MACHINE LEARNING

ASSIGNMENT-1 REPORT

Swaransh Patel

2022aib2678

Q1 Linear Regression

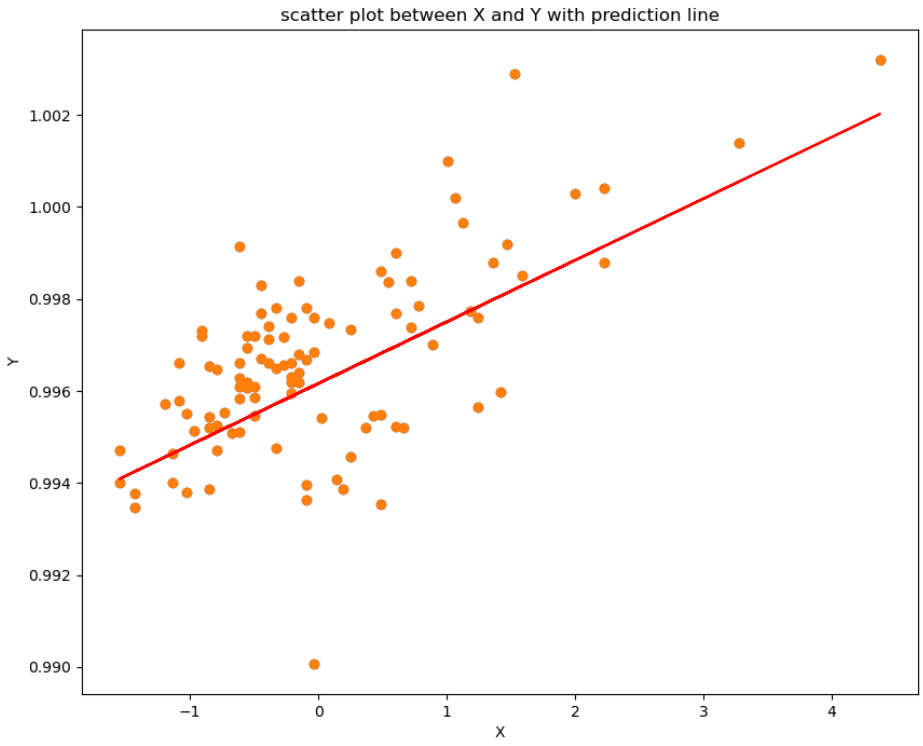
a.)

Batch Gradient Descent.

Learning Rate = 0.001
Stopping Criteria = abs(J_old - J_new) < 10 -15 Final Parameters = [0.99662009 0.0013402] Number of iterations taken = 176

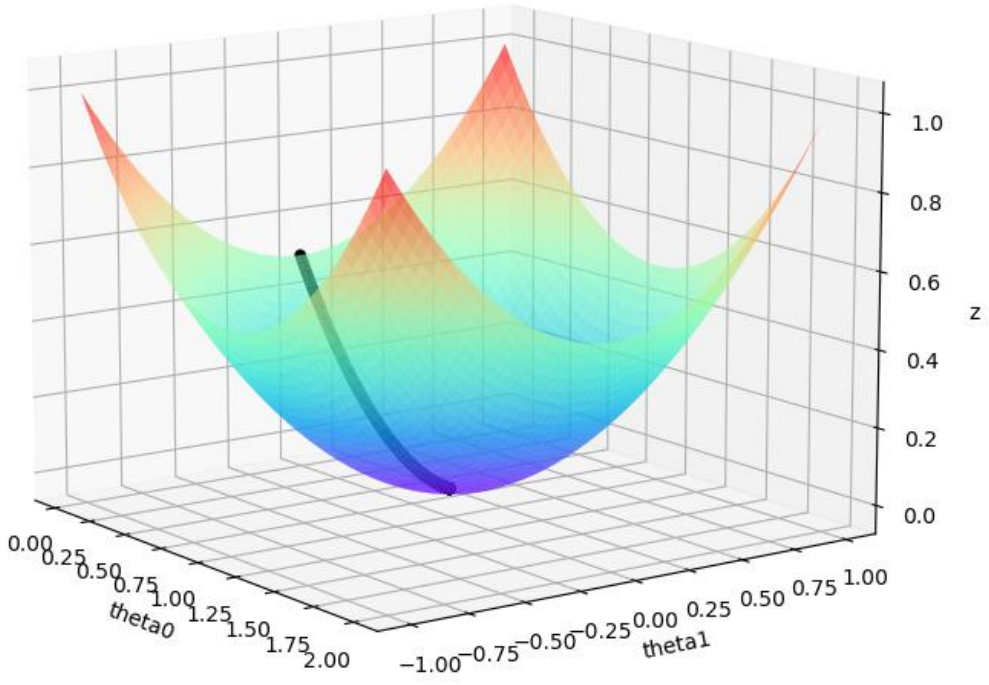
b.)

Plot of the hypothesis function learned:


scatter plot between X and Y with prediction line
1.002
1.000
0.998
0.996
0.994
0.992
0.990
Y
−1
0
1
2
3
4
X

c).

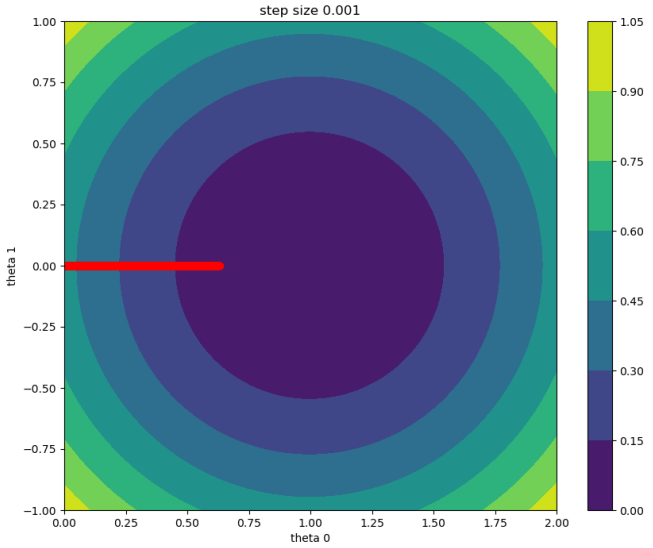
3D plot with error value



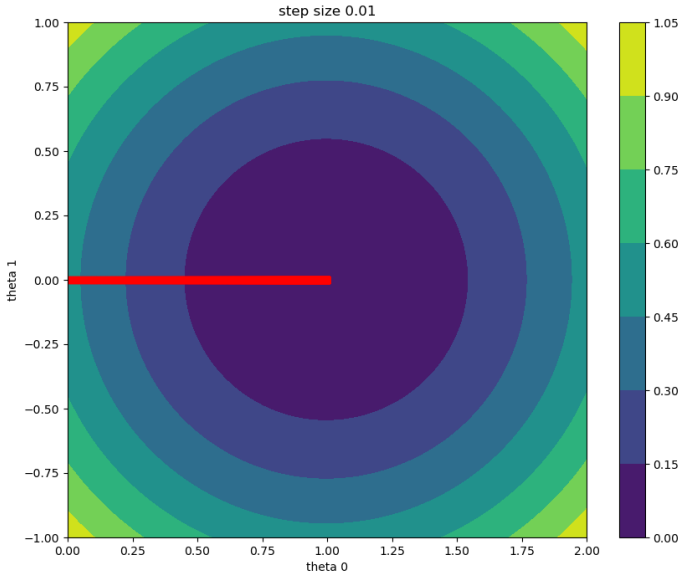
e)

contour plots

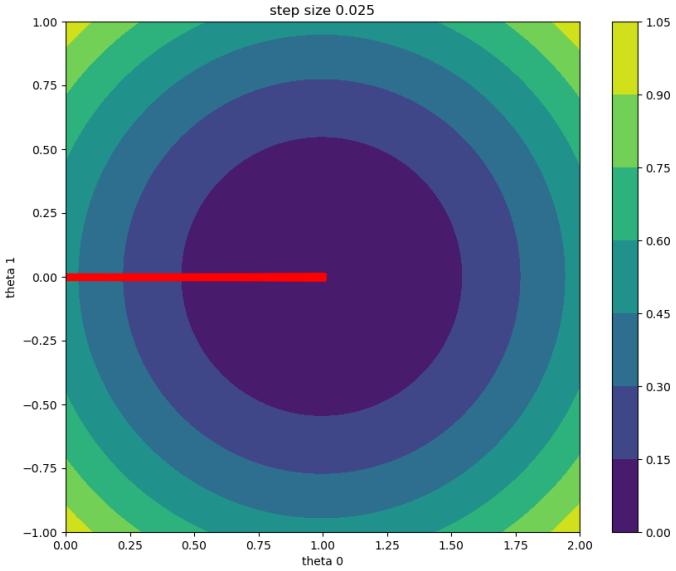
i. n=0.001



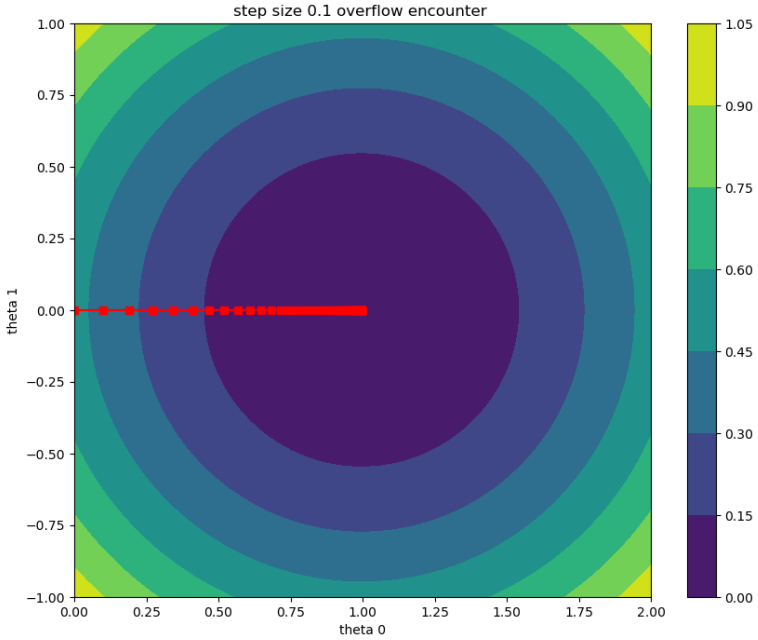
ii. n=0.01



iii. n=0.025



iv. n=0.1



As n(eta) increased then req the number of iterations required to converge is decrease in other words it converges faster. It shown in graph by lesser no of points in there

But then, when n(eta) is increased more the error function oscillates around the minima point
Further increasing eta values results in divergence so the contours hide from view.
In this case it is n(eta)=0.2

Observations:

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

Q2

Sampling and Stochastic Gradient Descent

In this question, we will use probabilistic interpretation of linear regression to generate the data and then apply stochastic gradient descent to optimize the cost.

A) COST FUNCTION

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

B)

batch size= 1,100,10000,1000000

θ learned =>

for batch size (1) is

[[2.99044637],[0.93186902],[1.97906341]]

for batch size (100) is

[[3.00391094],[0.99713277],[2.00637783]]

for batch size (10000) is

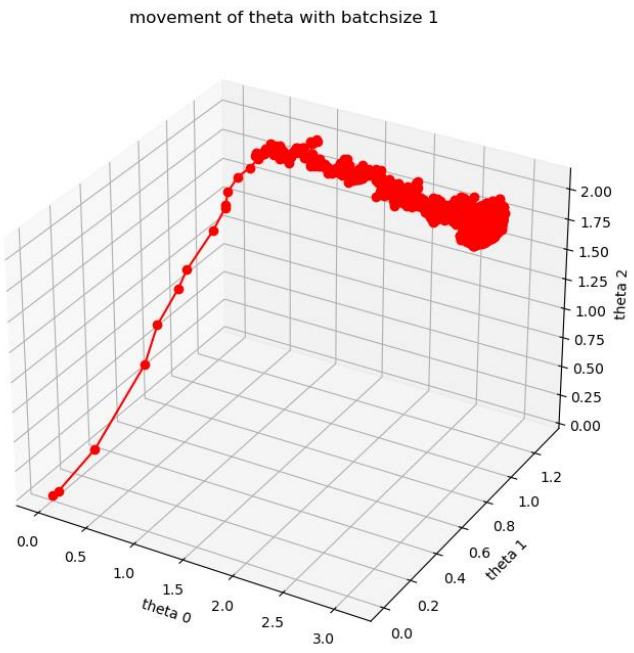
[[2.70443155],[1.1767313],[1.94156976]]

for batch size 1000000 is

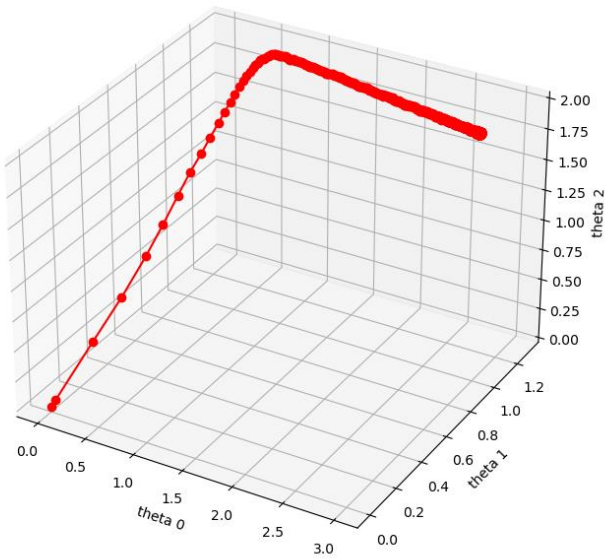
[[2.56965806],[1.28144957], [1.88709581]]

plots

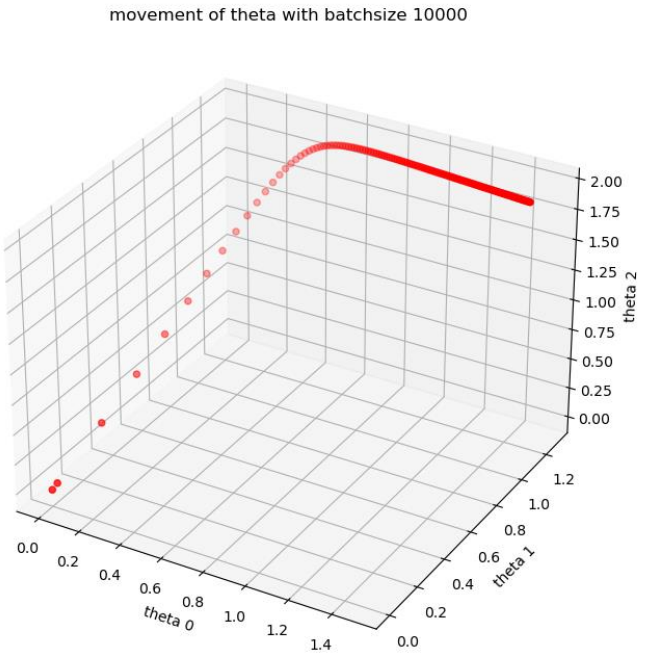
i. batch size 1



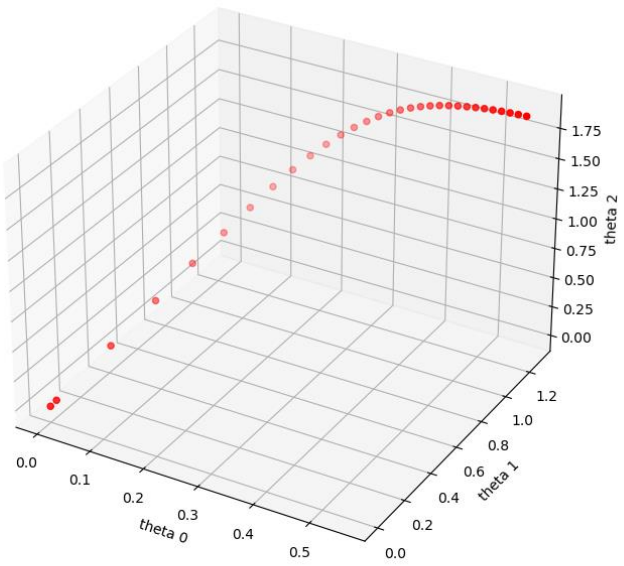
ii. batch size 100


movement of theta with batchsize 100
theta 0
theta 1
theta 2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

iii. Batch size 10000



iv. batch size 1000000


movement of theta with batchsize 1000000
theta 0
theta 1
theta 2
0.0
0.1
0.2
0.3
0.4
0.5
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75

convergence criteria:

for batch size 1
abs(ini_cost-final_cost)>1e-11
for batch size 100
abs(ini_cost-final_cost)>1e-8
for batch size 10000
abs(ini_cost-final_cost)>1e-2
for batch size 1000000
abs(ini_cost-final_cost)>1e-1

Observations :

The algorithms don’t exactly converge to the same point, but they are approximately same. We can see from above 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.

Q3 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.

Equations used:

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

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

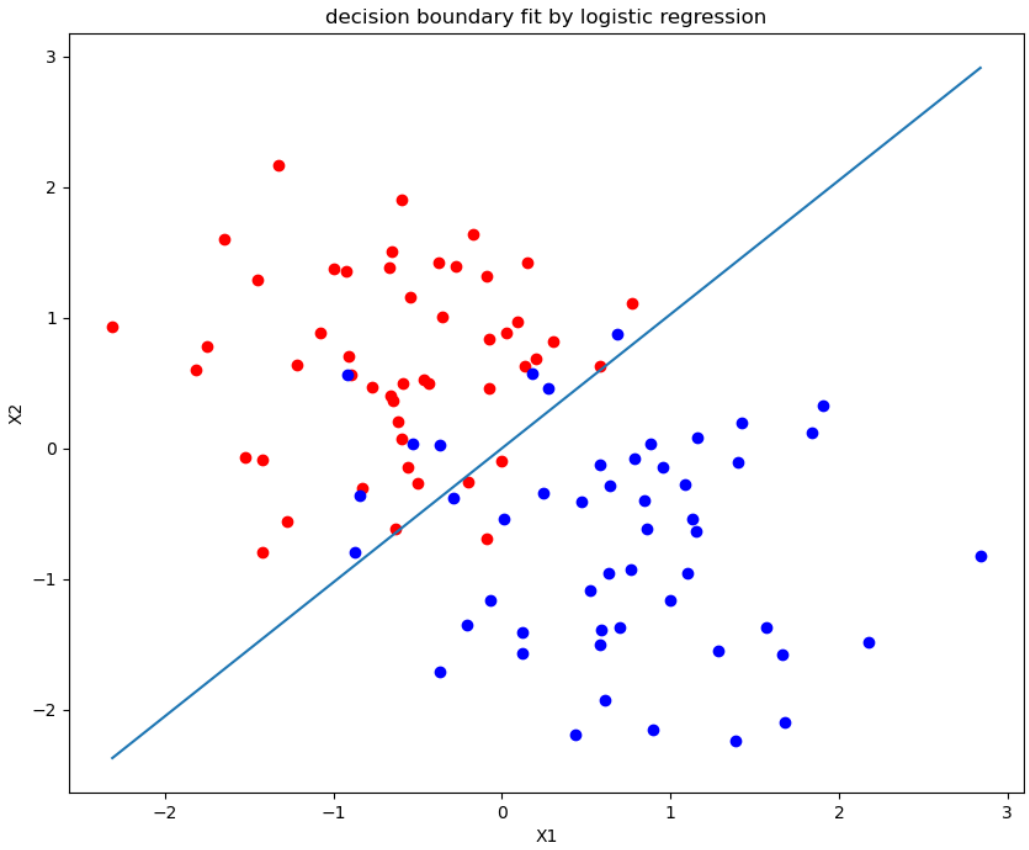
$$\nabla_\theta L(\theta) = \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})x_k^i$$

