CSE 440
Assignment 6

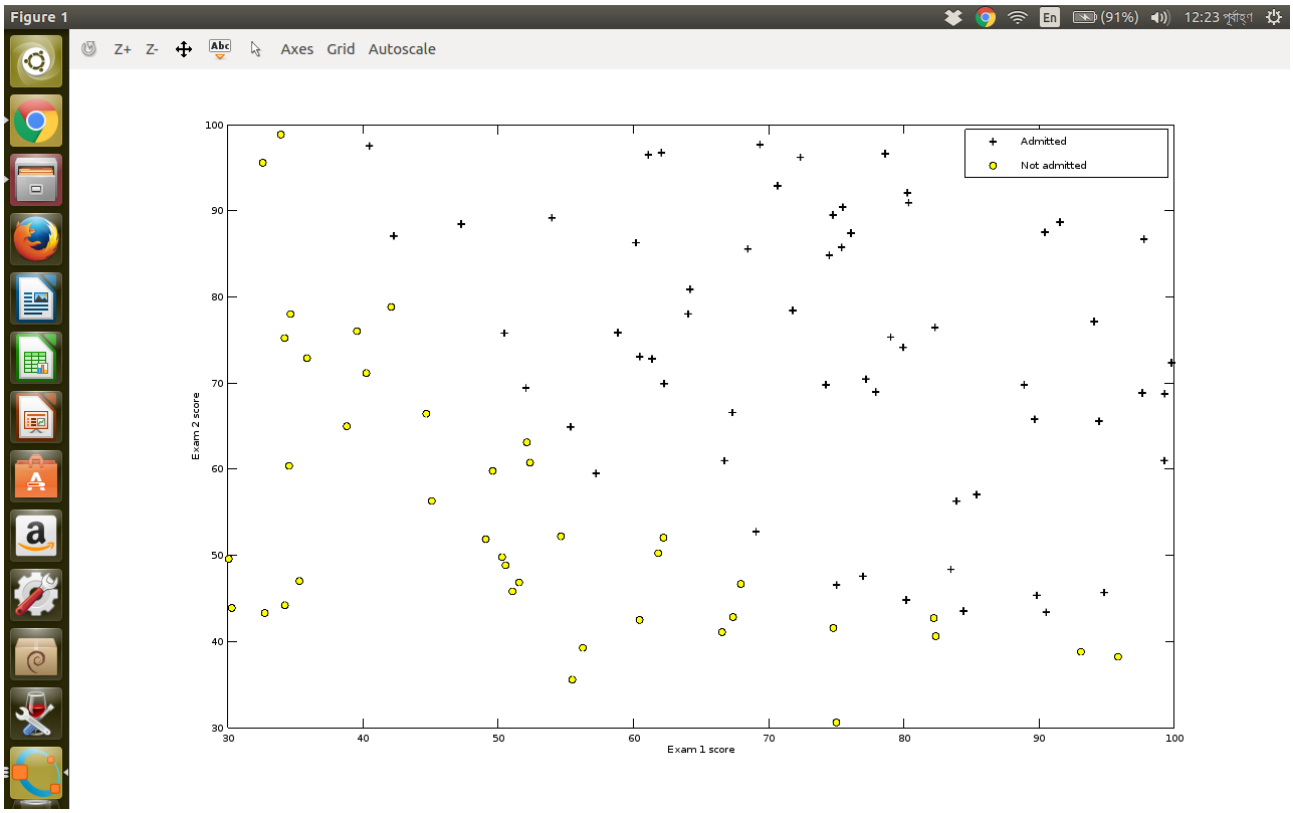
Submitted to -
M Ehsanul Karim

Submitted by -

Mahbuba Tasmin
ID – 1610064042

Scatter plot training data :

Before starting to implement any learning algorithm, it is always good to visualize the data if possible. In the first part of ex2.m, the code will load the data and display it on a 2-dimensional plot by calling the function plotData.

