ML Mini-Project Report

1 Linear Regression

(a) **Final Parameters :-**
Learning rate = 0.01

Stopping Criteria = When the difference between the two alternative J(θ) becomes less than 10^{-12}.

Number of Iterations, to finally get converged = 1146.

Theta Obtained:

$$\theta_0 = 0.99661018$$
$$\theta_1 = 0.00134018$$

(b) **Plot :-**

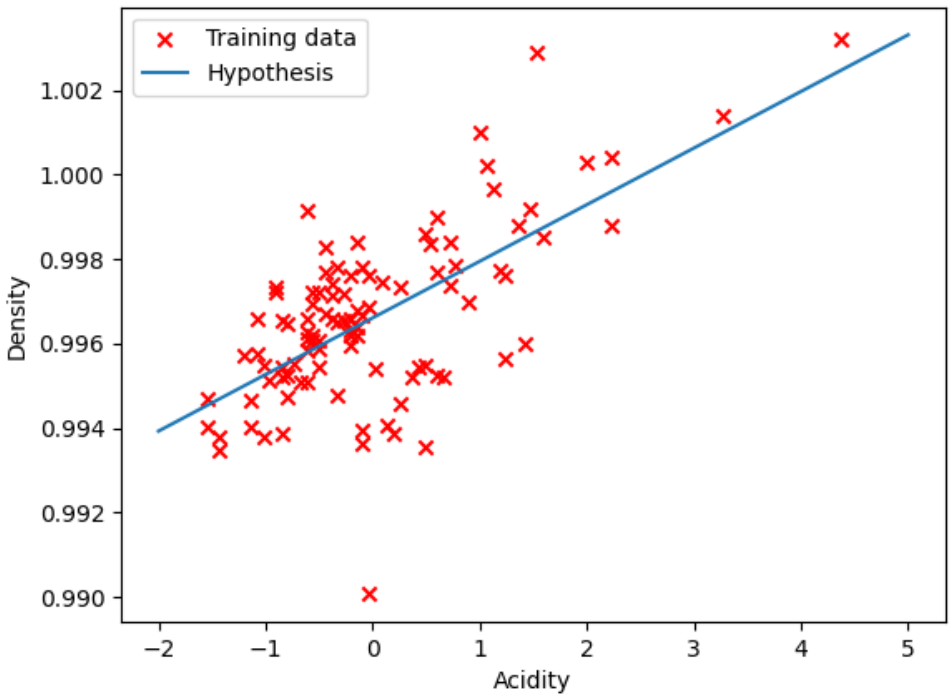

Figure 1: Training data with Hypothesis Line

(c) **3D Mesh for J(θ) :-**

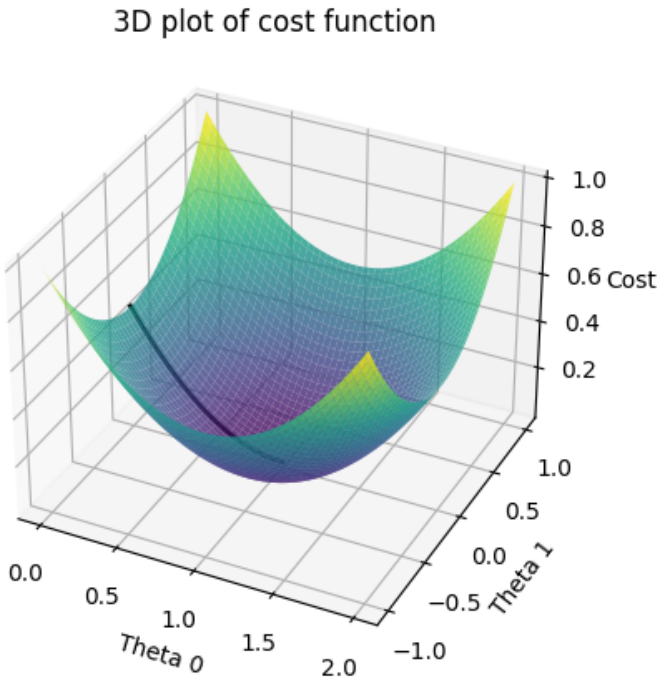

Figure 2: 3D Mesh for error function

(d) **Contour for Learning rate taken by me :-**

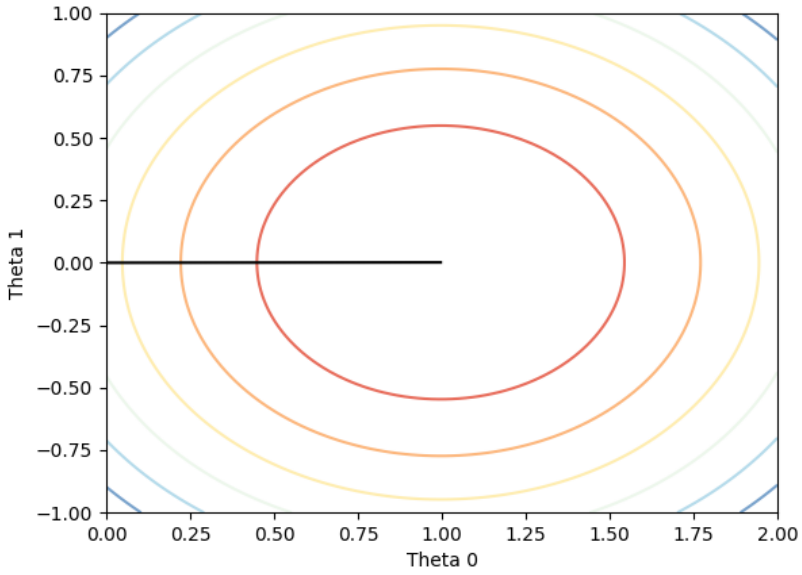

Figure 3: Contour for $\eta = 0.01$

(e) **Contour for different Learning rate :-**

