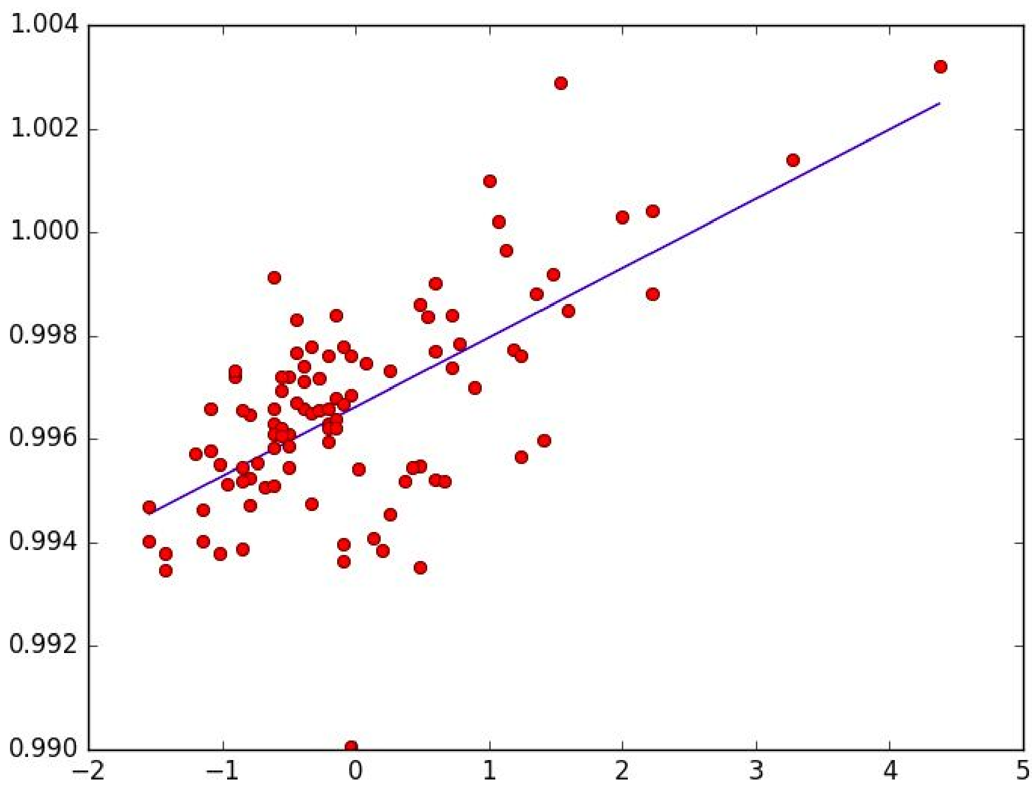
COL772 ASSIGNMENT 1
MACHINE LEARNING

Saket Dingliwal
2015CS10254

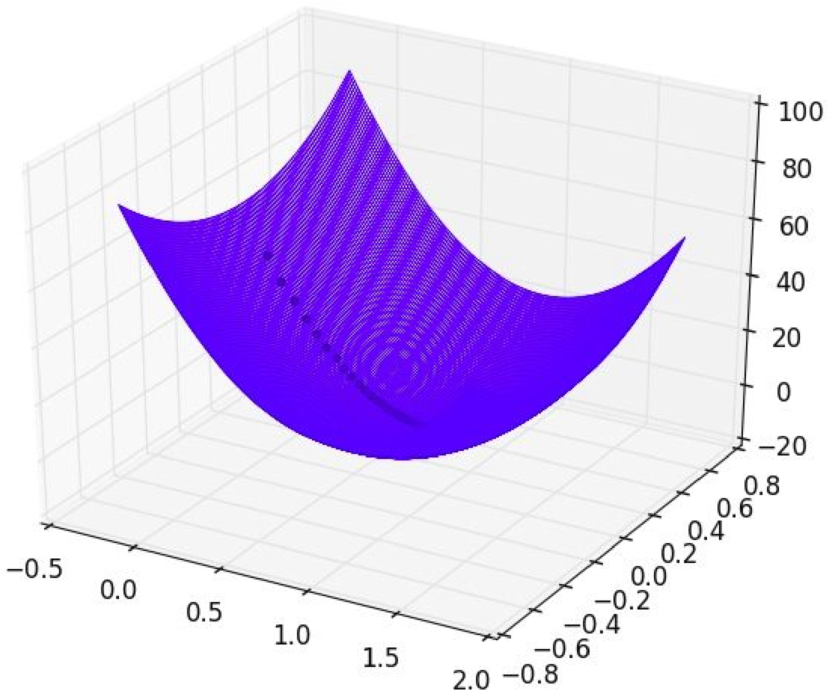
Part 1

Normalization of data is done. Linear Regression was done which works well for any number of input feature vectors. Gradient Descent was used for obtaining minima of the error function. Convergence condition used is the difference of error value approaching a small value.

