Assignement Number 4
20162007
Nishu Vats

TASK 1

Program:

```
#!/usr/bin/python

from scipy.misc import imresize
from random import seed
from random import randrange
from random import random
from csv import reader
from math import exp
from copy import deepcopy
import numpy as np
from sklearn.model_selection import KFold



# Calculate neuron activation for an input
def net(weights, inputs):
        activation = weights[-1]
        for i in range(len(weights)-1):
                activation += weights[i] * inputs[i]
        return activation


# Train a network for a fixed number of epochs and break when error is very less
def training(network, train):
        n_epochs=1000
        for n in range(n_epochs):
                sum_error=0
                for row in train:
                        intermediate_inputs = []
                        for j in range(len(network[0])):
                                activation = net(network[0][j]['weights'], row)
                                network[0][j]['output'] = 1.0 / (1.0 + exp(-activation))
                                intermediate_inputs.append(network[0][j]['output'])

                        outputs = []
                        for j in range(len(network[1])):
                                activation = net(network[1][j]['weights'], intermediate_inputs)
                                network[1][j]['output'] = 1.0 / (1.0 + exp(-activation))
                                outputs.append(network[1][j]['output'])

                        hotEncoding = [0 for i in range(n_outputs)]
                        hotEncoding[row[-1]] = 1
                        for i in range(len(hotEncoding)):
                                sum_error += (hotEncoding[i]-outputs[i])**2

                        for j in range(len(network[1])):
                                network[1][j]['delta'] = (hotEncoding[j] - network[1][j]['output']) * (network[1][j]
['output']* (1.0 - network[1][j]['output']))
```

```
                        for j in range(len(network[0])):
                                error = 0.0
                                for i in range(len(network[1])):
                                        error += (network[1][i]['weights'][j] * network[1][i]['delta'])
                                network[0][j]['delta'] = error * (network[0][j]['output']* (1.0 - network[0][j]
['output']))


                        for i in range(len(network[0])):
                                for j in range(len(row[:-1])):
                                        network[0][i]['weights'][j] += l_rate * network[0][i]['delta'] * row[:-1][j]
                                network[0][i]['weights'][-1] += l_rate * network[0][i]['delta']

                        inputs = [network[0][j]['output'] for j in range(len(network[0]))]
                        for i in range(len(network[1])):
                                for j in range(len(inputs)):
                                        network[1][i]['weights'][j] += l_rate * network[1][i]['delta'] * inputs[j]
                                network[1][i]['weights'][-1] += l_rate * network[1][i]['delta']
                if(sum_error<4):
                        break

def print_network_weight(network):
        for i in range(n_hidden):
                print network[0][i]["weights"]
        for i in range(n_outputs):
                print network[1][i]["weights"]

# Backpropagation Algorithm With Stochastic Gradient Descent
def neural_network(train, test):
        network = list()
        hidden_layer = [{'weights':[random() for i in range(n_inputs + 1)]} for i in range(n_hidden)]
        network.append(hidden_layer)
        output_layer = [{'weights':[random() for i in range(n_hidden + 1)]} for i in range(n_outputs)]
        network.append(output_layer)

# Train the network using train set
        training(network, train)

#printing network weights after learning is over
        print_network_weight(network)

#make predictions on test data
        predictions = list()
        for row in test:
                intermediate_inputs = [] #ouputs of hidden layer
                for j in range(len(network[0])):
                        activation = net(network[0][j]['weights'], row)
                        network[0][j]['output'] = 1.0 / (1.0 + exp(-activation))
                        intermediate_inputs.append(network[0][j]['output'])

                outputs = [] #outputs of output layer
                for j in range(len(network[1])):
                        activation = net(network[1][j]['weights'], intermediate_inputs)
                        network[1][j]['output'] = 1.0 / (1.0 + exp(-activation))
                        outputs.append(network[1][j]['output'])

                maxi=0 # predicts the class corresponding to node giving maximum output
                for j in range(len(outputs)):
```

```
                        if(outputs[j]>=outputs[maxi]):
                                maxi=j
                predictions.append(maxi)
        return(predictions)

##############################
##############################
##      MAIN        ##
##############################
##############################

#####################################################################
#   collects data from file digitises and downscales it     #
#####################################################################

seed(1)
fname = 'optdigits-orig.tra'
t=0
im=list()
newimg=list()
dataset= list()
j=0
i=0
str1=""
with open(fname, 'r') as file:
        csv_reader = reader(file)
        for row in csv_reader:
                if not row:
                        continue
                row=list(row[0].strip())
                row = [(int)(x.strip()) for x in row]
                if t==32:
                        if(row[0]==5 or row[0]==1 or row[0]==2):
                                im=np.array(im)
                                newimg=imresize(im,(8,8)) #Downscaled data
                                newimg = np.reshape(newimg, (1,np.product(newimg.shape)))
                                final_img=newimg[0].tolist()
                                label=deepcopy(row[0])
                                for i in range(len(final_img)): #digitizing data to grayscale
                                        if(final_img[i]>0):
                                                final_img[i]=final_img[i]/16
                                        if(final_img[i]<6):
                                                final_img[i]=0
                                        else:
                                                final_img[i]=1
                                final_img.append(label)
                                dataset.append(final_img)
                        t=0
                        im=list()
                        newimg=list()
                else:
                        im.append(row)
                        t=t+1
lengthOfDataset= len(dataset[0])

for column in range(lengthOfDataset-1):
        for row in dataset:
                row[column] = float(row[column])
```

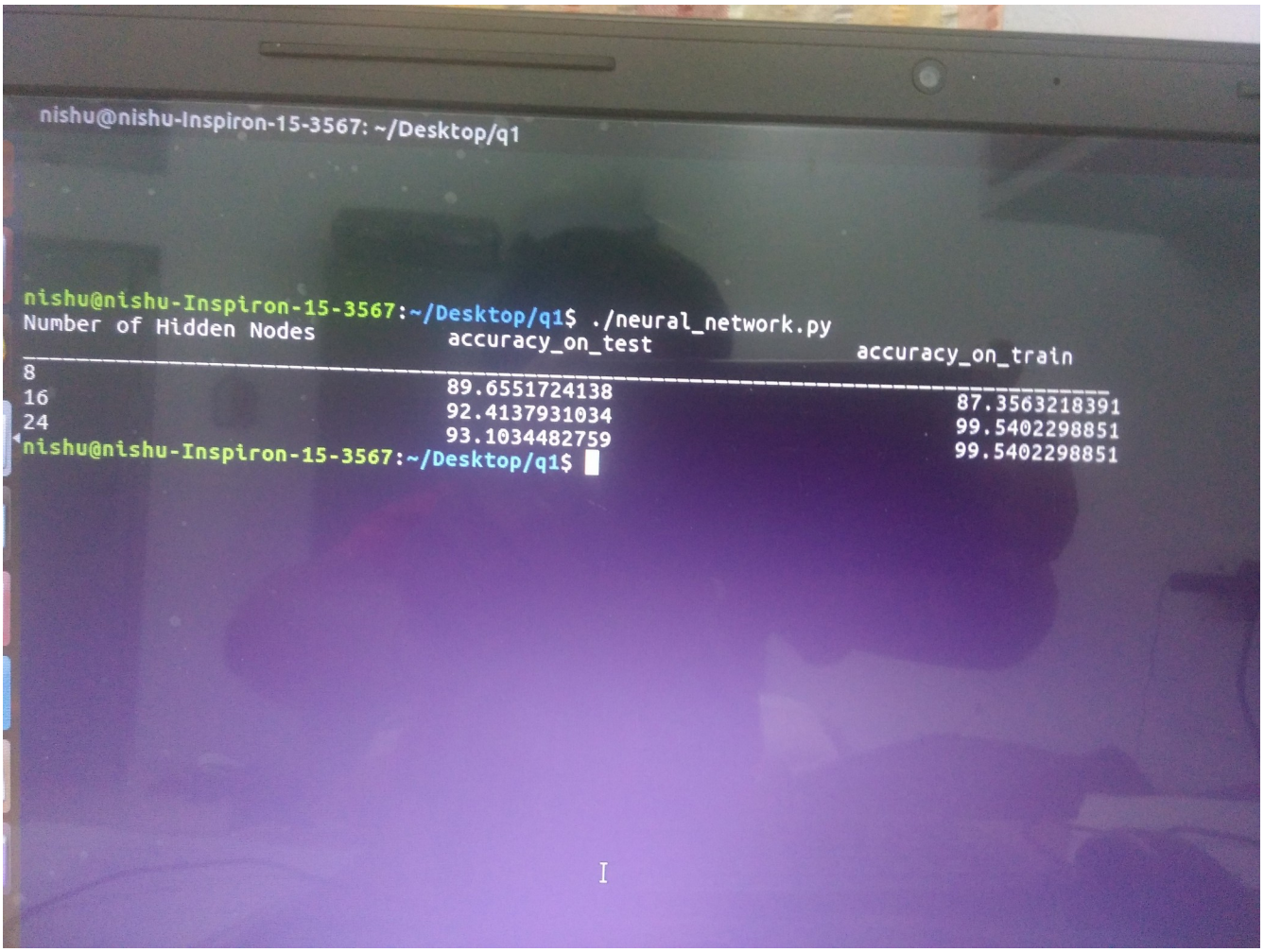
```
#####################################################
#     running neural network for digits 5,2,1     #
#####################################################
for row in dataset:
        if row[lengthOfDataset-1] == 5:
                row[lengthOfDataset-1]=0
        elif row[lengthOfDataset-1]==2:
                row[lengthOfDataset-1]=2
        elif row[lengthOfDataset-1]==1:
                row[lengthOfDataset-1]=1

n_folds = 5
l_rate = 0.3

########################################
#    Running kfold neural network      #
########################################
n_hidden_units = [8,16,24]
for n_hidden in n_hidden_units:
        folds = KFold(n_splits=4)
        scores = list()
        final_accuracy=0
        for train_set , test_set in folds.split(dataset):
                x_tr=[]
                x_test=[]
                for i in test_set:
                        x_test.append(dataset[i])
                for i in train_set:
                        x_tr.append(dataset[i])
        n_inputs = len(x_tr[0]) - 1
        n_outputs = len(set([row[-1] for row in x_tr]))
        predicted_test = neural_network(x_tr, x_test)
        predicted_train = neural_network(x_tr, x_tr)
        actual_test = [row[-1] for row in x_test]
        actual_train= [row[-1] for row in x_tr]
        correct_test = 0
        correct_train = 0
        for i in range(len(actual_test)):
                if actual_test[i] == predicted_test[i]:
                        correct_test += 1
        accuracy_test= correct_test / float(len(actual_test)) * 100.0

        correct_train = 0
        for i in range(len(actual_train)):
                if actual_train[i] == predicted_train[i]:
                        correct_train += 1
        accuracy_train= correct_train / float(len(actual_train)) * 100.0
        print "Number of Hidden Nodes\t\taccuracy_on_test\t\taccuracy_on_train"
        print "_____________________________________________________________________________"
        print str(n_hidden)+"\t\t\t"+str(accuracy_test)+"\t\t\t"+str(accuracy_train)
```

Output:

Number of Hidden Nodes	accuracy_on_test	accuracy_on_train
8	89.6551724138	87.3563218391

Number of Hidden Nodes	accuracy_on_test	accuracy_on_train
16	92.4137931034	99.5402298851

Number of Hidden Nodes	accuracy_on_test	accuracy_on_train
24	93.1034482759	99.5402298851



Observation:

Hidden nodes map non separable data to plane where data is more separable for better feature detection.
When number of hidden nodes are very less, after the mapping of non-separable data to new plane the separabilty is very less hence accuracy of prediction is less.
Separabilty generally increases with number of hidden nodes and hence accuracy generally increses with number of hidden nodes
But after a point,increase in number of hidded nodes cause overfitting and finally decreases the accuracy of predicion.

We can see that when number of hidden nodes is 8, accuracy on test_set is less.
Accuracy on test_set increases when number of nodes is 16.
Accuracy again increases when number of Hidden nodes is 24 .
Accuracy again decreased when number of Hidden nodes is 32 .

Part 1:

Implemented a simple 3-layer feed-forward neural network for Multi-class classification problem.
Used sigmoid activation function.
Implemented back-propagation for training the parameters.

Part 2:

Trained this network to learn a 3-class classifier for optical character recognitionfor any three digits between 1,2 and 5.

Part 3:

Used any three digits between 1,2 and 5 from the optdigits data set that comes from the UCI Machine Learning Repository.

Part 4:

Digitized and down-sampled images to 8 × 8.

Part 5:

Tried few configurations for a 3-Layer Neural Network with 8,16 and 24 number of hidden units and three output units and reported learned weights from backpropagation done using training data.

Part 6:

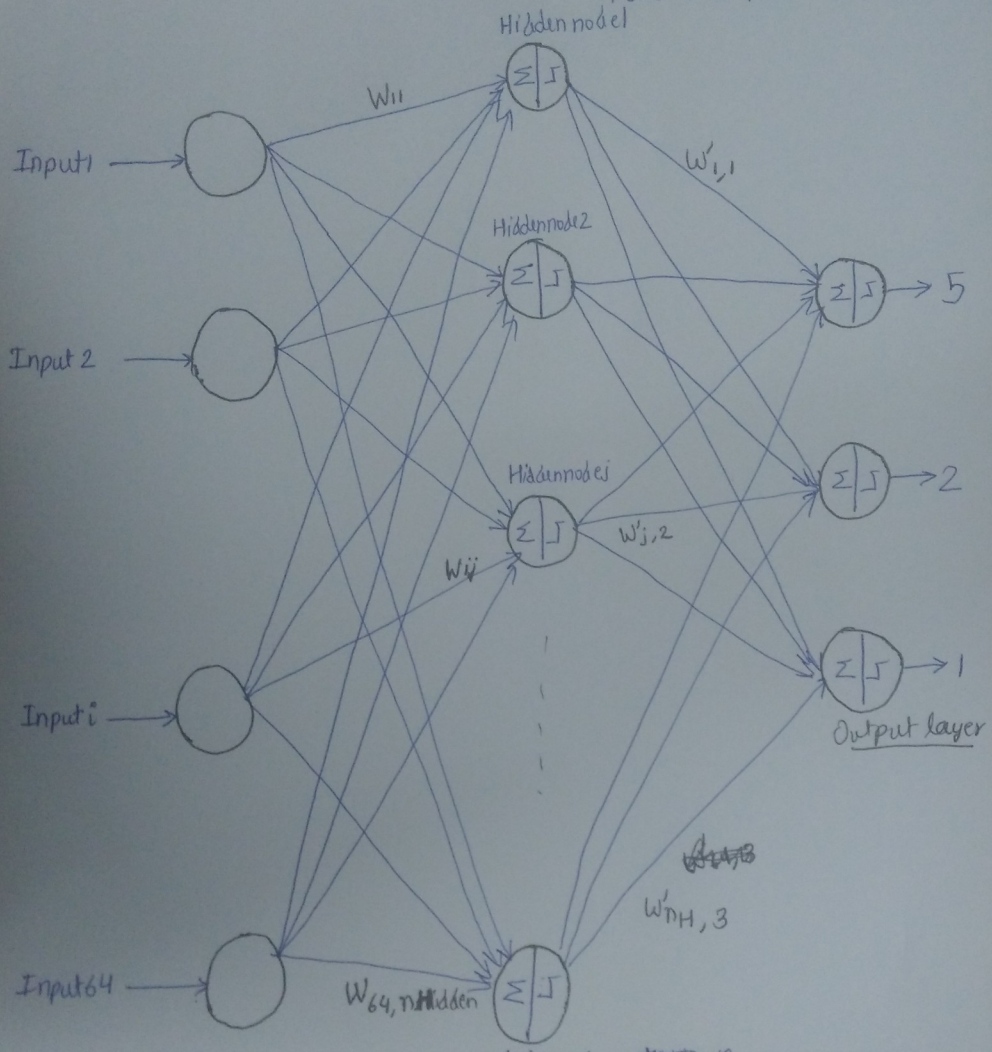
Drawn representative neural network architecture for number of hidden units as nH.

Σ - Weighted Summation

∫ - Sigmoid function.

W = Weight between input node to hidden layer node

W' = Weight between hidden layer node to output node.



