In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
```

In [2]:

```
bankdata = pd.read_csv("C:/Users/DEEP SUMAN/Desktop/Lab Prg Ready/bill_authentication.csv")
```

In [3]:

```
bankdata.shape
```

Out[3]:

```
(1372, 5)
```

In [4]:

```
bankdata.head(10)
```

Out[4]:

	Variance	Skewness	Curtosis	Entropy	Class
0	3.62160	8.6661	-2.80730	-0.44699	0
1	4.54590	8.1674	-2.45860	-1.46210	0
2	3.86600	-2.6383	1.92420	0.10645	0
3	3.45660	9.5228	-4.01120	-3.59440	0
4	0.32924	-4.4552	4.57180	-0.98880	0
5	4.36840	9.6718	-3.96060	-3.16250	0
6	3.59120	3.0129	0.72888	0.56421	0
7	2.09220	-6.8100	8.46360	-0.60216	0
8	3.20320	5.7588	-0.75345	-0.61251	0
9	1.53560	9.1772	-2.27180	-0.73535	0

In [5]:

```
X = bankdata.drop('Class', axis=1)
y = bankdata['Class']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
```

In [7]:

```
# PROCESS 1
model = GaussianNB()
```

In [8]:

```
kfold=model_selection.KFold(n_splits=5)
results = model_selection.cross_val_score(model,X_train, y_train, cv=kfold)
```

In [9]:

```
print("Results:", results)
print("Mean Results:", results.mean())
```

```
Results: [0.85       0.82727273 0.87214612 0.84931507 0.84018265]
Mean Results: 0.8477833125778332
```

In [10]:

```
#PROCESS 2
model = GaussianNB()
model.fit(X_train, y_train)
```

Out[10]:

```
▾ GaussianNB
GaussianNB()
```

In [11]:

```
y_pred = model.predict(X_test)
```

In [12]:

```
model.score(X_test, y_test)
```

Out[12]:

```
0.84
```

In [13]:

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
[[151  19]
 [ 25  80]]
              precision    recall  f1-score   support

           0       0.86      0.89      0.87       170
           1       0.81      0.76      0.78       105

    accuracy                           0.84       275
   macro avg       0.83      0.83      0.83       275
weighted avg       0.84      0.84      0.84       275
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js