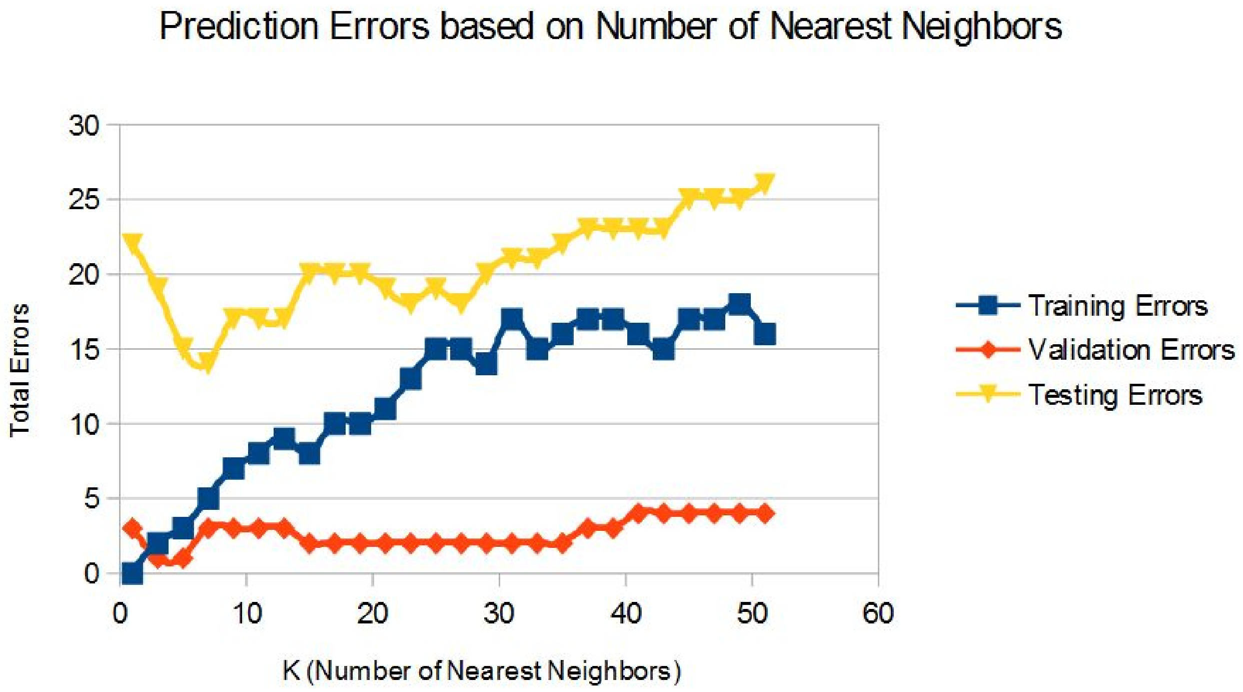
David Sahni, Burton Jaursch, Chase McWhirt
April 24nd, 2019
CS 434: Machine Learning and Data Mining

Assignment 2: Model Selection for KNN & Decision Tree

1. Model Selection for KNN

There are a few considerations to be made about which model to select. First, the data itself appears to rise fairly linearly over time. This is true of both training errors and testing errors. The relationship suggests there is a lot of overlap between benign and malignant cancer cells, which is to be expected.

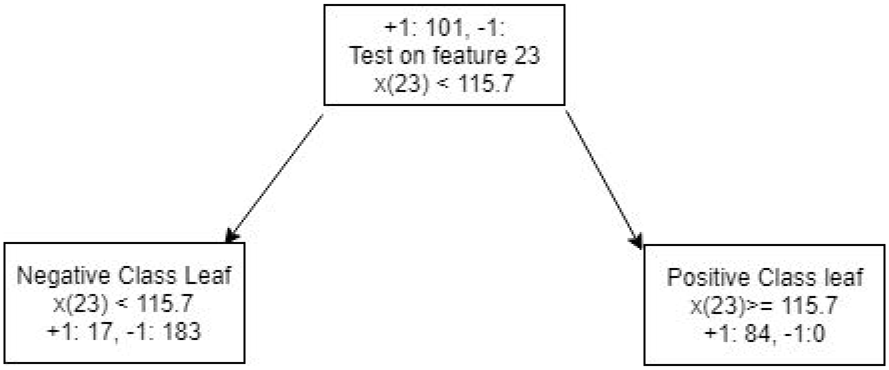
To select a model, we should first consider the two columns where k is 5 and k is 7. When k is 5, we can see the lower point where the combination of the three lie. However, point 7 has the minimum testing error without increasing training or validation errors too dramatically. Choosing either of these might lead to an over fitted model. If we choose the lower error combination that maximizes k, the best suggest would be k is 27. It's at this point that around 10% of the data is informing the prediction which means k is large enough to produce a good general model.