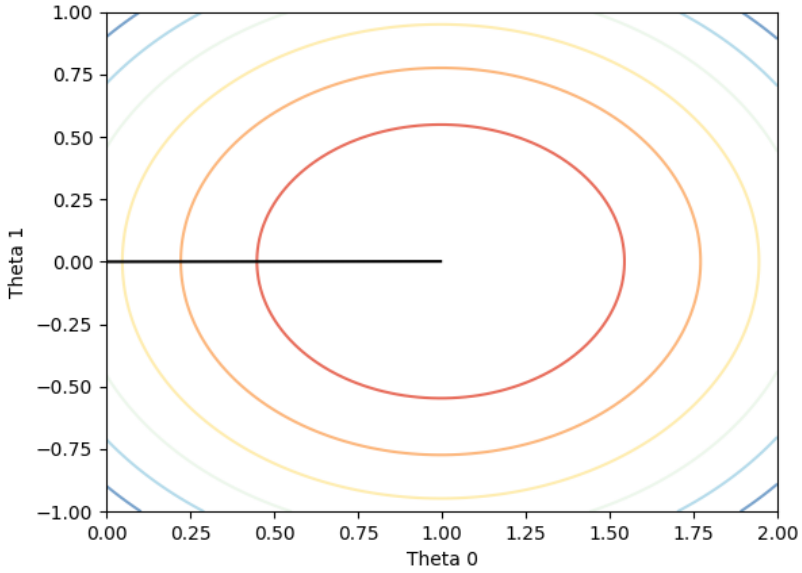

Figure 4: Contour for $\eta = 0.001$

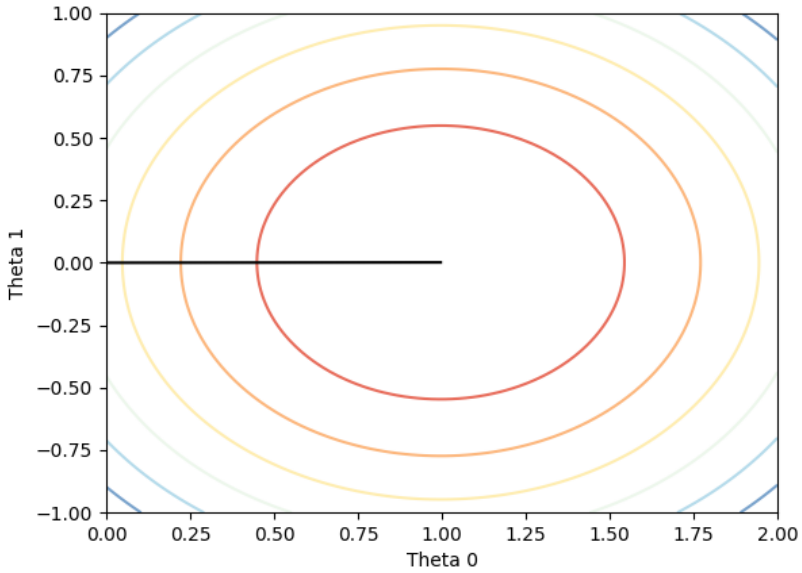

Figure 5: Contour for $\eta = 0.025$

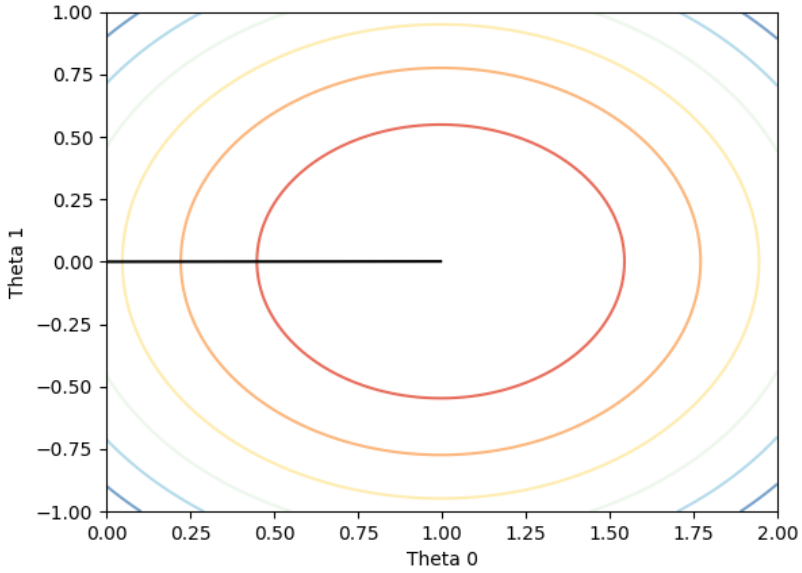

Figure 6: Contour for $\eta = 0.1$

Observations :-
Time taken by the model to get converge based on the stopping criteria, increases as we are decreasing the learning rate. As, θ updating steps depends directly on the value of learning rate(η).

$\theta = \theta$ - η*∇J(θ)

So, J(θ) converges very late as we decreases our learning rate.

Thus, T(η=0.001)>T(η=0.025)>T(η=0.1)

2 Stochastic Gradient Descent

(a) **Sampling**
Done sampling of 1 million data points based on the uni-variate Gaussian distribution given to us in the problem set.

(b) **Final Parameters :-**
Learning rate =0.001

Stopping Criteria = Captured θ after every 1000^{th} iterations and assumed that the model got converged when the difference between those alternative θ became less than 10^{-2}.

Table 1

Batch size	θ_0	θ_1	θ_2
1	3.00597606	1.101713657	1.99098366
100	2.98439702	1.003926570	1.99866407
10000	2.97802547	1.004772850	1.99833500
1000000	2.97805634	1.004871520	1.99820115

(c) **Observation:-**
No, the algorithms for different batch sizes doesn't converge to the same parameters, but approxi-

mately they are same and very close to the original parameters that we considered while generating the training data, i.e. $\theta_0 = 3$, $\theta_1 = 1$ and $\theta_2 = 2$.

