Assignment # 1

Danish Hafeez
Dept. of ECE
Course:Machine Learning
Air University
danish.hafeez76@gmail.com

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—Linear-Regression, Machine-Learning, Data-visualization, Hypothesis, Cost, Prediction**

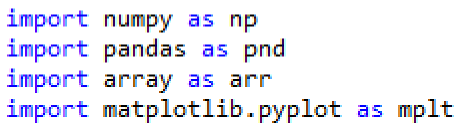
I. Introduction of Problem # 1

In this Assignment, I was to solve two Problems, In Problem 1 a data-set of a fruit distributing company was provided in which X is the population of a city and Y is the profit of a fruit truck in that city. By implementing Linear Regression I had to find a Hypothesis H(X) that best fits the data points available on the plot the data set we had.The main problem was finding the suitable θ's for x0 and x1. I also represented it graphically so that I can visually see what is happening.Different types of graphs can be plotted to represent the effects of Linear Regression being applied.This is a Uni-variate Linear Regression problem means the relationship is between one independent and one dependent variable

II. Explanation of Implantation

A. Importing Libraries

First of all I imported libraries which will be further used in my solution model of the problem

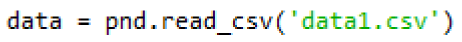


```
import numpy as np
import pandas as pnd
import array as arr
import matplotlib.pyplot as mplt
```

Snippet of Importing Libraries

B. Data File Input

I Used the **Panda** library to import the data set. I first converted it into .csv type file,so that i can access each column.

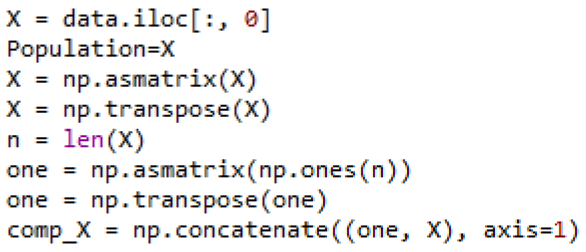


```
data = pnd.read_csv('data1.csv')
```

Data-frame input

C. Creating a Input Feature's Matrix

I imported the population set from data file and saved it in a matrix took transpose of it to convert it in a column matrix than created a column matrix of ones and concatenated the both now my on put featured vector variable created is **x_comp**

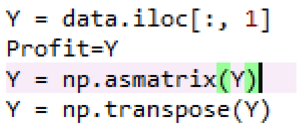


```
X = data.iloc[:, 0]
Population=X
X = np.asmatrix(X)
X = np.transpose(X)
n = len(X)
one = np.asmatrix(np.ones(n))
one = np.transpose(one)
comp_X = np.concatenate((one, X), axis=1)
```

Input Matrix

D. Creating a Output Matrix

I imported the profit set from data file and saved it in a matrix took transpose of it to convert it in a column matrix my output Vector is **Y**

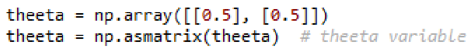


```
Y = data.iloc[:, 1]
Profit=Y
Y = np.asmatrix(Y)
Y = np.transpose(Y)
```

Create an output matrix

E. Declaring Theeta

I created an array of theeta upon which I will first create a hypothesis than will find the cost until valid theeta's are found after that we can say that our machine is trained.

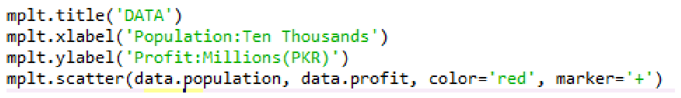


```
theeta = np.array([[0.5], [0.5]])
theeta = np.asmatrix(theeta)  # theeta variable
```