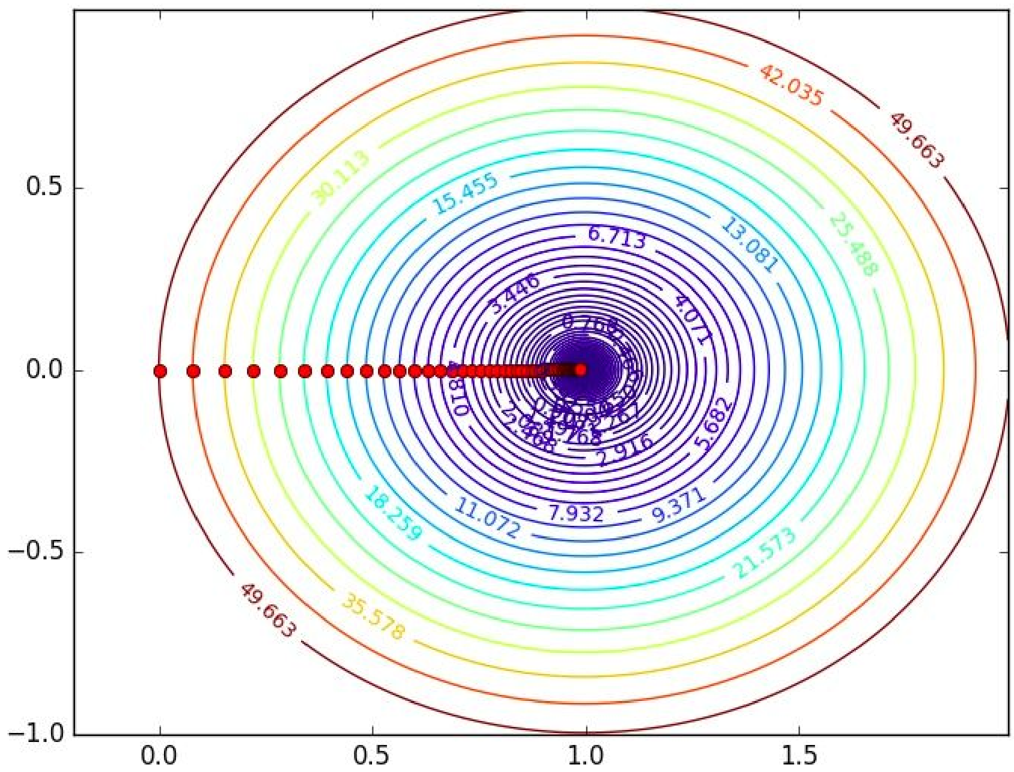
Epsilon used = 10^(-15)
Learning rate = 0.0008
Number of Iterations = 222
Value of J-theta = 0.00011947898
Result => y = 0.00134 x + 0.9966
Plot with normalized x on x-axis and y value on y-axis



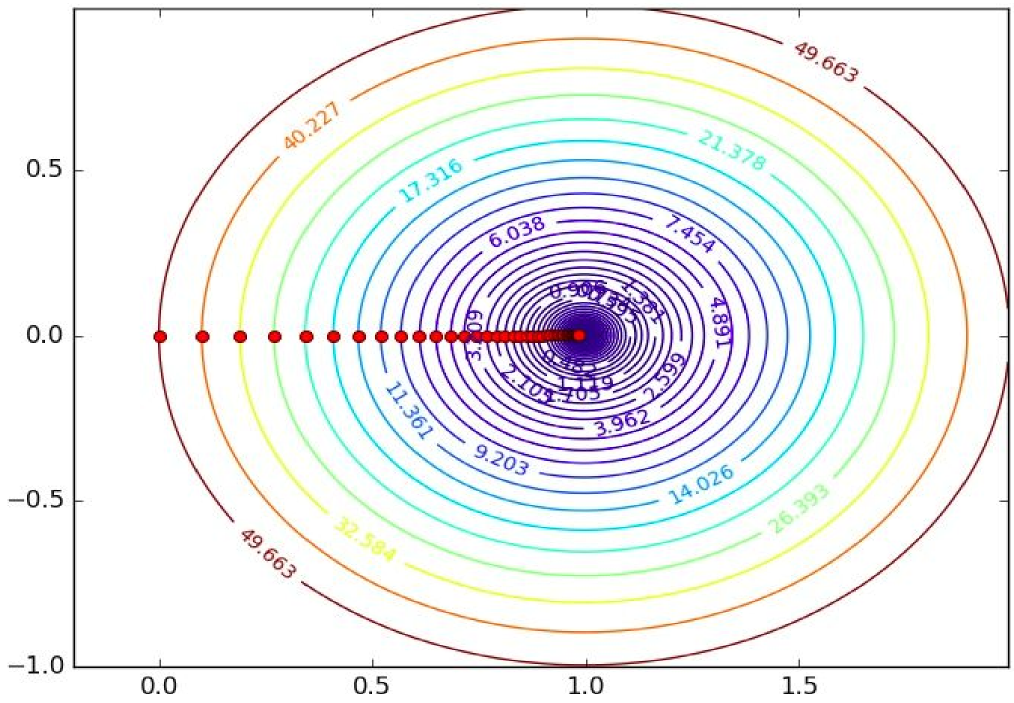
The 3-d plot for error function on Z-axis and theta0,theta1 on x,y-axis



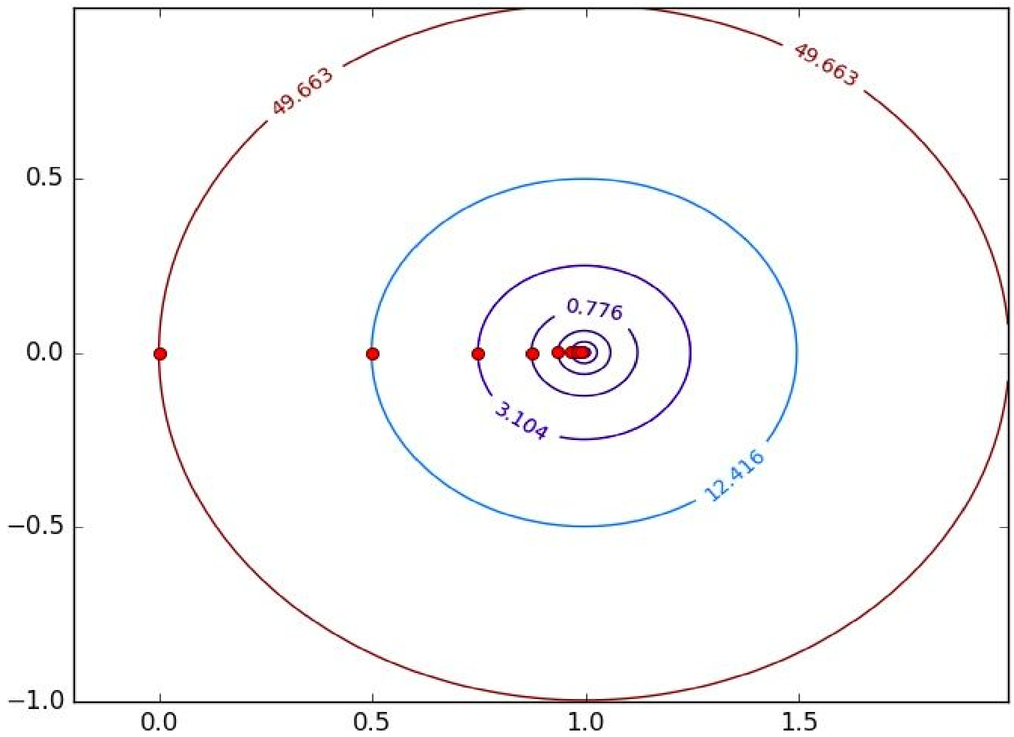
Contour plot for the learning rate = 0.0008 with theta0,theta1 on x,y-axis



