# COL774 - MACHINE LEARNING

## Assignment 1 Report

Submitted By
Mehak
2018MCS2143

Ques 1. Linear Regression

a. Batch Gradient Descent

Learning Rate : 0.001

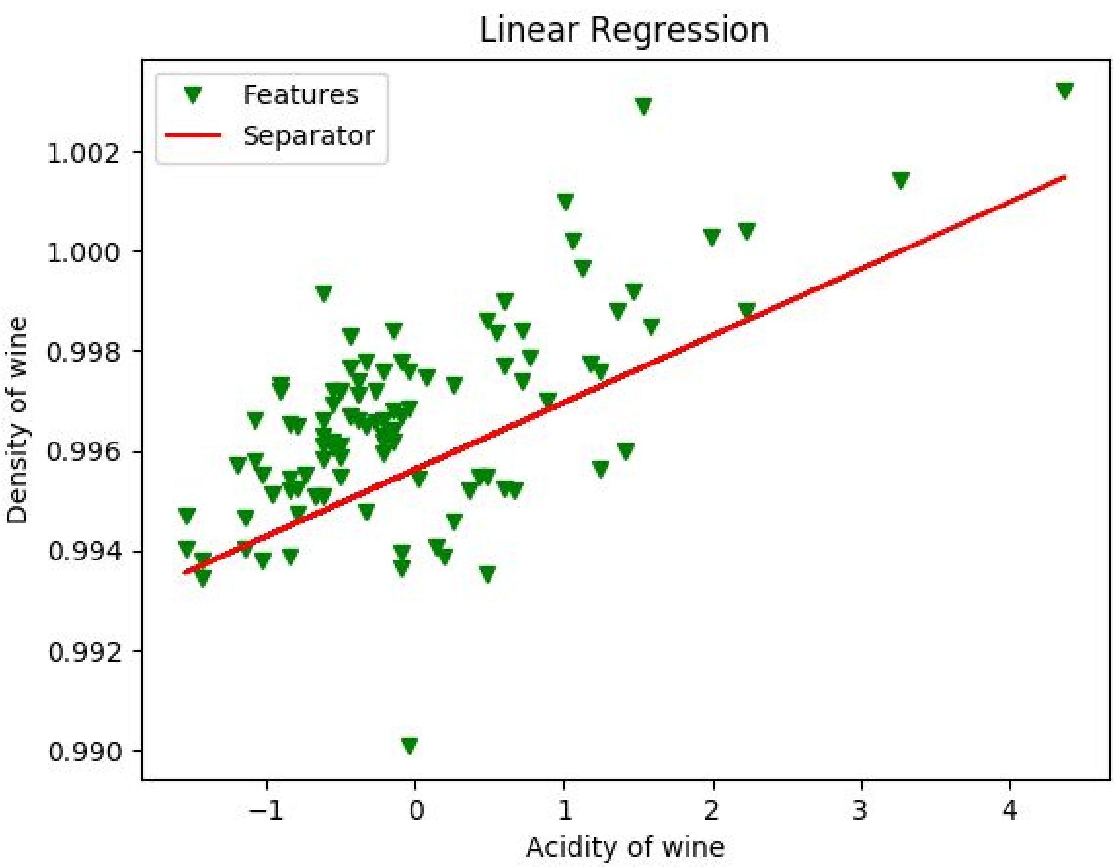
Stopping Criteria : J(theta+1)-J(theta) < epsilon where epsilon = 10^-9

Final sets of parameters obtained :

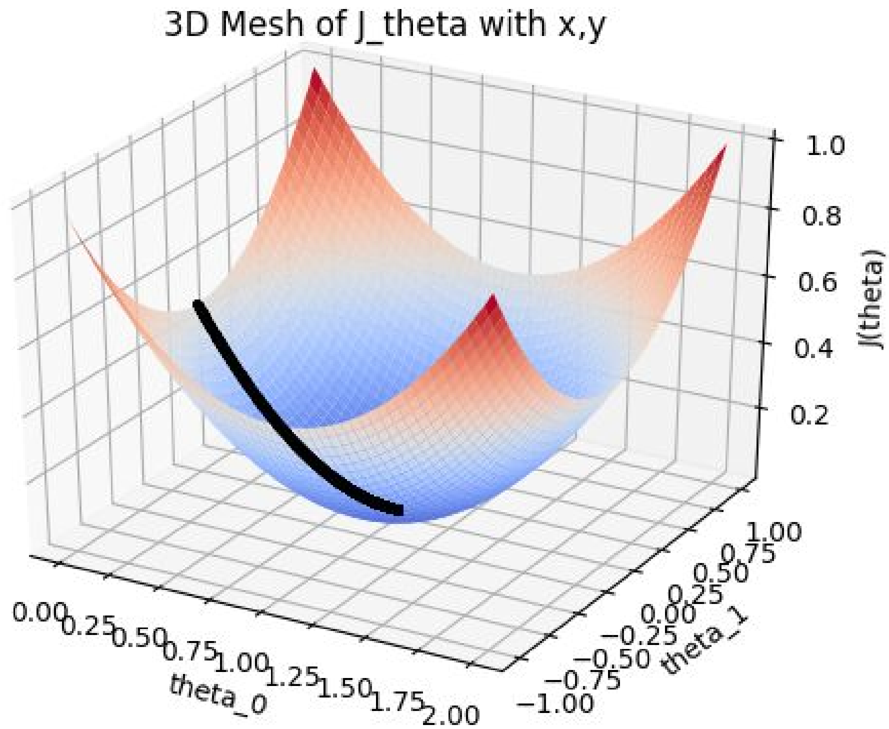
Theta0 =0.9965357445819697 Theta1=0.0013388527338389679

Error : 1.6937082384585006e-06

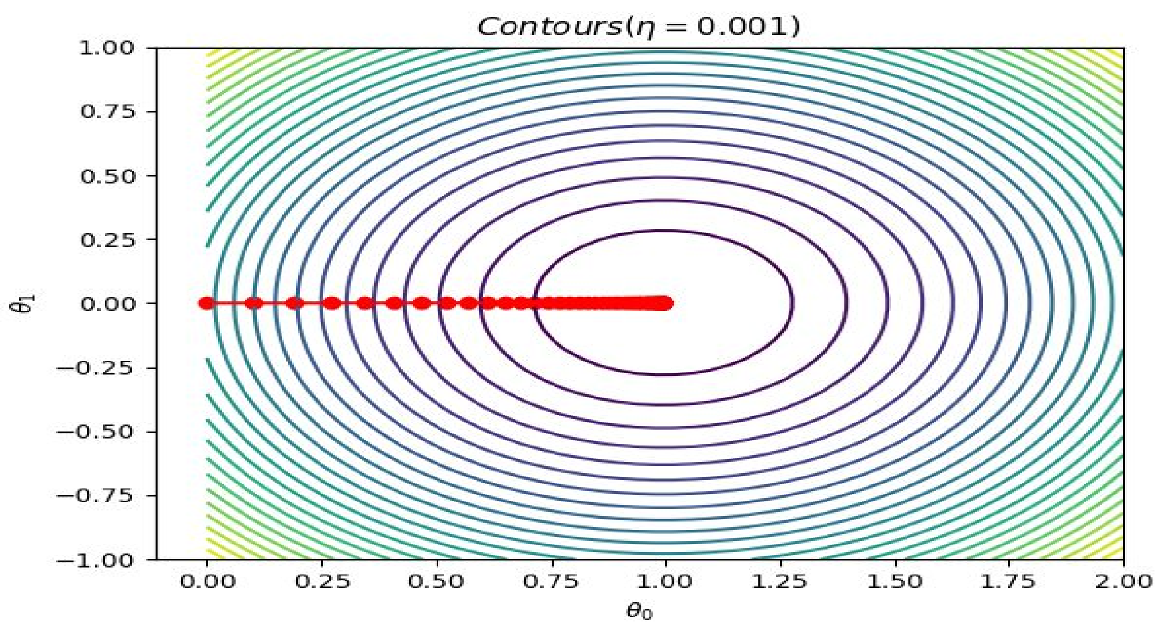
b.



