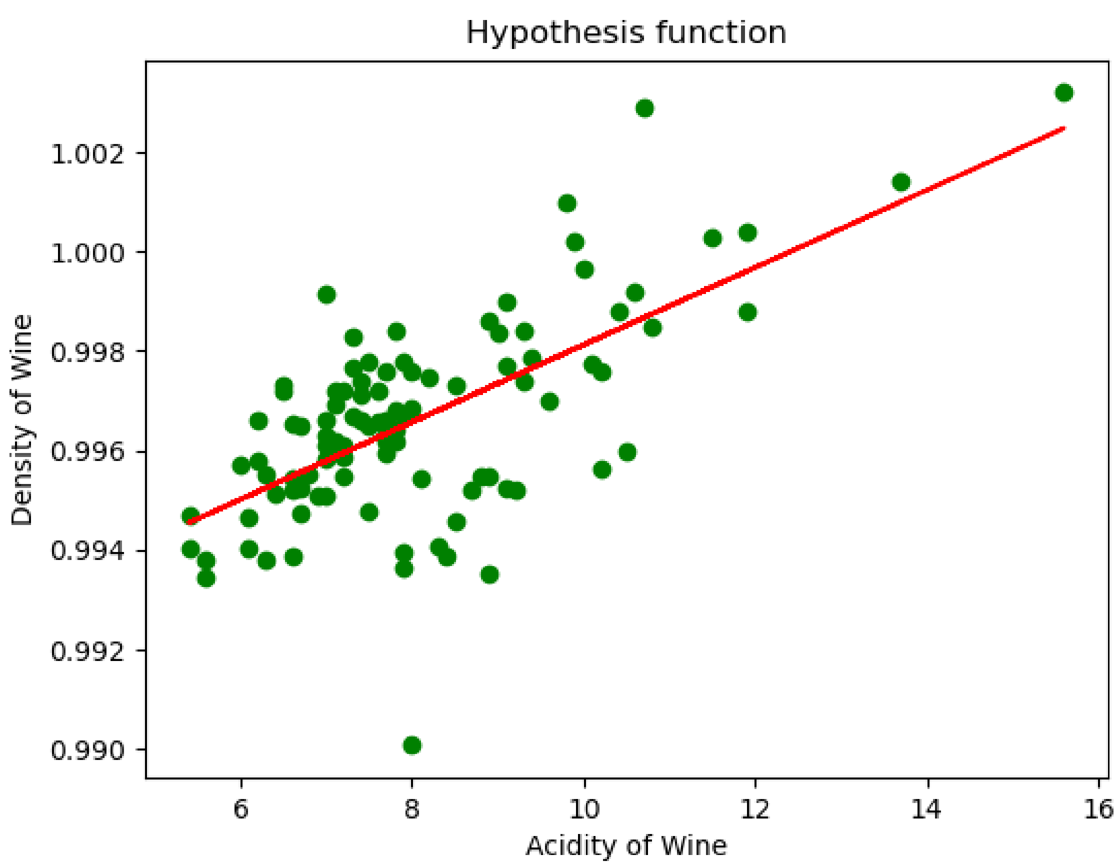
COL774 - ASSIGNMENT - 1

Gattu Karthik - 2019CS10348

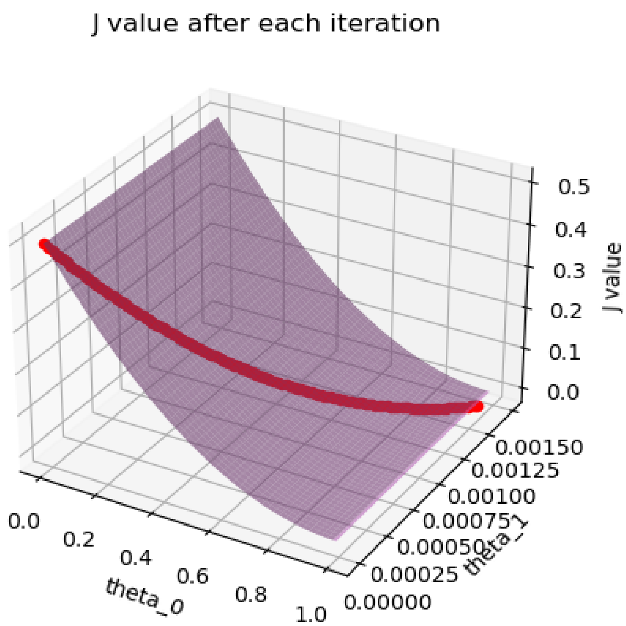
1. Linear Regression

a) Learning rate = 0.01
Stopping criteria = abs(J - J_prev) <= 1e-14
Converged theta = [[0.99661912]
[0.00134019]]
Final loss value = 1.194e-06
No of iterations to converge = 1376