Declaration of theeta

F. Visualizing Data

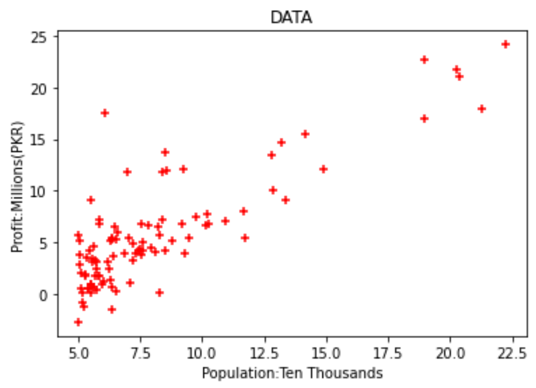
I plot the input and output to visualize what kind of data is given



```
mplt.title('DATA')
mplt.xlabel('Population:Ten Thousands')
mplt.ylabel('Profit:Millions(PKR)')
mplt.scatter(data.population, data.profit, color='red', marker='+')
```

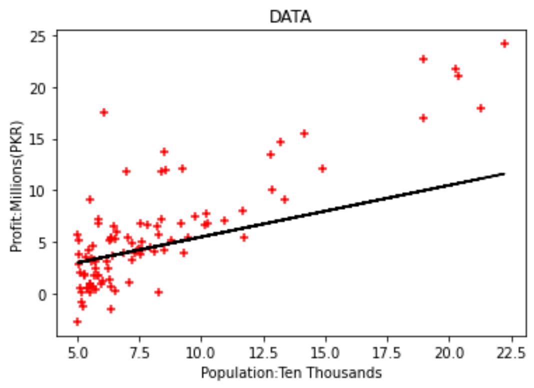
Plotting Data

The plot is as shown in figure



Plot of Give Data

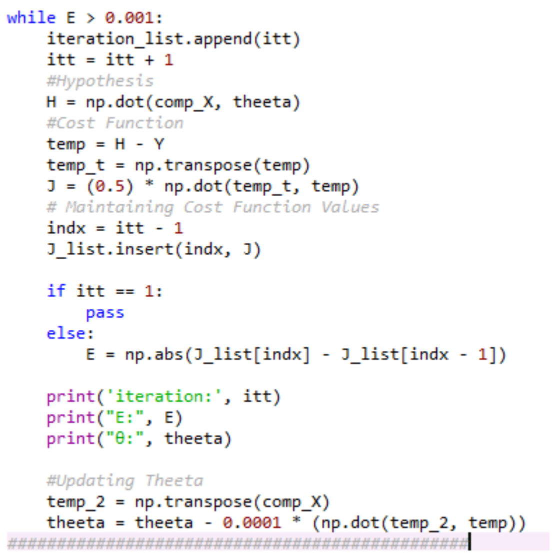
Plot of our initial Hypothesis is also shown



initial Hypothesis Plot

G. *While loop for Calculating Gradient Descend*

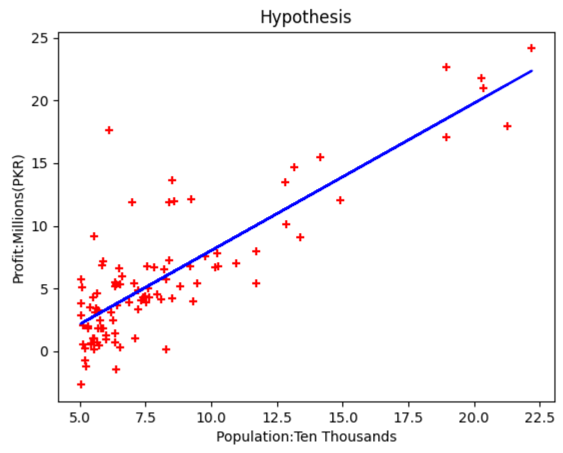
This loop is the Core of our model, in this we have st Criteria of based on E which is the difference betweer of last two iterations if E is less than 0.001 than loop stop and we will consider that our θas are predicted R and Machine is learned.



```
while E > 0.001:
    iteration_list.append(itt)
    itt = itt + 1
    #Hypothesis
    H = np.dot(comp_X, theeta)
    #Cost Function
    temp = H - Y
    temp_t = np.transpose(temp)
    J = (0.5) * np.dot(temp_t, temp)
    # Maintaining Cost Function Values
    indx = itt - 1
    J_list.insert(indx, J)

    if itt == 1:
        pass
    else:
        E = np.abs(J_list[indx] - J_list[indx - 1])

    print('iteration:', itt)
    print("E:", E)
    print("θ:", theeta)

    #Updating Theeta
    temp_2 = np.transpose(comp_X)
    theeta = theeta - 0.0001 * (np.dot(temp_2, temp))
################################################
```

