LAB3 OPTIMIZATION FOR MACHINE LEARNING

FALL SEMESTER 2023

Kshitij Jaiswal

B20CS028

**Code for Problem 1 -**

#kshitij jaiswal

#b20cs028

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

R = 8

def openCSV():

path = "6 columns - Copy.csv"

dt = pd.read\_csv(path)

return dt

if \_\_name\_\_ == '\_\_main\_\_':

dt = openCSV()

B=dt.values

x,y=B[:,1],B[:,0]

A=np.column\_stack((x,np.ones((len(x),1),dtype=float)))

beta=np.dot(np.linalg.inv(np.dot(A.T,A)),np.dot(A.T,y.T))

print("Cost of " + str(R) + "-thousand sq ft area house ",beta[0]\*R + beta[1])

plt.plot(x, y, 'b.')

plt.plot(x, beta[0]\*x+beta[1], 'r')

plt.xlabel('x')

plt.ylabel('y')

plt.show()

**Code for Problem 2 -**

#kshitij jaiswal

#b20cs028

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

R = 8

lb = 2

ub = 7

def openCSV():

path = '2\_col.xlsx'

dt = pd.read\_excel(path)

return dt

if \_\_name\_\_ == '\_\_main\_\_':

dt = openCSV()

B=dt.values

x,y=B[:,0],B[:,1]

numPnts = len(x)

A=np.column\_stack((x,np.ones((len(x),1),dtype=float)))

for n in range(lb, ub+1):

A = np.column\_stack((x\*\*n, A))

beta=np.dot(np.linalg.inv(np.dot(A.T,A)),np.dot(A.T,y.T))

Y = beta[0]\*(x\*\*n)

cost = beta[0]\*(R\*\*n)

for i in range(1, n+1):

Y += (beta[i]\*(x\*\*(n-i)))

cost += (beta[i]\*(R\*\*(n-i)))

print("cost of " + str(R) + "-thousand sq ft house " + str(n) + " poly fitting ", cost)

print("\n")

plt.plot(x, y, 'b.')

plt.plot(x, Y, 'r')

plt.xlabel('x')

plt.ylabel('y')

plt.show()

**Code for Problem 3 -**

#kshitij jaiswal

#b20cs028

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

R = 8

def openCSV():

path = "6 columns - Copy.csv"

dt = pd.read\_csv(path)

return dt

if \_\_name\_\_ == '\_\_main\_\_':

dt = openCSV()

B=dt.values

x,y,z=B[:,1],B[:,2],B[:,0]

numPnts = len(x)

A = np.ones((numPnts,1),dtype=float)

A = np.column\_stack((y, A))

A = np.column\_stack((x, A))

beta=np.dot(np.linalg.inv(np.dot(A.T,A)),np.dot(A.T,z.T))

price = beta[0]\*R + beta[1]\*(R+3) + beta[2]

print("cost of " + str(R) + "-thousand sq ft house & " + str(R+3) + " bedrooms ", price)

print("\n")

**Code for Problem 4 -**

#kshitij jaiswal

#b20cs028

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

R = 8

def openCSV():

path = "6 columns - Copy.csv"

dt = pd.read\_csv(path)

return dt

if \_\_name\_\_ == '\_\_main\_\_':

dt = openCSV()

B=dt.values

x,y,z=B[:,1],B[:,2],B[:,0]

numPnts = len(x)

A = np.ones((numPnts,1),dtype=float)

A = np.column\_stack((y, A))

A = np.column\_stack((x, A))

A = np.column\_stack((y\*\*2, A))

A = np.column\_stack((x\*y, A))

A = np.column\_stack((x\*\*2, A))

beta=np.dot(np.linalg.inv(np.dot(A.T,A)),np.dot(A.T,z.T))

price = beta[0]\*(R\*\*2) + beta[1]\*(R\*(R+3)) + beta[2]\*((R+3)\*\*2) + beta[3]\*(R) + beta[4]\*(R+3) + beta[5]

print("cost of " + str(R) + "-thousand sq ft house & " + str(R+3) + " bedrooms ", price)

print("\n")

**Code for Problem 5 -**

#kshitij jaiswal

#b20cs028

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

R = 8

def openCSV():

path = "6 columns - Copy.csv"

dt = pd.read\_csv(path)

return dt

def numBath(R):

if R&1:

return 2

return 1

if \_\_name\_\_ == '\_\_main\_\_':

dt = openCSV()

