Jared Frazier | CSCI 4350 | OLA 3 | Supervised Learning with Decision Trees | Due 2021/11/12

Introduction

When identifying a flower that you see in the wild, one might consider several factors such as its size, color, the season in which you encounter the flower. These factors are intrinsically related, that is the color will be invariable linked to the environment and season—since, naturally, a flower that is quite grey or dying is likely "out of season." This iterative process of decision making can be visualized as a simple tree, where the branches (arcs) of the tree between different leaves (nodes) inform the decision on what kind of flower you have encountered in the wild.

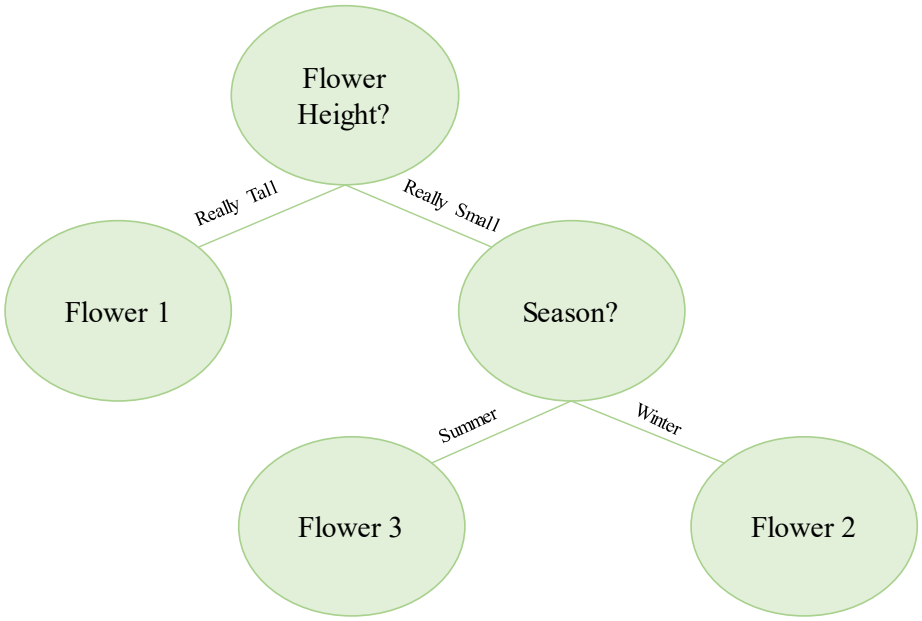

Figure 1. Sample Decision Tree

Note, when a leaf in the tree is has no branches after it, that is a leaf with no more green lines pointing down and away from it, then the leaf is called a terminal node. The terminal nodes in such a tree represent decisions, while non-terminal nodes represent some descriptive information, or feature, about whatever it is that is being observed. The branches are categories that are directly related to their parent feature. For example, the "Flower Height" feature has only two categories: "Really Tall" and "Really Small." The nature of these categories and the current feature of node determines the resulting decision.

Methods

The construction of such a tree involves the maximation of information gain: that is non-terminal nodes and terminal nodes should be constructed based how certain a user is about a particular feature or decision.

One such method to construct a decision tree is known as the Iterative Dichotomiser 3 algorithm [1]. The algorithm proceeds by adding a learning set, or data with features and labels—labels being the resulting classification associated with a set of features—to a node, computing the information entropy—a measure of uncertainty in the labels—associated with the node, and then iteratively evaluating the a given features distinct categories for their entropy relative to the label's entropy [2], [3]. In this way, a decision tree can be constructed, by searching (traversing) the tree.

In constructing the decision tree, one must carefully decide on the data structures, or internal parts of the tree itself. The tree, being made of nodes, must therefore be constructed with nodes that contain information which can be used to construct more decision sub-trees or to direct the tree search for a decision.

