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

df = pd.read_csv("/content/HousingData.csv")

df
```

```
        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX
\
0    0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296

1    0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242

2    0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242

3    0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222

4    0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222

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

501  0.06263   0.0  11.93   0.0  0.573  6.593  69.1  2.4786    1  273

502  0.04527   0.0  11.93   0.0  0.573  6.120  76.7  2.2875    1  273

503  0.06076   0.0  11.93   0.0  0.573  6.976  91.0  2.1675    1  273

504  0.10959   0.0  11.93   0.0  0.573  6.794  89.3  2.3889    1  273

505  0.04741   0.0  11.93   0.0  0.573  6.030   NaN  2.5050    1  273


     PTRATIO       B  LSTAT  MEDV
0       15.3  396.90   4.98  24.0
1       17.8  396.90   9.14  21.6
2       17.8  392.83   4.03  34.7
3       18.7  394.63   2.94  33.4
4       18.7  396.90    NaN  36.2
..       ...     ...    ...   ...
501     21.0  391.99    NaN  22.4
502     21.0  396.90   9.08  20.6
503     21.0  396.90   5.64  23.9
504     21.0  393.45   6.48  22.0
505     21.0  396.90   7.88  11.9

[506 rows x 14 columns]
```

```python
df.isnull().sum()
```

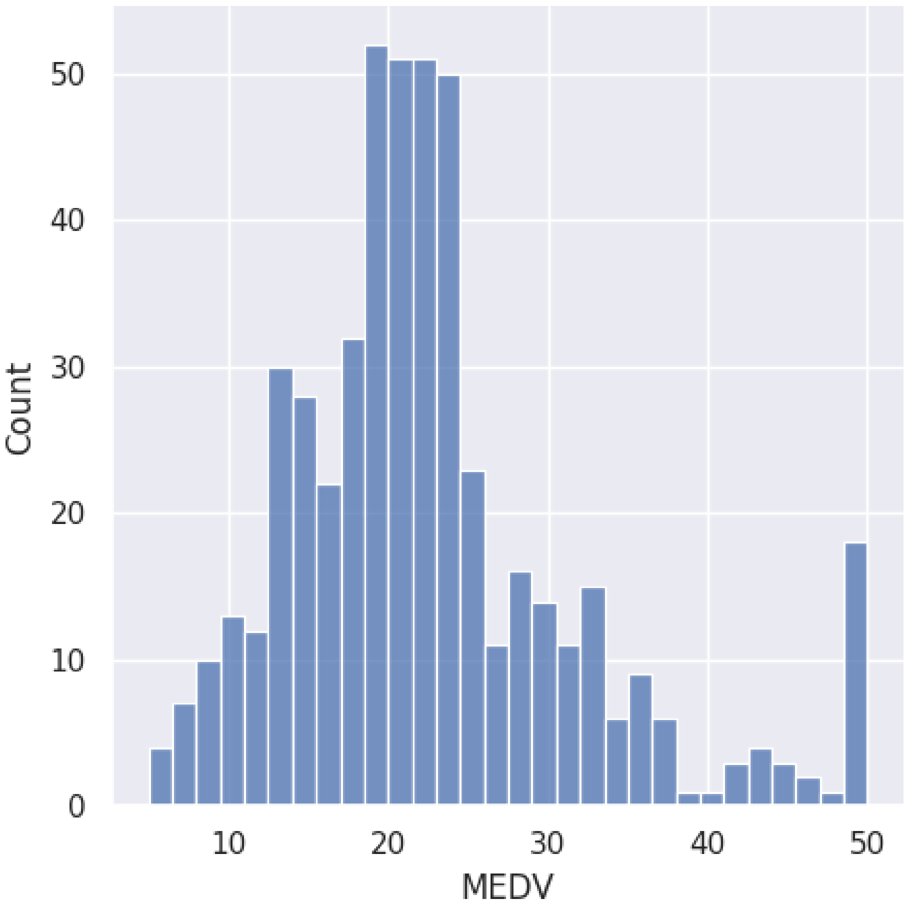
```
CRIM       20
ZN         20
INDUS      20
CHAS       20
NOX         0
RM          0
AGE        20
DIS         0
RAD         0
TAX         0
PTRATIO     0
B           0
LSTAT      20
MEDV        0
dtype: int64
```

