James May

Garrett Bauer

Peter Dorich

Assignment 2 Report

Part 1 KNN

|  |
| --- |
| **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.

Part 2, Section 1:

**Information gain: .109542253521**

**Training error rate: .0598591549296**

**Testing error rate: 0.105633802817**

**Thoughts on these values: When I implemented this functionality, I tested using both the data given when normalizing the values, and when not. In both cases the results were almost the exact same, so I left normalization out.**

Part 2, Section 2:

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 | 0.05985915492957746 | 0.1056338028169014 |
| 2 | 0.05985915492957746 | 0.1056338028169014 |
| 3 | 0.03169014084507042 | 0.08098591549295775 |
| 4 | 0.017605633802816902 | 0.07394366197183098 |
| 5 | 0.0 | 0.08450704225352113 |
| 6 | 0.0 | 0.08450704225352113 |

As you can see from the data, the greater the depth, the lower the training error rate becomes. 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 get worse. 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.