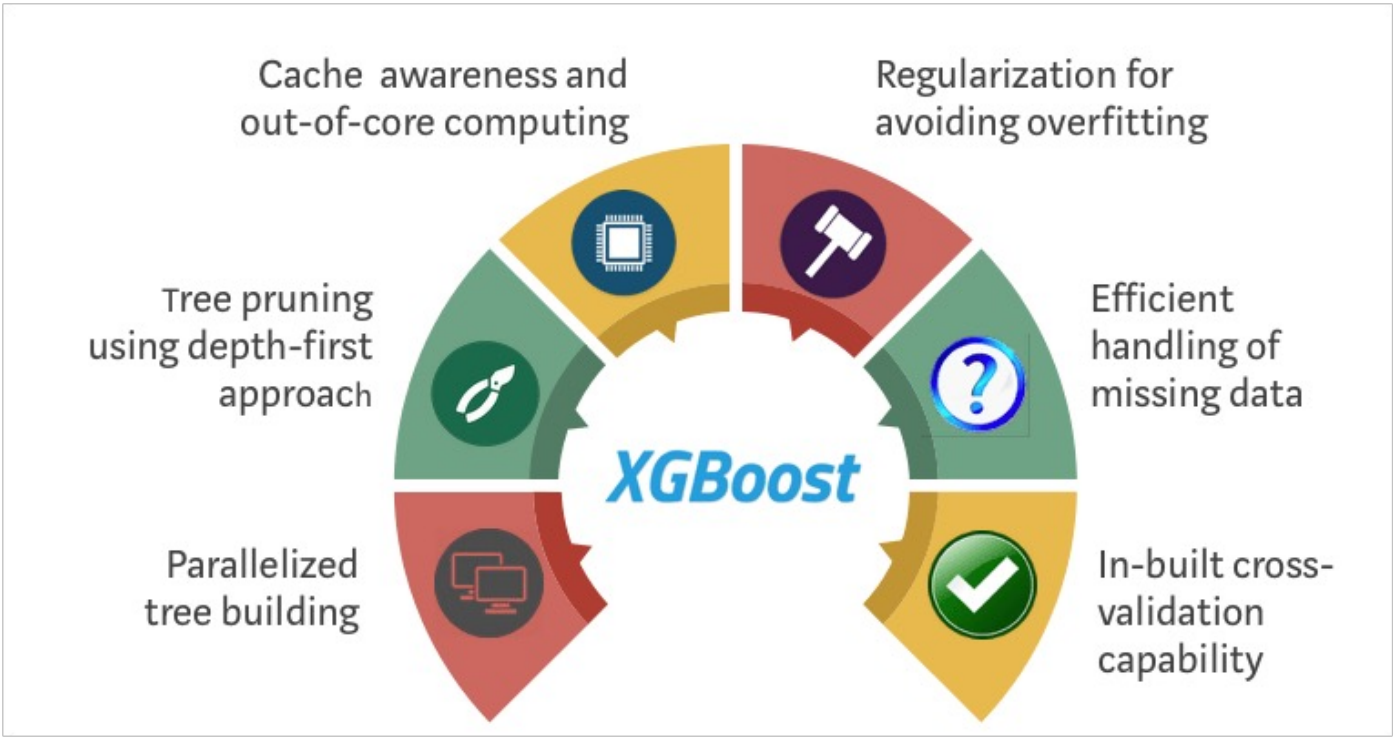
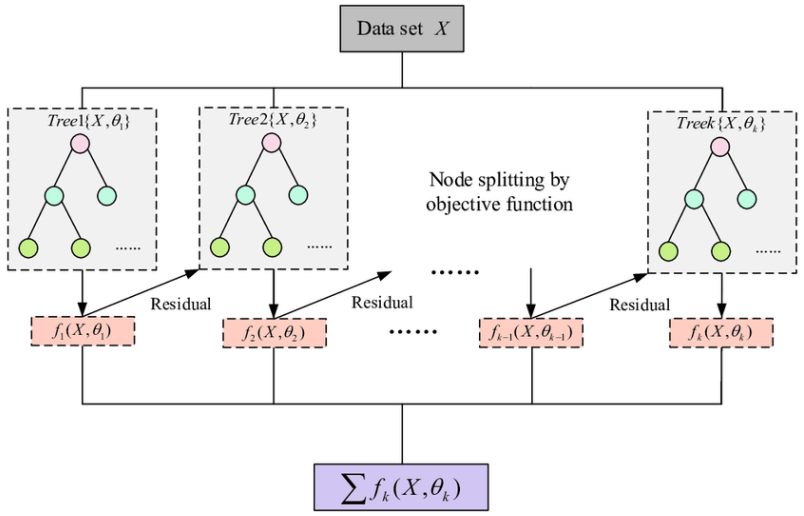
AI VIETNAM
All-in-One Course

XGBoost





Vinh Dinh Nguyen
PhD in Computer Science

Year 2023

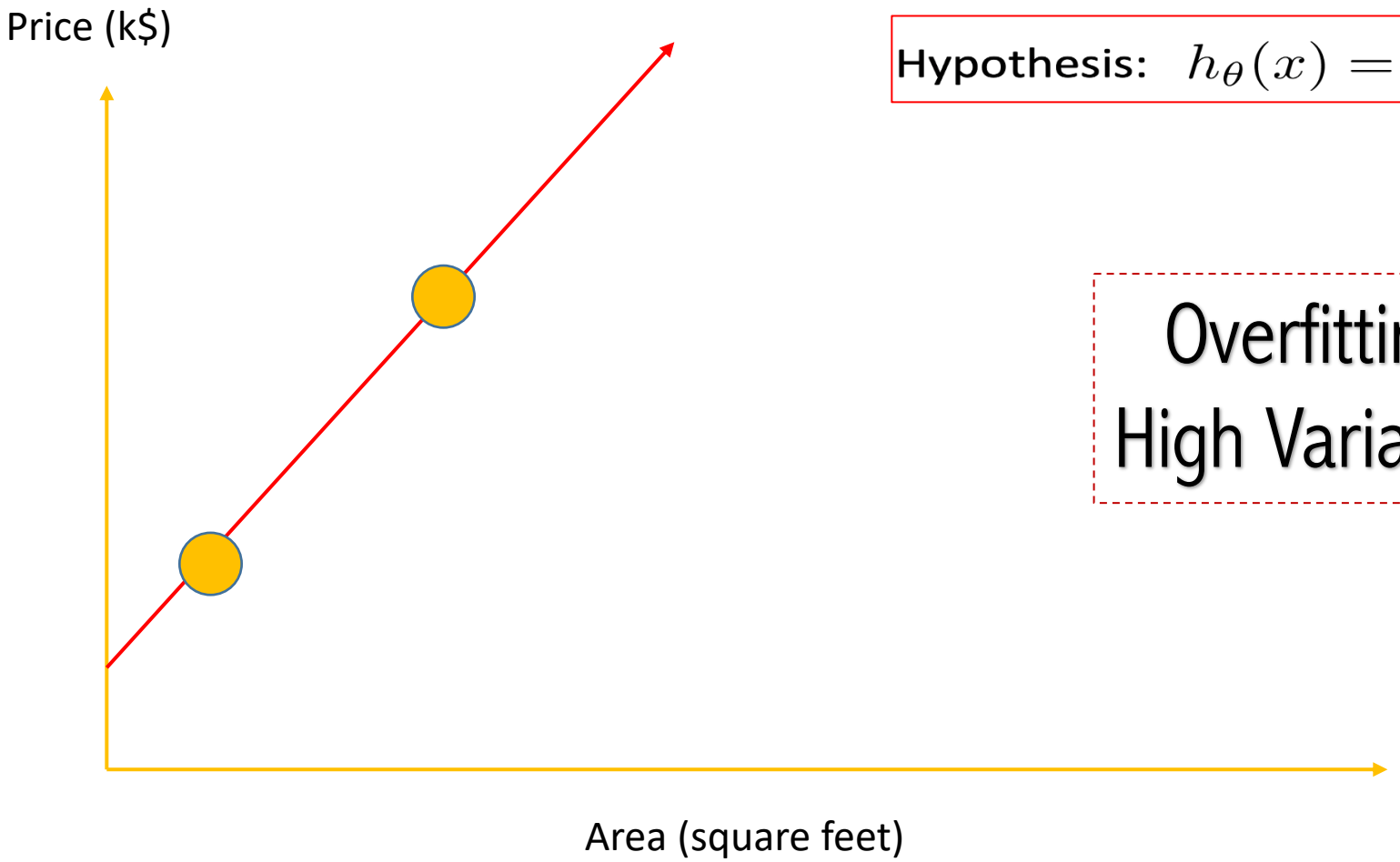
Outline

- Regularization
- XGBoost For Regression
- XGBoost For Classification
- XGBoost: Mathematical Explanation
- How To Fill Missing Values
- Example

Outline

- Regularization
- XGBoost For Regression
- XGBoost For Classification
- XGBoost: Mathematical Explanation
- How To Fill Missing Values
- Example

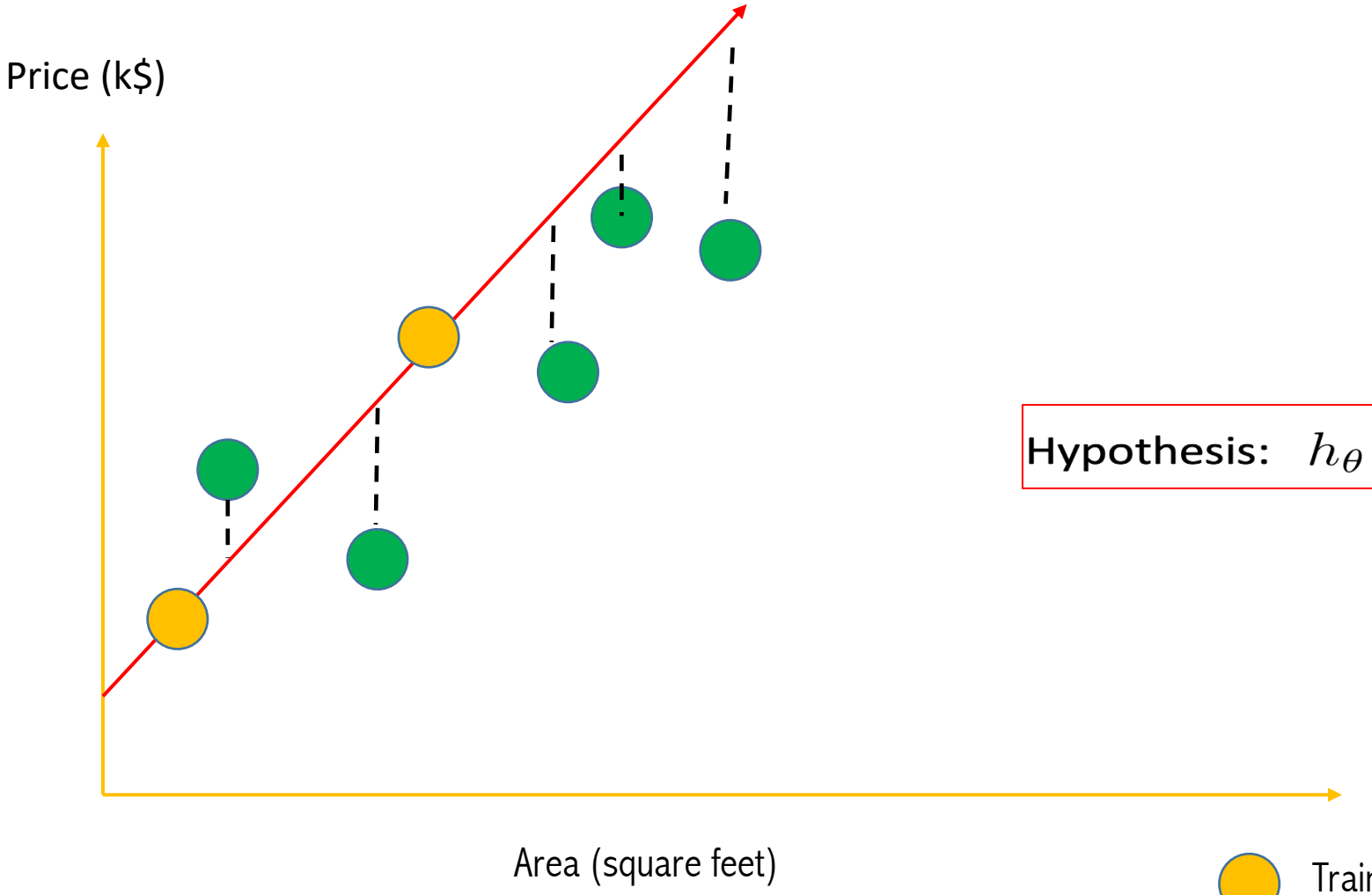
Regularization



Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Overfitting
High Variance

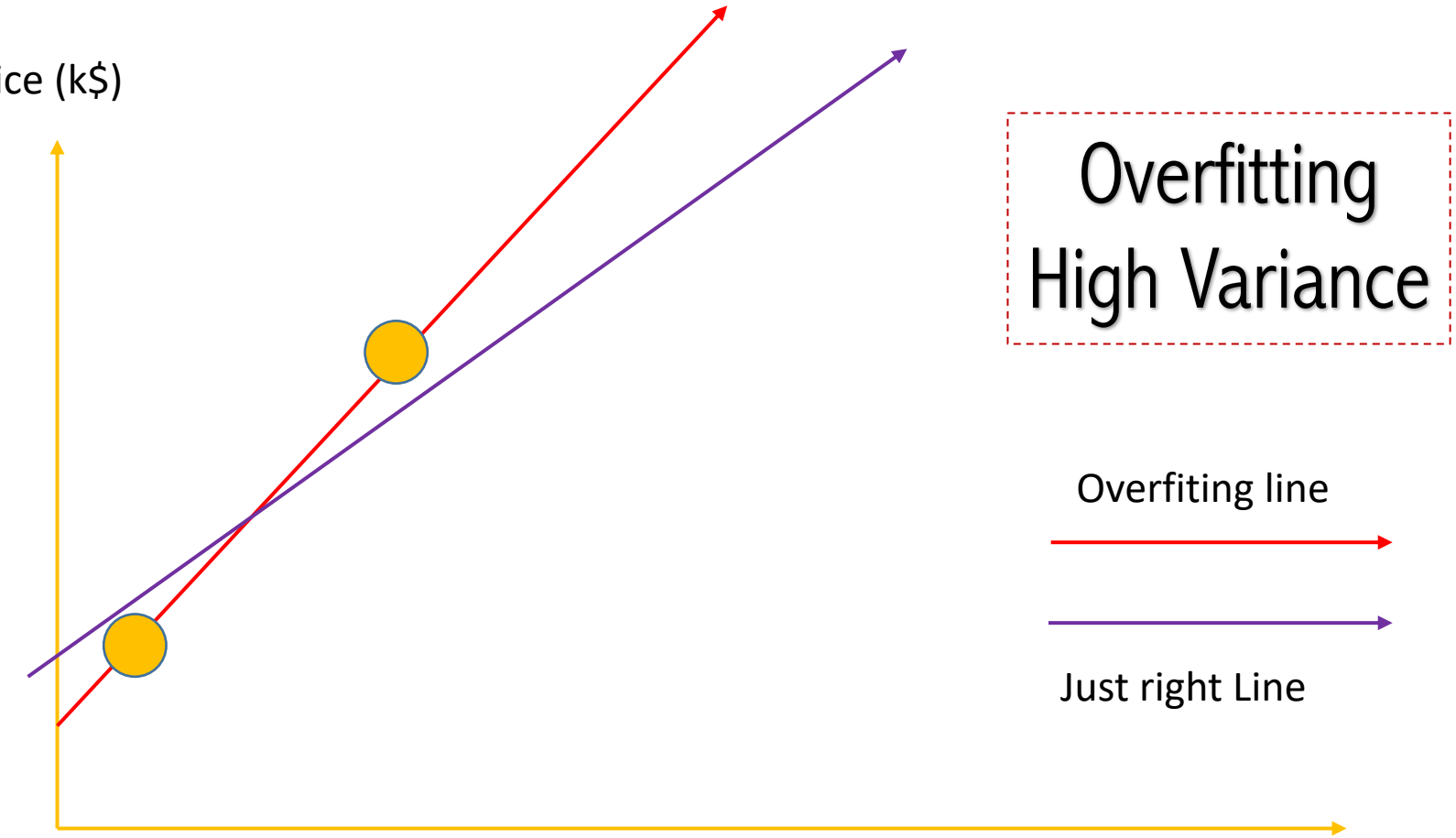
Regularization



Overfitting
High Variance

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Regularization



Overfitting
High Variance

Overfiting line

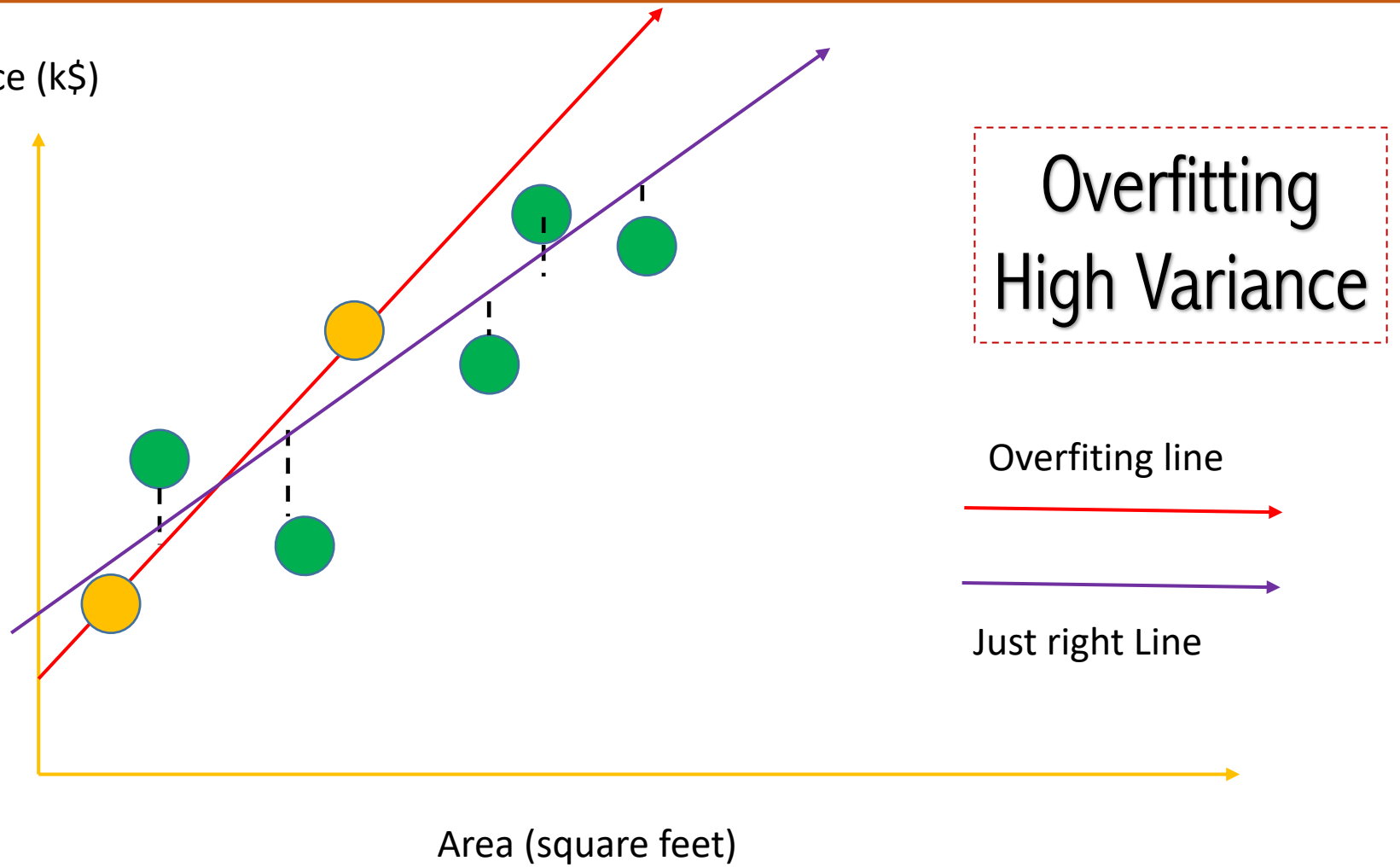
Just right Line

Train data

Test data

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Regularization



Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Price = Intercept + slope * area

Regularization

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Regularization.

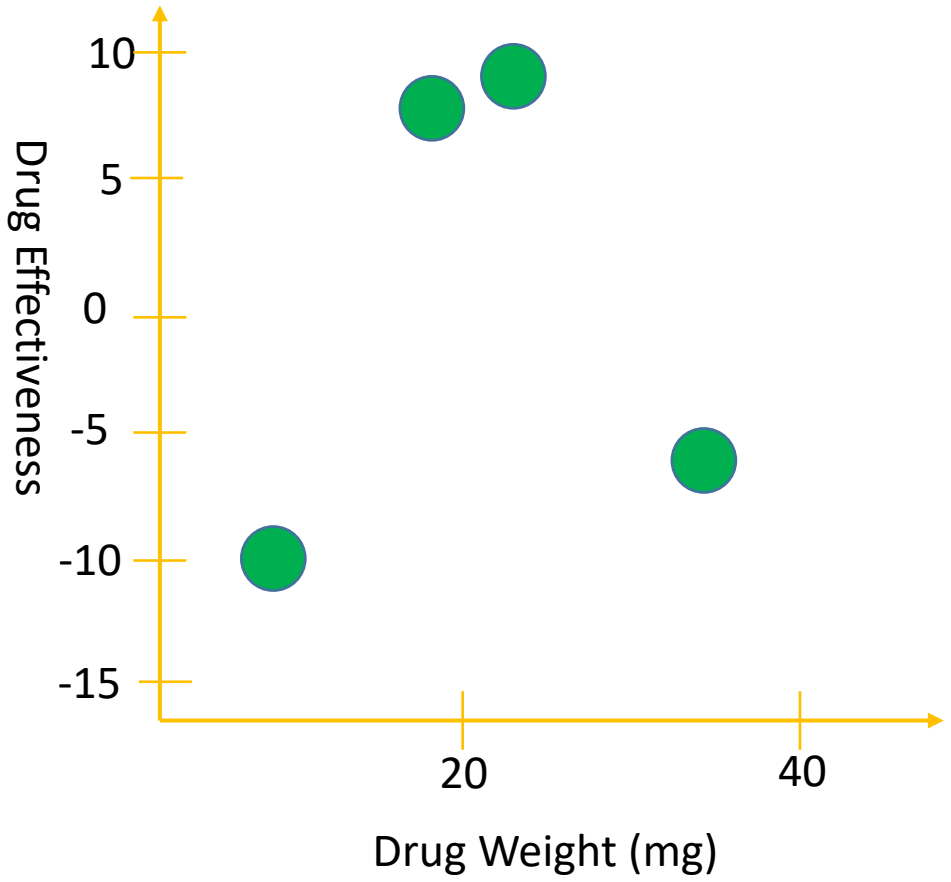
$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right]$$

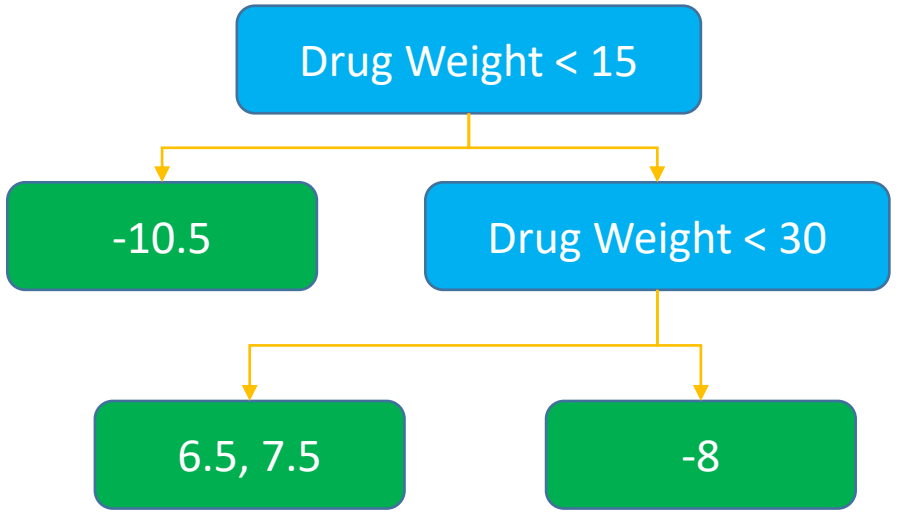
Price = [Intercept + slope * area] + $\boldsymbol{\lambda}$*slope^2

Outline

- Regularization
- XGBoost For Regression
- XGBoost For Classification
- XGBoost: Mathematical Explanation
- How To Fill Missing Values
- Example

XGBoost For Regression

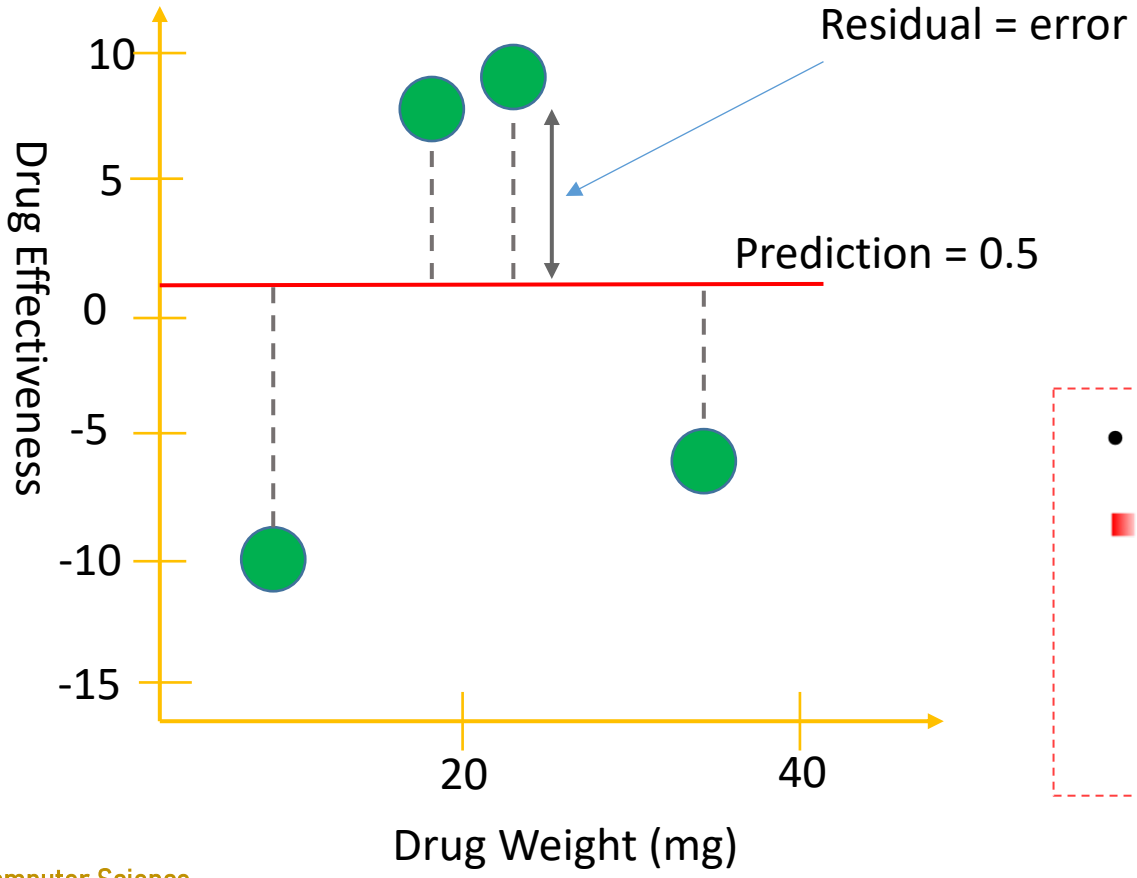




XGBoost For Regression

Step 1

- Initialize the first prediction for drug effectiveness
- Any number, for default, we set 1st prediction = 0.5