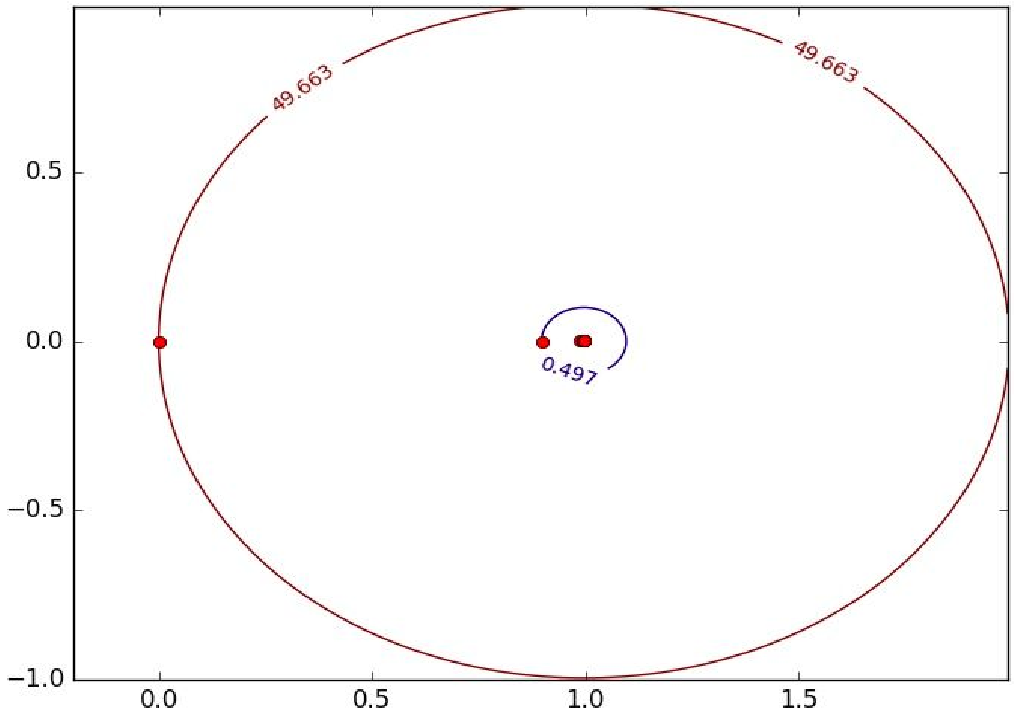
Contour Plots for different learning rates with theta0,theta1 on x,y-axis



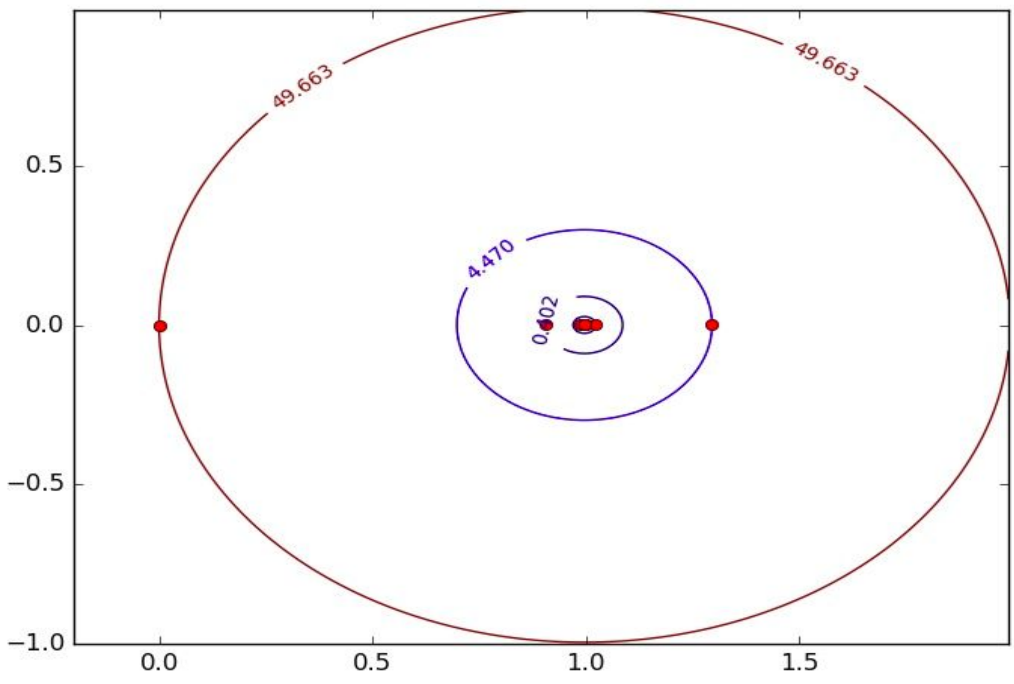
Alpha = 0.001 There is no oscillation in the value of theta



