Danning Yu, 305087992
CS M146 Discussion 1A
PS1
1/18/2020

Note: all logarithms are base 2.

1. (a) Using a simple majority vote, the best 1-leaf decision tree always assigns output to 1. For $n = 4$, this results in 2 errors. For $n = 5$, this results in 4 errors. For n boolean features, there will be 2^{n-3} mistakes, which results in an error rate of $2^{n-3}/2^n = 0.125$. Accuracy is $1 - 0.125 = 0.875$.

(b) No, consider the two possible cases for splitting: either split on feature X_1, X_2, or X_3, or on feature X_i, where $i \geq 4$.
Case 1: Splitting by X_1, X_2, or X_3: Assuming we split on X_1 (results will be same if we split on X_1 or X_2), then the two resulting leaves will still both predict 1 by majority vote. For the leaf reached by going down the $X_1 = 0$ path, the prediction of 1 has a 0.75 accuracy rate. This is because there will still be 2^{n-3} errors, but now out of 2^{n-1} possible states (because the results have been split once), so the error rate is 0.25, which is a 0.75 accuracy rate. By going down the $X_1 = 1$ path, the prediction of 1 has a 100% accuracy rate. Thus, the overall accuracy is

$$(0.5)(0.75) + (0.5)(1) = 0.875,$$

which is the same accuracy as before.

Case 2: Splitting by X_i, where $i \geq 4$: By majority vote, the two resulting leaves will still predict 1. Because we split on a variable that is not included in the function f, the resulting accuracy for each of the leaves will still be 0.875, so the total accuracy is

$$(0.5)(0.875) + (0.5)(0.875) = 0.875,$$

which is the same as before. Thus, there's not a split that reduces the number of mistakes.

(c) $$H[Y] = -\left(P(Y=1)\log P(Y=1) + P(Y=0)\log P(Y=0)\right)$$
$$= -\left(\frac{7}{8}\log\frac{7}{8} + \frac{1}{8}\log\frac{1}{8}\right) = 0.544 \text{ bits}.$$

(d) Yes, by splitting along X_1, X_2, or X_3. Using X_1:
Conditional entropy:

$$H[Y \mid X_1] = -\frac{1}{2}\left(P(Y=1 \mid X_1=1)\log P(Y=1 \mid X_1=1) + P(Y=0 \mid X_1=1)\log P(Y=0 \mid X_1=1)\right)$$
$$-\frac{1}{2}\left(P(Y=1 \mid X_1=0)\log P(Y=1 \mid X_1=0) + P(Y=0 \mid X_1=0)\log P(Y=0 \mid X_1=0)\right)$$

$$H[Y \mid X_1] = -\frac{1}{2}(1\log 1 + 0\log 0) - \frac{1}{2}\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right) = 0.406 \text{ bits}.$$

2. (a) Let $\frac{p}{p+n} = q$. Then we take the first derivative of B with respect to q (use chain rule) and apply the first derivative test:

$$\frac{dB}{dq} = -\log q - q\frac{q}{\ln 2} - \left(-\log(1-q) + (1-q)\frac{1}{(1-q)(-1)\ln 2} \right) = -\log q + \log(1-q) = 0$$

We get $q = 1-q$, giving $q = 0.5$ as a critical point. $B'(0.3) > 0$ and $B'(0.7) < 0$, so $q = 0.5$ is a maximum. The value is $B(0.5) = 1$. We then check the endpoints: the variable q is restricted to the range [0, 1], and we have $B(0) = 0$, and $B(1) = 0$. The only critical point in the interval [0, 1] is 0.5, and it is a maximum, so thus, $0 \le B(q) \le 1$, which gives $0 \le H(S) \le 1$, as desired.

Then, when $p = n$, $\frac{p}{p+n} = \frac{p}{p+p} = 0.5$, and according to above, $H(S = 0.5) = 1$, as desired.

