In [1]:
```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
import pandas as pd
```

In [6]:
```
df=pd.read_excel("f:/dataset/classification/creditcard_copy.xlsx")
X=df.iloc[:,1:-2].values
y=df.iloc[:,-1].values
```

In [7]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,random_state=1)
model=KNeighborsClassifier()
model.fit(X_train,y_train)
print(model.score(X_train,y_train))
print(model.score(X_test,y_test))
```

```
0.9855567226890757
0.9921259842519685
```

In [8]:
```
#Confusion Matrix---->it shows model behaviour for each class
from sklearn.metrics import confusion_matrix
pred_train=model.predict(X_train)
confusion_matrix(y_train,pred_train)
```

Out[8]:
```
array([[3420,    4],
       [  51,  333]], dtype=int64)
```

In [9]:
```
3471-3424
```

Out[9]:
```
47
```

In [15]:
```
Precision_Score_0=3420/3471
print(Precision_Score_0)

Precision_Score_1=333/337
print(Precision_Score_1)

Recall_score_0=3420/3424
print(Recall_score_0)

Recall_score_1=333/384
print(Recall_score_1)

#f1-score=harmonic mean of precision and recall

f1_0=2*Precision_Score_0*Recall_score_0/(Precision_Score_0+Recall_score_0)
print(f1_0)

f1_1=2*Precision_Score_1*Recall_score_1/(Precision_Score_1+Recall_score_1)
print(f1_1)
```

```
0.9853068280034573
0.9881305637982196
0.9988317757009346
0.8671875
0.9920232052211747
0.9237170596393898
```

In [ ]:
```

```