final parameters:

theta= [[-0.00064394],[0.00921424], [-0.00898329]]

plot:

Data plotted depicting different classes



Q4 Gaussian Discriminant Analysis

Equations used:

a) Linear Decision Boundary

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

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.

b) parameter for the same Co-variance Matrix

c)

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

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

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

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

parameters obtained for same Co-variance Matrix

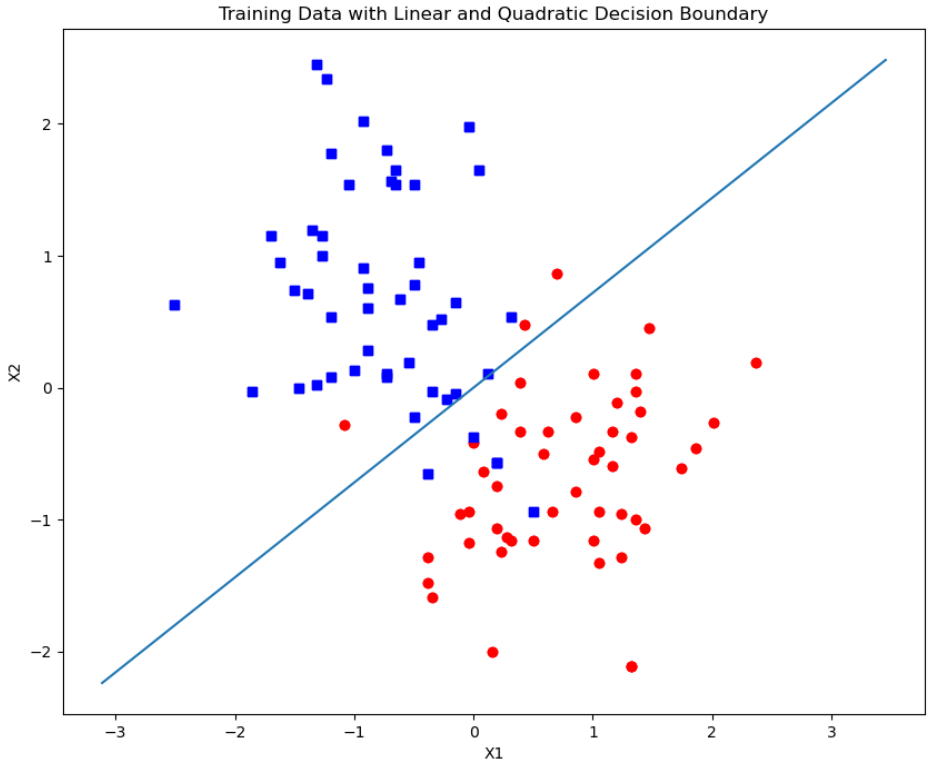
$$\mu_1 = \begin{bmatrix} 0.75529433 \\ -0.68509431 \end{bmatrix}$$

