CM146, Winter 2020
Problem Set 1: Decision trees, Nearest neighbors
Due Jan 27, 2020

1 Splitting Heuristic for Decision Trees

Solution:

(a) In the 1-leaf decision tree, the decision of Y would be the value of Y which is the most probable. In the case where $n \geq 4$, there are 2^{n-3} examples where $Y = 0$, and $2^n - 2^{n-3}$ examples where $Y = 1$. Therefore, the best 1-leaf decision tree chooses $Y = 1$, and makes $\boxed{2^{n-3} \textbf{ mistakes}}$.

(b) **No**. In all of the mistakes: $X_1 = 0, X_2 = 0, X_3 = 0$, because then Y is supposed to be 0. In a split of X_i where $1 \leq i \leq 3$:

- If $X_i = 1$ then there is no mistake since in this case all of the examples are with $Y = 1$, which is the correct decision given that $X_i = 1$.
- If $X_i = 0$, there are still 2^{n-3} examples with $Y = 0$, but now we have 2^{n-1} examples in total. Thus, the majority of the examples still favor $Y = 1$, and so the decision remains $Y = 1$. This is the incorrect decision for all 2^{n-3} examples with $Y = 0$, and so the number of mistakes are still 2^{n-3}

Thus, in this case the number of mistakes remain the same. Also, in a split of X_i where $4 \leq i \leq n$:

- If $X_i = 1$, there are 2^{n-4} examples with $Y = 0$, but the decision remains $Y = 1$, and so the decision is a mistake for all of these examples.
- If $X_i = 0$, there are 2^{n-4} examples with $Y = 0$, but the decision remains $Y = 1$, and so the decision is a mistake for all of these examples.

Thus, in this case, we have $2 \cdot 2^{n-4} = 2^{n-3}$ mistakes, and so in every possible single split the number of mistakes remain 2^{n-3}.

(c) The entropy is:

$$\begin{aligned} H[Y] &= -Pr(Y=1)\log Pr(Y=1) - Pr(Y=0)\log Pr(Y=0) \\ &= -\frac{2^n - 2^{n-3}}{2^n}\log\frac{2^n - 2^{n-3}}{2^n} - \frac{2^{n-3}}{2^n}\log\frac{2^{n-3}}{2^n} \\ &= -\frac{7}{8}\log\frac{7}{8} - \frac{1}{8}\log\frac{1}{8} \approx \boxed{0.54} \end{aligned}$$

.

(d) From the symmetry of the problem, we can understand that a split of either X_1, X_2, X_3 would result in the same reduction in entropy, because all of these variables have the same effect on Y. The rest of the variables doesn't affect Y, so splitting them will not give us any information. Thus, we will consider the split of X_1, and see if it reduces the entropy:

- If $X_1 = 1$ there is no entropy, since in that case $Y = 1$ regardless of the other variables ($Pr(Y=1) = 1$).
- If $X_1 = 0$, the entropy is:
  $H[Y|X_1 = 0] = -Pr(Y = 1|X_1 = 0)\log Pr(Y = 1|X_1 = 0) - Pr(Y = 0|X_1 = 0)\log Pr(Y = 0|X_1 = 0) = -\frac{2^{n-1}-2^{n-3}}{2^{n-1}}\log\frac{2^{n-1}-2^{n-3}}{2^{n-1}} - \frac{2^{n-3}}{2^{n-1}}\log\frac{2^{n-3}}{2^{n-1}} = -0.75\log 0.75 - 0.25\log 0.25 \approx 0.811.$

Thus, the entropy when splitting X_1 is (half of the examples are with $X_1 = 0$):

$$H[Y|X_1] = H[Y|X_1 = 1]Pr(X_1 = 1) + H[Y|X_1 = 0]Pr(X_1 = 0) = 0\cdot\frac{1}{2} + 0.811\cdot\frac{1}{2} = 0.405$$

Therefore, the resulting conditional entropy of Y given this split is $\boxed{0.405}$. The entropy is reduced by 0.1345.