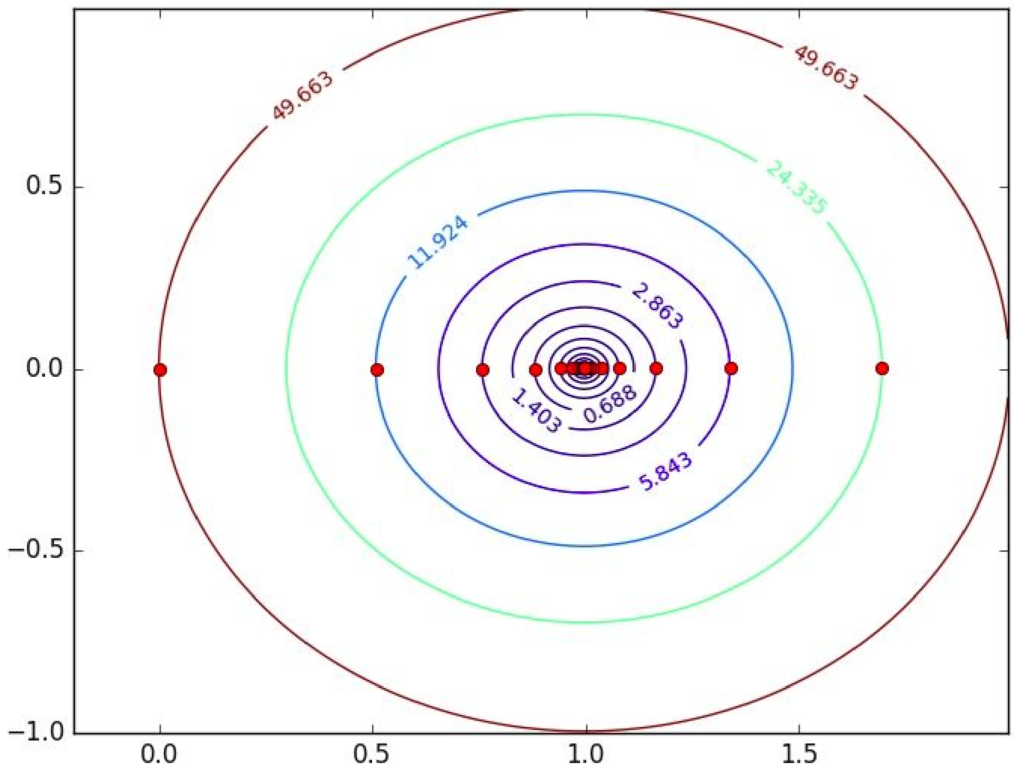
Alpha = 0.005 The number of steps for convergence decreases



Alpha = 0.009 the number of iterations for convergence decrease further



Alpha = 0.013 there is oscillations in theta values on left and right side and hence the number of iterations increased a little



Alpha = 0.017 The oscillations increases and so does the number of iterations

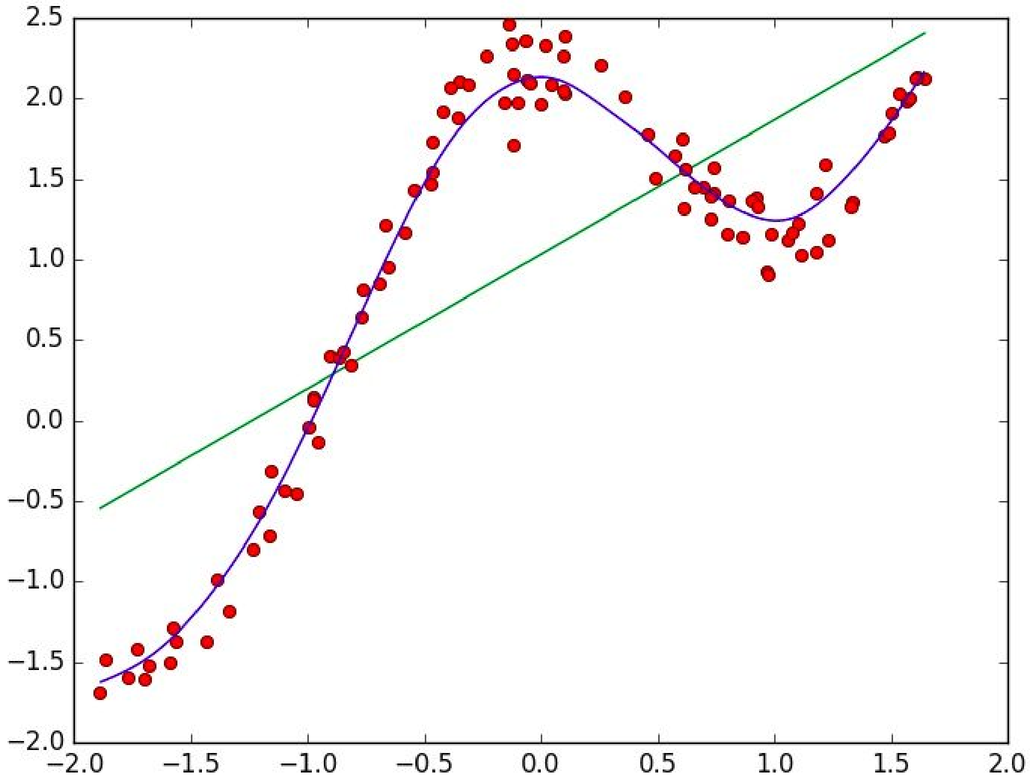
The gradient descent do not converge for alpha values = 0.021 and 0.025. The values of J-theta shoots to infinity after a few iterations.
We observe that the number of iterations required are minimum for alpha = 0.009 as there is no oscillations and value of j-theta decreases very quickly. So too small learning rate may increase the number of steps to be taken, while too large learning rates may lead to oscillations or failure of gradient descent by taking j-theta to infinity. Hence it s value must be optimum between the two extremes.

Part 2

The values of theta obtained without using weights are
theta0 = 1.03128116
Theta1 = 0.83519315
Resulting line => y = 0.83 x + 1.031

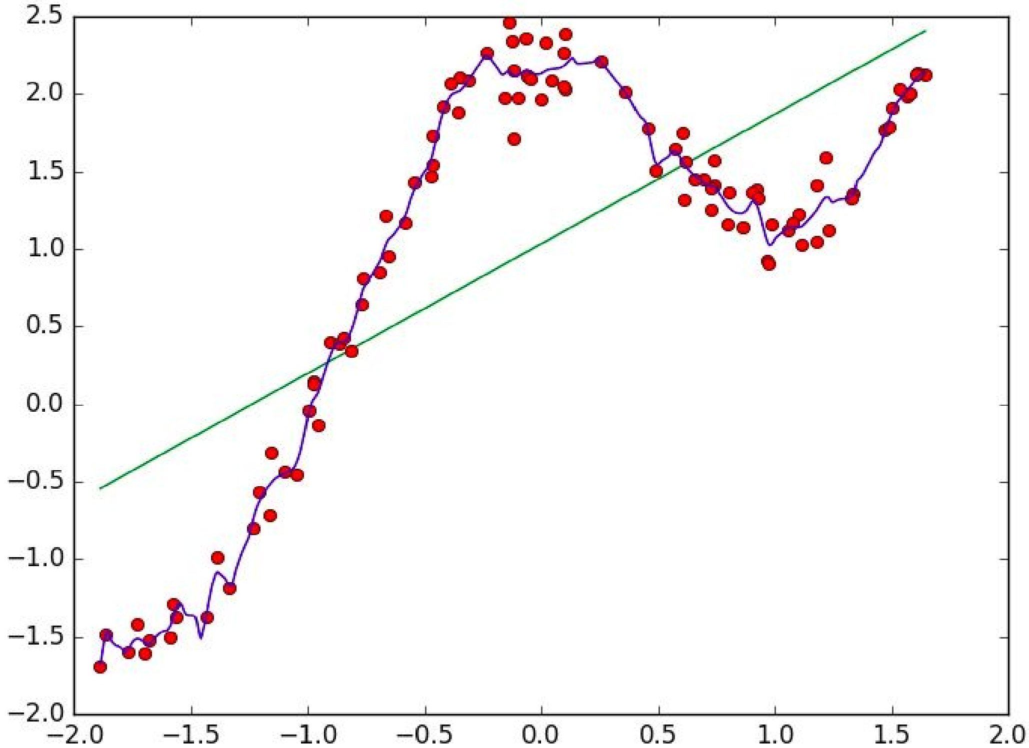
The plot for weighted linear model is done by choosing 1000 random points between maximum and minimum of x values and plotting them on graph.

