Decision Trees

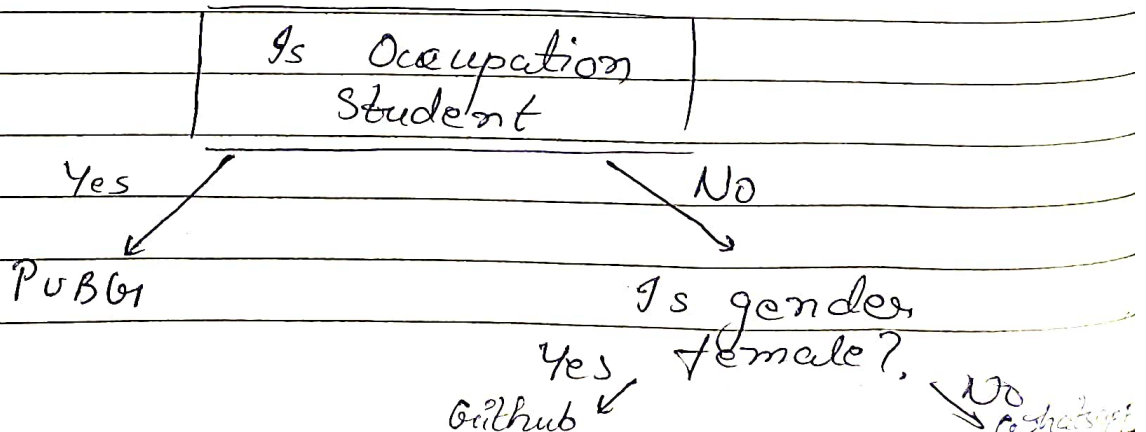
E.g :-

Gender	Occupation	Suggestion
F	Student	PUBG
F	Programmer	Github
M	Programmer	Whatsapp
F	Programmer	Github
M	Student	PUBG
M	Student	PUBG.

In python using nested if-else

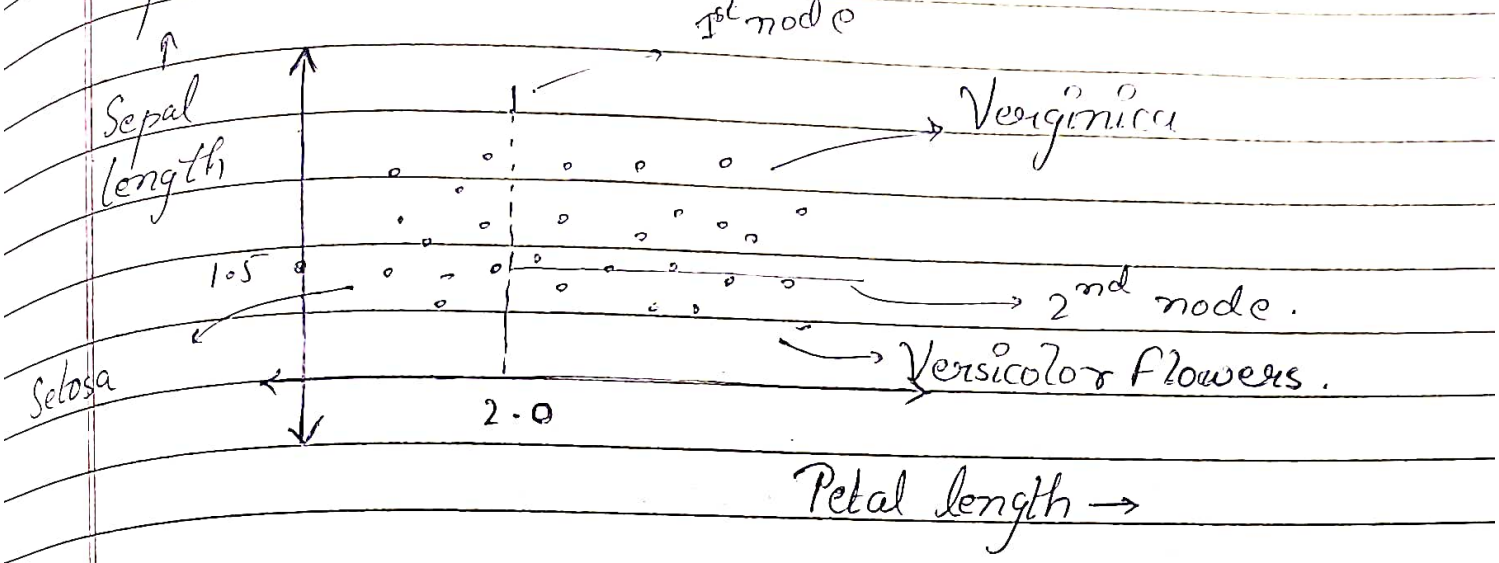
```
If occupation == student
        print (PUBG)
Else
        if gender == female
                print (Github)
        else
                print (whatsapp).
```

↳ Where is the Tree ?



