DecisionTree

Overview

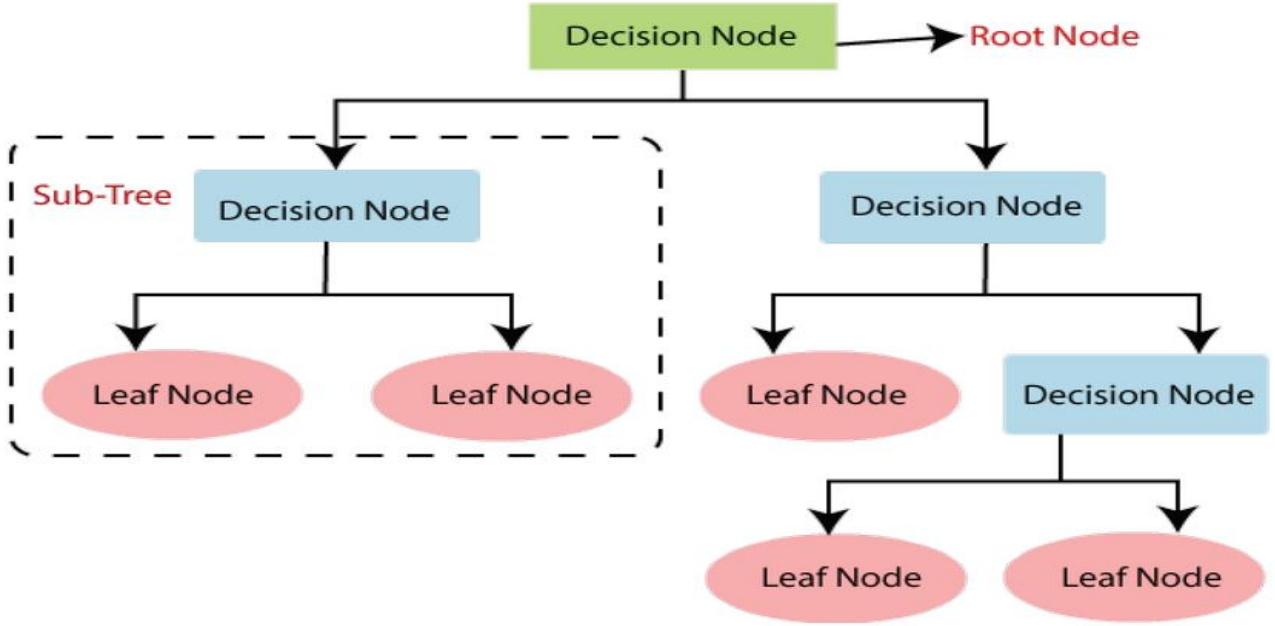
- Tree-Based Models Introduction .
- Decision Trees Basics .
- Applications & Usage
- Algorithm (Process)
- Mathematical & Statistical Intuition
- Hyperparameter Tuning Methods
- Advantages & Disadvantages
- Decision Trees note_book

decision tree :

is a powerful and intuitive machine learning tool used for both classification and regression tasks. It's structured like a tree, where each internal node represents a decision based on a specific feature, each branch corresponds to the outcome of the decision, and the leaf nodes represent the final prediction or output (class or value).

A decision tree uses a tree structure to represent sev
possible decision paths.

- It is like a flow-chart that mapping out the outcome



Tree-based Models:

- Decision Trees
- Ensemble Techniques
  - Random Forest
  - Gradient Boosting

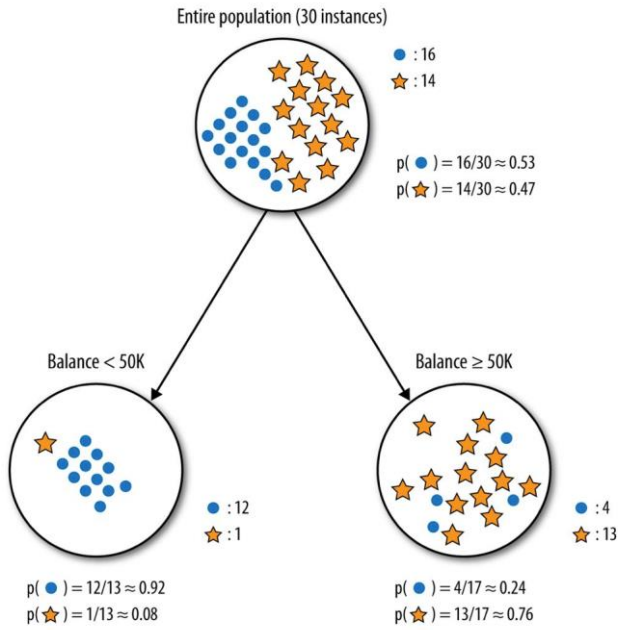
How It Works :

•**Starting at the Root**: The algorithm begins at the top, called the “root node,” representing the entire dataset.

