

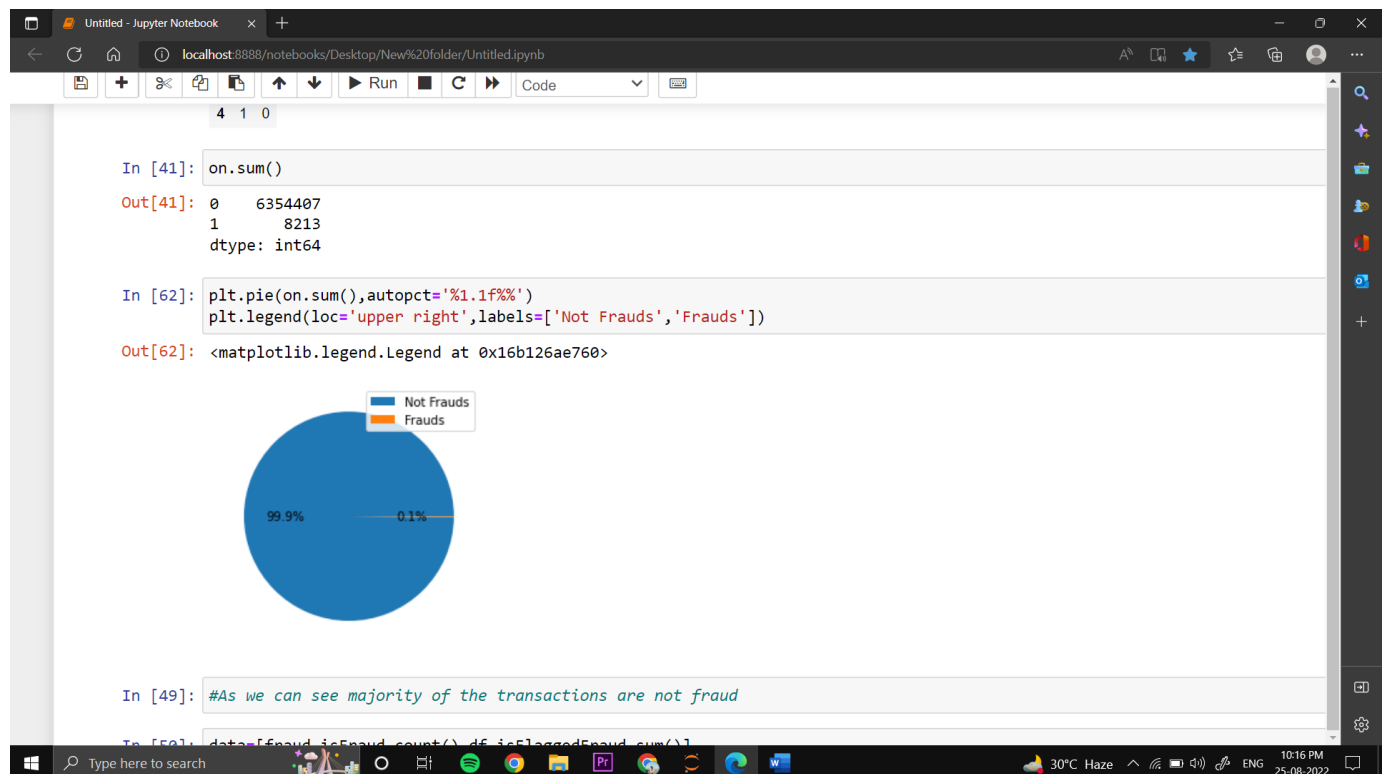
## Data cleaning including missing values, outliers and multi-collinearity.

From the figure it can be seen that there are no null values and therefore there is not need to perform work on null values.