B=dt.values

x1,x2,x3,x4=B[:,1],B[:,2],B[:,3],B[:,0]

numPnts = len(x1)

A = np.ones((numPnts,1),dtype=float)

A = np.column\_stack((x3, A))

A = np.column\_stack((x2, A))

A = np.column\_stack((x1, A))

beta=np.dot(np.linalg.inv(np.dot(A.T,A)),np.dot(A.T,x4.T))

price = beta[0]\*R + beta[1]\*(R+3) + beta[2]\*(numBath(R)) + beta[3]

print("cost of " + str(R) + "-thousand sq ft house, " + str(R+3) + " bedrooms & " + str(numBath(R)) + " bathroom(s) ", price)

print("\n")

**Code for Problem 6 -**

#kshitij jaiswal

#b20cs028

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

R = 8

def openCSV():

path = "6 columns - Copy.csv"

dt = pd.read\_csv(path)

return dt

def numBath(R):

if R&1:

return 2

return 1

def numStorey(R):

return (R%4) + 1

def numAC(R):

if R&1:

return 0

return 1

if \_\_name\_\_ == '\_\_main\_\_':

dt = openCSV()

B=dt.values

x1,x2,x3,x4,x5,x6=B[:,1],B[:,2],B[:,3],B[:,4],B[:,5],B[:,0]

numPnts = len(x1)

A = np.ones((numPnts,1),dtype=float)

A = np.column\_stack((x4, A))

A = np.column\_stack((x3, A))

A = np.column\_stack((x2, A))

A = np.column\_stack((x1, A))

beta=np.dot(np.linalg.inv(np.dot(A.T,A)),np.dot(A.T,x6.T))

print(“coefficients of hyperplane for five columns”, beta)

price = beta[0]\*R + beta[1]\*(R+3) + beta[2]\*(numBath(R)) + beta[3]\*(numStorey(R))

print("cost of " + str(R) + "-thousand sq ft house, " + str(R+3) + " bedrooms, " + str(numBath(R)) + " bathroom(s) & " + str(numStorey(R)) + " storeys ", price)

print("\n")

A = np.ones((numPnts,1),dtype=float)

A = np.column\_stack((x5, A))

A = np.column\_stack((x4, A))

A = np.column\_stack((x3, A))

A = np.column\_stack((x2, A))

A = np.column\_stack((x1, A))

beta=np.dot(np.linalg.inv(np.dot(A.T,A)),np.dot(A.T,x6.T))

print(“coefficients of hyperplane for six columns”, beta)

price = beta[0]\*R + beta[1]\*(R+3) + beta[2]\*(numBath(R)) + beta[3]\*(numStorey(R)) + beta[4]\*(numAC(R))

print("cost of " + str(R) + "-thousand sq ft house, " + str(R+3) + " bedrooms, " + str(numBath(R)) + " bathroom(s), " + str(numStorey(R)) + " storey(s) & " + str(numAC(R)) + " AC(s) ", price)

print("\n")

**Code for Problem 7 -**

#kshitij jaiswal

#b20cs028

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

R = 8

def openCSV():

path = "Census data (Chandigarh).csv"

dt = pd.read\_csv(path)

return dt

if \_\_name\_\_ == '\_\_main\_\_':

dt = openCSV()

B=dt.values

x,y=B[:,0],B[:,1]

print(x)

yp = np.log(y)

print(yp)

A=np.column\_stack((x,np.ones((len(x),1),dtype=float)))

beta=np.dot(np.linalg.inv(np.dot(A.T,A)),np.dot(A.T,yp.T))

pop = (np.e)\*\*(beta[0]\*2021 + beta[1])

print("Population of Chandigarh in 2021 ", int(pop))

plt.plot(x, y, 'b.')

Y = (np.e)\*\*(beta[0]\*x+beta[1])

plt.plot(x, Y, 'r')

plt.xlabel('x')

plt.ylabel('y')

plt.show()

**Code for Problem 8 -**

#kshitij jaiswal

#b20cs028

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

lb = 1

ub = 20

def polynomial\_fit(x, y, p):

numPnts = len(x)

A = np.ones((numPnts,1),dtype=float)

for i in range(1, p+1):

A = np.column\_stack((x\*\*i, A))

beta=np.dot(np.linalg.inv(np.dot(A.T,A)),np.dot(A.T,y.T))

Y = beta[0]\*(x\*\*p)

for i in range(1, p+1):

Y += (beta[i]\*(x\*\*(p-i)))

