Cody Hubbard 004843389
CSM146, Winter 2018
Problem Set 1

Problem 1

(a) Solution: The best one leaf decision tree should label every example "one", this is due to the nature of the target function we want to learn. With this method the only incorrect labels will be those with with three specific features, X_1, X_2, and X_3 all equal to zero. Since all errors must have these three features with value zero that leaves 2^{n-3} other feature choices for errors. Thus the error amount is 2^{n-3}. Since there are 2^ntraining examples and 2^{n-3} of those will be labeled with errors that gives and error rate of $\frac{2^{n-3}}{2^n} = 2^{-3} = \frac{1}{8}$.

(b) Solution: No, all decision trees with one internal node will still have to label every example as "one" regardless of what feature you split one. This is again because of the number of "one"s given by our target function. No matter what node you split on, even if its X_1, X_2, or X_3 both of the new subsets will have a majority of "correct" labels being "one", making the optimal decision to just label everything as "one". This results in an identical error rate of $\frac{1}{8}$.

(c) Solution: Entropy =

$$\begin{aligned} E(S) &= -\left(\frac{1}{8} \cdot Log\frac{1}{8} + \frac{7}{8} \cdot Log\frac{7}{8}\right) \\ &= -\left(-.11 - .05\right) \\ &= .16 \end{aligned}$$

(d) Solution: A split on any of the three critical features (X_1, X_2, X_3) should reduce the entropy by a nonzero amount.

$$\begin{aligned} E(S) &= \frac{1}{2}\left(0\right) + \frac{1}{2}\left(-\left(\frac{1}{4} \cdot Log\frac{1}{4} + \frac{3}{4} \cdot Log\frac{3}{4}\right)\right) \\ &= \frac{1}{2}\left(-\left(-.15 - .09\right)\right) \\ &= .12 \end{aligned}$$

The new conditional entropy is .12, thus the split resulted in a loss in .04 entropy.

Problem 2

Solution: Need to show that the information gain is zero, that is

$$Gain(S, A) = Entropy(S) - \sum_{v \epsilon Values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S) = 0$$

So using this formula we can plug in our info and

$$Gain = B\left(\frac{p}{p+n}\right) - \sum_{i=1}^{k} \frac{|p_i + n_i|}{|p+n|} B\left(\frac{p_i}{p_i + n_i}\right)$$

And we know that $\sum p_i = p$ because p_i is just p split into k disjoint subsets, this also holds for $\sum n_i$ thus by distributing the summation the gain equation becomes,

$$\begin{aligned} Gain =& B\left(\frac{p}{p+n}\right) - \frac{|p+n|}{|p+n|} B\left(\frac{p}{p+n}\right) \\ =& 0 \end{aligned}$$

as to be proved.

Problem 3

(a) Solution: If a point can be its own neighbor $k = 0$ should give a training set error of 0 minimizing the training set error as required.

(b) Solution: Large values of k will be bad in this data set because they can lead to easier miss-classification. There are not a lot of points in this data set so if k is too large you will start considering points which are not really "neighbors" because all the real neighbors will have been exhausted.

Small values of k will be bad in this data set because it will lead to over-fitting as every point will have a complex nearby decision boundaries.