```
df.fillna(method ='ffill',inplace=True)
```

```
df.isnull().sum()
```

```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
MEDV       0
dtype: int64
```

```
sns.set(rc={'figure.figsize':(11.7,8.27)})
sns.displot(df['MEDV'], bins=30)
plt.show()
```



```
corr = df.corr()
corr
```

```
                 CRIM        ZN     INDUS      CHAS       NOX        RM
AGE  \
CRIM     1.000000 -0.200017  0.407614 -0.055486  0.422366 -0.221545
0.349144
ZN      -0.200017  1.000000 -0.526815 -0.040399 -0.511056  0.302417 -
0.565795
INDUS    0.407614 -0.526815  1.000000  0.054064  0.763651 -0.391676
0.639985
CHAS    -0.055486 -0.040399  0.054064  1.000000  0.070867  0.106797
0.087057
NOX      0.422366 -0.511056  0.763651  0.070867  1.000000 -0.302188
0.729830
RM      -0.221545  0.302417 -0.391676  0.106797 -0.302188  1.000000 -
0.235960
AGE      0.349144 -0.565795  0.639985  0.087057  0.729830 -0.235960
1.000000
DIS     -0.379364  0.653589 -0.708027 -0.092318 -0.769230  0.205246 -
0.744977
RAD      0.627112 -0.310159  0.595129 -0.003339  0.611441 -0.209847
```

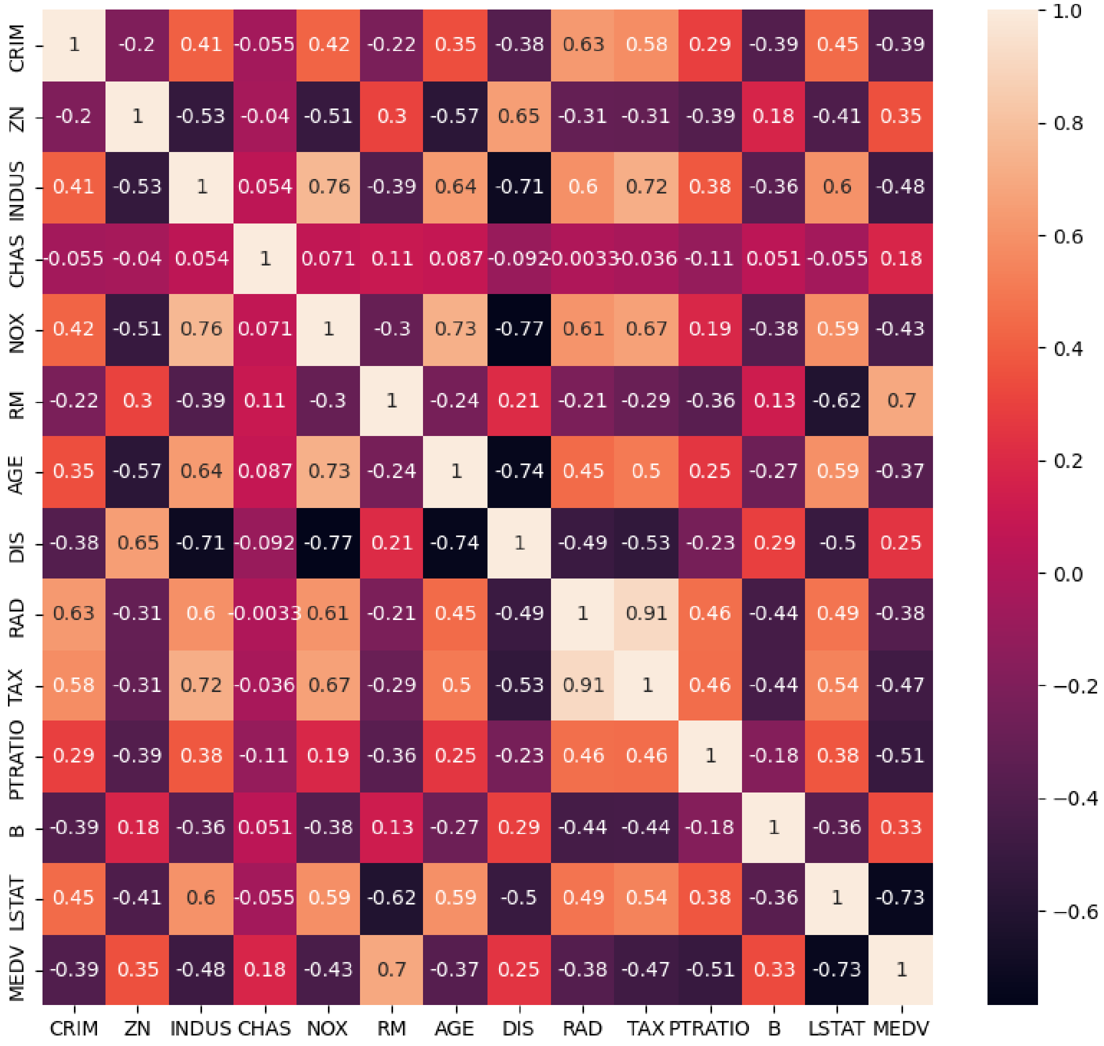
```
0.452431
TAX      0.584277 -0.313370  0.720760 -0.035822  0.668023 -0.292048
0.503086
PTRATIO  0.290771 -0.385452  0.383248 -0.109451  0.188933 -0.355501
0.254110
B       -0.386169  0.175763 -0.356977  0.050608 -0.380051  0.128069 -
0.272501
LSTAT    0.452563 -0.408091  0.601579 -0.054583  0.589646 -0.615892
0.592692
MEDV    -0.391776  0.353895 -0.483725  0.183844 -0.427321  0.695360 -
0.371413

