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

```
Boston = pd.read_csv("https://raw.githubusercontent.com/YBI-Foundation/Dataset/main/Boston.csv")
print(Boston.head())
```

```
      CRIM    ZN  INDUS  CHAS     NX     RM   AGE     DIS  RAD    TAX  \
0  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296.0
1  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242.0
2  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242.0
3  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222.0
4  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222.0

   PTRATIO       B  LSTAT  MEDV
8     15.3  396.98   4.98  24.8
1     17.8  396.98   9.14  21.6
2     17.8  392.83   4.03  34.7
3     18.7  394.63   2.94  33.4
4     18.7  396.96   5.33  36.2
```

```
data = pd.DataFrame(Boston)
```

```
data.head()
```

	CRIM	ZN	INDUS	CGAS	NX	RM	AGE	DIS	RAD	TAE	PTRATIO	
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	
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	
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	

```
data.isnull( ).sum()
```

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

```
print(data.info())
print(data.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
Rangelndex:  506 entries,  0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype

 0   CRIM     506 non-null    float64
 1   ZN       S06 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 nou-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(t2), int64(2)
memory usage: 55.5 KB
None
```

	CRIH	ZN	INDUS	CHAS	NX	RM	\
count	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	
std	8.68154S	23.322453	6.860353	0.253994	0.115878	0.702617	
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	
max	88.976200	100.000000	27.740000	1.000B00	0.871000	8.780000	

	AGE	DIS	RAD	TAX	PTRATIO	B	\
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	
mean	68.574901	3.795643	9.549407	408.237154	18.4SS534	356.674032	
std	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864	
min	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	
25%	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500	
50%	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000	
75%	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000	
max	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000	

	LSTAT	MEDV
count	506.000000	506.000000
mean	12.653063	22.532806
std	7.141062	9.1971B4
min	1.730000	5.000000
25%	6.950000	17.025000
50%	11.360000	21.200000
75%	16.955000	25.000000
max	37.970000	50.000000

```
print(type(data))
print('\n')
print(data.keys())
print('\n')
print(data.shape)
print('\n')
```

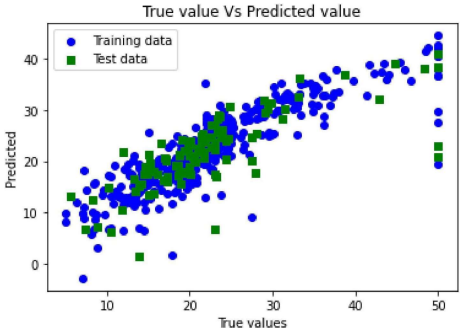
```
<class 'paNdas.core.frame.DataFrame'>


Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
       'PTRATIO', 'B', 'LSTAT', 'NEDV'],
      dtype='object')


(506, 14)
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state = 0)

LM  = LinearRegression()
model=LM.fit(x_train,y_train)

y_train_pred = LM.predict(x_train)
y_test_pred = LM.predict(x_test)

df = pd.DataFrame(y_train_pred,y_train)
df = pd.DataFrame(y_test_pred,y_test)

from sklearn.metrics import mean_squared_error, r2_score
MS = mean_squared_error(y_test,y_test_pred)
print(MS)
```

```
MS = mean squared error(y train pred, y train)
print(MS)
```

```
34 . 48 937 39647 3988
2 B . 3648 B6 28 B49 7234
```

```
plt.scatter(y train,y train pred, c=”blue" , marker='o', label= 'Training data')
plt.scatter(y test,y_test pred, c='green', marI‹er='s', label='Test data')
pit.xlabel('True values')
pit.ylabel'Predicted')
pit.title('True value Us Predicted value')
plt.legend(loc-'upper left')
pit.plot()
plt.show()
```



0s completed at 7:38 PM