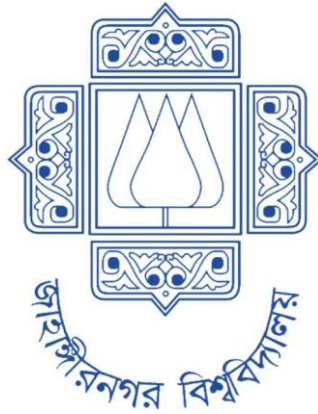


Institute of Information Technology (IIT)
Jahangirnagar University



Lab Report: 04

Submitted by:

Name: Md. Shakil Hossain

Roll No: 2023

Lab Date: 11-July-2023

Submission Date: 11-Aug-2023

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [6]: house = pd.read_csv('housing.csv')
```

```
In [7]: house.head()
```

```
0.00632 18.00 2.3100 0.5380 6.5750 65.20 4.0900 1 296.0 15.30 396.90 4.98 24.00
```

Out[7]:

```
0          0.02731 0.00 7.070 0 0.4690 6.4210 78...
1          0.02729 0.00 7.070 0 0.4690 7.1850 61...
2          0.03237 0.00 2.180 0 0.4580 6.9980 45...
3          0.06905 0.00 2.180 0 0.4580 7.1470 54...
4          0.02985 0.00 2.180 0 0.4580 6.4300 58...
```

```
In [8]: house.columns
```

Out[8]: Index(['0.00632' 18.00 2.310 0 0.5380 6.5750 65.20 4.0900 1 296.0 15.30 396.90 4.98 24.00'], dtype='object')

```
In [16]: col = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
```

```
In [19]: house_data = pd.read_csv('housing.csv', header=None, delimiter = r"\s+", names=
```

```
In [20]: house_data.head()
```

Out[20]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTA
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.9
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.0
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.9
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.3

```
[21]: house_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
```

```

#   Column   Non-Null Count  Dtype
---  -
0CRIM      506 non-null    float64
1ZN         506 non-null    float64
2INDUS      506 non-null    float64
3CHAS       506 non-null    int64
4NOX        506 non-null    float64
5RM         506 non-null    float64
6AGE        506 non-null    float64
7DIS        506 non-null    float64
8RAD        506 non-null    int64
9TAX        506 non-null    float64
10          PTRATIO  506 non-null    float64
11          B        506 non-null    float64
12          LSTAT   506 non-null    float64
13          MEDV    506 non-null    float64 dtypes: float64(12), int64(2)
memory usage: 55.5 KB

```

In [22]: `house_data.describe()`

```

Out[22]:

```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12

In [24]: `house_data.columns`

```

Out[24]: Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
                'PTRATIO', 'B', 'LSTAT', 'MEDV'],
                dtype='object')

```

In [25]: `house_data.index`

```

Out[25]: RangeIndex(start=0, stop=506, step=1)
[27]: house_data.isnull().sum()

```

In