              DIS       RAD       TAX   PTRATIO         B     LSTAT
MEDV
CRIM    -0.379364  0.627112  0.584277  0.290771 -0.386169  0.452563 -
0.391776
ZN       0.653589 -0.310159 -0.313370 -0.385452  0.175763 -0.408091
0.353895
INDUS   -0.708027  0.595129  0.720760  0.383248 -0.356977  0.601579 -
0.483725
CHAS    -0.092318 -0.003339 -0.035822 -0.109451  0.050608 -0.054583
0.183844
NOX     -0.769230  0.611441  0.668023  0.188933 -0.380051  0.589646 -
0.427321
RM       0.205246 -0.209847 -0.292048 -0.355501  0.128069 -0.615892
0.695360
AGE     -0.744977  0.452431  0.503086  0.254110 -0.272501  0.592692 -
0.371413
DIS      1.000000 -0.494588 -0.534432 -0.232471  0.291512 -0.501481
0.249929
RAD     -0.494588  1.000000  0.910228  0.464741 -0.444413  0.487754 -
0.381626
TAX     -0.534432  0.910228  1.000000  0.460853 -0.441808  0.542354 -
0.468536
PTRATIO -0.232471  0.464741  0.460853  1.000000 -0.177383  0.377126 -
0.507787
B        0.291512 -0.444413 -0.441808 -0.177383  1.000000 -0.363509
0.333461
LSTAT   -0.501481  0.487754  0.542354  0.377126 -0.363509  1.000000 -
0.732489
MEDV     0.249929 -0.381626 -0.468536 -0.507787  0.333461 -0.732489
1.000000
```