Table 2

Batch size	Iterations	Time Taken(sec)
1	139000	0.972869873046875
100	18000	0.12930035591125488
10000	17000	4.944814682006836
1000000	17000	246.057433128

Relative speed of convergence:-

- The batch size of 1 took less time to converge despite taking large number of iterations.
- The batch size of 100 took less time as well as less number of iterations than previous batch.
- The batch size of 10000 took relatively more time than 1 and 100 batch size, but approximately same number of iterations as 100 batch size.
- The batch size of 1000000 took longest time among all, as whole 1 million examples are being operated in each iteration.

Error for test data of size 10,000:-

Table 3

Batch size	Error
1	1.0025881695825698
100	0.9838918111222789
10000	0.9844034448090304
1000000	0.9844826617998529

Comments:-
The error on this test data set with respect to our original hypothesis comes out to be around 0.9829469215000003. The difference in error for all batch sizes and original hypothesis is not that much.
The error for batch size 10000 and 1000000 is equal as the test data is of size 10000 only and in both the algorithms, the operations are performed on all 10000 data points in each iteration. That's why they both will take same time and same number of iterations to get converge and hence, will produce the same error.
The error for original hypothesis is minimum of all as the test data was produced using the same parameters as used in original hypothesis.

(d) **3D plot for movement of theta:-**

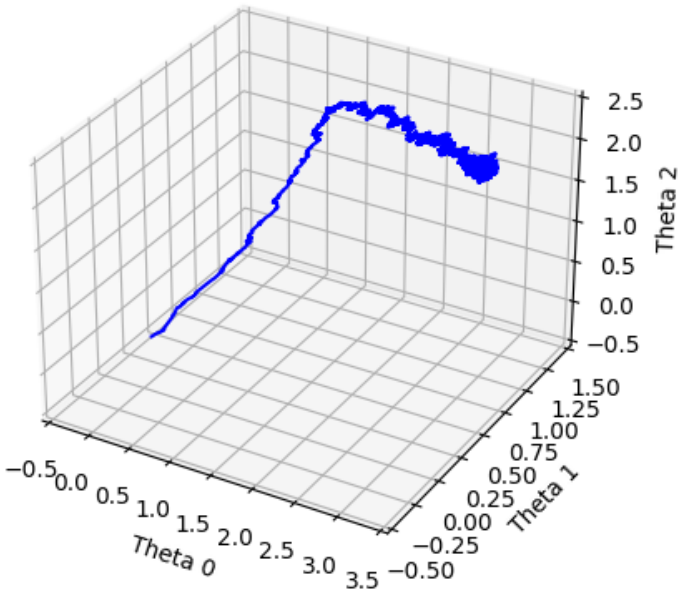

Figure 7: Movement of θ for Batch size = 1

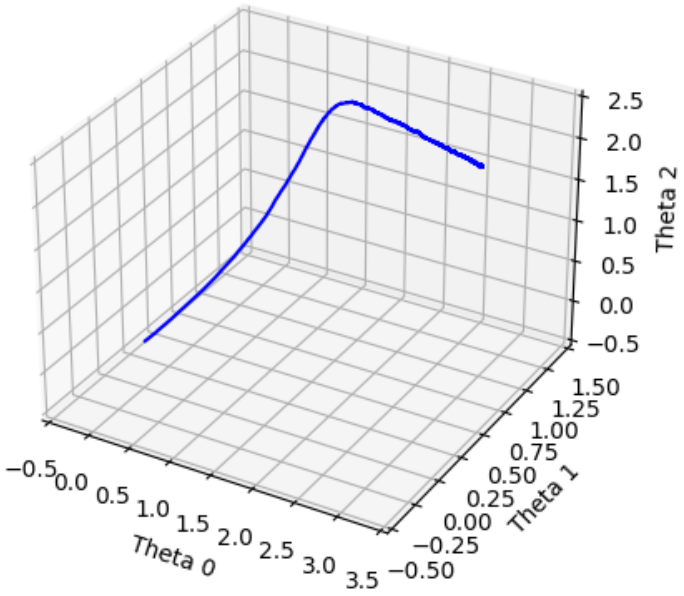

Figure 8: Movement of θ for Batch size = 100

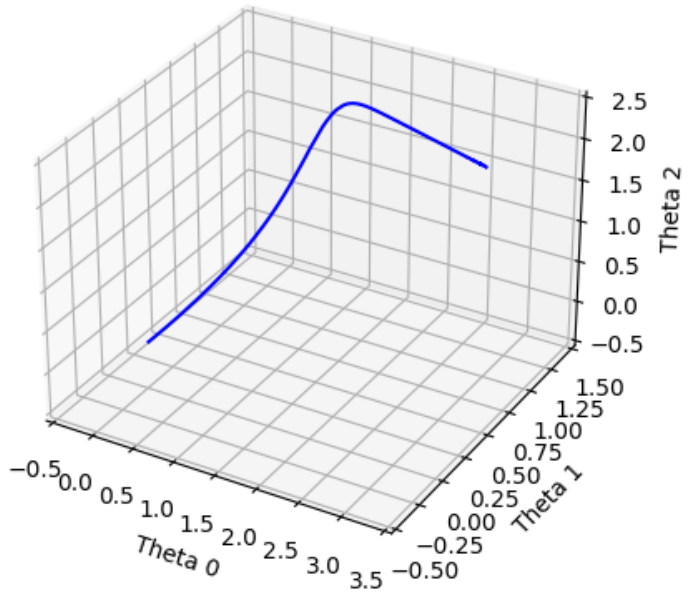

Figure 9: Movement of θ for Batch size = 10000

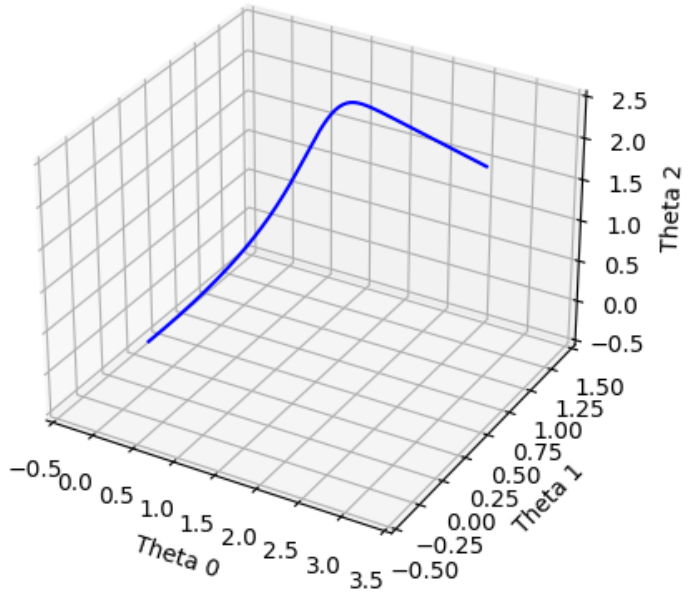

Figure 10: Movement of θ for Batch size = 1000000

Comments:-
Plotting is done for the θs captured after every 1000^{th} iterations.

- In case of batch size 1, movement of θ is very zig-zag. As in each iteration only one data point is being operated.
- In case of batch size 100, movement of θ is little smoother than previous one.
- In case of batch size 10000, movement of θ is even more smoother than previous two.
- But, In the case of batch size 1000000, the graph is smoothest of all, as operations are being performed on 1 million data points in each iteration.

All the graphs converges approximately to the same point. But, the time differs for convergence, which can be seen in the table 2 above.