Complete code for Gradient Descend calculation

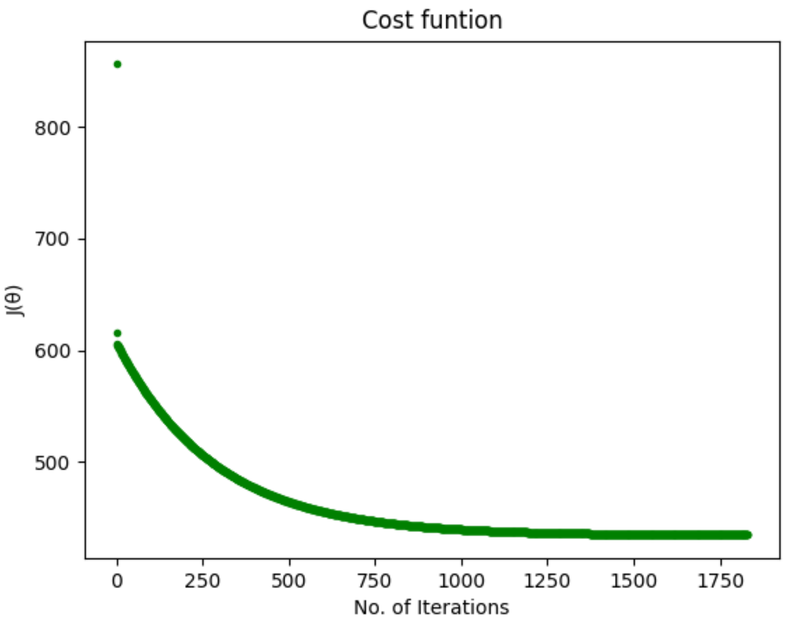
In this plot we set predicted the thetas and found the hypothesis upon that theta and our Line Fits perfectly on the data



Final Predicted Hypothesis

H. *Cost Function Plot*

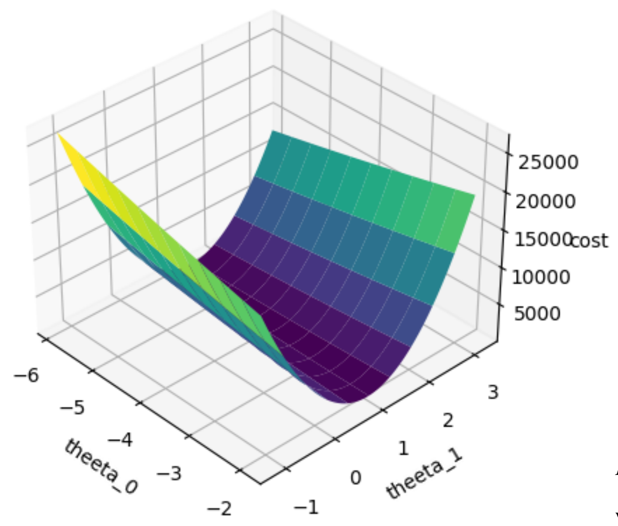
I saved the value of each cost during while loop and used those values to plot the cost function an exponential decline can clearly be seen in figure



Cost function

I. *Surface Plot*

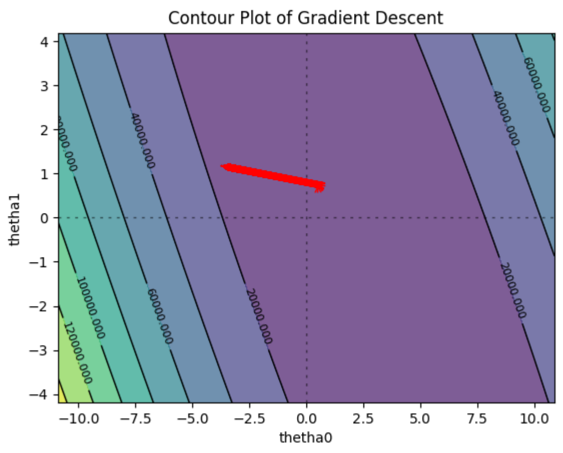
surface Plot is actually a three dimensional plot two variables are in-dependant one is dependant,in our case our in-dependant variables were theta0 and theta1 which were a range of values created by using mesh-Grid and we used those value to find our cost function at different combinations of theta and than plot it



