Problem Set 1: Decision Trees, Nearest Neighbors

Bonnie Liu
UID: 005300989
Discussion 1A (Ulzee An)

Due Date: January 25, 2022

For this problem set, I collaborated with Henry Li, Hannah Zhong, David Xiong, Justin He, and Nicholas Dean.

1 Splitting Heuristic for Decision Trees

1.1

A decision tree makes decisions based on the majority of labels. If $n \geq 4$ and $Y = X_1 \vee X_2 \vee X_3$, it does not matter how big n is as long as $n \geq 3$ because our target function only relies on those three variables. Mistakes will only be made when $X_1 = X_2 = X_3 = 0$. The proportion of possible examples that fit this criteria so long as $n \geq 3$ is $\frac{1}{8}$. Thus, the best 1-leaf decision tree makes $\frac{2^n}{8}$ mistakes.

1.2

Splitting by anything other than X_1, X_2, and X_3 would make no difference since the proportion of mistakes would stay the same. If we split by some X_i such that $1 \leq i \leq 3$, then we would get 2^{n-1} correctly classified examples for the $X_i = 1$ branch, and for the $X_i = 0$ branch, we would get $\frac{2^n}{8}$ incorrectly classified examples and $3 * \frac{2^n}{8}$ correctly classified examples. Thus, there is not a split that reduces the number of mistakes by at least one.

1.3

$$\begin{aligned} H(Y) &= -P(Y=0)logP(Y=0) - P(Y=1)logP(Y=1) \\ &= -\frac{1}{8}log(\frac{1}{8}) - \frac{7}{8}log(\frac{7}{8}) \\ &\approx 0.5436 \end{aligned}$$

1.4

Yes, there is a split that reduces the entropy of the output Y by a non-zero amount. If we split by any X_i such that $1 \leq i \leq 3$, we get the following:

- For the $X_i = 0$ branch, $P(Y = 0|X_i = 0) = \frac{1}{4}$ and $P(Y = 1|X_i = 0) = \frac{3}{4}$, so its entropy is $-\frac{1}{4}log(\frac{1}{4}) - \frac{3}{4}log(\frac{3}{4}) \approx 0.8113$.
- For the $X_i = 1$ branch, $P(Y = 0|X_i = 1) = 0$ and $P(Y = 1|X_i = 1) = 1$, so its entropy is $-0log(0) - 1log(1) = 0$.

Thus, we can calculate the conditional entropy for splitting on X_i:

$$\textit{conditional entropy} \approx \frac{1}{2}(0.8113) + \frac{1}{2}(0) = 0.4056$$

2 Entropy and Information

2.1

To show $0 \leq H(S) \leq 1$, let's first rewrite $H(S)$.

$$H(S) = B(\frac{p}{p+n}) = -\frac{p}{p+n}log(\frac{p}{p+n}) - (1 - \frac{p}{p+n})log(1 - \frac{p}{p+n})$$

Let x be the proportion of positives in S. Then $x = \frac{p}{p+n}$ and

$$H(x) = -xlog(x) - (1 - x)log(1 - x)$$

In order to find the minimum and maximum of H(x), we first need to find the critical points of H(x) by taking its derivative with respect to x.

$$\frac{dH(x)}{dx} = -log(x) - \frac{1}{ln(2)} + log(1 - x) + \frac{1}{ln(2)} = -log(x) + log(1 - x)$$

If we set this equation to 0, we get

$$-log(x) + log(1 - x) = 0 \Rightarrow log(\frac{1 - x}{x}) = 0 \Rightarrow \frac{1 - x}{x} = 2^0 = 1$$

Notice how when x=0, the above equation is undefined, so x=0 is one of our critical points. Let's solve for x now.

$$1 - x = x \Rightarrow 2x = 1 \Rightarrow x = \frac{1}{2}$$

If we plug in our critical points $x = 0$ and $x = \frac{1}{2}$, we get the following pair of equations:

$$H(x = 0) = 0log(0) - (1 - 0)log(1 - 0) = 0log(0) - 1log(1) = 0 - 0 = 0$$

$$H(x = \frac{1}{2}) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = \frac{1}{2} + \frac{1}{2} = 1$$

Thus, $0 \leq H(S) \leq 1$.

When $p = n$, $H(S) = B(\frac{p}{p+p}) = B(\frac{p}{2p}) = B(\frac{1}{2}) = -\frac{1}{2}log\frac{1}{2} - (1 - \frac{1}{2})log(\frac{1}{2}) = 1$.

2.2

Suppose the attribute we're splitting on labeled X. X has k different possibilities, labeled X_1, X_2, ..., X_k. That means if we split on X, S can be split into k subsets, labeled S_1, S_2, ..., S_k. For all k, the ratio $\frac{p_k}{p_k+n_k}$ is the same.

