Preyasi Gaur

705704939

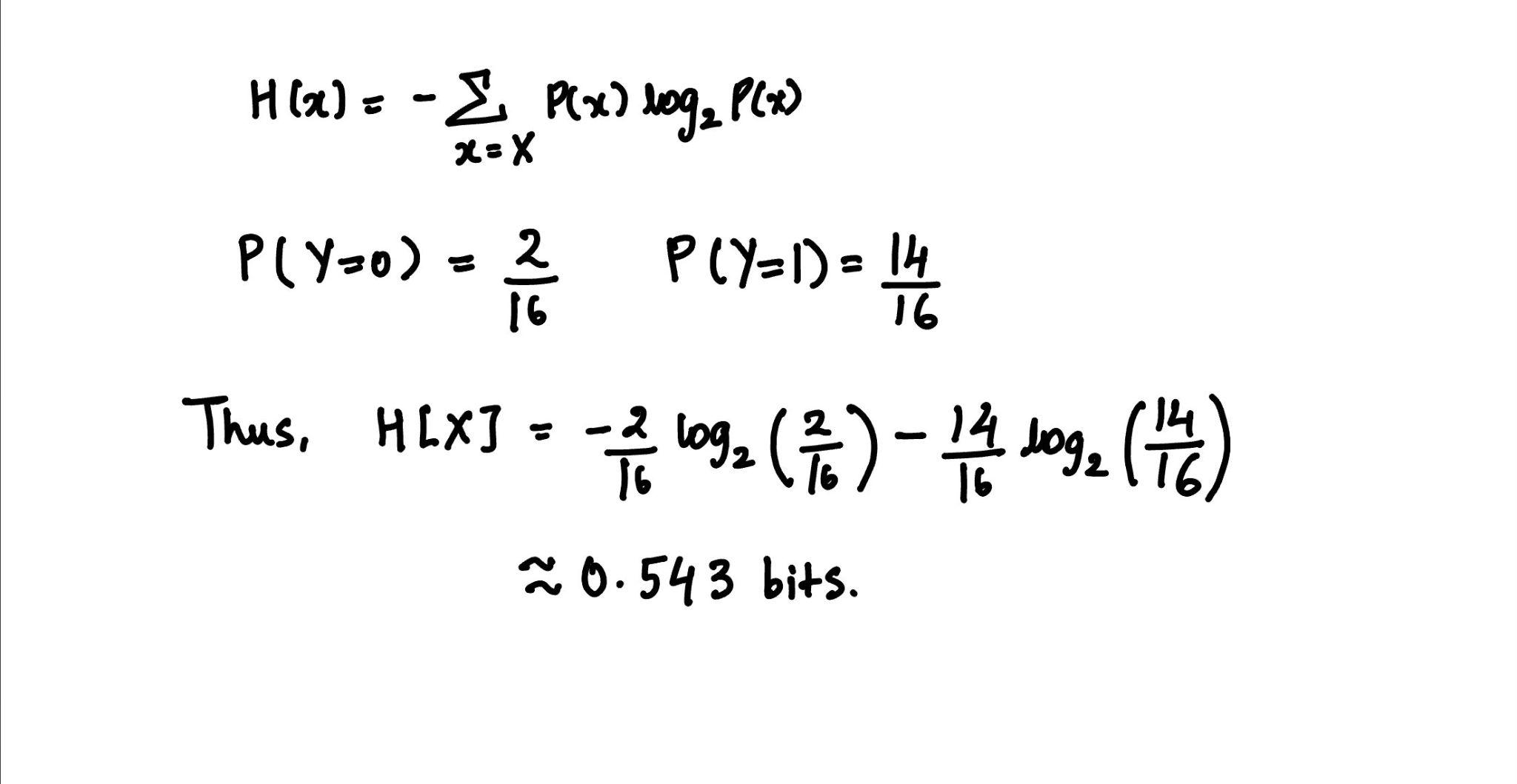
Computer Science M146

Problem Set 1

1 Splitting Heuristic for Decision Trees

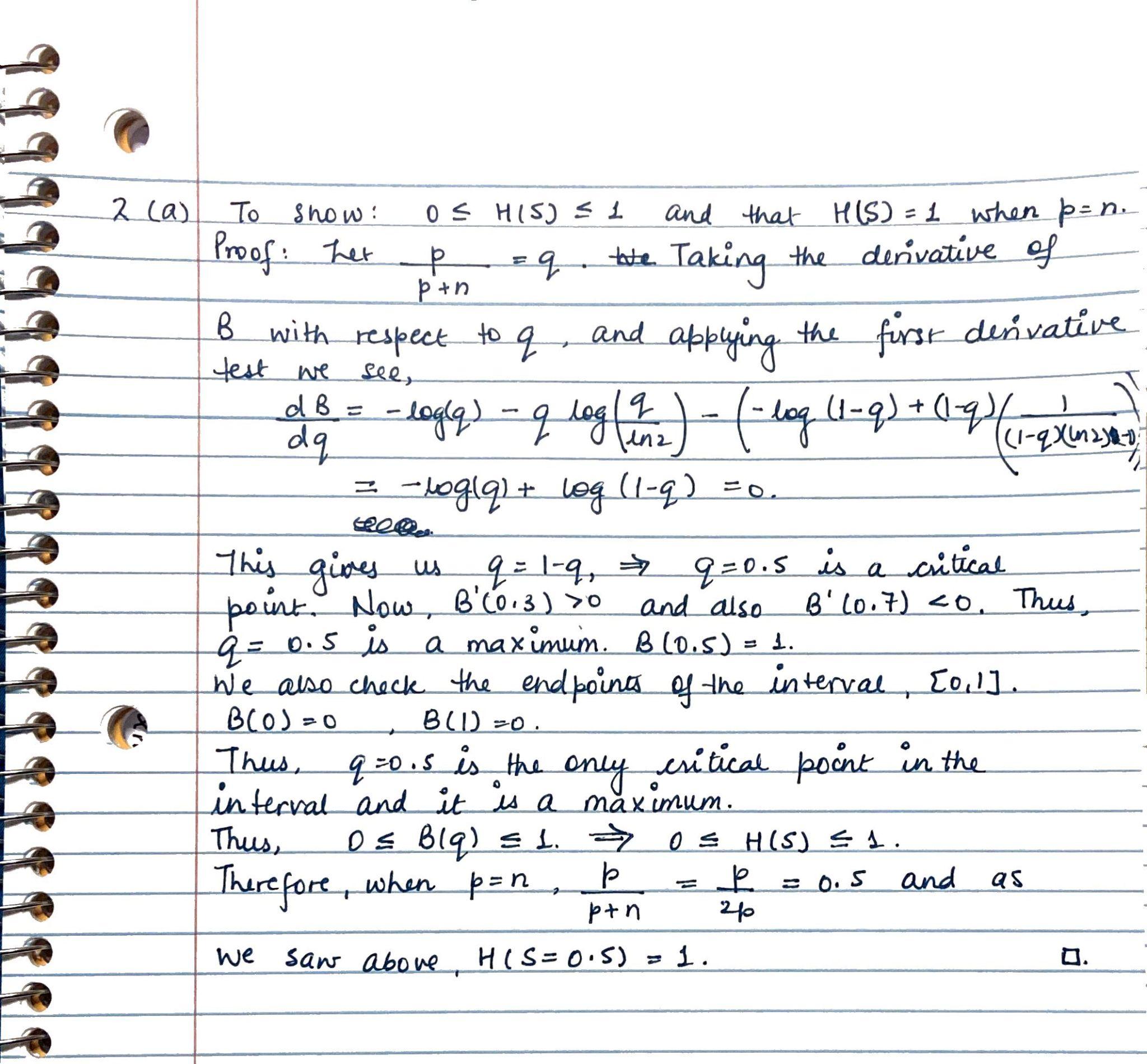
1. For the given data notice that Y is labeled 0 only when X1 = X2 = X3 = 0. The number of such binary vectors is given by because there are two choices for each of the remaining (n − 3) features. Since − **≈** , as >> . Thus, for the best decision tree, the best 1-leaf decision tree predicts 1 every time, and ends up making mistakes. This corresponds to making an error / = 1/8 th of the time.
2. No. It does not matter what variable we put at the root, because no matter what attribute we split on, the error will be the same. We are given the function:

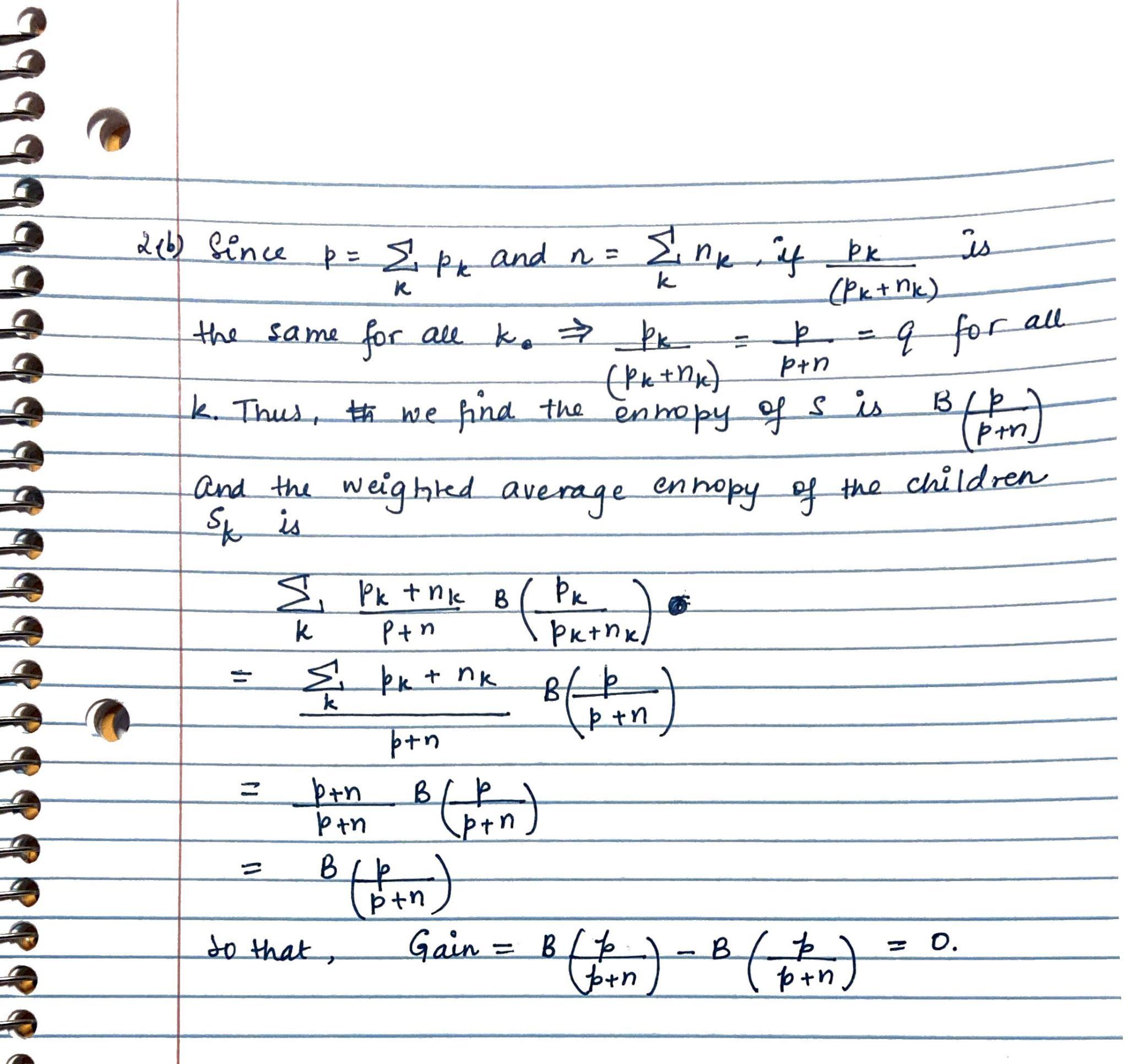
Y = ∨ ∨ ⇒ Y = 1 when = 1, =1 or = 1, and Y = 0 for every other case. Thus, it does not matter how we choose the first attribute there will always be a case when = 1, =1 or = 1 and Y = 1. Thus, any split based on one attribute will result in a mistake. Thus, it is not possible that there is a split that reduces the number of mistakes by at least one.



1. Yes, splitting with any of X1 or X2 or X3 gives a tree with entropy H(Y |Xi) = (1/2)[0] + 1/2[(1/4) log(4) + (3/4) log(4/3)] = 0.406 bits.

2 Entropy and Information



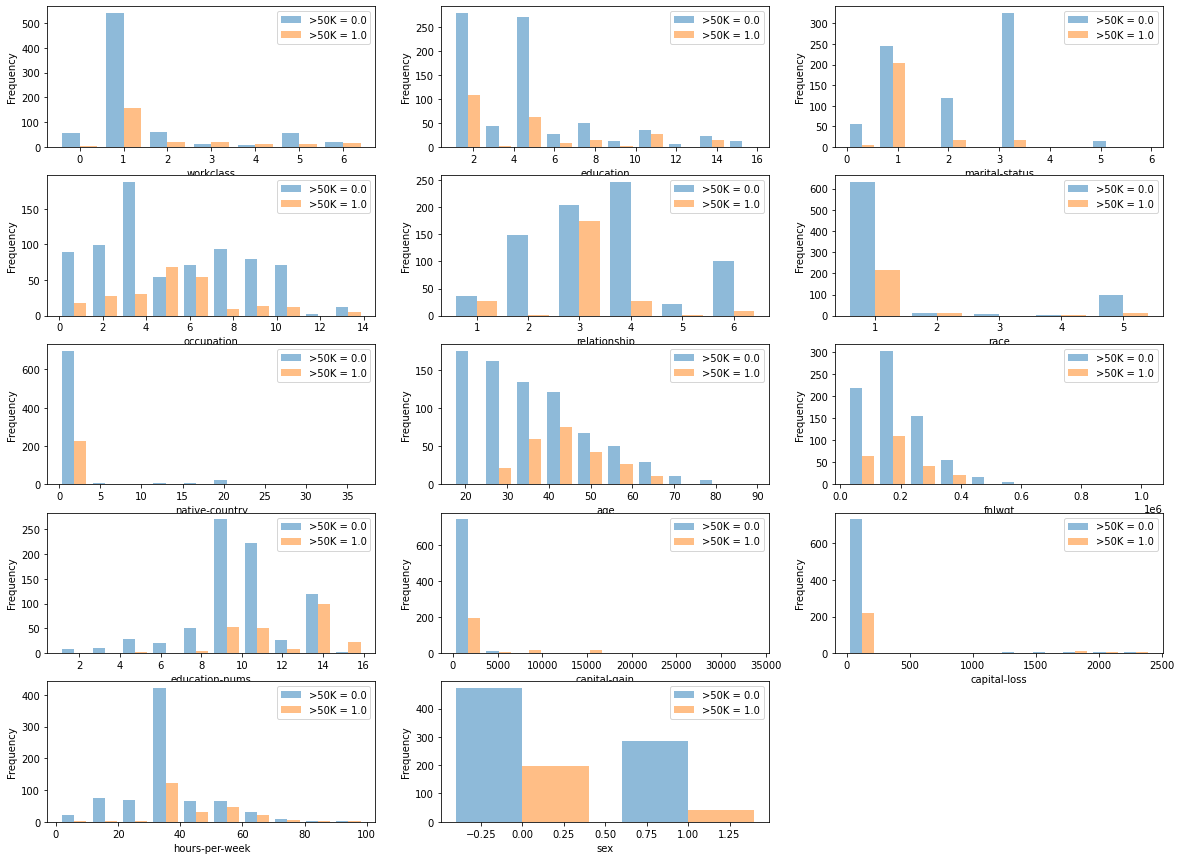


3 k-Nearest Neighbour

1. The value of k = 1(a data point is its own nearest neighbor) minimizes the training set error of the dataset. For this value of k, the KNN model closely follows the training dataset, thus minimizing the training error. The error is 0. Such a value of k however does not perform well on the test data, i.e. results in high test data, thus suggesting overfitting. Thus, the training set error is not a good estimate for the test set error.
2. The values k = 5 or k = 7 minimize the leave-one-out cross-validation error. In this case, the error is 4/14.
3. The LOOCV errors for different values of k:  
   Highest value of k: error is infinity   
   Lowest value of k: error is zero

Using too large a value of k would be underfitting and would not be able to accurately predict the data. Using a very small value of k would be overfitting and thus, the test data would not perform well on the model.

4 Programming Exercise : Apply Decision Trees and K-Nearest Neighbors

1. 

Analyzing the data:

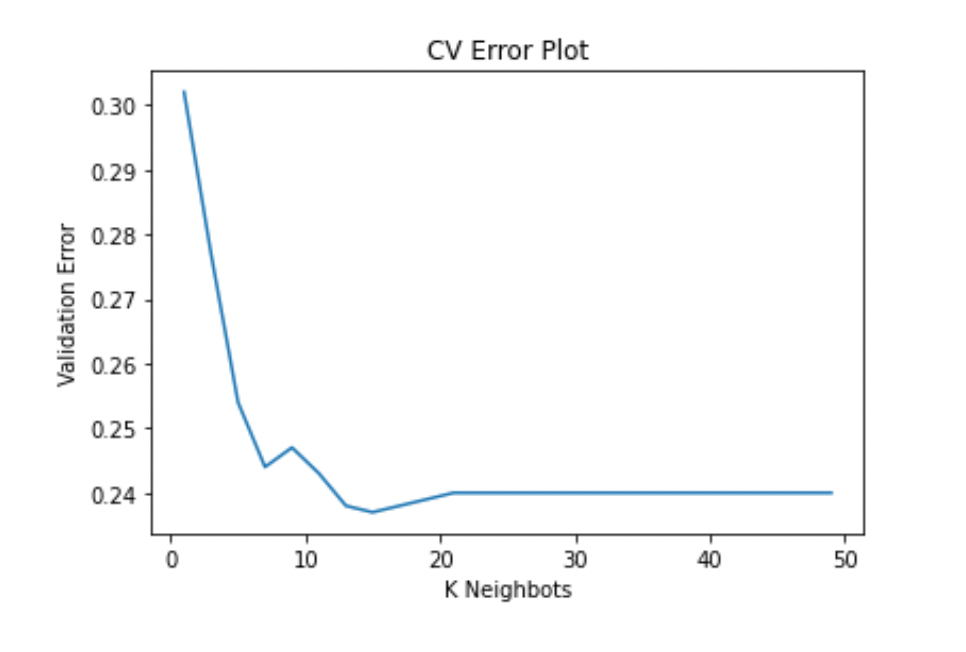
* Workclass: Majority of the people are self-employed. Government employees are most likely to earn >50k.
* Education: The higher the education, the higher the probability that they person earns >50k
* Marital Status: People who are divorced are most likely to make >50k.
* Occupation: The occupations that are most likely to earn >50k are Sales and Exec-managerial
* Relationship: Husbands are most likely to earn >50k
* Race: Although most people in the dataset are white, we see that Asian-Pac-Islanders have a pretty equal ratio for those earning below and above 50k.
* Native Country: Most of the people in the dataset are from the United States and earn less than 50k.
* Age: Younger people tend to earn <50k, and people in the age group of 40-50 are most likely to earn >50k.
* Fnlwgt: The fraction of people that earn <50k is greater than the fraction of people who earn >50k. More so, the fraction of people who earn more than >50k stays relatively constant.
* Education Nums: Majority of the people in the dataset have 8-10 years of education. Also, those with ≥14 years of education are most likely to earn >50k
* Capital Gain: Majority of the people in the dataset have low capital. However, those with high capital gain (≥10000) are more likely to earn >50k.
* Capital Loss: Majority of the people in the dataset have low capital loss. People earning >50k are spread across capital loss of 1000 and 2500.
* Hours-Per-Week: Majority of the people in the dataset work 40 hours per week, but those who work more, tend to earn >50k.
* Sex: Majority of the respondents are males, however looking at the ratio we can conclude that males are more likely to earn >50k.

