CODE REPORT

Theoretical time complexity

As the time complexity is merely the upper bound of the implementation time. It might look to disagree with the practical time for smaller values So, let us look at the theoretical time complexity from the code and analyze:

Real input Real output:

Construction of Decision tree: Let us suppose the criterion as information gain as given in the code and the function's input is on a series of size N and M features. Here if we start out with the calculation of all possible splits(N) to find out max info gain it takes O(N**2) and for each feature O(M). So, Total time complexity: O(M*N**3) and the extra N for varying the N value from 0 to N.

Prediction: As we recursively search for the output value down the tree, this is pretty much equivalent to the height of the tree. Thus, the time complexity is O(N)

Real Input Discrete Output:

Construction of Decision tree: The above explanation holds true for this as this falls under the same input hood. Thus Total time complexity: O(M*N**3)

Prediction: The above explanation holds true, Thus, the time complexity is O(N)

Discrete Input Real Output:

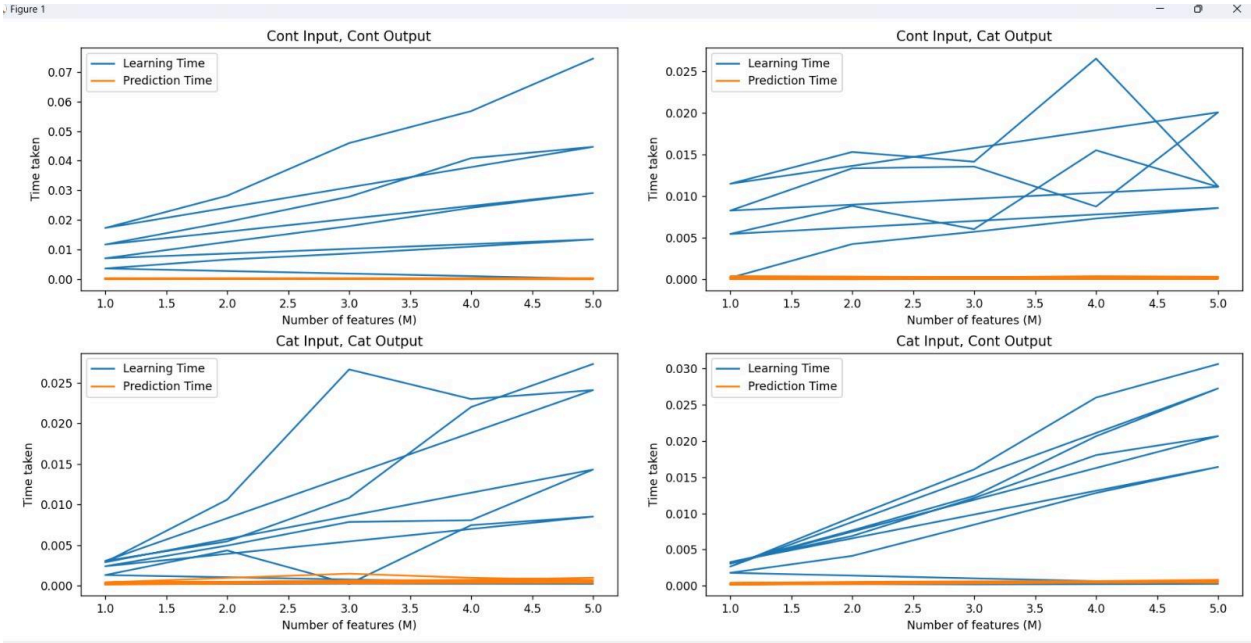
Construction of Decision tree: The information gain is calculated for each feature so it takes O(M). Now we need to compute for a series of N and for each calculation it is less than N samples So, The time complexity will be O(MN**2)

Prediction: Here the height of the tree is limited by both N and the features So, O(M)

Discrete Input Discrete Output:

The above explanation holds true for Construction of Decision trees and the same applies to Discrete output.

Practical Complexity vs Theoretical complexity:



As discussed the practical is as good as the theoretical one due to smaller value it just got squeezed where as the construction one doesn't show any doesn't show any as such but with the rise in the N it sure gets a significant rise similar to the one.(the above values are so small to observe the difference).