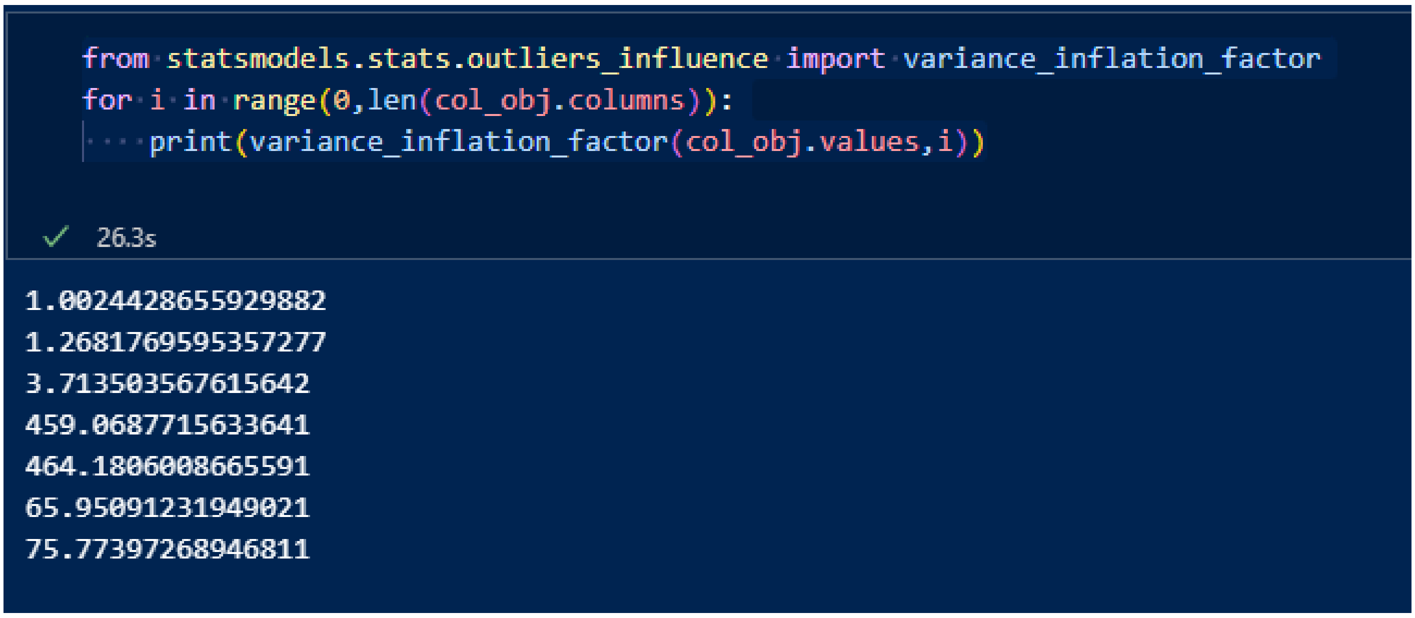


04

VIF

Multicollinearity :
Multicollinearity is a statistical phenomenon that occurs when two or more independent variables in a regression model are highly correlated with each other. In other words, multicollinearity indicates a strong linear relationship among the predictor variables. This can create challenges in the regression analysis because it becomes difficult to determine the individual effects of each independent variable on the dependent variable accurately.

Detecting Multicollinearity Using a Variance Inflation Factor (VIF)



```
from statsmodels.stats.outliers_influence import variance_inflation_factor
for i in range(0,len(col_obj.columns)):
    print(variance_inflation_factor(col_obj.values,i))
```

✓ 26.3s

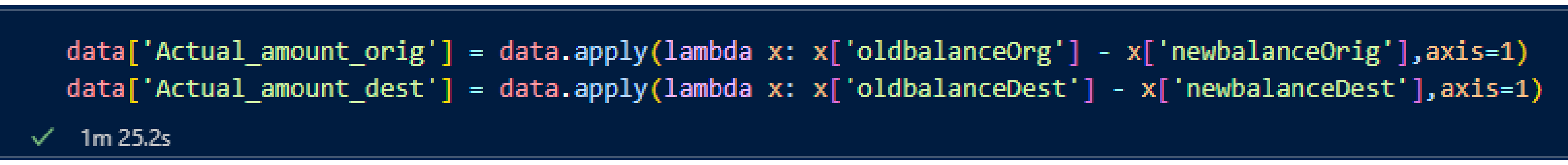
```
1.0024428655929882
1.2681769595357277
3.713503567615642
459.0687715633641
464.1806008665591
65.95091231949021
75.77397268946811
```