E.g Iris Dataset.

Geometric Intution:



Pseudo Code:

↳ Begin with your training dataset, which should have some feature variables as classification or regression output.

↳ Determine the 'best feature' in the dataset to split the data on; more on how we define "best feature" later

↳ Split the data into subsets that contain the correct values for this best feature. This splitting basically defines a node on the tree. i.e each node is a splitting point based on a certain features from our data.

↳ Recursively generate new tree nodes by using the subset of data created from step 3

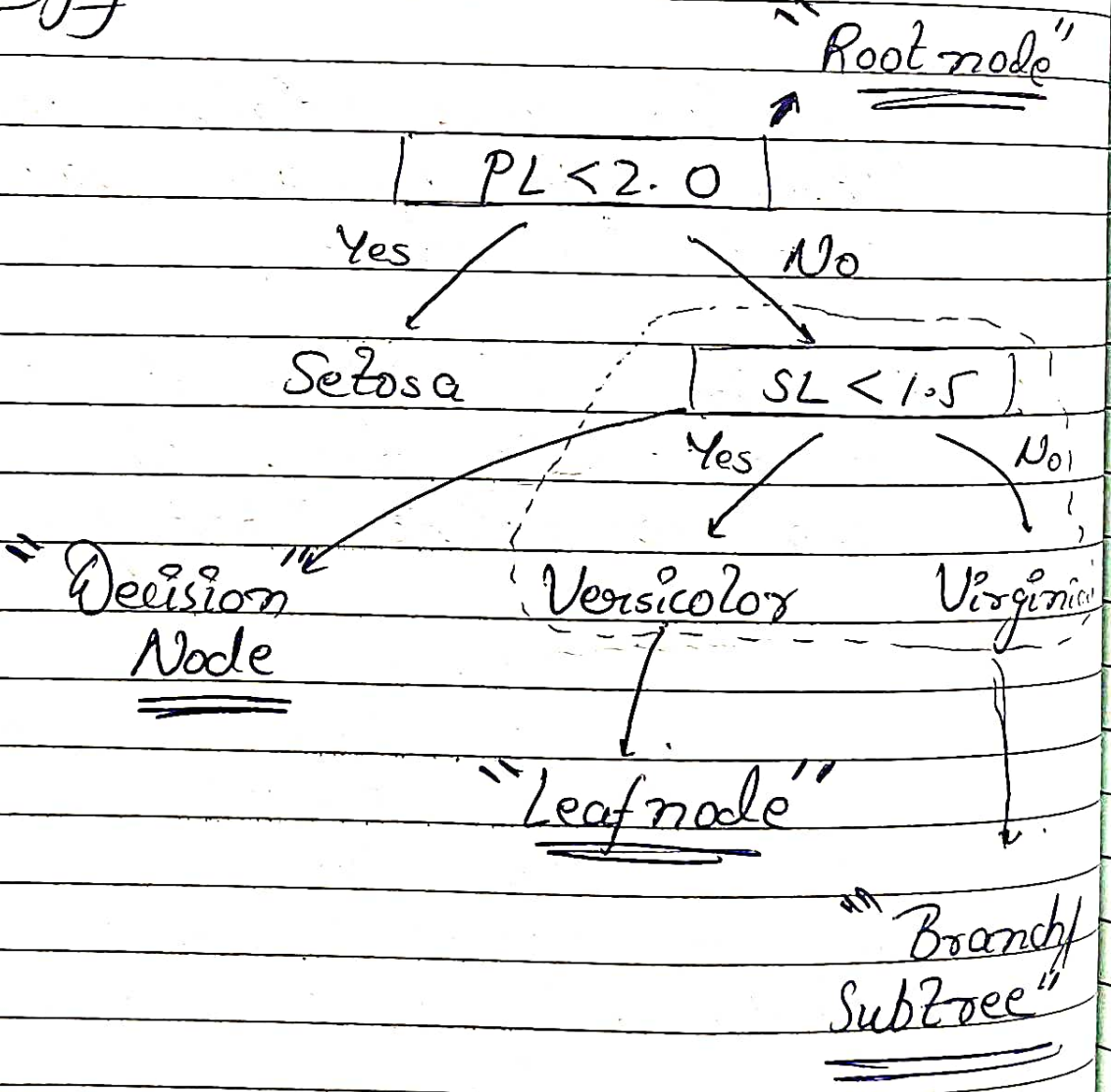
Conclusion :-

Programatically speaking, Decision trees are nothing but a giant structure of nested if-else condition.

