Andrew Lake
COMP 379
HW 3

I started this assignment by researching various datasets on Kaggle that were meant for classification tasks. I ended up settling on the heart disease dataset that I worked with for a few reasons. To start, the dataset had many features that were very interesting (age, cholesterol level, resting heart rate, etc.). There were a total of 13 features that all seemed quite relevant to predicting heart disease in patients upon first glance. Secondly, all of these features were already numerical and ready for analysis with a classifier. Given that the point of this assignment was to analyze the effect of hyperparameter tuning and to implement a KNN algorithm, I decided to work with this set. I started by splitting the original dataset into training, development, and test sets with splits of 70%, 15%, and 15% respectively. In summary, the dataset contains 13 features with the target feature being 0 for no heart disease or 1 for a patient that does have heart disease.

By using the default parameters of the Logistic Regression from scikit-learn, I saw an accuracy score of 82%. By tweaking the C value to 100 and penalty type to L1 Regularization, I was able to see an accuracy score of 85% with my model. This demonstrated the effects that hyperparameter tuning could have on model performance with a specific dataset and use case.

For the next part of the assignment, I implemented a KNN classifier. My implementation contained two simple methods. The constructor simply assigned the number of neighbors. The predict method looped through the rows in the test set and for each row it would loop through each row in the training set to find the number of nearest neighbors that the given row and contained features had with the training set.

With my implementation of KNN, I received an accuracy of 100%. I studied my solution thoroughly, as I believed I had probably made a mistake since my accuracy was perfect. I will continue to look into my algorithm to find out if there might be an issue that I have not found yet. It might make some sense though that a classifier would be extremely accurate for predicting heart disease in patients, as the features in this dataset are very telling of whether or not a patient would have heart disease. For instance, a patient with high cholesterol and high blood pressure would most likely have heart disease. These two patients would be close neighbors and would lead the classifier to predict that one has heart disease given that the other does. However, I was unable to find a problem with my solution which leads me to conclude that KNN is the best solution for this dataset and scenario. This makes some sense as these features are very telling of whether or not a patient would have heart disease. The test sample size is only 46 as well. However, I still have reason to believe that my implementation is faulty in some aspect as 100% is most likely too good to be true.