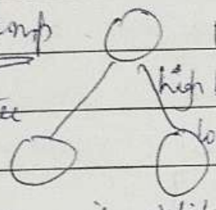
Boosting Algorithm

↳ Sequential weak learners

① Adaboost

Stump  Underfitting

high bias → Training Data Acc ↓↓ 40%

low Var → Test Data Acc ↑↑ 45%

~~simplest form of decision tree~~

Stump → Decision tree is split to one level or one split.

Aim

High Bias $\xrightarrow{\text{To}}$ low Bias

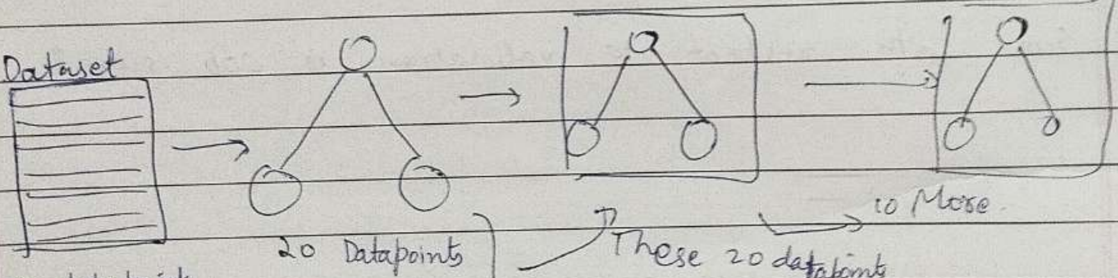
Low/High Var $\xrightarrow{\text{To}}$ low Variance

Weak Learners → Weak learners are added sequentially.

~~In case~~ of Random forest → Majority Voting Classifier [Classification]

Average of o/p [Regression]

Adaboost → Weak learners → Add the o/p of Weak learners with some weights assigned to it.



100 data points

20 Datapoints have wrongly predicted → These 20 datapoints along with some other will go to it.

10 More

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

α_1 ↓ Weight.

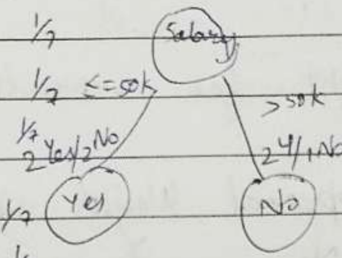
→ If weight is ↑↑↑ importance of model is ~~true~~ high.
→ If wt. is -ve the model is not doing anything.

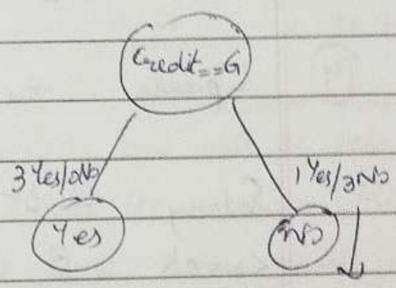
$M_1, M_2, M_3 \ldots M_n \rightarrow$ Weak Learners
$\alpha_1, \alpha_2, \alpha_3 \ldots \alpha_n \rightarrow$ Weights.

Salary	Credit	Approval	wt.
<= 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

G = Good
B = Bad
N = Normal

(1) Create Decision Tree Stump





This includes Normal also

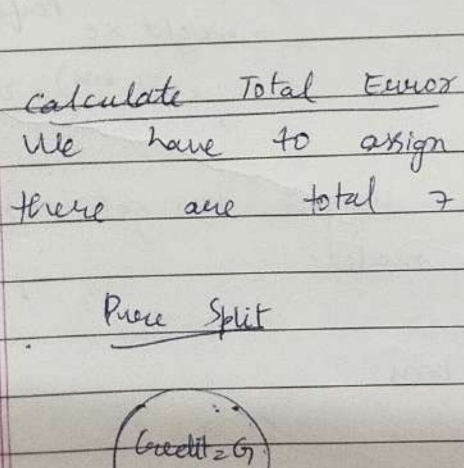
(1) To determine best weak learner we use Entropy or Gini impurity

$H(s) = -P_+ \log_2 P_+ - P_- \log_2 P_-$

- This decision tree stump is made using best one using Entropy or Gini impurity

(2) Calculate Total Error
We have to assign weights → Initial assigning would be 1/7 since there are total 7 data points.

Pure Split



→ 1 Yes is a wrong prediction

Total Error = $\frac{1}{7}$

(3) Performance of Stump $\rightarrow$ $= \frac{1}{2} \ln \left[\frac{1-TE}{TE} \right] = \frac{1}{2} \ln \left[\frac{1-\frac{1}{7}}{1/7} \right]$

$= \underline{0.896}$

function

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

$\alpha_1 = \underline{0.896}$

(4) Update the weight for correctly and incorrectly data points

Salary	Credit	Approval	Weights	Updated weights
<=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

for correctly classified point

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

$= \frac{1}{7} \times e^{-(0.896)}$

$= \underline{0.058}$

for incorrect classified points

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

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

$\Rightarrow$ All the incorrect data points weight has got increased. This is to pass it to next model.

(5) Normalize weights and assign bins.

Bin Assignment

Salary	Credit	Approval	Weights	Update wt.	Normalized wt	
<=50k	B	No	1/7	→ 0.058÷0.697	0.08	0 – 0.08
<=50k	G	Yes	1/7	0.058÷0.697	0.08	0.08 – 0.16
<=50k	G	Yes	1/7	0.058÷0.697	0.08	0.16 – 0.24
>50k	B	No	1/7	0.058	0.08	0.24 – 0.32
>50k	G	Yes	1/7	0.058	0.08	0.32 – 0.40
>50k	N	Yes	1/7	0.349 ~~0.0~~	0.50	0.40 – 0.90
<=50k	N	No	1/7	0.058	0.08	0.90 – 1
				0.697	≈ 1	

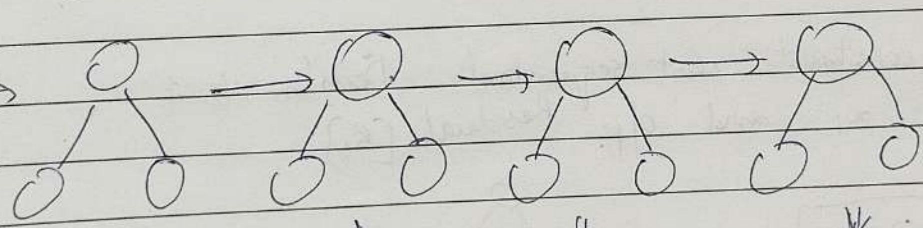
Random bins will be selected (random value)

⇒ Our aim is to pass this wrong record to the Decision Tree again & again to train. Along with that some other value can also get selected.

* Final Prediction

Test ($\leq$ 50k, G)

Test →



⇓ Yes ⇓ No ⇓ Yes ⇓ No

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

$$= 0.896\,(Yes) + 0.650\,(No) + 0.38\,(Yes) + 0.20\,(No)$$

$$= 1.2\,(Yes) + 0.85\,(No)$$

⇓

Because there is more weightage on Yes, the final result will be Yes.

Gradient Boosting Algorithm

Regression

x_1 Exp.	x_2 Degree	o/p Salary	→ Predicted $\hat{y}$	Residual $(y-\hat{y})$ R1	R_2	→ o/p of decision tree $\hat{y}$	R_3	R_4 $\hat{y}$
2	B.E	50k	75k	-25k	-23	72.7	-22.7	
3	Masters	70k	75k	-5k	-3	74.7 → Based on α (0.1)	-4.7	
5	Master	80k	75k	5k	3	75.3	4.7	
6	PHD	100k	75k	25k	20	77	23	
		75k Avg.						

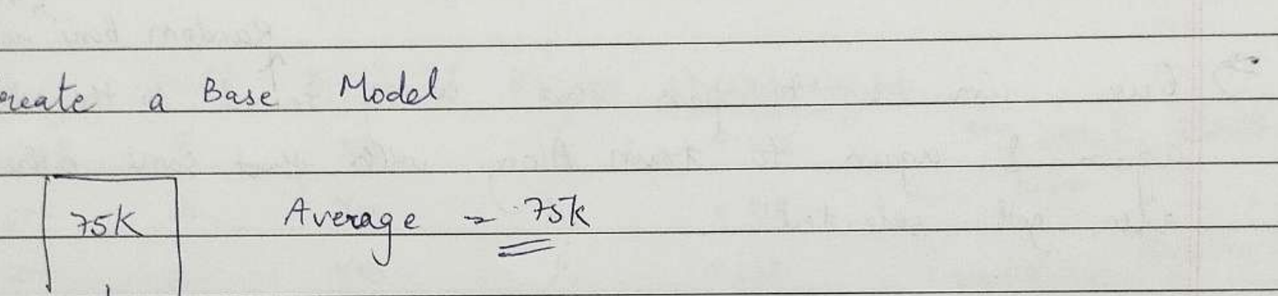
Step-2. Create a Base Model

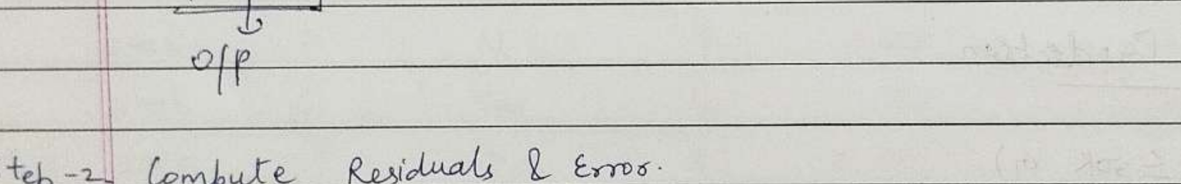
$\boxed{75k}$ → o/p

Average = 75k

Step-2 Compute Residuals & Error.

Step 3 - We construct next sequential Decision tree
i/p x_i and o/p Residuals $[R_1]$





Predicted $= 75 + \alpha(-23)$

$= 75 + 0.1 \times -23$

$= 75 - 2.3$

$\alpha \rightarrow$ learning Rate

$\alpha \rightarrow 0$ to 1

let $\alpha = 0.1$

$\Rightarrow$ R_2 is assumption & o/p of decision tree. Since Dependent feature with which decision tree is made is R_1, R_2 will be close to it.

$\Rightarrow$ This loop will go on and on until the predicted value keeps on decreasing and error is coming towards 0.

$\Rightarrow$ By default no. of DT is 100.

Final function

$$F(x) = \underset{\text{Base learner}}{\alpha_0 h_0(x)} + \underbrace{\alpha_1}_{\text{learning Rate}} \overbrace{(h_1(x))}^{\text{Model 1 }(M_1)} + \alpha_2 (h_2(x)) + \dots \alpha_n(h_n(x))$$

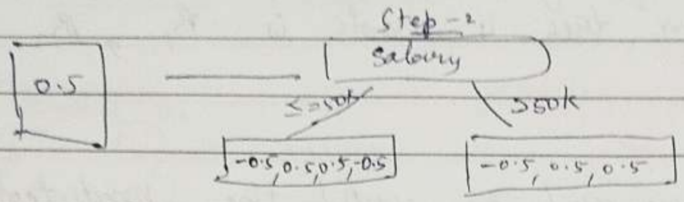
$$\boxed{F(x) = \sum_{i=0}^{n} \alpha_i h_i(x)}$$

* xg boost Classifier (Extreme Gradient Boost)

Dataset

Salary	Credit	Approval	R_1	$\hat{y}$	R_2
<= 50k	B	0	-0.5	0.52	-0.52
<= 50k	G	1	0.5	0.58	0.42
<= 50k	G	1	0.5	0.58	0.42
> 50k	B	0	-0.5	-	-
> 50k	G	1	0.5	-	-
> 50k	N	1	0.5	-	-
<= 50k	N	0	-0.5	-	-

Step 2. Base Model



Step 3 → Calculate similarity weight $= \dfrac{(\sum \text{Residuals})^2}{\sum Pr(1-Pr)}$

$Pr \rightarrow$ Probability of base learner (Base learner O/P)

Similarity weight (left node) $= \dfrac{(-0.5+0.5+0.5-0.5)^2}{0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)} = 0$

S.W (Right Node) $= \dfrac{(0.5)^2}{0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)}$

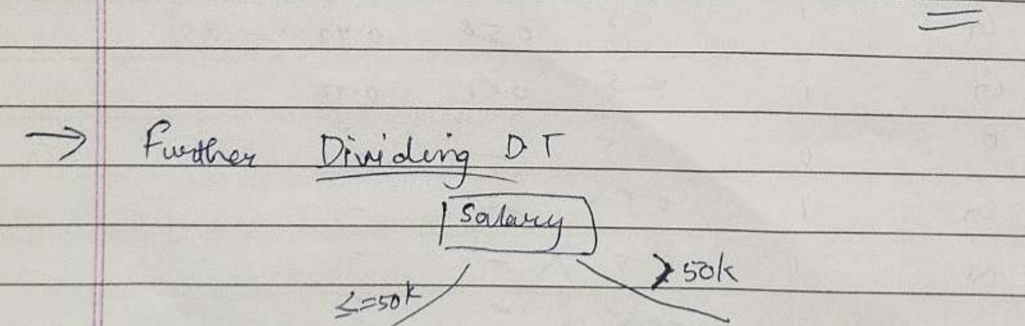
$= \dfrac{0.25}{3 \times 0.25} = \dfrac{1}{3} = 0.33$

S.W (Root Node) $= \dfrac{(0.5)^2}{7 \times 0.25} = \dfrac{1}{7} = 0.142$

Calculate Gain = SW(left) + S.W(Right) − S.W(Root Node)

$= 0 + 0.33 - 0.142 = 0.19$

→ Further Dividing DT



→ ~~S.W~~ Now Calculate similarity ~~Gains~~ weights

Final O/p →

$$\rightarrow \boxed{0.5} \rightarrow$$

↓

O/p of base model

$\log(\text{odds}) = \log\left(\frac{p}{1-p}\right)$ → log loss of logistic Regression.

$= \log\left(\frac{0.5}{0.5}\right)$

$= \log 1 = \underline{\underline{0}}$

Model O/p $= \sigma\left[0 + \alpha(\underset{\downarrow}{1})\right]$

1 is SW of Node as per the data point.

→ considering α b/w $[0 \text{ to } 1]$ & let $\alpha = 0.1$ → learning Rate

Model O/p $= \sigma(0.1) = \frac{1}{1+e^{-0.1}} = \underline{\underline{0.52}}$ → P.T.A

→ for second pt., O/p $= \sigma(0 + 0.1 \times 0.33) = \sigma(0.033) = 0.58$

→ After getting $\hat{y}$ calculate R^2, then next decision tree will be based on x_i & Residual R_2.

$$\boxed{\text{O/p} = \sigma\left(\text{Base learner} + \alpha_1(DT_1) + \alpha_2(DT_2) + \ldots \alpha_n(DT_n)\right)}$$

Xgboost Regressor

Exp	Gap	Salary	Res 1	O/P
2	Yes	40k	-11k	46
2.5	Yes	42k	~~-10k~~ -9k	46.
3	No	52k	1k	53.5
4	No	60k	9k	53.5
4.5	Yes	62k	11k	
		Avg 51k		

Similarity wt. = $\dfrac{(\sum \text{Residual})^2}{\sum Pr(1-Prob)}$

$\boxed{\text{Base Model}}$

↳ 51k

⟹ In XG boost we create binary Trees only.

[-11, -9, 1, 9, 11]

$\boxed{\text{Exp}}$

<=2 : [-11] — S.W = 60.5

>2 → first pt. is 2. : [-9, 1, 9, 11] — SW = 28.5

⟹ Similarity weight = $\dfrac{\sum(\text{Residuals})^2}{\text{No. of Residuals} + \lambda}$

λ → Hyperparameter.

$SW = \dfrac{121}{1+1} = \dfrac{121}{2} = 60.5$ → let $\lambda = 1$

$S.W(\text{Right Node}) = \dfrac{(-9+1+9+11)^2}{4+1} = \dfrac{144}{5} = 28.5$

$S.W(\text{root}) = \frac{1^2}{5+1} = \frac{1}{6} = 0.16$

→ Calculate Information Gain / Gain = SW(right) + SW(left) − S.W(root)

$= 65.5 + 28.5 - 0.16 = 93.84$

* Go to 2nd record.

[-11, -9, 1, 9, 11]

Exp

<=2.5 / \ >2.5

[-11, -9] [1, 9, 11]

$S.W = \frac{400}{3} = 133.33$ $SW = \frac{441}{3+1} = \frac{441}{4} = 110.25$

$SW(\text{root Node}) = \frac{1}{5+1} = \frac{1}{6} = 0.16$

$\text{Gain} = 133.33 + 110.25 - 0.16 = 243.42$

* By checking this Gain whichever has high split we will split from there.

⇓ Now let's compare

[-11, -9, 1, 9, 11]

Exp

<=2.5 / \ >2.5

[-11, -9] [1, 9, 11]

Grape (Yes) / \ No

$o/p = \text{Avg} = \frac{20}{2} = -10$

o/p = 11 ← [11] [1, 9] → o/p = 5

→ Here we will calculate S.W of this 11 & [1, 9].

f = Base Model $+ \alpha$ [O/p of Tree]

Lets consider first pt. [2 Yes 40k -11]

Base Model o/p is 51k

$51 + (0.5)\left[\frac{-20}{2}\right]$

α — o/p from node

⇒ like this we can create Multiple DT w.r.t to Grap etc.

$= 51 + (0.5)\left(\frac{-20}{2}\right)$

$= 51 - 5 = 46k$

Predicted o/p is 46k

* for 3rd point $= 51 + 0.5(5)$

$= 53.5$

$$\boxed{F = \text{Base Model} + \alpha_1(T_1) + \alpha_2(T_2) + \ldots \alpha_n(T_n)}$$

⇒ There is a hyperparameter γ = 150.5 → let it be

Gamma

let I.G = 140, ∴ ⇒ 140 − 150.5 → −ve value

If value is −ve [Post Prune it]

If +ve [we should not proon it]