(b) Gain is defined as $H[S] - H\left[S \mid \text{split}\right]$. Taking advantage of the fact that the ratio $\frac{p_k}{p_k + n_k}$ is identical for all the partitions, and that sum of p_k over all k is p (and similarly for n_k over all n), this means that $\frac{p_k}{p_k + n_k} = \frac{p}{p+n}$ for every single partition. The entropy resulting from the split can be calculated using the weighted average of the entropy of each partition:

$$H\left[S \mid \text{split}\right] = \sum_k \left(\frac{p_k + n_k}{p+n} \left(B\left(\frac{p_k}{p_k + n_k} \right) \right) \right)$$

$$H\left[S \mid \text{split}\right] = \left(\frac{1}{p+n} \right) B\left(\frac{p}{p+n} \right) \sum_k \left(p_k + n_k \right)$$

$$H\left[S \mid \text{split}\right] = \left(\frac{1}{p+n} \right) B\left(\frac{p}{p+n} \right) (p+n) = B\left(\frac{p}{p+n} \right)$$

Substituting this expression and the expression for $H[S]$ into the gain equation gives

$$H[S] - H\left[S \mid \text{split}\right] = B\left(\frac{p}{p+n} \right) - B\left(\frac{p}{p+n} \right) = 0$$

Thus, it has been shown that this particular split results in no information gain.

3. (a) The value $k = 1$ minimizes the training set error, and it results in an error of 0. This is because you are simply assigning a label based on the label that it has, as it is its own closest neighbor. This is not a reasonable estimate of the test set error because you are overfitting the data and not taking any of the data's neighbors into account.

(b) Using either $k = 5$ or $k = 7$ minimizes the LOOCV error: for both values of k, you get 4 misclassified points and 10 correctly classified points, resulting in an error of $4/14 = 0.286$. Cross validation is a better measure of test set performance because you are cycling through which data is used as test data (and the complement being used as training data), so you are removing any bias that could result from training or testing on a particular set of data.

(c) If we use $k = 0$, the error is 0. If we use $k = 1$, we get an error of $10/14 = 0.714$. If we use $k = 13$, we get an error of $14/14 = 1$. A too small of a value of k results in overfitting, while a too high value of k results in the problem degenerating into a simple majority vote, potentially causing all points to be misclassified, as demonstrated in this example.

4. (a) If we divide by passenger class, it seems that twice as many first class passengers survived than the number that died, approximately equal numbers of second class passengers survived and died, and three times as many third class passengers died compared to the number that lived. Thus, a lower class seems to correlate to a higher survival rate.

If we divide by gender, it seems that women had a much better chance of surviving (survival:death ratio of 2.5:1) than men (survival:death ratio of 1:3).

If we divide by age, children were the most likely to survive, old passengers had about the same chance of survival and death, and those in their 20s and 30s were most likely to die.

If we divide by the number of siblings and/or spouses the passenger was traveling with, those traveling alone (zero siblings and/or spouses) were twice as likely to die than to live, while others had roughly the same chances of dying and living.

If we divided by number of parents and/or children the passenger was traveling with, we see the same trend as the one exhibited by number of siblings and/or spouses traveling with that passenger: those traveling alone were much more likely to die.

If we divide by the fare paid, those with cheaper fares are twice as likely to die than to live, while those that paid higher fares had a much better chance of survival.

If we divide by embarkation location, those who boarded from Cherbourg had an roughly equal chance of dying or surviving, while those who boarded from Queenstown or Southampton were more likely to die (survival:death ratio of 1:2).

