In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
A=pd.read_csv("C:/Users/HP/Downloads/diabetes.csv")
A.head()
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	O
0	6	148	72	35	0	33.6	0.627	50	
1	1	85	66	29	0	26.6	0.351	31	
2	8	183	64	0	0	23.3	0.672	32	
3	1	89	66	23	94	28.1	0.167	21	
4	0	137	40	35	168	43.1	2.288	33	

In [3]:
```
x=A.iloc[:,: -1].values
y=A.iloc[:, -1].values
```

In [4]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70,test_size=0.30)
```

In [ ]:
```

```

In [5]:
```
from sklearn.neighbors import KNeighborsClassifier
S=KNeighborsClassifier()
S.fit(x_train,y_train)
```

Out[5]:
```
KNeighborsClassifier()
```

In [6]:
```
y_pred=S.predict(x_test)
```

In [7]:
```
y_pred
```

Out[7]:
```
array([1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1], dtype=int64)
```

In [8]:
```
y_test
```

Out[8]:
```
array([0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,
       0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1], dtype=int64)
```

In [9]:
```
from sklearn.metrics import confusion_matrix
result=confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(result)
```

```
Confusion Matrix:
[[124  31]
 [ 33  43]]
```

In [10]:
```
from sklearn.metrics import  accuracy_score
result1=accuracy_score(y_test, y_pred)
print("Accuracy_score:")
print(result1)
```

```
Accuracy_score:
0.7229437229437229
```

In [11]:
```
from sklearn.metrics import classification_report
D=classification_report(y_test,y_pred,digits=4)
print(D)
```

```
              precision    recall  f1-score   support

           0     0.7898    0.8000    0.7949       155
           1     0.5811    0.5658    0.5733        76

    accuracy                         0.7229       231
   macro avg     0.6854    0.6829    0.6841       231
weighted avg     0.7211    0.7229    0.7220       231
```

In [12]:
```
from sklearn.model_selection import GridSearchCV #to access k value
P= {'n_neighbors' : [2,6,7,54,23,20,34,65,42,31,19,82,5]}
```

In [13]:
```
W=GridSearchCV(S, P,cv=10)
```

In [14]:
```
W.fit(x_train,y_train)
```

Out[14]:
```
GridSearchCV(cv=10, estimator=KNeighborsClassifier(),
             param_grid={'n_neighbors': [2, 6, 7, 54, 23, 20, 34, 65, 42, 31,
                                         19, 82, 5]})
```

In [15]:
```
# let's see the  best parameters according to gridsearch
W.best_params_
```

Out[15]:
```
{'n_neighbors': 19}
```

In [16]:
```
W.best_score_
```

Out[16]:
```
0.7634171907756813
```

In [ ]:
```

```