ASSIGNMENT 3 - IIT2016039

Problem Statement 1:

Using the data set of two examination results design a predictor using logistic regression for predicting whether a student can get an admission in the institution. Use regularizer to further tune the parameters. Use 70 % data for training and rest 30% data for testing your predictor and calculate the efficiency of the predictor/hypothesis.

Hints: 1. You can pre process the data for convenience
2. You must use Python program for evaluating parameters using batch gradient descent algorithm (GDA). No function should be used for GDA.

Analysis:

The Output for this can be seen in IIT2016039_Assignment3_q1.html
In the given data we have two features exam1 and exam2 and we have 100 samples.
We have separated the data into 70 training samples and 30 testing samples

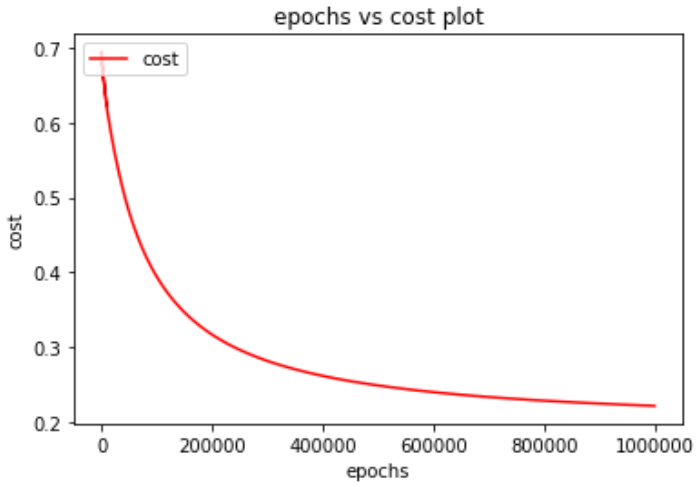
We are taking input features X = [1, exam1, exam2]
Our weights will be W = [W0, W1, W2]'

Logistic regression predictor function H(X) = 1 / (1 + e^(- W.X))

For H(x) <= 0.5 we take it as 0 i.e W.X >= 0
For H(x) > 0.5 we take it as 1 i.e W.X >= 0

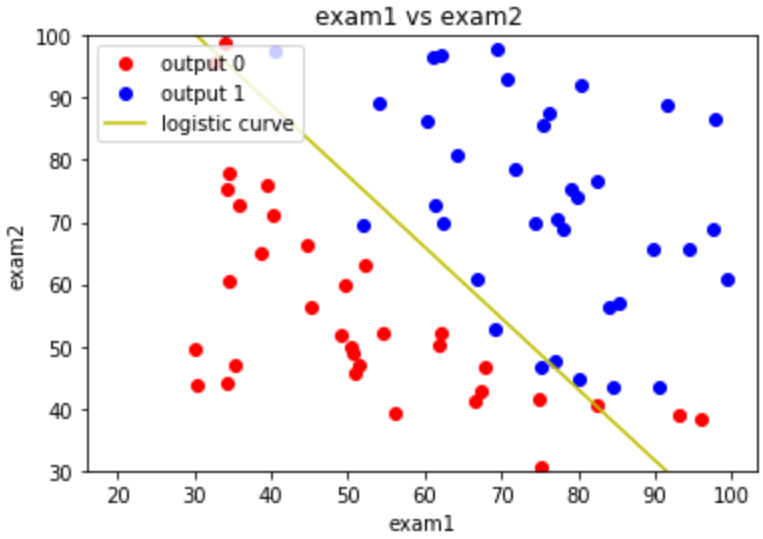
Cost function for logistic regression is - (1 / len(Y)) * (sum(Y * log(H) + (1 - Y)*log (1 - H)))

We took learning rate = 10^-3
Epochs = 1,000,000
After applying gradient descent we got Final cost of 0.2211087572819281 graph is as follows :

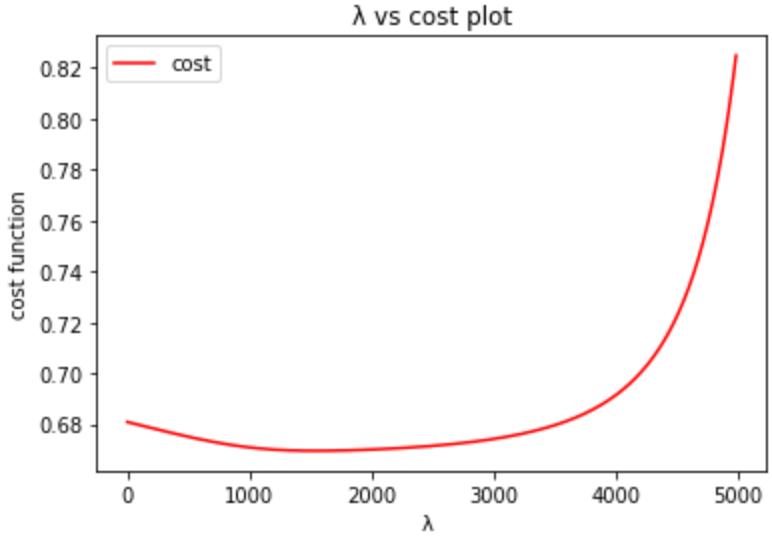


We got weights W = [-15.90210222 0.13522939 0.11802759]
When we predicted with the 30 test samples we got 26 of the predictions correctly
So accuracy is 86.66666666666667

