**Code for Synthetic1 :**

clc;

clear all;

%Load the data

load('D:\python3\_files\python3\synthetic1.mat');

%Size of data points

row = ((size(feature\_train,1)));

col = ((size(feature\_train,2)));

%Augmented data

Augdata = ones(row,col+1);

Augdata(:,2) = feature\_train(:,1);

Augdata(:,3) = feature\_train(:,2);

%Reverse the data points

for i = 1:row

if(label\_train(i) == 2)

Augdata(i,:) = Augdata(i,:)\*(-1);

end

end

%Randomize the training data

augdata = Augdata(randperm(row),:);

%Define weight vector

w = 0.1\*ones(row,col+1);

%Define counter for ad-hoc condition 1

count = 0;

%Define cost function

J = zeros(100,1);

%Classifier Loop

for j = 0:1000

for i = 1:row-1

if (w(i,1)\*augdata(i,1)+w(i,2)\*augdata(i,2)+w(i,3)\*augdata(i,3) <= 0)

w(i+1,:) = w(i,:)+augdata(i,:);

else

w(i+1,:) = w(i,:);

count = count+1;

end

end

if (count == row)

break

end

w(1,:) = w(i+1,:);

%Second adhoc condition

if (j==1000)

for k = 1:row

for i = 1:row

if (w(k,1)\*augdata(i,1)+w(k,2)\*augdata(i,2)+w(k,3)\*augdata(i,3) <= 0)

J(k) = J(k) - w(k,1)\*augdata(i,1)-w(k,2)\*augdata(i,2)-w(k,3)\*augdata(i,3);

end

end

if (J(k)==0)

J(k) = 11111;

end

end

[~,index] = min(J);

end

end

%Final Weight Vector

weight = w(index,:);

%Compute train error rate with final W

count2 = 0;

for i = 1:row

if (weight(1,1)\*augdata(i,1)+weight(1,2)\*augdata(i,2)+weight(1,3)\*augdata(i,3) <= 0)

count2=count2+1;

end

end

train\_error = count2/row;

%Plot Decision Boundary

plotDecBoundaries1(feature\_train,label\_train,weight);

%Applying Classifier to test data

row1 = ((size(feature\_test,1)));

col1 = ((size(feature\_test,2)));

augdata1 = ones(row1,col1+1);

augdata1(:,2) = feature\_test(:,1);

augdata1(:,3) = feature\_test(:,2);

%Compute test error rate with final W

count3 = 0;

for i = 1:row1

if (weight(1,1)\*augdata1(i,1)+weight(1,2)\*augdata1(i,2)+weight(1,3)\*augdata1(i,3) <= 0 && (label\_test(i)==1))

count3 = count3+1;

elseif (weight(1,1)\*augdata1(i,1)+weight(1,2)\*augdata1(i,2)+weight(1,3)\*augdata1(i,3) > 0 && (label\_test(i)==2))

count3 = count3+1;

end

end

test\_error = count3/row1;

**Code for Synthetic2:**

clc;

clear all;

%Load the data

load('D:\python3\_files\python3\synthetic2.mat');

%Size of data points

row = ((size(feature\_train,1)));

col = ((size(feature\_train,2)));

%Augmented data

Augdata = ones(row,col+1);

Augdata(:,2) = feature\_train(:,1);

Augdata(:,3) = feature\_train(:,2);

%Reverse the data points

for i = 1:row

if(label\_train(i) == 2)

Augdata(i,:) = Augdata(i,:)\*(-1);

end

end

%Randomize the training data

augdata = Augdata(randperm(row),:);

%Define weight vector

w = 0.1\*ones(row,col+1);

%Define counter for ad-hoc condition 1

count = 0;

%Define cost function

J = zeros(100,1);

%Classifier Loop

for j = 0:1000

for i = 1:row-1

if (w(i,1)\*augdata(i,1)+w(i,2)\*augdata(i,2)+w(i,3)\*augdata(i,3) <= 0)

w(i+1,:) = w(i,:)+augdata(i,:);

else

w(i+1,:) = w(i,:);

count = count+1;

end

end

if (count == row)

break

end

w(1,:) = w(i+1,:);

%Second adhoc condition

if (j==1000)

for k = 1:row

for i = 1:row

if (w(k,1)\*augdata(i,1)+w(k,2)\*augdata(i,2)+w(k,3)\*augdata(i,3) <= 0)

J(k) = J(k) - w(k,1)\*augdata(i,1)-w(k,2)\*augdata(i,2)-w(k,3)\*augdata(i,3);

end

end

if (J(k)==0)

