ML Lab Assignment 5: Decision Tree Algorithms – CART, ID3, and C4.5

25MCMI20, Sachin A

27/02/2026

1 Introduction

In this experiment, I implemented and compared three decision tree algorithms: CART (Classification and Regression Trees), ID3 (Iterative Dichotomiser 3), and C4.5. All three were applied to the same small dataset representing a student activity prediction problem. The goal was to understand how each algorithm differs in the way it selects splitting attributes, and to observe how the resulting decision trees differ from one another.

Additionally, the second and third questions of the assignment explore a scenario where two attributes have equal Information Gain and what happens when the tree is built using only those two attributes.

2 Dataset Description

The dataset used in this experiment is a small dataset with 11 instances. Each instance describes a situation and the activity my friend might engage in. The attributes are:

- **Weather**: The current weather condition (*sunny, windy, rainy*)
- **Parents**: Whether the friend's parents are visiting (*visit, no-visit*)
- **Cash**: Whether the friend has money (*rich, poor*)
- **Exam**: Whether the friend has an upcoming exam (*yes, no*)
- **Activity**: The activity the friend chooses (target variable: *cinema, tennis, shopping, stay-in*)

All attribute values are categorical.

3 Libraries Used

- **math**: Used for computing logarithms in entropy and information gain calculations.
- **Pandas**: Used to create and manipulate the dataset as a DataFrame.
- **Matplotlib**: Used to visualize the decision tree produced by CART.
- **Scikit-learn**: Used for the CART implementation via `DecisionTreeClassifier` and for label encoding.
- **collections.Counter**: Used to count label frequencies when calculating entropy.

4 Data Validation and Scope

Before running the algorithms, the dataset was validated to ensure all attribute values fall within the predefined domain. A `validate_dataset` function was implemented to check each column against a dictionary of allowed values. If any value falls outside the valid set, an error is raised and processing is halted. This prevents incorrect or out-of-scope data from being used during training.

```
VALID_VALUES = {
    "Weather": {"sunny", "windy", "rainy"},
    "Parents": {"visit", "no-visit"},
    "Cash": {"rich", "poor"},
    "Exam": {"yes", "no"},
    "Activity": {"cinema", "tennis", "stay-in", "shopping"}
}

def validate_dataset(df, valid_values):
    errors = []
    for col, valid_set in valid_values.items():
        out_of_scope = set(df[col].unique()) - valid_set
        if out_of_scope:
            errors.append(f"Column␣'{col}'␣has␣out-of-scope␣values:␣
               {out_of_scope}")
    if errors:
        for e in errors:
            print(f"[VALIDATION␣ERROR]␣{e}")
        raise ValueError("Dataset␣has␣invalid␣values.")
    else:
        print("[VALIDATION␣PASSED]␣All␣values␣are␣within␣expected␣
            scope.")
```

5 Question 1: Implementing CART, ID3, and C4.5

5.1 CART Implementation

CART was implemented using the `DecisionTreeClassifier` from Scikit-learn with the Gini impurity criterion. Since Scikit-learn only accepts numerical inputs, the categorical values were first encoded using `LabelEncoder`.

The Gini impurity for a node is calculated as:

$$\text{Gini}(S) = 1 - \sum_{i=1}^{c} p_i^2 \tag{1}$$

where p_i is the proportion of samples belonging to class i in the node.

The model was trained on the entire dataset and the training accuracy was computed. Since the dataset is small and we are using the training set itself for evaluation (since the dataset is so small that there is no point in doing a split of the dataset), a high accuracy is expected.

