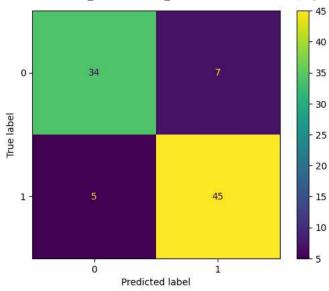
```
import pandas as pd
df=pd.read_csv('/content/heart.csv')
df
```

| | age | sex | ср | trtbps | chol | fbs | restecg | thalachh | exng | oldpeak | slp | caa | thall | output |
|-----|-----|-----|----|--------|------|-----|---------|----------|------|---------|-----|-----|-------|--------|
| 0 | 63 | 1 | 3 | 145 | 233 | 1 | 0 | 150 | 0 | 2.3 | 0 | 0 | 1 | 1 |
| 1 | 37 | 1 | 2 | 130 | 250 | 0 | 1 | 187 | 0 | 3.5 | 0 | 0 | 2 | 1 |
| 2 | 41 | 0 | 1 | 130 | 204 | 0 | 0 | 172 | 0 | 1.4 | 2 | 0 | 2 | 1 |
| 3 | 56 | 1 | 1 | 120 | 236 | 0 | 1 | 178 | 0 | 8.0 | 2 | 0 | 2 | 1 |
| 4 | 57 | 0 | 0 | 120 | 354 | 0 | 1 | 163 | 1 | 0.6 | 2 | 0 | 2 | 1 |
| | | | | | | | | | | | | | | |
| 298 | 57 | 0 | 0 | 140 | 241 | 0 | 1 | 123 | 1 | 0.2 | 1 | 0 | 3 | 0 |
| 299 | 45 | 1 | 3 | 110 | 264 | 0 | 1 | 132 | 0 | 1.2 | 1 | 0 | 3 | 0 |
| 300 | 68 | 1 | 0 | 144 | 193 | 1 | 1 | 141 | 0 | 3.4 | 1 | 2 | 3 | 0 |
| 301 | 57 | 1 | 0 | 130 | 131 | 0 | 1 | 115 | 1 | 1.2 | 1 | 1 | 3 | 0 |
| 302 | 57 | 0 | 1 | 130 | 236 | 0 | 0 | 174 | 0 | 0.0 | 1 | 1 | 2 | 0 |

```
303 rows × 14 columns
df.isna().sum()
(2)
        0
  age
  sex
        0
  ср
        а
  trtbps
        0
  chol
        0
  fbs
        0
  restecg
        0
  thalachh
        0
        0
  exng
  oldpeak
        0
  slp
        0
  caa
        0
  thall
        0
  output
        0
  dtype: int64
x=df.iloc[:,:-1].values
  array([[63., 1., 3., ..., 0., 0., 1.],
      [37., 1., 2., ..., 0., 0., 2.],
      [41., 0., 1., ..., 2., 0., 2.],
     [68., 1., 0., ..., 1., 2., 3.],
     [57., 1., 0., ..., 1., 1., 3.],
[57., 0., 1., ..., 1., 1., 2.]])
y=df.iloc[:,-1].values
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     from sklearn.model_selection import train_test_split
x\_train, x\_test, y\_train, y\_test=train\_test\_split(x, y, test\_size=0.30, random\_state=42)
x_train
  array([[39., 0., 2., ..., 2., 0., 2.], [29., 1., 1., ..., 2., 0., 2.],
     [50., 0., 2., ..., 1., 0., 2.],
```

```
[69., 1., 3., ..., 1., 1., 2.],
            [46., 1., 0., ..., 2., 0., 3.],
[63., 0., 1., ..., 2., 2., 2.]])
y_train
     array([1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,
            1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1,
            1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1,
            0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1,
            0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,
            0,\ 0,\ 1,\ 0,\ 1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1,\ 1,\ 1,\ 1,\ 0,\ 1,\ 1,\ 0,\ 1,\ 1,\ 0,
            1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0,
            0,\ 0,\ 1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 0,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 0,\ 1,\ 1,\ 1,\ 0,\ 0,\ 1,
            1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0,
            1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1])
# preprocessing
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
scaler.fit(x_train)
x_train=scaler.fit_transform(x_train)
x_test=scaler.fit_transform(x_test)
x test
     {\sf array}([[\ 0.29414409,\ 0.5985056\ ,\ -0.84722571,\ \dots,\ -0.58445999,
               0.0575118 , -1.97484177],
             [\ 0.52098403,\ 0.5985056\ ,\ 2.11806429,\ \ldots,\ -0.58445999,
            -0.81475054, 1.09713431],
[ 0.29414409, 0.5985056 , 1.12963429, ..., 1.0272327 ,
              0.0575118 , 1.09713431],
            [ 1.42834376, -1.67082814, -0.84722571, ..., 1.0272327 ,
              0.92977415, -0.43885373],
             [\ 0.40756406,\ 0.5985056\ ,\ 1.12963429,\ \dots,\ -0.58445999,
              -0.81475054, 1.09713431],
             [ 2.44912347, -1.67082814, 1.12963429, ..., -0.58445999,
              -0.81475054, -0.43885373]])
from sklearn.neighbors import KNeighborsClassifier
model=KNeighborsClassifier(n_neighbors=5)
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred
     array([0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0,
            0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1,
            1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0,
            1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1,
from sklearn.metrics import confusion_matrix,accuracy_score,ConfusionMatrixDisplay
result=confusion_matrix(y_test,y_pred)
result
     array([[34, 7],
            [ 5, 45]])
cmd=ConfusionMatrixDisplay(result,display_labels=cm)
cmd.plot()
```

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7d7011165de0>



score=accuracy_score(y_test,y_pred)
score

0.8681318681318682