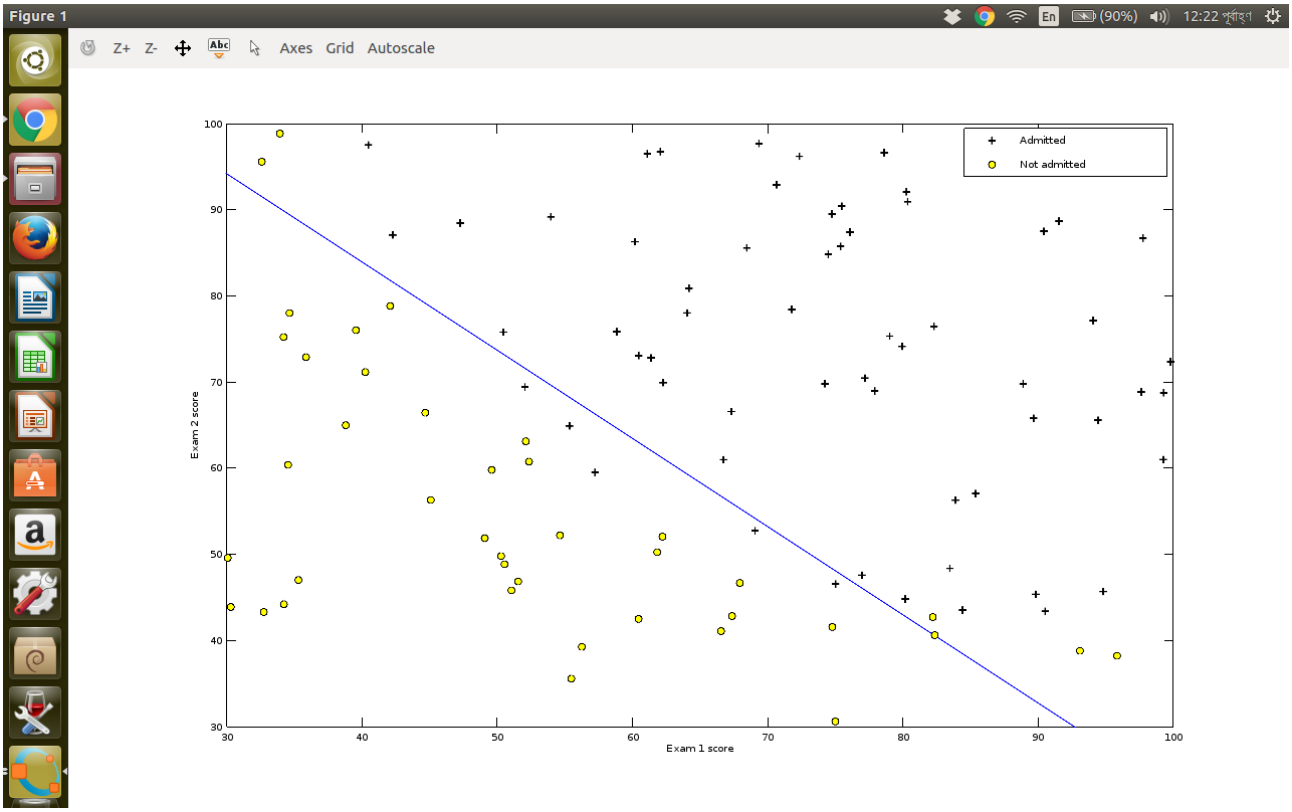


On the x- axis, exam 1 score is plotted and on the y axis, exam 2 score is plotted. As the problem mentioned, probability of a student getting admitted in the university based on the score of this two exams has to be obtained. Hence, this two scores are the training set for logistic regression and the value of y for each training set is either '+' or yellow circle plotted in the graph.

Training Data with Decision Boundary :

After implementing costFunction.m , cost and gradient for logistic regression will be returned. For logistic regression, fminunc is going to be used to optimize the cost function J(θ) with parameters θ, to find the best parameters θ for a given fixed dataset(of X and Y values).

Once fminunc completes, ex2.m will call costFunction function
using the optimal parameters of θ and this final θ value will then be used to plot the decision boundary on the training data.