Weights for nH=8:

[0.13436424411240122, 0.9675612111175075, 1.7038127117440915, -0.30741290476981675,
1.2744286333458892, 1.1301037045994302, 1.7893026558376002, 0.7887233511355132,
0.0938595867742349, 1.624148899366469, 2.977148080159181, 1.0361235024466762,
2.4806930214269944, -0.188543162413509, 0.5834847185612136, 0.7215400323407826,
0.22876222127045265, 0.11517685476395274, 1.4907991710260673, -2.029659114958099,
-1.6047439028379622, -1.2784067576658062, 0.9111403188478212, 0.38120423768821243,
0.21659939713061338, -1.5941343124197063, -0.9883972273213583, -2.5090557321767224,
-0.022548356373244982, -0.39225244723402203, 0.20873067343152021, 0.2308665415409843,
0.2187810373376886, 0.1368994571853648, -0.13805808942129275, -1.1211516714457692,
-1.555275026230815, -0.010074942546424397, 1.0062618630044022, 0.1859062658947177,
0.9925434121760651, 0.8614591119547264, -1.249570304070736, -0.8472870765383039,
-1.9099195254319645, -0.14476843948631413, -0.010716285106132492, 0.4221069999614152,
0.830035693274327, 0.9170466466082814, -0.7089500753612571, 0.6518430839170162,
1.4588377030997015, -0.22545784062690072, 0.6141496415868823, 0.13117762254733953,
0.034525830151341586, 0.24420988120040546, 1.3306968547811853, 0.7698401572376831,
-1.5348394834944554, -1.585539943093888, 0.902679809427722, -1.823047599795524,
0.8466949571019498]
[0.4389616300445631, 0.5084288095945279, 0.7776678210216198, 0.5180620432046352,
0.38958686592731784, 0.4880620534753652, 0.02920882672198378, 0.04348729035652743,
0.703382088603836, 0.9829489019897498, 0.5915360366907321, 0.3905247089028714,
0.16756017321482974, 0.501100105432119, 0.9819205612076835, 0.7705231398308006,
0.5396174484497788, 0.8601903203389643, 0.22886059931476524, 0.5102956219934973,
0.9501777359478738, 0.5774378098064765, 0.4591264583480534, 0.2692794774414212,
0.5479963094662489, 0.9563602876061131, 0.0019169087906687127, 0.7796382234910078,
0.817958858734931, 0.8850281295139122, 0.7405031248847186, 0.8091399008724796,
0.518678283523002, 0.5611547012952025, 0.4247217677417115, 0.05367417813322605,
0.8674214531887582, 0.5688957160116357, 0.19975803167487746, 0.5047204674288633,
0.48492511222773416, 0.35678988849383453, 0.34559232080761565, 0.537628695914962,
0.6210936089585464, 0.610708507688342, 0.4580953059237525, 0.027974984083842358,
0.22960503127702392, 0.17717405020782762, 0.5834648920222615, 0.8594913305419115,
0.7953513424571339, 0.7947030465457925, 0.8163187086751786, 0.255290776774212,
0.841744832274096, 0.673118606731952, 0.08288082029682128, 0.01392039588738136,
0.010565790596548875, 0.7534061374149313, 0.24923704777255334, 0.10945713073578293,
0.6213160369904106]
[0.3444228640964949, 0.07015332613509627, 0.1944623527716401, 0.5109394359088353,
0.11350430554090651, 0.2589495013103149, 0.7158691986788727, 0.4547016300456639,
0.3220017663873259, 0.4794628921386036, 0.051124640296539264, 0.3576823087599482,
0.3648825241468158, 0.16094810613197966, 0.10459564188557458, 0.8998185003560202,
0.5101159809286764, 0.21362355380148673, 0.6051072817631843, 0.751276200252899,
-0.05495991199780693 4, -0.007389945564304287, 0.14597057344819322, 0.7188354727617898,
0.16022759262970465, 0.6875061123232633, 0.6391547723148715, 0.482445927018535,
0.15081302202646157, 0.9486956972046492, 0.7982703765227217, 0.516599516949393,
0.22319578024667075, 0.6403353860833089, 0.3695962936940395, 0.5223968619576539,
0.2631987363014478, 0.6200998385445138, 0.061291084720919715, 0.29860594962301334,
0.9679033101508892, 0.8755403593406303, 0.2934125569901424, 0.8224572503428003,
0.2539155753422701, 0.9258282023652998, 0.7459714557698797, 0.4161722627650255,
0.25235810227983535, 0.012333118117667234, 0.8846508381793406, 0.014971235441369368,

0.768590068835466, 0.9467808164641649, 0.5718623980408781, 0.17106909297148187,
0.8677810644349934, 0.9740730146620111, 0.7411713285037237, 0.4921852345181696,
0.3239413878377732, 0.3111045023329195, 0.21009048092652313, 0.6736427476677392,
0.38079725418325205]
[0.1941186449851896, 0.10448201770900623, 0.662865902746389, 0.29304780804185937,
0.4974197024094059, 0.3230075091063551, 0.8713311714920458, 0.8996782696347811,
0.018092983640471738, 0.1997797931379023, 0.324377213882564, 0.9839522041748439,
0.780925243142637, 0.33741006829487435, 0.21271733721907868, 0.6744550697237632,
0.8377010701539643, 0.93083188024002, 0.3408970904309201, 0.8801284274284761,
0.6867864389558026, 0.4836441421830135, 0.9855035242537493, 0.23464043487103847,
0.7254651862412724, 0.08374979256052979, 0.16765263696289492, 0.9079563877528294,
0.2113227102853302, 0.7578908176975788, 0.6002049598549781, 0.8411321957058551,
0.3681079994056491, 0.34023849894197444, 0.2902982929012748, 0.8654818333980288,
0.6025227632520678, 0.9530781088365138, 0.8872039443037434, 0.13534597739545295,
0.5511704740692165, 0.10427480762387731, 0.03858994291877345, 0.0717605135566007,
0.864920118577296, 0.7865523702356663, 0.8284383909399763, 0.3408974641165834,
0.6151860325590366, 0.7818651934379813, 0.3761002651090923, 0.5686860067656672,
0.22049702312222522, 0.07877696940911003, 0.26650931838019587, 0.8907663710824744,
0.5644468332401974, 0.9250823442573959, 0.45518805583704064, 0.2740375393547812,
0.7836204549216013, 0.8251445507045156, 0.01144673383257946, 0.6705021421260264,
0.08822315034721286]
[0.1151024984279273, 0.8842481180768575, 0.0007568296791108083, 0.1983966501846446,
0.9700754958038489, 0.4241708280862498, 0.1169529820044098, 0.16738343746133177,
0.24142028509784308, 0.7254016192733974, 0.06159517048729356, 0.8712545000453693,
0.3542819908060024, 0.9684874131204275, 0.9088925458399085, 0.29402358494854774,
0.2534101360411267, 0.46421795757922857, 0.0810653625445845, 0.6252072227347519,
0.009425325827363855, 0.011184131264239134, 0.9825843022162504, 0.2955498600489178,
0.5965706431884413, 0.44482197143411994, 0.32234376633404693, 0.03679584148938767,
0.8862505042018887, 0.9689081008867704, 0.9699373115799624, 0.1113623101268919,
0.21519327003609845, 0.6179451316565858, 0.9785456241998566, 0.5045909375045534,
0.6805554437630312, 0.6641733029696223, 0.25976566785158545, 0.5416022629129655,
0.3073211178125135, 0.2463985044277015, 0.04989480809435108, 0.2381826734136139,
0.9853576845783832, 0.4514786206046566, 0.6528209092969887, 0.6434660802698416,
0.940734522249, 0.3863962670394718, 0.2614875851216532, 0.28680464607808703,
0.27855389853905155, 0.8184138672090515, 0.8864557531249243, 0.3030806040337472,
0.33433340565076186, 0.5442122989695838, 0.5542644678650714, 0.5583535842412041,
0.20638458341576726, -0.027125999892502722, 0.2191675353034909, 0.07284781227443574,
0.512537839312038]
[0.07091636753953445, 0.07496208842195957, 0.6295374686065147, 0.2836134250981673,
0.7898915138411758, 0.4957202982608688, 0.8633654268350514, 0.15417959616284405,
0.5014295859466933, 0.792966175632261, 0.07022168677884512, 0.9415872509006705,
0.16579615123742542, 0.7748171521375624, 0.9848803846849206, 0.8215501447435144,
0.3197840027930057, 0.10432749148850075, 0.5108499028414845, 0.9172507301078788,
0.28485618525226664, 0.8912789395308548, 0.14166121128999704, 0.9104816743927341,
0.03175994589733666, 0.31510912634868415, 0.9041021417948565, 0.8031368896045521,
0.9003376507637061, 0.8386367690630268, 0.7463064293070701, 0.6895951793002646,
0.1781548656443236, 0.4326641034892469, 0.15822730183032183, 0.710224471270795,
0.6621003341294791, 0.25207114644456813, 0.06495432648489059, 0.9633858833215757,
0.8082526283723965, 0.549272206513154, 0.5388731887517485, 0.8430648683787519,

0.44778390538969076, 0.3957338709493839, 0.33928273107909807, 0.2579690924717717,
0.024408502825104206, 0.646339668322516, 0.4109709014374406, 0.5622614407926233,
0.05348571326197687, 0.34691344847530897, 0.13232864265255012, 0.1250800715542617,
0.259112968915828, 0.8289445419587171, 0.3917496752587668, 0.39448037111831064,
0.6049431878115118, 0.22385536656838992, -0.001028397975827128, 0.5285021293272488,
0.49299456579352646]
[0.6488395923408533, 0.43829896983289135, 0.6840233517800144, 0.7339008641923881,
0.24399766015016341, 0.49677542859792007, 0.4785519560717145, 0.225062085038767,
0.4122461329173408, 0.5600135279250382, 0.9045015197500843, 0.9209921914129157,
0.28130803640594426, 0.6484869216825132, 0.04824101458590677, 0.07155138822789708,
0.5116917092002066, 0.877137304860212, 0.15994155173219743, 0.7733259186640733,
0.8908321591949637, 0.3136088459892707, 0.6926698328122978, 0.8489911224865752,
0.3716143307475649, 0.702353367315195, 0.7405369806008654, 0.6016955961435296,
0.863082997327405, 0.898359775489105, 0.9600653142959845, 0.5712326942175455,
0.17627589520647535, 0.25159776174253345, 0.22073827293800188, 0.5750112347963287,
0.7641115546001784, 0.05341957179834087, 0.6815828977528927, 0.7171532633675107,
0.3479815079568077, 0.5150556390222184, 0.16556438739093146, 0.7336239731375389,
0.047208326864904476, 0.9827240247122027, 0.8079249534150208, 0.6284485019821408,
0.2675262446471117, 0.9127946943261489, 0.9585657404672594, 0.14193882376855108,
0.781571229962485, 0.8432841129580866, 0.6593498882596912, 0.7004053177036595,
0.44505873211451163, 0.9243018894869062, 0.9689042527401686, 0.38511782780772613,
0.8082575768763169, 0.4352266467652895, 0.16442078704039703, 0.32545879320044296,
0.1320895108725339]
[0.9088847599027046, 0.5777748406726019, -0.9072308371241327, -0.3342613150321725,
-1.6107470528149912, -2.1359711590056385, -2.7490820825769764, 0.2482163710806481,
0.7495768111897567, 0.7461810575036187, 0.3150478841667201, -0.43291561314515287,
-0.5808512419236092, -1.6056241293288755, 0.8686505316094658, 0.8353323508828638,
0.20660581568518466, 0.10390690812704763, -0.3493082651697569, -2.8344341199009824,
3.3444150617642023, 0.7072134304139497, -0.9137279795920479, 0.7910907183680019,
0.8086546201638074, -1.3887297532488379, -3.3163608168716063, -7.528399897491552,
0.8982410723614569, -1.0093983910153081, -3.620971961869591, 0.3832563847662638,
0.2840474457335592, -0.1786875632401907, -1.3953632252631591, -0.38537043013366257,
0.7138668694467113, 0.44225592643392697, -6.1110751582394185, 0.7610656598531885,
0.9735197845800538, 0.08373041286519191, 2.1888881670791918, 1.2940163389615058,
-3.730883490335234, -1.3022691774082573, -2.6845285880088117, 0.5283939713514435,
0.0008447179488895173, 0.7263629636245373, 0.3312099110134169, 1.4180286737094683,
3.3326886459867513, 0.053360919314904984, 1.9572211874194618, -2.3632186603846317,
0.6476682418421831, -0.06971226422346236, 1.6312567975508718, -2.9275559871328825,
-0.6457386745221044, 2.1670795938214815, 2.77958131331284, 0.5585996491764542,
6.129858833661542]
[7.330628813541239, -0.6728483685701596, -0.25297425488773967, -0.7099538613817684,
-0.3605085893970296, -0.12666926535674494, -0.8111942955848142, -25.05749236592839,
-0.4985739238066743]
[-7.865191673246608, 0.41735118875386873, 0.013571609450760459, 0.4673762717600694,
0.5639007457209273, 0.48920842286038335, 0.9096613992543957, -2.7680227920494342,
0.7944295094838397]
[-0.29152186292756704, -2.4415606326994093, -2.732076958375723, -2.4707871166091477,
-2.6183855186192004, -2.573056803969243, -2.5698916514427395, 25.89712345381046,
-2.5739359678834 38]

Weights for nH=16:
[0.5491230721259409, 0.05381368498259822, 0.4962272073190437, 0.17362900973956646,
0.22037493372097652, 0.4295538988016091, 0.5436526381465532, 0.25041213288033703,
0.27093438017989424, 0.5301900687918419, 0.4614862660901011, 0.40411543226235946,
0.11235692915930932, 0.3762085449390508, 0.6541935632169271, 0.5441989404197219,
0.5447527137062477, 0.8435916582923236, 0.716838006916829, 0.6925831933205525,
0.04593856915246313, 0.3129555225385244, 0.682446042176694, 0.1557727759655333,
0.9134730441814907, 0.1441269618565545, 0.8826016230998798, 0.2213139045711153,
0.8524559378298474, 0.8507587801363867, 0.33507806677341795, 0.8885923720325766,
0.15976779278818198, 0.8509219502072443, 0.38656889132434785, 0.44778069458501857,
0.12693449142134575, 0.6001119734109669, 0.2684756769264653, 0.6668793014210087,
0.7993879440342788, 0.6036743607497893, 0.010683445321130866, 0.9602596904475142,
0.9300131086489828, 0.6416844538542184, 0.3780235144218427, 0.5619137655369324,
0.8828120686199001, 0.458353205459923, 0.775698742116943, 0.6012575166324452,
0.4287654330586146, 0.9345020401354961, 0.40944060754432754, 0.6059385161609837,
0.05327428951253488, 0.47086031978682735, 0.026348402441669812, 0.7027330672762585,
0.00640321940991861, 0.048052936940374075, 0.11341564252529393, 0.13999837591850708,
0.5135407370058719]
[0.35628839992753547, 0.2709680162776969, 0.9841392446762876, 0.9103519718637688,
0.656138634239146, 0.801768616204405, 0.8195082964093924, 0.24517343884360088,
0.8082860605552211, 0.2400899277693268, 0.5629840951359333, 0.35911631121388177,
0.16076110259807086, 0.7776027501813486, 0.9163946203064909, 0.31369855569597016,
0.8797625357454809, 0.34633338036401035, 0.6575118539925056, 0.9965776405689185,
0.7746684048675094, 0.05680006438570135, 0.43489336208545876, 0.37630325823086663,
0.2939317953132611, 0.8161922346562316, 0.44068055537525896, 0.699499125926934,
0.6372451100495803, 0.519833707634627, 0.05600263862067609, 0.6730352499596889,
0.891383085497047, 0.17228977505540713, 0.6429349331669518, 0.4888454788197716,
0.34299084114812173, 0.7106356161700921, 0.9751370075832311, 0.021664682628631304,
0.897305758366094, 0.3832384440306281, 0.8346035573494665, 0.17681731587065747,
0.7180689825602385, 0.09955614104984865, 0.3355159274899977, 0.9699086794931329,
0.6566155049590191, 0.7845252169793638, 0.4625099356186453, 0.47290274411705,
0.4944705923678444, 0.7744131109623448, 0.723939859464673, 0.19378128897471317,
0.44060439168556, 0.5420503344817501, 0.5719862992657668, 0.9278724392911208,
0.8413692431470488, 0.1520171152173091, 0.3774201537700716, 0.1090415426341275,
0.02811287890161703]
[0.0745859588783212, 0.1832002891523455, 0.7683950772116227, 0.6679966569820421,
0.7958138458632232, 0.2851140286412408, 0.15450727658498978, 0.9721002692327158,
0.8260249130855136, 0.9485740753398436, 0.021118229157903727, 0.3979880491340081,
0.6350478610282626, 0.7357160757614157, 0.9126813748552579, 0.5377317942344237,
0.39079239958264134, 0.006851285789091736, 0.8031468641735569, 0.9807612916788341,
0.9095475374364804, 0.6631113946049625, 0.34246736599141636, 0.23915025648517396,
0.7750196869400034, 0.9352555496325353, 0.9559674412726186, 0.17235563907349294,
0.5857442683110979, 0.5124096148829701, 0.42738860127440126, 0.7944006922875018,
0.9357823842440698, 0.724488428297693, 0.6989795090894935, 0.6924009435184248,
0.6536167371693181, 0.5343737728975363, 0.24755884844391085, 0.7794770186017971,
0.11909343724707233, 0.6438869849594454, 0.38938755562838373, 0.5642185999866914,
0.640835298747325, 0.4756804673174634, 0.9778792121632218, 0.23919305039462202,
0.012168333089732086, 0.9553663469824465, 0.31582195310243616, 0.2814720617903365,

0.41680048066493575, 0.5961396497121334, 0.9877586863441924, 0.7075229928016432,
0.31832021303921443, 0.5347505096785525, 0.45108723056132927, 0.5025921571584951,
0.4183305684642465, 0.17156482671072795, 0.3990319883762337, 0.389134643002817,
0.20145501649091396]
[0.8169186732056046, 0.3591536761033838, 0.12162668507593206, 0.5506816392588036,
0.8513896749343094, 0.7881591689091912, 0.622411440782163, 0.7310380068460375,
0.3361145774153067, 0.13223835889355648, 0.22544816209209165, 0.33388110208353977,
0.273733434301789, 0.47114574335266635, 0.14880495446085001, 0.130261785975196,
0.2527238668942108, 0.18916811705040934, 0.7937188940415945, 0.5500492658803184,
0.1906105268301962, 0.42949142034045434, 0.8721811192312302, 0.5776121477722593,
0.5539142523743498, 0.39396814140217806, 0.21233293223836908, 0.6391113046932656,
0.07040015759969806, 0.7902126357763098, 0.05757668364856072, 0.7463473111792467,
0.38262914432029493, 0.6834441856010225, 0.5981647529312412, 0.12283285427967496,
0.5375745933943206, 0.0809655953493602, 0.24244934863907705, 0.38166891142299064,
0.2856711685837189, 0.6617593087576076, 0.9756923945840301, 0.33729552512018385,
0.8456768155547714, 0.23497236210616587, 0.7099076318183135, 0.3477203659126339,
0.5353633261603788, 0.0863840929252913, 0.80006496306724, 0.18972935398571822,
0.45390540632095433, 0.2757033706432404, 0.7977006725029889, 0.5927752781783059,
0.6151849357234862, 0.7547595786279222, 0.22542216755347336, 0.04167999003517767,
0.817053109979715, 0.2934186232195174, 0.7916530857172086, 0.9567518553579886,
0.6190048836598113]
[0.10329198921112503, 0.8540363592356099, 0.6380644555278513, 0.2476059194707907,
0.20681233941517813, 0.5081608935827167, 0.12272225029538011, 0.9060200824268411,
0.7078621924830589, 0.8198111342030746, 0.38822025706135116, 0.924261933094683,
0.13255718330081007, 0.7154635048714427, 0.25468532481836603, 0.0036316269465580 53,
0.12089146531089001, 0.20214644697699735, 0.7653451454033979, 0.3746420212118191,
0.478308294360518, 0.6120487043530762, 0.2675893158083297, 0.6384335843307868,
0.6715719302788205, 0.9209743993311136, 0.501355605968314, 0.8522650125265189,
0.9657266935530413, 0.7676522837941613, 0.42122837591937695, 0.2719797975866193,
0.09773187837962227, 0.8307558033845335, 0.12798456364131267, 0.5581409567003749,
0.45175177627870583, 0.04471231614635702, 0.21465374288671032, 0.8228965828576935,
0.5386596159811745, 0.9243965699567274, 0.9076712392253207, 0.09319327292921889,
0.6749833417167902, 0.042248203208757586, 0.4230533744843047, 0.44177494338744194,
0.956872732737817, 0.5955465673148702, 0.19243699516453805, 0.5101971539321527,
0.5206040024453507, 0.19765897268099056, 0.36084474631255287, 0.8774697394900788,
0.9814709257866746, 0.7768786678399712, 0.06918463326148162, 0.9075762706182814,
0.4579677824435123, 0.8335011029975897, 0.17811001505436522, 0.14773280853920914,
0.9060014801614771]
[0.28552344045904365, 0.04298218498783023, 0.5008948757551597, 0.9878673525393789,
0.8342974031618559, 0.3979172790114098, 0.9933206113586807, 0.7966701948025767,
0.8420658675763089, 0.6454838156402155, 0.3941492742282061, 0.9033856898173355,
0.4686240265085851, 0.9349245375961495, 0.5521973752057712, 0.9098574658614854,
0.47715640081037314, 0.4269478647640567, 0.588938069910245, 0.3162681056555189,
0.1453876200777236, 0.5896021295840216, 0.8509633993891845, 0.27777624924381694,
0.8650214121278488, 0.7869871181452224, 0.7772707548379497, 0.4144766996749044,
0.9949308103757256, 0.790242062596517, 0.5756763386005098, 0.11350996819836934,
0.5738154912706415, 0.014421812827433548, 0.9024085592167563, 0.33444225519406257,
0.36594731261919, 0.551379053397298, 0.6377228379116066, 0.5827270677250831,
0.4849252171533167, 0.6343555635680588, 0.8457644620384664, 0.4422357254664286,

0.4987931996946828, 0.8111791961178718, 0.003652169266166223, 0.1607104980189884,
0.32502993465104124, 0.21375952070953583, 0.8940970608788497, 0.14548383411886565,
0.10524901438216529, 0.31515416646256894, 0.5073967921424217, 0.8214684610600855,
0.9956510837481631, 0.8518699574729153, 0.6087688419116575, 0.03553478761678564,
0.06108975586916676, 0.6276038820059356, 0.8179414799365077, 0.26548330011546745,
0.9668523792060326]
[0.5503873026658288, 1.3798903570502705, 1.8058689818694909, 0.16331217283665664,
2.6714323875679895, 1.9775911038887073, 2.659084683116987, 0.08329304401782134,
0.282428939274216, 2.573045876206164, 6.169654901185215, 1.4691757888454011,
2.859132242162027, 0.881992793288241, 0.9615480562238871, 0.301322965230045,
0.8735096217006009, 0.10530187064745565, 0.6185395570133263, -2.3210329479378116,
-2.3792429317024024, -3.103901289982824, 0.7844703479055171, 0.13325329151192067,
0.4422243447164529, -0.8989976245507058, -1.119935944207069, -3.411686283700421,
0.36133266872386266, -0.7170007299885852, 0.5642775811460432, 0.04062634488104666,
0.5883534304951312, -0.19378463966600484, -0.6808632203797045, -2.0487913910509166,
-1.269651511757323, -1.6676616130697874, 1.3477370717445727, 0.13004550318256214,
0.58672377844334, 0.17975406234962713, -0.954017150630434, -2.1923112768680086,
-3.1459802177175176, 0.6341446084736564, 0.4336210149459999, 0.9640157641060344,
0.7232340153655419, 0.7652189788657503, -0.25348754949474145, -0.6355372828104077,
0.9166391257652665, 1.0150354746954167, -0.03636938745402512, -1.3867307725107332,
0.009177761004041995, 0.7098247687168714, 0.8753799259555303, 2.781609206987242,
-4.790668721247168, -2.5600561197236735, 2.3624644203963183, -1.5202276920387954,
1.8800480133477875]
[0.37137797716929866, 0.3917050198900176, 0.6371054149282289, 0.18785380177250277,
0.18332950003034465, 0.6847850363794387, 0.2969024387117096, 0.93295518741951,
0.42624009112762795, 0.4712407835216363, 0.00421258881271792, 0.014577316268868148,
0.1022272401769808, 0.625314742166973, 0.6644972225566472, 0.952197558029905,
0.432469433749579, 0.7063900487542806, 0.3381591672204354, 0.08230252063771788,
0.41856772146635374, 0.70014733251935, 0.8042438267645861, 0.9519835922269637,
0.8321729347701794, 0.5666008868061243, 0.5593934784483766, 0.5096077677927674,
0.4739896657959744, 0.6792316814558591, 0.5756950786945507, 0.8571618256569391,
0.4500741556564184, 0.4718936931207409, 0.8370212020343207, 0.673954432662491,
0.5247092300807876, 0.5634896913101409, 0.8059993396870934, 0.6073820035137986,
0.2591504026605581, 0.3102438021651079, 0.5962786697143736, 0.038077828038767816,
0.46091324793682015, 0.8941768516628364, 0.23265244456724227, 0.44415604478975845,
0.6995045876204917, 0.9245640610740848, 0.6802222393567127, 0.6172274794341055,
0.37885878654119387, 0.4274922968618458, 0.6369976599384782, 0.3561154273961151,
0.7848712870144318, 0.008317567670499857, 0.7382429162067012, 0.7379377739122842,
0.30527743090962767, 0.0010117605232295916, 0.3301232707108888, 0.5887676868991895,
0.7823082293627679]
[0.8703662160764746, 0.2087073487580508, 0.08525450860253388, 0.12097736706123653,
0.988287496515166, 0.646012908978176, 0.12939983177629086, 0.6907652563921265,
0.9594778523568883, 0.607977745105632, 0.2361747631183514, 0.9626313498823442,
0.6986415151630683, 0.18204782687325302, 0.7662966706905571, 0.5041746404010078,
0.5740429881272324, 0.3663093272998529, 0.29487014604728395, 0.4173701127584078,
0.5220247427412105, 0.45945884910411, 0.8662435899399796, 0.07420553967563837,
0.19898931240818762, 0.9369861739113812, 0.6060826886552411, 0.6148164554518091,
0.6271104719550169, 0.2422300805929478, 0.39477133846445317, 0.21014841364230163,
0.15198323667114044, 0.9893637005718661, 0.742763279036556, 0.8768437292378292,

-0.0018971888793802832, 0.7045483057065085, 0.30799459862628303, 0.4979107243633002,
0.6752514254032553, 0.031191951519900635, 0.36984886707779824, 0.552406518030342,
0.8704937418424574, 0.5127236057987327, 0.3184045753275917, 0.6037607350313465,
0.5836119252877, 0.2923300552975058, 0.5488599607165945, 0.27574392324731667,
0.009265892531286127, 0.3097274940829199, 0.0866267186389896, 0.49171593498720617,
0.5011489035231449, 0.8702606613230105, 0.7514736633994326, 0.7506611373093242,
0.9884719474584387, 0.2624603328541488, 0.37280542120397514, 0.23007802455541793,
0.10098028129303795]
[0.5152291321320495, 0.3088852064575548, -1.0408222621446264, 0.5923545563001724,
-0.497916520764644, -1.3292442446321564, -0.38940304975336953, 0.0617972941915782,
0.7317304217503617, 1.767397767111775, 1.5497411979226876, -0.16987587353402783,
0.49099364409433066, -0.4322852058210594, 0.9710416117688655, 0.008799244158186337,
0.21135937026857277, -0.32673625077941887, 0.5704852803923524, -3.3618415187964272,
2.2012730578308863, 0.5833085879494853, 0.19036090347044426, 0.8870010231418268,
0.23859133549179212, -0.6478098801818069, -3.396130894690826, -3.727639925488942,
1.0504870946336893, -1.6137776361303213, -1.8944225636218646, 0.6401390663808352,
0.76148735124507, -0.013588910853154727, -1.3813701986394826, -0.6956925648559807,
-0.6194691280515559, -0.3312964873037728, -2.1289212858451365, 0.1462988262088517,
0.8329743924973544, 0.14127151996547688, 1.483321065239265, 1.9321598176402845,
-2.205500728483044, -0.901925068874769, -3.328638843041564, 0.7934570478484626,
0.6647668867194073, -0.1573853291163699, -0.37954220087074575, 0.3993483104431894,
3.4723755257228, 0.2940511073379373, 2.394069208827199, -1.177277784991391,
0.28611141230202086, 0.010650930741779339, 1.2090559055786783, -1.3653234731493253,
-1.230843021184773, 2.4804185953157787, 1.5464563888199452, 0.6716574318260297,
0.18926666663718644]
[0.8743122139751826, 0.6595244625101839, 0.6132028988830276, 0.8671896772853335,
0.3974119229118305, 0.4333541200516399, 0.24422032392976106, 0.8302020690541426,
0.8773606966711958, 0.9105550378936241, 0.6029342777852488, 0.1196125747989402,
0.08375268475147626, 0.8054172689635886, 0.8855885737991821, 0.5322688840613624,
0.9207674262789277, 0.9306772511869903, 0.7581362180032973, 0.3828329069620283,
0.4685944796753375, 0.3576130403244491, 0.3961637531319844, 0.47131536536840735,
0.017110372858409884, 0.13111336191381706, 0.17893270926512003, 0.5790220329247604,
0.883031747287136, 0.7163794759977172, 0.14947723402133076, 0.457683933115711,
0.6273041974919829, 0.1371295052813361, 0.08661168164753642, 0.6206957658958372,
0.2467538778008707, 0.6488891047350949, 0.17149610860123343, 0.8559051638324977,
0.3097403204237128, 0.4283580289836204, 0.551168191940024, 0.888935867520526,
0.9283958418975179, 0.8499886372586369, 0.6844866826310778, 0.06918778780927826,
0.1867894417016298, 0.5344132254506492, 0.9840725099655749, 0.7284017050340851,
0.20277306903950948, 0.36257016668228514, 0.962259811150385, 0.5077591803193302,
0.8703182217128731, 0.8579955309916518, 0.7795882696643724, 0.6301239473836622,
0.675679678710292, 0.350541139690598 2, 0.12110899159958789, 0.9486012890355182,
0.04370164734811261]
[0.2708849600348776, 0.6138359826290882, 0.9547329279308218, 0.22936239754845913,
0.283817572050275, 0.8595986460937219, 0.32532071333286583, 0.40295530292646475,
0.3597401810619114, 0.046702202754720354, 0.9322409319590653, 0.7237183450646477,
0.04450835573724023, 0.11165140971060931, 0.13567905384271373, 0.368886750237332,
0.8903220241258988, 0.14069115305310517, 0.24090729322882934, 0.35274257437121737,
0.5546500251115813, 0.91324859363526, 0.5397106527332359, 0.9035544832302415,
0.5419270306025946, 0.4408850052732792, 0.8992222873774863, 0.620425307727546,

0.5146000900508976, 0.5250085269171987, 0.35560745563106577, 0.4331013545726463,
0.07415858862491798, 0.20830689037316164, 0.772721361984584, 0.16494344229140717,
0.2456124190813763, 0.17172120982220468, 0.3634693624350392, 0.04928816693536242,
0.36032943489000346, 0.609691152223068, 0.6793601703139913, 0.8853460398798737,
0.12643336643614203, 0.6551380162047136, 0.1971437855089785, 0.3424250942841378,
0.5751293502107571, 0.8376475559122094, 0.6657781505650282, 1.0004896219187365,
0.05479430509041061, 0.33213279714741323, 0.4802508953875757, 0.03626212397693761,
0.052372596064729215, 0.3667663705363275, 0.5489706350392322, 0.15461662425804418,
0.10347378420184823, 0.3386955718601809, 0.7414902093816013, 0.5673177486396033,
1.0327518083796243]
[0.6051042103599796, 0.8903679301313383, 0.5994254028080178, 0.4874848952888491,
0.4106306781740554, 0.0899607889410023, 0.07343362811354133, 0.6584068808391955,
0.8591728160005081, 0.01922167210527499, 0.2033684428699752, 0.3276534462935838,
0.29423247055896473, 0.8313461541248505, 0.2535564839160543, 0.3062135499129053,
0.4875841469761729, 0.9524241916259274, 0.3082593665568244, 0.6239774399625698,
0.005979593397217951, 0.4158640334280777, 0.927004705865046, 0.2174049040549511 6,
0.356461612642898, 0.6516481020836608, 0.5699685648423918, 0.5766334594359022,
0.5842405736179631, 0.6651963724886618, 0.323282995560413, 0.35171617638022656,
0.3970061865215033, 0.5209574199105682, 0.5664288981140989, 0.8615918732791409,
0.3772288998528479, 0.4531166999763146, 0.8342007642414707, 0.9710749365228007,
0.24289556620832364, 0.7304537811422287, 0.23776259302358108, 0.7191094170372824,
0.01718083679576374, 0.5133787032289137, 0.5723974448441035, 0.6995913857762528,
0.917030561867377, 0.7960909179196093, 0.5663352040752481, 0.49150725822741,
0.00016839539299566856, 0.5414824729647841, 0.5569765332134684, 0.7419873421135873,
0.16539805079536674, 0.5886941224208168, 0.07777593505784595, 0.7369894738624951,
0.8137384423068532, 0.41097214347015565, 0.679464981217095, 0.6617751071582145,
0.2944730764158645]
[0.0882451194663475, 0.7577290983095516, 0.35166270835314517, 0.15651896139550822,
0.44520664847728736, 0.8405771633495708, 0.9553022930141775, 0.5673387299269459,
0.969856387437914, 0.17034859083313159, 0.4846197927839601, 0.0038214548414251107,
0.22894826956386996, 0.8785714603692094, 0.05945655779594677, 0.65443020435811,
0.5095412821201154, 0.9865332854729467, 0.9975762490500928, 0.12857585203000121,
0.2516671707361473, 0.9903295521904787, 0.3299450803743938, 0.180478601198241,
0.9117727483923297, 0.6182623199478403, 0.3190933477973179, 0.5614038168691724,
0.42356887324862685, 0.46023288818947633, 0.552218377322867 3, 0.1697781124705542,
0.615603495443488, 0.9553379486402953, 0.5934654504101309, 0.7824809201876627,
0.27790844807477605, 0.16115943640943056, 0.009626109099253431, 0.9813188786839407,
0.11906217355609527, 0.3800183634179082, 0.6494535734427263, 0.721363617844313,
0.6137050325459869, 0.4481868581023314, 0.8161135553659422, 0.4423597854656651,
0.8353037104488943, 0.05318903225755933, 0.7131481961304227, 0.08767575610956789,
0.38272651704938615, 0.4404167149334847, 0.1752284540633, 0.4489301758866368,
0.8528943925616802, 0.03640939821649437, 0.18686303250121908, 0.9712017968793633,
0.446555343061778, 0.37995922305808416, 0.9034816203701046, 0.7758511905434942,
0.16975158822451575]
[0.5978860651946428, 0.18059312444618644, 0.7673744999120309, 0.5333675688527089,
0.7606280952765176, 0.033812343228271695, 0.9246595599894639, 0.22981106076880609,
0.8497813814803977, 0.44292696440565693, 0.87682362183143, 0.07474852976468377,
0.022428129849525805, 0.4471730037317438, 0.929952601964014, 0.465313928807347,
0.5074718452694191, 0.1631832933182325, 0.518769239451339, 0.39533717317504563,

