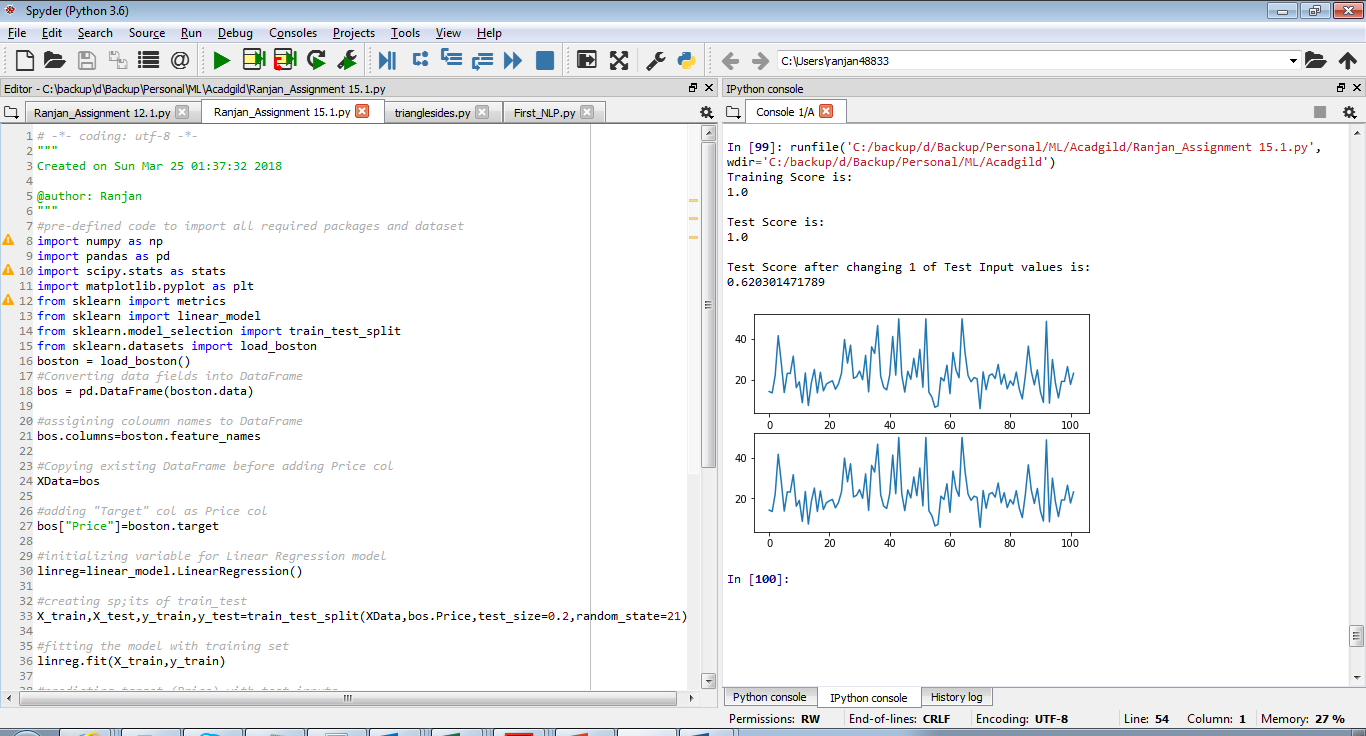
**Response for Assignment 15.1**



**Summary**

1. Above screenshot shows Training Score and Test Score as 100%
2. It also shows 2 plot charts of y\_test (test data of labels) and y\_pred (predicted data)
   1. **These charts are exactly the same**
3. After changing a value within the label test data after the model is created, test accuracy becomes 62.03%

**Below is the source code**

#pre-defined code to import all required packages and dataset

import numpy as np

import pandas as pd

import scipy.stats as stats

import matplotlib.pyplot as plt

from sklearn import metrics

from sklearn import linear\_model

from sklearn.model\_selection import train\_test\_split

from sklearn.datasets import load\_boston

boston = load\_boston()

#Converting data fields into DataFrame

bos = pd.DataFrame(boston.data)

#assigining coloumn names to DataFrame

bos.columns=boston.feature\_names

#Copying existing DataFrame before adding Price col

XData=bos

#adding "Target" col as Price col

bos["Price"]=boston.target

#initializing variable for Linear Regression model

linreg=linear\_model.LinearRegression()

#creating sp;its of train\_test

X\_train,X\_test,y\_train,y\_test=train\_test\_split(XData,bos.Price,test\_size=0.2,random\_state=21)

#fitting the model with training set

linreg.fit(X\_train,y\_train)

#predicting target (Price) with test inputs

y\_pred=linreg.predict(X\_test)

#printing training accuracy score

print("Training Score is: ")

print(linreg.score(X\_train,y\_train))

#printing test accuracy score

print("\nTest Score is: ")

print(linreg.score(X\_test,y\_test))

# plotting test label data and predicted data

plt.subplot(2,1,1)

plt.plot(y\_test.values)

plt.subplot(2,1,2)

plt.plot(y\_pred)

#updating random value of test data

y\_test.update(pd.Series([100],index=[321]))

#printing test accuracy score

print("\nTest Score after changing 1 of Test Input values is: ")

print(linreg.score(X\_test,y\_test))