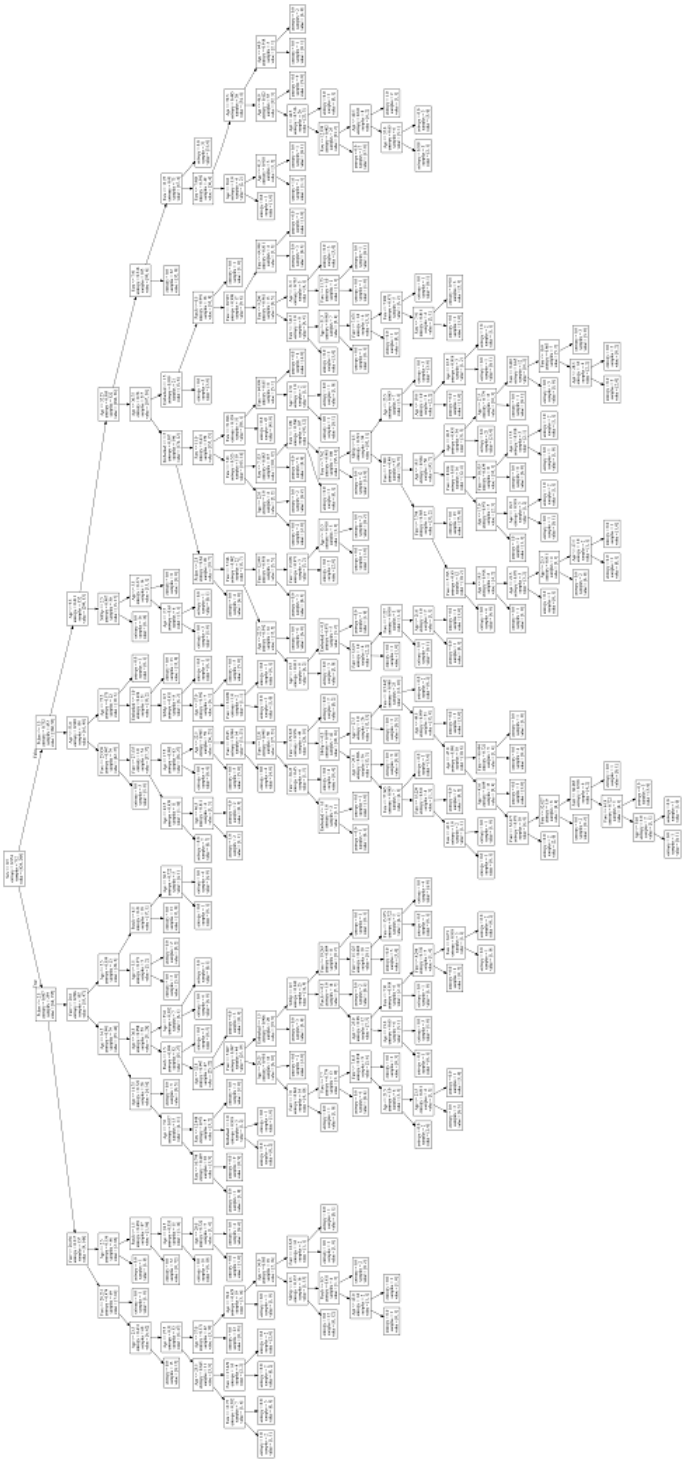
(c) Solution: $k = 5$ minimizes leave-one-out cross-validation error for this data set. This causes an error of $\frac{2}{7}$.

Problem 4

(a) Solution: For pclass 3rd class members had a drastically higher mortality rate. For Sex more females survived. For age, small children (age < 10) had the only only distribution where survived $>$ deceased. For siblings/spouses aboard those with more spouses/siblings had a higher chance of survival. For parents/children aboard those with more parents/children had a higher chance of survival. For fare those who paid more seemed to have much higher survival ratios, which probably correlates to class as well. For embarked those who came from "0" had the only positive survival ratio.

(b) Solution: I got the correct error of 0.485.

(c) Solution: The training error of my DecisionTreeClassifier with the criterion set to entropy was 0.014



