```
import os
os.getcwd()
```

```
'/content'
```

```
import pandas as pd
```

```
#importing dataset
df=pd.read_csv('/content/archive heart.zip')
```

```
df.head
```

```
<bound method NDFrame.head of      Unnamed: 0  Age  Sex     ChestPain  RestBP  Chol  Fbs  RestECG  MaxHR  \
0             1   63    1       typical     145   233    1        2    150
1             2   67    1  asymptomatic     160   286    0        2    108
2             3   67    1  asymptomatic     120   229    0        2    129
3             4   37    1    nonanginal     130   250    0        0    187
4             5   41    0    nontypical     130   204    0        2    172
..          ...  ...  ...           ...     ...   ...  ...      ...    ...
298         299   45    1       typical     110   264    0        0    132
299         300   68    1  asymptomatic     144   193    1        0    141
300         301   57    1  asymptomatic     130   131    0        0    115
301         302   57    0    nontypical     130   236    0        2    174
302         303   38    1    nonanginal     138   175    0        0    173

     ExAng  Oldpeak  Slope   Ca        Thal  AHD
0        0      2.3      3  0.0       fixed   No
1        1      1.5      2  3.0      normal  Yes
2        1      2.6      2  2.0  reversable  Yes
3        0      3.5      3  0.0      normal   No
4        0      1.4      1  0.0      normal   No
..     ...      ...    ...  ...         ...  ...
298      0      1.2      2  0.0  reversable  Yes
299      0      3.4      2  2.0  reversable  Yes
300      1      1.2      2  1.0  reversable  Yes
301      0      0.0      2  1.0      normal  Yes
302      0      0.0      1  NaN      normal   No

[303 rows x 15 columns]>
```

```
#find shape of data
df.shape
```

```
(303, 15)
```

```
#find missing values
df.isnull()
```

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
299	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
300	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
301	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
302	False	False	False	False	False	False	False	False	False	False	False	False	True	False	False

303 rows × 15 columns

```
#find missing values
df.isnull().sum()
```

```
Unnamed: 0    0
Age           0
Sex           0
ChestPain     0
RestBP        0
Chol          0
Fbs           0
RestECG       0
MaxHR         0
ExAng         0
Oldpeak       0
Slope         0
Ca            4
Thal          2
AHD           0
dtype: int64
```

```
#find missing values
df.count()
```

```
Unnamed: 0    303
Age           303
Sex           303
ChestPain     303
RestBP        303
Chol          303
Fbs           303
RestECG       303
MaxHR         303
ExAng         303
Oldpeak       303
Slope         303
Ca            299
Thal          301
AHD           303
dtype: int64
```

```
#find datatypes of each column
df.dtypes
```

```
Unnamed: 0      int64
Age             int64
Sex             int64
ChestPain      object
RestBP          int64
Chol            int64
Fbs             int64
RestECG         int64
MaxHR           int64
ExAng           int64
Oldpeak       float64
Slope           int64
Ca            float64
Thal           object
AHD            object
dtype: object
```

```
#find out zeros
df==0
```

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
0	False	False	False	False	False	False	False	False	False	True	False	False	True	False	False

```
#For highlighting
df[df==0]
```

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	0.0	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN	0.0	NaN	NaN	0.0	NaN	NaN
4	NaN	NaN	0.0	NaN	NaN	NaN	0.0	NaN	NaN	0.0	NaN	NaN	0.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN	0.0	NaN	NaN	0.0	NaN	NaN
299	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN
300	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
301	NaN	NaN	0.0	NaN	NaN	NaN	0.0	NaN	NaN	0.0	0.0	NaN	NaN	NaN	NaN
302	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN	0.0	0.0	NaN	NaN	NaN	NaN