•**Asking the Best Questions:** It looks for the most important feature or question that splits the data into the most distinct groups. This is like asking a question at a fork in the tree.

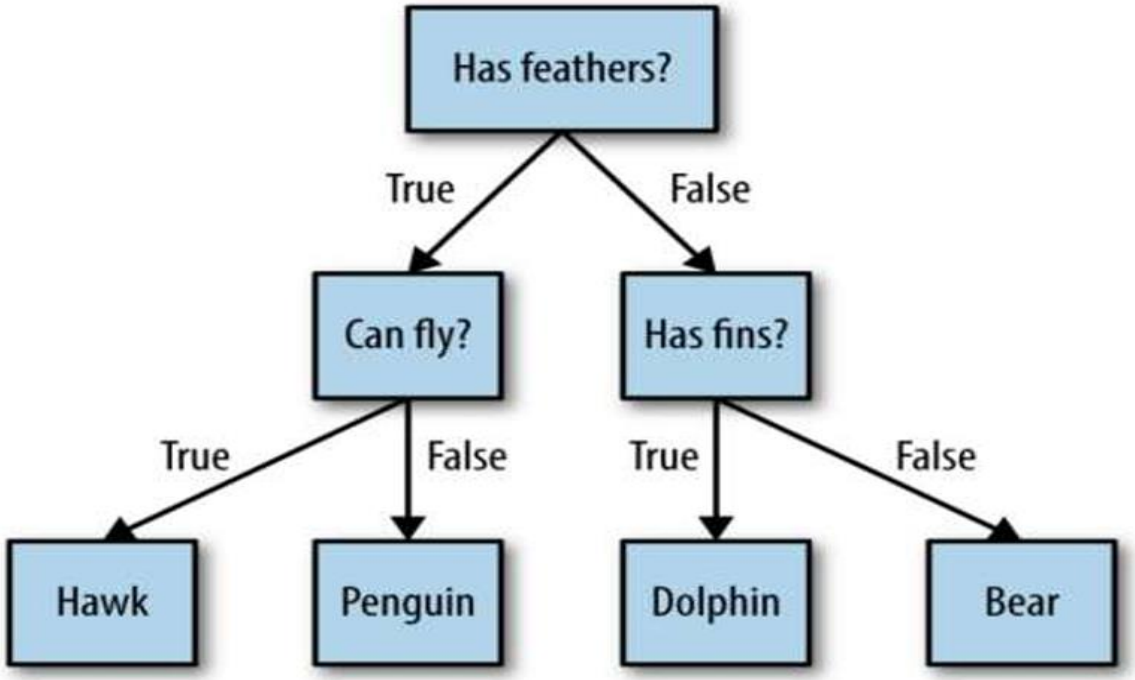
•**Branching Out**: Based on the answer to that question, it divides the data into smaller subsets, creating new branches. Each branch represents a possible route through the tree.

•**Repeating the Process**: The algorithm continues asking questions and splitting the data at each branch until it reaches the final “leaf nodes,” representing the predicted outcomes or classifications.