```
Out[27]: 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
```

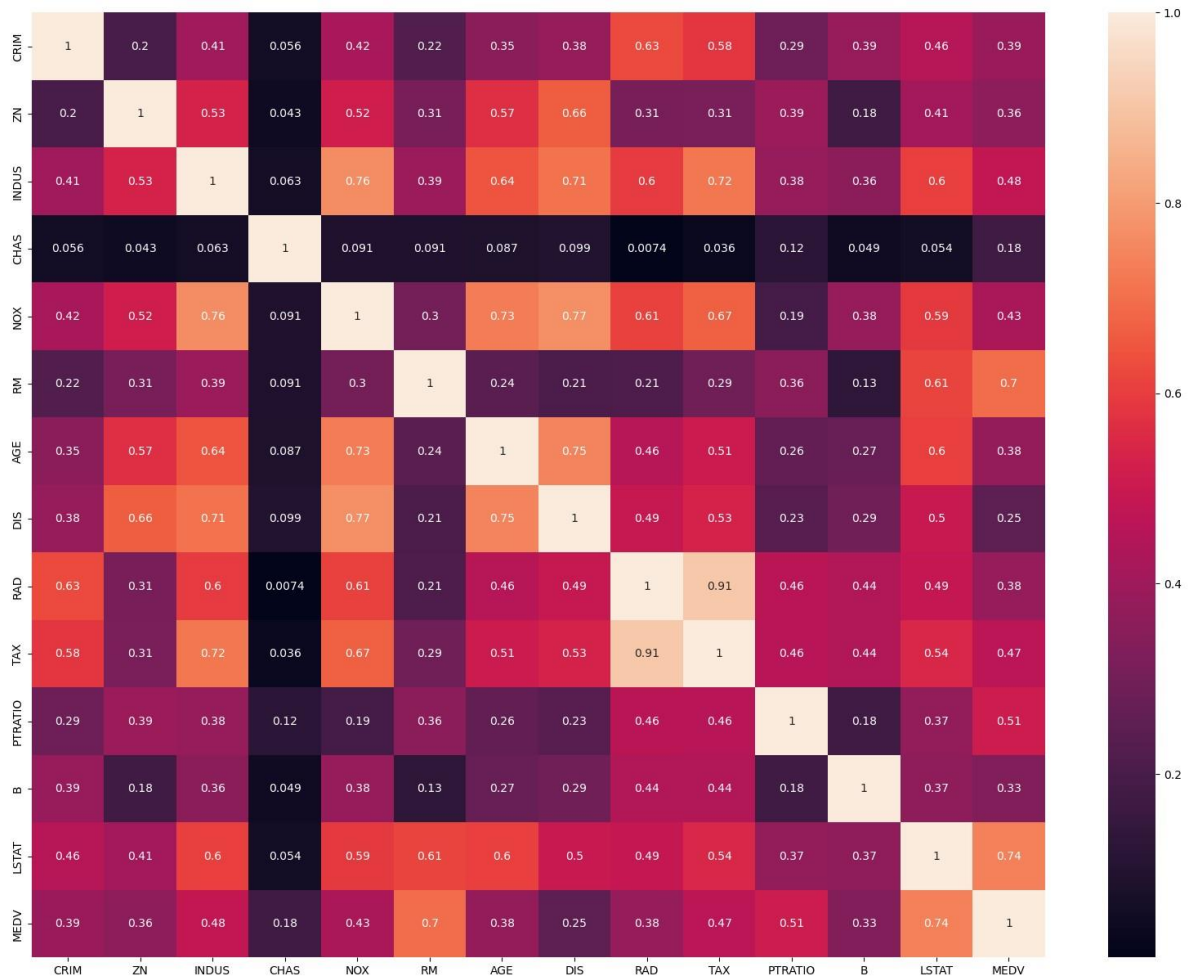
```
In [29]: corr = house_data.corr()
          print(house_data)
```

	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	296.0	
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	..
...	
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	
	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	5.33	36.2	
501	21.0	391.99	9.67	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]

