Homework 3

CSDS340

Ethan Fang

ewf22@case.edu

October 3, 2024

1 Problem 1

Height	**Hair**	**Eyes**	**Attractive?**
Small	Blonde	Brown	No
Tall	Dark	Brown	No
Tall	Blonde	Blue	Yes
Tall	Dark	Blue	No
Small	Dark	Blue	No
Tall	Red	Blue	Yes
Tall	Blonde	Brown	No
Small	Blonde	Blue	Yes

Table 1: Data

I will start with the Gini impurities for the first attribute, which is height.

$$Gini_{\text{small}} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{1}{3}$$

$$Gini_{\text{tall}} = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = \frac{12}{25}$$

$$Gini_{\text{height}} = \frac{3}{8} * \frac{1}{3} + \frac{5}{8} * \frac{12}{25} = \frac{17}{40}$$

Then hair:

$$Gini_{\text{blonde}} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = \frac{1}{2}$$

$$Gini_{\text{Dark}} = 1 - \left(\frac{3}{5}\right)^2 = 0$$

$$Gini_{\text{Red}} = 1 - \left(\frac{1}{1}\right)^2 = 0$$

$$Gini_{\text{hair}} = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$$

Then Eyes:

$$Gini_{\text{Brown}} = 1 - \left(\frac{3}{3}\right)^2 = 0$$

$$Gini_{\text{Blue}} = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = \frac{12}{25}$$

$$Gini_{\text{Eyes}} = \frac{5}{8} * \frac{12}{25} = \frac{3}{10}$$

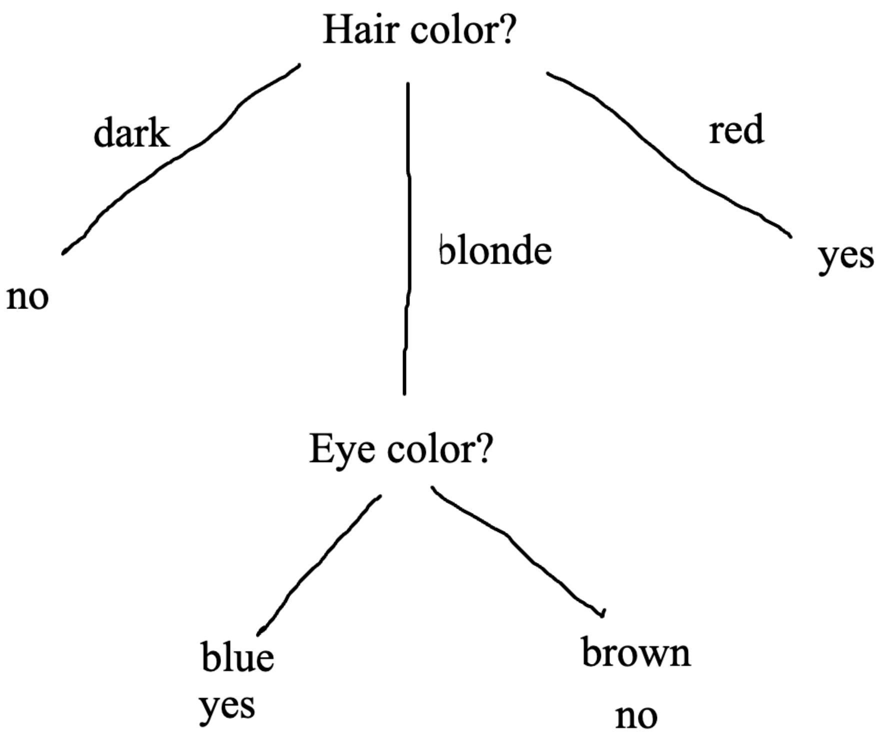

Figure 1: Politically Incorrect Decision Tree on Attractiveness

2 Problem 2

We are asked to compare the graphs of a decision tree vs. a perceptron on the logical AND function using entropy as the splitting criterion. I wrote a python script to do this for me: and here are the screenshots of my results.

