HW6程式採用python語言，以Jupiter notebook形式呈現

import math

import numpy as np

import matplotlib.pyplot as plt

def ReadData(dataFile):

    with open(dataFile, 'r') as f:

        lines = f.readlines()

        data\_list = []

        for line in lines:

            line = line.strip().split()

            data\_list.append([float(l) for l in line])

        dataArray = np.array(data\_list)

        return dataArray

def sign(n):

    if(n>=0):

        return 1

    else:

        return -1

def GetSortedArray(dataArray,i):

     # 根据dataArray第i列的值对dataArray进行从小到大的排序

    data\_list=dataArray.tolist()

    sorted\_data\_list=sorted(data\_list,key=lambda x:x[i],reverse=False)

    sortedDataArray=np.array(sorted\_data\_list)

    return sortedDataArray

def GetUZeroOneError(pred,dataY,u):

    return np.sum(u\*np.not\_equal(pred,dataY))/np.sum(u)

def GetZeroOneError(pred,dataY):

    return np.sum(np.not\_equal(pred,dataY))/dataY.shape[0]

def decision\_stump(dataArray,u):

    num\_data=dataArray.shape[0]

    num\_dim=dataArray.shape[1]-1

    min\_e=np.inf

    min\_s = np.inf

    min\_d=np.inf

    min\_theta = np.inf

    min\_pred = np.zeros((num\_data,))

    for d in range(num\_dim):

        sortedDataArray=GetSortedArray(dataArray,d) # 确保有效theta

        d\_min\_e=np.inf

        d\_min\_s = np.inf

        d\_min\_theta = np.inf

        d\_min\_pred = np.zeros((num\_data,))

        for s in [-1.0,1.0]:

            for i in range(num\_data):

                if(i==0):

                    theta=-np.inf

                    pred=s\*np.ones((num\_data,))

                else:

                    if sortedDataArray[i-1,d]==sortedDataArray[i,d]:

                        continue

                    theta=(sortedDataArray[i-1,d]+sortedDataArray[i,d])/2

                    pred=np.zeros((num\_data,))

                    for n in range(num\_data):

                        pred[n]=s\*sign(dataArray[n,d]-theta)

                d\_now\_e=GetUZeroOneError(pred,dataArray[:,-1],u)

                if(d\_now\_e<d\_min\_e):

                    d\_min\_e=d\_now\_e

                    d\_min\_s=s

                    d\_min\_theta=theta

                    d\_min\_pred=pred

        if(d\_min\_e<min\_e):

            min\_e=d\_min\_e

            min\_s=d\_min\_s

            min\_d=d

            min\_theta=d\_min\_theta

            min\_pred=d\_min\_pred

    return min\_s,min\_d,min\_theta,min\_pred,min\_e

def Pred(paraList,dataX):

    # paraList=[s,d,theta]

    num\_data=dataX.shape[0]

    pred=np.zeros((num\_data,))

    for i in range(num\_data):

        pred[i]=paraList[0]\*sign(dataX[i,paraList[1]]-paraList[2])

    return pred

def plot\_line\_chart(X=np.arange(0,500,1).tolist(),Y=np.arange(0,500,1).tolist(),nameX="t",nameY="Ein(gt)",saveName="12.png"):

    plt.figure(figsize=(30,12))

    plt.plot(X,Y,'b')

    plt.plot(X,Y,'ro')

    plt.xlim((X[0]-1,X[-1]+1))

    for (x,y) in zip(X,Y):

        if(x%10==0):

            plt.text(x+0.1,y,str(round(y,4)))

    plt.xlabel(nameX)

    plt.ylabel(nameY)

    plt.title(nameY+" versus "+nameX)

    plt.savefig(saveName)

    return

dataArray=ReadData("hw6\_train.dat.txt")

dataY=dataArray[:,-1]

dataX=dataArray[:,:-1]

num\_data=dataArray.shape[0]

u=np.full(shape=(num\_data,),fill\_value=1/num\_data)

ein\_g\_list=[]

alpha\_list=[]

g\_list=[]

ein\_G\_list=[]

u\_sum\_list=[]

epi\_list=[]

min\_pred\_list=[]

# adaboost

for t in range(500):

    u\_sum\_list.append(np.sum(u))

    min\_s,min\_d,min\_theta,min\_pred,epi=decision\_stump(dataArray,u)

    g\_list.append([min\_s,min\_d,min\_theta])

    min\_pred\_list.append(min\_pred)

    ein\_g=GetZeroOneError(min\_pred,dataY)

    ein\_g\_list.append(ein\_g)

    epi\_list.append(epi)

    para=math.sqrt((1-epi)/epi)

    alpha\_list.append(math.log(para))

    for i in range(num\_data):

        if min\_pred[i]==dataY[i]:

            u[i]/=para

        else:

            u[i]\*=para

    predG=np.zeros((num\_data,))

    for ta in range(t):

        predG+=alpha\_list[ta]\*min\_pred\_list[ta]

    for n in range(num\_data):

        predG[n]=sign(predG[n])

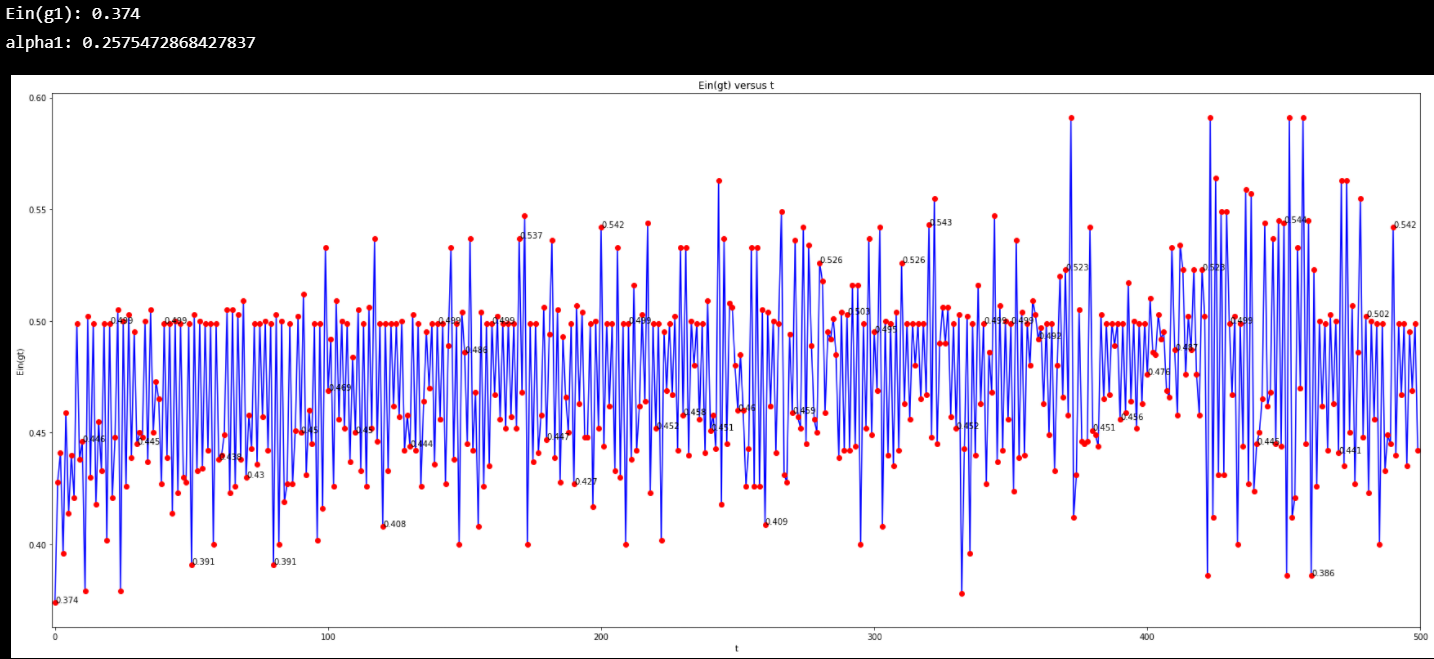
ein\_G\_list.append(GetZeroOneError(predG,dataY))

#Problem 11 and Problem 12

plot\_line\_chart(Y=ein\_g\_list)

print("Ein(g1):",ein\_g\_list[0])

print("alpha1:",alpha\_list[0])



# Problem 13

plot\_line\_chart(Y=ein\_G\_list,nameY="Ein(Gt)",saveName="14.png")

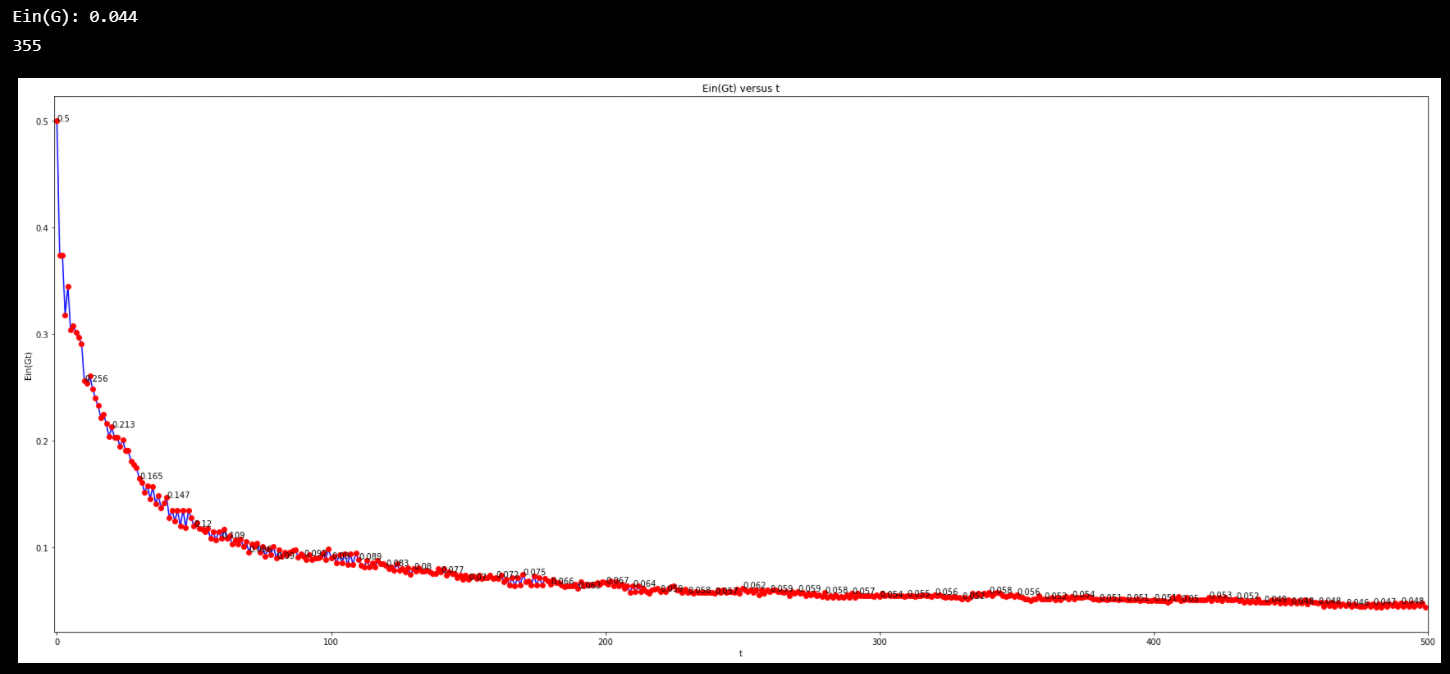
print("Ein(G):",ein\_G\_list[-1])

for i in range(500):

    if ein\_G\_list[i]==0.05:

        print(i)

        break



#計算Eout

testArray=ReadData("hw6\_test.dat.txt")

num\_test=testArray.shape[0]

testX=testArray[:,:-1]

testY=testArray[:,-1]

pred\_g\_list=[]

eout\_g\_list=[]

eout\_G\_list=[]

eout\_Guni\_list=[]

for t in range(500):

    pred\_g=Pred(g\_list[t],testX)

    pred\_g\_list.append(pred\_g)

    eout\_g\_list.append(GetZeroOneError(pred\_g,testY))

    pred\_G=np.zeros((num\_test,))

    pred\_Guni=np.zeros((num\_test,))

    for ta in range(t):

        pred\_G+=alpha\_list[ta]\*pred\_g\_list[ta]

    for ta in range(t):

        pred\_Guni+=pred\_g\_list[ta]

sign\_ufunc=np.frompyfunc(sign,1,1)

pred\_G=sign\_ufunc(pred\_G)

pred\_Guni=sign\_ufunc(pred\_Guni)

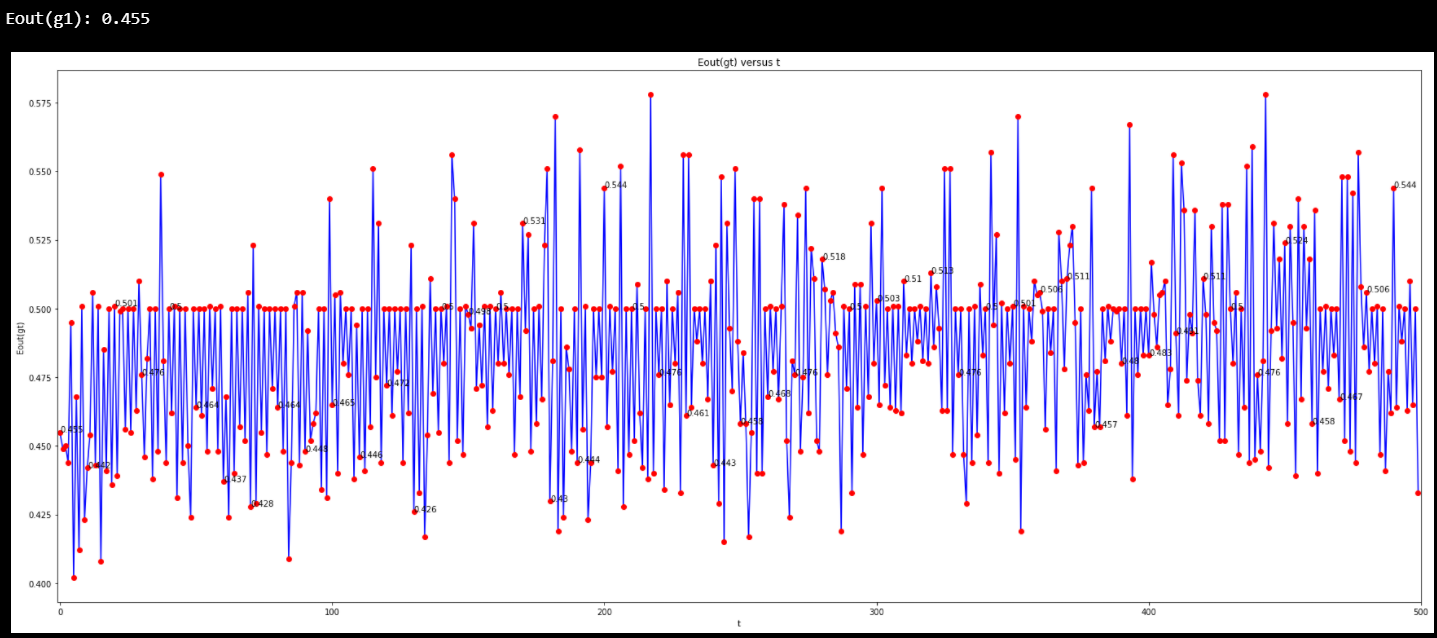
eout\_G\_list.append(GetZeroOneError(pred\_G,testY))

eout\_Guni\_list.append(GetZeroOneError(pred\_Guni,testY))

#Problem 14

plot\_line\_chart(Y=eout\_g\_list, nameY="Eout(gt)", saveName="17.png")

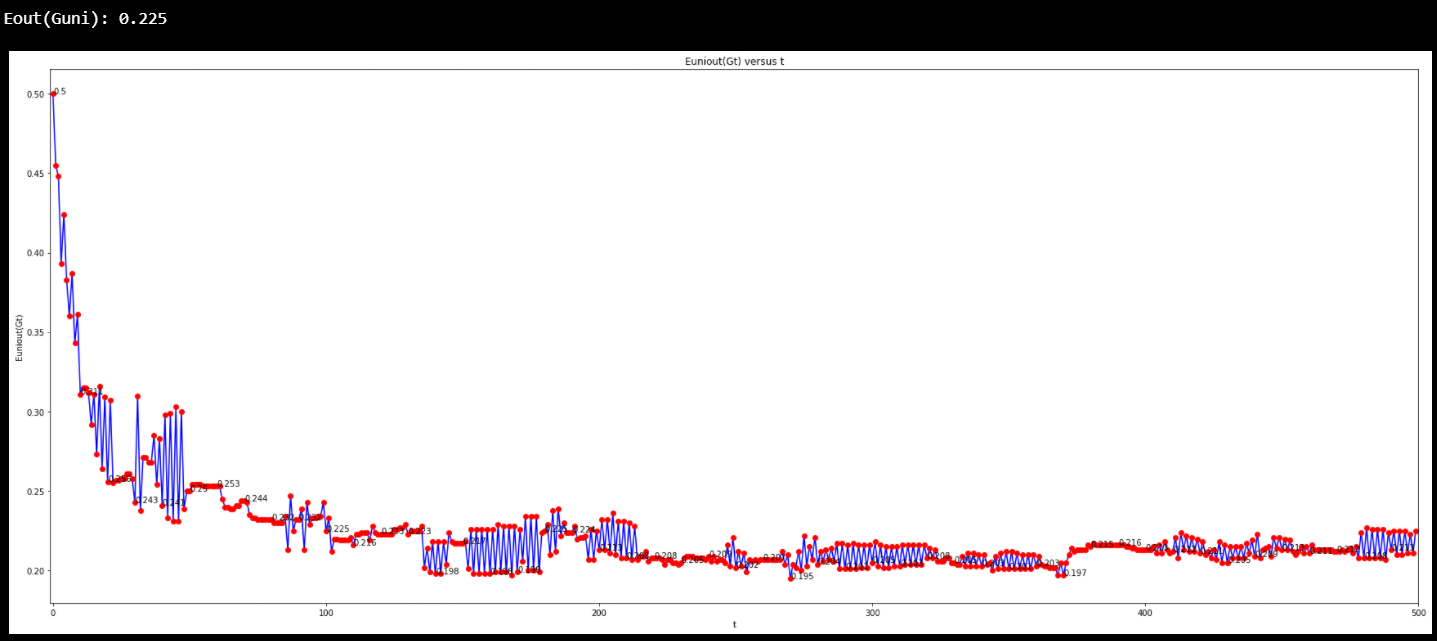
print("Eout(g1):",eout\_g\_list[0])



#Problem 15

plot\_line\_chart(Y=eout\_Guni\_list, nameY="Euniout(Gt)", saveName="19.png")

print("Eout(Guni):",eout\_Guni\_list[-1])



#Problem 16

plot\_line\_chart(Y=eout\_G\_list, nameY="Eout(Gt)", saveName="18.png")

print("Eout(G):",eout\_G\_list[-1])