# plt.plot(x, y, 'b.')

# plt.plot(x, Y, 'r')

# plt.xlabel('x')

# plt.ylabel('y')

# plt.show()

loss = 0.0

for i in range(numPnts):

loss += np.abs(Y[i] - y[i])

loss /= numPnts

return loss

def openCSV():

path = '2\_col.xlsx'

dt = pd.read\_excel(path)

return dt

if \_\_name\_\_ == '\_\_main\_\_':

dt = openCSV()

B=dt.values

x,y=B[:,0],B[:,1]

avgLoss = []

minLoss = 1e18

bestDeg = 0

for p in range(lb, ub+1):

lossVal = polynomial\_fit(x,y,p)

avgLoss.append(lossVal)

if lossVal < minLoss:

minLoss = lossVal

bestDeg = p

print("minimum average loss is " + str(minLoss) + " which is achieved at degree " + str(bestDeg))

plt.plot(list(range(lb, ub+1)), avgLoss, 'r')

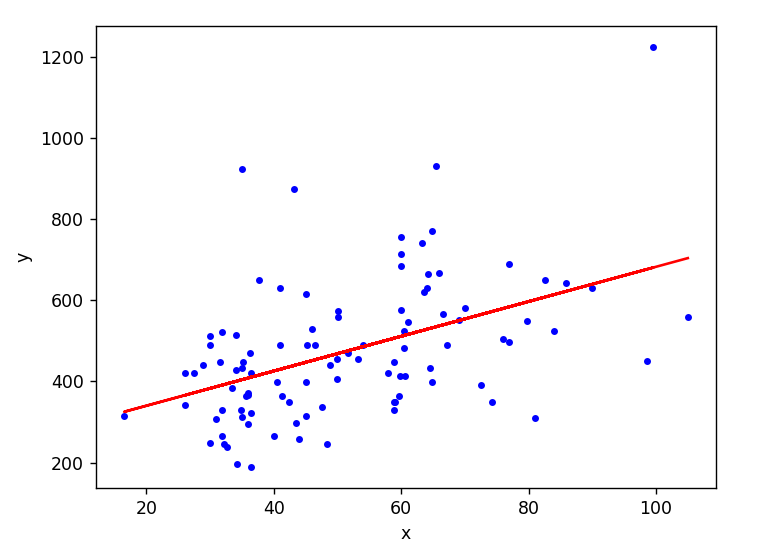
plt.xlabel('polynomial degree')

plt.ylabel('average absolute loss')

plt.show()

