Import Libraries

In [1]:
```
import pandas as pd
```

In [6]:
```
bostonhousing = pd.read_csv('Boston.csv')
```

In [8]:
```
bostonhousing
```

Out[8]:

	Unnamed: 0	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0	1	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	2	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	3	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	4	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	5	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	502	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9.67	22.4
502	503	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08	20.6
503	504	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64	23.9
504	505	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48	22.0
505	506	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7.88	11.9

506 rows × 15 columns

Split dataset to X and Y variables

In [10]:
```
Y = bostonhousing.medv
Y
```

Out[10]:
```
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 [12]:
```
X = bostonhousing.drop(['medv'], axis=1)
X
```

Out[12]:

	Unnamed: 0	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat
0	1	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98
1	2	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14
2	3	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03
3	4	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94
4	5	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	502	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9.67
502	503	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08
503	504	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64
504	505	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48
505	506	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7.88

506 rows × 14 columns

Data split

In [13]:
```
# Import library

from sklearn.model_selection import train_test_split
```

Perform 80/20 Data split

In [14]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
```

Data dimension

In [15]:
```
X_train.shape, Y_train.shape
```

Out[15]:
```
((404, 14), (404,))
```

In [16]:
```
X_test.shape, Y_test.shape
```

Out[16]:
```
((102, 14), (102,))
```

Linear Regression Model

In [18]:
```
# Import library

from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
```

Build linear regression

Defines the regression model

In [19]:
```
model = linear_model.LinearRegression()
```

Build training model

In [20]:
```
model.fit(X_train, Y_train)
```

Out[20]:
```
LinearRegression()
```

Apply trained model to make prediction (on test set)

In [21]:
```
Y_pred = model.predict(X_test)
```

Prediction results

Print model performance

In [22]:
```
print('Coefficients:', model.coef_)
print('Intercept:', model.intercept_)
print('Mean squared error (MSE): %.2f'
      % mean_squared_error(Y_test, Y_pred))
print('Coefficient of determination (R^2): %.2f'
      % r2_score(Y_test, Y_pred))
```

```
Coefficients: [-2.63574341e-03 -1.16595919e-01  4.69216810e-02 -2.35131792e-02
  1.86771761e+00 -1.51058363e+01  3.74748184e+00 -5.35028136e-03
 -1.58584524e+00  3.33065038e-01 -1.15905207e-02 -9.33238295e-01
  1.04371084e-02 -5.53439776e-01]
Intercept: 36.52638473239881
Mean squared error (MSE): 18.04
Coefficient of determination (R^2): 0.74
```

String formatting

In [23]:
```
r2_score(Y_test, Y_pred)
```

Out[23]:
```
0.7413997618975603
```

In [24]:
```
r2_score(Y_test, Y_pred).dtype
```

Out[24]:
```
dtype('float64')
```

We will be using the modulo operator to format the numbers by rounding it off.

In [25]:
```
'%f' % 0.5238108335636016
```

Out[25]:
```
'0.523811'
```

In [26]:
```
'%.3f' % 0.5238108335636016
```

Out[26]:
```
'0.524'
```

We will now round it off to 2 digits

In [27]:
```
'%.2f' % 0.5238108335636016
```

Out[27]:
```
'0.52'
```

Scatter plots

In [28]:
```
# Import library

import seaborn as sns
```

In [29]:
```
# The Data

Y_test
```

Out[29]:
```
297    20.3
220    26.7
319    21.0
13     20.4
139    17.8
       ...
475    13.3
445    11.8
440    10.5
421    14.2
182    37.9
Name: medv, Length: 102, dtype: float64
```

