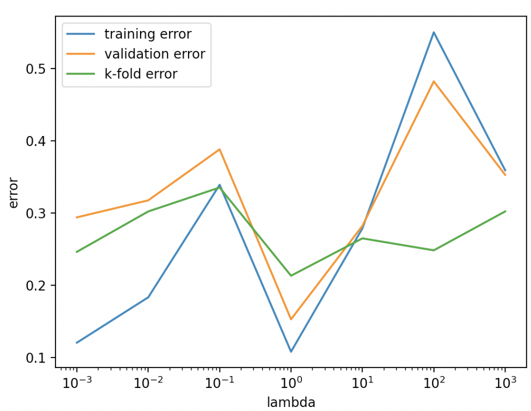
Project Report

- At the start of my train function, I initialized the weights as zero. Then, for each epoch, I shuffled the data and split it into different batches. Looping through each batch, I summed up the gradient by iteratively adding the sigmoid function output of the dot product of the weights and example. After iterating through the entire batch, I divided the gradient by the batch size (length of Y batch vector). I then added the regularization factor 2 * lambda * the weight vector. I then updated the weights in a similar fashion to as HW3. The main difference from HW3 is that if the weights are large in magnitude, then the regularization factor will dramatically increase the magnitude of the gradient. Thus, the regularization factor guides the model towards smaller weights.

- Regularization would likely have improved accuracy for the census logistic regression. The regularization discourages the model from overfitting so it would have improved the estimation error.

- Based on the graph, it appears that a lambda value of 1 generally results in the lowest error across all types.



- I think that it would be a bad idea to repeat patients across the training and validation sets even if they do exist as different data points. The point of the validation set is that we can properly adjust our hyperparameters before showing the model the test set. If the same patient appears in both the training and validation sets, this might cause us to inappropriately tune the hyperparameters. To remedy this, I would split up the dataset by patient into training, validation, and test sets. I would try to get as close to the 70-15-15 split as possible but I would guarantee that the model would see entirely different patients when it saw the training and test sets.