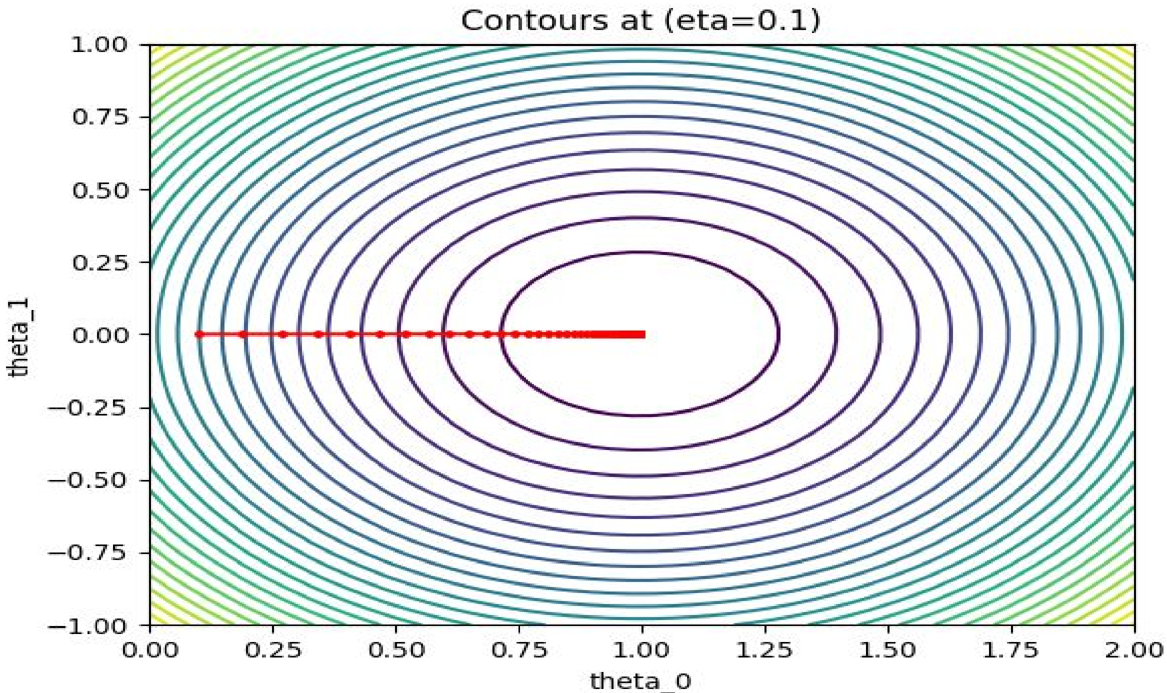
c.

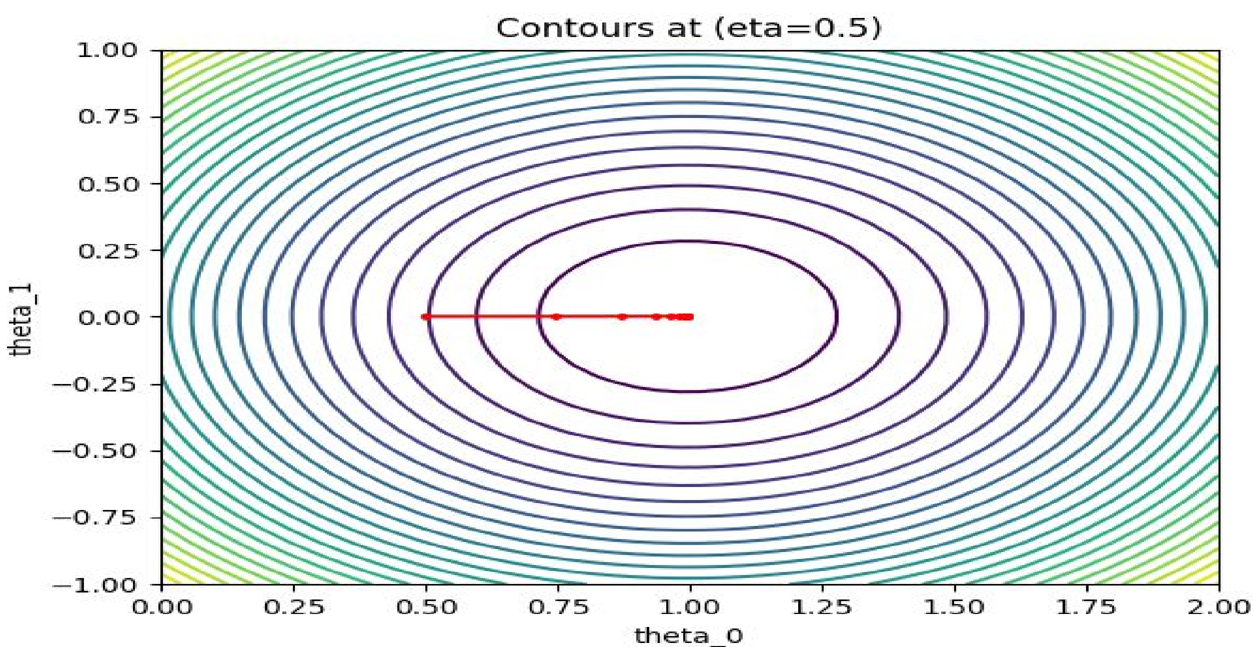

3D Mesh of J_theta with x,y
1.0
0.8
0.6
0.4
0.2
J(theta)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
theta_0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
theta_1

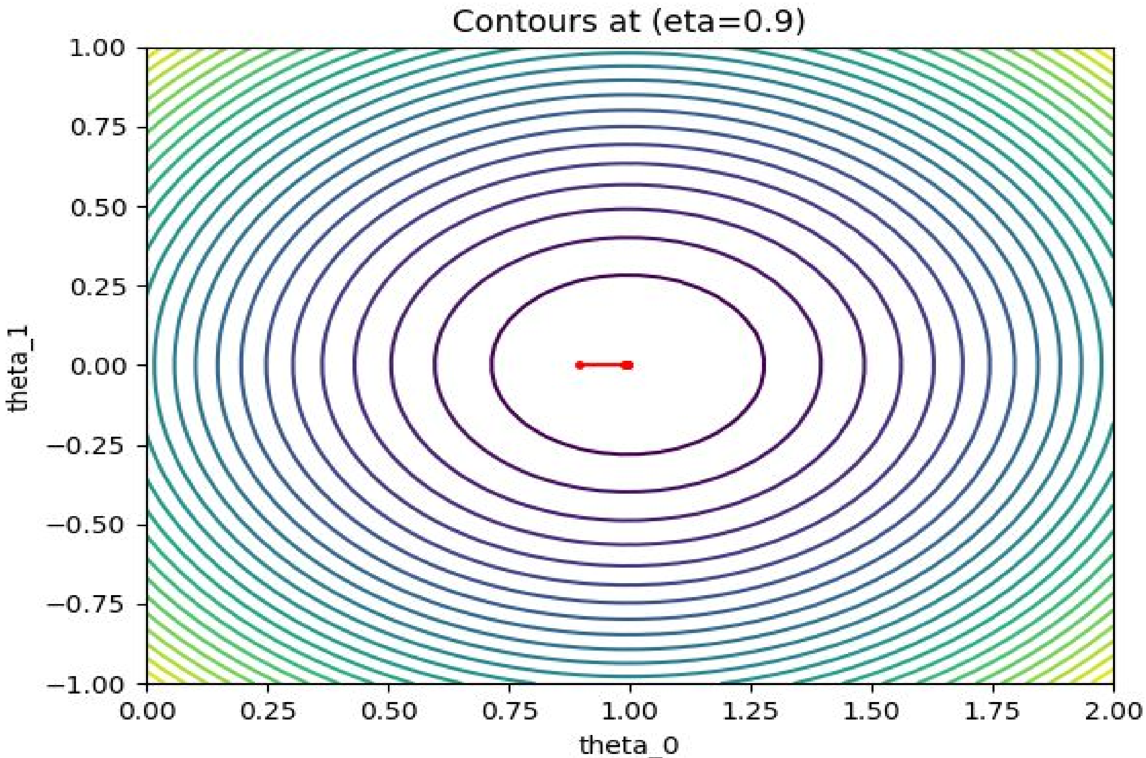
d.

