Started machine learning course from coursera .

Topics covered till date (under supervised learning)

1. Linear regression.

Linear regression

In linear regression we are given with the dataset which contains features (upon which our hypothesis depends upon) and target (which we need to predict later).

We train our model on the dataset (training set) with different values of the theta and come up with the model which best fit in the given dataset and predict continuous values such as price, population , profit depending upon what we set our target.

To completely implement our linear regression model we need 3 things to work upon:

1. Hypothesis.
2. Cost function.
3. Gradient descent.
4. Hypothesis : is the model we are trying to predict based on the given dataset.

h(x)= Ɵ0 + Ɵ1x1  + Ɵ2x2 + Ɵ3x3……..

is the way we represent hypothesis for linear regression .

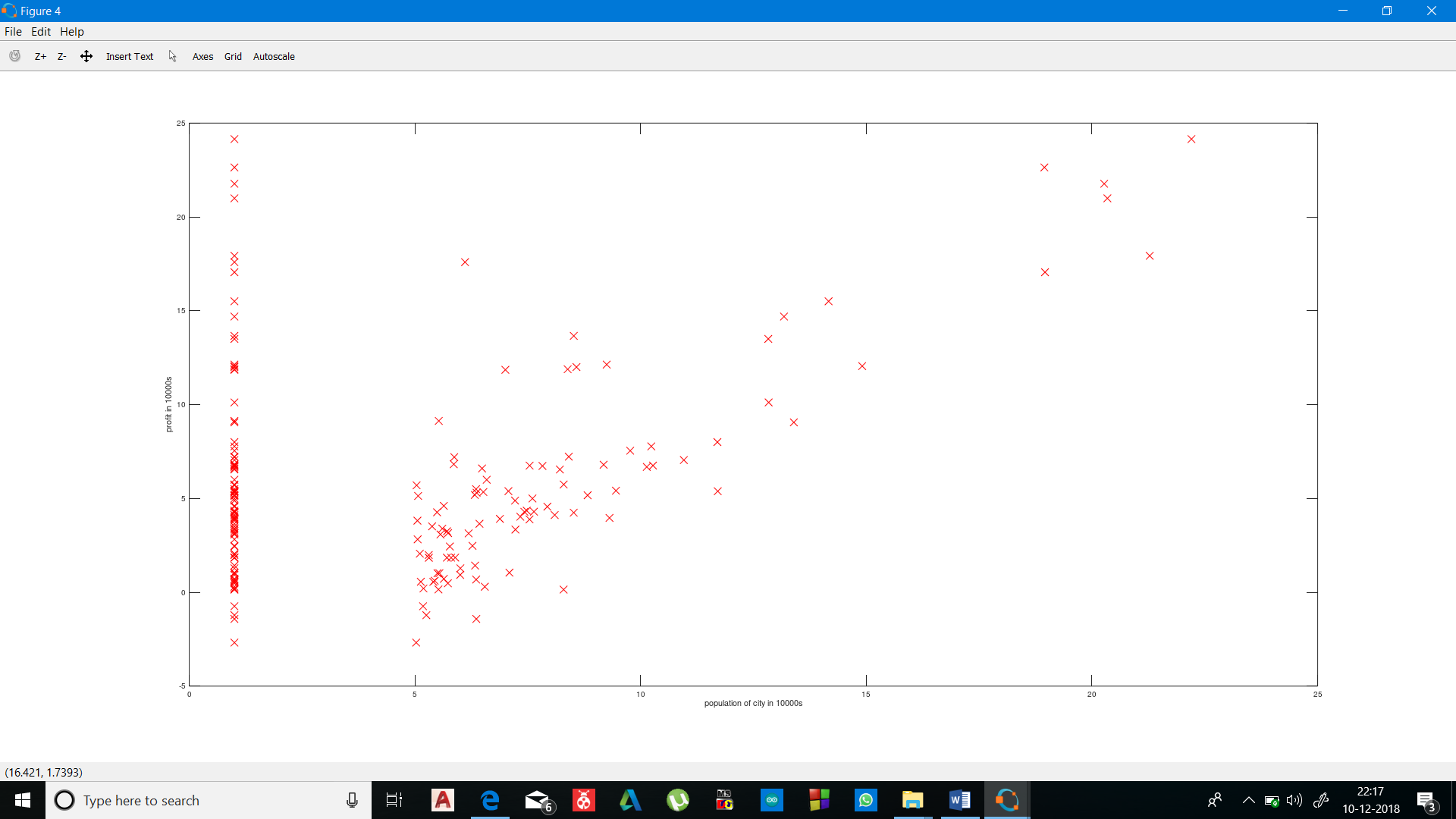
here Ɵj (j>=0) is the parameter we need to find out to best fit our model in the dataset and xi (i>0) are the features from the dataset.

