Project Report
Guiding Questions

- Briefly explain how you used batch stochastic gradient descent with regularization to learn the weights.

For 5000 iterations, I shuffled the X and Y arrays to randomize their orders and divided them into batches. And for each batch, I calculated the average loss and with regularization of 2 * self.lmbda * weights, so that for each weight is penalized with a constant lambda.

- Think back to when you implemented Logistic Regression on the Census dataset. How would it have been different if you applied Tikhonov regularization? Specifically, how would the regularization affect the accuracy and the types of errors?

If I had implemented Tikhonov regularization, the accuracy could increase because the regularization decreases the effect of overfitting, which could have occurred due to features such as gender or race. This accuracy increase is expected to happen due to the decrease in estimation error and increasing approximation error because regularization has a role of penalizing each weight applied to input.

- Use `plotError()`, which we have implemented for you, to produce a model selection curve. Then,conclude what the best value of lambda is and explain why. NOTE: It takes about five minutes to generate a graph. Please set your default lambda in the constructor to your optimal lambda you discovered for TA testing purposes.

It appears that the model performs the best on all training data sets, validation data set, k-folds when lambda is set to '1'. This is because when lambda is too large, the effect of penalizing each weight is so large that the model underfits the data and when lambda is too small, the effect of penalizing each weight is so small that the model overfits the data. I ran my program several times and compared each result to see which values of lambda are likely to minimize all the errors. Then, I figured out the value around 10^(-1) ~ 10 is neither too large nor too small and generates the best possible model among given lambdas.

- In this project, you used validation data to select a model. Suppose that each patient might've had multiple samples (e.g., multiple lab tests or x-rays) collected and entered into the dataset. Would you need to account for this when splitting your train-validation-test data? If yes, how? If no, why not?
(3-5 sentences)

Yes, and this is mainly because of two reasons. One reason is because if one patient submits multiple data while others don't, then there is a chance of extracting more than one example from one patient. This seems trivial at first, but it would violate the i.i.d assumption because multiple data from one patient are likely to be dependent on other data from the same patient. Another reason is based on an ethical thought that sampling multiple data from one patient could possibly result in giving more importance to that specific person as it is more likely that

data of the patient with multiple data are more likely to be selected for training the model and other patients might feel unfair about it.