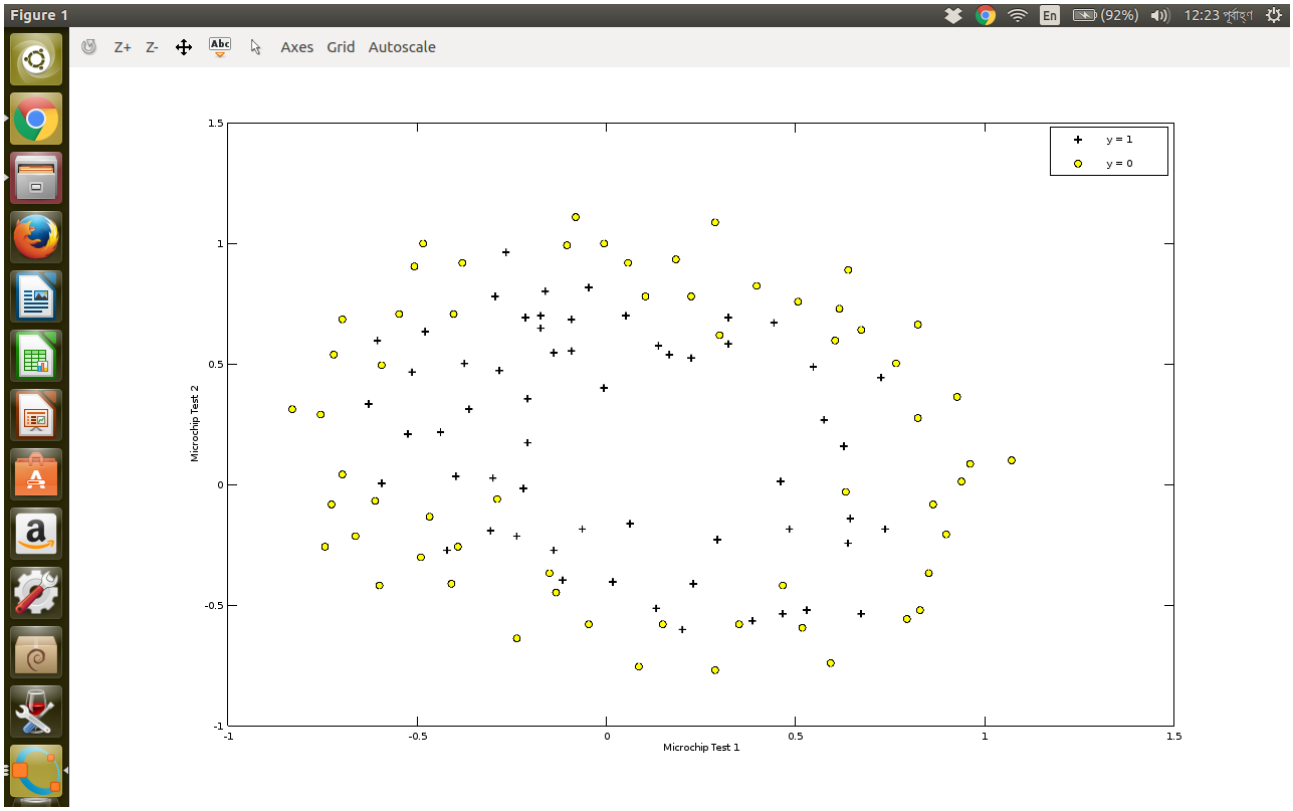


For $h_\theta(x) = g(\theta_0 + \theta_{1x1} + \theta_{2x2})$
from the graph , for $\theta_0 = -94$ (approximate) ,predict y = 1 if $-94 + x_1 + x_2 > 0$,
($-94 + x_1 + x_2$ this is the $\theta^t X$, used in calculating $h_\theta(x)$)

which comes as $x_1 + x_2 > 94$, when y = 1. Hence, the values upward the straight line x/94+y/94 = 0 is positive i.e. students get admitted above this limit and vice versa below.

Regularize logistic regression plot data:

In this part of the exercise, regularized logistic regression is implemented to predict whether microchips from a fabrication plant passes quality assurance (QA) , either accepted or rejected.
Given a dataset of test results on past microchips, from which a logistic regression model is built and after wards,a graph is plotted with positive and negative scores from training set.



On the x axis, microchip test 1 and on the y axis, microchip test 2 is placed. And the positive (y = 1, accepted) and negative (y = 0, rejected) examples are shown with different markers.

