Decision Tree - Machine Learning - Rylie Johnson

Classifying Synthetic Data

The implementation choice I struggled with the most was the way in which I wanted to represent my data. I initially considered a struct with members for class label and count, but in the end I opted to use a list of lists. Even so, I had to decide between representing them as [[v1, label], [v1, label], …], [[v2, label], [v2, label], …]] or [[v1, v1, v1, …], [v2, v2, …], [label, label, …]]. I ended up going with neither of these, and chose to work with the data as a list of 'points', each point containing 1 value for each attribute and the label. This worked out well for me, as I could assume in all my functions that the class label would be at the end of the list.

The type of the data threw me for a loop; upon importing the data, I assumed everything would be floats or integers, but they were actually strings. It took multiple go-arounds of typecasting to get everything to work.

I also struggled with the binary tree itself. I was hoping to write it myself, but in the end opted to use the anytree library. I had trouble importing it at first, but once I got it working I found it very convenient.

I'm still learning how to use Python, so I ran into some unexpected results when creating lists - I had many, many bugs that came from an empty list appearing where I didn't expect it to.

Training set error:

1) Error 0.0 / accuracy 1.0
2) Error 0.0 / accuracy 1.0
3) Error 0.026 / accuracy 0.974
4) Error 0.025 / accuracy 0.975

Classifying Pokemon

I ran into many of the same problems as with the synthetic data, like how to represent the lists and type errors.

It took quite a bit of tweaking to adapt my synthetic data functions to work with the Pokemon data. I had written everything to work with what I assumed to be integers, and hard-coded in the labels I was expecting, '0' and '1'. I was able to generalize the functions such that, rather than defining the labels myself, it simply referenced what was present in the data set.

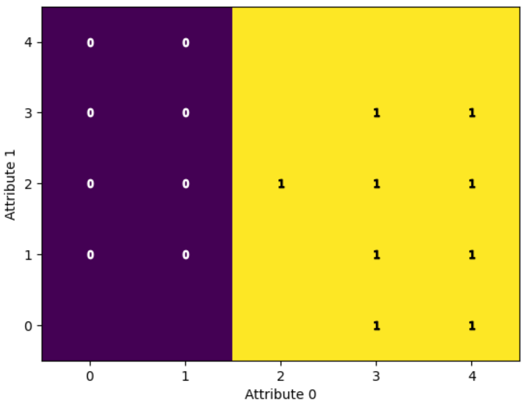
Training set error:

Error 0.0102 / accuracy 0.9898

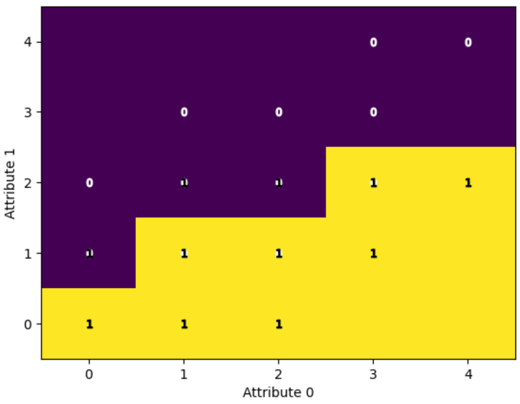
Visualizing Synthetic Data

‘0’ class labels are represented with white zeros, and ‘1’ class labels are represented with black ones. The purple areas correspond to the function approximation of ‘0’ class labels, and the yellow with ‘1’.

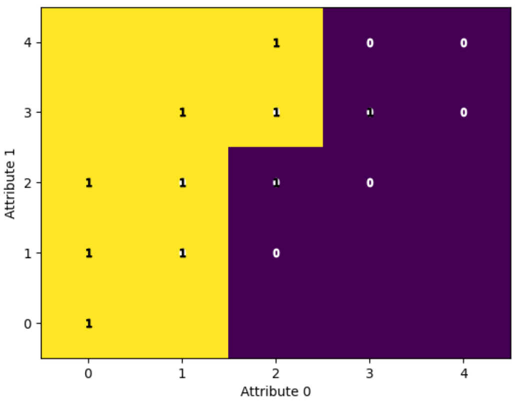
synthetic-1-discrete.csv



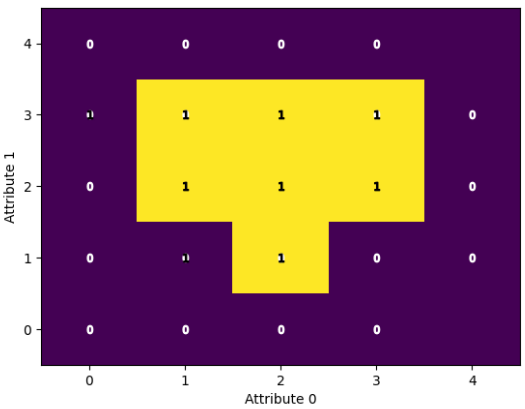
synthetic-2-discrete.csv



synthetic-3-discrete.csv



synthetic-4-discrete.csv


Attribute 1
Attribute 0