$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$$

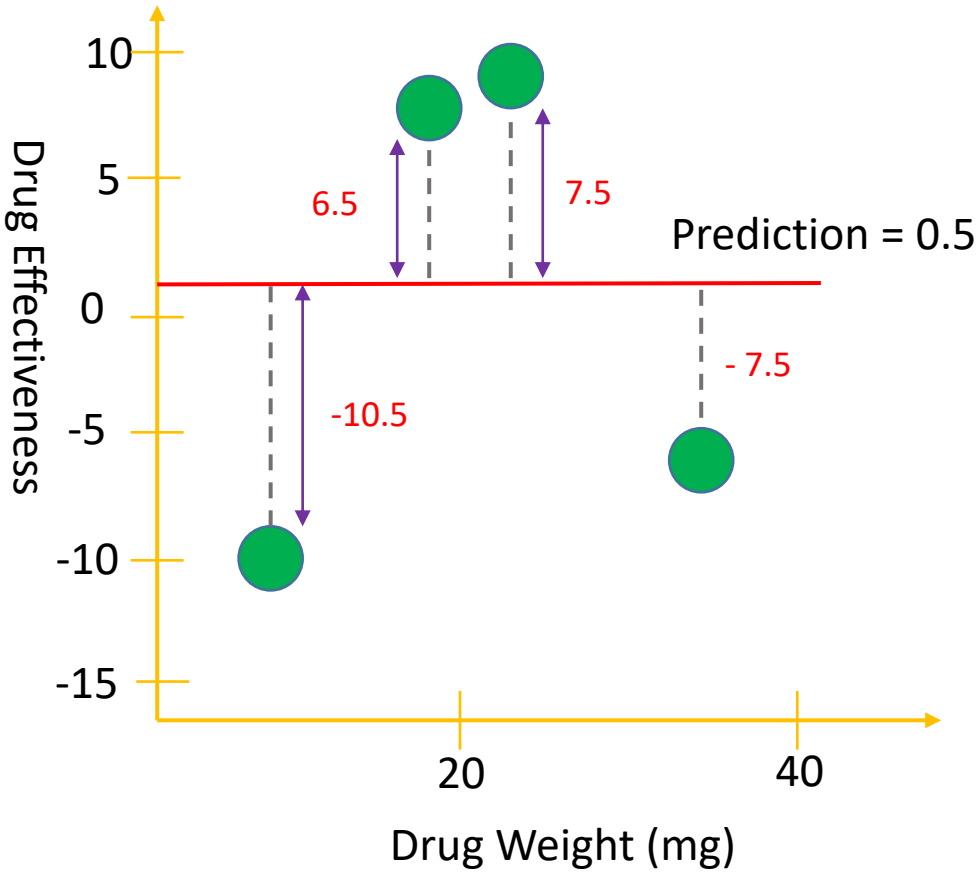
- $\{(x_i, y_i)\}^n_1$
- Loss function = $L(y_i, F(x)) = 1/2 * (\text{Output} - \text{Predicted})^2$

$$\frac{dL}{dPredicted} = 2/2\ (\text{Output} - \text{Predicted}) * -1 = -(\text{Output} - \text{Predicted})$$

Tricky implementation here

XGBoost For Regression

Step 1



Start with single Leaf of residuals

-10.5, 6.5, 7.5, -7.5

Compute Similarity Score

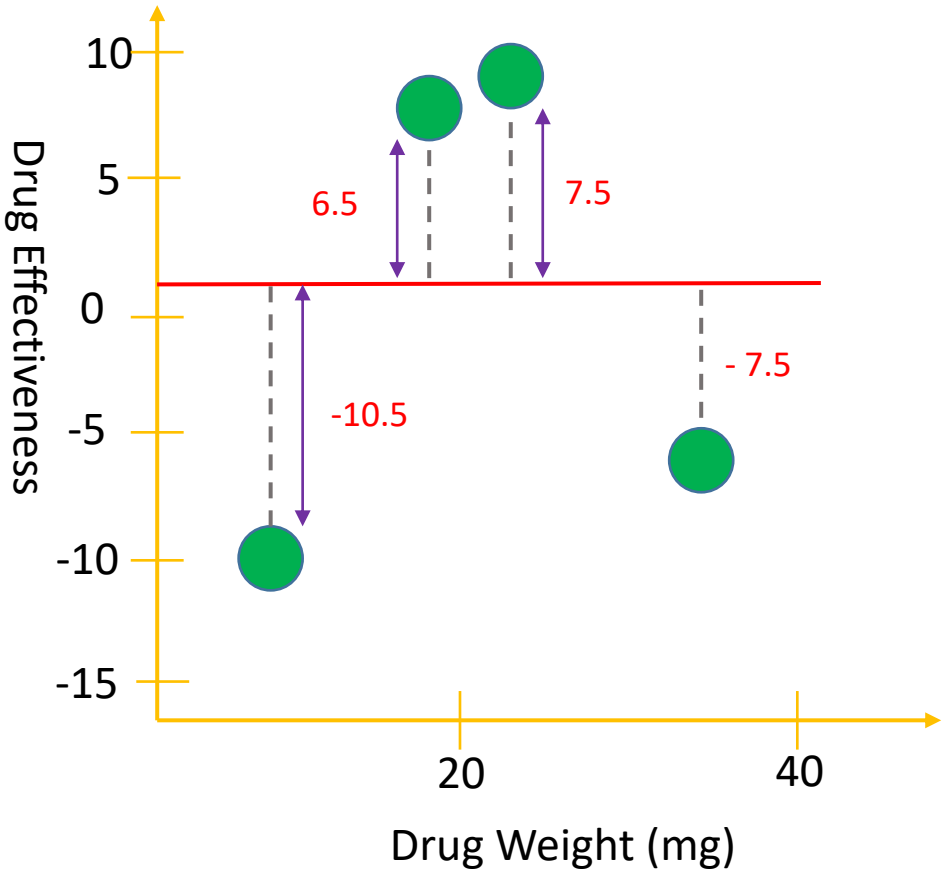
$$SC = \frac{[\sum (output - predicted)]^2}{m + \lambda}$$

m: number of samples

λ : *regularization parameters*

XGBoost For Regression

Step 1



Start with single Leaf of residuals

-10.5, 6.5, 7.5, -7.5

m = 4
$\lambda = 0$

Compute Similarity Score

$$SC = \frac{[\sum (output - predicted)]^2}{m + \lambda}$$

$$SC = \frac{[-10.5+7.5+6.5+(-7.5)]^2}{4} = 4$$

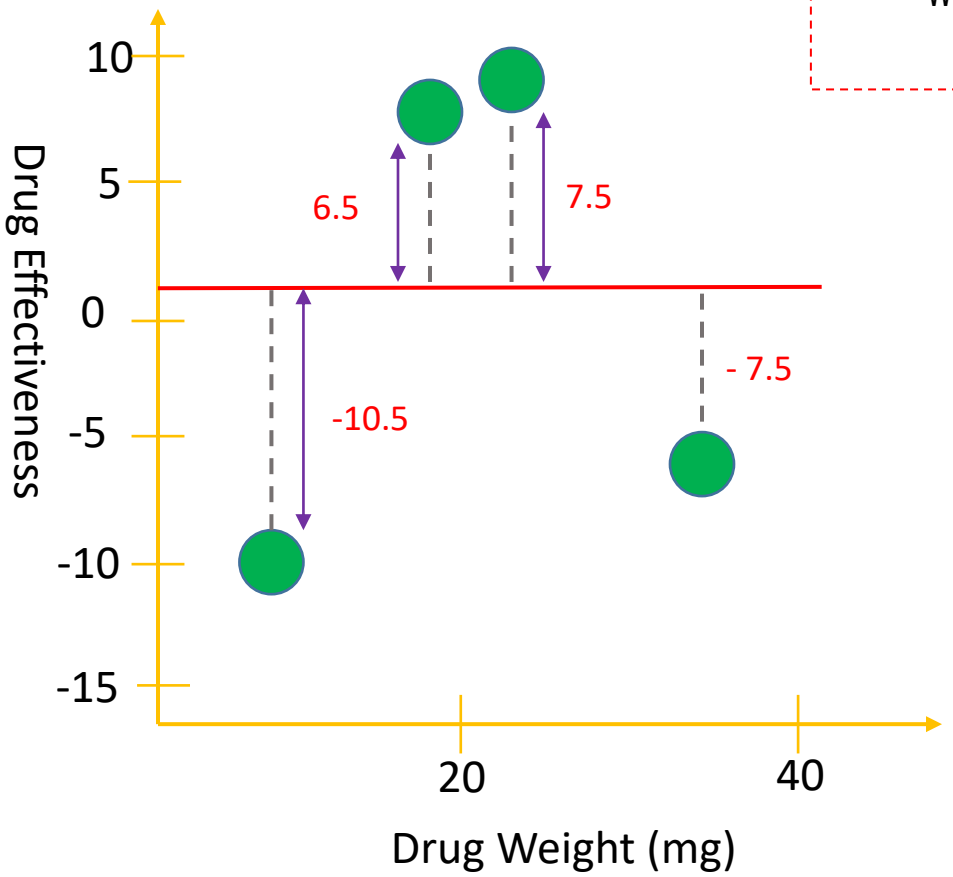
XGBoost For Regression

Step 1

SC = 4

-10.5, 6.5, 7.5, -7.5

What happens if we try to split residuals into two groups => measure the similarity score



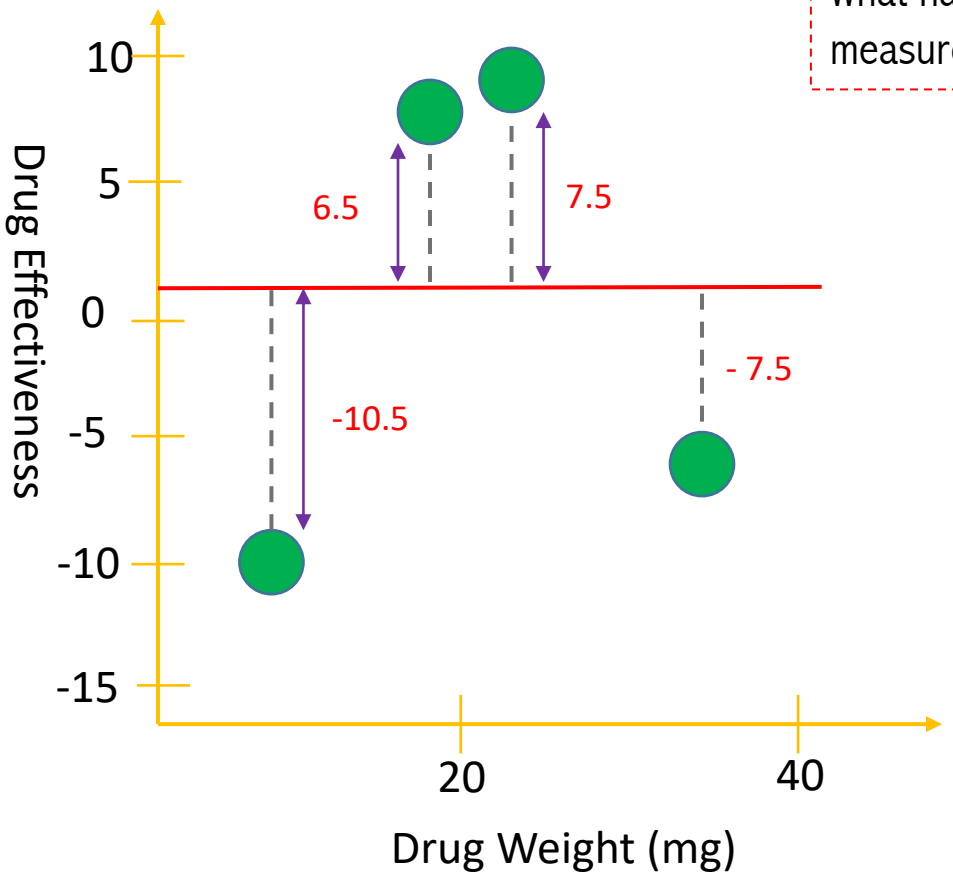
XGBoost For Regression

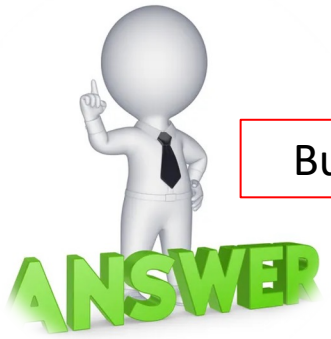
Step 1

SC = 4

-10.5, 6.5, 7.5, -7.5

What happens if we try to split residuals into two groups => measure the similarity score

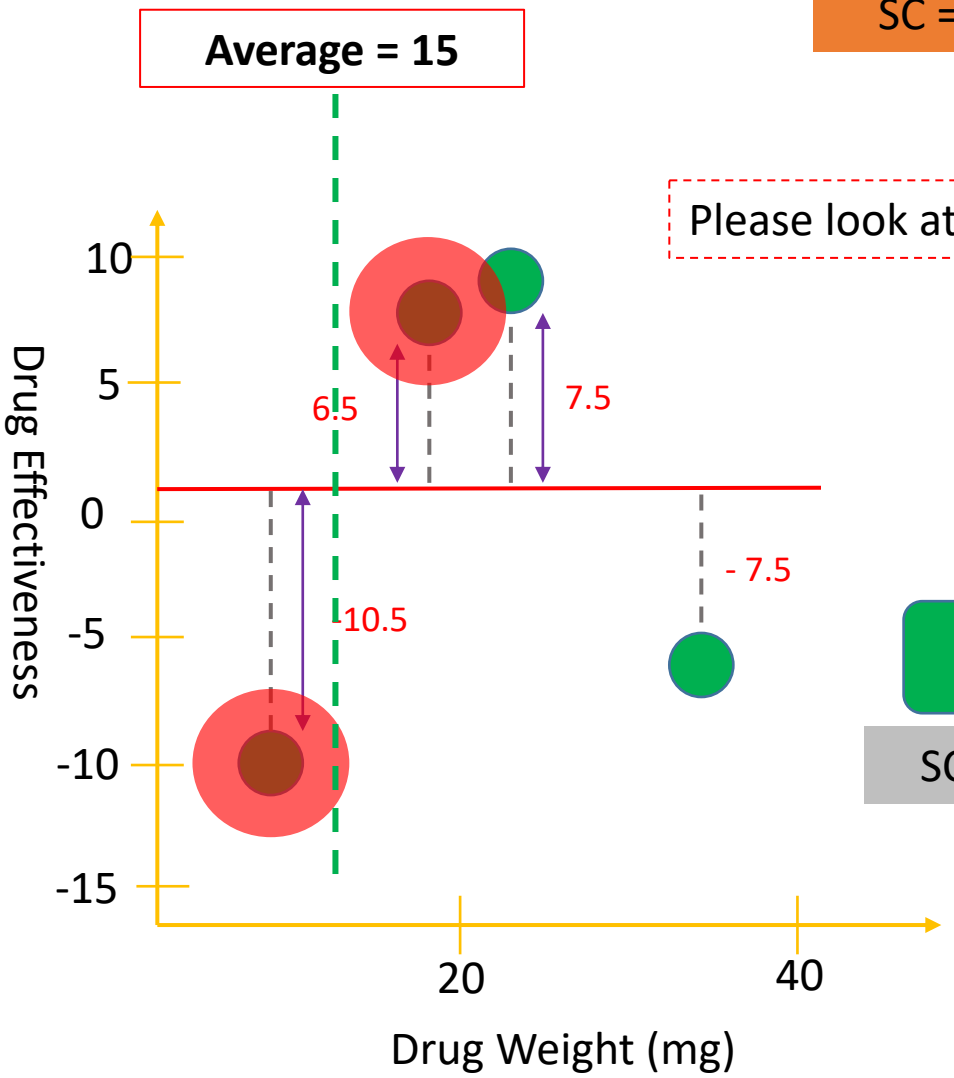




Build a tree on it

XGBoost For Regression

Step 1



SC = 4

-10.5, 6.5, 7.5, -7.5

Please look at the two outputs with lowest drug weights

Drug weights < 15

SC = 4

-10.5

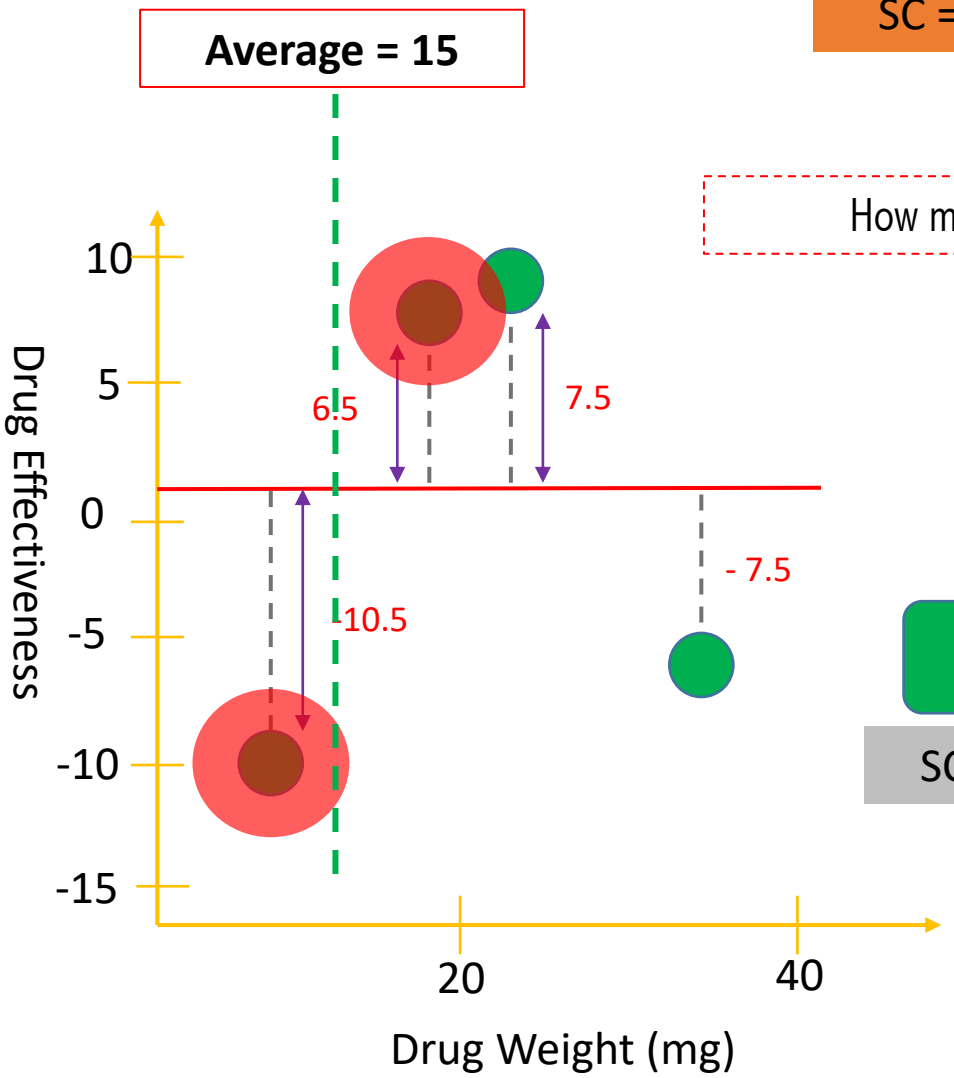
SC = 110.25

6.5, 7.5, -7.5

SC = 14.08

XGBoost For Regression

Step 1



SC = 4

-10.5, 6.5, 7.5, -7.5

How much better leaves cluster similar Residual than the root?

Drug weights < 15

SC = 4

-10.5

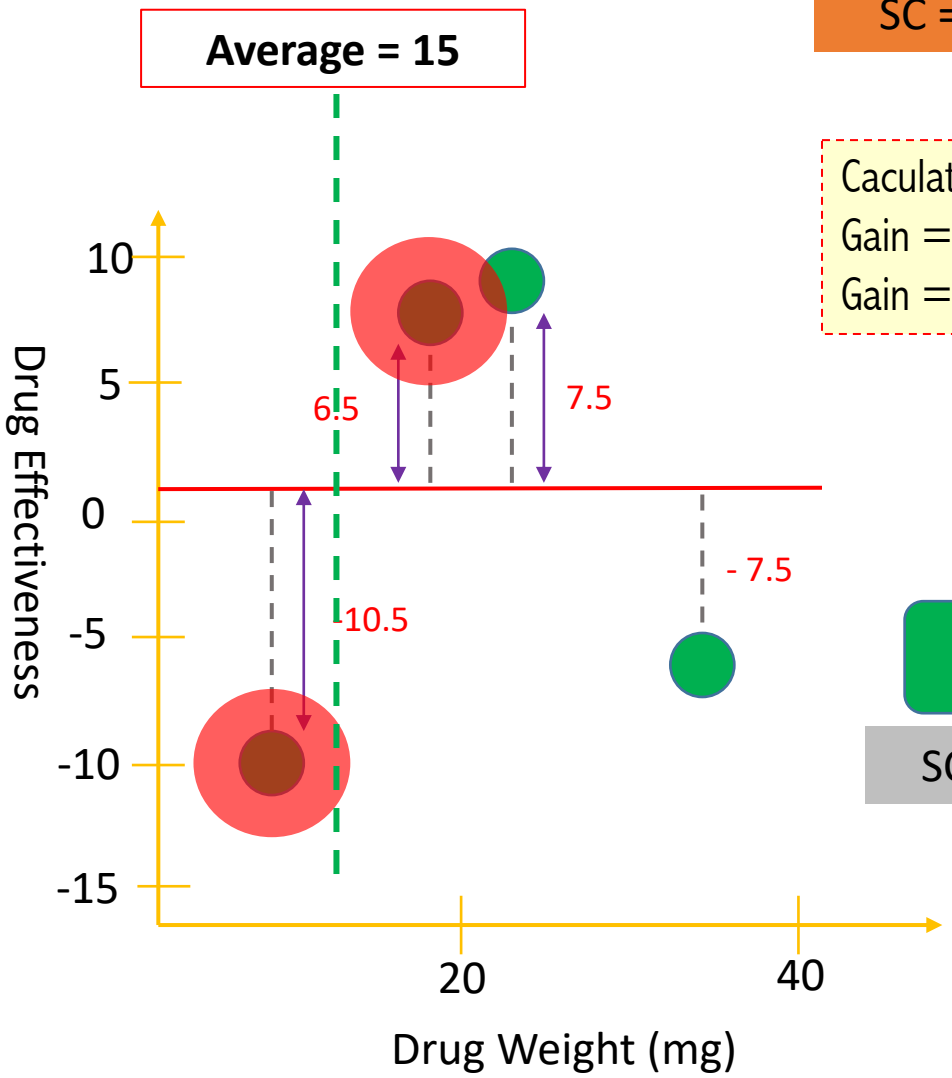
SC = 110.25

6.5, 7.5, -7.5

SC = 14.08

XGBoost For Regression

❑ *Step 1*



SC = 4

-10.5, 6.5, 7.5, -7.5

Residual rất khác nhau, triệt tiêu lẫn nhau, nên SC nhỏ

Caculate the Gain.
Gain = Left SC + Right SC - Root SC
Gain = 120.33

Drug weights < 15

SC = 4

-10.5

SC = 110.25

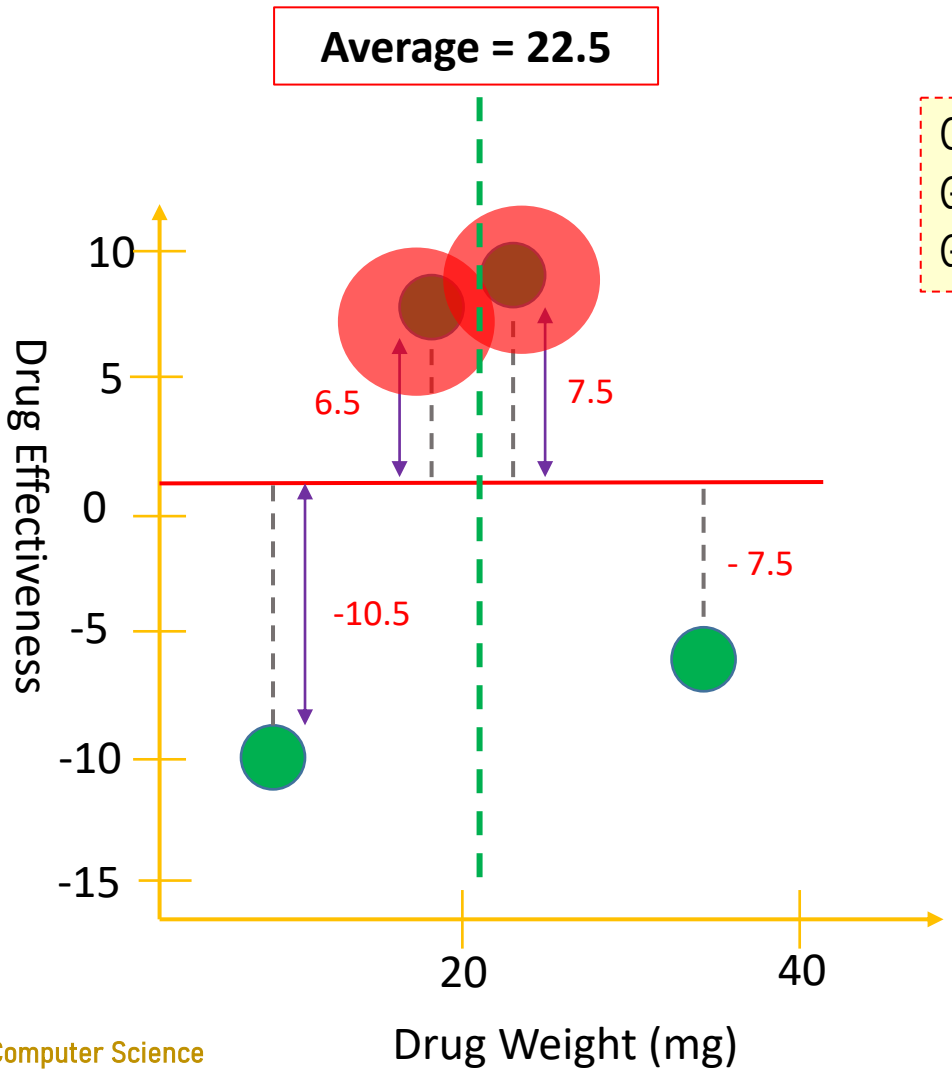
6.5, 7.5, -7.5

SC = 14.08

Residual giống nhau hoặc không triệt tiêu lẫn nhau, nên SC khá lớn

XGBoost For Regression

❑ *Step 1*

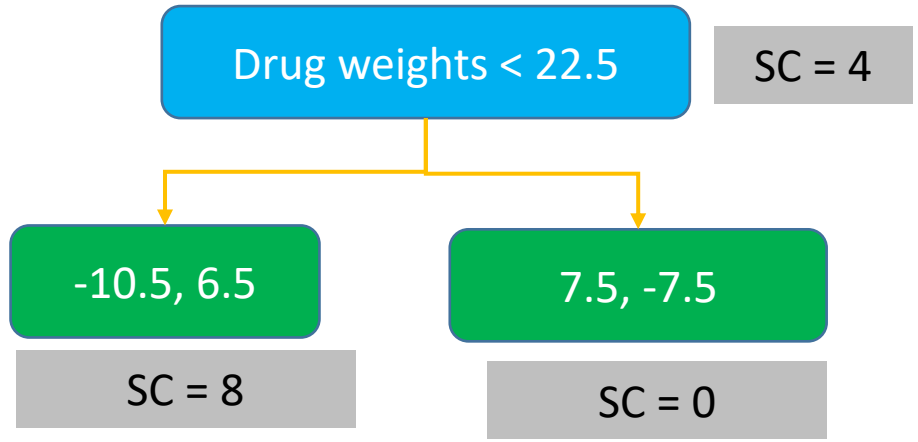


SC = 4

-10.5, 6.5, 7.5, -7.5

Residual rất khác nhau, triệt tiêu lẫn nhau, nên SC nhỏ

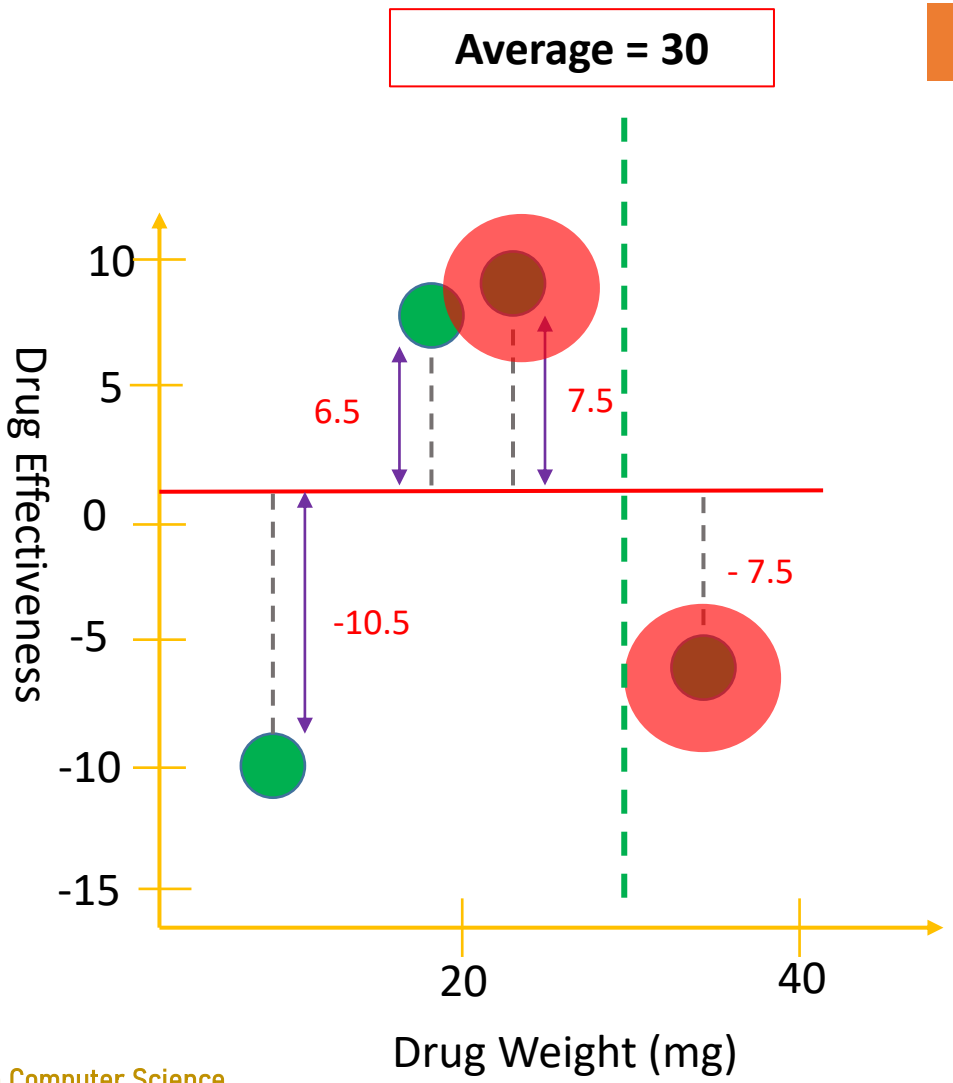
Caculate the Gain.
Gain = Left SC + Right SC - Root SC
Gain = 4.0



Residual giống nhau hoặc không triệt tiêu lẫn nhau, nên SC khá lớn

XGBoost For Regression

❑ *Step 1*

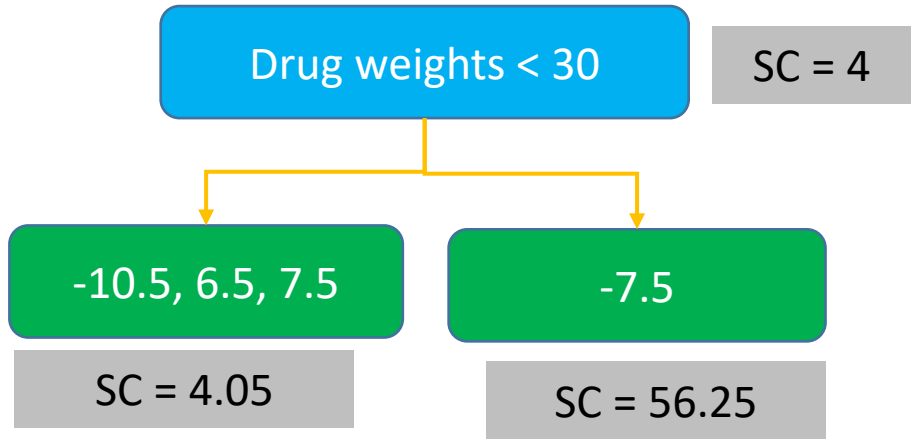


SC = 4

-10.5, 6.5, 7.5, -7.5

Residual rất khác nhau, triệt tiêu lẫn nhau, nên SC nhỏ

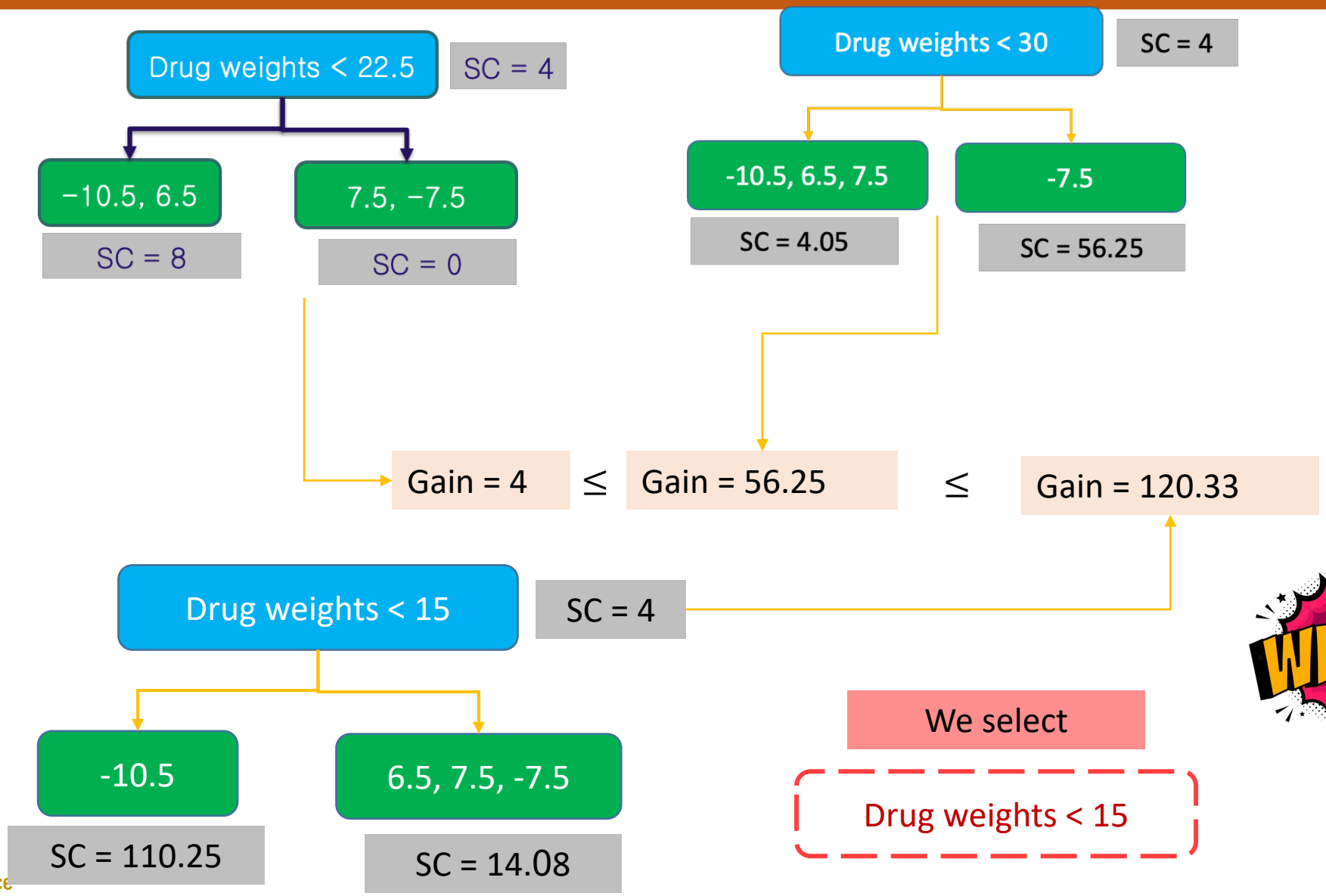
Caculate the Gain.
Gain = Left SC + Right SC - Root SC
Gain = 56.33



Residual giống nhau hoặc không triệt tiêu lẫn nhau, nên SC khá lơn

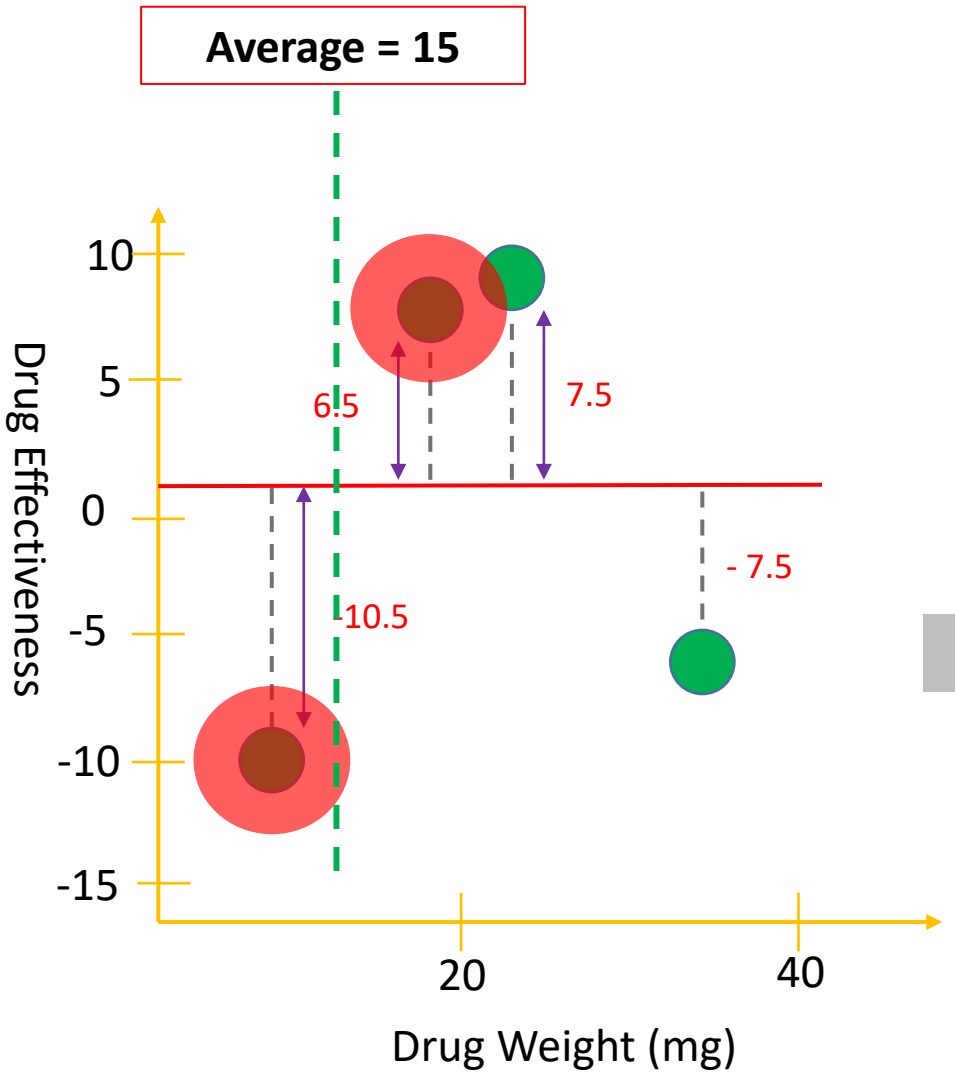
XGBoost For Regression

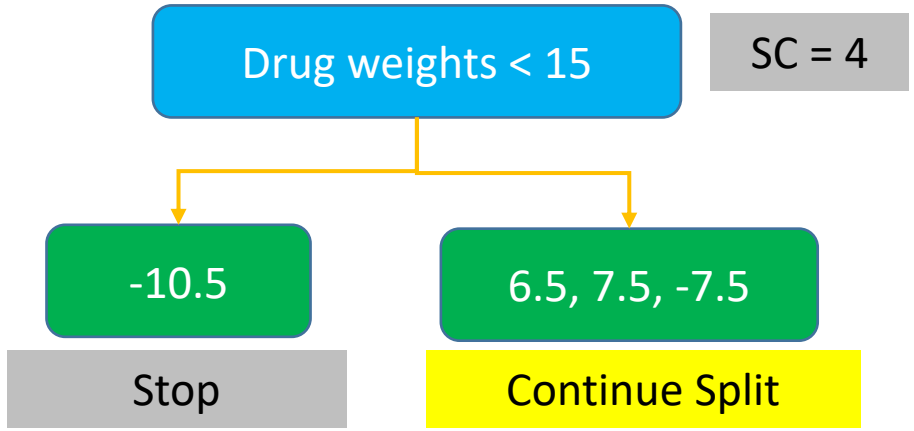
❑ *Step 1*



XGBoost For Regression

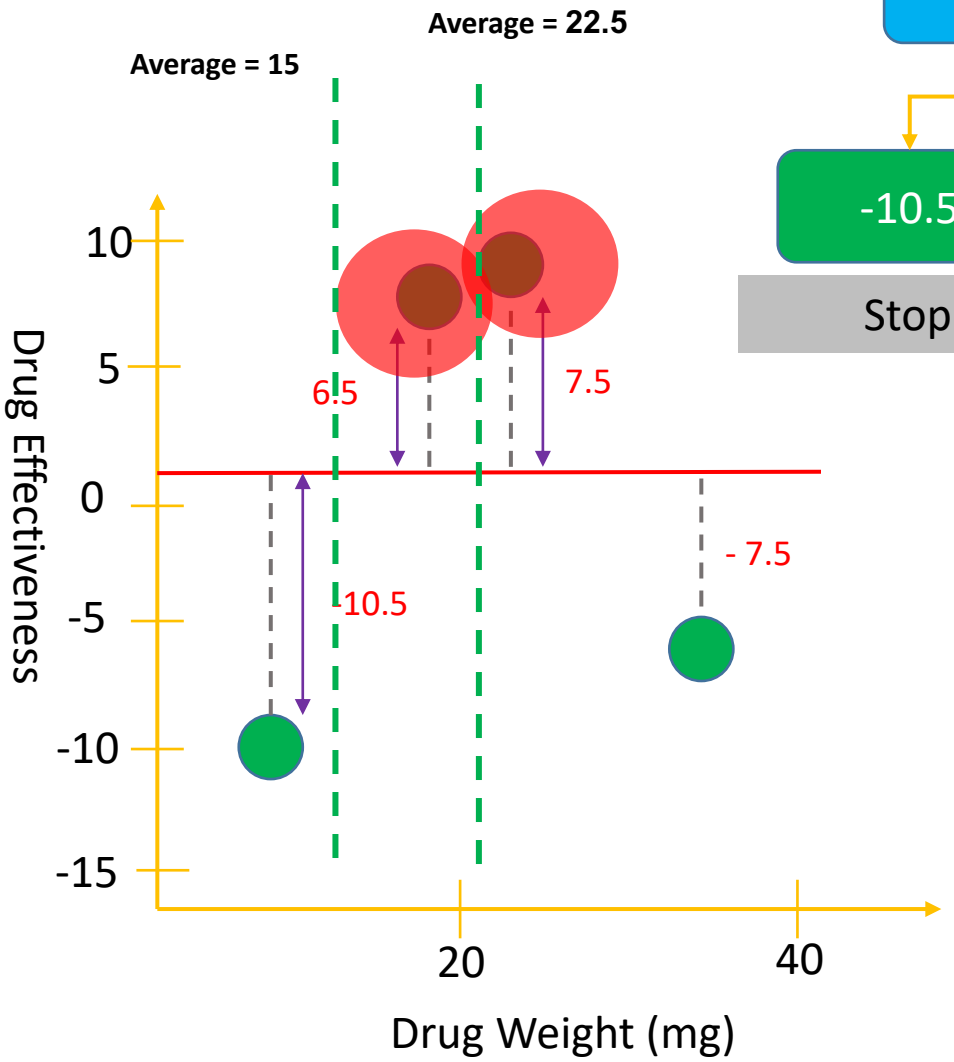
❑ *Step 1*

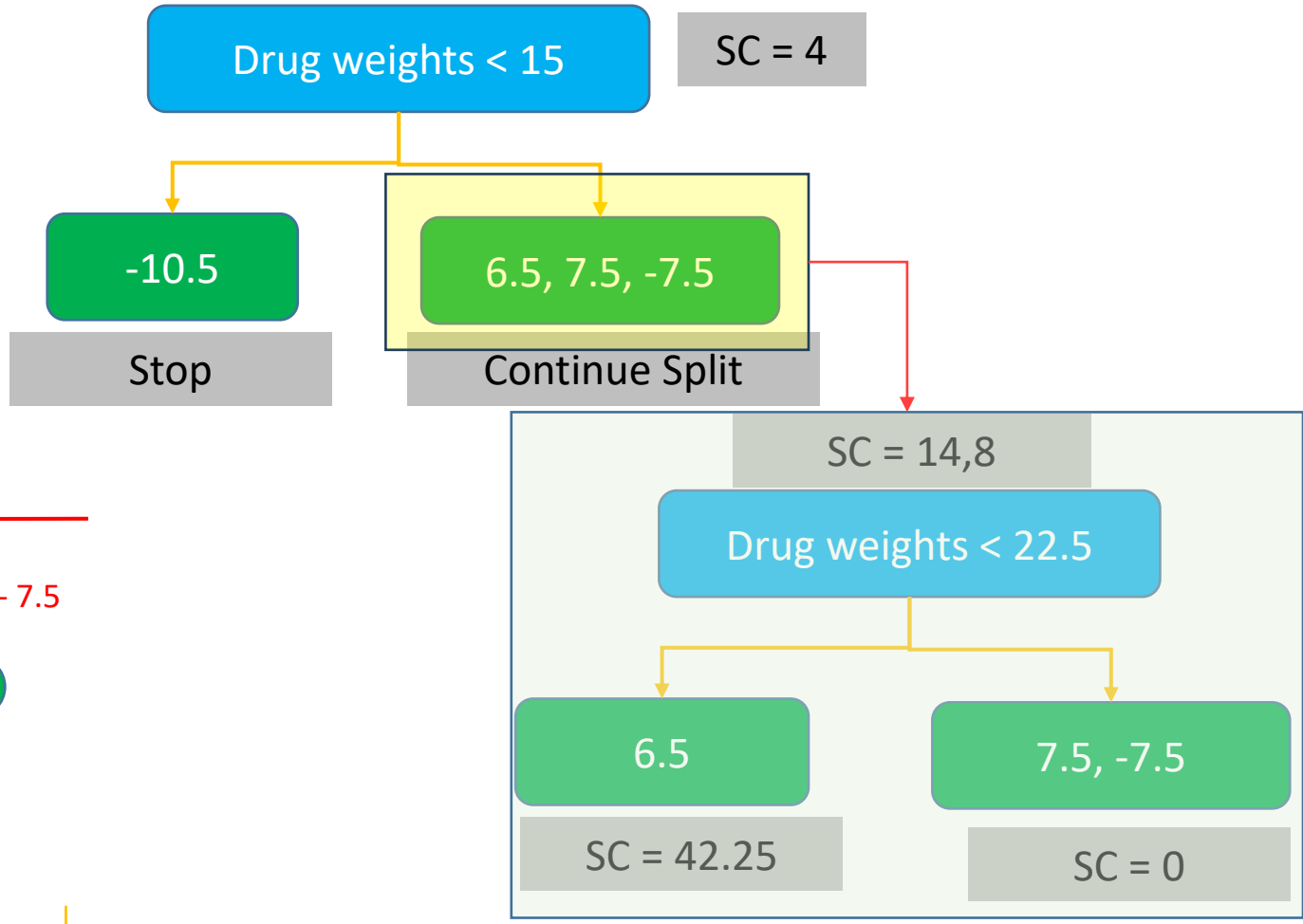




XGBoost For Regression

❑ *Step 1*

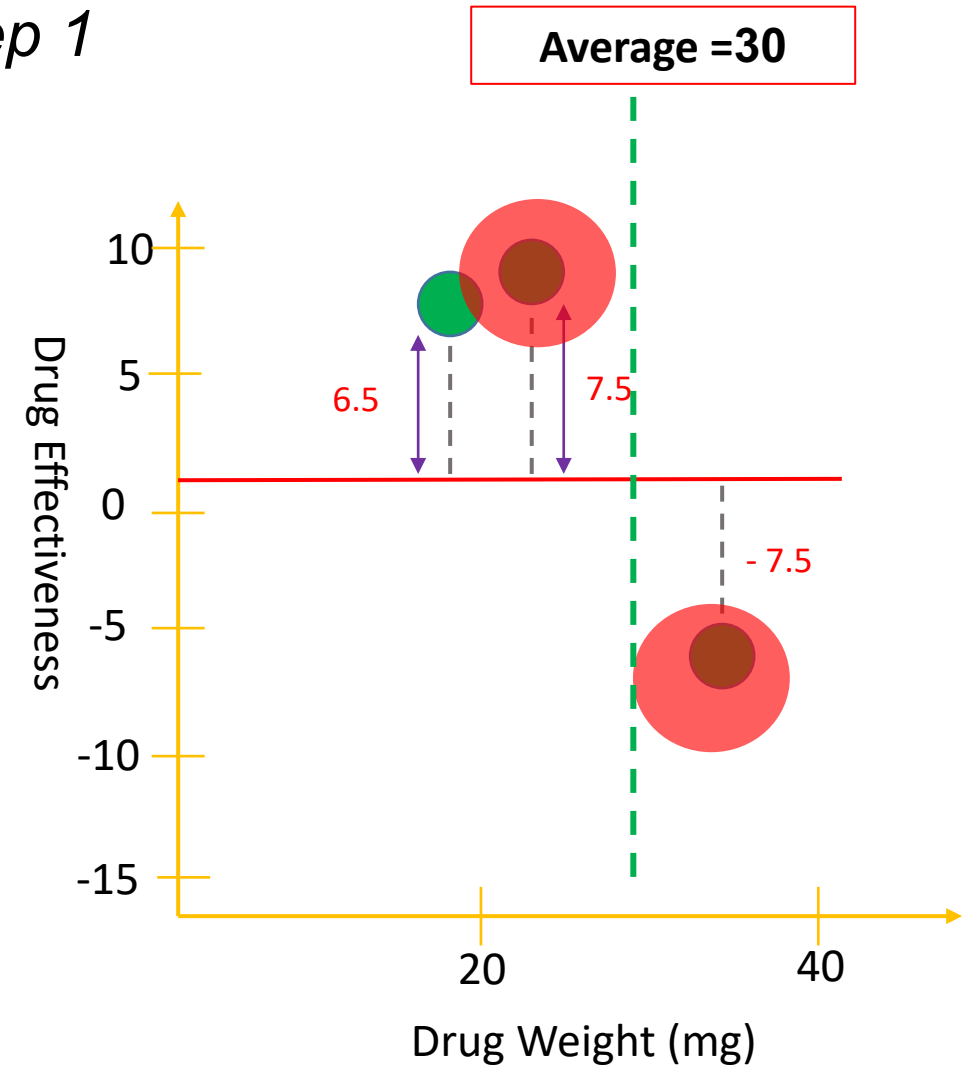


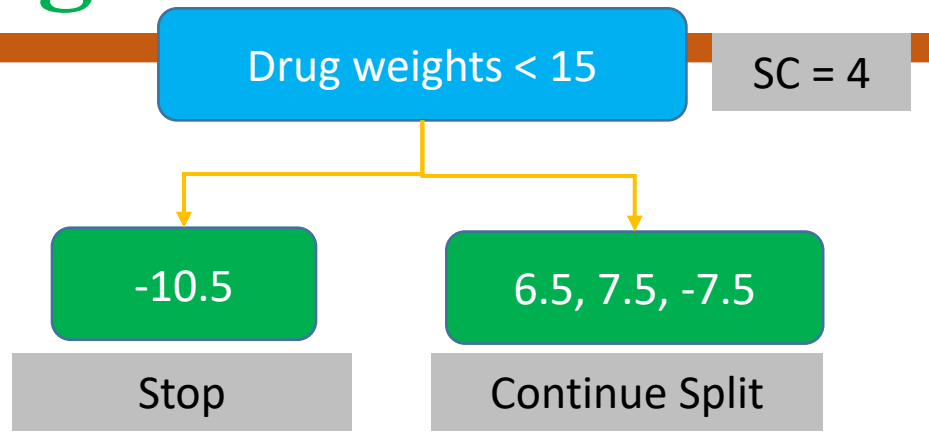


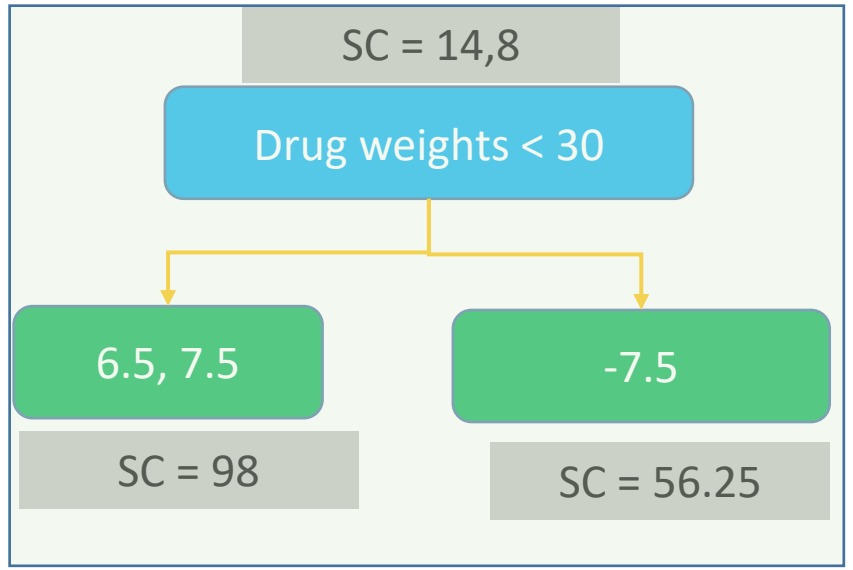
Caculate the Gain
Gain = Left SC + Right SC - Root SC
Gain = 28.17

XGBoost For Regression

❑ *Step 1*



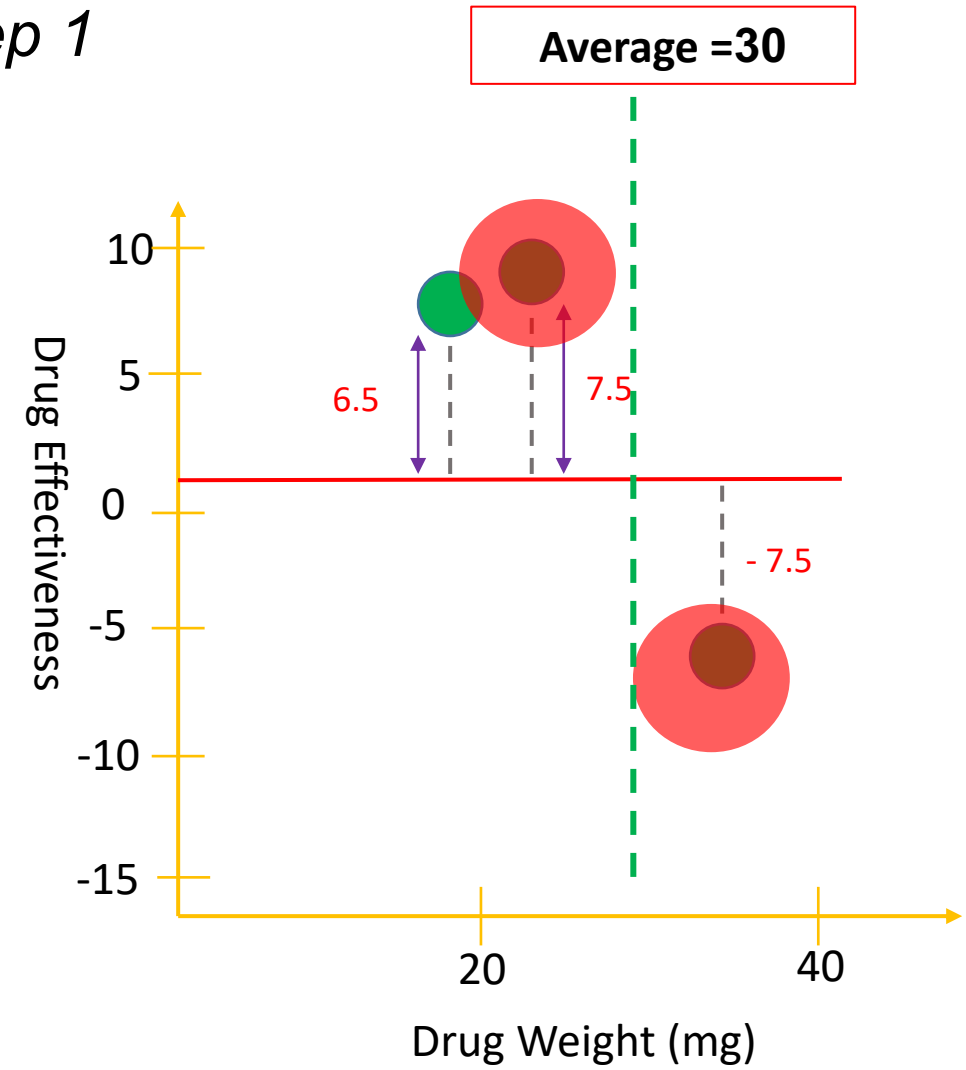


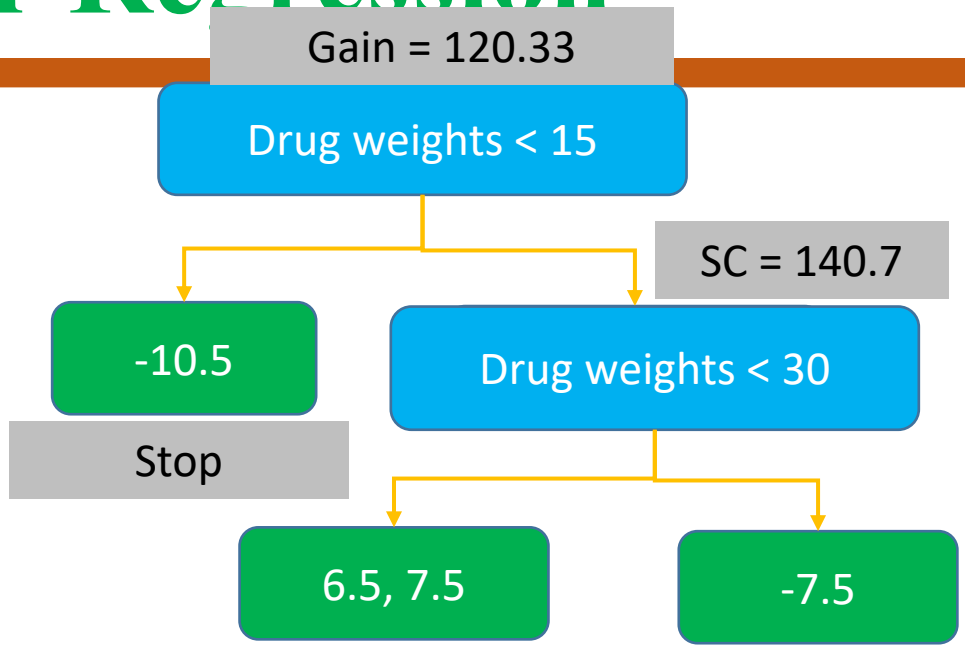


Caculate the Gain
Gain = Left SC + Right SC - Root SC
Gain = 140.17

XGBoost For Regression

❑ *Step 1*



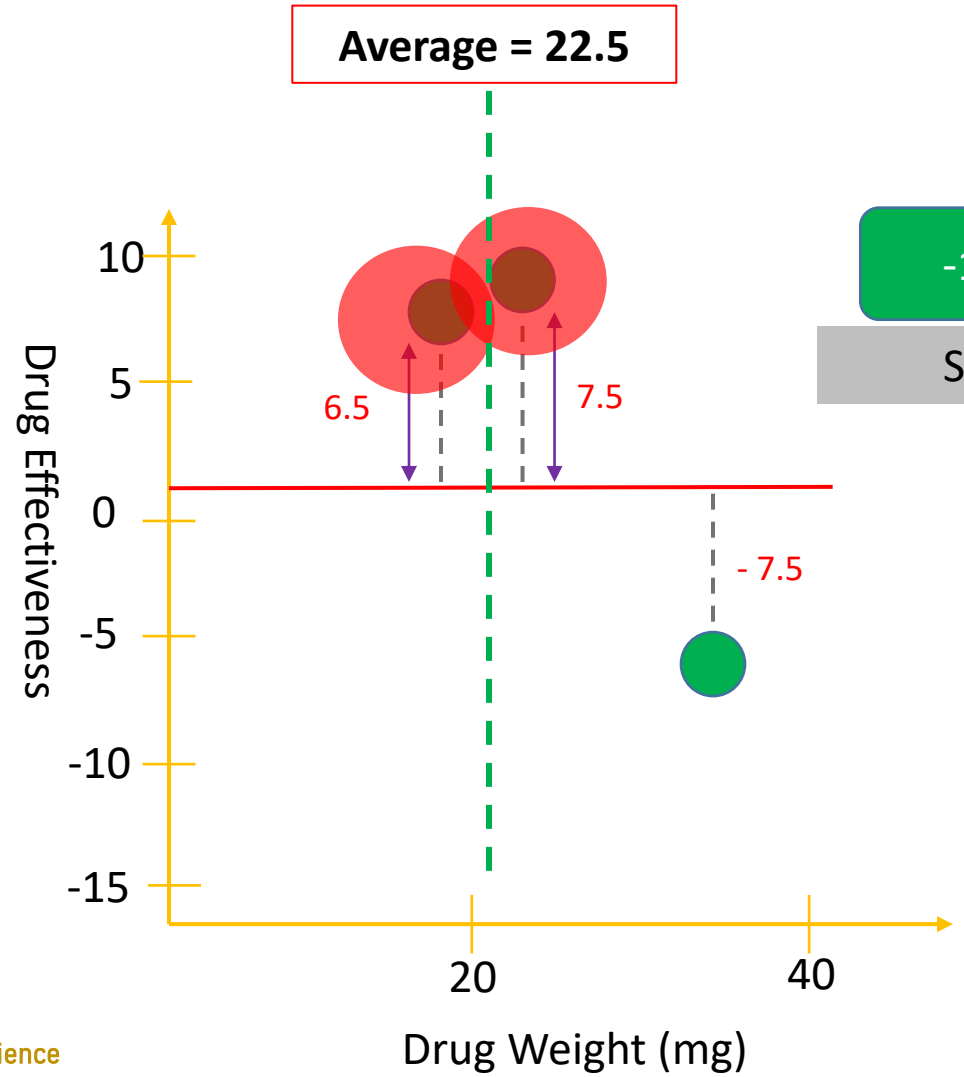


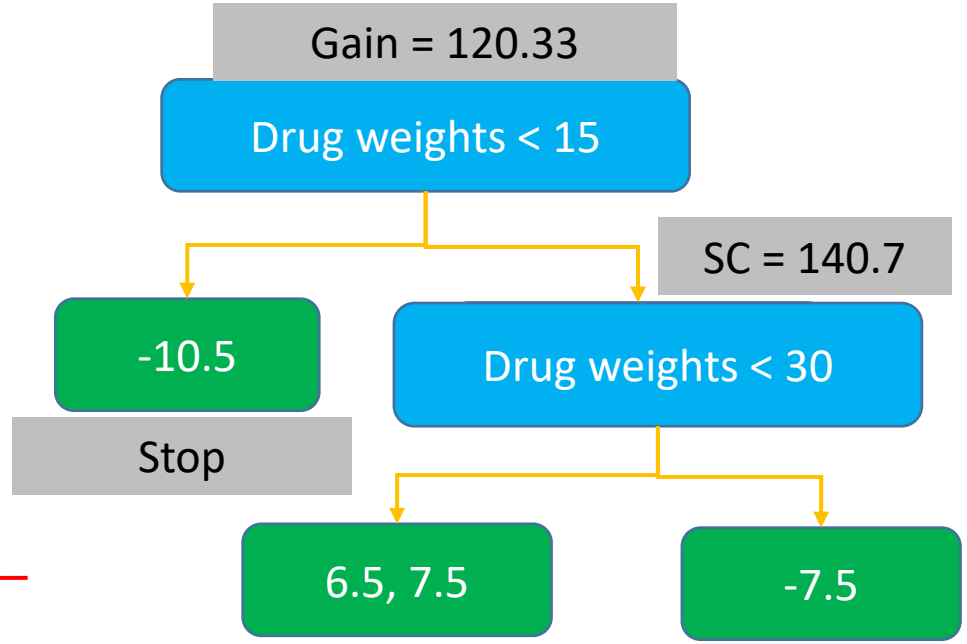
How to prune the tree to prevent Overfitting ? Gain information

$\gamma = 130$

XGBoost For Regression

❑ *Step 1*





How to prune the tree to prevent Overfitting ? Gain information

$\gamma = 130$

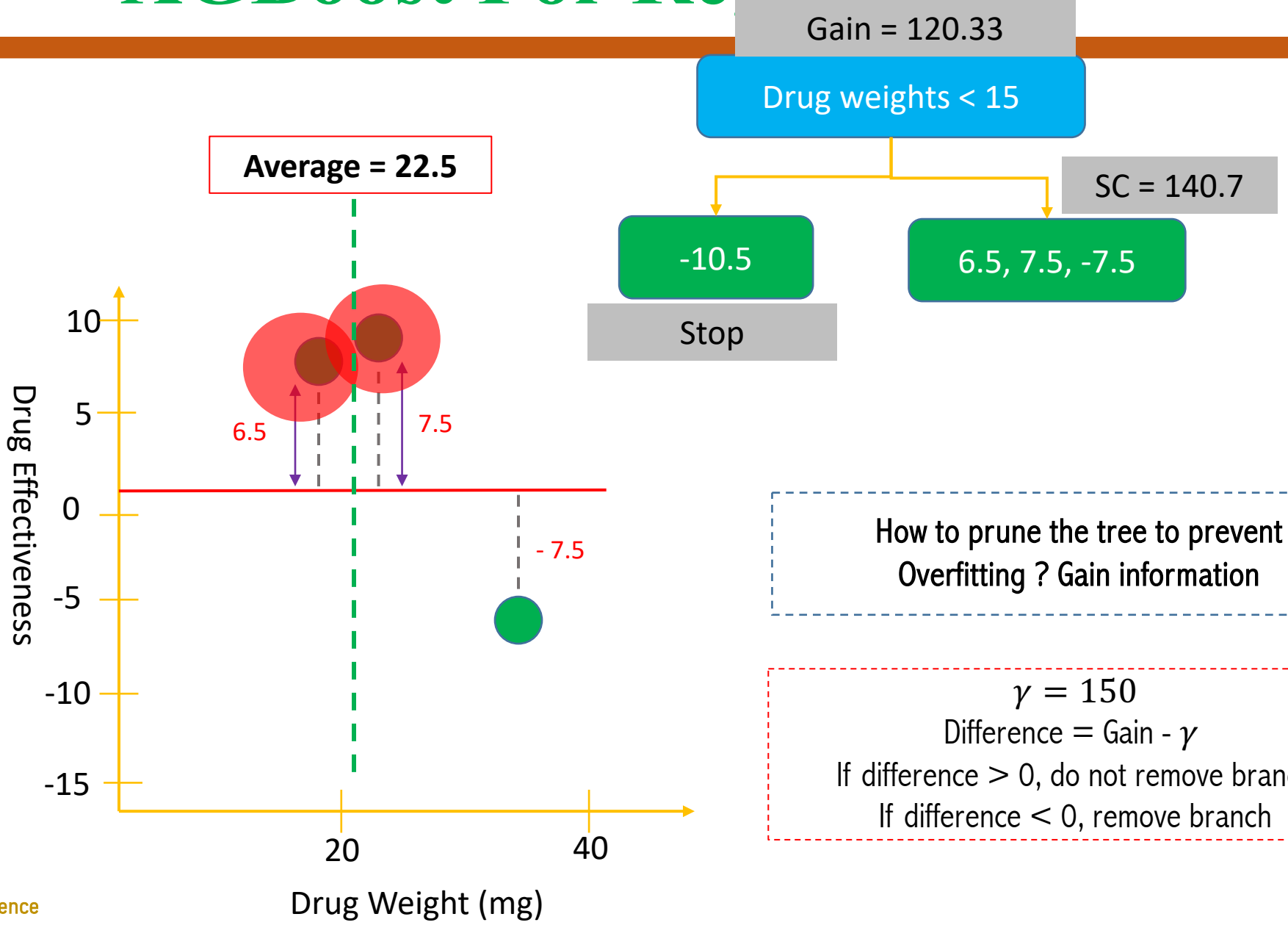
Difference = Gain - γ

If difference > 0, do not remove branch

If difference < 0, remove branch

XGBoost For Regression

❑ *Step 1*



How to prune the tree to prevent Overfitting ? Gain information

$\gamma = 150$

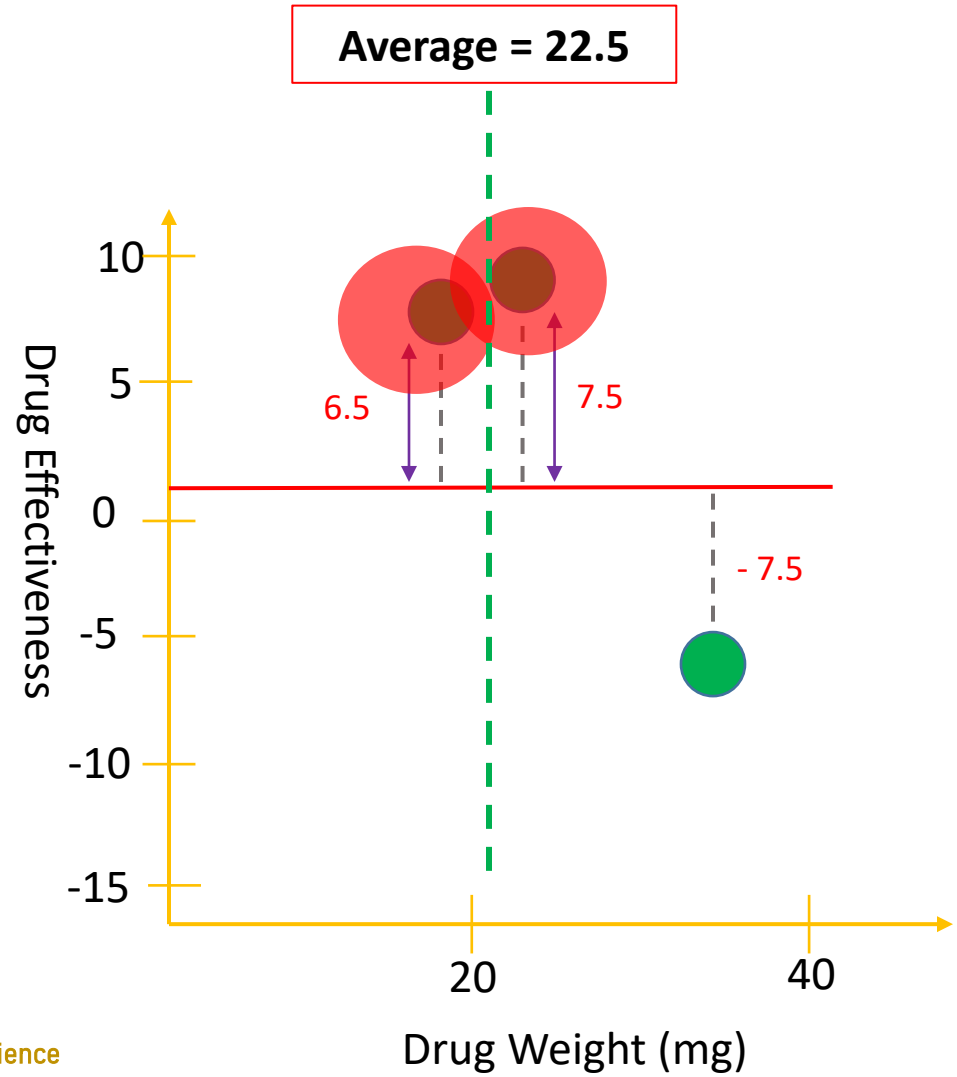
Difference = Gain - γ

If difference > 0, do not remove branch

If difference < 0, remove branch

XGBoost For Regression

❑ *Step 1*



0.5

How to prune the tree to prevent Overfitting ? Gain information

$\gamma = 150$

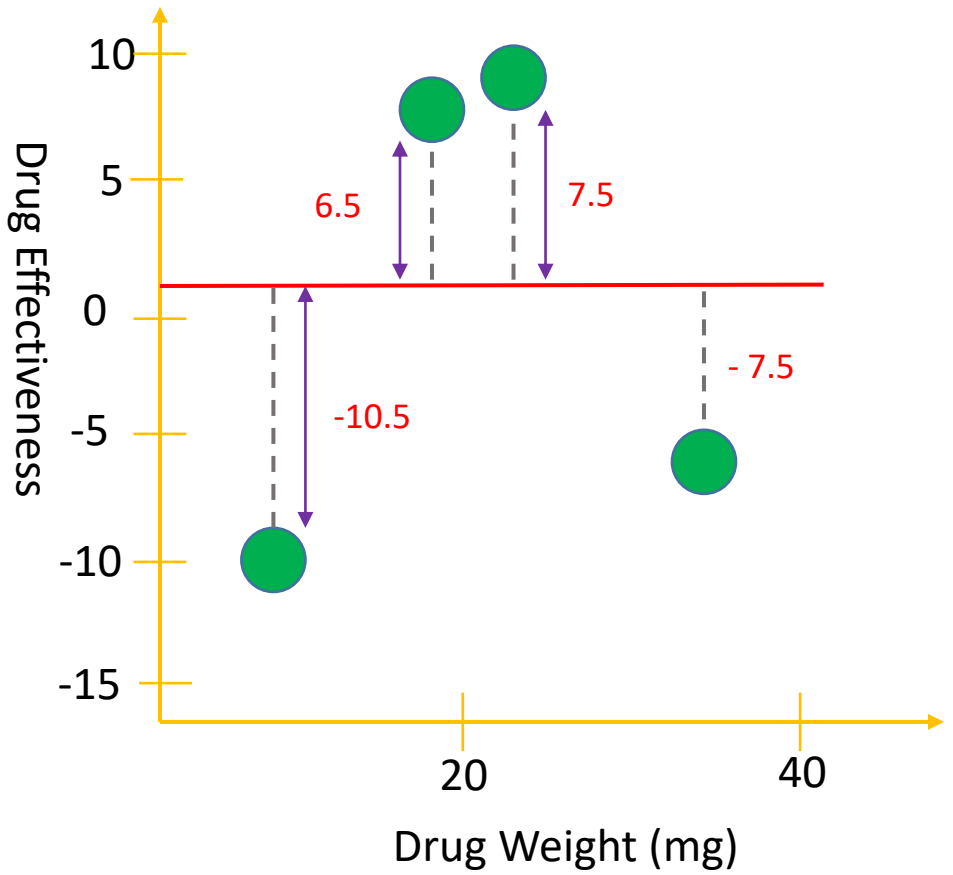
Difference = Gain - γ

If difference > 0, do not remove branch

If difference < 0, remove branch

XGBoost For Regression

❑ *Step 1*



Start with single Leaf of residuals

-10.5, 6.5, 7.5, -7.5

m = 4
λ = 1

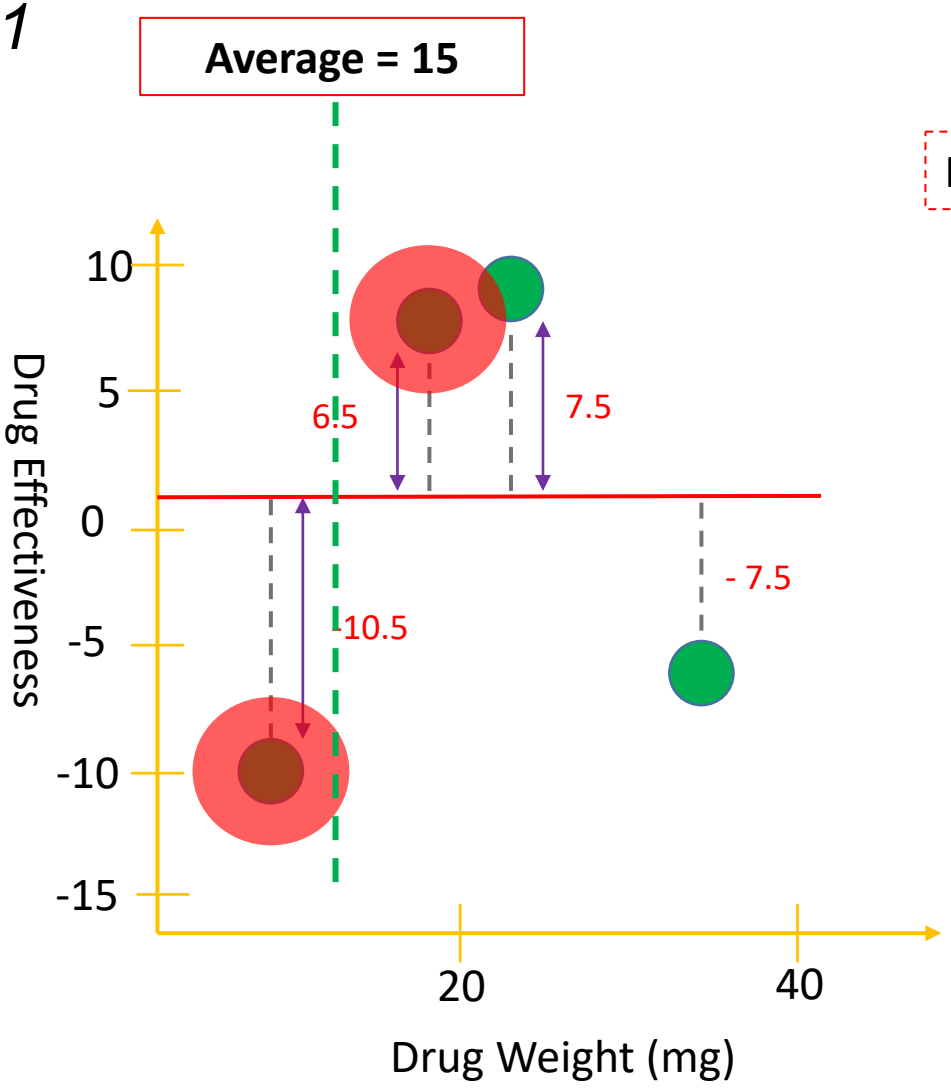
Compute Similarity Score

$$SC = \frac{\sum (output - predicted)^2}{m + \lambda}$$

$$SC = \frac{[-10.5+7.5+6.5+(-7.5)]^2}{4+1} = 3.2$$

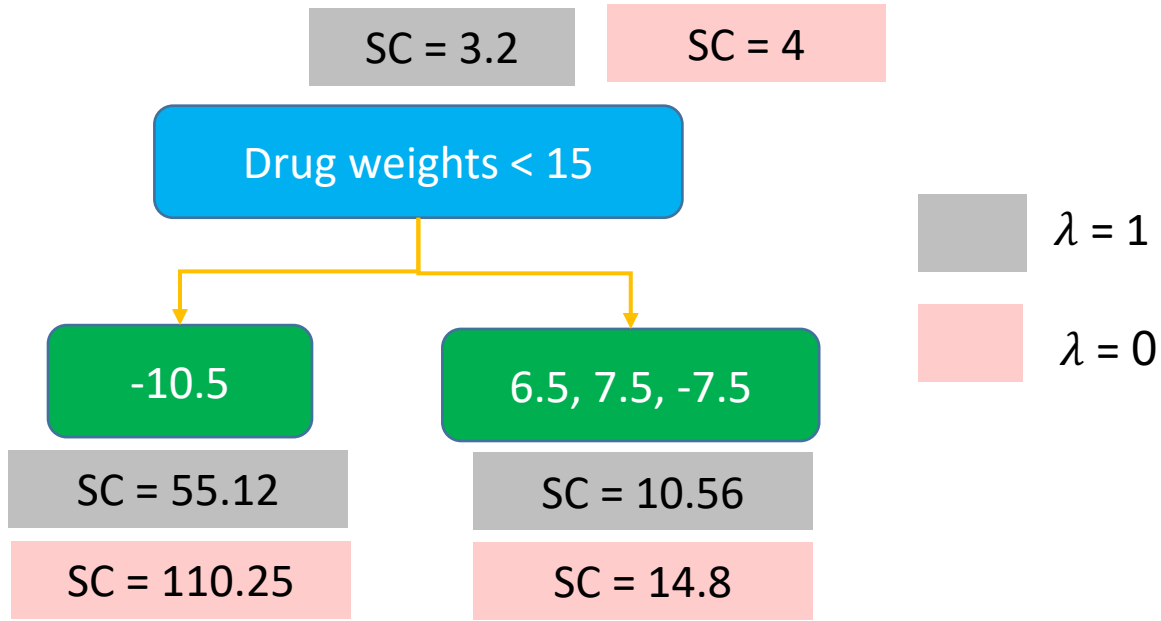
XGBoost For Regression

❑ *Step 1*



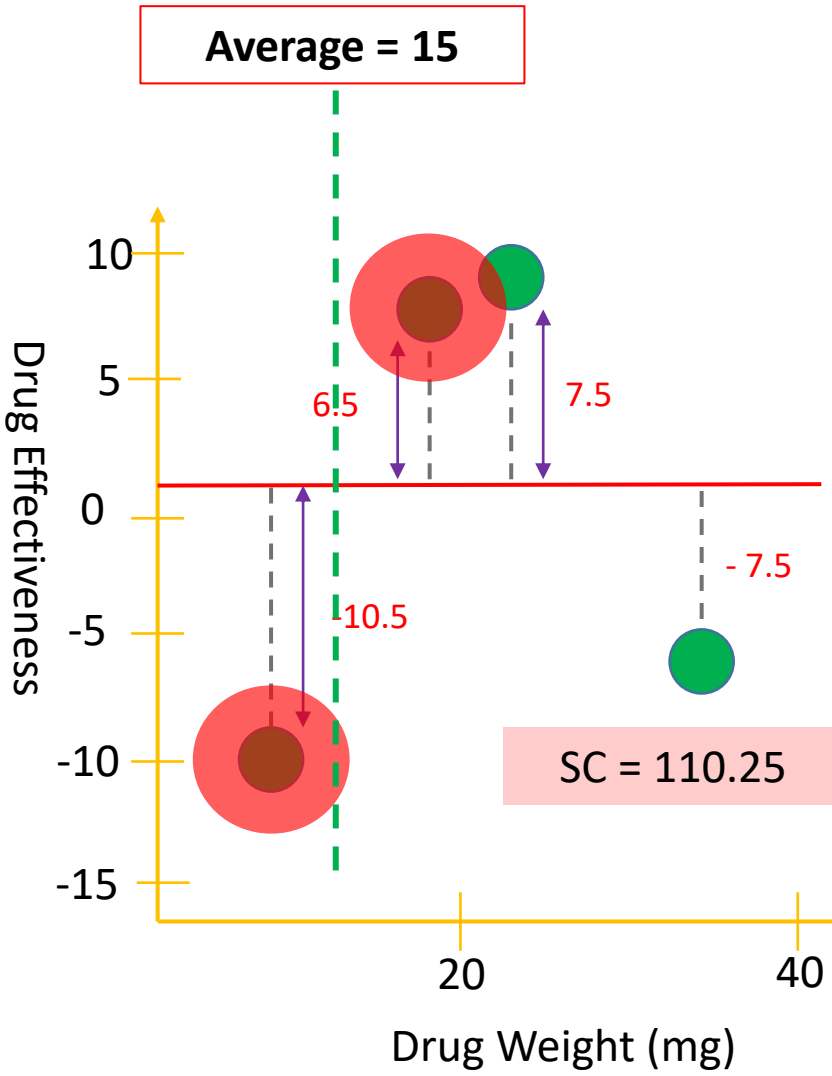
SC = 3.2 | -10.5, 6.5, 7.5, -7.5

Please look at the two outputs with lowest drug weights



When $\lambda > 0$, the similarity score are smaller
Inversely proportional to the number of residuals

XGBoost For Regression



$\lambda = 1$ $\lambda = 0$

Gain = 55.12 + 10.56 – 3.2 = 62.48

Gain = 55.12 + 10.56 – 4 = 120.33

Drug weights < 15

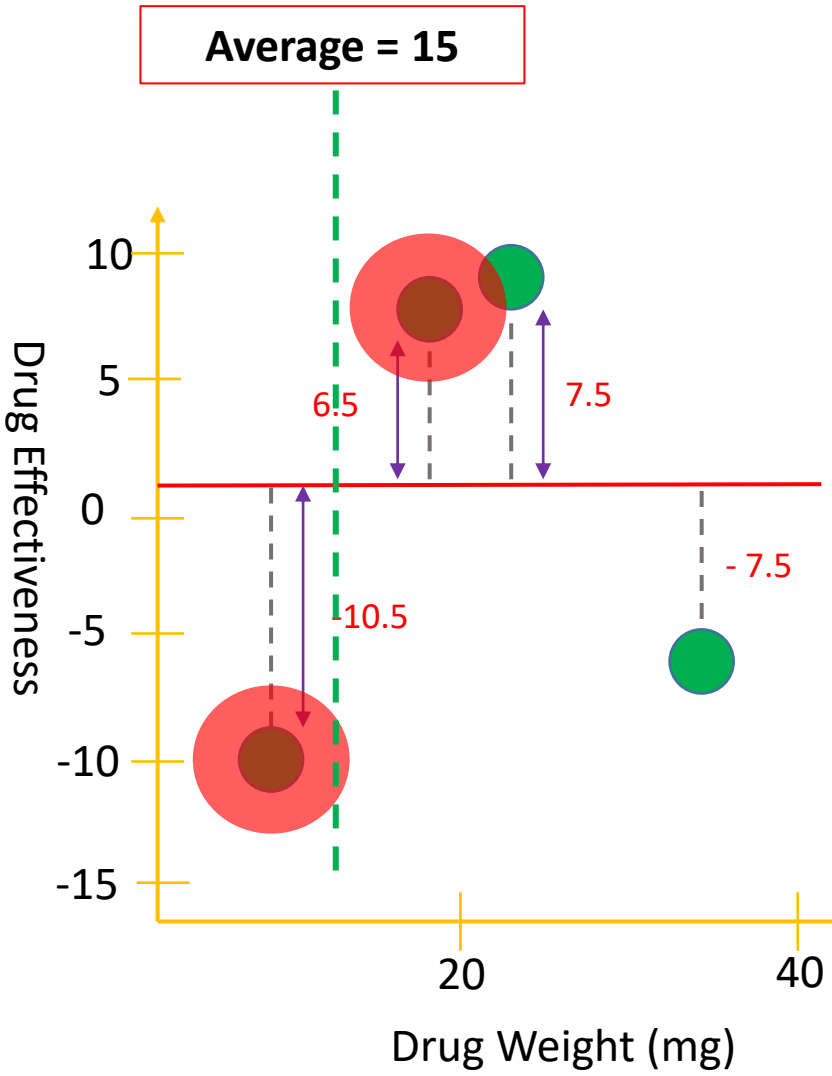
Gain = 82.9 Gain = 140.17

-10.5 6.5, 7.5, -7.5

SC = 110.25 SC = 55.12 SC = 10.56 SC = 14.8

The amount of decrease is invertly propotional to the number of Residual in the nodes

