Decision Tree

The Decision Tree algorithm is a popular machine learning algorithm used for both classification and regression tasks. It is a tree-like model that makes decisions based on feature values, where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents the outcome or prediction.

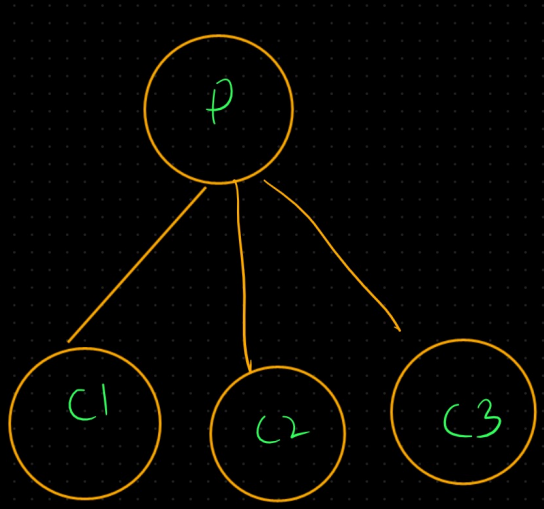
Types of Decision Trees:

① Classification Trees
(Decision Tree classifier)

Eg → Predicting whether an email is spam or not

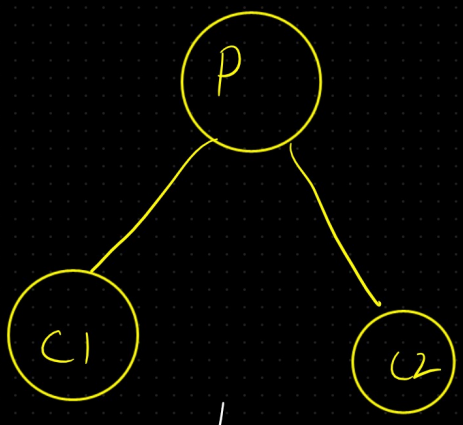
② Regression Trees [Decision Tree Regressor]

Eg → Predicting the price of the house based on its features.



↓

[Multi-Tree]



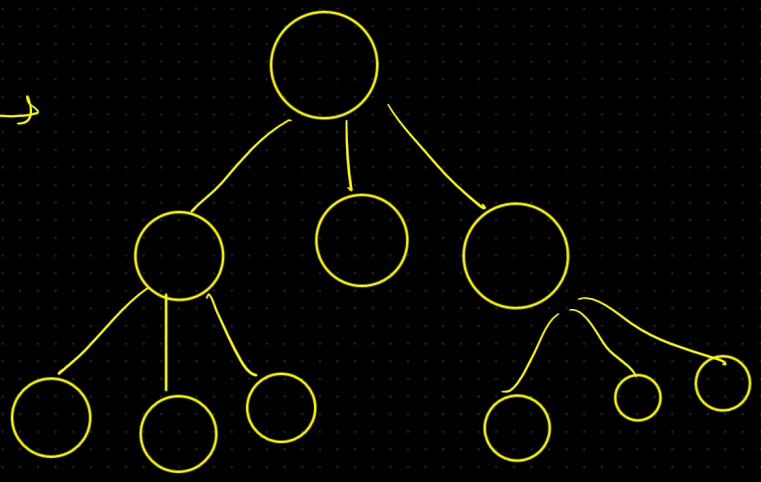
↓

Binary Tree
(1 Node → 2 child)

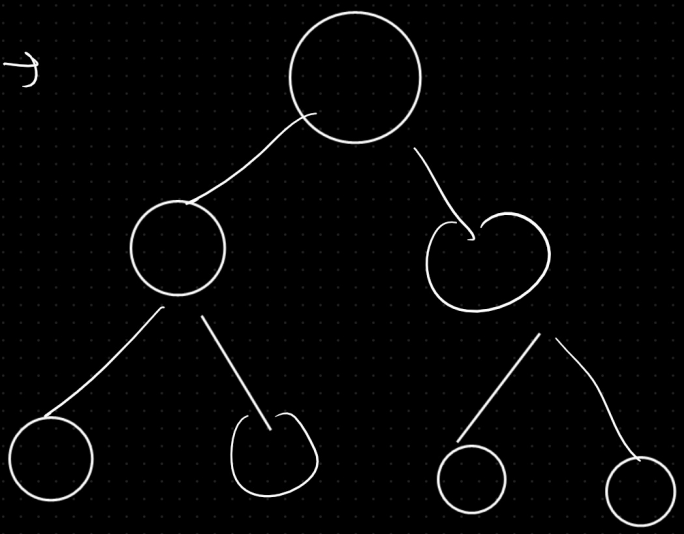
Decision Tree Classifier

1. ID3 → [Iterative Dichotomiser 3]
2. CART → [Classification And Regression Trees]
3. Random Forest
4. Gradient Boosted Trees (GBT)
5. XGBoost (Xtreme Gradient Boosting)

ID3 →

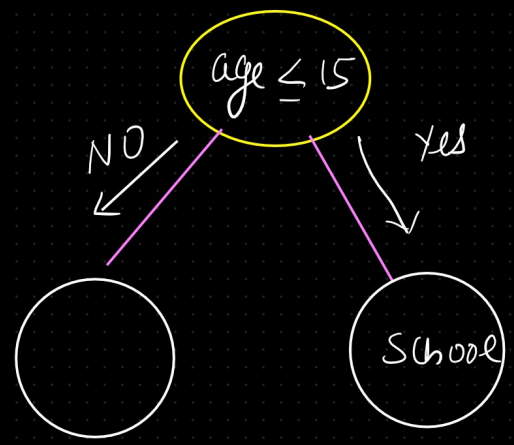


CART →

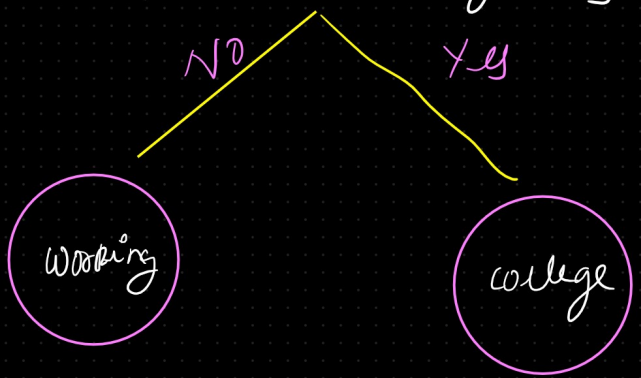


Age = 14 years

```
if (age ≤ 15)
    print("School")
elif (age > 15 and age ≤ 21)
    print("college")
else:
    print("Working")
```

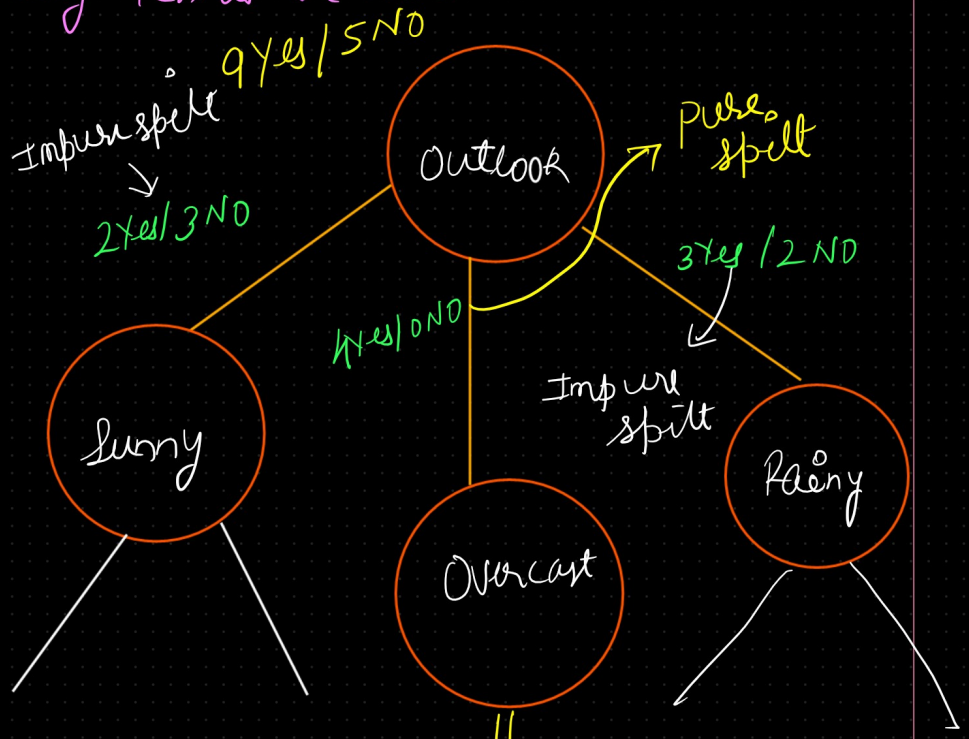


[age > 15 and age ≤ 21]



Dataset: Predict Play Tennis or Not

Outlook	Temperature	Humidity	Wind	Play Tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rainy	Mild	High	Weak	Yes
Rainy	Cool	Normal	Weak	Yes
Rainy	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rainy	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rainy	Mild	High	Strong	No



① Purity Check

Pure Impure

Leaf Node

Entropy
Gini Impurity
} Measure of purity

Decision Trees split nodes based on the concept of purity. A node is considered pure if all its data belongs to the same class (for classification) or has similar values (for regression).

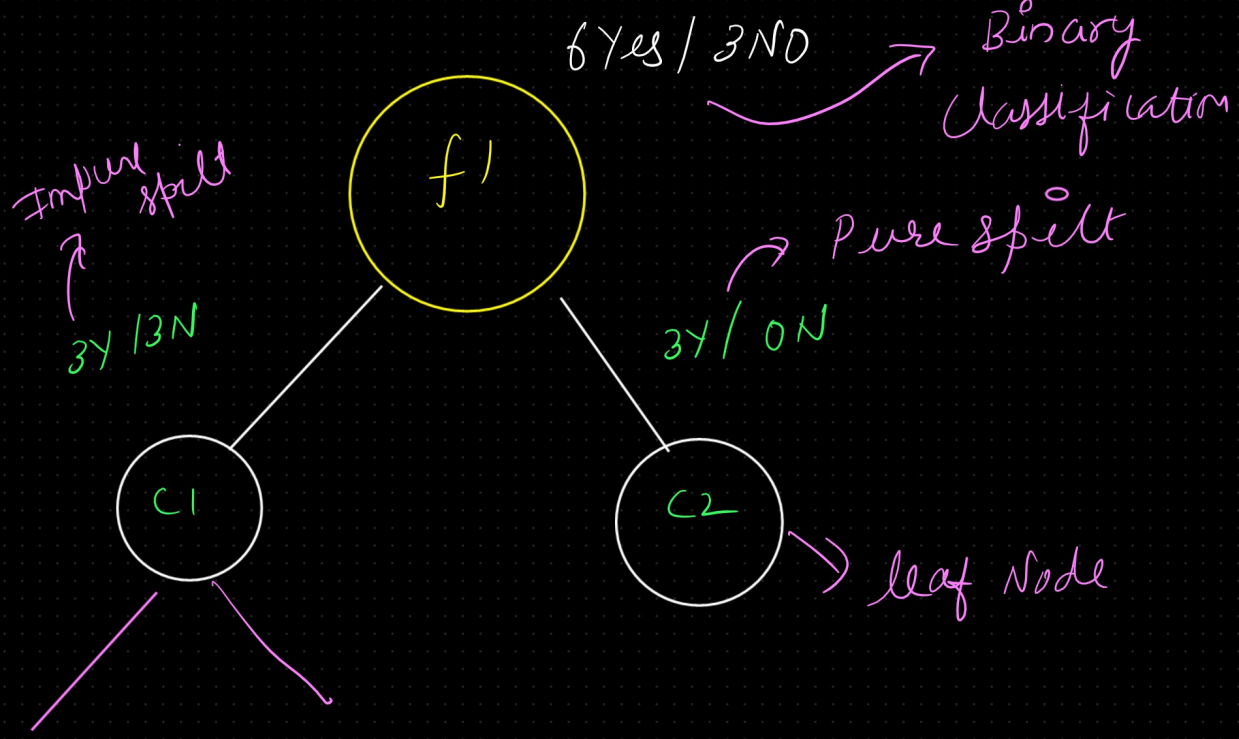
Entropy Vs Gini Impurity

→ Entropy : Measures the Impurity or uncertainty in a dataset

$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

P_+ = Prob of +ve category

P_- = Prob of -ve category



$$H(C1) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

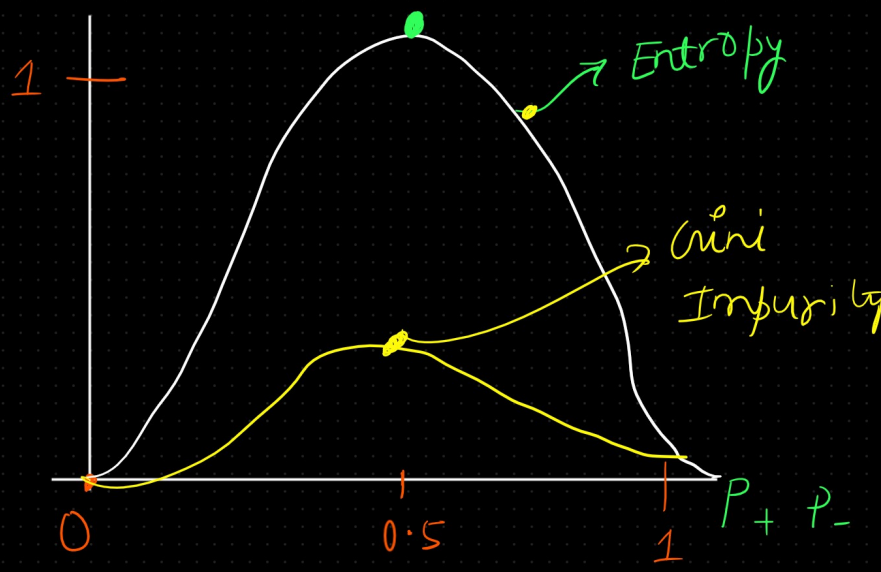
$$= -\frac{3}{6} \log_2\left(\frac{3}{6}\right) - \left(\frac{3}{6}\right) \log_2\left(\frac{3}{6}\right)$$

$$= -\frac{1}{2} \log_2\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) \log_2\left(\frac{1}{2}\right)$$

$$= \underline{1} \Rightarrow \text{Impure Split}$$

$$H(C2) = -\left(\frac{3}{3}\right) \log_2\left(\frac{3}{3}\right) - 0/3 \log_2(0/3)$$

$$= -\log_2 = 0 \Rightarrow \underline{\text{Pure Split}}$$

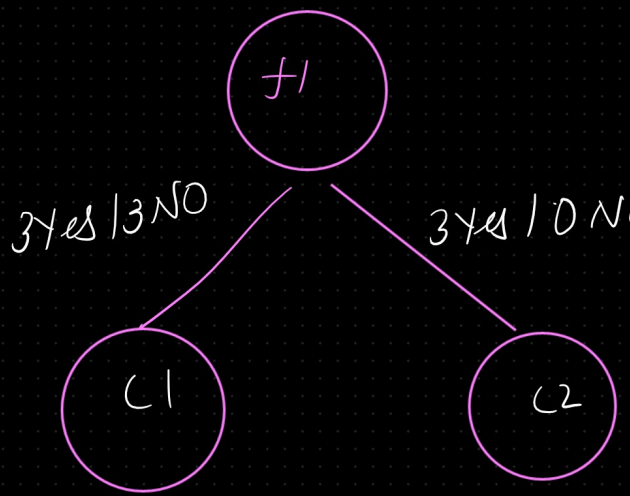


Multi class → $-P_{c_1} \log P_{c_1} - P_{c_2} \log P_{c_2} - P_{c_3} \log P_{c_3}$

② Gini Impurity

$$G.I = 1 - \sum_{i=1}^{n} (P)^2$$

$$G.I = 1 - \sum_{i=1}^{n} (P)^2$$



$$C1 = 1 - [(P_+)^2 + (P_-)^2]$$

$$= 1 - \left[\left(\frac{3}{6}\right)^2 + \left(\frac{3}{6}\right)^2\right]$$

$$= 1 - \left[\frac{1}{4} + \frac{1}{4}\right]$$

$$= \frac{1}{2} = 0.5 \Rightarrow \text{Impure split}$$

$$G.I(C2) = 1 - [(1)^2 + 0$$

$$= 1 - 1$$

$$= 0$$

$\Downarrow$

pure split

Information Gain

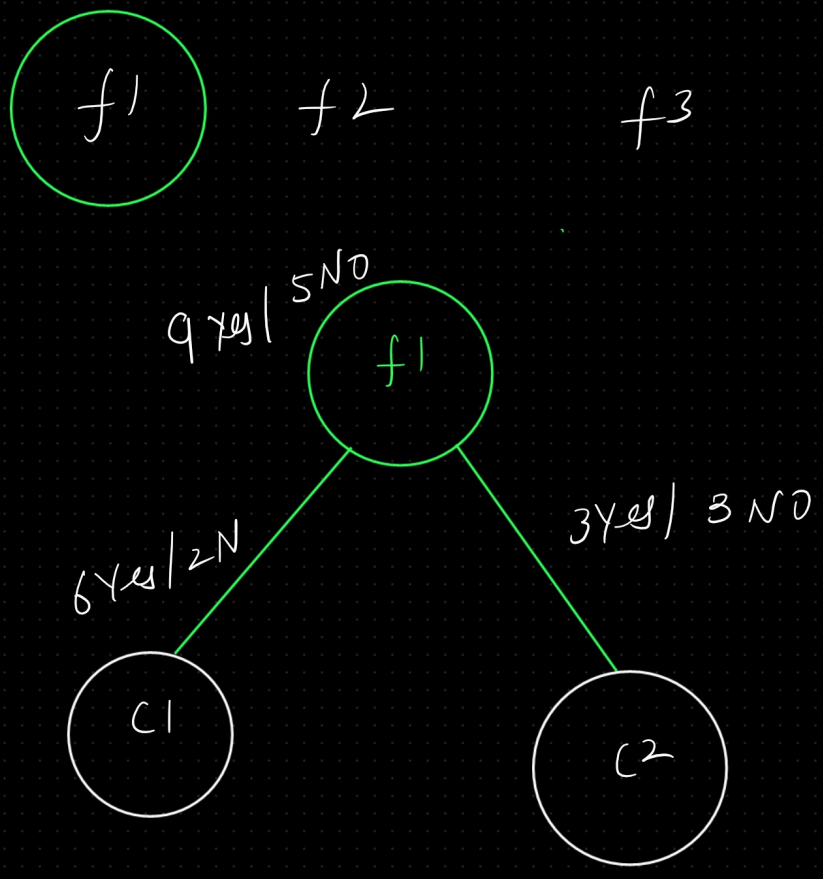
Information gain measures the effectiveness of a particular attribute in classifying the training data. It is calculated by comparing the entropy (or Gini impurity) before and after splitting the dataset based on that attribute.

$$Gain(S, f_1) = H(S) - \sum_{v \in val} \frac{|S_v|}{|S|} H(S_v)$$

where $H(S)$ = Entropy

$f1 \rightarrow$ feature

f1, f2, f3



$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

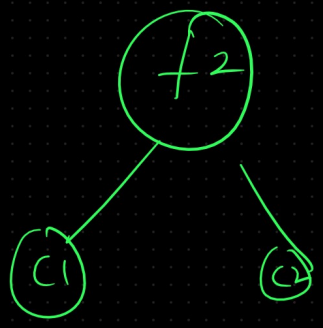
$$= -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} \simeq 0.94$$

$H(C1) = \approx 0.81$

$H(C2) = 1$

$$Gain(S, f1) = 0.94 - \left[\frac{8}{14} \times 0.81 + \frac{6}{14} \times 1\right]$$

$$\boxed{Gain(S, f1) = 0.049}$$



$$Gain(S, f2) = 0.051$$

Information gain is more when we split using f2

Entropy → Dataset → small

Gini Impurity → Dataset → large

When to Use Decision Trees:

a. Pros:

Easy to understand and interpret.

Can handle both numerical and categorical data.

Nonlinear relationships between features and target variable are well captured.

b. Cons:

Prone to overfitting, especially with deep trees.

Can be sensitive to small variations in the data.

Biased towards features with more levels.

Mathematical Intuition:

Splitting Decision: At each node, Decision Trees select the feature that provides the best split, maximizing information gain or minimizing impurity.

Entropy/Gini Calculation: These metrics quantify the uncertainty or impurity in the dataset. Lower values indicate higher purity.

Information Gain Calculation: Measures the reduction in entropy (or decrease in impurity) after splitting the dataset based on a particular attribute.

Recursive Process: The decision-making process continues recursively until a stopping criterion is met, such as reaching a maximum depth or purity threshold.