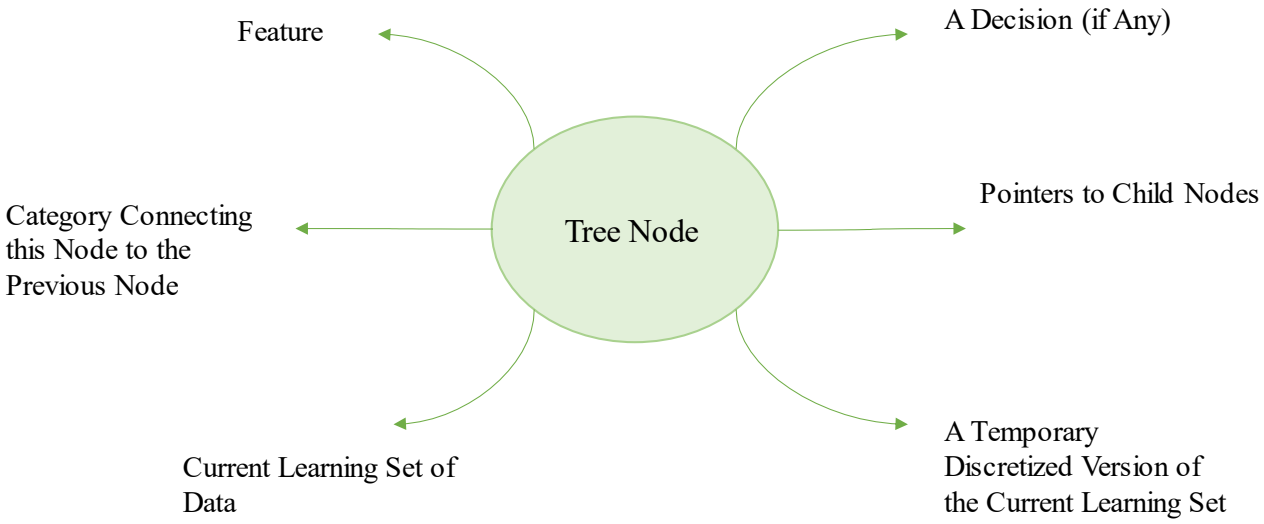

Figure 2. Tree Node for ID3 Decision Tree

Moreover, in the evaluation of the decision tree, different random samples of the data set were used for construction and testing. In this way, a more statistically robust measure of my algorithm's performance was achieved, per the lab specification.

Two data sets were used in the construction of this tree: the iris flower dataset and the breast cancer diagnostic data set [4]. The objective for the constructed decision is to decide on what kind of iris and if cancer is present given a number of features, respectively.

My particular algorithm was conceptually influenced by a number of sources [1]–[3], [5], [6]

Results and Discussion

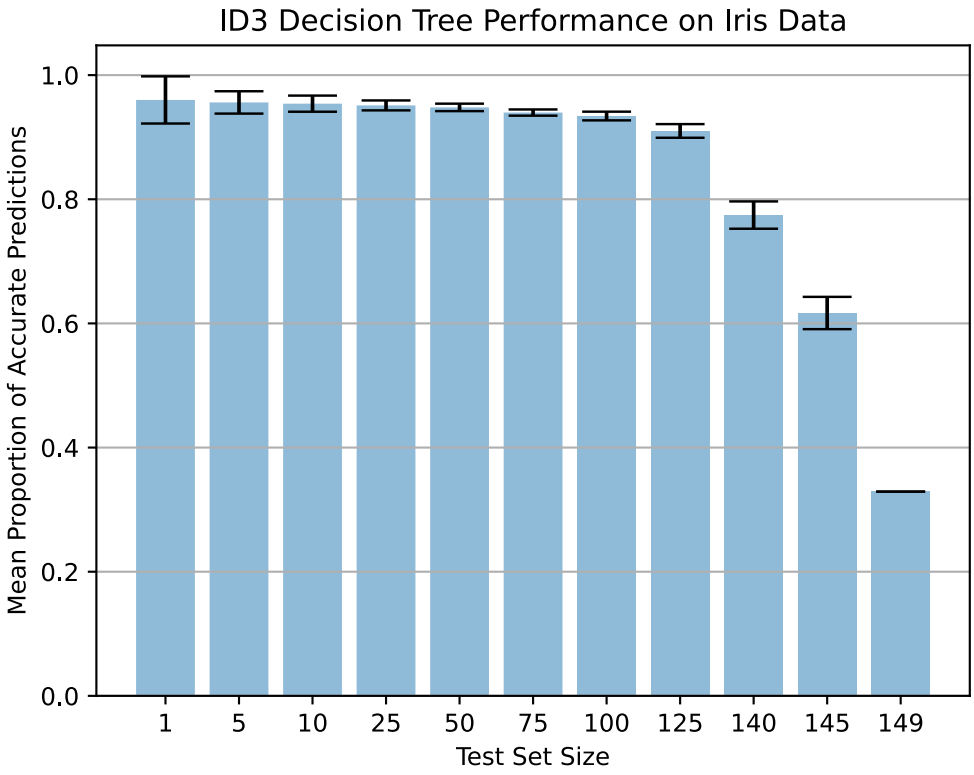

*Figure 3. Performance on iris data. Error bars determined by 1.96 * standard error.*

Figure 3 demonstrates my algorithm's performance on the cancer dataset. There is a distinct, but non-linear decline in performance as the test set size increases. This is to be expected since in supervised learning problems—problems where features and labels are known—a lack of construction (or learning) data for the model can lead to poor performance. The iris data set is also relatively simple in that it only contains four features (or dimensions). Eliminating the number of factors that can influence a decision invariably simplifies the decision trees.

```
TREE
[2:None:None]
├── [None:(-inf, 2.3):0.0]
└── [None:[2.3, inf):1.0]
```

```
TREE
[2:None:None]
├── [None:(-inf, 2.45):0.0]
└── [3:[2.45, inf):None]
    ├── [0:(-inf, 1.75):None]
    │   ├── [1:(-inf, 7.1):None]
    │   │   ├── [None:(-inf, 2.8499999999999996):1.0]
    │   │   └── [None:[2.8499999999999996, inf):1.0]
    │   └── [None:[7.1, inf):2.0]
    └── [None:[1.75, inf):2.0]
```

Moreover, the sample of data will influence the construction of the tree. The above tree figures show the construction of two distinct decision trees for the iris data set. The right decision tree was constructed using the entire data set, while the left decision tree was constructed using a random sample of 55 rows (from 150) for its construction.

