In [ ]:

```
#************************************************************************************
#                      Assignment : Machine Learning 2-Linear Regression
#
#************************************************************************************

# Problem Statement :  To Build Regression Model using Sklearn to predict price based on other dependent
#                   :  variables


# Step 0 :importing Librabries
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import sklearn
from sklearn.datasets import load_boston

# Step1 : Loading the Data from the dataset , setting all the columns and column names

boston = load_boston()

# understanding the characteristics of dataset
# printing the boston Dataset contents
print(" printing the Boston Data Contents \n",'-'*80)
print(boston.keys())

# Understanding the Data from the DESCR and identifying the dependent variables
print(" The Attributes , columns of the DataSet and Meaning of these attributes \n",'-'*80)
print(boston.feature_names)
print(boston.DESCR)


# Loading the data in the DataFrame
bos = pd.DataFrame(boston.data)
bos.columns = boston.feature_names
bos['PRICE'] = boston.target


print(" Concatenating the Target price field to DataFrame \n",'-'*80)
print(bos.head())

```

In [ ]:

```
# Step2 : Analysing the Data

# To find the number of rows and columns in the dataset

print(" Analysing the Data , by seeing at its shape, information,describe, functions \n",'-'*80)
print(bos.shape)
print("-"*80)


print(bos.info())
print("-"*80)

print(bos.describe())
print("-"*80)

print(" Checking if any of the features have null value  \n",'-'*80)
# To check if any of the fields/features have null data
print(bos.isnull().sum())
print("-"*80)

```

In [ ]:

```
# Step3: Data Visualisation for better understanding of the data


# importing seaborn and matplotlib for visualising the Data
import seaborn as sns
import matplotlib.pyplot as plt

# allow plots to appear within the notebook
%matplotlib inline


# plotting the pair plot for all the features with respect to Price , to get an idea of dependency of variables
features = bos.columns
count = len(features)

i = 0
while i < (count-2):
    sns.pairplot(bos,x_vars= features[i:i+4] ,y_vars = 'PRICE', aspect=1,kind='reg')
    i = i + 4

sns.pairplot(bos,x_vars= features[i:] ,y_vars = 'PRICE', aspect=1,kind='reg')

#sns.pairplot(bos,x_vars=['CRIM','ZN','INDUS','CHAS','NOX','RM','AGE','DIS','RAD','TAX','PTRATIO','B','LSTAT'] ,y_var

# WE observe that RM( Average Rooms per House) has a linear relation and dependent on the price , hence we will use R
# the dependent or predictor variable
print(" We observe from the pairplot , that Price is linearly dependent on the RM feature than others\n")



```

In [ ]:

```
# Step 4 : Training Model using by creating training and test sets

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

# Using RM as predictor and PRICE as Target
#features = ['RM','AGE','DIS','B']
features = ['RM','AGE','DIS','B','CRIM','ZN','INDUS','CHAS','NOX','RAD','TAX','PTRATIO','LSTAT']
#
#,'CHAS','NOX'
X = bos[features]
Y = bos.PRICE

# reshaping the training and Test Set for getting 2 D array as having only 1 D
print(X.shape)
print(Y.shape)

Y = Y.values.reshape(-1,1)

# splitting the Data into training and Test Set , 25 percent is Testing Data
x_train, x_test, y_train,y_test = train_test_split(X,Y,test_size=0.25,random_state=1234)

print(" The shape of the train and test sets after splitting \n",'-'*80)
print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)



# creating an instance of Linear Regression
lm = LinearRegression()

# fitting / Training data with the training set data
lm.fit(x_train,y_train)

# Predicting the prices for the test set
pred_test = lm.predict(x_test) # test case prediction

print(" Following are the predicted price values for the test set , after fitting the model with train set\n",'-'*80)
print(pred_test)

```

In [ ]:

```
# Step 5: Visualisation of the observed and predicted Data


print(y_test.shape)
print(pred_test.shape)
plt.scatter(y_test, pred_test)
plt.xlabel("Observed Price ")
plt.ylabel(" Predicted House Price ")
plt.title("Observed Vs Predicted Price Graph")
```

In [ ]:

```
#Step 6 : Model Evaluation by finding the error and accuracy of prediction

# Step Evvaluation of Data by finding the MSE and Rsquare
from sklearn.metrics import mean_squared_error,r2_score

MSE = np.sqrt(mean_squared_error(y_test,pred_test))
print("The Root Mean squared error for Price of house is : %.4f"%MSE)

R2square = r2_score(y_test,pred_test)
print("The R2 score for Model for Price of House : %.4f"%R2square)

acc = round(R2square,2)*100

print("This Model can predict the price of House at %.2f percent accuracy "%acc)
```