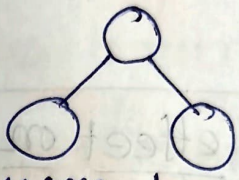
AdaBoost:

1. combines lot of weak learners.

↓

stump



2. some stumps have more-to-say in the final decision unlike RF.

3. Each stump try to reduce previous stump's mistake.

↳ by sampling wrongly classified data more times by increasing the weights of wrongly classified data.

Steps:

① assign sample weight.

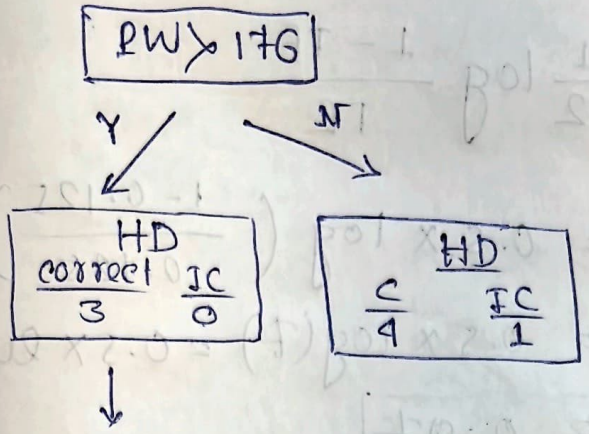
say, $n = 8$

$w_i = 0.125$

initially, $w_i = \frac{1}{n}$

n = row numbers

② split by every attribute and calculate Gini Impurity & weighted GI.



- $\left(\frac{3}{3}\right)^2 + \left(\frac{0}{0}\right)^2$ = 1
- $\left(\frac{4}{5}\right)^2 + \left(\frac{1}{5}\right)^2 = \frac{17}{25} = 0.68$

- weight for Y = $\frac{3}{8}$
- weight for N = $\frac{5}{8}$

- $WGI = 1 - \left(1 \cdot \frac{3}{8} \times 1 + \frac{5}{8} \times 0.68\right)$
  $= 1 - (0.375 + 0.425$
  $= 1 - 0.8$
  $= 0.2$

→ based on the lowest WGI, choose the attribute and split it.

③ Now calculate Total Error : # of rows misclassified by the attribute.

Ex: say if we split by FW, 1 row get misclassified.

↳ Then total error = $\frac{1}{8} = \frac{1}{n}$

④ Now calculate amount of say for that stump:

- Amount of say $= \frac{1}{2} \log \frac{1 - TE}{TE}$

For, f_W, $AoS = 0.5 \times \log\left(\frac{1 - 0.125}{0.125}\right)$

$= 0.5 \times \log(7) = 0.5 \times \ln(7)$

$\boxed{AoS = 0.97}$

⑤ Now assign new sample weight:

a) for Misclassified:

- New SW = old SW $\times e^{AoS}$

b) for correctly classified:

- New SW = old SW $\times e^{-AoS}$

and nomalize them:

$\tilde{w}_i = \frac{w_i}{\sum w_i}$ w_i = New SW

⑥ Now assign range for each row:

w_i		r_i	range	
0.07		1	0 - 0.07	↓ cummulative sum
0.07				
0.07	→	2	0.07 - 0.14	
0.49		3	0.14 - 0.28	
0.07				
0.07		4	0.24 - 0.77	→ misclassified one have max width.
0.07		5	0.77 - 0.84	
0.07				
⋮		⋮	⋮	
			1	

⑦ Now generate random number b/w 0 to 1 and choose the row in which bin it falls.

↳ misclassified one ie, having max bin width will have max probability to get picked up.

↳ this way sampling → create new dataset.

⑧ Now repeat from step 2:

↳ and make all sample weights to again = $\frac{1}{n}$

→ can not handle missing values by default.

→ sensitive to outliers.

↳ can have large impact on the weights in each iterations.

⑦ Now generate random number b/w 0 to 1 and choose the row in which bin it falls.

↳ misclassified one ie, having max bin width will have max probability to get picked up.

↳ this way sampling → create new dataset.

⑧ Now repeat from step 2:

↳ and make all sample weights to again = $\frac{1}{n}$