T value used is 0.2 for the normalized data

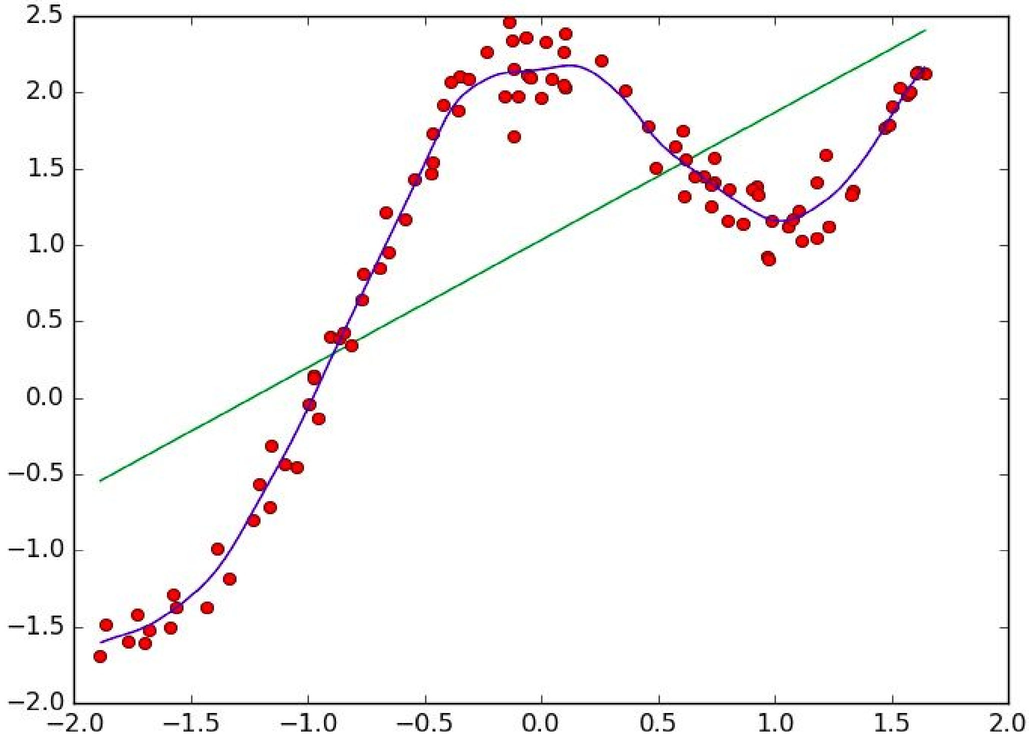


It can be observed that if smaller values of t ie 0.03,0.1 etc are used there is overfitting of the training data. The curve tries to pass through each and every training point and hence might not perform well on new data. For t = 0.2 a better fit is obtained. On increasing the value further, underfitting is observed. The obtained curve do not pass close to the training points. At t = 2,10 the curve is almost linear and is mostly same as unweighted linear regression.