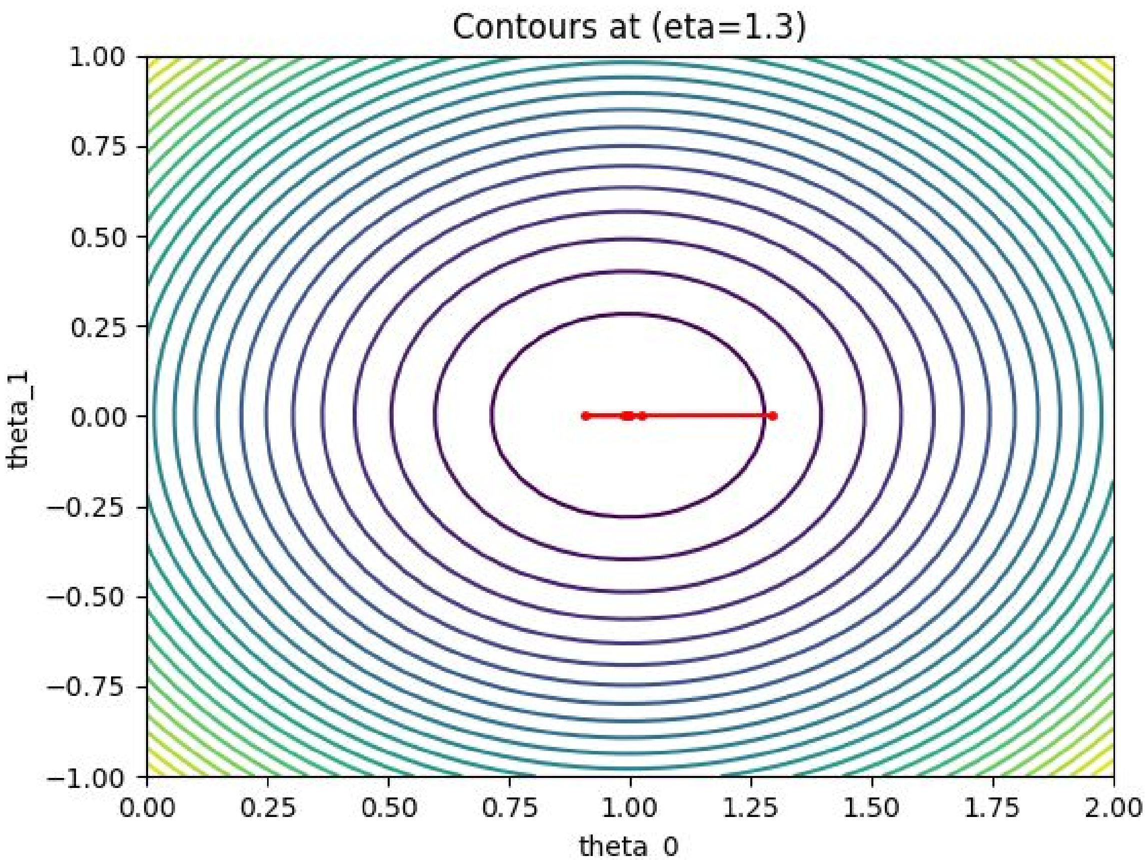

$Contours(\eta = 0.001)$
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
$\theta_1$
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
$\theta_0$

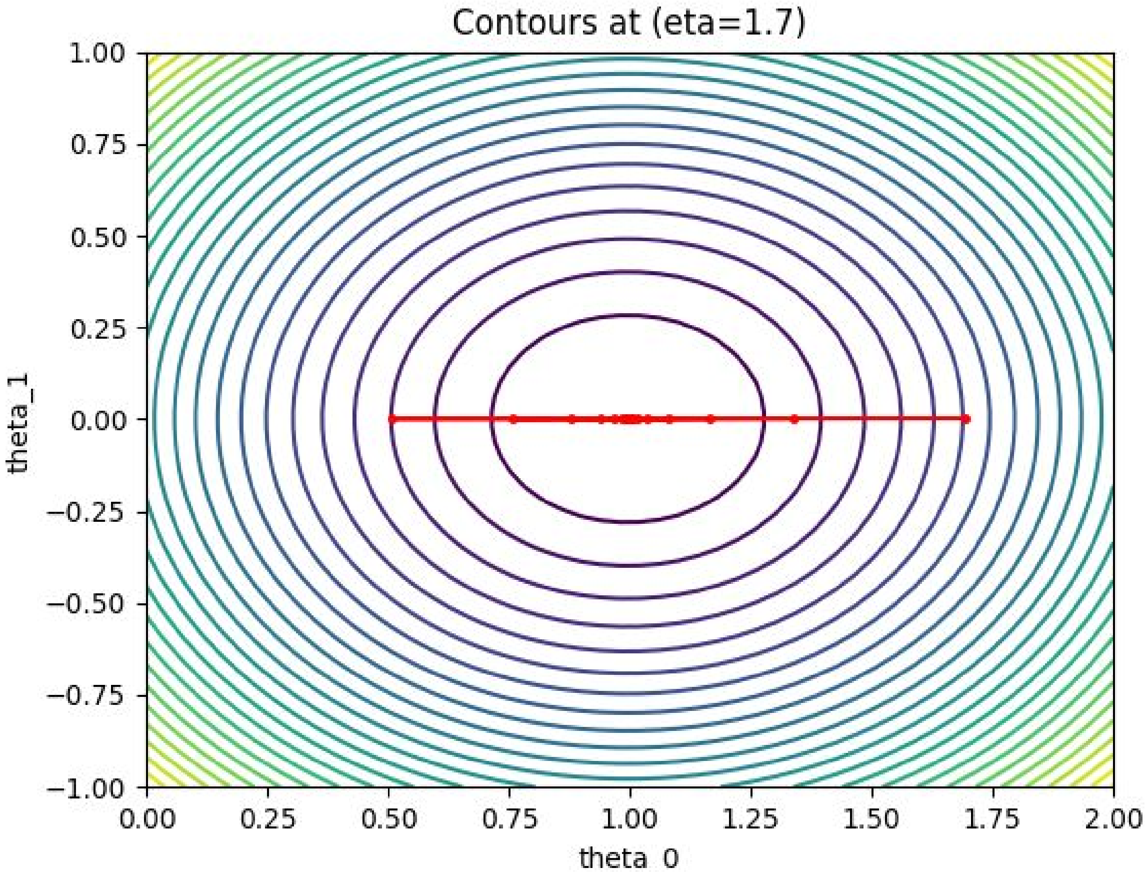
e.

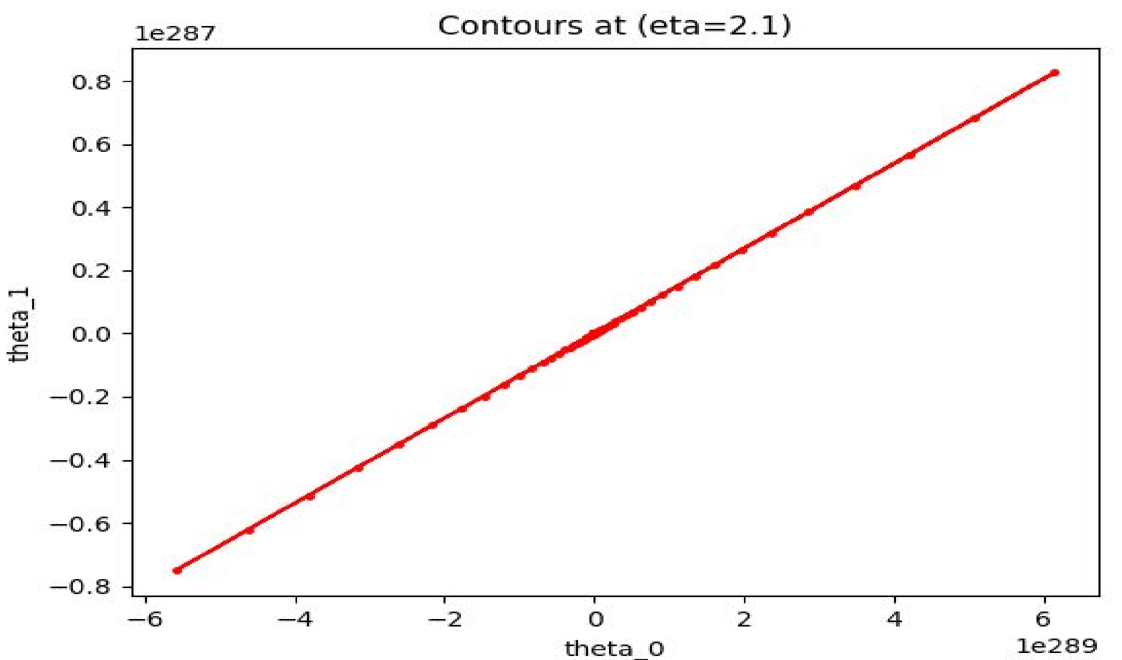

Contours at (eta=0.1)
theta_1
theta_0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

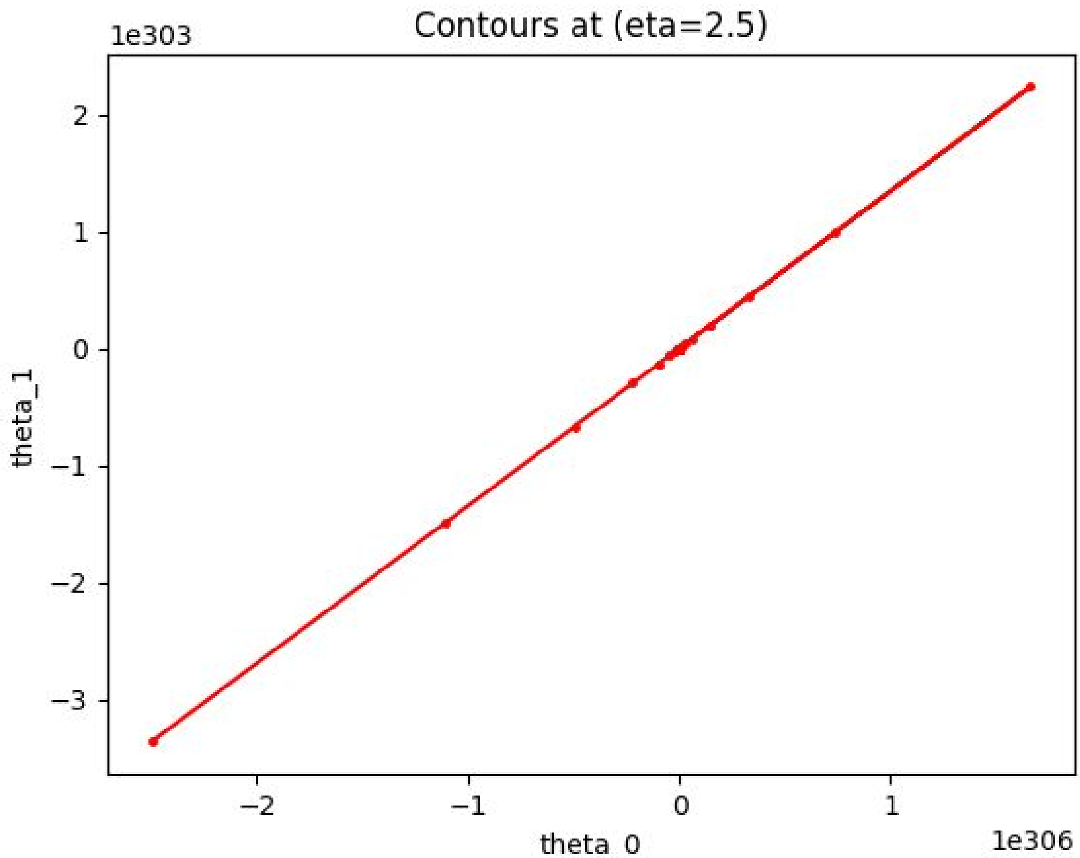

Contours at (eta=0.5)
theta_1
theta_0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Contours at (eta=0.9)
theta_1
theta_0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Contours at (eta=1.3)
theta_1
theta_0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Contours at (eta=1.7)
theta_1
theta_0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Contours at (eta=2.1)
1e287
theta_1
theta_0
1e289
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−6
−4
−2
0
2
4
6



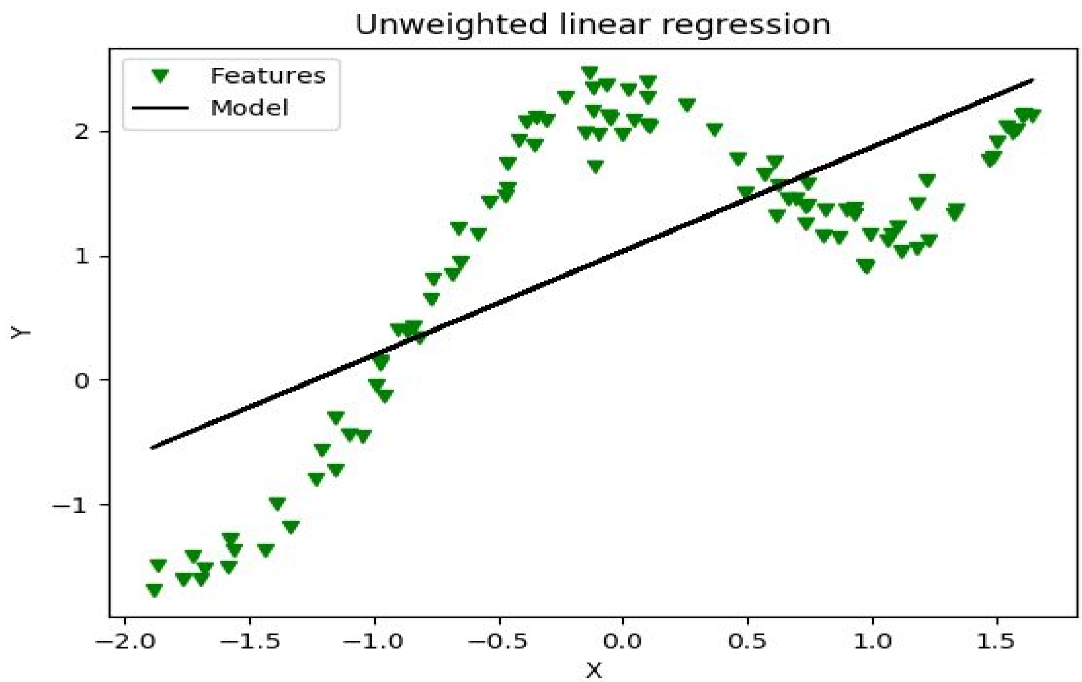

Observations

When the learning rate is small as eta = 0.001, linear regression will take very small steps and converge to the local minima after large number of iterations. When learning rate is increased, the step size increases and algorithm converges to local minima in less number of iterations.

