What is a Decision Tree?
A tree-like model that predicts a class (or value) by asking a sequence of questions about the attributes.

- Each internal node → asks about an attribute
- Each branch → represents an outcome of that test
- Each leaf → represents a class label or decision

Hunt's Algorithm (How a Decision Tree Learns)
It's a recursive algorithm that keeps splitting the dataset until each subset is as "pure" as possible (meaning it mostly contains one class).

Algorithm Steps:

1. Start with all the data at the top of the tree.
2. Check:
    a. If every record in this group has the same class → stop! That becomes a leaf with that class.
    b. If there are no records left → make it a leaf that predicts the majority class.
3. Otherwise:
    a. Pick the best feature to ask about — the one that separates the classes most clearly.
    b. Split the data into smaller groups based on that feature's values.
    c. Repeat the same process (go back to step 2) for each smaller group.

What Do We Mean by "Best Split"?

When building a decision tree, we want to choose the question (feature split) that best separates the data into purer groups — groups that mostly contain only one class.

So a good split makes each branch have data points that are mostly of a single class (like mostly "Yes" or mostly "No"). A bad split leaves branches that are still very mixed (lots of both "Yes" and "No").

Types of Splits: two main kinds

1. Binary split: at a split, only split to 2 groups
Binary splits are common because they keep the tree simpler.

2. Multi-Way Split
Splits the data into more than two groups at once.

Gini Index

The Gini index measures how impure or mixed a node is.
It tells us how often a randomly chosen sample from that node would be misclassified if we labeled it randomly according to the class distribution.

- Gini = 0 → pure node (only one class)
- Gini = high → mixed node

$$Gini(t) = 1 - \sum_j p(j|t)^2$$

p(j|t) = proportion of samples in node t that belong to class j.

Suppose a node has:

- 8 total samples
- 1 “No” and 7 “Yes”

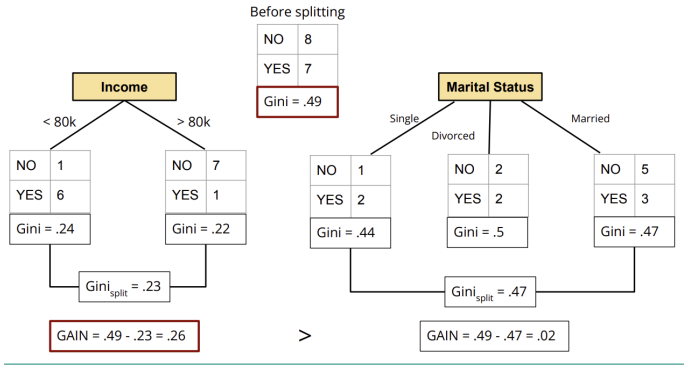
Then p(yes) = ⅞ , p(no) = ⅛
Gini = 1 - (⅞)^2 - (⅛)^2 = 1 - (49/64 + 1/64) = 1 - 50/64 = 14/64 = 0.22

When you split a node into several child nodes, the total impurity is the weighted average of their Gini indices:

$$Gini_{split} = \sum_t \frac{n_t}{n} Gini(t)$$

Goal: pick the attribute/split with the lowest Gini_split (i.e., the highest purity).

Example of deciding what to split on:



Limitations of Decision Trees

1. Overfitting
    - Trees can easily become too complex, splitting until every leaf perfectly fits the training data.
    - This makes the model perform poorly on new data.
    - Example: the tree keeps creating small rules just to fit outliers.

2. Unstable
    - Small changes in the data can lead to a completely different tree structure.

3. Bias toward attributes with many values
    - A feature with many unique values (like “Customer ID”) can create many small splits — even if it doesn’t truly help classification.

4. Hard to capture smooth or linear boundaries
    - Trees split data into boxes — they can’t easily represent gradual changes like a line or curve.

Solutions:
- Early termination
    - Stop at some specified depth
    - Stop if size of node is below some threshold
    - Stop if gini does not improv
- Pruning
    - Create fully grown tree then trim

Extensions

1. Entropy: Measures information disorder — used in Information Gain

2. Misclassification error: Simpler measure that looks only at the most common class in a node.
    - It’s less sensitive than Gini or Entropy — so it’s used less often when growing the tree, but sometimes used for pruning (simplifying the tree).
    - Ex: If 70% “Yes”, 30% “No” ….. error = 1 - 0.7 = 0.3