Assignment1 Report

Priyadarshi Hitesh

10 February 2020

1 Linear Regression

In this, i have implemented least square linear regression to predict density of wine based on its acidity.

1.1 Cost Function

$$J(\theta) = 1/2m \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))^2$$

1.2 Final parameter:

Learning Rate : 0.01

Stopping Criteria : When difference of cost computer between the alternate iterations is less than 10^{-15}

Theta Obtained :

$$\theta 0 = 0.99662001$$

$$\theta 1 = 0.0013402$$

1.3 Plots:

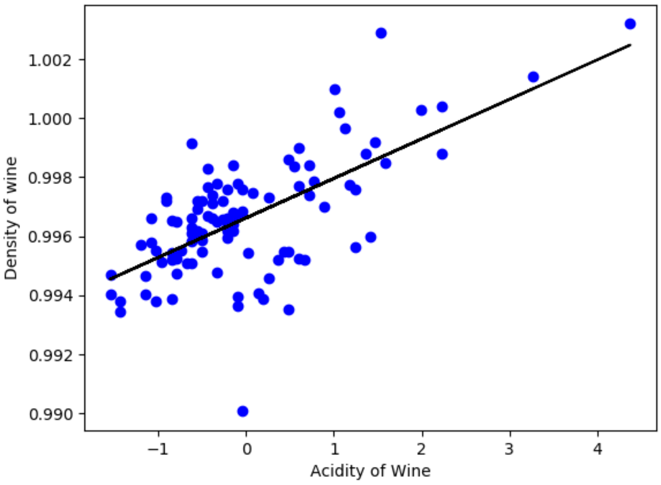

Figure 1: Data with Hypothesis

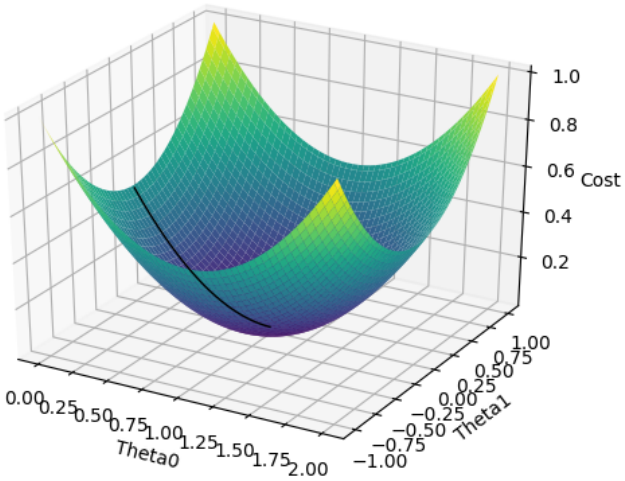

Figure 2: 3D-Mesh

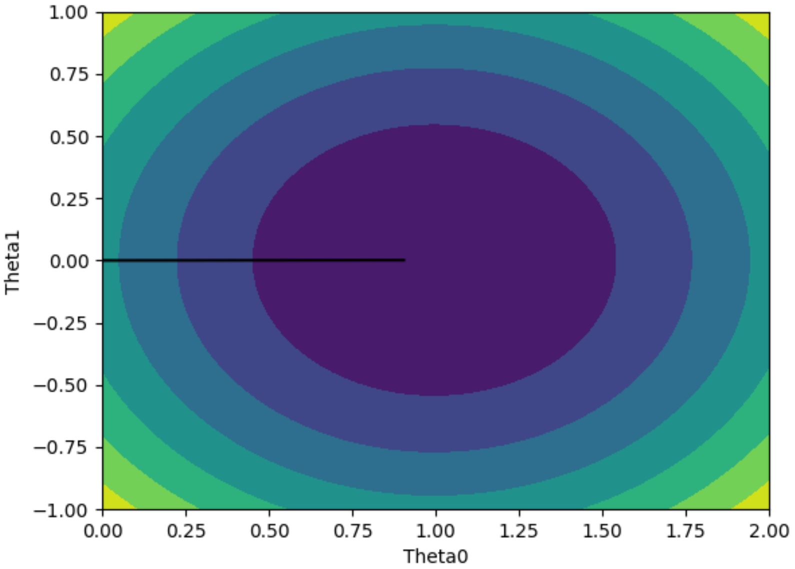

Figure 3: Contour for $\eta = 0.001$

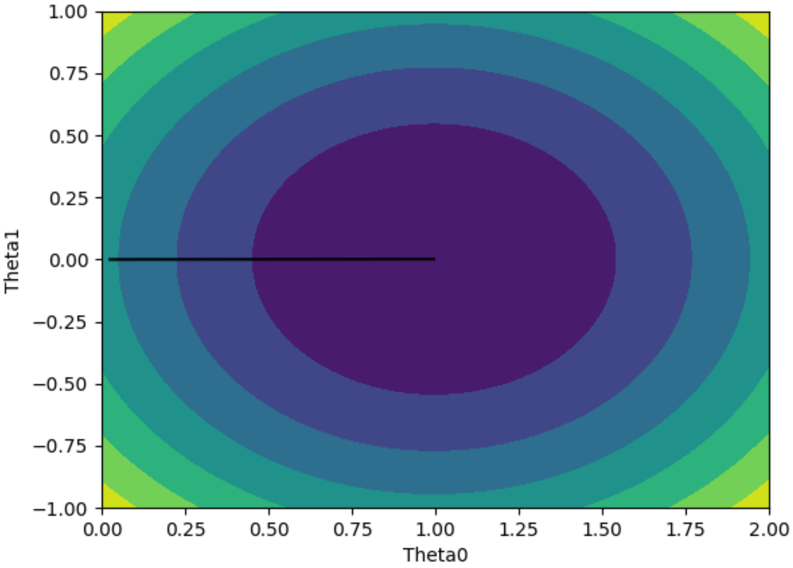

Figure 4: Contour for $\eta = 0.025$

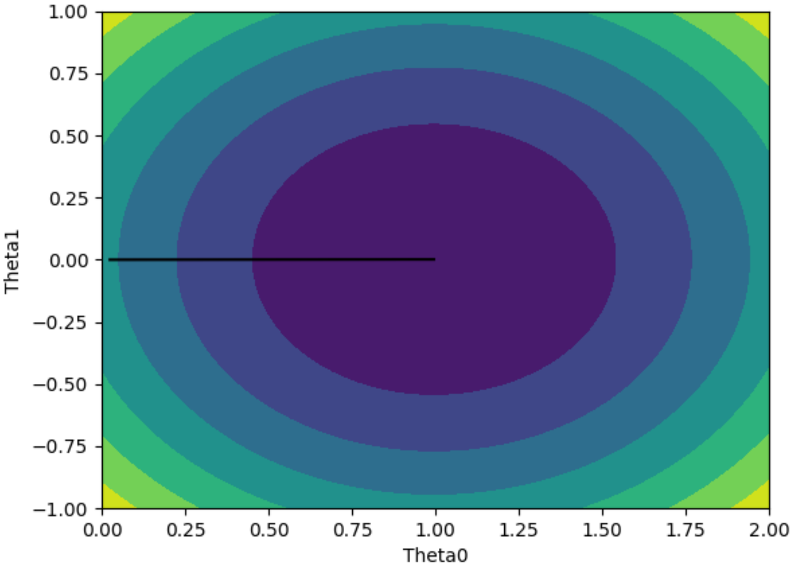

Figure 5: Contour for $\eta = 0.1$

1.4 Observations:

The time contour is taking to converge varies in the order of batch sizes T($\eta = 0.001$) > T($\eta = 0.025$) > T($\eta = 0.1$). This is because cost function depends on η and when it is comparatively small, it is taking long to converge.

2 Sampling and Stochastic Gradient Descent

In this, i have done sampling of 1 million points for the given normal distribution and then implemented stochastic gradient descent and found convergence criteria for different batch sizes.

2.1 Cost Function

$$J_b(\theta) = 1/2r \sum_{k=1}^{r} (y^{(i_k)} - h_\theta x^{(i_k)})^2$$

2.2 Final parameter:

Learning Rate : 0.001

Stopping Criteria : It is computed by taking difference of average of 1000 alternate iterations and when it is less then γ where, γ is different for different batch sizes.

Batch Sizes	$\theta 0$	$\theta 1$	$\theta 2$
1	2.94010294	1.03112853	1.97253566
100	2.92231722	1.01892543	1.99386426
10000	3.00346197	0.99943629	2.0009043
1000000	2.99718894	1.00063337	2.00041179

Batch Sizes	**Convergence Criteria(γ)**	**Number of Iterations**	**Time Taken(sec)**
1	10^{-3}	167000	5.153075218200684
100	10^{-3}	13000	0.50669264793396
10000	10^{-7}	30000	11.211018800735474
1000000	10^{-5}	22000	410.7630362510681

2.2.1 Observations :

The algorithms don't exactly converge to the same point, but they are approximately same. We can see from the table, values of the theta are converging approximately to 3,1,2 and original hypothesis was exactly 3,1,2. So, not much difference is observed.

2.2.2 Relative Speed of convergence :

We can see from the table, time taken by the algorithm for different batch sizes

- Batch size 1 take less time to converge despite taking more iterations.
- Batch size 100 also takes less time to converge and less iterations than the batch size 1.
- Batch size 10000 takes comparatively more time than batch size 1 and 100.
- Batch size 1000000 took longest to converge as the operations are performed on 1 million data at the same time, number of iterations it took is comparatively less than other.

2.2.3 Errors :

Batch Sizes for different hypothesis	**Error**
1	1.072045322367526
100	1.0041499717505575
10000	0.9828952488270722
1000000	0.9828998936085278

Error obtained from original hypothesis is 0.9829469215000001. We can see the difference between error obtained from our hypothesis and original one. The error obtained from hypothesis of batch size 10000 and 1000000 are almost same and there is very small difference for the other batch sizes.

The data wasn't generated in chunks from original hypothesis and that's why when we are calculating for large batch sizes, the error is almost same.

2.3 Plots:

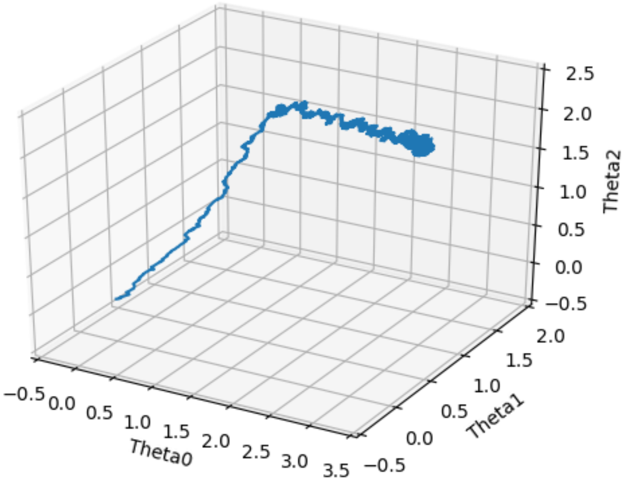

Figure 6: Movement of θ for batch size 1

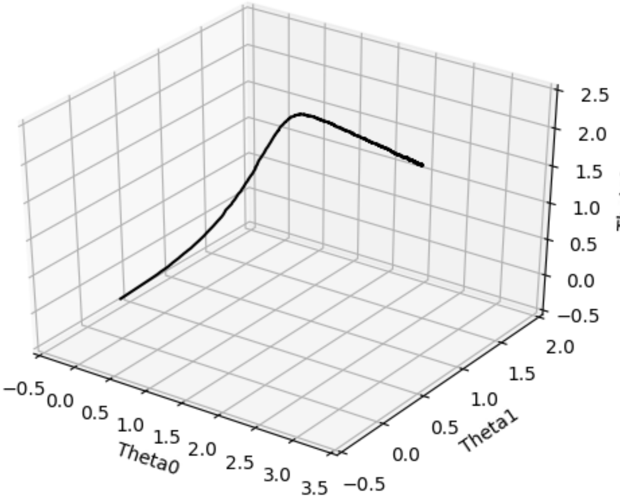

Figure 7: Movement of θ for batch size 100

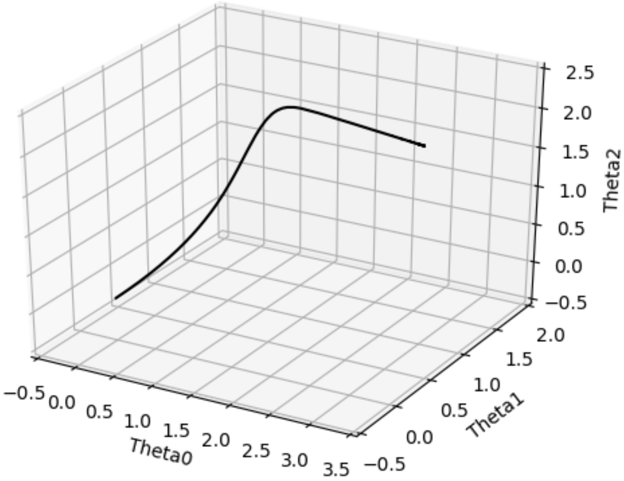

Figure 8: Movement of θ for batch size 10000

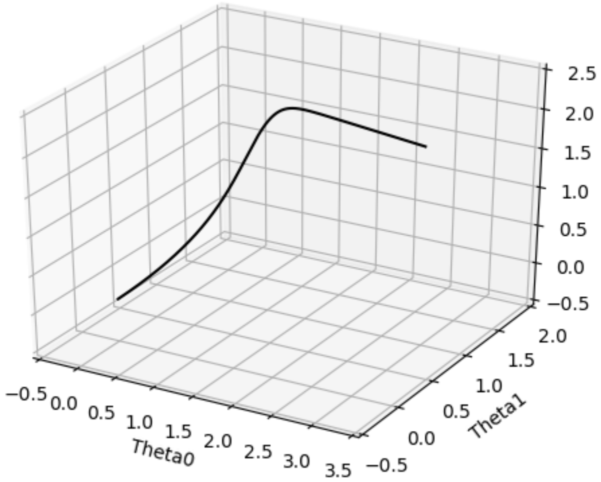

Figure 9: Movement of θ for batch size 1000000

- For batch size 1, line is doing zig-zag movement before converging. It took the longest time.
- For batch size 100, line is going less zig-zag than batch size 1 and it took quite less time to converge.
- For batch size 10000, line has gotten smoother than the earlier ones, it takes comparatively less time than batch size 1 and bit more than batch size 100.
- For batch size 1000000, line is smoothest than all the other ones, it takes comparatively less time than batch size 1,10000 and bit more than batch size 100.

It makes intuitive sense that as the size of batch is increasing, the values of theta is not doing to and fro movement and it also depicts the convergence rate for every batch size.

3 Logistic Regression

In this, i have implemented the logistic regression algorithm to classify the given data into two classes and found a hypothesis line at which probability of a data point to be of either classes have equal probability. It has been implemented by Newton's method.

3.1 Equations obtained in Logistic Regression

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

$$H^{-1} = \sum_{i=1}^{m} -x_{\mathrm{j}}^{(i)} h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))x_{\mathrm{k}}^{(i)}$$

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

3.2 Final Parameters:

Theta Obtained :

