Decision Tree Algorithm

Agenda: Exploring Decision Trees

- AI Overview
- Introduction Basic Concepts
- How Decision Trees Work
- Pruning and Over-fitting
- Advantages and Disadvantages
- When To Use Decision Tree Algorithm
- Real-World Applications
- Summary

Machine Learning (ML)

Machine learning (ML) enables machines to consume data and solve problems.

Machine learning methods fall into three main types: supervised, unsupervised, and reinforcement learning.

Supervised learning uses labeled data to train models, while unsupervised learns patterns from unlabeled data.

Reinforcement learning improves decisions through trial and error with rewards.

Supervised Learning

Models learn from labeled input-output pairs to predict outcomes.

Unsupervised Learning

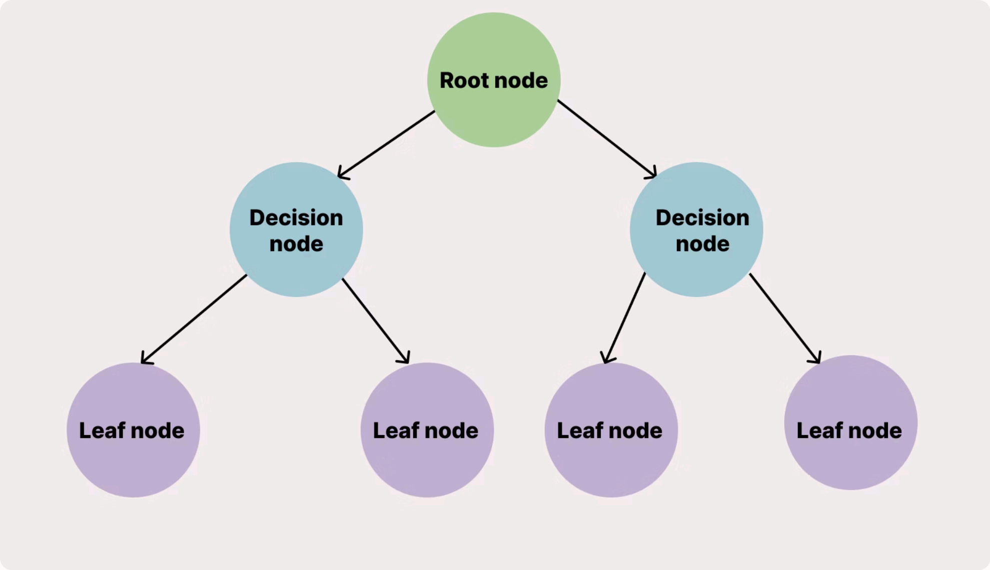
Algorithms identify hidden structures in unlabeled data.

Reinforcement Learning

Agents learn via rewards and penalties to maximize goals.

Introduction to Decision Trees Algorithm

- A Decision Tree is a supervised learning algorithm.
- It represents a function that takes as an input a vector of value and returns returns a "decision" a single output value.
- It is used for classification and regression tasks.
- This model visually represents decisions step-by-step.
- It reach to a decision by performing a sequence of steps.



Basic Concepts

- **Node :** Each point where a decision is made.
- **Root Node :** The first decision node, it represent the entire population , this further will be divided it two or more decision nodes.
- **Internal Nodes :** Represent tests on features
- **Leaf Nodes :** Represent class labels or output values, also known as Terminal node (do not split further)
- **Splitting :** Process of dividing a node into sub-nodes
- **Branch :** A decision path from one node to another

How Decision Trees Work

Entropy

Entropy measures the disorder or uncertainty in the dataset.

- measures error of the model

Formula: **$\text{Entropy}(S) = -\Sigma\ p(i) \log_2 p(i)$**,

where p(i) is the probability of class i in set S.

Information Gain

Information Gain quantifies the reduction in entropy after a split.

- measures how much a particular attribute reduces the entropy (uncertainty) of a dataset.

Formula: **$IG(S, A) = \text{Entropy}(S) - \Sigma\ (|S_v|/|S|)\ \text{Entropy}(S_v)$**,

where S_v are subsets after splitting by attribute A.

- In a decision tree, the algorithm selects the attribute that provides the highest information gain, effectively reducing the uncertainty (entropy) in the dataset. By recursively splitting the data using the attribute that maximizes information gain, the model builds a tree that classifies the data accurately.
- The purity of node increases with respect to the target variables.

Building a Decision Tree: Algorithm Steps

Choose Best Attribute

Use selection measures to find optimal splits.

