In [6]:
```
import pandas as pd
import numpy as num
```

In [ ]:

In [7]:
```
dfnpd.read_csv("D:\heart.csv")
df.head()
```

Out[7]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.0	fixed	No
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0	normal	Yes
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0	reversable	Yes
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0	normal	No
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0	normal	No

In [ ]:

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    303 non-null    int64
 1   Age           303 non-null    int64
 2   Sex           303 non-null    int64
 3   ChestPain     303 non-null    object
 4   RestBP        303 non-null    int64
 5   Chol          303 non-null    int64
 6   Fbs           303 non-null    int64
 7   RestECG       303 non-null    int64
 8   MaxHR         303 non-null    int64
 9   ExAng         303 non-null    int64
 10  Oldpeak       303 non-null    float64
 11  Slope         303 non-null    int64
 12  Ca            299 non-null    float64
 13  Thal          301 non-null    object
 14  AHD           303 non-null    object
dtypes: float64(2), int64(10), object(3)
memory usage: 35.6+ KB
```

In [ ]:

In [9]:
```
size=df.size
print("size of the dataframe is:",size)
```

```
size of the dataframe is: 4545
```

In [ ]:

In [10]:
```
df.shape
```

Out[10]:
```
(303, 15)
```

In [ ]:

In [11]:
```
print(df.describe())
```

```
       Unnamed: 0         Age         Sex      RestBP        Chol         Fbs  \
count  303.000000  303.000000  303.000000  303.000000  303.000000  303.000000
mean   152.000000   54.438944    0.679868  131.689769  246.693069    0.148515
std    87.412784    9.038662    0.467299   17.599748   51.776918    0.356198
min     1.000000   29.000000    0.000000   94.000000  126.000000    0.000000
25%   76.500000   48.000000    0.000000  120.000000  211.000000    0.000000
50%   152.000000   56.000000    1.000000  130.000000  241.000000    0.000000
75%   227.500000   61.000000    1.000000  140.000000  275.000000    0.000000
max   303.000000   77.000000    1.000000  200.000000  564.000000    1.000000

          RestECG       MaxHR       ExAng     Oldpeak       Slope          Ca
count  303.000000  303.000000  303.000000  303.000000  303.000000  299.000000
mean     0.990099  149.607261    0.326733    1.039604    1.600660    0.672241
std      0.994971   22.875003    0.469794    1.161075    0.616226    0.937438
min      0.000000   71.000000    0.000000    0.000000    1.000000    0.000000
25%      0.000000  133.500000    0.000000    0.000000    1.000000    0.000000
50%      1.000000  153.000000    0.000000    0.800000    2.000000    0.000000
75%      2.000000  166.000000    1.000000    1.600000    2.000000    1.000000
max      2.000000  202.000000    1.000000    6.200000    3.000000    3.000000
```

In [ ]:

In [12]:
```
print(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]>
```

In [ ]:

In [13]:
```
print("Data Type for each columns are\n",df.dtypes.value_counts())
```

```
Data Type for each columns are
 int64    10
object    3
float64   2
Name: count, dtype: int64
```

In [ ]:

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
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
```

In [ ]:

In [15]:
```
df['Age'].mean()
```

Out[15]:
```
np.float64(54.438943894389441)
```

In [ ]:

In [16]:
```
df.columns
```

Out[16]:
```
Index(['Unnamed: 0', 'Age', 'Sex', 'ChestPain', 'RestBP', 'Chol', 'Fbs',
       'RestECG', 'MaxHR', 'ExAng', 'Oldpeak', 'Slope', 'Ca', 'Thal', 'AHD'],
      dtype='object')
```

In [ ]:

In [17]:
```
print(df['Age'])
print(df['Sex'])
print(df['RestBP'])
print(df['Chol'])
```

```
0      63
1      67
2      67
3      37
4      41
       ..
298    45
299    68
300    57
301    57
302    38
Name: Age, Length: 303, dtype: int64
0      1
1      1
2      1
3      1
4      0
      ..
298    1
299    1
300    1
301    0
302    1
Name: Sex, Length: 303, dtype: int64
0      145
1      160
2      120
3      130
4      130
      ...
298    110
299    144
300    130
301    130
302    138
Name: RestBP, Length: 303, dtype: int64
0      233
1      286
2      229
3      250
4      204
      ...
298    264
299    193
300    131
301    236
302    175
Name: Chol, Length: 303, dtype: int64
```

In [ ]:

In [18]:
```
df[['Age','Sex','ChestPain','RestBP','Chol','RestECG']]
```

Out[18]:

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

303 rows × 6 columns

In [ ]:

In [19]:
```
from sklearn.model_selection import train_test_split
```

In [ ]:

In [20]:
```
train, test=train_test_split(df,random_state=0,test_size=0.25)
```

In [ ]:

In [21]:
```
train.shape
```

Out[21]:
```
(227, 15)
```

In [ ]:

In [22]:
```
test.shape
```

Out[22]:
```
(76, 15)
```

In [ ]:

In [23]:
```
import numpy as np
```

In [ ]:

In [24]:
```
actual=list(np.ones(45))+list(np.zeros(55))
```

In [ ]:

In [25]:
```
np.array(actual)
```

Out[25]:
```
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., 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., 0.])
```

In [ ]:

In [26]:
```
predicted=list(np.ones(35))+list(np.zeros(62))+list(np.ones(3))
```

In [ ]:

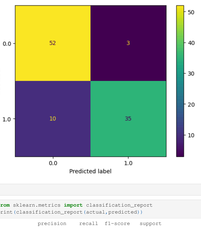
In [27]:
```
np.array(predicted)
```

Out[27]:
```
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., 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.])
```

In [ ]:

In [28]:
```
from sklearn.metrics import ConfusionMatrixDisplay
ConfusionMatrixDisplay.from_predictions(actual,predicted)
```

Out[28]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2887ff72500>
```



In [ ]:

In [29]:
```
from sklearn.metrics import classification_report
print(classification_report(actual,predicted))
```

```
              precision    recall  f1-score   support

         0.0       0.84      0.95      0.89        55
         1.0       0.92      0.78      0.84        45

    accuracy                           0.87       100
   macro avg       0.88      0.86      0.87       100
weighted avg       0.88      0.87      0.87       100
```

In [ ]:

In [30]:
```
from sklearn.metrics import accuracy_score
accuracy_score(actual,predicted)
```

Out[361]: 0.87

In [ ]: