

# ARTIFICIAL INTELLIGENCE METHODS

## Assignment 6

SINDRE LIAN THRONÆS      SINDRELT@STUD.NTNU.NO

Course: TDT4171  
Term: Spring 2023  
Handed in: 06.03.23

**1   Exercise 1 - Importance() and LearnDecisionTree()      1**



Norwegian University of  
Science and Technology

Department of Computer Science  
Norwegian University of Science and Technology

## 1 Exercise 1 - Importance() and LearnDecisionTree()

**I have not succeeded in implementing the LearnDecisionTree-algorithm and thus I haven't been able to compare the accuracy of both models.** Still, I will share my intuition on how I think the performance is affected by the randomized-importance-algorithm and importance using information gain to choose the best attribute to split upon.

I expect the performance of the random version of *importance()* to be quite bad. Firstly, I assume it will generate a very large tree with lots of nodes. This tree would be quite hard to traverse and also it would take up unnecessary memory. Secondly, it would be inefficient to use for classification. This would largely be due to a poor choice of attribute-index to split upon, giving birth to many excessive recursive calls of *LDT()*.

Comparing the information-gain version to the random version, I would expect the information-gain version to outperform the random version 9/10 times. The main difference would most likely be the total amount of recursive calls and thus a much more sparse tree. This tree would be easier to traverse and faster to use for classification. There are certainly a possibility that the randomized version will be lucky and pick the same order of attributes to split upon and due to the complexity of the "choose" be faster, but information-gain guarantees to select the attributes with the most information gain of the given list of attributes.