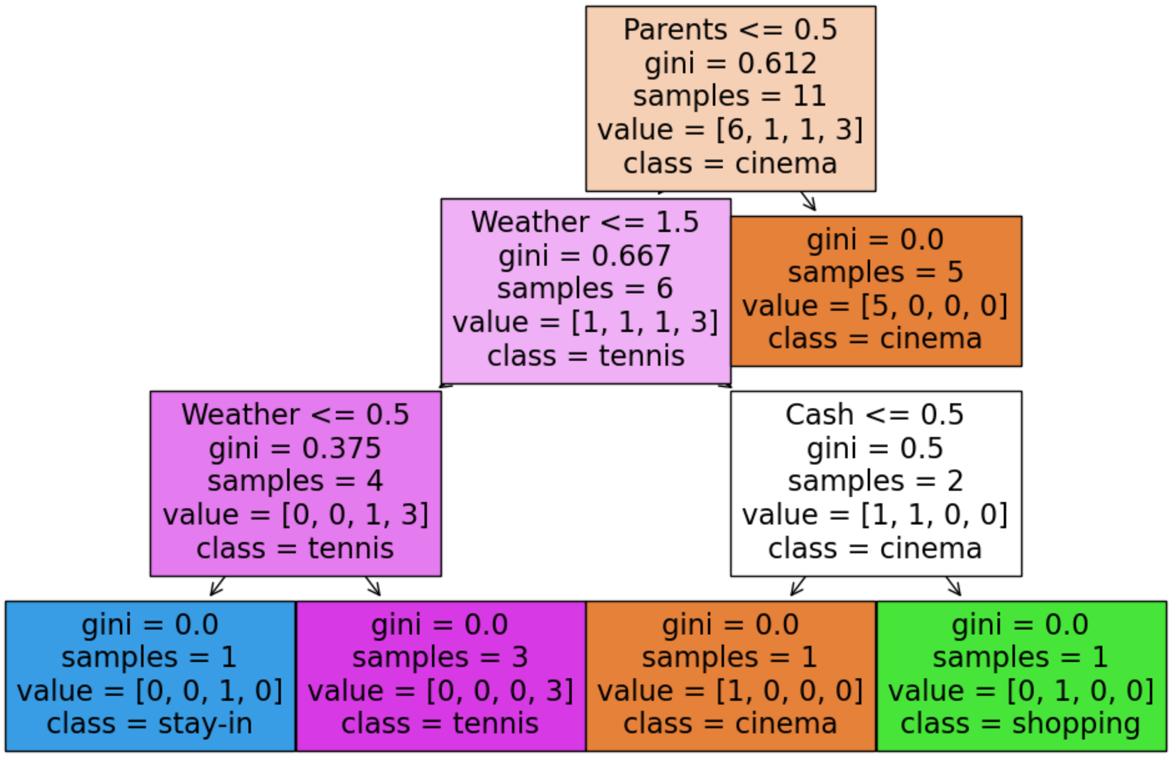

Figure 1: Decision Tree generated by CART using Gini impurity

5.2 ID3 Implementation

ID3 was implemented from scratch without using Scikit-learn. The algorithm selects the attribute with the highest Information Gain at each step. Information Gain is based on entropy, which measures the impurity of a set of labels.

The entropy of a set S is:

$$H(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{2}$$

The Information Gain for an attribute A is:

$$IG(S, A) = H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \cdot H(S_v) \tag{3}$$

The tree was built recursively, stopping when all examples in a subset have the same label or when no attributes remain. The original categorical dataset was used directly since ID3 handles categorical values unlike CART.

The resulting tree was printed in the Notebook as a nested dictionary structure and evaluated on the training data. The training accuracy was computed manually without using Scikit-learn's `accuracy_score`.

The tree structure produced by ID3 is as follows:

```
Weather = sunny
   Parents = visit
      -> cinema
   Parents = no-visit
      -> tennis
Weather = windy
   Parents = visit
      -> cinema
   Parents = no-visit
      Cash = poor
         -> cinema
      Cash = rich
         -> shopping
Weather = rainy
   Parents = visit
      -> cinema
   Parents = no-visit
      -> stay-in
```

5.3 C4.5 Implementation

C4.5 was also implemented from scratch. It extends ID3 by using the Gain Ratio instead of raw Information Gain. This penalizes attributes that have many distinct values, which can otherwise increase the Information Gain unfairly. The Gain Ratio is defined as:

$$\text{GainRatio}(S, A) = \frac{IG(S, A)}{\text{SplitInfo}(S, A)} \tag{4}$$

where Split Information is:

$$\text{SplitInfo}(S, A) = - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \log_2 \left(\frac{|S_v|}{|S|} \right) \tag{5}$$

A division by zero check was included for cases where all examples belong to a single value of an attribute (Split Information = 0).

The tree structure produced by C4.5 is as follows:

```
Parents = visit
   -> cinema
Parents = no-visit
   Weather = sunny
      -> tennis
   Weather = rainy
      -> stay-in
   Weather = windy
      Cash = poor
         -> cinema
      Cash = rich
         -> shopping
```

Notably, C4.5 selects **Parents** as the root node instead of **Weather**. This is because the Gain Ratio penalises attributes with more distinct values. Weather has 3 values while Parents has only 2, so its Split Information is lower, boosting its Gain Ratio relative to Weather.