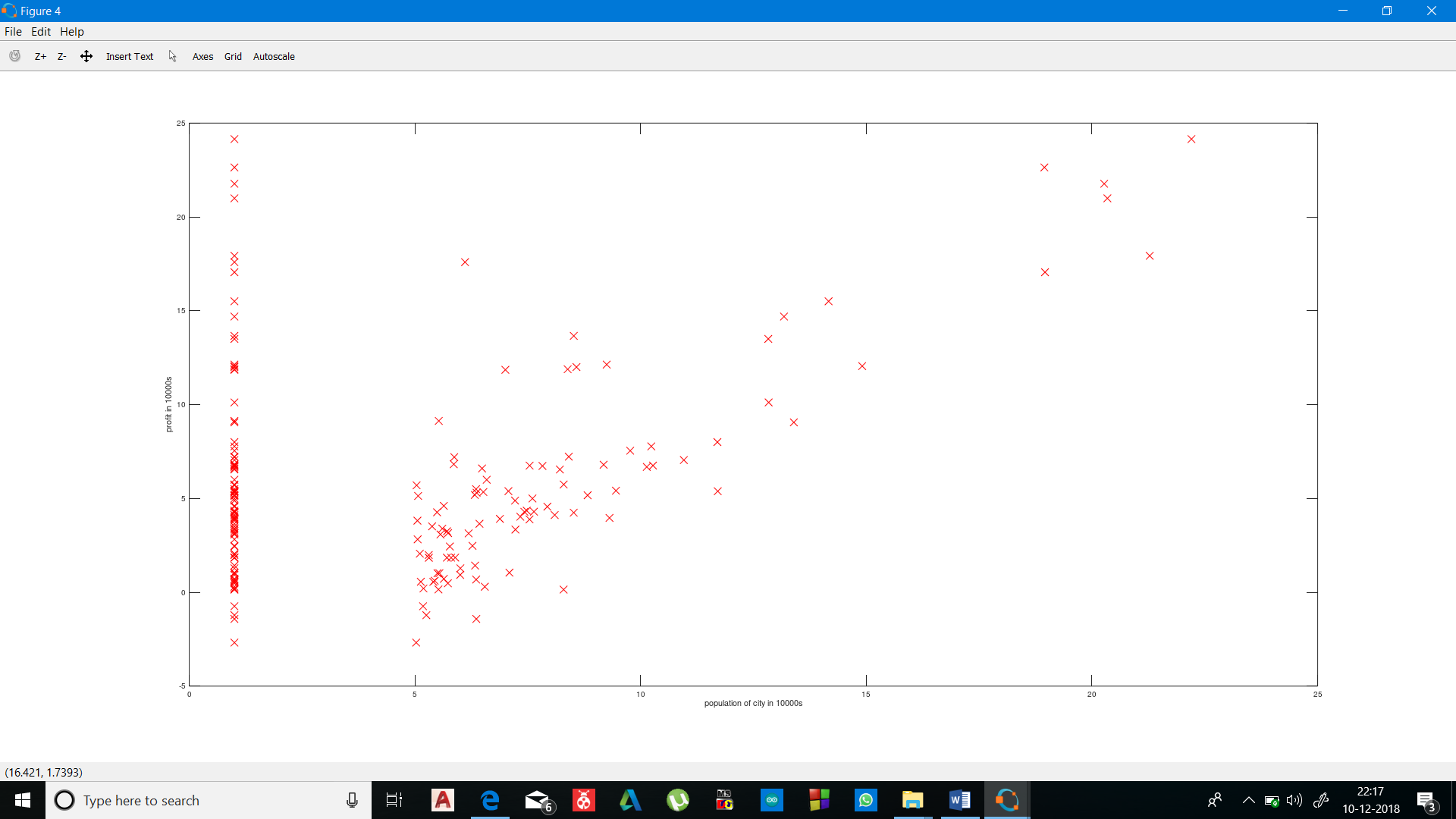
1. Costfunction : we can visualize the costfunction as function which penalize our model for different values of theta …closer the penalty to 0 , more accurate our model is and vice versa . it depends upon the value of theta we chose for our hypothesis.
2. Gradient descent : is the function which minimizes the costfunction i.e gives us the values of theta for which costfunction penalize our model least .(our hypothesis best fit the model ).

I implemented the linear regression in octave .

Problem statement : a food company want to set up layout’s in different cities . it has dataset which contains population of the city and corresponding profit from the layout of that city. Now they want to predict the profit from the layout if they open it in a city which has population x .

1. Plot of the data I was given:





In the plot x axis is the population and y axis is the profit .

(dataset contains only one feature )

1. Hypothesis: h(x)= Ɵ0 x0 + Ɵ1x1  (x0 =1 for all )
2. Costfunction :

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

% parameter for linear regression to fit the data points in X and y

function J = computeCost(X, y, theta)

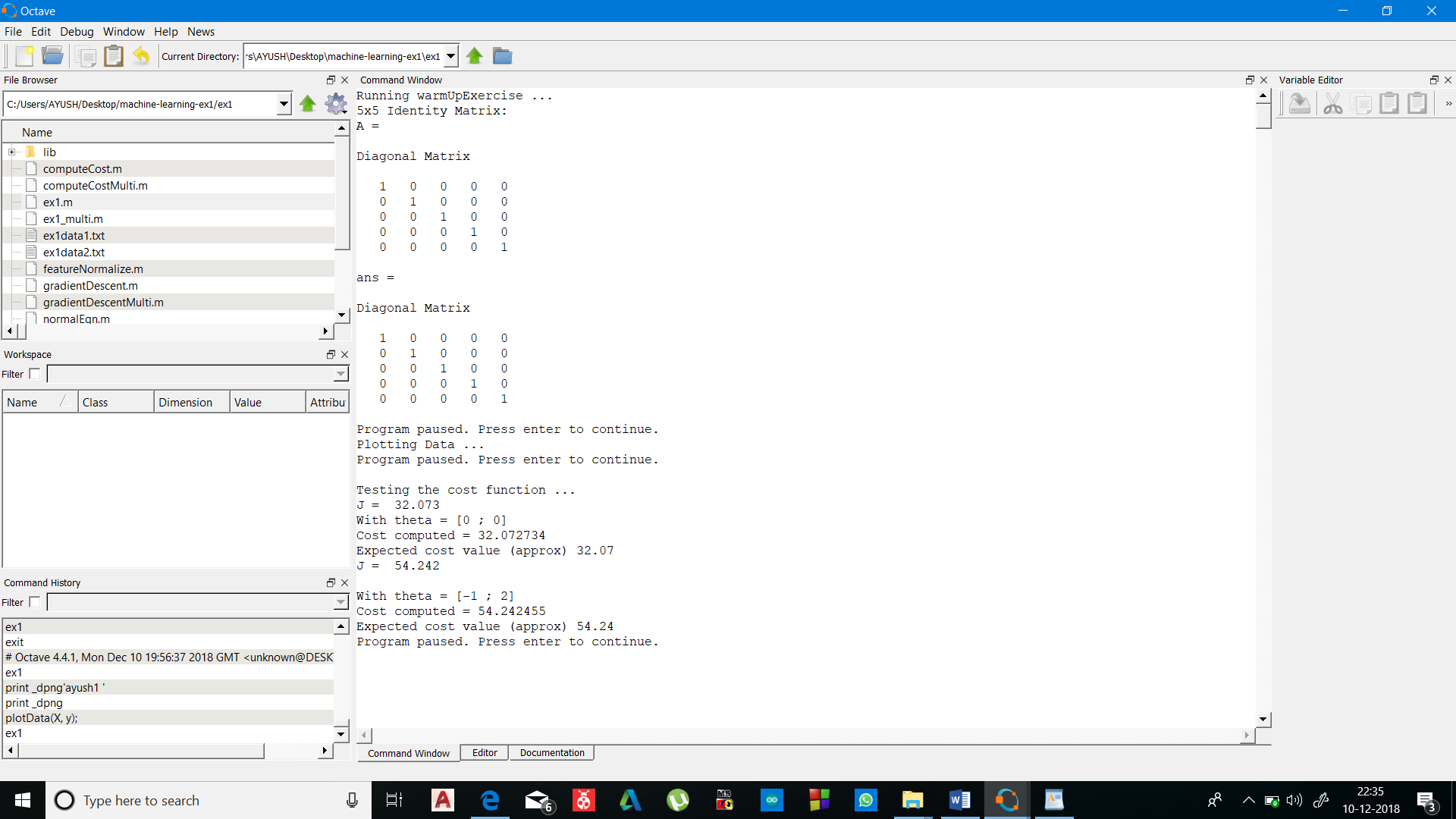
m = length(y); % number of training examples

