XGBoost

Xtream Gradient Boosting

It is also works for both problem, Regression and classification.

It is used to solve complex data problem.

It is implemented in c++ and provids interface for python and other language

Incorporate regularization L_1 & L_2 for overfitting

Support parallel tree construction to speed up training.

Handle missing data automatically

pros -

1. High performance
2. speed and scalability
3. Flexibility

cons -

1. Complexity
2. computational Resource
3. Not ideal for unstructue data

After 1st step

Sal.	Credit	Approved (Y)	pred. ($\hat{Y}_1$)	$Y - \hat{Y} = R$ (Residual)	$\hat{Y}_2$
<=50	B	0.	0.5	0-0.5 = -0.5	0
<=50	G	1	0.5	1-0.5 = 0.5	0
<=50	G	1	0.5	1-0.5 = 0.5	1
>50	B	0	0.5	0-0.5 = -0.5	0
>50	G	1	0.5	1-0.5 = 0.5	1
>50	N	1	0.5	1-0.5 = 0.5	0
<=50	N	0	0.5	0-0.5 = -0.5	1

Step-1 Base model

class 0 and 1

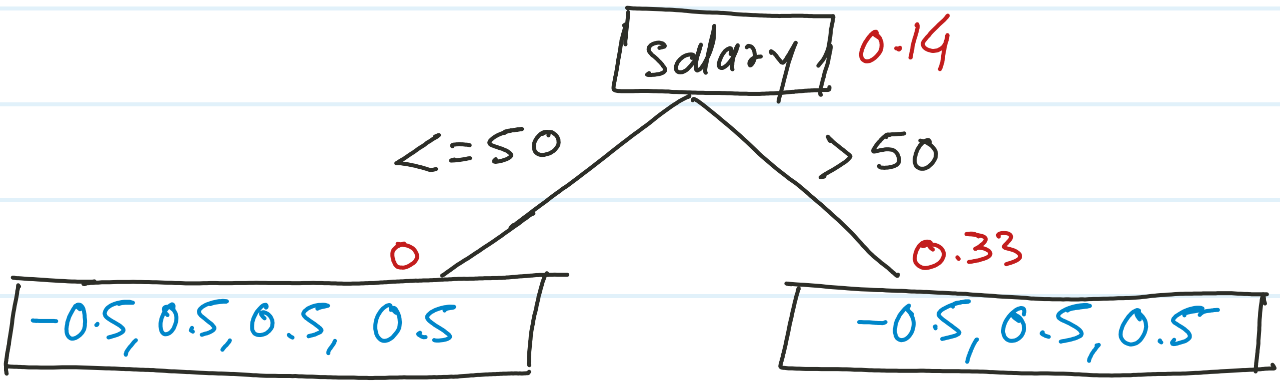
$$\frac{0+1}{2} = 0.5$$

It called probability 0.5

To calculate residual initially use probability

After getting residual construct a tree

[-0.5, 0.5, 0.5, -0.5, 0.5, 0.5, -0.5]



Doesn't matter how category will be there, it split or construct the tree with two leafe only mean binary tree only created.

step-2

Calculate the similarity weight

$$\text{Similarity wt} = \frac{\sum (\text{Residual})^2}{\sum [P_r (1 - P_r)]}$$

1st leaf

$$<=50 \quad = \frac{[-\cancel{0.5} + \cancel{0.5} + \cancel{0.5} - \cancel{0.5}]^2}{[0.5(1+0.5) + 0.5(1-0.5) + 0.5(1-0.5) - 0.5(1+0.5)]}$$

$$<=50 = \frac{0}{1} = 0$$

2nd leaf

$$>50 = \frac{[-\cancel{0.5}+\cancel{0.5}+0.5]^2}{[0.5(1+0.5)+0.5(1-0.5)+0.5(1-0.5)]}$$