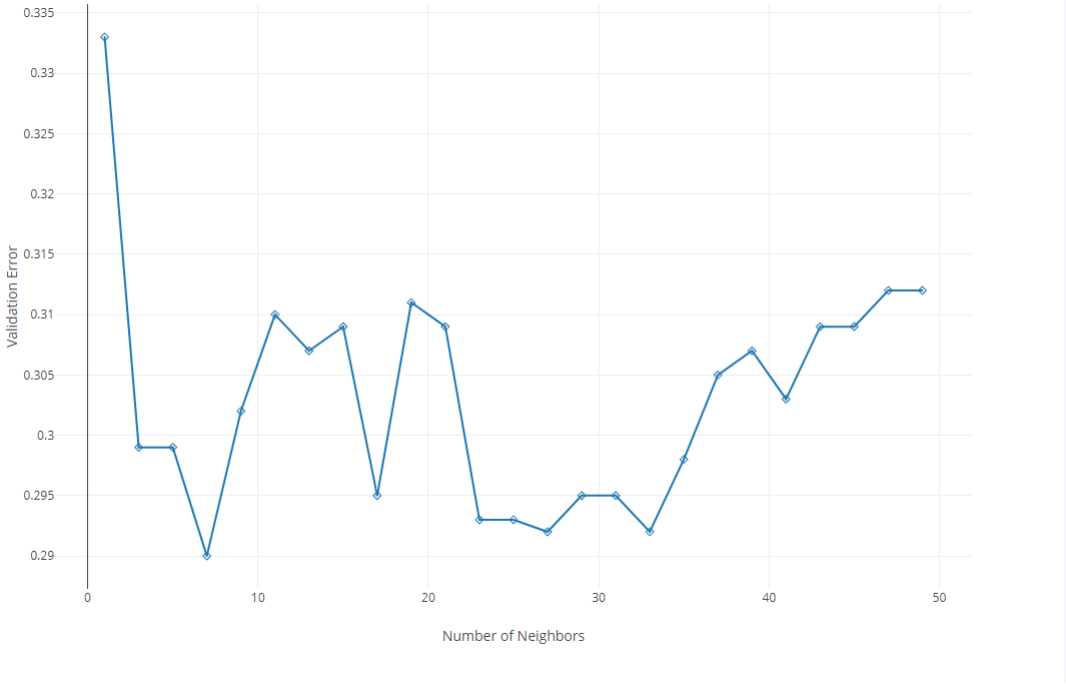
The generated tree is as follows

(d) Solution: The error of my three nearest neighbor classifiers is as follows:
–For 3-Nearest Neighbors: 0.167
–For 5-Nearest Neighbors: 0.201
–For 7-Nearest Neighbors: 0.240

(e) Solution: The average training and test error of each of my classifiers on the Titanic data set is as follows:

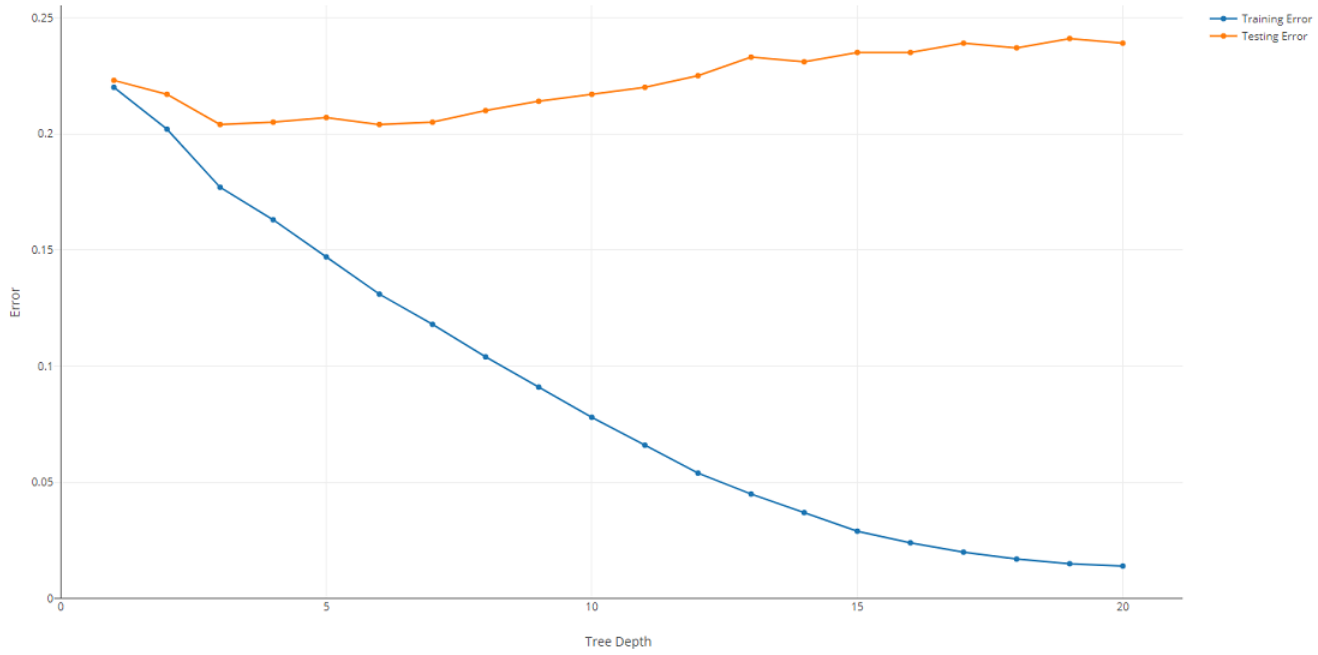
For Majority Vote classifier:
– training error: 0.404, testing error: 0.407
For Random classifier:
– training error: 0.489, testing error: 0.487
For Decision Tree classifier:
– training error: 0.012, testing error: 0.241
For 5-Nearest Neighbors classifier:
– training error: 0.212, testing error: 0.315

