First, I will summarize my process. I started by loading in the training data file, then I created the mappings for the Booleans, month, and visitor predictors. I then changed these predictors to strings, used .upper or .lower to match how I described them in the mappings, and then mapped them. Then I had to deal with outliers, which would prove to be the hardest part of the project.

In my first try at this project, I did not consider outliers, and everything worked fine until I got to the scaled datasets, and I was getting losses of over 5 million. But after realizing there were some extreme outliers, I removed them and kept trekking along. I then defined the scaling functions, and then tested 9 different models of different amounts of layers and different number of units per hidden layer. I found a model that balanced low loss and not too high complexity and continued with it.

Using this chosen model, I then tested the unscaled training data, getting a loss of around .275. Through my trial and error processes to this point, I had found that the best losses I had seen were roughly .3 or slightly below. So from this I found that .275 was a pretty good loss for the training set.

Next, I used the model on the unscaled test data, getting a loss of roughly .36. This shows that my model has room for improvement, however, human behavior is very hard to model, so this loss is not too bad.

Satisfied with the results thus far, I tested my model on feature scaled training sets. I found training losses of ~.24, ~.23, ~.21 for the minMax, meanNorm, and z score scaled sets, respectively. Then testing on the scaled test sets, I found losses of ~.28, ~.26, ~.29 in the same order. So, I chose mean normalization as my chosen feature scaling method due to the low losses.

After choosing mean normalization scaling, I made a regularized model of the same size and same units per hidden layer, and tested my mean normalized scaled training data, and got a training loss of ~.43 and a test loss of ~.44. I was very disappointed to see this, as thus far all losses have been lower than this. I have now come to realize that the added penalty term involved with the regularization will increase the loss, but in theory should reduce the test loss.

Overall, I am very happy with how my model performed. Throughout the process, my losses purely decreased, some even by millions. I feel confident that a roughly .26 loss for my mean normalized, unregularized test error is good and that I achieved what I sought out to do.