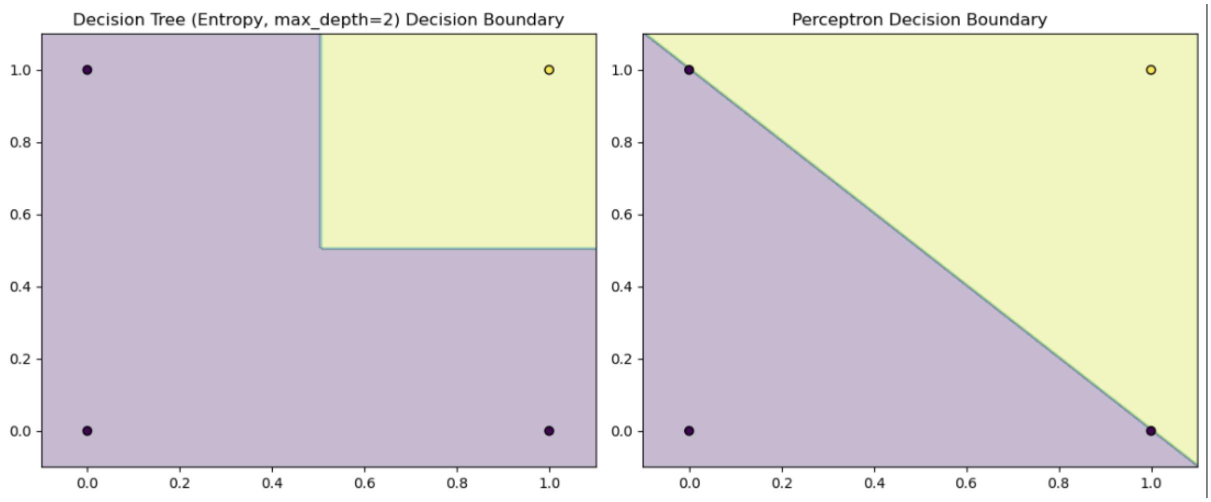

Figure 2: Decision Tree vs. Perceptron Boundaries on Logical AND

As we can see, the decision tree is not limited to linear boundaries, so it was able to "box" out the (1,1) point. The perceptron, on the other hand, must generate a straight line boundary. This works for the logical AND operator, since you can use a diagonal line, but may not work for some not-linearly-separable datas. The results show the difference in how these 2 algorithms function.

3 Problem 3

The curse of dimensionality is evident as the number of dimensions increase in the graph.

The Unit Hypershere is the area where points are ≤ 1 unit away from the origin. The fraction of points lying within the unit hypersphere rapidly decreases as the number of dimensions increase. In lower dimensions, points close to the origin are more likely to be within the unit circle. However, as the number of dimensions increases, more volume is spread from the center. This can be observed in the graph, where the proportion of points in the hypersphere take a big hit every time we increase a dimension.

The mean pairwise distance also increases as the number of dimensions grow. This is because in higher dimensions, points have "more ways to go" and therefore become more spread out. While the nearest neighbor may still be closer than other points, it becomes increasingly further away because they just have more directions to move in. As a result, even the closest neighbor in a high dimensional space is considered far compared to lower dimensional spaces.