Time taken to converge = 54ms

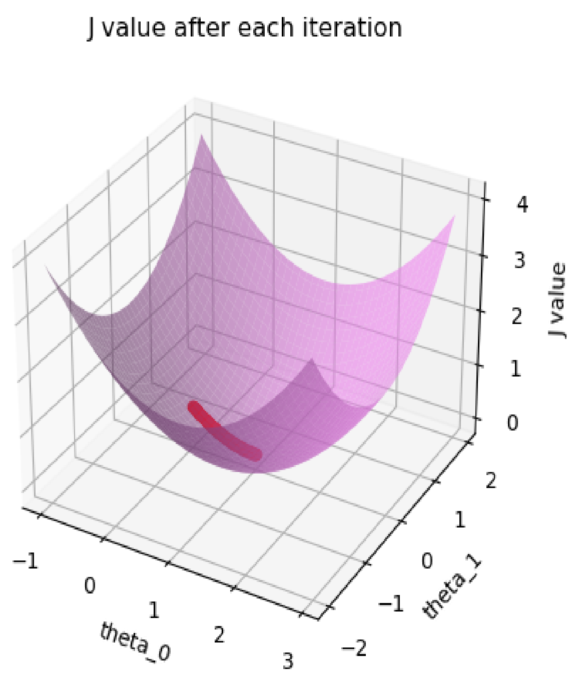
b) Plot of learned hypothesis on the given data



c) 3D mesh showing the error function -J(θ) after each iteration



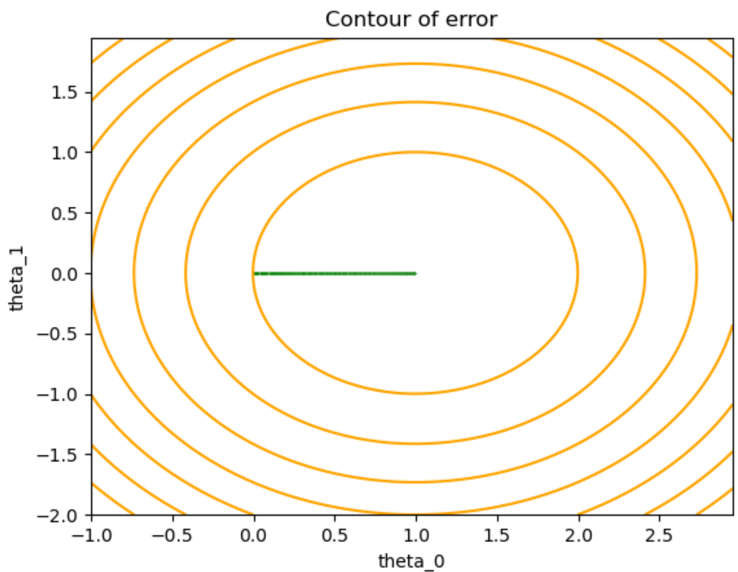
- The J(θ) value is decreasing with the increase in no of iterations



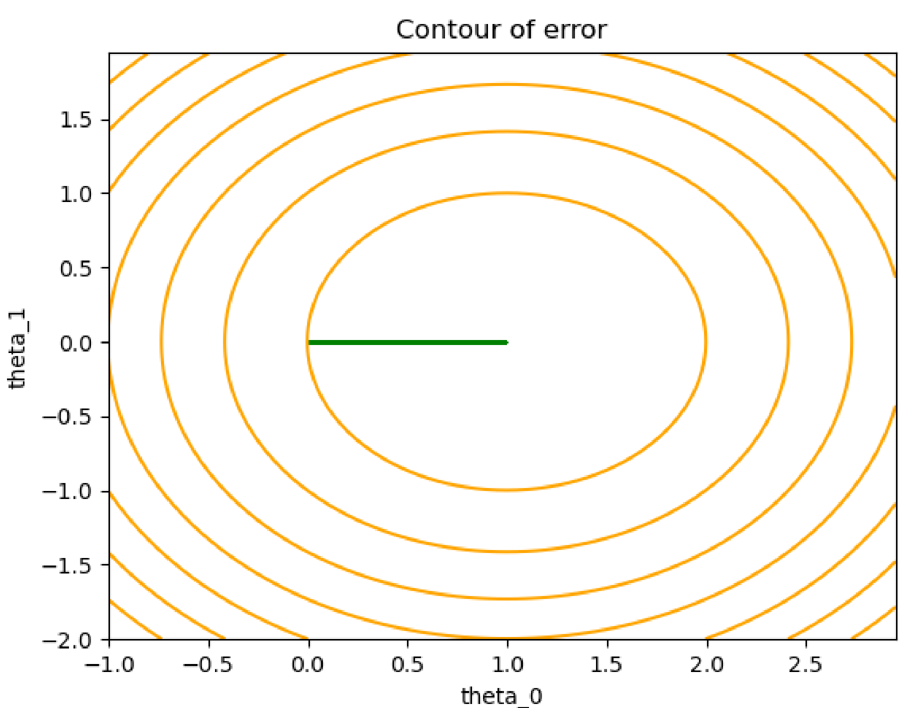
The big picture for observing the movement of J(θ)

d) contours of the cost function & value of $J(\theta)$ with iterations

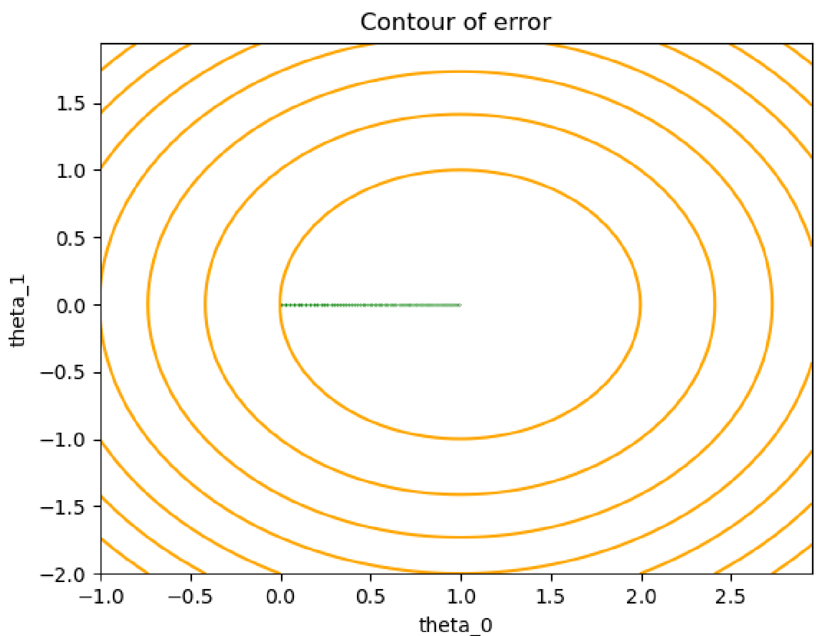
$\eta = 0.01$



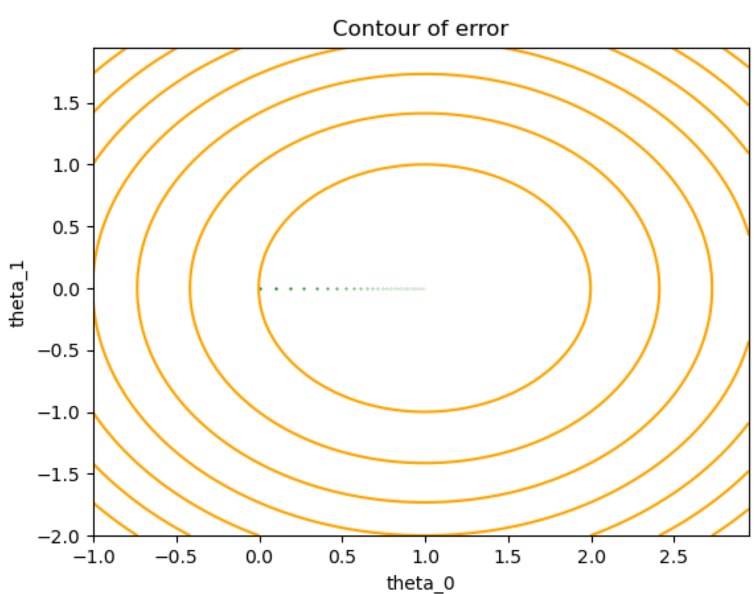
e) $\eta = 0.001$ & delta = 1e-12



η = 0.025 & delta = 1e-12



η = 0.1 & delta = 1e-12



With increase in η the time taken to converge is decreasing,but overly increasing η may have chance of skipping the minima value.

2. Sampling and Stochastic Gradient Descent

a) The output is stored in ‘sampled_data.csv’.

b) Convergence criteria =
abs(avg cost for b batches - avg cost for prev b batches)<delta

Batch size = r

Final converged parameters for different r values

- r = 1 , b = 10000 , delta = 1e-7
- r = 100 , b = 10000 , delta = 1e-9
- r = 10000, b = 1000 , delta = 1e-9
- r = 1000000, b = 1 , delta = 1e-7

- Converged theta values

r = 1

```
[[2.77006286]
 [0.93207003]
 [1.85751314]]
```

r = 100

```
[[2.75473946]
 [0.94045724]
 [1.87972566]]
```

r = 10000

```
[[2.75258036]
 [0.94137092]
```

[1.88096406]]

r = 1000000
[[2.71678164]
[0.94926096]
[1.87835031]]

c) The final parameters obtained for different r values are different but they are close to the original theta from which we generated the data.

- How much different are these from original hypothesis:
  RMSE from the given hypothesis
  r =1 : 0.1610259548936549

  r =100 : 0.16141456936714718

  r=10000 : 0.16209400800261925

  r=1000000 : 0.1803568099613533

  This RMSE value depends on delta we choose but here but on decreasing delta the theta will come close to the given hypothesis

- Speed of convergence
  r=1 : 1 epoch

  r = 100 : 8 epochs

r = 10000 : 381 epochs

r = 1000000 : 15705 epochs

If we increase the batch size the no of epochs are increasing and converging fast.

- Test error value

  Error with original hypothesis = 0.9829469215

  r = 1 : 2.2693726956049933

  r = 100 : 1.9245807442343918

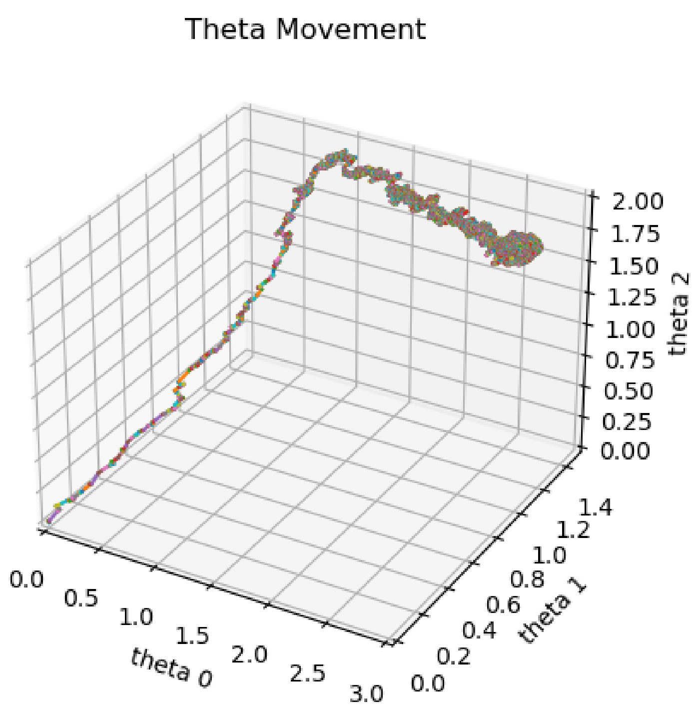
  r = 10000 : 1.9045722259647722

  r = 1000000 : 1.8982679133345504

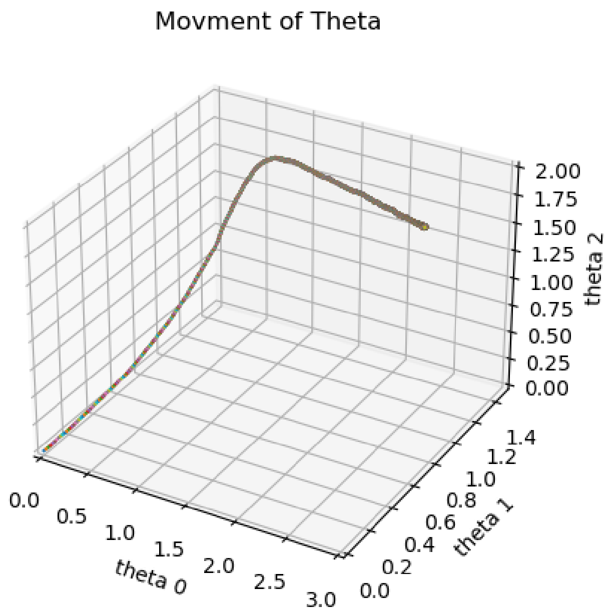
  This depends on the delta value we choose for convergence,with increase in r the value is getting close to the error with the given hypothesis

d) plot of the movement of θ as the parameters are updated (until convergence) for varying batch sizes

