Assignment 1

Machine Learning (SE-807)

Submitted By:-

Shahzeb Awan

Reg no:-

362585

Submitted To:-

Dr. Qasim Umar Khan

Date: - 21, March, 2022

Q: Take a 5-featured Multi-Linear Regression Problem having at least 1000 training examples, plot the cost function vs. number of iterations, after that show the surface 3D plot of Cost function having variable θ_0 and θ_1, at the end also show the contour plot of the cost function.

I have taken a problem of climate **Temperature** prediction using other features in relation to it like, **Humidity**, **Wind direction**, **Atmospheric pressure**, **Wind-speed** and **Visibility**.

First of all I downloaded the dataset from Kaggle having link as follows: -

https://www.kaggle.com/datasets/zakriarehman/weather-data-for-linear-regression

Then I selected my feature for prediction as Temperature and selected other features as inputs.

After arranging in a useable format in excel as follows:-

L1 f_x

	A	B	C	D	E	F	G	H	I	J	K
1	Humidity	,	Wind_Bearing_degrees	,	Pressure_millibars	,	Wind_Speed_kmh	,	Visibility_km	,	Temperature_c
2	0.92	,	130	,	1021.6	,	11.27	,	8.05	,	-0.555555556
3	0.73	,	330	,	1017	,	20.93	,	16.1	,	21.11111111
4	0.97	,	193	,	1013.99	,	5.9731	,	14.9086	,	16.6
5	0.82	,	300	,	1031.59	,	3.22	,	16.1	,	1.6
6	0.6	,	116	,	1020.88	,	10.8836	,	9.982	,	2.194444444
7	0.32	,	190	,	1015.33	,	21.4613	,	10.3523	,	27.53888889

Then I converted this excel data file to text form for usage in my MATLAB Program as shown on next page: -

Shahzeb_Awan_data_weather - Notepad

File Edit Format View Help

0.92 ,	130 ,	1021.6 ,	11.27 ,	8.05 ,	-0.555555556
0.73 ,	330 ,	1017 ,	20.93 ,	16.1 ,	21.11111111
0.97 ,	193 ,	1013.99 ,	5.9731 ,	14.9086 ,	16.6
0.82 ,	300 ,	1031.59 ,	3.22 ,	16.1 ,	1.6
0.6 ,	116 ,	1020.88 ,	10.8836 ,	9.982 ,	2.194444444
0.32 ,	190 ,	1015.33 ,	21.4613 ,	10.3523 ,	27.53888889
0.84 ,	170 ,	1009.04 ,	7.9695 ,	11.1251 ,	19.97777778
0.86 ,	30 ,	1009.6 ,	14.49 ,	15.134 ,	11.11111111
0.73 ,	351 ,	1018.39 ,	14.007 ,	15.8263 ,	8.405555556
0.81 ,	320 ,	1003.89 ,	6.44 ,	7.8568 ,	1.7
0.88 ,	141 ,	1021.28 ,	14.007 ,	6.0214 ,	-2.222222222
0.6 ,	204 ,	1019.52 ,	1.4168 ,	15.8263 ,	21.9
0.87 ,	1 ,	1015.92 ,	11.0285 ,	14.9086 ,	17.10555556
0.73 ,	297 ,	1013.06 ,	4.0733 ,	9.7566 ,	17.77222222
0.39 ,	35 ,	1025.59 ,	7.6636 ,	9.982 ,	24.95
0.92 ,	310 ,	1024.3 ,	3.22 ,	3.4615 ,	-2.711111111
0.78 ,	180 ,	1018.76 ,	4.83 ,	9.982 ,	18.88888889

