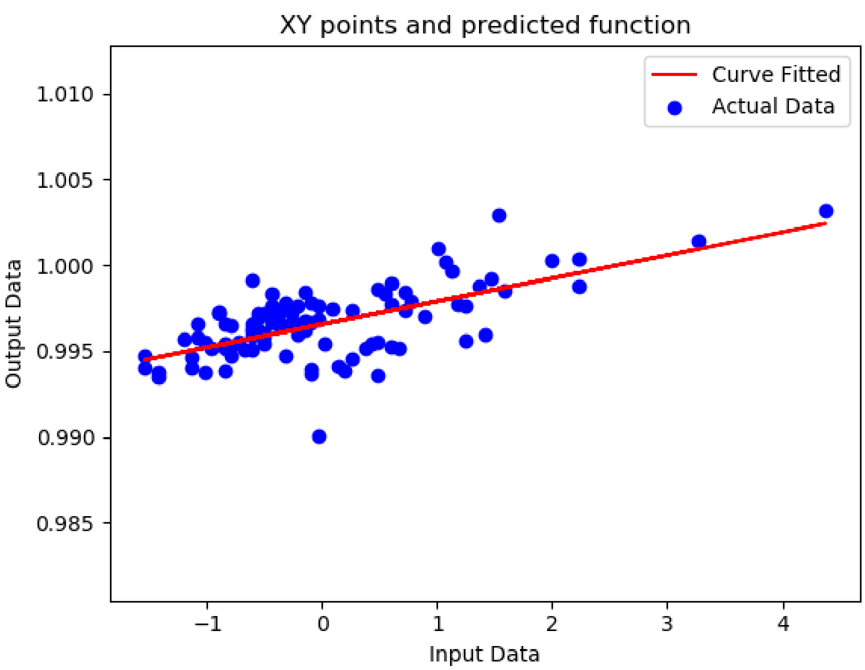
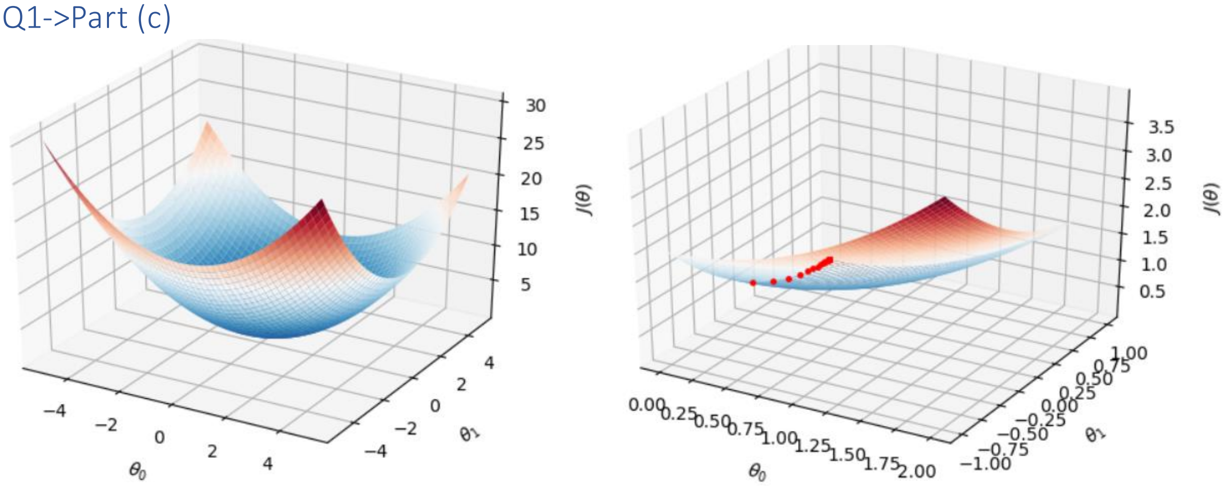
Machine Learning (COL774)
Assignment #1

Tanu Kanvar
(2018EEY7537)

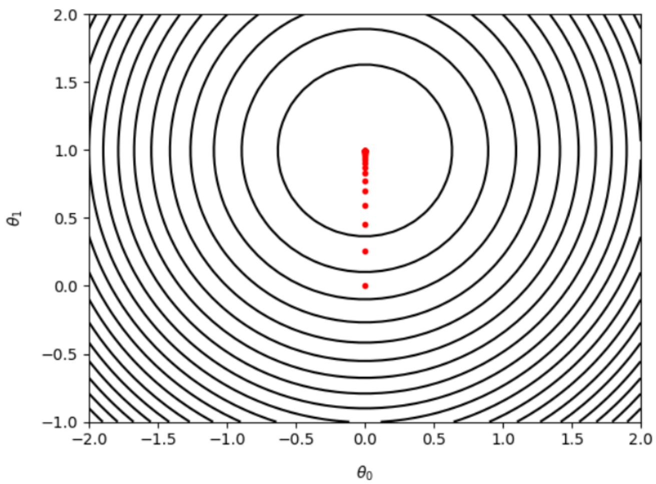
Q1->Part (a)
Learning Rate = 0.003
Stopping Criteria = J0 - JPvs <= 1e-13
Theta Vector = [0.0013401936587243866, 0.9966183451740515]

Q1->Part (b)


Q1->Part (c)


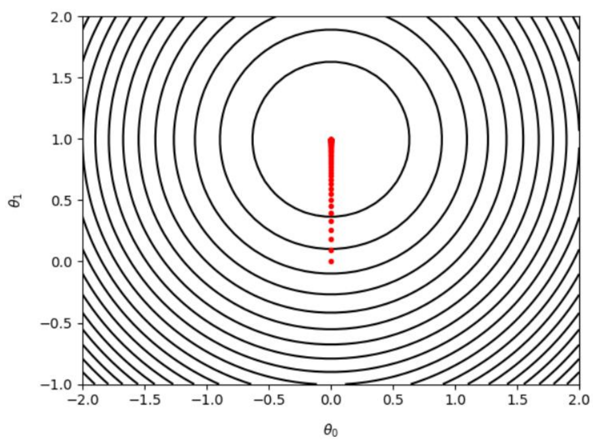
Q1->Part (d)

Contour for learning rate = 0.003

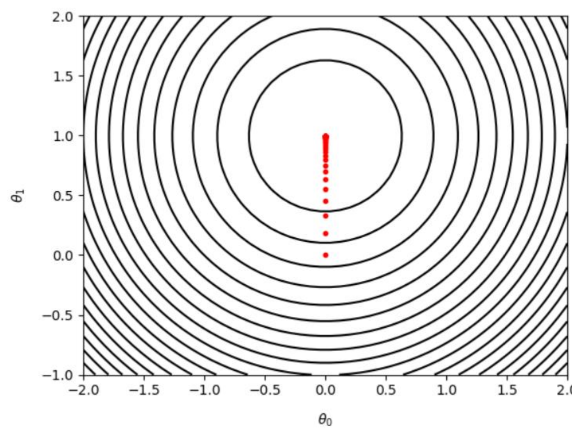


Q1->Part (e)