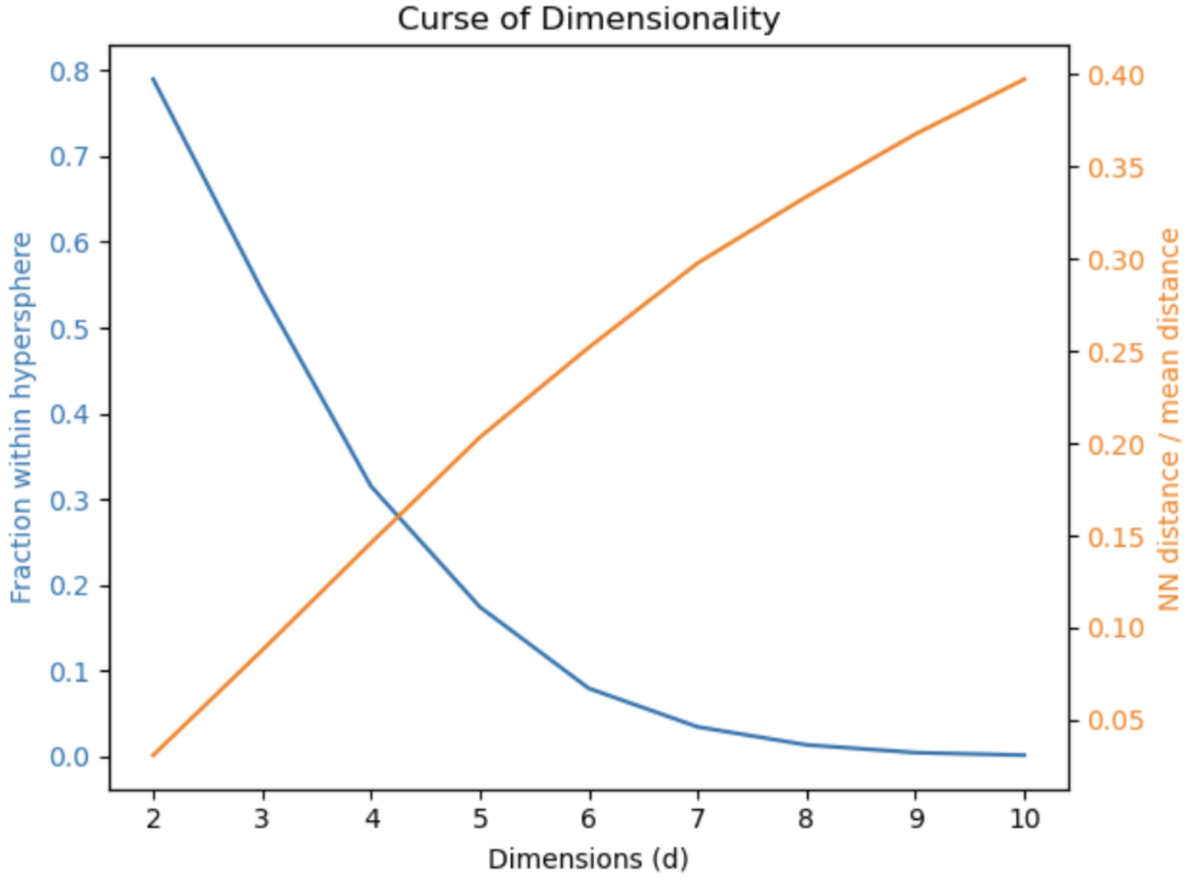

Figure 3: Curse of Dimensionality Analysis

4 Problem 4

For problem 4, we were asked to fit a K-Nearest-neighbors classifier to the pima indians dataset from Assignment 1. I used many different scalers including Standard Scaler, MinMax Sclaer, Normalizer, Power Transformer, Robust Scaler, and MaxAbs Scaler. I tried k values from range(1,20) for each one, and tested many different distance metrics including euclidean, manhattan, chebyshev, minkowski, and cosine similarity. I also tried removing outliers by calculating the z-score of the data (but this didn't help me very much)...

In the end, my highest accuracy was 0.776 with the Standard Scaler, k=15, and the manhattan distance metric.

5 Problem 5

This problem I found to be the trickiest, just because of working with sklearn. I tested the decision tree classifier with many types of imputation, including iterative, KNN, mean, median, constant, mostfrequent, and even no imputation at all. For the KNN Imputation, I also tested many different values of neighbors, and played around with the maxdepth of the decision tree classifier as well (which I found the most accurate to be at maxdepth=3). Here is a graph I generated with my results:

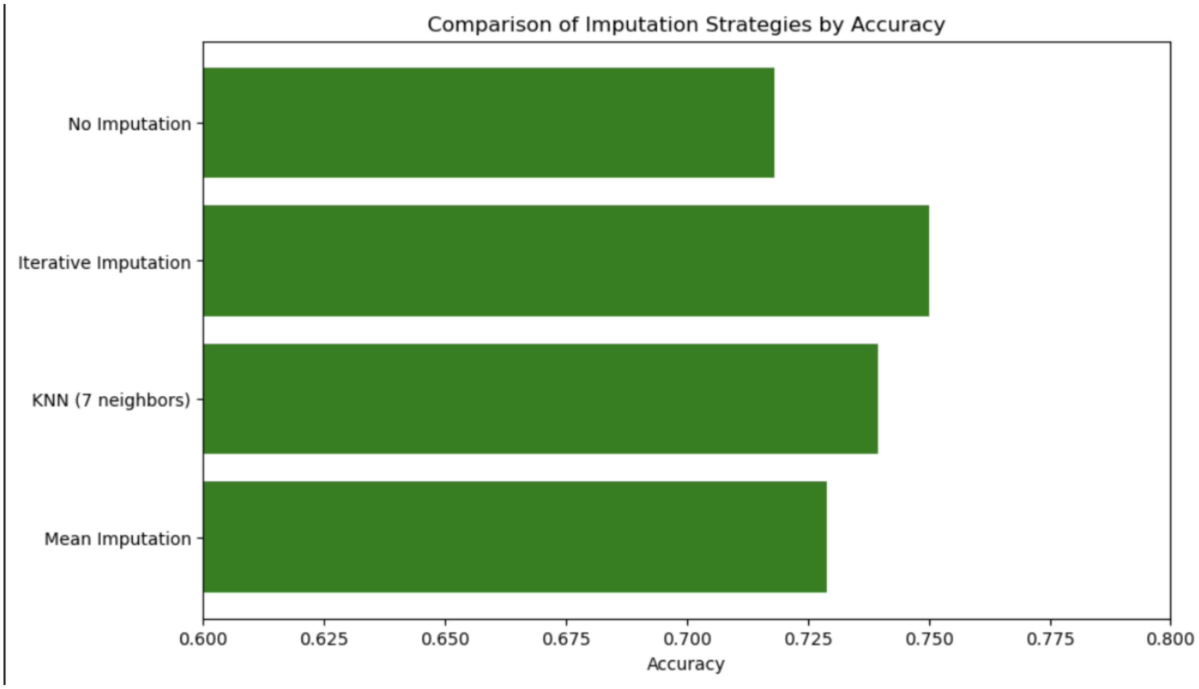

Figure 4: Imputation Analysis

I didn't graph all of the different imputation methods that I tried, just because I felt that some accuracies weren't worth showing. Using the SimpleImputation function, I found mean to be the best one, so that was the one that I showed.

Something that I found very interesting was how accurate "no imputation" was. When discussing it with Professor Xu, It seems that sklearn handles missing data in the Decision Tree Classifier by either choosing to "always go left" or "always go right," which is really interesting how accurate the model ended up being. Thankfully, of course the imputation methods provided higher accuracies than the no imputation model, but it is interesting how relatively accurate it was.