303 rows × 15 columns

```
#for counting
df[df==0].count()
```

```
Unnamed: 0      0
Age             0
Sex            97
ChestPain       0
RestBP          0
Chol            0
Fbs           258
RestECG       151
MaxHR           0
ExAng         204
Oldpeak        99
Slope           0
Ca            176
Thal            0
AHD             0
dtype: int64
```

```
#displaying columns
df.columns
```

```
Index(['Unnamed: 0', 'Age', 'Sex', 'ChestPain', 'RestBP', 'Chol', 'Fbs',
       'RestECG', 'MaxHR', 'ExAng', 'Oldpeak', 'Slope', 'Ca', 'Thal', 'AHD'],
      dtype='object')
```

```
#find mean age of patients
df['Age']
df['Age'].mean()
```

```
54.43894389438944
```

```
#Now extract only Age, Sex, ChestPain, RestBP, Chol. Randomly divide dataset in training
#(75%) and testing (25%)
df[['Age','Sex','ChestPain','RestBP','Chol']]
```

	Age	Sex	ChestPain	RestBP	Chol
0	63	1	typical	145	233
1	67	1	asymptomatic	160	286
2	67	1	asymptomatic	120	229
3	37	1	nonanginal	130	250
4	41	0	nontypical	130	204
...	...	...	...	...	...
298	45	1	typical	110	264

```
newdf=df[['Age','Sex','ChestPain','RestBP','Chol']]
newdf
```

	Age	Sex	ChestPain	RestBP	Chol
0	63	1	typical	145	233
1	67	1	asymptomatic	160	286
2	67	1	asymptomatic	120	229
3	37	1	nonanginal	130	250
4	41	0	nontypical	130	204
...	...	...	...	...	...
298	45	1	typical	110	264
299	68	1	asymptomatic	144	193
300	57	1	asymptomatic	130	131
301	57	0	nontypical	130	236
302	38	1	nonanginal	138	175

303 rows × 5 columns

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

train,test=train_test_split(df,random_state=0,test_size=0.25)

train.shape
```

```
(227, 15)
```

```
test.shape
```

```
(76, 15)
```

```
import numpy as np

actual=list(np.ones(45))+list(np.zeros(55))

np.array(actual)
```

```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
predicted=list(np.ones(40))+list(np.zeros(52))+list(np.ones(8))

np.array(predicted)
```

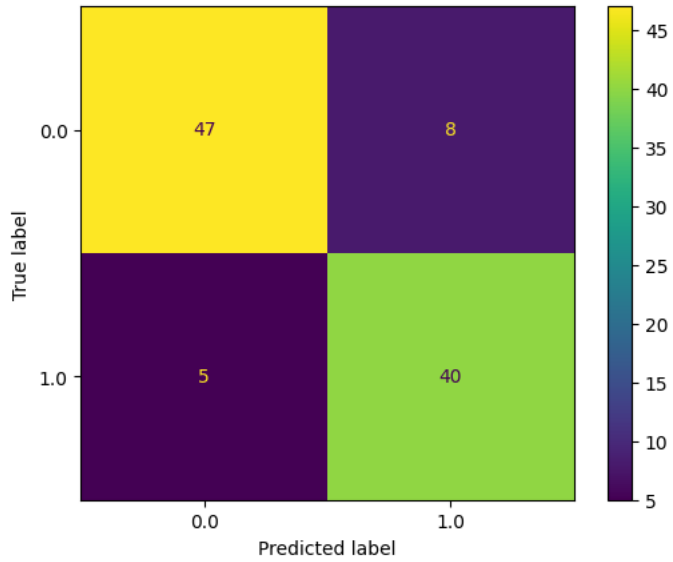
```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
```

```
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 1., 1., 1., 1.])
```

```
from sklearn.metrics import ConfusionMatrixDisplay
```

```
ConfusionMatrixDisplay.from_predictions(actual,predicted)
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x78630bc33eb0>
```




```
from sklearn.metrics import classification_report
```

```
print(classification_report(actual,predicted))
```

```
              precision    recall  f1-score   support

         0.0       0.90      0.85      0.88        55
         1.0       0.83      0.89      0.86        45

    accuracy                           0.87       100
   macro avg       0.87      0.87      0.87       100
weighted avg       0.87      0.87      0.87       100
```

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(actual,predicted)
```

```
0.87
```