

Azalea Rohr

Homework 9

April 8st, 2024

Collaboration: Angelo Amato

Sources: None

Question 1

The solution to this problem works since it first generates a text file with random data and then it reads from that file. Once that data has been created, it predicts the class for each data point based on the split point and the class direction. It also determined the best-split point and class direction combination by considering all of the possible combinations and finding the accuracy for each combination. It also checks the possibility of each point being either class 0 or 1 and sees if it is less than or greater than the partition. The time complexity of reading the data and finding the best split (with the predict function), which involves iterating over all of the possible combinations is $O(n)$. As the code iterates over the data points multiple times, the full operation is $O(n^2)$. So the complexity of the solution is $O(n^2)$.

Below is the sample output code after running cargo run

The best partition for the data is to split at 779531 with any $x \leq 779531$ being classified as class 1.

Any $x \leq$ the partition will be classified as 1. Any $x >$ the partition will be classified as 0.

The accuracy of this partition is 0.526.

After running cargo run 5 times, the average accuracy of the partition is 0.543.

For my tests, I did 4 different tests, each test for specific functions of the decision tree implementation. The first test, test_predict, evaluates the prediction function by looking at a known data set that was provided where the accuracy of the partition point and class direction is expected to be 0.6 when the split point of 0 is used. This test verifies if the predicted function can accurately compute the accuracy when the partition and class direction are given. The second test, test_best_split, looks at the best_split function using another predetermined dataset and tests to make sure the function correctly identifies the best-split point and class direction. For the second test is it expected to be (0, 0, 0.7). The test_predict_one_class_left test checks the predict functions performance when one class is only on the left side of the partition. This test aims to check that the predict function handles cases where the classes are imbalanced. And the 4th test, test_off_center_left, checks the behavior of the best_split function with the provided data set that is skewed left (towards negative numbers), checking its ability to

identify the best-split point and class direction accurately. These 4 tests are designed to validate the correctness of the decision tree with multiple scenarios.