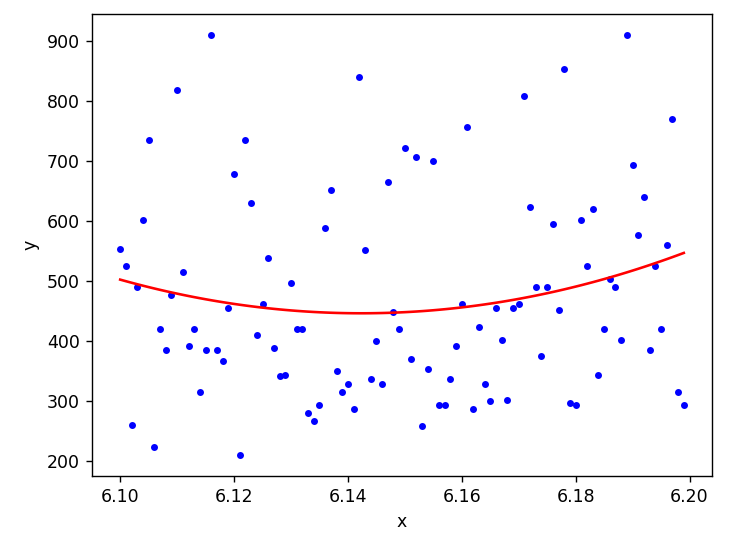
**Graphs -**

**Q1**

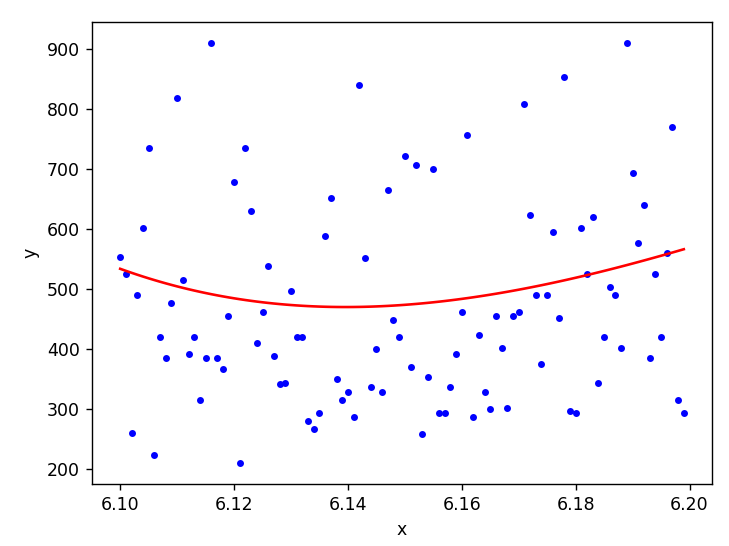
****

**Q2**

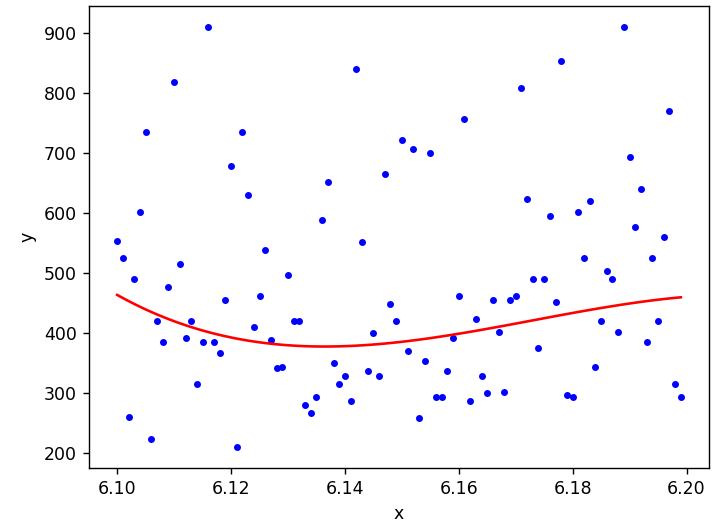
**n=2**

****

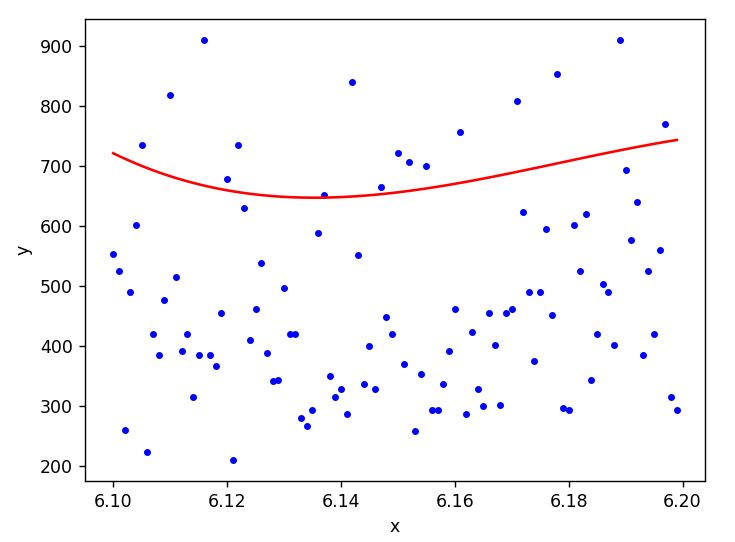
