Problem 1

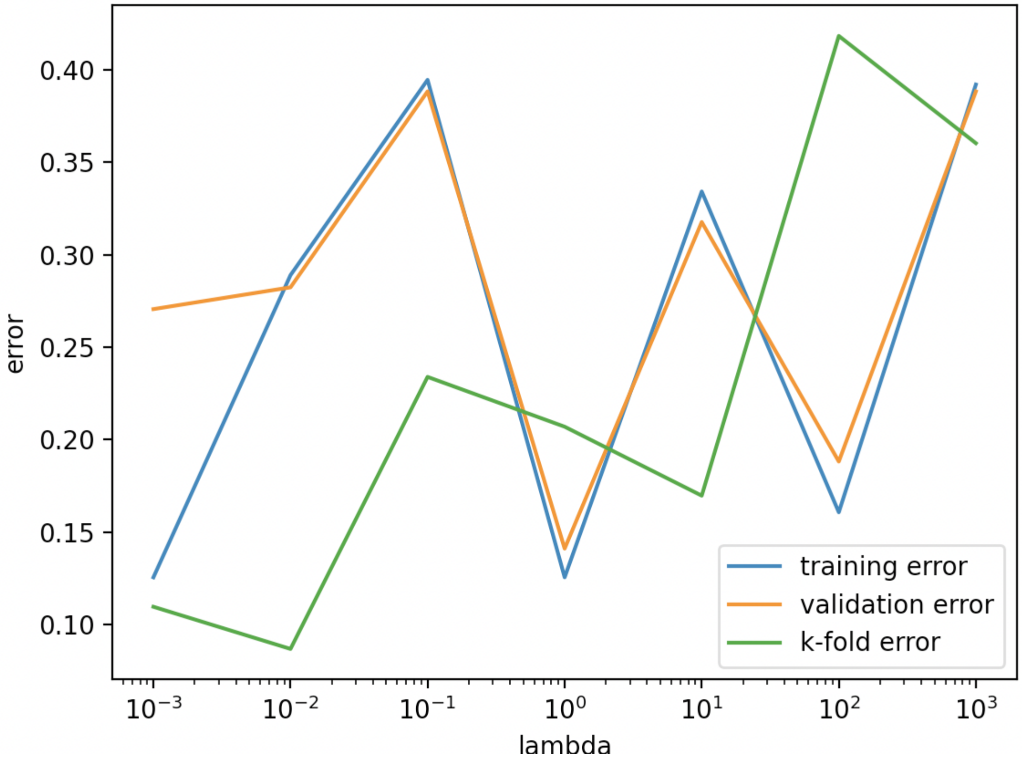
To learn the weights using batch stochastic gradient descent, the program had a smaller learning rate of 0.00001. Moreover, within the train() function of the program, as part of this form of gradient descent, a constant is added to the weights when they are updated at the end of the iteration over the batches. This constant– that being, $2 \times \lambda \times w$, where w denotes the weights, is added in order to 'punish' the hypotheses that are much too complex. That is, by having this constant be added to the weights, this prevents the L2 regularization from growing too large.

Problem 2

When utilized, Tikhonov regularization lowers estimation error, so that the model would be less likely to overfit. Therefore, had I used Tikhonov regularization, the model being trained on the Census dataset would have been less likely to overfit, as well has have had a lower testing error, and increased accuracy. This is because regularization ensures that errors don't grow too large (as previously mentioned), therefore ensuring that all of the features are weighed in a more even manner. Although it may lead to an increase in approximation error, estimation error would be reduced, as well as any chances of the model overfitting.

Problem 3

The best value of lambda is 1 (or, 10^0 in the plot), which gives a train accuracy of 0.8793969849246231, and validation accuracy of 0.8588235294117647. The reason that I chose this lambda was because it lowered both validation error and training error, while at the same time maximizing validation accuracy. Therefore, I concluded that it was the most optimal lambda that could be utilized.



Problem 4

Yes, if each patient had multiple samples entered into the dataset, then this would need to be accounted for when splitting the train-validation test data. Otherwise, there would be double the amount of data for each patient, which would violate our initial assumption that the data is independently and identically distributed (i.i.d), given that certain data points would be dependent on each other. In essence, this would cause the model to not generalize, given that it would be assuming that one patient would be more likely to be chosen over another, simply because they would have more medical samples associated with them. To fix this, each patient could be assigned a unique medical ID. This would ensure that the splits of data would occur based off of each patient, rather than example (since each example can have multiple samples).