```
import numpy as np
import pandas as pd
df=pd.read_csv('/content/heart_missing.csv')
df
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
0	52	1	0.0	125	212	0.0	1.0	168	0	1.0	2	2
1	53	1	0.0	140	203	1.0	0.0	155	1	3.1	0	0
2	70	1	0.0	145	174	0.0	1.0	125	1	2.6	0	0
3	61	1	0.0	148	203	0.0	1.0	161	0	0.0	2	1
4	62	0	0.0	138	294	1.0	1.0	106	0	1.9	1	3
...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1.0	140	221	0.0	1.0	164	1	0.0	2	0
1021	60	1	0.0	125	258	0.0	0.0	141	1	2.8	1	1
1022	47	1	0.0	110	275	0.0	0.0	118	1	1.0	1	1
1023	50	0	0.0	110	254	0.0	0.0	159	0	0.0	2	0
1024	54	1	0.0	120	188	0.0	1.0	113	0	1.4	1	1

1025 rows × 14 columns

```
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	52	1	0.0	125	212	0.0	1.0	168	0	1.0	2	2	
1	53	1	0.0	140	203	1.0	0.0	155	1	3.1	0	0	
2	70	1	0.0	145	174	0.0	1.0	125	1	2.6	0	0	
3	61	1	0.0	148	203	0.0	1.0	161	0	0.0	2	1	
4	62	0	0.0	138	294	1.0	1.0	106	0	1.9	1	3	

```
df.tail()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
1020	59	1	1.0	140	221	0.0	1.0	164	1	0.0	2	0
1021	60	1	0.0	125	258	0.0	0.0	141	1	2.8	1	1
1022	47	1	0.0	110	275	0.0	0.0	118	1	1.0	1	1
1023	50	0	0.0	110	254	0.0	0.0	159	0	0.0	2	0
1024	54	1	0.0	120	188	0.0	1.0	113	0	1.4	1	1

```
df.isna().sum()
```

```
age          0
sex          0
cp           7
trestbps     0
chol         0
fbs         13
restecg      4
```

```
     thalach      0
     exang        0
     oldpeak     11
     slope        0
     ca           0
     thal         0
     target       0
     dtype: int64
```

```
x=df['cp'].mean()
x
```

```
     0.9390962671905697
```

```
df['cp'].fillna(x,inplace=True)
df.isna().sum()
```

```
     age          0
     sex          0
     cp           0
     trestbps     0
     chol         0
     fbs         13
     restecg      4
     thalach      0
     exang        0
     oldpeak     11
     slope        0
     ca           0
     thal         0
     target       0
     dtype: int64
```

```
y=df['fbs'].mean()
y
```

```
     0.15118577075098813
```

```
df['fbs'].fillna(y,inplace=True)
df.isna().sum()
```

```
a=df['restecg'].mean()
a
```

```
     0.5318315377081293
```

```
df['restecg'].fillna(a,inplace=True)
df.isna().sum()
```

```
b=df['oldpeak'].mean()
b
```

```
     1.0784023668639053
```

```
df['oldpeak'].fillna(b,inplace=True)
df.isna().sum()
```

```
     age         0
     sex         0
     cp          0
     trestbps    0
     chol        0
     fbs         0
     restecg     0
     thalach     0
     exang       0
     oldpeak     0
     slope       0
     ca          0
     thal        0
     target      0
     dtype: int64
```

```
df.dtypes
```

```
     age           int64
     sex           int64
     cp          float64
     trestbps      int64
     chol          int64
     fbs         float64
     restecg     float64
     thalach       int64
     exang         int64
     oldpeak     float64
     slope         int64
     ca            int64
     thal          int64
     target        int64
     dtype: object
```

```
x=df.iloc[:,:-1].values
x
```

```
     array([[52.,  1.,  0., ...,  2.,  2.,  3.],
            [53.,  1.,  0., ...,  0.,  0.,  3.],
            [70.,  1.,  0., ...,  0.,  0.,  3.],
            ...,
            [47.,  1.,  0., ...,  1.,  1.,  2.],
            [50.,  0.,  0., ...,  2.,  0.,  2.],
            [54.,  1.,  0., ...,  1.,  1.,  3.]])
```