Surface Plot

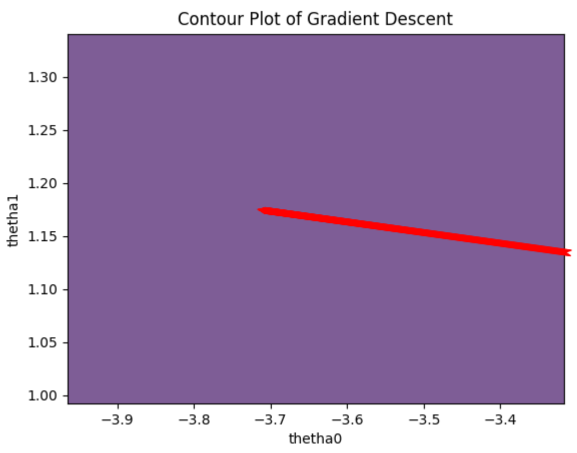
J. Contour Plot

A contour plot show us the slicing of surface plot the same matrix I created for the surface plot was used to plot contour plot than with a technique I showed moving arrows on that plot It was an amazing experience Contour is shown in this figure



Contour Graph

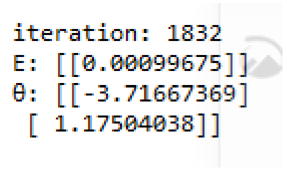
and If zoomed in the contour we can clearly see that it shows us the path toward our desired theta



Zoomed in contour graph

K. Final Readings

when I took α $\bar{0}$.0001 and E $\bar{0}$.001 my hypothesis was predicted with the line fitting perfectly when ever I Decreased the value of α the function converge with less iterations and when I increased α the function did never converged and nan value was stored in θas



```
iteration: 1832
E: [[0.00099675]]
θ: [[-3.71667369]
 [ 1.17504038]]
```

Final Predicted Hypothesis

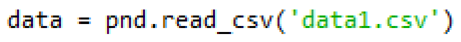
III. Introduction of Problem # 2

In this problem we were approached by a Real state investor which provided us with the data set of Area of house,# of bedrooms and Price of that house in a comma separated text file.I proposed linear regression.Procedure is same as explained in Problem 1 but their is a difference we will see that data set it-self is not that much in one scale like area will be in 100s price will be in 10000s and no. of bedrooms will be less than 20 which will directly effect our regression,So I came up with the Idea of Mean normalization After that all my data was in one scale an it converged faster than I expected lets look into the implementation of our Model

IV. Explanation of Implementing the Code

A. Importing Data file

I imported data file a txt file was provided i first made a csv file of our data and imported it in my code I used the following statement

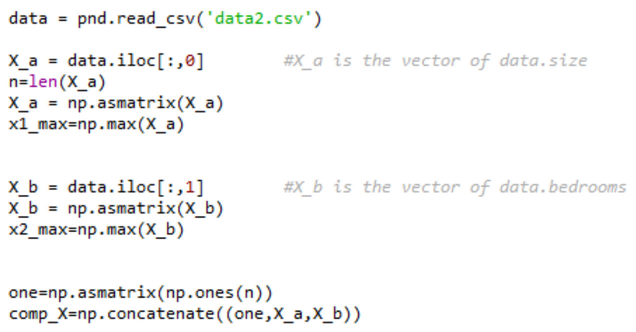


```
data = pnd.read_csv('data1.csv')
```

Final Predicted Hypothesis

B. *Creating a Input Feature's Matrix*

I imported size and # of bedrooms which will be our x1 and x2 respectively and for x0 i created a column matrix of ones.I concatenated these matrices to form one Matrix of size 3*n



```
data = pnd.read_csv('data2.csv')

X_a = data.iloc[:,0]        #X_a is the vector of data.size
n=len(X_a)
X_a = np.asmatrix(X_a)
x1_max=np.max(X_a)


X_b = data.iloc[:,1]        #X_b is the vector of data.bedrooms
X_b = np.asmatrix(X_b)
x2_max=np.max(X_b)


one=np.asmatrix(np.ones(n))
comp_X=np.concatenate((one,X_a,X_b))
```

