⇒ Xgboost Classification:

Let we have a dataset

Salary (Independent)	Credit (Independent)	Y(Approval) (Dependent)	$\hat{y}$ (Base model output)	R_1 ($y-\hat{y}$)
<=50K	B	0	0.5	-0.5
<=50K	G	1	0.5	0.5
<=50K	G	1	0.5	0.5
>50K	B	0	0.5	-0.5
>50K	G	1	0.5	0.5
>50K	N	1	0.5	0.5
<=50K	N	0	0.5	-0.5

Steps

1. Construct a base model
2. Construct a Decision Tree with root
3. Calculate similarity weight

$$= \frac{\Sigma (\text{residual})^2}{\Sigma Pr(1-Pr)+\lambda}$$

4. Calculate gain

Steps Implementation

1. Construct a base model. It's probability should be 0.5 as two categories and it is unbiased

Base Model

⇓

Pr=0.5

2. Construct a decision tree with root using inputs (salary credit) and output R1

For now let's consider salary is best feature to construct DT

[−0.5, 0.5, 0.5, −0.5, 0.5, 0.5, −0.5]

⇓

Pr=0.5

Salary → Similarity weight = 0.14

≤50K

>50K

−0.5, 0.5, 0.5, −0.5

LC

Similarity weight = 0

−0.5, 0.5, 0.5

RC

Similarity weight = 0.33

3. Calculate similarity weight

- For left child:

$$\text{Similarity weight}(LC) = \frac{\sum(\text{residual})^2}{\sum pr(1-pr)}$$

$$\text{Similarity weight}(LC) = \frac{[(-0.5+0.5+0.5-0.5)^2]}{[0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)]}$$

$$\boxed{\text{Similarity weight}(LC) = 0}$$

- For right child

$$\text{Similarity weight}(RC) = \frac{\sum(\text{residual})^2}{\sum pr(1-pr)}$$

$$= \frac{[(-0.5+0.5+0.5)^2]}{[0.5(1-0.5)+0.5(1-0.5)+(1-0.5)]}$$

$$\boxed{\text{Similarity weight}(RC) = 0.33}$$

- For root:

Calculate similarly for root

$$\boxed{\text{Similarity weight}(root) = 0.14}$$

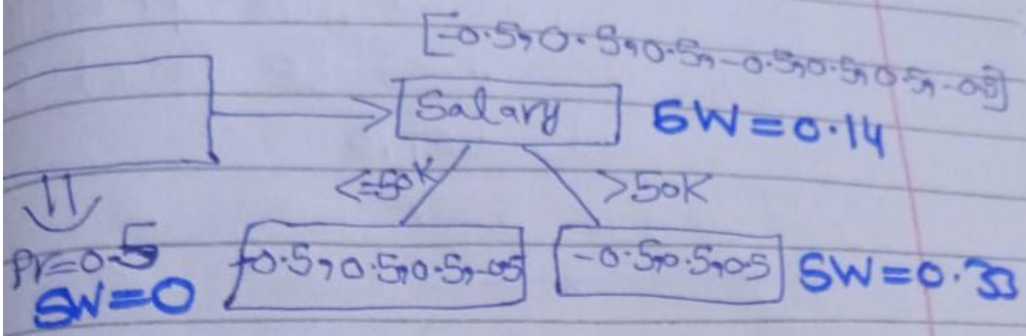
4. Calculate gain weight

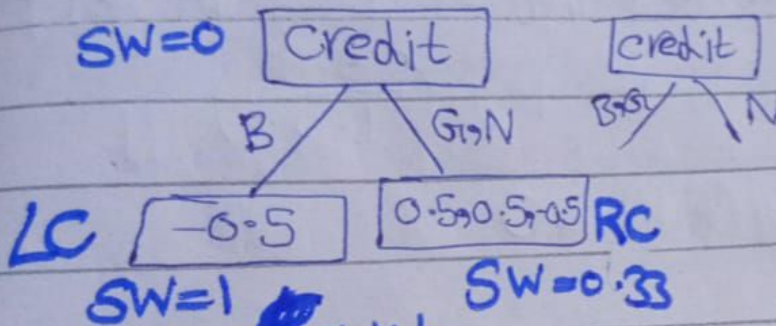
Gain = Similarity wt (LC) + Similarity weight (RC) − Similarity weigt (root)