2. Decision Tree

Part 1:

Learned Decision stump:



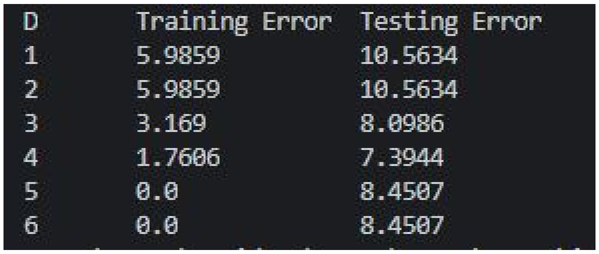
Information Gain: 0.64354

Training Pct Error: 5.986%

Testing Pct Error: 10.563%

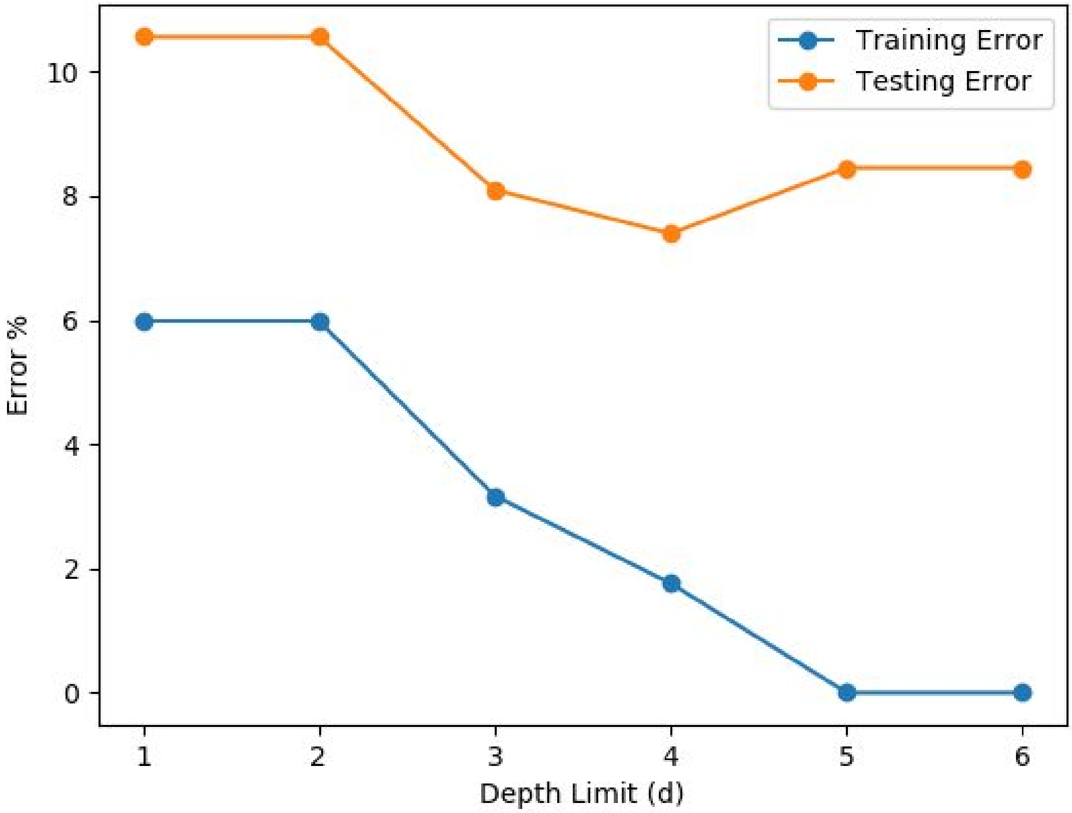
Part 2:

Data table of Training/Testing Error by depth limit:



```
D       Training Error  Testing Error
1       5.9859          10.5634
2       5.9859          10.5634
3       3.169           8.0986
4       1.7606          7.3944
5       0.0             8.4507
6       0.0             8.4507
```

Plot of Training/Testing Error as a function of D:



Observations:
We can see the training error decrease as the tree is allowed to grow deeper, which is expected as the algorithm will allow it to perfectly learn training data set. We can see this as the depth limit gets to 5 and 6 as the algorithm has created enough tests to represent every decision boundary in the training set. Regarding the testing error, we see it dip as the depth limit increases and the learned decision tree becomes more complex, however it rises again as the training error approaches zero. This is the result of overfitting the training data as the algorithm learns to test for each individual boundary in the training set.