(f) Solution:



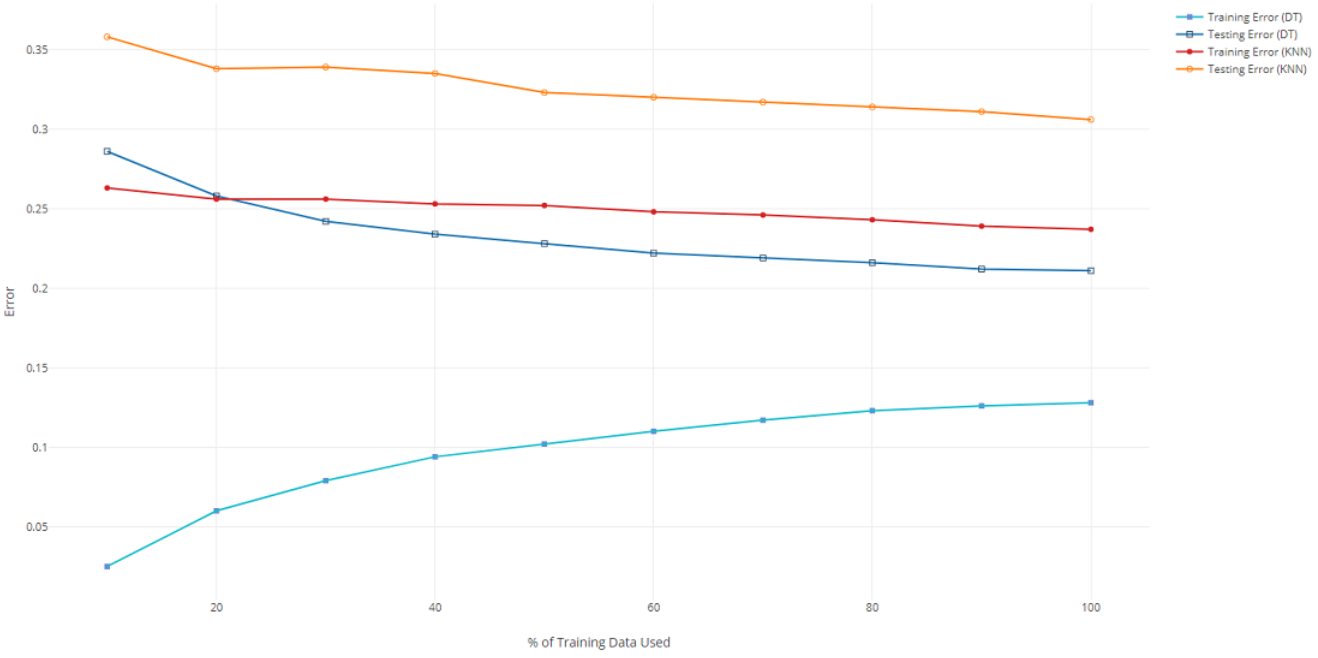
The results above show that there is a decent amount of optimaility in the 23-33 nearest neighbors range, after that the error slowly climbs as you would expect as overfitting takes over. Additionally the early values are all over the place, the 1-21 nearest neighbors range has wild changes in error. The graph shows an optimal K for K-Nearest-Neighbors as 7-neighbors with a validation error of 0.290.

(g) Solution:



From the above graph you can see that the best depth limit to use for the data is 6, which gives an approximate testing error of 0.204. You can see overfitting in these results. As the depth of the tree increases the training error decrees as it should, however the testing error increases steadily, this is caused by overfitting. the tree becomes more complex and the model has a harder time classifying new data correctly.

(h) Solution:



From the above data you can see that in almost all cases the error decreases as more training data becomes available, this makes sense as our models should both become more accurate with more training since we already isolated the optimal depth for our decision tree (6) and the optimal amount of neighbors for our K-nearest neighbors algorithm (7-neghibros). However there is some sort of odd behavior with the

Decision Tree's training error. As the amount of data increases the error increases. Ii am unsure if it is due to an error in my programming or if this is correct behavior.