```
[34]: plt.figure(figsize=(20,15))
sns.heatmap(house_data.corr().abs(),annot=True)
```

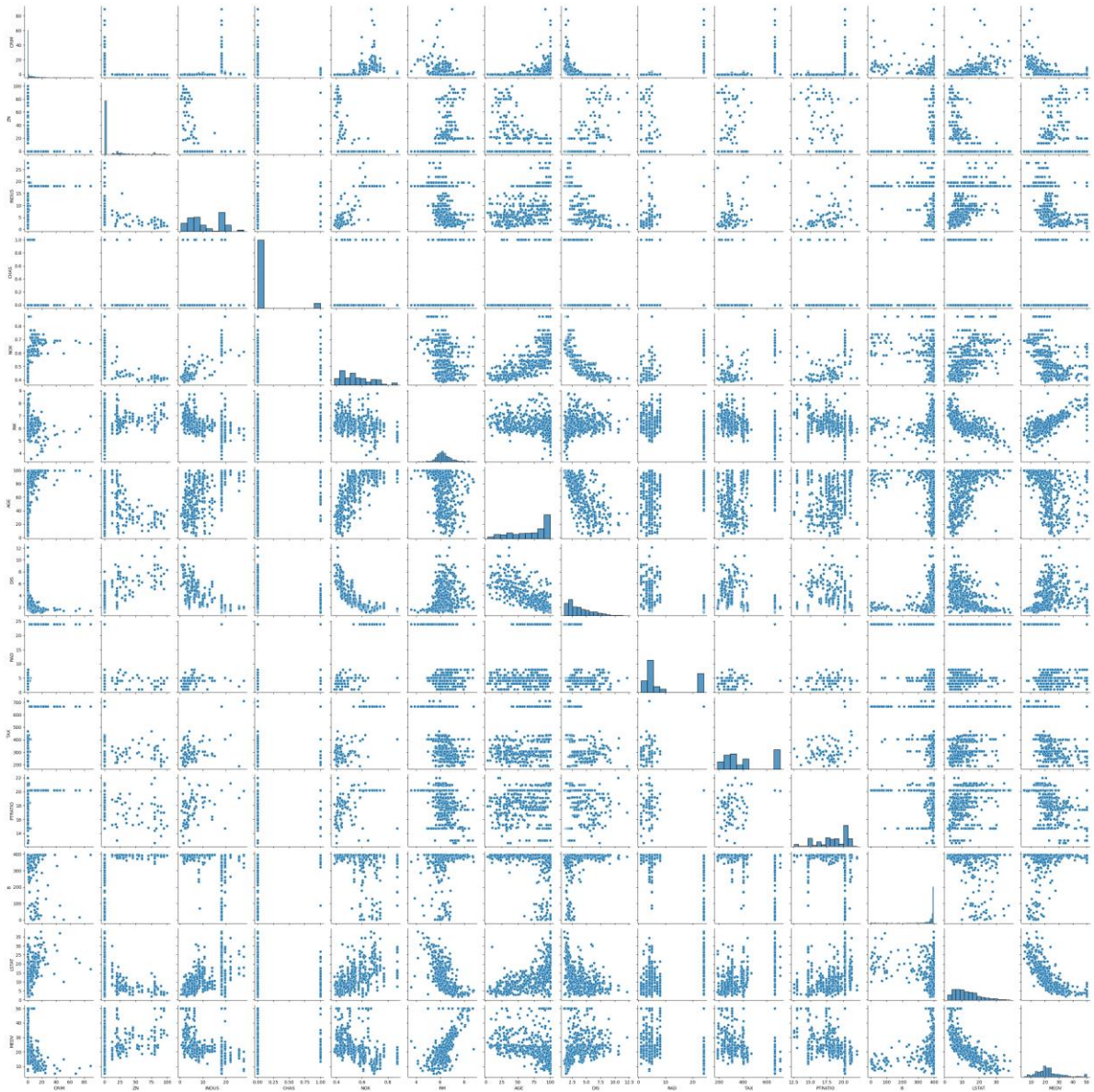
Out[34]: <Axes: >