This figure shows that our dataset cannot be separated into positive and negative examples by a straight-line through the plot. Therefore, a straight-forward application of logistic regression will not perform well on this dataset since logistic regression will only be able to find a linear decision boundary. In this case, non linear decision boundary will be the solution to find the best possible parameters.

Training Data with Decision boundary :

To help visualize the model learned by this classifier, we have provided the function plotDecisionBoundary.m which plots the (non-linear)decision boundary that separates the positive and negative examples. In plotDecisionBoundary.m, we plot the non-linear decision boundary by computing the classifier's predictions on an evenly spaced grid and then and drew a contour plot of where the predictions change from y = 0 to y = 1.

After learning the parameters θ, the next step in ex reg.m will plot a decision boundary similar to the below figure.

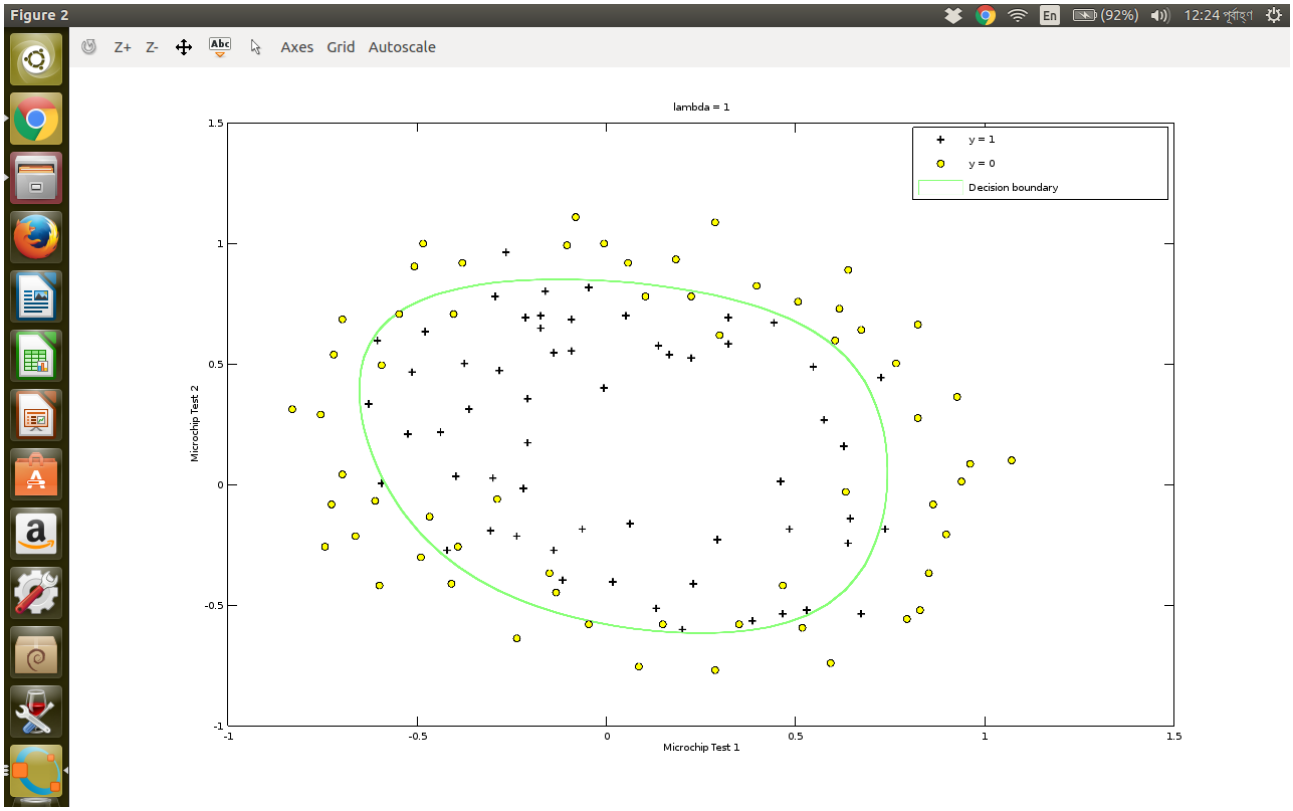


Figure 4: Training data with decision boundary (λ = 1)

λ, which is the regularization parameter and which task is to shrink all parameters except θ_0 to perfectly fit all the training set examples and keep an option for predicting future generalized parameter values.

