```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

dataset=pd.read_csv("information.csv")

dataset.head()
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX
PTRATIO  \
0  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296
15.3
1  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242
17.8
2  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242
17.8
3  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222
18.7
4  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222
18.7

        B  LSTAT  MEDV
0  396.90   4.98  24.0
1  396.90   9.14  21.6
2  392.83   4.03  34.7
3  394.63   2.94  33.4
4  396.90   5.33  36.2
```

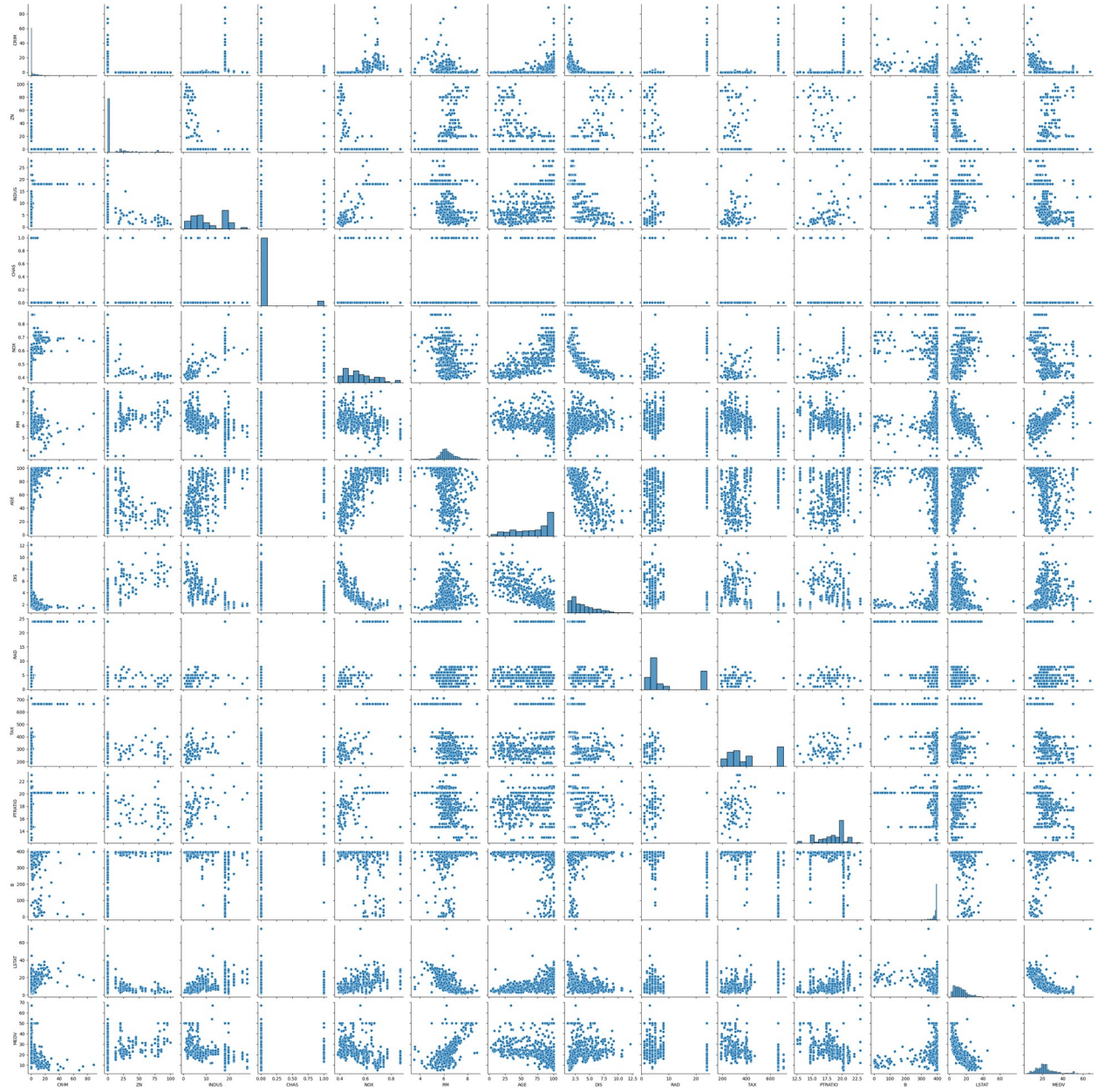
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 511 entries, 0 to 510
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   CRIM     511 non-null    float64
 1   ZN       511 non-null    float64
 2   INDUS    511 non-null    float64
 3   CHAS     511 non-null    int64
 4   NOX      511 non-null    float64
 5   RM       511 non-null    float64
 6   AGE      511 non-null    float64
 7   DIS      511 non-null    float64
 8   RAD      511 non-null    int64
 9   TAX      511 non-null    int64
 10  PTRATIO  511 non-null    float64
 11  B        511 non-null    float64
 12  LSTAT    511 non-null    float64
 13  MEDV     511 non-null    float64
```

```
dtypes: float64(11), int64(3)
memory usage: 56.0 KB

dataset.describe()

              CRIM          ZN       INDUS        CHAS         NOX
RM  \
count  511.000000  511.000000  511.000000  511.000000  511.000000
511.000000
mean     3.584139   11.252446   11.151096    0.068493    0.554757
6.279393
std      8.564433   23.234838    6.828175    0.252838    0.115310
0.729270
min      0.006320    0.000000    0.460000    0.000000    0.385000
3.520000
25%      0.082325    0.000000    5.190000    0.000000    0.449000
5.882000
50%      0.261690    0.000000    9.690000    0.000000    0.538000
6.209000
75%      3.621175   12.500000   18.100000    0.000000    0.624000
6.629500
max     88.976200  100.000000   27.740000    1.000000    0.871000
8.780000