J = 0;

J=(1/(2\*m))\*sum((((X\*theta)-y).^2))

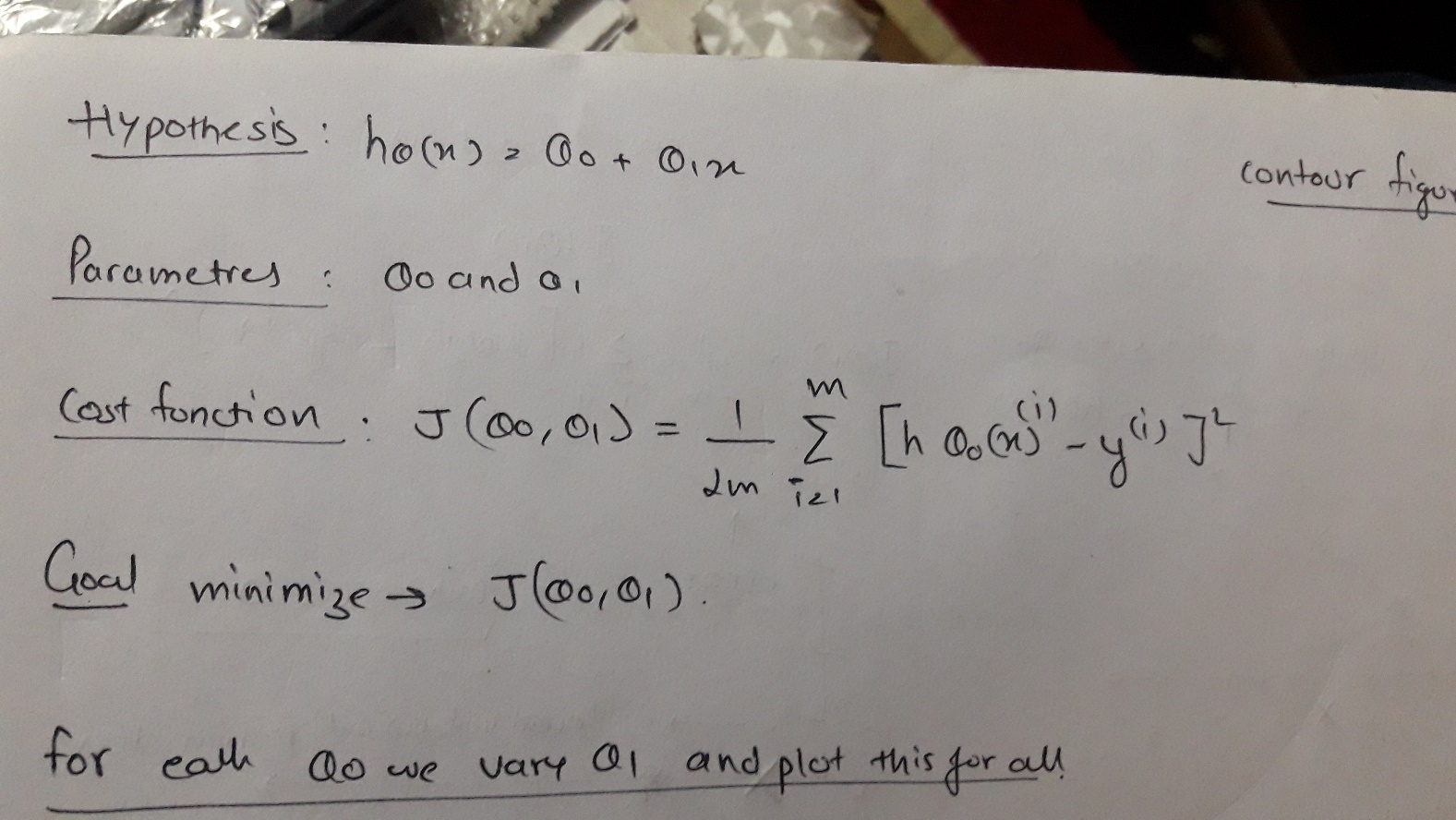
end

output:



it is the definition of cost function which returns the penalty as explained earlier .