2 Entropy and Information

Solution:

(a) We know that $0 \le \frac{p}{p+n} \le 1$. Let $q = \frac{p}{p+n}$. Therefore:
$H(S) = B(\frac{p}{p+n}) = B(q) = -q\log q - (1-q)\log(1-q) = q(\log(1-q) - \log q) - \log(1-q) = q\log\frac{1-q}{q} - \log(1-q) = q\log\frac{1-q}{q} + \log\frac{1}{1-q} =$

$\log \frac{(1-q)^q}{q^q} + \log \frac{1}{1-q} = \log \frac{1}{(1-q)^{1-q} q^q}$. To prove that $0 \leq H(S)$, we need to prove that $1 \leq \frac{1}{(1-q)^{1-q} q^q}$. Indeed:

$$0 \leq q \leq 1 \Rightarrow (1-q)^{1-q} q^q \leq 1 \Rightarrow \mathbf{1 \leq \frac{1}{(1-q)^{1-q} q^q}}$$

And thus: $0 \leq H(S)$. Also, from $H(S)$, we know that the maximum entropy happens when $q = \frac{1}{2}$:

$$\frac{dH(S)}{dq} = -\log q - 1 + \log(1-q) + 1 = \log(1-q) - \log q = 0$$

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

By plugging back $q = \frac{1}{2}$ into $H(S)$, we get: $H(S) = -\frac{1}{2}\log(\frac{1}{2}) \cdot 2 = \mathbf{1}$. Therefore, we proved that $0 \leq H(S) \leq 1$.
In our proof, we also showed that the max entropy happens when $q = \frac{1}{2}$. This is equivalent to the case in which $p = n$, because when $p = n$ we get:

$$q = \frac{p}{p+n} = \frac{1}{2}$$

which is exactly the maximum point of the entropy ($H(S) = 1$ when $q = \frac{1}{2} \Rightarrow$ when $p = n$).

(b) It's given that $\frac{p_j}{p_j+n_j}$ is the same for all j. Therefore:

$$\frac{p_j}{p_j+n_j} = \frac{p_1}{p_1+n_1} \Rightarrow p_j = \alpha_j p_1, p_j + n_j = \alpha_j(p_1+n_1)$$

. Thus:

$$\frac{p}{p+n} = \frac{p_1+p_2+\ldots+p_k}{p_1+p_2+\ldots+p_k+n_1+n_2+\ldots+n_k}$$

$$= \frac{p_1+\alpha_2 p_1+\ldots+\alpha_k p_1}{p_1+\alpha_2 p_1+\ldots+\alpha_k p_1+n_1+\alpha_2 n_1+\ldots+\alpha_k n_1}$$

$$= \frac{p_1(1+\alpha_2+\ldots+\alpha_k)}{(1+\alpha_2+\ldots+\alpha_k)(p_1+n_1)} = \frac{p_1}{p_1+n_1}$$

So we got $\frac{p}{p+n} = \frac{p_1}{p_1+n_1}$. By definition, the information gain of X_j is:

$$Gain = H[Y] - H[Y|X_j] = B(\frac{p}{p+n}) - \sum_k Pr(X_j = a_k) H[Y|X_j = a_k]$$

$$= B(\frac{p}{p+n}) - \sum_k Pr(X_j = a_k) B(\frac{p_k}{p_k + n_k}) = B(\frac{p}{p+n}) - \sum_k Pr(X_j = a_k) B(\frac{p_1}{p_1 + n_1})$$

$$= B(\frac{p}{p+n}) - \sum_k Pr(X_j = a_k) B(\frac{p}{p+n}) = B(\frac{p}{p+n}) - B(\frac{p}{p+n}) \underbrace{\sum_k Pr(X_j = a_k)}_{=1} = \boxed{0}$$

Thus, there's no information gain in that case.

3 k-Nearest Neighbors

Solution:

(a) For $\boxed{\text{k=1}}$, each example in the training set is the closest neighbor of itself, and so each example in the training set will be correctly classified. Thus, $k = 1$ minimizes the training set error, and the minimum error is 0. The training set error is not a reasonable estimate of test set error since $k = 1$ extremely over-fits the model to the training data, which limits the generalization of the model to make a good estimate for testing data.

