Gradient Boost for Classification →

Consider.

Likes Popcorn	Age	Favorate color	Loves Jawan movie. ↓ (Probability)
Yes	12	Blue	Yes (1)
Yes	87	Green	Yes (1)
NO	44	Blue	NO (0)
Yes	19	Red	NO (0)
NO	32	Green	Yes (1)
NO	14	Blue	Yes (1)

(Actual values)

Step1: Start with Initial Prediction →

[Note: When we use gradient boost for classification, the initial prediction for every individual sample :→ log(odd)]

$$odd = \frac{\text{No of Yes}}{\text{No of NO}} = \frac{4}{2} = 2$$

$$\log_e(odd) = \log_e(2) = 0.7$$

To use this we have to Convert it into Probability =

Probability of loving Jawan movie = $\frac{1}{1+e^{-\log(odd)}} = \frac{1}{1+e^{-0.7}} \approx 0.7$ ↑

So Initial Prediction for Probability of all the Samples for loving Jawan movie = 0.7 //

Initial Prediction=0.7

Likes Popcorn	Age	Favorate color	Loves Jawan movie. ↓	Residual Error
Yes	12	Blue	Yes (1)	0.3
Yes	87	Green	Yes (1)	0.3
		Blue	NO (0)	-0.7

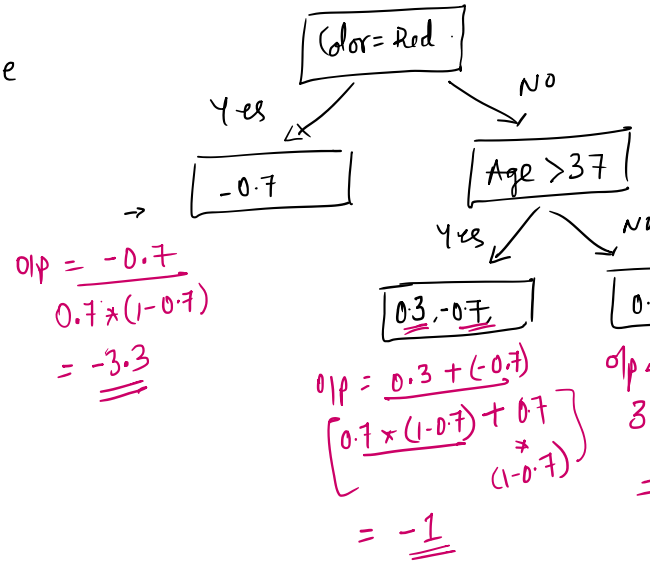
Yes	87	Green	Yes (1)	0.3
NO	44	Blue	NO (0)	-0.7
Yes	19	Red	NO (0)	-0.7
NO	32	Green	Yes (1)	0.3
NO	14	Blue	Yes (1)	0.3

Actual

(Initial Residual Error)

Now let us Build Tree Color And Age

Let the Tree be



o/p = -0.7
0.7 * (1-0.7)
= -3.3

o/p = 0.3 + (-0.7)
[0.7 × (1-0.7) + 0.7 × (1-0.7)]
= -1

o/p = 0.3+0.3+0.3
3 * (0.7) × (1-0.7)
= 1.4

For Classification:

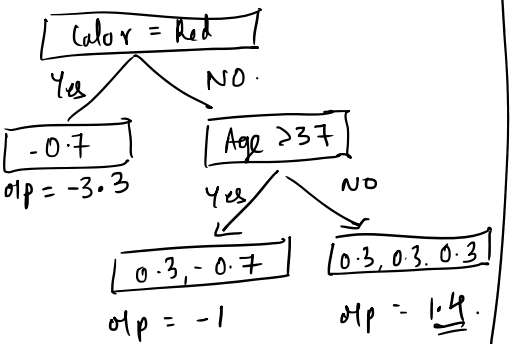
o/p value at every Leaf

= Σ Residual

Σ (Previous Prob * (1 − Previous Prob)

Here this time Previous Prob = Initial Prediction

∴ New Prediction = Initial Prediction + 0.8 * (learning Rate)



Consider first Sample

Like Popcorn	Age	Color	Initial
Yes	12	Blue	0.7

New Prediction = 0.7 + 0.8 * 1.4 = 1.8

Convert into Probability = $\frac{1}{1+e^{-1.8}}$ = 0.9

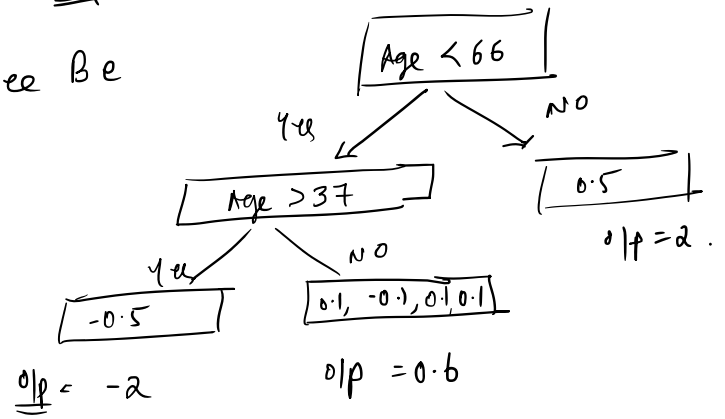
Similarly Calculate for all the other Samples -

Now the New Prediction be :→ Initial Prediction = 0.7

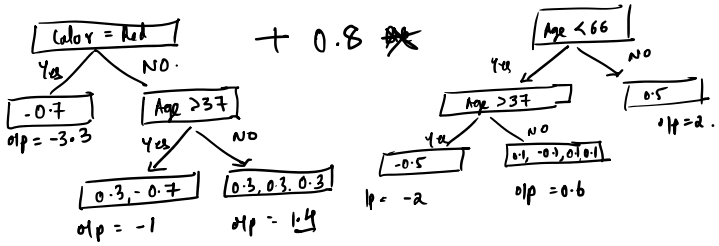
Likes Popcorn	Age	Favorate color	Loves Jawan movie (Actual values)	Residual Error	New Prediction	New Residual
Yes	12	Blue	Yes (1)	0.3	0.9	0.1
Yes	87	Green	Yes (1)	0.3	0.5	0.5
NO	44	Blue	NO (0)	-0.7	0.5	-0.5
Yes	19	Red	NO (0)	-0.7	0.1	-0.1
NO	32	Green	Yes (1)	0.3	0.9	0.1
NO	14	Blue	Yes (1)	0.3	0.9	0.1

* Here New prediction is not Same for all the Samples.
* Each Sample has its own prediction and calculated New residual for each Sample.
* Again on the basis of new Residual let us Build a Tree Using only Age only

Let the Tree Be



∴ New Prediction = Initial Prediction + 0.8 * (learning Rate)



* Apply each Sample in dataset on the above model and calculate New Predictions

Apply ... sample ... number ...

New Predictions

* Again calculate New Residual and so on -
* This process repeats untill we have made the maximum specified number of Trees or the residual gets very very small.

Summary : Gradient Boost for classification.

(1) Consider the dataset
(2) Find log(odd) on the target
(3) Find Initial Probability $= \frac{1}{1+e^{-\log(odd)}}$.
(4) Now calculate Initial Residual based on Initial Probability
(5) Build a Tree using some of the features and Initial Residual (-Tree1)
(6) Calculate New Prediction for = Initial + learning * Tree1
Every sample Prediction Rate
(7) Using New Predicate Calculate New Residual.
(8) Construct a new Tree based on some feature and New Residual and so on.
(9) Repeat untill max no of Trees are constructed or the Residuals are insignificant.

Summary : Gradient Boost for Regression.

(1) Consider the dataset
(2) Find Initial Prediction = Avg of observed value.
(3) Now calculate Initial Residual based on Initial Prediction
(4) Build a Tree using some of the features and Initial Residual (-Tree1)
(5) Calculate New Prediction for = Initial + learning * Tree1
Every sample Prediction Rate

(5) Calculate New Prediction for = Initial + learning rate * Every sample Prediction Rate

(6) Using New Predicate Calculate New Residual.

(7) Construct a new Tree based on some feature and New Residual and so on.

(8) Repeat untill max no of Trees are constructed or the Residuals are insignificant.

XG Boost (Xtreme Gradient Boost)

① XGBoost for Regression

Consider

x : y (Actual)

Dosage of Drug	Effectiveness of Drug
10	-10
20	7
25	8
35	-7

For Classification

Dosage	Effectiveness
10	0
20	1
25	1
35	0

Build Prediction Model Using XGBoost

Step1: Assume Initial threshold for Effectiveness = 0.5

Step 2 Calculate Residual.

