**MACHINE LEARNING PROJECT**

In this project, I have used Linear Regression Machine Learning model for the boston housing dataset from Kaggle website

**PROBLEM STATEMENT:**

Each record in the database describes a Boston suburb or town. The data was drawn from the Boston Standard metropolitan Statistical Area(SMSA)in 1970. The attributes are defined as follows (taken from the UCI Machine Learning Repository1): CRIM: per capita crime rate by town

**ML METHODOLOGY:**

Linear Regression is the methodology used for training and testing the dataset.

Linear Regression is a method of modeling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationship between variables. Linear Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.

**DATASET DESCRIPTION:**

Some relevant columns in the dataset

* ZN: proportion of residential land zoned for lots over 25,000 sq .ft.
* INDUS: proportion of non-retail business acres per town
* CHAS: Charles River dummy variable(= 1 if tract bounds river; 0 otherwise)
* NOX: nitric oxides concentration(parts per 10 million)
* RM: average number of rooms per dwelling
* AGE: proportion of owner-occupied units build prior to 1940
* RAD: index of accessibility to radial high ways
* TAX: full-value property-tax rate per $10,000
* PTRATIO: pupil- teacher ratio by town 12. B: 1000(Bk-0.63)2 where Bk is the proportion of blacks by town 13.LSTAT:% lower status of the population
* MEDV: median value of owner-occupied homes in $1000s we can see that the input attributes have a mixture of units.

**PRE\_PROCESSING:**

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.

%matplotlib inline

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

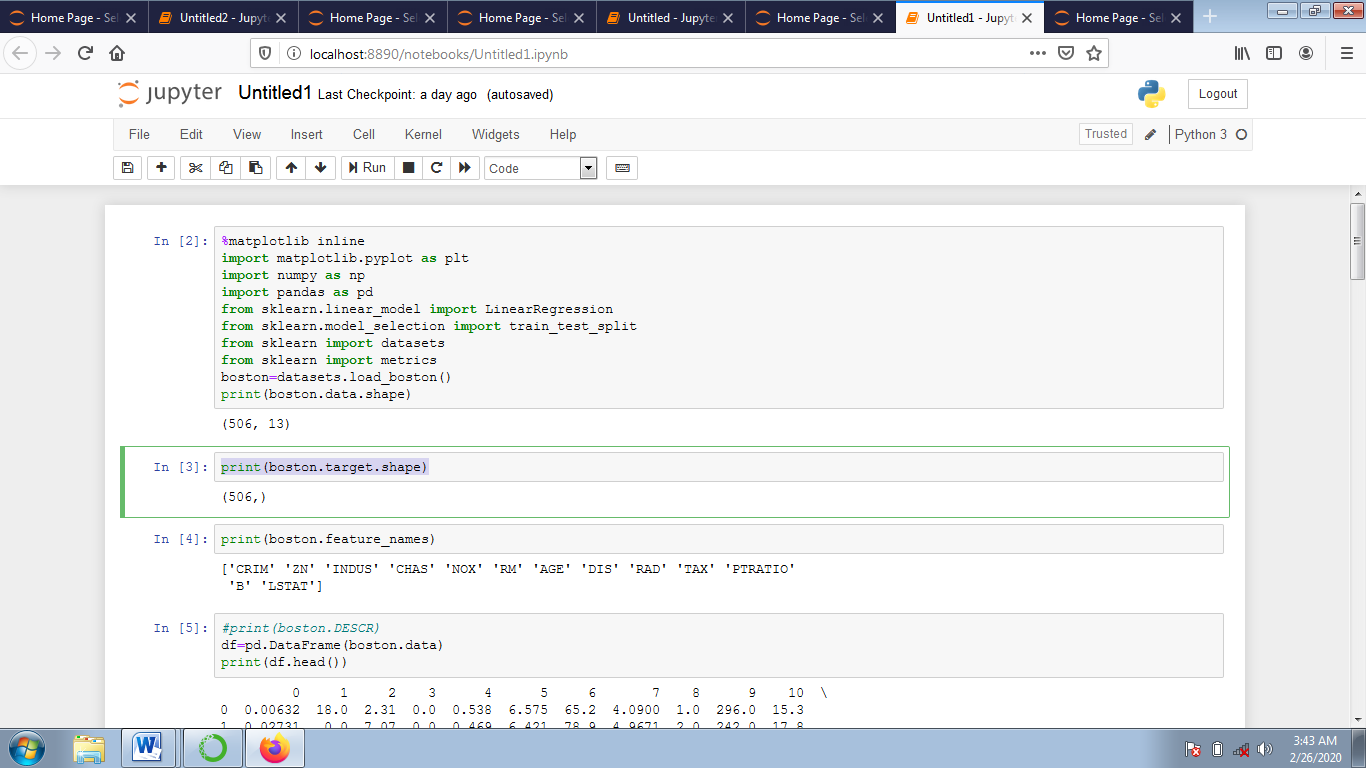
from sklearn.linear\_model import LinearRegression

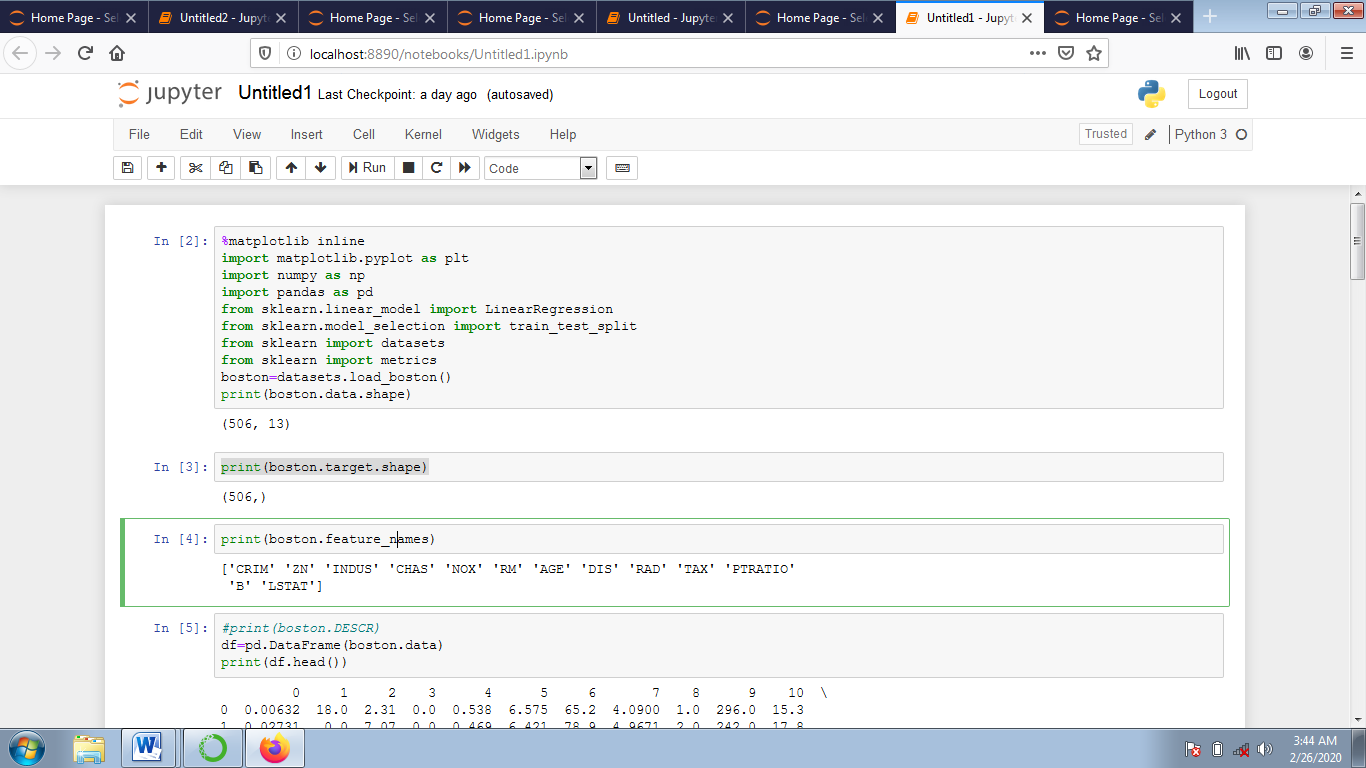
from sklearn.model\_selection import train\_test\_split

