Logistic Regression:

A & B]

Code:

```
% Task (a) : Load given data

x = load("q1x.dat") ;    %disp(x)
y = load("q1y.dat") ;    %disp(y)

% Task (b) : Implementation of Gradient Descent for LR optimization

theta = zeros(size(x,2)+1,1) ;  % Theta initialization with zeros
                                % disp(theta);
G = zeros(size(theta)) ;        % gradient initialization with zeros
x = [ones(size(x,1), 1) x] ;    % Adding intercept term

for i = 1:100
    % cost equation given in assignment
    LR_cost = sum( y.*log(sigma(x*theta)) + (1-y).*log(1-sigma(x*theta))) ;
    % gradient of LR problem
    G = x' * ( sigma(x*theta) - y ) ;
end

for i = 1:10000
    % updating values of theta
    theta = theta - (0.0003) * G ;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function sigma_val = sigma(ip)
    sigma_val = 1./(1 + exp(-ip));
end
```

Coefficient theta resulting from above fit: *[-1.5000 ; 143.0071 ; 164.7846]*

C]

Code:

```
% Task (c) : implementation of Newton's method for LR optiimzation

% g_z = 1 + exp(y.*(x*theta)) ;
% h_desh = 1 - g_z ;
% h_double_desh = g_z.*( g_z - 1 ) ;
% Gradient_f = @(x) sum( h_desh * (y'*(x*theta)) * x ) ;
% Hessain_f = @(x) sum( h_double_desh * (y'*(x*theta))*(x*x')) ;

% Gradient equation for LR problem
Gradient_f = @(x) sum(x'*(sigma(x*theta)-y)) ;
% Hessain equation for LR problem
Hessain_f = @(x) sum(sum((x'*x) * (sigma(x*theta)' * (1-sigma(x*theta)))));

conv = 1000;
```

```
n = 0;
while(n <= conv)
    Delta = -( Gradient_f(x)/Hessain_f(x) );
    % taking step size eta = 0.0003 and Updating x
    x = x + (0.0003 * Delta);
    n = n + 1;
end
```

Gradient matrix: (3 x 1) [34.9991 ; 118.20224 ; -36.66159]

Hessian matrix: (3 x 3)

$$\begin{bmatrix} 0.08471 & 0.33237 & -0.02962 \\ 0.33237 & 1.61638 & -0.06157 \\ -0.02962 & -0.06157 & 0.27360 \end{bmatrix}$$

<u>D</u>]

Code:

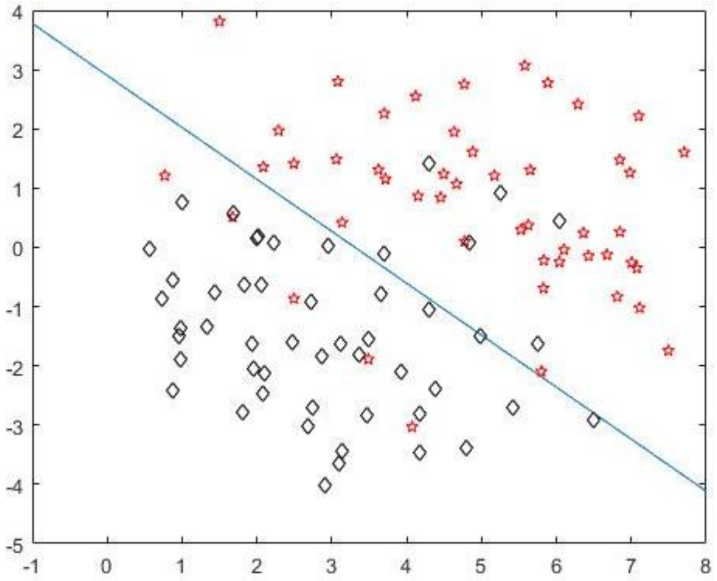
```
% Task (d) : plotting the training data

figure
x = load("q1x.dat") ;
y = load("q1y.dat") ;

% plotting all positive data points with reference to y=1
plot(x(y == 1,1),x(y == 1,2), 'p', 'color', 'red')
hold on
% plotting all negative data points with reference to y=0
plot(x(y == 0,1),x(y == 0,2), 'd', 'color', 'black')

% drawing line
difference = (theta(1)- theta(2)) ;
X1 = linspace(-1,8,500) ;
M = difference*X1 / theta(3)+2.9 ; % slope
% plotting the line between data points
plot(X1,M)
```

Figure:


4
3
2
1
0
-1
-2
-3
-4
-5
-1
0
1
2
3
4
5
6
7
8