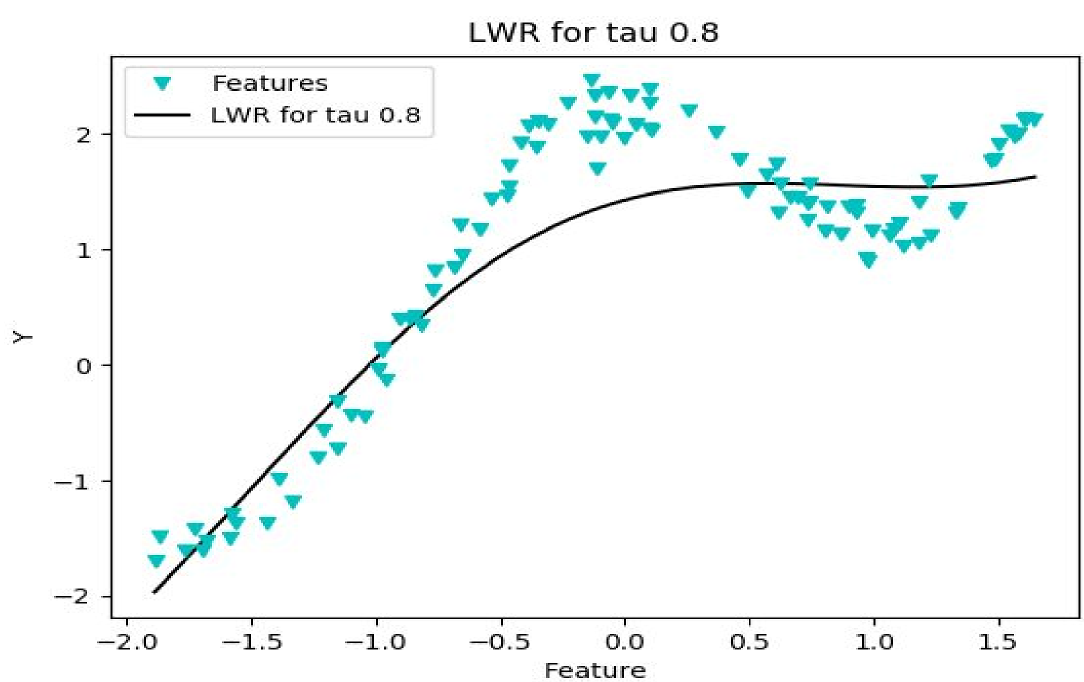
However, learning rate is further increased to a large value as 2, the algorithm will start taking very large steps thus missing local minima. And hence it will keep oscillating and never converge to the local minima.

Q2. Locally weighted Linear Regression

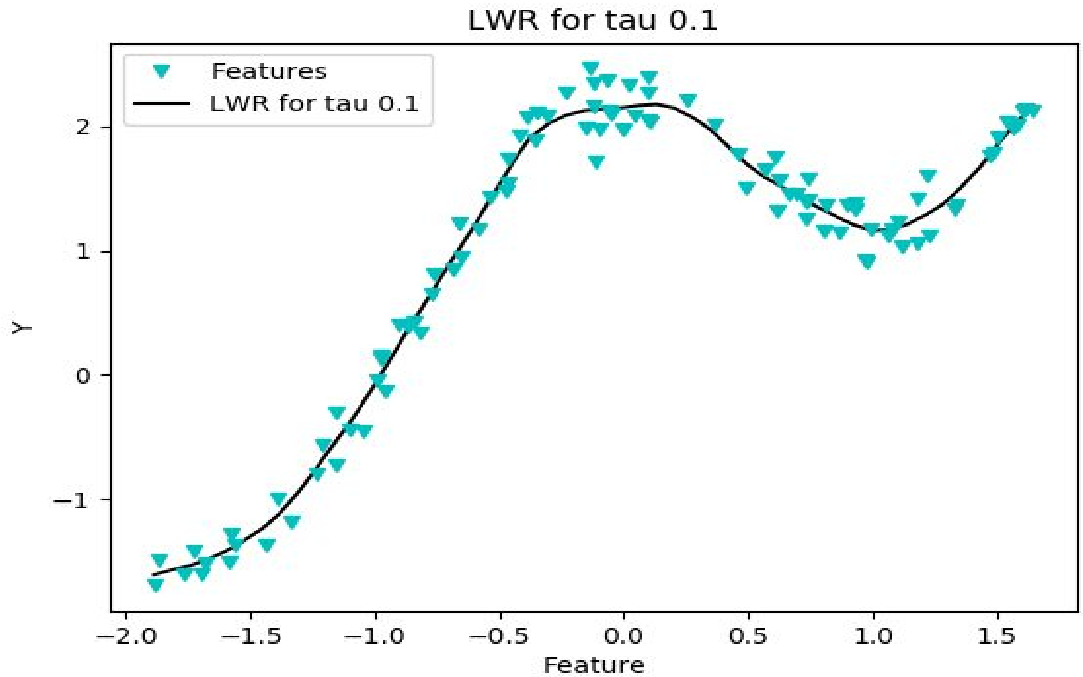
a.

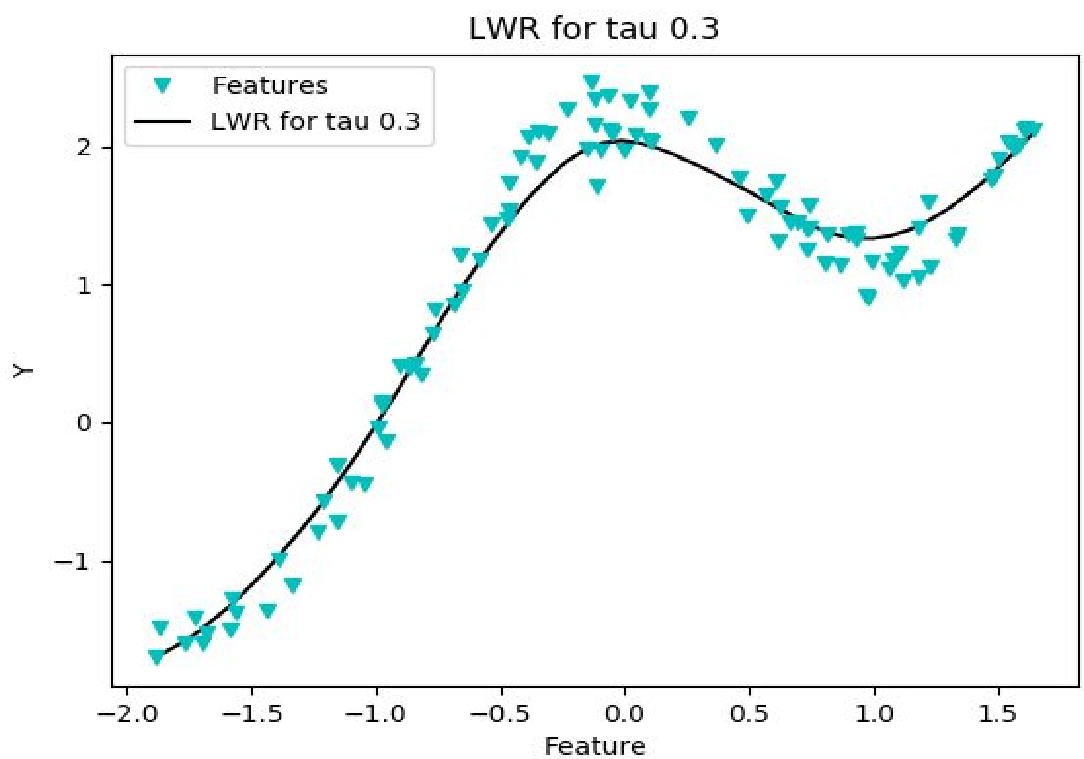


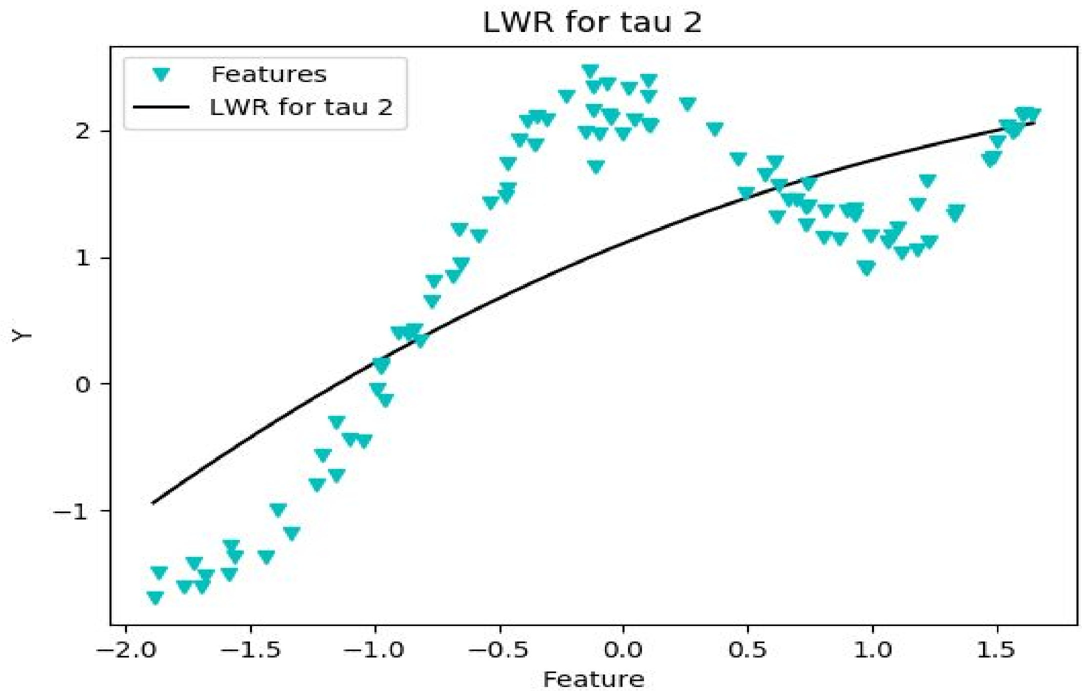

b.