0.8704901751242972, 0.7318812190911528, 0.47735146595785377, 0.1489549004994799,
0.14595486484430253, 0.9660873074970202, 0.5726955520765928, 0.183668042715845,
0.7835293494932792, 0.20093311006797962, 0.4538501756837777, 0.8771760922354259,
0.10340270920963646, 0.0991046999035933, 0.031824404189932376, 0.12573846272481426,
0.34228629298257107, 0.30154563247291394, 0.2776888086352384, 0.014162759389941493,
0.48711800364210034, 0.4453659952300972, 0.7421585948081556, 0.29748144848563585,
0.5502849400827814, 0.2855324783691055, 0.7519355130337078, 0.1741455651843582,
0.48935061082033504, 0.44561799103995653, 0.45852660002753787, 0.523752856051954,
0.5046900161978041, 0.2950316609891612, 0.8268527549442907, 0.9514355990878389,
0.5590075398596664, 0.635466967112807, 0.7166902135885996, 0.29602843193048367,
0.561087291486032, 0.45114488932065144, 0.4891005303217828, 0.3940871702790435,
0.5009943978401208]
[0.2181347546434632, 0.24107322320278524, 0.18710396616633596, 0.5847296906203769,
0.23606567823828983, 0.7686885801370611, 0.9037302523051499, 0.7597268973405501,
0.3282812567036614, 0.9421217827245794, 0.3309445004100781, 0.35160299695805664,
0.5903487431185653, 0.6568160469267428, 0.40841710667669, 0.7866659157149083,
0.8535253852090725, 0.28759564055807824, 0.21185170886472204, 0.3905739719056478,
0.7013187607959391, 0.6705735910590146, 0.17565573623481912, 0.3886977929720531,
0.9018589702872081, 0.9592469253796004, 0.5951321602352414, 0.7693514951135306,
0.8354242523656232, 0.22054226032838206, 0.06563085391223203, 0.611358555557219,
0.38469160499991883, 0.7105989697571662, 0.2917707795367483, 0.4294820641539743,
0.8028268857249181, 0.08799787189905245, 0.40689253917917106, 0.15324150108844592,
0.5336140453133892, 0.7327700031128304, 0.9872560542404482, 0.7536586061037402,
0.14000547730865123, 0.43014651411578075, 0.5416383314728672, 0.6377611379649377,
0.7008425110796995, 0.9729947230899415, 0.9380258422070348, 0.20306610162613753,
0.15012348931222874, 0.9651342456254431, 0.16195281872154751, 0.9681884523626187,
0.11983713836985044, 0.5849882914901702, 0.11776013042070002, 0.12328802339344816,
0.324773699704277, 0.7931296876900423, 0.7040141169022811, 0.3176671040885297,
0.12742554056973932]
[-0.7840811954737894, -0.998896971181475, -0.9436096583451007, -0.6759610105182148,
-0.6937145262139799, -0.2494269023705794, 15.168915624502857, -0.5718347372949172,
-0.6031875479195752, -10.751696623058825, -0.8087914436956232, -0.9990707030457416,
-0.28429526980340514, -0.8100101689615614, -1.1140182906226934, -0.7434339654874633,
-0.6489549616417505]
[0.5991774234337601, 0.4415678244222347, -0.06415233805385086, 0.6385070977274406,
-0.26854523001680036, 0.41471287214888897, -12.941713518244708, 0.4981780871674075,
-0.07831895869244797, -3.101764301773774, 0.5103276624155086, 0.41955255028107347,
-0.17295473997318378, 0.0912700665208932, -0.20954478484797834, 0.4557644355082651,
0.7204877096849306]
[-0.4270231255124818, -0.27703242259216215, -0.2644736841193044, -0.7975151033527685,
-0.5535967369927376, -0.5770584979410087, 1.0741673782866954, -0.535598568385989,
-0.5755480717483898, 11.691724350990555, -0.70136212924404, -0.8186227287115576,
-0.6759632390773694, -0.8886327791635494, -0.2164531709690365, -0.44065393547206294,
-0.311521915746274]

Weights for nH=24:

[0.3836577771074978, 0.7675857449307816, 1.1547454108107225, 0.8345052779560511,
1.8763250022354108, 1.0565117825915908, 1.7674005091386755, 0.8663825107121359,

0.3784075202852446, 2.332977852450054, 3.144564940183717, 0.6957682715548222,
1.5568309225558694, -0.2416268078337071, 0.8999789956806344, 0.11950013245412117,
0.9725295748130851, -0.13898399528532937, 0.14290370625182758, -2.2570246232339883,
-0.841892000743852, -2.520258459351645, 0.2787514003311193, 0.6784256856743994,
0.1274307077182104, -1.2016984691342296, -2.1750964898399023, -2.9385370762994825,
0.2258775115918024, -0.9251065714698647, 0.3444847870601477, 0.764399484996985,
0.6020305336769644, -0.051896429534927895, -0.2246921695772775, -0.6247155414905035,
-0.8472061750264815, 0.3035622429388266, 0.8008854289363118, 0.20391049290268382,
0.6825436066958162, 0.24831204714143948, -1.263229539776653, -0.41534656931174024,
-1.7037284607816328, 0.09283965763529835, 0.12472630481102645, 0.9915905354150086,
0.7515738595609315, 0.7785044287508275, 0.31072949291111995, 0.4964255919848802,
0.5421636261779751, 0.5363484942067084, 1.565289253901932, -0.3840405703507762,
0.5568147221005418, 0.575995028368318, 1.2242197444476968, 0.9184407788966558,
-1.3326618530118541, -1.0670806953285321, 0.7798062372131553, -1.7576341426771689,
0.5950781100028198]
[0.44375914838371544, 0.5621870535099014, 0.45113140946763985, 0.6754371896604162,
0.9339892017692355, 0.03136499399994107, 0.9300198715465997, 0.9900439110941297,
0.8629802809584769, 0.46377348606011565, 0.718770974802869, 0.45379448801841543,
0.2881333837290751, 0.09839797988585516, 0.19832431733686048, 0.47723671448199523,
0.2821679234341109, 0.21612796742371296, 0.6251912534688628, 0.2092283295595123,
0.6987480559416667, 0.013795802627922384, 0.2233008724691076, 0.4244167395834072,
0.21419724576886756, 0.8257034249438167, 0.4519402078939862, -5.97879132844377e-05,
0.10524893892875965, 0.633048500052324, 0.309475272989495, 0.29939698890856026,
0.2107572214950899, -0.006180119720811172, 0.725049896792721, 0.2670582177220405,
0.15973781975067694, 0.07452058977665037, 0.5085612908880587, 0.7387715085013629,
0.9470770371178014, 0.932654518054309, -0.0034317275406712876, 0.44373520417518086,
0.393760590710861, 0.5191958207009338, 0.42951421504395554, 0.9275701317594791,
0.5939310987747468, 0.43617200024810115, 0.9693210534042462, 0.5003754392054535,
-0.06839564977122371, 0.5747759916002072, 0.9570514523493433, 0.6524936254705392,
0.1997319284610123, 0.8383084041934025, 0.09376927352340715, 0.859486137855361,
0.06814072713972107, 0.7453652892403273, 0.16146789129833672, 0.14491965028951403,
0.1808614366461831]
[0.5039746864288395, 0.6924252055428969, 0.5159491468451664, 0.07280882637422224,
0.9448477027873883, 0.8358681591522475, 0.3662196920662446, 0.03764342166345169,
0.4093524085026319, 0.9643754018680875, 0.9581241969993041, 0.17044823283407,
0.198156917201321, 0.9764915141446152, 0.3387289307690506, 0.6960429198942161,
0.928945850412855, 0.8265255761007889, 0.8049261336881535, 0.608834716663445,
0.8091590840209893, 0.23506592938811557, 0.9041055281602194, 0.21304586739531683,
0.12326385803069928, 0.7691166680556614, 0.9880577388694605, 0.754922116470197,
0.62447328049646, 0.08782241041257777, 0.5281305973076073, 0.17853290927498255,
0.5407512864503733, 0.40957621172469616, 0.1389657391673204, 0.5448808197242175,
0.792439117016555, 0.7134135557875058, 0.1389553361167844, 0.9363363881963134,
0.9319802205701049, 0.35869864924438605, 0.9794878256095965, 0.19456681282387872,
0.40001492447808196, 0.8389451371990584, 0.08481868371881442, 0.3943272558292381,
0.009808907546960066, 0.211230302062695, 0.3599731280759008, 0.8377004276475765,
0.6270743746133287, 0.09718958622095404, 0.7989442614481795, 0.7589895357875815,
0.3066933281734099, 0.17331014937799064, 0.738513994792246, 0.9391135908160918,
0.21989436646158034, 0.33374814065694575, 0.33687305108031207, 0.9670328210765338,
0.051824700217807045]

[0.018591990375907308, 0.18210641342224426, 0.47063948299026354, 0.9574798210539127,
0.01548852738405556, 0.7904632913165603, 0.5291800705417149, 0.8902185430491149,
0.9638991807834547, 0.7205587319763567, 0.25706229920538104, 0.8969943211413951,
-0.011612344722979333, 0.9745168022091428, 0.4129247629117473 5, 0.2699714810282322,
0.883010960008947, 0.4999881476438834, 0.37440383765297774, 0.28282116460227896,
0.38720201049613745, 0.23652344956600685, 0.5179503709263276, 0.43597552602754464,
0.2639824003755633, -0.0016592632708420274, 0.1352862092444568, 0.7597459584448704,
0.6808471868232344, 0.9148488200991499, 0.14932354278981824, 0.9265068693291914,
0.887264377504174, 0.34960329398072015, 0.9537447424431914, 0.9689261891155618,
0.8165724400894099, 0.5522805855840153, 0.8716943182412116, 0.1543447151192433,
0.1327485174016182, 0.1388622813465114, 0.6913106571021654, 0.9465929738689082,
0.16729520611807366, 0.12743848005071032, 0.8867283686823907, 0.8781681218480367,
0.12719888072692775, 0.03365878639948213, 0.2789225152252875, 0.2651477087925574,
-0.01671550873122 7045, 0.13058611161615716, 0.8326662308019569, 0.06334759474446816,
0.45619285808559806, 0.7165708263097762, 0.7224322566385134, 0.3037257182893237,
0.11230088228580261, 0.40370553216950306, 0.3566290962115341, 0.44685166306051516,
0.27046041538440396]
[0.47518827904492533, 0.9740927361424487, -0.02813082034906954, 0.24535048908809412,
0.8770477680054758, 0.4816855687218141, 0.7793281760089956, 0.07915157522179472,
0.9899011959374497, -0.002255563929128436, 0.751898528144858, 0.26522398740647424,
0.7466099366024461, 0.8052567010338912, 0.7339713185953968, 0.9548137365910017,
0.22248881007395682, 0.8491153085708629, 0.6054891309498338, 0.2189100608477923,
0.27310091266761405, 0.09885364762971385, 0.9025925985814625, 0.3303781449061197,
0.3469152755003422, 0.1277231358640496, 0.8084583441140384, 0.5873676255791803,
0.24268803909438647, 0.8053078783698764, 0.5362336860592328, 0.7926748303781457,
0.7058547517867373, 0.43253844058543073, 0.3636532889855885, 0.9286121144478262,
0.2162652457608609, 0.09610176363311249, 0.8572112070973062, 0.6874170517494126,
0.9257782808935802, 0.6106484261223324, 1.0020503227403537, 0.4681086942040154,
0.10180465392111518, 0.6615600179535774, 0.2493368846419001, 0.3920520681100138,
0.36836941973547743, 0.32478678181489457, 0.502468660641366, 0.6229518386902003,
0.09996725381659473, 0.8902302234584818, 0.3474207458752519, 0.7378776906465024,
0.3465878270066046, 0.11932556955001741, 0.2930028967090507, 0.23979391634015318,
0.3311259840479569, 0.6771491134027328, 0.6775889371735632, 0.19273505106355132,
0.7306682058342358]
[0.6160356703867356, 0.16592283373091402, 0.6512792009223479, -0.05037946305801945,
0.48962463339008633, 0.23812803925904416, 0.592775396105868 4, 0.9377205012716218,
0.7063327777407302, 0.572716688085623, 0.6415028305443791, 0.1285183979541698,
0.32837335239023696, 0.02885328032056147, 0.4931681288484954, 0.9681888845552649,
0.5030499494951645, 0.7494975295061924, 0.8823474900483301, 0.36741387903648043,
-0.025879767087163805, 0.971504258364 1862, 0.5083210001939256, 0.4279878846316286,
0.06265714459654126, 0.7010713420932733, 0.7691043160040755, 0.3274821979656021 3,
0.03475702847982466, 0.6699867312491462, 0.18943235985254595, 0.34707599310296733,
0.2388291486318681, 0.5220356256460689, 0.6076518035426746, 0.39912619321439774,
0.016531053158526245, 0.21580268346685824, 0.2195647897026469 8, 0.5104276875607231,
0.9636856247243017, 0.6556432383922174, 0.8600162879152652, -0.018404016618568 75,
-0.13177415154276792, 0.5081483587620494, 0.22920192870145117, 0.33257859190740824,
0.7309770743384236, 0.0967120326533444, 0.9145046741221466, 0.3076567641471311,
0.4516851780064096, 0.053748131406081624, 0.10237367384041546, 0.6711671443204745,
0.03782205083236945, 0.6369146780095002, 0.38410550473477334, 0.19101607422201416,

0.8143914782528254, 0.8721411990738521, 0.3533993797614572, 0.38478165351193006, 0.15526069943783863]
[0.11990226496042389, 0.2956073577862081, 1.0976492635729296, 0.5806769734327799, 1.7790713262035605, 1.5876311864270773, 2.898939049307342, 0.24562846473779398, 0.9134564404999961, 0.15415032029920916, 0.6019120755657026, -0.3572570016426147, 0.31683428660095536, 0.9941775100212611, -0.3389649961058487, 0.5582934362675425, 0.7138825245904499, 0.5247212380209626, -0.3288049452274941, 0.001264590824295805, -2.8805355316305503, -1.4651424052509732, 0.9898925027376513, 0.6218775171739609, 0.20484899521075361, -0.3820285707878942, 1.4767471861178565, 5.243640813339126, -0.016851372986349273, 0.03181449772793143, 3.5913430517741185, 0.8437858808935765, 0.7279649932908665, 0.6082833394821643, 0.029178256087129012, -0.15064045391792044, -0.47887289527762067, -0.3677636320166196, 6.651701064505279, 0.352050496299232, 0.10944123263057037, 0.7252772864513215, -2.0436854627460477, -0.5470444647080918, 1.5257006419640198, 1.4973795645710761, 1.8151286306803036, 0.9406605938733351, 0.8323281345961557, -1.0436107462947457, 0.9232849542177359, -1.8721932319570056, -1.2553802190151315, -0.16095200100509377, -0.8449382539078845, 2.390936670820957, 0.1597511324655545, 0.5926946273768381, -0.46630190628996004, 2.9268459653321823, -0.578882439997766, -1.0070655431127369, -1.250641955745261, -0.7743632846935979, -6.529145579310595]
[0.7073156047700468, 0.433713199656922, 0.6569892238121472, 0.5711380389939203, 0.4297575106093364, 0.35225403530314403, 0.21242484677696352, 0.9913324733842115, 0.2033171107511037, 0.4615781474567164, 0.8819958471396178, 0.8116629598184762, 0.9844368238645894, 0.6218688733913367, 0.28400514867479953, 0.8680388960528205, 0.06582450949525565, 0.942245572465879, 0.4383980233470926, 0.6846080915123773, 0.9032762019517323, 0.5216045937660495, 0.8205007515000963, 0.8386585356809791, 0.5090375386492554, 0.9110369982020692, 0.23979372776368432, 0.15243268229418022, 0.707447166475226, 0.8239393655778929, 0.41169257401364384, 0.03021172191114685, 0.429294171915124, 0.2256746249786589, 0.0689878249670952, 0.16568656201065754, 0.6015707936672992, 0.3726387598180206, 0.7962177596290194, 0.21989064136249092, 0.603815513784034, 0.26833971792725836, 0.2864261697607066, 0.15472562430802725, 0.7443409709148675, 0.2734636897899255, 0.6696982081155437, 0.6884970174632501, 0.5541652818112324, 0.6620821707882063, 0.6786334226262198, 0.09770767174201125, 0.7720476184920281, 0.5139253078064708, 0.8149999376403538, 0.11095274180912923, 0.05751725553483478, 0.646545379711295, 0.29594406395996253, 0.9772437280093856, 0.30271628590757044, 0.4530427914312051, 0.3457931369243928, 0.558894347562018, 0.20816256956512116]
[0.9603611137661437, 0.27377361386101556, 0.9040566806100147, 0.9948457785760447, 0.3860712184073642, 0.8406325560696359, 0.8945088202201981, 0.035625533228329065, 0.9708066741494044, 0.381268588305668, 0.3264514426847061, 0.8253009144942831, 0.21878475937115313, 0.06075758358136188, 0.9685079744295018, 0.017633047278253766, 0.63787078568881, 0.5666906703397768, 0.43079810734856855, 0.873446426634714, 0.8840830620480483, 0.059497211489674226, 0.6183389200905572, 0.8585149932024465, 0.3891088448927317, 0.13773109039959897, 0.6095404403332273, 0.0604900508117435, 0.4437702551803999, 0.5308325750688767, 0.2146159217688292, 0.5854505973495308, 0.8276937410961236, 0.4771654811483912 5, 0.5584895690095472, 0.22312146156661417, 0.06209761880041368, 0.4944371034652139, 0.24955550260411555, 0.6312961542237212, 0.10059031409163499, 0.45252555918883225, 0.5238030578427235, 0.8112361161016415, 0.10350146291783162, 0.911611413964952, 0.15285194175049974, 0.9182638030568925, 0.811789578021936, 0.8901932973796601, 0.923576997814774, 0.09807862826467416,

0.4114640050420038, 0.6281235522893276, 0.3351422240832726, 0.6604594808461646,
0.8703075121502039, 0.203423844056661, 0.3017876889885184, 0.8801489505899239,
0.37617927547476077, 0.1282704751730488, 0.751635630033418, 0.2938605916563677,
0.6327126304963093]
[0.06581682332370054, 0.5649828703668123, 0.7907109253417547, 0.45254844542091255,
0.3741410306315324, 0.7492844211813067, 0.5192503870611249, 0.47496399141014256,
0.17434204993371683, 0.6782692801689052, 0.8080690029235967, 0.00969928001295487,
0.32621730137658267, 0.6451093330514718, 0.14910464433192597, 0.858452452532792,
0.13437977204881957, 0.24269954782434697, 0.44473828422367, 0.15380147498398986,
0.9417798898625132, 0.5102283143289287, 0.3896055877396292, 0.610671629302935,
0.7473963754184244, 0.5466537046249978, 0.6379765652763101, 0.6908527974603293,
0.45587426886160726, 0.2836360206936756, 0.12497265435986525, 0.6627147212460307,
0.3780189829703957, 0.9535635447015177, 0.2759153707596524, 0.8029605039059838,
0.45979634603857733, 0.1441510531287304, 0.3881538473772745, 0.6276685744459687,
0.6733619575337484, 0.34750165675453726, 0.23064413981453286, 0.18366931328646727,
0.614256248377419, 0.6676707966904268, 0.8995059775719069, 0.4897254402661393,
0.33079069221825097, 0.7979832462914206, 0.6295459061748719, 0.7418678370638703,
0.07795179153529197, 0.8180573859027639, 0.3949463703507852, 0.6708145903046854,
0.07312759067245822, 0.00015819259282025855, 0.007237560181931602, 0.08182633326288771,
0.10033621608927938, 0.8689773467416126, 0.5519781939753882, 0.061973324232031016,
1.0095325224090623]
[0.4307383100518397, 0.6817149863114615, 0.6505237646868386, 0.3500521798082789,
0.8294071232313699, 0.969058179960212, 0.4092744154493355, 0.3543324926143999,
0.42274460532311464, 0.8288499838068489, 0.48976238726579985, 0.09807071466760509,
0.009898963549918363, 0.6628831333717876, 0.14818712923484134, 0.7667462967817731,
0.5997234487012529, 0.20845035298677694, 0.06417290349465986, 0.6784828895438573,
0.16137950410278895, 0.9819986718448559, 0.5138667700817918, 0.32705805227063045,
0.8041560202533509, 0.619127644625949, 0.6665414518740779, 0.7980822587707889,
0.4520528193763283, 0.045039597767936326, 0.15938577277712007, 0.9912684785157757,
0.6239589550774126, 0.02477687760362162, 0.15155081341794982, 0.10644300422280675,
0.25756285021653363, 0.6553707293095955, 0.927324999239895, 0.5717505104440603,
0.4990657954800918, 0.9221069765516797, 0.12838874714835888, 0.8934981323538387,
0.646783597374627, 0.23908279044895192, 0.7676149864378774, 0.1749186682964744,
0.5919439615080287, 0.45926990055571937, 0.09708877507049114, 0.9305349538462756,
0.19389224283970966, 0.3601748805035936, 0.6321984275848661, 0.27022576064148246,
0.09473762721303791, 0.3003014819442557, 0.36497924756034095, 0.8030710271798208,
0.5450796774081464, 0.9159250805204577, 0.13182822554167997, 0.8270663912823246,
0.9841246273877206]
[0.9831712701954762, 0.2993662710645667, 0.10242498646994157, 0.7658428555942667,
0.4416264010642891, 0.6422449226148086, 0.09882691499208267, 0.9640735001267688,
0.39535671827759866, 0.9655609873295371, 0.009803876739672434, 0.29552377869143615,
0.46978102208479683, 0.369075271579829, 0.2093556047765857, 0.9802897264899629,
0.5481293563488099, 0.9369638247304216, 0.5208845214515937, 0.7872792623840126,
0.6410299827425456, 0.13723019396950087, 0.6982659064188931, 0.7476240485798494,
0.7168500376095548, 0.4914386568106796, 0.05092272003486768, 0.8321986423351065,
0.9691965082795815, 0.06743285930308128, 0.042728133454625665, 0.4319370711674815,
0.73696391587715, 0.4494154488178673, 0.8190428761327088, 0.8804827353016964,
0.5953431100483839, 0.6177663160562654, 0.6444204597055423, 0.3711012679280552,
0.20358353097512283, 0.9103344600315492, 0.25915883738269624, 0.8258492393003324,

0.5176742506295221, 0.7411911106611211, 0.7879255270706103, 0.883120435588625,
0.8383393526076189, 0.3278093298914289, 0.834776263029952, 0.8694586680546529,
0.642682408404741, 0.19967663882393363, 0.9849080237797264, 0.7989000437812981,
0.3842019974808617, 0.303014568647466, 0.816457380871733, 0.6616969656066002,
0.37582162206899483, 0.9064863981407938, 0.9896028546596563, 0.6785222757721574,
0.8330787011431708]
[0.5966131111788976, 0.3049967378166899, 0.7974259360153219, 0.7003668799294297,
0.8795713872075162, 0.3448544129176718, 0.9380848402527194, 0.6948342737988397,
0.750465686542124, 0.8434965122161903, 0.4545332761878652, 0.8398968390738953,
0.8921323527997458, 0.9703689839730041, 0.5993948996082383, 0.0617264515319107,
0.2414042813460131, 0.1981291128916732, 0.1400491663935119, 0.3789157952149872,
0.5914194476026372, 0.9898530213851985, 0.08228853831071936, 0.9572822068102208,
0.5469837669686077, 0.8011833952794889, 0.8254269481059723, 0.35279602300514085,
0.7039828905356902, 0.743484767992646, 0.32900800946849595, 0.9246379462937042,
0.8819446558850201, 0.01097637636054092, 0.854361839553176, 0.38904128024177426,
0.9581234058430405, 0.056478520811609846, 0.84258168702453, 0.5116020029766039,
0.2764570791439733, 0.49165367106709096, 0.8362323999467174, 0.4317702195951509,
0.4562158358427786, 0.8818570287624374, 0.8402422121816454, 0.22018046059119278,
0.5114962908717022, 0.40117052196938174, 0.19318995449835327, 0.20337168962945262,
0.11318122458482147, 0.16866257373228946, 0.4226648772904436, 0.2529169109267646,
0.7252084398950263, 0.09280201178599828, 0.23870992699551366, 0.9614549809935492,
0.5710881892133934, 0.18698494246674566, 0.8231395071806239, 0.01690174696595363,
0.3059949214228783]
[0.1385372664186072, 0.4113358314209933, 0.4592968182100176, 0.9074503741157838,
0.4930486252430719, 0.9020217487041828, 0.9584712099014515, 0.04469719072375933,
0.106004943618907, 0.8083342916490666, 0.28472477948428837, 0.4548844603406412,
0.646743146865862, 0.4264775597986966, 0.8667481669919179, 0.02665708109987064,
0.2624947622476944, 0.05889834335069123, 0.2545016545885251, 0.9875317844856674,
0.019903313531595697, 0.5418938310864538, 0.2183048553921046, 0.7412472523537161,
0.16390388683868828, 0.7433957313305484, 0.5227353807986561, 0.9657622087863239,
0.07211475782813186, 0.5229790379461665, 0.9104178988672285, 0.5209698234723555,
0.4257846427376334, 0.9236466521688498, 0.8298255924853756, 0.7759718297667803,
0.39971537811728686, 0.3292444395748965, 0.2825983449083065, 0.9737163748942792,
0.07766092218340481, 0.1351112165565548, 0.6293537002637115, 0.8734242567993028,
0.9741824597831429, 0.6113183247682672, 0.860769150675679, 0.36374808460425456,
0.5153987567749581, 0.7685020223615333, 0.16601397498439746, 0.6915829894602101,
0.4165690382019118, 0.038489612895810016, 0.3500193463325097, 0.7408867962606595,
0.7995988004803323, 0.684660015773975, 0.7154723037427091, 0.36218356094128085,
0.3032277357570851, 0.3890506440388705, 0.4302185735599205, 0.32574322352595203,
0.8778002088542886]
[0.7005537108531836, 0.15161468367573994, 0.14001817141911463, 0.4791796721563159,
0.6443934450434046, 0.714019948742994, 0.43312623130411315, 0.5776137297715126,
0.6052434224526434, 0.2910582419720195, 0.8113173020605132, 0.34467552253972583,
0.12801114926196325, 0.1515796845064570 4, 0.648516987351674 5, 0.4743059199649750 6,
0.8449691545306829, 0.8767985593366372, 0.24468755544818402, 0.7813223002338726,
0.6625301711373012, 0.5548199021408797, 0.9955699557629815, 0.16996090704851585,
0.4199953945396890 4, 0.4122140208415127 6, 0.289124367993537 7, 0.627579735862527 8,
0.8448083498435905, 0.4773088627035401, 0.08132232051437303, 0.26270088170411154,
0.7194488077825302, 0.45956213426707904, 0.05567641073186017, 0.2505229448394201,

0.5259322402675939, 0.32803818978929533, 0.20786984043553025, 0.3812691727408839,
0.11086220878888664, 0.8868628971807472, 0.31187638125793155, 0.45402237476062074,
0.5214253532774119, 0.4596488635988293, 0.9232163098440521, 0.15440946773073505,
0.4939881104384314, 0.15539956698806703, 0.058120683182181325, 0.6971313421074695,
0.331379693454304, 0.7360523461117532, 0.4811216092485517, 0.03681847499892087,
0.08047086160323824, 0.027082590773073484, 0.5828200665112123, 0.4076615695577438,
0.2881511365640951, 0.2833707945217809, 0.41993324593205056, 0.8646364636649653,
0.8913446214565767]
[0.8239319077226289, 0.7461949459670046, 0.07094059985120445, 0.3441350812081854,
0.7903187637942471, 0.16121685996235144, 0.7173564507331901, 0.1087315798004359,
0.8148991967211918, 0.2533140835396972, 0.20760989358950974, 0.9267512582211263,
0.08522547768447152, 0.8875417936814649, 0.2944388222395565, 0.2796425698156626,
0.14443191507601283, 0.0361614142116071, 0.3298838995622446, 0.41138019017090155,
0.32408062656751296, -0.0006151833808153263, 0.8911724456851406, 0.8424828310814182,
0.6291674542068829, 0.12985012676492383, 0.09953542669342628, 0.43847612912759676,
0.1685575530711914, 0.27150527768007987, 0.2177386323942343, 0.7694648481848224,
0.6682810104457573, 0.20763869257276918, 0.9389136465742242, 0.5855688477583934,
0.7935956237338114, 0.648332548798604, 0.45609457488602195, 0.46972691539810885,
0.42720961885843156, 0.06823443941425414, 0.947751044494842, 0.3529501353893378,
0.040701358316678427, 0.07273990795675904, 0.37838798365815723, 0.7148504606904454,
0.49987528470051834, 0.3299281137154002, 0.9041356801732005, 0.04299640334222073,
0.2380711825293274, 0.8723381096809981, 0.4766463563380743, 0.4655725373601182,
0.08010239520994877, 0.5970565672065807, 0.6886546591479349, 0.7951476985878768,
0.07844828681539005, 0.5813901075123412, 0.8285813046287511, 0.8923982799705724,
0.056877034002965454]
[0.14453863288145163, 0.9052843328544674, 0.36735119704073105, 0.38866912177458585,
0.7274151485744021, 0.2526813712308081, 0.15897786176555853, 0.5598008087193751,
0.6174486138856887, 0.9224774899436455, 0.0782491761791217, 0.6064951975863356,
0.07865224069679715, 0.5826212816088652, 0.6435253894912647, 0.4167570746176753,
0.3945344915585426, 0.4066965555462304, 0.8250535063651909, 0.8427551053978031,
0.37915797395668366, 0.7267063100075407, 0.43552889340292006, 0.4406228840194665,
0.6225591491508786, 0.6073526443097554, 0.4831977140631813, 0.48628591149657985,
0.8125979097453098, 0.3234895091981687, 0.6277414140228614, 0.1411202793176629,
0.9777048651789804, 0.0430965666842506, 0.7872711418047009, 0.1285622464231641,
0.6202144437053256, 0.01514168625072241, 0.3323197255314305, 0.04119496913123866,
0.4218691024119261, 0.6191678586411691, -0.012512219958530463, 0.4853688365774933,
0.371536160561408, 0.8890761996014739, 0.40365175561133726, 0.02749686044745725,
0.24014939342793318, 0.871771768982168, 0.44659279700556415, 0.3381137076106839,
0.264164463985863, 0.247141697876908, 0.10349947943728739, 0.5656651658149489,
0.1399662088902055, 0.14138062892875827, 0.13131759994863923, 0.20998023040947894,
0.8549864033446407, 0.13484946503420467, -0.019635689133346176, 0.9482098637295892,
0.45002520877417146]
[0.969983812108392, 0.7876144343672766, 0.4703080545389494, 0.4242434783378429,
-0.008237311062962169, 0.417030333153629 4, 0.2403134783104408, 0.3790086346030469,
0.85824369092743, 0.618652593314948, 0.9415661897070103, 0.7634079809405896,
0.8426190753492041, 0.10138385151790302, 0.5131981089090044, 0.8225324066482965,
0.7769592764486889, 0.8188516945678803, 0.4226849913036419, 0.12187473585521828,
0.5094094264545769, 0.5595172887334108, 0.7623762794739238, 0.3674024734789103,
0.8974494421565649, 0.8756363240772165, 0.7448933428463025, 0.7383578467143339,

0.9799380405754695, 0.08418451066382297, 0.34990537919900516, 0.44189925018498377,
0.42187244383561184, 0.6760298106217631, 0.43991144717360825, 0.3719793507105564,
0.017387846000685686, 0.04379318893592699, 0.48830450415812904, 0.3226225283998968,
0.23101722525890267, 0.019581440138484247, 0.07488319446044214, 0.17261090381129074,
0.4642122286987968, 0.22761012089669952, 0.4606182673702371, 0.7411674334760908,
0.07557893669392801, 0.6283099830648359, 0.7548758911617024, 0.8984880872177726,
0.9829882994900843, 0.18065259501454403, 0.040569615751535504, 0.648514915659751,
0.5437695350797416, 0.9057377957381704, 0.5961446994810309, 0.8192758089026708,
0.0011649461117175894, 0.46101318602277586, 0.19535559791145093, 0.18107478470351587,
0.06275657995413403]
[0.945655958120173, 0.39169268537179197, 0.7953937040436992, 0.772666393811084,
0.8055472326308646, 0.29828620715904564, 0.7669819892469842, 0.45457529940027597,
0.9345831602672291, 0.6806376184052226, 0.8962654419258432, 0.9523813694280543,
0.3576567992378874, 0.3556212377167634, 0.0027378094663761334, 0.832688462291885,
0.6175933553600116, 0.16426611845547587, 0.6196282276150168, 0.9416764408329902,
0.9287332682825491, 0.9913877540335925, 0.33697773858498076, 0.25058251495836215,
0.0031931167482028844, 0.1421286458490337, 0.2532185374753985, 0.6230839460389425,
0.00625237791287469, 0.9288183025254089, 0.6481831830432288, 0.8575246517307803,
0.2845384197119065, 0.5235216114981854, 0.913044579184284, 0.10933624864344152,
0.07722786827283516, 0.14716446165773908, 0.20249548955474275, 0.1781856781863711,
0.031227375833419502, 0.1568613673116619, 0.44086599930495873, 0.3231580840757903,
0.446851269858572, 0.49853564350766866, 0.7070938846492167, 0.5786361107799265,
0.9708457257108402, 0.9448335488184142, 0.8998891389773245, 0.5222406527965127,
0.24258614154927188, 0.4516339984730561, 0.4928998691540892, 0.5139734439954883,
0.939687044259812, 0.6704342141374043, 0.9350245966243802, 0.14944904619698,
0.6383892631516743, 0.2668714006470411, 0.020485157459284805, 0.004185165616124922,
0.36955431440065184]
[0.18407964003735244, 0.3285737041249625, 0.0028752364452135715, -0.06010308802621361,
0.09682099761644378, 0.5720548327967574, 0.458050800549288, 0.7219698995220968,
0.9157559919715841, 0.5172575575983251, 0.5599981214520406, 0.10176040160601435,
0.26400859317628755, 0.6707790843371789, 0.788809122213041, 0.1444394913317013,
0.19387551333521935, 0.46140480402447515, 0.3212873310837706, 0.47261529126628465,
0.855176760615542, 0.25901926952290005, 0.09472048787713476, 0.08814913579141315,
0.31223555298632866, 0.3261892680496183, -0.0849741198397964, 0.6454094179170531,
0.29458349276912105, 0.22220536182727646, 0.2825660565747037, 0.8835754760590351,
0.6290911757177926, 0.8033864551213821, 0.017484558705722505, 0.07353332818985457,
0.32049409457581923, 0.39006051746089304, 0.12494376310442862, 0.19693347689080953,
0.07936368460869936, 0.07266622979887072, 0.18423317962146143, 0.24946076700351555,
0.18883872064722088, 0.5001177995624249, 0.9930125527756765, 0.5734579331433408,
0.4789692003962206, 0.3859151243616119, 0.11830940495282623, 0.6295308385386174,
0.35076581778696736, 0.6315122388454438, 0.43214464384823575, 0.21193816352523584,
0.3334928078771514, 0.14996971066179388, 0.23337711691459362, 0.8520830969471368,
0.0982112092579579, 0.6500226397688985, 0.9859552615764291, 0.4030886684901397,
0.20282074190028324]
[0.10744033479042325, 0.13258695174989396, 0.9247517956444574, 0.46952490321419554,
0.3273635128758902, 0.0435357637451629, 0.7083505884848978, 0.13413887809403247,
0.49894514314364113, 0.7542679723684799, 0.7999275958243748, 0.8596591493525785,
0.261634102049977, 0.15874089224942517, 0.11881638250625096, 0.31779077070587836,
0.34165020520001244, 0.19610246265124093, 0.12212765284615776, 0.4190779528476755,