- Here we have calculated the Variance Inflation Factor of the independent variables
- Before doing this, we have converted the datatype of ‘type’ column from object to numerical using the label encoder.
- We can see that oldbalanceOrg and newbalanceOrig have too high VIF thus they are highly correlated. Similarly oldbalanceDest and newbalanceDest. Also nameDest is connected to nameOrig.
- Thus combine these pairs of collinear attributes and drop the individual ones.
- But these columns are significant for model building and analysis. Thus we will adjust the columns in the further steps.

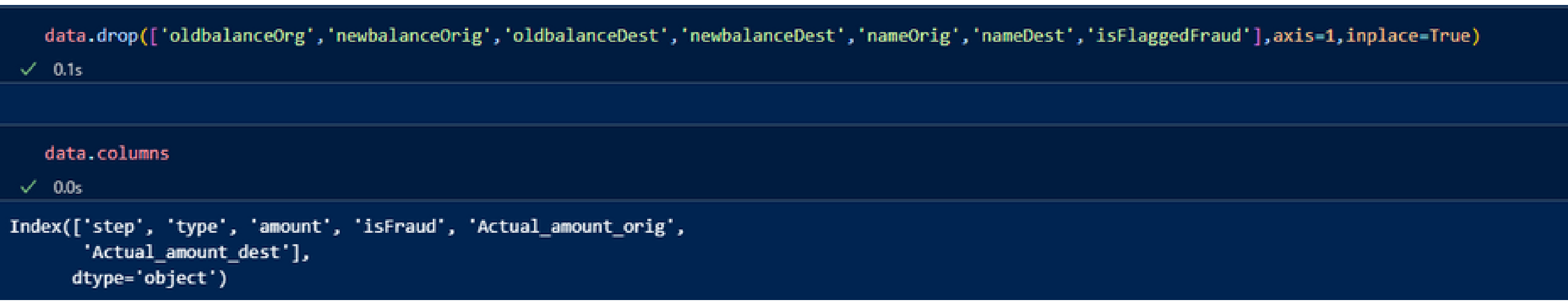
VIF

Handling highly correlated parameters

We can see that oldbalanceOrg and newbalanceOrig have too high VIF thus they are highly correlated. Similarly oldbalanceDest and newbalanceDest. Also nameDest is connected to nameOrig.
Thus combine these pairs of collinear attributes and drop the individual ones.



```
data['Actual_amount_orig'] = data.apply(lambda x: x['oldbalanceOrg'] - x['newbalanceOrig'],axis=1)
data['Actual_amount_dest'] = data.apply(lambda x: x['oldbalanceDest'] - x['newbalanceDest'],axis=1)
```
✓ 1m 25.2s

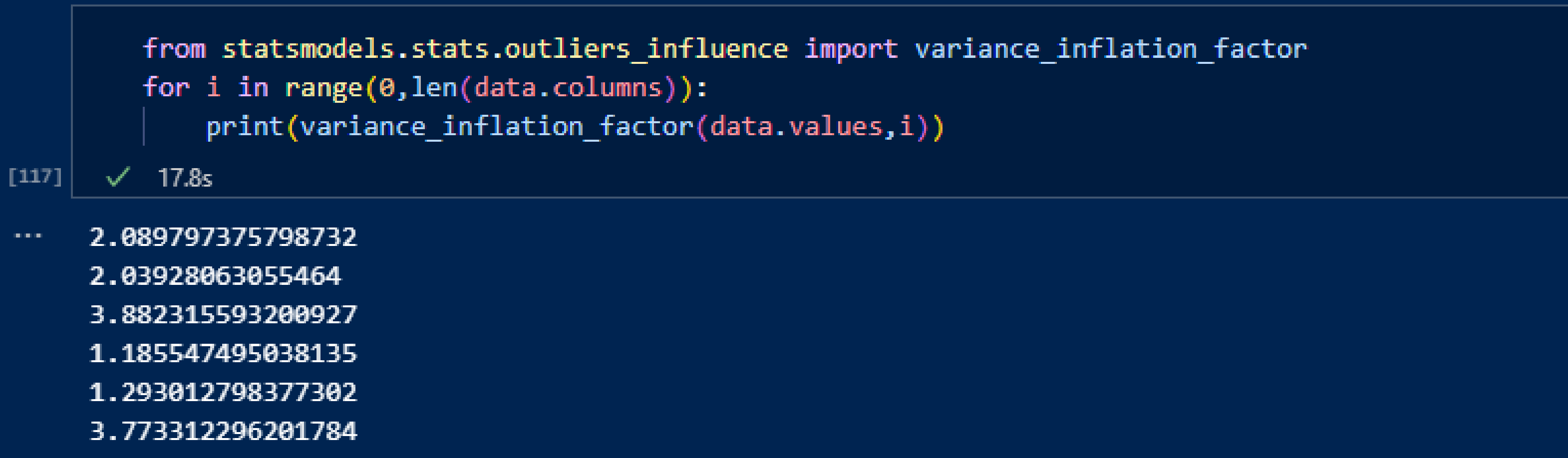


```
data.drop(['oldbalanceOrg','newbalanceOrig','oldbalanceDest','newbalanceDest','nameOrig','nameDest','isFlaggedFraud'],axis=1,inplace=True)
```
✓ 0.1s

```
data.columns
```
✓ 0.0s

```
Index(['step', 'type', 'amount', 'isFraud', 'Actual_amount_orig',
       'Actual_amount_dest'],
      dtype='object')
```

Rechecking VIF



```
from statsmodels.stats.outliers_influence import variance_inflation_factor
for i in range(0,len(data.columns)):
    print(variance_inflation_factor(data.values,i))
```
[117] ✓ 17.8s

```
2.089797375798732
2.03928063055464
3.882315593200927
1.185547495038135
1.293012798377302
3.773312296201784
```

Thus the strong colinearity is removed. And we can proceed with model building

05 TRAINING AND TESTING DATA

FORMATION OF INDEPENDENT AND DEPENDENT VARIABLES FOR MODEL TRAINING AND TESTING

Test Split	**25 %**
Random State	**1**
Training and test data scale	**Robust Scalar**

SCALING USING ROBUST SCALAR

As we can see above, we need to normalize the range of features in a dataset before applying any machine learning algorithm on them. As we have outliers in our dataset we will go for Robust Scaler as it is less prone to outliers.
As robust Scaler works with median and Inter quartile range

MODEL BUILDING

MODEL CHOICE:

Highly unbalanced data as Percentage of Legit transactions= 99.87 % and Percentage of Fraud transactions= 0.13 %. SO **DECISION TREES** AND **RANDOM FORESTS** ARE GOOD METHODS FOR IMBALANCED DATA.

THUS WE SELECTED THE FOLLOWING ALGORITHMS:

DECISION TREE

RANDOM FOREST

LOGISTIC REGRESSION

KNN

While Creating Model

- We have dropped these two columns ‘nameOrig’, ‘nameDest’ which will not cause much impact on our model.
- And Transformed 4 columns into two columns for adjusting the VIF.

MODEL COMPARISION

Models	Accuracy Score	Precision	Recall	AUC
Logistic Regression	0.9992	0.89	0.46	0.73
Decision Tree(Entropy)	0.9994	0.81	0.78	0.89
Random Forest	0.9996	0.95	0.74	0.87
KNN	0.9994	0.88	0.70	0.85

- We have seen that Accuracy of both Random Forest and Decision Tree is almost equal, although teh precision of Random Forest is more. In a fraud detection model, Precision is highly important because rather than predicting normal transactions correctly we want Fraud transactions to be predicted correctly and Legit to be left off

- Also the reason we have chosen this model is because of highly unbalanced dataset (Legit: Fraud :: 99.87:0.13). Random forest makes multiple decision trees which makes it easier (although time taking) for model to understand the data in a simpler way since Decision Tree makes decisions in a boolean way.

CONCLUSIONS

LEVEL 1 CHECK FOR FRADULENT TRANSACTIONS :

- Source of transaction is out of anamoly.
- Sender and receiver of the transaction are legit and verified with background check.
- Past transactions can be used to predict the authenticity of the party.

PREVENTION METHODS:

- Do no disclose OTP's, sensitive passwords and other data that can lead to unforseen circumstances.
- Make sure the device from which the transaction will take place is secure and malware/spyware free.
- Keep the devices security patch up to date.
- Immediately inform the cyber cell if any suspicious action takes place without your notice.
- Beware of spams,spoofs,and do not fall for social engineering of the attackers.