In [5]:
```
import pandas as pd                                                    ##Importing libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [12]:
```
ds=pd.read_csv(r"C:\Users\SWATHI\Downloads\archive (2).zip")        ##getting dataset for prediction
ds
```

Out[12]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
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	0.8	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

In [13]:
```
df=pd.read_csv("heart.csv")                          #we can also get the dataset like this
df
```

Out[13]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
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	0.8	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

In [14]:
```
df.head()
```

Out[14]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
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	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

In [33]:
```
df.info()                                        ##for getting the info about dataset like datatypes, null value
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        303 non-null    int64  
 12  thal      303 non-null    int64  
 13  target    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

In [15]:
```
df.describe()
```

Out[15]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000

In [16]:
```
df.sum()                                    ##for getting the sum of the columns
```

Out[16]:
```
age         16473.0
sex           207.0
cp            293.0
trestbps    39882.0
chol        74618.0
fbs            45.0
restecg       160.0
thalach     45343.0
exang          99.0
oldpeak       315.0
slope         424.0
ca            221.0
thal          701.0
target        165.0
dtype: float64
```

In [17]:
```
df.isnull()                                 ##for checking the null values
```

Out[17]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	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
2	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
4	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
299	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
301	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	False	False

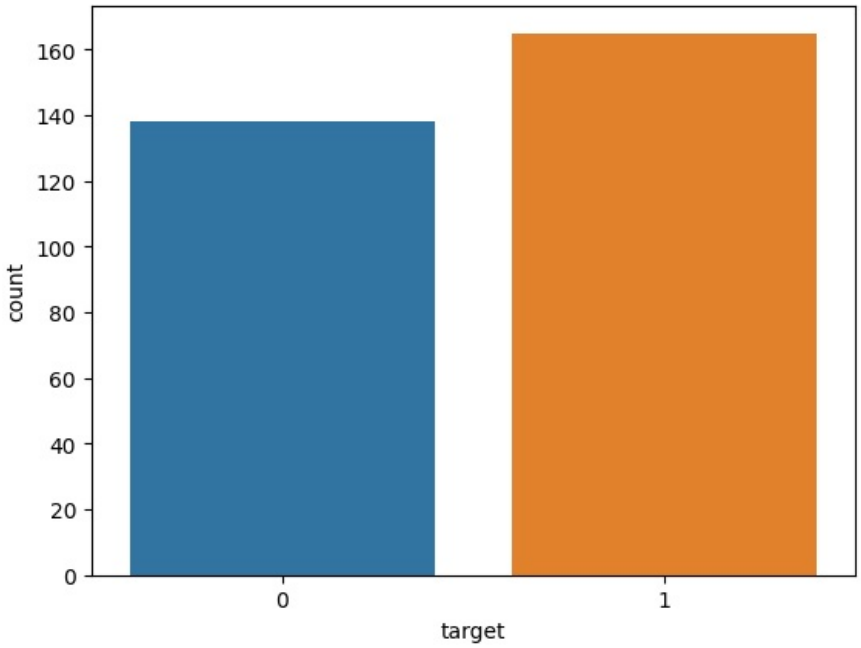
303 rows × 14 columns

In [18]:
```
df.isna().sum()                                    ##for getting the count of null values
```

