

## 1 Random importance

When using a random importance function the tree, is as expected, random. The trees also grows very big. The range of nodes spans from as low as 40 and as high as 120, depending on the order the attributes are chosen. Considering that  $2^7 = 128$  is the maximum, the random importance function is not very space efficient. The tree is thus too big to be represented in the report, but can be seen when running the code.

The results from running the tests after learning the decision tree varies. The accuracy ranges from 50% – 100%.

## 2 Expected information gain

The expected information gain performs much better, consistently creating a tree of just 9 nodes as can be seen in figure 1.

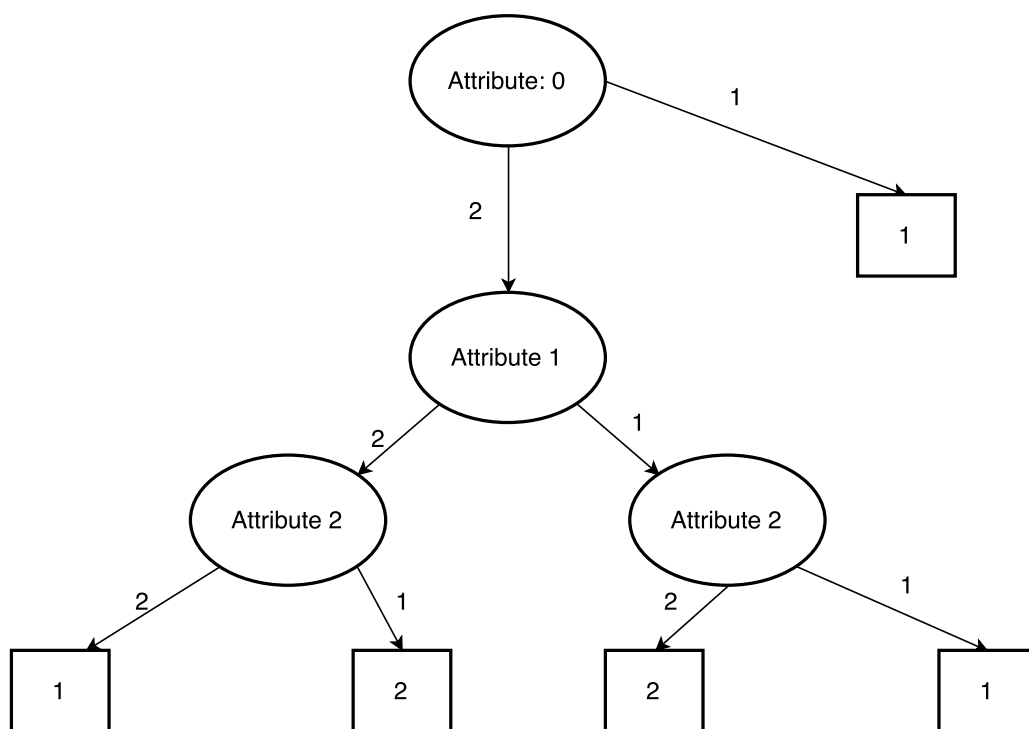


Figure 1: The graph after running the training data with the information gain function

After learning the decision tree, the test was run and gives 100% accuracy every time.

## 3 Discussion

The expected information gain is the importance function that gives the best accuracy, with an always 100% accuracy. The random function is random and performs anywhere in the range 50% – 100%. The advantage that the random importance function has is that it requires less computation when choosing an attribute, but at the cost of (much) more storage.

Since the random importance function is random, the results I receive is random every time.

The information gain importance function is consistent and gives the same result every time.