from sklearn import datasets

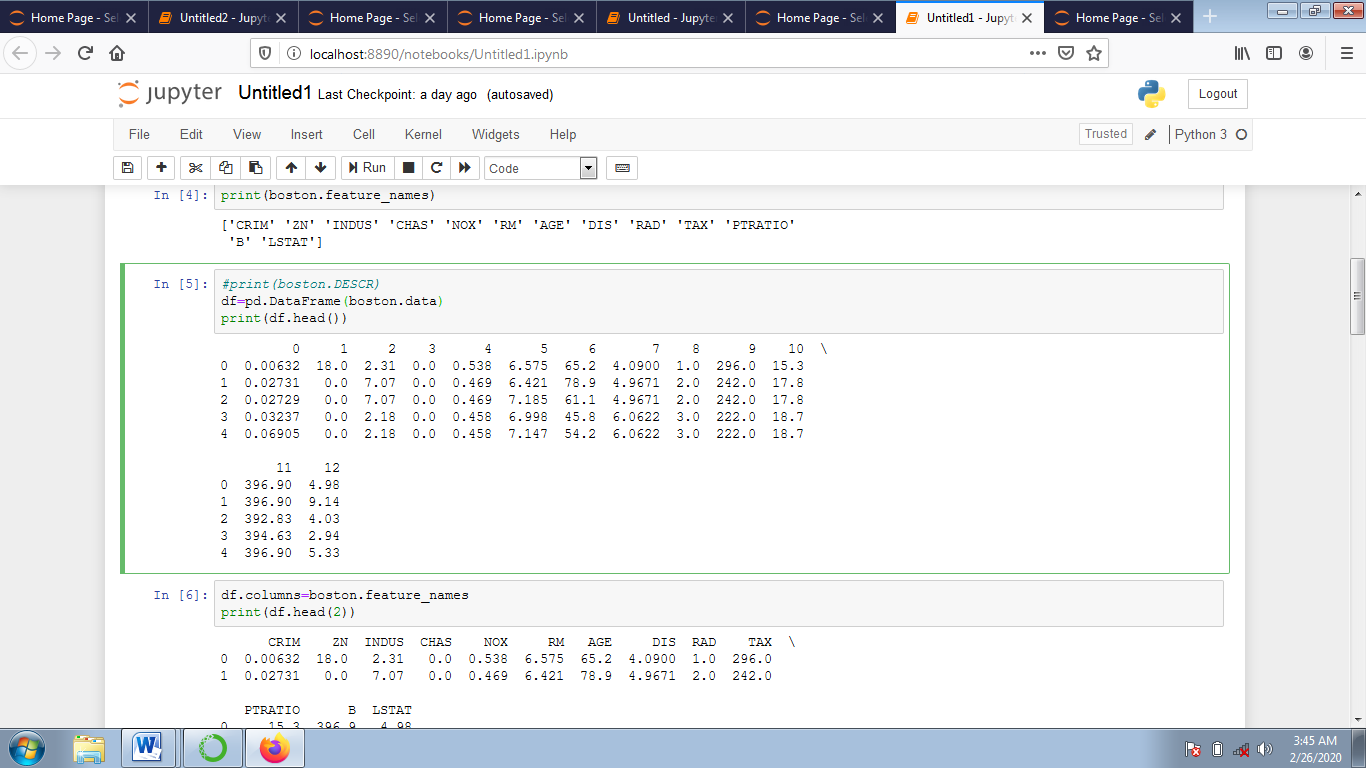
from sklearn import metrics

boston=datasets.load\_boston()

print(boston.data.shape) 

print(boston.target.shape)

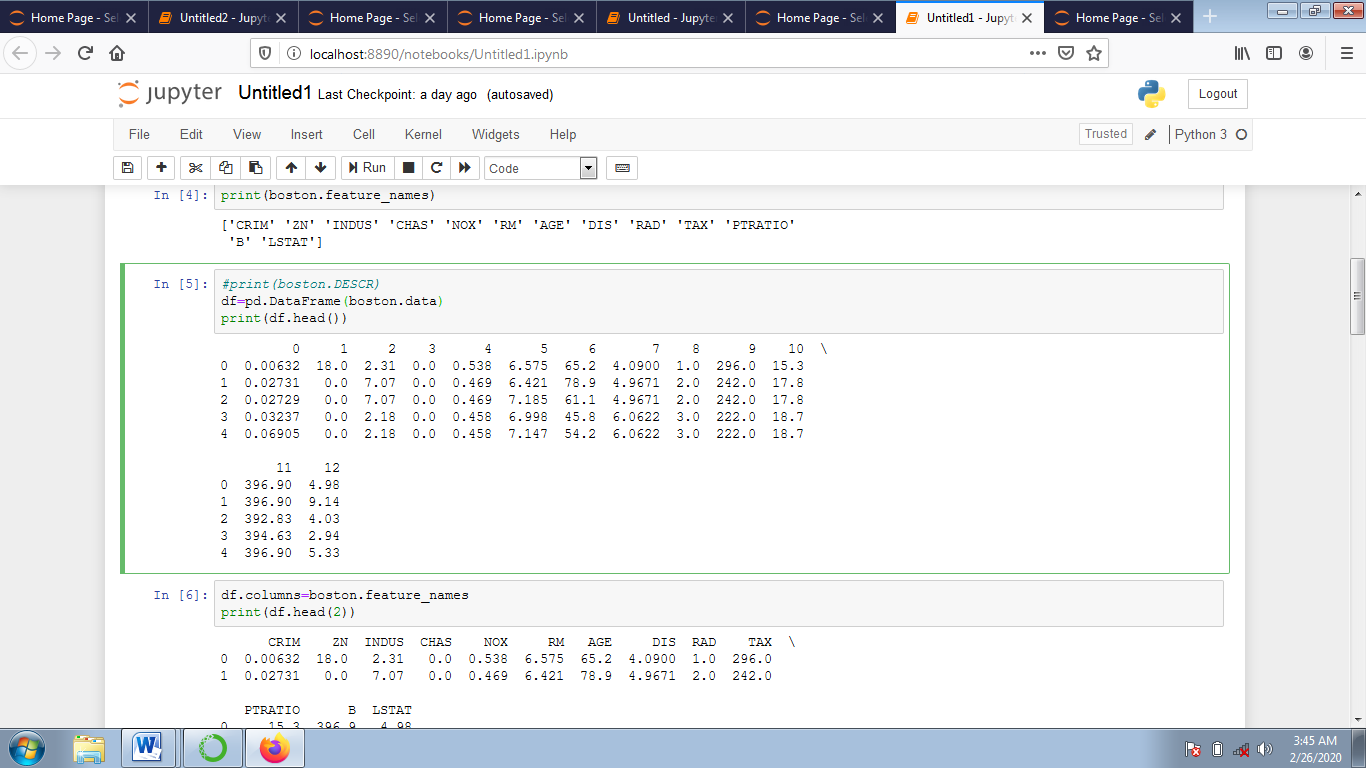
print(boston.feature\_names)