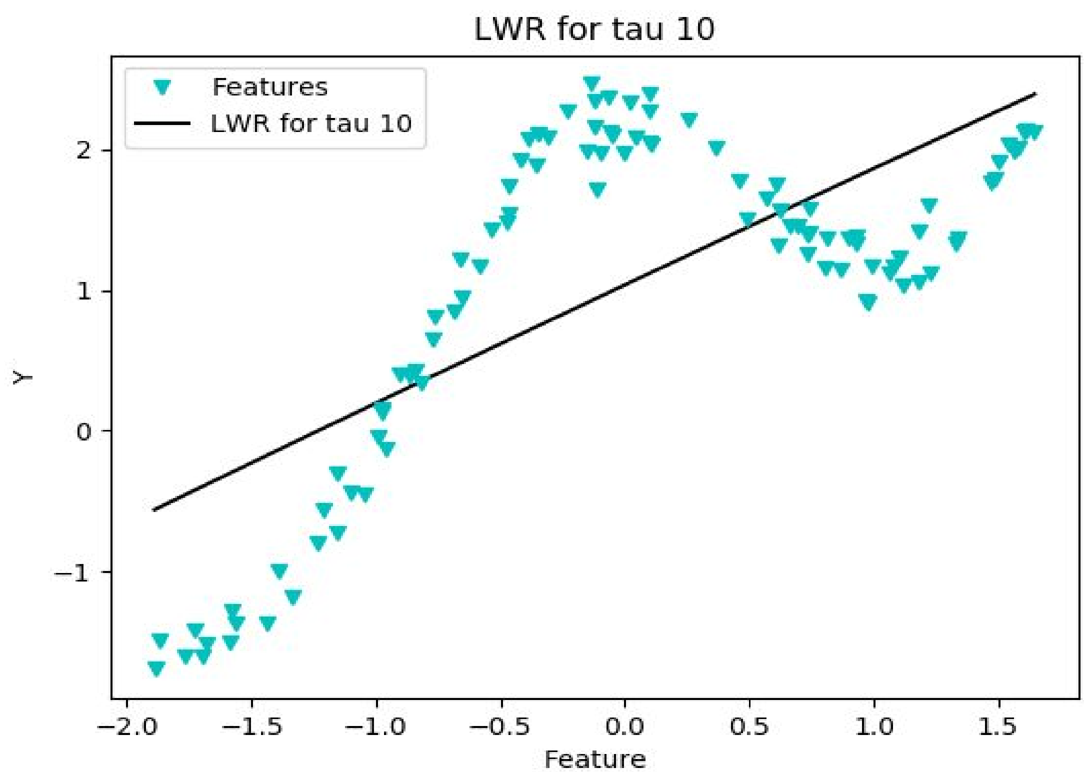


c.


LWR for tau 0.1
Features
LWR for tau 0.1
2
1
0
−1
Y
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Feature


LWR for tau 0.3
Features
LWR for tau 0.3
2
1
0
−1
Y
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Feature


LWR for tau 2
Features
LWR for tau 2
Y
Feature
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
−1
0
1
2


LWR for tau 10
Features
LWR for tau 10
Y
Feature
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
−1
0
1
2

Observations

When value of $\tau = 0.3$, the algorithm gives best fit. However, when the value of $\tau$ is decreased i.e. it approaches 0, the algorithm will try to overfit as in case of $\tau = 0.1$. While when value of $\tau$ is increased i.e. it approaches 1, the algorithm will underfit as in case of $\tau = 0.8,2,10$.