In [30]:
```
import numpy as np
np.array(Y_test)
```

Out[30]:
```
array([20.3, 26.7, 21. , 20.4, 17.8, 17. , 15.6, 10.8, 21.7,  8.3, 32. ,
       18. , 46. ,  8.5, 11.3, 23.3, 20.6, 24.8, 13.2, 11. , 26.4, 15.6,
       24.5, 13.1, 24.4, 21. , 24.4, 41.3, 35.2, 24.7, 16.6, 20.2, 24. ,
       24.6, 30.1, 34.9, 13.1, 18.9, 23.6, 17.9, 33. , 25.2, 16.8, 21.1,
        8.8, 32.9, 15.6, 16.1, 31.1, 18.4, 23.2, 22.5,  7.2, 17.3, 50. ,
       20.8, 25. ,  7.5, 20.6, 22. , 22.8, 15.6, 19.5, 18.4, 33.1, 20.1,
       23.2, 28. , 13.8, 19.5, 14.5, 20.5, 14.5, 17.2, 17.6, 50. , 16.2,
       13.6, 21.5, 14.9, 26.2, 21.4, 22.8, 29.1, 14.8, 15.2, 19.4,  8.3,
       19.6, 15. , 24.3, 22. , 14.1, 22.6, 11.8, 21.2, 14.4, 13.3, 11.8,
       10.5, 14.2, 37.9])
```

In [31]:
```
Y_pred
```

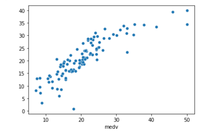
Out[31]:
```
array([18.59935136, 32.59692593, 21.16404792, 20.354801 , 16.45984152,
       22.15793225, 16.05706107, 11.46518899, 21.19663772,  9.45954931,
       33.93529891, 19.02987381, 39.46395322,  7.1791354 , 13.68163892,
       28.39149391, 27.00389682, 25.21334361,  8.77061615, 14.12403152,
       28.87238482, 16.26458202, 20.51219323, 20.62277603, 24.30832501,
       24.01401911, 22.51843377, 33.43313009, 34.53393112, 25.16529841,
       15.85993997, 22.77703804, 31.01301934, 29.66568887, 30.05586673,
       30.36026414, 14.61150457, 24.59581876, 29.53452074,  0.8731239 ,
       23.32875753, 27.39012247, 20.08851769, 20.61785418,  3.24926905,
       30.89568635, 12.58131569, 18.69916412, 32.30355745, 15.82317265,
       26.18577841, 22.5789498 ,  8.07868536, 15.6170231 , 39.92313033,
       18.59786748, 25.262397 , 12.92721025, 21.45373383, 28.0211893 ,
       28.64539803, 13.10583772, 20.1285186 , 19.96503991, 32.87744761,
       19.4022493 , 22.41223786, 28.96700682,  6.06609684, 17.56811559,
       18.46789492, 19.37031407, 14.08799999, 16.41935929, 15.84026005,
       34.46353125, 20.34331979, 12.72833708, 23.78371224, 17.89520327,
       24.07329631, 24.25626044, 27.10561578, 30.44579981, 14.96885631,
       19.52889914, 17.22633551, 13.15762302, 19.03355154, 18.95228829,
       24.70417128, 28.78735661, 17.68084657, 22.93320236,  9.14289182,
       23.96986512,  8.54557993, 15.62300708, 11.67537752, 12.83660473,
       18.44855824, 34.24534024])
```

Making the scatter plot

In [32]:
```
sns.scatterplot(Y_test, Y_pred)
```

```
C:\Users\Azlan\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel='medv'>
```

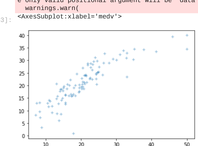
Out[32]:



In [33]:
```
sns.scatterplot(Y_test, Y_pred, marker="+")
```

```
C:\Users\Azlan\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel='medv'>
```

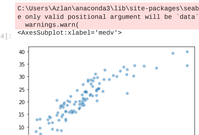
Out[33]:



In [34]:
```
sns.scatterplot(Y_test, Y_pred, alpha=0.5)
```

```
C:\Users\Azlan\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel='medv'>
```

Out[34]:



In [ ]: