

# Decision Tree ML Algorithms

---

A **Decision Tree** is a **supervised**, non-parametric machine-learning algorithm used for both **classification** and **regression** tasks. It models decisions as a tree-like structure where internal nodes represent features, branches represent decision rules, and leaf nodes represent outcomes (class labels or numeric values)

**OR**

Decision Tree is a supervised ML algorithm that uses a tree-like structure of decisions to perform classification or regression by recursively splitting data based on feature importance

## Types of Decision Trees

→ **Classification Tree**

- ◆ Output: Category (Yes/No, Spam/Not Spam)

→ **Regression Tree**

- ◆ Output: Continuous value (Salary, Price)

## How Decision Tree Works

1. Select the **best feature** to split data
2. Split data into subsets
3. Repeat recursively for each subset
4. Stop when:

All data is pure

- Max depth reached
- Minimum samples reached

## Key components and terminology

- Root node: topmost node; the starting point for all splits.
- Internal (decision) nodes: represent tests on features (e.g., “Income  $\geq$  50K?”).
- Branches: outcomes of tests (e.g., “Yes / No”).
- Leaf (terminal) nodes: final predictions (class label or numeric value)

## Advantages

- Easy to understand & visualize
  - No feature scaling required
  - Handles numerical & categorical data
  - Works well with non-linear data
- 

## Disadvantages

- **Overfitting** (major issue)
  - Sensitive to small data changes
  - Not good for very large datasets
- 

## How to Avoid Overfitting

- Set `max_depth`
- Set `min_samples_split`
- Set `min_samples_leaf`
- Use **Pruning**
- Use **Ensemble methods** (Random Forest, XGBoost)

## What is Gini?

Gini Index measures how impure a node is.

It tells us the probability of misclassification if we randomly label a data point.

### Range

- 0 → Pure node (all data belongs to one class)
- 0.5 (binary) → Highly impure

#### ◆ Formula

$$Gini = 1 - \sum p^2$$

Where  $p$  = probability of each class

## What is Entropy?

Entropy measures the randomness or uncertainty in data.

Higher entropy = more mixed classes

Lower entropy = purer node

### Range (Binary Class)

- 0 → Pure node
- 1 → Maximum impurity

Formula:

$$Entropy = - \sum p \log_2 p$$

## What is Information Gain?

Information Gain tells how much entropy is reduced after a split.

It measures the **effectiveness of a feature**.

### Rule

- Higher Information Gain → Better feature

**Example** If:

- Parent entropy = 1
- Child entropy after split = 0.6

$$IG = 1 - 0.6 = 0.4$$

---

## When to Use What?

- **Gini** → Faster, default in scikit-learn
  - **Entropy** → More theoretical
  - **Information Gain** → Feature selection logic
- 

## Overfitting and pruning

Decision trees can overfit by growing very deep and complex trees that memorize noise.

Common ways to control this:

- **Pre-pruning (early stopping):**
  - Limit maximum depth.
  - Require a minimum number of samples per node or per leaf.
  - Stop if gain from splitting is below a threshold.

- **Post-pruning:**
  - Grow a large tree first, then remove branches that do not improve performance on validation data (e.g., cost-complexity pruning).

Click here for More Details : [Decision Tree](#)