3 Logistic Regression

(a) **Equations for Logistic regression model using Newton's method:-**

$$\theta = \theta - H^{-1}\nabla_\theta L(\theta)$$

$$\nabla_\theta L(\theta) = \sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

$$H^{-1} = \sum_{i=1}^{m} -x_j^{(i)} h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))x_k^{(i)}$$

Final Parameters:-

$$\theta_0 = 0.40125316$$

$$\theta_1 = 2.5885477$$

$$\theta_2 = -2.72558849$$

(b) **Plot:-**

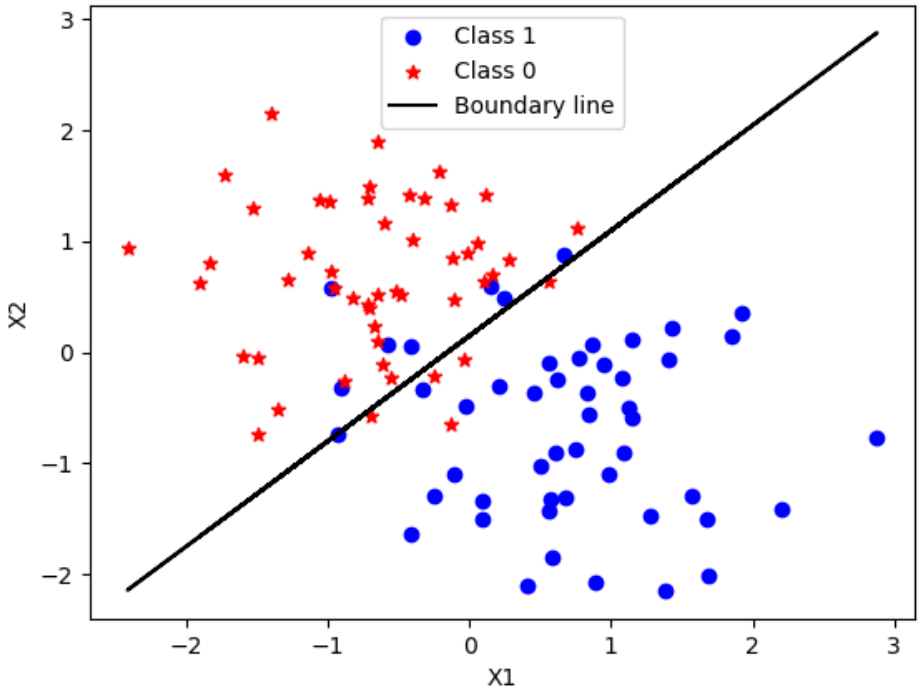

Figure 11: Training Data along with Decision Boundary

4 Gaussian Discriminant Analysis

(a) **Final Parameters for same Covariance matrix:-**

$$\phi = 0.5$$

$$\mu_0 = \begin{bmatrix} -0.75529433 \\ 0.68509431 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 0.75529433 \\ -0.68509431 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 0.42953048 & -0.02247228 \\ -0.02247228 & 0.53064579 \end{bmatrix}$$

