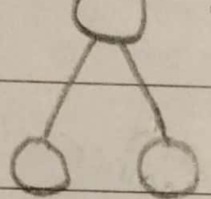
Boosting Algorithms:

→ We are adding sequential weak learner.

① Adaboost

Whenever we create a decision tree with just one depth.

This will lead to underfitting.



} Node with 2 leaves is called stump.

- → high bias [Training data ⇒ Accuracy ↓]
- → high/low variance [Test data ⇒ Accuracy ↓ or ↑]

Aim

1) high ~~variance~~ bias $\xrightarrow{\text{to}}$ Low ~~variance~~ bias.
2) Low/high variance $\xrightarrow{\text{to}}$ Low variance

} we are using boosting for this.

$\boxed{\text{Adaboost use stump}}$

Weak learner:

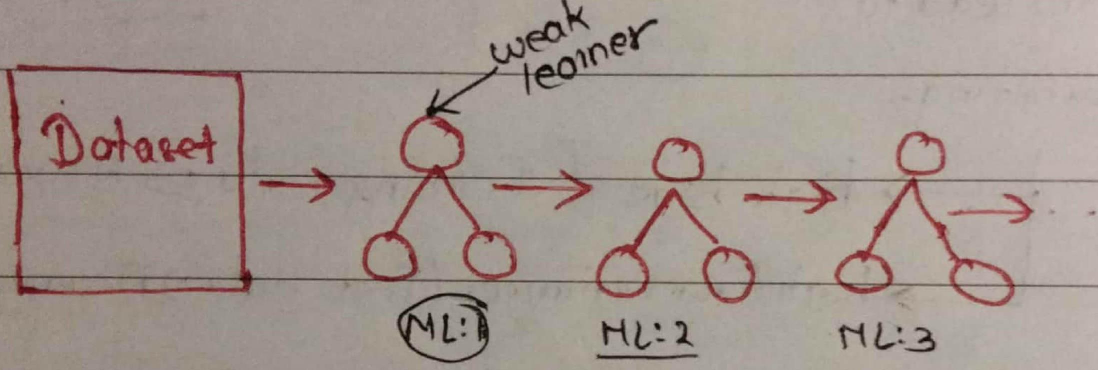
In Random Forest :- Majority Voting classifier [classification]

:- Average of O/P [Regression]

Memo No. ______
Date / /

In Adaboost:

Dataset are passed to the stump which is weak learner. Since, one can't predict correctly, it push those incorrectly predicted dataset to another weak learner and it goes on.



let us consider, we passed 100 datapoints to ML1, It predicted 80 datapoints correctly and 20 are wrongly predicted.

Now, these 20 points + other dataset are passed into ML2. lets say 10 points predicted correctly, and rest wrongly predicted are passed into ML3 and goes on.

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

Memo No. ____

Date / /

α is weight of model:

1) High value of α $\Rightarrow$ depicts high responsibility of model

2) -ve value $\Rightarrow$ depict model is not responsible

$M_1, M_2 \cdots M_n \longrightarrow$ weak learners

$\alpha_1 \alpha_2 \alpha_3 \cdots \alpha_n \longrightarrow$ weight / score

B - Bad

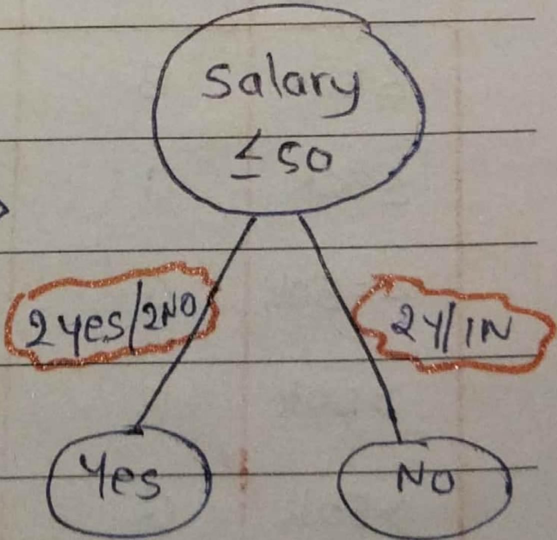
G - Good

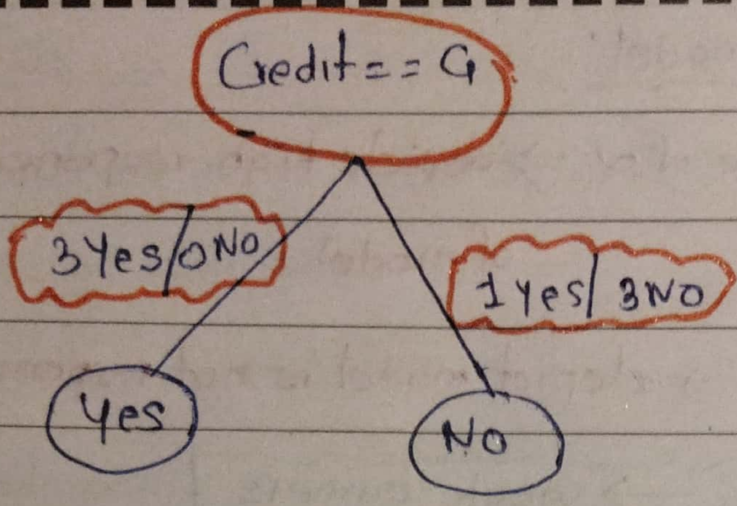
N - Normal

Example:

Salary	Credit	Approval
<=50K	B	No
<=50K	G	Yes
<=50K	G	Yes
>50K	B	No
>50K	G	Yes
<=50 K	N	No
>50K	N	Yes

$\Rightarrow$





Step 1: We create decision Tree Stamp by selecting the best one on the basis of entropy or Gini.

Step: 2 Assigning weight

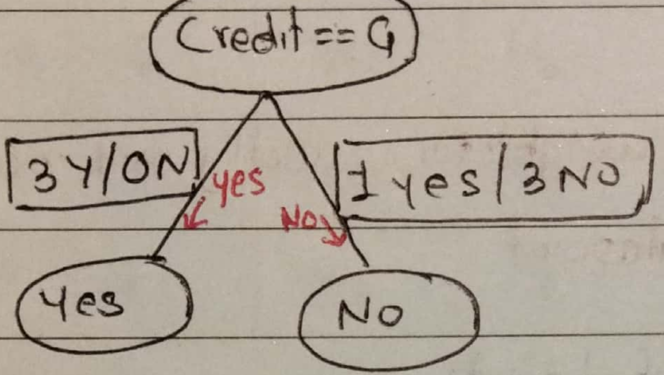
Salary	Credit	Approval	weight
≤ 50k	B	No	1/7
≤ 50k	G	yes	1/7
<= 50k	G	yes	1/7
> 50k	B	No	1/7
> 50k	G	yes	1/7
> 50k	N	yes	1/7
≤ 50k	N	No	1/7

Mo Tu We Th Fr Sa Su

Memo No. ______

Date / /

Here, split on the basis of credit == G was better since we had got a clean split on yes side.



Here, when we have Credit == G, yes is final verdict. When we don't have credit == G, No is verdict, still we have 1 point which is wrongly predicted based on this. So, this weak learner model has error of 1/7 $\therefore$ TE = 1/7 (Addition of weight of wrong data point)

* We have to give the wrongly predicted data to another weak learner.

Step 3: Calculating performance of stump:

$$\text{performance score} = \frac{1}{2} \ln \left[\frac{1 - T.E}{T.E}\right]$$

$$= \frac{1}{2} \ln [6]$$

$$= 0.896 \rightarrow \text{Good Performance}$$

Memo No. ______

Date / /

$$f = \alpha_1(m_1) + \alpha_2(m_2) + \alpha_3(m_3) + \dots + \alpha_n(m_n)$$

$$\boxed{\alpha_1 = 0.896}$$

$$\therefore f = 0.896(m_1) + \dots$$

Step 4: Update the weight for correctly and incorrectly predicted datapoint

for correctly classified point:

$$= \text{weight} * e^{-\text{Performance}}$$

$$= 1/7 * e^{-(0.896)} = 0.058$$

for incorrectly classified point

$$= \text{weight} * e^{\text{Performance}}$$

$$= \frac{1}{7} \times e^{0.896} = 0.349$$

Here, all the datapoints, which were correctly predicted, those weight has decreased and incorrectly predicted datapoint's weight has been increased.

Memo No. ______

Date / /

Salary	Credit	Approval	Weight	Updated W.
<=50K	B	No	1/7	0.058
<=50K	G	Yes	1/7	0.058
<=50K	G	Yes	1/7	0.058
>50K	B	No	1/7	0.058
>50K	G	Yes	1/7	0.058
>50K	N	Yes	1/7	0.349
<=50K	N	No	1/7	0.058

wrongly predicted Hence weight increased

Step 5: Normalize weight and assign bins

- After this step, we have push wrongly predicted data-points to another model. Hence, it is very important step. We have make the total summation of the updated weight to 1. If we add the updated weight we will get 0.697.

$$\text{Normalized weight} = \frac{\text{updated weight}}{\text{sum of updated weight}}$$

$\frac{0.058}{0.697} = 0.08$

Memo No.

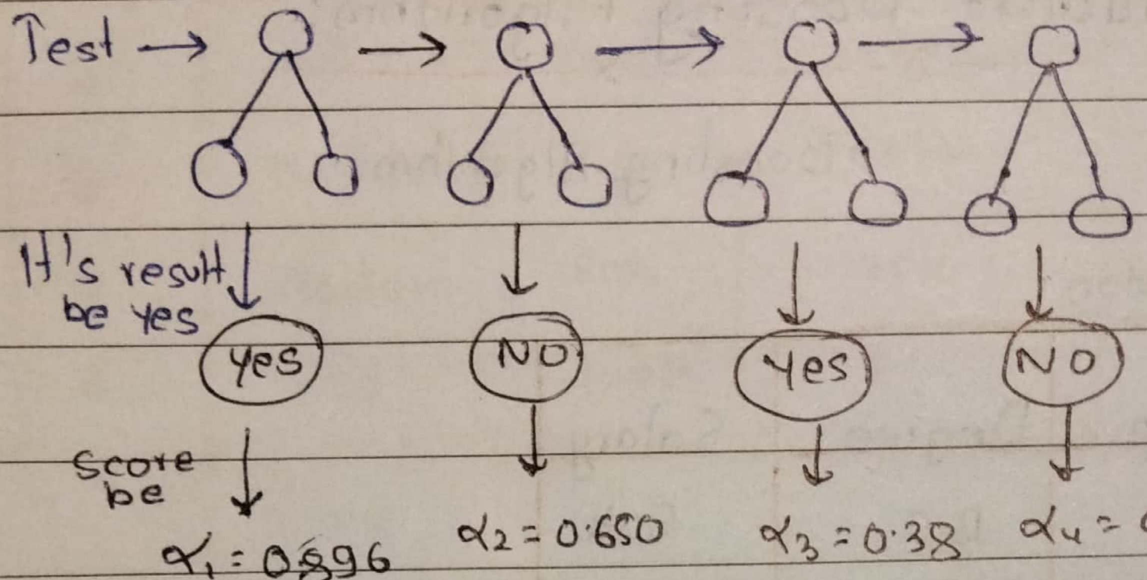
Date / /

Salary	Credit	Approval	Update weight.	Normalize weight	Bin Assign.
≤50k	B	No	0.058	0.08	0-0.08
≤50k	G	Yes	0.058	0.08	0.08 -0.16
≤50k	G	Yes	0.058	0.08	0.16 -0.24
>50k	B	NO	0.058	0.08	0.24-0.32
>50k	G	Yes	0.088	0.08	0.32 -0.4
>50k	N	Yes	0.349	0.50	0.4 -0.90
≤50k	N	No	0.058	0.08	0.90 - 1
			0.697	≅1	

Since, the bin for the incorrectly predicted is high, so, the next model, the chances of data being selected from data that is wrongly predicted. and it continue in a same steps in another model.

Final Prediction:

Suppose, we are using 4 decision tree for the some problem. So,



$f: \alpha_1(M_1) + \alpha_2(M_2) + \alpha_3(M_3) + \alpha_4(M_4)$

$= 0.0896(\text{yes}) + 0.650(\text{No}) + 0.38(\text{Yes}) + 0.20(\text{No})$

$= \boxed{1.2(\text{Yes}) + 0.85(\text{No})}$

Since, the weight of yes is more, so, the final classification is "Yes".