Main code:-

```
%% Load Data
data = load('Shahzeb_Awan_data_weather.txt');
X = data(:, 1:5);% input features
y = data(:, 6); % output column is number 6
m = length(y); % length of dataset

%% ================ Part 1: Feature Normalization ================
% Scale features and set them to zero mean
fprintf('Normalizing Features ...\n');

[X mu sigma] = featureNormalize(X);% This function is explained in this
%report on later pages in detail

% Add intercept term to X
X = [ones(m, 1) X];

%% ================ Part 2: Gradient Descent ================

fprintf('Running gradient descent ...\n');

% Choosing some alpha value
alpha = 0.01;
num_iters = 1000;

% Init Theta and Running Gradient Descent

theta =[1;44;9;2;5;1];%nice then zeros(comparitively better)
 %gradient decent function is also explained later on
[theta, J1] = gradientDescentMulti2(X, y, theta, alpha, num_iters);
```

```
% Plot the convergence graph
figure;
plot(1:numel(J1), J1, 'b');
xlabel('Number of iterations');
ylabel('Cost J');

% Display gradient descent's result
fprintf('Theta computed from gradient descent: \n');
fprintf(' %f \n', theta);
fprintf('\n');
%if Humidity is 0.88 , Wind bearing degrees 141, pressure in mb is 1021.28,
wind speed in km/h is 14.007
%and visibility in km is 6.0214 then Temperature should be round about :-2.22
C
Temperature=[1,(0.88-mu(1))/sigma(1),(141-mu(2))/sigma(2),(1021.28-
mu(3))/sigma(3),(14.007-mu(4))/sigma(4),(6.0214-mu(5))/sigma(5)]*theta;

% ============================================================

fprintf(['Predicted Temperature ' ...
        '(using gradient descent):\n %f\n'], Temperature);

%% ================ Part 3: Normal Equations ================
%% Analytical solution through ordinary least squares

fprintf('Solving with normal equations...\n');

%% Load Data
data = csvread('Shahzeb_Awan_data_weather.txt');
X = data(:, 1:5);
y = data(:, 6);
m = length(y);

% Add intercept term to X
X = [ones(m, 1) X];

% Calculate the parameters from the normal equation
theta = normalEqn(X, y);%this is also explained separately in following pages

% Display normal equation's result
fprintf('Theta computed from the normal equations: \n');
fprintf(' %f \n', theta);
fprintf('\n');

%if Humidity is 0.88 , Wind bearing degrees 141, pressure in mb is 1021.28,
wind speed in km/h is 14.007
%and visibility in km is 6.0214 then Temperature should be round about :-2.22
C

Temperature = [1,0.88   ,   141 ,   1021.28 ,   14.007  ,   6.0214]*theta;


% ============================================================
```

```
fprintf(['Predicted Temperature ' ...
         '(using normal equations):\n %f\n'], Temperature);


%% ============= Part 4: Visualizing J(theta_0, theta_1) =============
fprintf('Visualizing J(theta_0, theta_1) ...\n')

% Grid over which we will calculate J
theta0_vals = linspace(-60, 100, 100);
theta1_vals = linspace(-60, 60, 100);

% initialize J_vals to a matrix of 0's
J_vals = zeros(length(theta0_vals), length(theta1_vals));

% Fill out J_vals
for i = 1:length(theta0_vals)
    for j = 1:length(theta1_vals)
      t = [theta0_vals(i); theta1_vals(j);0.002519;-0.003176;-
0.179258;0.361726];
      J_vals(i,j) = computeCostMulti(X, y, t);
    end
end


% Because of the way meshgrids work in the surf command, we need to
% transpose J_vals before calling surf, or else the axes will be flipped
J_vals = J_vals';
% Surface plot
figure;
surf(theta0_vals, theta1_vals, J_vals)
xlabel('\theta_0'); ylabel('\theta_1');


% Contour plot
figure;
% Plot J_vals as 15 contours spaced logarithmically between 0.01 and 100
contour(theta0_vals, theta1_vals, J_vals, logspace(-2, 3, 20))
xlabel('\theta_0'); ylabel('\theta_1');
hold on;
plot(theta(1), theta(2), 'rx', 'MarkerSize', 10, 'LineWidth', 2);

%my minimum cost Function Result at my Selected thetas from gradient decent:-

J=computeCostMulti(X, y, theta)
```

Gradient-decent function:-

```
function [theta, J_history] = gradientDescentMulti2(X, y, theta, alpha, 
num_iters)
% Initialize some useful values
m = length(y); % number of training examples
J_history = zeros(num_iters, 1);
for iter = 1:num_iters
% Perform a single gradient step on the parameter vector theta.
gradient=zeros(6,1);
  for i=1:m,
    for j=1:6,
    gradient(j,1)=gradient(j,1)+(theta'*X(i,:)'-y(i))*X(i,j);
    end
  end
theta=theta-alpha/m*gradient;
    % Save the cost J in every iteration
    J_history(iter) = computeCostMulti(X, y, theta);
end
end
```