The tree figure shows three aspects of a node from left to right: (1) the attribute to which a node belongs—iris petal size, for example—(2) the category of the previous node, and (3) the decision for a particular node. Notably, the data set contained continuous (i.e., numerical data). Thus, the categories for nodes correspond to potential ranges that value might fall into based on some threshold. It is therefore shown that a decision tree can be quite simply constructed given a particular set of data from the same population, but such a tree may be over fitted to that sample, thus leading to poor generalization. That is, in the construction of the decision tree, its decisions and leaves are naturally well-suited for the set from which it was constructed, but *not* necessarily for other sets from the same population of sets.

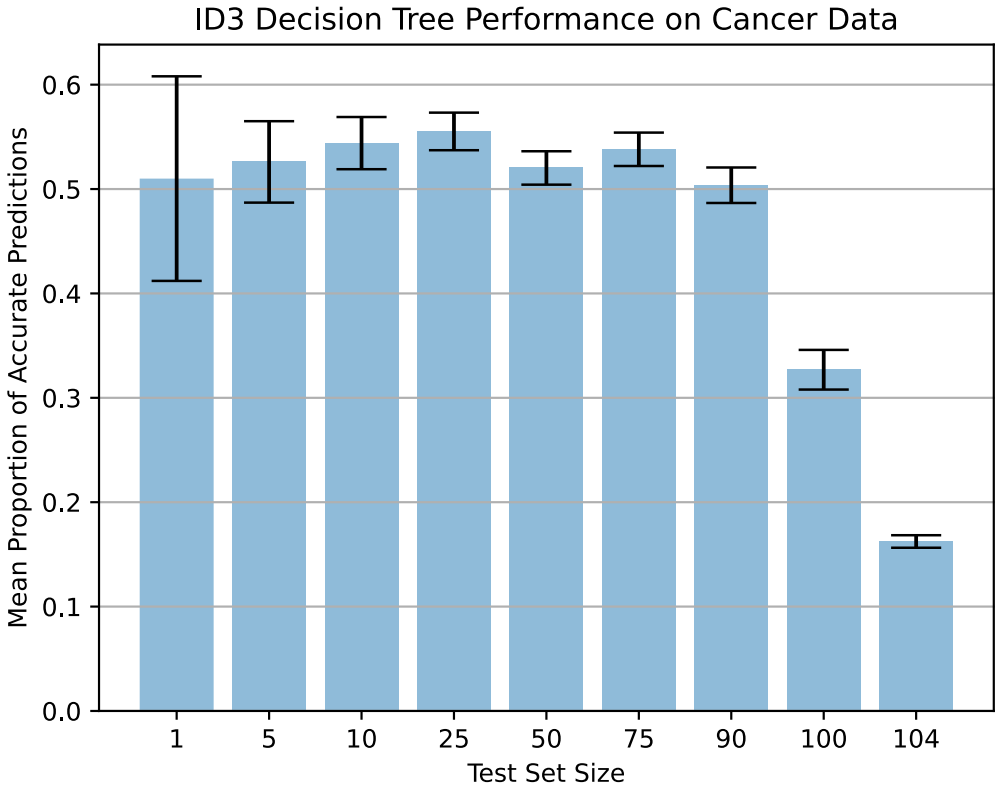

*Figure 4. Performance on cancer data. Error bars determined by 1.96 * standard*

The decision tree was also tested on the cancer data set, with the results shown in Figure 4. The performance on the cancer data was significantly worse compared with performance on the iris dataset. In the iris data set, the decision tree at its best correctly classified leaves with an accuracy rate of about 97%, while for the cancer dataset, the maximum average classification for cancer was around 55%. I suspect this performance drop is due primarily to two-reasons: (1) the cancer data set is far more complex, having nine features from which a decision is constructed,

and (2) in attempting tie-breaking mechanisms for maximizing information gain, I believe I likely made an implementation error that I could not identify.

This project was critical for illustrating a simple supervised learning task as well as a exploration of careful debugging. The algorithm has many moving parts, so the use of Python's 'pdb', 'logger', and 'anytree' libraries were critical for development.

References

[1] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986, doi: 10.1007/BF00116251.

[2] W. Peng, J. Chen, and H. Zhou, "An implementation of ID3-decision tree learning algorithm," *From web. arch. usyd. edu. au/wpeng/DecisionTree2. pdf Retrieved date: May*, vol. 13, 2009.

[3] R. Bhardwaj and S. Vatta, "Implementation of ID3 algorithm," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 6, 2013.

[4] D. Dua and C. Graff, "UCI Machine Learning Repository." 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[5] N. Mantri, "Using ID3 Algorithm to build a Decision Tree to predict the weather," *OpenGenus IQ: Computing Expertise & Legacy*. OpenGenus IQ: Computing Expertise & Legacy, Jun. 2021. [Online]. Available: https://iq.opengenus.org/id3-algorithm/

[6] Wikipedia contributors, "ID3 algorithm — Wikipedia, The Free Encyclopedia." 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=ID3_algorithm&oldid=1008669949