

# Assignment 6

|         |            |
|---------|------------|
| ▼ Class | TDT4171    |
| ▼ Type  | Assignment |

## Implementation of Learn-Decission-Tree

Using the implementation of `Learn-Decission-Tree` provided in `assignment_6.py`, we see that with `measure = "information_gain"` we get a training accuracy at 0.74, and test accuracy at 0.7857, consistently. With `measure = "random"`, the accuracies are more varying. However, they are quite high, both values varying from approx. 0.68 to as high as 0.94.

One would assume that with more information for a heuristic, rather than a random heuristic, would provide better accuracy. Here we see that that is not always the case. This could be caused by multiple reasons:

1. The training sets are not that big. Random heuristics could therefore more often “accidentally” give a quite good heuristic.
2. The Learn-Decission-Tree algorithm is greedy. Greedy algorithms works well in some cases, but is not guaranteed to give the best solution. A random heuristic could therefore randomly generate a better heuristic. Even though the greedy algorithm here works pretty well.

From some testing, I observed that the importance based on a information-gain heuristic performed better than importance based on a random heuristic in most cases, but some times the random heuristic beat the information-gain heuristic.