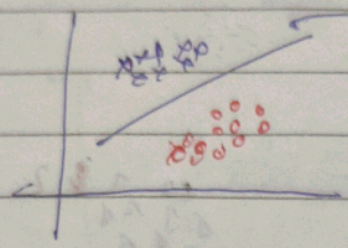
Decision Tree

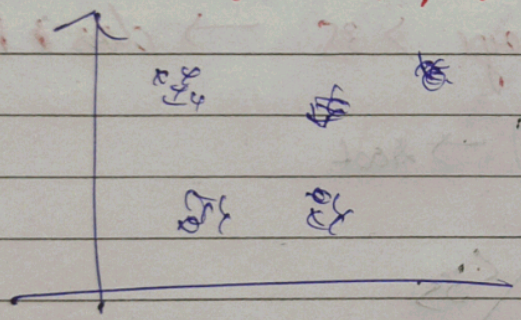


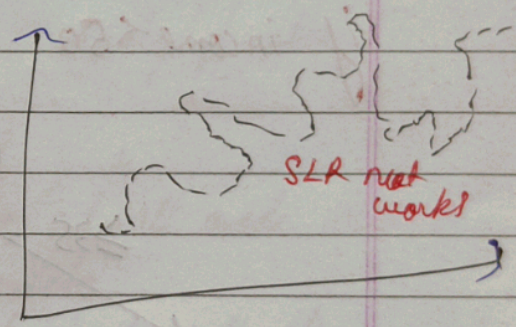
→ till now we have learnt about logistic Regression & linear Regression

Logistic Regression → makes for linear decision boundary

Linear Regression → captures linear Relationship b/w independent variable

Problems so far with linear regression and logistic regression → captures linear relationship b/w variables. they were not able to capture relationship among non-linear variables

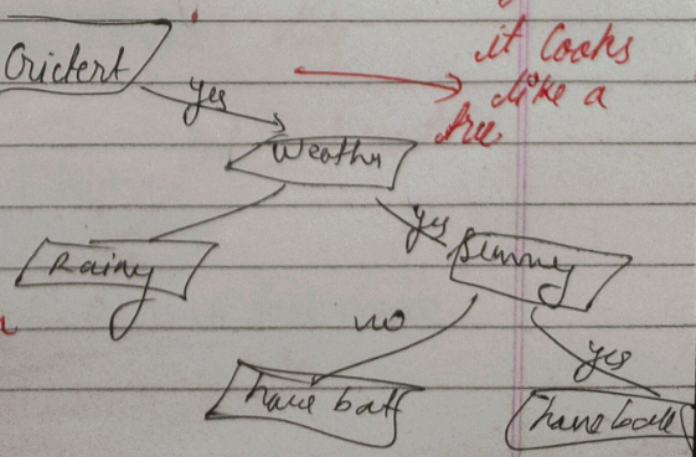




for that we have Decision Tree

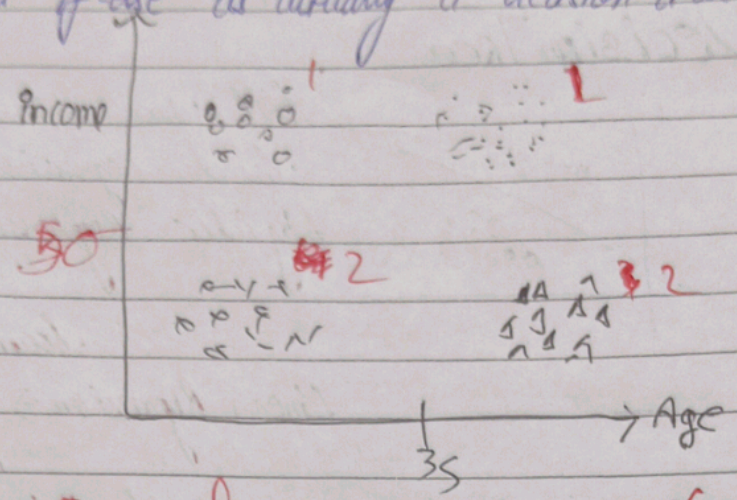
Decision Tree Classifier — Decision Tree Regressor

Note ⇒ Decision Tree works On if-else condition
↓
Decision Tree
At every node a decision is taken.