Feature normalization:-

```
function [X_norm, mu, sigma] = featureNormalize(X)
%   the mean value of each feature is 0 and the standard deviation
%   is 1.
X_norm = X;
mu = zeros(1, 5);
sigma = zeros(1, 5);

mu=mean(X);
sigma=std(X);
X_norm=(X-mu)./sigma;
end
```

Closed Form Solution using normal Eqn:-

```
function [theta] = normalEqn(X, y)
%Computes the closed-form solution to linear regression
theta = zeros(5, 1);
theta=pinv(X'*X)*X'*y;
end
```

My Cost function:-

```
function J = computeCostMulti(X, y, theta)
m = length(y); % number of training examples
J = 0;
J=1/(2*m)*(X*theta-y)'*(X*theta-y);
end
```

Command Window Result:-

>>Shahzeb_Awan_Multi_linear_regression

Normalizing Features ...

Running gradient descent ...

Theta computed from gradient descent:

12.391067

-6.045475

0.255844

-0.367468

-1.279270

1.557708

Predicted Temperature (using gradient descent):

5.504655

Solving with normal equations...

Theta computed from the normal equations:

35.008717

-29.869259

0.002519

-0.003176

-0.179258

0.361726

Predicted Temperature (using normal equations):

5.502272

Visualizing J(theta_0, theta_1) ...