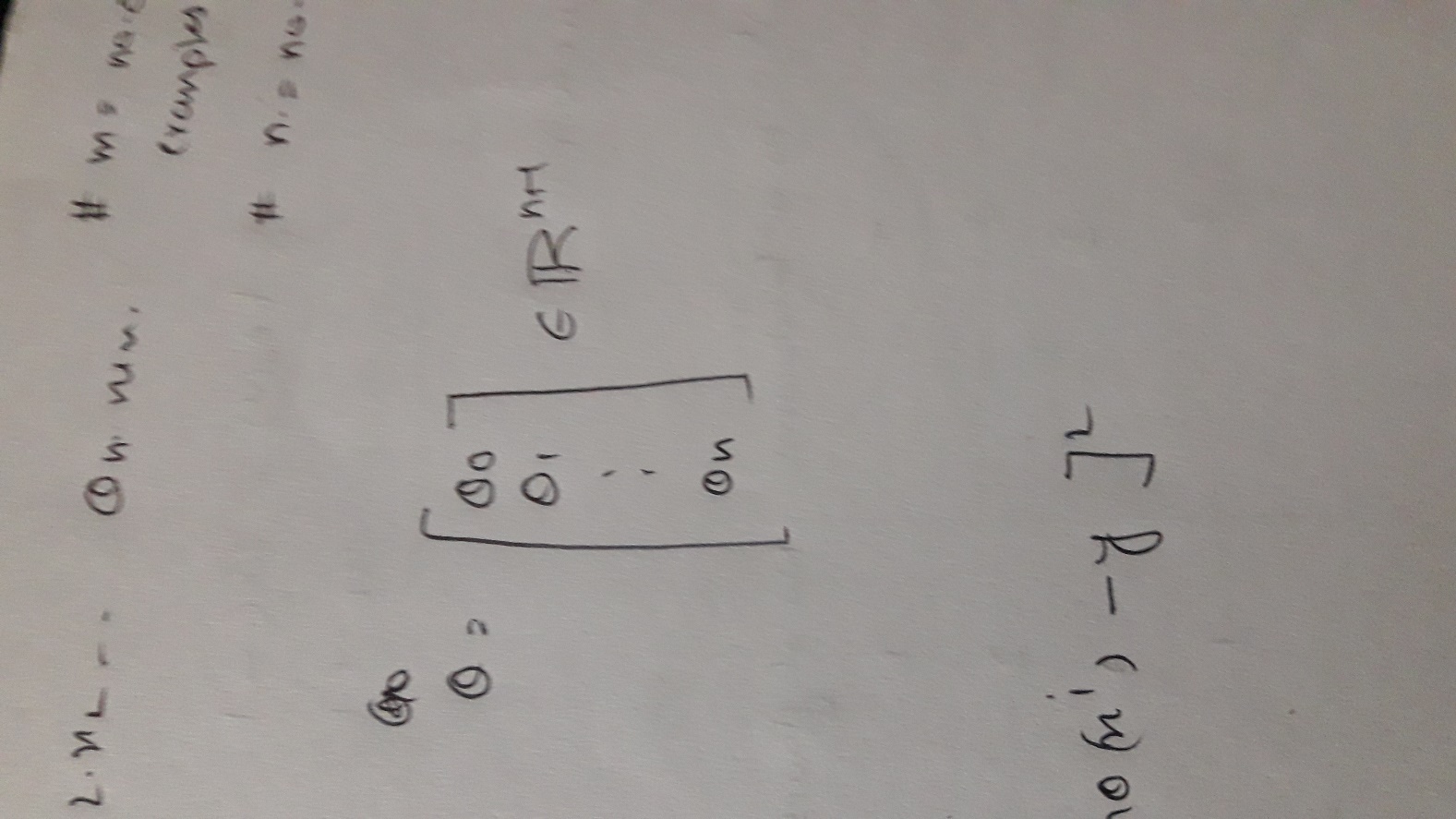
here the equation :



is implemented with the help of vectors

1. X is a feature matrix : 1st column contains value 1 and 2nd column contain population .

(the first column’s value for every row is 1 so as we can do matrix operation )

1. Theta ( Ɵj  ) is a vector representation of Ɵ0 , Ɵ1  in hypothesis Ɵ0 x0 + Ɵ1x1  . 

Gradient descent :

function [theta, J\_history] = gradientDescent(X, y, theta, alpha, num\_iters)

%GRADIENTDESCENT Performs gradient descent to learn theta

% theta = GRADIENTDESCENT(X, y, theta, alpha, num\_iters) updates theta by

% taking num\_iters gradient steps with learning rate alpha

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

J\_history = zeros(num\_iters, 1);

for iter = 1:num\_iters

theta=theta-((X')\*((alpha/m).\*((X\*theta)-y)) )