```
plt.subplots(figsize=(10,9))
```

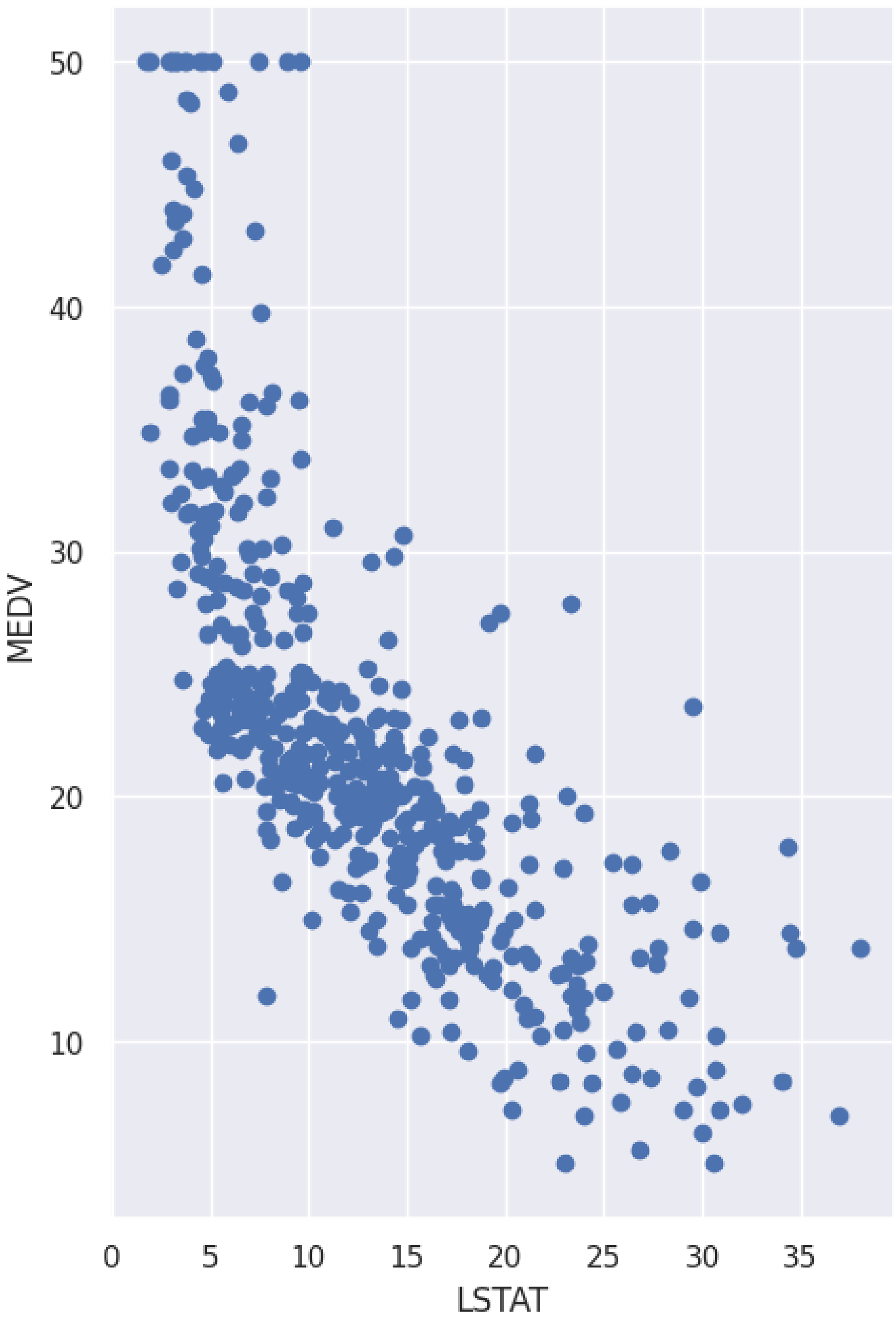
```
sns.heatmap(corr,annot = True)
```

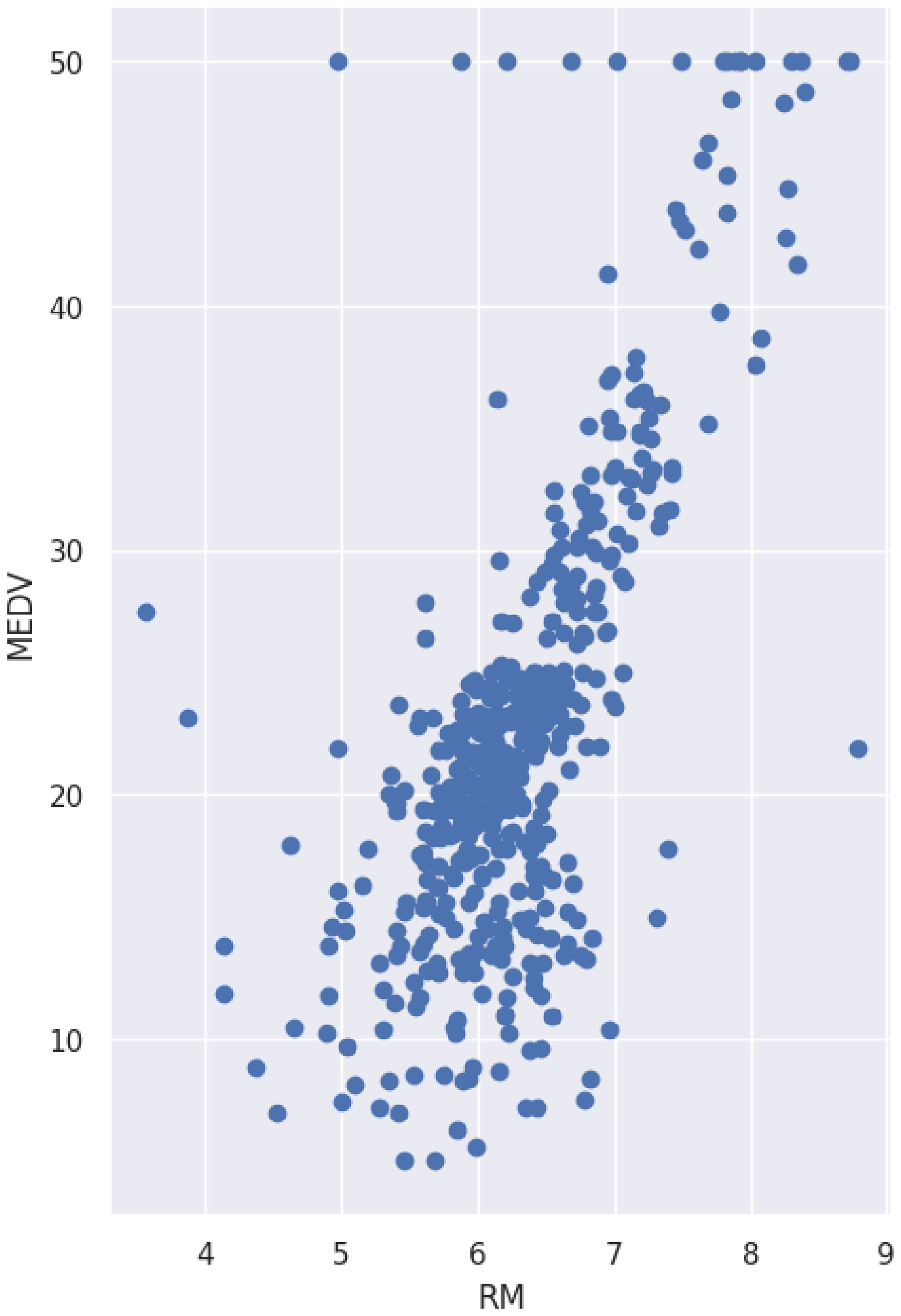
```
<Axes: >
```



| | CRIM | ZN | INDUS | CHAS | NOX | RM | AGE | DIS | RAD | TAX | PTRATIO | B | LSTAT | MEDV |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CRIM | 1 | -0.2 | 0.41 | -0.055 | 0.42 | -0.22 | 0.35 | -0.38 | 0.63 | 0.58 | 0.29 | -0.39 | 0.45 | -0.39 |
| ZN | -0.2 | 1 | -0.53 | -0.04 | -0.51 | 0.3 | -0.57 | 0.65 | -0.31 | -0.31 | -0.39 | 0.18 | -0.41 | 0.35 |
| INDUS | 0.41 | -0.53 | 1 | 0.054 | 0.76 | -0.39 | 0.64 | -0.71 | 0.6 | 0.72 | 0.38 | -0.36 | 0.6 | -0.48 |
| CHAS | -0.055 | -0.04 | 0.054 | 1 | 0.071 | 0.11 | 0.087 | -0.092 | -0.0033 | -0.036 | -0.11 | 0.051 | -0.055 | 0.18 |
| NOX | 0.42 | -0.51 | 0.76 | 0.071 | 1 | -0.3 | 0.73 | -0.77 | 0.61 | 0.67 | 0.19 | -0.38 | 0.59 | -0.43 |
| RM | -0.22 | 0.3 | -0.39 | 0.11 | -0.3 | 1 | -0.24 | 0.21 | -0.21 | -0.29 | -0.36 | 0.13 | -0.62 | 0.7 |