(b) **Plot:-**

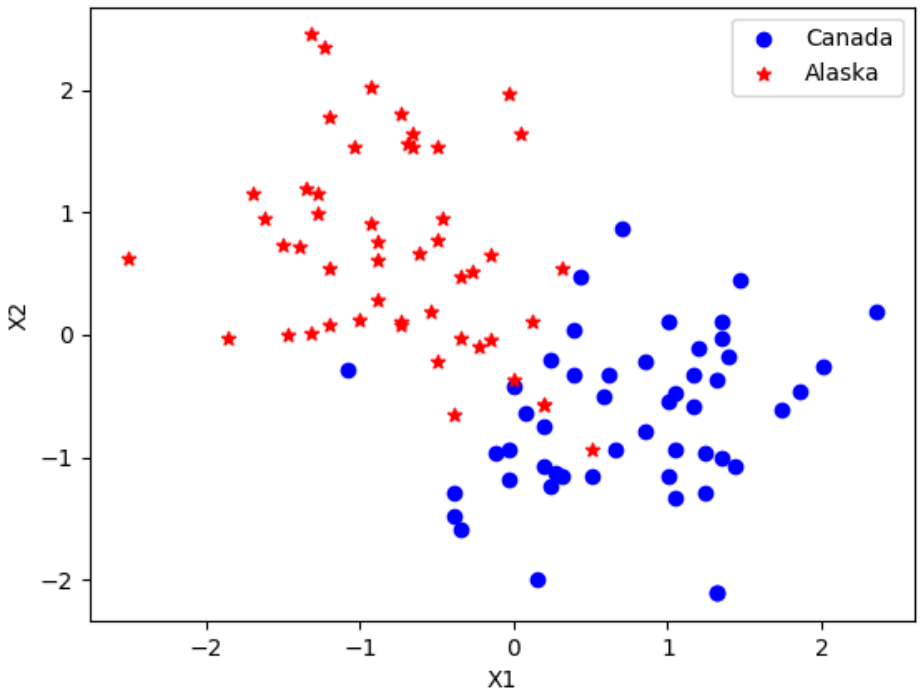

Figure 12: Training Data

(c) **Equation of Linear Decision Boundary for same Covariance matrix:-**

$$\log(\frac{\phi}{1-\phi}) + \frac{1}{2}(\mu_0^T\Sigma^{-1}\mu_0 - \mu_1^T\Sigma^{-1}\mu_1) + x^T(\Sigma^{-1}\mu_1 - \Sigma^{-1}\mu_0) = 0$$

$$(constant) + A_{00}x_1 + A_{10}x_2 = 0$$

$$where, A = \Sigma^{-1}\mu_1 - \Sigma^{-1}\mu_0$$

$$constant = \log(\frac{\phi}{1-\phi}) + \frac{1}{2}(\mu_0^T\Sigma^{-1}\mu_0 - \mu_1^T\Sigma^{-1}\mu_1)$$

Plot:-

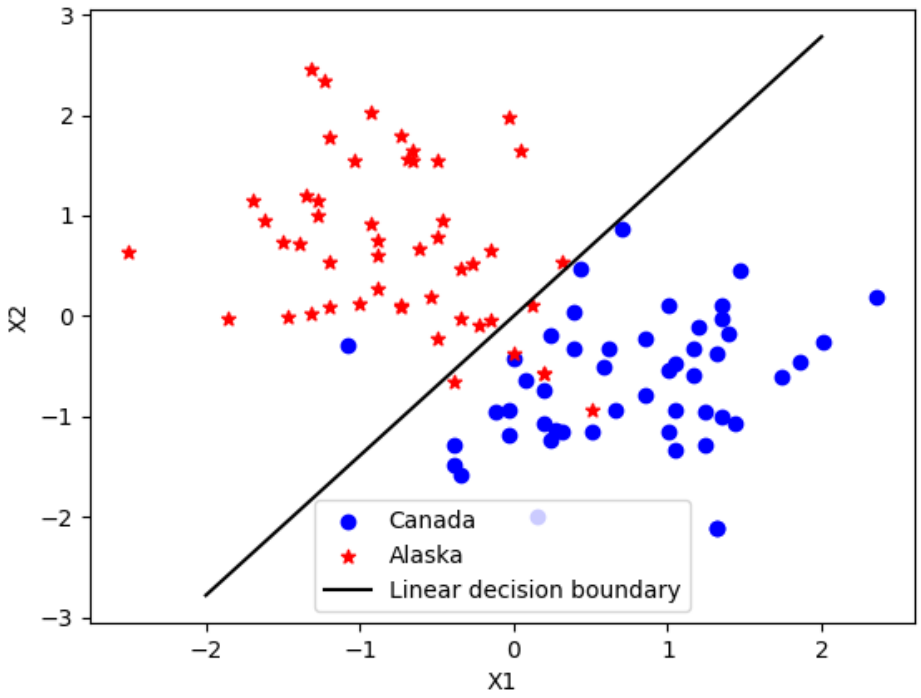

Figure 13: Training Data along with Linear Decision Boundary

(d) **Final Parameters for different Covariance matrix:-**

$$\phi = 0.5$$

$$\mu_0 = \begin{bmatrix} -0.75529433 \\ 0.68509431 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 0.75529433 \\ -0.68509431 \end{bmatrix}$$

$$\Sigma_0 = \begin{bmatrix} 0.38158978 & -0.15486516 \\ -0.15486516 & 0.64773717 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 0.47747117 & 0.1099206 \\ 0.1099206 & 0.41355441 \end{bmatrix}$$

