**EX2.m – Logistic Regression**

data = load('ex2data1.txt');

X = data(:, [1, 2]); y = data(:, 3);

%% ============ Part 1: Compute Cost and Gradient ============

[m, n] = size(X);

X = [ones(m, 1) X];

initial\_theta = zeros(n + 1, 1);

[cost, grad] = costFunction(initial\_theta, X, y);

fprintf('Cost at initial theta (zeros): %f\n', cost);

fprintf('Gradient at initial theta (zeros): \n');

fprintf(' %f \n', grad);

%% ============= Part 2: Optimizing using fminunc =============

options = optimset('GradObj', 'on', 'MaxIter', 400);

[theta, cost] = ...

fminunc(@(t)(costFunction(t, X, y)), initial\_theta, options);

fprintf('Cost at theta found by fminunc: %f\n', cost);

fprintf('theta: \n');

fprintf(' %f \n', theta);

%% ============== Part 3: Predict and Accuracies ==============

prob = sigmoid([1 45 85] \* theta);

fprintf(['For a student with scores 45 and 85, we predict an admission ' ...

'probability of %f\n\n'], prob);

% Compute accuracy on our training set

p = predict(theta, X);

fprintf('Train Accuracy: %f\n', mean(double(p == y)) \* 100);

fprintf('\nProgram paused. Press enter to continue.\n');

pause;

function g = **sigmoid**(z)

%SIGMOID Compute sigmoid functoon

% J = SIGMOID(z) computes the sigmoid of z.

g = zeros(size(z));

[r,c] = size(z);

for i=1:r

for j=1:c

g(i,j) = 1/(1+exp(-z(i,j)));

end

end

end

function [J, grad] = **costFunction**(theta, X, y)

%COSTFUNCTION Compute cost and gradient for logistic regression

% J = COSTFUNCTION(theta, X, y) computes the cost of using theta as the

% parameter for logistic regression and the gradient of the cost

% w.r.t. to the parameters.

m = length(y); % number of training examples

J = 0;

grad = zeros(size(theta));

[r, c] = size(X);

h\_theta = sigmoid(X\*theta);

J = -(1./m)\*sum(y.\*log(h\_theta)+(1-y).\*log(1-h\_theta));

for i=1:c

grad(i) = (1./m)\*sum(X(:,i).\*(h\_theta-y));

end

end

function p = **predict**(theta, X)

%PREDICT Predict whether the label is 0 or 1 using learned logistic

%regression parameters theta

m = size(X, 1); % Number of training examples

p = zeros(m, 1);

for i = 1:m

val = sigmoid(X(i,:)\*theta);

if val>= 0.5

p(i) = 1;

else

p(i) = 0;

end

end

end