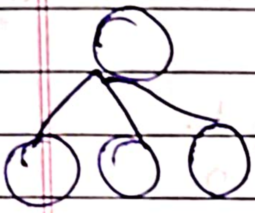
Decision Tree :-

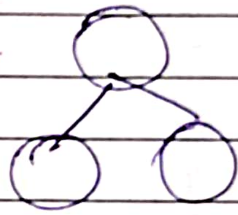
- Hierarchical tree based algorithm that is used to classify or predict outcomes based on a set of rules.

- It have two types :- ID3
  CART ✓ Scikit-learn uses this type of tree



ID3: It may or may not have Binary split

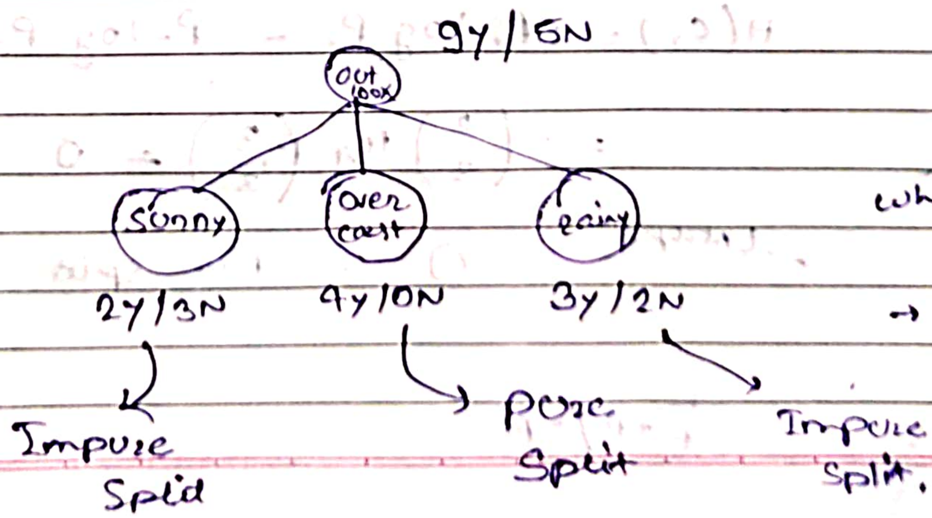
CART: It have Binary split



- Let suppose we have dataset :
  Features like outlook, Temp, Humidity, wind
  play tennis {yes or no}

- Select outlook as root node & count unique categories.
  {sunny, overcast, rainy}

- With respect to outlook lets to figure out how many yes & no's and same for each unique category.



why Impure Split?
→ Both values are present.

How to find mathematically whether split is pure & impure?

1) Entropy :-

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

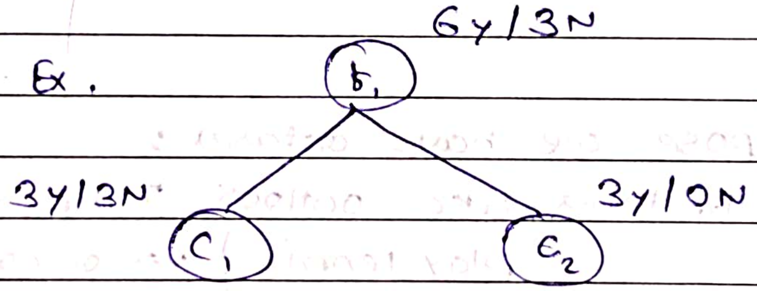
— for Binary classification

Where,

P_+ → Probability of being 1.

P_- → Probability of being 0.

Ex.



$$H(C_1) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

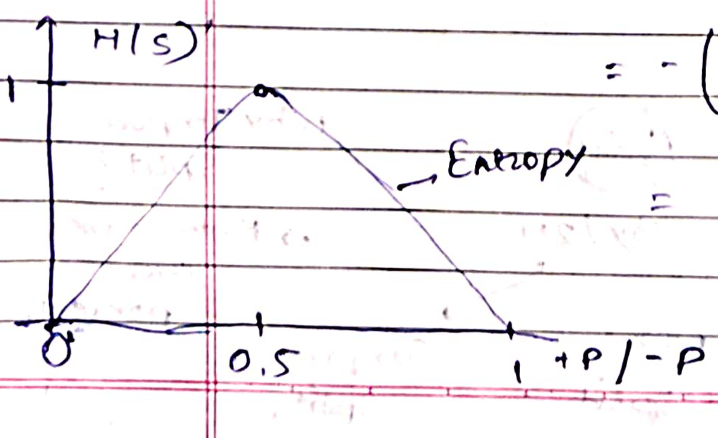
$$= -\frac{3}{6} \times \log_2\left(\frac{3}{6}\right) - \frac{3}{6} \times \log_2\left(\frac{3}{6}\right)$$

$$= 1$$ → Impure split.

$$H(C_2) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

$$= -\left(\frac{3}{3}\right) \log_2\left(\frac{3}{3}\right) = 0.$$

$$= 0$$ → Pure split



When to use Gini Index or Entropy:—

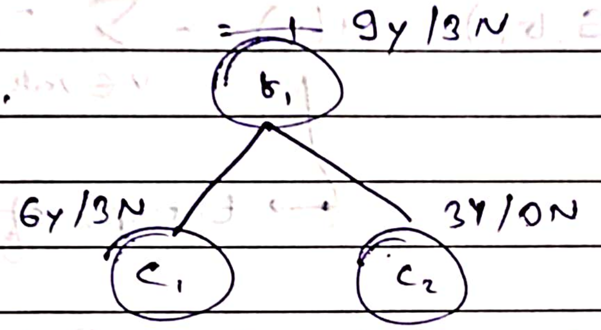
→ wherever the data small use entropy ≈ 10,000 Re

→ If you not choose b/w them, default use gini impurity.

2. Gini Impurity :—

$$G.I = 1 - \sum_{i=1}^{n} p^2$$

ex.



$$G.I(c_1) = 1 - \left(P_+^2 + P_-^2\right)$$

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

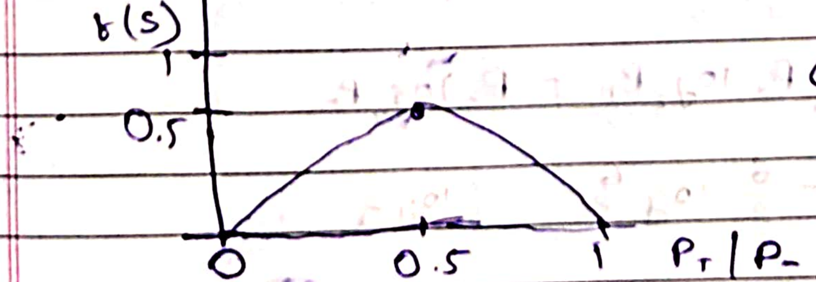
$$= 1 - \frac{1}{2}$$

$$= 0.5 \rightarrow \text{Impure split.}$$

$$G.I(c_2) = 1 - \left(P_+^2 + P_-^2\right)$$

$$= 1 - \left(\left(\frac{3}{3}\right)^2 + 0^2\right)$$

$$= 1 - 1^2 = 0 \rightarrow \text{pure split.}$$



G.I = 0 to 0.5

Which feature would you select for splitting?

⇒ Information Gain.

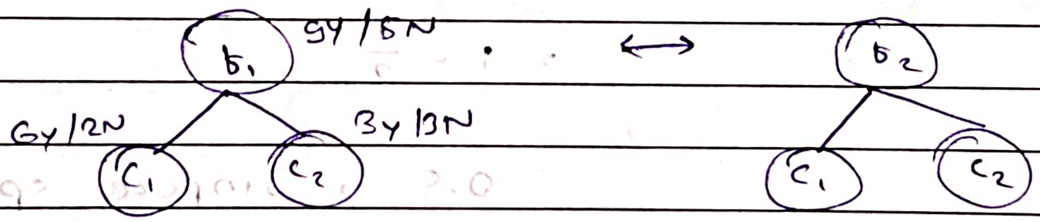
$$Gain(S, f_i) = H(S) - \sum_{v \in value} \frac{|S_v|}{|S|} \times H(S_v)$$

$H(S)$ → Entropy of root node.

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

Ex.

f_1 f_2 f_3 O/P.



$$H(f_1) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

$$= -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 5/14$$

$$= 0.94$$

$$H(c_1) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

$$= -\frac{6}{8} \log \frac{6}{8} - \frac{2}{8} \log \frac{2}{8}$$

$$= 0.81$$

$$H(c_2) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

$$= 1$$

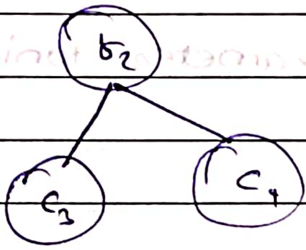
$$Gain(S, f_1) = H(S) - \sum_{v \in val} \frac{|S_v|}{|S|} H(S_v).$$

$$= 0.94 - \left[\frac{8}{14} \times H(c_1) + \frac{6}{14} \times H(c_2) \right]$$

$$= 0.94 - \left[\frac{8}{14} \times 0.81 + \frac{6}{14} \times 1 \right]$$

$Gain(S, f_1) = 0.049.$

for f_2 :-



let suppose
$Gain(S, f_2) = 0.051$

So we find out $Gain(S, f_1) < Gain(S, f_2)$

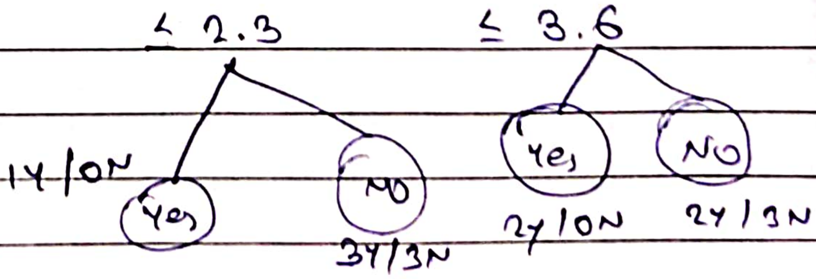
$\therefore$ So we start splitting from f_2.

for numerical category :-

1. Sort the feature value.

f_1	O/P
2.3	Yes
3.6	Yes
4	NO
5.2	No
6.7	Yes
8.9	NO
10.5	Yes

★ Thershold :-



3. Calculate Information Gain

Post Pronning & Pre Pruning :-

Pruning :- Used for reduce overfitting.

* Post Pruning :- Applied for small Dataset.
  1) Grow full / complete Decision tree.
  2) Prone it with respect to depth.

* Pre - Prunning :-
  1) Hyperparameter tuning with constructing tree.