CS 434 Assignment 3

John Miller and Brandon Lee

May 1, 2017

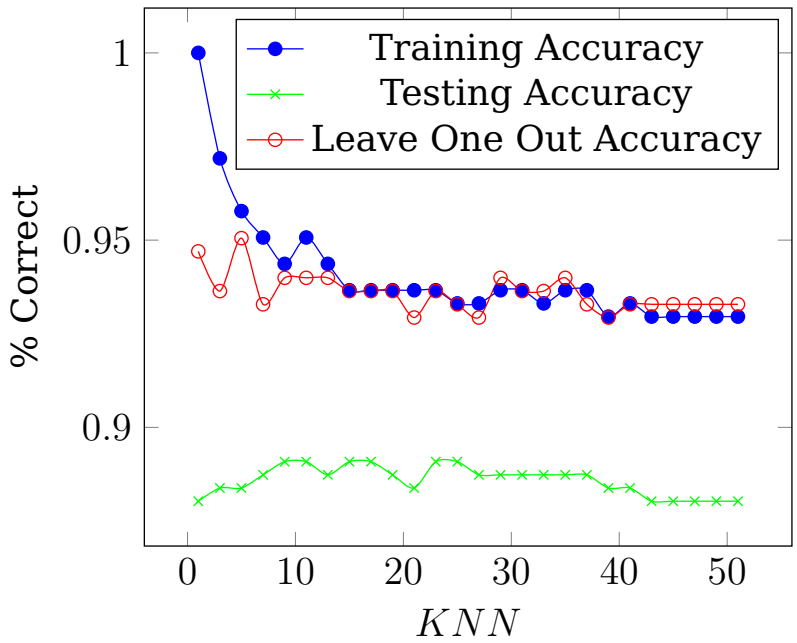
Part I: Model selection for KNN

Building KNN

We implemented the algorithm as described, but ran into performance issues when trying to generate the data for plotting. To get around this we find all neighbors of each point (row) and then when trying out different k values we just grab the k best neighbors from our list. This change massively increased performance because it no longer had to re-calculate distances to every other point for each new k value.

Plotting Error

Figure 1: Accuracy of model by various KNNs



Analyzing Error and Relationships

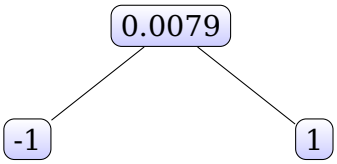
Look at the data it makes sense that the training accuracy for a k value of 1 has 100% accuracy because the nearest neighbor to each training point will be itself. We can also see that the leave

one out cross validation accuracy follows the same trend as the training accuracy. After comparing graphs with some other students in the class it seems like the calculation of the training accuracy is off. This could be due to some error in our normalization, distance calculation, or the way the accuracy for each k value is calculated.

Part 2: Decision tree

Stump

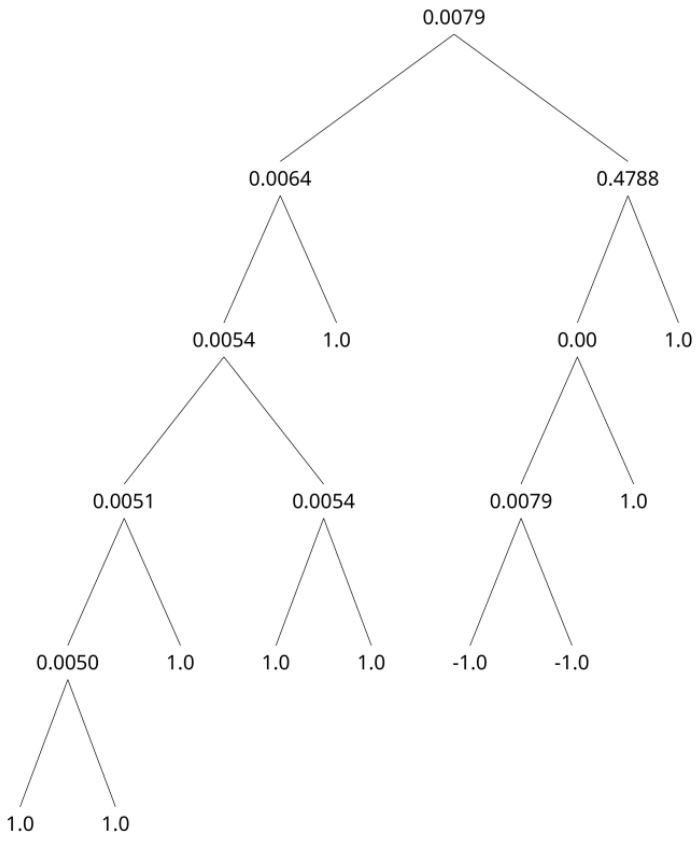
Figure 2: Decision Tree Stump



Information Gain: 0.1478750536602611
Training Error Rate: 0.05633802816901412
Testing Error Rate: 0.0880281690140845

Top-Down Greedy

Figure 3: Decision Tree



Observe that the tree in Figure 3 represents the value of the node at best split of the subsection. The childless leaves represent the majority classifier, [-1, 1]. For additional details such as the specific feature per branch split, please run the project code.

Figure 4: Information Gain by Depth

Depth	Information Gain
0	0.1478750536602611
1	0.0
2	0.0
3	0.0
4	0.0
3	0.0
2	0.0
1	0.11550295857988166
2	0.1069720480391115
3	0.1069720480391115

Training Error Rate: 0.03873239436619713
Testing Error Rate: 0.08450704225352113

Analysis

From our values in both decision stump and decision tree error rates, we observe that the training data has the lower rate of error over the testing data. This is consistent with the logic of the model as it is based off of training data, regardless of depth. As for the values between stump data and tree data, we can see that in both instances of testing and training, the tree error rates are slightly lower. This is due to the additional complexity of introducing further levels for the tree to identify more majority classifiers.