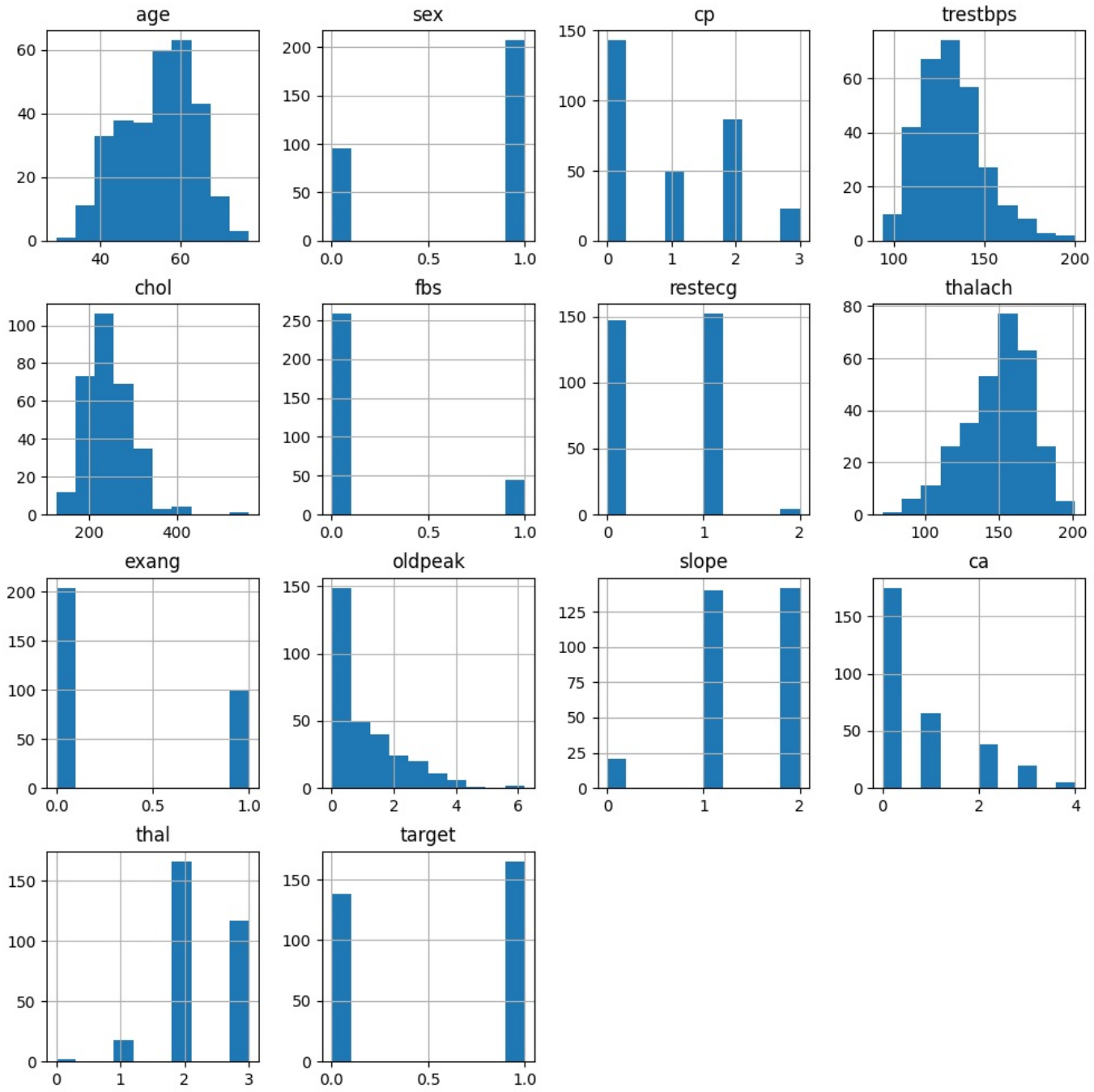
Out[18]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [19]:
```
sns.countplot(x='target', data=df)  ##for getting the plotting of the dataset here 0 is not having heart attack
```

Out[19]:
```
<Axes: xlabel='target', ylabel='count'>
```



In [20]:
```
df.hist(figsize=(12,12))   #getting the histograms of dataset and also checking the given data is categorical o
plt.show()
```



In [21]:
```
df2= pd.get_dummies(df, columns=['sex','cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal']) #creating dummy
```

In [25]:
```
df2
```

Out[25]:

	age	trestbps	chol	thalach	oldpeak	target	sex_0	sex_1	cp_0	cp_1	...	slope_2	ca_0	ca_1	ca_2	ca_3	ca_4	thal_
0	63	145	233	150	2.3	1	False	True	False	False	...	False	True	False	False	False	False	Fals
1	37	130	250	187	3.5	1	False	True	False	False	...	False	True	False	False	False	False	Fals
2	41	130	204	172	1.4	1	True	False	False	True	...	True	True	False	False	False	False	Fals
3	56	120	236	178	0.8	1	False	True	False	True	...	True	True	False	False	False	False	Fals
4	57	120	354	163	0.6	1	True	False	True	False	...	True	True	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
298	57	140	241	123	0.2	0	True	False	True	False	...	False	True	False	False	False	False	Fals
299	45	110	264	132	1.2	0	False	True	False	False	...	False	True	False	False	False	False	Fals
300	68	144	193	141	3.4	0	False	True	True	False	...	False	False	False	True	False	False	Fals
301	57	130	131	115	1.2	0	False	True	True	False	...	False	False	True	False	False	False	Fals
302	57	130	236	174	0.0	0	True	False	False	True	...	False	False	True	False	False	False	Fals

303 rows × 31 columns

In [22]:
```
cols=['cp_0', "cp_1", 'cp_2', 'trestbps','chol', 'fbs_0', 'fbs_1', 'restecg_0','restecg_1','restecg_2']
```

In [23]:
```
x=df2[cols]
y=df2.target
y
```

Out[23]:
```
0      1
1      1
2      1
3      1
4      1
      ..
298    0
299    0
300    0
301    0
302    0
Name: target, Length: 303, dtype: int64
```

In [45]:
```
from sklearn.model_selection import train_test_split         #Splitting data into training and testing sets
x_train,x_test,y_train,y_test=train_test_split(x , y ,test_size=0.25,random_state=42)
```