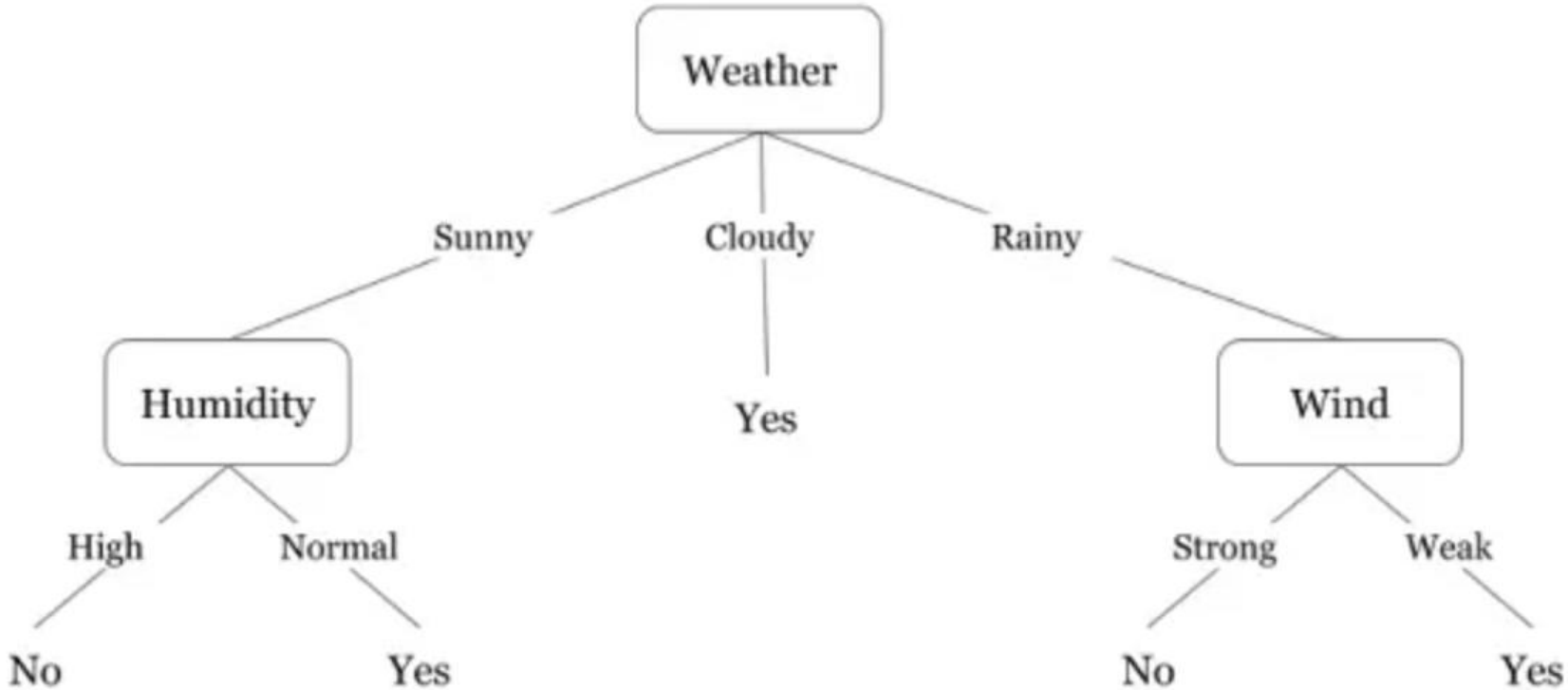
Advantages of Decision Trees

- **Easy to Understand .**
- **Handles Both Numerical and Categorical Data.**
- **No Need for Data Scaling**
- **Automated Feature Selection:**
- **Handles Non-Linear Relationships:**



Disadvantages of Decision Trees

- **Overfitting**
- **Instability**: Small changes in the data can result in a completely different tree being generated.
- **Biased with Imbalanced Datasets**
- **Inability to Extrapolate**: Cannot make predictions beyond the range of the training data.


Weather
Sunny
Cloudy
Rainy
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

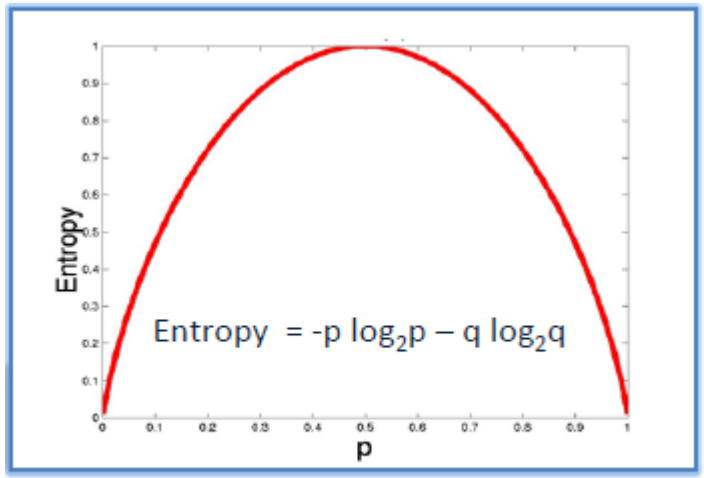
Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

How to Choose the Best Feature for Splitting

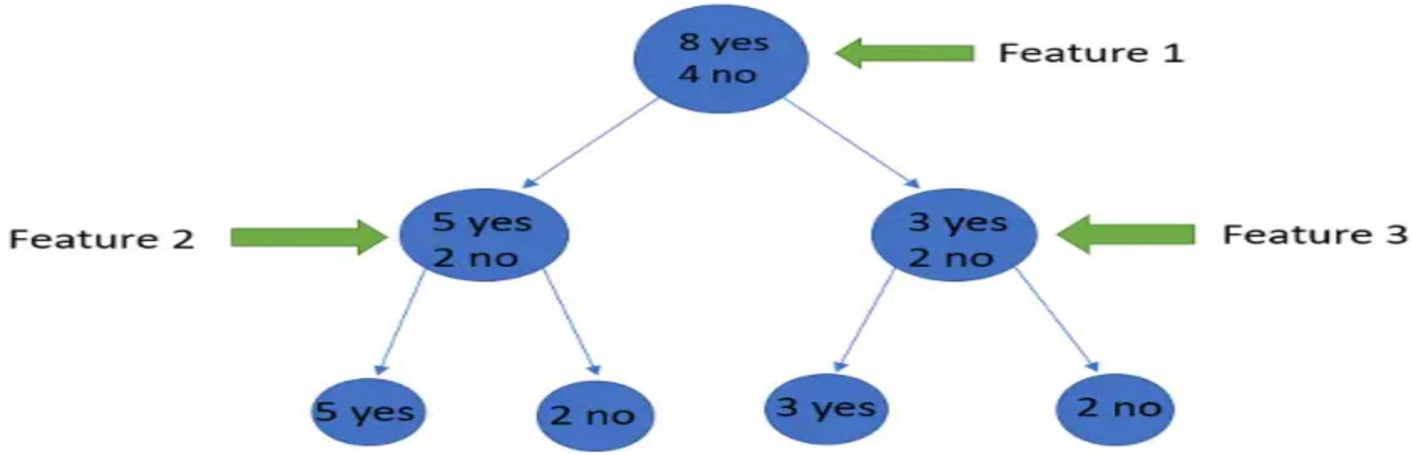
We use metrics like **Entropy**, **Information Gain**, **Gini Index**, and the **Gain Ratio** to measure how good a split is.

Entropy

$$E = -\sum_{i=1}^{n} p_i \, log_2(p_i)$$



Entropy = -0.5 $\log_2$0.5 – 0.5 $\log_2$0.5 = 1



$$\Rightarrow \quad -\left(\frac{5}{7}\right)log_2\left(\frac{5}{7}\right) - \left(\frac{2}{7}\right)log_2\left(\frac{2}{7}\right)$$

$$\Rightarrow \quad -(0.71 * -0.49) - (0.28 * -1.83)$$

$$\Rightarrow \quad -(-0.34) - (-0.51)$$

$$\Rightarrow \quad 0.34 + 0.51$$

$$\Rightarrow \quad 0.85$$

$$\left(\frac{2}{5}\right)^2 \left(\frac{2}{5}\right) \quad \left(\frac{2}{5}\right) log_2\left(\frac{2}{5}\right)$$

$$\Rightarrow \quad -(0.6 * -0.73) - (0.4 * -1.32)$$

$$\Rightarrow \quad -(-0.438) - (-0.528)$$

$$\Rightarrow \quad 0.438 + 0.528$$

$$\Rightarrow \quad 0.966$$

Information Gain

$$Information\ Gain = Entropy_{parent} - Entropy_{children}$$

How to Build a Good Decision Tree

The process of building a decision tree involves making decisions on which feature to split on and when to stop. We aim to build a tree that accurately predicts outcomes without overcomplicating things.

At each level, the algorithm evaluates all features and picks the one that results in the greatest improvement in accuracy. This is called the **greedy approach**, where the tree aims to reduce error (or increase accuracy) step by step.
However, it's important to avoid **overfitting**, where the tree becomes too complex and fits the training data perfectly but performs poorly on new data. A good decision tree balances complexity and accuracy by setting limits like:

- **Maximum depth** (how deep the tree can go).

- **Minimum samples per node** (how many data points should be at each node).

The CART algorithm follows these main steps:

Feature selection:
CART evaluates all features and selects the one that best splits the data into homogeneous subsets. For classification, it uses measures like Gini impurity or entropy. For regression, it uses measures like mean squared error.

Splitting the data:
Once the best feature is selected, CART creates a binary split in the data based on that feature. This process is repeated recursively on each subset until a stopping criterion is met (e.g., maximum tree depth, minimum samples per leaf).

Tree building:
As the algorithm progresses, it builds a tree structure where each node represents a decision based on a feature, and each leaf represents a final prediction

Pruning (optional):
After building the full tree, CART may prune it back to prevent overfitting and improve generalization.

Avoiding Overfitting and Optimizing Trees

Pruning is a technique used in decision trees to simplify the model and improve its accuracy, especially on unseen data.

1. Pre-Pruning (Stopping Early)

Maximum depth (k)
Limit the depth of the tree to prevent it from becoming too complex.

Minimum samples per node
Stop splitting if a node has too few samples.

2. Post-Pruning (Simplifying the Tree)

involves growing the tree fully and then trimming back branches that don't add significant value. The goal is to balance the model's complexity with its ability to generalize.

Applications of Decision Trees

- **Healthcare**

Diagnosing diseases based on patient symptoms

- **Finance**

Assessing credit risk for loan approvals

Detecting fraudulent transactions

Recommending products based on customer preferences

- **Marketing**

Segmenting customers for targeted campaigns

Predicting customer churn and retention

- **Education**

Predicting student performance and outcomes

References

Data Science from Scratch, Joel Grus

Introduction to Machine Learning with Python, Andreas C. Muller • Ch2 (Pg. 70 – 92)

Python Data Science Handbook, Jake VanderPlas