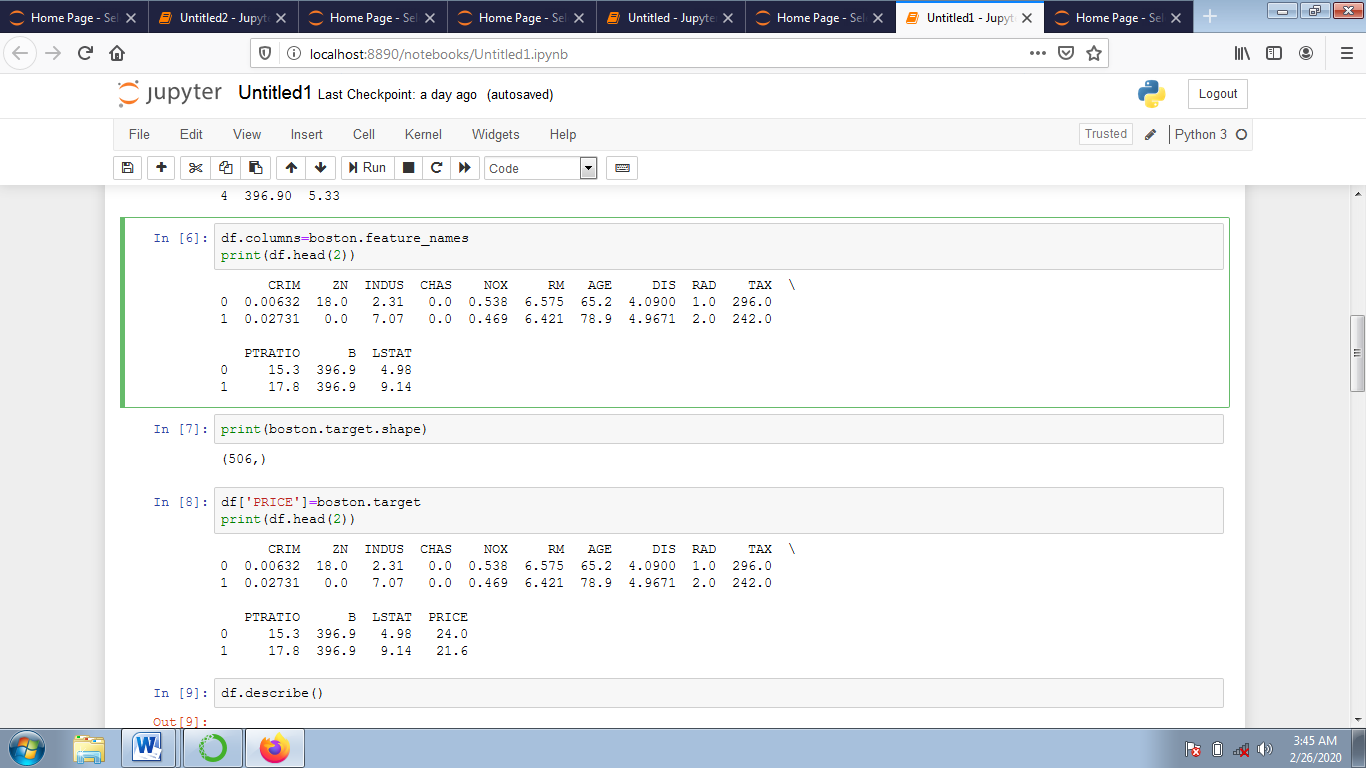
#print(boston.DESCR)

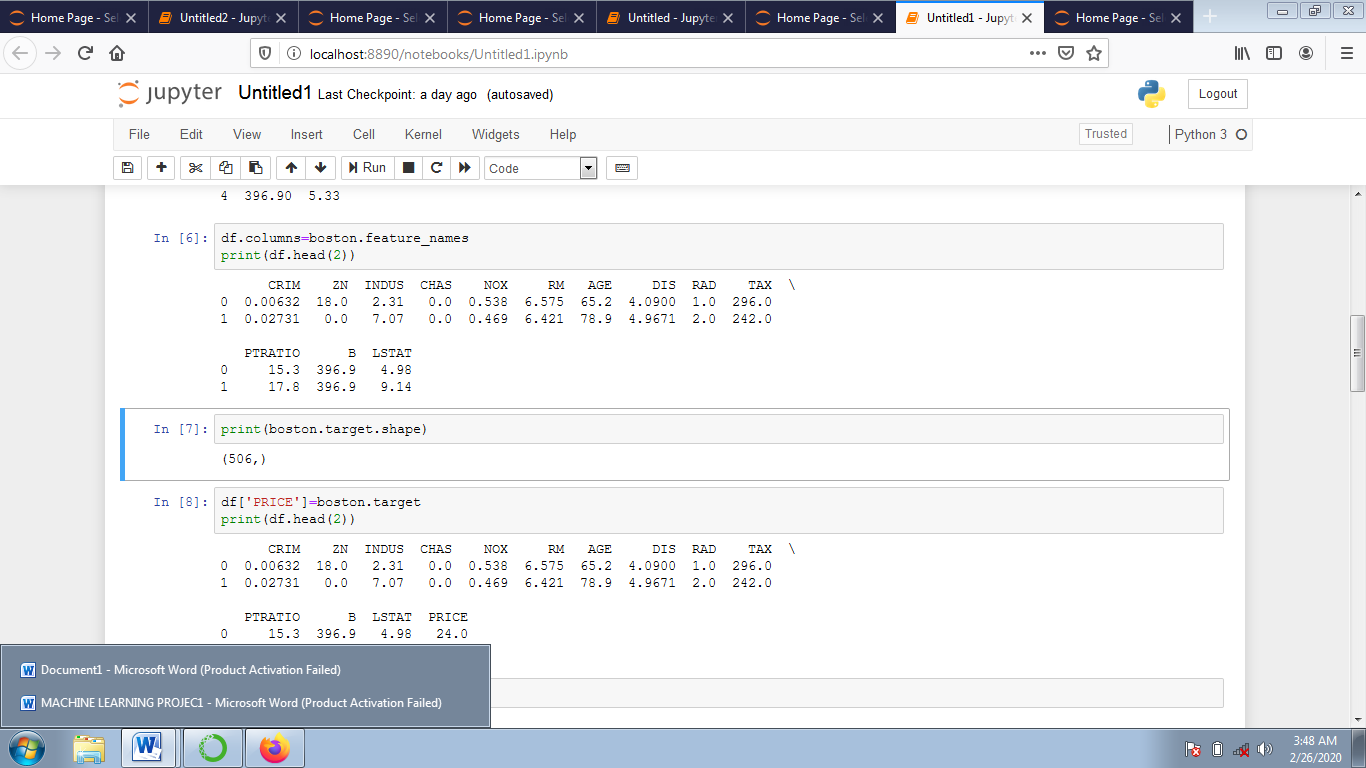
df=pd.DataFrame(boston.data)

print(df.head())