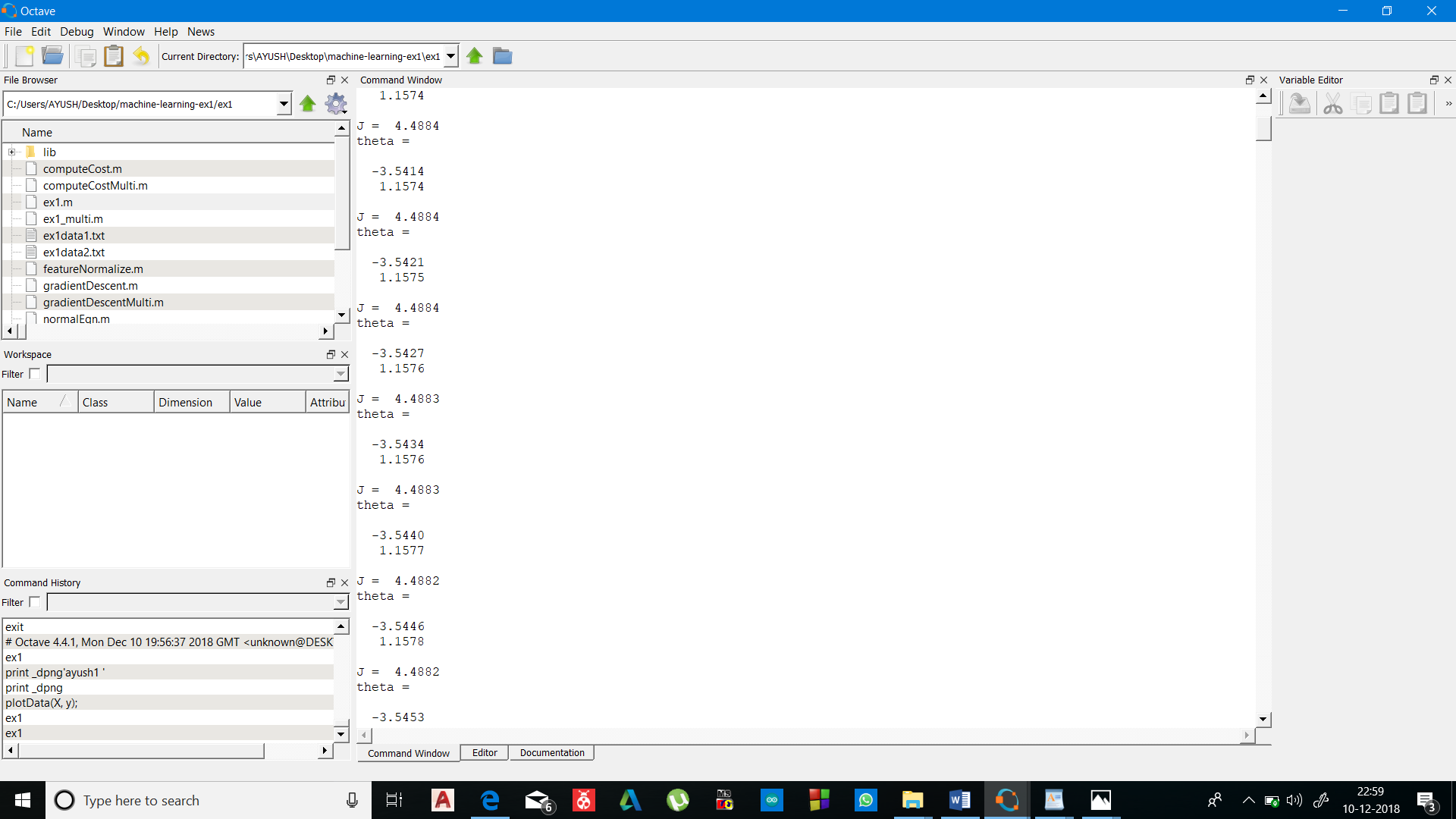
J\_history(iter) = computeCost(X, y, theta);

end

end

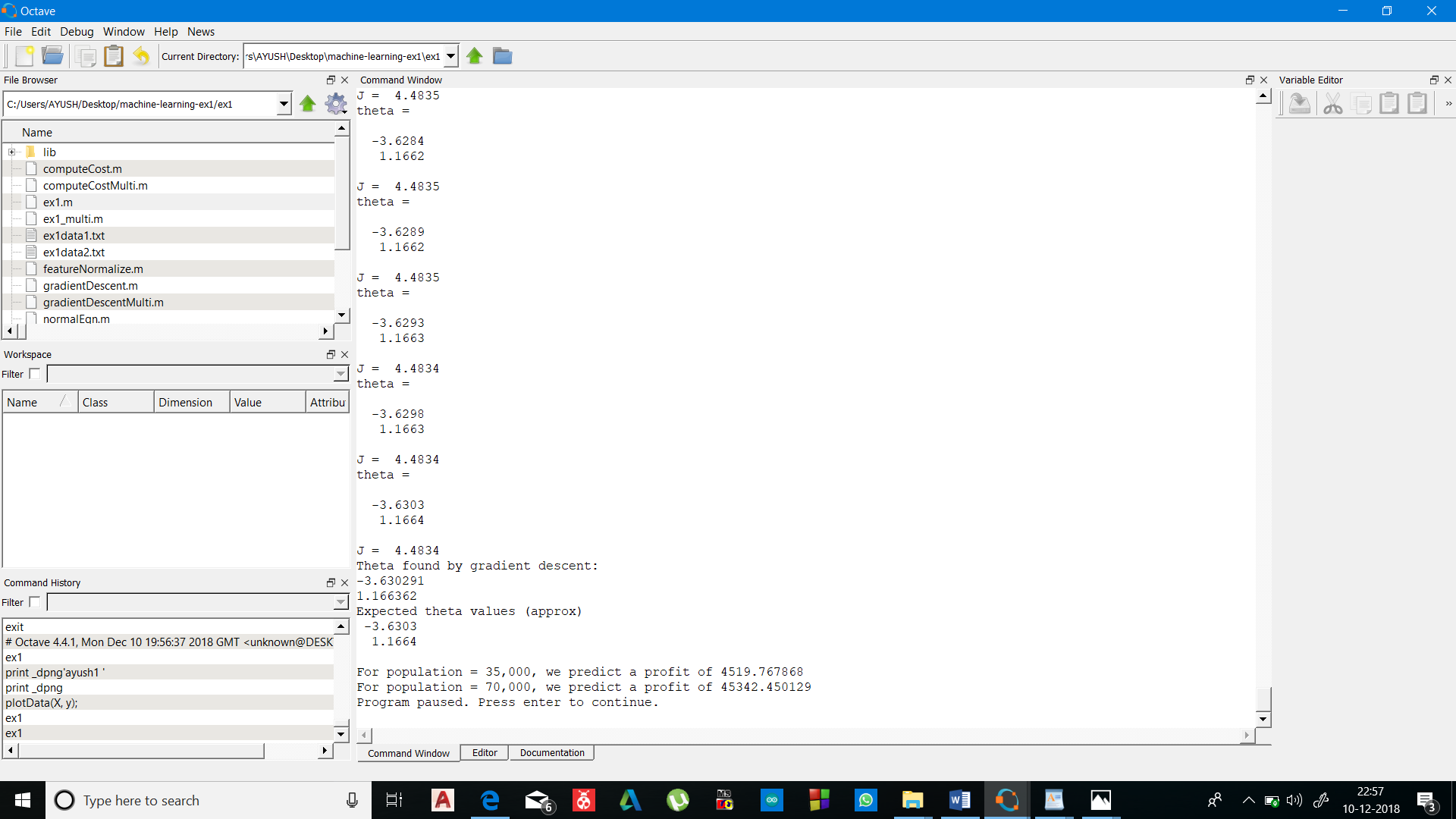
output:

since 1500 interation were made only showing part of the result:



Now whats happening is my gradient function is working to minimize the cost function by working with different set of values of theta which least penalize my model or best fit in my hypothesis.

We can observe with interations theta is decreasing (thus minimizing ) .



Theta we assume to be minimized after 1500 interations

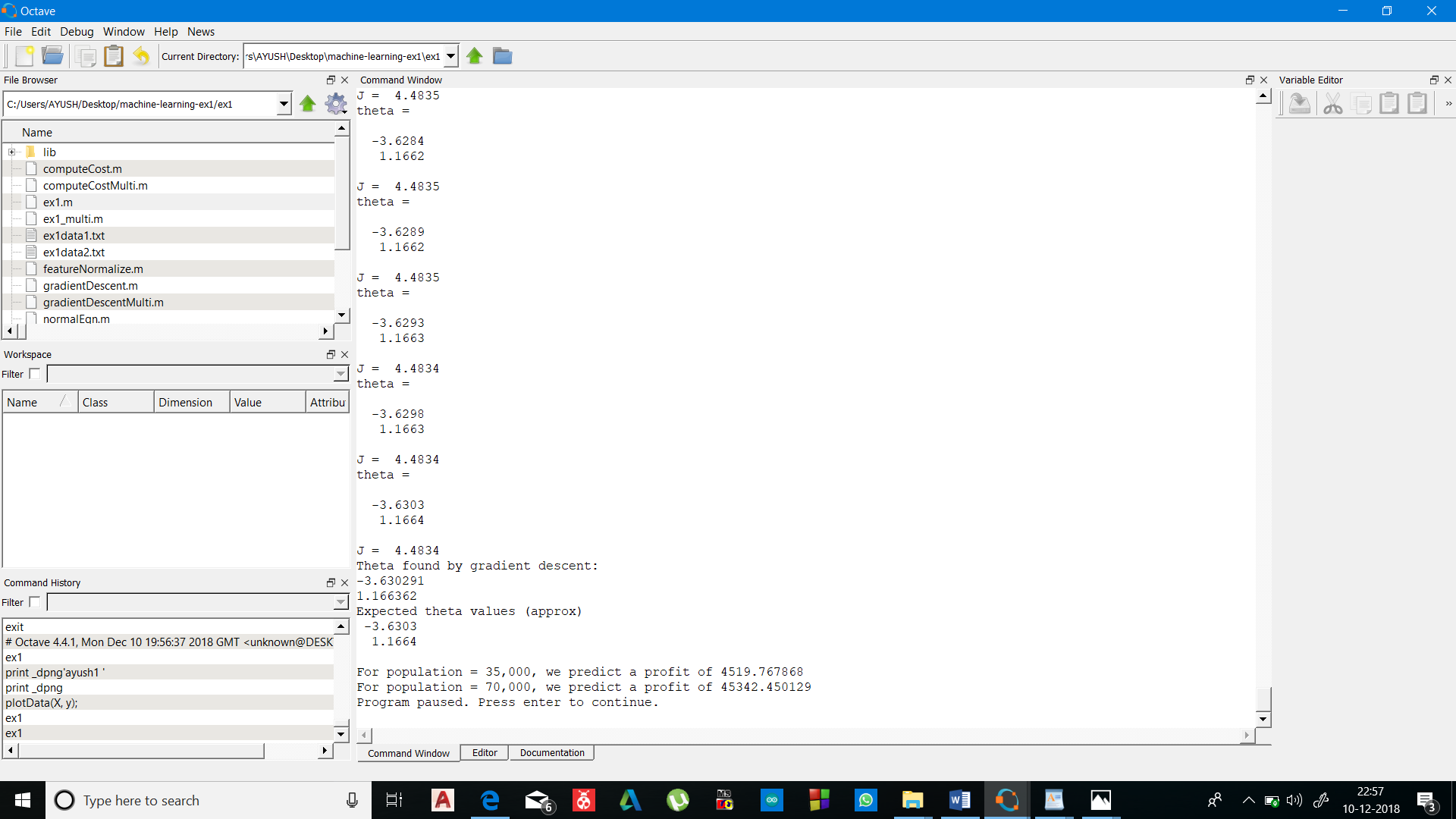
After 1500 interations

J=4.4834

Ɵ0 = -3.6304 Ɵ1 =1.1664

Thus my hypothesis after analysing dataset is :

Profit =-3.6304 + 1.1664\*(population)