In [46]:
```
x_train.shape
```

Out[46]:
```
(227, 10)
```

In [47]:
```
x_test.shape
```

Out[47]:
```
(76, 10)
```

In [48]:
```
from sklearn.preprocessing import StandardScaler                    ##feature Scaling
sc=StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

In [49]:
```
x_train
```

Out[49]:
```
array([[-0.91139737,  2.34216018, -0.6815542 , ...,  1.07791686,
        -1.04046567, -0.13392991],
       [-0.91139737, -0.42695628,  1.46723474, ..., -0.92771533,
         0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 , ..., -0.92771533,
         0.96110812, -0.13392991],
       ...,
       [-0.91139737, -0.42695628, -0.6815542 , ...,  1.07791686,
        -1.04046567, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 , ...,  1.07791686,
        -1.04046567, -0.13392991],
       [-0.91139737,  2.34216018, -0.6815542 , ..., -0.92771533,
         0.96110812, -0.13392991]])
```

In [50]:
```
x_test
```

Out[50]:
```
array([[ 1.09721625, -0.42695628, -0.6815542 ,  1.16826465,  0.55847168,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737, -0.42695628, -0.6815542 ,  2.34508328,  0.78298125,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737, -0.42695628,  1.46723474,  1.16826465, -2.24789797,
        -2.61296502,  2.61296502, -0.92771533,  0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  0.22680976,  3.04678611,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737, -0.42695628,  1.46723474, -1.18537259,  0.35267124,
        -2.61296502,  2.61296502,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737, -0.42695628,  1.46723474,  1.16826465, -1.46211447,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  0.46217348, -0.05892964,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  4.11031121,  0.78298125,
        -2.61296502,  2.61296502,  1.07791686, -1.04046567, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 , -0.47928142,  0.74556299,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737, -0.42695628,  1.46723474, -0.30275862,  0.50234429,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  0.57985534, -1.01309532,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [-0.91139737, -0.42695628, -0.6815542 ,  2.81581073,  0.4462169 ,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737,  2.34216018, -0.6815542 , -0.59696328,  1.4752191 ,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 , -1.18537259, -1.4808236 ,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737,  2.34216018, -0.6815542 , -1.47957725, -0.78858575,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [-0.91139737, -0.42695628,  1.46723474,  0.57985534, -0.20860269,
```