XGBoost For Regression



$\lambda = 0$

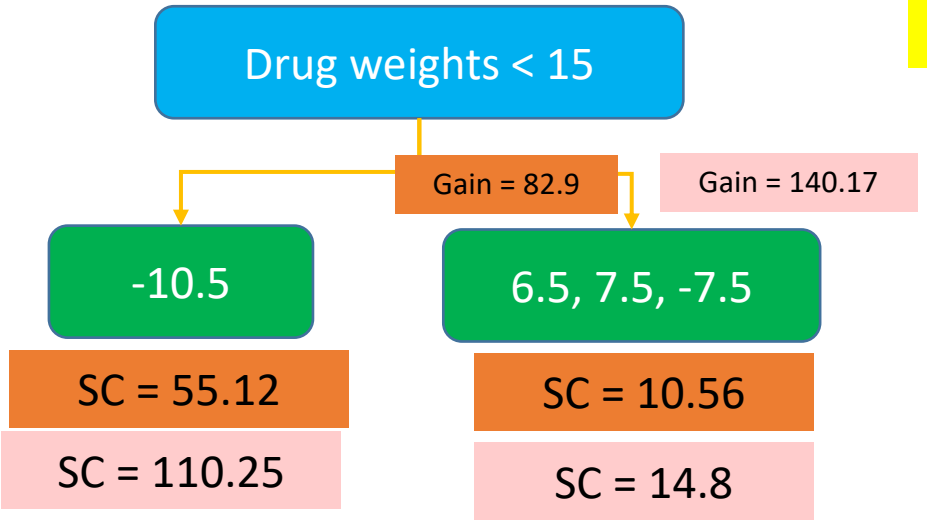
$\lambda = 1$

$\lambda > 0$: easy to prune the tree
Prevent overffiting

Gain = 55.12 + 10.56 – 4 = 62.48

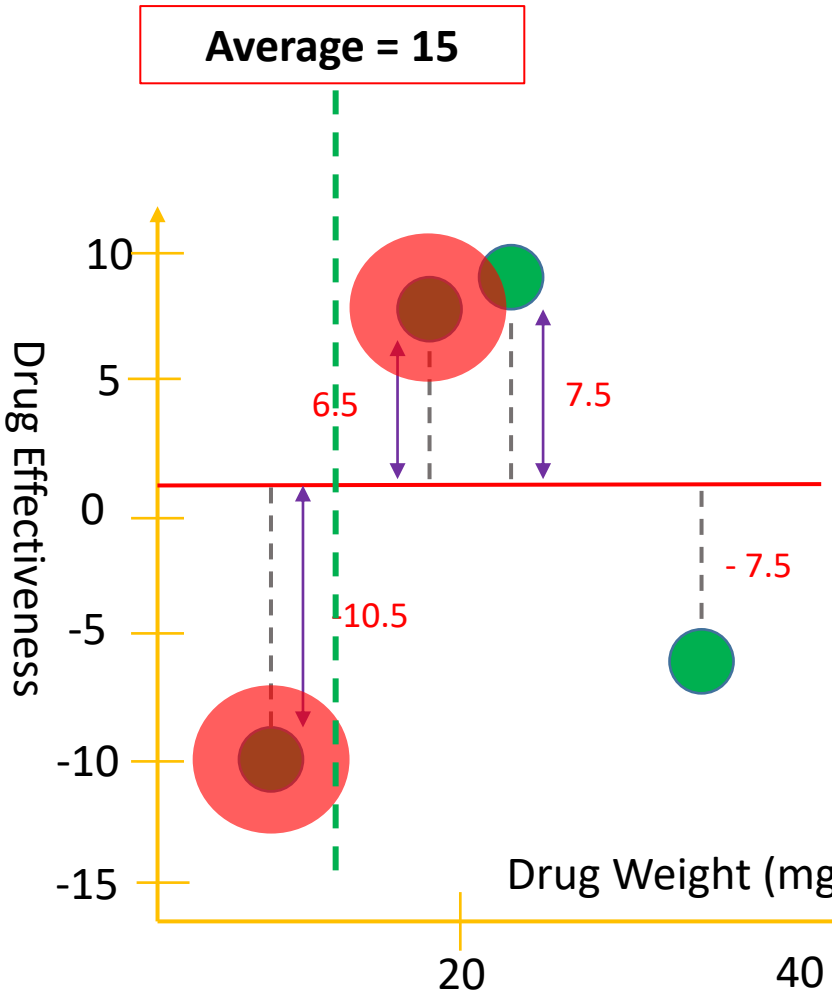
Gain = 55.12 + 10.56 – 4 = 120.33

Prunning parameter: $\gamma = 130$



The amount of decrease is invertly propotional to the number of Residual in the nodes

XGBoost For Regression



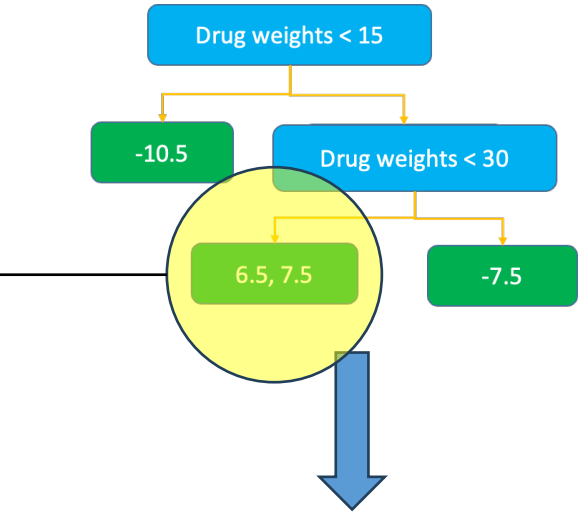
Setting $\gamma = 0$ do not turn off prunning

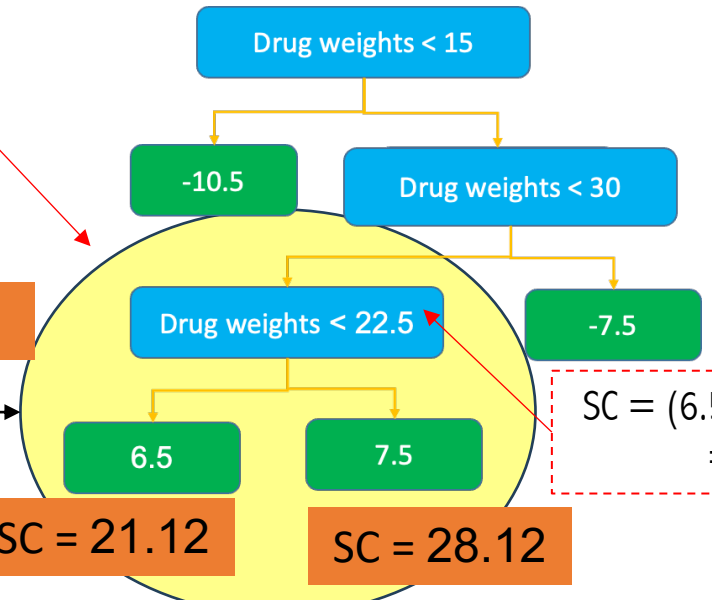
We will remove this branch: -16.06- $\gamma < 0$

Prunning parameter: $\gamma = 0$

$\lambda = 1$

SC = 65.3

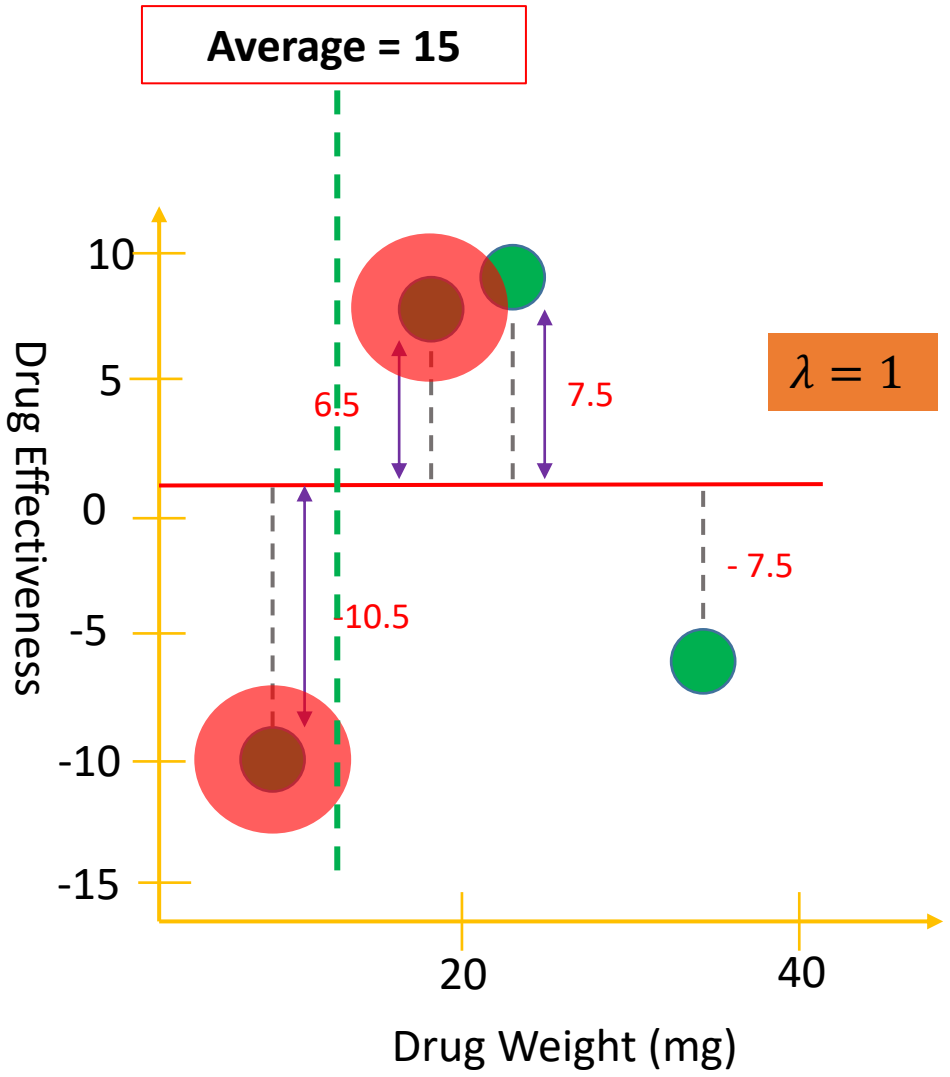




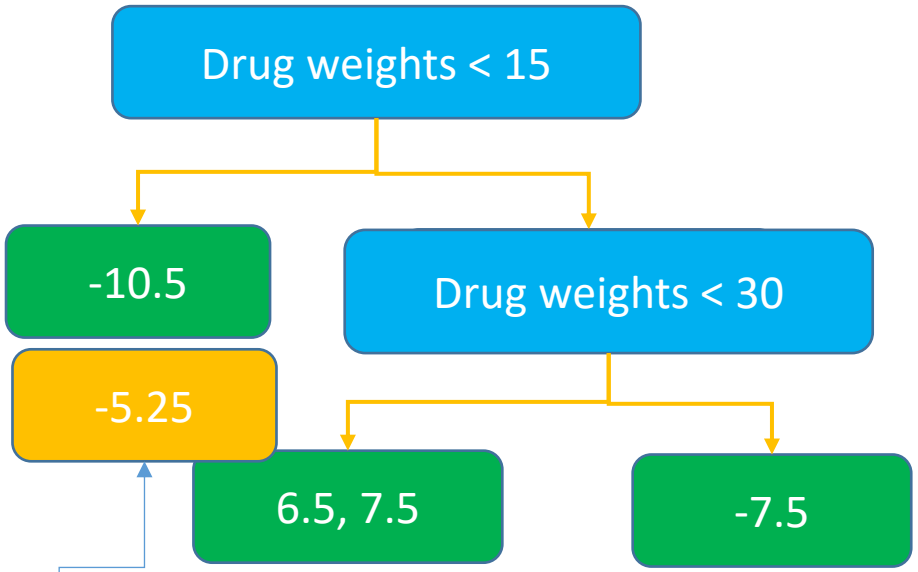
SC = $(6.5+7.5)^2/(2+1)$ = 65.3

Gain = 21.12 + 28.12 – 65.3 = -16.06

How to Predict Value



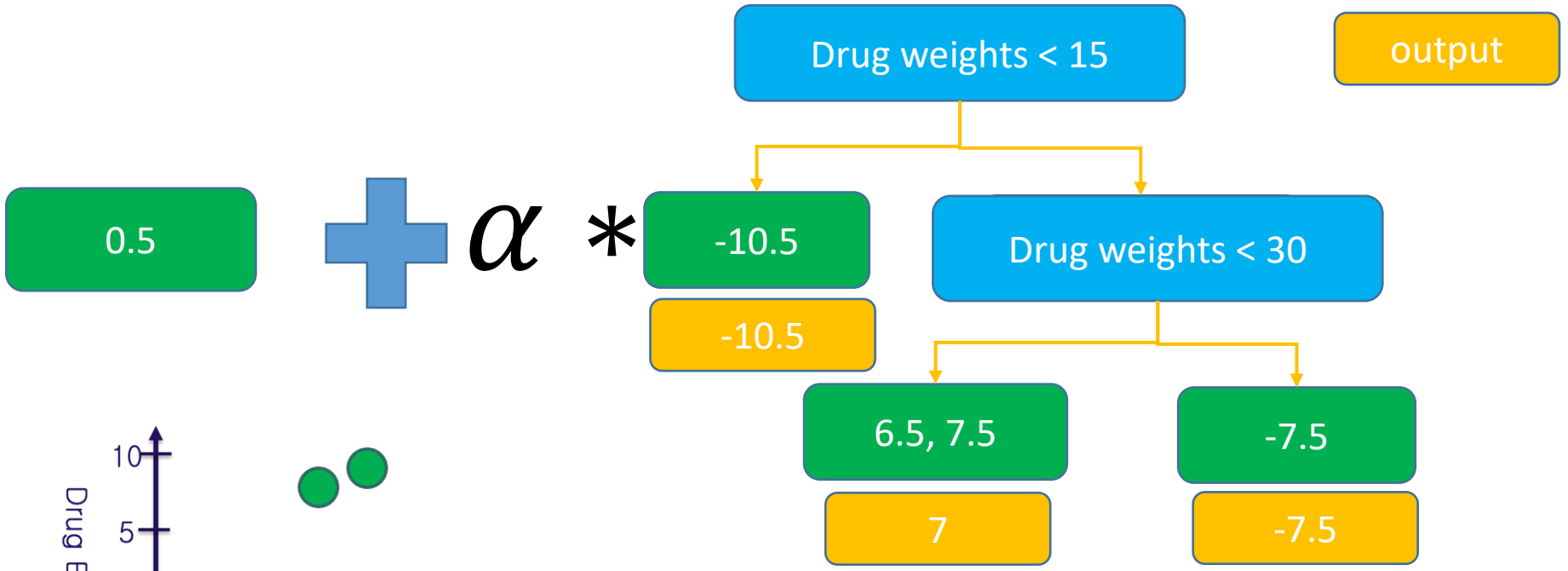
$\lambda = 1$

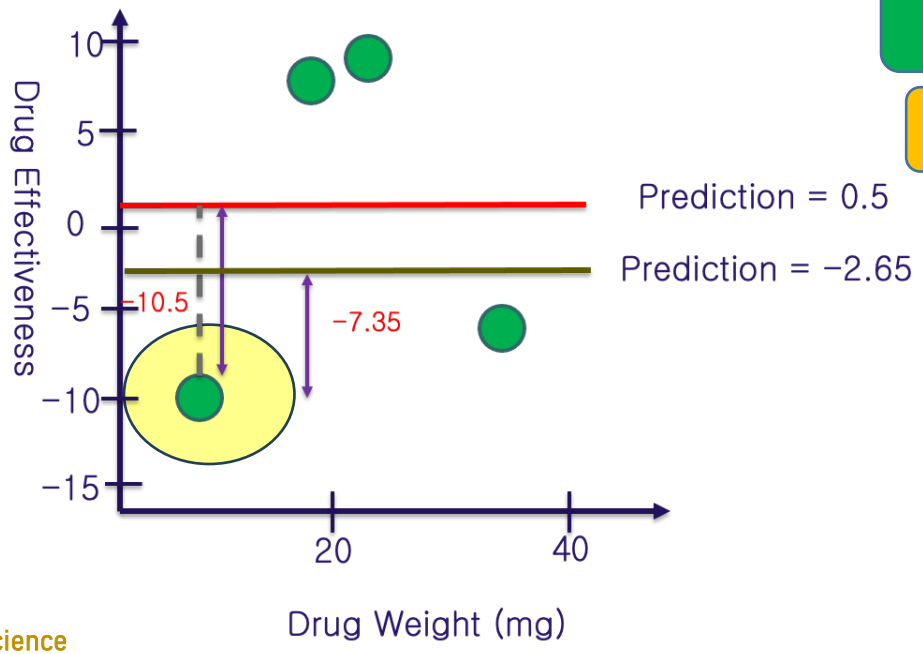


$$\text{Output Value} = \frac{\text{Sum of Residuals}}{\text{Number of Residuals} + \lambda}$$

When $\lambda > 0$, it will reduce the amount that this indiviual observation add to the overal prediction

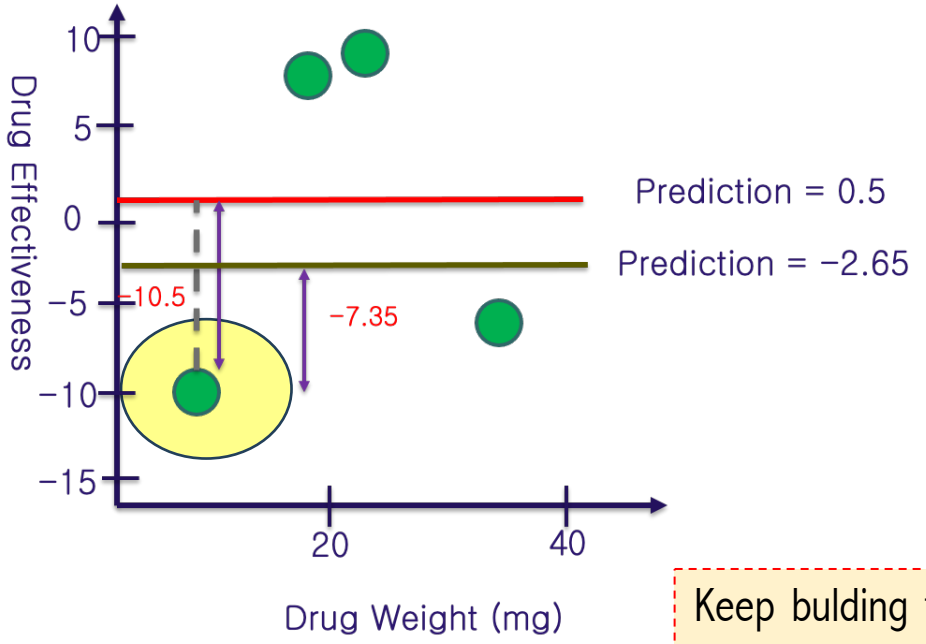
How to Predict Value

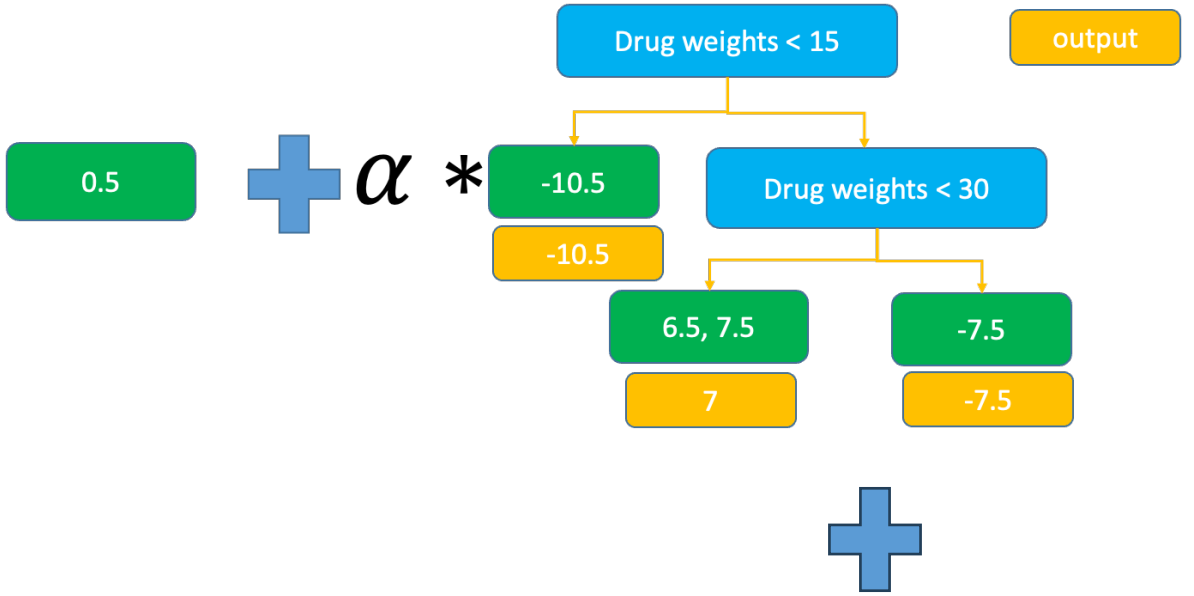




$\lambda = 0$ $\alpha = 0.3$

Building the Next Tree

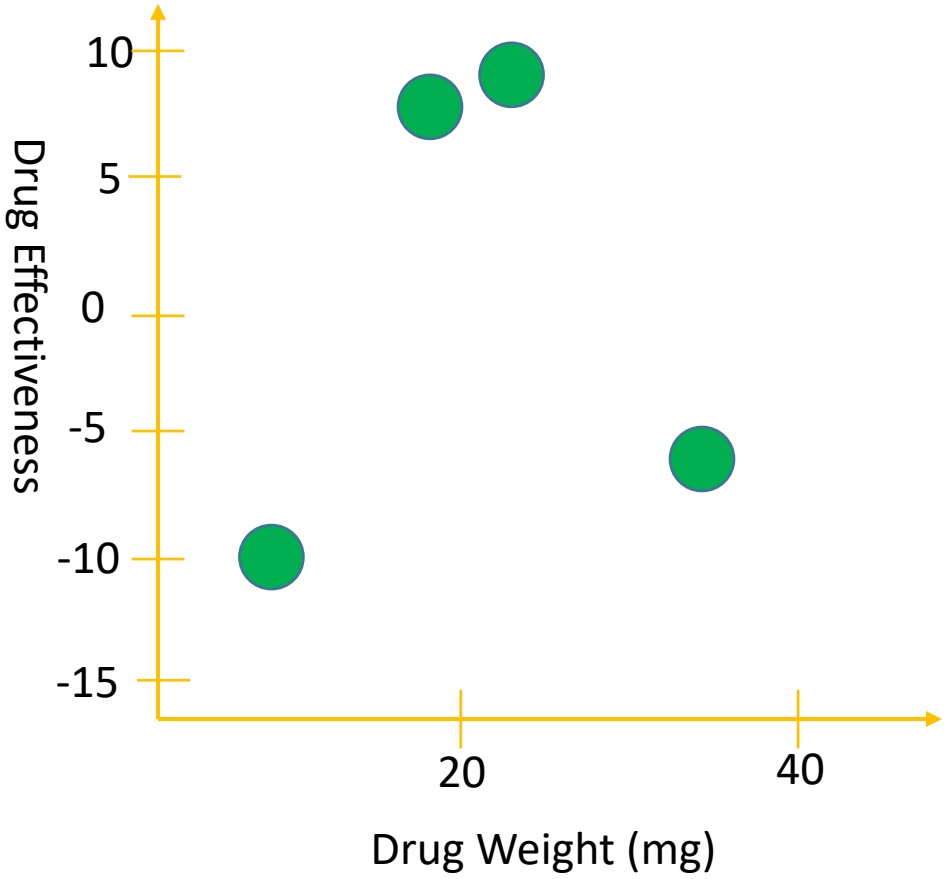


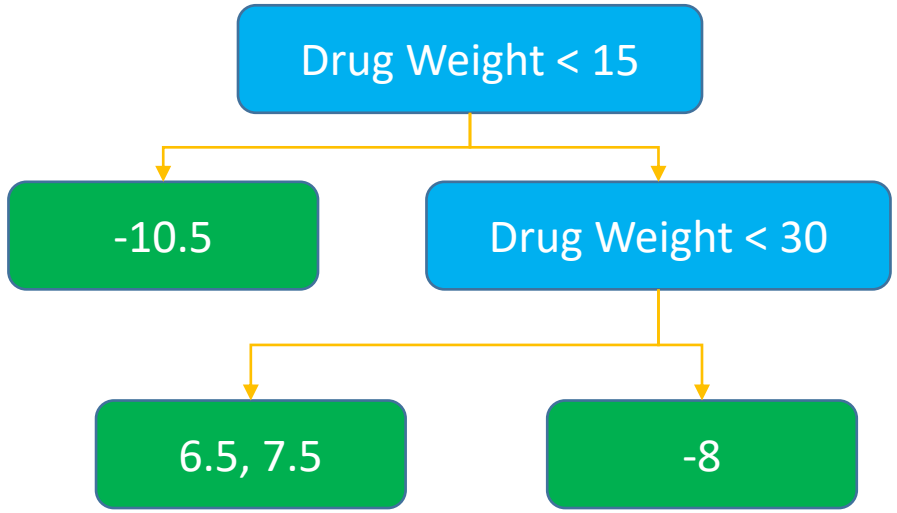


$\alpha *$ Next Tree Result

Keep bulding the Tree until the Residual are reach the predefined threshold. Or we reach to the maximum number of Tree

XGBoost for Regression



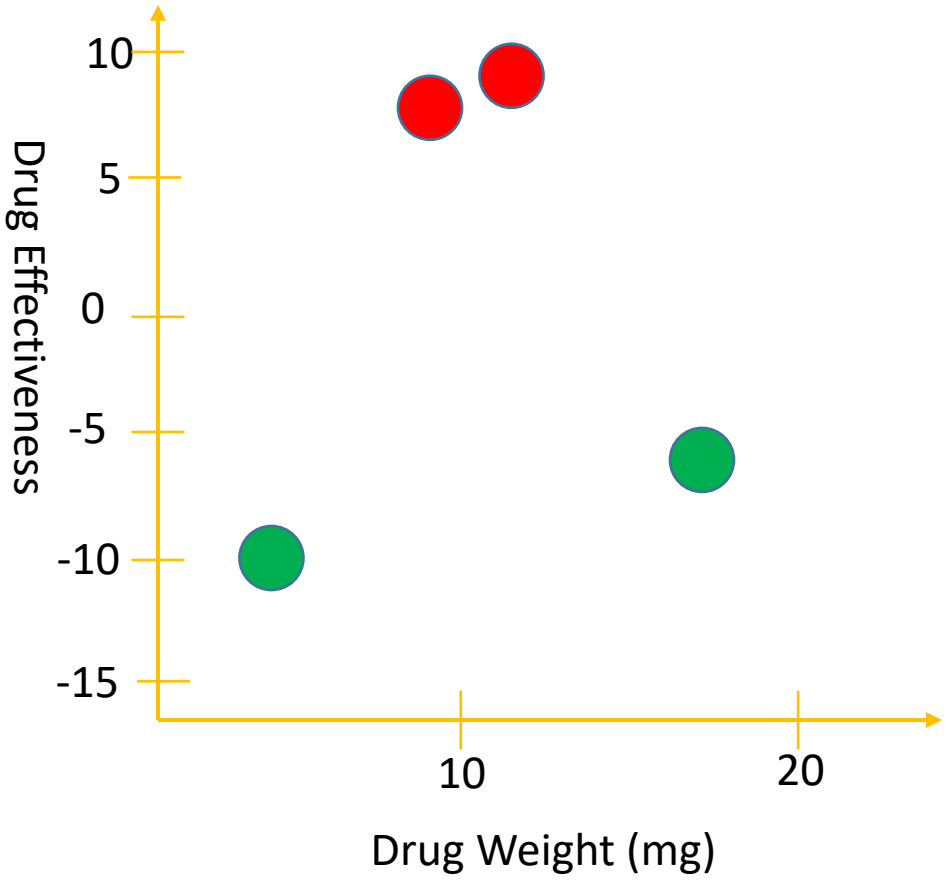


HOW TO FIND QUANTILES? => QUANTILE SKETCH APPROXIMATE SOLUTION

Outline

- Regularization
- XGBoost For Regression
- XGBoost For Classification
- XGBoost: Mathematical Explanation
- How To Fill Missing Values
- Example