→ it looks like a tree

Now nested if-else is actually a decision tree

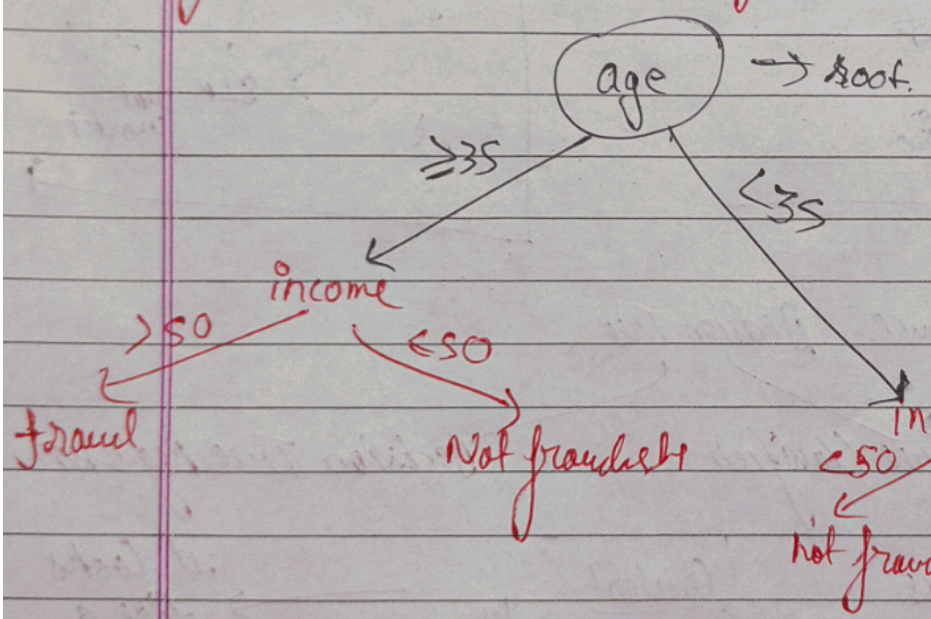


if income ≥ 50 & age ≤ 35 → class 1

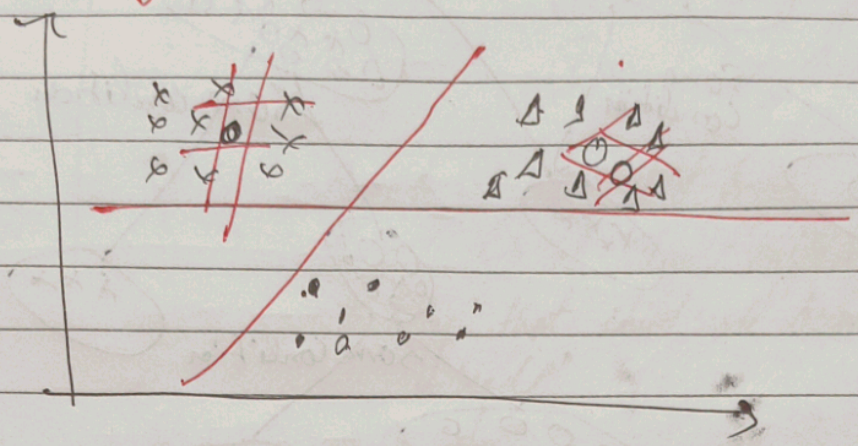
if income < 50 & age ≤ 35 → class 2

if income ≥ 50 & age > 35 → class 2

if income < 50 & age > 35 → class 1



in Decision tree it divides the data into pockets ~~atts~~ (also called as nodes) until all / each data point of same class



Advantage

→ Simple to understand

→ Captures non-linear relationship

→ DT → divides data into nodes/ pocket

↓

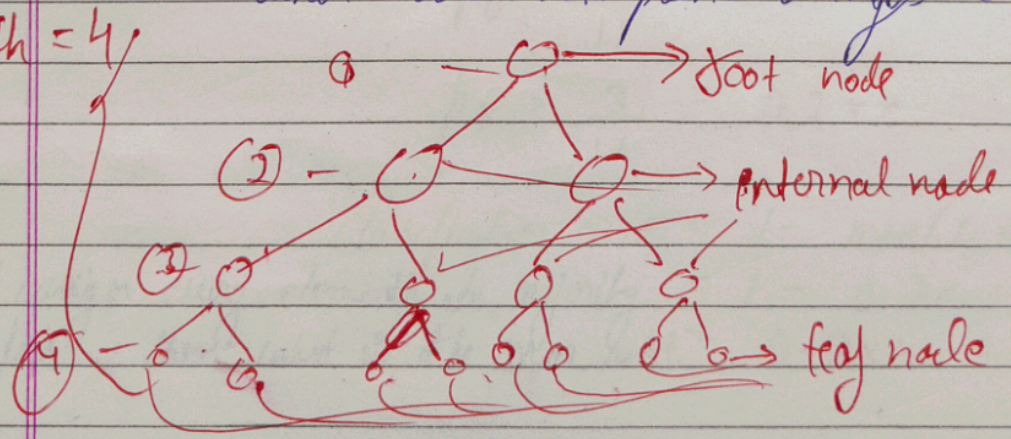
but until when?

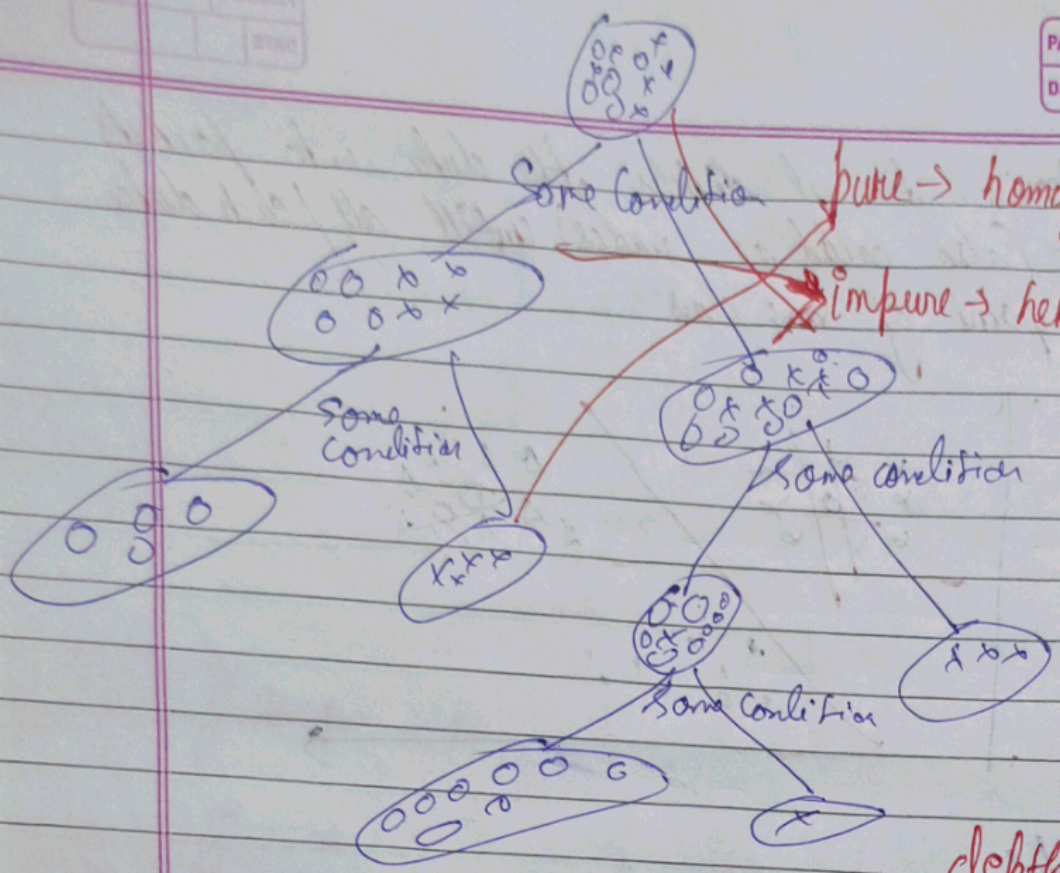
↳ till we get all node become pure node

↳ pure node → all dp's were of same class

DT → intends is to create pure leaf node/pocket where each data points belongs to one class

depth = 4





depth = 3

next que is How to split ~~a~~ that on some condition & what will be that some condition?

① which feature should we take at each step?

impurity measure

Age	Salary	fraud
–	–	1
–	–	1
–	–	0
–	–	0
–	–	1

① How to split
① on what features
② on what condition

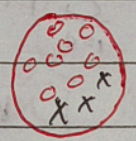
Agenda → How split happen
↳ it should happen in such a way that child node becomes pure node.

↓

for that we have impurity measure

Types of impurity measure
① classification error
② entropy
③ Gini

① Classification error



0 : 8
x : 3

classification error is given by $= (1 - \max(p_i))$

Classification error $= 1 - \max(p_i)$

$$p_0 = \frac{8}{11} = 0.7xx$$

$$p_x = \frac{3}{11} = 0.2xx$$

Classification error $= 1 - \max(0.7x, 0.2x)$

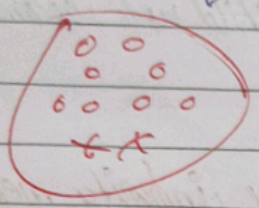
* If i assign every element to majority class, then what is the error rate? $= 1 - 0.7x$
$= 0.3xx$

Classification error is simplest among all, but now in practical implementation we do not use it as majority.

(2) gini

$$\boxed{GI = 1 - \sum_{i=0}^{n-1} (p_i)^2 \quad \text{or} \quad \sum_{i=0}^{n-1} p(i)(1-p(i))}$$

↳ from class (0 to n-1) / (1 to n)



