```
import pandas as pd

df=pd.read_csv('C:/Users/Admin/Desktop/lp3/diabetes.csv')

df.columns
```

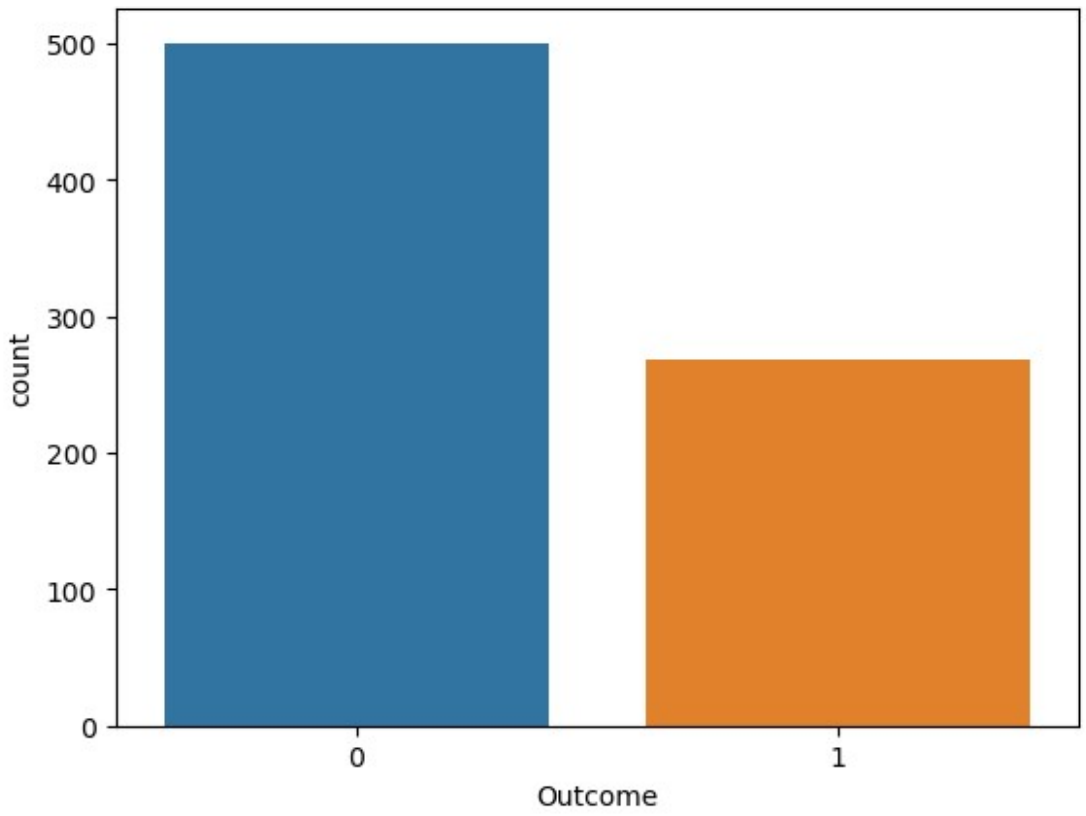
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
'Insulin',
       'BMI', 'Pedigree', 'Age', 'Outcome'],
      dtype='object')
```

```
import seaborn as sns

#input data
x=df.drop(['Outcome'],axis=1)
#output data
y=df['Outcome']

sns.countplot(x=y)
```

```
<Axes: xlabel='Outcome', ylabel='count'>
```




```
y.value_counts()
```

```
0    500
1    268
Name: Outcome, dtype: int64
```

```
#scaling
from sklearn.preprocessing import MinMaxScaler

scaler=MinMaxScaler()
x_scaled=scaler.fit_transform(x)

x_scaled
```

```
array([[0.35294118, 0.74371859, 0.59016393, ..., 0.50074516,
0.23441503,
        0.48333333],
       [0.05882353, 0.42713568, 0.54098361, ..., 0.39642325,
0.11656704,
        0.16666667],
       [0.47058824, 0.91959799, 0.52459016, ..., 0.34724292,
0.25362938,
        0.18333333],
       ...,
       [0.29411765, 0.6080402 , 0.59016393, ..., 0.390462  ,
0.07130658,
        0.15      ],
       [0.05882353, 0.63316583, 0.49180328, ..., 0.4485842 ,
0.11571307,
        0.43333333],
       [0.05882353, 0.46733668, 0.57377049, ..., 0.45305514,
0.10119556,
        0.03333333]])
```

```
#cross validation
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.
25)
```

```
x.shape
```

```
(768, 8)
```

```
x_train.shape
```

```
(576, 8)
```

```
x_test.shape
```

```
(192, 8)
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn= KNeighborsClassifier(n_neighbors=5)

knn.fit(x_train,y_train)
```

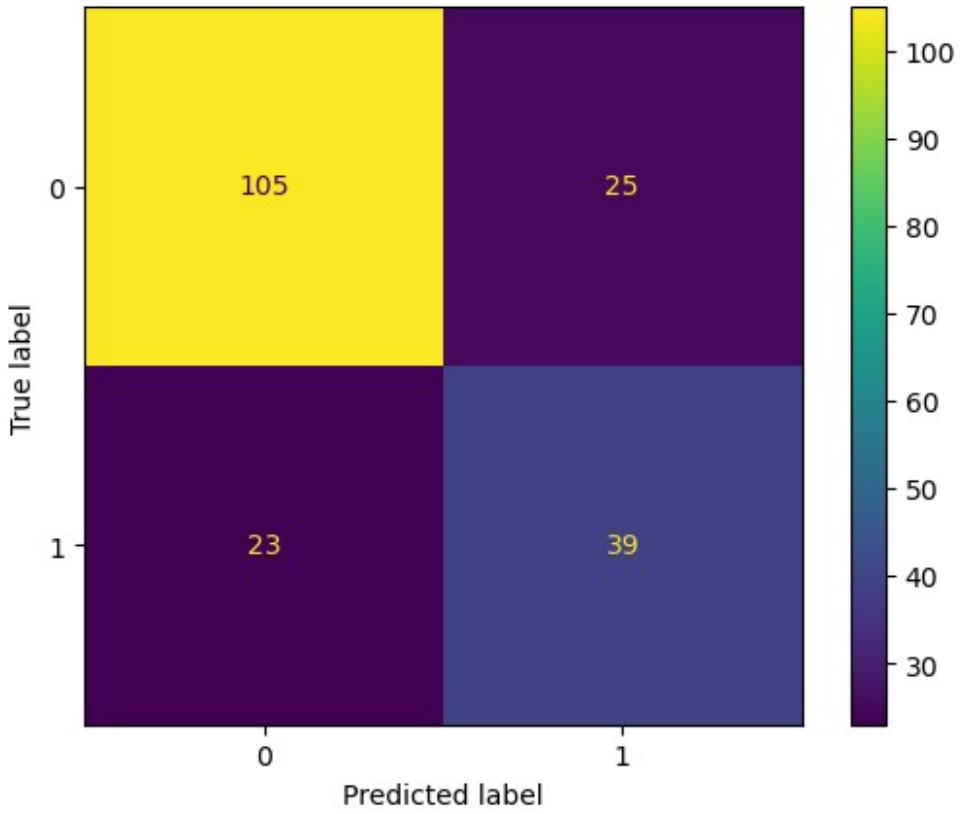
```
KNeighborsClassifier()
```

```
from sklearn.metrics import accuracy_score,ConfusionMatrixDisplay
from sklearn.metrics import classification_report

y_pred=knn.predict(x_test)

ConfusionMatrixDisplay.from_predictions(y_test,y_pred)
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x27aeb73add0>
```




```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.82      0.81      0.81       130
           1       0.61      0.63      0.62        62

    accuracy                           0.75       192
   macro avg       0.71      0.72      0.72       192
```

```
weighted avg       0.75      0.75      0.75       192
```

```
import matplotlib.pyplot as plt
import numpy as np

error=[]
for i in range(1,20):
    knn= KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train,y_train)
    y_pred=knn.predict(x_test)
    error.append(np.mean(y_pred!=y_test))

error
```

```
[0.296875,
 0.2708333333333333,
 0.23958333333333334,
 0.24479166666666666,
 0.25,
 0.24479166666666666,
 0.21875,
 0.21875,
 0.2708333333333333,
 0.25,
 0.2604166666666667,
 0.22395833333333334,
 0.2604166666666667,
 0.23958333333333334,
 0.25,
 0.23958333333333334,
 0.22916666666666666,
 0.22916666666666666,
 0.22395833333333334]
```

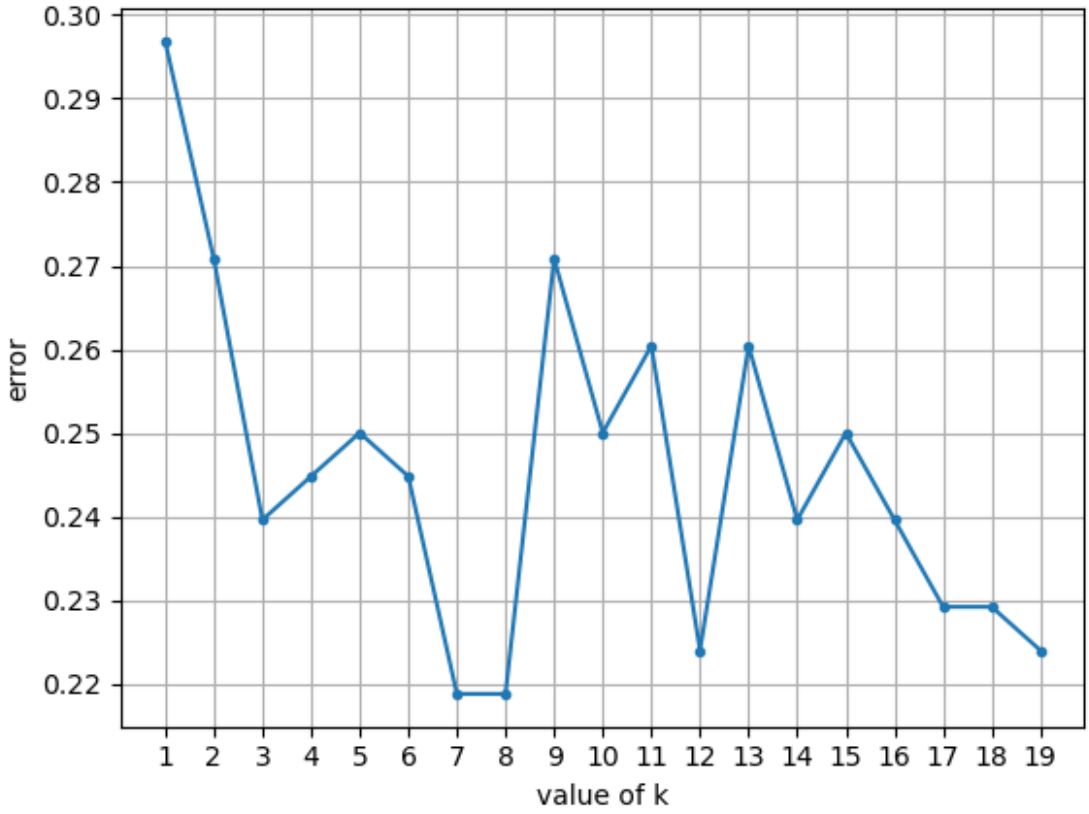
```
plt.figure(figsize=(16,9))
```

```
<Figure size 1600x900 with 0 Axes>
```

```
<Figure size 1600x900 with 0 Axes>
```

```
plt.xlabel('value of k')
plt.ylabel('error')
plt.grid()
plt.xticks(range(1,20))
plt.plot(range(1,20),error,marker='.')
```

```
[<matplotlib.lines.Line2D at 0x27af324cdd0>]
```


0.30
0.29
0.28
0.27
0.26
0.25
0.24
0.23
0.22
error
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
value of k