XGBoot For Classification



Not effectiviness

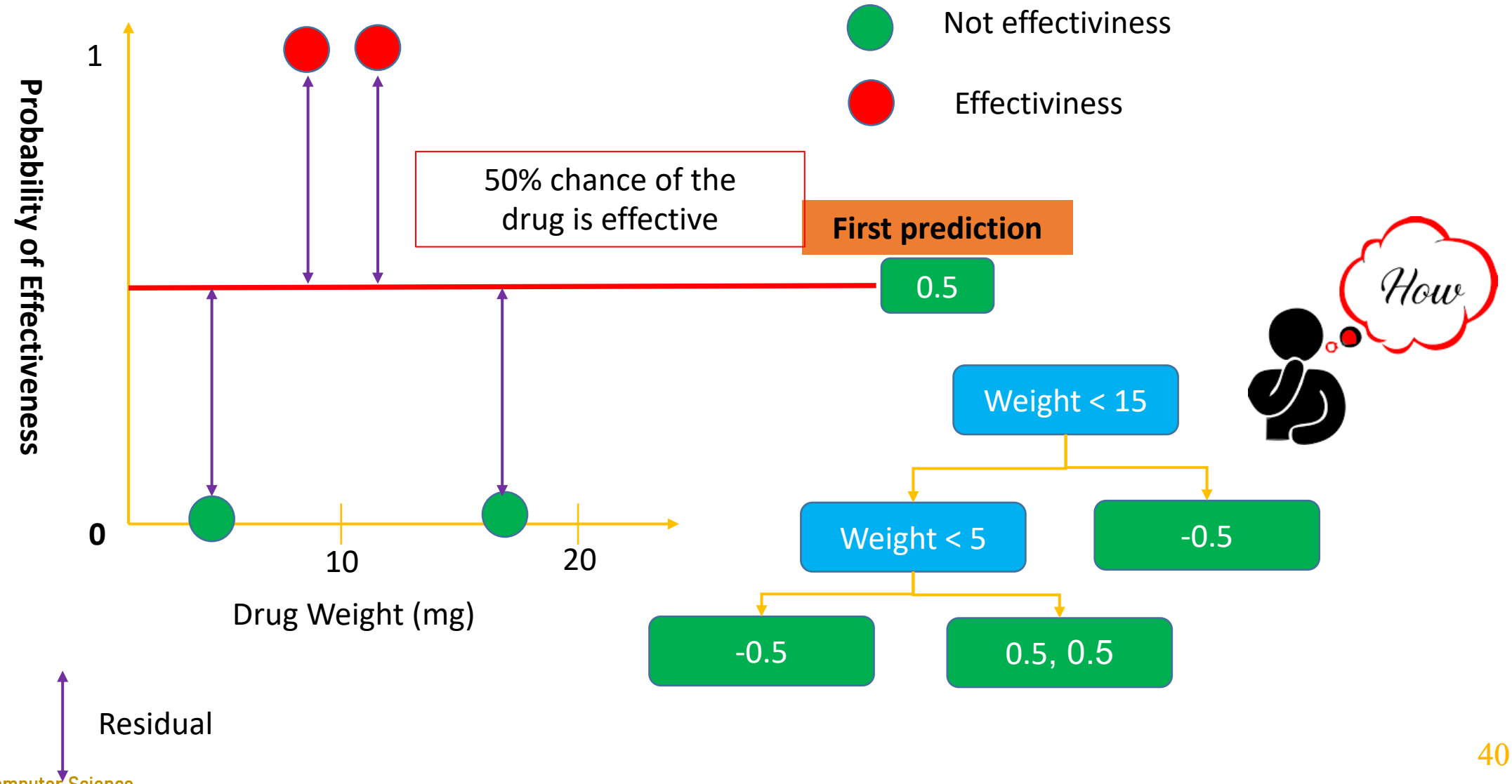
Effectiviness

First prediction

0.5

50% chance of the drug is effective

XGBoot For Classification



XGBoot For Classification

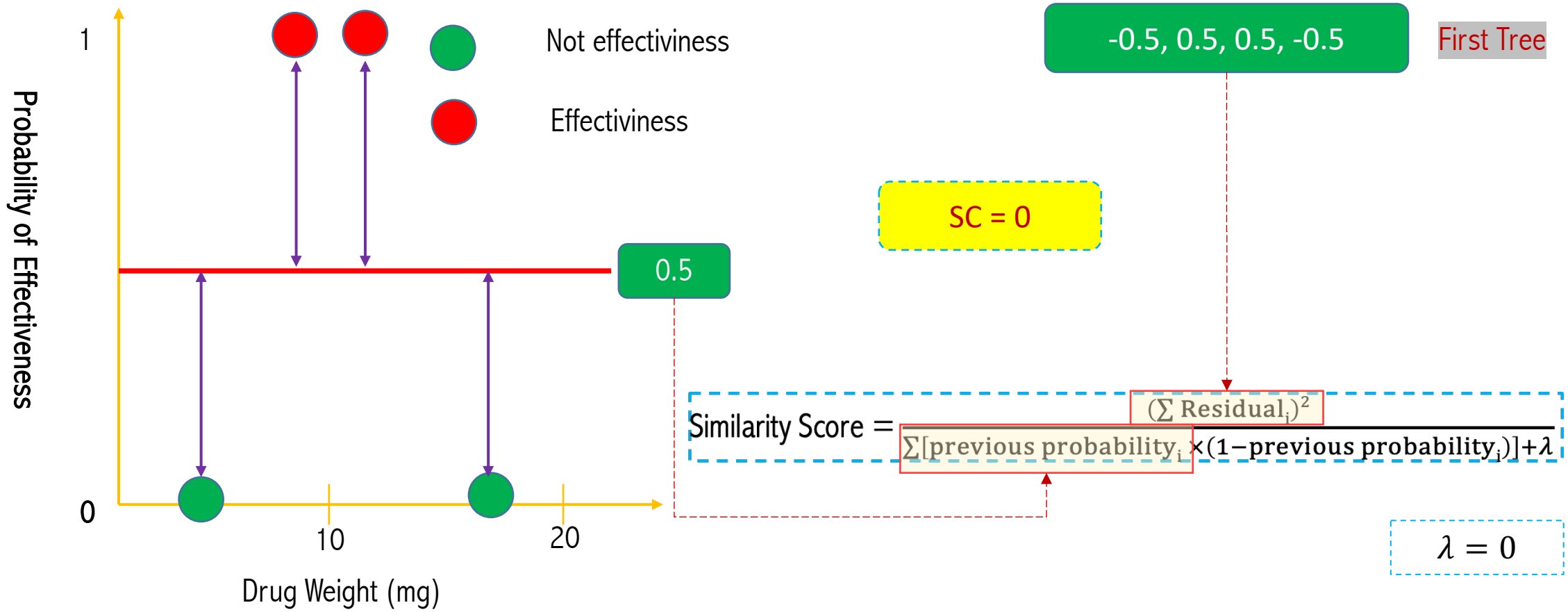
Similarity Score for Classification:

$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum[\text{previous probability}_i \times (1 - \text{previous probability}_i)] + \lambda}$$

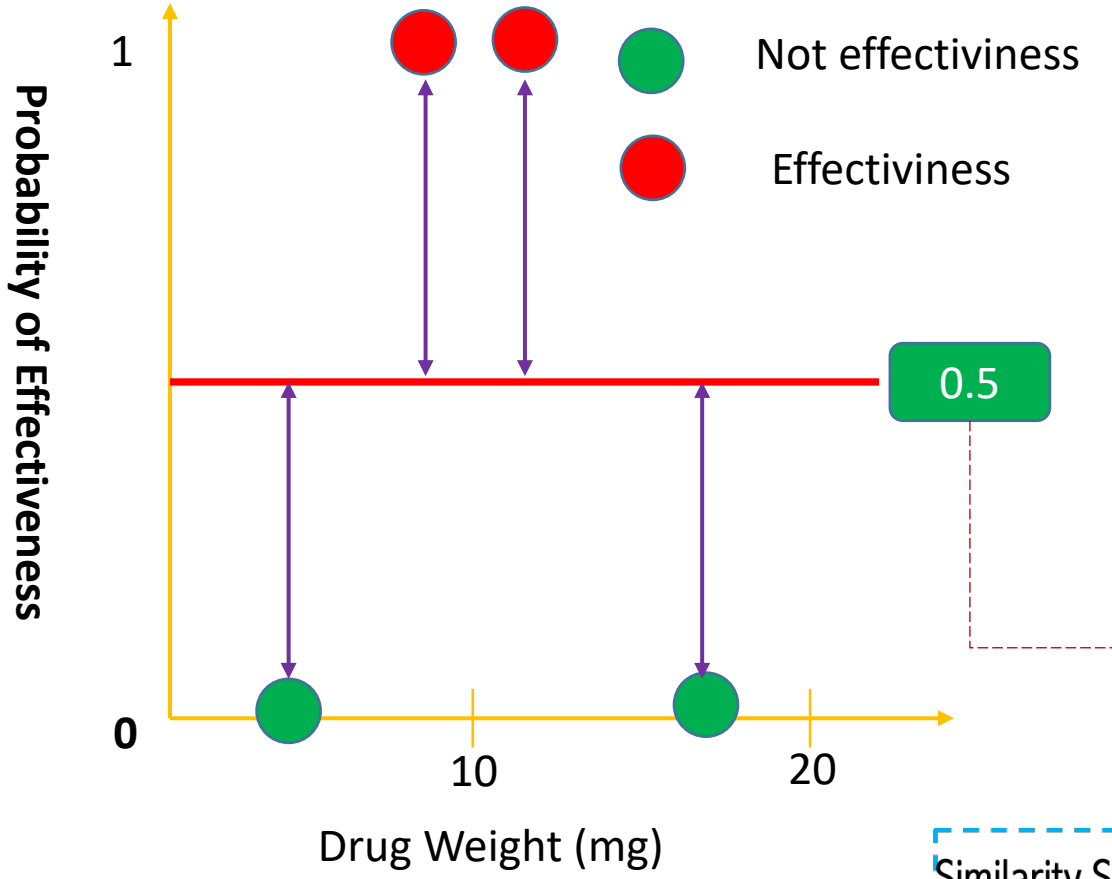
Similarity Score for Prediction (regression):

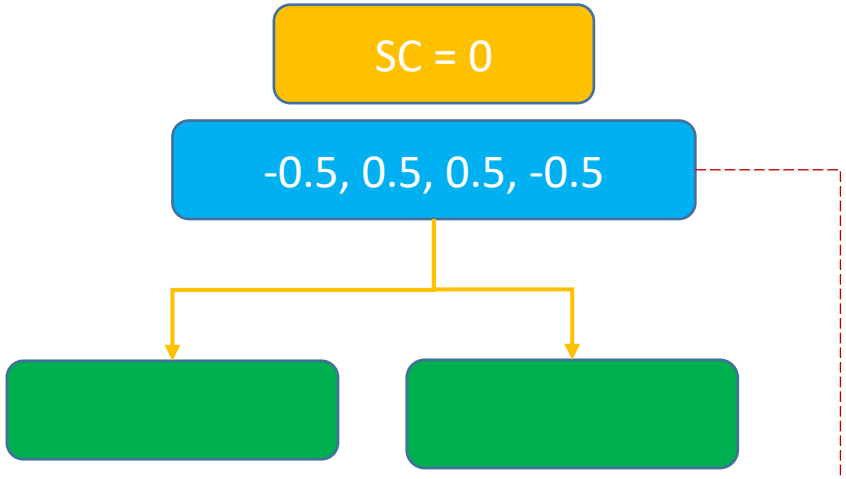
$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\text{number of residual} + \lambda}$$

XGBoost for Classification



XGBoost For Classification

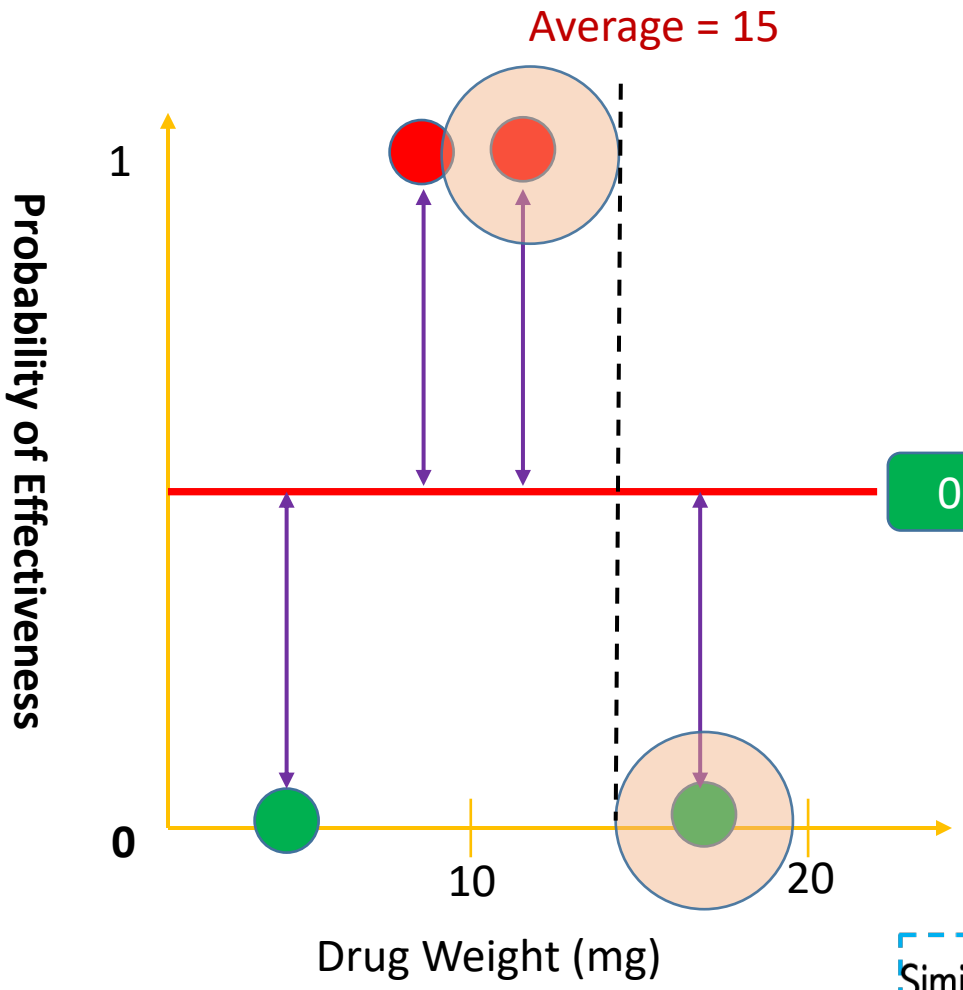


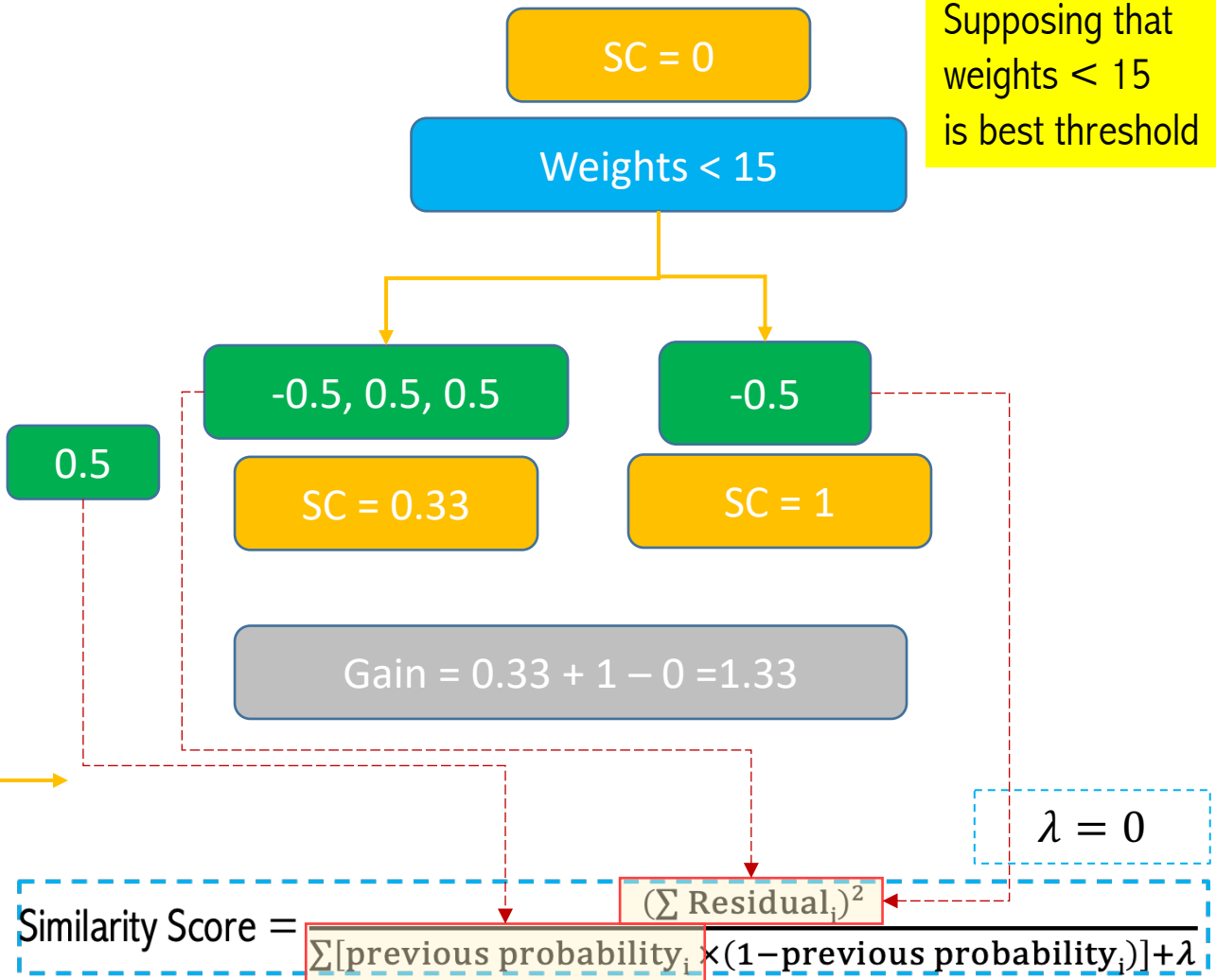


$$\lambda = 0$$

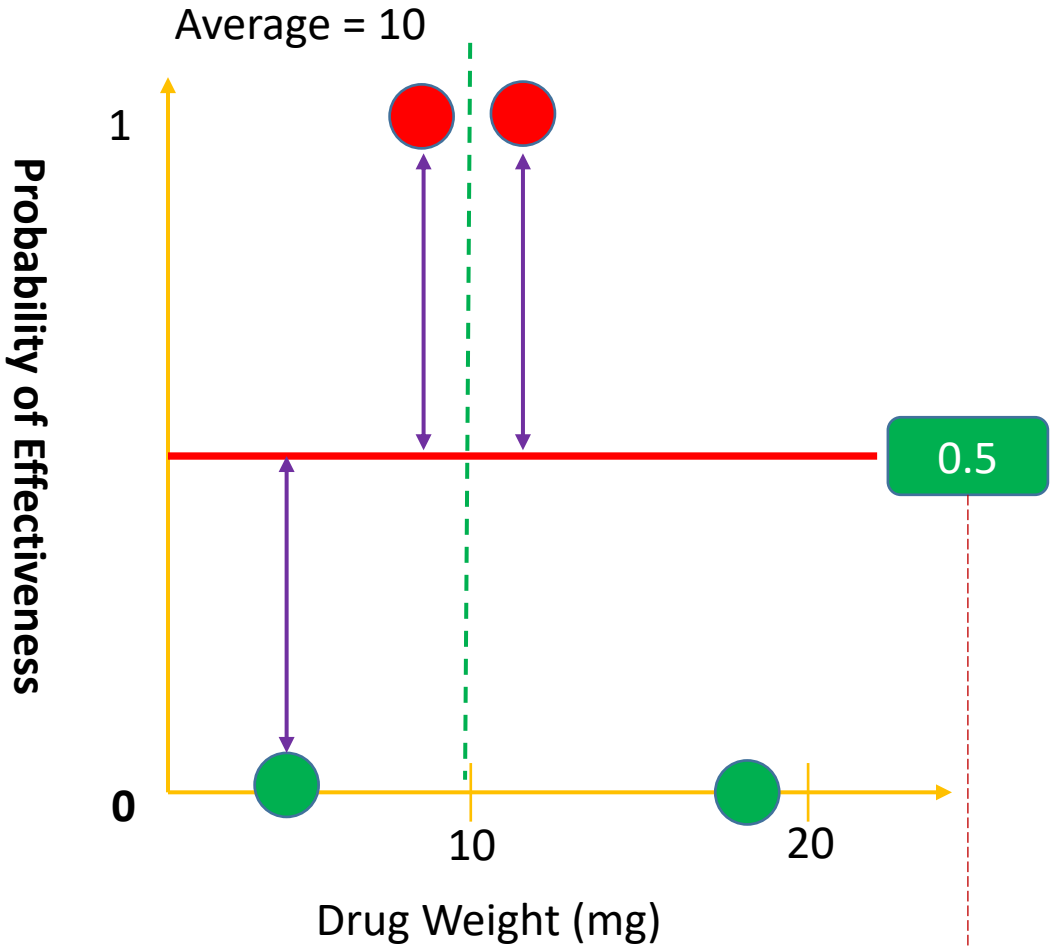
$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum[\text{previous probability}_i \times (1 - \text{previous probability}_i)] + \lambda}$$

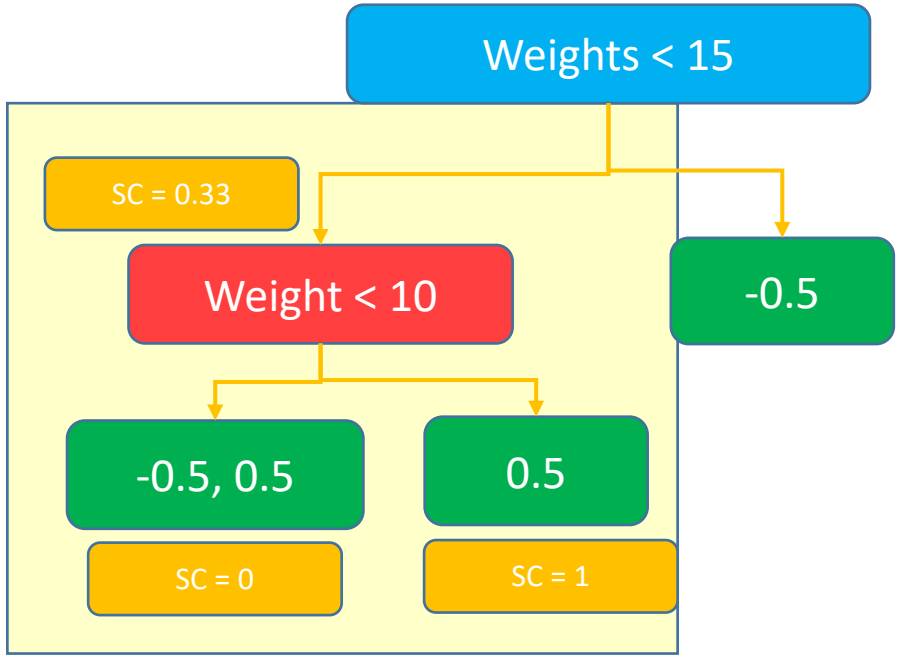
XGBoost For Classification





XGBoost For Classification



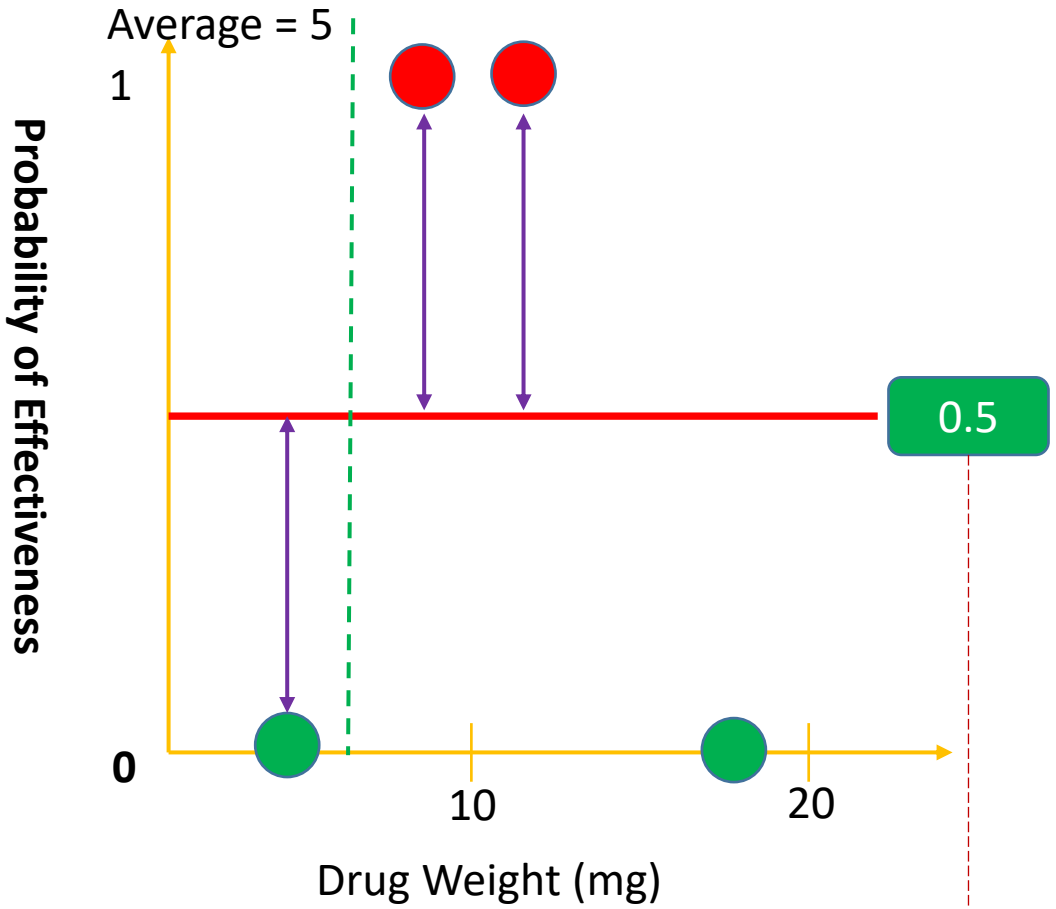


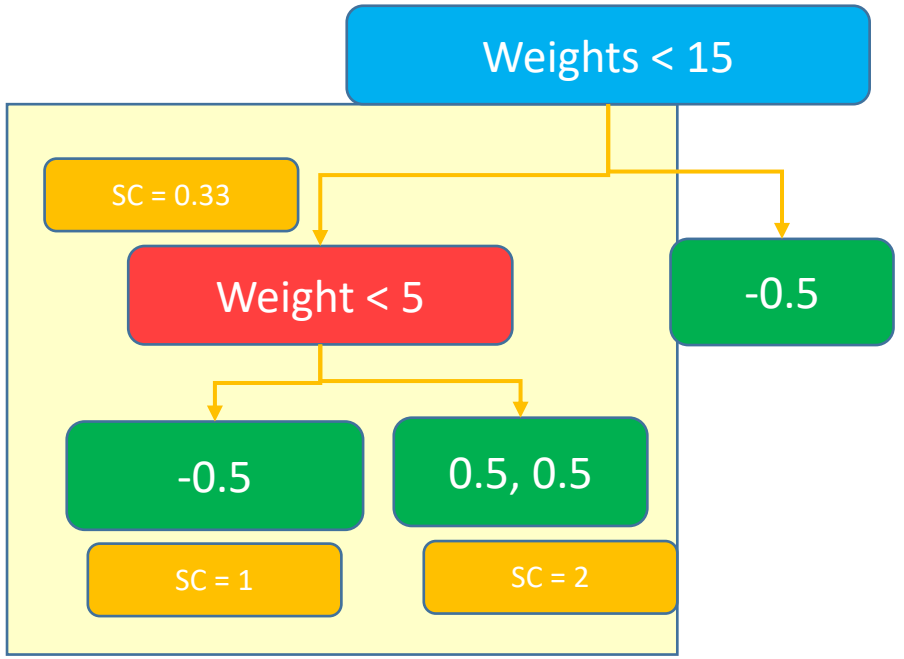
Gain = 0 + 1 – 0.33 = 0.66

$\lambda = 0$

$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum[\text{previous probability}_i \times (1 - \text{previous probability}_i)] + \lambda}$$

XGBoost for Classification

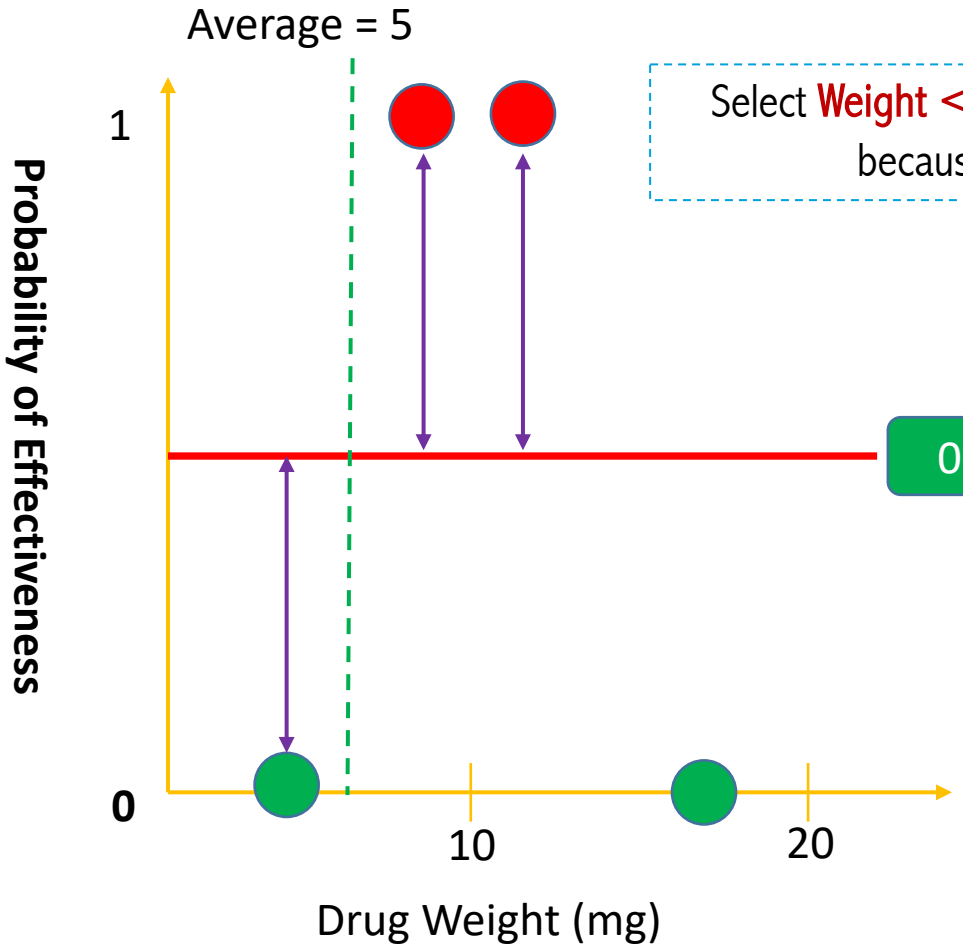




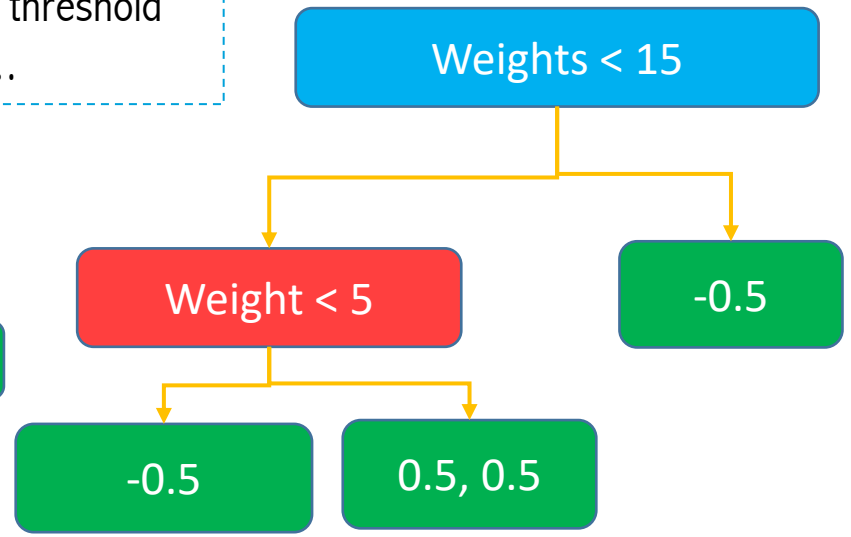
Gain = 1 + 2 – 0.33 = 2.66

$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum[\text{previous probability}_i \times (1 - \text{previous probability}_i)] + \lambda}$$

XGBoost for Classification



Select **Weight < 5** is threshold because ....