Gain = 0 + 0.33 − 0.14

$$\boxed{\text{Gain} = 0.19}$$

Now let's increase our decision tree and add credit in it





For Left Split

Similarity (R) = 0

Similarity (LC) = 1

Similarity (RC) = 0.33

$\boxed{\text{Gain} = 1 + 0.33 - 0 = 1.33}$

Now we can calculate Gain similarly for right split and if it is greater than gain of left split then we will select right split otherwise left split

Let left split selected in our case

• Prediction for new test data:

As base model gives probability so our data must be passed from log of odds before

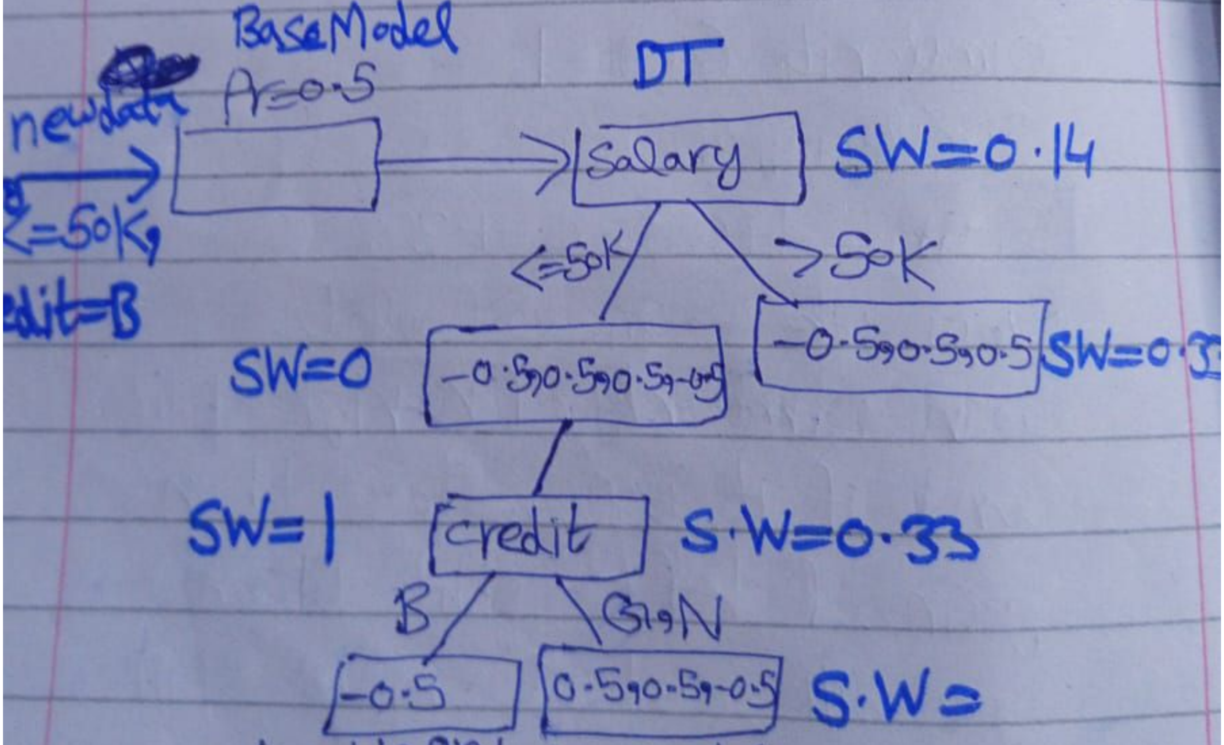
$$\boxed{\log(\text{odds}) = \log\left(\frac{P}{1-P}\right)}$$

As probability of base model pr=0.5

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

$$\boxed{\log(\text{odds}) = 0}$$

• Predicted output



logodds on base model

New data = $\sigma\,(0 + \alpha(1))$

Sigmoid activation Similarity weight

$\alpha \Rightarrow 0$ to $1 \Rightarrow$ Learning rate

Let $\alpha = 0.1$

Prediction of new data $= \sigma(0 + (0.1)(1)) = \sigma(0.1)$

$$= \frac{1}{1+e^{-0.1}} \qquad \therefore \sigma(z) = \frac{1}{1+e^{-z}}$$

Prediction of new data $= 0.52$

Similarly calculate for 2nd record salary<=50k and credit=G

Prediction) $= \sigma(0 + \alpha(0.33))$

Let $\alpha = 0.1$

Prediction $= \sigma(0 + (0.1)(0.33))$

$$= \sigma(0.033)$$

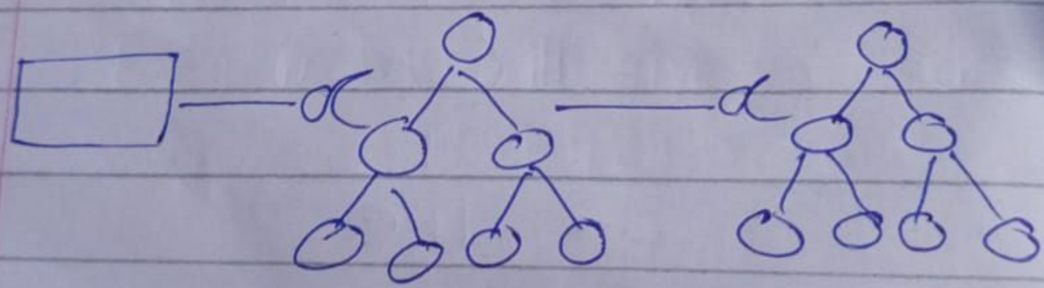
$$= \frac{1}{1+e^{-0.033}}$$

Prediction $= 0.508$

Let's put results in table:

Salary	credit	Approval	$\hat{y}$	R1	$\hat{y}$	R2
<=50K	B	0	0.5	-0.5	0.52	-0.52
<=50K	G	1	0.5	0.5	0.58	0.42
<=50K	G	1	0.5	0.5	—	—
>50K	B	0	0.5	-0.5	—	—
>50K	G	1	0.5	0.5	—	—
>50K	N	1	0.5	0.5	—	—
<=50K	N	0	0.5	-0.5	—	—

Now we will keep on making Decision Tree using salary, credit as input and R2 as output



$$Output = \sigma(Base Model + \alpha_1(DT1) + \alpha_2(DT2) + \alpha_3(DT3))$$

Let the output we get is 0.52
so as threshold = 0.50
and $0.52 > 0.50$
So output class will be
1

And in Similarity weight

$$\text{Similarity Weight} = \frac{\Sigma(\text{residual})^2}{\Sigma pr(1-pr)+\lambda}$$

λ → Hyperparameter

- There is always a thinking that till one we are going to have splitting we have to define restriction according to similarity weight for that we specify a **cover value**

cover value = Pr(1-pr)

cover value = 0.5(1-0.5)

$\boxed{\text{cover value} = 0.25}$

So, when similarity weight will be less than cover value we stop splitting.

⇒ Xg Boost Regressor:

Let we have a dataset

Exp (Independent)	Gap (Independent)	Salary(y) (Dependent)	$\hat{y}$ (Base model output)	R_1 ($\hat{y} - y$)	$\hat{y}$	R_2
2	Yes	40K	51K	-11	49.9	-9.9
2.5	Yes	42K	51K	-9	49.9	-7.9
3	No	52K	51K	1	51.5	0.5
4	No	60K	51K	9	51.5	8.5
4.5	Yes	62K	51K	11	52.5	9.9

Steps:

① Create a base model

Base model will give us average

$$\text{average} = \frac{40+42+52+60+62}{5}$$

average = 51K

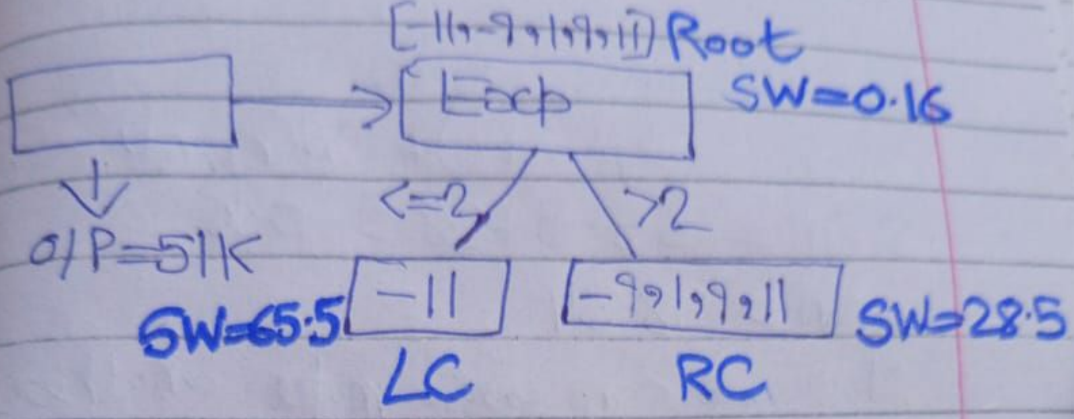
Base model

⇓

O/P = 51K

② Residual computation

(3) Now create Decision Tree using inputs Experience,gap and output R_1



(4) Calculate similarity weight

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

$\lambda \Rightarrow$ Hyperparameter

As λ increases similarity weight decreases

Let $\lambda = 1$

$$SW(LC) = \frac{(11)^2}{1+1} = \frac{121}{2}$$

$$\boxed{SW(LC) = 65.5}$$

$$SW(RC) = \frac{(-9+1+9+11)^2}{4+1} = \frac{144}{5}$$

$$\boxed{SW(RC) = 28.5}$$

$$SW(root) = \frac{(-11+9+1+9+11)^2}{5+1}$$

$$\boxed{SW(root) = 0.16}$$

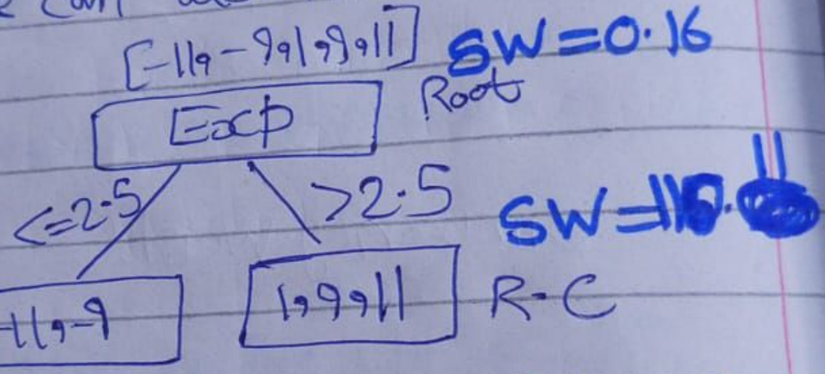
5) Calculate Gain

Gain = SW(left) + SW(right) − SW(root)

Gain = 65.5 + 28.5 − 0.16

Gain = 98.34

Now we can also make another split



S·W=133.33

Calculate similarity weight and gain

$$SW(L\cdot C) = \frac{\Sigma (\text{residual})^2}{\text{No. of residuals} + \lambda}$$