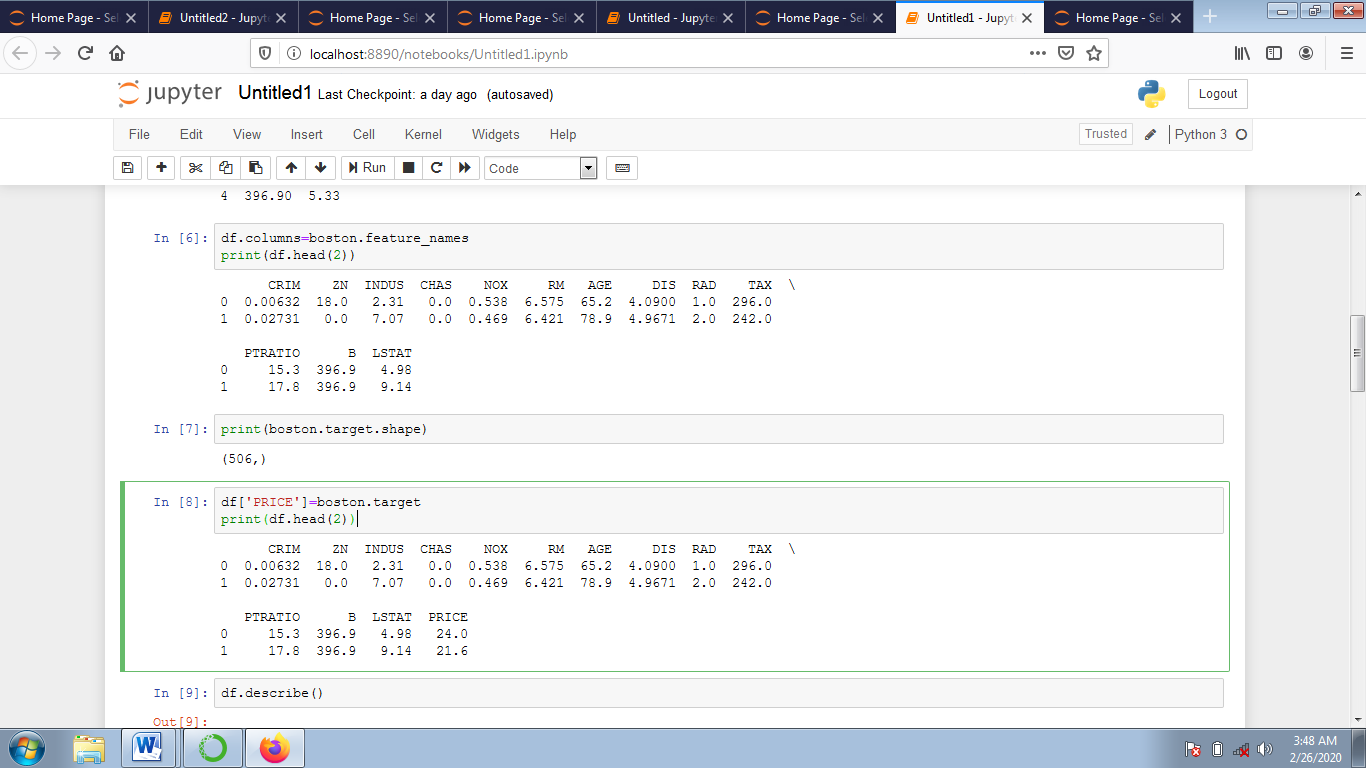
df.columns=boston.feature\_names

print(df.head(2))

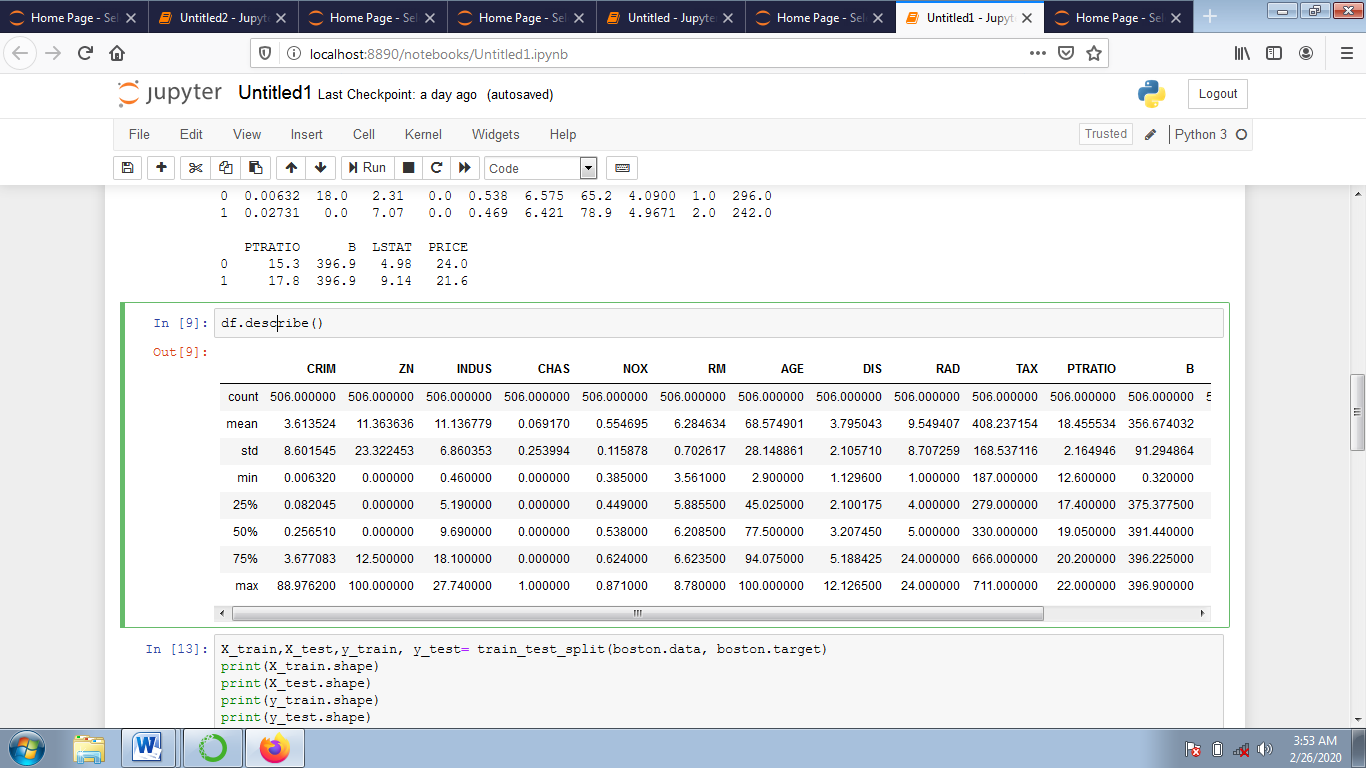
print(boston.target.shape)

df['PRICE']=boston.target

print(df.head(2))



df.describe()



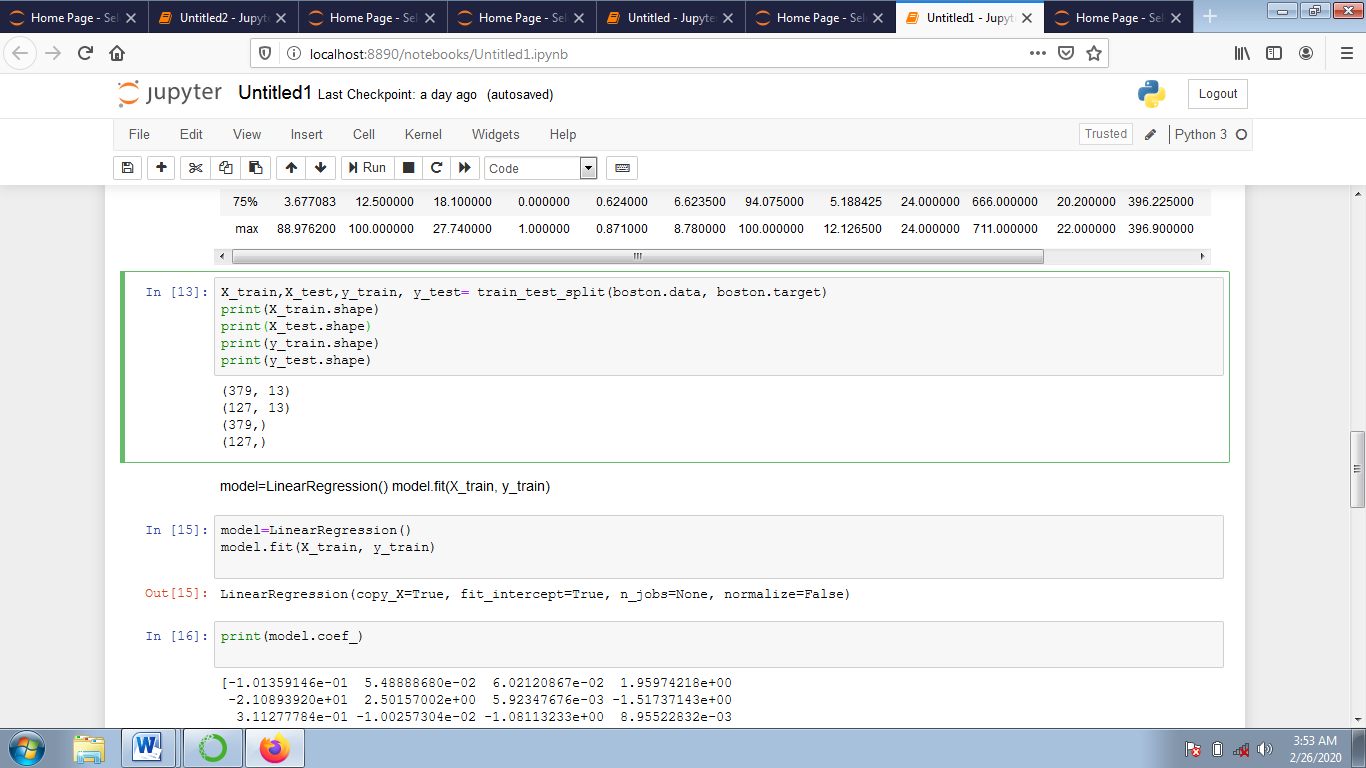
X\_train,X\_test,y\_train, y\_test= train\_test\_split(boston.data, boston.target)

print(X\_train.shape)