```
[35]: sns.pairplot(house_data)
```

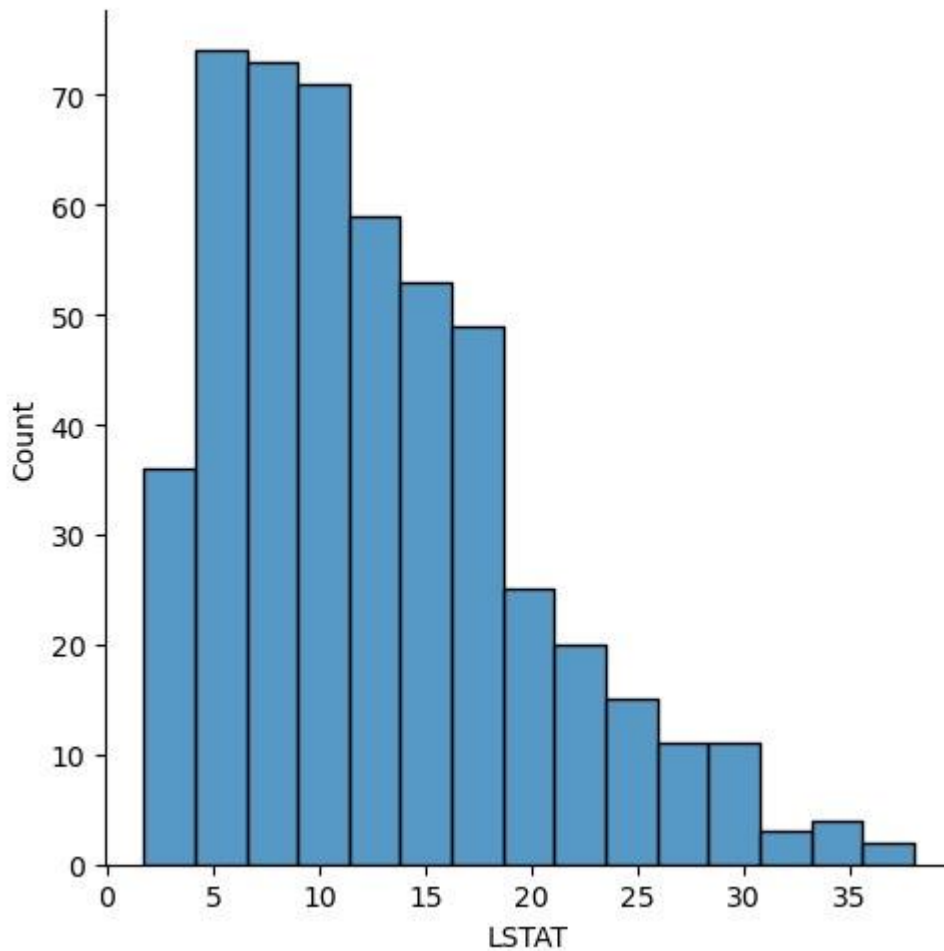
Out[35]: <seaborn.axisgrid.PairGrid at 0x289de2236d0>

In



