In [19]:

```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score,confusion_matrix
```

In [2]:

```python
data=pd.read_csv('diabetes.csv')
data.head()
```

Out[2]:

	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 [6]:

```python
not_zero=['Glucose','BloodPressure','SkinThickness','Insulin','BMI']
for colum in not_zero:
    mean=int(data[colum].mean())
    data[colum]=data[colum].replace(0,mean)
```

In [9]:

```python
data['Insulin']
```

Out[9]:

```
0       79
1       79
2       79
3       94
4      168
      ... 
763    180
764     79
765    112
766     79
767     79
Name: Insulin, Length: 768, dtype: int64
```

In [14]:

```
x=data.iloc[:,:8]
y=data.iloc[:,8]
```

In [16]:

```
#x
#y
```

In [17]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=13)
```

In [18]:

```
print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
(537, 8) (231, 8) (537,) (231,)
```

In [20]:

```
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
```

In [21]:

```
knn=KNeighborsClassifier(n_neighbors=11,p=2,metric='euclidean')
```

In [26]:

```
knn.fit(x_train,y_train)
```

Out[26]:

```
KNeighborsClassifier(metric='euclidean', n_neighbors=11)
```

In [28]:

```
y_predict=knn.predict(x_test)
```

In [29]:

```
accuracy_score(y_test,y_predict)
```

Out[29]:

```
0.7056277056277056
```

In [30]:

```
confusion_matrix(y_test,y_predict)
```

Out[30]:

```
array([[120,  24],
       [ 44,  43]], dtype=int64)
```

In [ ]:

```

```