| AGE | 0.35 | -0.57 | 0.64 | 0.087 | 0.73 | -0.24 | 1 | -0.74 | 0.45 | 0.5 | 0.25 | -0.27 | 0.59 | -0.37 |
| DIS | -0.38 | 0.65 | -0.71 | -0.092 | -0.77 | 0.21 | -0.74 | 1 | -0.49 | -0.53 | -0.23 | 0.29 | -0.5 | 0.25 |
| RAD | 0.63 | -0.31 | 0.6 | -0.0033 | 0.61 | -0.21 | 0.45 | -0.49 | 1 | 0.91 | 0.46 | -0.44 | 0.49 | -0.38 |
| TAX | 0.58 | -0.31 | 0.72 | -0.036 | 0.67 | -0.29 | 0.5 | -0.53 | 0.91 | 1 | 0.46 | -0.44 | 0.54 | -0.47 |
| PTRATIO | 0.29 | -0.39 | 0.38 | -0.11 | 0.19 | -0.36 | 0.25 | -0.23 | 0.46 | 0.46 | 1 | -0.18 | 0.38 | -0.51 |
| B | -0.39 | 0.18 | -0.36 | 0.051 | -0.38 | 0.13 | -0.27 | 0.29 | -0.44 | -0.44 | -0.18 | 1 | -0.36 | 0.33 |
| LSTAT | 0.45 | -0.41 | 0.6 | -0.055 | 0.59 | -0.62 | 0.59 | -0.5 | 0.49 | 0.54 | 0.38 | -0.36 | 1 | -0.73 |
| MEDV | -0.39 | 0.35 | -0.48 | 0.18 | -0.43 | 0.7 | -0.37 | 0.25 | -0.38 | -0.47 | -0.51 | 0.33 | -0.73 | 1 |