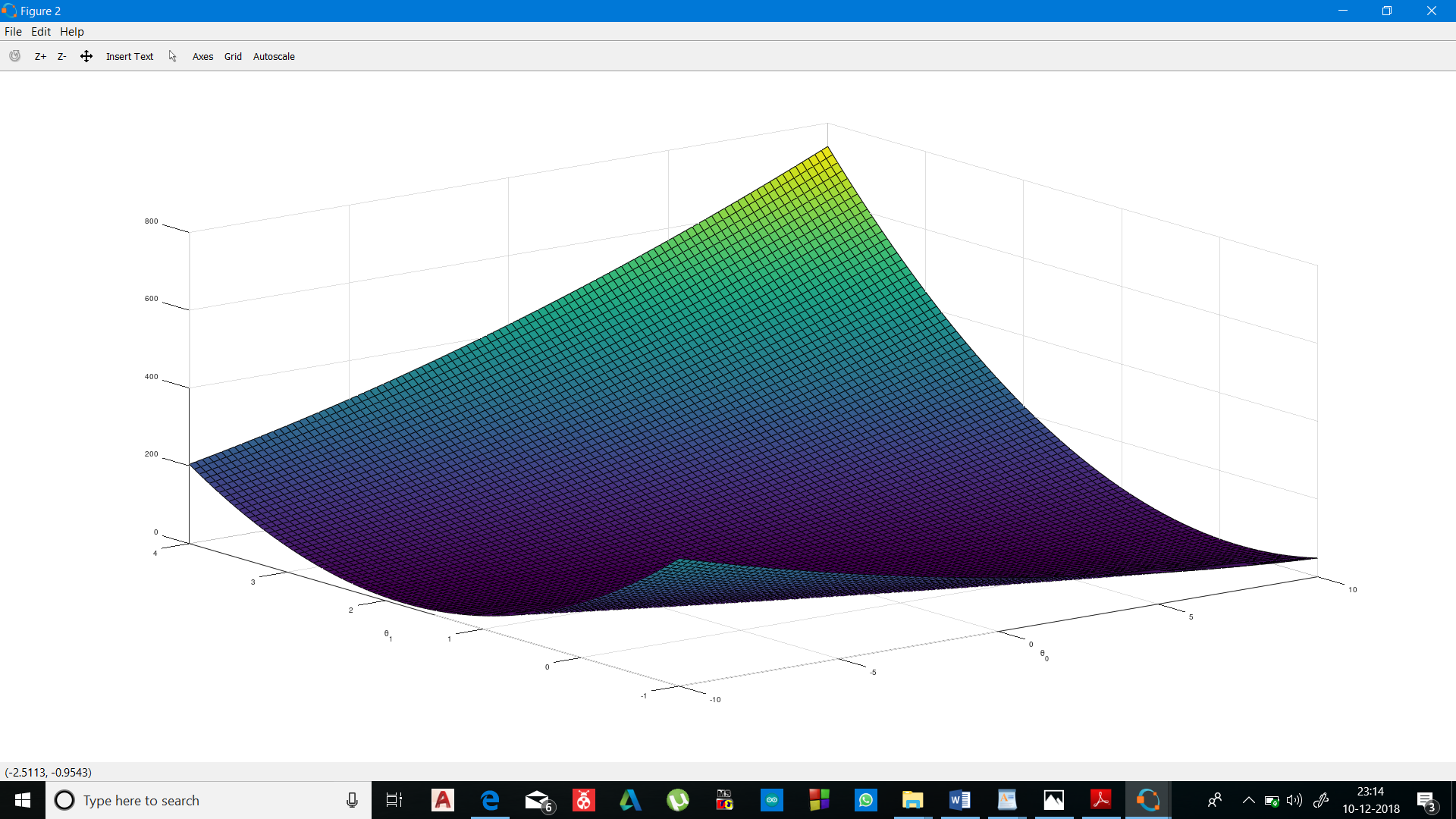


Here at last my model is predicting values of profit for 35000 and 70000 population .

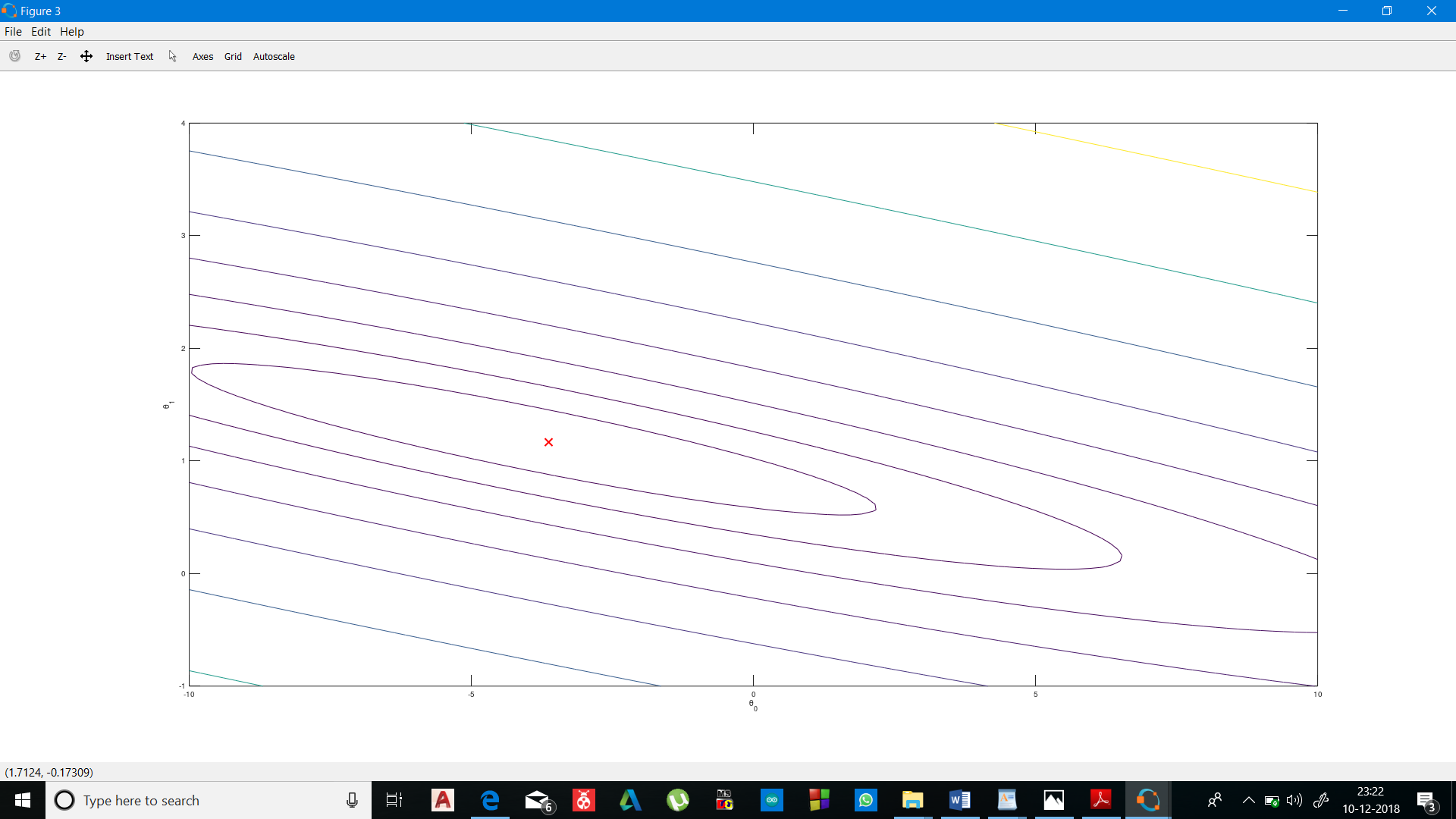
To understand the functioning of gradient descent more clearly here we have ploted the value of Ɵ0 , Ɵ1  and corresponding costfunction on y axis