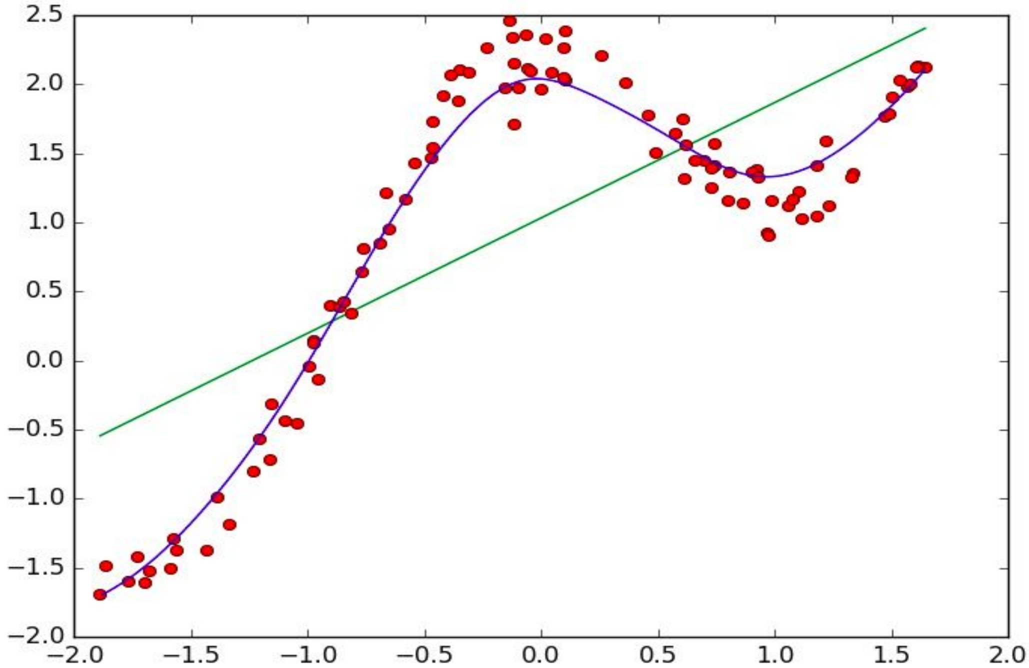
Plots for different t values are shown



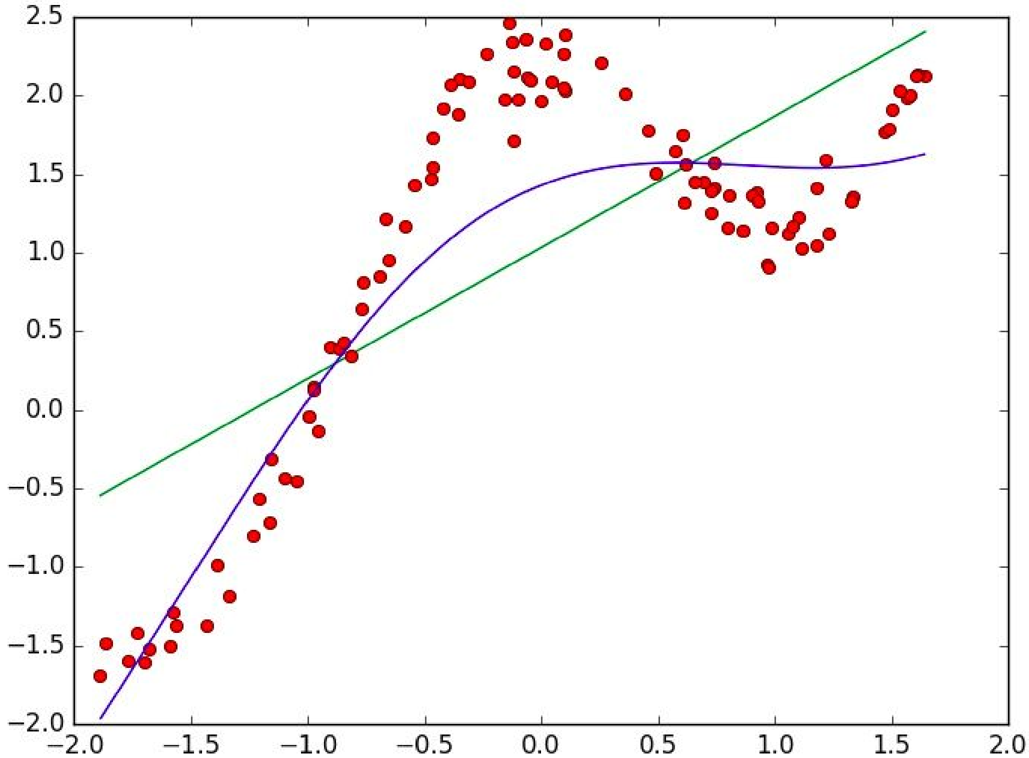
T = 0.01 (overfitting)