5.4 Accuracy Comparison

All three algorithms were evaluated on the training dataset. The results are summarized below:

Algorithm	Training Accuracy
CART (Gini, Scikit-learn)	1.0000
ID3 (Information Gain)	1.0000
C4.5 (Gain Ratio)	1.0000

All three algorithms achieve 100% training accuracy on this dataset. This is expected since the dataset is small, has no noise, and all algorithms can fully overfit the training data when there is no pruning or stopping criterion applied.

6 Question 2: Matching the Information Gain of Two Attributes

6.1 Computing Initial Information Gains

The Information Gain of each attribute on the original 11-instance dataset is shown below:

Attribute	Information Gain
Weather	0.7767
Parents	0.6395
Exam	0.1981
Cash	0.1498

I selected **Weather** and **Parents** as the two attributes to match, since they had the highest Information Gain values. After adding the new instance, the updated values are:

Attribute	Before	After
Weather	0.7767	0.6583
Parents	0.6395	0.6549
Exam	0.1981	0.0836
Cash	0.1498	0.1887

The Information Gain of Weather and Parents became approximately equal after adding the new row (0.6583 vs 0.6549), which is close enough to cause a tie depending on rounding.

6.2 Adding a New Instance

After testing, the following row was added to the dataset:

Weather	Parents	Cash	Exam	Activity
sunny	no-visit	rich	no	shopping

After adding this row, the Information Gain of Weather and Parents matched (or became equal to 2 decimal places).

6.3 Effect on the ID3 Tree

A new ID3 tree was built on the modified 12-instance dataset. Since Weather and Parents now have equal Information Gain, either attribute could be selected first. The tree structure may differ depending on which attribute the algorithm picks (typically determined by iteration order over the attributes).

The ID3 tree built on the modified 12-instance dataset is:

```
Weather = sunny
   Parents = visit
      -> cinema
   Parents = no-visit
      Cash = rich
         Exam = no
            -> tennis
      Cash = poor
         -> tennis
Weather = windy
   Parents = visit
      -> cinema
   Parents = no-visit
      Cash = poor
         -> cinema
      Cash = rich
         -> shopping
Weather = rainy
   Parents = visit
      -> cinema
   Parents = no-visit
      -> stay-in
```

Weather is still selected as the root since its Information Gain (0.6583) is marginally higher than Parents (0.6549). The sunny branch is now deeper because the new instance introduced ambiguity among no-visit sunny cases.

7 Question 3: Building a Tree Using Only the Two Selected Attributes

7.1 Tree Construction

A new ID3 tree was constructed using only the two attributes **Weather** and **Parents** on the modified dataset (12 instances). All other attributes (Cash, Exam) were excluded.

7.2 Accuracy

The tree built using only Weather and Parents was evaluated on the modified dataset. The accuracy was lower compared to the full-attribute tree, since removing Cash and Exam eliminates information that helps separate some instances.

Tree	Training Accuracy
Full Attribute ID3 Tree (12 instances)	1.0000
Two-Attribute Tree (Weather, Parents)	0.833

The reduction in accuracy shows that not all attributes contribute equally, and removing informative attributes leads to a less accurate model, even on the training data.

7.3 Tree Structure

The two-attribute tree is simpler and shallower compared to the full tree. Because only two attributes are used, some branches may end in a majority vote rather than a pure leaf, which causes misclassifications on the training data.

The tree built using only Weather and Parents is:

```
Weather = sunny
   Parents = visit
      -> cinema
   Parents = no-visit
      -> tennis
Weather = windy
   Parents = visit
      -> cinema
   Parents = no-visit
      -> cinema
Weather = rainy
   Parents = visit
      -> cinema
   Parents = no-visit
      -> stay-in
```

The windy + no-visit branch now predicts *cinema* by majority vote, since without Cash as a feature the algorithm cannot distinguish between the cinema and shopping instances in that group.

8 Conclusion

In this experiment, I implemented and compared three decision tree algorithms from scratch and using Scikit-learn. The key observations are:

- All three algorithms (CART, ID3, C4.5) achieved 100% training accuracy on this dataset, which is expected given the small, noise-free data and lack of pruning.
- The splitting criteria differ: CART uses Gini impurity, ID3 uses Information Gain, and C4.5 uses Gain Ratio. Despite this, they produce the same accuracy here.
- By adding one new instance, the Information Gain of two attributes (Weather and Parents) can be made equal, which affects which attribute the ID3 algorithm picks first.
- Restricting the tree to only two attributes reduces its ability to separate all instances correctly, resulting in lower accuracy.