               AGE         DIS         RAD         TAX     PTRATIO
B  \
count  511.000000  511.000000  511.000000  511.000000  511.000000
511.000000
mean    68.616243    3.783876    9.485323  407.440313   18.500000
356.600900
std     28.099130    2.098631    8.688469  167.903532    2.200348
90.882679
min      2.900000    1.129600    1.000000  187.000000   12.600000
0.320000
25%     45.050000    2.100350    4.000000  279.500000   17.400000
374.710000
50%     77.300000    3.152300    5.000000  330.000000   19.100000
391.340000
75%     94.050000    5.118000   24.000000  666.000000   20.200000
396.210000
max    100.000000   12.126500   24.000000  711.000000   23.000000
396.900000

             LSTAT        MEDV
count  511.000000  511.000000
mean    12.879550   22.682192
std      7.797416    9.484262
min      1.730000    5.000000
25%      7.065000   17.050000
50%     11.450000   21.200000
```

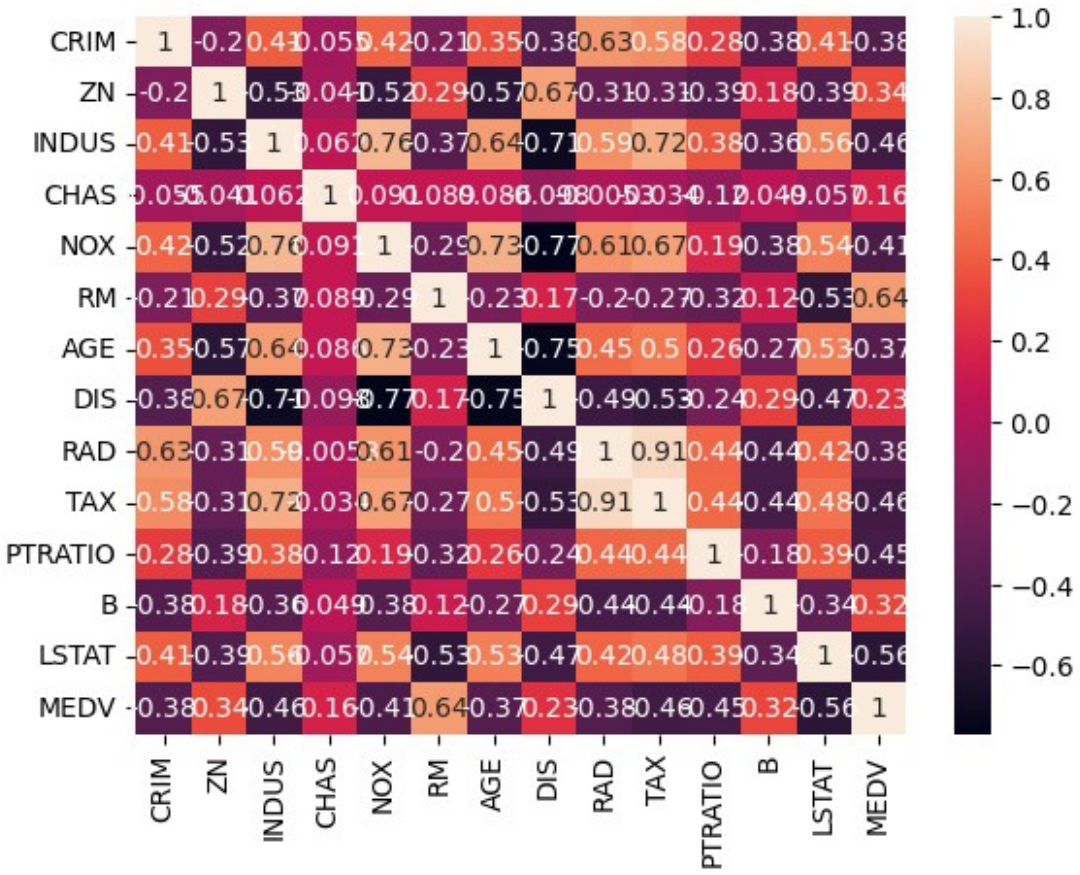
```
75%     17.105000   25.000000
max     76.000000   67.000000

dataset.corr()

             CRIM        ZN     INDUS      CHAS       NOX        RM
AGE  \
CRIM     1.000000 -0.198451  0.405524 -0.054906  0.420524 -0.206994
0.350847
ZN      -0.198451  1.000000 -0.534106 -0.041333 -0.516256  0.288855 -
0.567796
INDUS    0.405524 -0.534106  1.000000  0.062332  0.763583 -0.370202
0.642817
CHAS    -0.054906 -0.041333  0.062332  1.000000  0.091023  0.089403
0.085814
NOX      0.420524 -0.516256  0.763583  0.091023  1.000000 -0.285351
0.729233
RM      -0.206994  0.288855 -0.370202  0.089403 -0.285351  1.000000 -
0.227381
AGE      0.350847 -0.567796  0.642817  0.085814  0.729233 -0.227381
1.000000
DIS     -0.377028  0.665185 -0.707886 -0.097541 -0.768309  0.166746 -
0.745097
RAD      0.625964 -0.307146  0.591784 -0.005343  0.609343 -0.203099
0.452229
TAX      0.583389 -0.311531  0.718764 -0.034244  0.666982 -0.274950
0.503360
PTRATIO  0.276695 -0.392844  0.379441 -0.124415  0.186073 -0.320076
0.258251
B       -0.384356  0.175637 -0.356955  0.048970 -0.379936  0.117844 -
0.271925
LSTAT    0.405030 -0.390029  0.556707 -0.057013  0.540050 -0.534628
0.529994
MEDV    -0.380072  0.339767 -0.463269  0.164782 -0.411486  0.644951 -
0.368203

