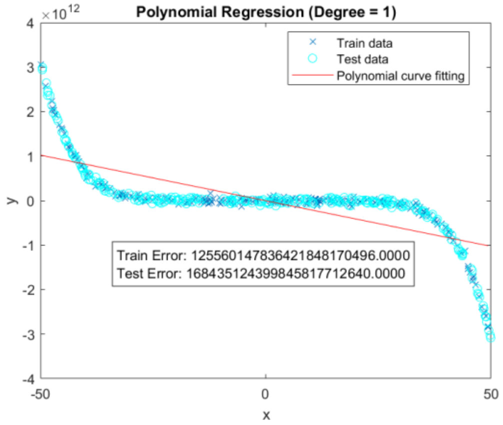
Homework 1

Komal Niraula (N16417290)

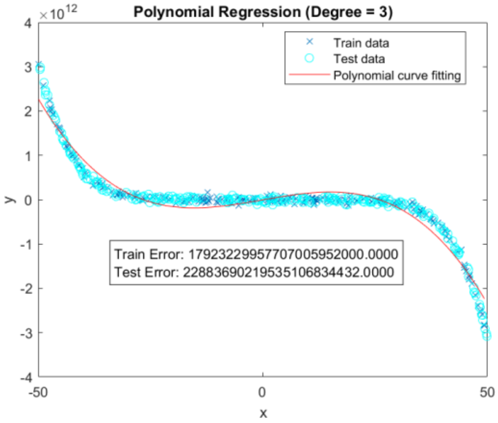
Kn2505@nyu.edu

Problem 1

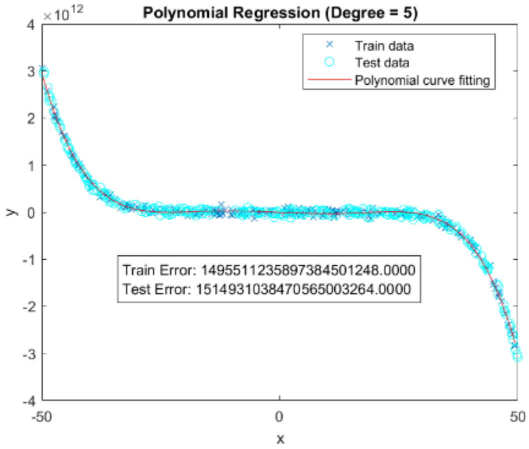
We get the following plot at different degrees by splitting the data into two halves.



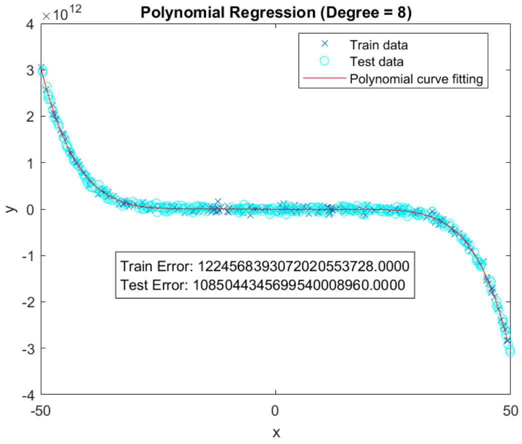
```
Train Error: 125560147836421848170496.0000
Test Error: 168435124399845817712640.0000
```



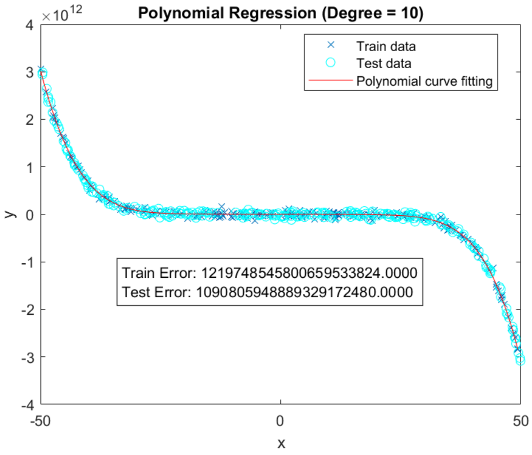
```
Train Error: 17923229957707005952000.0000
Test Error: 22883690219535106834432.0000
```

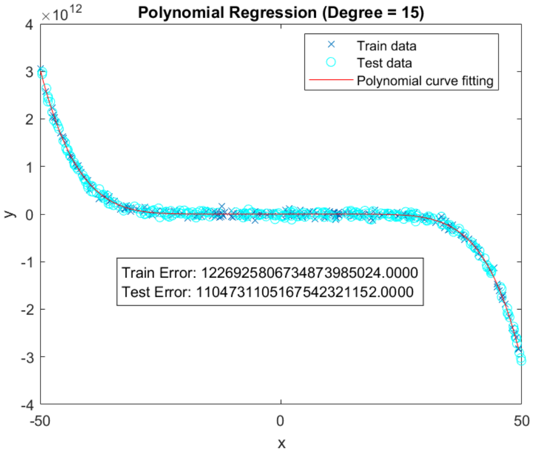


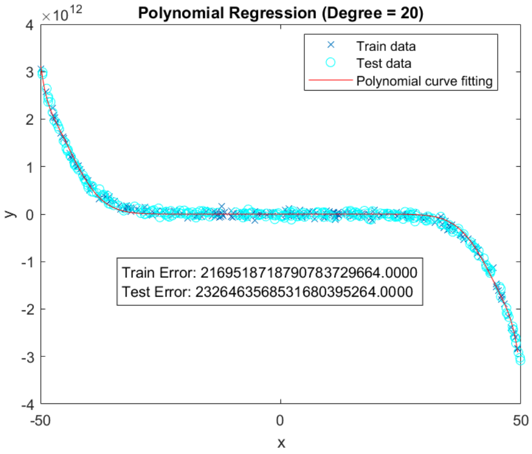
```
Train Error: 1495511235897384501248.0000
Test Error: 1514931038470565003264.0000
```

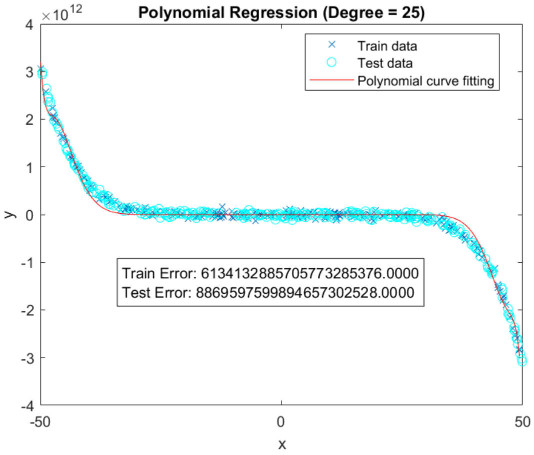


```
Train Error: 1224568393072020553728.0000
Test Error: 1085044345699540008960.0000
```





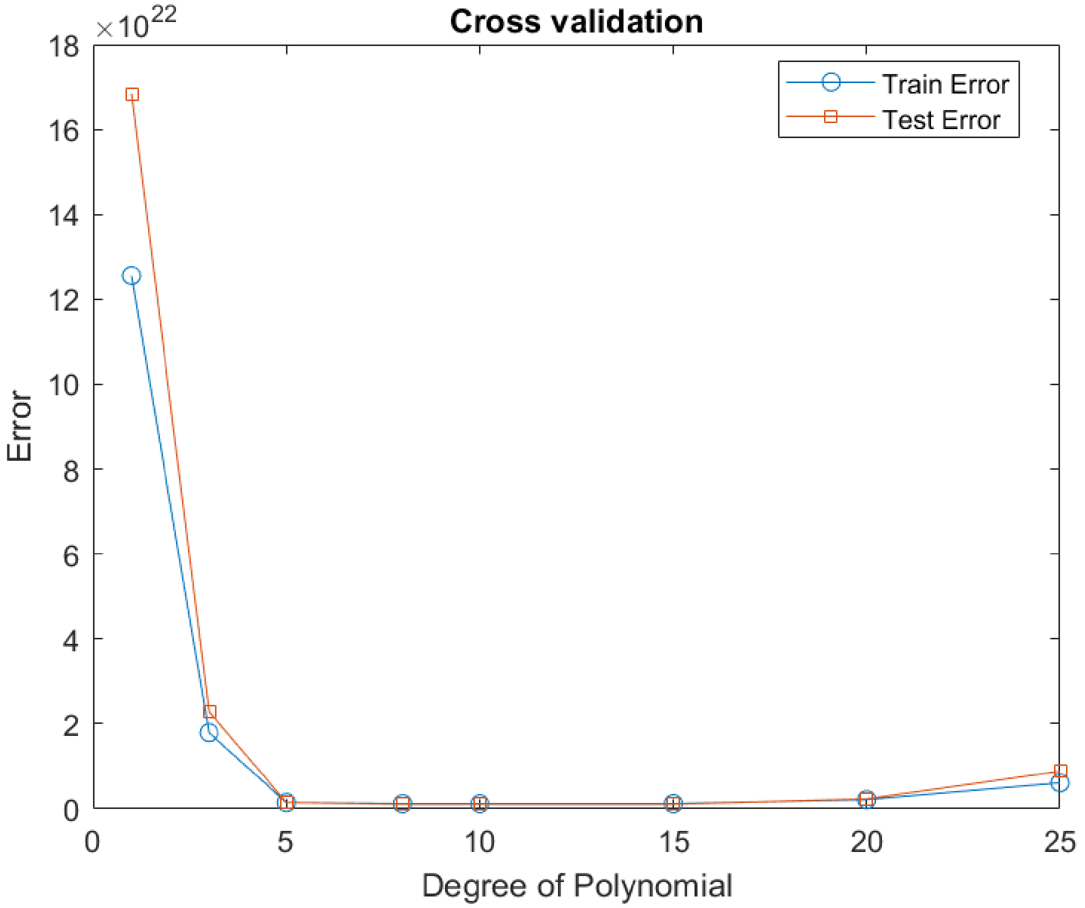




The training and testing error seems to decrease when the degree of the polynomial is being increased. This shows that, with the increase in the number of degrees, the model is becoming better at understanding the patterns in the data. However, after degree 8, the test error starts increasing. This indicates that, after degree 8, the model began to overfit.

The training error also increases significantly for very high degrees, such as 20 and 25. So, degree 8 is the optimal choice here. The test error at degree 8 is better than both smaller and larger degree models.

After cross validation we get the following chart:

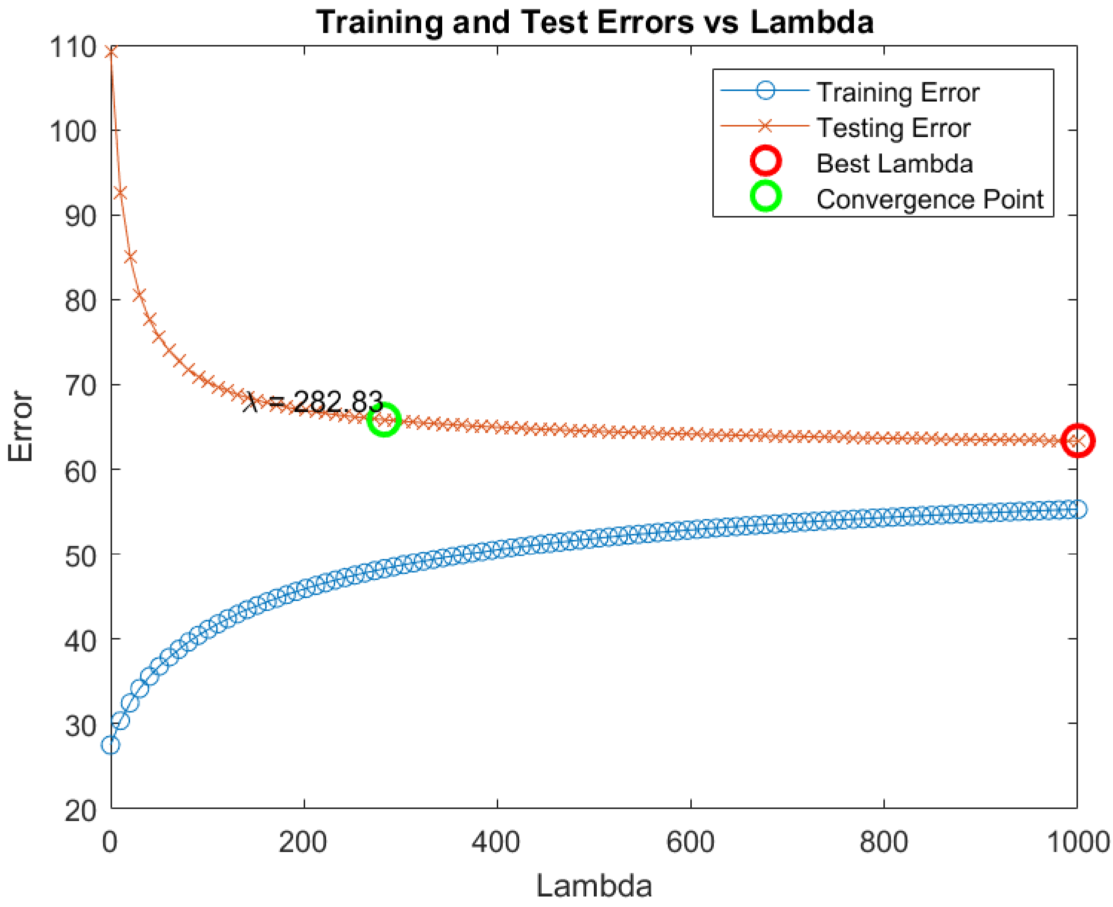


According to the cross-validation, the test error is lower when the degree of polynomial is 8.

Problem 2

The polyreg.m function has been edited in the polyreg2.m

Running the problem2.m code to apply two-fold cross-validation, we get the following plot.



The convergence point occurs when the change in testing error between consecutive values of λ becomes negligible. Using a threshold of 0.1, the convergence point was found to be λ=282.83.

This means the testing error stabilizes here. Also, this point is where further regularization yields minimal improvements.

According to the graph, the testing error is lowest when λ is 1000.

From the graph, we can make the following observations:

- The training error increases as the regularization parameter λ increases. This is because increasing λ adds more regularization, making the model fit the training data less accurately.
- The testing error initially decreases with the increase in λ. Regularization helps to reduce overfitting here, and the testing error stabilizes. Further increases in λ have little to no effect on the testing error.

Problem 3

First question:

Given, $g(z) = \frac{1}{1+\exp(-z)}$

First, let's calculate $(g(-z))$:

$$g(-z) = \frac{1}{1 + \exp(z)}$$

Using the given formula for $(g(z))$:

$$1 - g(z) = 1 - \frac{1}{1 + \exp(-z)}$$

$$1 - g(z) = \frac{(1 + \exp(-z)) - 1}{1 + \exp(-z)}$$

$$1 - g(z) = \frac{\exp(-z)}{1 + \exp(-z)}$$

$$1 - g(z) = \frac{1}{1 + \exp(z)}$$

This is the expression for g(-z). Therefore, we have shown that:

$$g(-z) \;=\; 1 \;-\; g(z)$$

Second question: Proof for the inverse: $g^{-1}(y) = \ln\left(\frac{y}{1-y}\right)$

We are given the logistic function $(\mathrm{g}(\mathrm{z}))$ and need to find its inverse. Let's start with the equation for $(\mathrm{g}(\mathrm{z}))$:

$$y = g(z) = \frac{1}{1 + \exp(-z)}$$

$$y(1 + \exp(-z)) = 1$$

$$y + y\exp(-z) = 1$$

$$y\exp(-z) = 1 - y$$

$$\exp(-z) = \frac{1-y}{y}$$

$$-z = \ln\left(\frac{1-y}{y}\right)$$

This can be written as:

$$z = \ln\left(\frac{y}{1-y}\right)$$

Therefore, the inverse is:

$$g^{-1}(y) = \ln\left(\frac{y}{1-y}\right)$$

Problem 4

Logistic function $\mathrm{f}(\mathrm{x_i};\theta)=\frac{1}{1+\exp(-\theta^{\mathrm{T}}\mathrm{x_i})}$

Empirical risk function $R_{\mathrm{emp}}(\theta)=\frac{1}{N}\sum_{i=1}^{N}\left[(y_i-1)\log\left(1-f(x_i;\theta)\right)-y_i\log\left(f(x_i;\theta)\right)\right]$

$$\nabla_\theta R(\theta)=-\frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i}{f(x_i;\theta)}\frac{\partial}{\partial\theta}f(x_i;\theta)-\frac{1-y_i}{1-f(x_i;\theta)}\frac{\partial}{\partial\theta}f(x_i;\theta)\right)$$

$$\nabla_\theta R(\theta)=-\frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i}{g(\theta^T x_i)}\frac{\partial}{\partial\theta}g(\theta^T x_i)-\frac{1-y_i}{1-g(\theta^T x_i)}\frac{\partial}{\partial\theta}g(\theta^T x_i)\right)$$

Substituting $\frac{\partial}{\partial\theta}g(\theta^T x_i)=g(\theta^T x_i)\left(1-g(\theta^T x_i)\right)x_i$ in the equation:

$$\nabla_\theta R(\theta)=-\frac{1}{N}\sum_{i=1}^{N}\Bigg(\frac{y_i}{g(\theta^T x_i)}g(\theta^T x_i)\left(1-g(\theta^T x_i)\right)x_i \\ -\frac{1-y_i}{1-g(\theta^T x_i)}g(\theta^T x_i)\left(1-g(\theta^T x_i)\right)x_i\Bigg)$$

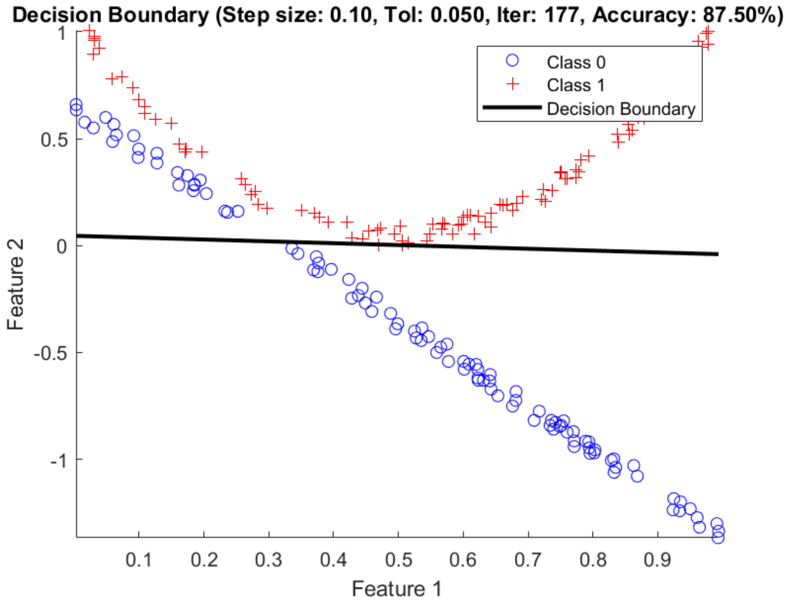
$$\nabla_\theta R(\theta)=-\frac{1}{N}\sum_{i=1}^{N}\left(y_i\left(1-g(\theta^T x_i)\right)x_i-(1-y_i)g(\theta^T x_i)x_i\right)$$

$$\nabla_\theta R(\theta)=-\frac{1}{N}\sum_{i=1}^{N}\left(y_i-y_i g(\theta^T x_i)-g(\theta^T x_i)+y_i g(\theta^T x_i)\right)x_i$$

$$\nabla_\theta R(\theta)=-\frac{1}{N}\sum_{i=1}^{N}\left(y_i-g(\theta^T x_i)\right)x_i$$

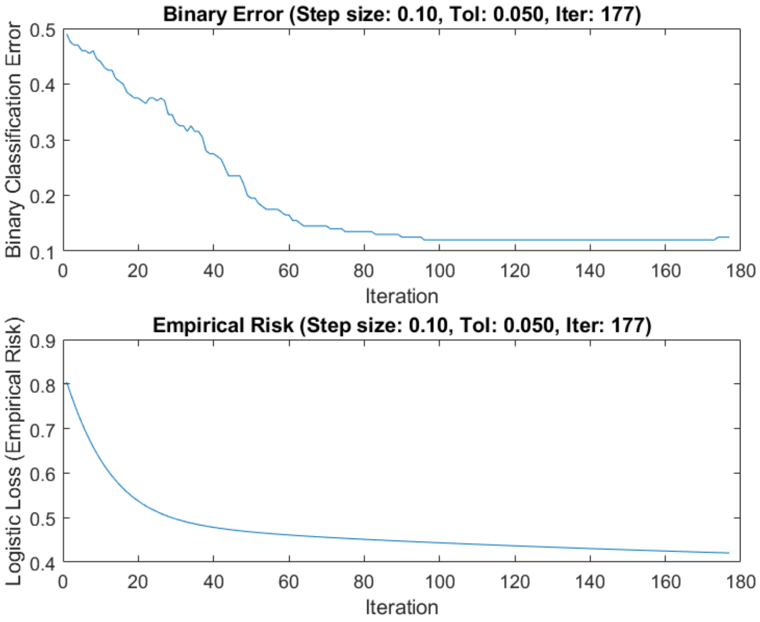
$$\nabla_\theta R(\theta)=\frac{1}{N}\sum_{i=1}^{N}(g(\theta^T x_i)-y_i)x_i$$

When η = 0.1 and ϵ = 0.05, the decision boundary is:



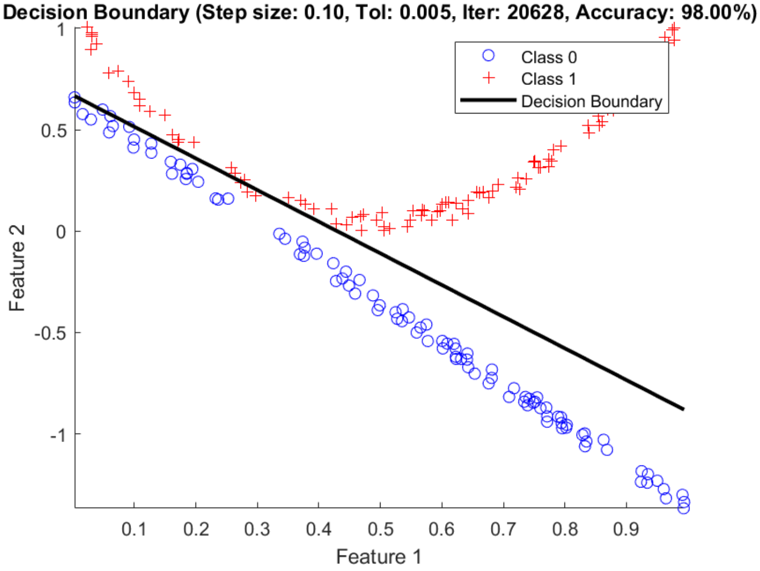
Here, the model took 177 iterations to converge with accuracy of 87.5%

The graph below contains binary error and empirical risk:



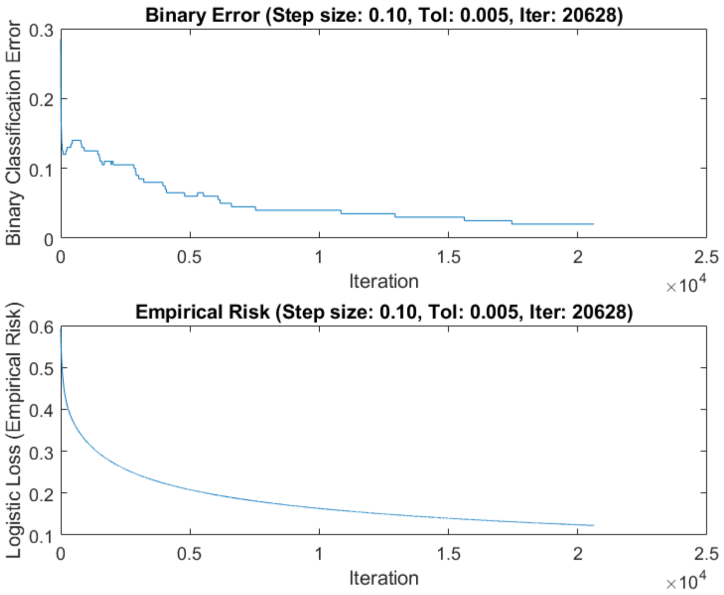
The binary classification error and empirical risk error both decrease steadily, reflecting the optimization process of minimizing errors in predictions.

Decreasing the tolerance, ϵ = 0.005, the decision boundary is:



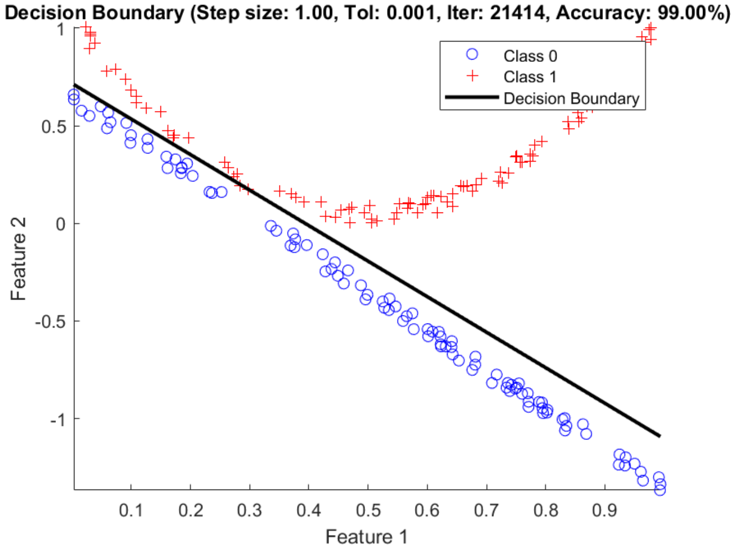
Here, the model took 20,628 iterations to converge with accuracy of 98%

The graph below contains binary error and empirical risk:



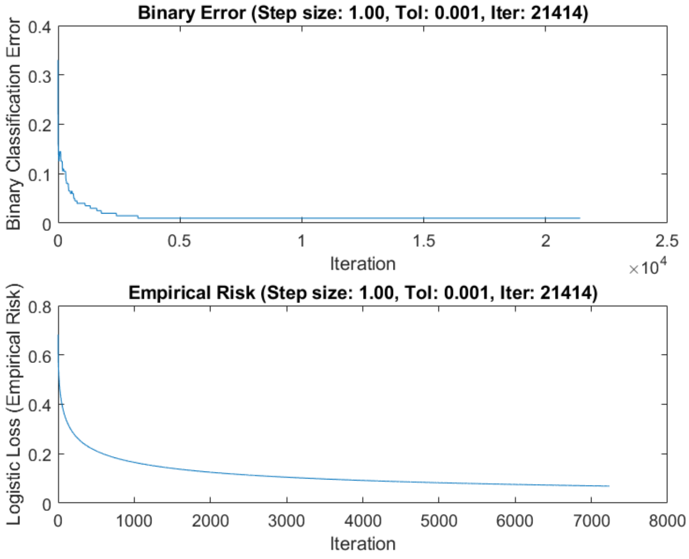
The binary classification error decreases steadily over 20,628 iterations, reaching near-zero values towards the end. This indicates improvement in the model performance. The empirical loss also reflects model prediction becoming more accurate as the loss decreases.

Now, using larger step size, η = 1 and smaller tolerance, ϵ = 0.001, the decision boundary is:



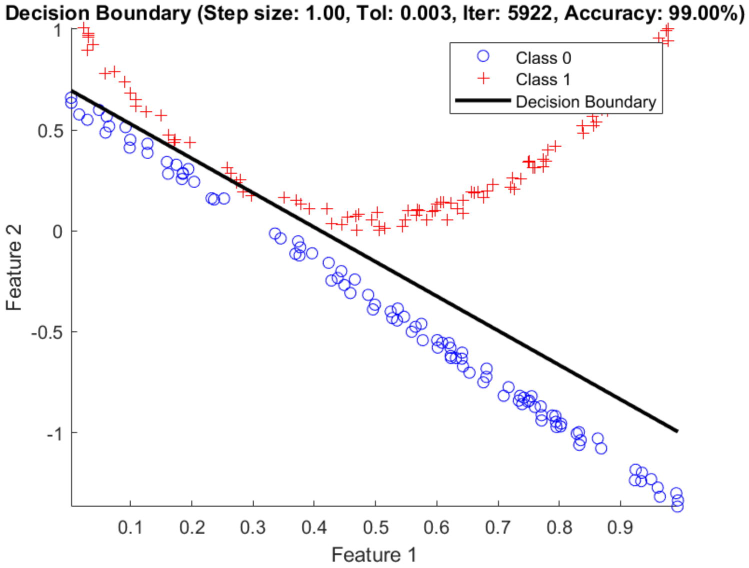
Here, the model took 21,414 iterations to converge with accuracy of 99%

The graph below contains binary error and empirical risk:



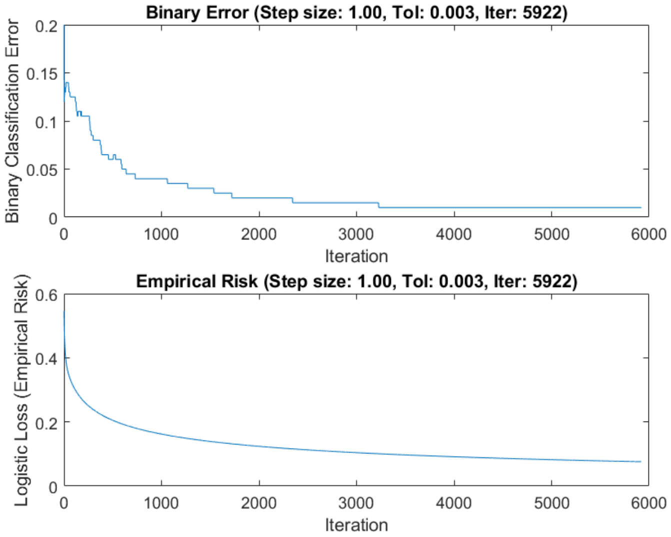
The binary classification error decreases rapidly for the first few thousand iterations, indicating efficient learning early in the training process. The empirical risk also decreases sharply in the early stages, though the rate of improvement slows down as it converges around 7,000 iterations.

Using the same step size, η = 1 but decreasing the tolerance, ϵ = 0.003, decision boundary is:



Here, the model took 5,922 iterations to converge with an accuracy of 99%

The graph below contains a binary error and empirical risk:



The rapid decline of binary classification error in the initial iterations stabilizes close to zero at around 3200 iterations. This indicates effective learning and classification performance. Similarly, the empirical risk exhibits a gradual downward trend as the iterations increase. This reflects the model's progression towards minimizing prediction errors.