```
[36]: sns.displot(house_data['LSTAT'])
```

```
Out[36]: <seaborn.axisgrid.FacetGrid at 0x289ea07a230>
```



```
[37]: sns.distplot(house_data['LSTAT'])
```

C:\Users\User\AppData\Local\Temp\ipykernel_1800\3880656448.py:1: UserWarning:

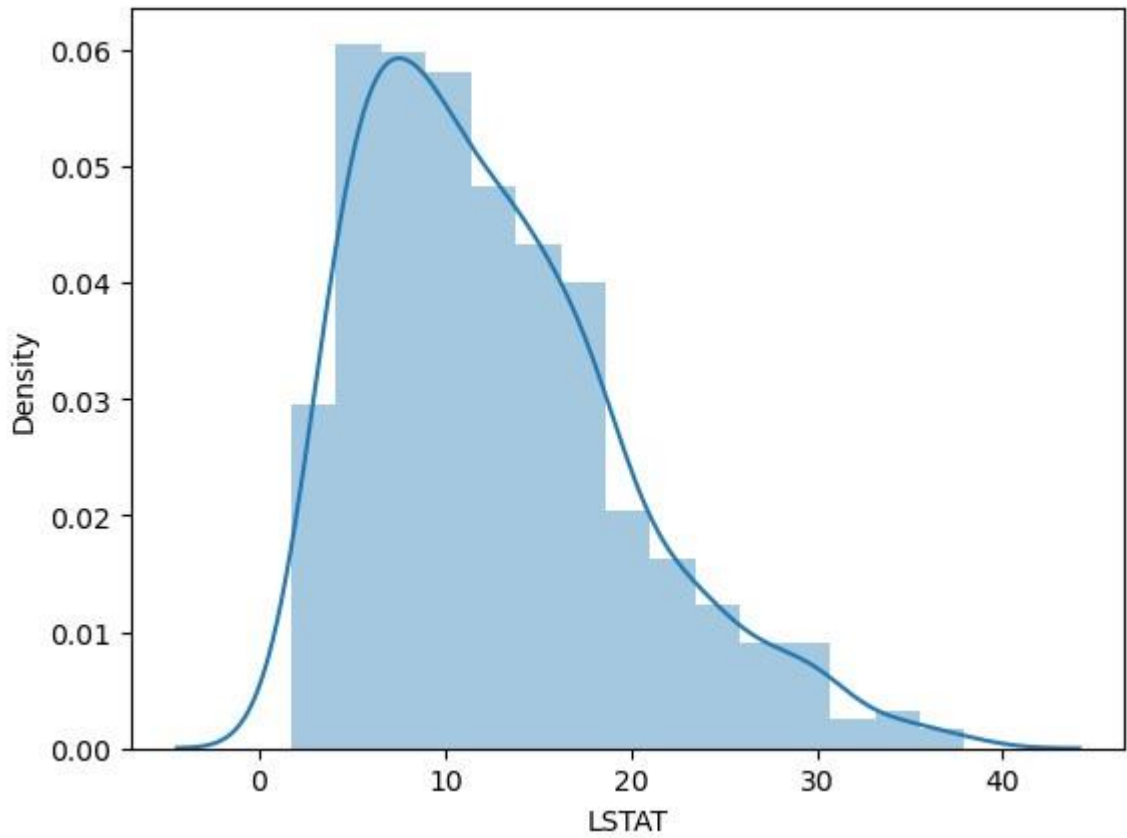
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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

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

```
sns.distplot(house_data['LSTAT'])
```

```
Out[37]: <Axes: xlabel='LSTAT', ylabel='Density'>
```



```
In [39]: X = house_data.drop('MEDV',axis = 1)
```

```
In [41]: y = house_data['MEDV']
```

```
[42]: X
```

```
Out[42]:
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LS
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	
...	
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	

In [43]: y

```
Out[43]: 0      24.0
         1      21.6
         2     34.7
         3     33.4
         4     36.2      ... 501     22.4
        502     20.6
        503     23.9
        504     22.0
        505     11.9
        Name: MEDV, Length: 506, dtype: float64
```

In [44]: `from sklearn.model_selection import train_test_split`

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_stat
```

In [45]: `from sklearn.linear_model import LinearRegression`

In [46]: `model = LinearRegression()`

[47]: `model.fit(X_train,y_train)`

Out[47]: `LinearRegression()`

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [48]: `predictions = model.predict(X_test)`

Model Evaluation

In [49]: `print(model.intercept_)`

```
31.63108403569186
```

In [50]: `print(model.coef_)`

```
[-1.33470103e-01  3.58089136e-02  4.95226452e-02  3.11983512e+00
 -1.54170609e+01  4.05719923e+00 -1.08208352e-02 -1.38599824e+00
 2.42727340e-01 -8.70223437e-03 -9.10685208e-01  1.17941159e-02
 -5.47113313e-01]
```

In

```
In [51]: coeff = pd.DataFrame(model.coef_,X.columns,columns=["Coefficient"])
```

Out[52]:

	Coefficient
--	-------------

CRIM	-0.133470
ZN	0.035809
INDUS	0.049523
CHAS	3.119835
NOX	-15.417061
RM	4.057199
AGE	-0.010821
DIS	-1.385998
RAD	0.242727
TAX	-0.008702
PTRATIO	-0.910685
B	0.011794
LSTAT	-0.547113

```
[53]: house_data.head()
```

Out[53]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTA
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	9.
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.0
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.9
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.3

