```
#import all required libraries
import pandas as pd
import numpy as np
insaid = pd.read_csv("/content/drive/MyDrive/Internships/INSAID_AUGUST/Fraud.csv")
insaid.head()
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDe
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M19797871
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M20442822
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C5532640
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C389970
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M12307017

Task1: Data cleaning including null values, outliers and multicollinearilty

Null values

```
insaid.isnull().sum()
```

```
step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

There are no null values in any of the columns in data set.

Outliers

```
insaid_numeric_dataset = insaid.select_dtypes(include = [np.number])
insaid_numeric_col = insaid_numeric_dataset.columns.values
for col in insaid_numeric_col:
        missing = insaid[col].isnull()
        no_of_missing = np.sum(missing)
```

```
        if no_of_missing > 0:
            medn =insaid[col].median()
            insaid[col] =insaid[col].fillna(medn)
```

```
### step
abs((insaid[insaid_numeric_col[0]].mean() ) - (insaid[insaid_numeric_col[0]].median() ))
```

```
     4.397245631516569
```

```
###     amount
abs((insaid[insaid_numeric_col[1]].mean() ) - (insaid[insaid_numeric_col[1]].median() )  )
```

```
     104989.96354912291
```

```
###  oldbalanceOrg
abs((insaid[insaid_numeric_col[2]].mean() ) - (insaid[insaid_numeric_col[2]].median() ))
```

```
     819675.1040744851
```

```
### newbalanceOrig
insaid_numeric_col[3]
abs((insaid[insaid_numeric_col[3]].mean() ) - (insaid[insaid_numeric_col[3]].median() ))
```

```
     855113.6685785672
```

```
### oldbalanceDest
insaid_numeric_col[4]
abs((insaid[insaid_numeric_col[4]].mean() ) - (insaid[insaid_numeric_col[4]].median() ))
```

```
     967996.0015196998
```

```
### newbalanceDest
insaid_numeric_col[5]
abs((insaid[insaid_numeric_col[5]].mean() ) - (insaid[insaid_numeric_col[5]].median() ))
```

```
     1010334.9582020713
```

```
### isFraud
insaid_numeric_col[6]
abs((insaid[insaid_numeric_col[6]].mean() ) - (insaid[insaid_numeric_col[6]].median() ))
```

```
     0.001290820448180152
```

```
### isFlaggedFraud
insaid_numeric_col[7]
abs((insaid[insaid_numeric_col[7]].mean() ) - (insaid[insaid_numeric_col[7]].median() ))
```

```
     2.51468734577894e-06
```

oldbalanceOrg , **newbalanceOrig** , **oldbalanceDest** , **newbalanceDest** all have outliers.

Remove outliers using Quantile based flooring and clapping

Amount

```
print(insaid['amount'].skew())
lessthan_10 = insaid['amount'].quantile(0.10)
morethan_90 = insaid['amount'].quantile(0.90)
```

```
30.99394948249038
```

```
insaid['amount'] = np.where(insaid['amount']< lessthan_10 , lessthan_10, insaid['amount'])
insaid['amount'] = np.where(insaid['amount']> morethan_90 , morethan_90, insaid['amount'])
insaid['amount'].skew()
```

```
0.8046460444556197
```

newbalanceOrig

```
print(insaid['newbalanceOrig'].skew())

insaid['newbalanceOrig'] = np.where(insaid['newbalanceOrig']< lessthan_10 , lessthan_10, i
insaid['newbalanceOrig'] = np.where(insaid['newbalanceOrig']> morethan_90 , morethan_90, i
insaid['newbalanceOrig'].skew()
```

```
5.176884001159233
1.2382985209345365
```

oldbalanceDest

Double-click (or enter) to edit

```
print(insaid['oldbalanceDest'].skew())
insaid['oldbalanceDest'] = np.where(insaid['oldbalanceDest']< lessthan_10 , lessthan_10, i
insaid['oldbalanceDest'] = np.where(insaid['oldbalanceDest']> morethan_90 , morethan_90, i
insaid['oldbalanceDest'].skew()
```

```
19.921757915791062
0.10285904254115923
```

newbalanceDest

```
print(insaid['newbalanceDest'].skew())
insaid['newbalanceDest'] = np.where(insaid['newbalanceDest']< lessthan_10 , lessthan_10, i
insaid['newbalanceDest'] = np.where(insaid['newbalanceDest']> morethan_90 , morethan_90, i
insaid['newbalanceDest'].skew()
```

```
19.352302057660165
-0.06748173621229811
```

Outliers are removed.