Recall the equation for information gain is $gain = H[S] - H[S|X = X_k]$. To show that $gain = 0$, we can show that $H[S] = H[S|X = X_k]$. We are given that $H[S] = B(\frac{p}{p+n})$, and it follows that $H[S|X = x_k] = B(\frac{p_k}{p_k+n_k})$. Thus, if we can show $\frac{p}{p+n} = \frac{p_k}{p_k+n_k}$, we can show that $gain = 0$.

Notice how $\frac{p}{p+n}$ is the proportion of positives in S and how $\frac{p_k}{p_k+n_k}$ is the proportion of positives in S_k. Since the proportion of positives for all S_k are the same, it follows that the proportion of positives in S is the same as well. Hence, $\frac{p}{p+n} = \frac{p_k}{p_k+n_k}$, and $gain = 0$.

3 k-Nearest Neighbor

3.1

$k = 1$ minimizes the training set error for this test set because our training set consists of each point in our given data set. $k = 1$ means that for each test data point we examine, we will select the data point in our training set that is closest to that test data point. In our case, since the test data point is in the training set already, we will always predict the label of the test data point correctly. Hence, the resulting training set error is 0. Training set error is not a reasonable estimate of test set error, especially given this value of k, because the training set error does not tell us how well our k-nearest neighbor model performs on other data points. Using the training set error means that we are trying to evaluate our model on what it learned from.

3.2

$k = 5$ minimizes the training set error for this data set, and the resulting training set error is $\frac{4}{14} \approx 0.2857$. Cross validation is a better measure of test set performance because cross validation involves breaking up the training data into equal parts, using each part as validation data, and then finding the average performance across the folds. We can use the "unknown" validation data that we did not use for training to evaluate our performance.

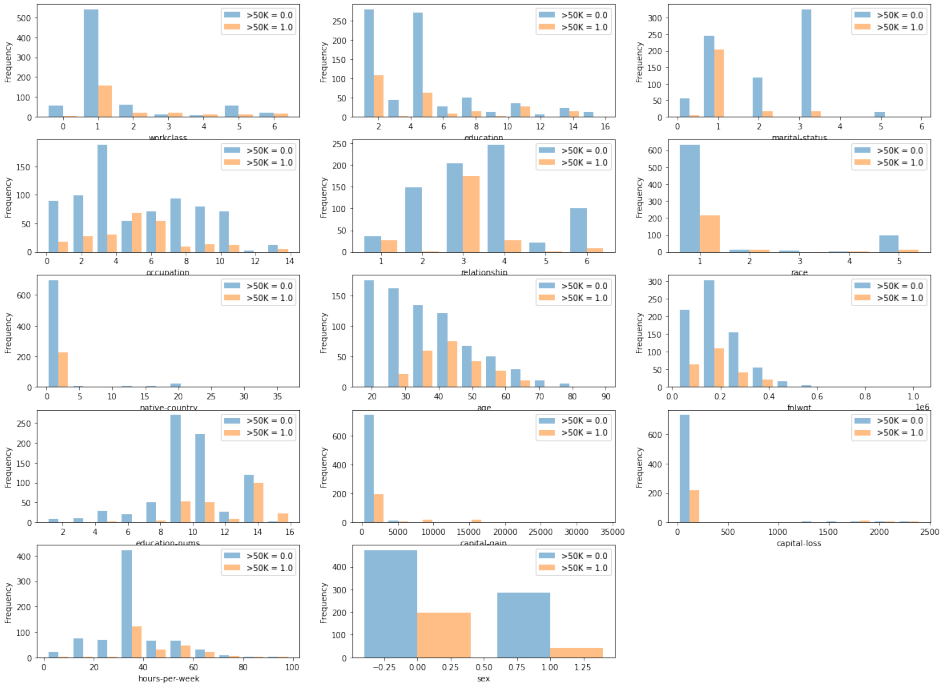
3.3

The LOOCV error for the lowest k in this data set (i.e. $k = 1$) is $\frac{10}{14}$, and the LOOCV error for the highest k in this data set (i.e. $k = 13$) is $\frac{14}{14} = 1$. Using too large a value of k may lead to underfitting whereas using too small a value of k may lead to overfitting.

4 Programming exercise: Applying decision trees and k-nearest neighbors

4.1 Visualization

4.1.1



- workclass: Most of our data points fall under category 1 for workclass. For all workclasses besides 3 and 4, more people make less than or equal to 50k than those who make more than 50k.
- education: Most of our data points fall under two categories. For every education category, more people make less than or equal to 50k than those who make more than 50k.
- marital status: For every marital status category, more people make less than or equal to 50k than those who make more than 50k. However, those whose marital status falls under category 1 are more likely to make more than 50k than any other category.
- occupation: Only for one occupation category is it more likely to make more than 50k.
- relationship: For every relationship category, more people make less than or equal to 50k than those who make more than 50k. However, those who

relationship falls under category 3 are more likely to make more than 50k than any other category.