Create Decision Node

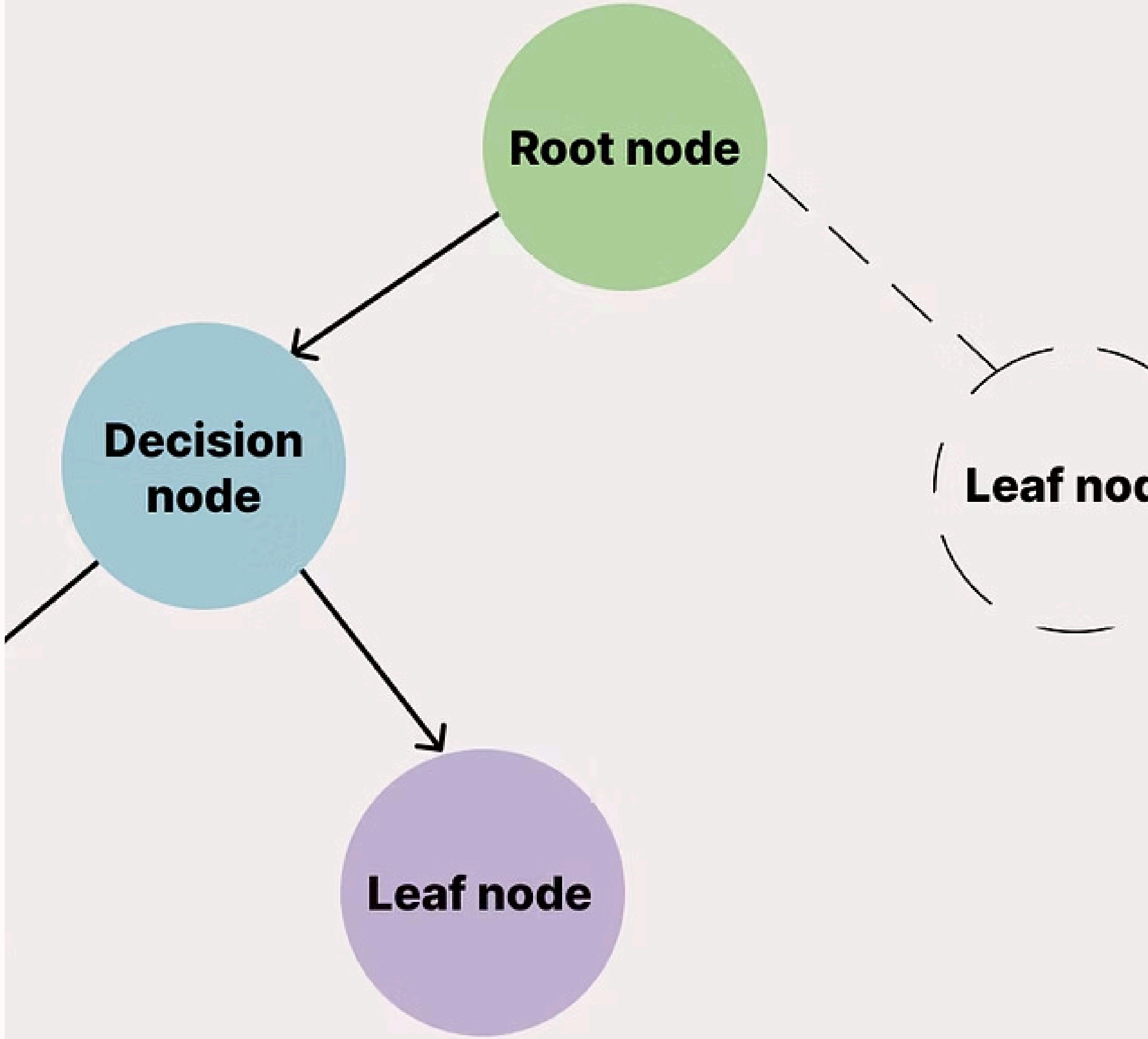
Split data into subsets by chosen attribute.

Repeat Recursively

Continue until stopping conditions are met.

Pruning and Over-fitting

- Pruning is the opposite of splitting
- Pruning removes leaf nodes that add little value.
- This simplifies the tree and improves generalization on new data.
- Over-fitting occurs when a tree fits training data too closely, harming accuracy on unseen data. The model learns not only the underlying patterns in the training data but also the noise and outliers
- Pruning helps prevent overfitting by removing branches that have little predictive power.



Advantages and Disadvantages of Decision Trees

Advantages

- Easy to interpret and understand.
- Handles both categorical and numerical data.
- No assumptions on data distribution.

Disadvantages

- Prone to overfitting with noisy data.
- Unstable to small data changes.
- Training can be computationally expensive.
- pruning is necessary for large set of data

