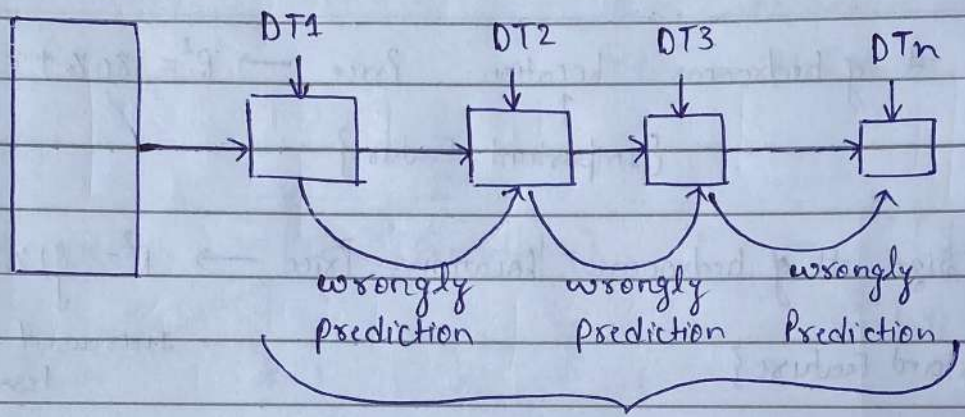
# AdaBoost

→ It is a supervised machine learning algorithm which is used to solve both classification and regression problem by using boosting technique.




$$f = \alpha_1(M_1) + \alpha_2(M_2) + \alpha_3(M_3) + \cdots + \alpha_n(M_n)$$

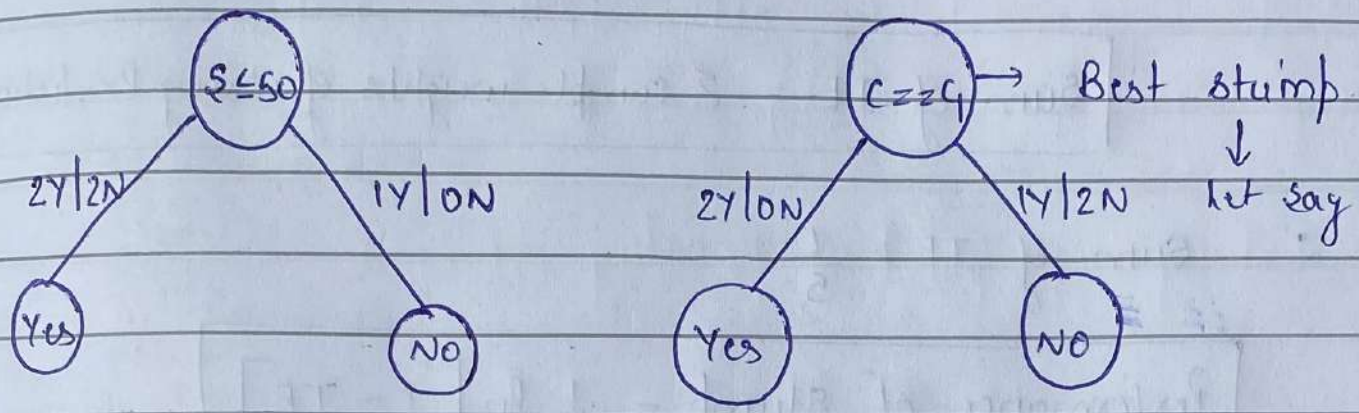
↳ $\alpha$ → weights

## * Working of AdaBoost :

Dataset

| I/P | | O/P |
|---|---|---|
| Salary | Credit | Approval |
| ≤50K | B | NO |
| ≤50k | G | Yes |
| ≤50k | G | Yes |
| >50k | N | Yes |
| ≤50k | N | NO |

① Create Decision tree Stump (Decision tree with level 1 only) and select the best stump :




Note : Select best Stump using Entropy or Gini and Information Gain.

② Initialize the sample weights

| Salary | credit | Approved | Sample weight | |
|---|---|---|---|---|
| ≤ 50k | B | NO | 1/5 | |
| ≤ 50k | G | Yes | 1/5 | |
| ≤ 50k | G | Yes | 1/5 | |
| >50k | N | Yes | 1/5 | → wrongly Predicted |
| ≤ 50k | N | NO | 1/5 | |

③ Calculate Sum of total errors and performance of stump:

$$\boxed{\text{Sum of TE} = \Sigma \text{ sample weights of wrong Prediction}}$$

$\therefore$ Sum of TE $= \frac{1}{5}$

$$\boxed{\text{Performance of Stump} = \frac{1}{2} \ln\left[\frac{1 - TE}{TE}\right]}$$

$\hookrightarrow \alpha$.

$$\Rightarrow \alpha_1 = \frac{1}{2} \ln\left[\frac{1 - 1/5}{1/5}\right] \approx 0.6$$

④ Update the weights for correctly and incorrectly classified point:

For correct classified point
$= \text{weight} * e^{-\alpha}$

$= \frac{1}{5} \times e^{-0.6} \approx 0.109$

For incorrectly classified point
$= \text{weight} * e^{+\alpha}$

$= \frac{1}{5} \times e^{+0.6} \approx 0.364$

| Salary | credit | Approval | SW | updated weight |
|---|---|---|---|---|
| ≤50k | B | No | 1/5 | 0.109 ↓ |
| ≤50k | G | Yes | 1/5 | 0.109 ↓ |
| ≤50k | G | Yes | 1/5 | 0.109 ↓ |
| >50k | N | Yes | 1/5 | 0.364 ↑ |
| ≤50k | N | No | 1/5 | 0.109 ↓ |

$\Sigma = 0.8$

⑤ Normalize weights computation and assigning bins :

$$\text{Normalize} = \frac{\text{updated weight}}{\Sigma \text{updated weights}}$$

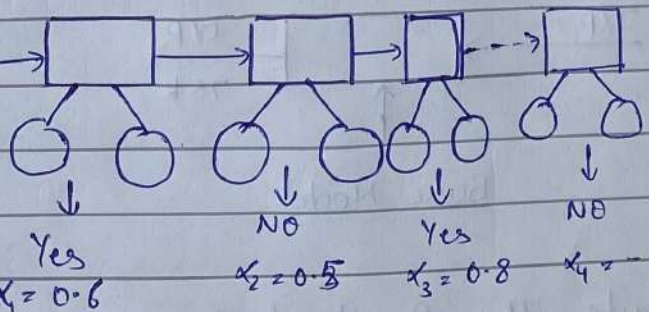
| Normalized Weight | Bins Assignment |
|---|---|
| 0.136 | 0 – 0.1 |
| 0.136 | 0.1 – 0.2 |
| 0.136 | 0.2 – 0.3 |
| 0.455 | 0.3 – 0.8 → Biggest Bin Size (wrongly predicted data point) |
| 0.136 | 0.8 – 0.9 |

Note : Now, algorithm will run and iteration process in which it select random numbers between 0 and 1. Most of the numbers will lie inside biggest bin. Hence, the wrongly predicted data point will be selected and send to next decision tree stump.

→ Prediction :

Test data ( ≤ 50k, ?)

f : Test data →




$f = 0.6(\text{Yes}) + 0.5(\text{NO}) + 0.8(\text{Yes}) + (-0.3)(\text{NO})$

$= 1.4(\text{Yes}) + 0.2(\text{NO})$ ⇒ Performance of say (Yes) = 1.4

Performance of say (NO) = 0.2

↓

Yes