- race: For every racial category, more people make less than or equal to 50k than those who make more than 50k. There are significantly more people who fall under racial category 1 in our dataset.
- native-country: For every native country, more people make less than or equal to 50k than those who make more than 50k. A large majority of people in our data set come from one native country.
- age: As age increases, the number of subjects in our data set decreases as well, and so does the number of people who make less than or equal to 50k. For every age group, more poeple make less than or equal to 50k than those who make more than 50k.
- fnlwgt: Note fnlwgt describes the number of people the census believes the entry represents. For every fnlwgt group, more poeple make less than or equal to 50k than those who make more than 50k.
- education-nums: For every education-nums group besides one (the one with maximum education-nums), more people make less than or equal to 50k than those who make more than 50k. The more number of years of education one received, the more likely they were to make more than 50k a year.
- capital-gain: A large majority of people seem to have little to no capital gain, but the scale for capital gain ranges anywhere from 0 to 35000.
- capital-loss: A large majority of people seem to have little to no capital loss, but the scale for capital loss ranges anywhere from 0 to 2500.
- hours-per-week: For every hours per week bucket in the histogram, more people make less than or equal to 50k than those who make more than 50k. A large majority of people seem to work about 40 hours per week.
- sex: For both sexes, more people make less than or equal to 50k than those who make more than 50k. Those whose sex falls under category 0 are more likely to make more than 50k compared to those whose sex falls under category 1.

4.2 Evaluation

4.2.1

I implemented fitting for my random classifier by creating a dictionary whose keys were the possible values for y and whose values were the proportion of appearances each key made in the data set. I found the proportion by utilizing the sum() function to find the total number of ones in our data set since our only possible values for y are 0 and 1 in this case.

Then to predict the values of y, I used the np.random.choice() function to randomly generate n values of 0 or 1 with associated probabilities according to their corresponding values in self.probabilities_.

Using the same method as the majority vote classifier, I obtained a training error of 0.374.

4.2.2

The training error of the decision tree classifier with criterion="entropy" is 0. All the other parameters were set to their default values.