              DIS       RAD       TAX   PTRATIO         B     LSTAT
MEDV
CRIM    -0.377028  0.625964  0.583389  0.276695 -0.384356  0.405030 -
0.380072
ZN       0.665185 -0.307146 -0.311531 -0.392844  0.175637 -0.390029
0.339767
INDUS   -0.707886  0.591784  0.718764  0.379441 -0.356955  0.556707 -
0.463269
CHAS    -0.097541 -0.005343 -0.034244 -0.124415  0.048970 -0.057013
0.164782
NOX     -0.768309  0.609343  0.666982  0.186073 -0.379936  0.540050 -
0.411486
RM       0.166746 -0.203099 -0.274950 -0.320076  0.117844 -0.534628
0.644951
```

```
AGE     -0.745097  0.452229  0.503360  0.258251 -0.271925  0.529994 -
0.368203
DIS      1.000000 -0.488474 -0.530379 -0.238155  0.290997 -0.467063
0.233469
RAD     -0.488474  1.000000  0.910211  0.438646 -0.442406  0.422389 -
0.379016
TAX     -0.530379  0.910211  1.000000  0.440962 -0.440830  0.482088 -
0.459274
PTRATIO -0.238155  0.438646  0.440962  1.000000 -0.175251  0.393263 -
0.447464
B        0.290997 -0.442406 -0.440830 -0.175251  1.000000 -0.339910
0.317941
LSTAT   -0.467063  0.422389  0.482088  0.393263 -0.339910  1.000000 -
0.562960
MEDV     0.233469 -0.379016 -0.459274 -0.447464  0.317941 -0.562960
1.000000

sns.pairplot(df)

<seaborn.axisgrid.PairGrid at 0x208c256d810>
```



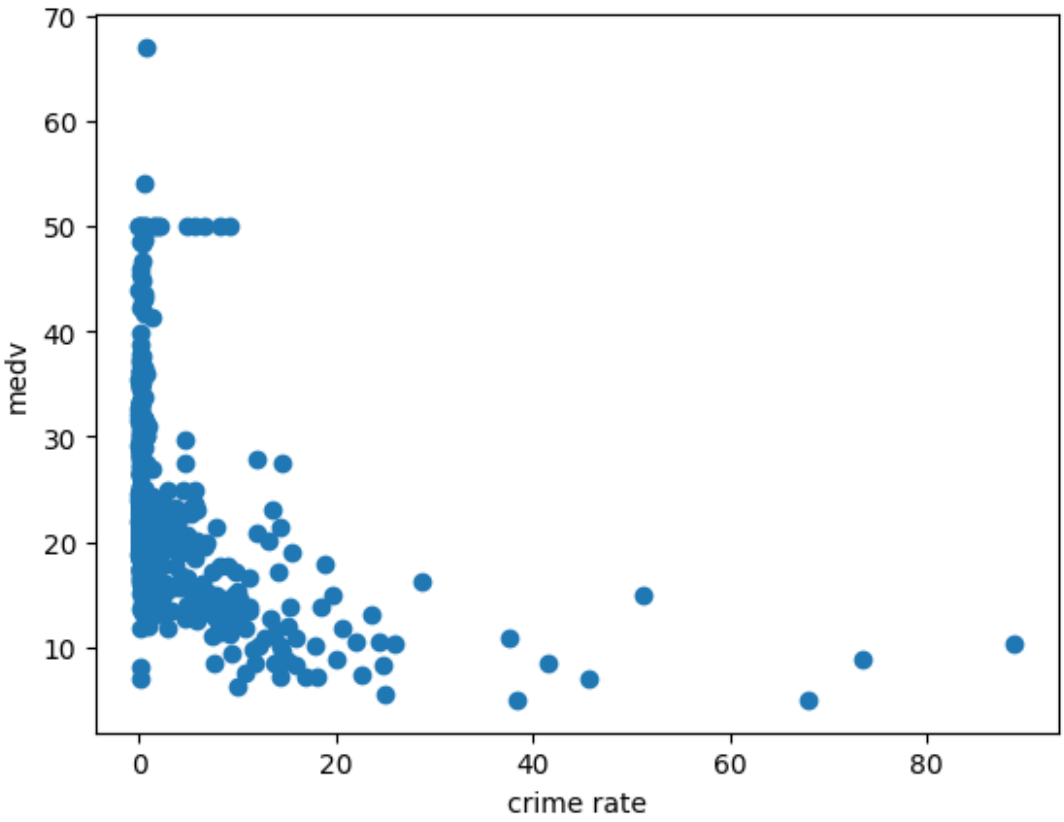