Giả sử quy định depth level = 2, dừng xây dựng Tree

How to estimate the minimum number of Residuals in each leaf
=> **XGBoot Cover**

By default: Mininmum XGBoot Cover is set to 1

What is a Cover

Similarity Score for Classification:

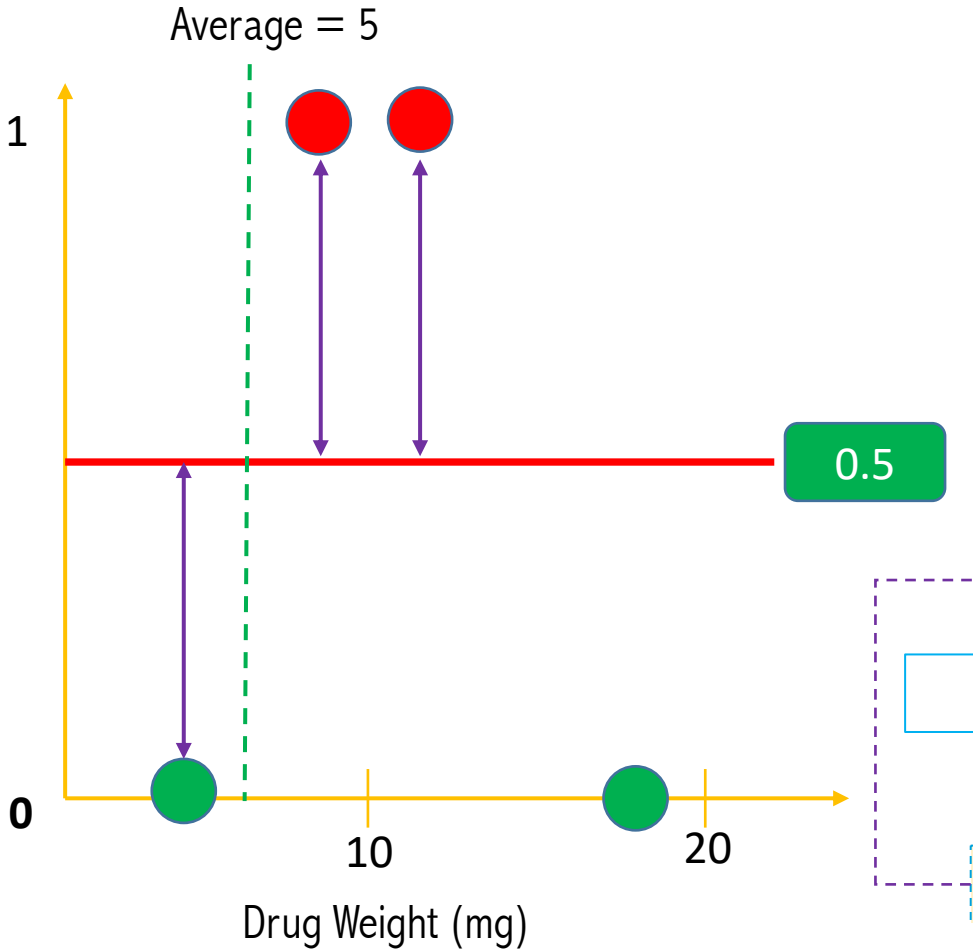
$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum[\text{previous probability}_i \times (1 - \text{previous probability}_i)] + \lambda}$$

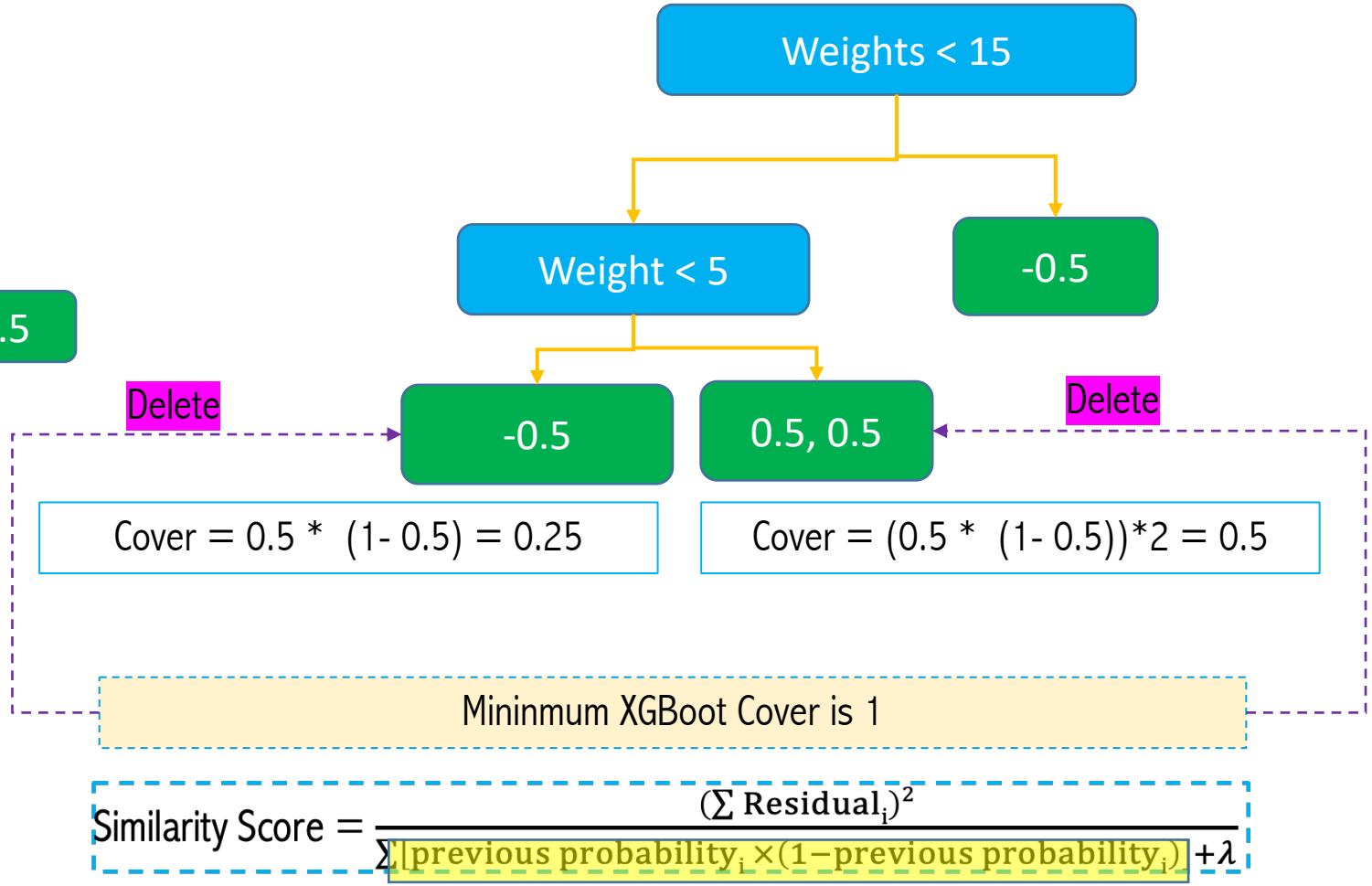
Cover

Similarity Score for Prediction:

$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\text{number of residual} + \lambda}$$

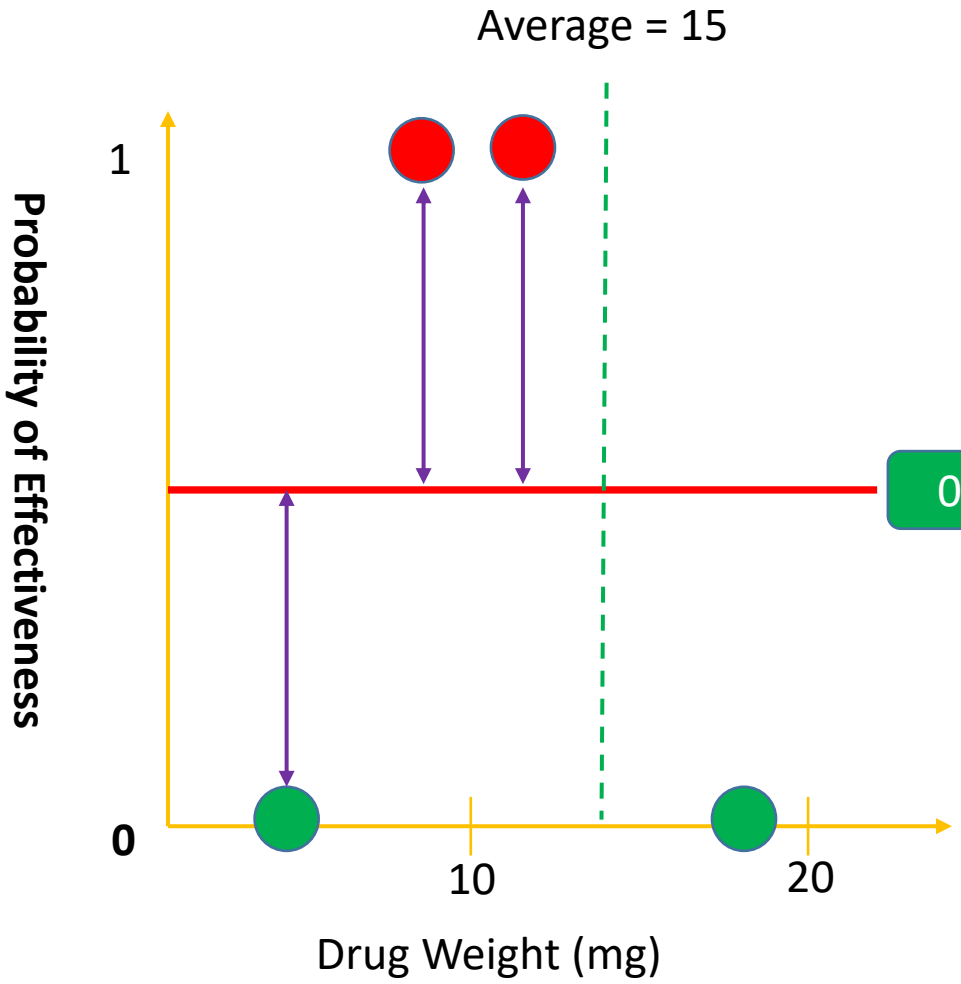
What is a Cover

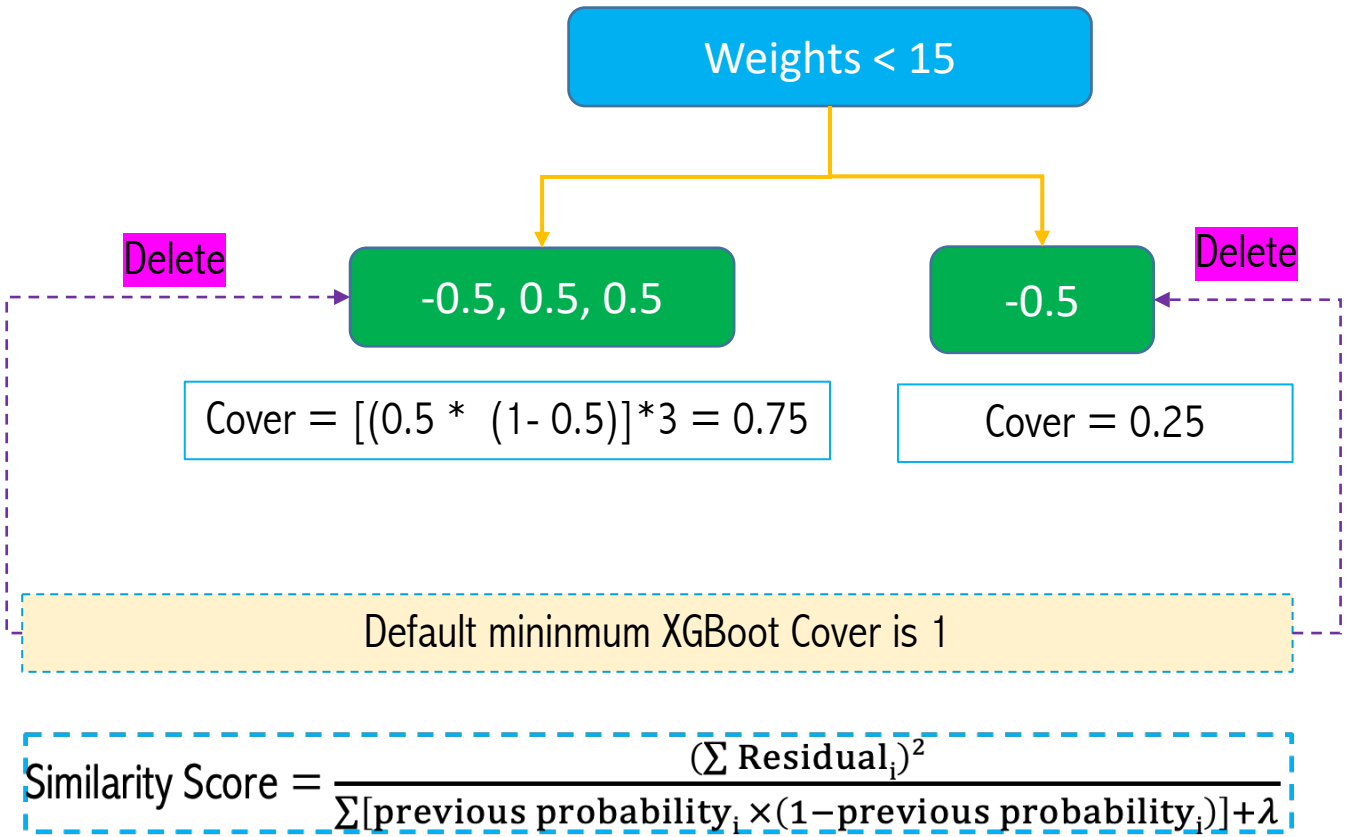




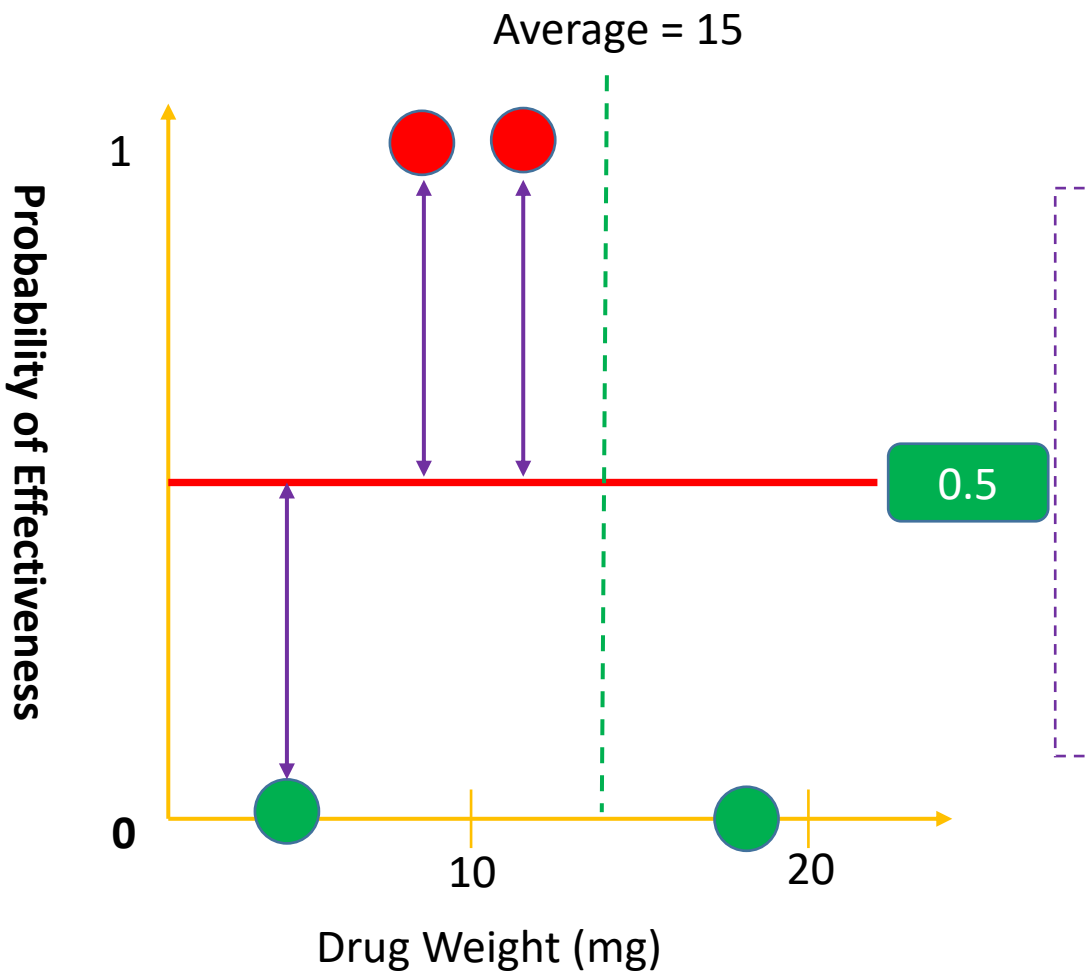
$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum_i [\text{previous probability}_i \times (1 - \text{previous probability}_i)] + \lambda}$$

Xgboost for Classification





Xgboost for Classification



Keep this node

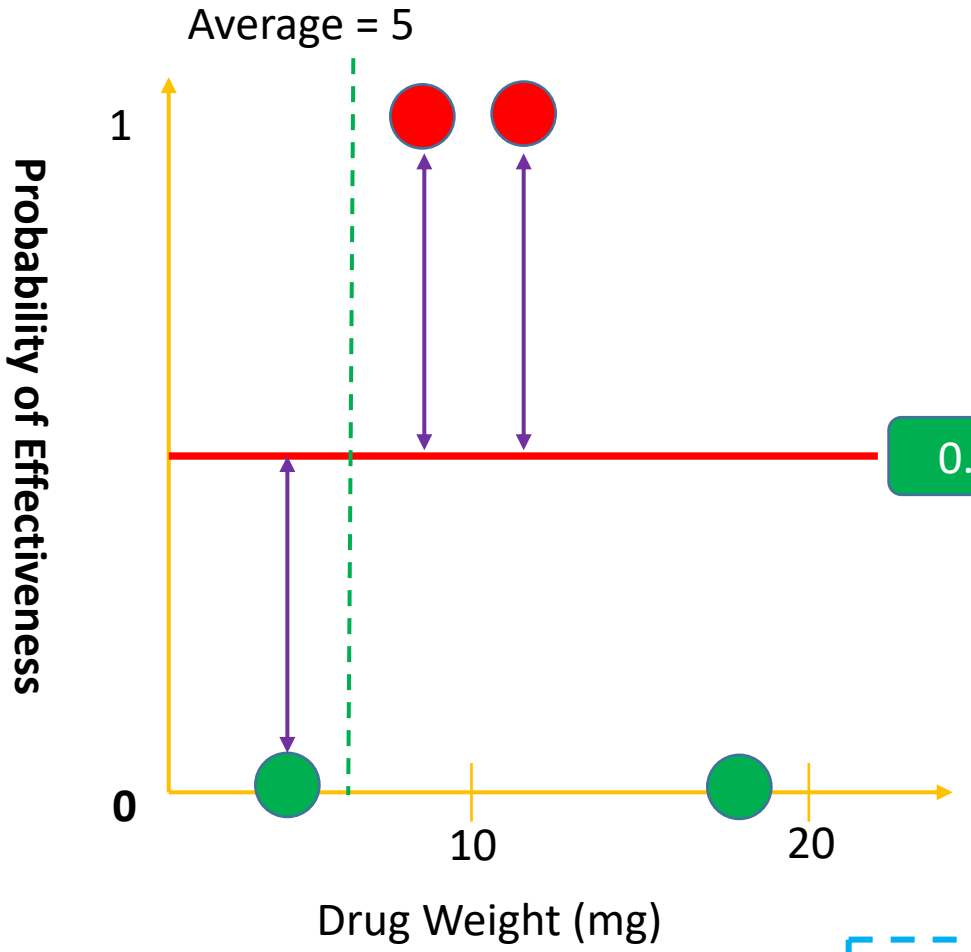
-0.5, 0.5, 0.5, -0.5

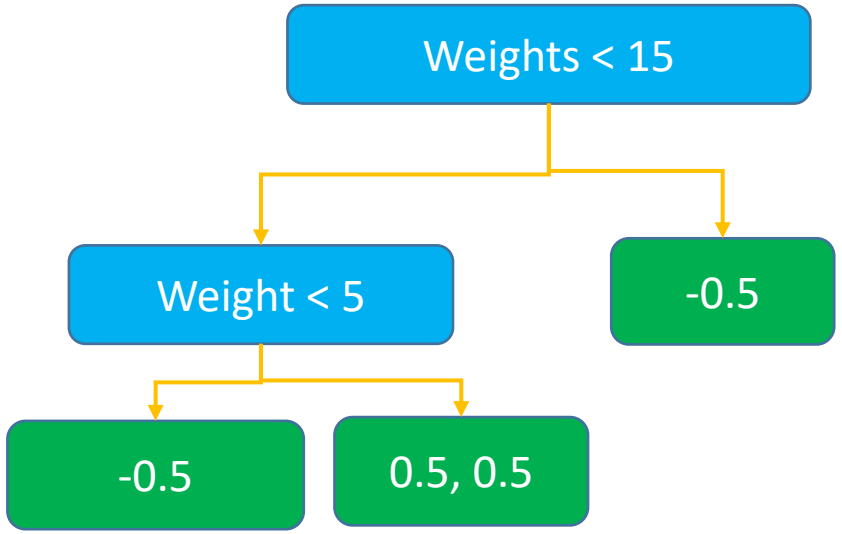
Cover = [(0.5 * (1- 0.5)]*4 = 1

Default mininmum XGBoot Cover is 1

$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum[\text{previous probability}_i \times (1-\text{previous probability}_i)]+\lambda}$$

How to predict the value





$$\text{Output Value} = \frac{(\sum \text{Residual}_i)}{\sum[\text{previous probability}_i \times (1 - \text{previous probability}_i)] + \lambda}$$

How to predict the value

Drug Weight	Drug Effectiveness
	No
	Yes
	Yes
	No

Initial prediction is that the probability of drug effective is 50%

2 Yes and 2 No => Probablity Yes = 2/4 = 1/2 = 0.5

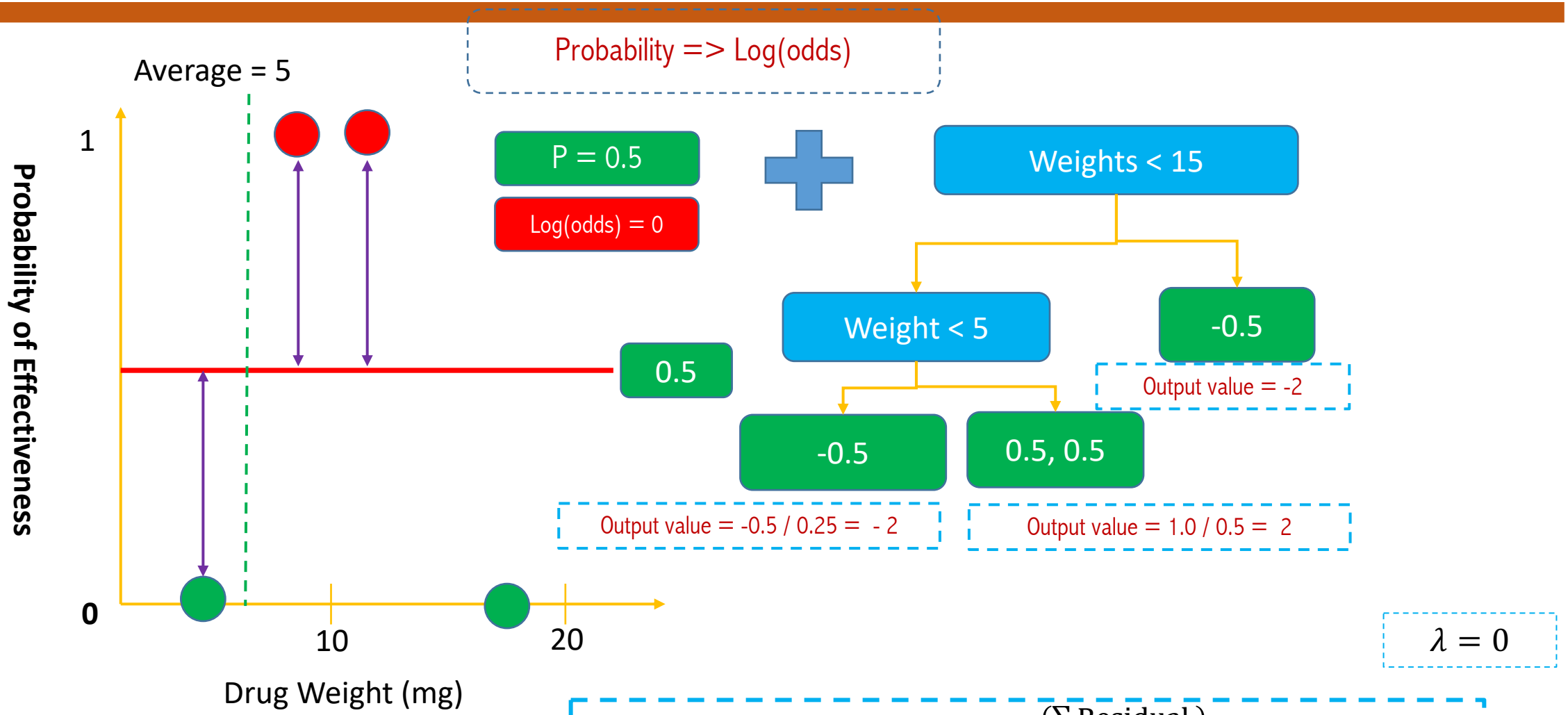
$$\text{Log(odds)} = \log\left(\frac{\text{Probablity Yes}}{\text{Probablity No}}\right) = 0$$

In XGBoost (or Gradient Boost), the initial prediction is that the log(odds)

$$\text{Probability of Drug Effectiveness} = \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}}$$

$$\text{Probability of Drug Effectiveness} = \frac{e^{0}}{1+e^{0}} = 0.5$$

How to predict the value



Tranformation formular for getting value at a leaf.