**n=3**

****

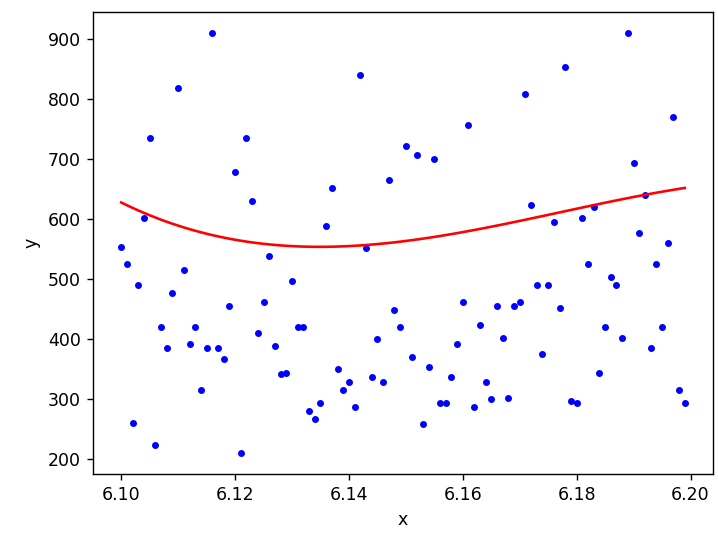
**n=4**

****

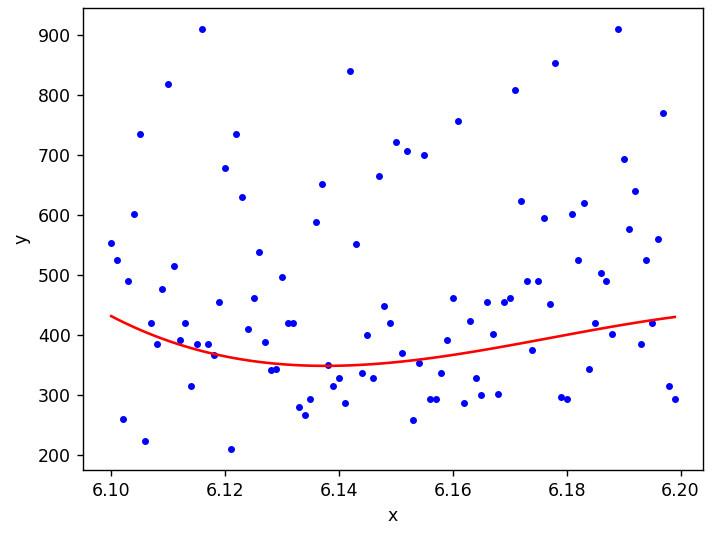
**n=5**

****

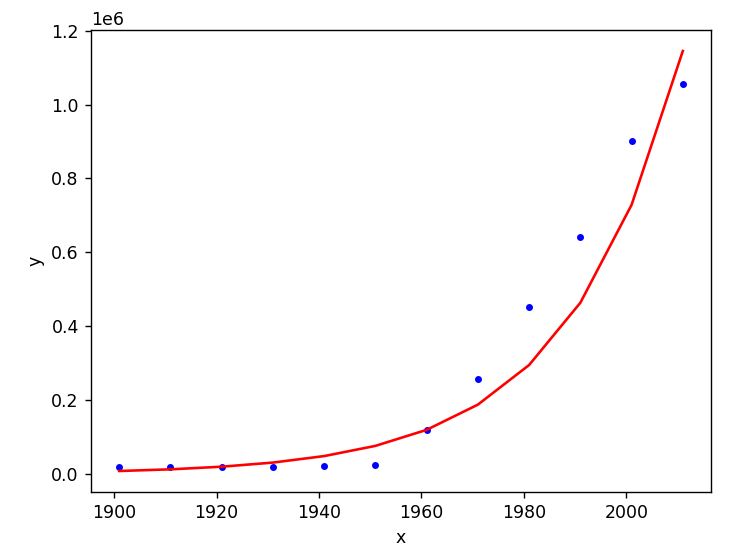
**n=6**

****

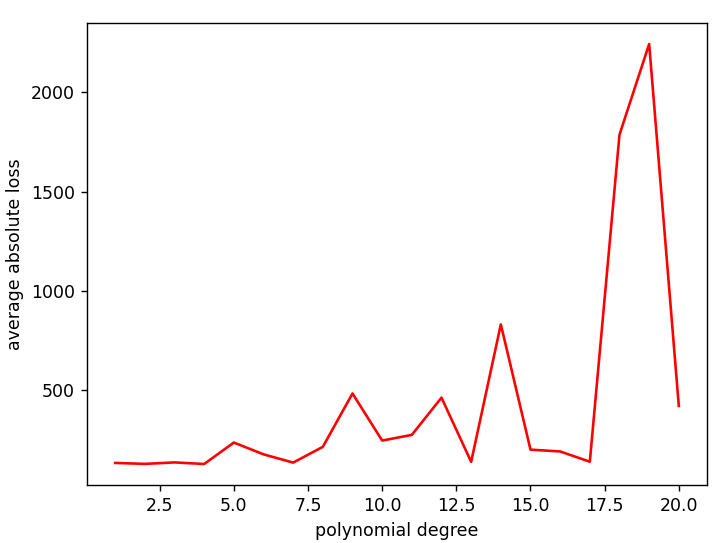
**n=7**

****

**Q7**

****

**Q8**

****