James May

Garrett Bauer

Peter Dorich

Assignment 2 Report

K Nearest Neighbor

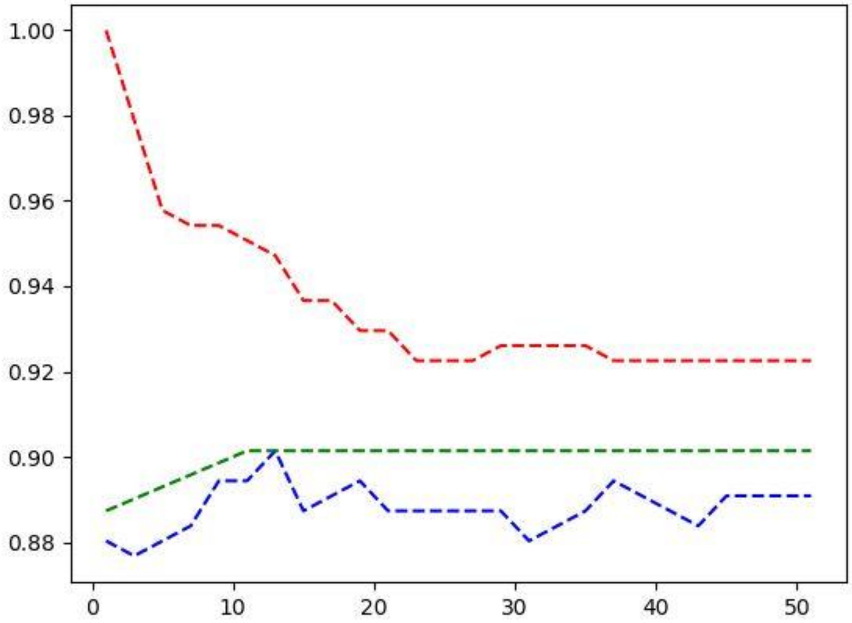

Figure 1: Accuracy is plotted as a function of K, red is training error, blue is test error, and green is testing error of the leave-one-out cross-validation method.

The training error, shown in figure 1, starts out at 100%. Since k equals 1, it is choosing itself as the nearest neighbor and thus is not a good indicator of training error. It can be seen converging at an accuracy of 93% around k = 23. Testing error, shown in figure 1, remains relatively stable around 89%, though it does seem to increase slightly as the training error drops. The cross-validation testing error has a similar, but higher trend in accuracy which converges at 90% after k = 11.

Since the cross-validation error doesn't change after k = 11, it is a good assumption that having a higher k would not be useful.

Decision Tree

Information gain: .109542253521

Training error rate: .0598591549296

Testing error rate: 0.105633802817

When implementing this functionality, both normalized and not normalized data was tested. In both cases the results were almost the exact same, so the normalization was left out.

Top-Down Greedy Induction

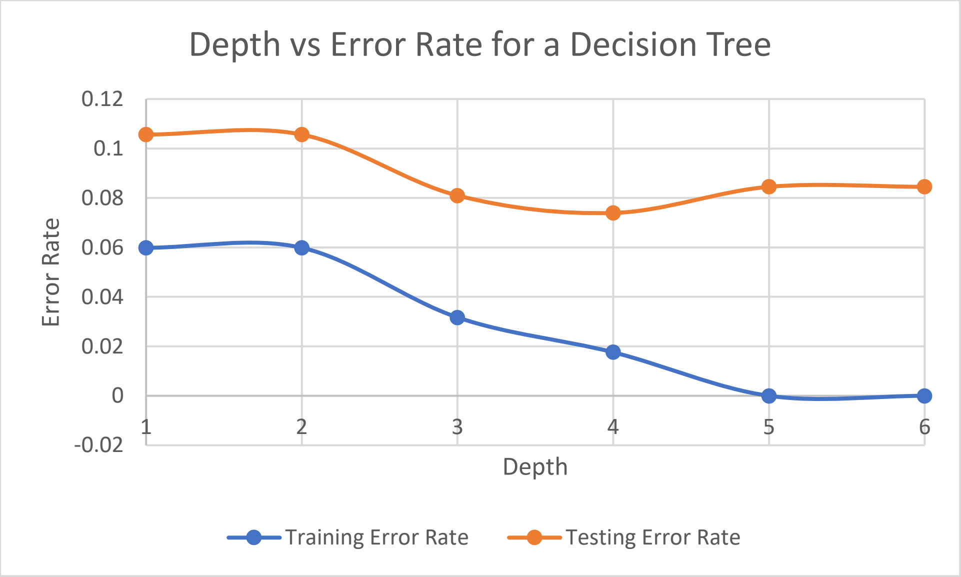

Figure 2: Error Rate is plotted as a function of depth, where blue is training error, and orange is testing error. The data points are recorded in the table below.

The error rate is calculated as the number of incorrect predictions over the total number of data points.

Depth:	Training Error Rate:	Testing Error Rate:
1	0.0599	0.1056
2	0.0599	0.1056
3	0.0317	0.0810
4	0.0176	0.0739
5	0.0	0.0845
6	0.0	0.0845

Table 1: The data points from figure 2 are recorded here.

As you can see from figure 2 and the data table, as the depth increases, the training error rate decreases. However, this does not directly translate to a decrease in the testing error rate. The testing error rate did follow the same trend as the training error rate until a tree depth of 4, where it began to increase. This is likely due to the decision tree overfitting to the training data which leads to a lower training error rate while the real-world performance suffers. The testing error rate was lowest with a tree of depth 4, suggesting that this is probably the best value to use in practice.