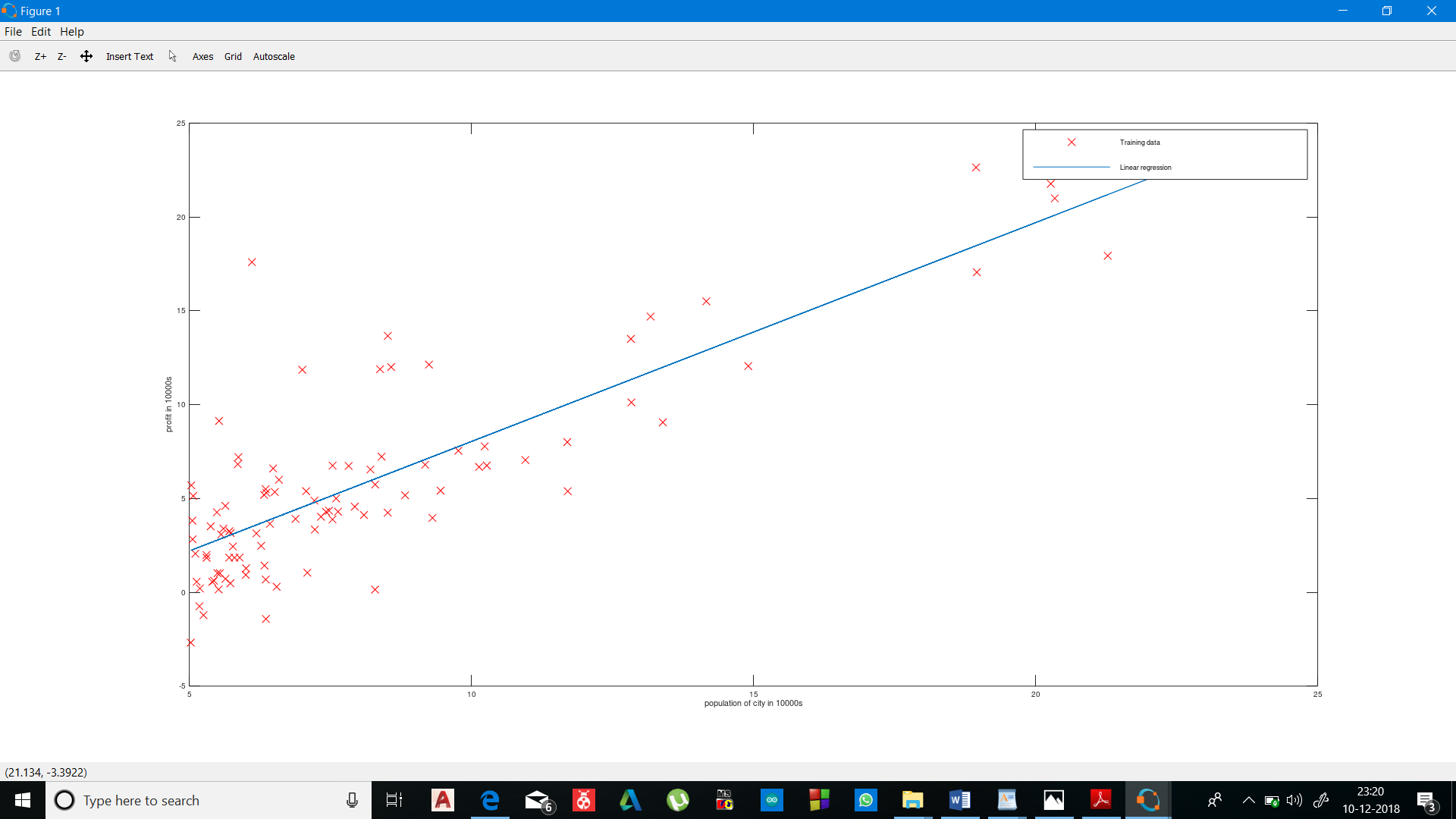
First I have initialized my gradient function at theta=[0,0] but as interations proceeded it automatically moved down towards darker blue colour from (say green portion) and at last returned the value of theta

Which best suits the given dataset.



Contour plot of above 3-D figure : the red cross shows gradient descent has returned values of theta where costfunction has minimized





Red cross : data set

Blue line : predicted model ..my hypothesis:

Profit =-3.6304 + 1.1664\*(population)