r = 1

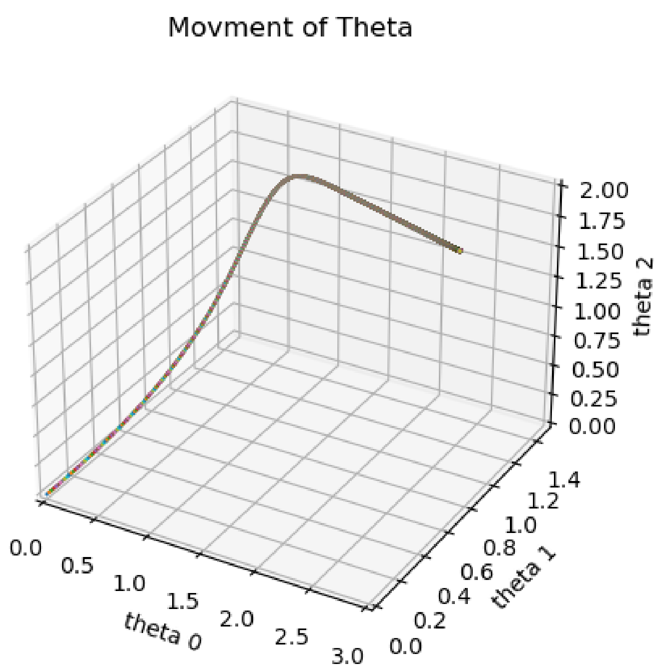


r = 100



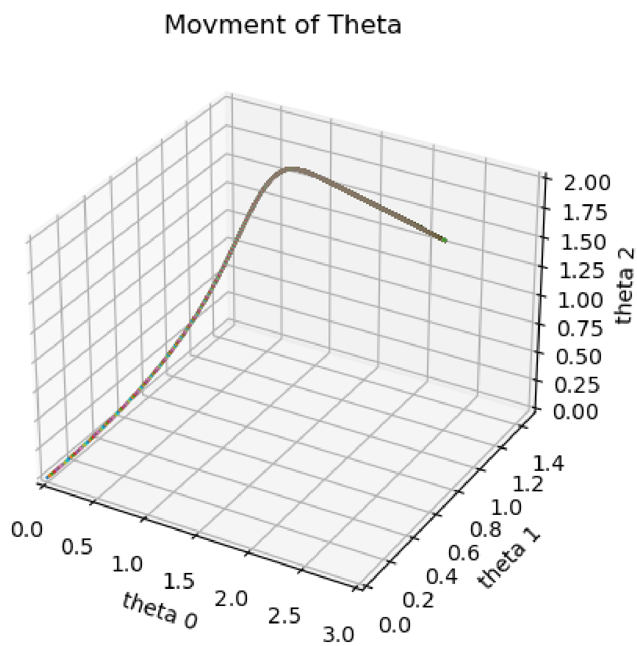
Movment of Theta
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
theta 2
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
theta 1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
theta 0

r = 10000



Movment of Theta
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
theta 2
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
theta 1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
theta 0

r = 1000000



Observations :
For r =1 the movement of theta is not smooth but for large values of r the movement of the theta is smooth and the smoothness is increasing with increase in r. But the overall movement of theta when r=1 is same as the other values of r.This behavior is because when r is small we are considering only a very small portion of the training data for updating the parameters hence we will miss out the majority of the information of the dataset , but in case of larger r values we are considering a significant amount of data while updating the parameters.