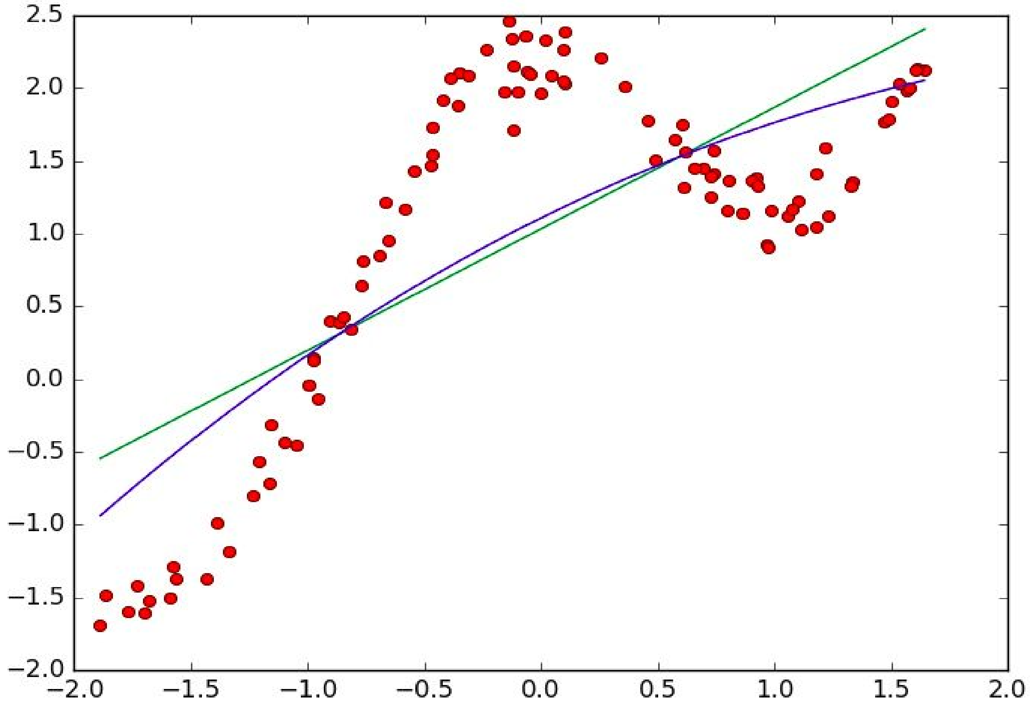
T = 0.1



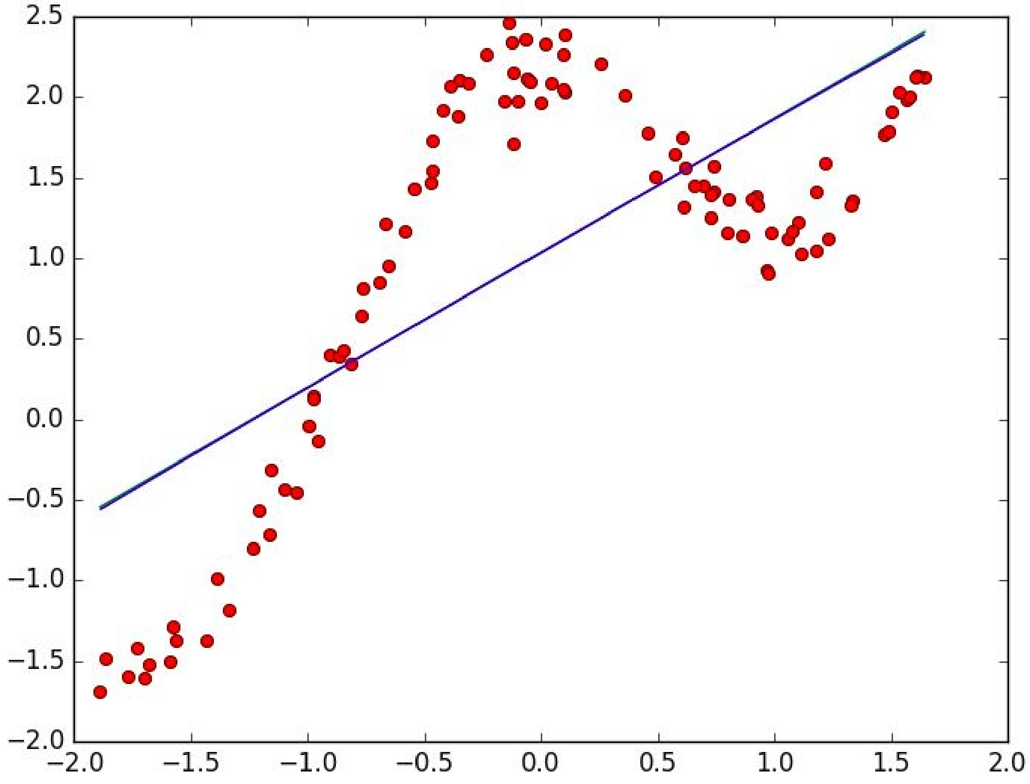
T = 0.3 (good fit)



T = 0.8 (underfit)



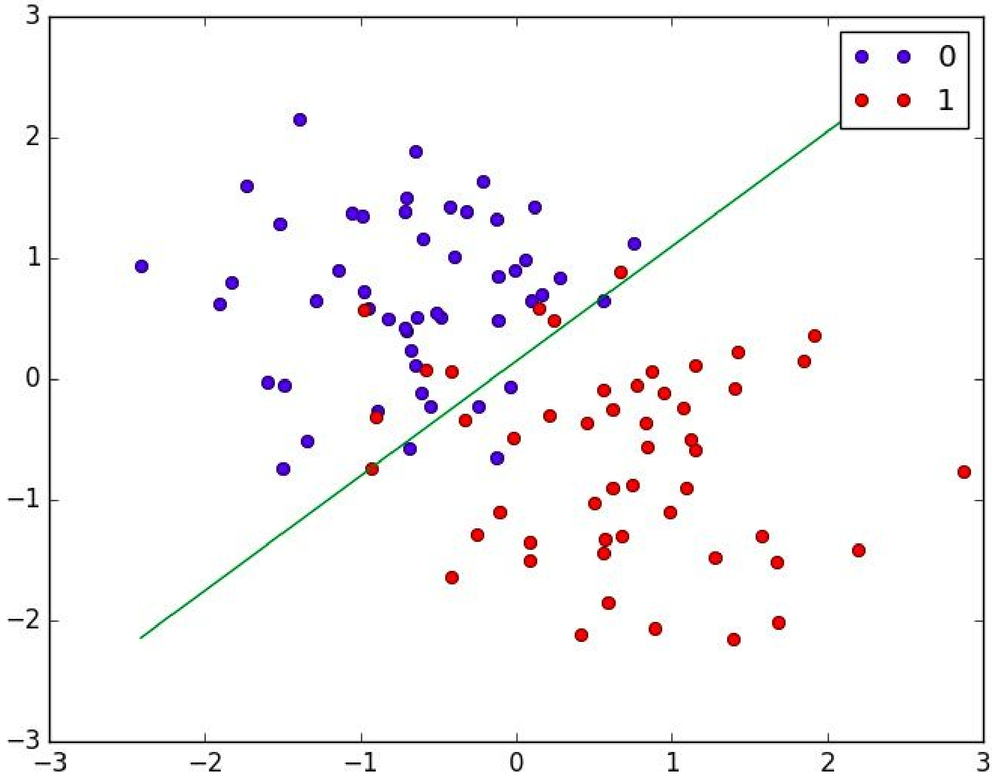
T = 2 (underfit)



T = 10 (completely linear)

Part 3

The values of the theta parameter obtained from newton method for epsilon = 10^(-10) is [[0.40125316], [2.5885477] ,[-2.72558849]]
The corresponding equation of the line of the decision boundary is
0.401 + 2.588 x - 2.725 y = 0

