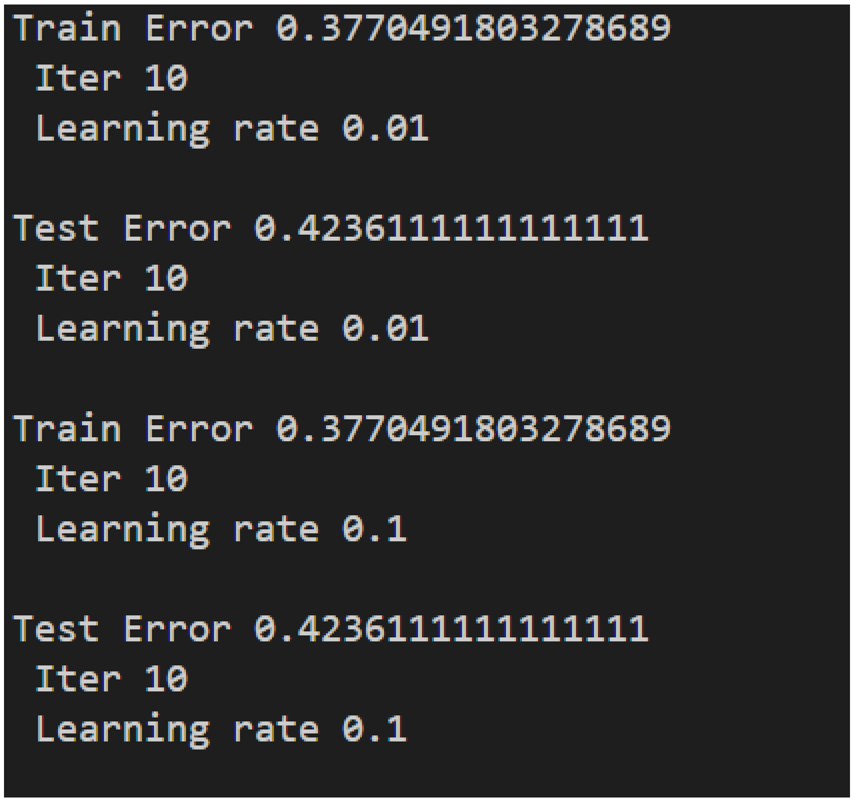
Programming Assignment 1

Members:

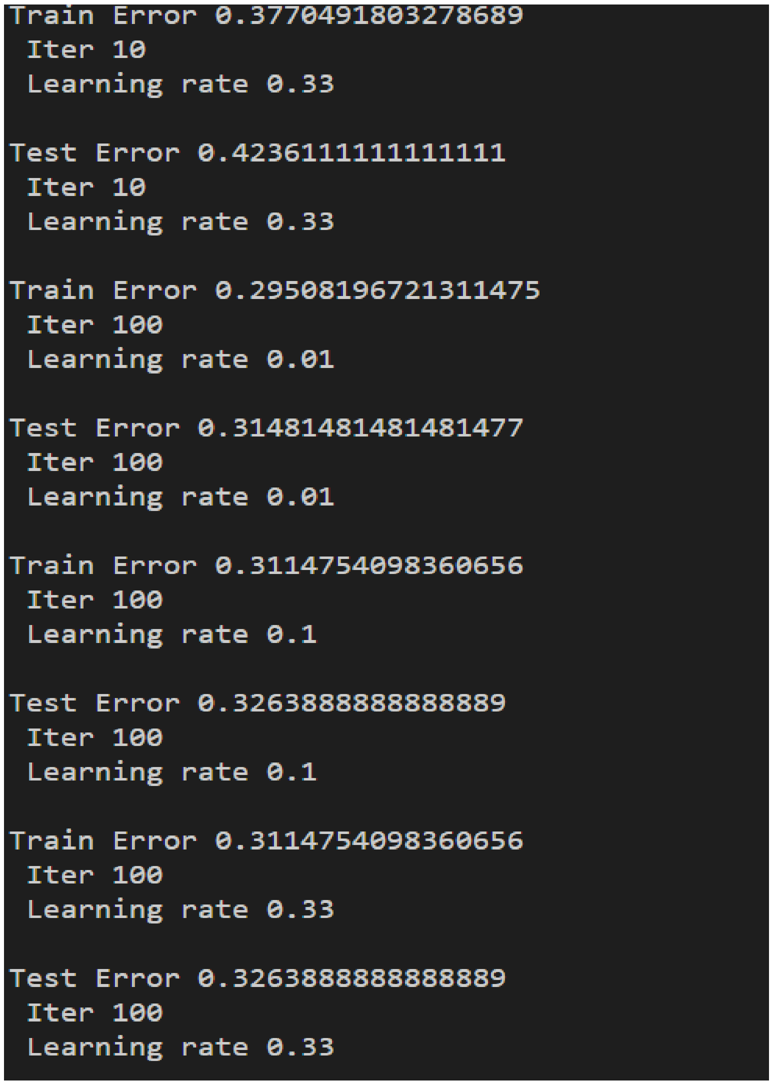
Erik Hale (netid: emh170004)

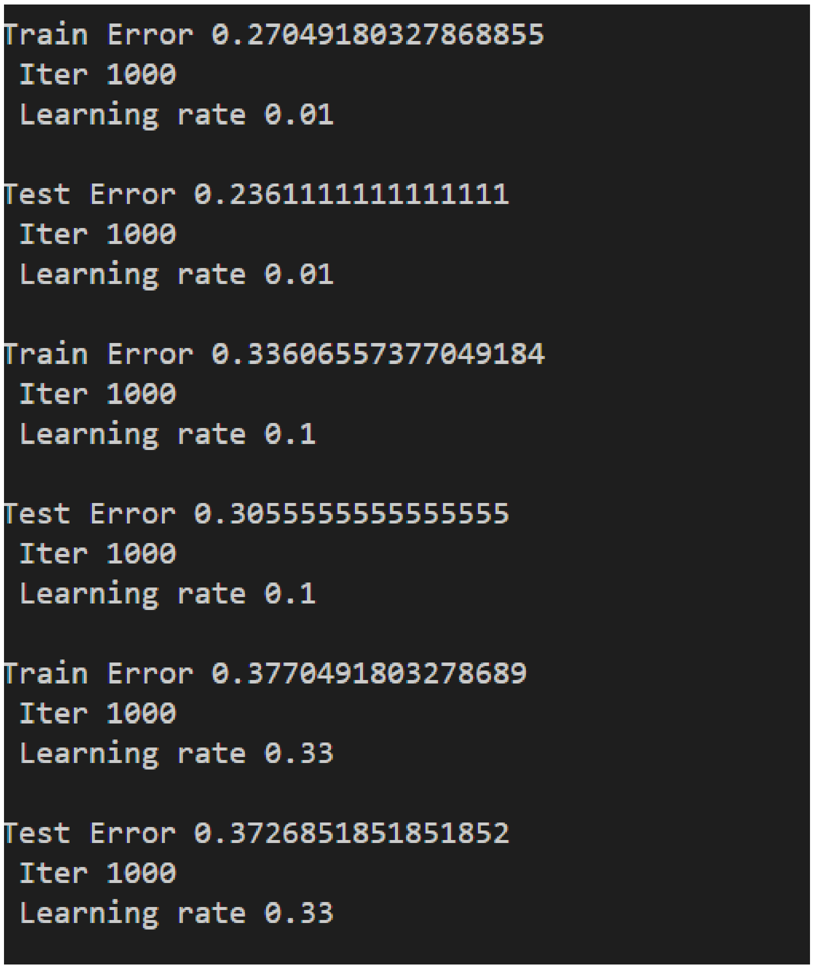
Shiva Kumar(netid: sak220007)

Part a)

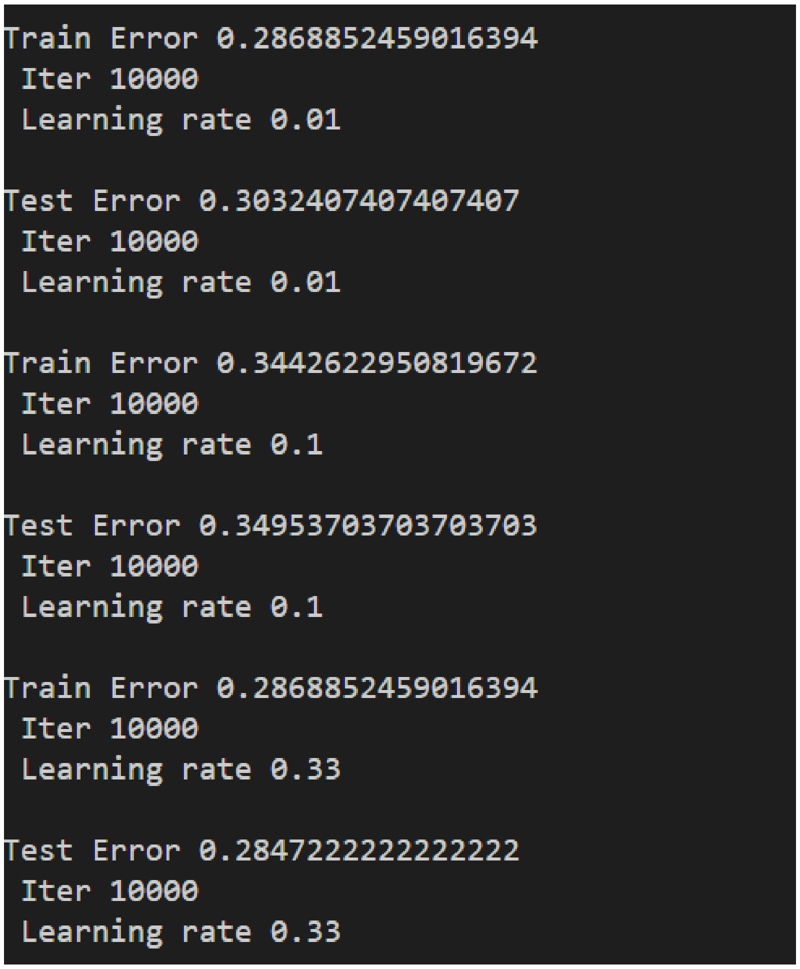


```
Train Error 0.3770491803278689
 Iter 10
 Learning rate 0.01

Test Error 0.4236111111111111
 Iter 10
 Learning rate 0.01

Train Error 0.3770491803278689
 Iter 10
 Learning rate 0.1

Test Error 0.4236111111111111
 Iter 10
 Learning rate 0.1
```


```
Train Error 0.3770491803278689
 Iter 10
 Learning rate 0.33

Test Error 0.4236111111111111
 Iter 10
 Learning rate 0.33

Train Error 0.29508196721311475
 Iter 100
 Learning rate 0.01

Test Error 0.31481481481481477
 Iter 100
 Learning rate 0.01

Train Error 0.3114754098360656
 Iter 100
 Learning rate 0.1

Test Error 0.3263888888888889
 Iter 100
 Learning rate 0.1

Train Error 0.3114754098360656
 Iter 100
 Learning rate 0.33

Test Error 0.3263888888888889
 Iter 100
 Learning rate 0.33
```



```
Train Error 0.27049180327868855
 Iter 1000
 Learning rate 0.01

Test Error 0.2361111111111111
 Iter 1000
 Learning rate 0.01

Train Error 0.33606557377049184
 Iter 1000
 Learning rate 0.1

Test Error 0.3055555555555555
 Iter 1000
 Learning rate 0.1

Train Error 0.3770491803278689
 Iter 1000
 Learning rate 0.33

Test Error 0.3726851851851852
 Iter 1000
 Learning rate 0.33
```



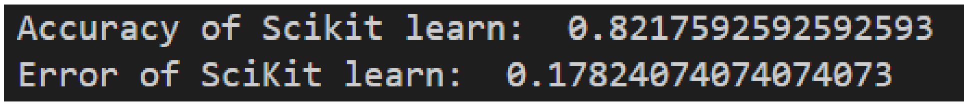
```
Train Error 0.2868852459016394
 Iter 10000
 Learning rate 0.01

Test Error 0.3032407407407407
 Iter 10000
 Learning rate 0.01

Train Error 0.3442622950819672
 Iter 10000
 Learning rate 0.1

Test Error 0.34953703703703703
 Iter 10000
 Learning rate 0.1

Train Error 0.2868852459016394
 Iter 10000
 Learning rate 0.33

Test Error 0.2847222222222222
 Iter 10000
 Learning rate 0.33
```

The best parameters are when the learning rate is 0.01 and the iteration is 1000. We can conclude this because this is when the test error is the smallest.

Part b)

Check out emh170004_sak220007_model_1.obj file.

Part c)



```
Accuracy of Scikit learn:  0.8217592592592593
Error of SciKit learn:  0.17824074074074073
```

For the SciKit Learn’s Logistic Regression model, the default values differ slightly from the values we chose to get a better error than we were able to. The Model that SciKit Learn provides has only 100 iterations compared to our 1000, but the Scikit Learn's version uses Regularization by default.

Regularization (more specifically L2 Regularization which is the default for SciKit Learn) is a process that prunes the classifiers weights so that they stay small, and if they get too small, treat them as redundant and take out the weight. The rationale behind this is that models can get overfit very easily with lots of features with large weights, so using the L2 regularization method keeps the weights in check. The constant in the L2 regularization is known as lambda which is 1/‘C’ in SciKit Learn’s default parameters and is equal to 1.0 (Thus lambda is equal to 1.0). This means that if we want the strength of the regularization to grow, we need to increase the lambda value.

Also, the SciKit learn model makes use of tolerance which we do not use in our model. The model stops looking for better weights when it reaches a certain tolerance.

Interestingly, the fit_intercept variable in the scikit learn model is set to true which specifies that a constant should be added to the decision function. The extra column of ones we added is most likely what was affected by the fit_intercept variable.

SciKit Learn uses much more advanced methods to predict whether a set of features is a 1 or a 0 and thus has a better error rate than our model.

Part d)

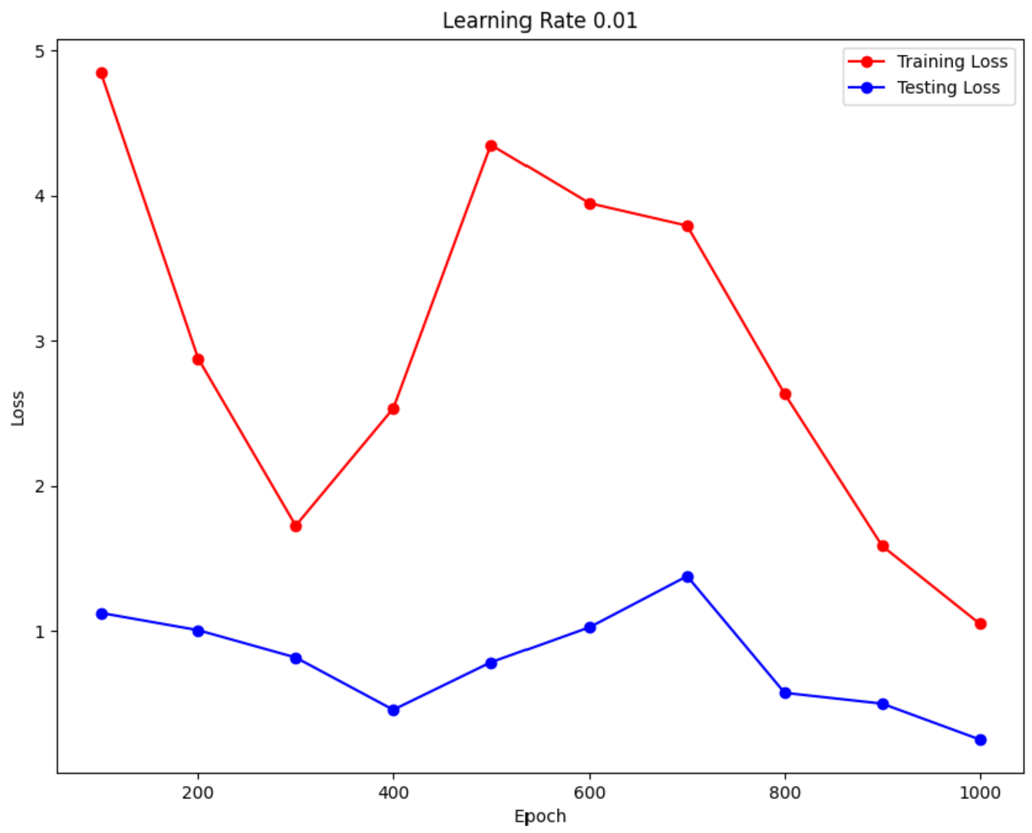
Red Lines: Training loss at every 100th Iteration