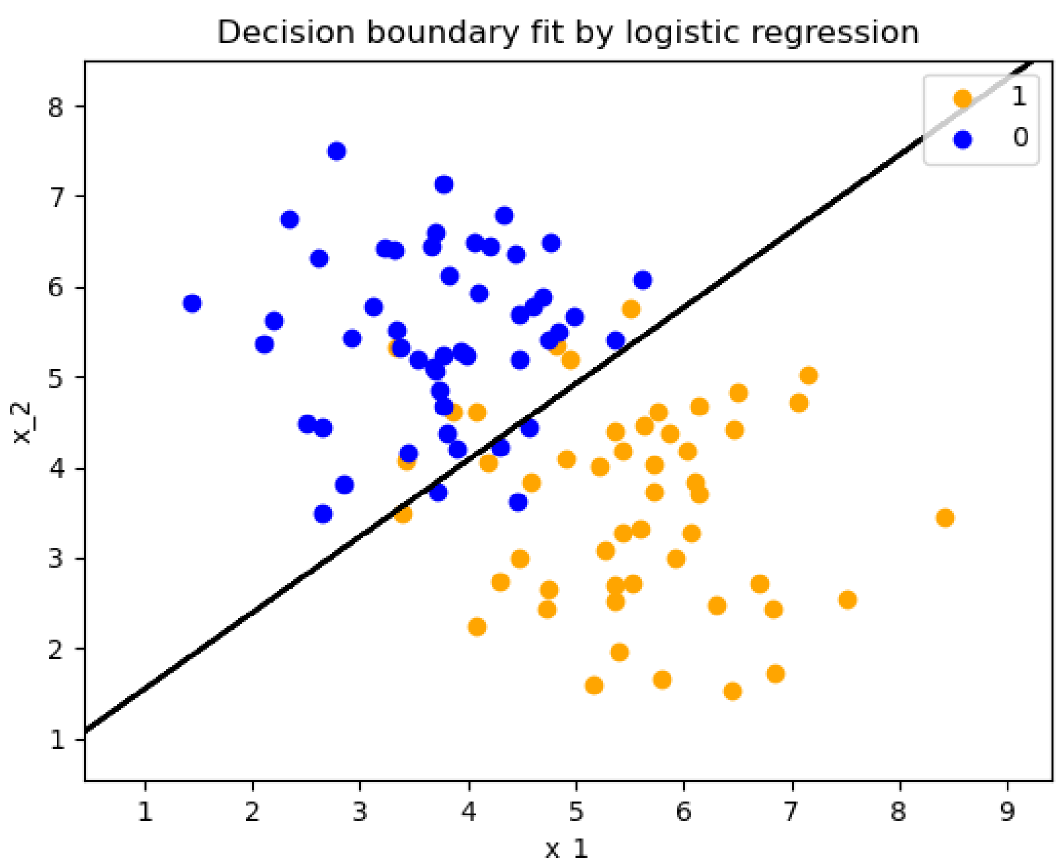
3. Logistic Regression

a) Converged Theta vector: [[0.40125316]
[2.5885477]
[-2.72558849]]

Total no of iterations to converge = 8

Final loss after convergence = -22.834144984472392

b) Plot of Training data and Decision boundary



The black line is the linear decision boundary that is learnt.

4.Gaussian Discriminant Analysis

a) Assuming that both the classes have the same covariance matrix i.e. $\Sigma 0 = \Sigma 1 = \Sigma$.

- ϕ = 0.5

- μ_0 = [[-0.75529433]
  [0.68509431]]

- μ_1 = [[0.75529433]
  [-0.68509431]]

- $\boldsymbol{\Sigma}$ = [[0.42953048 -0.02247228]
  [-0.02247228 0.53064579]]

c) Equation of decision boundary ($\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1$)

$$-[(\mu_1 - \mu_0)^T \boldsymbol{\Sigma}^{-1} x - \tfrac{1}{2} * (\mu_1^T \boldsymbol{\Sigma}^{-1} \mu_1 - \mu_0^T \boldsymbol{\Sigma}^{-1} \mu_0)] + \log((1-\phi)/\phi) = 0$$

d) $\boldsymbol{\Sigma}_0 \neq \boldsymbol{\Sigma}_1$

- ϕ = 0.5

- μ_0 = [[-0.75529433]
  [0.68509431]]

- μ_1 = [[0.75529433]
  [-0.68509431]]

- $\boldsymbol{\Sigma}$ = [[0.42953048 -0.02247228]
  [-0.02247228 0.53064579]]

- $\boldsymbol{\Sigma}_0$ = [[0.38158978 -0.15486516]
  [-0.15486516 0.64773717]]

