**PREREQUISITES:**

The train.txt file has to be pasted on MATLAB root folder. The MATLAB root folder can be known by typing pwd on command window of MATLAB

**KEY EQUATIONS:**

x=[x1  y=[y1  theta=[0

x2  y2  0]

… ….

x97] y97]

x=[1 x1

1 x2

……

1 x97]

m=90(7 cases are removed for testing)

d=2

h=theta1+theta2\*x

h=x\*theta

MSE=cost=sum((h-y).^2)/(2\*m)

Learning rate=alpha=0.001

theta1=theta1-alpha\*( ∂ cost)/ (∂ theta1)

theta1=theta1-(alpha/m)\*sum(x\*theta-y)\*x(:,1)

theta2=theta2-alpha\*( ∂ cost)/ (∂ theta2)

theta2=theta2-(alpha/m)\*sum(x\*theta-y)\*x(:,2)

**Code with Explanation:**

**%initially data is loaded and separated into x and y**

data=load('train.txt');

d=size(data);

d=d(2);

x=data(:,1);

y=data(:,d);

**%data is then randomly separated into 90 and 7 values for training and testing**

q=ceil(rand(7,1)\*50);

testx=zeros(7,1);

testy=zeros(7,1);

for v=1:length(q);

testx(v,1)=x(q(v));

testy(v,1)=y(q(v));

x(q(v))=[];

y(q(v))=[];

end

**%now ones column is added to make matrix multiplication easier**

m=length(x);

x1=ones(m,1);

x=[x1,x];

**%initial theta values are set**

theta=zeros(d,1);

**%h=theta1+theta2\*x is found using**

h=x\*theta;

**%cost function is found using**

c=sum((h-y).^2);

cost=c/(2\*m);

**%learning rate is set**

alpha=.001;

**%Iterations is done to find the correct weights**

for i=1:20000

z=sum(x\*theta-y);

z1=sum((x\*theta-y).\*x(:,2));

theta(1)=theta(1)-((alpha/m)\*z);

theta(2)=theta(2)-((alpha/m)\*z1);

z=0;

z1=0;

end;

**%the hyperplane is found and plotted**

p=x\*theta;

plot(nx,y,'rx')

hold on;

plot(nx,p)

hold on;

**%now we predict for the testing data and MSE is found**

m=length(testx);

x1=ones(m,1);

testnx=[x1,testx(:,1)];

prediction=testnx\*theta;

MSE=(sum(testnx\*theta-testy).^2)/(2\*m)

**% we plot the testing data to see how well the algorithm predicts**

plot(testx,prediction,'bo')

plot(testx,testy,'gx')

**OUTPUT:**



MSE=0.0731