```
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[-0.91139737,  2.34216018, -0.6815542 , -0.71464514, -0.67633097,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -0.59696328, -1.08793185,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  0.57985534,  1.21329127,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  0.46217348, -1.49953273,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[-0.91139737, -0.42695628, -0.6815542 ,  2.34508328, -0.35827574,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -0.00855397,  0.6894356 ,
 -2.61296502,  2.61296502,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -0.59696328, -0.17118443,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[-0.91139737,  2.34216018, -0.6815542 ,  1.75667397,  1.04490909,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[-0.91139737, -0.42695628,  1.46723474, -0.71464514, -0.07763877,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  1.28594652, -0.43311226,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[-0.91139737, -0.42695628, -0.6815542 ,  1.75667397,  0.50234429,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  0.57985534,  0.27783472,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  2.05087862,  0.80169039,
 -2.61296502,  2.61296502,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -1.18537259,  0.1468708 ,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -0.12623583,  0.31525298,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  0.57985534, -0.73245836,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -0.59696328,  0.3900895 ,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -1.53841818, -0.71374923,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[-0.91139737, -0.42695628, -0.6815542 ,  1.16826465, -0.37698487,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -0.00855397,  0.1468708 ,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[-0.91139737, -0.42695628,  1.46723474, -0.30275862, -0.02151138,
 -2.61296502,  2.61296502,  1.07791686, -1.04046567, -0.13392991],
[-0.91139737, -0.42695628,  1.46723474, -0.36159955,  0.16557993,
 -2.61296502,  2.61296502, -0.92771533,  0.96110812, -0.13392991],
[-0.91139737, -0.42695628, -0.6815542 ,  0.57985534, -0.88213141,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -0.89116793,  0.25912559,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[-0.91139737, -0.42695628,  1.46723474,  0.46217348, -0.43311226,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[-0.91139737,  2.34216018, -0.6815542 ,  0.57985534, -0.47053052,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[-0.91139737,  2.34216018, -0.6815542 , -0.12623583, -0.71374923,
 -2.61296502,  2.61296502,  1.07791686, -1.04046567, -0.13392991],
[-0.91139737, -0.42695628,  1.46723474, -0.00855397, -0.60149444,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[-0.91139737,  2.34216018, -0.6815542 , -0.59696328,  0.31525298,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  0.10912789,  0.01590688,
 -2.61296502,  2.61296502,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -0.12623583, -0.78858575,
 -2.61296502,  2.61296502, -0.92771533,  0.96110812, -0.13392991],
[-0.91139737,  2.34216018, -0.6815542 , -0.12623583, -0.76987662,
 -2.61296502,  2.61296502, -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  2.34508328, -0.395694  ,
 -2.61296502,  2.61296502,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  1.75667397,  0.80169039,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -0.00855397,  0.18428906,
 -2.61296502,  2.61296502,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  0.69753721,  1.17587301,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[-0.91139737,  2.34216018, -0.6815542 ,  0.28565069, -0.80729488,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[-0.91139737, -0.42695628,  1.46723474, -1.65610004,  1.34425518,
  0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 , -0.30275862,  1.00749083,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[ 1.09721625, -0.42695628, -0.6815542 ,  1.75667397, -1.53695099,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
[-0.91139737, -0.42695628,  1.46723474,  0.57985534,  0.1468708 ,
  0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
```

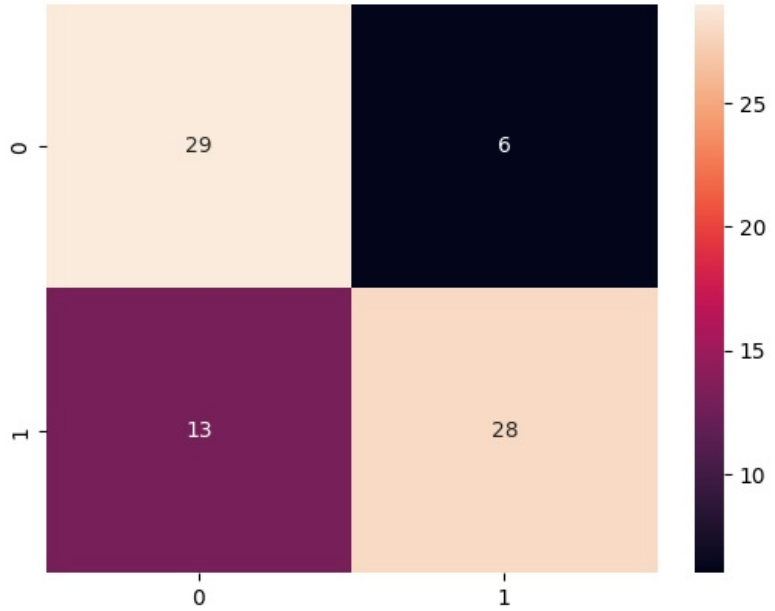
```
       [-0.91139737, -0.42695628,  1.46723474, -0.0673949 , -0.9382588 ,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  0.81521907, -0.99438619,
        -2.61296502,  2.61296502, -0.92771533,  0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  0.87406   ,  0.67072647,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  1.16826465, -0.05892964,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737,  2.34216018, -0.6815542 , -0.59696328,  0.91394517,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  1.16826465,  0.4462169 ,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737, -0.42695628,  1.46723474,  1.16826465, -0.28343921,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [-0.91139737,  2.34216018, -0.6815542 ,  0.28565069,  0.07203428,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 , -0.77348607, -0.30214835,
        -2.61296502,  2.61296502, -0.92771533,  0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 , -0.12623583,  1.06361822,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737,  2.34216018, -0.6815542 ,  0.10912789,  0.78298125,
        -2.61296502,  2.61296502,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737,  2.34216018, -0.6815542 ,  0.57985534,  0.89523604,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737, -0.42695628,  1.46723474,  0.57985534,  1.66231041,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [-0.91139737,  2.34216018, -0.6815542 , -1.47957725, -0.90084054,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  0.57985534, -0.37698487,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  1.16826465, -0.04022051,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 , -0.47928142, -0.45182139,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [ 1.09721625, -0.42695628, -0.6815542 ,  0.87406   , -0.6389127 ,
         0.382707  , -0.382707  ,  1.07791686, -1.04046567, -0.13392991],
       [-0.91139737,  2.34216018, -0.6815542 , -0.00855397,  0.29654385,
         0.382707  , -0.382707  , -0.92771533,  0.96110812, -0.13392991]])
```

In [41]:
```
from sklearn.neighbors import KNeighborsClassifier          ##Selecting the model
classifier= KNeighborsClassifier()
classifier.fit(x_train, y_train)
yp=classifier.predict(x_test)
yp
```

Out[41]:
```
array([0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1,
       1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 1], dtype=int64)
```

In [52]:
```
from sklearn.metrics import confusion_matrix             ##getting confusion matrix using seaborn plots
cm=confusion_matrix(y_test,yp)
sns.heatmap(cm, annot= True)
```

Out[52]:
```
<Axes: >
```



In [54]:
```
from sklearn.metrics import accuracy_score   ##Checking the accuracy score
accuracy_score(y_test,yp)
```

Out[54]: 0.75

In [ ]: