# Decision Tree [Advance & complex]

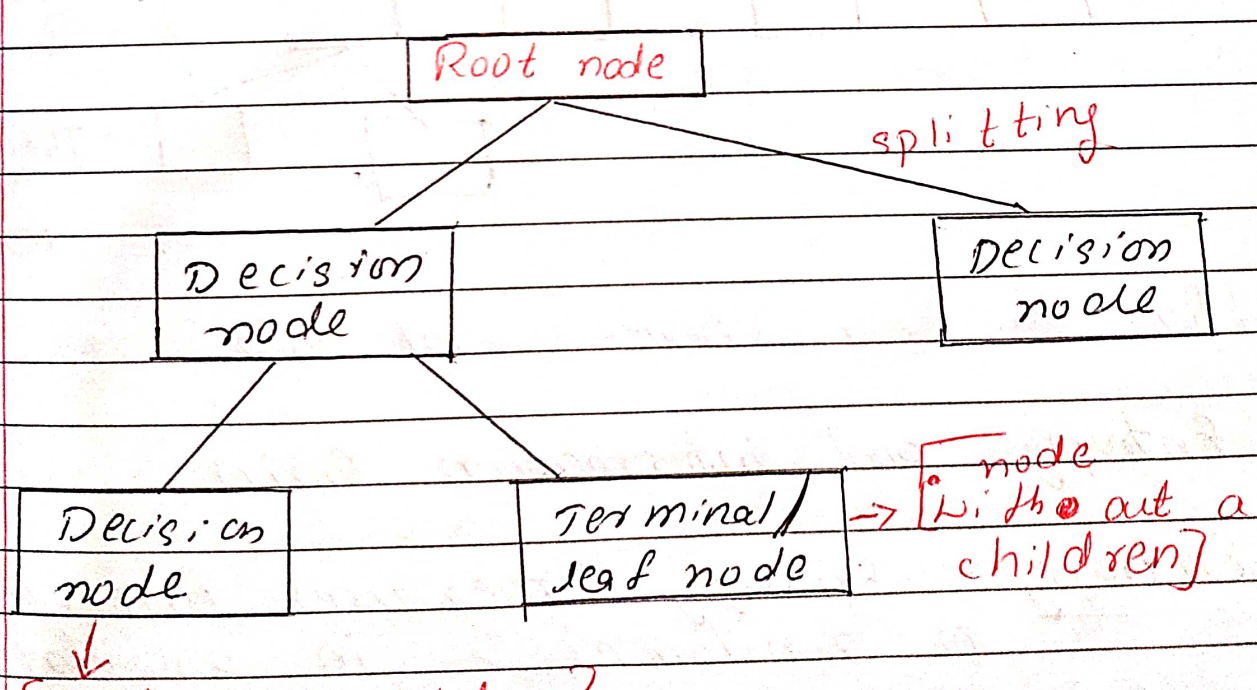
- It comes under supervised learning
- It is used to solve both Regression & classification problem

(1) Decision Tree Regressor.
It is used to solve data which has continuous target.

(2) Decision Tree classifier.
It is used to solve data which has categorical target.

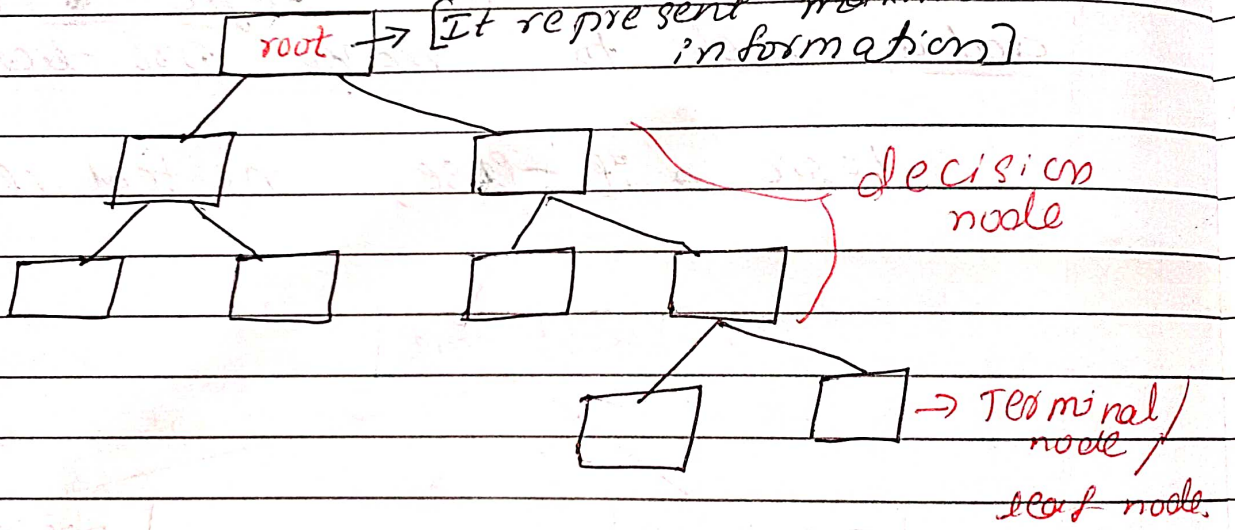
- Decision Tree uses Logic to understand data and to make prediction

Logic → if-else, nested if ...



## How decision Tree work?

- Decision Tree builds Regression and classification models in the form of tree.
- It divides entire data into subsets such that an incremental tree will be developed.
- The final result will be the tree which includes decision node and leaf nodes



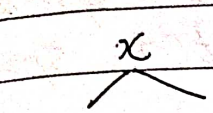
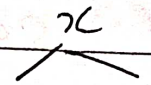
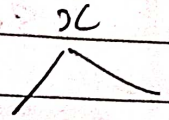

→ How do you choose root node?

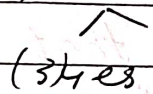
- Entropy and information Gain.

① Entropy :- Entropy measures uncertainty or randomness in a data.

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

$P_+$ → probability of positive class
$P_-$ → Probability of negative class

| Purity | Impurity | Impurity |
|---|---|---|
|   |   |   |
| Entropy = 0 | Entropy will be high. | Entropy = 1 |

- Range of Entropy = [0, 1]
- Entropy = 0, → pure split →  
- Entropy > 0 and <= 1, impurity.

Note:-
Entropy is calculated on categories of a column.

## (2) Information Gain:-

- Information Gain tells about how much information a feature is giving.
- Gain is used to select the root node among input variables.
- Root node will be the feature which gives maximum information Gain.

$$Gain(S) = H(S) - \sum \frac{|S_v|}{|S|} \cdot H(S_v)$$

$H(S)$ → Entropy before split
$H(S_v)$ → Entropy after split
$S$ → no of samples before split.
$S_v$ → no of samples after split

→ Steps to calculate information gain

1. Find entropy of data $H(S)$
2. select column
3. Find Entropy of categories in that column
4. Calculate the information Gain of that column
5. Repeat above 3 steps for different columns
6. select the column which gives maximum information Gain.

Eg: Weather → Sunny, rainy, cloudy

(H) H(sunny) H(rainy) H(cloudy)
Entropy

Gain (weather)

| Weather | Humidity | Wind | play |
|---|---|---|---|
| sunny | high | weak | no |
| sunny | high | strong | no |
| cloudy | high | weak | yes |
| rainy | high | weak | yes |
| rainy | normal | weak | yes |
| rainy | normal | strong | no |
| cloudy | normal | strong | yes |
| sunny | high | weak | no |
| sunny | normal | weak | yes |
| rainy | normal | weak | yes |
| sunny | normal | strong | yes |
| cloudy | high | strong | yes |
| cloudy | normal | weak | yes |
| rainy | high | strong | no |

Eg 1:- To select root for above table.

① find Entropy of data H(S)

→ Data

9 yes 5 NO

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

$H(S) = -P(yes) \cdot \log_2 P(yes) - P(NO) \cdot \log_2 P(NO)$

$P(yes) = 9/14 \qquad P(NO) = 5/14$

$H(S) = (9/14) \log_2 (9/14) - (5/14) \cdot \log_2 (5/14)$

$H(S) = 0.94$

② find entropy of weather & gain of weather

→ Sunny — 2 yes 3 NO

Cloudy — 4 yes 0 NO

Rainy — 3 yes 2 NO

- $H(S) = -P(yes) \log_2 P(yes) - P(NO) \cdot \log_2 P(NO)$

→ Calculate H(Sunny) :-

$P(yes) = 2/5 \qquad P(NO) = 3/5$

$H(Sunny) = -2/5 \log_2 (2/5) - (3/5) \log_2 (3/5)$

$H(Sunny) = 0.97$ //

→ Calculate Entropy of cloudy (H(cloudy)) :

P(yes) = 4/4 P(No) = 0

H(cloudy) = $-(4/4)\log_2(4/4) - (0)\log_2(0)$

H(cloudy) = 0 //

→ Calculate H(Rainy) :-

P(yes) = 3/5 P(No) = 2/5

H(Rainy) = $-(3/5)\log_2(3/5) - (2/5)\log_2(2/5)$

H(Rainy) = 0.97

→ H(S) = 0.94 , H(sunny) = 0.97

H(cloudy) = 0, H(Rainy) = 0.97

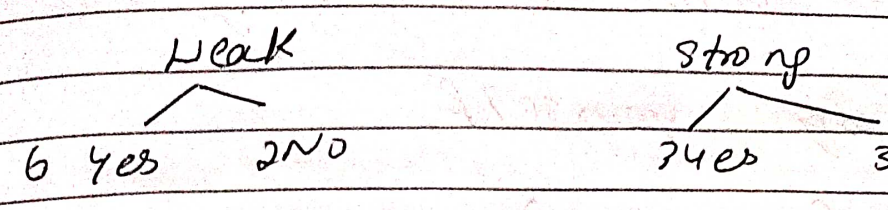
→ Gain(weather) = $H(S) - \sum \frac{|S_v|}{|S|} . H(S_v)$

Gain = $H(S) - (S_v/S)\,H(\text{sunny}) - (S_v/S)\,H(\text{cloudy}) - (S_v/S)\,H(\text{rainy})$

Gain = $0.94 - (5/14)(0.94) - (4/14)(0) - (5/14)(0.97)$

Gain = 0.24

(3) Find Entropy of Wind and Gain of Wind




→ Calculate H(Weak) :-

$P(Yes) = 6/8$ , $P(NO) = 2/8$

$H(Weak) = -6/8 \log_2 6/8 - 2/8 \log_2 2/8$

$H(Weak) = 0.81$

~~→ H(Strong) = $-1/2 \log_2 1/2 = 1/2 \log_2 1/2$.~~

→ Calculate H(Strong) :-

$P(Yes) = 1/2$ $P(NO) = 1/2 = 3/6 = 1/2$.

$H(Strong) = -1/2 \log_2 1/2 - 1/2 \log_2 1/2$

$H(Strong) = 1$ //

→ $Gain = H(S) - \left(\frac{S_v}{S}\right) H(Weak) - \left(\frac{S_v}{S}\right) H(Strong)$

$Gain = 0.94 - (8/14)(0.81) - (6/14)(1)$

$Gain = 0.048$ //

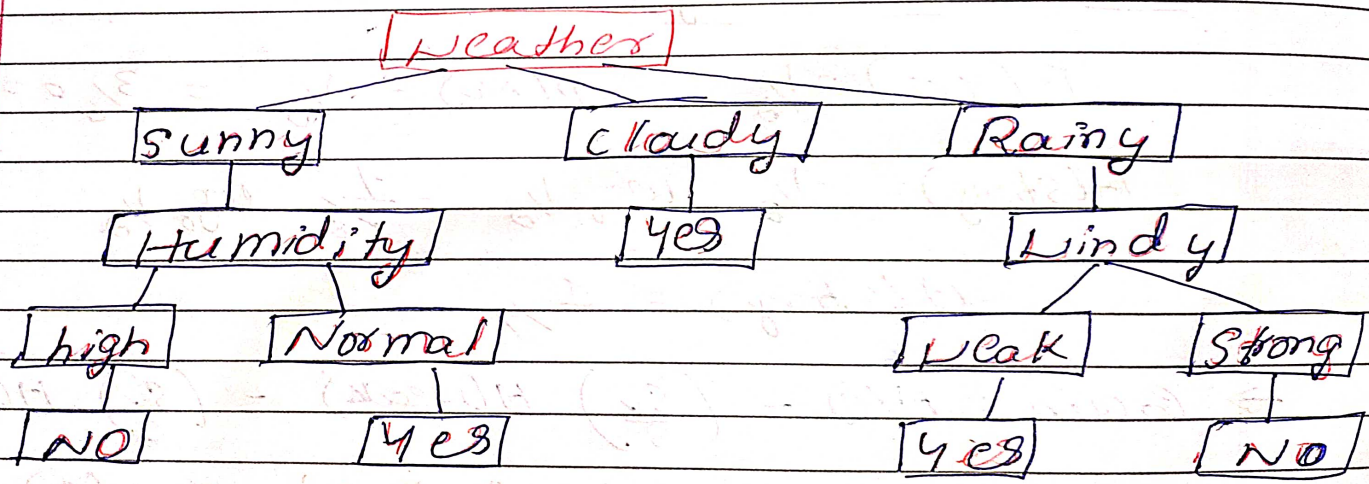
(4) Calculate Entropy of Humidity and Gain of Humidity.

→ solve by yourself.

→ Gain (Humidity) = 0.12

(5) Gain (Weather) = 0.24
Gain (Wind) = 0.04
Gain (Humidity) = 0.12

Weather is selected as root node because it is giving maximum information gain.




Question:-

(1) If Cloudy, Hig
Question?

Cloudy, high, Weak → Yes play
Sunny, high, Weak → NO
Rain, Normal, Weak → Yes.

Imp
Note:- Scaling and handling outliers are not required in tree based algorithms.

→ Limitation of Gain & Entropy:-

- Time taken to compute is very high because of log function.

→ How to overcome the limitation of Entropy and Information gain

1) Gini Index
2) Weighted Gini

1) Gini:-

- It measures impurity of the data
- It is used to select root node.
- Select the column which has less weighted Gini.

- $Gini = 1 - \sum P_i^2 = 1 - [P(yes)]^2 - [P(no)]^2$

2) Weighted Gini:-

$$\text{Weighted Gini} = \frac{\sum |S_v|}{|S|} Gini(S_v)$$

$S \rightarrow$ no of samples before split

$S_v \rightarrow$ no of samples after split

Note: Select the column which gives less weighted gini as root node.

① How to calculate weighted gini for above table

Ans:

→ ① for weather column

| Sunny | Cloudy | Rainy |
|---|---|---|
| 2 Yes 3 NO | 4 Yes 0 NO | 3 Yes 2 NO |
| P(Yes) = 2/5 P(NO) = 3/5 | P(Yes) = 4/4 = 1 P(NO) = 0 | P(Yes) = 3/5 P(NO) = 2/5 |

→ • $Gini(sunny) = 1 - [P(yes)]^2 - [P(NO)]^2$

$= 1 - (2/5)^2 - (3/5)^2$

$Gini(sunny) = 0.48$

• $Gini(cloudy) = 1 - (1)^2 - (0)^2$

$= 0$ //

• $Gini(rainy) = 1 - (3/5)^2 - (2/5)^2$

$= 0.48$

→ Weighted Gini: $\frac{|Sv|}{|S|} Gini(sunny) + \frac{|Sv|}{|S|} Gini(cloudy) + \frac{|Sv|}{|S|} Gini(rainy)$

$= \left(\frac{5}{14}\right) \cdot (0.48) + \text{ot} \left(\frac{5}{14}\right) (0.48)$

$= 0.34$

Similarly calculate for wind & humidity and compare all three result.

- Weighted Gini (weather) = 0.34
- Weighted Gini (Wind) = 0.48
- Weighted Gini (Humidity) = 0.52

→ from above we need to select which has less weighted gini.
→ weather has less weighted gini so it is considered as root node.

Note:- Gini range = [0, 0.5]

→ Converting Categorical column to numerical column

① Label Encoder:-

- When data is odinal
- use ~~data~~ when data has 2 categories

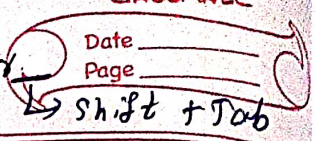
Eg:- Yes, No, Positive, Negative

② one hot Encoder:-

- use when data is nominal [no order]

shift + Tab ⇒ after Import algorith name

Ex:- from sklearn.ensemble import GradientBoostingClassifier




- use this one hot Encoding then you have 2 (or) 3 categories.

(3) manual Encoder:-

- use this when column has more than 3 categorical.
- check the categories relationship with target and assign values according to that.
- It is used to Encode target column.

| Overfitting | underfitting |
|---|---|
| • training score is high testing score is less | • Training score is less, testing score is less |
| • Low bias, high variance | • high bias, high variance. |

⇒ Bias ⇒ Training Error / Error made on Train data.

⇒ Variance ⇒ Testing Error / Error made on Test data.

Note:- In General, if we find a model with low bias & low variance. It is the most suitable model to select.

# Hyper parameter tuning in Decision tree :-

1) Criterion :- It decides which method is used to measure the purity (or) quality of the split.

→ Criterian :- [ Gini , Entropy ]

↓

It select any one.

2) Splitter :-

It is the strategy or a techingue used to select variables to split at node. [Eg :- like root, sub node)

splitter = ( best, random )

3) max depth :-

It tells about depth of the tree (or) levels (or) height of a tree.

4) min sample split :-

- minimum number of samples required to split internal node (sub node)

- An internal node will have further splits (also called children)

5) min sample leaf :-

- minimum no of sample required to split at leaf node.

- A leaf node is node without any children (without any further splits).

Note:- building model

- n-Jobs = 1 :- It will use only one processor.
- n-Jobs = -1 :- It will make use of all the processor.
- scoring = select "accuracy" or "f1 score"
- verbos = To print message
- Cross validation [CV] :- cross validation. ~~database~~ no
  - define how many CV required