$$>50 = \frac{0.25}{0.75} = \frac{1}{3} = 0.33$$

3rd root

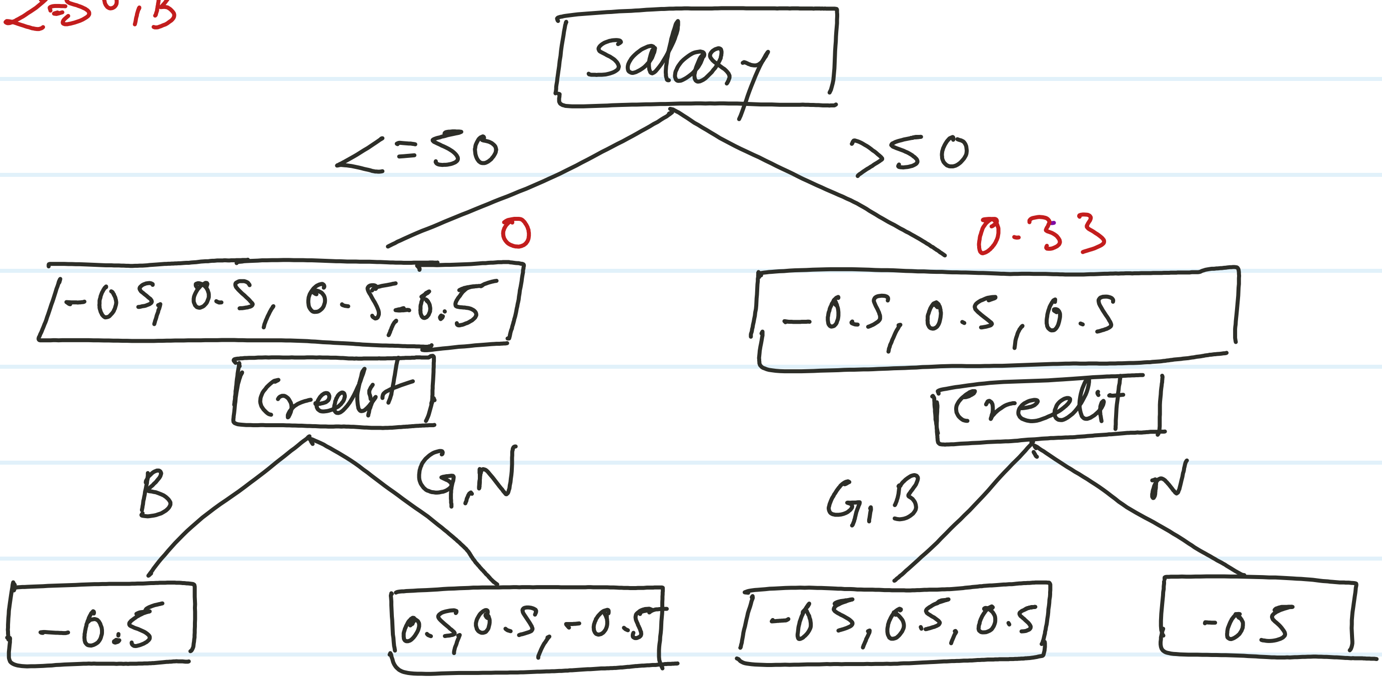
$$\text{salary} = \frac{[-\cancel{0.5}+\cancel{0.5}+0.5-\cancel{0.5}+\cancel{0.5}+\cancel{0.5}-\cancel{0.5}]^2}{[0.5(1+0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1+0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1+0.5)]}$$

$$\text{salary} = \frac{0.25}{1.75} = 0.14$$

$$\text{Gain} = (\text{Sm wt}_1 + \text{Sm wt}_2) - \text{Sm wt Root}$$
$$= (0 + 0.33) - 0.14$$
$$= 0.21$$

For each feature we calculate gain, whichever is higher will be consider as root

<=50, B



$$\text{Sm wt}_1 = \frac{(-0.5)^2}{[0.5(1+0.5)]} = \frac{0.25}{0.75} = 0.33$$

$$\text{Sm wt}_2 = \frac{(0.5 + \cancel{0.5} - \cancel{0.5})^2}{[0.5(1-0.5) + 0.5(1-0.5) + 0.5(1+0.5)]}$$

$$= \frac{0.25}{0.25 + 0.25 + 0.75} = 0.2$$

$$\text{Gain} = (\text{Sm wt}_1 + \text{Sm wt}_2) - \text{Sm wt root}$$

$$= (0.33 + 0.2) - 0$$

$$= 0.53$$

$$sw_1 = \frac{(-\cancel{0.5} + \cancel{0.5} + 0.5)^2}{[0.5(1+0.5) + 0.5(1-0.5) + 0.5(1-0.5)]}$$

$$= \frac{0.25}{1.25} = 0\ 2$$

$$sw_2 = \frac{(-0\ 5)^2}{[0.5(1+0.5)]}$$

$$= \frac{0.25}{0.75} = 0.33$$

$$\text{Gain} = (0.2 + 0.33) - 0.33$$

$$= 0.2$$

After this we check "cover value" for post pruning, $[P_r(1-P_r)]$

Cover value $= 0.5(1-0.5) = 0.25$

If cover value greater the gain value then we will use post pruning otherwise build continue tree in our case gain value

$0.53 > 0.25$

we build continue tree.

we calculat 1st decision tree, we can build any number of decision tree once residuals are calculated.

Now for predicting new record first we use odd ratio

$$\text{odd ratio} = \log\left(\frac{p}{1-p}\right)$$

suppose we are predicting, <=50, B

First calculate base model

$$= \frac{0.5}{1-0.5}$$

$$= \frac{\cancel{0.5}}{\cancel{0.5}} = 1 = \ln(1) = 0$$

Then we use function of XGBoost

$$\Rightarrow \sigma(BM + \alpha \; \text{Sm wt})$$

σ = Sigmoid activation function

α = learning rate (0 - 1)

$$\Rightarrow \sigma(0 + 0.1(\text{Sm wt of leaf}))]$$

$$\Rightarrow \sigma[0 + 0.1(0.33)]$$

$$\Rightarrow \sigma(0.033) \qquad \frac{1}{1+e^{(0.033)}}$$

$$\Rightarrow \sigma = \frac{1}{1+e^{(0.033)}}$$

$$\Rightarrow \boxed{\sigma = 0.492}$$

This is our new probability for new record

Salary	credit	App	R_1	New prob		R_2
<=50	B	0	-0.5	0.4	0 - 0.4 =	-0.6
<=50	G	1	0.5	0.5	1 - 0.5 =	0.5
>50	B	1	0.5	0.6	1 - 0.6 =	0.4
<=50	N	1	0.5	0.3	1 - 0.3 =	0.7
>50	G	0	-0.5	0.2	0 - 0.2 =	-0.2

Again based on new Residual R_2 we will build new decision tree

Final model formula will be

$$= \sigma\left[BM + \alpha . DT_1 + \alpha DT_2 + \cdots + \alpha DT_n\right]$$

XGBoost classification