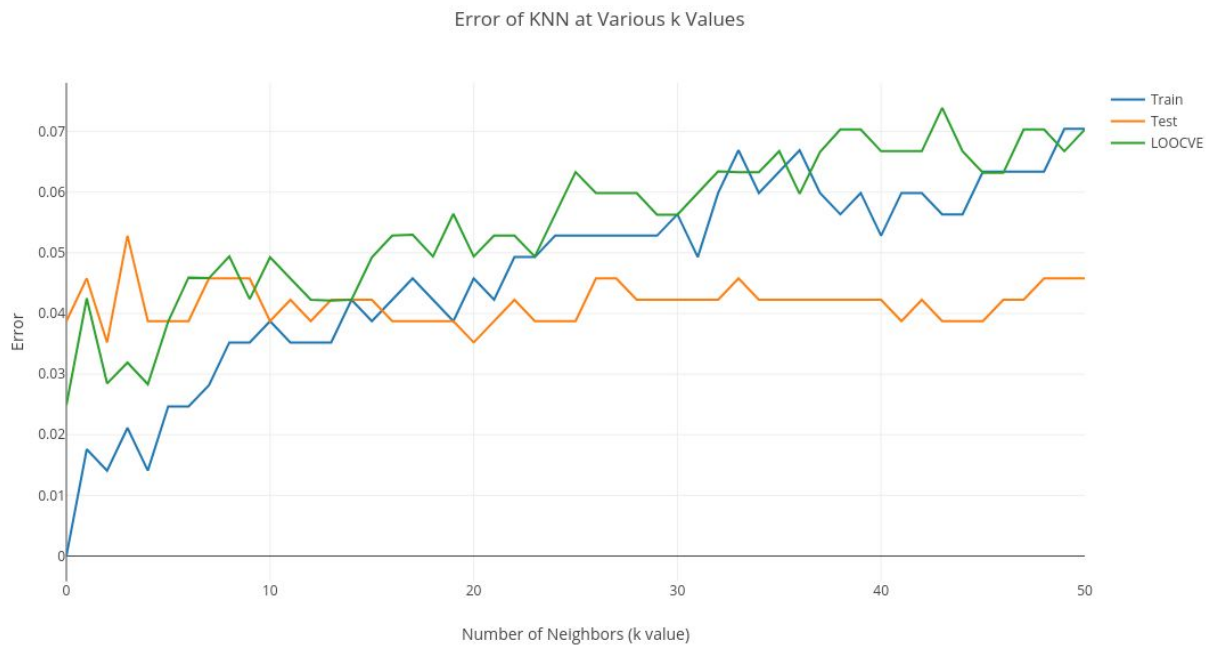
Parker Bruni and Brian Blythe
CS434
4/26/2018

Implementation Assignment 2

PART 1: Model selection for KNN

1.1
(as seen in the submitted python file “knn.py”)