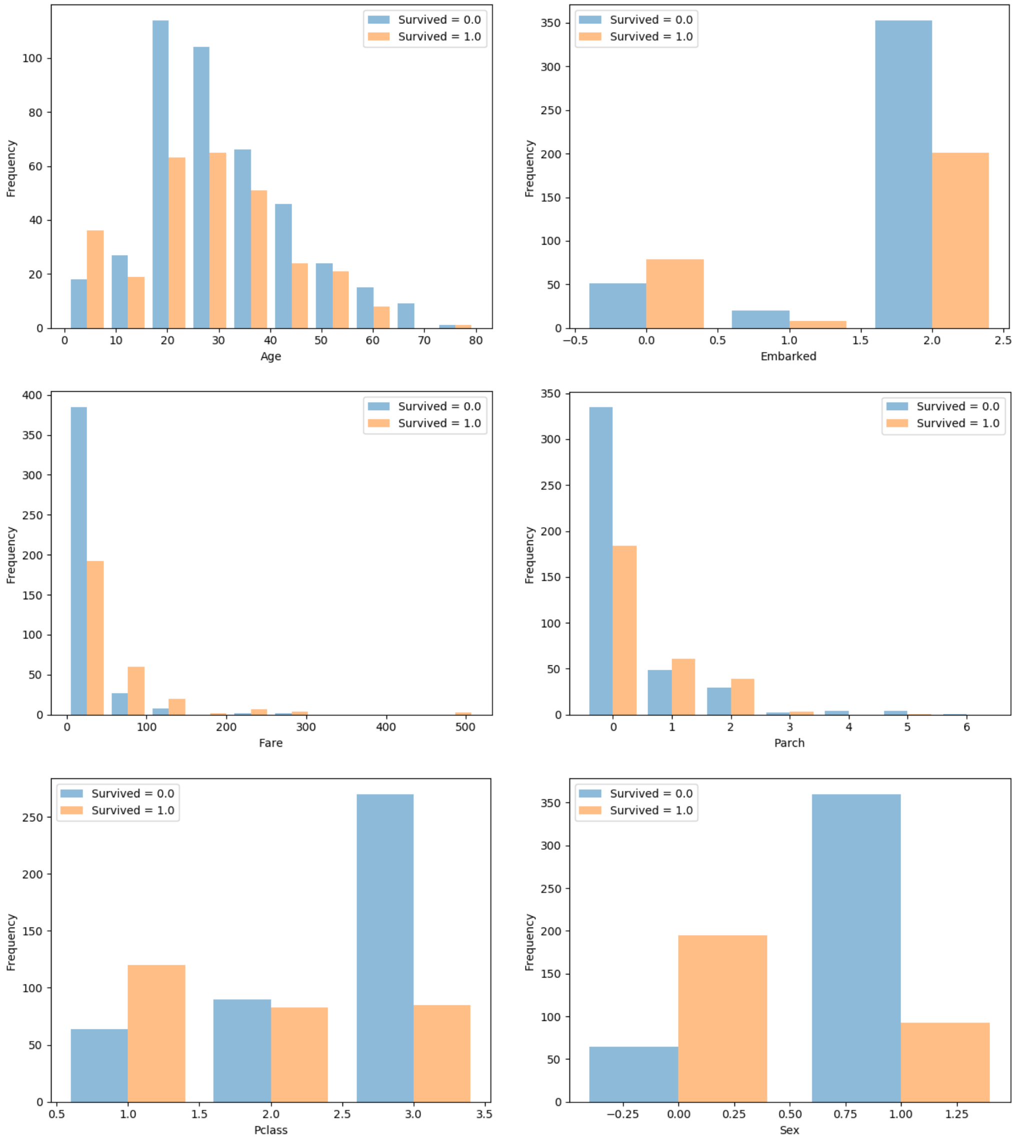
(b) For the examples at $(7, 3)$ and $(3, 7)$, we need $k = 5$ to make sure they are correctly classified, because for $k = 4$, their class will be chosen arbitrarily (they have 2 neighbors from each class), and for $k < 4$ their class will be chosen to be incorrect (their $k < 4$ closest neighbors are of the opposite class). For all other examples, we need $k < 5$ for each one of them to be correctly classified if it would be the entire validation set. Thus, $\boxed{\text{k=5}}$ minimizes the leave-one-out cross-validation error, since for $k = 5$ the validation error would be 0 for each fold. Cross-validation is a better measure of test-set performance since it helps avoid overfitting to hyperparameters.