(e) **Equation of Quadratic Decision Boundary for different Covariance matrix:-**

$$\log(\frac{\phi}{1-\phi}) + \frac{1}{2}\log(\frac{|\Sigma_0|}{|\Sigma_1|}) + \frac{1}{2}(\mu_0^T\Sigma_0^{-1}\mu_0 - \mu_1^T\Sigma_1^{-1}\mu_1) + x^T(\Sigma_1^{-1}\mu_1 - \Sigma_0^{-1}\mu_0) + \frac{1}{2}x^T(\Sigma_0^{-1} - \Sigma_1^{-1})x = 0$$

$$(constant) + A_{00}x_1 + A_{10}x_2 + M_{00}x_1^2 + M_{11}x_1^2 + (M_{01} + M_{10})x_1x_2 = 0$$

$$where, A = \Sigma^{-1}\mu_1 - \Sigma^{-1}\mu_0$$

$$M = \frac{1}{2}(\Sigma_0^{-1} - \Sigma_1^{-1})$$

$$constant = \log(\frac{\phi}{1-\phi}) + \frac{1}{2}\log(\frac{|\Sigma_0|}{|\Sigma_1|}) + \frac{1}{2}(\mu_0^T\Sigma_0^{-1}\mu_0 - \mu_1^T\Sigma_1^{-1}\mu_1)$$

Plot:-

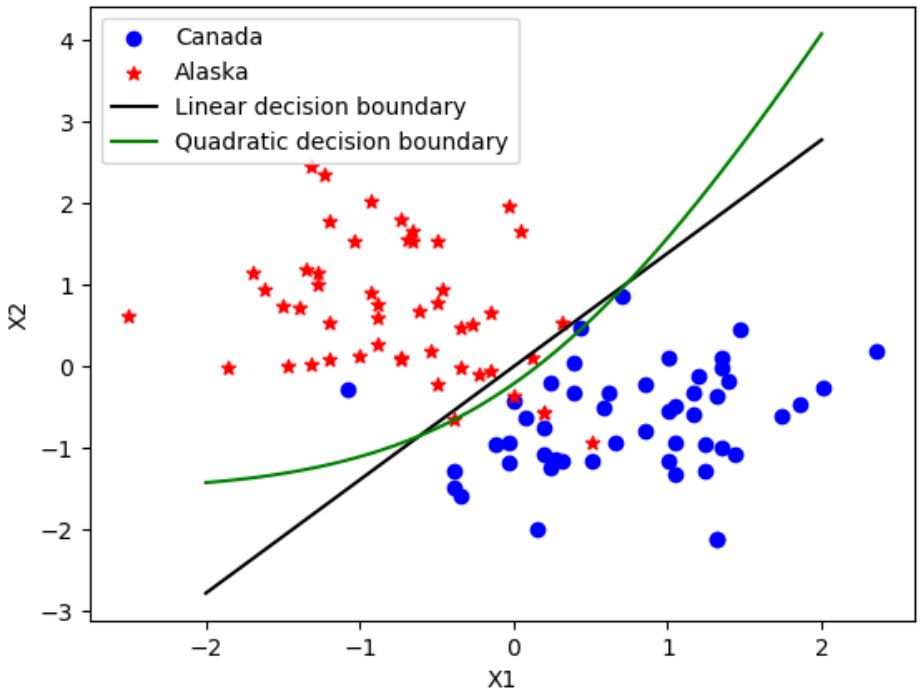

Figure 14: Training Data with Linear and Quadratic Decision Boundary

(f) **Observation:-**
By analyzing the plot of both linear and quadratic decision boundary, we saw that Quadratic decision boundary is making more accurate predictions about classification of the data points. By analyzing the plot above in Figure 14, we can tell that there are some points of Class Alaska that the linear decision boundary predicted as Class Canada, whereas quadratic boundary is still predicting them correctly, making less errors than linear boundary.