```
sns.heatmap(dataset.corr(),annot = True)
```

```
<Axes: >
```



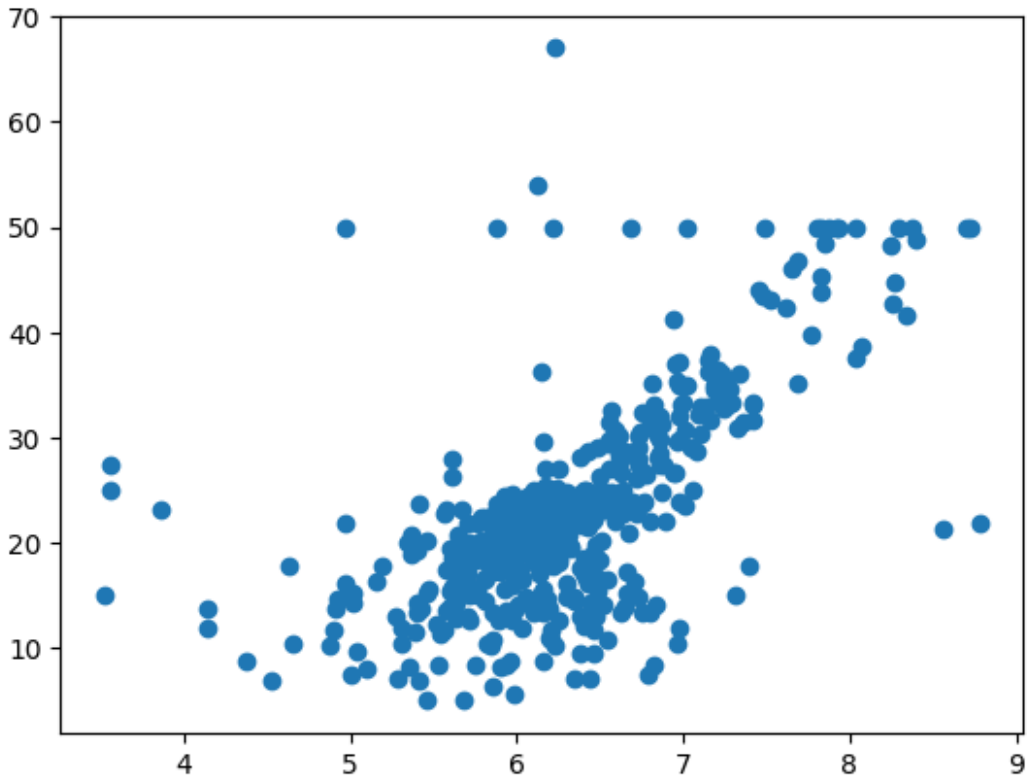

```
plt.scatter(dataset['CRIM'],dataset['MEDV'])
plt.xlabel("crime rate")
plt.ylabel('medv')
```