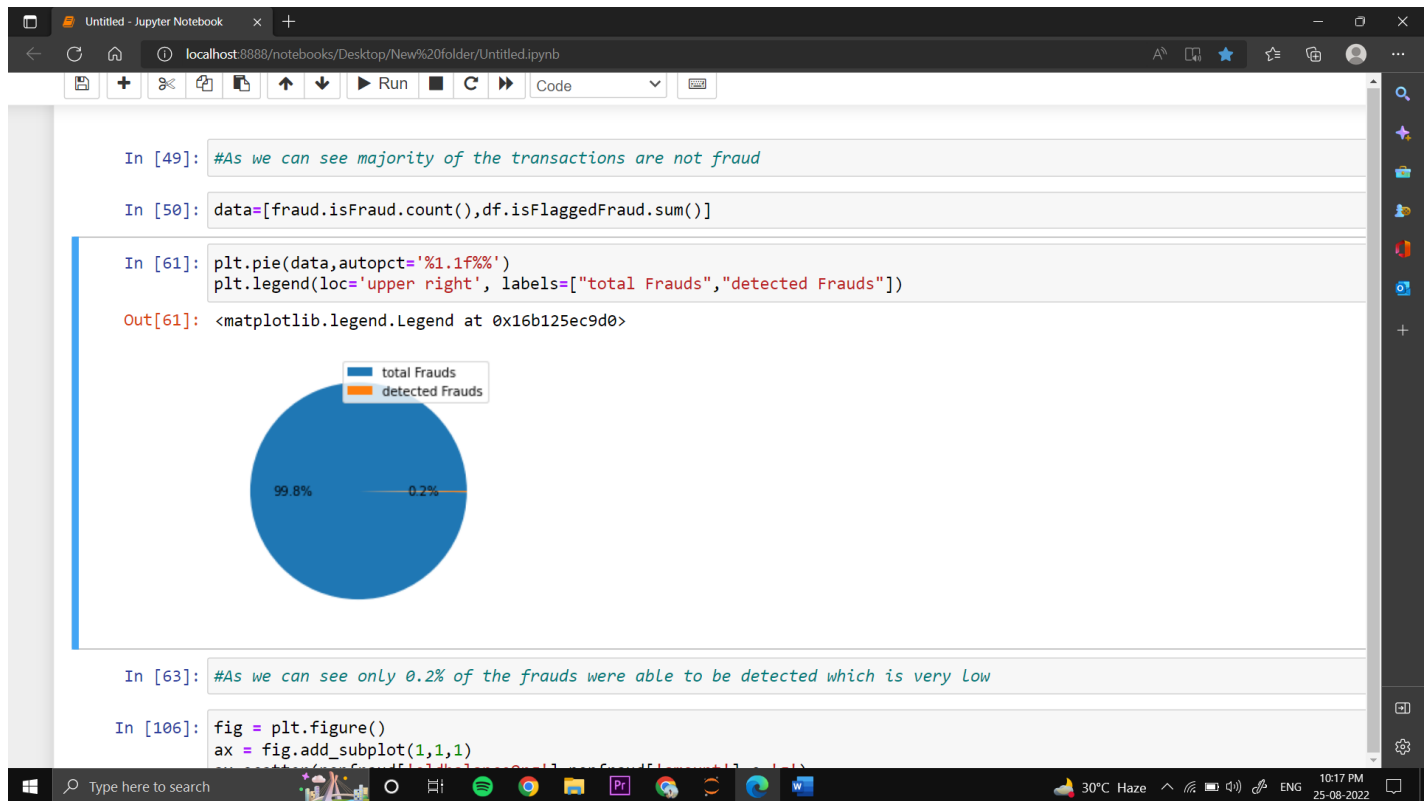
It can however be seen that there are some outliers in the code which can be removed by the following code and it has somewhat increased the efficiency of the model.

## Describe your fraud detection model in elaboration.

First, after finding out if there are null values or not I have performed data analysis. In which I have gathered some information like 99.9% of the transactions has no fraud where 0.1% has it.