Mathematically speaking Decision trees use hyperplanes which run parallel to any one of the axes to cut the coordinate system into hyper cuboids.

Terminology :

Sample Decision Tree



Some unanswered ques.

- How to decide which col. should be considered as root node?
- How to select subsequent decision node?
- How to decide splitting criteria in case of numerical columns?

Advantages:

- Intuitive and easy to understand.
- Minimal Data preparation is required.
- The cost of using the tree for inference is logistic in the number of data points used to train the tree.

Disadvantages:

- Overfitting.
- Prone to errors for imbalanced datasets.

Points:

- Decision Trees are mostly used for classification problems but can also be applied to regression, hence the name CART.

/.

Classification & Regr Trees

Working of decision Tree

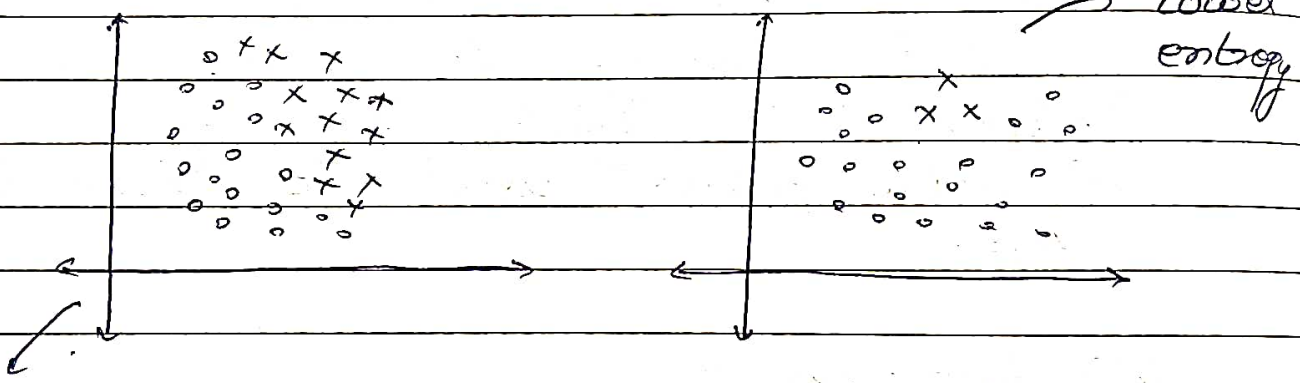
Entropy ? (Degree of randomness)

Entropy is nothing but the measure of disorder. Or we can also call it the measure of purity / impurity.

Calculated only on the basis of label

Higher entropy → impure
Lower entropy → pure

E.g. :



Higher entropy — more knowledge less entropy

Mathematical formula of entropy.

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

i => classes in label

Where P_i is simply the frequentist probability of an element / class 'i' in our data.

for e.g if our data has only 2 class labels Yes and No.

$$E(D) = -p_{yes}\log_2(p_{yes}) - p_{no}\log_2(p_{no})$$

Eg. Dataset

D(1)

Salary	Age	Purchase
20000	21	Yes
10000	45	No
60000	27	Yes
15000	31	No
12000	18	No

D(2)

Salary	Age	Purchase
34000	31	No
15000	25	No
69000	57	Yes
25000	21	No
32000	28	No.

$H(d) = -P_y\log_2(P_y) - P_n\log_2(P_n)$

$H(d) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)$

$H(d) = 0.97$

→ Higher entropy.

$H(d) = -\frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{4}{5}\log_2\left(\frac{4}{5}\right)$

$H(d) = 0.72$

---

for 3 class or more classes. (Yes, No, May be)

$$H(d) = -P_y\log_2(P_y) - P_n\log_2(P_n) - P_m\log_2(P_m)$$

Observation :

↳ More the uncertainty more is entropy.

↳ For a 2 class problem the min entropy is 0 and the max is 1.

↳ For more than 2 classes the min entropy is 0 but the max can be greater than 1.

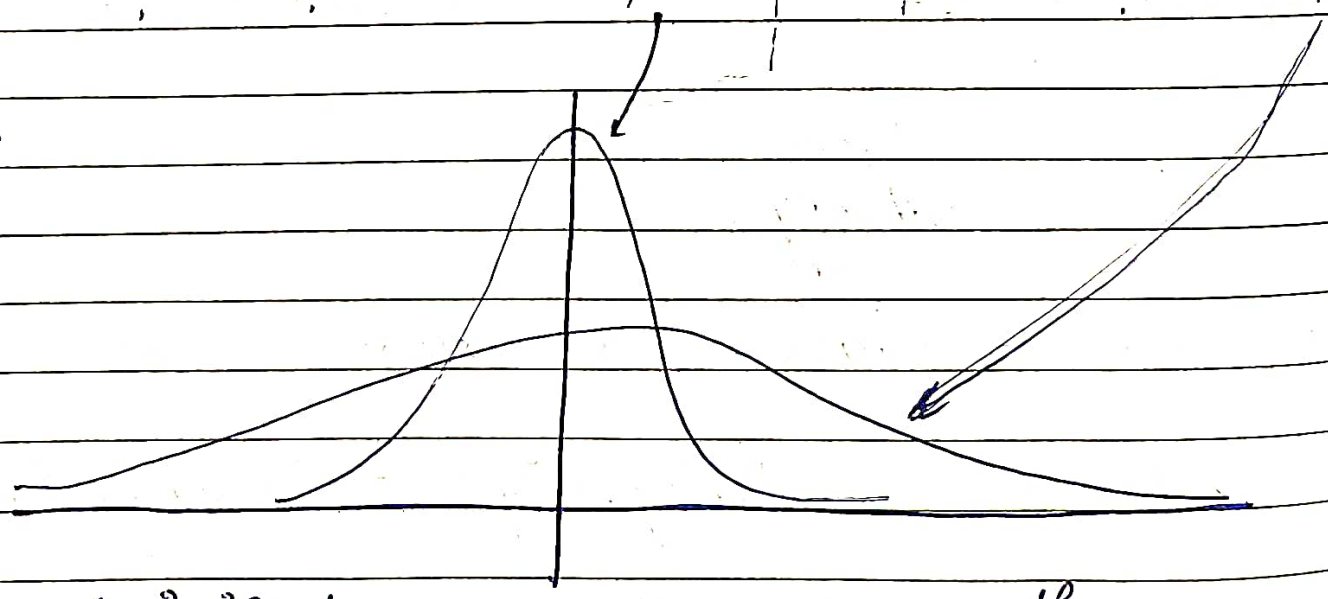
↳ Both $\log_2$ or $\log_e$ can be used to calculate entropy.

- Entropy for continuous variable

Eg:

Area	Built in	Price	Area	Built in	Price
1200	1999	3.5	2200	1989	4.6
1800	2011	5.6	800	2018	6.5
1400	2000	7.3	1100	2005	12.8
⋮	⋮	⋮	⋮	⋮	⋮

↳



↳ So intuitively we can say the

randomness of label in dataset 2 is more. So, its entropy is higher than dataset 1.

Information Gain:

It is a metric used to train Decision Trees. Specifically, this metric measures the quality of a split.

The information gain is based on the decrease in entropy after a data-set is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain.

Information gain = E(parent) − E[{Weighted Average} * E{children}]

weighted entropy

Step 1:- entropy of parent.

→ entire dataset

Outlook

Sunny [consider smaller dataset contain rows where outlook is sunny]

Overcast [" " is overcast]

Rain [" " is rain]

Step 2 : Calculate entropy for children

Each child i.e for sunny, overcast & rain

Step 3 : Calculate weighted Entropy of children

Weighted Entropy = $\frac{\text{no. of row in sunny dataset}}{\text{Total no. of row in the entire dataset}}$ × Entropy of Sunny

\+ $\frac{\text{" " in overcast}}{\text{" " " "}}$ × " " overcast

\+ $\frac{\text{" " in rain}}{\text{" " " "}}$ × " " rain

Step 4 : Calculate Information Gain

So the information gain when we split this data on the basis of Outlook col. is say 0.28

Step 5 : Calculate Information Gain for all the col..

"Whichever column has the highest Information Gain (maximum decrease in entropy) the algorithm will select that column to split the data"

- Gini Impurity :

This is just like entropy, a measure to check the purity / impurity of data column.

Only difference is formula :-

$$G_I = 1 - \Sigma p^2(i)$$

i.e for 2 class classification.

$$G_I = 1 - (p^2(y) + p^2(n))$$

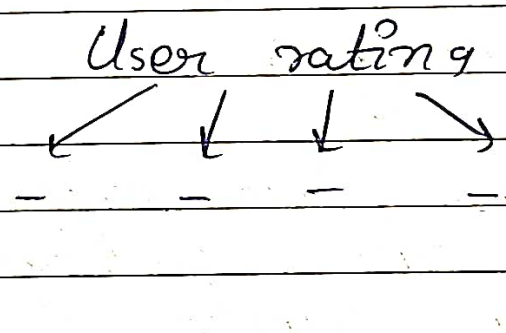
Higher Gini → Impure

The only benifit is, it is computationally fast.

Handling Numerical Data.

E.g.

S.No	User Rating	Downloaded
1	3.5	Yes
2	4.6	Yes
3	2.5	No
4	1.6	Yes
⋮	⋮	⋮



So computationally very much infeasible.

Step 1 :

Sort the data on the basis of numerical column.

Step 2 :

Split the entire data on the basis of every value of user-rating.

i.e say User rating ≥ 1.6

Yes → dataset containing every row except one

No → only one row logically

again say

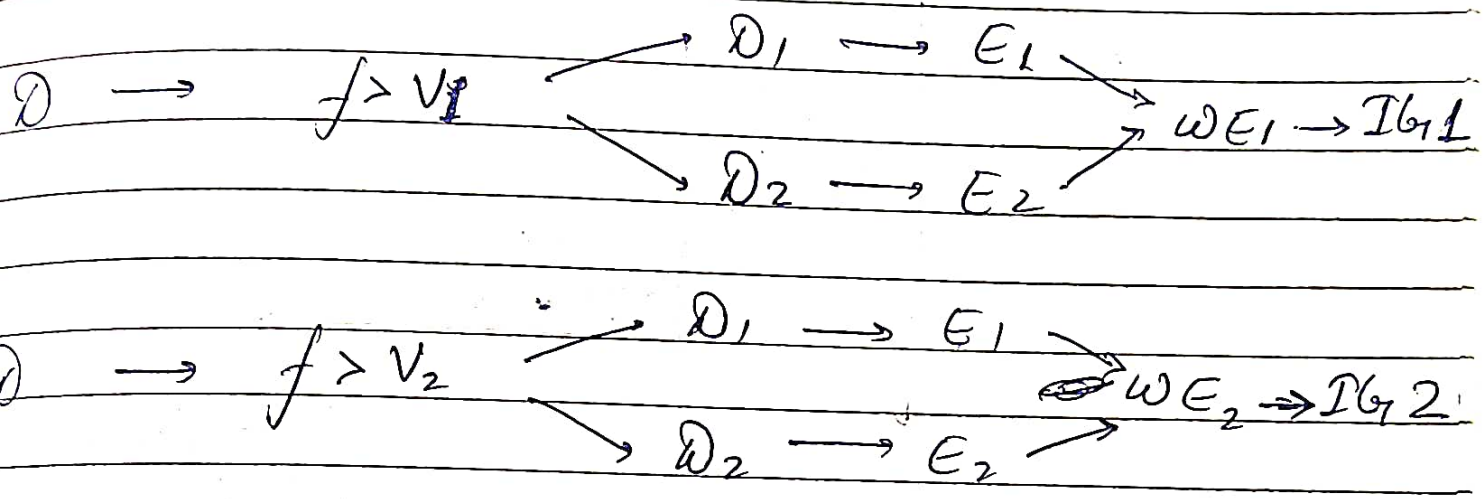
User rating > 2.5 ?

Yes → 2 rows (3.5, 4.6) dataset

No → 2 row (2.5, 1.6) dataset

and keep on repeting.

So.



Step 3 :-

Max { IG1, IG2, IG3 .... }.

based onto which we select ouce splitting creiteria.

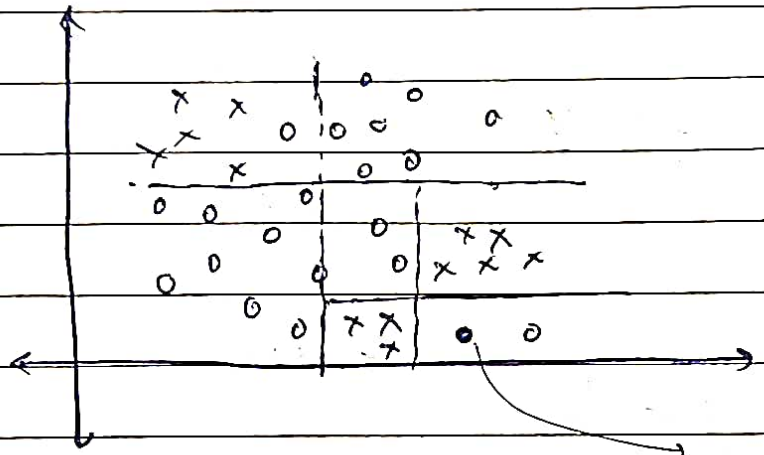
Decision Trees Hyperparameter

To reduce overfitting

Performs well → Training
Performs badly → Test

Max_depth

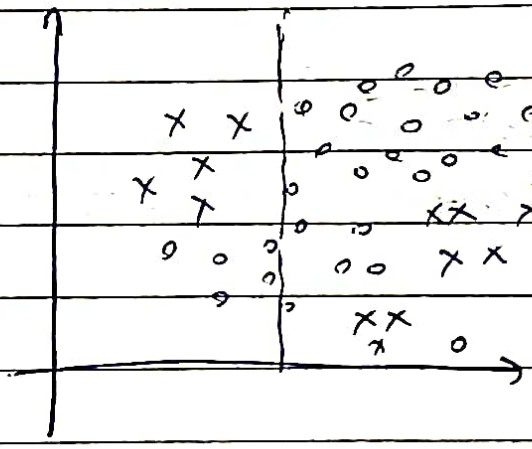
E.g:



consider this point.

- max_depth = 'None'
- ↓
- overfitting

logically it seems that it should be x but decision tree classified it as o.



- max-depth = '1'
- ↓
- Underfitting

Splitter = 'best' or 'random'

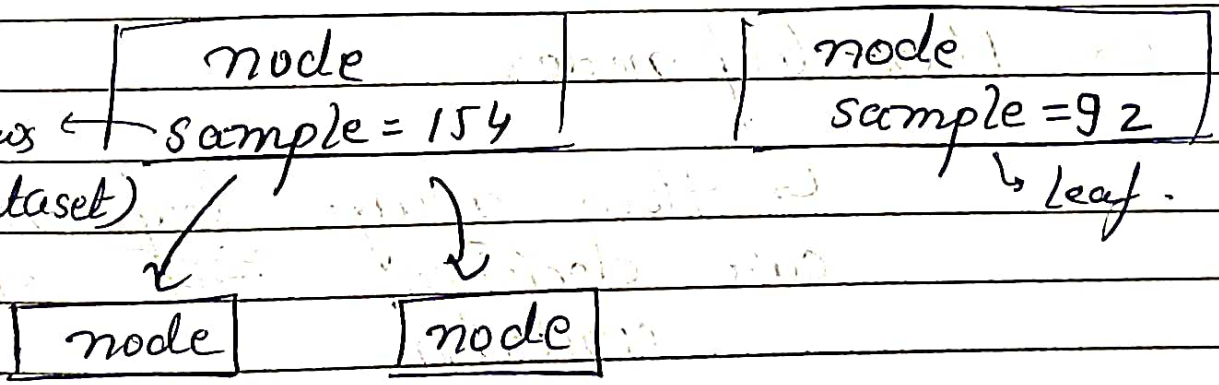
'best': Split best fitting criteria i.e highest Info. gain

'random': Makes any random split (best for reducing over-fitting)

Min Samples split :

Using this we can decide that if, minimum = x rows will present on a node (means node dataset) then an only then the split happen rather the node becomes a leaf

Say min Samples split = 100



Min Sample leaf :

means if the leaf node on splitting comes down to min sample leaf value, it will not allow to split.

E.g minSampleleaf = 100

node
sample = 116

↳ no further split (because if we split the no. of sample reduces below 100)

Means the leaf node must have 100 no. of samples.

↳ Max features
↳ How many cols. to be given to decision tree

↳ Max leaf nodes
↳ How many leaf node our decision tree should contain