```
Text(0, 0.5, 'medv')
```



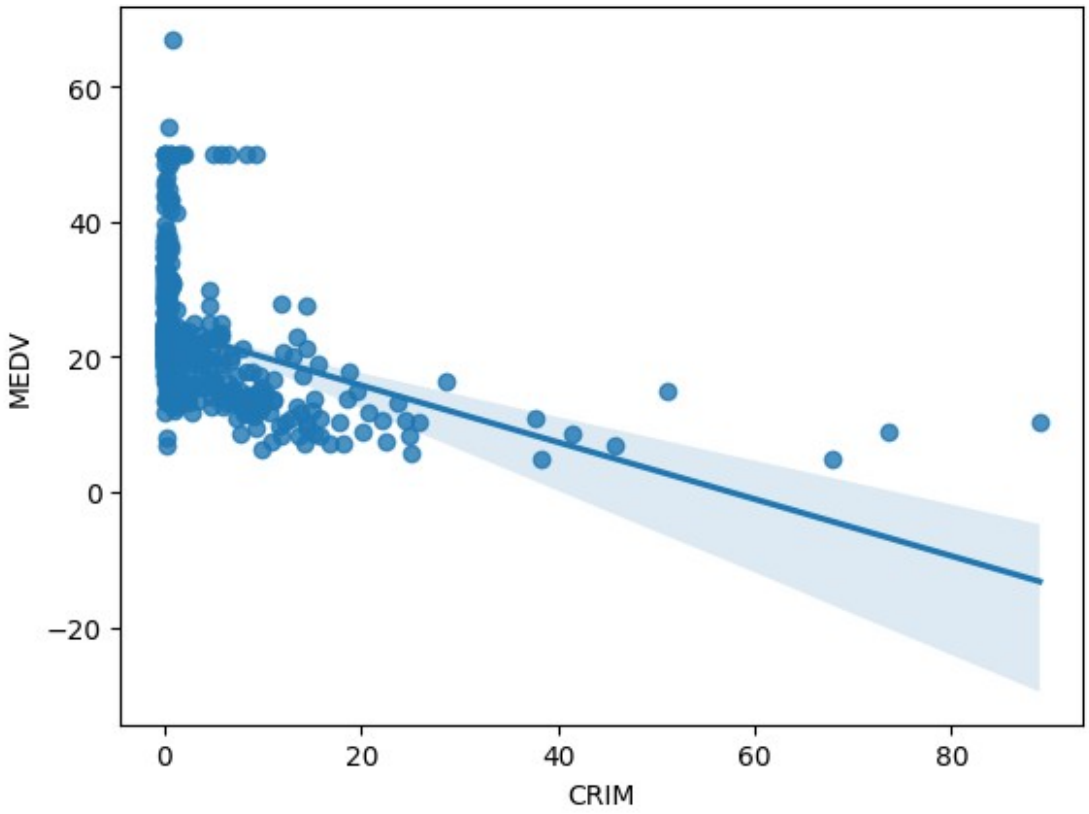

```
plt.scatter(dataset['RM'],dataset['MEDV'])
```

```
<matplotlib.collections.PathCollection at 0x208cf415f90>
```



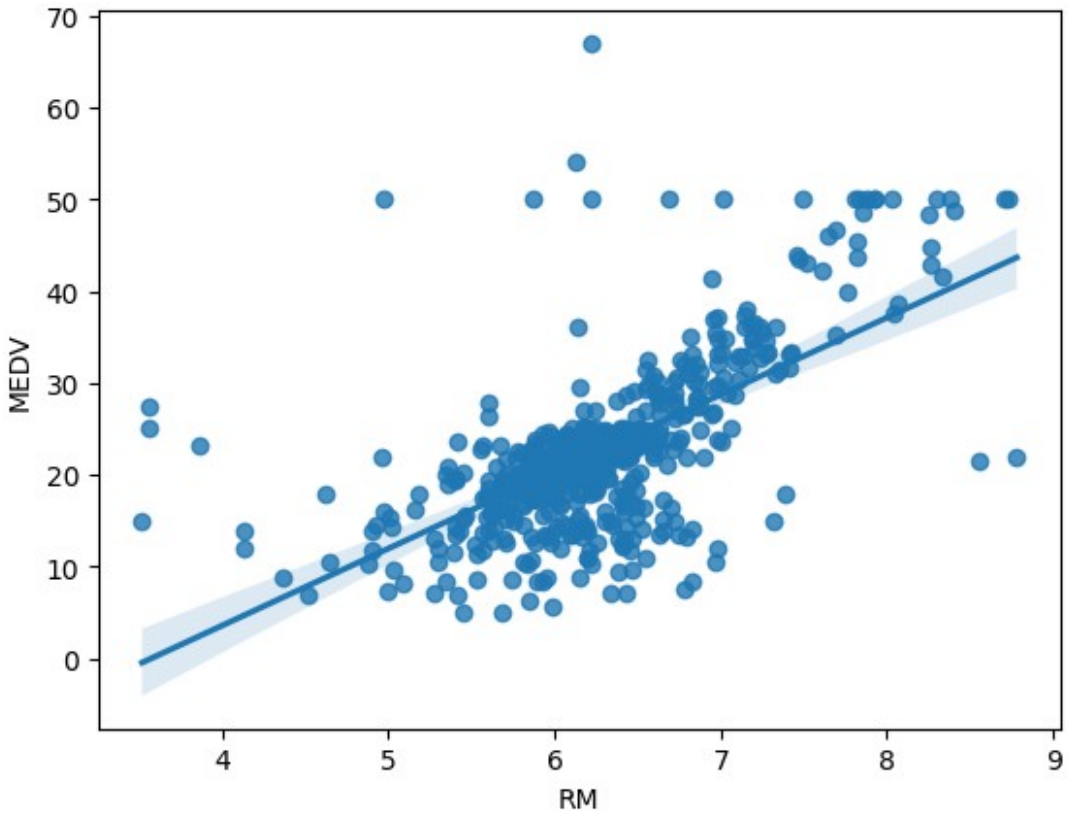

```
sns.regplot(x="CRIM",y="MEDV",data = dataset)
```

```
<Axes: xlabel='CRIM', ylabel='MEDV'>
```

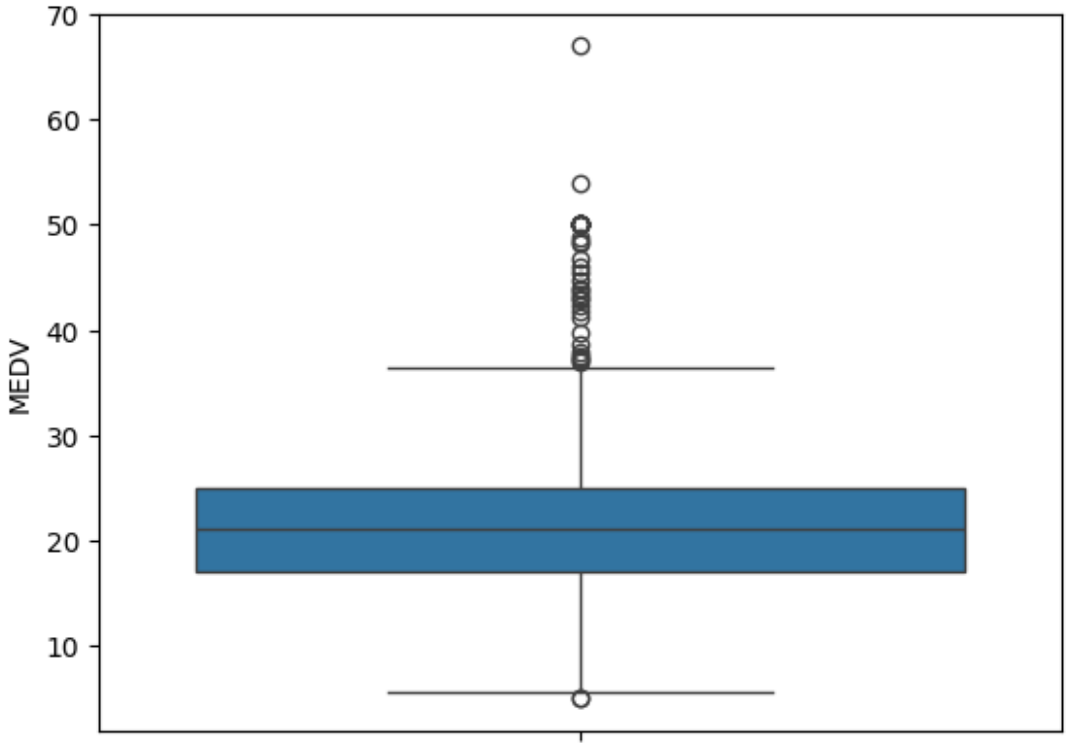



```
sns.regplot(x="RM",y="MEDV",data = dataset)
```

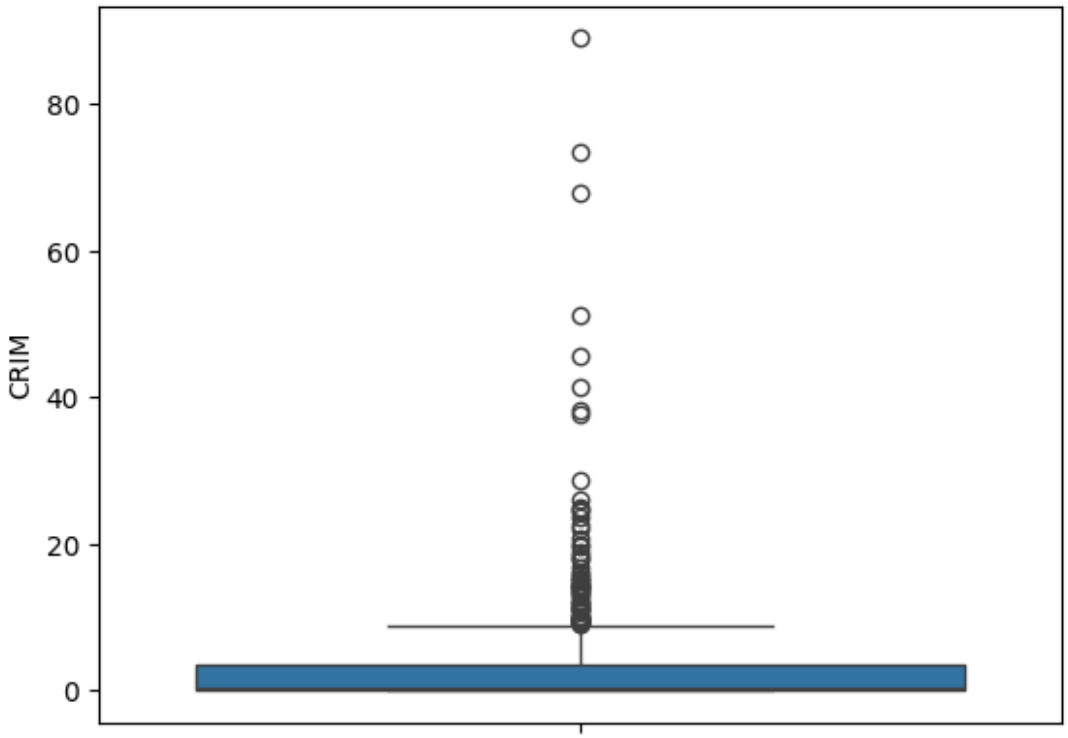
```
<Axes: xlabel='RM', ylabel='MEDV'>
```