$$\theta 0 = 0.40125316$$

$$\theta 1 = 2.5885477$$

$$\theta 2 = -2.72558849$$

3.3 Plots:

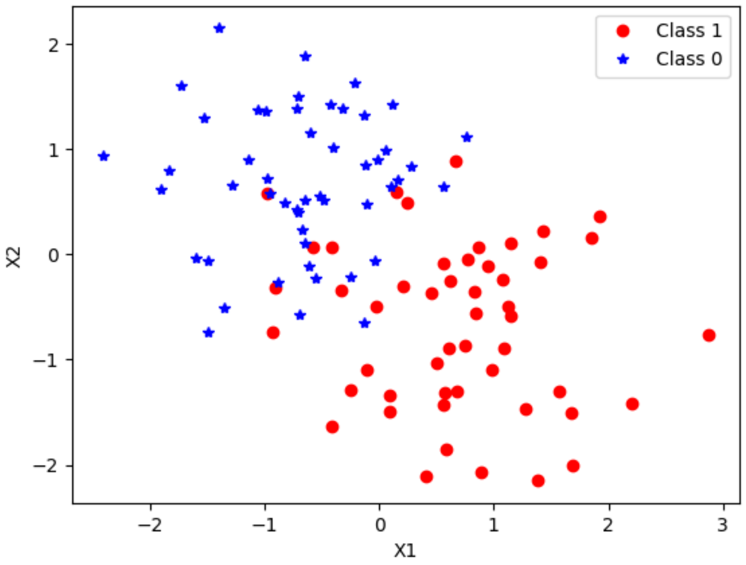

Figure 10: Data plotted depicting different classes

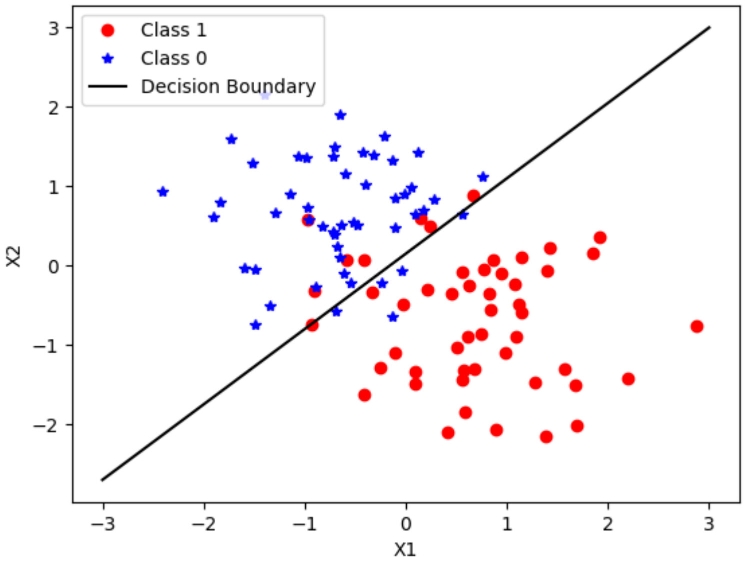

Figure 11: Data along with decision boundary

4 Gaussian Discriminant Analysis

In this, I have implemented Gaussian Discriminant Analysis for separating out two classes Alaska and Canada. It has been implemented for the two cases, when covariances are equal and when covariances are different.

4.1 Equations of Linear Decision Boundary

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

The above equation is of the form $\theta^T x = 0$, which is basically $\theta_0 x_0 + \theta_1 x_1 = 0$ So for a given value of x_0 we can find what is x_1 and thereby decision boundary can be plotted.

4.2 Equations of parameter for the same Co-variance Matrix

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} x^{(i)}}{1\{y^{(i)} = 1\}}$$

$$\mu_0 = \frac{\sum_{i=0}^{m} 1\{y^{(i)} = 0\} x^{(i)}}{1\{y^{(i)} = 0\}}$$

$$\Sigma = \sum_{i=1}^{m} \frac{(x^{(i)} - \mu_y^{(i)})(x^{(i)} - \mu_y^{(i)})^T}{m}$$

