Implementation of an algorithm to learn decision trees.

# Introduction

What the problem is what I am doing why interesting and what my contribution is

# Background

Explain:

* Binary Classification instance
* Decision tress
* learning decision trees

Decision trees

Decision trees represent a hierarchical structure for decision-making, where each node signifies a decision based on certain attributes, and the branches denote the outcome of these decisions (REFERENCE). The process initiates from the root node and progresses through internal nodes (representing attributes tests) to the leaf nodes, which hold the decision outcomes – class labels for classification tasks or continuous values for regression tasks. This process of systematically dividing the entire dataset into smaller subsets mirrors the divide and conquer approach often used in problem solving. By applying this method decision trees aim to simplify complex dataset into subsets that are easier to analyse and make predictions from.(REFERENCE)

The simplicity and interpretability of decision trees make them suitable for a wide array of applications:

Classification and prediction: Their primary use in categorising instances into distinct classes based on attribute values supports applications in a variety of fields such as medical diagnosis and customer segmentation, among others (reference)

Regression: Beyond classification, decision trees can be predictors of continuous outcomes, making them invaluable tools in forecasting sales, evaluating real estate prices, and other quantitative analyses (reference)

The representation of decision trees is both straightforward and visually engaging, enhancing their appeal:

Nodes: The nodes in the decision tree are the entities that make decisions. The root node represents the entire dataset, internal nodes correspond to attribute tests which split the dataset, and leaf nodes represent the outcome of these decision paths.

Branches: These represent the decision outcome at each node, guiding the path later nodes or leaf outcomes

This structured approach allows decision trees to transparently communicate the logic behind decision making processes, which makes it easier to grasp and apply them in a variety of disciplines (reference)

DIAGRAM EXAMPLE

Binary Classification Instance

Talk about:

* Algorithms which are known 2/3, show my understanding, link to my work
* 2 types, heuristic, exact
* Show smallest decision trees are needed

Explain:

* Algorithm to implement.
* How algorithm works with pseudo code and running time
* Complexity – parameterised , what is this
* Completeness of the algorithm

## Method

* Design of solution
* How I implemented, project management methodology
* What improvements I did
  + Extended domain to discrete
  + Removed symmetries.
  + Error that caused loop
* How I showed that its correct
* Challenges of implementation
* Evidence of good coding practice and version control

# Results and observations

* Detailed description of software implementation with justifications
* Run it on different benchmarks tests
  + How long it takes.
  + Compare to SAT algorithm
* to find what s is
  + From 1 and go up to find solution
* Extended domain to discrete, performance of this
* Testing structure, how we test the correctness

Conclusion

* Ideas for future work

Problem definition

* Goal is to decision tree with smallest mumb

# Self-appraisal:

* Considered how my work could be extended?
* Included mature personal reflection which leads to lessons learnt?
* Suggested how the problems encountered might be avoided?
* Considered legal, social, ethical and professional issues, with justification if one or more are not relevant?

Remove symmetries.

Enumerate everything to avoid do same thing again.

Do some kind of ordering.

Avoid repeating features in branch

Getting same trees in different ways/orders

Have an ordering on the feature

* **Don’t consider putting a feature that is smaller in the ordering above(higher in tree) a feature that is larger, only on the path you are extending,**
* **In report show we still generate all possibilities but we don’t generate the same tree twice**

**For the error, when inserting feature, make sure everything below is reachable before adding, else skip**