Multicollinearity

Building model using all variables

```
from sklearn.linear_model import LogisticRegression
logistic1= LogisticRegression()
###fitting logistic regression for active customer on rest of the variables#######
logistic1.fit(insaid[["step"]+['amount']+['oldbalanceOrg']+['newbalanceOrig']+['oldbalance
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:760: DataConversic
  y = column_or_1d(y, warn=True)
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
print("Intercept", logistic1.intercept_)
print("Coefficients", logistic1.coef_)
```

```
Intercept [-2.68107391e-05]
Coefficients [[-4.12345757e-03  4.03105915e-06  1.01923537e-05 -9.44858086e-04
  -1.22217228e-06 -1.11757954e-05  6.68196351e-08]]
```

```
import statsmodels.api as sm
m1=sm.Logit(insaid['isFraud'],insaid[["step"]+['amount']+['oldbalanceOrg']+['newbalanceOri
m1.fit()
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarnir
  import pandas.util.testing as tm
Optimization terminated successfully.
         Current function value: 0.009679
         Iterations 16
<statsmodels.discrete.discrete_model.BinaryResultsWrapper at 0x7f5de14b7350>
```

step, amount, oldbalanceOrg, newbalanaceOrig, oldbalanceDest, newbalanceDest these all variables are impactful variables.

Task 2: Describe your fraud detection model in ellaboration

I build a Logistic Model and found that the accuracy ofmy model was 99%. I concluded that using the confusion matrix. In the model I considered all the variable and found that only few of them were impactful. they are step, amount, oldbalanceOrg, newbalanaceOrig, oldbalanceDest, newbalanceDest.The most impact full is **newbalanceOrig** column.

Given any customer information such as step, amount, oldbalanceOrg, newbalanaceOrig, oldbalanceDest, newbalanceDest we can use our MOdel outputs to tell that whether the customer is Fraud or not. Simply by putting the data in following formula:

```
1.coef_[0][5] * f))/1+ (exp(logistic1.coef_[0][0])*a + logistic1.coef_[0][1]*b + logistic1

      File "<ipython-input-20-7abf17b7932e>", line 1
        y = (exp(logistic1.coef_[0][0])*a + logistic1.coef_[0][1]*b + logistic1.coef_[0]
    [2]*c + logistic1.coef_[0][3]*d +  logistic1.coef_[0][4] * e + logistic1.coef_[0][5]
    * f))/1+ (exp(logistic1.coef_[0][0])*a + logistic1.coef_[0][1]*b +
    logistic1.coef_[0][2]*c + logistic1.coef_[0][3]*d +  logistic1.coef_[0][4] * e +
    logistic1.coef_[0][5] * f))

    ^
```

where a = step , b = smount, c = oldbalanceOrg, d = newbalanaceOrig, e = oldbalanceDest, f = newbalanceDest

Task 3: How did you select variables to be included in the model.

I built a logistic regression model, from the summary of this model we concluded that the variables with pvalue less than 0.05 we reject the null hypothesis that coeficient of that variable is 0, so this are impactful variablees.

Task4: Demostrate the performance of the model by using best set of rtools

```
# Confusion Matrix & Accuracy
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

predict1=logistic1.predict(insaid[['step']+['amount']+['oldbalanceOrg']+['newbalanceOrig']
```

```
predict1

cm1 = confusion_matrix(insaid[['isFraud']],predict1)
print(cm1)



print("col sums", sum(cm1))
total1=sum(sum(cm1))
print("Total", total1)



accuracy1=(cm1[0,0]+cm1[1,1])/total1
accuracy1
```

Task 5: What are the key factors that predict fradulent customers.

```
# Wald Chi-square value
abs(m1.fit().tvalues).sort_values(ascending=False)
```

The most impactful variable is newbalanceOrig

Task6: Do these factors make sense?

Yes, This variable **newbalanceOrig** is the impactful variable as amount the person is adding as his/her new balance will make considerably tell whether the person will turn out to be faulted or not.

Task7 What kind of prevention should be adopted while company update its infrastructure

More importance should be ginven to security, may that be physical or digital. This can be done by setting up CCTV cameras at the place of customer arrival so that the fraud customer if detected after can be spotted easily.

Task 8 Assuming these actions have been implemented how would you determine if they work?

If this techniques are implemented then surely we can take action physically on the customer if he/she is detected fraud by our Model.