For λ = 1 , the decision boundary is able to predict mostly accurate values within the non linear curve . For different values of λ , logistic regression will be regularized differently and hence the boundary curve also gets different.

For λ = 0 (too small value) :

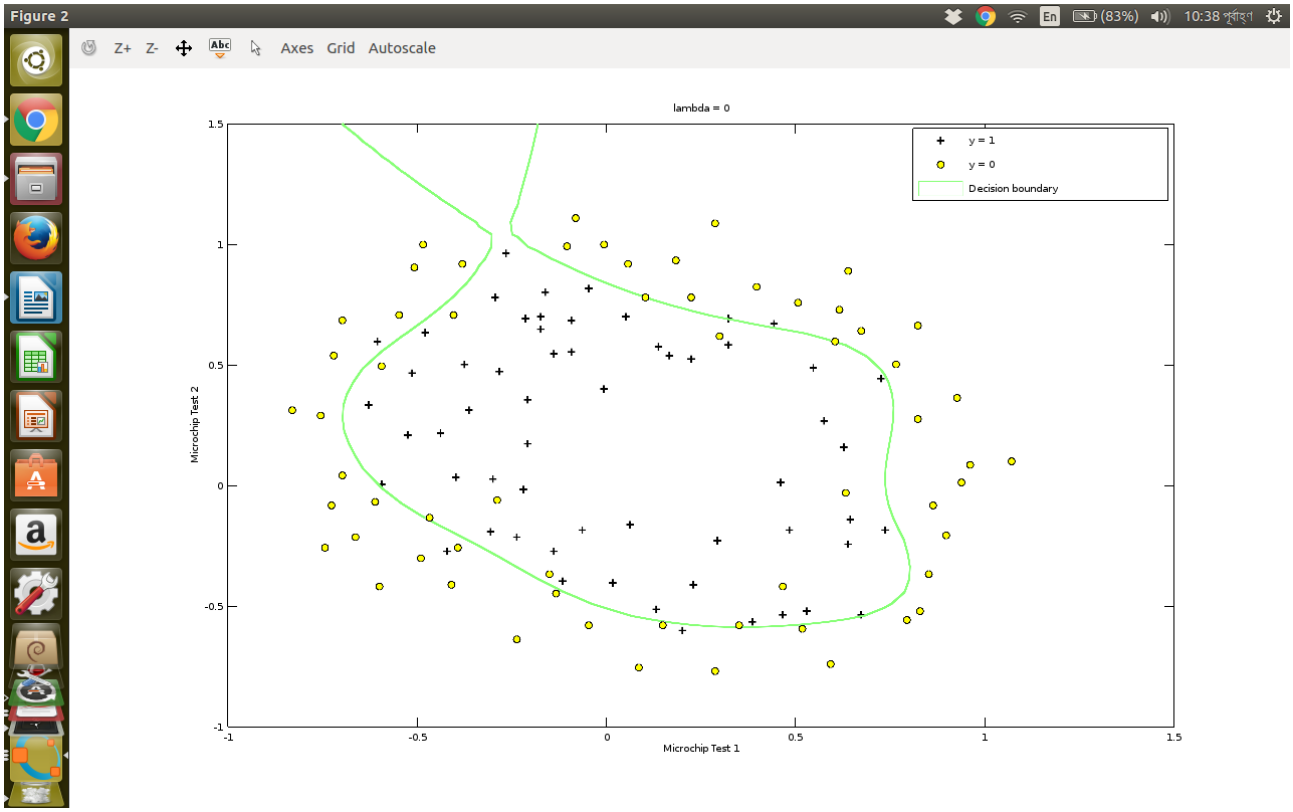


Figure 5: No regularization (Overfitting) (λ = 0)

With a small λ, we find that the classifier gets almost every training example correct, but draws a very complicated boundary, thus overfitting the data.And this boundary curve won't be able to predict generalized value for future training example set of data.This is not a good decision boundary: for example, it predicts that a point at x = (−0.75, 0) is accepted (y = 1), which seems to be an incorrect decision given the training set .