```python
features = ['LSTAT','RM']
target = df['MEDV']
for i,col in enumerate(features):
  plt.subplot(1 ,len(features), i+1)
  x = df[col]
  y  = target
  plt.scatter(x,y)
  plt.xlabel(col)
  plt.ylabel('MEDV')
  plt.show()
```


50
40
30
20
10
MEDV
0
5
10
15
20
25
30
35
LSTAT



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

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=0)

from sklearn.linear_model import LinearRegression
reg = LinearRegression()
```

```
reg.fit(x_train,y_train)
print(reg.coef_,reg.intercept_)
```

```
[-1.22341809e-01  4.41363869e-02 -1.34419395e-03  2.55872583e+00
 -1.56313766e+01  3.69035255e+00 -6.13607953e-03 -1.41415596e+00
  2.38632648e-01 -1.09194727e-02 -1.03687110e+00  8.11270396e-03
 -4.90318222e-01] 38.188474058519745
```

```
from sklearn.metrics import r2_score
y_pred_train = reg.predict(x_train)
r2_score(y_train,y_pred_train)
```

```
0.7731269304116333
```

```
y_pred_test = reg.predict(x_test)
r2_score(y_test,y_pred_test)
```

```
0.5724529611713153
```

```
reg.predict(df.iloc[3:4,0:13])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:432:
UserWarning: X has feature names, but LinearRegression was fitted
without feature names
  warnings.warn(
```

```
array([28.65583765])
```

```
df.iloc[3:4,:]
```

```
      CRIM   ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX
PTRATIO  \
3  0.03237  0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222
18.7


        B  LSTAT  MEDV
3  394.63   2.94  33.4
```

```
from sklearn.metrics import mean_squared_error
rmse = np.sqrt(mean_squared_error(y_train,y_pred_train))
rmse
```

```
4.395090120040799
```