J =

23.1443

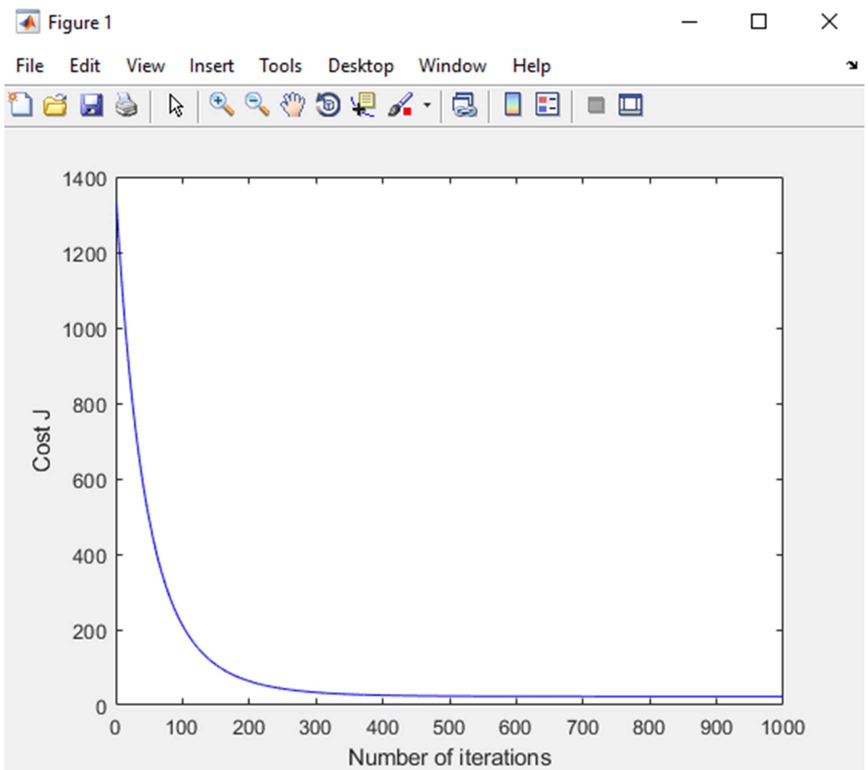

Figure 1 Cost function vs number of iterations

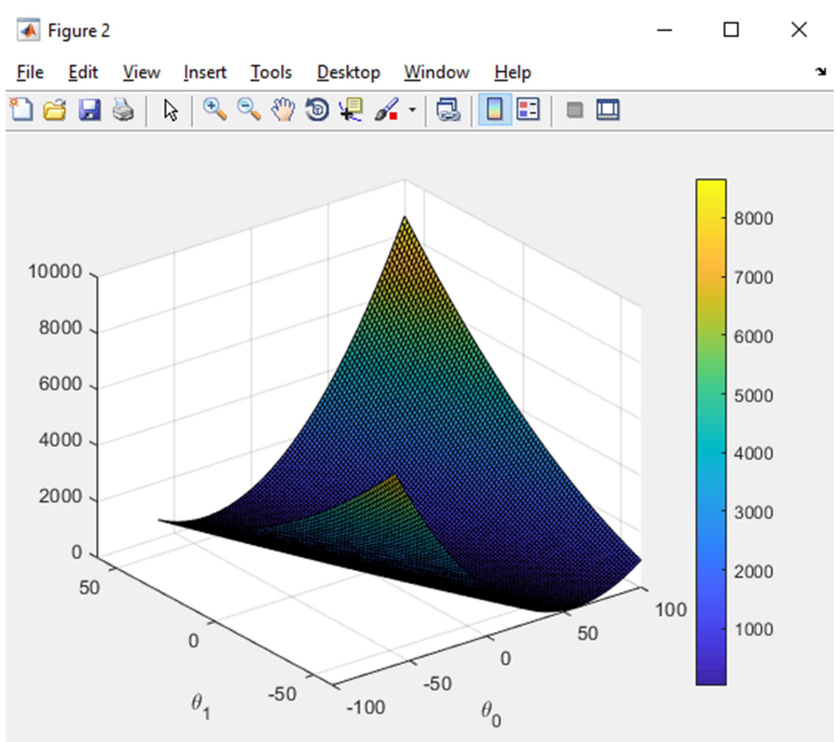

Figure 2 Cost Function vs variable Theta 0 and Theta 1

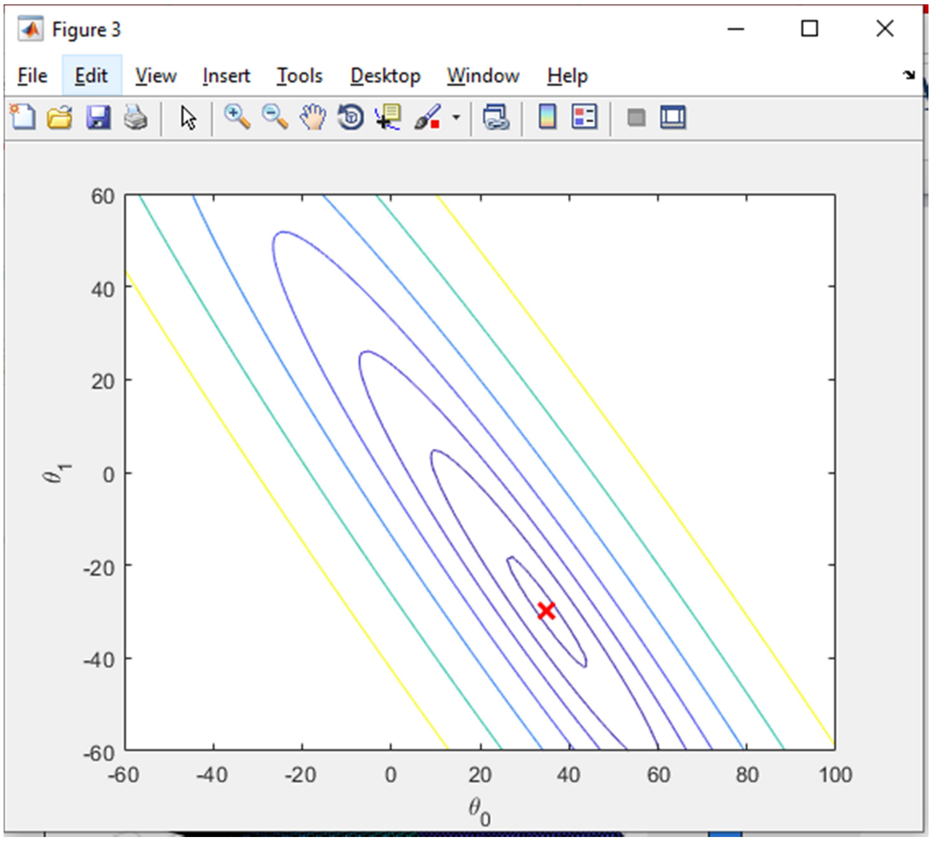

Figure 3 Contour plot of Cost function for theta 0 and theta 1

Following is my Github repository link for this project: -

https://github.com/Shahzeb-Awan/Multi-Linear-Regression

I have made this repository Private but after the marks allotment of this assignment I will make this repository public.

References:-

I have taken help from following two Github repositories and also through my Machine leaning class knowledge: -

https://github.com/kk289/ML-Linear_Regression-MATLAB

https://github.com/antoinevlt/Multivariate-linear-regression