0.4278701077843108, 0.1365774535599053, 0.8203481889105843, 0.0888380697507758,
0.38861722520906883, 0.3863819428786136, 0.39542452514637233, 0.6208020175919431,
0.5212702975413793, 0.6041286421911156, 0.7320838623355554, 0.09324176434074105,
0.9654517597410612, 0.5235730260585948, 0.7544662189611785, 0.20242525022616778,
0.47741832351272834, 0.7721754802843529, 0.3170769510134583, 0.5573963595155415,
0.8724475354616752, 0.5284285241470569, 0.5250934440362639, 0.7026283690557382,
0.9262796236448507, 0.2615700111413272, 0.9739777827620186, 0.7404976228016241,
0.39321746039884, 0.17052331036434654, 0.25895730933995625, 0.40753873238831334,
0.7102263632063798, 0.40700482482437783, 0.0543052225087694, 0.08164354336293797,
0.9393355068865382, 0.07950680933288982, 0.8996367020138186, 0.9125791222089484,
0.6139482126896665, 0.664411382781334, 0.8542626785864986, 0.835946667942128,
0.7161652111922691]
[0.7539493361821394, 0.9223891886077237, 0.2347698415126661, 0.39709408821751896,
0.7486377125242546, 0.6890680096923322, 0.4031794000791576, 0.9973004663510214,
0.046405078815649814, 0.1838476638354644, 0.13883342952937833, 0.32527551194787424,
0.9256756899951236, 0.9916564585074095, 0.7711780555382598, 0.4206411767944471,
0.9292856716568809, 0.3286075951251514, 0.5321566690028037, 0.7105633136812696,
0.3314053449156269, 0.7061815287686688, 0.9162023891979612, 0.35046166128768863,
0.7743428377862625, 0.07031490910319954, 0.9112612564128993, 0.4644157475214324,
0.3440529772560206, 0.9577495412128894, 0.0924421892298485, 0.515443763140301,
0.9547067986279703, 0.3775285039274238, 0.26379527007595566, 0.44547613924238877,
0.6548920421496318, 0.2669212967767734, 0.9442730961954559, 0.9155413285955938,
0.7843431453464802, 0.5276279909710425, 0.4165572062135734, 0.21873343137922413,
0.342494850025229, 0.46973262757561124, 0.3324566568676061, 0.519570013527581,
0.011572257762923144, 0.8056779444759549, 0.7481588192841705, 0.7707124698871395,
0.08423388200051309, 0.2630661440315428, 0.2165088384858733, 0.564822088425704,
0.9626300889612249, 0.35782798198064575, 0.8280806066406661, 0.027670223492719657,
0.6588768271469746, 0.3747000224174558, 0.3437056348065882, 0.6615154385558586,
0.8314903574908931]
[0.9335133848427308, 0.40861240087022327, 0.4964549309540353, 0.4412884138201523,
0.041512338621571436, 0.5647852798851848, 0.7444300416864922, 0.9196199090090872,
0.21247562339176085, 0.5537716489781093, 0.7802502004107232, 0.18341898366057496,
0.26173275577747135, 0.40889813086725396, 0.6786843244928781, 0.84781341469113,
0.29236084208554713, 0.6041995823003181, 0.9653679074853997, 0.9551893410308848,
0.8113061107285425, 0.39844986078306904, 0.7002803809305228, 0.4815997490387537,
0.5146693063624848, 0.36074667576327374, 1.003102241771978, 0.7570859907650199,
0.5263098270754467, 0.023212154608127575, 0.32345750420208563, 0.8601840171411504,
0.09074935460635203, 0.8847325628090774, 0.44592501191722617, 0.07381109540783046,
0.4544327219658966, 0.7944290438301358, 0.051340549428374, 0.4941855752874449,
0.6894658989772463, 0.43607816075728684, 0.2829554170043066, 0.585783549631944,
0.5379010595450965, 0.1223489803603791, 0.8935459960991414, 0.8541111396709337,
0.5340053704304839, 0.9192812449969434, 0.6018471759776004, 0.01326316852569964,
0.9313530492477581, 0.6587001286622707, 0.14287925538807958, 0.021569966681530314,
0.7518818000494101, 0.3809044982565362, 0.12012296761220194, 0.6551026160564061,
0.5388326388188822, 0.8045408155547471, 0.5892753892279211, 0.44978021709667754,
0.31218359368047205]
[0.991765010557749, 0.9514371393057501, 0.051456880678124726, 0.980872168990607,
0.7591747074676248, 0.2514500966693003, 0.7796470255358041, 0.6497143109340363,
0.6283937463142392, 0.8705114516151296, 0.2890658287564776, 0.8117933678091259,

0.5716692009336782, 0.3837439564841075, 0.972245252208247, 0.07564776134445228,
0.5876789025223378, 0.8414099016668681, 0.056417324404588444, 0.8654664412354842,
0.5363095844062643, 0.5597588385369963, 0.3112430570704226, 0.767808713261574,
0.14373151580942467, 0.4865236416267821, 0.3687119990411185, 0.5488929885343741,
0.20790334804243404, 0.5432565213226587, 0.014051887302413742, 0.6254098241195082,
0.5891570855144926, 0.1447641640163353, 0.316414067580066, 0.44350274234031845,
0.8408558825002217, 0.9889780484603662, 0.5910823639629802, 0.7844750350228243,
0.3703017212887001, 0.903713424501113, 0.9466498568660227, 0.9932845352662645,
0.625984727439435, 0.0828791533671551, 0.2128497999348036, 0.2228609118352578,
0.6096157407170201, 0.346920907715992, 0.42695165852778305, 0.7615942145238643,
0.09863808454243624, 0.028065348510279, 0.23148821921860996, 0.646231182116383,
0.24005310778537314, 0.9280409981825046, 0.516770676595653, 0.5383066009623005,
0.5077583913922653, 0.06417032223461962, 0.6365473754413155, 0.5866350146580775,
0.7080761776344409]
[0.33231993292432166, -0.40731562555325235, -0.9400120613787978, -0.561976076615292,
-1.3243033035392444, -0.6021283712670719, 25.798329337054362, -1.2060081671459741,
-1.2888815267337053, -1.3347621307454858, -0.9828117735178389, -0.5622456219371839,
-1.15474444587912, -0.7078958341704742, -0.5866602918668328, -1.097768582463778,
-0.4470061016270011, -0.7120283174139393, -0.9252100179264816, -1.1130977406870544,
-1.1873385408803827, -0.5166875420784043, -0.5392233614912952, -1.1917954276974774,
-1.3224432659166012]
[-8.586434538832549, -0.12101426491080089, -0.008366109421135698, 0.5351566442237832,
0.3866465688023888, -0.3781121564997748, -0.7717661189868874, 0.5907295892517184,
0.5660885737429724, 0.1224086697042304, 0.2762278055093289, 0.5410871905462611,
-0.1821281987311195, 0.6339135603838724, -0.07940285644145989, -0.08180451442118263,
-0.07900428879171205, 0.2969034743499171, 0.3501567883357249, -0.1480947837566685,
-0.15769647051096497, 0.6317774386039505, 0.437692485045212, -0.10028603553769248,
0.44688202175860176]
[5.7259900946221585, -0.020112466684693, 0.31870111198646983, 0.1900041250141514,
-0.3219222548917026, 0.3042611115093956, -26.860673686698348, 0.17901411250565222,
-0.059352032790649045, -0.1828070363188882, -0.4254456240603252, -0.09922921140209275,
-0.09659202041309893, -0.3158003244852662, -0.3778342285618486, 0.09069026127359005,
-0.3237204707126715, -0.009741720319714459, 0.16692296972519485, 0.405686034297984,
-0.08631675027128254, -0.045633706599782546, -0.5358200046453383, -0.3901167067457593,
-0.22775450553195806]

TASK 2

Program:

Training Non-Linear SVM classifier using rbf(Gaussian Kernel)

```
#!/usr/bin/python

from datetime import date
import numpy as np
from sklearn.model_selection import cross_val_score
import matplotlib.pyplot as plt
import pandas
from sklearn.cross_validation import train_test_split
```

```
from matplotlib.colors import ListedColormap
from sklearn.metrics import accuracy_score
from sklearn import model_selection
from sklearn.model_selection import KFold
from sklearn.svm import SVC

Cs=[5,50,100]
degrees=[2,3,4]

dataset=pandas.read_csv('Data_SVM.csv')
classes=dataset.values
maxFinalAccuracyMean=0.0
maxFinalStDev=0.0
cmax=0
degreesmax=0
print "C        degree        Accuracy Mean        Accuracy_Standard_Deviation"
print"_____"

#Training Non-Linear SVM classifier using rbf(Gaussian Kernel)
for Ci in Cs:
   for degreei in degrees:

      n_epochs=30
      finalAccuracyMean=0.0
      finalAccuracyStdev=0.0
      for i in range(n_epochs):
         features=dataset.iloc[:,: -1].values
         labels=dataset.iloc[:, 2].values

         x_tr,x_test,y_tr,y_test=train_test_split(features,labels,test_size=0.2,random_state=0)



         svmPoly=SVC(C=Ci,kernel='rbf',random_state=0,degree=degreei)
         svmPoly.fit(x_tr,y_tr)

         y_pred=svmPoly.predict(x_test)

         accuracies=cross_val_score(estimator=svmPoly,X=x_tr,y=y_tr,cv=10)
         finalAccuracyMean+=accuracies.mean()
         finalAccuracyStdev+=accuracies.std()

      finalAccuracyMean/=30
      finalAccuracyStdev/=30
      print str(Ci)+"        "+str(degreei)+"        "+str(finalAccuracyMean)+"        "+str(finalAccuracyStdev)

      if(finalAccuracyMean>maxFinalAccuracyMean):
         cmax=Ci
         degreesmax=degreei
         maxFinalAccuracyMean=finalAccuracyMean
         maxFinalStDev=finalAccuracyStdev

#Running SVM using rbf and C and D found corresponding to maximum accuracy
svmPoly=SVC(C=cmax,kernel='rbf',random_state=0,degree=degreesmax)

features=dataset.iloc[:,: -1].values
labels=dataset.iloc[:, 2].values
```

```
svmPoly.fit(features,labels)

accuracies=cross_val_score(estimator=svmPoly,X=features,y=labels,cv=10)

print
print "Maximum Accuracy mean corresponding to C ="+str(cmax)+"and d= "+str(degreesmax)
+"is:"+str(maxFinalAccuracyMean)
print "Maximum Accuracy mean corresponding to C ="+str(cmax)+"and d= "+str(degreesmax)
+"is:"+str(maxFinalStDev)


#Plotting the results
X_set,Y_set=features,labels
X1,X2=np.meshgrid(np.arange(start=X_set[:,0].min()-1,stop=X_set[:,0].max()+1,step=0.01),
                  np.arange(start=X_set[:,1].min()-1,stop=X_set[:,1].max()+1,step=0.01))
plt.contourf(X1,X2,svmPoly.predict(np.array([X1.ravel(),X2.ravel()]).T).reshape(X1.shape),
             alpha=0.75,cmap=ListedColormap(('red','green')))
plt.xlim(X1.min(),X1.max())
plt.ylim(X2.min(),X2.max())

for i,j in enumerate(np.unique(Y_set)):

    plt.scatter(X_set[Y_set==j,0],X_set[Y_set==j,1],
                c=ListedColormap(('red','green'))(i),label=j)

plt.title('SVM Classification {Test Set}')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.show()
```

OutPut:

```
nishu@nishu-Inspiron-15-3567:~$ cd Desktop/
nishu@nishu-Inspiron-15-3567:~/Desktop$ cd q2/
nishu@nishu-Inspiron-15-3567:~/Desktop/q2$ clear

nishu@nishu-Inspiron-15-3567:~/Desktop/q2$ ./svm
bash: ./svm: No such file or directory
nishu@nishu-Inspiron-15-3567:~/Desktop/q2$ ./svm.py
/home/nishu/.local/lib/python2.7/site-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 i
n favor of the model_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV
 iterators are different from that of this module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
C           degree          Accuracy Mean            Accuracy_Standard_Deviation
_____________________________________________________________________________
5           2              0.980833333333            0.0292973263854
5           3              0.980833333333            0.0292973263854
5           4              0.980833333333            0.0292973263854
50           2              0.9675           0.0441273409829
50           3              0.9675           0.0441273409829
50           4              0.9675           0.0441273409829
100           2              0.9675           0.0441273409829
100           3              0.9675           0.0441273409829
100           4              0.9675           0.0441273409829

Maximum Accuracy mean corresponding to C =5and d= 2is:0.980833333333
Maximum Accuracy mean corresponding to C =5and d= 2is:0.0292973263854
```

Training Non-Linear SVM classifier using Polynomial Kernel

```
#!/usr/bin/python

from datetime import date
import numpy as np
from sklearn.model_selection import cross_val_score
import matplotlib.pyplot as plt
import pandas
from sklearn.cross_validation import train_test_split
from matplotlib.colors import ListedColormap
from sklearn.metrics import accuracy_score
from sklearn import model_selection
from sklearn.model_selection import KFold
from sklearn.svm import SVC

Cs=[5,100,20000]
degrees=[2,3]

dataset=pandas.read_csv('Data_SVM.csv')
classes=dataset.values
maxFinalAccuracyMean=0.0
maxFinalStDev=0.0
cmax=0
degreesmax=0
print "C        degree        Accuracy Mean        Accuracy_Standard_Deviation"
print"_____"

#Training Non-Linear SVM classifier using Polynomial Kernel
for Ci in Cs:
   for degreei in degrees:

      n_epochs=30
      finalAccuracyMean=0.0
      finalAccuracyStdev=0.0
      for i in range(n_epochs):
         features=dataset.iloc[:,: -1].values
         labels=dataset.iloc[:, 2].values

         x_tr,x_test,y_tr,y_test=train_test_split(features,labels,test_size=0.2,random_state=0)



         svmPoly=SVC(C=Ci,kernel='poly',random_state=0,degree=degreei)
         svmPoly.fit(x_tr,y_tr)

         y_pred=svmPoly.predict(x_test)

         accuracies=cross_val_score(estimator=svmPoly,X=x_tr,y=y_tr,cv=10)
         finalAccuracyMean+=accuracies.mean()
         finalAccuracyStdev+=accuracies.std()

      finalAccuracyMean/=30
      finalAccuracyStdev/=30
      print str(Ci)+"        "+str(degreei)+"        "+str(finalAccuracyMean)+"        "+str(finalAccuracyStdev)

      if(finalAccuracyMean>maxFinalAccuracyMean):
         cmax=Ci
```

```
        degreesmax=degreei
        maxFinalAccuracyMean=finalAccuracyMean
        maxFinalStDev=finalAccuracyStdev

#Running SVM using polynimial kernel and C and D found corresponding to maximum accuracy
svmPoly=SVC(C=cmax,kernel='poly',random_state=0,degree=degreesmax)

features=dataset.iloc[:,: -1].values
labels=dataset.iloc[:, 2].values

svmPoly.fit(features,labels)

accuracies=cross_val_score(estimator=svmPoly,X=features,y=labels,cv=10)
print
print "Maximum Accuracy mean corresponding to C ="+str(cmax)+"and d= "+str(degreesmax)
+"is:"+str(maxFinalAccuracyMean)
print "Maximum Accuracy mean corresponding to C ="+str(cmax)+"and d= "+str(degreesmax)
+"is:"+str(maxFinalStDev)


#Plotting the results
X_set,Y_set=features,labels
X1,X2=np.meshgrid(np.arange(start=X_set[:,0].min()-1,stop=X_set[:,0].max()+1,step=0.01),
            np.arange(start=X_set[:,1].min()-1,stop=X_set[:,1].max()+1,step=0.01))
plt.contourf(X1,X2,svmPoly.predict(np.array([X1.ravel(),X2.ravel()]).T).reshape(X1.shape),
        alpha=0.75,cmap=ListedColormap(('red','green')))
plt.xlim(X1.min(),X1.max())
plt.ylim(X2.min(),X2.max())

for i,j in enumerate(np.unique(Y_set)):

    plt.scatter(X_set[Y_set==j,0],X_set[Y_set==j,1],
            c=ListedColormap(('red','green'))(i),label=j)

plt.title('SVM Classification {Test Set}')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.show()
```

Output:

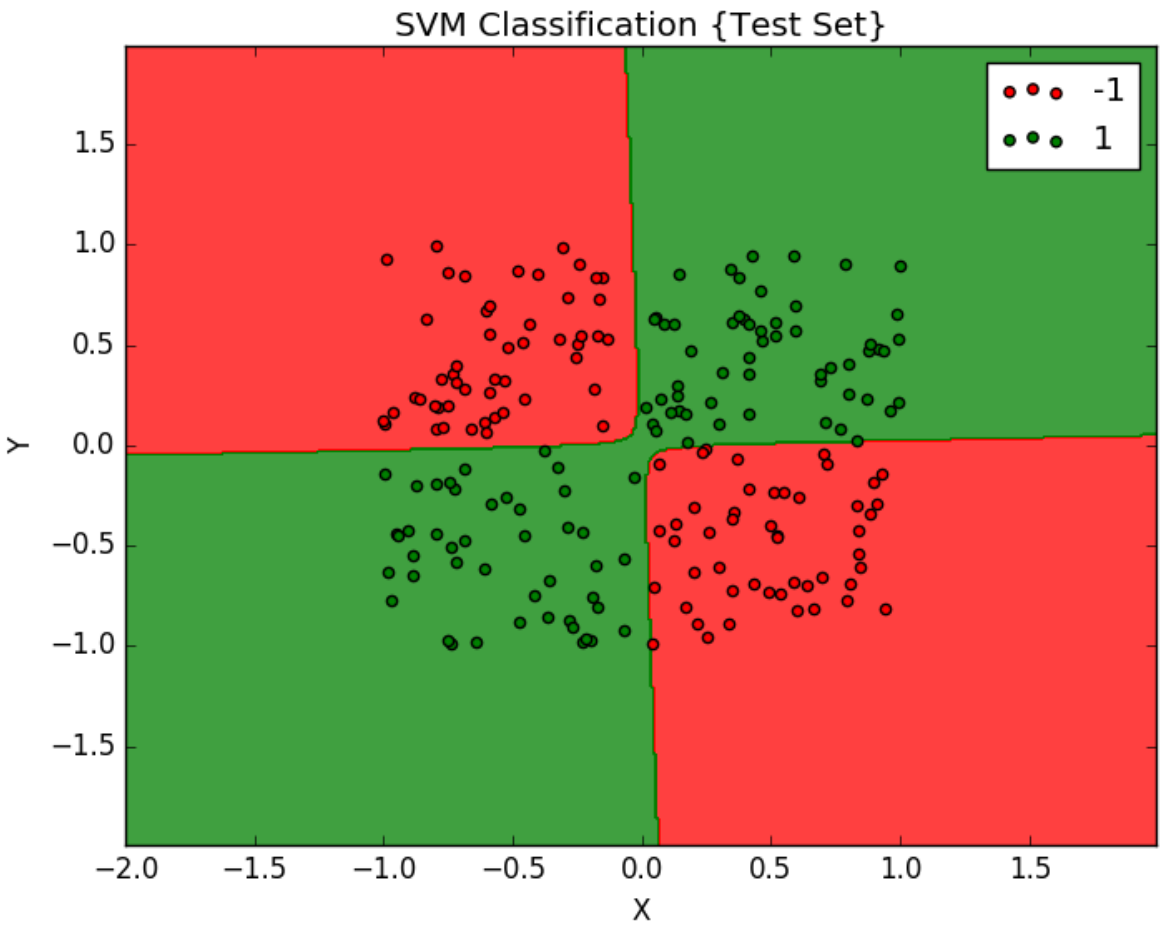
```
nishu@nishu-Inspiron-15-3567:~/Desktop/q2$ ./svmPoly.py
/home/nishu/.local/lib/python2.7/site-packages/sklearn/cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 i
n favor of the model_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV
 iterators are different from that of this module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
C           degree          Accuracy Mean           Accuracy_Standard_Deviation
_____________________________________________________________________________
5           2             0.967916666667            0.0425755048251
5           3             0.592328431373            0.0754940992434
100           2             0.967916666667            0.0425755048251
100           3             0.617745098039            0.0690253869016
20000           2             0.980416666667            0.0299333750928
20000           3             0.617745098039            0.0690253869016

Maximum Accuracy mean corresponding to C =20000and d= 2is:0.980416666667
Maximum Accuracy mean corresponding to C =20000and d= 2is:0.0299333750928
nishu@nishu-Inspiron-15-3567:~/Desktop/q2$
```

Part 1:

Plotted the data.

