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


data = pd.DataFrame(boston.data)


data.columns = boston.feature_names
data.head()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90

```
data['PRICE'] = boston.target


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

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

```
#Split dependent variable and independent variables
x = data.drop(['PRICE'], axis = 1)
y = data['PRICE']


#splitting data to training and testing dataset.
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest =train_test_split(x, y, test_size =0.2,random_state = 0)
```

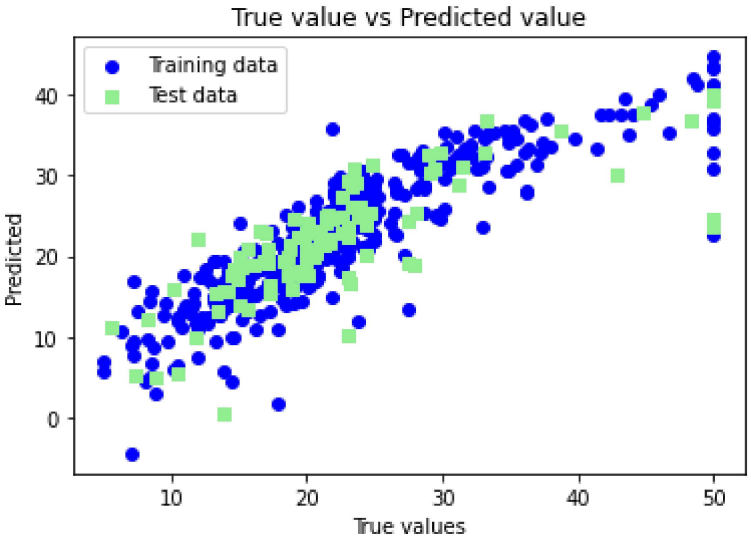
```
#Use linear regression( Train the Machine ) to Create Model
import sklearn
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
model=lm.fit(xtrain, ytrain)


#Predict the y_pred for all values of train_x and test_x
ytrain_pred = lm.predict(xtrain)
ytest_pred = lm.predict(xtest)


#Evaluate the performance of Model for train_y and test_y
df=pd.DataFrame(ytrain_pred,ytrain)
df=pd.DataFrame(ytest_pred,ytest)


#Calculate Mean Square error for train_y and test_y
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(ytest, ytest_pred)
print(mse)
mse1 = mean_squared_error(ytrain_pred,ytrain)
print(mse1)
```

```
33.44897999767638
19.32647020358573
```

```
from sklearn.metrics import mean_squared_error
#def linear_metrics():
r2 = lm.score(xtest, ytest)
rmse = (np.sqrt(mean_squared_error(ytest, ytest_pred)))
print('r-squared: {}'.format(r2))
print('---------------------------------------')
print('root mean squared error: {}'.format(rmse))
```

```
r-squared: 0.5892223849182526
---------------------------------------
root mean squared error: 5.783509315085122
```

```
#Plotting the linear regression model
plt.scatter(ytrain ,ytrain_pred,c='blue',marker='o',label='Training data')
plt.scatter(ytest,ytest_pred ,c='lightgreen',marker='s',label='Test data')
plt.xlabel('True values')
plt.ylabel('Predicted')
plt.title("True value vs Predicted value")
plt.legend(loc= 'upper left') #plt.hlines(y=0,xmin=0,xmax=50)
plt.plot()
plt.show()
```


True value vs Predicted value
Training data
Test data
Predicted
True values
0
10
20
30
40
10
20
30
40
50