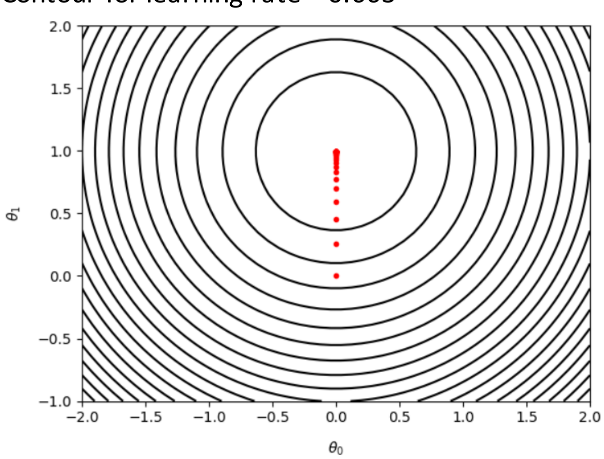
Contour for learning rate = 0.001



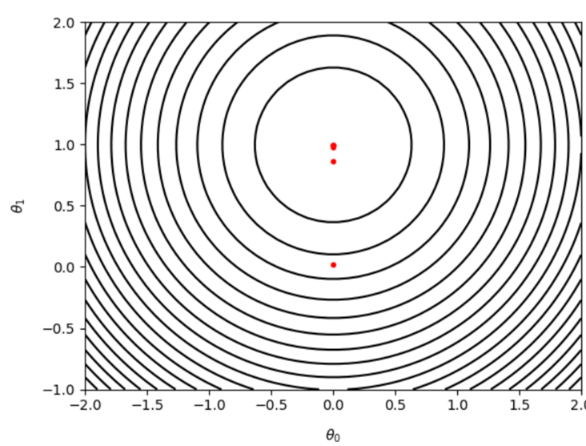
Contour for learning rate = 0.002



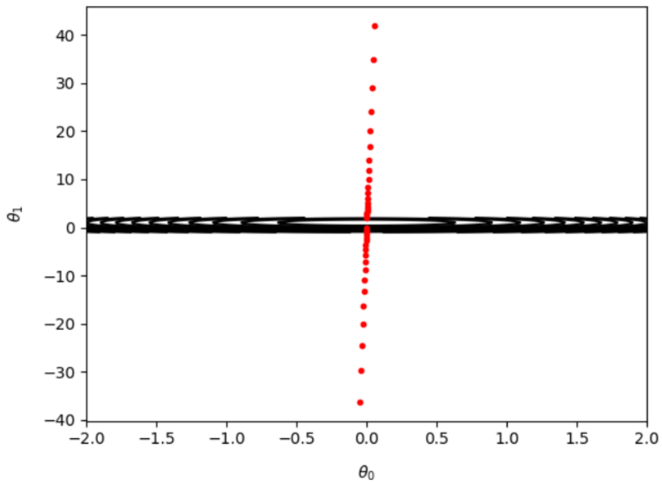
Contour for learning rate = 0.003



Contour for learning rate = 0.02



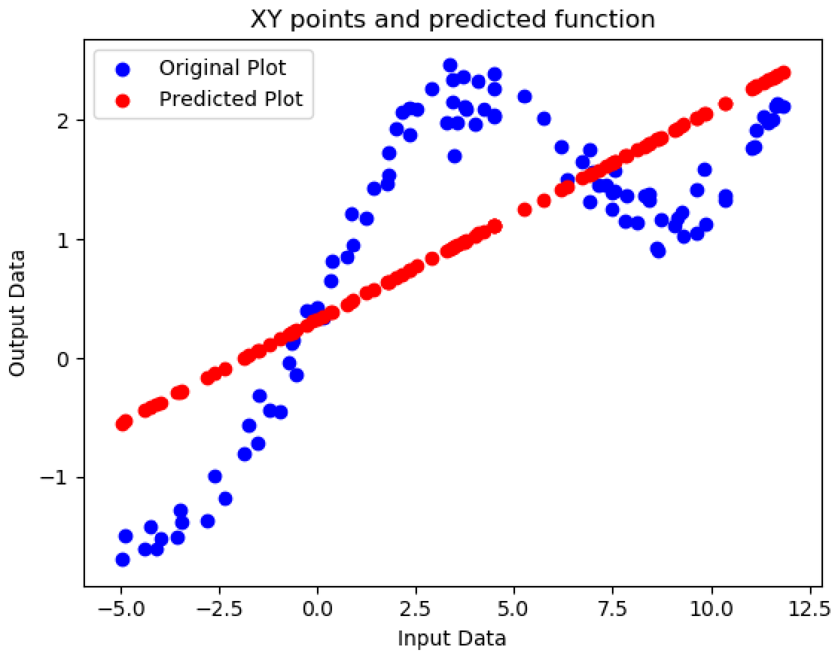
Contour for learning rate = 2.1



As the learning rate increases, the number of steps to converge to optimum point with minimum error decreases. After a certain value (for example in this case, learning rate above 2.1) the gradient descent starts to diverge. And thus never converges to optimum point.

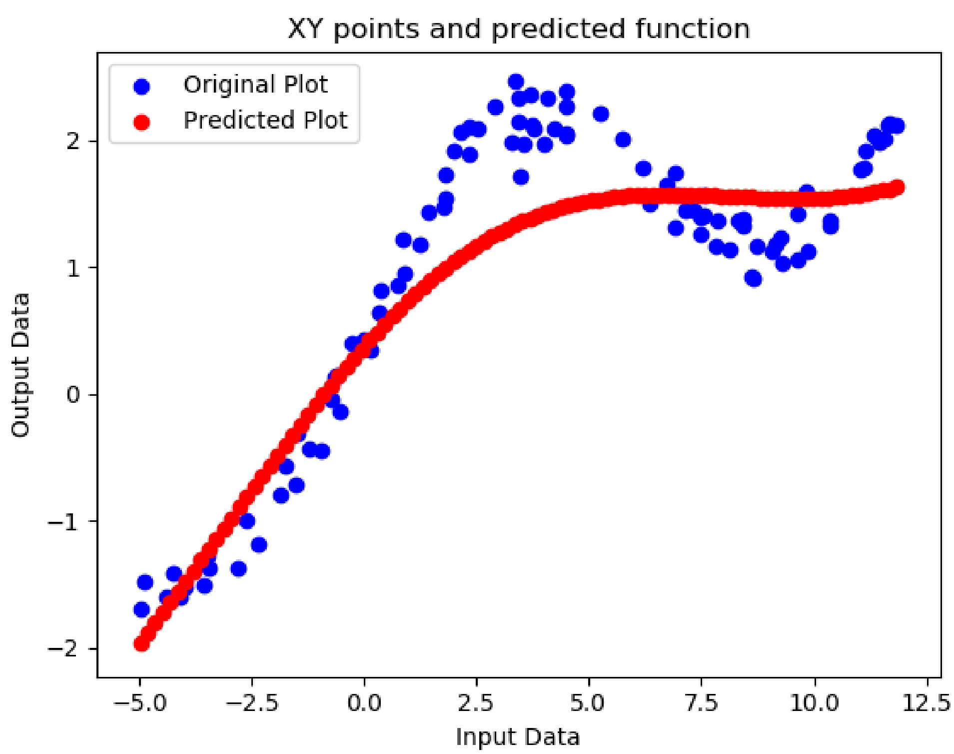
Q2->Part (a)

Theta Vec = [0.10277078407327364, 1.03128116]



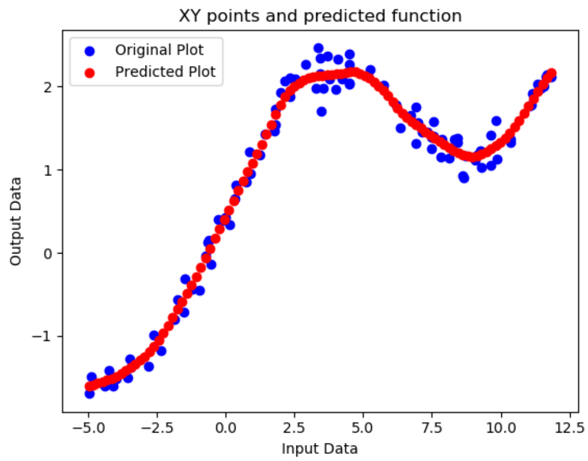
Q2->Part (b)

Tau = 0.8

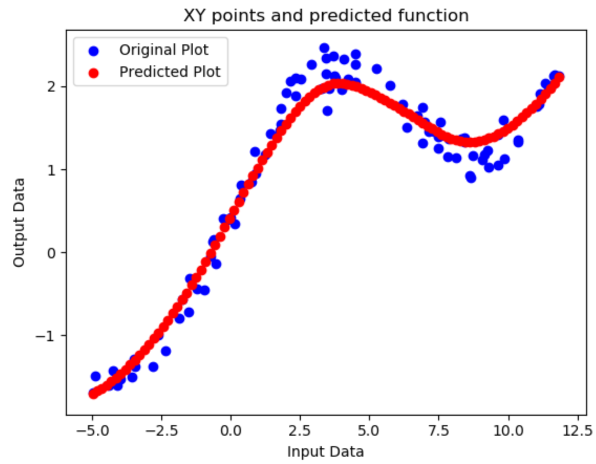


Q2 ->Part (c)