```
sns.boxplot(dataset["MEDV"])
plt.show()
```




```
sns.boxplot(dataset["CRIM"])
plt.show()
```



```
def outliers(s):
    q1=df[s].quantile(.25)
    q3=df[s].quantile(.75)
    iqr=q3-q1
    ul=q3+1.5*iqr
    ll=q1-1.5*iqr
    return(ll,ul)

outliers("CRIM")
```

```
(np.float64(-5.22595), np.float64(8.92945))
```

```
outliers("RM")
```

```
(np.float64(4.760749999999999), np.float64(7.750750000000001))
```

```
outliers("ZN")
```

```
(np.float64(-18.75), np.float64(31.25))
```

```
outliers("AGE")
```

```
(np.float64(-28.450000000000003), np.float64(167.55))
```

```
dataset.shape
```

```
(511, 14)
```

```
dataset["CRIM"].median()
```

```
np.float64(0.26169)
```

```
dataset=df[df['CRIM']<8.92945]
dataset
```

```
(dataset["CRIM"]>8.92945).sum()
```

```
np.int64(67)
```

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

X
```

| | CRIM | ZN | INDUS | CHAS | NOX | RM | AGE | DIS | RAD | TAX |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.00632 | 18.0 | 2.31 | 0 | 0.538 | 6.575 | 65.2 | 4.0900 | 1 | 296 |
| 1 | 0.02731 | 0.0 | 7.07 | 0 | 0.469 | 6.421 | 78.9 | 4.9671 | 2 | 242 |
| 2 | 0.02729 | 0.0 | 7.07 | 0 | 0.469 | 7.185 | 61.1 | 4.9671 | 2 | 242 |
| 3 | 0.03237 | 0.0 | 2.18 | 0 | 0.458 | 6.998 | 45.8 | 6.0622 | 3 | 222 |

```
4    0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222

..       ...   ...    ...   ...    ...    ...   ...     ...  ...  ...

506  0.98765   0.0  12.50     0  0.561  6.980  89.0  2.0980    3  320

507  0.23456   0.0  12.50     0  0.561  6.980  76.0  2.6540    3  320

508  0.44433   0.0  12.50     0  0.561  6.123  98.0  2.9870    3  320

509  0.77763   0.0  12.70     0  0.561  6.222  34.0  2.5430    3  329

510  0.65432   0.0  12.80     0  0.561  6.760  67.0  2.9870    3  345


     PTRATIO       B  LSTAT
0       15.3  396.90   4.98
1       17.8  396.90   9.14
2       17.8  392.83   4.03
3       18.7  394.63   2.94
4       18.7  396.90   5.33
..       ...     ...    ...
506     23.0  396.00  12.00
507     23.0  343.00  25.00
508     23.0  343.00  21.00
509     23.0  343.00  76.00
510     23.0  321.00  45.00

[511 rows x 13 columns]

y

0      24.0
1      21.6
2      34.7
3      33.4
4      36.2
       ...
506    12.0
507    32.0
508    54.0
509    67.0
510    24.0
Name: MEDV, Length: 511, dtype: float64

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.33, random_state=10)

X_train.shape
```

```
(342, 13)
```

```
y_train.shape
```

```
(342,)
```

```
X_test.shape
```

```
(169, 13)
```

```
y_test.shape
```

```
(169,)
```

```
dataset["CRIM"]
```

```
0      0.00632
1      0.02731
2      0.02729
3      0.03237
4      0.06905
        ...
506    0.98765
507    0.23456
508    0.44433
509    0.77763
510    0.65432
Name: CRIM, Length: 511, dtype: float64
```

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
```

```
X_train=scaler.fit_transform(X_train)
```

```
X_test=scaler.transform(X_test)
```

```
X_train
```

```
array([[-0.41439929, -0.48202008, -1.26559838, ..., -0.36860961,
         0.38410572,  0.14286711],
       [-0.16444387, -0.48202008,  1.16969306, ..., -1.76282977,
         0.39184726,  2.19433483],
       [-0.41792627, -0.48202008, -0.87726044, ...,  0.71078664,
         0.3700401 , -0.95212852],
       ...,
       [-0.1633557 , -0.48202008,  0.95916553, ...,  0.71078664,
         0.16309014,  0.05843371],
       [-0.40500056, -0.48202008,  2.03314043, ...,  0.21606336,
         0.31890231,  0.25236667],
       [-0.34019155,  0.32164046, -1.0508034 , ..., -2.52740211,
         0.39937073, -0.32283585]], shape=(342, 13))
```