```
In [55]: model.predict([[0.00632,18.0,2.31,0,0.538,6.575,65.2,4.0900,1,296.0,15.3,396.9
```

```
C:\Users\User\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X
does not have valid feature names, but LinearRegression was fitted with featu
re names
  warnings.warn(
```

Out[55]: array([30.08649576])

```
In [56]: model.predict([[0.06905,0.0,2.18,0,0.458,7.147,54.2,6.0622,3,222.0,18.7,396.90
```

```
C:\Users\User\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X
does not have valid feature names, but LinearRegression was fitted with featu
re names
  warnings.warn(
```

```
Out[56]: array([28.20837173])
```

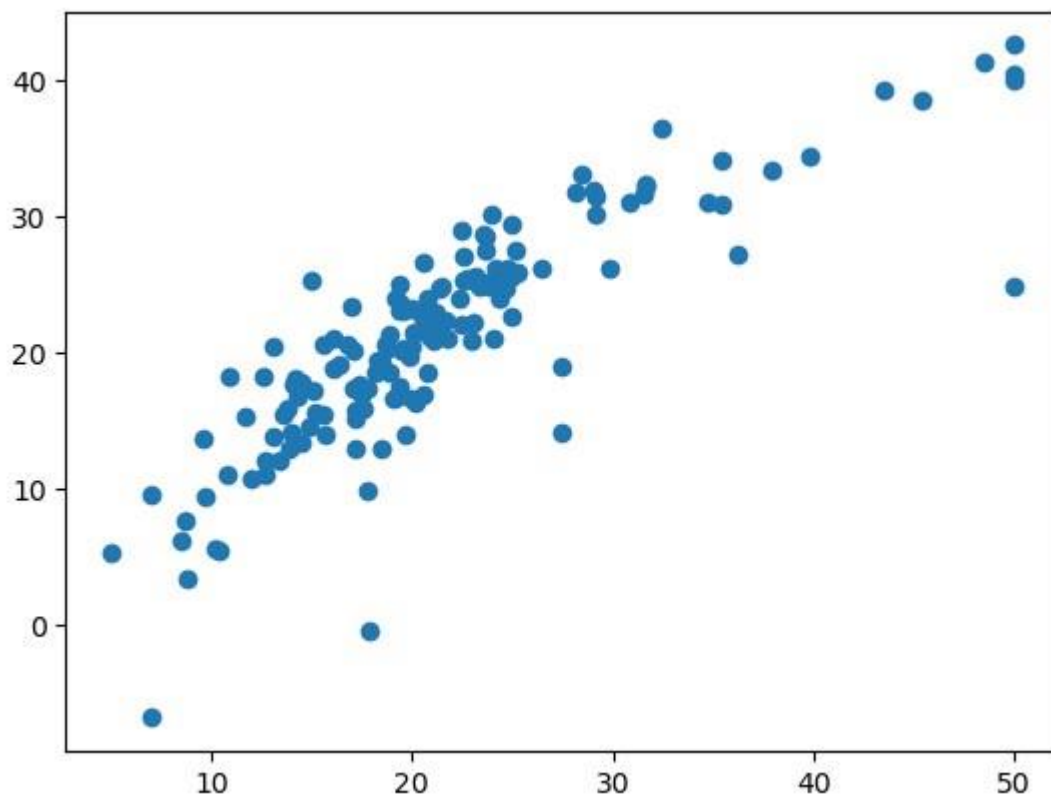
```
In [57]: model.predict([[0.06905,0.0,2.18,0,0.458,7.147,54.2,6.0622,3,222.0,18.7,396.90
```

```
C:\Users\User\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X
does not have valid feature names, but LinearRegression was fitted with featu
re names
  warnings.warn(
```

