In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [16]:
```
df = pd.read_csv("HousingData.csv")
df = df.dropna()
df
```

Out[16]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
499	0.17783	0.0	9.69	0.0	0.585	5.569	73.5	2.3999	6	391	19.2	395.77	15.10	17.5
500	0.22438	0.0	9.69	0.0	0.585	6.027	79.7	2.4982	6	391	19.2	396.90	14.33	16.8
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48	22.0

394 rows × 14 columns

In [17]:
```
df.columns
```

Out[17]:
```
Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
       'PTRATIO', 'B', 'LSTAT', 'MEDV'],
      dtype='object')
```

In [18]:
```
x = df[['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
        'PTRATIO', 'B', 'LSTAT']]
y = df['MEDV']
```

In [19]:
```
x
```

Out[19]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21
...	...	...	...	...	...	...	...	...	...	...	...	...	...
499	0.17783	0.0	9.69	0.0	0.585	5.569	73.5	2.3999	6	391	19.2	395.77	15.10
500	0.22438	0.0	9.69	0.0	0.585	6.027	79.7	2.4982	6	391	19.2	396.90	14.33
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48

394 rows × 13 columns

In [20]:
```
y
```

Out[20]:
```
0      24.0
1      21.6
2      34.7
3      33.4
5      28.7
       ...
499    17.5
500    16.8
502    20.6
503    23.9
504    22.0
Name: MEDV, Length: 394, dtype: float64
```

In [21]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=42)
```

In [22]:
```
model = LinearRegression()
```

In [24]:
```
model.fit(x_train, y_train)
```

Out[24]:
```
LinearRegression()
```

In [25]:
```
y_pred = model.predict(x_test)
print(y_pred)
```

```
[29.22850199 17.5037472  21.80803847 30.4942528  18.50265527 34.7432311
 22.07369779 30.7541295  33.52029866 14.82910783 22.14507214 41.26427356
 22.51242483 16.87300668 19.0023074  20.77110711 17.26921288 15.48568273
 22.69482862 14.12652975 18.19319969 20.48292217 17.04235173 29.66737037
 26.09718411 16.06132841 27.09746911 31.45656662 22.64238912 27.06681913
 41.10181897 18.21320236 23.062681   17.41202659 17.39546606 21.076803
 22.38598488 21.66963934 22.9118984  20.80249561 27.70046219 34.60121409
 22.25931447 30.83868791 35.33229137 19.76393425 24.99195034 10.39133132
 19.75538217 25.31114768 21.55934818 25.89776968 14.23212614 18.66822307
 18.34126683 24.31756905 43.40891373 22.87668508 15.3452895  23.17268086
 21.11796307 21.4821572  14.66508216 28.92300146 -3.71655126 32.60224615
 16.81628299 31.90077457 24.78013157 20.11600672 31.55246209 32.35712749
```

```
18.87519084 19.5714957  19.26577986 35.60775919 19.34796049 28.48982534
16.32963606]
```

In [ ]: