In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
#reading a csv file
df = pd.read_csv("https://raw.githubusercontent.com/YBIFoundation/Dataset/ma
```

In [3]:

```
df
```

Out[3]:

	CRIM	ZN	INDUS	CHAS	NX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
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

506 rows × 14 columns

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    int64  
 4   NX       506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    int64  
 9   TAX      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: 55.4 KB
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NX         0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
MEDV       0
dtype: int64
```

In [8]:
```
df.describe()
```

Out[8]:

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

In [13]:
```
y = df[['MEDV']]
```

In [14]:
```
y
```

Out[14]:

	MEDV
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

506 rows × 1 columns

In [16]:
```
x = df.drop("MEDV", axis = 1)
```

In [17]:
```
x
```

Out[17]:

	CRIM	ZN	INDUS	CHAS	NX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
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

506 rows × 13 columns

In [20]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.22, ra
```

In [21]:
```
X_train
```

Out[21]:

	CRIM	ZN	INDUS	CHAS	NX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
334	0.03738	0.0	5.19	0	0.515	6.310	38.5	6.4584	5	224.0	20.2	389.40
181	0.06888	0.0	2.46	0	0.488	6.144	62.2	2.5979	3	193.0	17.8	396.90
227	0.41238	0.0	6.20	0	0.504	7.163	79.9	3.2157	8	307.0	17.4	372.08
434	13.91340	0.0	18.10	0	0.713	6.208	95.0	2.2222	24	666.0	20.2	100.63
180	0.06588	0.0	2.46	0	0.488	7.765	83.3	2.7410	3	193.0	17.8	395.56
...	...	...	...	...	...	...	...	...	...	...	...	...
106	0.17120	0.0	8.56	0	0.520	5.836	91.9	2.2110	5	384.0	20.9	395.67
270	0.29916	20.0	6.96	0	0.464	5.856	42.1	4.4290	3	223.0	18.6	388.65
348	0.01501	80.0	2.01	0	0.435	6.635	29.7	8.3440	4	280.0	17.0	390.94
435	11.16040	0.0	18.10	0	0.740	6.629	94.6	2.1247	24	666.0	20.2	109.85
102	0.22876	0.0	8.56	0	0.520	6.405	85.4	2.7147	5	384.0	20.9	70.80

394 rows × 13 columns

In [22]:
```
X_test
```

Out[22]:

	CRIM	ZN	INDUS	CHAS	NX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
173	0.09178	0.0	4.05	0	0.510	6.416	84.1	2.6463	5	296.0	16.6	395.5
274	0.05644	40.0	6.41	1	0.447	6.758	32.9	4.0776	4	254.0	17.6	396.9
491	0.10574	0.0	27.74	0	0.609	5.983	98.8	1.8681	4	711.0	20.1	390.1
72	0.09164	0.0	10.81	0	0.413	6.065	7.8	5.2873	4	305.0	19.2	390.9
452	5.09017	0.0	18.10	0	0.713	6.297	91.8	2.3682	24	666.0	20.2	385.0
...	...	...	...	...	...	...	...	...	...	...	...	..
325	0.19186	0.0	7.38	0	0.493	6.431	14.7	5.4159	5	287.0	19.6	393.6
335	0.03961	0.0	5.19	0	0.515	6.037	34.5	5.9853	5	224.0	20.2	396.9
56	0.02055	85.0	0.74	0	0.410	6.383	35.7	9.1876	2	313.0	17.3	396.9
437	15.17720	0.0	18.10	0	0.740	6.152	100.0	1.9142	24	666.0	20.2	9.3
409	14.43830	0.0	18.10	0	0.597	6.852	100.0	1.4655	24	666.0	20.2	179.3

112 rows × 13 columns

In [23]:

```
y_train
```

Out[23]:

```
     MEDV
334  20.7
181  36.2
227  31.6
434  11.7
180  39.8
...   ...
106  19.5
270  21.1
348  24.5
435  13.4
102  18.6

394 rows × 1 columns
```

In [24]:

```
y_test
```

Out[24]:

```
     MEDV
173  23.6
274  32.4
491  13.6
 72  22.8
452  16.1
...   ...
325  24.6
335  21.1
 56  24.7
437   8.7
409  27.5

112 rows × 1 columns
```

In [27]:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[27]:

```
LinearRegression()
```

In [31]:

```
model.intercept_
```

Out[31]:

```
array([30.19463764])
```

```
In [29]: model.coef_

Out[29]: array([[-1.11553969e-01,  2.95152462e-02,  3.86075523e-02,
                  2.79971538e+00, -1.64618910e+01,  4.37750378e+00,
                 -7.77843388e-03, -1.43434767e+00,  2.55926441e-01,
                 -1.06030604e-02, -9.14793654e-01,  1.28877921e-02,
                 -5.15350135e-01]])
```

```
In [32]: # y = 30.194+(-1.11553969e-01)*CRIM +2.95152462e-02*ZN....................
         y_predict = model.predict(X_test)
```

```
In [33]: y_predict

Out[33]: array([[28.97529729],
                [36.06237867],
                [14.78397171],
                [25.12703189],
                [18.70716381],
                [23.17927344],
                [17.649153  ],
                [14.26888579],
                [22.86584011],
                [20.64941593],
                [24.87153295],
                [18.62998967],
                [-6.31043237],
                [21.73416181],
                [19.2665216 ],
                [26.08512913],
                [20.59744865],
                [ 5.70346772],
                [40.44438231],
                [17.55224001],
```

```
In [34]: #using  rmse
         from sklearn.metrics import mean_squared_error
```

```
In [35]:  mean_squared_error(y_test,y_predict)

Out[35]: 23.055926174103806
```

```
In [ ]:

In [ ]:

In [ ]:
```