$$\phi = 0.5$$

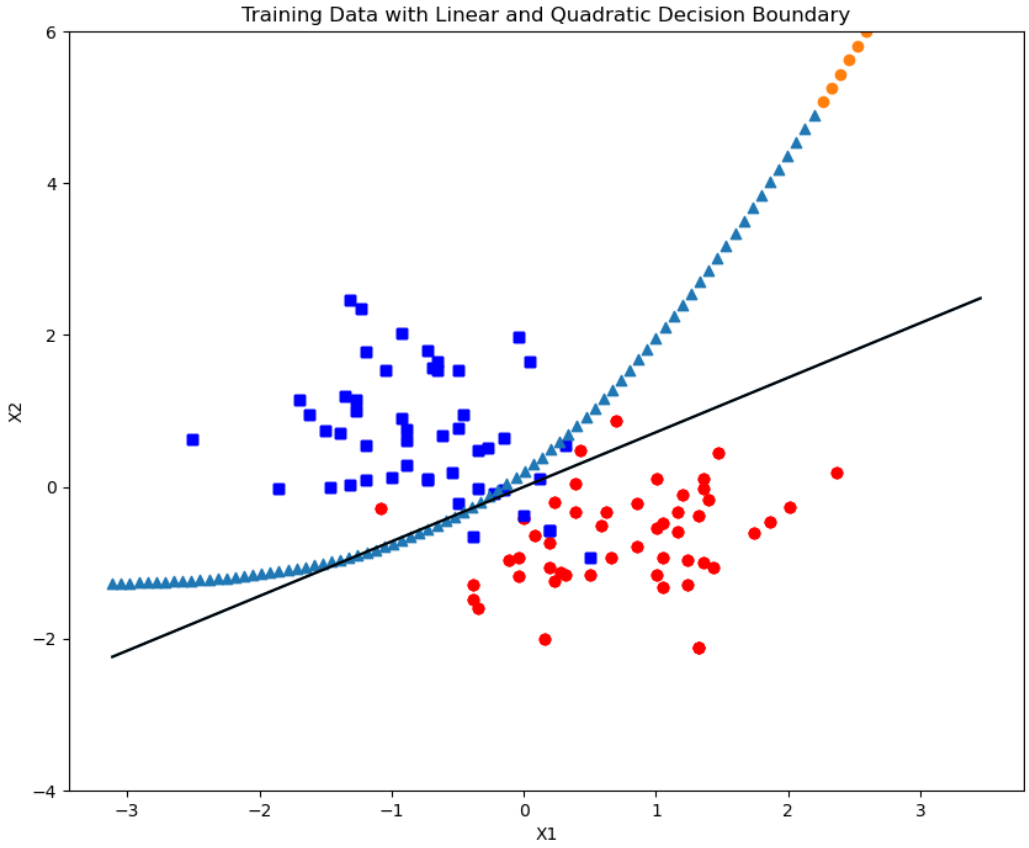
$$\mu_0 = \begin{bmatrix} -0.75529433 \\ 0.68509431 \end{bmatrix}$$

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

d) plots
with linear decision boundary


Training Data with Linear and Quadratic Decision Boundary
X2
X1
2
1
0
−1
−2
−3
−2
−1
0
1
2
3

e)plots with quadratic decision boundary



f) parameter for different Co-variance Matrix

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

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

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

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

g) Final values of parameter obtained for different Covariance Matrix

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

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

$$\phi = 0.5$$

```
cov0=[[0.47747117, 0.1099206 ]
      [0.1099206 , 0.41355441]]
cov1=[[ 0.38158978 ,-0.15486516]
       [-0.15486516 , 0.64773717]]
```

ANALYSIS:

from above data we clearly see quadratic boundary is predicting most accurately as comparatively linear decision boundary.