J(k) = 11111;

end

end

[~,index] = min(J);

end

end

%Final Weight Vector

weight = w(index,:);

%Compute train error rate with final W

count2 = 0;

for i = 1:row

if (weight(1,1)\*augdata(i,1)+weight(1,2)\*augdata(i,2)+weight(1,3)\*augdata(i,3) <= 0)

count2=count2+1;

end

end

train\_error = count2/row;

%Plot Decision Boundary

plotDecBoundaries1(feature\_train,label\_train,weight);

%Applying Classifier to test data

row1 = ((size(feature\_test,1)));

col1 = ((size(feature\_test,2)));

augdata1 = ones(row1,col1+1);

augdata1(:,2) = feature\_test(:,1);

augdata1(:,3) = feature\_test(:,2);

%Compute test error rate with final W

count3 = 0;

for i = 1:row1

if (weight(1,1)\*augdata1(i,1)+weight(1,2)\*augdata1(i,2)+weight(1,3)\*augdata1(i,3) <= 0 && (label\_test(i)==1))

count3 = count3+1;

elseif (weight(1,1)\*augdata1(i,1)+weight(1,2)\*augdata1(i,2)+weight(1,3)\*augdata1(i,3) > 0 && (label\_test(i)==2))

count3 = count3+1;

end

end

test\_error = count3/row1;

%Plot Decision Boundary

%plotDecBoundaries1(feature\_test,label\_test,weight);

**Code for Synthetic3:**

clc;

clear all;

%Load the data

load('D:\python3\_files\python3\synthetic3.mat');

%Size of data points

row = ((size(feature\_train,1)));

col = ((size(feature\_train,2)));

%Augmented data

Augdata = ones(row,col+1);

Augdata(:,2) = feature\_train(:,1);

Augdata(:,3) = feature\_train(:,2);

%Reverse the data points

for i = 1:row

if(label\_train(i) == 2)

Augdata(i,:) = Augdata(i,:)\*(-1);

end

end

%Randomize the training data

augdata = Augdata(randperm(row),:);

%Define weight vector

w = 0.1\*ones(row,col+1);

%Define cost function

J = zeros(row,1);

%Classifier Loop

for j = 0:1000

%Define counter for ad-hoc condition 1

count = 0;

flag = 0;

for i = 1:row

if (w(i,1)\*augdata(i,1)+w(i,2)\*augdata(i,2)+w(i,3)\*augdata(i,3) <= 0)

w(i+1,:) = w(i,:)+augdata(i,:);

else

w(i+1,:) = w(i,:);

count = count+1;

end

if (count == row)

weight = w(i+1,:);

flag = 1;

break

end

end

if (flag == 1)

break

end

w(1,:) = w(i+1,:);

%Second adhoc condition

if (j==1000)

for k = 1:row

for i = 1:row

if (w(k,1)\*augdata(i,1)+w(k,2)\*augdata(i,2)+w(k,3)\*augdata(i,3) <= 0)

J(k) = J(k) - w(k,1)\*augdata(i,1)-w(k,2)\*augdata(i,2)-w(k,3)\*augdata(i,3);

end

end

if (J(k)==0)

J(k) = 11111;

end

end

[~,index] = min(J);

%Final Weight Vector

weight = w(index,:);

end

end

%Compute train error rate with final W

count2 = 0;

for i = 1:row

if (weight(1,1)\*augdata(i,1)+weight(1,2)\*augdata(i,2)+weight(1,3)\*augdata(i,3) <= 0)

count2=count2+1;

end

end

train\_error = count2/row;

%Plot Decision Boundary

plotDecBoundaries1(feature\_train,label\_train,weight);

%Applying Classifier to test data

row1 = ((size(feature\_test,1)));

col1 = ((size(feature\_test,2)));

augdata1 = ones(row1,col1+1);

augdata1(:,2) = feature\_test(:,1);

augdata1(:,3) = feature\_test(:,2);

%Compute test error rate with final W

count3 = 0;

for i = 1:row1

if (weight(1,1)\*augdata1(i,1)+weight(1,2)\*augdata1(i,2)+weight(1,3)\*augdata1(i,3) <= 0 && (label\_test(i)==1))

count3 = count3+1;

elseif (weight(1,1)\*augdata1(i,1)+weight(1,2)\*augdata1(i,2)+weight(1,3)\*augdata1(i,3) > 0 && (label\_test(i)==2))

count3 = count3+1;

end

end

test\_error = count3/row1;

%Plot Decision Boundary

%plotDecBoundaries1(feature\_test,label\_test,weight);