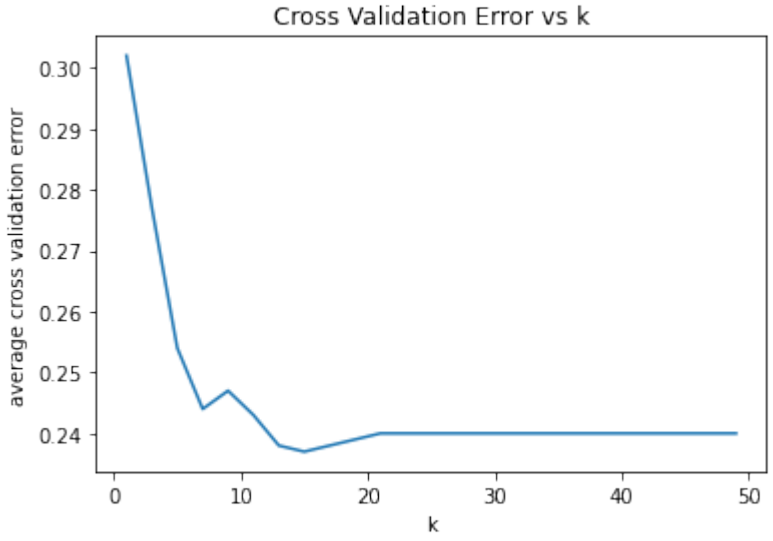
4.2.3

k	training error
3	0.153
5	0.195
7	0.213

4.2.4

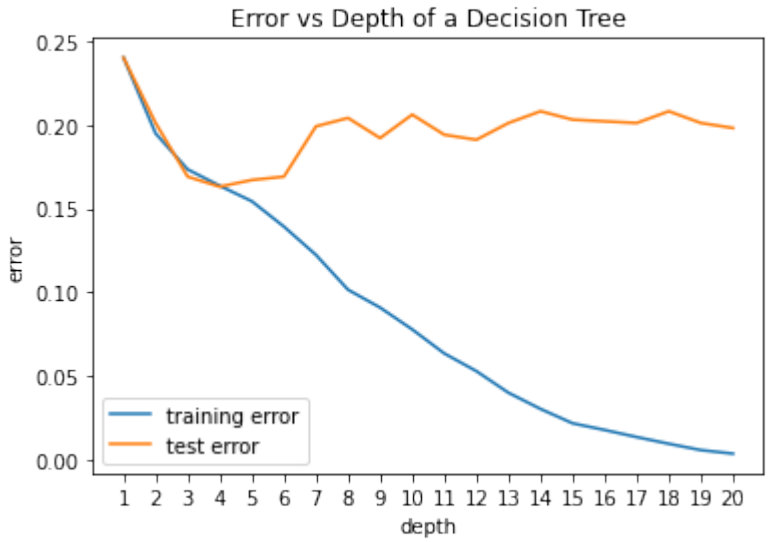
classifier	training error	test error	f1 score
majority vote	0.240	0.240	0.760
random	0.378	0.381	0.619
decision tree	0.000	0.201	0.798
5-neighbors	0.201	0.270	0.730

4.2.5



The best value of k seems to be 15 since that is when the cross validation error is lowest according to our analysis.

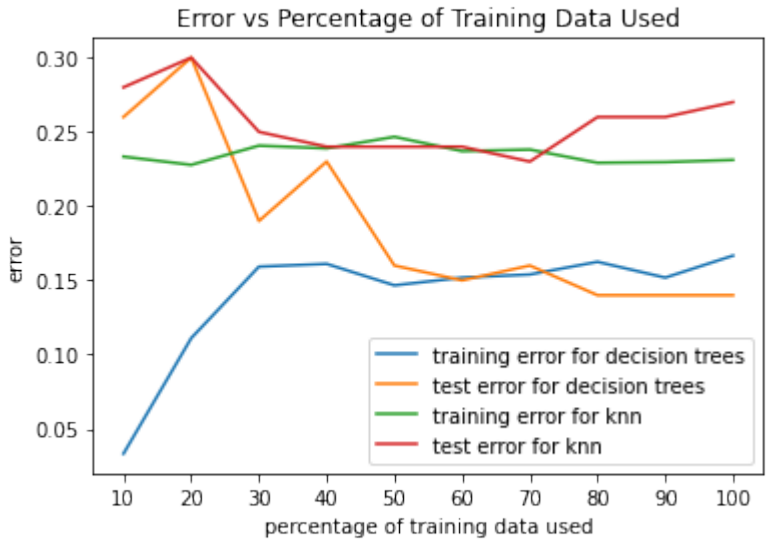
4.2.6



The best depth limit to use for this data is 4 because that is the depth at which test error reaches a minimum, implying that we have reached the ideal balance between overfitting and underfitting our data.

I see overfitting at depths greater than 4 because the training error keeps decreasing, which means the decision tree is performing well on our training data, but the test error is not decreasing anymore, which means performance on test data is not improving.

4.2.7



The training error for decision trees increases as the percentage of training data used increases because when we have fewer data points, the tree is more likely to overfit with a depth of 4, causing training error to be low by test error to be high. It seems that as we add more training data, the test error generally decreases.

4.2.8

After standardization...

- Part b: The majority vote classifier and random classifier retained the same training error because standardizing our X values did not change anything about our y values, which is what we solely used to implement our majority vote classifier and random classifier.
- Part c: The decision tree classifier retained a training error of 0 because the decision tree was still able to overfit on our training data since we did not specify a max depth for our tree.
- Part d: The 3-neighbors classifier, 5-neighbors classifier, and 7-neighbors classifier had lower training errors than before because standardizing our X values means each feature contributes to the distance for the k nearest neighbor classifier equally.

k	training error
3	0.114
5	0.129
7	0.152

- Part e: Upon performing cross validation, the average training and test errors of each my classifiers on the adult_subsample data set remained the same for all but the k nearest neighbor classifier because as mentioned before, standardizing our X values does not affect our performance for these three classifier. Standardizing our X values improves the performance of our k nearest neighbor classifier because it allows us to weight each feature equally.

classifier	training error	test error	f1 score
majority vote	0.240	0.240	0.760
random	0.378	0.381	0.619
decision tree	0.000	0.202	0.797
5-neighbors	0.135	0.206	0.793

- Part f: The best value of k is now 27 instead of 15.
- Part g: Our graph looks the same as before, and the best depth limit to use for this data still seems to be 4. As depth increases, overfitting occurs, and thus training error decreases while test error increases.
- Part h: Although the training error and test error for decision trees did not change from before, we see a pretty drastic difference for k nearest neighbors. Both training error and test error for k nearest neighbors has decreased by quite a bit, and now as the percentage of training data used increases, our test error is less than our training error. Previously, our decision tree algorithm had lower error rates than our k nearest neighbors algorithm, but now our k nearest neighbors algorithm outperforms our decision tree algorithm.