HERE ARE THE LINKS TO READ DOCUMENTATION OF USED LIBRARIES :

```
SKLEARN : http://scikit-learn.org/stable/documentation.html
NUMPY :   https://docs.scipy.org/doc/numpy-1.15.0/
PANDAS :  http://pandas.pydata.org/pandas-docs/version/0.23/
LinearRegression DOCUMENTATION : http://scikit-
learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
```

In [1]:

```
#IMPORTTED LIBRARIES FOR ANALYSING A DATASET WITH AN ML MODEL
from sklearn import datasets , model_selection , linear_model

#DATASETS - CONTAIN DATASETS FOR ANALYSIS
#MODEL SELECTION - USED FOR SPLITTING DATA INTO TRAINING AND TESTING
#LINEAR MODEL - CONTAINS ML ALGOS LIKE LINEAR REGRESSION , LOGISITC REGRESSION
```

In [2]:

```
#LOADING BOSTON DATASET FROM SKLEARN DATASETS

df = datasets.load_boston()
```

In [3]:

```
#DISPLAYING THE X AND Y IN TABULAR FORMAT

import pandas as pd
X_val , Y_val = pd.DataFrame(df.data) , pd.DataFrame(df.target)
print(X_val.head())
print(Y_val.head())
```

```
        0     1     2    3      4      5     6       7    8      9    10  \
0  0.00632  18.0  2.31  0.0  0.538  6.575  65.2  4.0900  1.0  296.0  15.3
1  0.02731   0.0  7.07  0.0  0.469  6.421  78.9  4.9671  2.0  242.0  17.8
2  0.02729   0.0  7.07  0.0  0.469  7.185  61.1  4.9671  2.0  242.0  17.8
3  0.03237   0.0  2.18  0.0  0.458  6.998  45.8  6.0622  3.0  222.0  18.7
4  0.06905   0.0  2.18  0.0  0.458  7.147  54.2  6.0622  3.0  222.0  18.7

       11    12
0  396.90  4.98
1  396.90  9.14
2  392.83  4.03
3  394.63  2.94
4  396.90  5.33
      0
0  24.0
1  21.6
2  34.7
3  33.4
4  36.2
```

In [4]:

```
#SPLITTED THE DATA INTO TRAINING AND TESTING
#df.data CONTAINS X VALUES
#df.target CONTAINS THE RESULT

X_train , X_test , Y_train ,Y_test = model_selection.train_test_split(df.data , df.target , random_
state = 1)
alg = linear_model.LinearRegression()
```

In [5]:

```
#FIT THE MODEL WITH TRAINING AND TESTING

alg.fit(X_train , Y_train)
```

Out[5]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [6]:

```
#GIVES THE SCORE OF THE APPLIED MODEL
alg.score(X_test , Y_test)
```

Out[6]:

```
0.7790257749137307
```

In [7]:

```
#REGRESSION COEFFICECIENTS
alg.coef_
```

Out[7]:

```
array([-1.13256952e-01,  5.70869807e-02,  3.87621062e-02,  2.43279795e+00,
       -2.12706290e+01,  2.86930027e+00,  7.02105327e-03, -1.47118312e+00,
        3.05187368e-01, -1.06649888e-02, -9.97404179e-01,  6.39833822e-03,
       -5.58425480e-01])
```

In [8]:

```
#INTERCEPT FOR THE MODEL
alg.intercept_
```

Out[8]:

```
45.23641584605663
```

In [9]:

```
#PREDICTED VALUES
y_pred = alg.predict(X_test)
y_pred
```

Out[9]:

```
array([32.37355169, 27.95629215, 18.07265446, 21.63752354, 18.92899165,
       19.96544181, 32.28164239, 18.06690441, 24.73681562, 26.85560915,
       27.23448864, 28.56695646, 21.19027273, 26.94926544, 23.38373688,
       20.89466993, 17.10735967, 37.72703514, 30.52697416,  8.43947453,
       20.87218975, 16.19528394, 25.13903989, 24.77783176, 31.41047378,
       10.97998688, 13.79742537, 16.80597997, 35.94111289, 14.71326657,
       21.23984248, 14.1500526 , 42.7175483 , 17.83367283, 21.8471085 ,
       20.39714439, 17.4827441 , 26.99954163,  9.82567275, 20.00045231,
       24.27086068, 21.06464573, 29.47336641, 16.46410291, 19.38632792,
       14.49124029, 39.41204737, 18.10233655, 26.21931924, 20.56700238,
       25.08766866, 24.48271997, 25.02751143, 26.85092251,  5.00787149,
       24.12962926, 10.7060661 , 26.83809421, 16.79966861, 35.47116284,
       19.49834974, 27.43500479, 16.57517584, 19.11046881, 10.97829442,
       32.04938666, 36.3187731 , 21.86383561, 24.82654636, 25.34497518,
       23.36898797,  6.99865029, 16.82926036, 20.2651223 , 20.74578444,
       21.85863245, 34.18467273, 27.95220552, 24.86087363, 34.4298983 ,
       18.6153324 , 24.02883897, 34.45483147, 13.27505837, 20.71858774,
       30.1615014 , 17.04490511, 24.20028896, 19.17968202, 16.97964859,
       26.80839937, 41.01402058, 14.45369995, 23.27510914, 14.92319964,
       21.93677781, 22.81592892, 29.16975402, 36.69413595, 20.40886123,
       17.82738237, 17.49005237, 25.07511894, 21.98189934,  8.28498632,
       21.52508548, 16.46257248, 33.01600715, 24.49465929, 25.08278997,
       38.29472881, 28.93626329, 14.80682003, 34.73501879, 35.49658763,
       32.89764656, 20.97873322, 16.66612666, 34.24466124, 39.00016305,
       21.57473927, 15.65632993, 27.32990822, 18.71715952, 27.27015969,
       21.16178312, 26.01358024])
```

In [10]:

```
#CALCULATING SCORE THROUGH FORMULA
1 - ((y_pred - Y_test)**2).sum()/((Y_test - Y_test.mean())**2).sum()
```

Out[10]:

```
0.7790257749137307
```