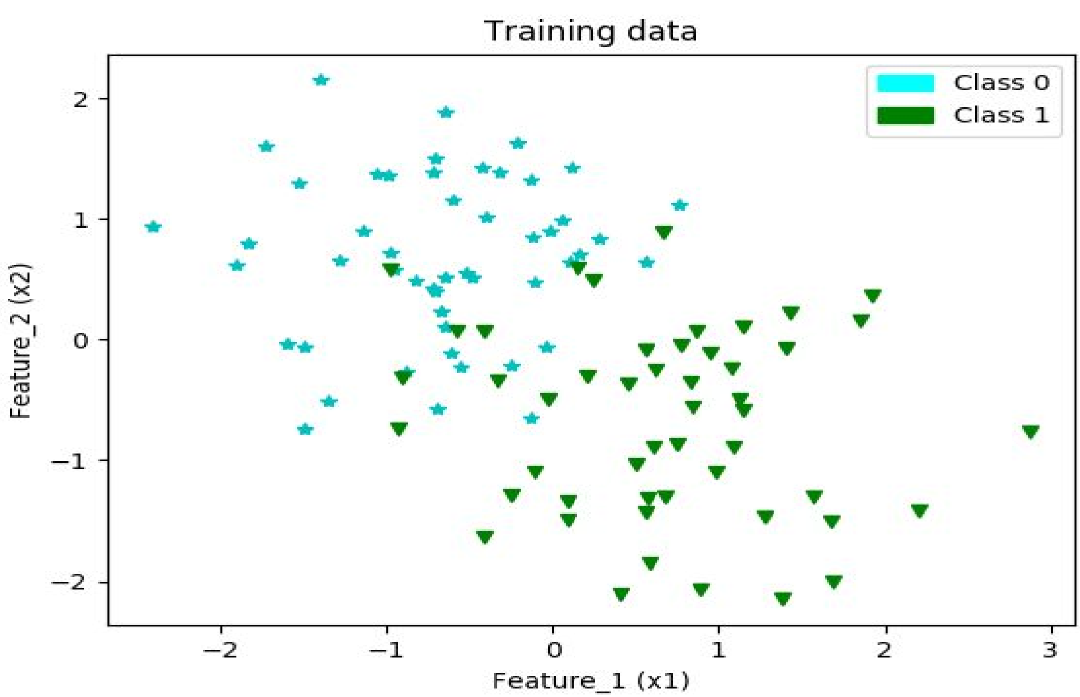
Q3. Logistic Regression

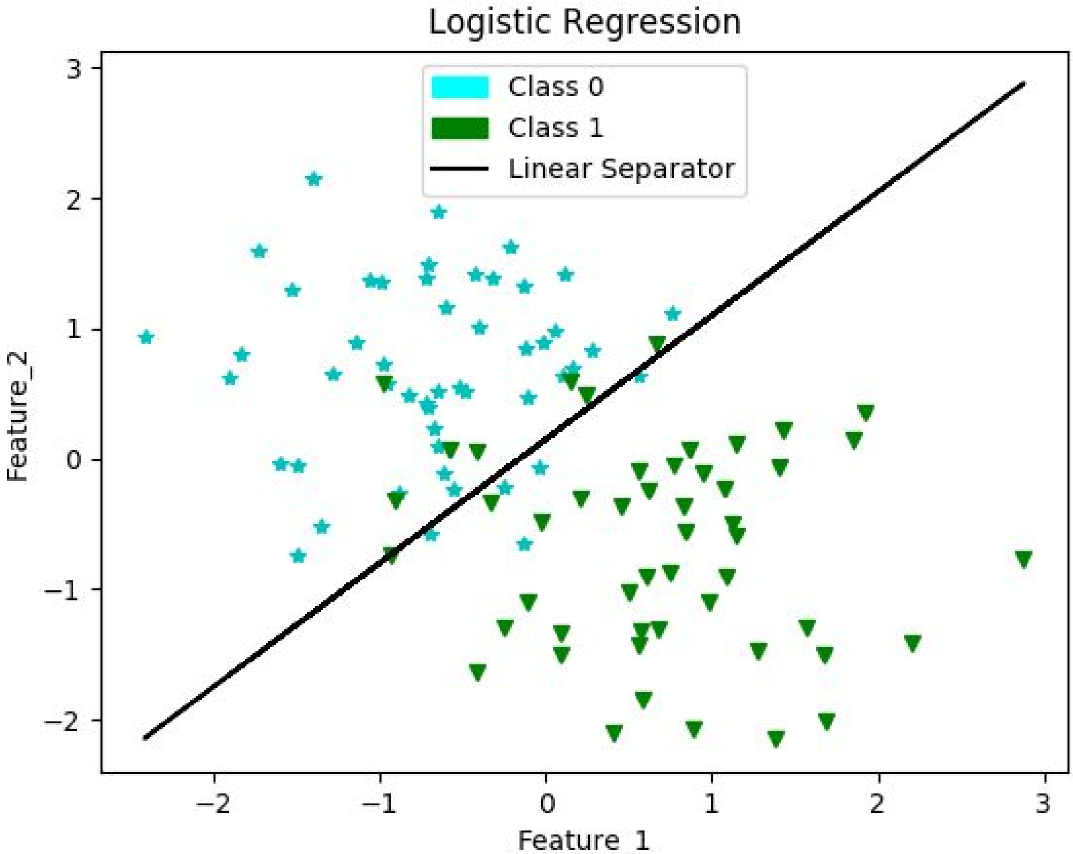
a. Parameters =

Theta0 = 0.40125316
Theta1 = 2.5885477
Theta2 = -2.72558849

Error = 22.83414498447239

b.



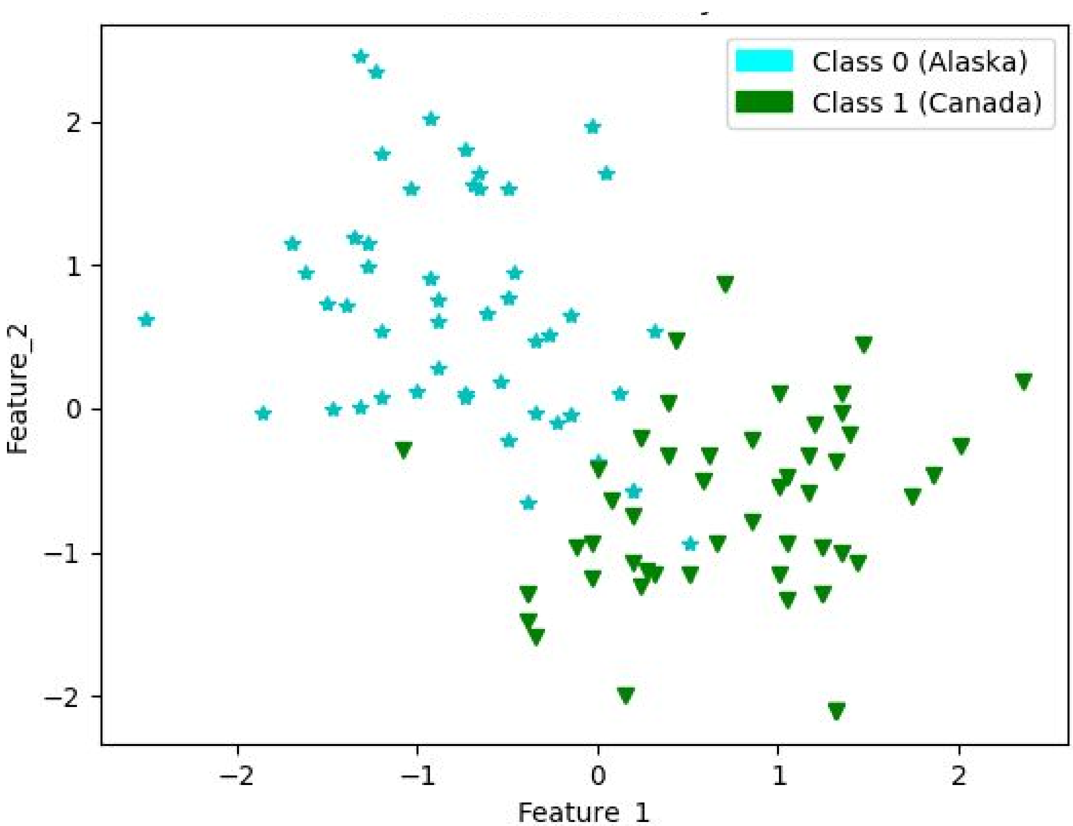



Q4. Gaussian Discriminant Analysis

a. Phi 0.5
Mu0 [-0.75529433 0.68509431]
Mu1 [ 0.75529433 -0.68509431]
Covariance = (Σ1 + Σ2)/2
[[ 0.42953048 -0.02247228]
[-0.02247228 0.53064579]]

b.

Plot of Features corresponding to Alaska and Canada



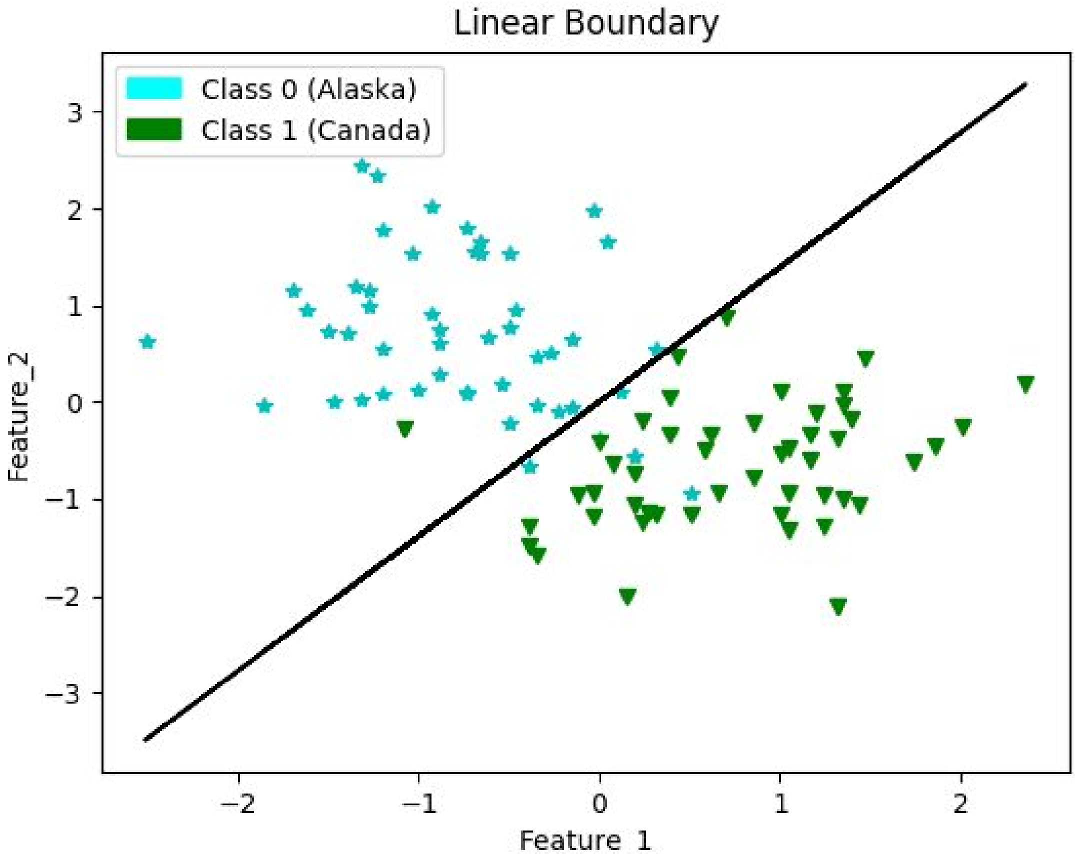

c. Equation for linear boundary i.e. when covariance matrix is same is

$$AX = B$$

where

$$A = 2 * ( \mu_0^T \Sigma^{-1} - \mu_1^T \Sigma^{-1})$$

$$B = (\mu_0^T \Sigma_0^{-1} \mu_0 - \mu_1^T \Sigma_1^{-1} \mu_1 - 2 * \log ( ((1/\phi) - 1) )$$




d. Phi 0.5

Mu0 [-0.75529433 0.68509431]

