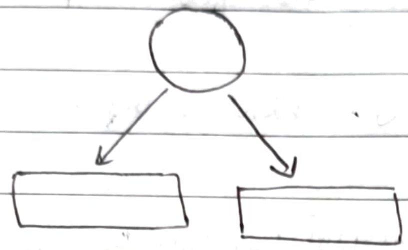
→ stands for (ADAptive BOOSTing)

Adaboost (theory) [statquest YT]

(1) In adaboost we have multiple trees but these trees have only one node and two leaves.



} such a tree is called a "STUMP"

stumps alone are not good at prediction

So in adaboost we have multiple stumps. ⇒ Forest of stumps.

(2) In random forest all trees have equal vote in final outcome.
In adaboost however some ~~stup~~ stumps have "more say" in final outcome than others.

(3) In a random forest each decision tree is made independent of each other.
In adaboost ⇒ order is important ⇒ errors made by first stump influence how second stump is made.

Adaboost is not a model in itself ⇒ Rather it can be applied on any classifier to learn from it's shortcomings and get a more accurate model.

Steps for adaboost in decision trees :→

1) A stump is made on training data based on weighted samples.

indicates how important it is to correctly classify a sample.

Initially we give all the samples equal weight.

2) We create decision stump for each variable

$\Downarrow$

And we see how well each stump performs.

3) More weight is assigned to incorrectly classified samples.

$\Downarrow$

so they are classified correctly in next stump.

Also, weight is assigned to each classifier too on the basis of accuracy.

4) Reiterate from step 2 until → All points are correctly classified

→ Maximum iteration level is reached.

<u>Mathematics behind adaboost for decision trees:</u>

~~$x^{(i)} \in R^n$ $y^{(i)} \in \{-1,1\}$~~

consider that our dataset has n features and m points

$$x_j \in R^m \ , \ y^{(i)} \in \{-1,1\}.$$

step 1) we assign weights to training data on basis of the significance of each data point.

Initially :→ $w_j = \dfrac{1}{M}$ $\quad j \in \{1,2,\dots m\}$

step 2) Then we create decision stump for each variable and we select the best stump from them ⇒ The one who classifies the best.

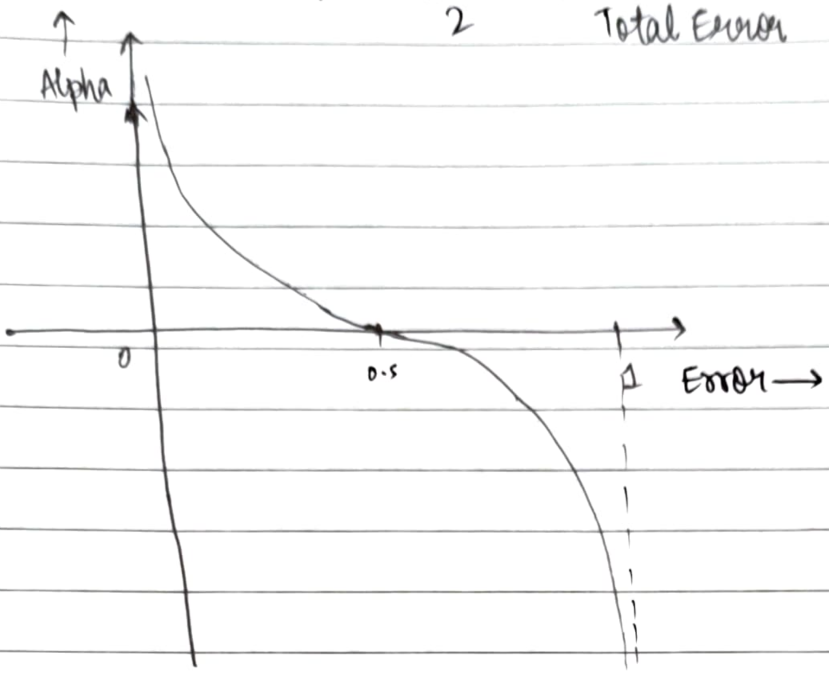
Sum of weights for incorrectly classified samples.

$$\alpha_t = \frac{1}{2} \ln \frac{(1-\text{Total Error})}{\text{Total Error}}$$

$t \in \{1, 2, \ldots n\}$

⇓

as we make stump for each feature and pick the one with maxm. α as it also has the least total error.



we choose stump such that

$$\text{stump selected} = \underset{\text{stumps}}{\text{argmax}}\ \alpha_{\text{stump}}.$$

step 3) Now we change the weights of data points

For incorrectly classified samples ⇒ $w_j := w_j * e^{\alpha}$

For correctly classified samples ⇒ $w_j := w_j * e^{-\alpha}$

These do not add upto 1 ~~thus~~ so we normalise ⇒ $w_j := \frac{w_j}{\sum_{k=1}^{m} w_k}$

here α is the weight assigned to the classifier

step 4) repeat from step 2.