Name : SHEIKPAREETH

Machine Learning, Supervised, classification, SVM_kernal classification

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 [ ]:

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

In [5]:

```
#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 [6]:

```
#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 [7]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724:
DataConversionWarning: A column-vector y was passed when a 1d array was expe
cted. Please change the shape of y to (n_samples, ), for example using ravel
().
  y = column_or_1d(y, warn=True)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWa
rning: The default value of gamma will change from 'auto' to 'scale' in vers
ion 0.22 to account better for unscaled features. Set gamma explicitly to 'a
uto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

Out[7]:

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=0,
    shrinking=True, tol=0.001, verbose=False)
```

In [8]:

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

In [9]:

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

In [10]:

```
print(cm)
```

```
[[4926    0]
 [   2    0]]
```

In [11]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\classification.p
y:1437: UndefinedMetricWarning: Precision and F-score are ill-defined and be
ing set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

In [12]:

```
print(clf_report)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4926
           1       0.00      0.00      0.00         2

    accuracy                           1.00      4928
   macro avg       0.50      0.50      0.50      4928
weighted avg       1.00      1.00      1.00      4928
```