$$\phi = \frac{\sum_{i=1}^{m} (1\{y^{(i)} = 1\})}{m}$$

4.3 Final values of parameters obtained for same Co-variance Matrix

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

$$\phi = 0.5$$

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

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

4.4 Plots in the case of equal covariances

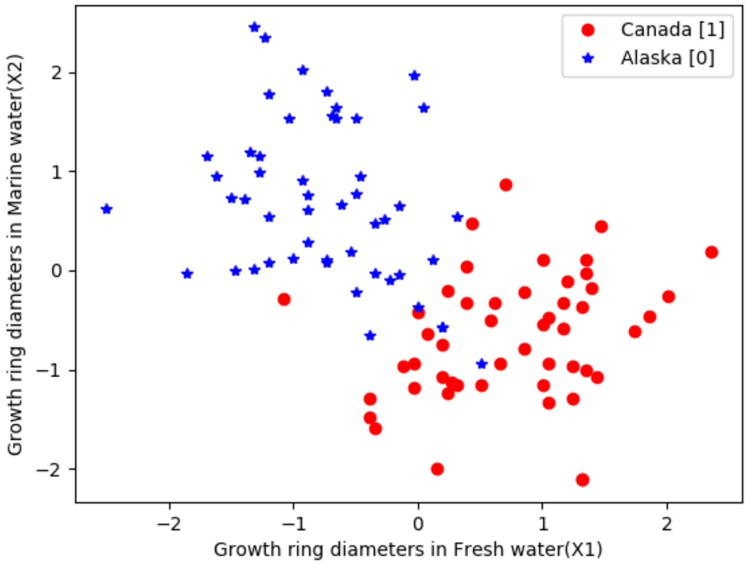

Figure 12: Plotting Training Data

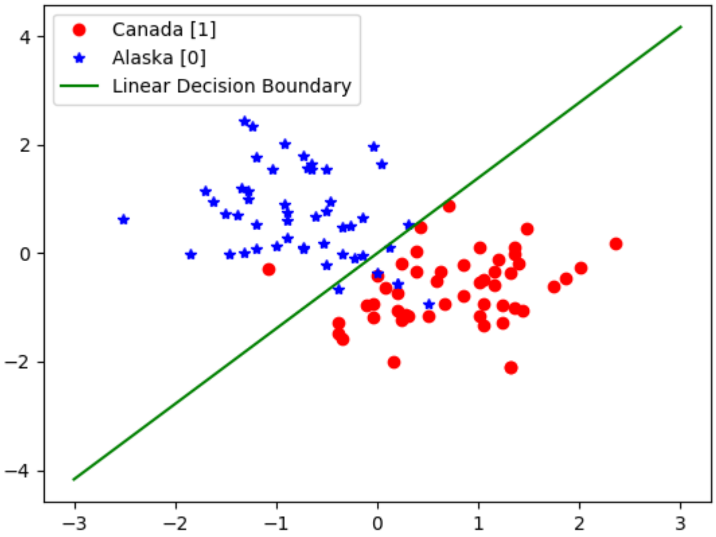

Figure 13: Plotting Training Data with Linear Decision Boundary

4.5 Equations of parameter for different Co-variance Matrix

We are finding here equations of decision boundary and then after finding the cofficient, plotted the quadratic decision boundary for the given data.

The equation of μ_1, μ_0 and ϕ would be same as it was in the case of equal co-variance matrix. Below is the equation derived for the quadratic decision boundary

$$\log(\frac{\phi}{(1-\phi)}) + \frac{1}{2}\log(\frac{|\sum_0|}{|\sum_1|} + x^T\frac{(\sum_0^{-1} - \sum_1^{-1})}{2}x + \frac{x^T\sum_1^{-1}\mu_1 - x^T\sum_0^{-1}\mu_0}{1} + \frac{\mu_0^T\sum_0^{-1}\mu_0 - \mu_1^T\sum_1^{-1}\mu_1}{2} = 0$$

$$\theta_0 + \theta_1 x_1^2 + \theta_2 x_2^2 + \theta_3 x_1 x_2 + \theta_4 x_1 + \theta_5 x_2 = 0$$