```
X_test
```

```
array([[-0.13525904, -0.48202008, -0.20727079, ..., -0.09876055,
        -0.05803447, -0.03391532],
       [-0.28966984, -0.48202008,  1.16969306, ..., -1.76282977,
         0.08349401, -1.09592915],
       [ 0.04254796, -0.48202008,  0.95916553, ...,  0.71078664,
         0.38127079, -0.0299575 ],
       ...,
       [ 0.09625216, -0.48202008,  0.95916553, ...,  0.71078664,
        -3.32398389,  0.69036493],
       [-0.42109345,  1.92896154, -1.19874167, ..., -1.35805617,
         0.4099472 , -1.03788119],
       [-0.41238153, -0.48202008,  0.20951682, ...,  0.03616398,
         0.19503763, -0.12230653]], shape=(169, 13))
```

```
from sklearn.linear_model import LinearRegression
```

```
regression=LinearRegression()
```

```
regression.fit(X_train,y_train)
```

```
LinearRegression()
```

```
reg_pred=regression.predict(X_test)
```

```
reg_pred
```

```
array([ 19.98589533,  31.45526324,  19.59311181,  24.32210658,
        28.65239095,  12.53813937,  24.07053296,  22.05487354,
        24.89438288,  24.25190669,  32.82469698,  16.34519674,
        21.56455789,  32.37302369,  22.81157581,  20.98905942,
        35.13749789,  36.13891021,  19.06078533,  18.50783589,
        19.61797837,  24.41306731,  19.31342742,  20.63629152,
        27.19826898,  32.17514179,  23.67638675,  21.56756887,
        31.85353367,  35.53316975,  27.66098093,  16.97125538,
       -10.27569514,  25.23276966,  23.36747744,  27.93420839,
        15.46511035,  32.04879176,  14.22574443,  23.29641925,
        24.36276523,  29.27148426,  33.95044511,  21.2986439 ,
        14.8935081 ,  24.79478944,   4.76895317,  19.67801423,
        31.6116077 ,  24.00135581,  19.24845622,  30.34424731,
        20.47177245,  21.33555603,  17.352779  ,  19.39571463,
        23.71026394,  40.07132318,  16.68689291,  28.46186716,
        34.36953395,  34.56498629,  18.15940759,  19.67860881,
        32.69745379,  20.7894055 ,  15.2048199 ,  16.25439973,
        23.7828687 ,  22.56373662,  32.24705347,  39.54029996,
        17.18436899,  15.9986714 ,  19.0917243 ,  21.50077783,
        34.13900845,  30.17485542,  30.1694527 ,  14.43289604,
        37.52295461,  19.13272   ,  24.39150369,  22.92656843,
        13.79039387,  36.06891963,  33.94588756,  27.63690063,
        21.86774329,  19.26172629,  26.37077488,  17.55771661,
```

```
       33.93727218,  14.02486744,  23.51597365,  24.69766797,
       24.18152769,  19.75581542,  10.27364156,  27.34721329,
       29.48391432,  19.06112325,  25.28128478,  17.96120099,
        3.92877823,  21.77087245,  27.8164002 ,  15.06357486,
       20.9073709 ,  30.53538851,  27.01387813,  21.39953587,
       23.32641309,  16.9841283 ,  23.54058351,  18.30513425,
       26.1564794 ,  25.07903691,  22.90716366,   9.89417608,
       18.26575831,  18.12962688,  34.33410557,   8.67245294,
       28.49975845,  24.47791253,  19.99168584,  15.64631398,
       16.42932162,  26.0183806 ,  20.96568148,  14.57644367,
       31.52386014,  26.27641314,  37.9927828 ,  19.54774116,
       19.74205644,  23.97378827,  24.69811483,  39.65534149,
       31.27639108,  27.71310802,  29.17987973,  28.5879573 ,
       12.68138443,  18.45594585,  31.23890506,  17.5373893 ,
       15.96218637,  14.54511307,  25.95493798,  23.83540554,
       27.49694938,  21.89388761,  21.17599861,  34.40904416,
       19.87379061,  26.43986821,  27.99812994,  24.88505292,
       19.95486094,  15.10219404,  25.65195737,  17.99968273,
       19.26950049,  28.12088645,  16.51334319,  32.47883265,
       22.74362601])
```

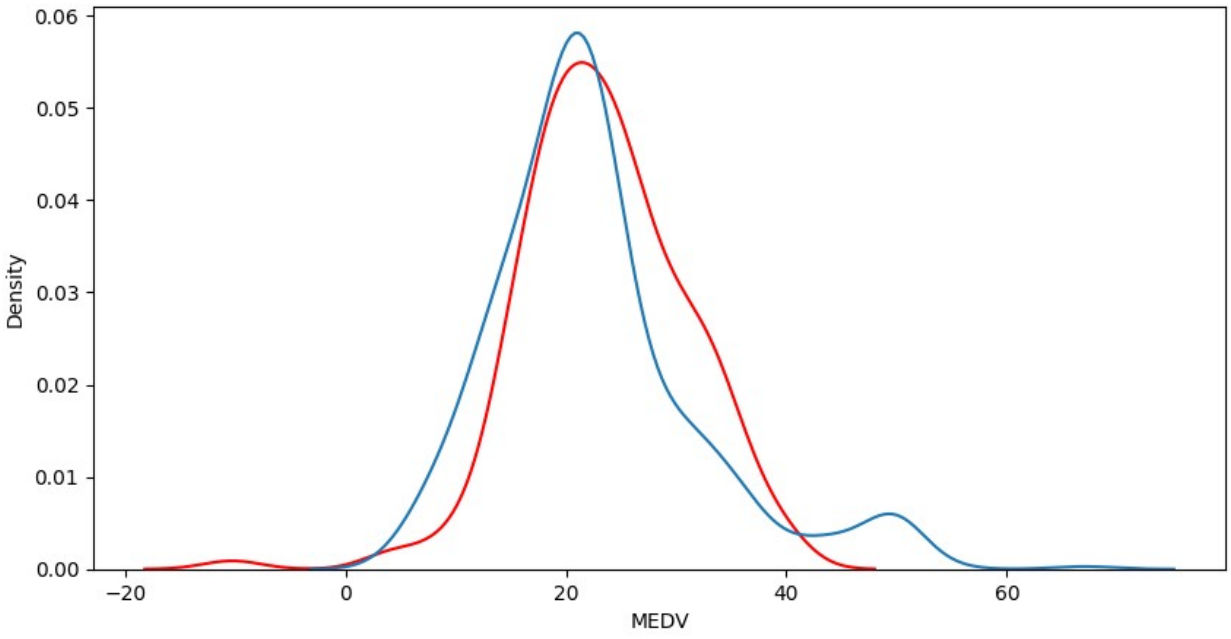
```
y_test
```

