20-NOV-2022

Gradient Boosting Algorithm

Gradient Boosting Regressor

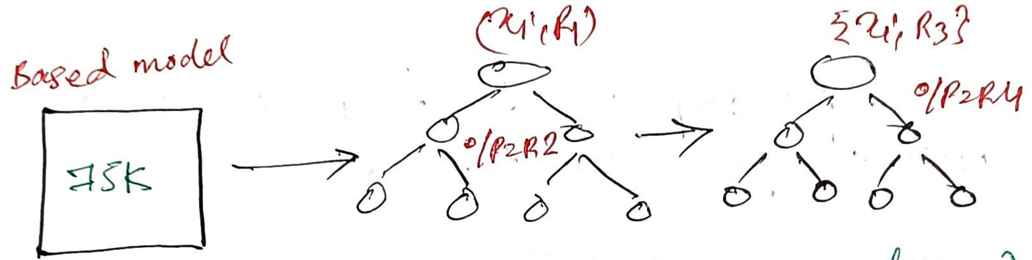
Regression

output of decision trees
(values are assumed)

Exp	Degree	O/P (y) Salary	$\hat{y}$	R1 (Residual) ($y - \hat{y}$)	R2	$\hat{y}$ (updated)
2	B.E	50K	75K	-25	-23	72.7
3	Masters	70K	75K	-5	-3	74.7
5	Masters	80K	75K	-5	3	75.3
6	PhD	100K	75K	25	20	77
Avg		75K				

Step 1: Create a base model

-



Avg = (50+70+80+100)/4
= 75K

75+(-23) = 52K (overfitting)
because based on two values, values are close to actual output.

Step 2: Compute the residuals or errors

Step 3: Construct the next sequential decision trees with input x_i and o/p is Residuals (R1).
- Complete decision trees

- Let's take R2 is the output of decision trees.
- Predicted out will be multiplied by 'α'.

$$\boxed{\text{Predicted} = 75 + \alpha(-23)}, \quad \left\{ \begin{array}{l} \alpha = \text{learning rate} \\ 0 \le \alpha \le 1 \end{array} \right\}$$

1st Record

Let's $\alpha = 0.1$

Predicted = 75 + 0.1(-23) = 72.7

2nd Record

Predicted = 75 + 0.1(-3) = 74.7

Like this we will find for all record.

Now, calculate R_3 = diff. b/w actual - $\hat{y}$ (updated).

R_3
-22.7
-4.7
4.7
23

Again, we will take R_3 as output and construct next decision tree

R4 (output of 2nd decision tree)	$\hat{y}$
—	—
—	—
—	—
—	—

Like this, we will keep constructing decision trees till the Residual gets reduced.

→ Residual should decrease.

Final Function → Base Learner

$$\boxed{F(x) = \alpha_0 h_0(x) + \alpha_1 (h_1(x)) + \alpha_2 (h_2(x)) + \alpha_3 (h_3(x)) + \dots + \alpha_n (h_n(x))}$$

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

where α = learning rate

Xgboost Classifier (Extreme Gradient Boost)

Dataset

Ned's dataset [illegible] data

Salary	Credit	Approval	(Approval − 0.5) R1	$\hat{y}$	$(y - \hat{y})$ $(0.5 - \hat{y})$ R2
<=50K	B	0	−0.5	0.52	−0.48 −0.52
<=50K	G	1	0.5	0.58	0.42
<=50K	G	1	0.5	0.58	
>50K	B	0	−0.5		
>50K	G	1	0.5		
>50K	N	1	0.5		
<=50K	N	0	−0.5		