Blue Lines: Test loss at every 100th Iteration

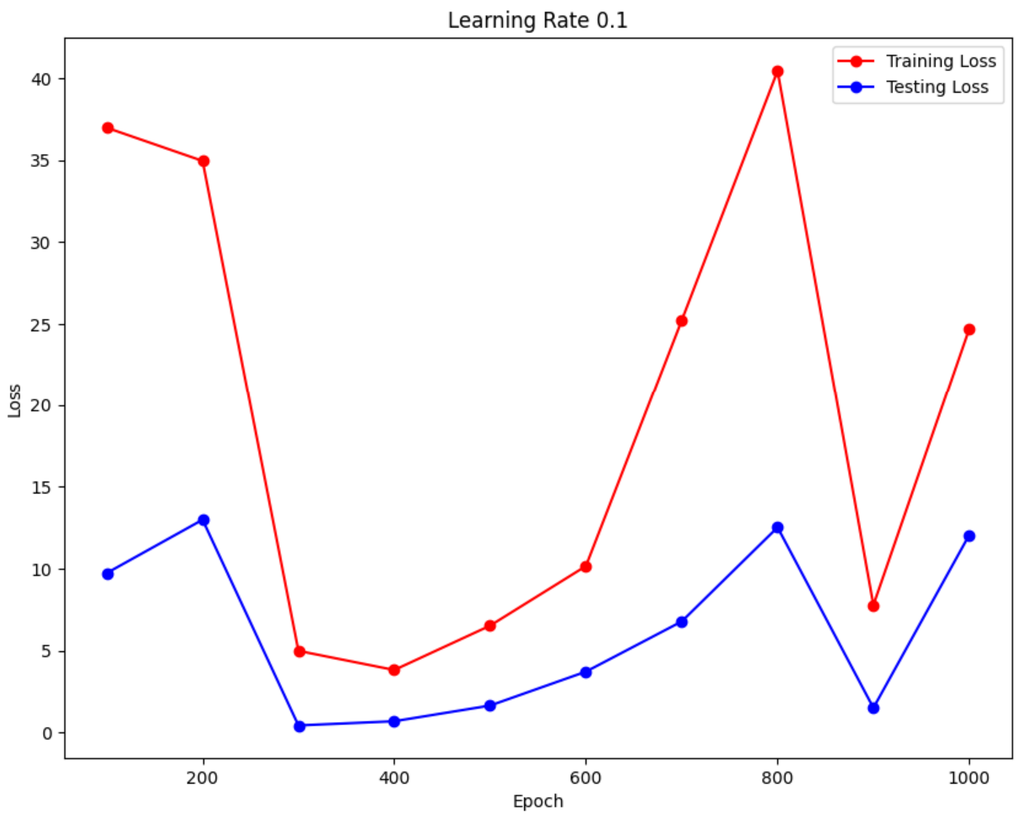
Y-axis: Loss

X-axis: Epoch number

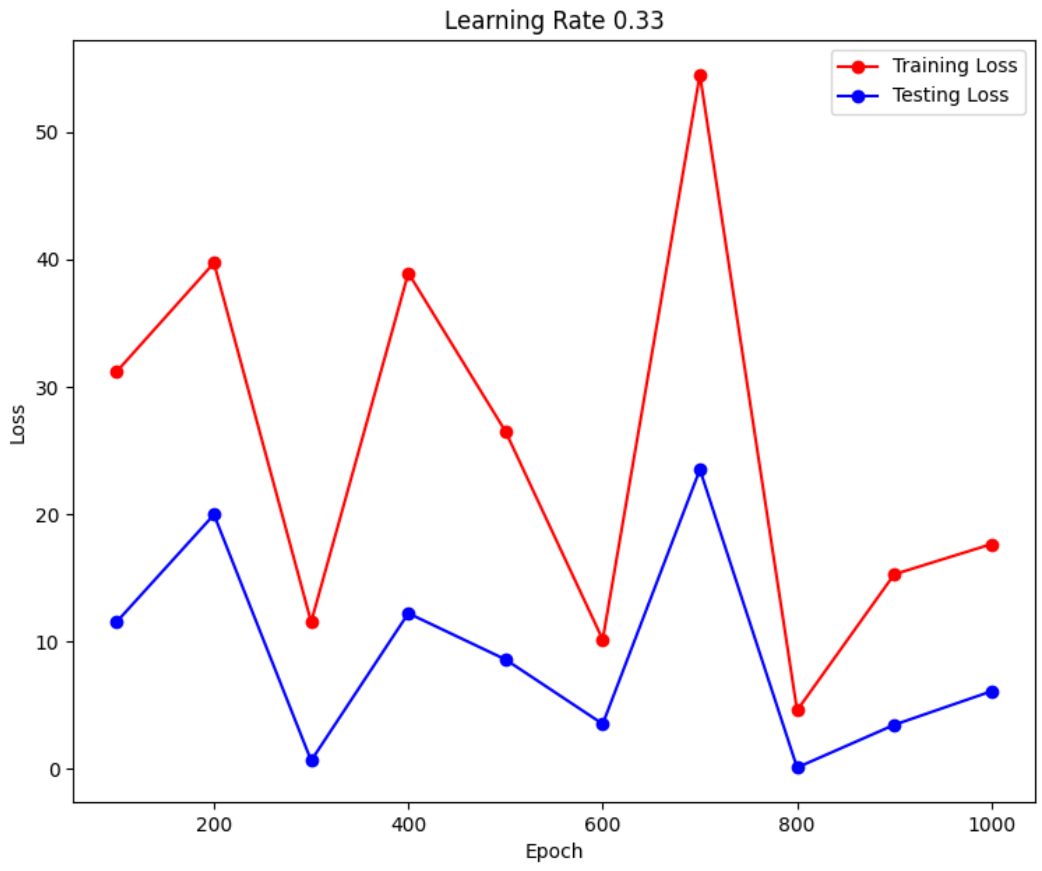
Learning Rate = .01



Learning Rate = 0.1



Learning Rate = 0.33


Learning Rate 0.33
Training Loss
Testing Loss
Loss
Epoch