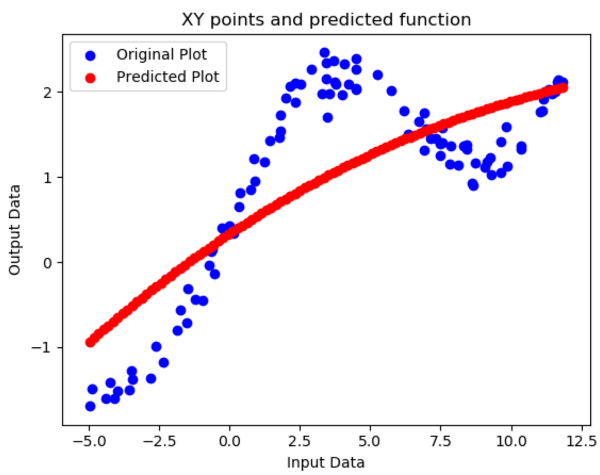
Tau = 0.1



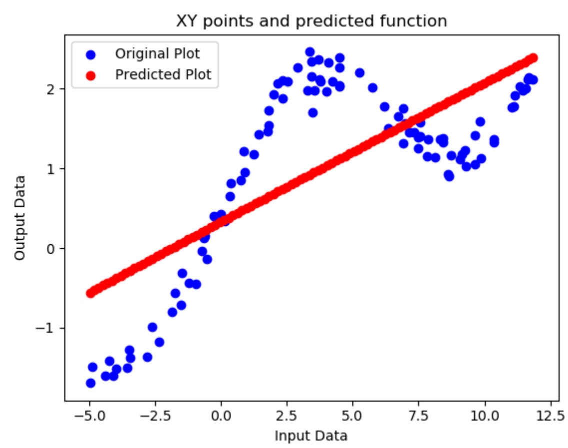
Tau = 0.3



Tau = 2



Tau = 10



Tau = 0.3 shows the best fit.

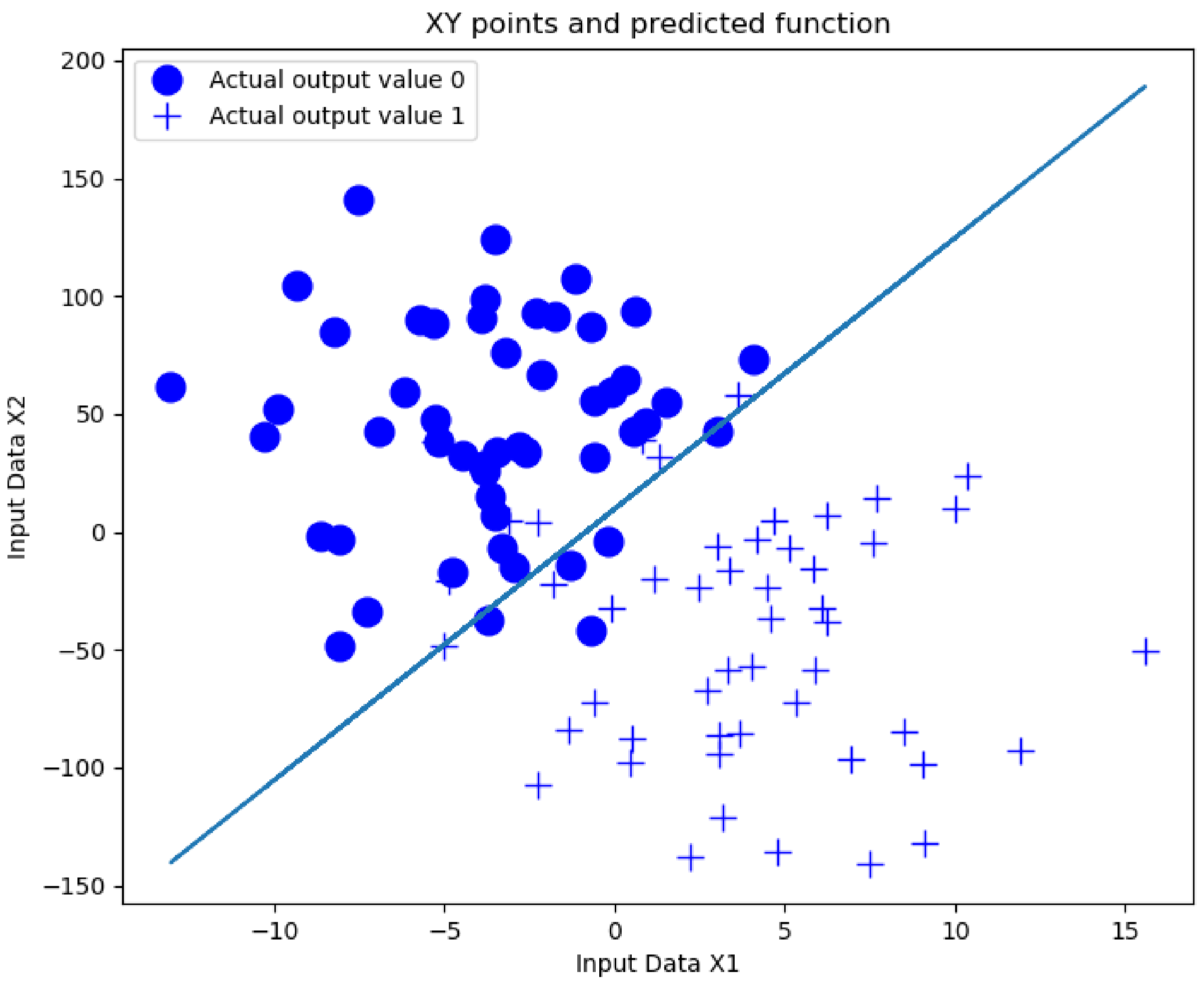
The dependence of a point (p) on its neighboring points decreases exponentially as we go farther away from the point p.
For a point p and considering its some neighboring point q, lower the value of tau, higher will be the term $\frac{(x-x^{(i)})^2}{2\tau^2}$. Therefore lower will be $\exp\left(-\frac{(x-x^{(i)})^2}{2\tau^2}\right)$ because of negative exponent. And the weight on point q will decrease exponentially for lower tau.
So for a given point (p), as tau decreases, the number of neighboring points considered for fitting a curve to p also decreases.

Hence very low value of tau might give wrong fit to points as not much neighboring points are considered (as can be seen for tau = 0.1 case). For very high tau, the curve fitting of a point depends on almost all points in graph and so approaches linear regression (as can be seen in tau = 10)

Q3->Part (a)

Theta Vector = [2.588547696944719, -2.7255884878403966, 0.40125316054552557]

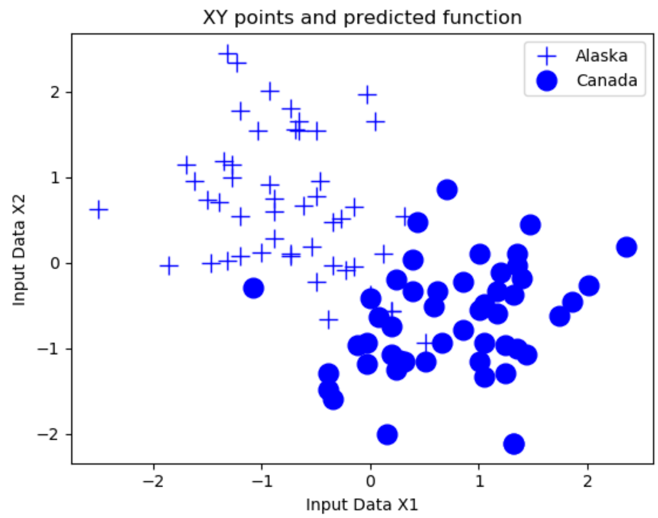
Q3->Part (b)



Q4->Part (a)

Mu0 = [[0.75529433] [-0.68509431]]
Mu1 = [[-0.75529433] [0.68509431]]
Sigma = [[0.42953048 -0.02247228]
[-0.02247228 0.53064579]]

Q4->Part (b)

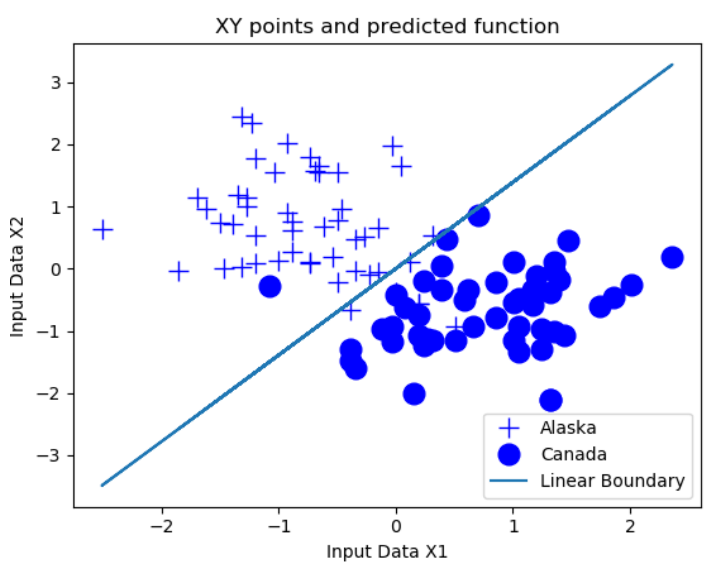


Q4->Part (c)

Linear Boundary is expressed as

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)$$

$$\begin{aligned} \mathbf{w} &= \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \\ w_0 &= -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}. \end{aligned}$$



Q4->Part (d)

Phi = 0.5
Mu0 = [[0.75529433] [-0.68509431]]
Mu1 = [[-0.75529433] [0.68509431]]
Sigma0 = [[0.47747117 0.1099206]
[0.1099206 0.41355441]]
Sigma1 = [[2.66346787 -2.22465653]
[-2.22465653 2.525154]]

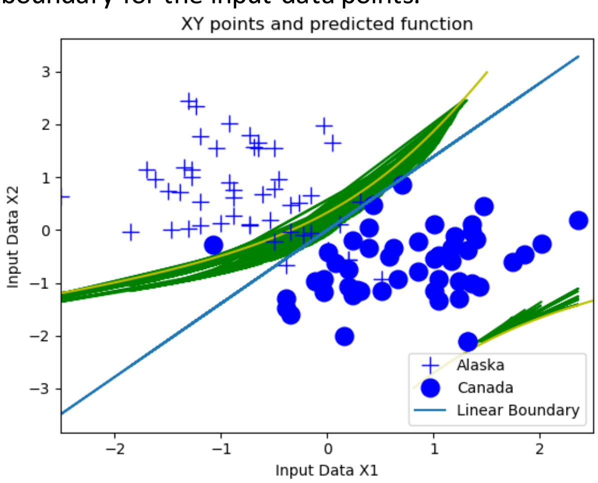
Q4->Part (e)

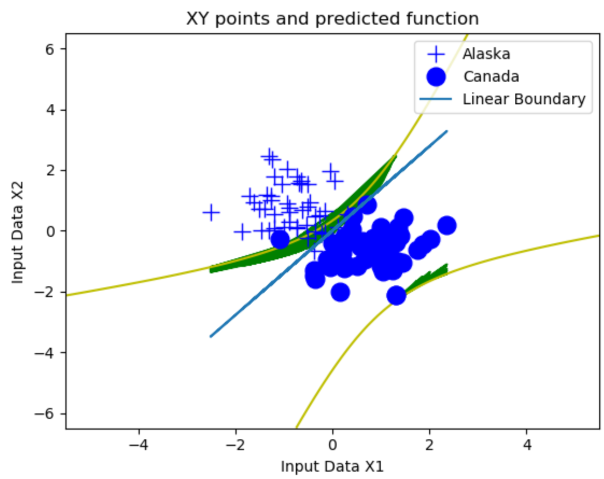
Quadratic Boundary is expressed as

$X^T AX + BX + C = 0$
Where $A = \Sigma_0^{-1} - \Sigma_1^{-1}$
$B = -2 * (\mu_0^T \Sigma_0^{-1} - \mu_1^T \Sigma_1^{-1})$
$C = (\mu_0^T \Sigma_0^{-1} \mu_0 - \mu_1^T \Sigma_1^{-1} \mu_1 - 2 * \log (((1/\phi) - 1) * (|\Sigma_1 | / \Sigma_0))$

Green shaded = 'Quadratic Boundary using input data'
Yellow line = 'Quadratic Boundary by taking points in range'

Yellow line is to show how a boundary would look like. However the green shaded area is the actual boundary for the input data points.





Q4->Part (f)

As can be seen in the above figures, the quadratic is partitioning the data better than linear. The quadratic boundary is specifying the boundary for + and circles more accurately that linear.