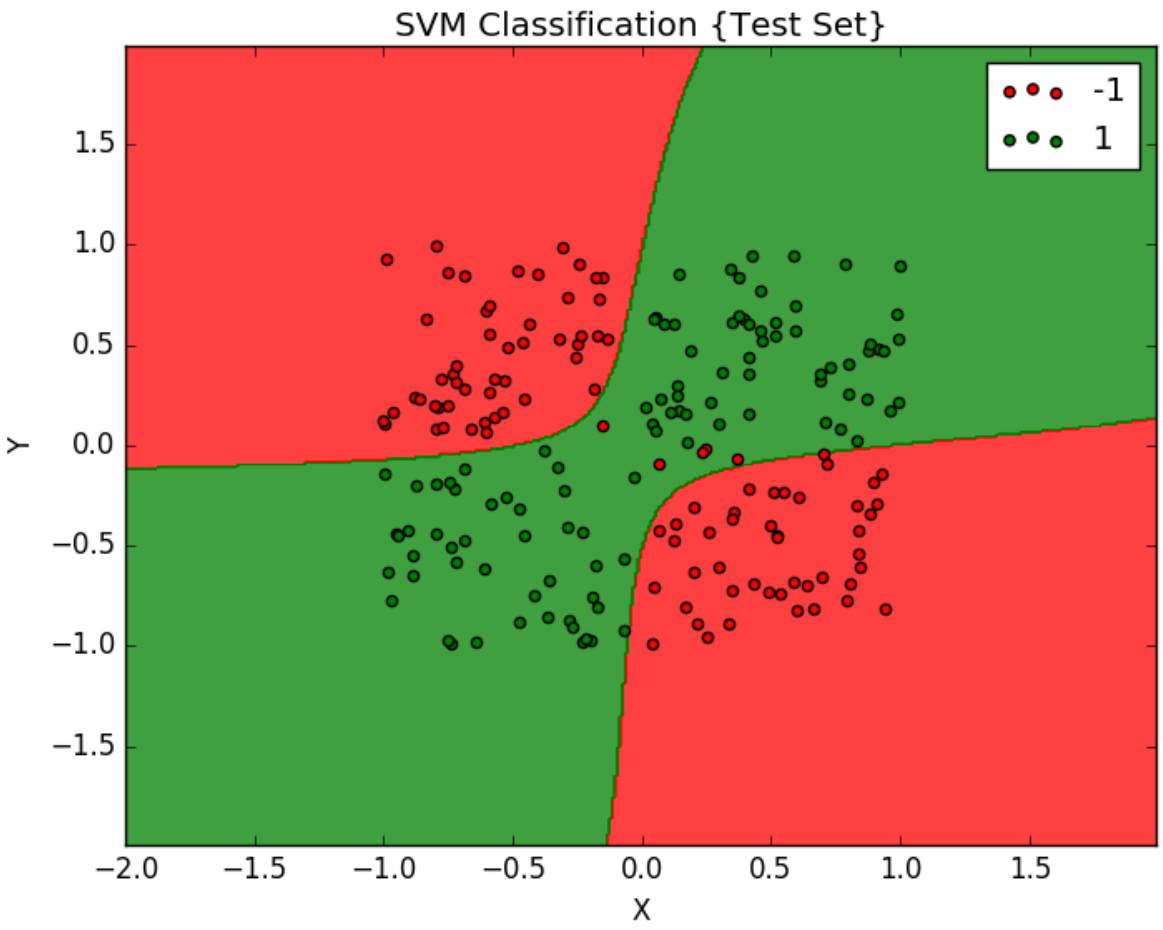
Part 2:

Trained nonlinear SVM classifier using polynomial kernel. Varied the values of C and d (degree of polynomial) in some range. For each combination of C and d, run 10-fold cross validation 30 times and report the average cross validation accuracy and standard deviation. Made a chart for that.
Find the best combination of C and d.
Used these parameters and train SVM using complete dataset. Plotted the final classifier.



Part 3:

Trained nonlinear SVM classifier using rbf (Gaussian) kernel. Varied the values of C and d (degree of polynomial) in some range. For each combination of C and d, run 10-fold cross validation 30 times and report the average cross validation accuracy and standard deviation. Made a chart for that.
Find the best combination of C and d.
Used these parameters and train SVM using complete dataset. Plotted the final classifier.



Part 4:

Compared the performances with different kernels and comment on the quality of classifiers in each case.

Polynomial Classifier is performing better than RBF.

TASK 3

Implemented a Naive Bayes Classifier for UCI Census-Income (KDD) Data Set using only the Discrete and Categorical attributes/features.

Program:

```
#!/usr/bin/python

from csv import reader
from collections import Counter
from math import sqrt
from sklearn.model_selection import KFold
```

```
import math

fname = 'census-income.csv'

def loadData(fname):
        dataset=list()
        with open(fname, 'r') as file:
                csv_reader = reader(file)
                for row in csv_reader:
                        if not row:
                                continue
                        dataset.append(row)
        return dataset

def calculateTotal(tset):
        total=0
        for row in tset:
                total+=1
        return total

def summarize(tset):
        summaries={"mean":{},"mode":{},"stdev":{},"freq":{},"classTotal":0,"classProbability":0.0,"dataTotal":0}
        summaries["classTotal"]=calculateTotal(tset)
        for column in range(len(tset[0])-1):
                if continuous_or_nominal[column]==1:
                        summaries["mean"][column]=calculate_mean(column,tset)
                        summaries["stdev"][column]=calculate_stdev(column,tset,summaries["mean"][column])
                else:
                        summaries["mode"][column]=calculate_mode(column,tset)
                summaries["freq"][column]=calculate_freq(column,tset)
        return summaries

def is_number(s):
   try:
      float(s)
      return True
   except ValueError:
      return False

def calculate_mean(column,tset):
        number_of_non_missing_entries=0
        sum_of_non_missing_entries=0
        for row in tset:
                if is_number(row[column]):
                        number_of_non_missing_entries+=1
                        sum_of_non_missing_entries+=row[column]

        mean=sum_of_non_missing_entries/number_of_non_missing_entries
        return mean

def calculate_mode(column,tset):
        wordfreq={}
        for row in tset:
                if '?' in row[column]:
                                pass
                else:
                        if row[column] not in wordfreq:
                                wordfreq[row[column]] = 0
```

```
                else:
                        wordfreq[row[column]] += 1
        maximum = [(value, key) for key, value in wordfreq.items()]
        return max(maximum)[1]

def calculate_stdev(column,tset,mean):
        avg = mean
        number_of_entries=0
        variance=0.0
        for row in tset:
                variance+=pow(row[column]-avg,2)
                number_of_entries+=1
        variance = variance/number_of_entries
        return sqrt(variance)

def calculate_freq(column,tset):
        wordfreq={}
        for row in tset:
                if row[column] not in wordfreq:
                        wordfreq[row[column]] = 0
                else:
                        wordfreq[row[column]] += 1
        maximum = [(value, key) for key, value in wordfreq.items()]
        return wordfreq

def separateByClass(tset):
        separated = {}
        for row in tset:
                if (row[-1] not in separated):
                        separated[row[-1]] = []
                separated[row[-1]].append(row)
        return separated

def summarizeByClass(tset):
        separated = separateByClass(tset)
        summariesByClass = {}
        total=calculateTotal(tset)
        for classValue, instances in separated.iteritems():
                summariesByClass[classValue] =summarize(instances)
                summariesByClass [classValue]["dataTotal"]=total
                summariesByClass[classValue]["classProbability"]=float(summariesByClass[classValue]
["classTotal"])/float(summariesByClass[classValue]["dataTotal"])
        return summariesByClass

def calculateProbability(x, mean, stdev):
        exponent = math.exp(-(math.pow(x-mean,2)/(2*math.pow(stdev,2))))
        res=float(1 / (math.sqrt(2*math.pi) * stdev) * exponent)
        if(res>0.0):
                return math.log(res)
        else:
                return 0.0

def calculateProbability_nom(freq, classTotal):
        if(freq>0):
                return math.log(float(freq)/float(classTotal))
        else:
                return 0.0
```

```
def calculateClassProbabilities(summaries, tseti):
	probabilities = {}
	for classValue, classSummaries in summaries.iteritems():
		probabilities[classValue] = 1
		for column in range(len(tseti)-1):
			if column==24:
				continue
			if continuous_or_nominal[column]==1:
				mean=classSummaries["mean"][column]
				stdev=classSummaries["stdev"][column]
				x = tseti[column]
#				probabilities[classValue] += calculateProbability(x, mean, stdev)
			else:
				x = tseti[column]
				if(x in (classSummaries["freq"][column])):
					freq= classSummaries["freq"][column][x]
					classTotal=classSummaries["classTotal"]
					probabilities[classValue] += calculateProbability_nom(freq, classTotal)

				else:
					continue
		probabilities[classValue] += math.log(classSummaries["classProbability"])
	return probabilities

def predict(summaries, tseti):
	probabilities = calculateClassProbabilities(summaries, tseti)
	bestLabel, bestProb = None, -1
	for classValue, probability in probabilities.iteritems():
		if bestLabel is None or probability > bestProb:
			bestProb = probability
			bestLabel = classValue
	return bestLabel

def getPredictions(summaries, tSet):
	predictions = []
	for i in range(len(tSet)):
		result = predict(summaries, tSet[i])
		predictions.append(result)
	return predictions

def getAccuracy(testSet, predictions):
	correct = 0
	for i in range(len(testSet)):
		if testSet[i][-1] == predictions[i]:
			correct += 1
	return (correct/float(len(testSet))) * 100.0

####################
####################
##    MAIN     ##
####################
####################

dataset=loadData(fname)

#Here 1 suggests continuous data and 0 suggests nominal(categorial) data in column
```

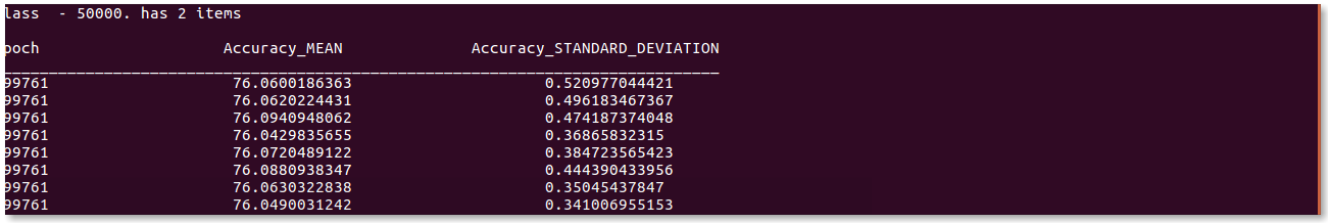
```
continuous_or_nominal={0:1,1:0,2:0,3:0,4:0,5:1,6:0,7:0,8:0,9:0,10:0,11:0,12:0,13:0,14:0,15:0,16:1,17:1,18:1,19:0,20:0,
21:0,22:0,23:0,24:1,25:0,26:0,27:0,28:0,29:0,30:1,31:0,32:0,33:0,34:0,35:0,36:0,37:0,38:0,39:1,40:0}

#Summary of whole data
FullDatasetSummary={"mean":{},"mode":{},"stdev":{},"freq":{},"total":0, "numberofEmptyEntries":{}}

################################################################################
# calculation of Full data summary and modifying empty fields with mean in #
# continuous data columns and mode in nominal(categorical) data column     #
################################################################################

print "column_no          mean/mode              stdev          Number of Empty Entries"
print "_____"

FullDatasetSummary["total"]=calculateTotal(dataset)
for column in range(len(dataset[0])-1):
        tempEmptyEntries=0
        if continuous_or_nominal[column]==1:  #Continuous data
                for row in dataset:
                        if is_number(row[column]):
                                row[column]=float(row[column])  #non empty entry made float
                FullDatasetSummary["mean"][column]=calculate_mean(column,dataset)
                #Filling with mean in empty entries
                for row in dataset:
                        #print row[column]
                        float(row[column])
                        if is_number(row[column]):
                                pass
                        else:
                                tempEmptyEntries+=1
                                row[column]=FullDatasetSummary["mean"][column]
                FullDatasetSummary["stdev"][column]=calculate_stdev(column,dataset,FullDatasetSummary["mean"]
[column])
                FullDatasetSummary["numberofEmptyEntries"][column]=tempEmptyEntries
                print str(column)+"\t\t\t"+str(FullDatasetSummary["mean"][column])
+"\t\t\t"+str(FullDatasetSummary["stdev"][column])+"\t\t\t"+str(FullDatasetSummary["numberofEmptyEntries"]
[column])

        else:                                                           # nominal(categorical) data
                FullDatasetSummary["mode"][column]=calculate_mode(column,dataset)
                #Filling with mode in empty entries
                for row in dataset:
                        if '?' in row[column]:
                                tempEmptyEntries+=1
                                row[column]=FullDatasetSummary["mode"][column]
                        else:
                                pass

                FullDatasetSummary["numberofEmptyEntries"][column]=tempEmptyEntries
                print str(column)+"\t\t\t"+str(FullDatasetSummary["mode"][column])
+"\t\t\tNA\t\t\t"+str(FullDatasetSummary["numberofEmptyEntries"][column])

                FullDatasetSummary["freq"][column]=calculate_freq(column,dataset)

#separate By classs
separated = {}
for row in dataset:
        if (row[-1] not in separated):
```

```
                separated[row[-1]] = []
        separated[row[-1]].append(row)

print
print "Number of classes= "+str(len(separated))

total=0
for i in separated:
        print "Class "+str(i)+" has "+str(len(separated[i]))+" items"
        total+=len(separated[i])
        print
print "Database "+str(i)+" has "+str(total)+" items"
print
for column in range(len(dataset[0])-1):
        if continuous_or_nominal[column]==1:  #Continuous data
                pass
        else:
                print column
                print FullDatasetSummary["freq"][column]



################################################################################
# Calculating the accuarcy by running naive bayes of train data and       #
# testing on test data                                    #
################################################################################
print
print "epoch                Accuracy_MEAN           Accuracy_STANDARD_DEVIATION"
print "_____________________________________________________________________________"
n_epochs=30
j=0
for i in range(n_epochs):
        j=j+1
        accuracies=[]
        folds = KFold(n_splits=10, shuffle=True)
        for train_set , test_set in folds.split(dataset):
                x_tr=[]
                x_test=[]
                for i in test_set:
                        x_test.append(dataset[i])
                for i in train_set:
                        x_tr.append(dataset[i])
                summariesByClass = summarizeByClass(x_tr)
                predictions = getPredictions(summariesByClass, x_test)
                accuracy = getAccuracy(x_test, predictions)
                accuracies.append(accuracy)

        number_of_accuracies=0
        sum_of_accuracies=0
        for acc in accuracies:
                number_of_accuracies+=1
                sum_of_accuracies+=acc
        accuracies_mean=sum_of_accuracies/number_of_accuracies

        avg =accuracies_mean
        number_of_accuracies=0
        accuracies_variance=0.0
```

```
    for acc in accuracies:
        accuracies_variance+=pow(acc-avg,2)
        number_of_accuracies+=1
accuracies_variance = accuracies_variance/number_of_accuracies
accuracies_variance=sqrt(accuracies_variance)
print " "+str(j)+"          "+str(accuracies_mean)+"          "+str(accuracies_variance)
```

OutPut:

With considering continuous data and calculation gaussion probabiility for that.



```
lass  - 50000. has 2 items

poch                Accuracy_MEAN          Accuracy_STANDARD_DEVIATION
_____________________________________________________________________
99761                76.0600186363              0.520977044421
99761                76.0620224431              0.496183467367
99761                76.0940948062              0.474187374048
99761                76.0429835655              0.36865832315
99761                76.0720489122              0.384723565423
99761                76.0880938347              0.444390433956
99761                76.0630322838              0.35045437847
99761                76.0490031242              0.341006955153
```

epoch	Accuracy_MEAN	Accuracy_STANDARD_DEVIATION
99761	84.9471771241	0.449623570527
99761	84.9722414882	0.391635521185
99761	84.9521977935	0.405714276234
99761	84.9221261226	0.441525812494
99761	85.0193465219	0.302562463194
99761	84.9221170802	0.303981227388
99761	84.9361435271	0.430834369571
99761	84.9521860383	0.237340705311
99761	84.9361521676	0.336794772444

Without considering continuous data and calculation gaussion probabiility for that.

```
25                      Nonmover                        NA                      49946
26                      Nonmover                        NA                      49946
27                      Nonmover                        NA                      49946
28                      Not in universe under 1 year old                        NA                      0
29                      Not in universe                         NA                      49946
30                     1.95615565045                   2.3643194504                    0
31                      Not in universe                         NA                      0
32                      United-States                   NA                      3429
33                      United-States                   NA                      3072
34                      United-States                   NA                      1764
35                      Native- Born in the United States                       NA                      0
36                      0                       NA                      0
37                      Not in universe                         NA                      0
38                      2                       NA                      0
39                     23.185331088                    24.389062115                    0
40                      95                      NA                      0

Number of classes= 2
Class  50000+. has 2 items
Class  - 50000. has 2 items

epoch                   Accuracy_MEAN           Accuracy_STANDARD_DEVIATION
_____________________________________________________________________________
 99761                   76.0600186363                   0.520977044421
 99761                   76.0620224431                   0.496183467367
 99761                   76.0940948062                   0.474187374048
 99761                   76.0429835655                   0.36865832315
 99761                   76.0720489122                   0.384723565423
 99761                   76.0880938347                   0.444390433956
 99761                   76.0630322838                   0.35045437847
 99761                   76.0490031242                   0.341006955153
 99761                   76.0700495261                   0.395424027445
 99761                   76.0780655572                   0.323753468623
 99761                   76.0309627339                   0.495621887089
 99761                   76.0149091708                   0.355526573444
 99761                   76.0580223648                   0.399945725005
 99761                   76.0660383959                   0.675718554214
 99761                   76.0710480135                   0.541259516815
 99761                   76.0600178325                   0.597264359467
 99760                   76.0269402505                   0.456382546509
 99761                   76.0690413934                   0.45786775173
 99761                   76.0610186307                   0.41262265751
```

Part 1:
Summary of the data.