Let $\lambda = 1$

$$SW(L\cdot C) = \frac{(-11-9)^2}{2+1} = \frac{400}{3}$$

$$\boxed{SW(L\cdot C) = 133.33}$$

$SW(R\cdot C) = \frac{(1+9+11)^2}{3+1} = \frac{441}{4}$ 110.1

$SW(R\cdot C) =$ 110.11

$SW(Root) = (-11-9+1+9+11)^2$

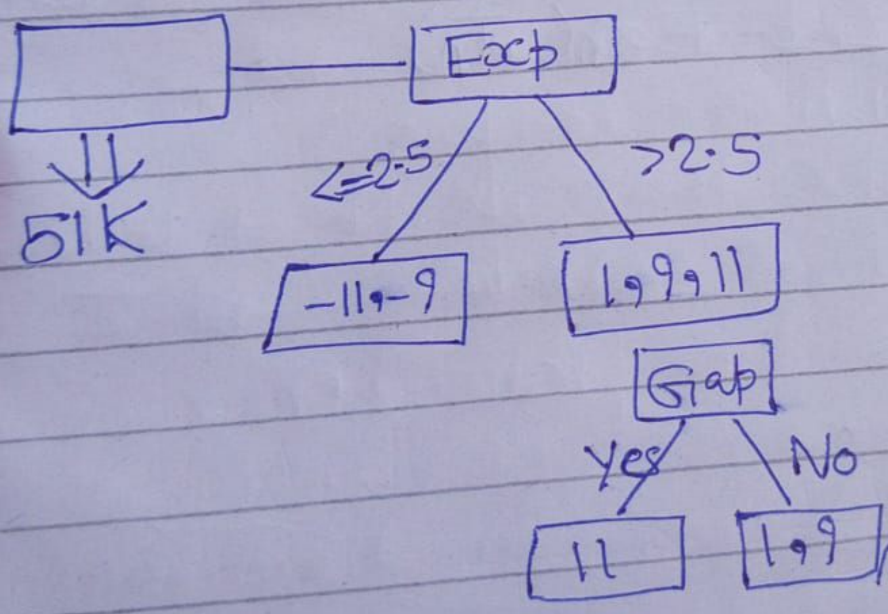
Calculate gain

Gain = 133.23 + 110.11 − 0.16

Gain = 243.18

As it's gain is higher so we will select this 2nd split

We can do further splitting using Gap



$Average = \frac{9+1}{2} = 5$

- **Final prediction:**

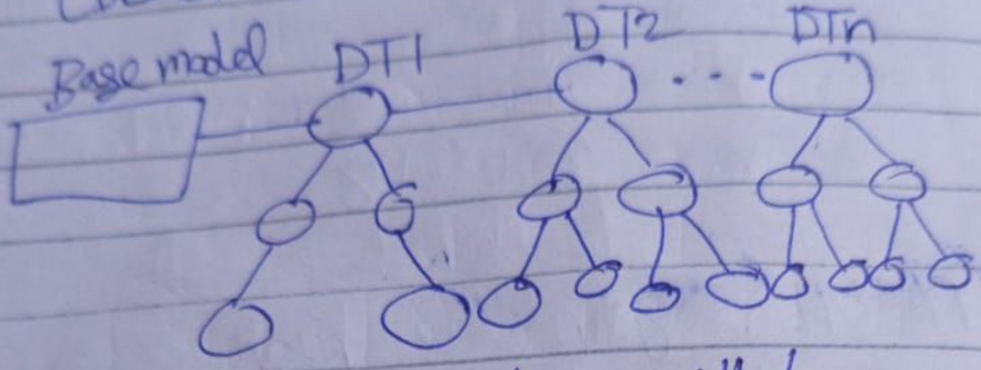
Input: exp=3 and gap=No

Output = 51 + 0.1(5)

51 → Base model; 0.1 → α ⇒ Learning rate (0-1)

$\boxed{\text{Output} = 51.5}$

Now after predicted outputs for all we calculate R2 and make next weak learner (Decision Tree) using output R2.



And the prediction will be

$$F = BaseModel + \alpha_1(DT_1) + \alpha_2(DT_2) + \cdots + \alpha_n(DT_n)$$