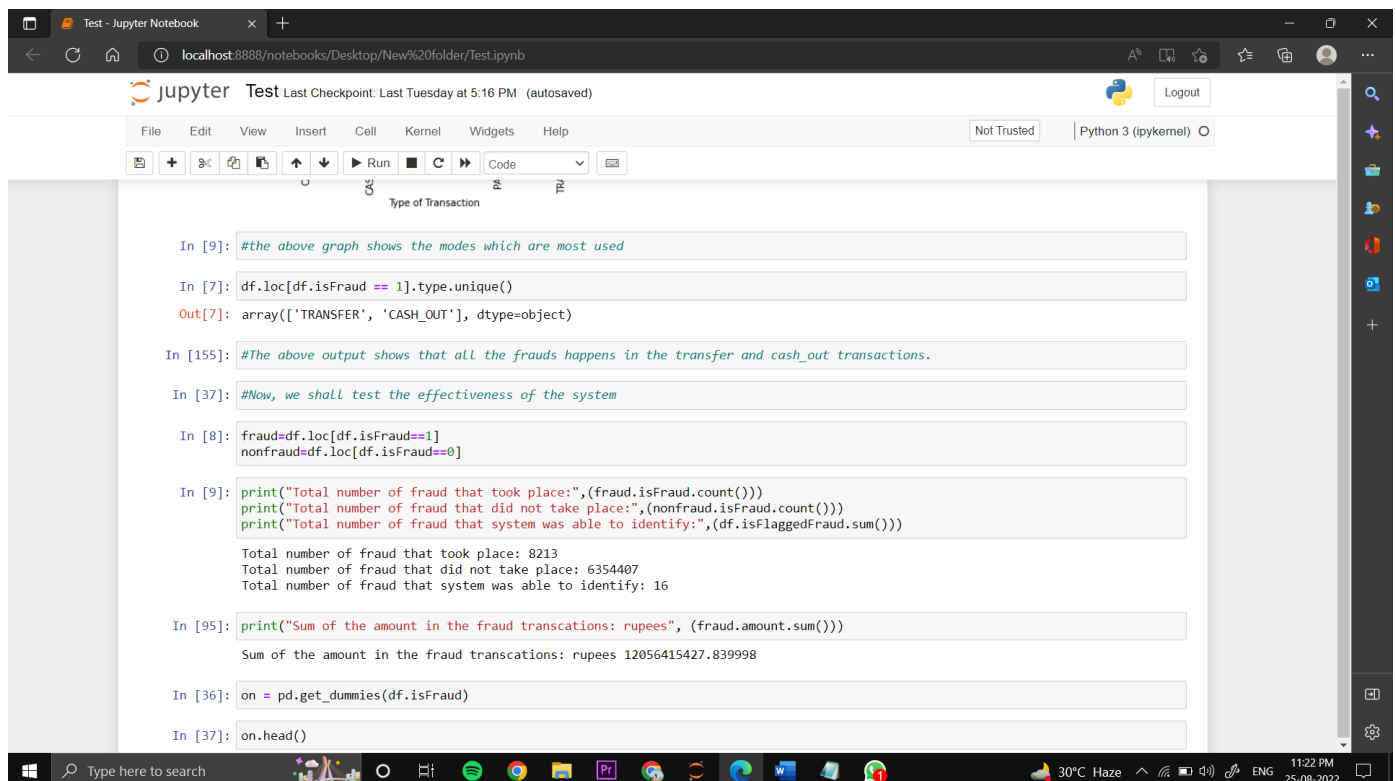
After that I have found out the efficiency of the 'isFlaggedFraud' to figure out fraud transactions which is very low since it was able to find out 0.2% of detected frauds.



Once that was done I tried to find out the relation of columns with the fraud transactions and almost all columns had some dependence on the fraud data except the 'nameOrig','nameDest','isFlaggedFraud'.

In addition to this it was also found out that all the fraud transactions were happening in the 'Transfer' and 'Cash out'.

The data is being divided into two datasets fraud and nonfraud.



And graphs were formed which gave me insights that in which range of amount does the fraud takes place.

After this the model was trained and accuracy was determined. However, after removing the outliers there was a slight increase in its accuracy and it was finally at 78%.

### How did you select variables to be included in the model?

Here there is clearly no correlation between the name and the fraud since all of them are unique in nature. Also, the same case is with the 'isflaggedfraud'.