$$\theta_0 = \log(\frac{\phi}{(1-\phi)}) + \frac{1}{2}\log(\frac{|\sum_1^{-1}|}{|\sum_0^{-1}|} + \frac{\mu_0^T\sum_0^{-1}\mu_0 - \mu_1^T\sum_1^{-1}\mu_1}{2}$$

let $A = \frac{(\sum_0^{-1} - \sum_0^{-1})}{2}$ $\theta_1 = A_{00}$ $\theta_2 = A_{11}$ $\theta_3 = A_{01} + A_{10}$ $[\theta_4, \theta_5] = \sum_1^{-1}\mu_1 - \sum_0^{-1}\mu_0$

4.6 Final values of parameter obtained for different Covariance Matrix

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

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

$$\phi = 0.5$$

$$\Sigma 1 = \begin{bmatrix} 0.47747117 & 0.1099206 \\ 0.1099206 & 0.41355441 \end{bmatrix}$$

$$\Sigma 0 = \begin{bmatrix} 0.38158978 - 0.15486516 \\ -0.154865160.64773717 \end{bmatrix}$$

4.7 Plot in case of different covariances

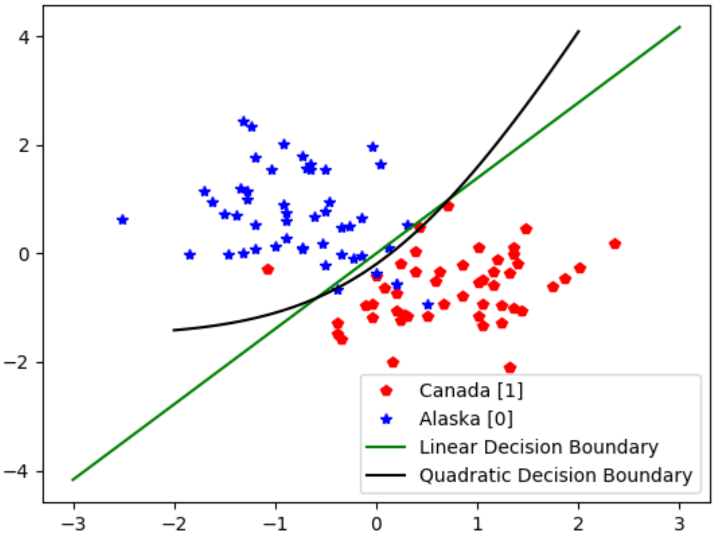

Figure 14: Training Data with Linear and Quadratic Decision Boundary

4.8 Analysis

The quadratic boundary obtained for the given data is predicting most accurately and would have less error than the linear decision boundary. We can clearly see from the graph there are few points of Class Alaska which our linear hypothesis would have predicted for Class Canada and quadratic boundary is handling such cases thereby less error than the linear one.