In [5]:

```
import pandas as pd
import numpy as np
```

In [9]:

```
pdata= pd.read_csv('C:\\Users\\Jai mata di\\Downloads\\diabetes.csv')
pdata.head()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28

In [12]:

```
pdata.dtypes
```

Out[12]:

```
Pregnancies                 int64
Glucose                     int64
BloodPressure               int64
SkinThickness               int64
Insulin                     int64
BMI                       float64
DiabetesPedigreeFunction  float64
Age                         int64
Outcome                     int64
dtype: object
```

In [15]:

```
pdata.describe()
```

Out[15]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabet
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

spliting data

In [16]:

```
from sklearn.model_selection import train_test_split
x=pdata.drop("Outcome",axis=1)
y=pdata[["Outcome"]]
```

In [17]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y, test_size=0.3,random_state=1)
```

In [20]:

```
from sklearn.naive_bayes import GaussianNB
model=GaussianNB()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
```

```
C:\Users\Jai mata di\anaconda3\lib\site-packages\sklearn\utils\validation.p
y:72: DataConversionWarning: A column-vector y was passed when a 1d array wa
s expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  return f(**kwargs)
```

In [22]:

```
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test,y_pred))
```

```
Accuracy: 0.7835497835497836
```

In [23]:

```
test_pred=model.predict(x_test)
print(metrics.classification_report(y_test,test_pred))
print(metrics.confusion_matrix(y_test,test_pred))
```

```
              precision    recall  f1-score   support

           0       0.80      0.88      0.84       146
           1       0.75      0.62      0.68        85

    accuracy                           0.78       231
   macro avg       0.77      0.75      0.76       231
weighted avg       0.78      0.78      0.78       231

[[128  18]
 [ 32  53]]
```