Mu1 [ 0.75529433 -0.68509431]

Covariance 0
[[ 0.38158978 -0.15486516]
[-0.15486516 0.64773717]]

Covariance 1
[[0.47747117 0.1099206 ]
[0.1099206 0.41355441]]

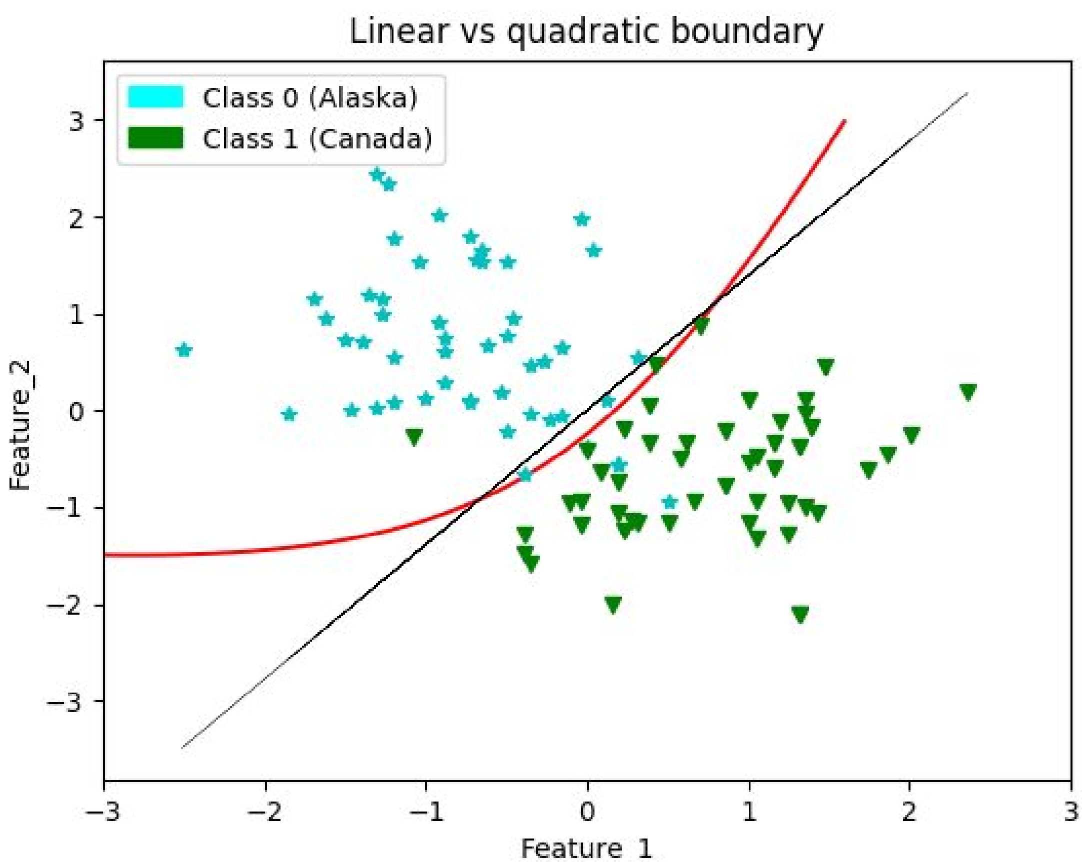
e. Equation of quadratic boundary when covariance matrix is different

$$X^T A X + B X + C = 0$$

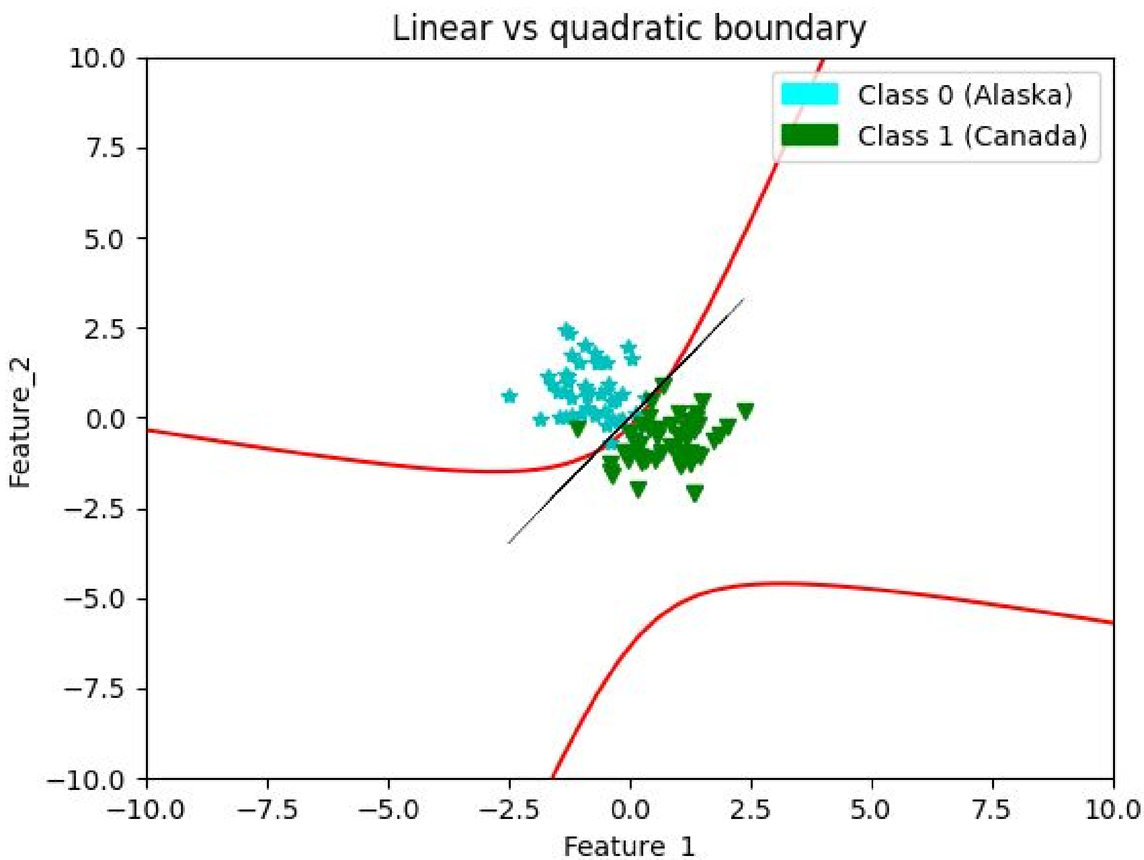
Where $A = \Sigma_0^{-1} - \Sigma_1^{-1}$

$B = -2 * ( \mu_0^T \Sigma_0^{-1} - \mu_1^T \Sigma_1^{-1})$

$C = (\mu_0^T \Sigma_0^{-1} \mu_0 - \mu_1^T \Sigma_1^{-1} \mu_1 - 2 * \log ( ((1/\phi) - 1) * (|\Sigma_1| / \Sigma_0) )$



e. Analysis of linear and quadratic boundary




As we can see from the plot in part d, we can observe that quadratic separator works better than linear separator. Because linear separator classifies some features of Alaska class incorrectly as features of Canada class which are classified correctly by quadratic separator.

Some of the references used for Python plotting:


1. https://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html
2. https://docs.scipy.org/doc/numpy/reference/generated/numpy.mgrid.html
3. https://matplotlib.org/users/pyplot_tutorial.html