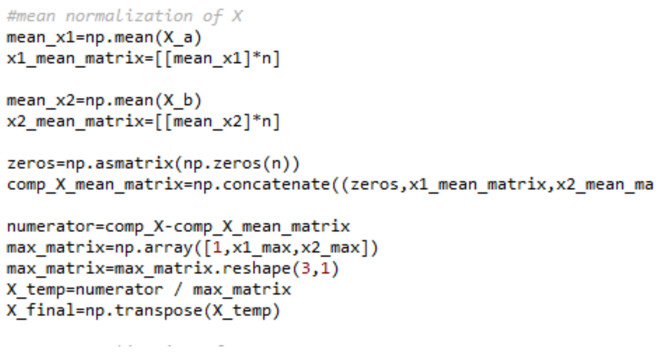
Input Matrix

C. *Mean Normalization of our Input Matrix*

Our Data is not in one scale and it is very difficult for it to converge or we can say predict the right hypothesis so to scale it down between -1 to 1 I used a technique called mean normalization using formula

$$\frac{X_i - mean(Xi)}{max(X_i)} \tag{1}$$

I used pure Vector implementation of this formula that is Complete Mean Normalization in one go Code is attached.I first created a temp matrix of 3*n with 1st rov of zeros,second having mean of x1 and third row with n of x2, subtracted x obtained from data frame and x creat by me than divided it with maximum value of each x0,x my Final Matrix **X_final**

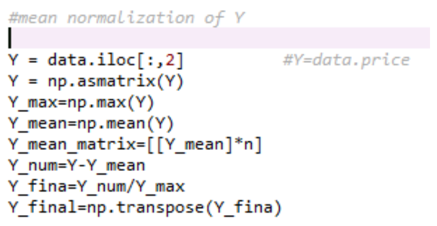


```
#mean normalization of X
mean_x1=np.mean(X_a)
x1_mean_matrix=[[mean_x1]*n]

mean_x2=np.mean(X_b)
x2_mean_matrix=[[mean_x2]*n]

zeros=np.asmatrix(np.zeros(n))
comp_X_mean_matrix=np.concatenate((zeros,x1_mean_matrix,x2_mean_ma

numerator=comp_X-comp_X_mean_matrix
max_matrix=np.array([1,x1_max,x2_max])
max_matrix=max_matrix.reshape(3,1)
X_temp=numerator / max_matrix
X_final=np.transpose(X_temp)
```

Input Matrix

D. *mean normalization in Given Output*

Also with the help of mean normalization i scaled down the output between -1 to 1. I implemented this with the help of vectors as shown in code

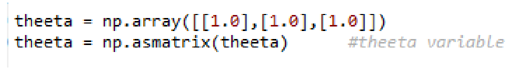


```
#mean normalization of Y

Y = data.iloc[:,2]          #Y=data.price
Y = np.asmatrix(Y)
Y_max=np.max(Y)
Y_mean=np.mean(Y)
Y_mean_matrix=[[Y_mean]*n]
Y_num=Y-Y_mean
Y_fina=Y_num/Y_max
Y_final=np.transpose(Y_fina)
```

Mean Normalized output matrix

E. *Θas*

in this we will take three θas which matrix is created as shown

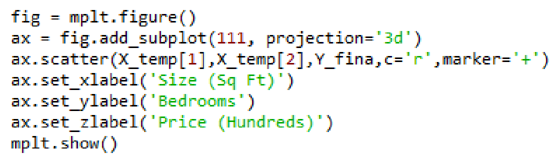


```
theeta = np.array([[1.0],[1.0],[1.0]])
theeta = np.asmatrix(theeta)      #theeta variable
```