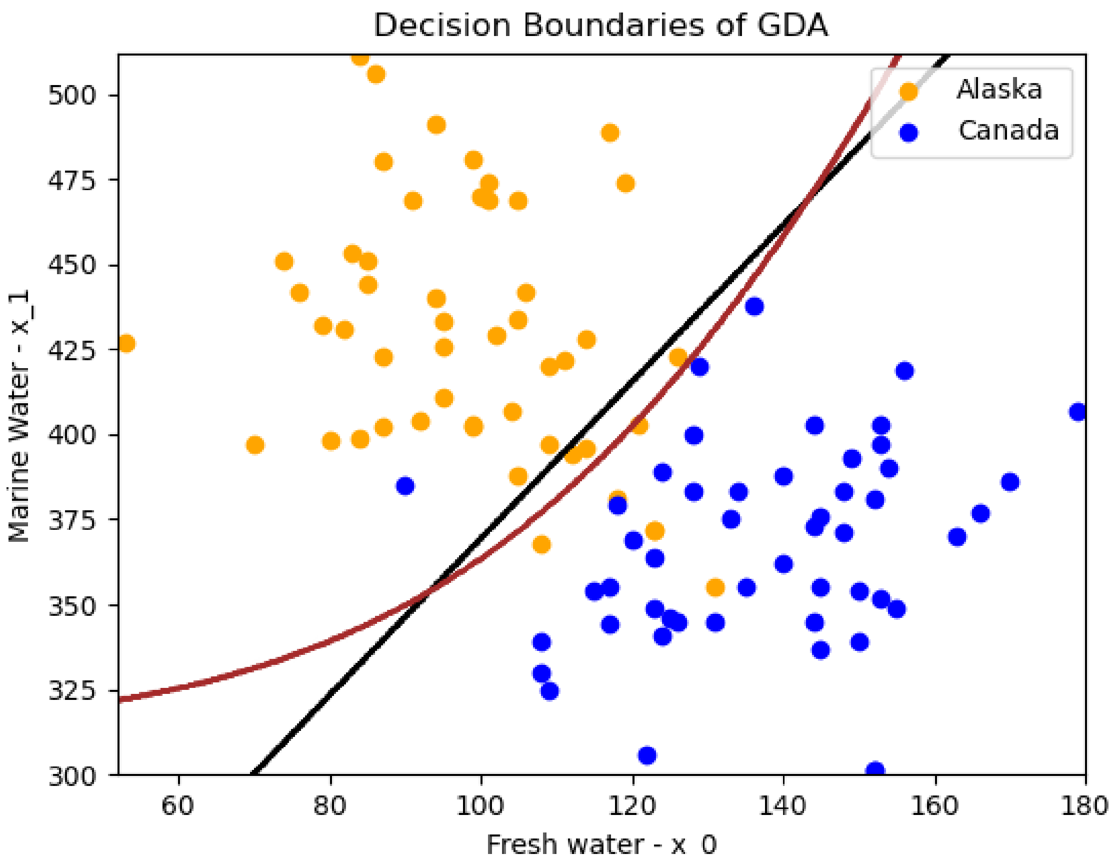
- $\boldsymbol{\Sigma}_1$ = [[0.47747117 0.1099206]
  [0.1099206 0.41355441]]

e) Equation of decision boundary ($\boldsymbol{\Sigma}_0 \neq \boldsymbol{\Sigma}_1$)

$\frac{1}{2}$ *[x^T ($\boldsymbol{\Sigma}^{-}{}_{1}{}^{1}$ - $\boldsymbol{\Sigma}_0^{-1}$) x - 2*($\mu_1$ $\boldsymbol{\Sigma}^{-}{}_{1}{}^{1}$ - μ_0 $\boldsymbol{\Sigma}_0^{-1}$)^T x - ($\mu_1^T$ $\boldsymbol{\Sigma}_1^{-1}$ μ_1 - μ_0^T $\boldsymbol{\Sigma}_0^{-1}$ μ_0)] + c =0

c = log(((1-ϕ)*($|\boldsymbol{\Sigma}_1|$)$^{\frac{1}{2}}$) / ϕ * ($|\boldsymbol{\Sigma}_0|$)$^{\frac{1}{2}}$)

b,c,e) Plot of Training Data, Linear Decision boundary, Quadratic Decision boundary:



f) Quadratic decision boundary is separating the points more accurately than the linear decision boundary. This is because linear decision boundary is assuming the covariance matrices are equal but in quadratic decision boundary it is not making any such assumptions.Hence it is able to separate more accurately than the other.