```
Out[57]: array([28.20837173])
```

```
[59]: plt.scatter(y_test,predictions)
```

```
Out[59]: <matplotlib.collections.PathCollection at 0x289f48bb0d0>
```



```
[73]: sns.distplot((y_test-predictions),kde = True,bins = 20)
```

C:\Users\User\AppData\Local\Temp\ipykernel_1800\248592448.py:1: UserWarning:

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

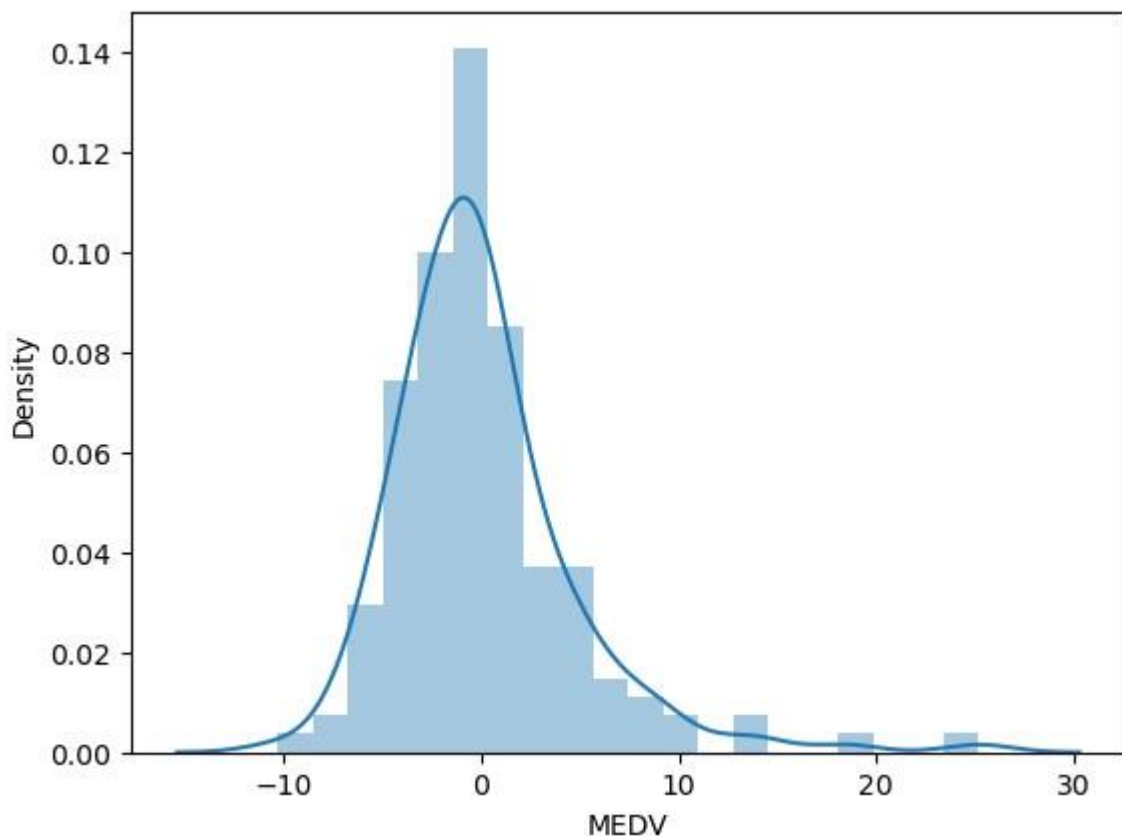
Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751> (<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

```
sns.distplot((y_test-  
predictions),kde = True,bins = 20)
```

Out[73]: <Axes: xlabel='MEDV', ylabel='Density'>



```
In [60]: from sklearn import metrics
```

```
In [72]: print('MAE : ',metrics.mean_absolute_error(y_test,predictions))  
print('MSE : ',metrics.mean_squared_error(y_test,predictions))  
print('RMSE: ',np.sqrt(metrics.mean_squared_error(y_test,predictions)))
```

MAE : 3.1627098714574253

MSE : 21.517444231177432

RMSE: 4.63868