```
y=df.iloc[:,-1].values
y
```

```
     array([0, 0, 0, ..., 0, 1, 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=0)
x_train
```

```
     array([[57.,  1.,  2., ...,  2.,  0.,  2.],
            [59.,  1.,  0., ...,  1.,  2.,  1.],
            [71.,  0.,  0., ...,  1.,  0.,  2.],
            ...,
            [65.,  1.,  3., ...,  1.,  1.,  2.],
            [67.,  1.,  0., ...,  1.,  0.,  2.],
            [60.,  1.,  2., ...,  1.,  0.,  2.]])
```

```
x_test
```

```
     array([[44.,  1.,  2., ...,  2.,  0.,  2.],
            [58.,  0.,  1., ...,  2.,  2.,  2.],
            [63.,  1.,  0., ...,  2.,  2.,  3.],
            ...,
            [56.,  1.,  2., ...,  1.,  1.,  1.],
            [57.,  0.,  0., ...,  2.,  0.,  2.],
            [54.,  1.,  0., ...,  1.,  1.,  3.]])
```

```
y_train
```

```
from sklearn.preprocessing import StandardScaler
norm=StandardScaler()
norm.fit(x_train)
x_train=norm.transform(x_train)
x_test=norm.transform(x_test)
x_train
```

```
x_test
```

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
x_test
```

```
from sklearn.metrics import confusion_matrix,accuracy_score
con_mat=confusion_matrix(y_test,y_pred)
print(con_mat)
```

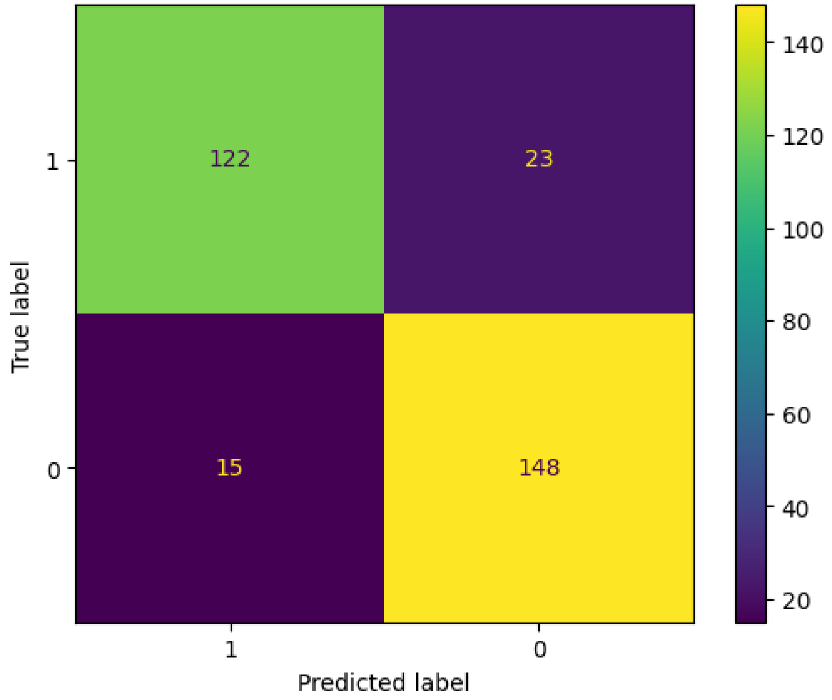
```
     [[122  23]
      [ 15 148]]
```

```
score=accuracy_score(y_test,y_pred)
score
```

```
     0.8766233766233766
```

```
from sklearn.metrics import ConfusionMatrixDisplay
labels=[1, 0]
cmd=ConfusionMatrixDisplay(con_mat,display_labels=labels)
cmd.plot()
```

```
     <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7831170ee110>
```


True label
Predicted label