```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv("Boston.csv")
```

```
df
```

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

```
df.isna().sum
```

```
<bound method NDFrame._add_numeric_operations.<locals>.sum of      Unnamed: 0   crim     zn  indus    chas     nox      rm     age  
dis  \
0         False  False  False  False  False  False  False  False  False
1         False  False  False  False  False  False  False  False  False
2         False  False  False  False  False  False  False  False  False
3         False  False  False  False  False  False  False  False  False
4         False  False  False  False  False  False  False  False  False
..          ...    ...    ...    ...    ...    ...    ...    ...    ...
501       False  False  False  False  False  False  False  False  False
502       False  False  False  False  False  False  False  False  False
503       False  False  False  False  False  False  False  False  False
504       False  False  False  False  False  False  False  False  False
505       False  False  False  False  False  False  False  False  False

       rad    tax  ptratio  black  lstat   medv
0    False  False    False  False  False  False
1    False  False    False  False  False  False
2    False  False    False  False  False  False
3    False  False    False  False  False  False
4    False  False    False  False  False  False
..     ...    ...      ...    ...    ...    ...
501  False  False    False  False  False  False
502  False  False    False  False  False  False
503  False  False    False  False  False  False
504  False  False    False  False  False  False
505  False  False    False  False  False  False

[506 rows x 15 columns]>
```

```
target_variables = "medv"
```

```
y = df[target_variables]
```

```
x = df.drop(target_variables, axis=1)
```

```
x.head()
```

	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

```
y.head()
```

```
0    24.0
1    21.6
2    34.7
3    33.4
4    36.2
Name: medv, dtype: float64
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2,random_state=2)
```

```
from sklearn.linear_model import LinearRegression
```

```
regression = LinearRegression()
```

```
regression.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
train_score=round(regression.score(x_train,y_train)*100,2)
```

```
print('Train score of linear regression',train_score)
```

```
Train score of linear regression 72.91
```

```
y_pred = regression.predict(x_test)
```

```
from sklearn.metrics import r2_score
score=round(r2_score(y_test,y_pred)*100,2)
print('r_2 score',score)
```

```
r_2 score 78.1
```

```
round(regression.score(x_test,y_test)*100,2)
```

```
78.1
```

```
from sklearn import metrics
print("mean absolute error on test data of linear regression",metrics.mean_absolute_error(y_test,y_pred))
print("mean squared error on test data of linear regression",metrics.mean_squared_error(y_test,y_pred))
print("root mean squared error on test data of linear regression",np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

```
mean absolute error on test data of linear regression 3.0812603233002447
mean squared error on test data of linear regression 18.321720821929564
root mean squared error on test data of linear regression 4.280387928906627
```

```
df1=pd.DataFrame({'Actual':y_test,'Predicted':y_pred,'Variance':y_test-y_pred})
```

```
df1.head()
```

	Actual	Predicted	Variance
463	20.2	22.935008	-2.735008
152	15.3	21.334270	-6.034270
291	37.3	33.643417	3.656583
183	32.5	31.381211	1.118789
384	8.8	3.218861	5.581139

```
df.head(14)
```

	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
5	6	0.02985	0.0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
6	7	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9
7	8	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.90	19.15	27.1
8	9	0.21124	12.5	7.87	0	0.524	5.631	100.0	6.0821	5	311	15.2	386.63	29.93	16.5
9	10	0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.10	18.9
10	11	0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15.0
11	12	0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.90	13.27	18.9
12	13	0.09378	12.5	7.87	0	0.524	5.889	39.0	5.4509	5	311	15.2	390.50	15.71	21.7