Shih-Han Wang, Shakti Suman

**Q1**

**Code:**

import math

import matplotlib.pyplot as plt

def Gradient\_descent(theata0, theata1, m, Alpha):

x=[3,1,0,4]

y=[2,2,1,3]

Alpha=0.05

resultJ=[]

resultTheata1=[]

#formula

#h=theata0+theata1\*x[i]

#J=1/(2\*m) \* math.pow((h-y[i]), 2)

#5 Round

for i in range(5):

sum=0

gradient1=0

gradient2=0

currentJ=0

#sum

for j in range(4):

h=theata0+theata1\*x[j]

#sum (h(x)-y)^2

sum+=math.pow((h-y[j]), 2)

#new gradient1 (h(x)-y)

gradient1+=(h-y[j])

#new gradient2 (h(x)-y)\*x

gradient2+=(h-y[j])\*x[j]

#update theata0 & theata1

theata0=theata0-Alpha/m\*gradient1

theata1=theata1-Alpha/m\*gradient2

resultTheata1.append(theata1)

#new J

currentJ=1/(2\*m)\*sum

print('J(Theata0, Theata1)=',currentJ)

resultJ.append(currentJ)

#plot

plt.xlabel('Theata1 ')

plt.ylabel('J(Theata0, Theata1) ')

plt.plot(resultTheata1, resultJ, 'bs')

plt.axis([0.7, 1, 0, 1])

plt.show()

#main

theata0=0

theata1=1

m=4

Alpha=0.05

x=Gradient\_descent(theata0, theata1, m, Alpha)

**Answer:**

When Alpha=0.05, error go down after 5 rounds.

J(Theata0, Theata1)= 0.5

J(Theata0, Theata1)= 0.4057812500000001

J(Theata0, Theata1)= 0.36099658203124996

J(Theata0, Theata1)= 0.33682167510986327

J(Theata0, Theata1)= 0.32135127205371866