θ

F. *Scatter plot of Given Data frame*

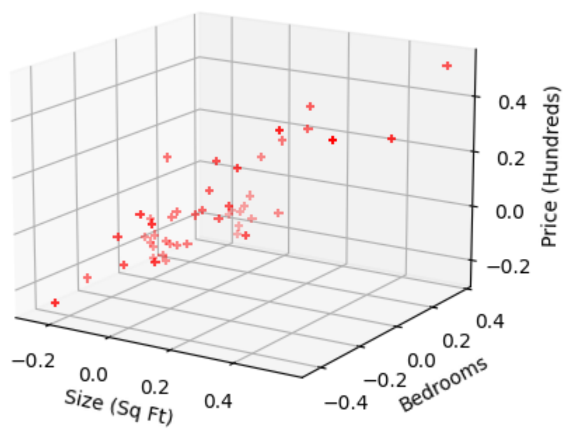
Visually represented data in 3D helps us to see things clearly



```
fig = mplt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X_temp[1],X_temp[2],Y_fina,c='r',marker='+')
ax.set_xlabel('Size (Sq Ft)')
ax.set_ylabel('Bedrooms')
ax.set_zlabel('Price (Hundreds)')
mplt.show()
```

Code for Scatter plot

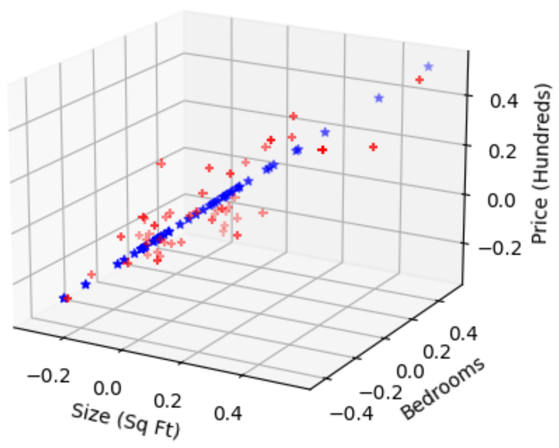
Scatter Plot is shown in figure



Scatter Plot

G. *Hypothesis*

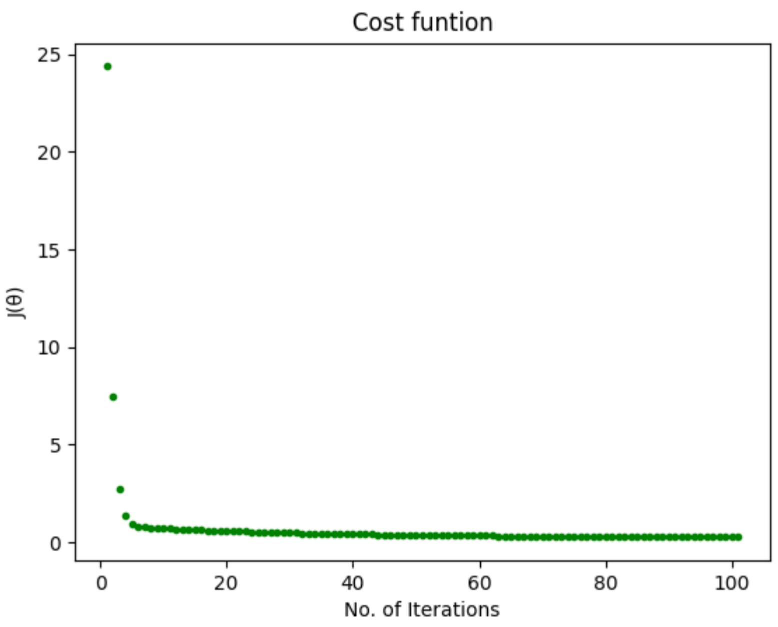
Visually represented data in 3D helps us to see things clearly so I calculated hypothesis like explained in previous example and plotted the final prediction which is shown



Final Hypothesis

H. Cost Function Plot

I saved the value of each cost during while loop and used those values to plot the cost function an exponential decline can clearly be seen in figure



Cost function

V. General review

You are indeed one of the best teachers I ever had,It is tottaly my blame for late submission
one of the thing I liked about this assignment or this this course is you are preparing us for real world problems