Step 1: Create Base Model

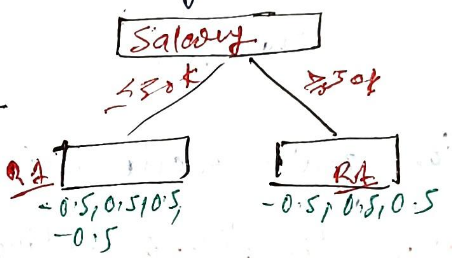
Base Model

$\boxed{0.5}$

Binary classification

output 0

Creating Decision Trees



→ Entire decision trees gets created in Gradient Boost

Step 2: Create Decision Trees

Step 3: Calculate Similarity Weight

$$\boxed{\text{Similarity Weight} = \frac{\sum (\text{Residuals})^2}{\sum P_r(1 - P_r)}}$$

where P_r = Probability w.r.t based model (Base learner output)

$$S.W(\leq 50K) = \frac{(-0.5 + 0.5 + 0.5 - 0.5)^2}{0.5(1-0.5) + 0.5(1-0.5) + 0.5(1-0.5) + 0.5(1-0.5)}$$

(Left Node)

$$= 0$$

$$S.W(> 50K) = \frac{(-0.5 + 0.5 + 0.5)^2}{0.5(1-0.5) + 0.5(1-0.5) + 0.5(1-0.5)}$$

(Right Node)

$$= 0.33$$

$$S.W = \frac{(-0.5+0.5+0.5-0.5-0.5+0.5+0.5)}{0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)}$$

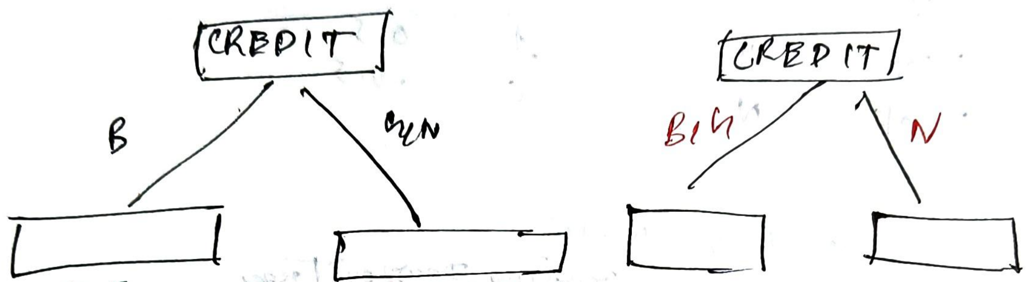
(Root Node)

$= 0.142$

Calculate Gain

Gain $= 0 + 0.33 - 0.142 = \cancel{0.21}\ 0.19$

Build next root,



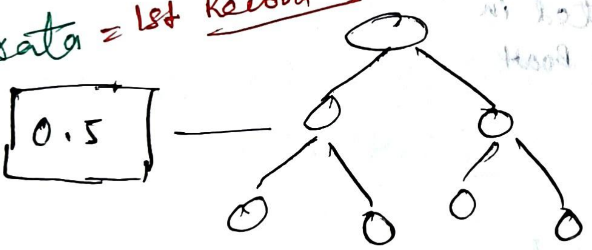
-0.5 $\quad$ $0.5, 0.5, -0.5$

$\boxed{SW = 1}$ $\quad$ $SW = 0.33$

→ Highest information gain node will get selected.

Final output $\quad$ Binary Classification → Linear Regression (log loss)

Test data = 1st Record



→ O/P should be b/w 0 to 1 in binary classification.

$$\boxed{\text{Log (odds)} = \log\left(\frac{P}{1-P}\right)}$$

$\log(0.5) = \log\left(\frac{0.5}{1-0.5}\right)$

$= \log(1)$

$= 0$

$\left\{ \text{Sigmoid fn} = \frac{1}{1+e^{-z}} \right\}$ → Binary classification.

~~Model output~~

$$\text{Model output} = \sigma\{0 + \alpha(S.W)\} = \frac{1}{1+e^{0.10}} = 0.52$$

$\cancel{= 0 + \alpha(1)} = \sigma\{0+\alpha(1)\}$

Sigmoid ~~function~~. $\quad$ Similarity Weight

2nd Record

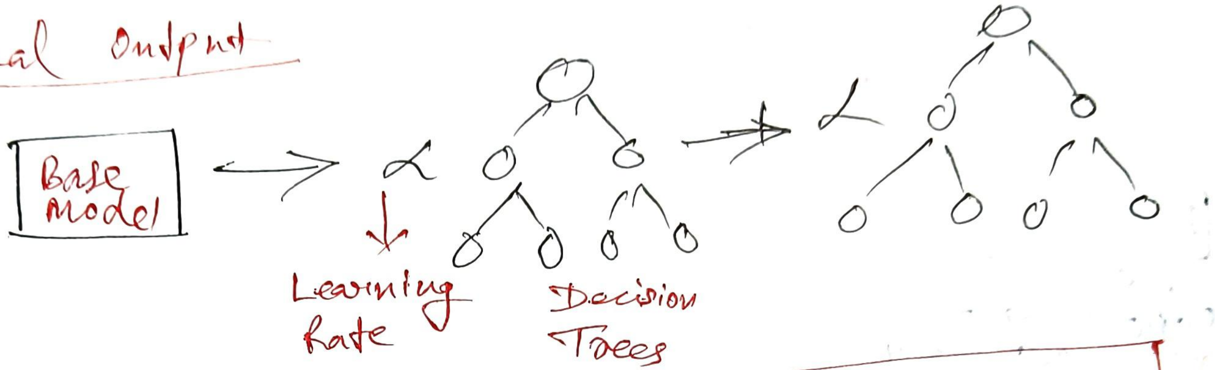
$$\text{Model output} = \sigma(0 + 0.1(0.33)) = 0.58$$
$$= \frac{1}{1+e^{-(0+0.1(0.33))}}$$

3rd Record.

$$\text{Model Output} = \sigma(0 + 0.1(0.33)) = 0.58$$
$$= \frac{1}{1+e^{-(0+0.1(0.33))}}$$

→ After this again, compute R2 and again create next decision trees and so on till we get less residual.

Final Output



$$\boxed{O/P = \sigma(\text{Base learner} + \alpha_1(DT_1) + \alpha_2(DT_2) + \dots \alpha_n(DT_n))}$$

↓ use log(odds) for base learner

→ In a binary classification, take default threshold as 0.5 but we can create our own threshold as well.

→ for Multi classification, we use softmax activation function.

→ for Binary classification, we use sigmoid activation function.

XgBoost Regressor

- Sequential Decision Trees.

Dataset

Exp	Gap	(y) Salary	(y − avg) (Residuals) R1	(DT 1) o/p	(Residuals) R2	(DT 2) o/p	R3	… so on	(DT n)
2	Yes	40K	−11K	46					
2.5	Yes	42K	~~−10K~~ −9K	46					
3	No	52K	−12 K	53.5					
4	No	60K	8K	62					
4.5	Yes	62K	12K	63					

Avg = 51K

Base Model

↓

output = avg of all salary
= 51K

1st Record:

Step 1:

Decision Trees

$\{-11, -9, 1, 9, 11\}$ $\left\{ S.W = \frac{(-11-9+1+9+11)^2}{5+1} = 0.16 \right\}$

$\boxed{\text{Exp}}$ → Root Node

$\boxed{\text{Gain} = 93.84}$

≤ 2 → $\boxed{-11}$ $\left\{ S.W = \frac{121}{1+1} = 65.5 \right\}$

> 2 → $\boxed{-9, 1, 9, 11}$ $\left\{ S.W = \frac{(-9+1+9+11)^2}{4+1} = \frac{144}{5} = 28.5 \right\}$

Step 2: Find Similarity weight. (To find out the root node)

$$\boxed{\text{Similarity Weight} = \frac{\Sigma(\text{Residuals})^2}{\text{No. of Residuals} + \lambda}}$$

λ = hyperparameter

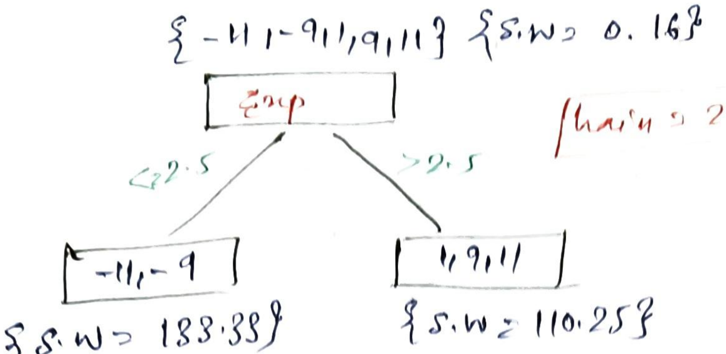
~~Here~~ Here, ~~S.W =~~

~~Assume~~, $\lambda = 1$

Step 3: Calculate Information Gain,

$\{$ Information Gain $= 65.5 + 28.5 - 0.16 = 93.84 \}$

2nd Record



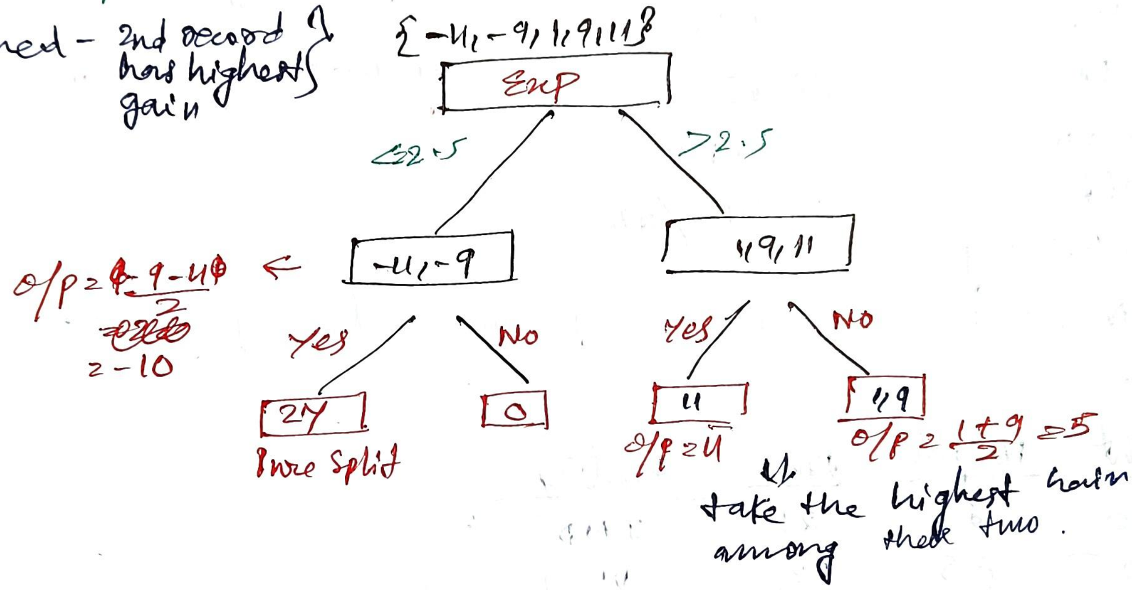
[Gain = 243.42]

Information Gain = 183.33 + 110.25 − 0.16 = 243.42

> Similarly, we will find gain for all the records and will take the highest gain record.

> After exp, we will do the same for gap.

{Assumed - 2nd record has highest gain}



> After the base model, record will be passed to base model.

Model Output

$o/p = 51 + \alpha(-20)$ ~~= 51 + 0.5(20)~~ = 50 + 0.5(-10) = 46 (o/p after 1st iterations)

w.r.t one DT, like that we can create many DTs.

α = learning rate parameter

For multiple DTs,

$$\boxed{o/p = \text{base model output} + \alpha_1(DT_1) + \alpha_2(DT_2) + \alpha_3(DT_3) + \dots + \alpha_n(DT_n)}$$

Gamma: $\boxed{\gamma = \text{gamma} - \text{Information Gain}}$

if γ = +ve, no pruning

if γ = -ve, post pruning (cut the tree)