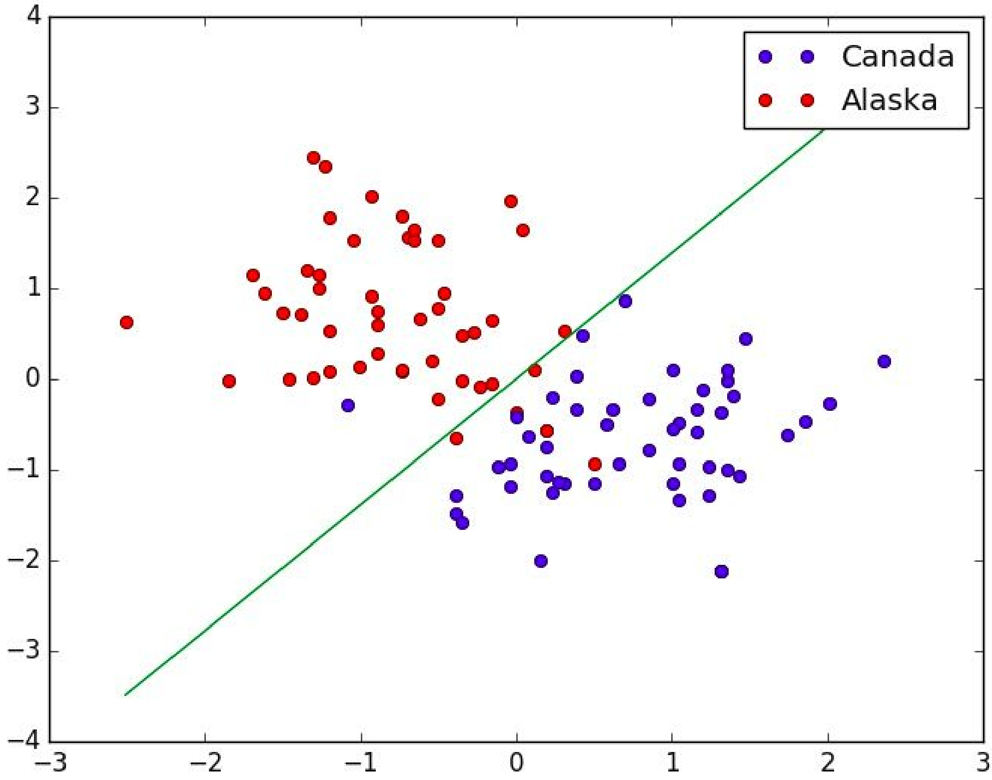


Part 4

The values of the parameters obtained if value of covariance matrices is assumed to be equal are
phi = 0.5
u(alaska) = [-0.75529433, 0.68509431]
u(canada) = [0.75529433, -0.68509431]

Covariance matrix = [[0.42953048, -0.02247228],
[-0.02247228, 0.53064579]]

The plot for the linear decision boundary comes out to be



The equation of the decision boundary comes out to be

Parameters = > theta [-1.7763568394e-15, -6.7785090, 4.877167]

Eqn = > -6.77 x + 4.877 y = 0

The formulae for the parameters of decision boundary is obtained as follows

For classification with 2 input features ->

theta[0] = 2 * log(1-phi) - 2 * log(phi) + u1_cov_u1 - u0_cov_u0

theta[1] = 2 * cov_u0[0][0] - 2 * cov_u1[0][0]

theta[2] = 2 * cov_u0[1][0] - 2 * cov_u1[1][0]

Where

$$u1\_cov\_u1 = u1\ Cov^{-1}\ u1^T$$

$$u0\_cov\_u0 = u0\ Cov^{-1}\ u0^T$$

$$cov\_u1 = Cov^{-1}\ u1^T$$

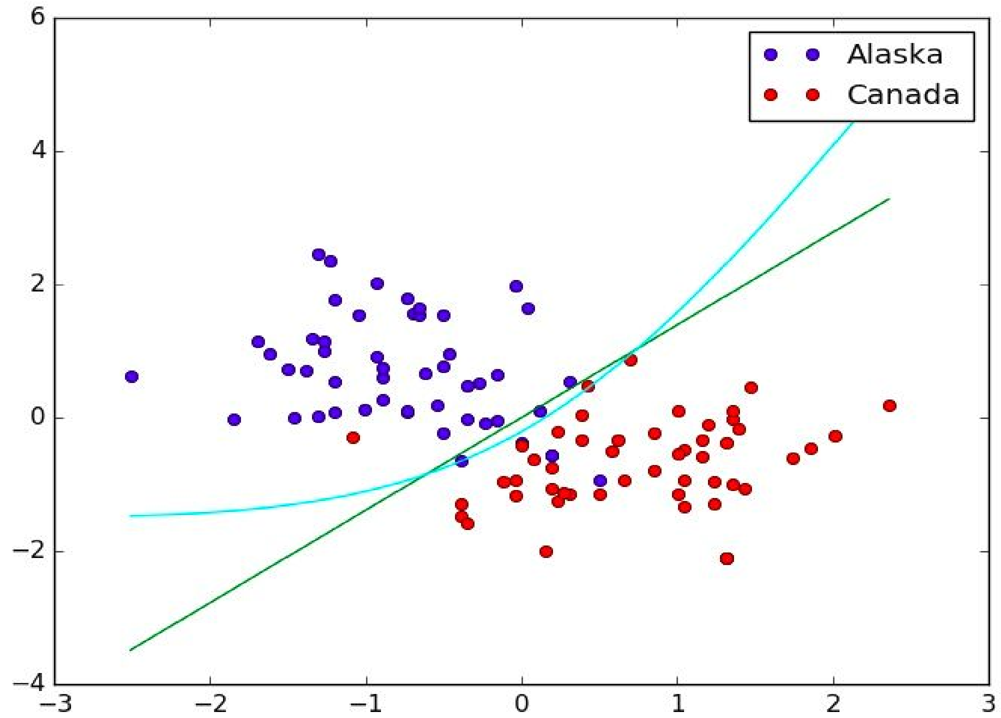
$$cov\_u0 = Cov^{-1}\ u0^T$$

For the quadratic decision boundary with different covariance matrices.
The values obtained are

$$\text{cov(alaska)} = \begin{bmatrix} 0.38158978 & -0.15486516 \\ -0.15486516 & 0.64773717 \end{bmatrix}$$

$$\text{cov(canada)} = \begin{bmatrix} 0.47747117 & 0.1099206 \\ 0.1099206 & 0.41355441 \end{bmatrix}$$

The plot for the same can be seen as



The boundary equation obtained is
-1.169 + 0.6713 x*x + 2.573 x*y - 0.865 y*y + 7.615 x - 5.71 y = 0

The formulae for the parameters obtained for 2 feature input is

const = log(abs(cov0_det/cov1_det)) + u0_cov0_u0 - u1_cov1_u1
Coefficient of x^2 = diff_mat[0][0]
Coefficient of y^2 = diff_mat[1][1]
Coefficient of xy = diff_mat[0][1] + diff_mat[1][0]
Coefficient of x = 2 * cov1_u1[0][0] - 2 * cov0_u0[0][0]
Coefficient of y = 2 * cov1_u1[1][0] - 2 * cov0_u0[1][0]

Where

$$cov0\_det = |cov0|$$
$$cov1\_det = |cov1|$$
$$u1\_cov1\_u1 = u1\ Cov1^{-1}\ u1^T$$
$$u0\_cov0\_u0 = u0\ Cov0^{-1}\ u0^T$$
$$cov1\_u1 = Cov1^{-1}\ u1^T$$
$$cov0\_u0 = Cov0^{-1}\ u0^T$$
$$diff\_mat = Cov0^{-1} - cov1^{-1}$$

The quadratic boundary obtained is more general form of GDA and have less stronger assumption about the data. The linear boundary seems to be a good separator if the x given y come from normal distribution with same covariance. However quadratic is more general, although more complex. Considering training data of the above problem, 2 more points are correctly classified by the quadratic separator over the linear one.