When we draw the graph between two features and then plotting the curve
y = - (W0 + W1 * x) / (W2) predicted is



When we do regularization for lambda values 0 to 5000 with 100 epochs and learning rate 10^-3we get the follow graph :



Problem Statement 2:

Using the data set of two quality test results of a microchip product, design a predictor using logistic regression which will predict the acceptance or rejection of the microchip given the two test results. Use regularizer to further tune the parameters. Use 70 %

data for training and rest 30% data for testing your predictor and calculate the efficiency of the predictor/hypothesis.

Hints: 1. You can pre process the data for convenience

2. You must use Python program for evaluating parameters using batch gradient descent algorithm (GDA). No function should be used for GDA.

Analysis

The Output for this can be seen in IIT2016039_Assignment3_q2.html

In the given data we have two features exam1 and exam2 and we have 100 samples.

We have separated the data into 70 training samples and 30 testing samples

We are taking input features X = [1, test1, test2, test1^2, test2^2, test1*test2]

Our weights will be W = [W0, W1, W2, W3, W4, W5]'

Logistic regression predictor function H(X) = 1 / (1 + e^(- W.X))

For H(x) <= 0.5 we take it as 0 i.e W.X >= 0

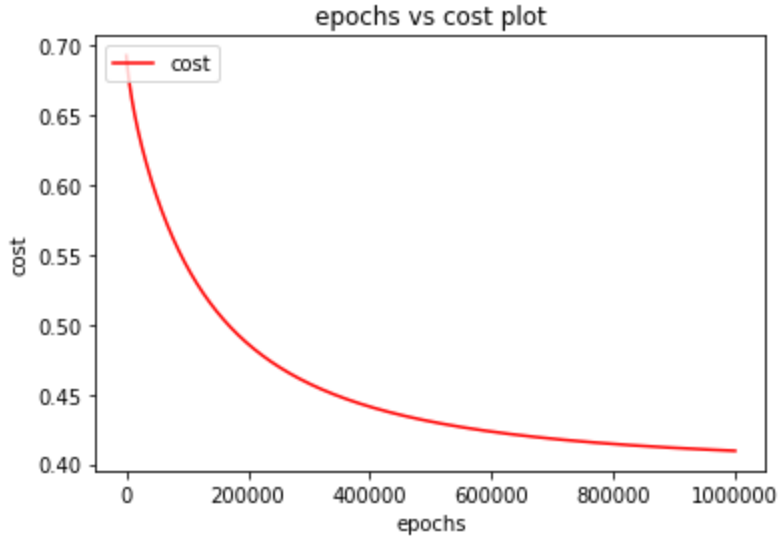
For H(x) > 0.5 we take it as 1 i.e W.X >= 0

Cost function for logistic regression is - (1 / len(Y)) * (sum(Y * log(H) + (1 - Y)*log (1 - H)))

We took learning rate = 10^-3

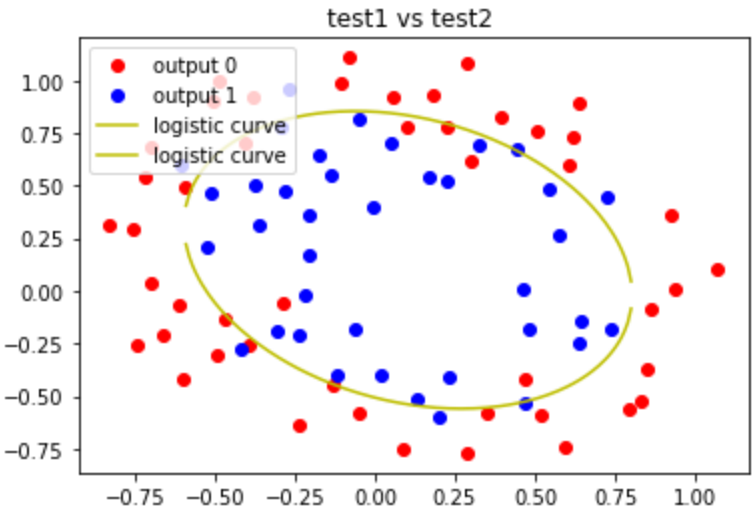
Epochs = 1,000,000

After applying gradient descent we got Final cost of 0.4095800517193937 graph is as follows :



We got weights W =[3.77545406,
2.46296211,
3.005693,
-8.92530275,
-8.74500698,
-4.19038197]

When we predicted with the 33 test samples we got 26 of the predictions correctly
So accuracy is 78.78787878787878
When we draw the graph between two features and then plotting the curve y = - ((W2 + W5 * x) (+/-) (W2 + (W5 * x)) * ((W2 +(W5 * x))) - (4 * W4 * (W0 + (W3 * x * x) + (W1 * x)))) / (2 * W4) predicted is



When we do regularization for lambda values 0 to 10000 with 100 epochs and learning rate 10^-3 we get the follow graph :