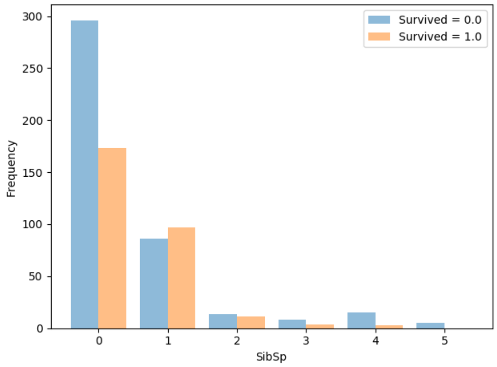
(c) For LOOCV, the lowest k is 1, and the highest k is 13 (13 examples left in the training set when we leave one out for validation). For $k = 1$, the examples $(6, 2)$, $(7, 3)$, $(8, 4)$, $(7, 2)$, $(8, 3)$, $(2, 6)$, $(3, 7)$, $(4, 8)$, $(2, 7)$, $(3, 8)$ are classified incorrectly when each one of them is left as the validation set. Thus, $\boxed{\text{for } k = 1 \text{ the error is } \frac{10}{14}}$ because out of 14 test examples in total, 10 results in a wrong classification. For $k = 13$, each example as the validation set is classified incorrectly because there are 6 examples in the training set from the same class, and 7 examples from the opposite class. Thus, $\boxed{\text{for } k = 13 \text{ the error is } 1}$ because all test examples are classified incorrectly. Using too large values of k can cause underfitting. Using too small values of k can cause overfitting.

4 Programming exercise : Applying decision trees

Solution:

(a) The code plotted the following histograms:

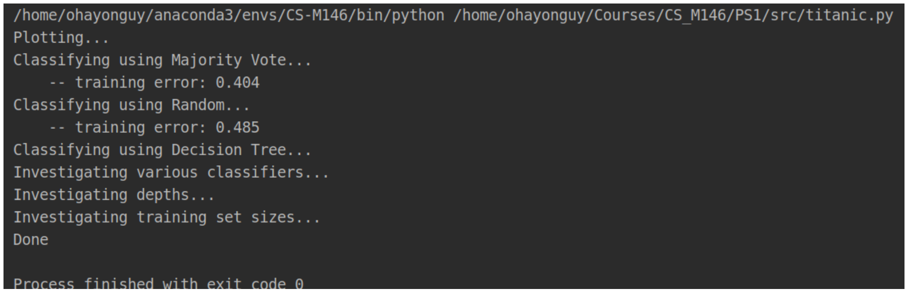




Thus, I observe the following trends:

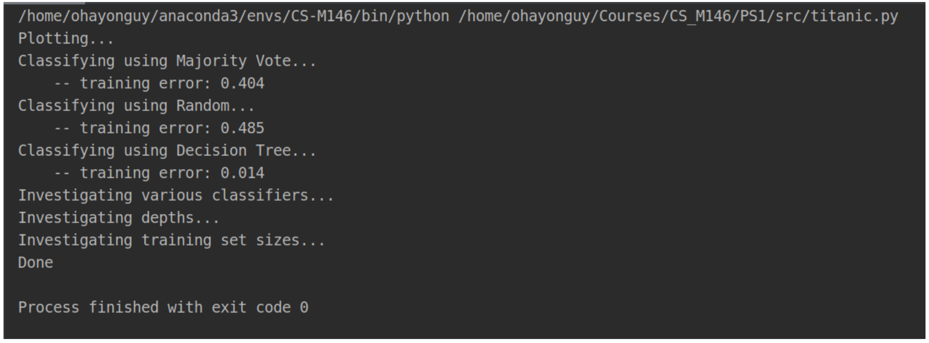
- **Age**: people who are less than 10 years old had a higher chance of survival. Also, the majority of the people on the ship were aged between 20 and 40.
- **Embarked**: people who were embarked at Cherbourg had a higher chance of survival. Also, the majority of the people were embarked at Southampton.
- **Fare**: people who paid more had a higher chance of survival. Also, the majority of the people paid a low fare.
- **Parch**: people with 1,2 or 3 parents/children aboard had a higher chance of survival. Also, the majority of the people had 0 parents/children aboard.
- **Pclass**: people from the 1st class had a higher chance of survival. Also, the majority of the people were from the 3rd class.
- **Sex**: women had a higher chance of survival.
- **SibSp**: people who had 1 sibling/spouse aboard had a higher chance of survival. Also, the majority of the people had 0 siblings/spouses aboard.