Number of classes= 2
Class 50000+. has 6186 items

Class - 50000. has 93576 items

Database - 50000. has 99762 items

1
{' Private': 36070, ' Self-employed-incorporated': 1647, ' State government': 2166, ' Self-employed-not incorporated': 4279, ' Not in universe': 50078, ' Without pay': 74, ' Federal government': 1404, ' Never worked': 203, ' Local government': 3832}
2
{' 10': 0, ' 11': 902, ' 38': 845, ' 39': 1455, ' 14': 141, ' 15': 221, ' 16': 252, ' 17': 68, ' 32': 1821, ' 33': 8711, ' 30': 594, ' 31': 569, ' 36': 494, ' 37': 2138, ' 34': 1353, ' 35': 1664, ' 13': 466, ' 28': 62, ' 49': 261, ' 48': 290, ' 50': 820, ' 18': 206, ' 19': 720, ' 12': 679, ' 25': 482, ' 24': 730, ' 27': 294, ' 26': 56, ' 21': 280, ' 20': 19, ' 23': 273, ' 22': 468, ' 47': 803, ' 43': 4226, ' 45': 2203, ' 44': 1221, ' 29': 2210, ' 42': 2303, ' 41': 1850, ' 40': 820, ' 2': 1076, ' 3': 324, ' 0': 50282, ' 1': 380, ' 6': 241, ' 7': 246, ' 4': 3066, ' 5': 268, ' 8': 252, ' 9': 489, ' 46': 101, ' 51': 15}
3

{' 10': 1805, ' 11': 296, ' 12': 1692, ' 13': 614, ' 14': 417, ' 15': 438, ' 16': 1741, ' 17': 902, ' 32': 1188, ' 33': 1707, ' 30': 940, ' 31': 1375, ' 36': 2092, ' 37': 1092, ' 34': 2021, ' 35': 1623, ' 39': 480, ' 9': 370, ' 18': 573, ' 19': 2770, ' 25': 415, ' 24': 893, ' 27': 425, ' 26': 3813, ' 21': 272, ' 20': 30, ' 23': 1634, ' 22': 219, ' 43': 660, ' 45': 79, ' 44': 778, ' 29': 2670, ' 28': 786, ' 41': 803, ' 42': 929, ' 2': 4355, ' 3': 1655, ' 0': 50282, ' 1': 275, ' 6': 199, ' 7': 363, ' 4': 680, ' 5': 438, ' 8': 1017, ' 40': 336, ' 46': 15, ' 38': 1558}
4
{' 1st 2nd 3rd or 4th grade': 905, ' 12th grade no diploma': 1136, ' Less than 1st grade': 415, ' Some college but no degree': 13953, ' Masters degree(MA MS MEng MEd MSW MBA)': 3305, ' 10th grade': 3812, ' 11th grade': 3522, ' Bachelors degree(BA AB BS)': 9884, ' Prof school degree (MD DDS DVM LLB JD)': 875, ' Associates degree-academic program': 2078, ' 5th or 6th grade': 1713, ' Doctorate degree(PhD EdD)': 619, ' High school graduate': 24146, ' 7th and 8th grade': 4148, ' Children': 23441, ' Associates degree-occup /vocational': 2689, ' 9th grade': 3104}
6
{' Not in universe': 93423, ' College or university': 2831, ' High school': 3505}
7
{' Separated': 1695, ' Divorced': 6449, ' Widowed': 5324, ' Married-spouse absent': 715, ' Married-civilian spouse present': 42092, ' Married-A F spouse present': 338, ' Never married': 43142}
8
{' Mining': 324, ' Transportation': 2210, ' Medical except hospital': 2303, ' Retail trade': 8711, ' Finance insurance and real estate': 3018, ' Construction': 3066, ' Wholesale trade': 1821, ' Manufacturing-nondurable goods': 3393, ' Private household services': 494, ' Communications': 594, ' Utilities and sanitary services': 569, ' Not in universe or children': 50282, ' Forestry and fisheries': 101, ' Other professional services': 2203, ' Education': 4226, ' Personal services except private HH': 1455, ' Agriculture': 1457, ' Entertainment': 820, ' Public administration': 2177, ' Hospital services': 1850, ' Manufacturing-durable goods': 4444, ' Social services': 1221, ' Business and repair services': 2984, ' Armed Forces': 15}
9
{' Other service': 6176, ' Technicians and related support': 1471, ' Not in universe': 50282, ' Private household services': 425, ' Professional specialty': 6868, ' Protective services': 786, ' Farming forestry and fishing': 1519, ' Sales': 6020, ' Machine operators assmblrs & inspctrs': 3185, ' Transportation and material moving': 2039, ' Handlers equip cleaners etc ': 2070, ' Precision production craft & repair': 5353, ' Adm support including clerical': 7251, ' Armed Forces': 15, ' Executive admin and managerial': 6287}
10
{' Other': 1902, ' White': 83612, ' Asian or Pacific Islander': 2912, ' Amer Indian Aleut or Eskimo': 1208, ' Black': 10123}
11
{' Cuban': 615, ' Chicano': 169, ' Other Spanish': 1242, ' Mexican-American': 4000, ' NA': 404, ' Do not know': 139, ' Mexican (Mexicano)': 3694, ' Central or South American': 1986, ' Puerto Rican': 1581, ' All other': 85922}
12
{' Male': 47970, ' Female': 51790}
13
{' Not in universe': 90282, ' Yes': 1444, ' No': 8033}
14
{' Job loser - on layoff': 507, ' Other job loser': 1107, ' Job leaver': 285, ' Not in universe': 96631, ' New entrant': 203, ' Re-entrant': 1023}
15

{' PT for econ reasons usually FT': 257, ' Children or Armed Forces': 61689, ' Not in labor force': 13412, ' Full-time schedules': 20466, ' PT for econ reasons usually PT': 582, ' PT for non-econ reasons usually FT': 1734, ' Unemployed part- time': 410, ' Unemployed full-time': 1204}
19
{' Nonfiler': 37367, ' Joint one under 65 & one 65+': 1976, ' Joint both 65+': 4193, ' Single': 18893, ' Joint both under 65': 33571, ' Head of household': 3756}
20
{' Northeast': 1321, ' West': 2045, ' Not in universe': 92067, ' Midwest': 1697, ' Abroad': 202, ' South': 2424}
21
{' Connecticut': 64, ' South Dakota': 74, ' Iowa': 64, ' Wisconsin': 61, ' New Jersey': 37, ' Kansas': 69, ' Michigan': 198, ' Arizona': 136, ' Utah': 533, ' North Carolina': 397, ' District of Columbia': 55, ' California': 883, ' Massachusetts': 72, ' Montana': 106, ' Florida': 449, ' Abroad': 264, ' Alaska': 123, ' Vermont': 98, ' Kentucky': 113, ' Wyoming': 102, ' Nebraska': 101, ' New Mexico': 227, ' Alabama': 107, ' Missouri': 69, ' Ohio': 99, ' Virginia': 57, ' Colorado': 113, ' Illinois': 84, ' Tennessee': 94, ' Not in universe': 92397, ' Georgia': 110, ' West Virginia': 102, ' New Hampshire': 115, ' South Carolina': 48, ' Nevada': 79, ' Oregon': 109, ' Maine': 65, ' North Dakota': 225, ' Mississippi': 77, ' Delaware': 47, ' Oklahoma': 295, ' New York': 106, ' Minnesota': 292, ' Idaho': 13, ' Arkansas': 99, ' Indiana': 296, ' Pennsylvania': 98, ' Maryland': 78, ' Texas': 94, ' Louisiana': 118}
22
{' Child under 18 of RP of unrel subfamily': 352, ' Spouse of householder': 20654, ' Other Rel <18 never marr not in subfamily': 302, ' Grandchild 18+ spouse of subfamily RP': 3, ' Child 18+ ever marr RP of subfamily': 353, ' Child 18+ never marr RP of subfamily': 308, ' Spouse of RP of unrelated subfamily': 23, ' Secondary individual': 3155, ' Other Rel <18 never married RP of subfamily': 1, ' Child <18 ever marr not in subfamily': 14, ' Other Rel <18 ever marr not in subfamily': 1, ' Other Rel 18+ ever marr not in subfamily': 1020, ' Grandchild <18 never marr child of subfamily RP': 888, ' Other Rel <18 never marr child of subfamily RP': 302, ' Child <18 spouse of subfamily RP': 0, ' Grandchild 18+ ever marr not in subfamily': 19, ' Grandchild <18 never marr not in subfamily': 501, ' Householder': 26690, ' Grandchild 18+ never marr not in subfamily': 206, ' Child 18+ spouse of subfamily RP': 76, ' Child 18+ never marr Not in a subfamily': 6133, ' Other Rel 18+ ever marr RP of subfamily': 309, ' Grandchild 18+ never marr RP of subfamily': 0, ' Child <18 never marr not in subfamily': 24989, ' Other Rel 18+ never marr RP of subfamily': 62, ' Grandchild <18 never marr RP of subfamily': 0, ' Other Rel 18+ spouse of subfamily RP': 353, ' Child <18 never marr RP of subfamily': 38, ' Child 18+ ever marr Not in a subfamily': 464, ' Grandchild 18+ ever marr RP of subfamily': 5, ' Other Rel <18 spouse of subfamily RP': 1, ' In group quarters': 82, ' RP of unrelated subfamily': 339, ' Child <18 ever marr RP of subfamily': 5, ' Other Rel 18+ never marr not in subfamily': 833, ' Nonfamily householder': 11242, ' Other Rel <18 ever marr RP of subfamily': 2}
23
{' Child under 18 ever married': 21, ' Child under 18 never married': 25038, ' Spouse of householder': 20656, ' Child 18 or older': 7338, ' Nonrelative of householder': 3876, ' Group Quarters- Secondary individual': 54, ' Householder': 37942, ' Other relative of householder': 4829}
25
{' Abroad to nonMSA': 35, ' Not in universe': 722, ' MSA to MSA': 5288, ' MSA to nonMSA': 363, ' Nonmover': 91344, ' NonMSA to MSA': 294, ' NonMSA to nonMSA': 1343, ' Not identifiable': 199, ' Abroad to MSA': 165}
26
{' Different division same region': 225, ' Same county': 4878, ' Not in universe': 722, ' Different region': 582, ' Nonmover': 91344, ' Different county same state': 1328, ' Abroad': 202, ' Different state same division': 473}

27
{' Different state in South': 474, ' Same county': 4878, ' Not in universe': 722, ' Different state in Northeast': 207, ' Different state in Midwest': 241, ' Nonmover': 91344, ' Different county same state': 1328, ' Different state in West': 357, ' Abroad': 202}
28
{' Not in universe under 1 year old': 50668, ' Yes': 41398, ' No': 7693}
29
{' Not in universe': 92067, ' Yes': 2882, ' No': 4810}
31
{' Not in universe': 72393, ' Mother only present': 6256, ' Neither parent present': 807, ' Both parents present': 19358, ' Father only present': 943}
32
{' Columbia': 289, ' Cuba': 587, ' Trinadad&Tobago': 68, ' El-Salvador': 501, ' Guatemala': 231, ' Outlying-U S (Guam USVI etc)': 75, ' Holand-Netherlands': 20, ' United-States': 82796, ' China': 395, ' Thailand': 54, ' Haiti': 156, ' Germany': 647, ' South Korea': 255, ' Iran': 102, ' Hungary': 167, ' Dominican-Republic': 680, ' Poland': 628, ' Philippines': 590, ' Puerto-Rico': 1287, ' Hong Kong': 50, ' Vietnam': 217, ' Honduras': 105, ' Mexico': 5047, ' Portugal': 196, ' England': 380, ' Jamaica': 250, ' India': 306, ' Yugoslavia': 122, ' Greece': 205, ' Japan': 208, ' Taiwan': 83, ' Nicaragua': 176, ' Canada': 659, ' Italy': 1118, ' Panama': 15, ' Scotland': 118, ' France': 84, ' Cambodia': 116, ' Ecuador': 193, ' Laos': 81, ' Peru': 157, ' Ireland': 306}
33
{' Columbia': 288, ' Cuba': 595, ' Trinadad&Tobago': 57, ' El-Salvador': 533, ' Guatemala': 229, ' Outlying-U S (Guam USVI etc)': 62, ' Holand-Netherlands': 19, ' United-States': 83084, ' China': 367, ' Thailand': 70, ' Haiti': 161, ' Germany': 675, ' South Korea': 296, ' Iran': 91, ' Greece': 154, ' Dominican-Republic': 537, ' Poland': 584, ' Philippines': 644, ' Puerto-Rico': 1192, ' Hong Kong': 48, ' Vietnam': 236, ' Honduras': 112, ' Mexico': 4986, ' Portugal': 179, ' England': 426, ' Jamaica': 250, ' India': 312, ' Yugoslavia': 101, ' Hungary': 170, ' Japan': 264, ' Taiwan': 103, ' Nicaragua': 184, ' Canada': 701, ' Italy': 916, ' Panama': 17, ' Scotland': 114, ' France': 88, ' Cambodia': 100, ' Ecuador': 200, ' Laos': 72, ' Peru': 166, ' Ireland': 337}
34
{' Columbia': 210, ' Cuba': 427, ' Trinadad&Tobago': 47, ' El-Salvador': 342, ' Guatemala': 167, ' Outlying-U S (Guam USVI etc)': 45, ' Holand-Netherlands': 10, ' United-States': 90116, ' China': 236, ' Thailand': 64, ' Haiti': 98, ' Germany': 419, ' South Korea': 235, ' Iran': 65, ' Hungary': 29, ' Dominican-Republic': 326, ' Poland': 170, ' Philippines': 453, ' Puerto-Rico': 691, ' Nicaragua': 129, ' Vietnam': 181, ' Honduras': 77, ' Mexico': 2935, ' Portugal': 93, ' England': 224, ' Jamaica': 175, ' India': 217, ' Yugoslavia': 47, ' Greece': 73, ' Japan': 177, ' Taiwan': 81, ' Hong Kong': 51, ' Canada': 345, ' Italy': 225, ' Panama': 10, ' Scotland': 34, ' France': 58, ' Cambodia': 62, ' Ecuador': 133, ' Laos': 53, ' Peru': 120, ' Ireland': 70}
35
{' Native- Born in Puerto Rico or U S Outlying': 737, ' Native- Born abroad of American Parent(s)': 950, ' Foreign born- U S citizen by naturalization': 3011, ' Native- Born in the United States': 88354, ' Foreign born- Not a citizen of U S ': 6705}
36
{' 2': 8233, ' 0': 90187, ' 1': 1339}
37
{' Not in universe': 98734, ' Yes': 198, ' No': 827}
38
{' 2': 75296, ' 0': 23437, ' 1': 1026}
40

{' 94': 49815, ' 95': 49945}

Part 2:

Chose mean of data to replace missing Continuous data because replacing by ean will not change overall mean and will not impact much on standard deviation too . This will keep the gaussian probabilty of variables almost same before and after replacement.

Chose mode to replace missing Categorical attributes because missing attributes are most likely to have the value of mode of attribute as probability of mode of value is maximum for that particular attribute.

Part 3:
Ran the 10-fold cross validation 30 times and then noted the average cross validation accuracy and its standard deviation.

With considering continuous data and calculation gaussion probabiility for that.

epoch	Accuracy_MEAN	Accuracy_STANDARD_DEVIATION
99761	84.9471771241	0.449623570527
99761	84.9722414882	0.391635521185
99761	84.9521977935	0.405714276234
99761	84.9221261226	0.441525812494
99761	85.0193465219	0.302562463194
99761	84.9221170802	0.303981227388
99761	84.9361435271	0.430834369571
99761	84.9521860383	0.237340705311
99761	84.9361521676	0.336794772444

Without considering continuous data and calculation gaussion probabiility for that.

epoch	Accuracy_MEAN	Accuracy_STANDARD_DEVIATION
1	76.0690479241	0.469581990093
2	76.0469954994	0.56518725195
3	76.1071434628	0.281446637975
4	76.0740598525	0.227426862483
5	76.1061459802	0.441153389508
6	76.0680385858	0.491128320762
7	76.0690405897	0.366994712013
8	76.049007344	0.532397997059
9	76.051010548	0.581881738651
10	76.0269330165	0.444594399033
11	76.022931833	0.36118565361
12	76.0670439163	0.30316824688
13	76.027945369	0.426863046375
14	76.0189280373	0.458212323859
15	75.9818401288	0.518007927136
16	76.1021129472	0.423814991802

17	76.0299549027	0.337478562185
18	76.0510014051	0.384249540508
19	76.0640244415	0.46740530579
20	76.0900851831	0.424469708601
21	76.0620379158	0.500281125893
22	76.0740588478	0.244475171345
23	76.037968422	0.546866175875
24	76.0690462161	0.232012311748
25	76.0339622149	0.397584066597
26	76.0259502027	0.610685765364
27	76.0650254406	0.387415768457
28	76.0369610931	0.308843576374
29	76.0750641672	0.520475484824
30	76.078075102	0.401274747554