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

In [2]:
```
import numpy as np
```

In [3]:
```
import matplotlib.pyplot as plt
```

In [4]:
```
import seaborn as sb
```

In [6]:
```
df = pd.read_csv("Boston.csv")
```

In [7]:
```
df
```

Out[7]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	BLAC
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.9
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.9
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.8
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.6
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.9
...	...	...	...	...	...	...	...	...	...	...	...	
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.9
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.9
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.9
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.4
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.9

506 rows × 14 columns

In [10]:
```
df.isnull().sum
```

Out[10]:
```
<bound method NDFrame._add_numeric_operations.<locals>.sum of       CRIM     ZN  I
NDUS   CHAS    NOX     RM    AGE    DIS    RAD    TAX
0    False  False  False  False  False  False  False  False  False  False  \
1    False  False  False  False  False  False  False  False  False  False
2    False  False  False  False  False  False  False  False  False  False
3    False  False  False  False  False  False  False  False  False  False
4    False  False  False  False  False  False  False  False  False  False
..     ...    ...    ...    ...    ...    ...    ...    ...    ...    ...
501  False  False  False  False  False  False  False  False  False  False
502  False  False  False  False  False  False  False  False  False  False
503  False  False  False  False  False  False  False  False  False  False
504  False  False  False  False  False  False  False  False  False  False
505  False  False  False  False  False  False  False  False  False  False

     PTRATIO  BLACK  LSTAT   MEDV
0      False  False  False  False
1      False  False  False  False
2      False  False  False  False
3      False  False  False  False
4      False  False  False  False
..       ...    ...    ...    ...
501    False  False  False  False
502    False  False  False  False
503    False  False  False  False
504    False  False  False  False
505    False  False  False  False

[506 rows x 14 columns]>
```

In [14]:
```
target_variables = "MEDV"
```

In [15]:
```
y = df[target_variables]
```

In [17]:
```
x = df.drop(target_variables, axis=1)
```

In [21]:
```
x.head()
```

Out[21]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	BLACK
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

In [22]:
```
y.head()
```

Out[22]:
```
0    24.0
1    21.6
2    34.7
3    33.4
4    36.2
Name: MEDV, dtype: float64
```

In [23]:
```
from sklearn.model_selection import train_test_split
```

In [26]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size = 0.2,random_state = 2
```

In [27]:
```
from sklearn.linear_model import LinearRegression
```

In [28]:
```
regression = LinearRegression()
```

In [29]:
```
regression.fit(x_train,y_train)
```

Out[29]:
```
▾ LinearRegression

LinearRegression()
```

In [31]:
```
train_score = round(regression.score(x_train,y_train)*100,2)
```

In [34]:
```
train_score
```

Out[34]:
```
72.86
```

In [42]:
```
y_pred = regression.predict(x_test)
```

In [43]:
```
y_pred
```

Out[43]:
```
array([23.01506153, 21.2115869 , 33.71590384, 31.56542369,  3.1826268 ,
        3.15381954, 27.40305304, 22.2126176 , 14.86506114, 21.34105453,
       30.95942941, 26.70065029, 21.12624382, 18.37282564, 17.64315354,
       25.38194186, 24.42970445, 13.36957057,  8.66686786, 18.57490534,
       21.73966467, 20.34270529, 36.5461105 , 20.59627495, 19.87979627,
       15.75766967, 37.11632999, 34.85897895, 30.83458635, 23.23441285,
       18.68278505, 20.749546  , 31.84560076, 30.20214207, 13.3861702 ,
       15.87078398, 13.70766096, 23.74163998, 25.95135088, 23.18325878,
       28.99906539, 12.50341936, 31.08347911,  6.39401895, 23.71801218,
       20.61523929, 33.15362417, 19.21862493, 35.89603081,  0.82365329,
       31.90288611, 31.69640543,  6.58849712, 34.62762996, 20.41162545,
       19.69277608, 19.53445865, 18.58689088, 15.81420496, 22.98764309,
       19.65947045, 16.36377019, 18.48783369, 32.76568172, 35.49022568,
       24.58349631, 41.5854766 , 32.94818456, 14.60990256, 27.43178268,
        8.04470074,  5.61185652, 22.21428332, 18.72817007, 31.02824788,
       26.04494485, 24.60357003, 24.84231113, 25.38796252, 24.87762205,
       33.71343923, 19.72606026, 20.60046055, 27.82692882, 38.0055624 ,
       37.24265207, 22.16841364, 29.6160177 , 31.07303315, 17.93399181,
       20.87524555, 19.48170453, 18.61409692, 37.13055111, 39.81659125,
        9.1811861 , 35.30202671, 30.28664671, 21.0820992 , 13.65467682,
       31.38696603, 24.99174874])
```

In [44]:
```
from sklearn.metrics import r2_score
```

In [45]:
```
score = round(r2_score(y_test,y_pred)*100,2)
```

In [47]:
```
print("r2 score is ",score)
```

```
r2 score is  77.89
```

In [48]:
```
round(regression.score(x_test,y_test)*100,2)
```

Out[48]:
```
77.89
```

In [49]:
```
from sklearn import metrics
```

In [51]:
```
print("mean absolute error :",metrics.mean_absolute_error(y_test,y_pred))
print("mean squared absolute error :",metrics.mean_squared_error(y_test,y_pred))
print("root mean squared absolute error :", np.sqrt(metrics.mean_squared_error(y_te
```

```
mean absolute error : 3.1130437468934313
mean squared absolute error : 18.495420122448426
root mean squared absolute error : 4.300630200615768
```

In [52]:
```
df1 = pd.DataFrame({"Actual":y_test,"Predicted": y_pred,"Variance":y_test-y_pred})
```

In [53]:
```
df1.head()
```

Out[53]:

	Actual	Predicted	Variance
463	20.2	23.015062	-2.815062
152	15.3	21.211587	-5.911587
291	37.3	33.715904	3.584096
183	32.5	31.565424	0.934576
384	8.8	3.182627	5.617373

In [54]:
```
df1.head(14)
```

Out[54]:

	Actual	Predicted	Variance
463	20.2	23.015062	-2.815062
152	15.3	21.211587	-5.911587
291	37.3	33.715904	3.584096
183	32.5	31.565424	0.934576
384	8.8	3.182627	5.617373
141	14.4	3.153820	11.246180
240	22.0	27.403053	-5.403053
349	26.6	22.212618	4.387382
410	15.0	14.865061	0.134939
150	21.5	21.341055	0.158945
175	29.4	30.959429	-1.559429
289	24.8	26.700650	-1.900650
67	22.0	21.126244	0.873756
310	16.1	18.372826	-2.272826

In [ ]: