

```
In [1]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

In [2]: data = pd.read_csv('D:\Accredian task\Fraud.csv')

In [3]: print(data[0:10],"\n")

      step      type  amount  nameOrig  oldbalanceOrg  newbalanceOrig  \
0      1  PAYMENT   9839.64  C1231006815    170136.00    160296.36
1      1  PAYMENT   1864.28  C1666544295    21249.00    19384.72
2      1  TRANSFER    181.00  C1305486145     181.00         0.00
3      1  CASH OUT    181.00  C040083671     181.00         0.00
4      1  PAYMENT  11668.14  C2048537720    41554.00    29885.86
5      1  PAYMENT   7817.71  C90045638   53860.00    46042.29
6      1  PAYMENT   7107.77  C154988899   183195.00    176087.23
7      1  PAYMENT   7861.64  C1912850431   176087.23    168225.59
8      1  PAYMENT   4024.36  C1265012928    2671.00         0.00
9      1   DEBIT   5337.77  C712410124    41720.00    36382.23

      nameDest  oldbalanceDest  newbalanceDest  isFraud  isFlaggedFraud
0  M1979787155         0.0         0.00         0         0
1  M2044282225         0.0         0.00         0         0
2  C553264065         0.0         0.00         1         0
3  C38997010         21182.0         0.00         1         0
4  M1230701703         0.0         0.00         0         0
5  M573487274         0.0         0.00         0         0
6  M408069119         0.0         0.00         0         0
7  M633326333         0.0         0.00         0         0
8  M1176932104         0.0         0.00         0         0
9  C195600860         41898.0        40348.79         0         0

In [4]: #print the complete shape of dataset
print("Shape of Complete Dataset")
print(data.shape,"\n")

Shape of Complete Dataset
(6362620, 11)

In [5]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
#   Column      Dtype
---  -
0   step        int64
1   type         object
2   amount      float64
3   nameOrig     object
4   oldbalanceOrg float64
5   newbalanceOrig float64
6   nameDest     object
7   oldbalanceDest float64
8   newbalanceDest float64
9   isFraud      int64
10  isFlaggedFraud int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB

In [6]: data.isnull().any().any()

Out[6]: False

In [7]: false = data[data['isFraud']==1]
true = data[data['isFraud']==0]
n=len(false)/float(len(true))
print (n)
print('False Detection : {}'.format(len(data[data['isFraud']==1])))
print('True Detection:{}'.format(len(data[data['isFraud']==0])), "\n")

0.0012924888191769902
False Detection : 8213
True Detection:6354407

In [8]: data['isFraud'].value_counts(normalize=True)*100

Out[8]:
0    99.870918
1     0.129082
Name: isFraud, dtype: float64

In [9]: print("False Detection Transaction")
print("")
print(false.amount.describe(), "\n")

#True Detection Transaction
print("True Detection Transaction")
print("")
print(true.amount.describe(), "\n")

False Detection Transaction

count      8.213000e+03
mean       1.467967e+06
std        2.404253e+06
min         0.000000e+00
25%        1.270913e+05
50%        4.414234e+05
75%        1.517771e+06
max         1.000000e+07
Name: amount, dtype: float64

True Detection Transaction

count      6.354407e+06
mean       1.781970e+05
std        5.962370e+05
min         1.000000e-02
25%        1.336840e+04
50%        7.468472e+04
75%        2.083648e+05
max         9.244552e+07
Name: amount, dtype: float64

In [10]: #select all columns except the last for all rows
X=data.iloc[:, :-1].values
#select the las columns of all rows
Y=data.iloc[:, -1].values

print(X.shape)
print('-----')
print(Y.shape)

(6362620, 10)
-----
(6362620,)

In [11]: df = data.drop(['type', 'nameOrig', 'nameDest', 'isFraud'], axis=1)

In [12]: df.head()

Out[12]:
   step  amount  oldbalanceOrg  newbalanceOrig  oldbalanceDest  newbalanceDest  isFlaggedFraud
0      1   9839.64      170136.0      160296.36         0.0         0.0             0
1      1   1864.28      21249.0      19384.72         0.0         0.0             0
2      1    181.00        181.0         0.00         0.0         0.0             0
3      1   181.00        181.0         0.00      21182.0         0.0             0
4      1  11668.14      41554.0      29885.86         0.0         0.0             0

In [13]: #select all columns except the last for all rows
X=df.iloc[:, :-1].values
#select the las columns of all rows
Y=df.iloc[:, -1].values

print(X.shape)
print('-----')
print(Y.shape)

(6362620, 6)
-----
(6362620,)

In [ ]:

In [14]: import pandas as pd
from sklearn.model_selection import train_test_split

#train test split method
X_train, X_test, Y_train, Y_test= train_test_split(X,Y, test_size=0.3)

In [15]: import numpy as np
import scipy as sp

Y_train.shape

Out[15]:
(4453834,)

In [ ]:

In [16]: #Decision Tree Classifier
from sklearn.tree import DecisionTreeClassifier
classifier=DecisionTreeClassifier(max_depth=4)
classifier.fit(X_train, Y_train)
predicted=classifier.predict(X_test)
print("\n Predicted value:\n",predicted)

Predicted value:
[0 0 0 ... 0 0 0]

In [17]: from sklearn import metrics
DecisionTree= metrics.accuracy_score(Y_test, predicted) * 100
print("\n The Accuracy Score Using Algorithm Decision Tree Classifier: ", DecisionTree)

The Accuracy Score Using Algorithm Decision Tree Classifier: 99.99979044272118

In [18]: from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score

In [19]: #Precision
print('Precision')
# use TP/(TP+FP) where (TP=True Positif, TN=True Negative, FP = False Positive, FN = False Negative)
precision=precision_score(Y_test, predicted, pos_label=1)*100
print("\n Score Precision : \n",precision )

Precision

Score Precision :
50.0

In [20]: #Recall
# Recall = TP / (TP + FN)
print("Recall")
recall=recall_score(Y_test, predicted, pos_label=1)*100
print("\n Recall Score : \n", recall)

Recall

Recall Score :
25.0

In [21]: #F1-Score
print('F1-Score')
fscore=f1_score(Y_test, predicted, pos_label=1)*100
print("\n F1 Score : \n", fscore)

F1-Score

F1 Score :
33.33333333333333

In [22]: pip install nbconvert

Requirement already satisfied: nbconvert in c:\users\nadee\anaconda\lib\site-packages (6.4.4)
Requirement already satisfied: testpath in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (0.5.0)
Requirement already satisfied: pygments>=2.4.1 in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (2.11.2)
Requirement already satisfied: nbformat>=4.4 in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (5.3.0)
Requirement already satisfied: entrypoints>=0.2.2 in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (0.4)
Requirement already satisfied: beautifulsoup4 in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (4.11.1)
Requirement already satisfied: mistune>=0.8.1 in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (0.8.4)
Requirement already satisfied: bleach in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (4.1.0)
Requirement already satisfied: pandocfilters<=1.4.1 in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (1.5.0)
Requirement already satisfied: defusedxml in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (0.7.1)
Requirement already satisfied: jupyter-core in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (4.9.2)
Requirement already satisfied: jinja2>=2.4 in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (2.11.3)
Requirement already satisfied: jupyterlab-pygments in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (0.1.2)
Requirement already satisfied: traitlets>=5.0 in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (5.1.1)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in c:\users\nadee\anaconda\lib\site-packages (from nbconvert) (0.5.13)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\nadee\anaconda\lib\site-packages (from jinja2>=2.4->nbconvert) (2.0.0)
Requirement already satisfied: pyzmq>=13 in c:\users\nadee\anaconda\lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (6.1.12)
Requirement already satisfied: nest-asyncio in c:\users\nadee\anaconda\lib\site-packages (from nbclient<0.6.0,>=0.5.0->nbconvert) (1.5.5)
Requirement already satisfied: tornado>=4.1 in c:\users\nadee\anaconda\lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (22.3.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\nadee\anaconda\lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (2.8.2)
Requirement already satisfied: pywin32>=1.0 in c:\users\nadee\anaconda\lib\site-packages (from jupyter-core->nbconvert) (302)
Requirement already satisfied: fastjsonschema in c:\users\nadee\anaconda\lib\site-packages (from nbformat>=4.4->nbconvert) (2.15.1)
Requirement already satisfied: jsonschema>=2.6 in c:\users\nadee\anaconda\lib\site-packages (from nbformat>=4.4->nbconvert) (4.4.0)
Requirement already satisfied: attrs>=17.4.0 in c:\users\nadee\anaconda\lib\site-packages (from jsonschema>=2.6->nbformat>=4.4->nbconvert) (21.4.0)
Requirement already satisfied: six>=1.5 in c:\users\nadee\anaconda\lib\site-packages (from python-dateutil>=2.1->jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (0.18.0)
Requirement already satisfied: soupsieve>1.2 in c:\users\nadee\anaconda\lib\site-packages (from bleach->nbconvert) (2.3.1)
Requirement already satisfied: packaging in c:\users\nadee\anaconda\lib\site-packages (from bleach->nbconvert) (21.3)
Requirement already satisfied: webencodings in c:\users\nadee\anaconda\lib\site-packages (from bleach->nbconvert) (0.5.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\nadee\anaconda\lib\site-packages (from packaging->bleach->nbconvert) (3.0.4)
Note: you may need to restart the kernel to use updated packages.
```