(b) After implementing the code, I correctly received the desired error rate:



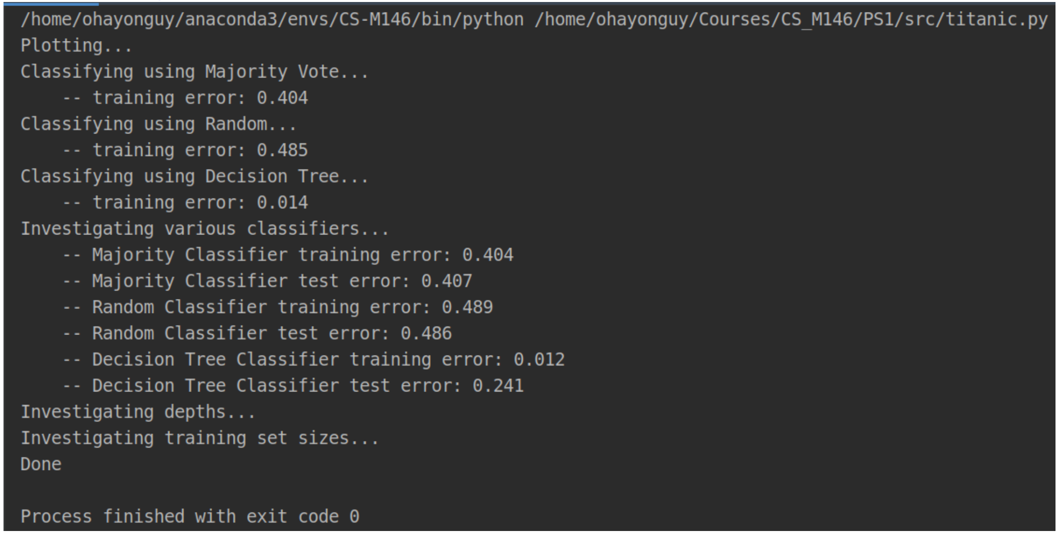
```
/home/ohayonguy/anaconda3/envs/CS-M146/bin/python /home/ohayonguy/Courses/CS_M146/PS1/src/titanic.py
Plotting...
Classifying using Majority Vote...
    -- training error: 0.404
Classifying using Random...
    -- training error: 0.485
Classifying using Decision Tree...
Investigating various classifiers...
Investigating depths...
Investigating training set sizes...
Done

Process finished with exit code 0
```

(c) By using the entropy criterion, I received a training error of 0.014 for this classifier:



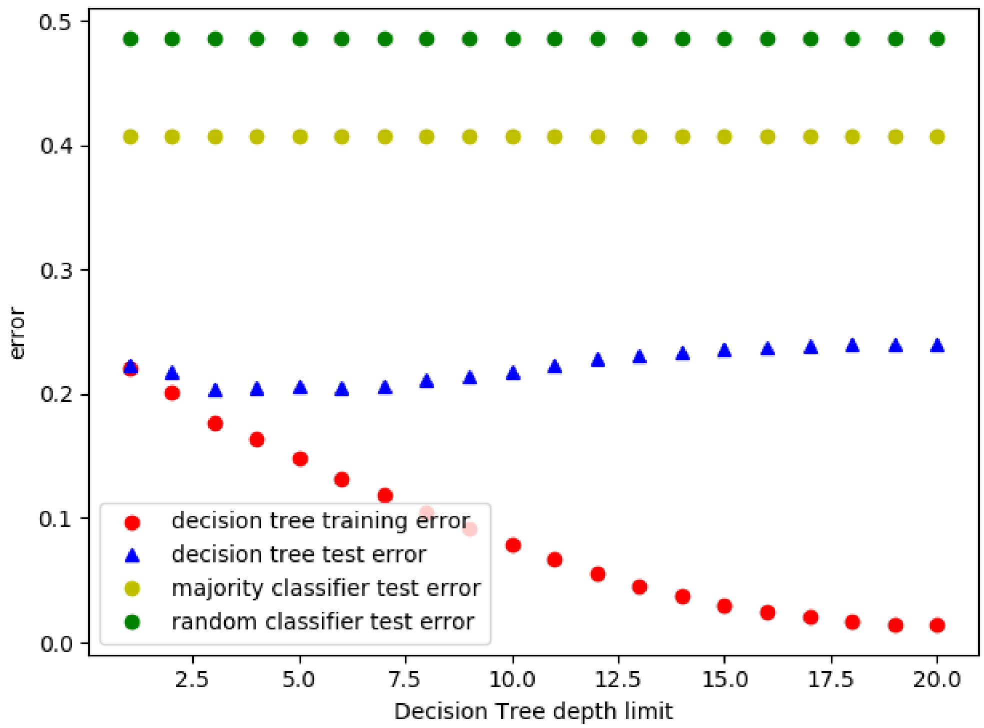
```
/home/ohayonguy/anaconda3/envs/CS-M146/bin/python /home/ohayonguy/Courses/CS_M146/PS1/src/titanic.py
Plotting...
Classifying using Majority Vote...
    -- training error: 0.404
Classifying using Random...
    -- training error: 0.485
Classifying using Decision Tree...
    -- training error: 0.014
Investigating various classifiers...
Investigating depths...
Investigating training set sizes...
Done

Process finished with exit code 0
```