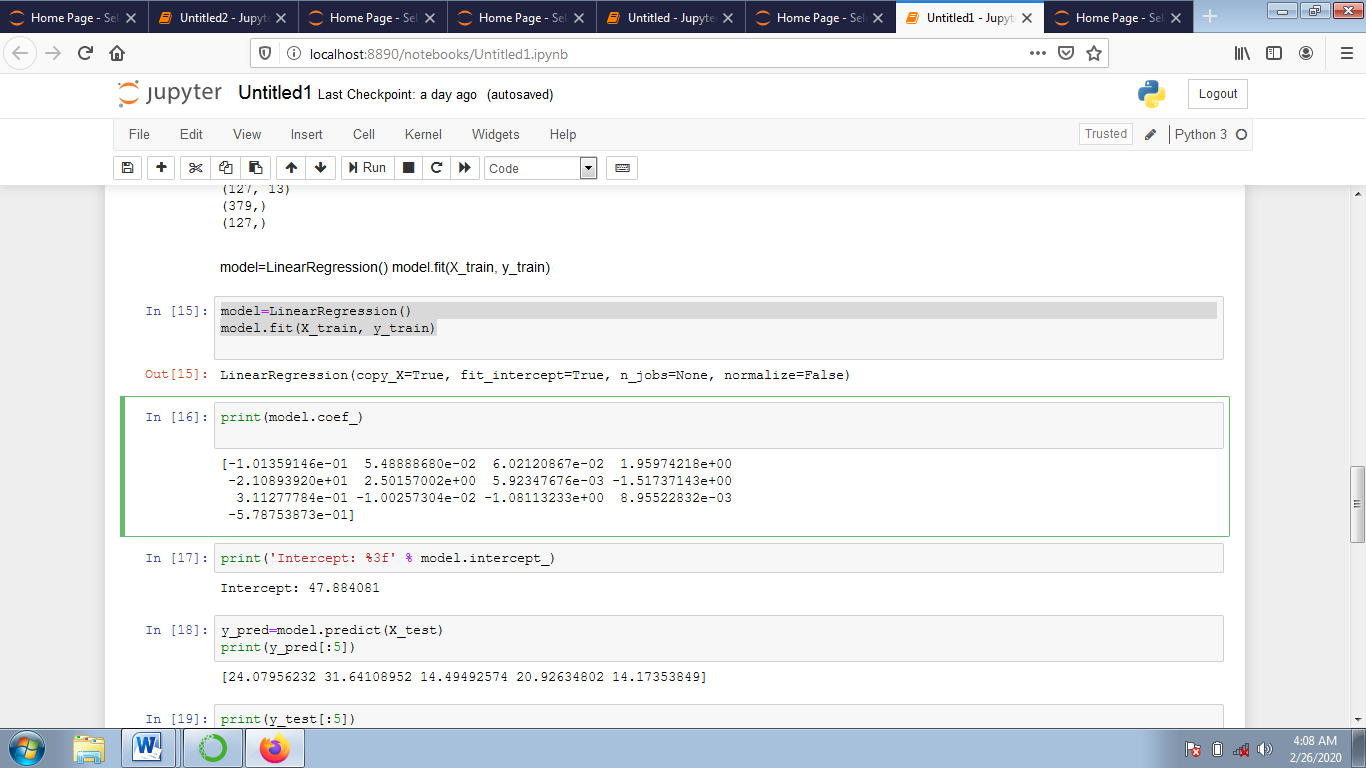
print(X\_test.shape)

print(y\_train.shape)

print(y\_test.shape)

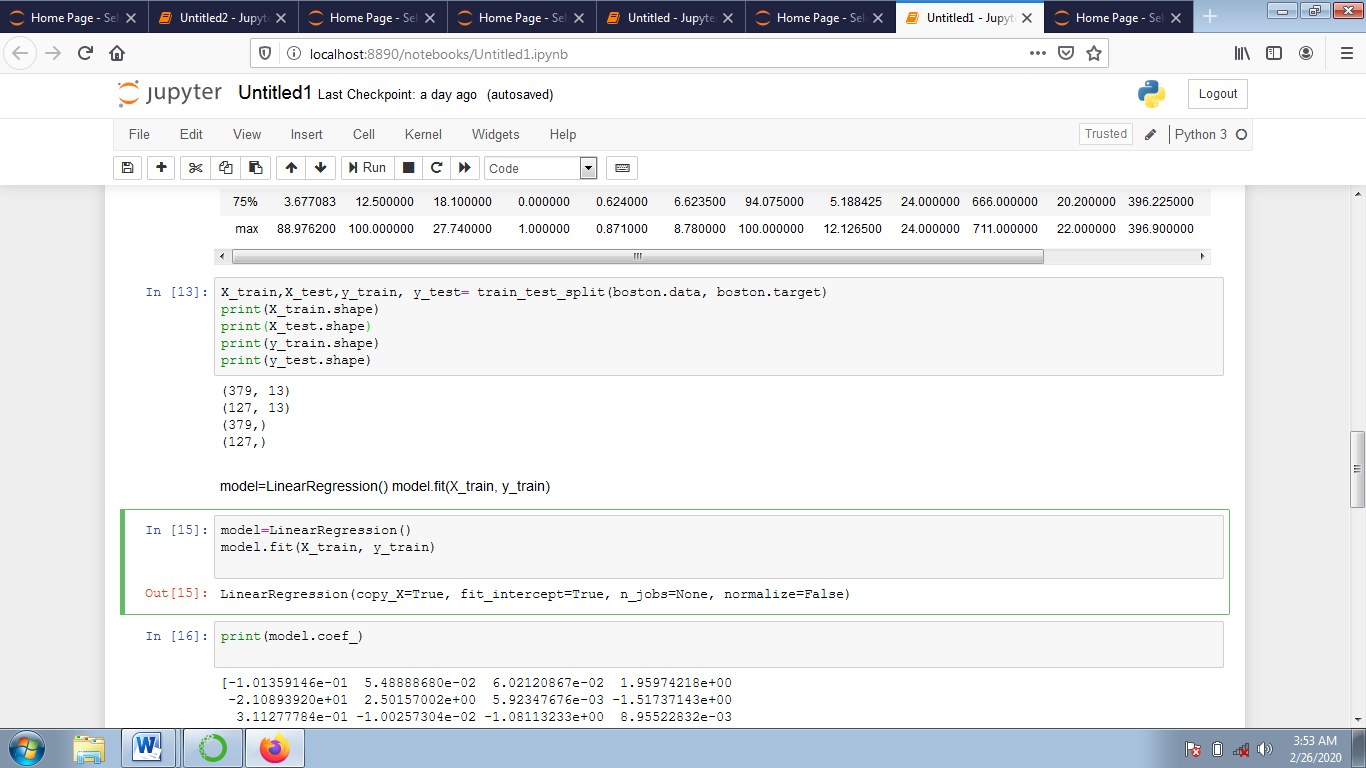


model=LinearRegression()

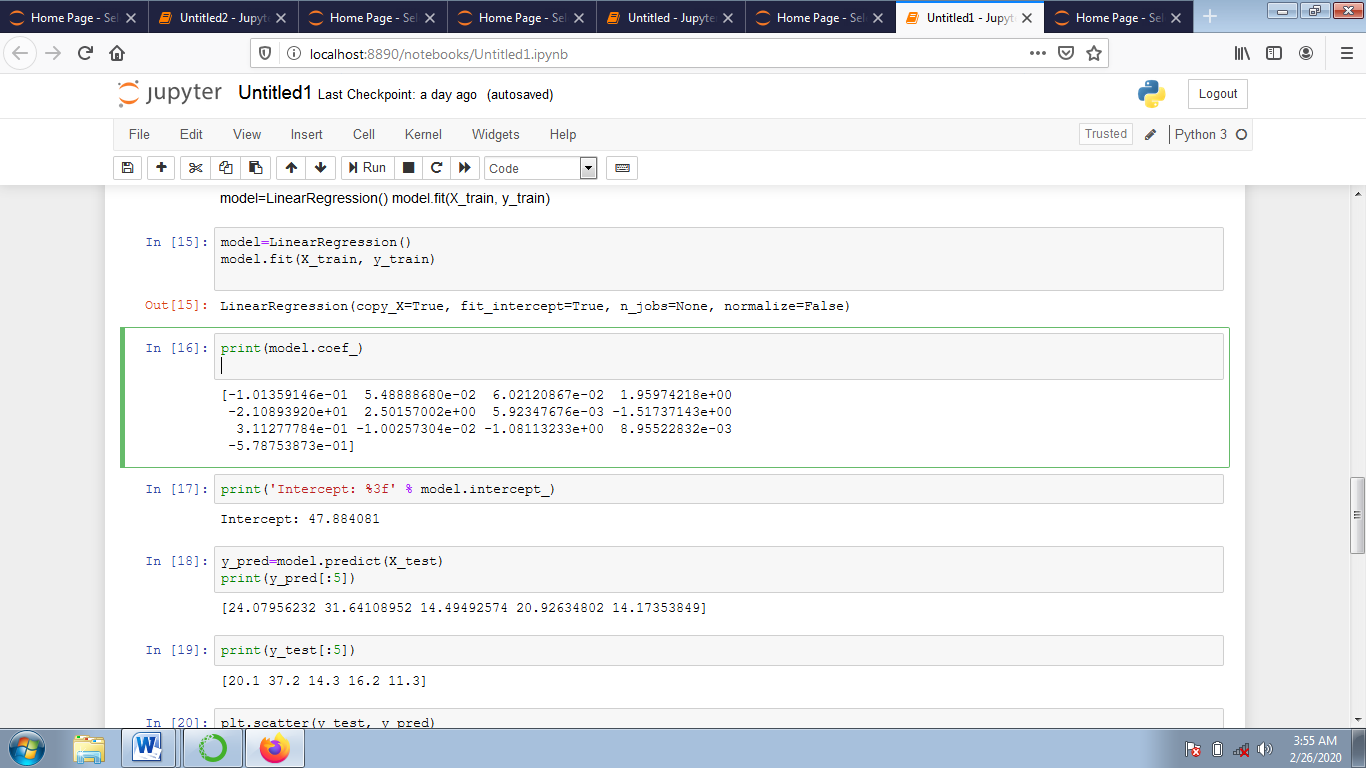
model.fit(X\_train, y\_train) 

model=LinearRegression()

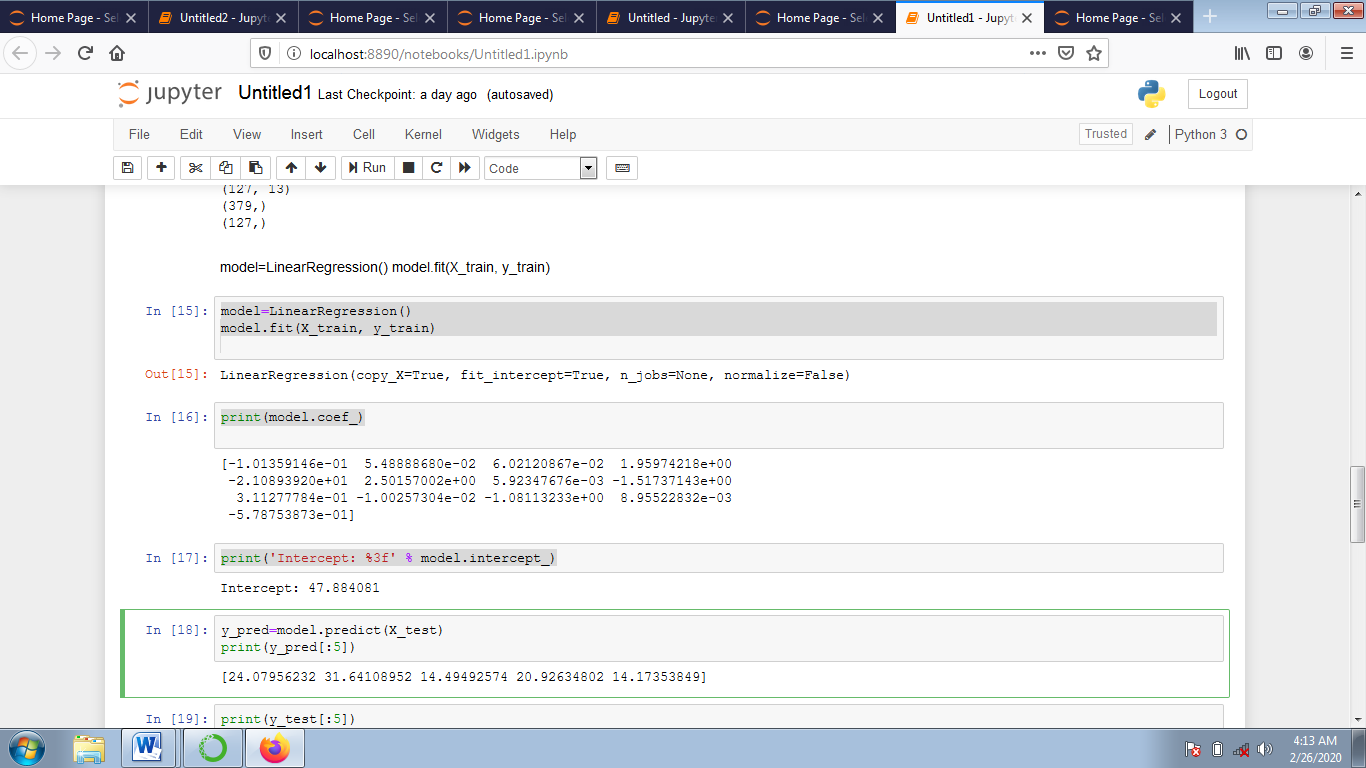
model.fit(X\_train, y\_train)