$$GI = 1 - \sum_{i=0}^{2-1} (p(i))^2$$

$$= 1 - [(0.8)^2 + (0.2)^2]$$

$$= 1 - [0.64 + 0.04]$$

$$= 1 - 0.68$$

$$= 0.32$$

for two class → $\sum_{i=0}^{2-1} p_i(1-p_i)$

$$= p_0(1-p_0) + p_x(1-p_x)$$

$$= 0.8(0.2) + 0.2 \times 0.8$$

$$= 0.16 + 0.16$$

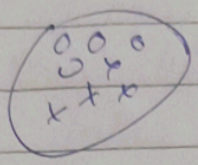
$$= 0.32$$

path are ~~[illegible]~~ same

Note → Highest gini = 0.5

~~plot of gini is symmetrical for us~~

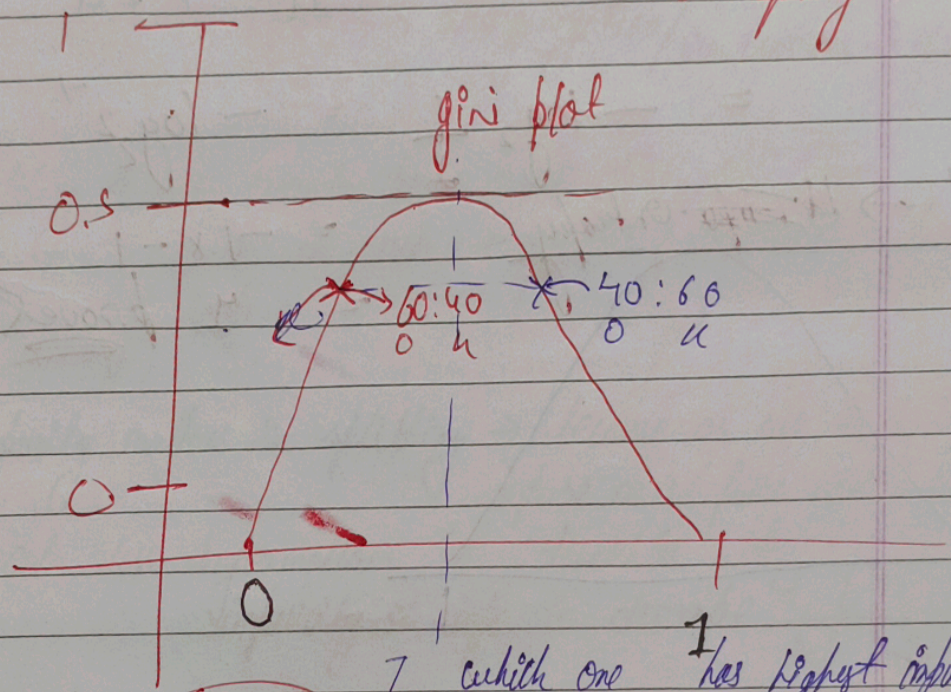
impure node



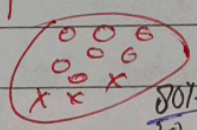
$p_o = 50\%$
$p_x = 50\%$

both 50% → highest impurity
↓
highest gini
↓
all classes are equally distributed

$$gini = 1 - 0.25 - 0.25$$
$$= \underline{0.5}$$

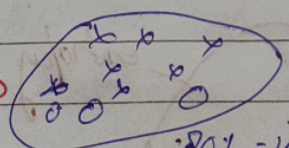


Node 1 →



80% - o
20 - x

Node 2 →



80% = x
20% = o

which one has highest impurity?

Ans → both are equally impure

(3) entropy

$$H(S) = -\sum_{i=0}^{n-1} p_i \log_2 p_i$$

$$-p_o \log p_o - p_x \log p_x$$

max impurity → 0.50 → gini

highest value of entropy = 1

Highest value of entropy = 1 (for $H_0 = 0.5$, $H_1 = 0.5$)

Proof

$p_0 = 0.5 = 1/2 = p_x$

$$\underbrace{-\frac{1}{2}\log_2\frac{1}{2}}_{a} \underbrace{-\frac{1}{2}\log\frac{1}{2}}_{a} = 2 \times -\frac{1}{2}\log_2\frac{1}{2}$$

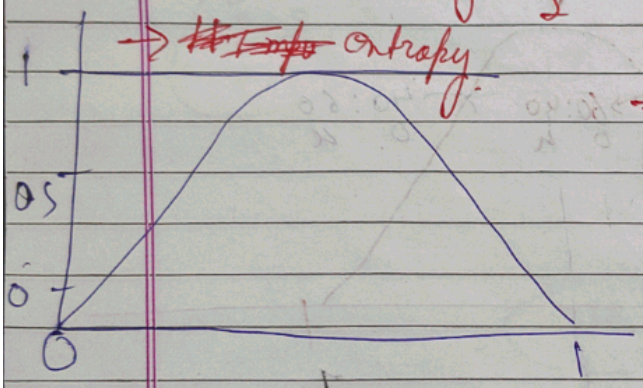
$-2a$

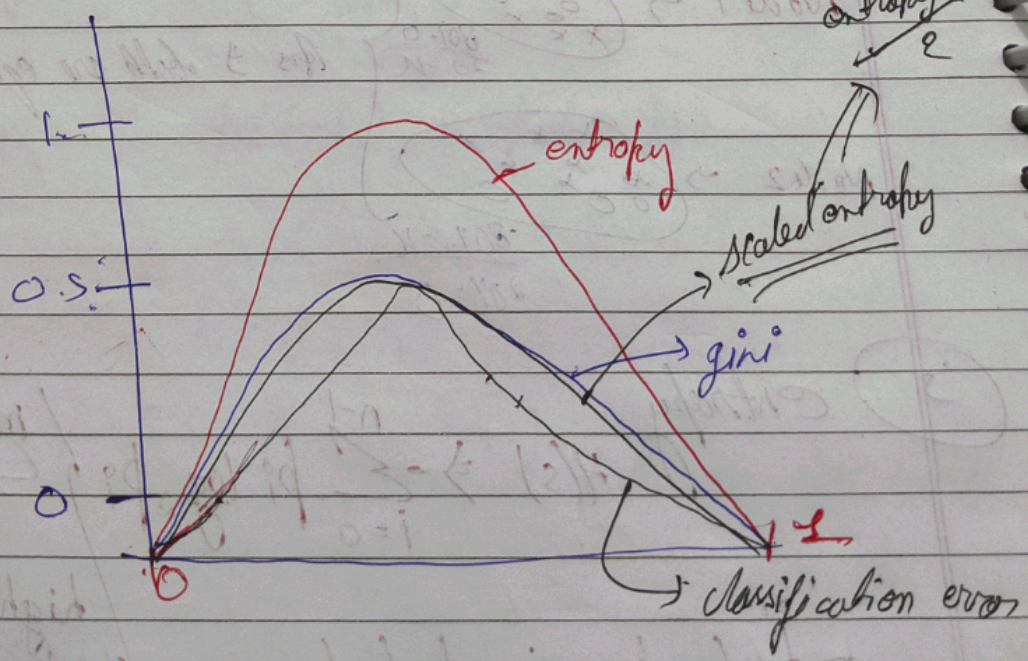
$$= -\log_2\frac{1}{2} = -\log_2 2^{-1}$$

$$= -1 \times -1$$

$= 1$ proved

→ ~~Info~~ entropy

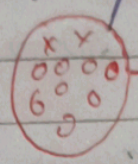




Note → feature with highest info gain is best to split

Till now we have learnt about impurity, impurity measure
Now lets understand ~~to~~ How we split and ~~on~~ what features to split

on what features to split ⇒ information gain



→ disorganization/randomness is there

*Aim of spliting → we want pure nodes after splitting
→ split in such a way we get maximizes purity of node.

Why impurity matters in splitting → because as we have less impure node (pure node) we need less information to describe it

impurity ∝ info to describe $\propto \frac{1}{\text{info gain}}$

impurity (post ~~split~~ split) < impurity (pre split)

→ split in such a way that it satisfy the above sentence at best.

* this change impurity is given by the difference of post split and pre-split impurity

Δ impurity = impurity (pre) − impurity (post)

less the impurity higher the information gain

information gain is a measure to define the degree of disorganization.

Info gain → gini to calculate IG
→ entropy to calculate IG

gender	class	cricket
—	—	Y
—	—	N
—	—	Y
—	—	N
—	—	N

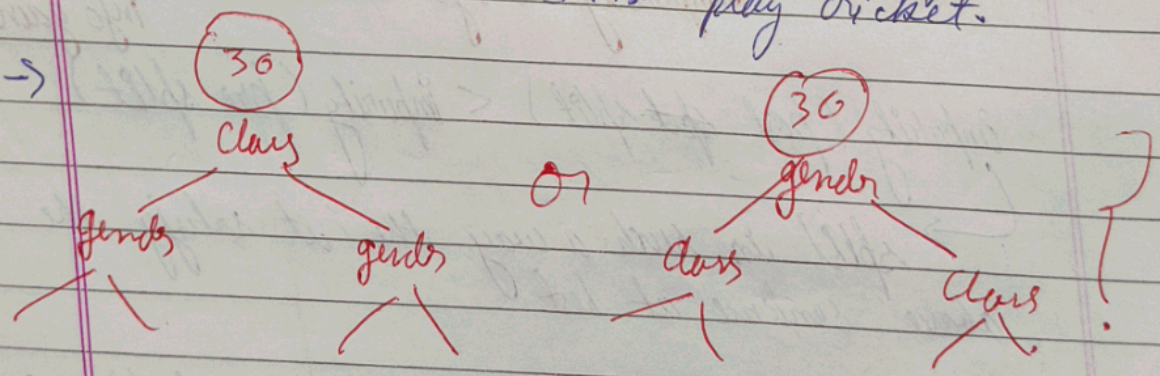
Total = 30

30 → 50% play cricket

gender → out of 10 female - 2 play cricket
↳ out of 20 male → 13 play cricket

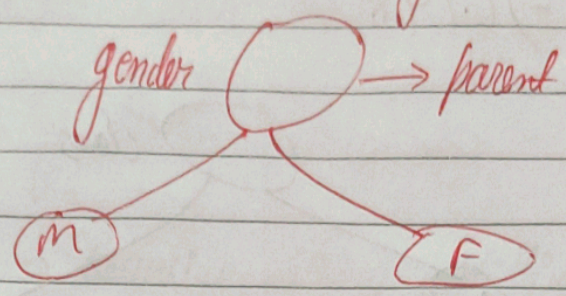
Class → out of 14 student of class 9, 43% play cricket.
out of 16 student of class 10, 57% play cricket.

Q →



whichever feature has highest IG, will be used

IG for "class col" & "gender col"



$$IG = \text{entropy(pre)} - \text{entropy(post)}$$

$$= en(\text{parent}) - en(\text{child combined})$$

$$en \Rightarrow -p_i \log_2 p_i - (1-p_i)\log_2(1-p_i)$$

$$en(\text{parent}) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$en(\text{male child}) = -\frac{13}{20}\log_2\frac{13}{20} - \frac{7}{20}\log_2\frac{7}{20} \approx 0.92$$

$$en(\text{Female child}) = -\frac{2}{10}\log_2\frac{2}{10} - \frac{8}{10}\log_2\frac{8}{10} \approx 0.72$$

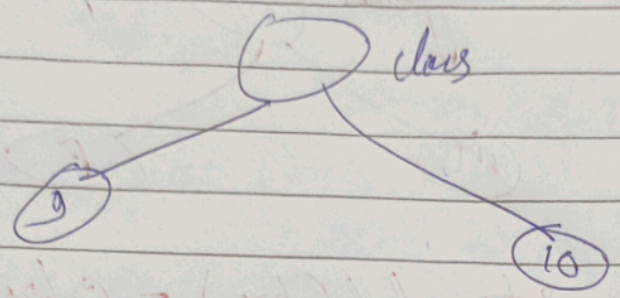
$$en(\text{child combined}) = \frac{\text{no. of female}}{\text{total no. of stu}} \times en(\text{female}) + \frac{\text{no. of male}}{\text{take student}} + en(\text{male})$$

$$= \frac{10}{30} \times 0.72 + \frac{20}{30} \times 0.92$$

$$= 0.24 + 0.612$$

$$en(\text{child combined}) = 0.852$$

$IG = 1 - 0.852$
$\Rightarrow 0.148$



en(parent) − en(Child Combined) = IG

en(parent) = 1

en(9th class) = $-0.43 \log 0.43 - 0.57 \log_2 0.57$
=

en(class 10) = $-0.57 \log 0.57 - 0.43 \log_2 0.43$

En(Child Combined) $\Rightarrow (-0.43 \log 0.43 - 0.57 \log 0.57) \times \frac{14}{30}$
$+ (-0.57 \log 0.57 - 0.43 \log_2 0.43) \times \frac{16}{30}$
≈ 0.99

$Ig = 1 - 0.99$
$= 0.01$

Since IG of gender > IG of class, then using gender column to split

We will repeat this step at every stage to decide ~~which~~ which column to use for splitting.

Algorith of decision tree

step 1 -> Recursive binary splitting / partitioning of data into smaller subset.

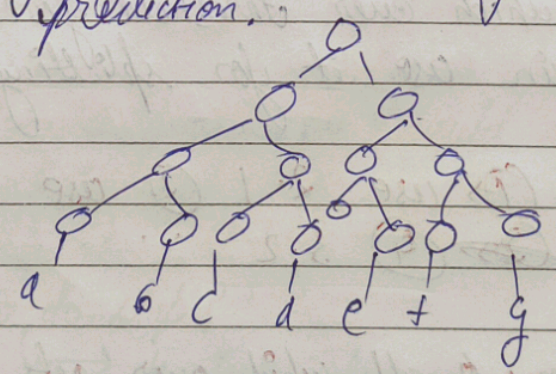
Step 2 -> Select the best variable for split (IG)

Step 3 -> Apply the split

Step 4 -> repeat the process for subset obtained

Step 5 -> Continue the process until every node is a pure node / Stopping criteria is reached

Step 6 -> majority value in a leaf node will be prediction.



Decision Tree is a greedy algo → keeps splitting till all node become pure.

What if the feature is continous?

# of hours studied	# of output / P/F
2.1	F
3.5	F
4.1	P
3.2	F
5.3	P
6.8	P
9.8	P
3.0	F

1. Sort the feature as P/F
2. Treat each value/row as cut off / threshold and try to build decision tree.
3. which ever threshold give maximum info gain use it for splitting.

Eg → (1) use 2.1 (2) use 3.5 (3) use 4.1 (4) 3.2 ..... (9.8)

→ among all which ever gets highest I.G use it for splitting.

disadvantage = Time Complexity ↑

Decision Tree is greedy algorithm

↓

DT will be fully grown until each or leaf node is not pure node

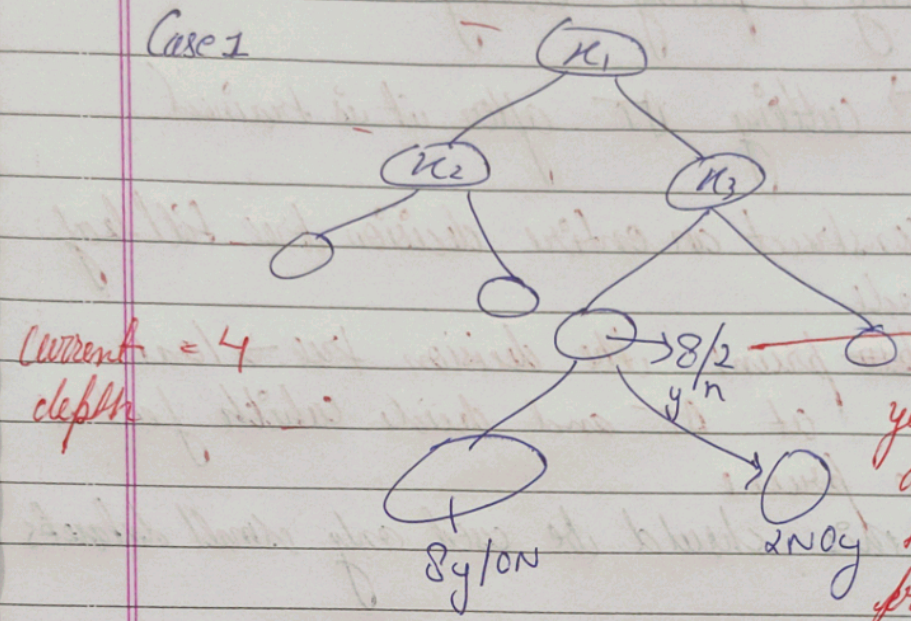
⇓

and it leads to overfitting

* but we want a generalise model → low bias, low variance

train acc ↓, test acc ↑

Case 1



current depth = 4

Here in this node majority is yes class, so all data point comes to this node should be predicted 'yes', then why we split it again.

This will decrease depth ← we should stop Here.

& reduce overfitting (stop 1 node earlier)

↓

But How to stop?

→ Two Techniques

1 → post pruning

2 → pre pruning

3 → ensembling.

Parameter to create Decision Tree

Criterion = ['gini', 'entropy', 'log_loss']
splitter
max depth = ; ~~min depth~~ = min_sample_split,
min samples , max_features

① Post pruning → pruning = cutting
→ cutting DT after it is trained

Step1 → construct an entire decision tree till leaf node
Step 2 → ~~pu~~ prune the decision tree → look at DT and decide which feature to prune
Note → This should be used only small datasets

② Pre pruning →
↳ limit the growth of DT
→ Hyperparameter tuning to select best params.