Initial Prediction = 0.5

Dosage of Drug	Effectiveness of Drug	Residual
10	-10	-10.5
20	7	6.5
25	8	7.5
35	-7	-7.5

Let $\lambda = 0$

Step3: Consider all the Residuals in Root initially.

-10.5, 6.5, 7.5, -7.5

$$\text{Similarity} = \frac{(-10.5 + 6.5)^2}{4 + 0} = 4$$

Now let us see if we can cluster the residual better on Similarity by split

Note

$$\text{Similarity Score} = \frac{(\text{Sum of Residual})^2}{\text{No of Residual} + \lambda}$$

[λ = Regularization Parameter ⇒ prevents Overfitting]

Gain = Similarity (left) + Similarity (Right) − Similarity (Root).

Determine Dosage value for split ⇒.

Case1) Consider Average of first 2 dosage, i.e 10 & 20

Avg = 15

Similarly = 4

Case2) Consider dosage as Average of 20 & 25

Average = 22.5

Similarly = 4.

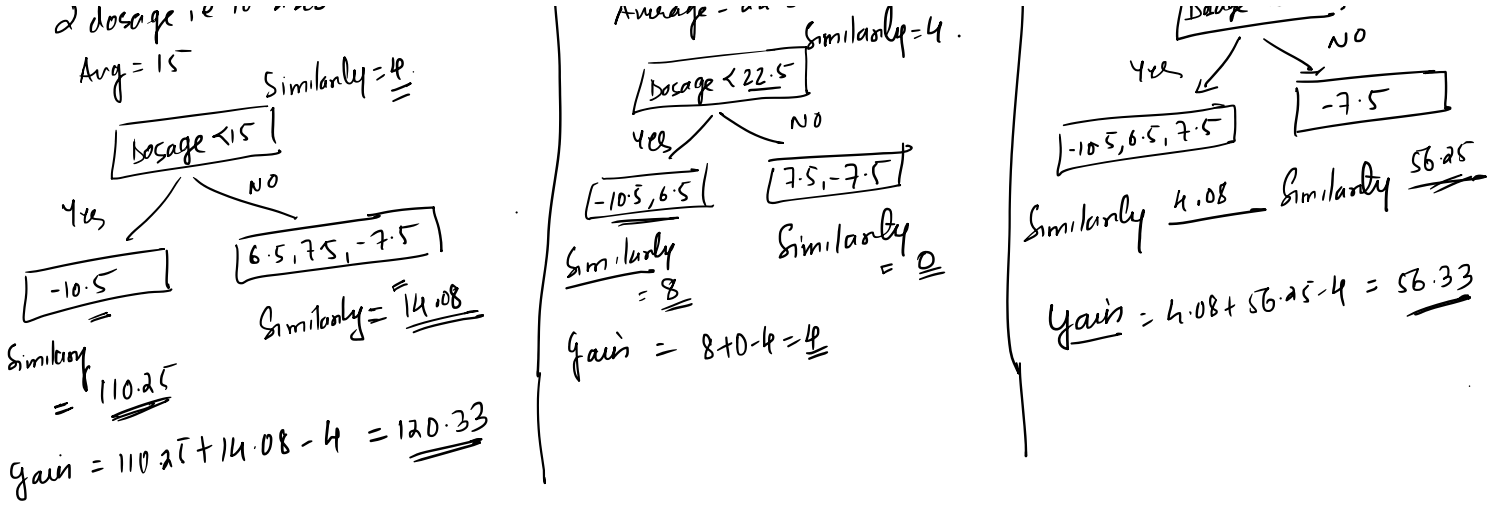
Dosage < 22.5

Case3: Dosage as Average of 25 & 35 Avg = 30

Similarly = 4

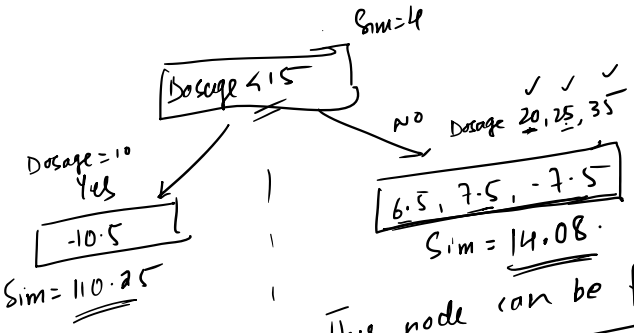
Dosage < 30

Yes ↙ NO ↘

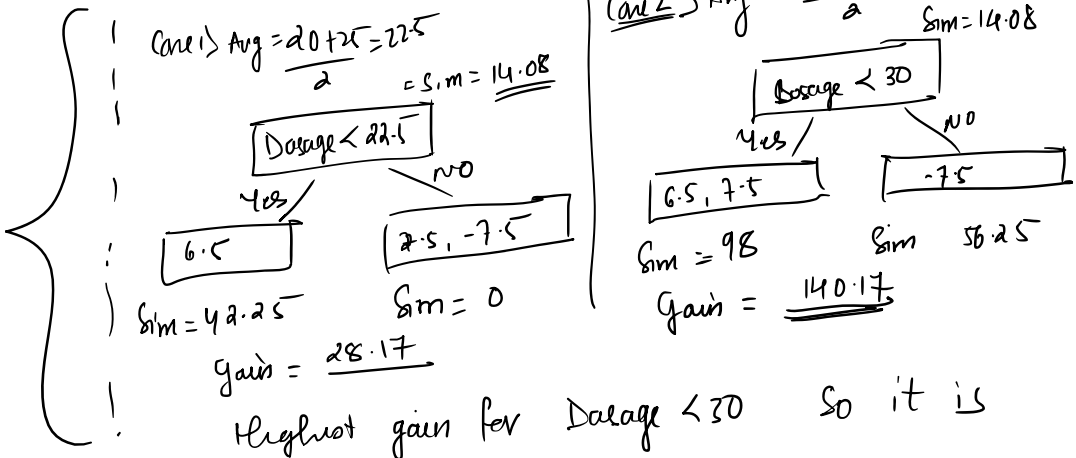


Since the Dosage < 15 gives highest gain = 120.33

The split factor = Dosage < 15

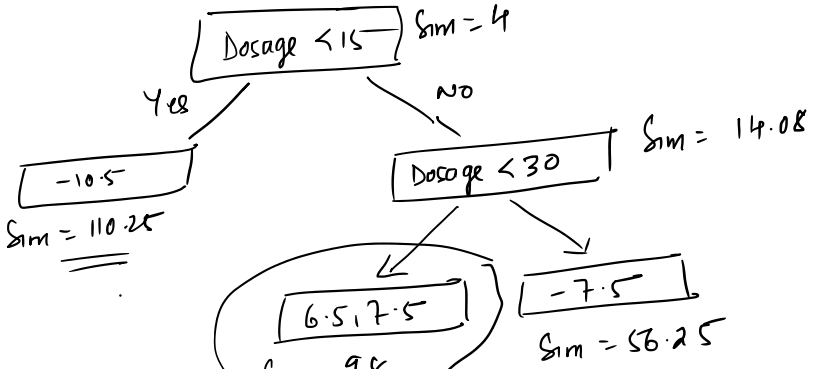


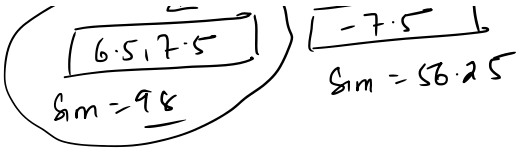
This node can be further splitted



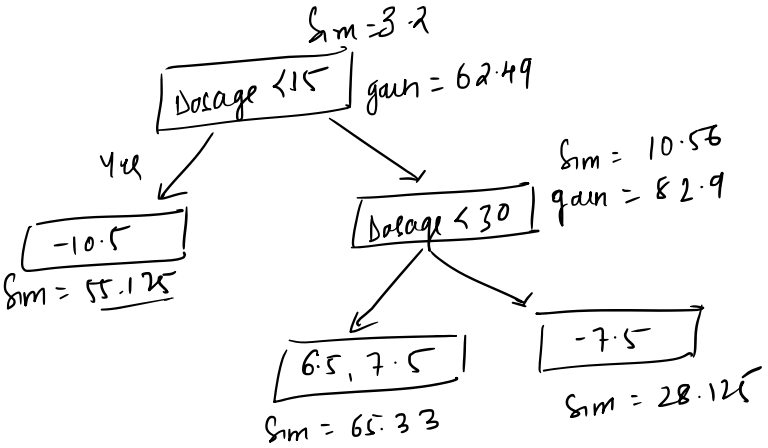
Highest gain for Dosage < 30 So it is the split condition

Now. Tree $\lambda = 0$





Now for $\lambda = 1$



Note

When $\lambda \geq 0$ the Similarity Score decreases (↓)

Also Gain decreases (↓).