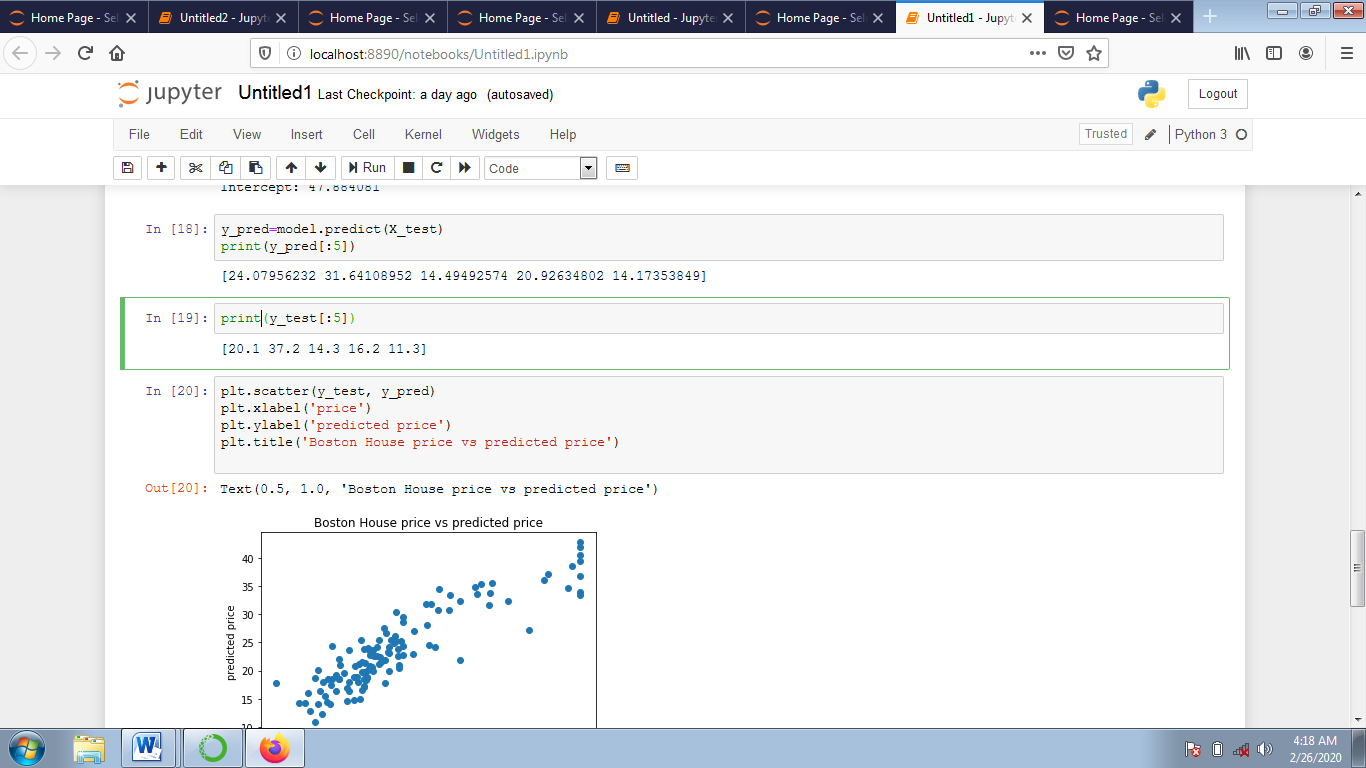
print(model.coef\_)



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

print(y\_pred[:5])

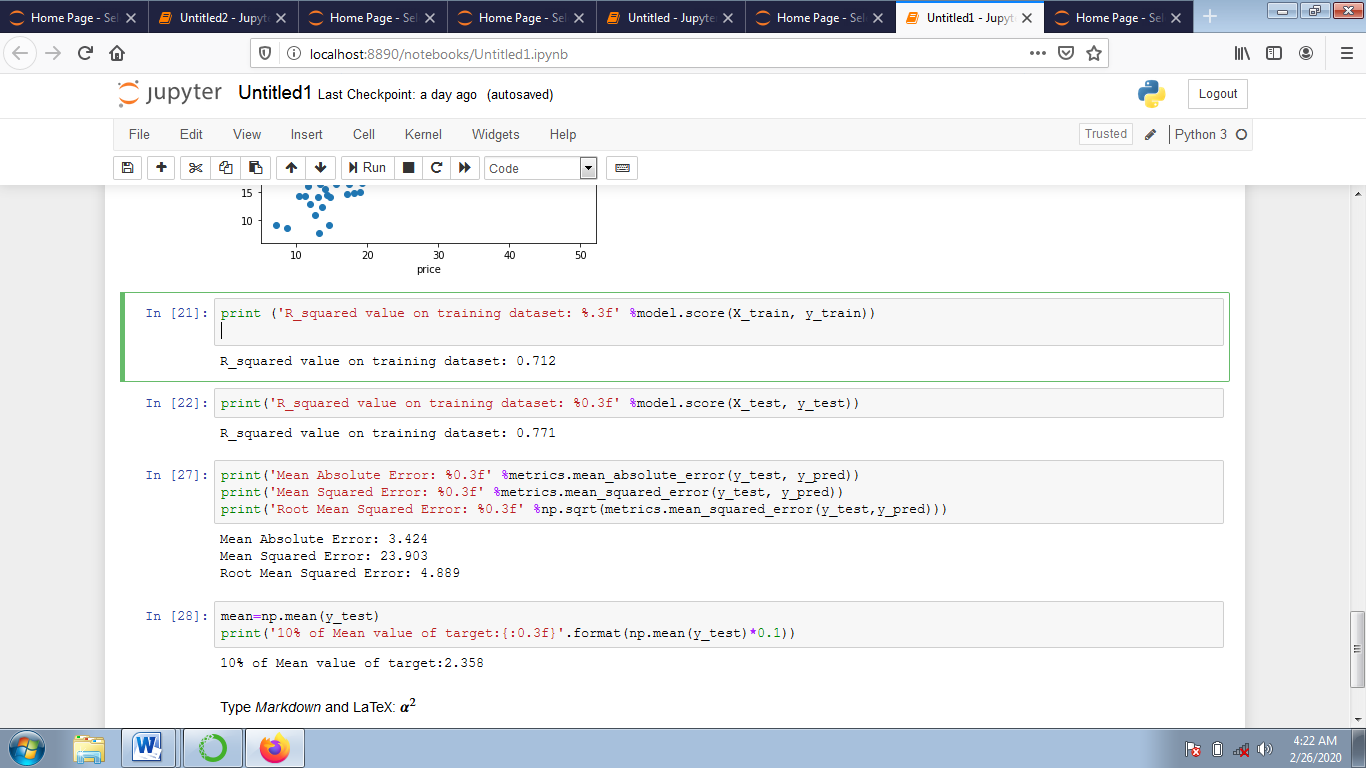
print(y\_test[:5])



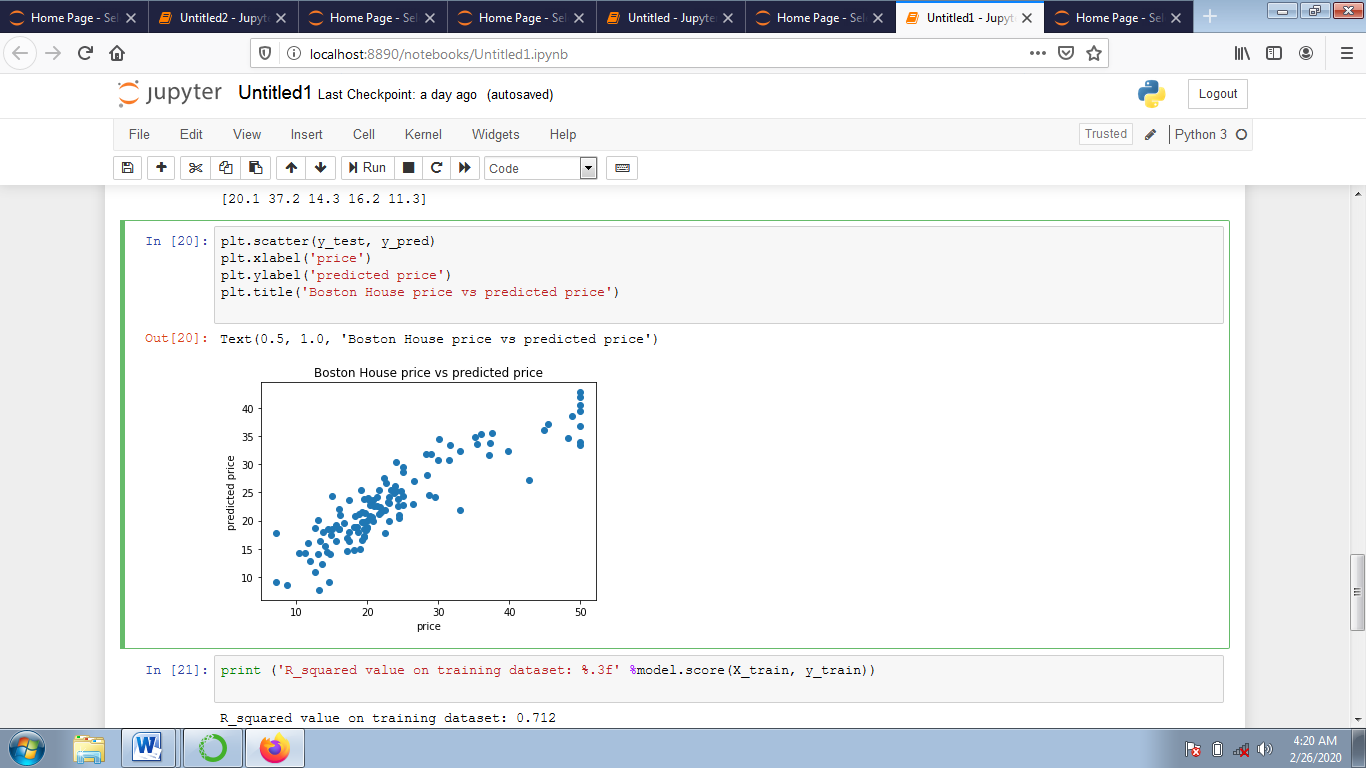
plt.scatter(y\_test, y\_pred)

plt.xlabel('price')

plt.ylabel('predicted price')

plt.title('Boston House price vs predicted price') 

