Name : SHEIKPAREETH

Machine Learning, Supervised, classification, Decision tree ¶

In [1]:

```
#Import the libraries and put nicknames
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:

```
#Reading the dataset
dataset=pd.read_csv("fraud.csv")
```

In [3]:

```
#It have 16426 rows and 9 columns
dataset
```

Out[3]:

	step	type	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalar
0	1	PAYMENT	9839.64	170136.00	160296.36	0.00	
1	1	PAYMENT	1864.28	21249.00	19384.72	0.00	
2	1	PAYMENT	11668.14	41554.00	29885.86	0.00	
3	1	PAYMENT	7817.71	53860.00	46042.29	0.00	
4	1	PAYMENT	7107.77	183195.00	176087.23	0.00	
...	...	...	...	...	...	...	
16421	743	CASH_OUT	339682.13	339682.13	0.00	0.00	33
16422	743	TRANSFER	6311409.28	6311409.28	0.00	0.00	
16423	743	CASH_OUT	6311409.28	6311409.28	0.00	68488.84	637
16424	743	TRANSFER	850002.52	850002.52	0.00	0.00	
16425	743	CASH_OUT	850002.52	850002.52	0.00	6510099.11	736

16426 rows × 9 columns

In [4]:

```
#Above the value is categorical value so i have used "one hot encoding method" (we cannot a
#I have removed the duplcates or dummy value
dataset=pd.get_dummies(dataset,drop_first=True)
```

In [5]:

```
#provide input columns name for the easily without seeing the table
dataset.columns
```

Out[5]:

```
Index(['step', 'amount', 'oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDes
t',
       'newbalanceDest', 'isFraud', 'isFlaggedFraud', 'type_CASH_OUT',
       'type_DEBIT', 'type_PAYMENT', 'type_TRANSFER'],
      dtype='object')
```

In [6]:

```
#Put the input and aouput column name
indep=dataset[["step", "amount", "oldbalanceOrg", "newbalanceOrig", "oldbalanceDest", "newb
dep=dataset[["isFlaggedFraud"]]
```

In [ ]:

```
#output values (dependent)
dep
```

In [ ]:

```
#Input values (independent)
indep
```

In [7]:

```
#split inti training set and test test
#take 30% of sample
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(indep,dep,test_size=0.3,random_state=0)
```

In [8]:

```
#model creation process for classification
#formul loaded this libraries and use fit method substitute the value
#finally create a model
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier.fit(X_train,y_train)
```

Out[8]:

```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=Non
e,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=0, splitter='best')
```

In [9]:

```
#Evaluation metrics to use test set
#y_test output of prdicted value
y_pred=classifier.predict(X_test)
```

In [10]:

```
#Calculate confusion matrix to evaluate the accuracy of a classification
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In [11]:

```
print(cm)
```

```
[[4926    0]
 [   1    1]]
```

In [12]:

```
#find clssification report (precision,recall,f1-score,accuracy )
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
```

In [13]:

```
print(clf_report)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4926
           1       1.00      0.50      0.67         2

    accuracy                           1.00      4928
   macro avg       1.00      0.75      0.83      4928
weighted avg       1.00      1.00      1.00      4928
```