```
310    16.1
157    41.3
359    22.6
43     24.7
181    36.2
       ...
460    16.4
308    22.8
455    14.1
193    31.1
76     20.0
Name: MEDV, Length: 169, dtype: float64
```

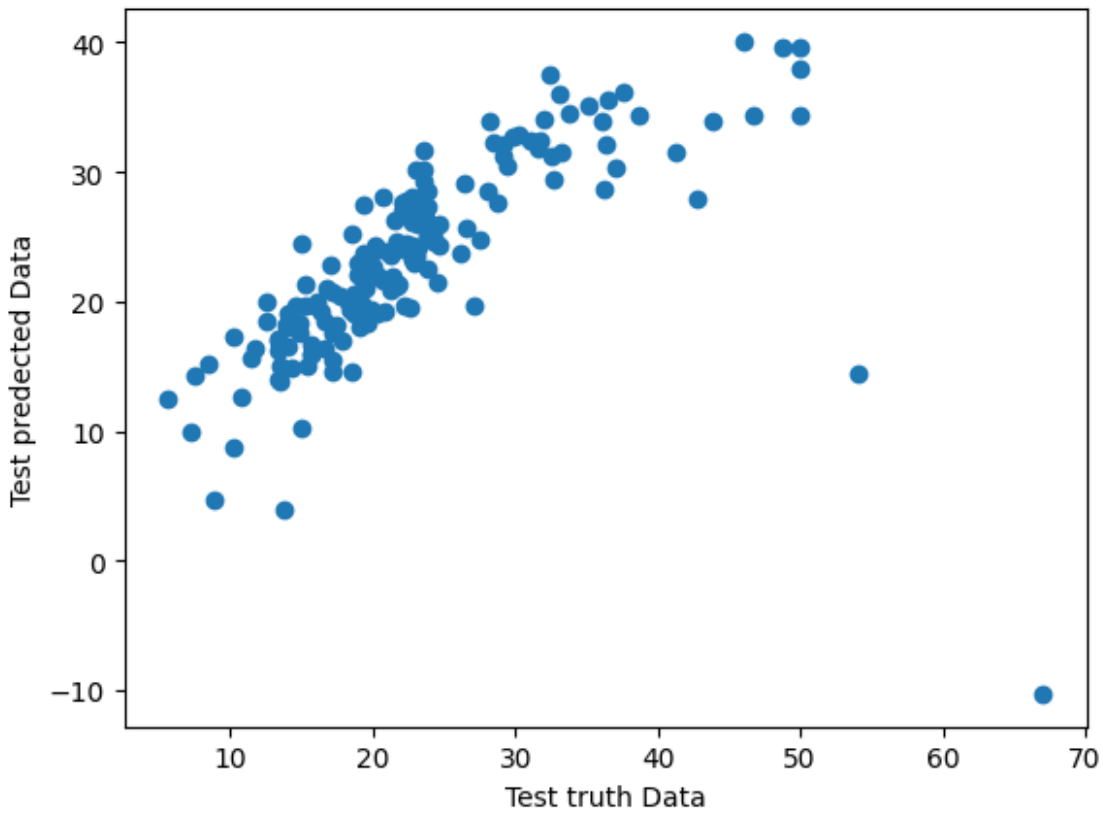
```
plt.figure(figsize=(10,5))
sns.distplot(reg_pred,hist=False,color="red")
sns.distplot(dataset["MEDV"],hist=False)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_14888\1488617724.py:2:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).
```

```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(reg_pred,hist=False,color="red")
C:\Users\hp\AppData\Local\Temp\ipykernel_14888\1488617724.py:3:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `kdeplot` (an axes-level function for kernel
density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(dataset["MEDV"],hist=False)

<Axes: xlabel='MEDV', ylabel='Density'>
```




```
plt.scatter(y_test,reg_pred)
plt.xlabel('Test truth Data')
plt.ylabel('Test predected Data')

Text(0, 0.5, 'Test predected Data')
```




```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
print(mean_squared_error(y_test,reg_pred))
print(mean_absolute_error(y_test,reg_pred))
print(np.sqrt(mean_squared_error(y_test,reg_pred)))
```

```
64.01362373830402
3.992903676554756
8.000851438334799
```

```
from sklearn.metrics import r2_score
```

```
r2_score(y_test,reg_pred)
```

```
0.3065524754881511
```

```
from sklearn.linear_model import Ridge
```

```
clf = Ridge()
```

```
clf.fit(X_train, y_train)
```

```
Ridge()
```

```
r_pred=clf.predict(X_test)
```

```
y_test
```

```
310    16.1
157    41.3
359    22.6
43     24.7
181    36.2
       ...
460    16.4
308    22.8
455    14.1
193    31.1
76     20.0
Name: MEDV, Length: 169, dtype: float64
```

```
r2_score(y_test,r_pred)
```

```
0.30851570863297906
```