For λ = 100(large value):

here , for larger value of λ , all the parameters (θ) tends to approximately 0, hence in the end, only $h_\theta(x) = \theta_0$ remains and it results in underfitting the data set.

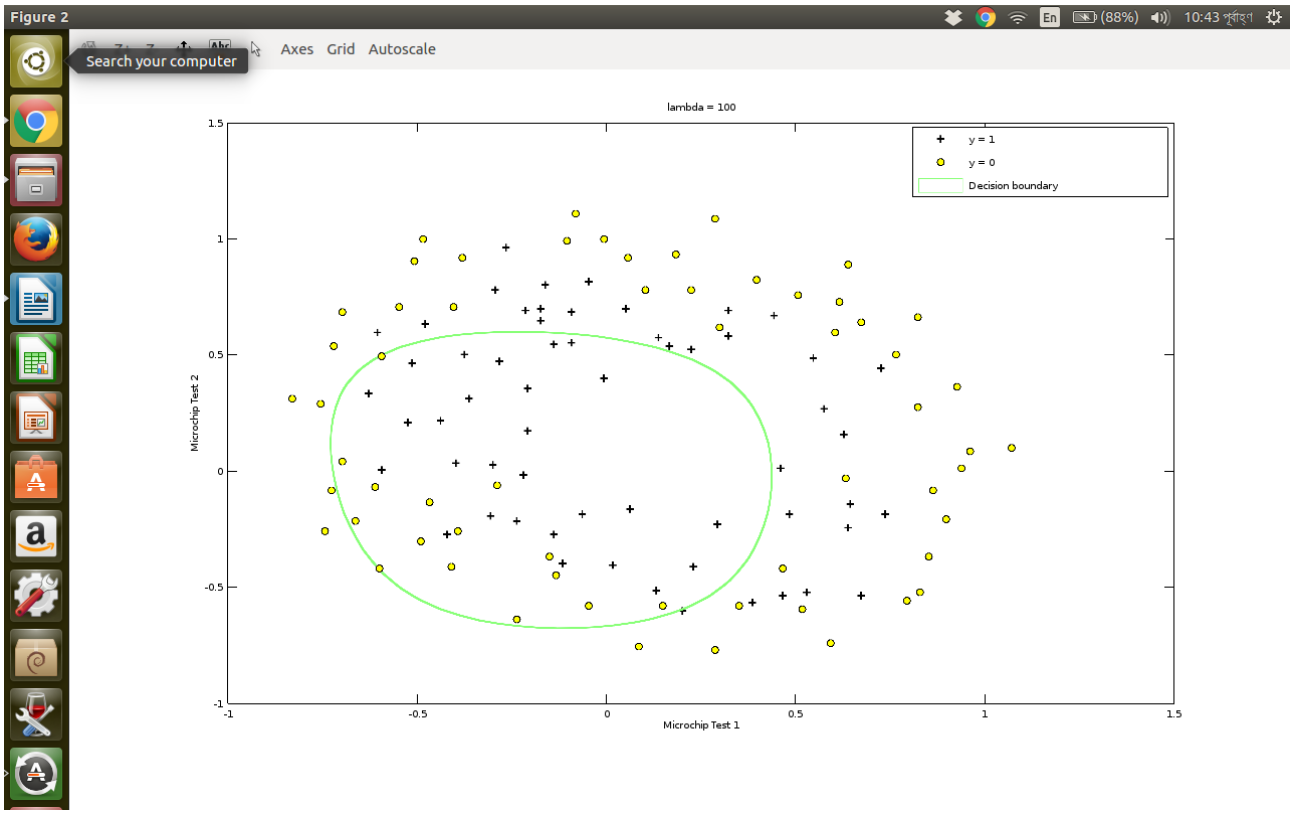


Figure : Too much regularization (Underfitting) (λ = 100)

With a larger λ, a plot is found that shows a decision boundary which separates the positives and negatives fairly well but excludes some good amount of data too. Thus , those data which are outside the boundary proves that this decision boundary is underfitting the data set.