1.2

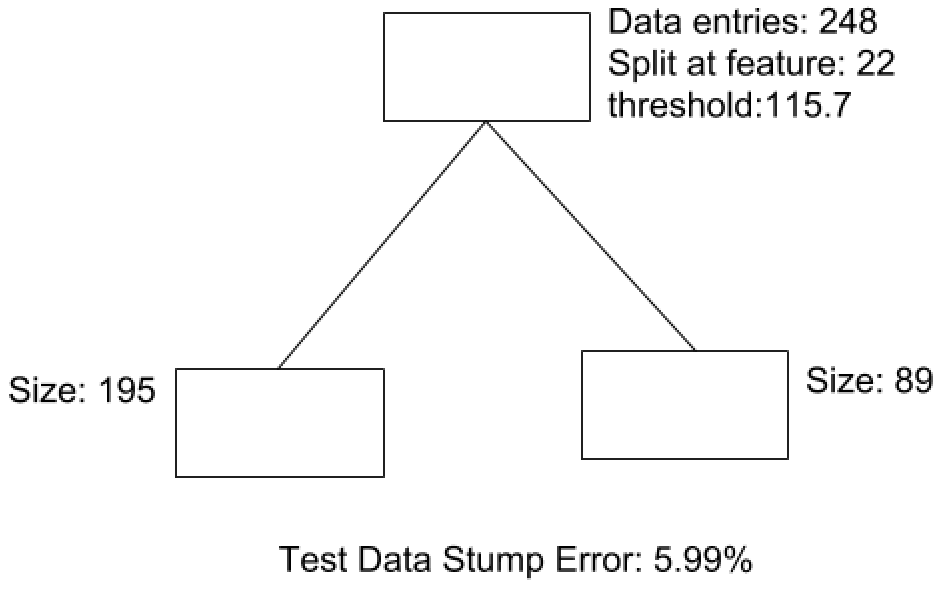


1.3
For the training data, the error was 0% when k = 1, this is because the nearest neighbor on the training data point is itself, so every guess will be 100% accurate for training data at k = 1. The training data steadily increased as k iterated up to k = 50. The testing data error was somewhat consistent for all values of k from 1 to 50, ranging from about 3% error to 5% error. For leave-one-out cross validation, the error was similar to our training error, except that it was slightly greater error at each corresponding k-value point than the training data. We used a fold value of 10 (big K) for the leave-one-out cross validation. The k (for knn) that seems ideal to us is about k = 15.

PART 2: Decision Tree

1.1

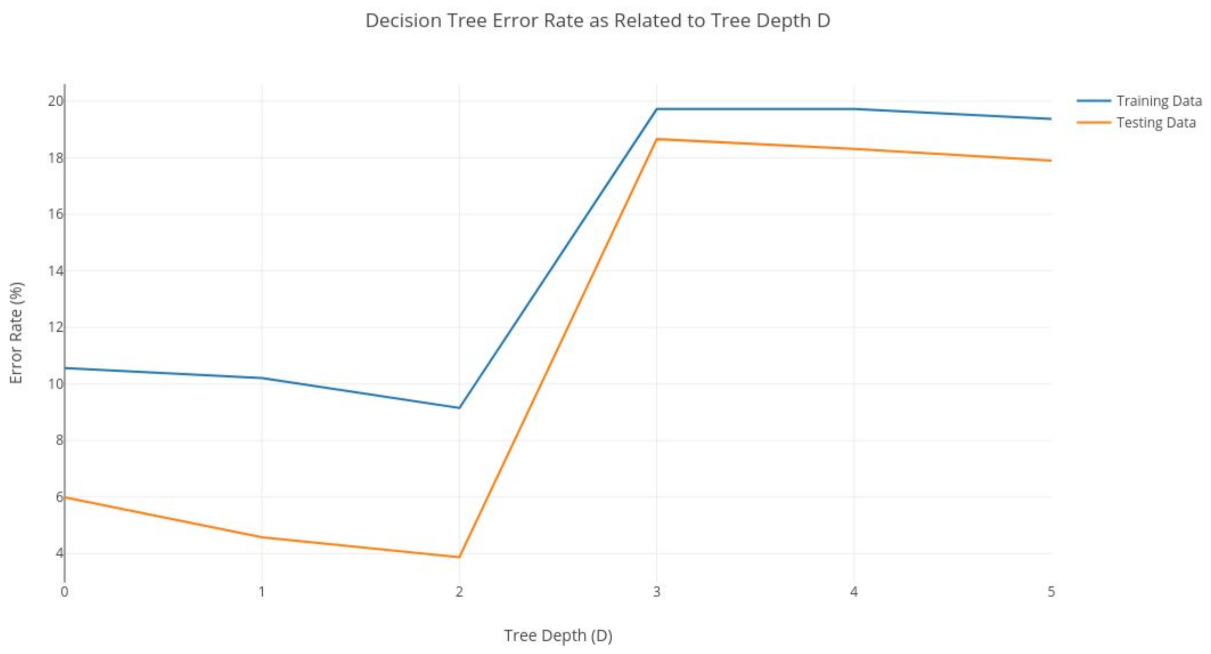
1) **the learned decision stump (be sure to provide the label for each branch)**



2) **the computed information gain of the selected test**
   Information Gain: 0.65
3) **the training and testing error rates (in percentage) of the learned decision stump**
   Training Error: 5.99%
   Testing Error: 10.56%

1.2

	D = 1	D = 2	D = 3	D = 4	D = 5	D = 6
Training Error	5.99%	4.58%	3.87%	18.66%	18.31%	17.96%
Testing Error	10.56%	10.21%	9.15%	19.72%	19.72%	19.37%



The behavior that we observe is that the the training data is more fit and has less error than the testing data but we still achieve decent error rates for both. The training data starts at about 6% for d = 1 and steadily gets better up to d = 3, where it jumps up to about 18% error rate and continues to get steadily better up to d = 6. Our testing data followed a similar trend but with slightly more error at all points than the training data. The more error on the testing data is to be expected as the model and thresholds were determined using the training data, but jump in error may be due to an incorrect selection of the feature that would give us the most gain at that level of the tree or an incorrect selection of threshold for that level and higher. All other selections appear to be correct as the tree gets steadily better from d = 3 and deeper.