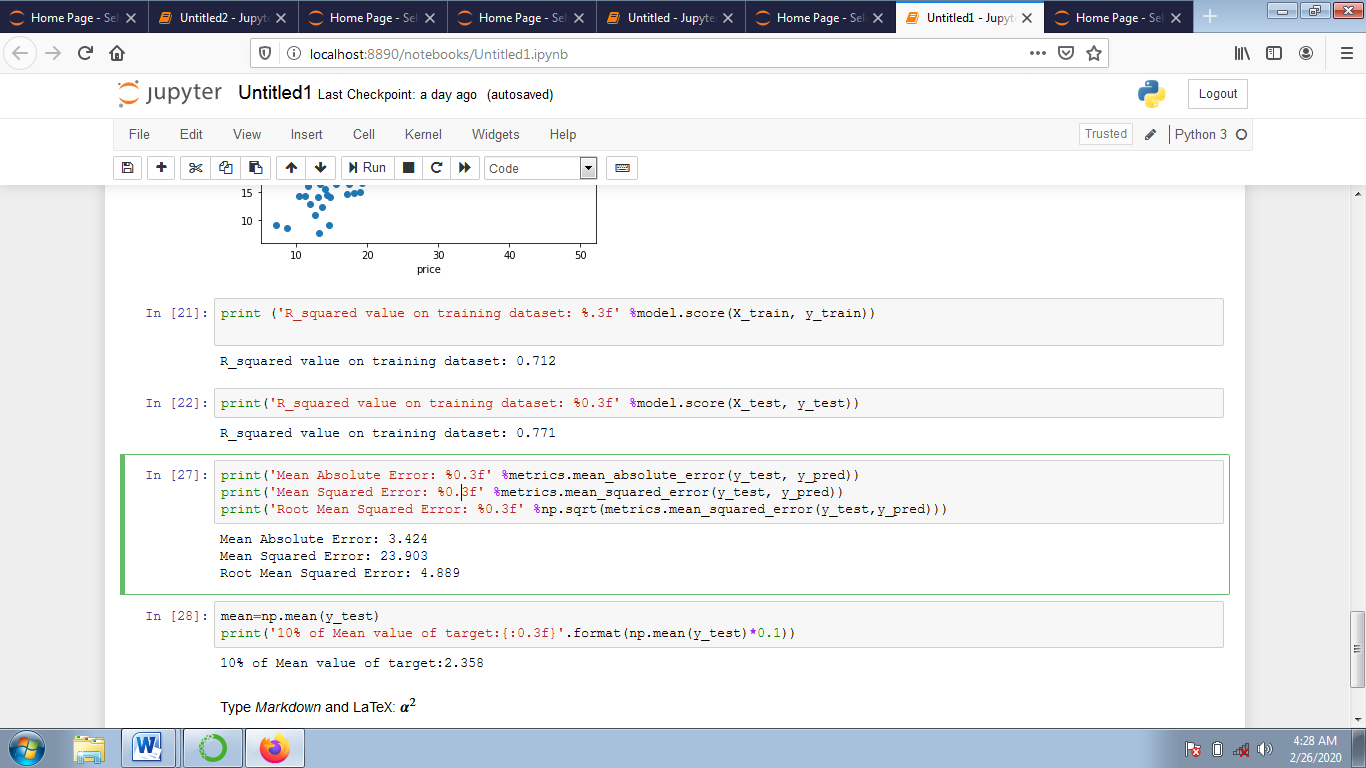
print ('R\_squared value on training dataset: %.3f' %model.score(X\_train, y\_train))



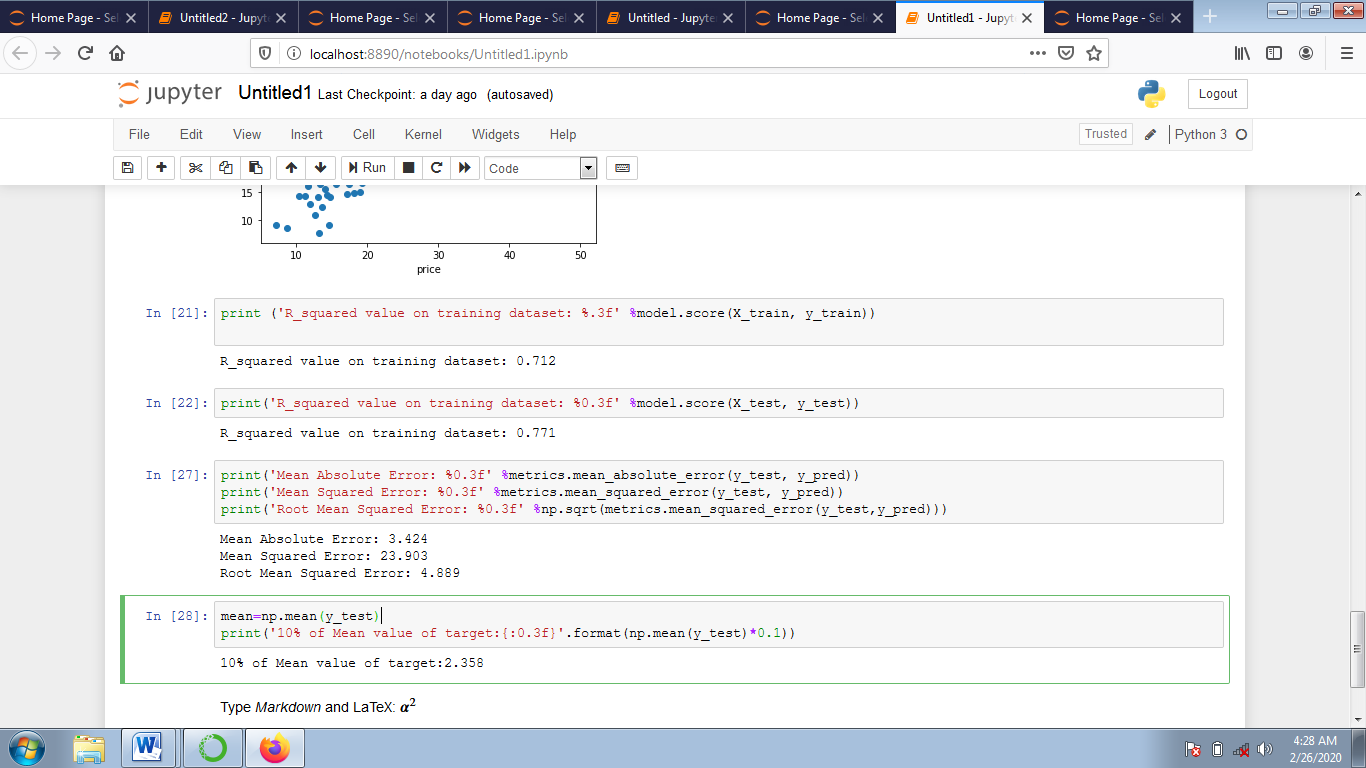
print('Mean Absolute Error: %0.3f' %metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error: %0.3f' %metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error: %0.3f' %np.sqrt(metrics.mean\_squared\_error(y\_test,y\_pred)))



mean=np.mean(y\_test)

print('10% of Mean value of target:{:0.3f}'.format(np.mean(y\_test)\*0.1))

**CONCLUSION:**

Price of Boston housing dataset is predicted by using linear regression method by using training and testing the data.