Memo No. ______
Date / /

Gradient Boosting Algorithm:

→ Boosting Algorithm

Regression:

Exp	Degree	Salary
2	B.E	50k
3	Master	70k
5	Master	80k
6	PHd	100k

Table: Dataset

→ Average = 75K

Step1: Create a base model.

The base model predict the average of output feature. For the problem the average is 300/4 = 75K. So, whatever the input feature, it will predict 75K as output.

Step2: Compute residual or error

Mo Tu We Th Fr Sa Su

Memo No. (R1)

Date / /

Exp	Degree	Salary	Predicted $\hat{y}$	$(y - \hat{y})$ Error (R1)
2	B.G	50k	75K	-25K
3	Master	70k	75K	-5K
5	Master	80k	75K	5K
6	PHd	100k	75K	25K

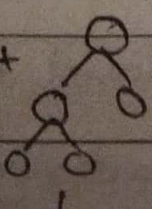
Step3: Construct next sequential decision tree with input Xi (Exp, Degree) and output as Residual or Error (R1).

Exp	Degree	Salary	$\hat{R}_1$	$\hat{R}_2$
2	B.G	50K	-25	-23
3	Master	70K	-5	-3
5	Master	80K	5	3
6	PHd	100K	25	20

assumed value.

Pipeline

Overfitting.

Dataset ⟶ Base model +  = 75-23 = 52

75

-23

Memo No. ______

Date / /

To avoid Overfitting:

Predicted $= 75 + \alpha(-23)$ ⟶ Learning rate (0 to 1)

$$\boxed{\text{Predicted} = \text{O/p of base model} + \alpha \cdot R_2}$$

Say, $\alpha = 0.1$

Predicted $= 75 + 0.1 \times (-23) = 72.7k$

Exp	Degree	Salary	$\hat{y}$	R_1	R_2	$\hat{y}$
2	B.E	50k	75	-25	-23	72.7
3	Master	70k	75	-5	-3	74.7
5	Master	80k	75	5	3	75.3
6	PHd	100k	75	25	20	77

Again for R_3

R_3 = output feature (salary) - $\hat{y}$

And again,

we will make another decision tree and we have input or independent features (Exp, Degree) and dependent label R_3 and we will get R_4.

This runs sequentially.

Mo | Tu | We | Th | Fr | Sa | Su

Memo No.

Date / /

Final function

base learner   M1   M2

$$f(x) = \alpha_0 h_0(x) + \alpha_1 h_1(x) + \alpha_2 (h_2(x)) + \alpha_3 h_3(x) + \dots + \alpha_n h_n(x)$$

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

Gradient Boost for Classification

Dataset:

likes Popcorn	Age	Favourite color	Loves Troll 2
Yes	12	Blue	Yes
Yes	87	Green	Yes
No	44	Blue	No
Yes	19	Red	No
No	32	Green	Yes
No	14	Blue	Yes

Memo No. ______

Mo Tu We Th Fr Sa Su   Date / /

Step 1: Initial Prediction with base model

when we use gradient boost for classification, the initial prediction for every individual is log(odds).

Calculating log(odds) that Someone Loves troll 2 is:

No. of people who loves troll 2 = 4

No. of people who doesn't love troll 2 = 2

Log(Odd) who loves troll-2 = $\log\left(\frac{4}{2}\right)$

This is the initial Predict ← = 0.7

Just like with Logistic Regression, the easiest way to use the log(odds) for classification is to convert it to a probability...

...and we do that with a logistic function:

$$\text{Probability of Loving troll 2} = \frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}} \Rightarrow \text{0.7 (because of rounding)}$$

Since the probability of loving troll 2 is greater than 0.5, we can classify everyone in the training dataset as someone who Loves troll 2

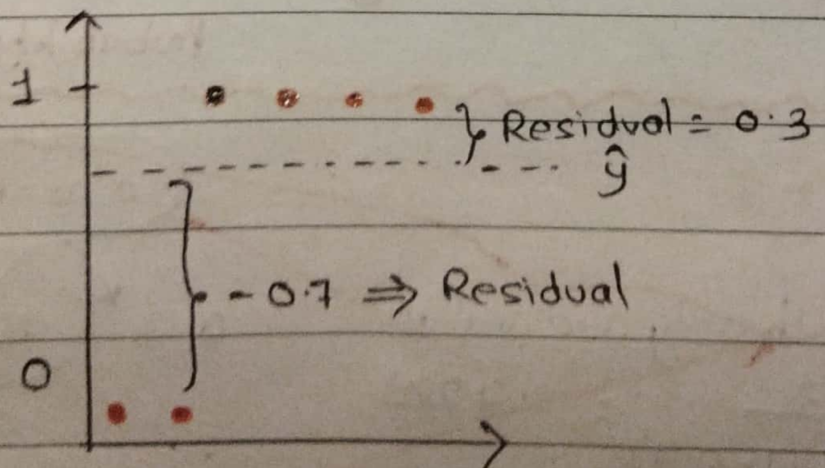
Now, classifying everyone in the training dataset as someone who loves troll 2 is pretty lame becouse two of the people do not love the movie.

We can measure how bad the initial prediction is by calculating Pseudo Residuals, the difference between the observed and the predicted values.

$$\boxed{\text{Residual} = (\text{observed} - \text{Predicted})}$$

Residuals on a graph:

x—x—x



Yes → 1
No → 0

Memo No.

Date / /

likes Popcorn	Age	Favourite Color	Loves Troll 2	$\hat{y}$	Residual
Yes	12	Blue	Yes	0.7	0.3
Yes	87	Green	Yes	0.7	0.3
No	44	Blue	No	0.7	-0.7
Yes	19	Red	No	0.7	-0.7
No	32	Green	Yes	0.7	0.3
No	14	Blue	Yes	0.7	0.3

Step 3: Using input features (likes Popcorn, Age, Favorite Color) we predic Residual (R2):

$$\boxed{R2 = \frac{\sum \overbrace{\text{Residual}}^{R1}}{\sum \text{Previous Probability} \times (1 - \text{Previous Probability})}}$$