1. By classifying using Random, we get the training error to be 0.374
2. By classifying using Decision Tree, we get the training error to be 0.000
3. By classifying using k-Nearest Neighbors, we get the training error
   1. k = 3  
      The training error is 0.153
   2. k = 5  
      The training error is 0.195
   3. k = 7

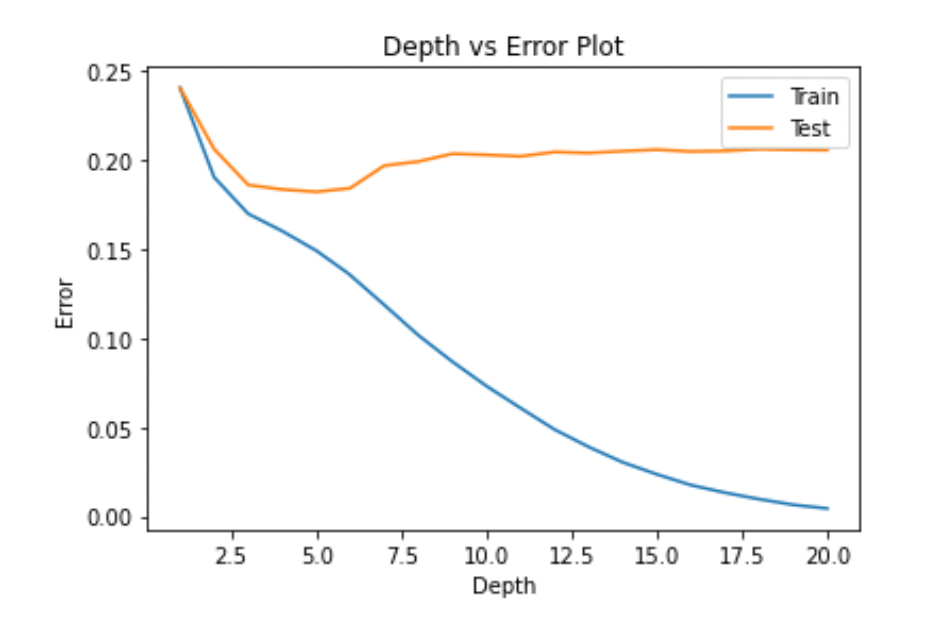
The training error is 0.213

1. Analyzing various classifiers, we reach the following results:

|  | Training Error | Test Error | F1 Score |
| --- | --- | --- | --- |
| Decision Tree | 0.000 | 0.205 | 0.795 |
| K-Nearest Neighbor | 0.202 | 0.259 | 0.741 |
| Majority | 0.240 | 0.240 | 0.760 |
| Random | 0.375 | 0.382 | 0.618 |

1. 

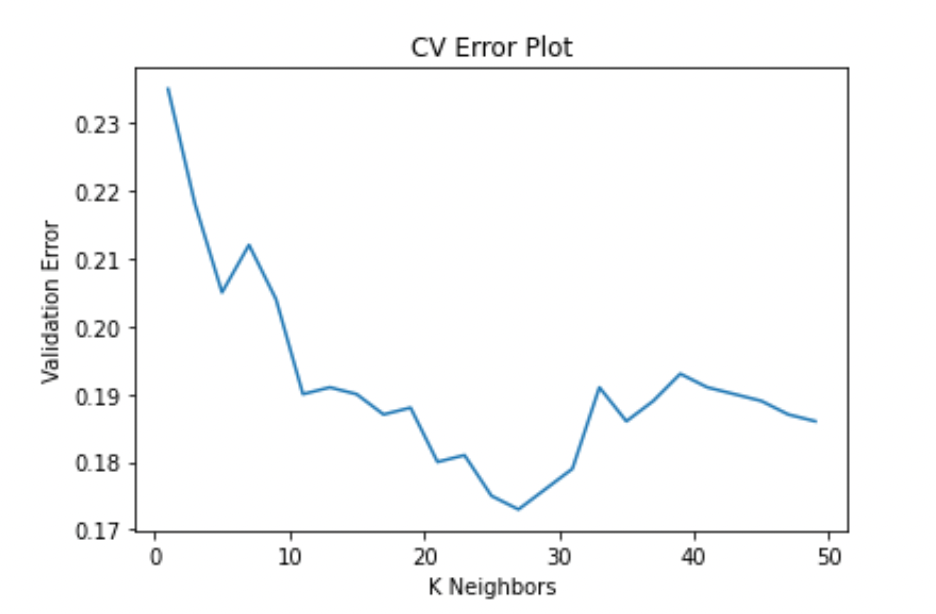
It is clear that upon running 10-fold cross validations for all odd numbers from 1-50, we see that the validation error generally decreases for k ≤ 15, and then increases in the interval 15 ≤ k ≤ 21. For k > 21, the validation error stagnates to approximately 0.242. Thus, the best value for k is 15, as that is the local minimum of the error plot above and has the least validation error.

1.   
   The best depth limit to use here would be ~5.0. We can see that before depth = 5.0, there is underfitting as the training error and test error are both very high (~ 0.20+). We can also see that for depth > 5.0, the training error reduces greatly, while the test error increases, indicating overfitting.
3. Pre-processing the data by standardizing it and redoing the parts b. to h. we get,

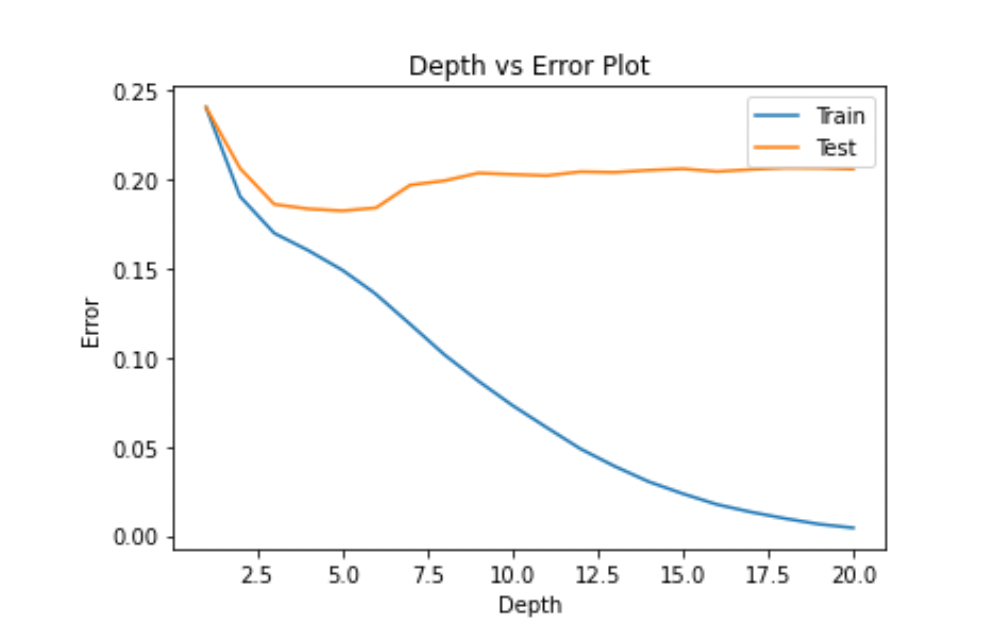
By classifying using Random, we get the training error to be 0.374  
By classifying using Decision Tree, we get the training error to be 0.000  
By classifying using Majority Vote, we get the training error to be 0.240  
  
By classifying using K-Nearest Neighbour, we get   
 k = 3 => Training Error: 0.114

k = 5 => Training Error: 0.129  
 k = 7 => Training Error: 0.152

|  | Train Error | Test Error | F1 Score |
| --- | --- | --- | --- |
| K-Nearest Neighbors | 0.133 | 0.209 | 0.791 |
| Decision Tree | 0.000 | 0.205 | 0.795 |
| Majority Vote | 0.240 | 0.240 | 0.760 |
| Random | 0.374 | 0.382 | 0.618 |



It is clear that upon running 10-fold cross validations for all odd numbers from 1-50, we see that the validation error generally decreases for k ≤ 27, and then increases. Thus, the best value for k is 27, as that is the local minimum of the error plot above and has the least validation error (~ 0.0174)



The best depth limit to use here would be ~5.0. We can see that before depth = 5.0, there is underfitting as the training error and test error are both very high (~ 0.20+). We can also see that for depth > 5.0, the training error reduces greatly, while the test error increases, indicating overfitting.