(d) After using cross-validation with the required settings, I have received the following results:



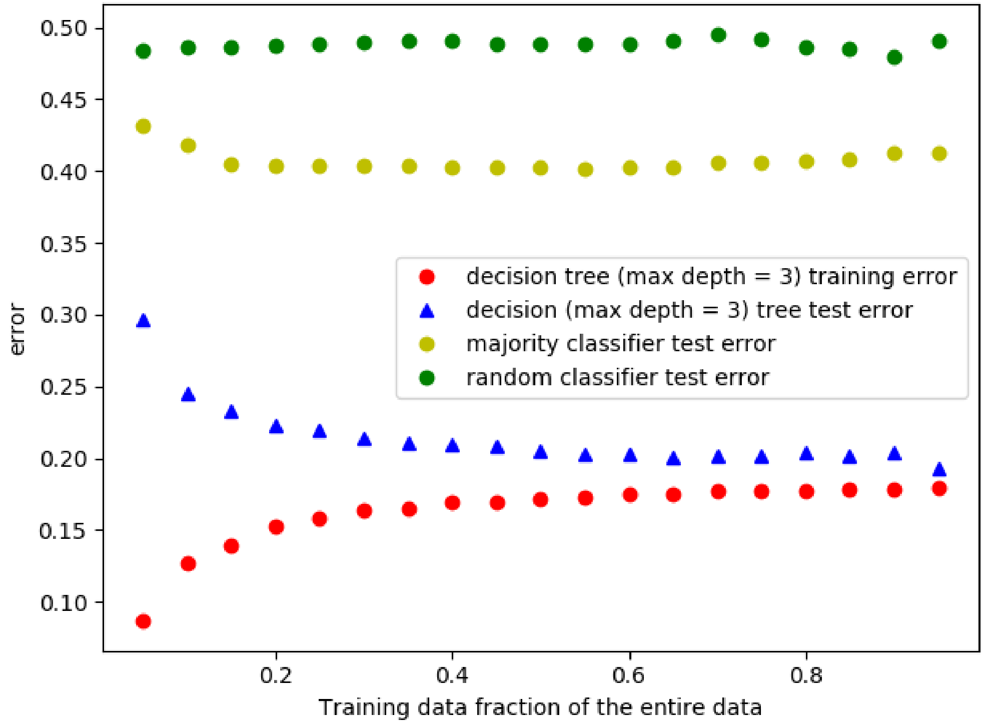
```
/home/ohayonguy/anaconda3/envs/CS-M146/bin/python /home/ohayonguy/Courses/CS_M146/PS1/src/titanic.py
Plotting...
Classifying using Majority Vote...
    -- training error: 0.404
Classifying using Random...
    -- training error: 0.485
Classifying using Decision Tree...
    -- training error: 0.014
Investigating various classifiers...
    -- Majority Classifier training error: 0.404
    -- Majority Classifier test error: 0.407
    -- Random Classifier training error: 0.489
    -- Random Classifier test error: 0.486
    -- Decision Tree Classifier training error: 0.012
    -- Decision Tree Classifier test error: 0.241
Investigating depths...
Investigating training set sizes...
Done

Process finished with exit code 0
```

(e) I received the following plot:



The minimal test error is achieved when the depth limit is 3. Thus, the best depth limit is 3. In addition, we can definitely see overfitting: as the depth limit goes above 3, the test error increases while the training error decreases, which is exactly the behavior of overfitting.

(f) I received the following plot:



We can see that as the fraction of the training data goes up, the test error goes down and the training error goes up. This is expected since when more training data is present, the model could possibly generalize to better fit the distribution of the data.