Practical No. 04

by Kartik Deshpande

Data Analytics I Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (https://www.kaggle.com/c/boston-housing). The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 feature variables in this dataset. The objective is to predict the value of prices of the house using the given features.

In [24]:

```
import pandas as pd
import numpy as np

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

In [51]:

```
df = pd.read_csv("/home/kartik/Documents/Python Notebooks/BostonHousing.csv")
df = df.dropna()
df.head()
```

Out[51]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
0	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	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	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	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	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

In [52]:

```
df.columns
```

Out[52]:

```
Index(['crim', 'zn', 'indus', 'chas', 'nox', 'rm', 'age', 'dis', 'rad', 'tax',
       'ptratio', 'b', 'lstat', 'medv'],
      dtype='object')
```

In [53]:

```
x = df[['crim', 'zn', 'indus', 'chas', 'nox', 'rm', 'age', 'dis', 'rad', 'tax',
       'ptratio', 'b', 'lstat']]
x.head()
```

Out[53]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33

In [54]:

```
y = df['medv']
y.head()
```

```
Out[54]:  0    24.0
          1    21.6
          2    34.7
          3    33.4
          4    36.2
          Name: medv, dtype: float64
```

```
In [55]:  x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=42)
```

```
In [56]:  model = LinearRegression()
          model.fit(x_train,y_train)
```

```
Out[56]:  ▾ LinearRegression
          LinearRegression()
```

```
In [59]:  y_pred = model.predict(x_test)
          y_pred
```

```
Out[59]:  array([10.82520289, 22.97716771, 15.45617932, 33.55363131, 22.96357871,
                 11.52151263, 12.76018157, 19.74412591, 21.33180568, 11.7372368 ,
                 18.75187948, 30.04070255, -0.73011025, 25.78030298,  3.02335542,
                  8.49359394, 24.07065874, 18.57018302, 25.24003893, -6.24945751,
                 13.33486252, 19.08911255, 27.0053246 , 19.59024598, 22.40273032,
                 16.47206196, 28.79995249, 26.24334357, 18.42194929, 21.27338464,
                 20.62838908, 30.49181729, 17.87807473, 31.53661897, 31.16125663,
                 22.20316674,  7.79878712, 23.70737642,  8.54510946, 25.0261323 ,
                 12.99764774, 36.12050346, 14.45054578, 30.51121076, 13.02756177,
                 28.48505695, 30.34475695, 20.15771804, 18.46362559, 13.69183882,
                 24.00613417, 32.99780499, 16.4544118 , 11.66937979, 34.39689874,
                 33.37924364, 17.77929903, 18.70970757, 16.25656178, 27.35347057,
                 20.48252629, 40.60322048, 20.53694472,  8.20383246, 25.97767891,
                 27.81783878, 12.08008232,  7.62795819, 27.14868012, 16.44871208,
                 23.46295285, 14.63324084, 40.28319824, 28.66936219, 23.1422757 ,
                 23.95467347, 35.49409707, 24.49032705, 20.75456047, 15.97157605,
                 27.18392572, 27.90827964, 21.23340735, 29.37584949, 23.9104647 ,
                 29.29067164, 24.22591482, 20.08729338, 18.20901184, 44.2614741 ,
                  4.63790216, 19.31301769, 17.2763475 , 23.72223401,  7.38111706,
                 17.02032604, 31.01206401, 21.14872276, 10.9653362 , 20.85193641,
                 24.18859543, 17.31441353, 12.19815419, 19.11197493, 19.50296116,
                 22.01189876, 35.62919117, 31.55632051, 20.24630891, 20.2365227 ,
                 14.26461161, 11.71171865, 21.66497318, 15.74320729, 20.29084409,
                 19.52326714, 25.23052357, 23.83851879, 23.14474265, 22.78985166,
                 40.21608485, 27.45423907, 24.8064738 , 30.06864408, 30.07124307,
                 38.69282771])
```

```
In [61]:  model.score(x_train,y_train)
```

```
Out[61]:  0.7335900413194543
```

```
In [62]:  model.score(x_test,y_test)
```

```
Out[62]:  0.7459403901980342
```

```
In [63]:  np.sqrt(mean_squared_error(y_test,y_pred))
```

```
Out[63]:  4.387285229095364
```

```
In [ ]:
```