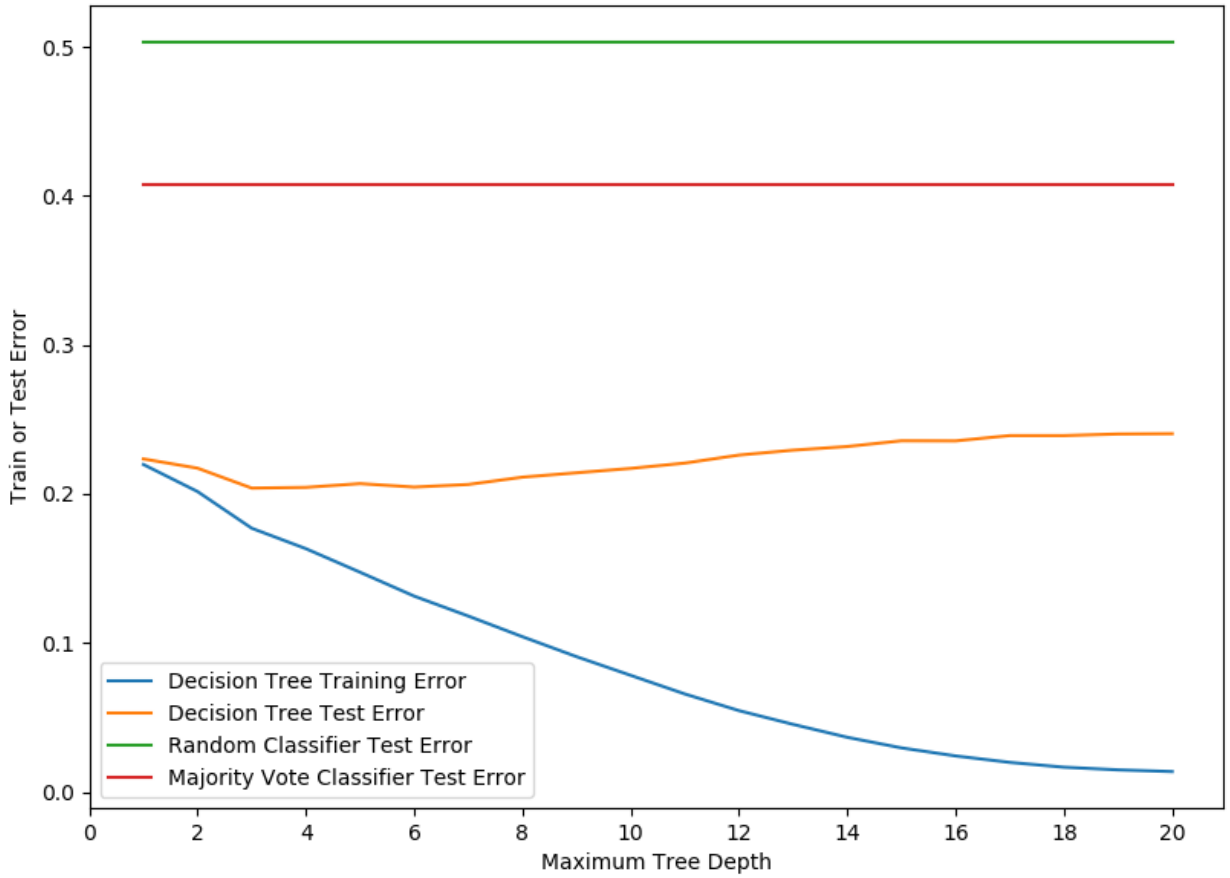
(b) I implemented the functions in the RandomClassifier class, including `fit(...)` and `predict(...)`. In the end, I got that 59.55% of passengers died and 40.45% survived, leading to the training algorithm having an error of 0.483, which is slightly different from the expected 0.485 value given in the problem statement.

(c) Using the DecisionTreeClassifier class provided in scikit-learn with the criterion set to entropy, the model yielded a training error of 0.014.

(d) Implementing the `error(...)` function, I got the following training and test errors using the three classifiers:

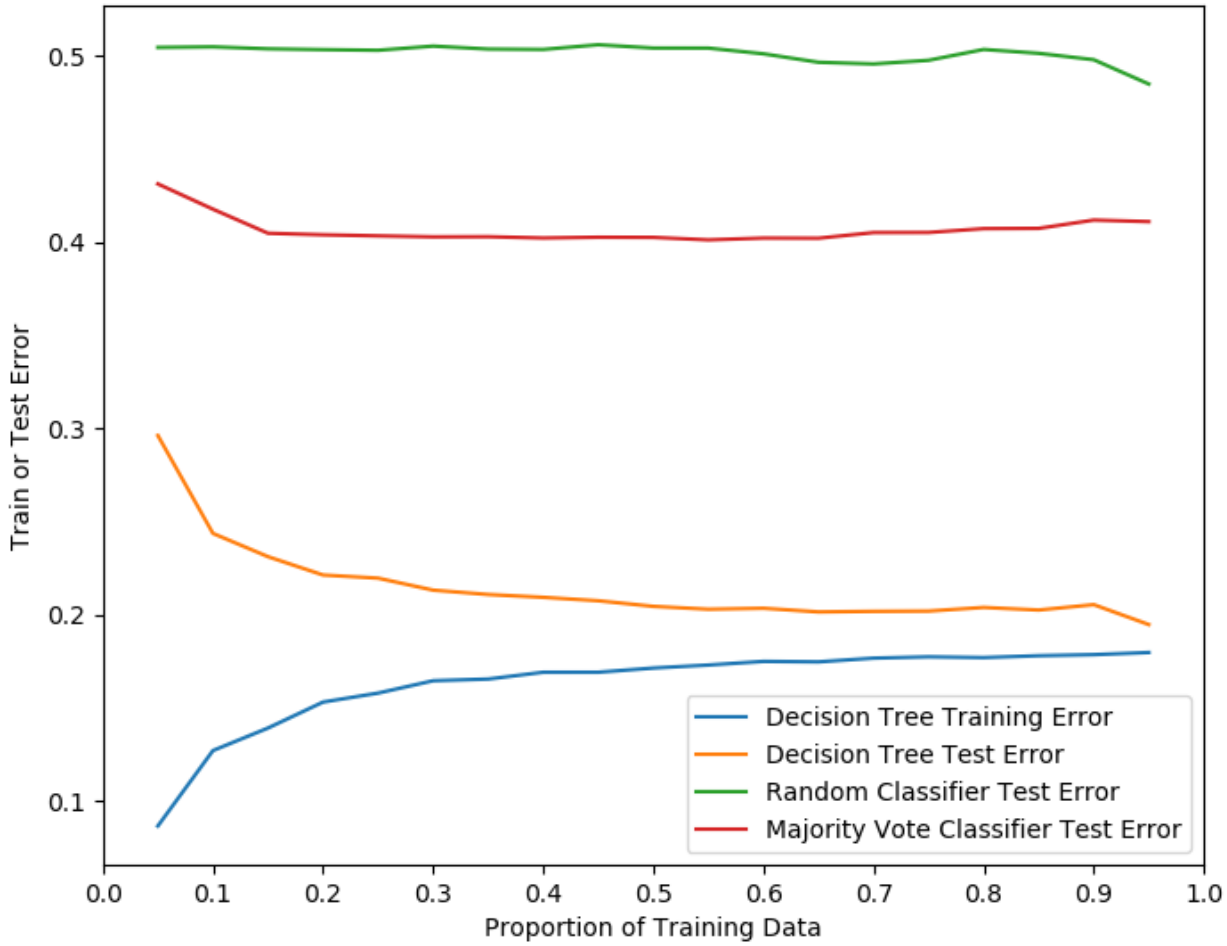
majority vote training error: 0.404; testing error: 0.407
random classifier training error: 0.503; testing error: 0.504
decision tree training error: 0.012; testing error: 0.241

(e) Letting the depth range from 1 to 20, I got the following graph:



The best depth to use is 3, which corresponds to the smallest test error of 0.204. I see overfitting: as the maximum tree depth increases, the decision tree training error (blue line) decreases, but the decision tree test error (orange line) actually increases, indicating that presence of overfitting.

(f) Letting the proportion of training data vary from 5% to 95%, I got the following graph:



From this plot, we can see that as the proportion of data allocated towards training increases, the decision training error increases, and then mostly levels out, while the decision tree test error decreases, and then mostly levels out; the training and test error seem to be converging towards the same value. The random and majority vote classifier test errors stay relatively constant as the proportion of training data increases, which makes sense, as they classify using heuristics that are not susceptible to overfitting.