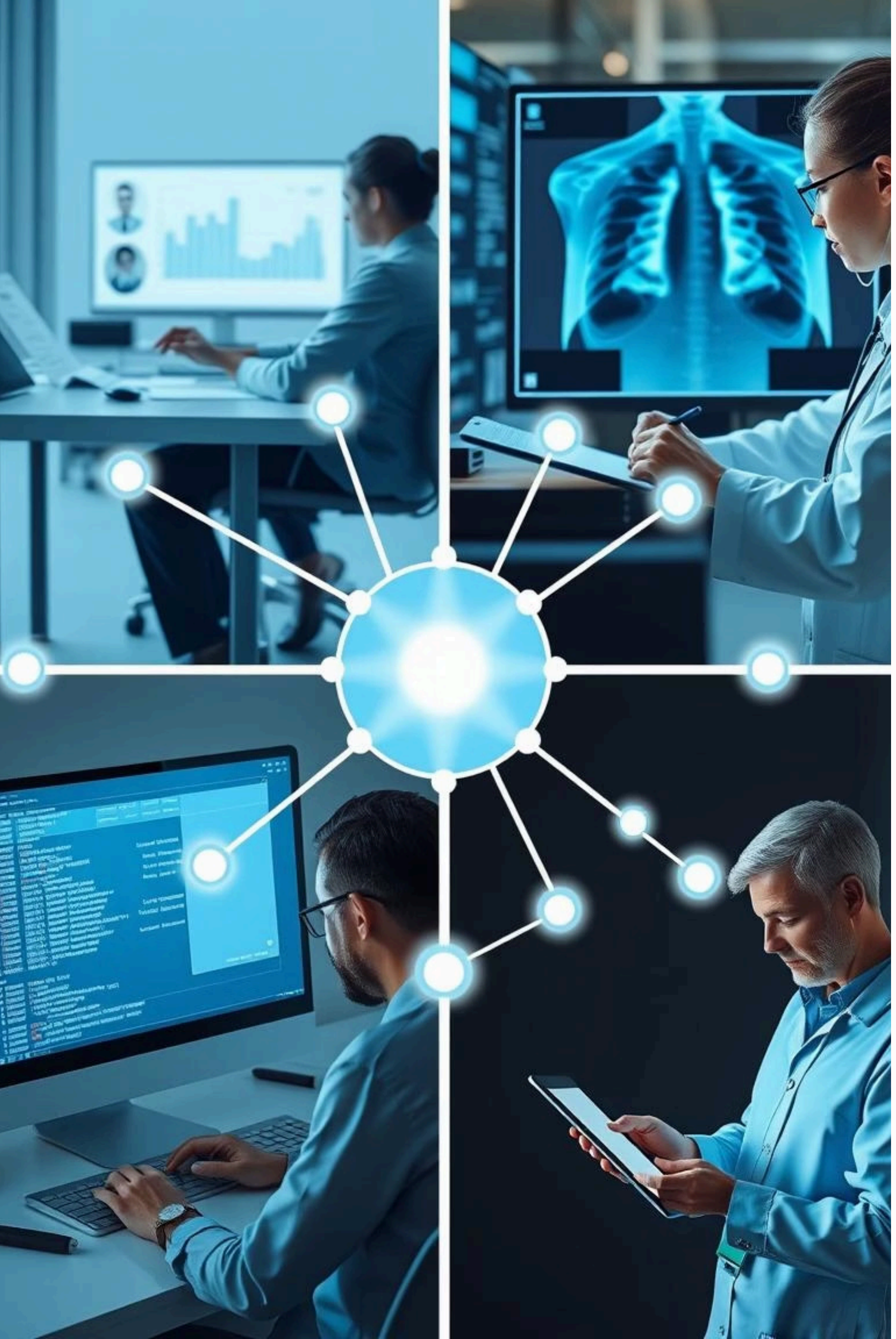
When To Use Decision Tree Algorithm

When interpretability is crucial: Decision trees provide clear, easy-to-understand visual models that show exactly how decisions are made. This transparency makes them ideal when you need to explain your model's reasoning to stakeholders or domain experts. *For example, in healthcare, doctors use decision trees to explain patient risk factors and diagnoses clearly to patients and colleagues.*

As a baseline model: Decision trees are quick to train and can provide a strong baseline for classification or regression tasks. Using them early in your modeling process helps gauge overall feature importance and sets a benchmark for more complex models. *For instance, e-commerce businesses use decision trees early to identify top features influencing customer purchase behavior before deploying advanced models.*

When your data has a mixture of feature types: Decision trees naturally handle both categorical and numerical data without requiring complex preprocessing. They can seamlessly split on different feature types in a single model. *For example, in credit scoring, decision trees use categorical features like job type and numerical features like income to predict creditworthiness.*

When the dataset is small to medium-sized: Decision trees work well with limited data and can capture patterns effectively without extensive training data. However, with very large datasets, training time and overfitting risk may increase, so other techniques might be preferable. *As an example, startups analyzing customer feedback with limited entries use decision trees to extract insights before scaling up their data collection.*



Real-World Applications

Applications

- Credit Risk Assessment
- Medical Diagnosis
- Fraud Detection
- Recommendation Systems

Summary of Decision Trees

Core Concept

Decision trees split data using attributes that maximize information gain.

Key Process

Recursive splitting builds a tree that classifies data accurately.

Challenges

Pruning reduces overfitting by simplifying the tree structure.

Advantages

Easy to interpret, handles various data types, and requires no assumptions on data distribution.

Disadvantages

Prone to overfitting, sensitive to small data changes, and can be computationally expensive.

Applications

Used in credit risk, medical diagnosis, fraud detection, and recommendation systems.

Any Questions?

Thank You