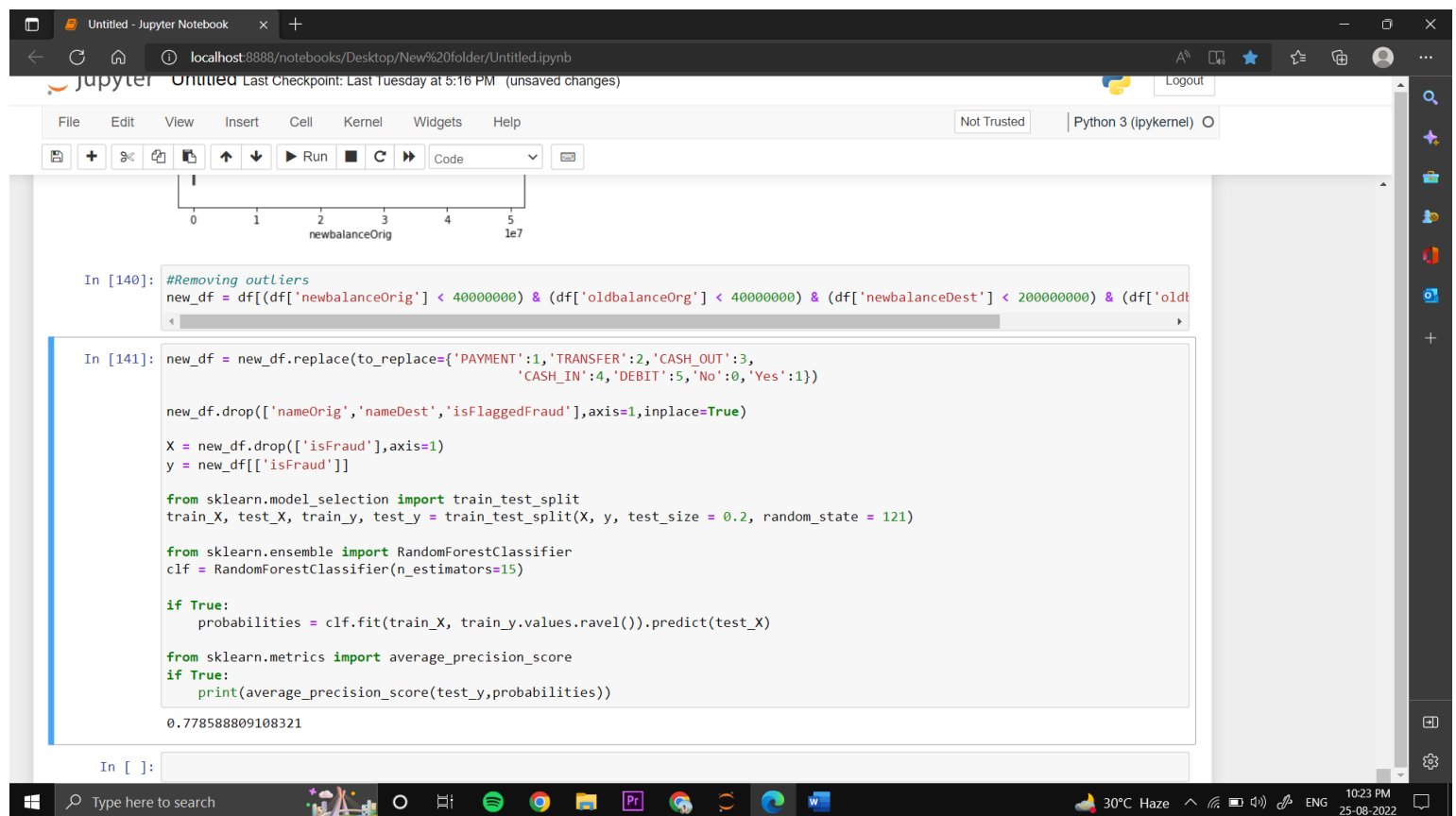
Furthermore, on creating graphs of fraud and nonfraud data we realize that most of the frauds are happening in a certain range and that is around 0 to 1 crore.

Later, we also see that the 'transfer' and the 'cash\_out' are the only two transaction in which fraud occurs.

Based on these things I was able to select variables.

### Demonstrate the performance of the model by using best set of tools.

Here, we have selected the variables and after that removed outliers and applied the randomforestclassifiers.



The screenshot shows a Jupyter Notebook interface with the following code in two cells:

```
In [140]: #Removing outliers
new_df = df[(df['newbalanceOrig'] < 40000000) & (df['oldbalanceOrig'] < 40000000) & (df['newbalanceDest'] < 200000000) & (df['oldbalanceDest'] < 200000000)]

In [141]: new_df = new_df.replace(to_replace={'PAYMENT':1,'TRANSFER':2,'CASH_OUT':3,
                                             'CASH_IN':4,'DEBIT':5,'No':0,'Yes':1})

new_df.drop(['nameOrig','nameDest','isFlaggedFraud'],axis=1,inplace=True)

X = new_df.drop(['isFraud'],axis=1)
y = new_df[['isFraud']]

from sklearn.model_selection import train_test_split
train_X, test_X, train_y, test_y = train_test_split(X, y, test_size = 0.2, random_state = 121)

from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=15)

if True:
    probabilities = clf.fit(train_X, train_y.values.ravel()).predict(test_X)

from sklearn.metrics import average_precision_score
if True:
    print(average_precision_score(test_y,probabilities))

0.778588809108321
```

This gives us an accuracy of 78%.

### What are the key factors that predict fraudulent customer?

The key factors that are responsible for fraudulent customers are

- The transactions should be from 'Transfer' or 'Cash\_Out'
- Most of the frauds range from 0 to 1 crore, whereas the nonfraud transactions range from 0 to 9 crore. Not a single fraud was above than 1 crore.

### Do these factors make sense? If yes, How? If not, How not?

Yes, these factors make sense as using the above two mentioned points we will be able to narrow down our research and it would help us to find out the frauds very easily.

**What kind of prevention should be adopted while company update its infrastructure?**

Instead of having the threshold of 200 rupees the company should keep it near 1 crore and should later equate the transactional balances.

The equation should be  $(oldorg - olddest) = (neworg - newdest)$

If the values are same then we could say that the transaction was not a fraud.

This feature could be incorporated in the system.

**Assuming these actions have been implemented, how would you determine if they work?**

We could determine using the isFlaggedFraud. That is to see if the 'isFlaggedFraud' equal to the number of frauds that have occurred in the system. If the numbers are same, then the system is perfect. However, if the numbers are not similar, then we need to incorporate other methods to this.