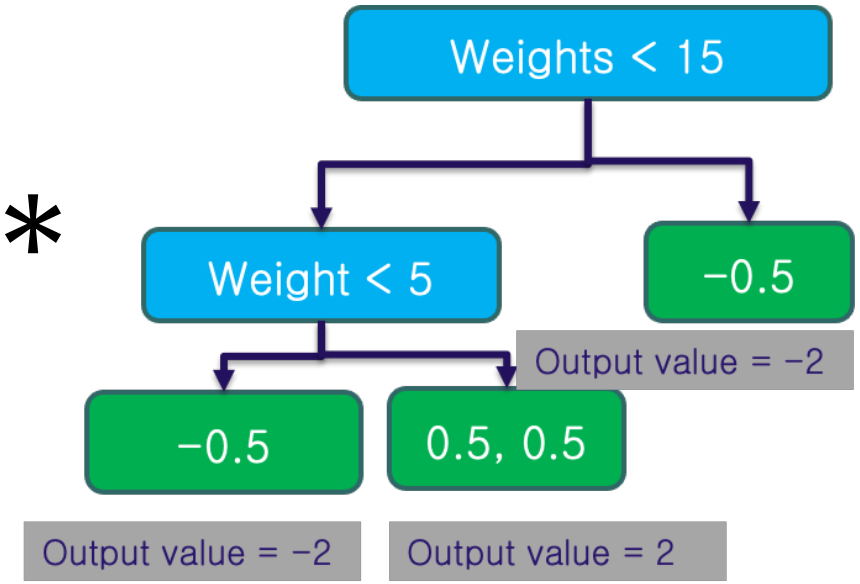
How to predict the value

Probability => Log(odds)

P = 0.5

Log(odds) = 0

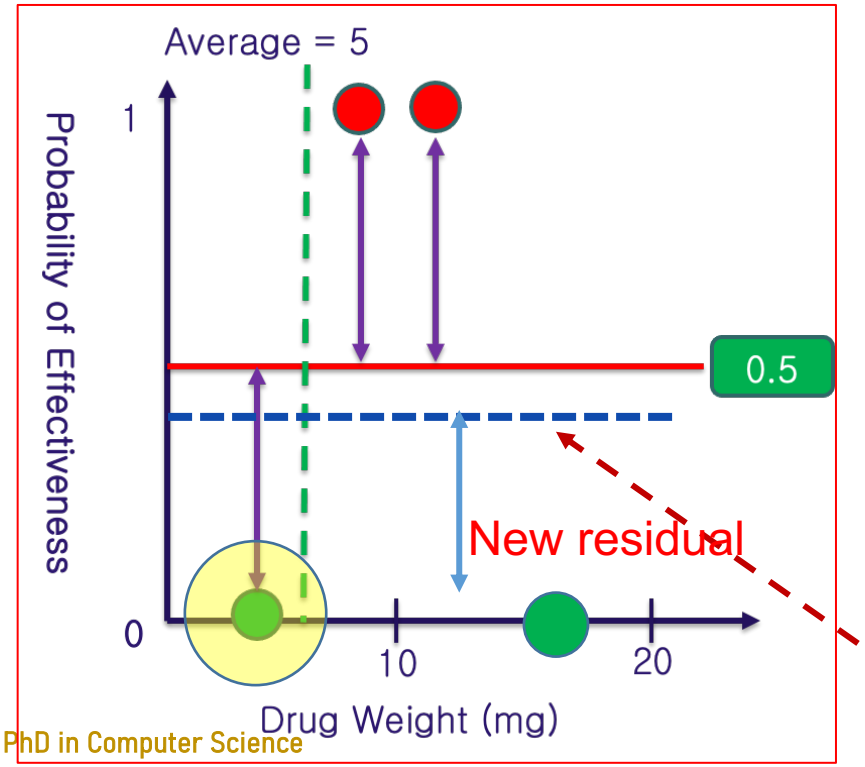
α *



$\frac{p}{1-p}$ = odds

$\text{Log}(\frac{p}{1-p})$ = log(odds)

α = 0.3



Prediction = 0 + 0.3 * (-2) = -0.6

$$\text{Probability} = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

$$\text{Probability} = \frac{e^{-0.6}}{1+e^{-0.6}} = 0.35$$

How to predict the value

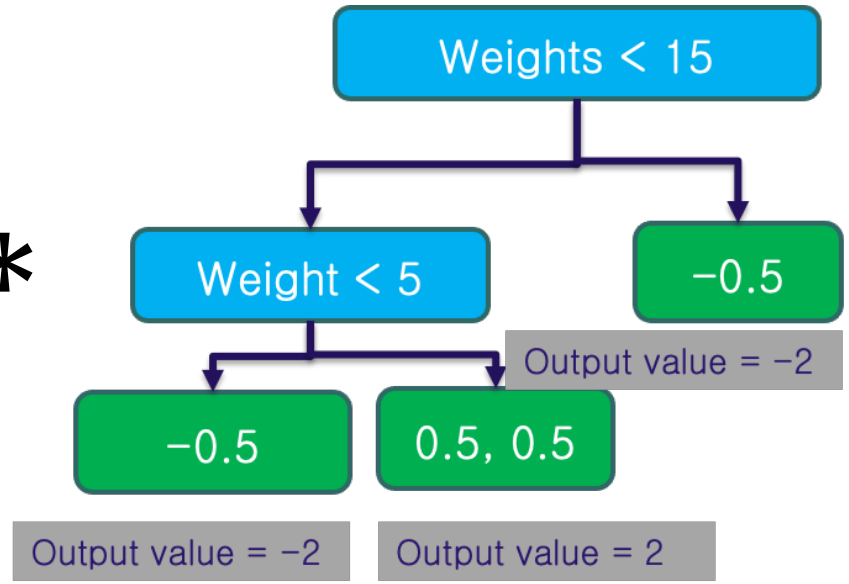
Probability => Log(odds)

Can we change P?

P = 0.5

Log(odds) = 0

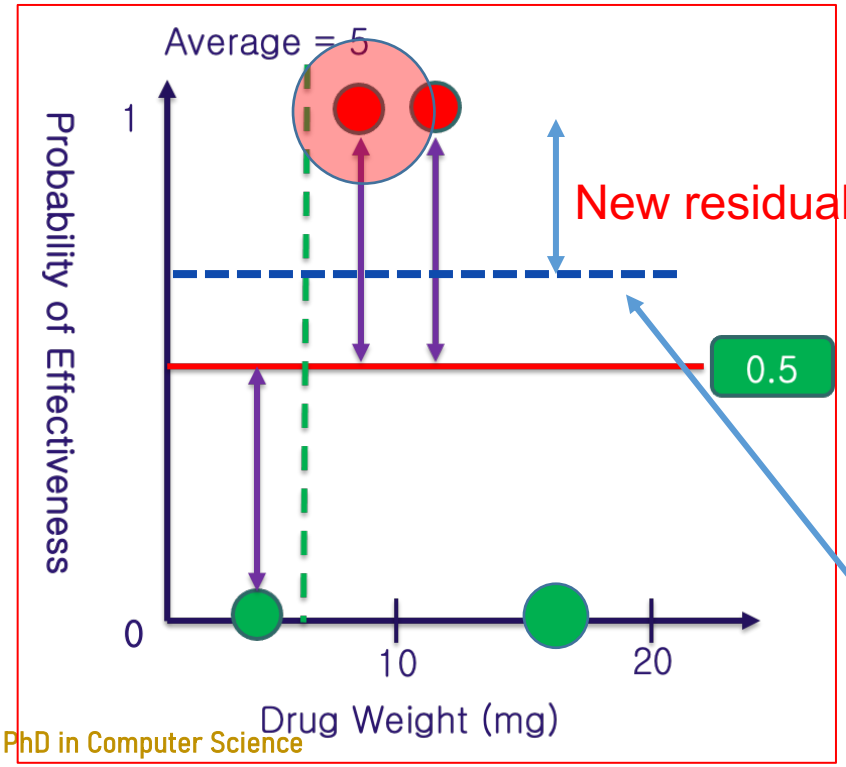
α *



$\frac{p}{1-p}$ = odds

$\text{Log}(\frac{p}{1-p})$ = log(odds)

α = 0.3



Log(odds) = Prediction = 0 + 0.3 * (2) = 0.6

$$\text{Probability} = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

$$\text{Probability} = \frac{e^{0.6}}{1+e^{0.6}} = 0.65$$

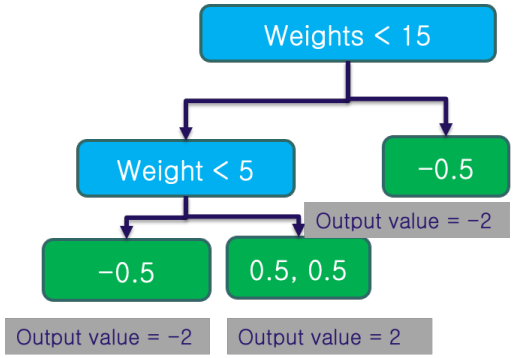
Build 2nd Tree

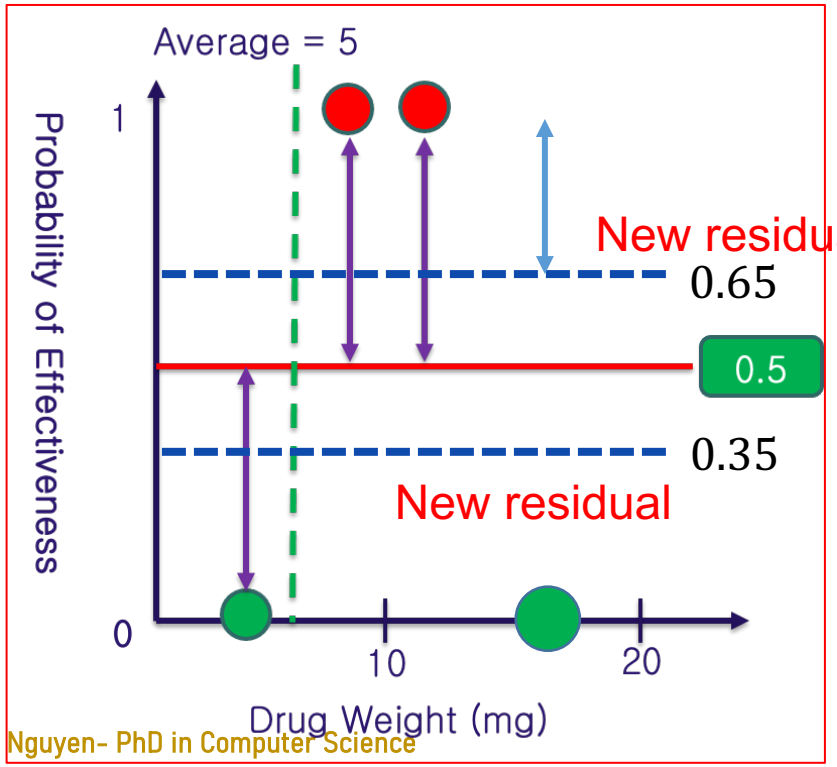
Probability => Log(odds)

P = 0.5

Log(odds) = 0

$\propto *$





$\propto *$

-0.35, 0.35, 0.35, -0.35

$$\text{Similarity Score} = \frac{(-0.35 + 0.35 + 0.35 - 0.35)^2}{0.35\times(1-0.35)+0.65\times(1-0.65)+0.65\times(1-0.65)+0.35\times(1-0.35)}$$

$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum[\text{previous probability}_i \times(1-\text{previous probability}_i)]+\lambda}$$

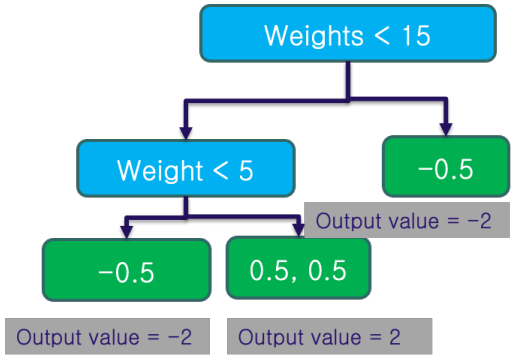
Build 2nd Tree

Probability => Log(odds)

P = 0.5

Log(odds) = 0

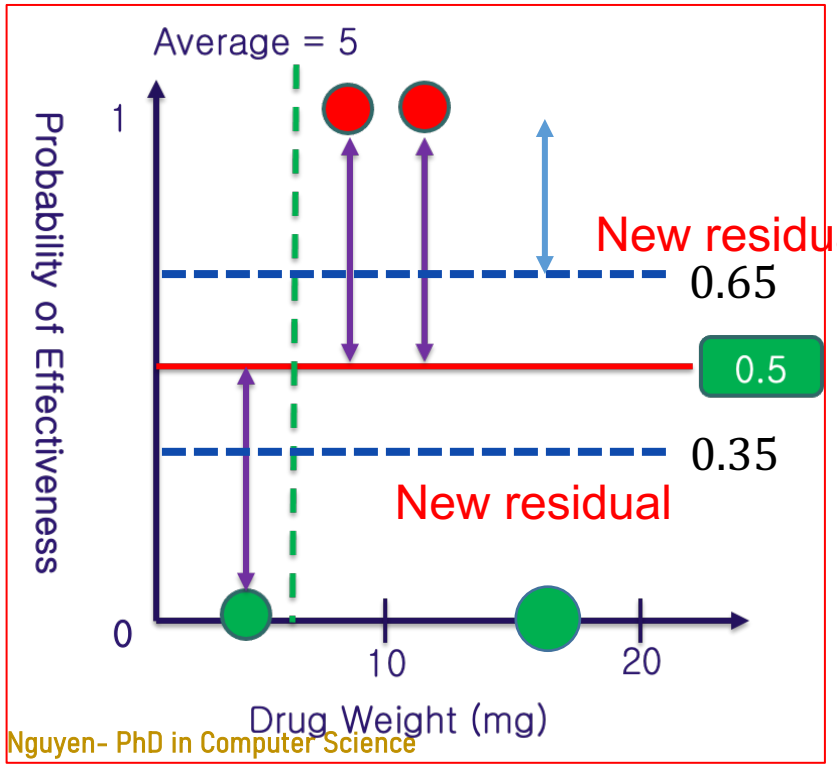
$\propto *$



$+$

$\propto *$

-0.35, 0.35, 0.35, -0.35



$$\text{Output Score} = \frac{(-0.35 + 0.35 + 0.35 - 0.35)}{0.35\times(1-0.35)+0.65\times(1-0.65)+0.65\times(1-0.65)+0.35\times(1-0.35)+\lambda}$$

$$\text{Output Score} = \frac{(\sum \text{Residual}_i)}{\sum[\text{previous probability}_i \times(1-\text{previous probability}_i)]+\lambda}$$

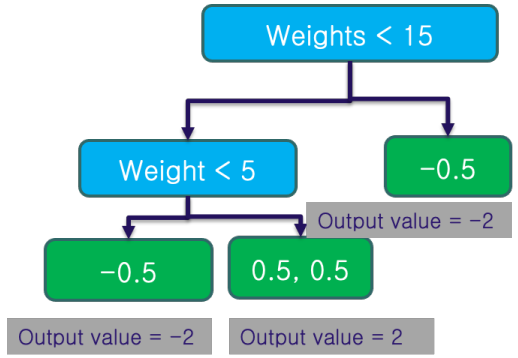
Build 2nd Tree

Probability => Log(odds)

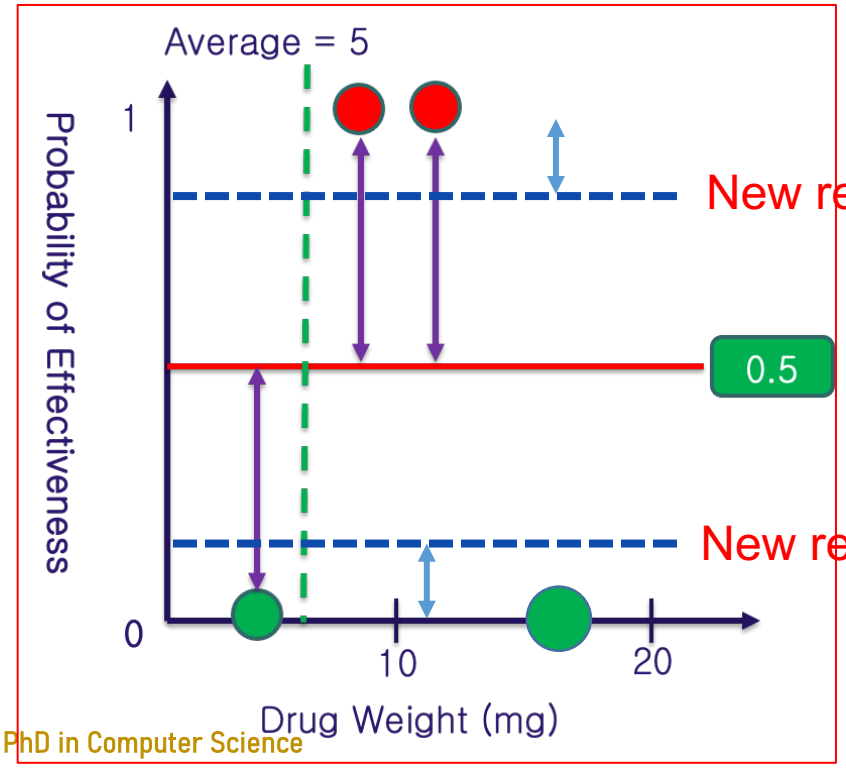
P = 0.5

Log(odds) = 0

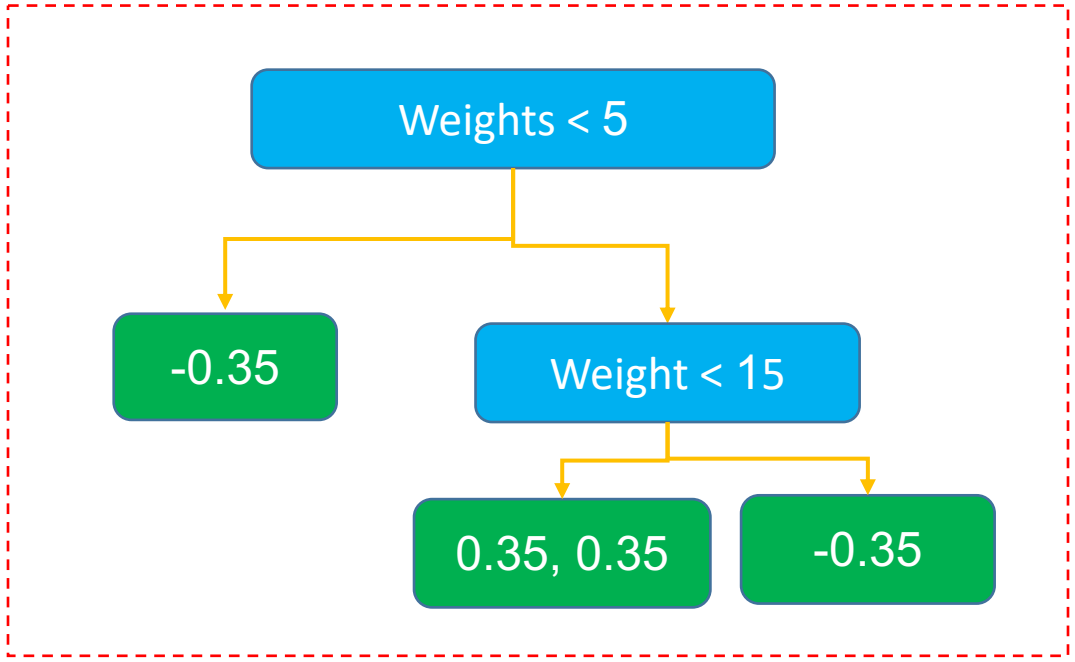
$+ \propto *$



$+$



$\propto *$



Review Questions

1. When do you stop to build the Tree

2. What's happen when $\lambda > 0$

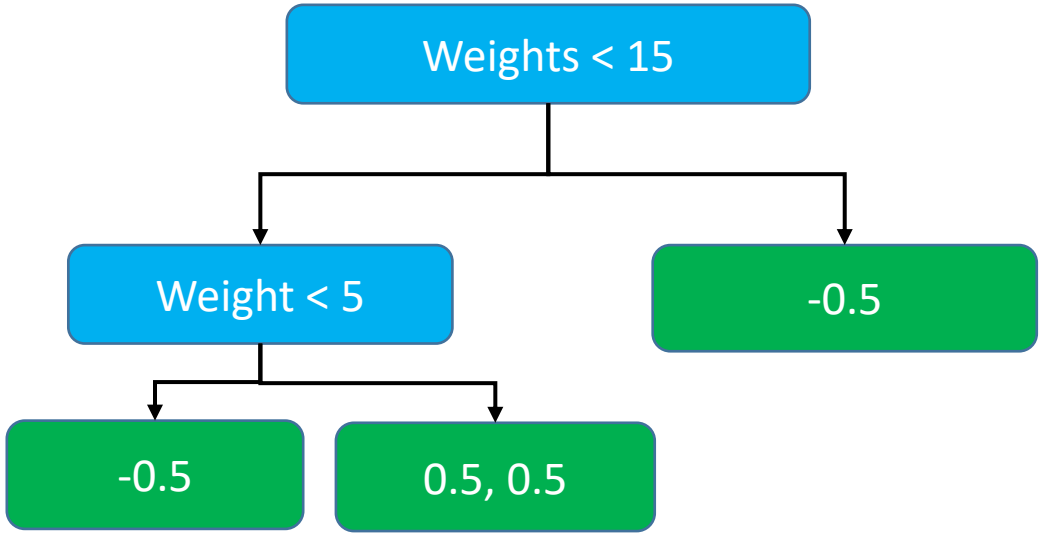
$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum[\text{previous probability}_i \times (1 - \text{previous probability}_i)] + \lambda}$$

Outline

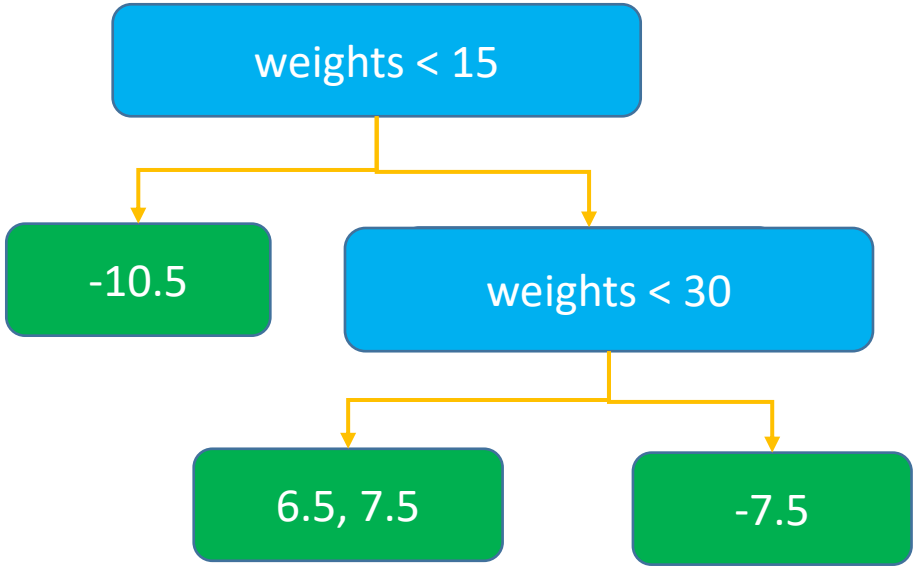
- Regularization
- XGBoost For Regression
- XGBoost For Classification
- XGBoost: Mathematical Explanation
- How To Fill Missing Values
- Example

XGBoost: Behind The Scenes

Classification



Regression



$$Similarity\ Score\ = \frac{(\sum \text{Residual})^2}{\sum \bar{y}_i \times (1-\bar{y}_i) + \lambda}$$

$$Ouput\ value\ = \frac{(\sum \text{Residual})}{\sum \bar{y}_i \times (1-\bar{y}_i) + \lambda}$$

$$Similarity\ Score\ = \frac{(\sum \text{Residual})^2}{\text{Number of Residual} + \lambda}$$

$$\text{Output Value} = \frac{(\sum \text{Residual})}{\text{Number of Residual} + \lambda}$$

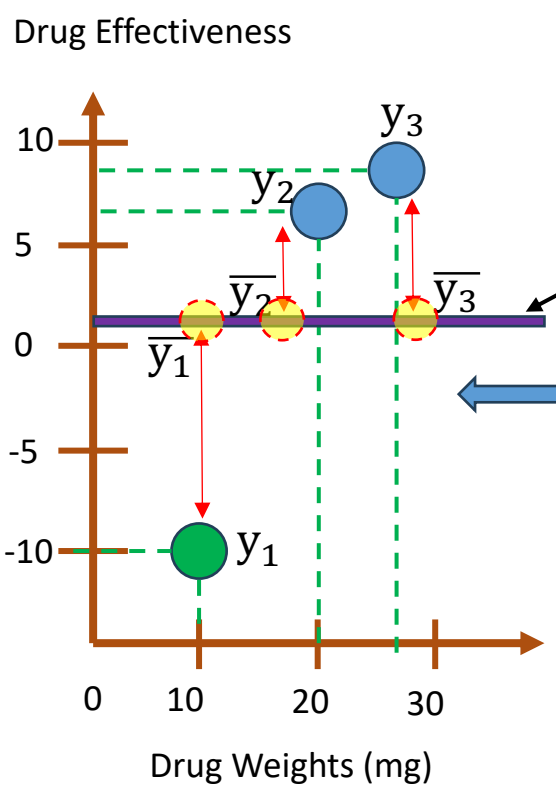
XGBoost: Behind The Scenes

Regression

Dự đoán ban đầu hiệu quả thuốc

0.5

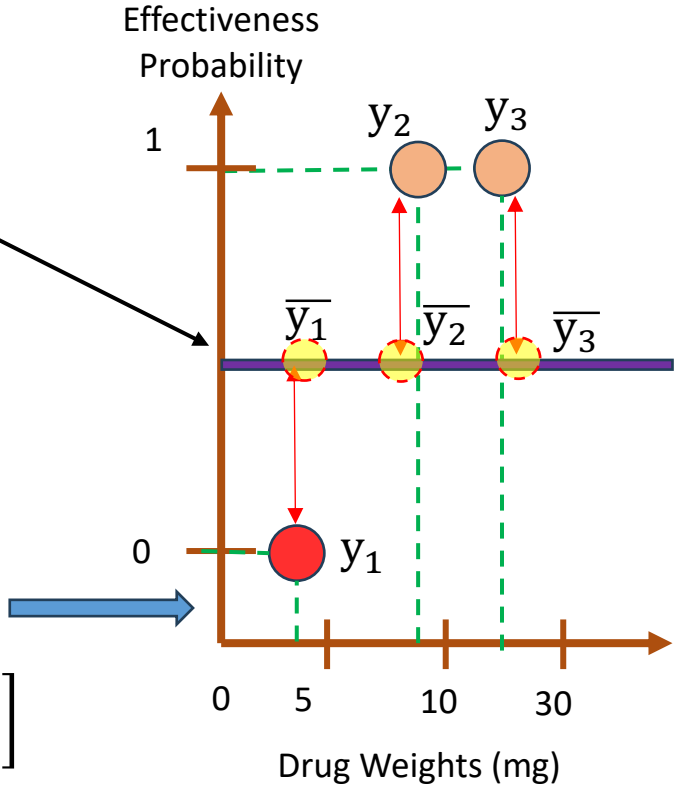
Classification



$$\sum_{i=1}^{3} \mathcal{L}(y_i, \overline{y_i}) \quad \text{Where} \quad \mathcal{L}(y_i, \overline{y_i}) = \frac{1}{2}(y_i - \overline{y_i})^2$$

$$\sum_{i=1}^{3} \mathcal{L}(y_i, \overline{y_i}) \quad \text{Where}$$

$$\mathcal{L}(y_i, \overline{y_i}) = -\left[\begin{matrix} y_i \log(\overline{y_i}) + \\ (1 - y_i)\log(1 - \overline{y_i}) \end{matrix}\right]$$



Sử dụng loss functions xây dựng cây

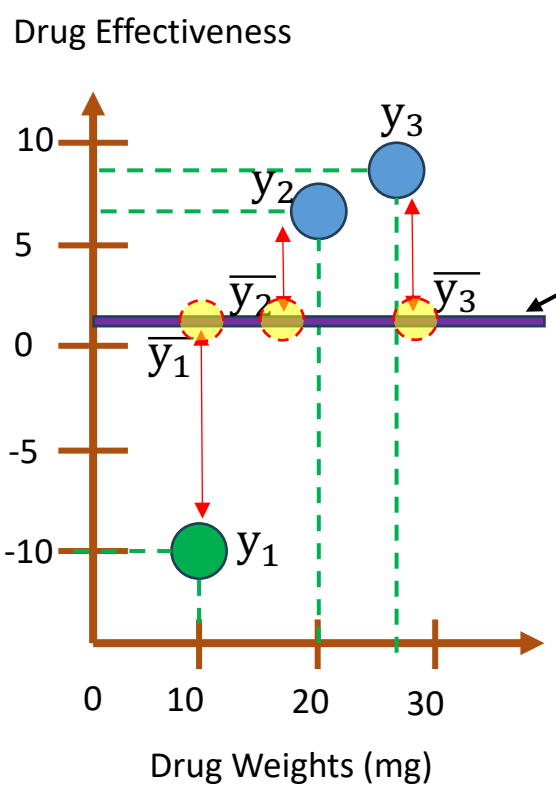
XGBoost: Behind The Scenes

Regression

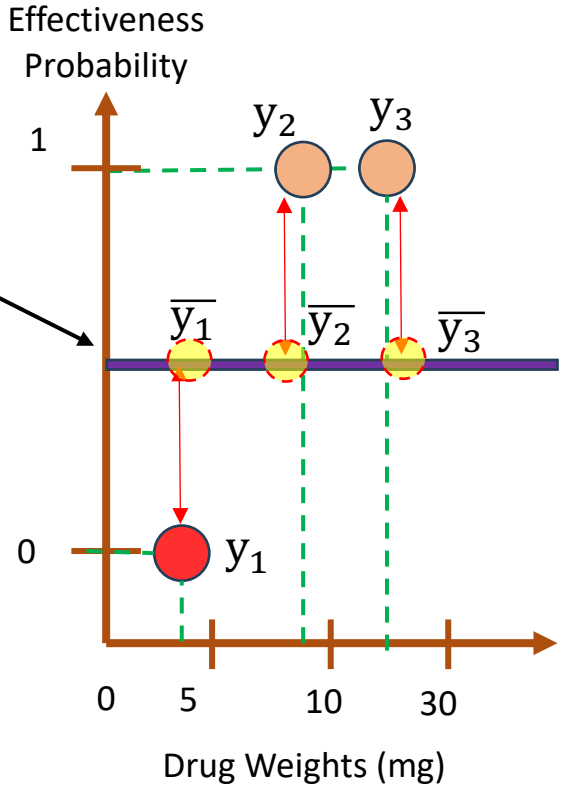
Dự đoán ban đầu hiệu quả thuốc

0.5

Classification



$$\sum_{i=1}^{n} \mathcal{L}(y_i, \overline{y_i}) + \gamma T + \lambda P^2$$



γ is a user definable penalty to encourse pruning

XGBoost can prune even when $\gamma = 0$

Pruning is excuted after the full tree built => It plays no role in deriving the Optimal Output Values

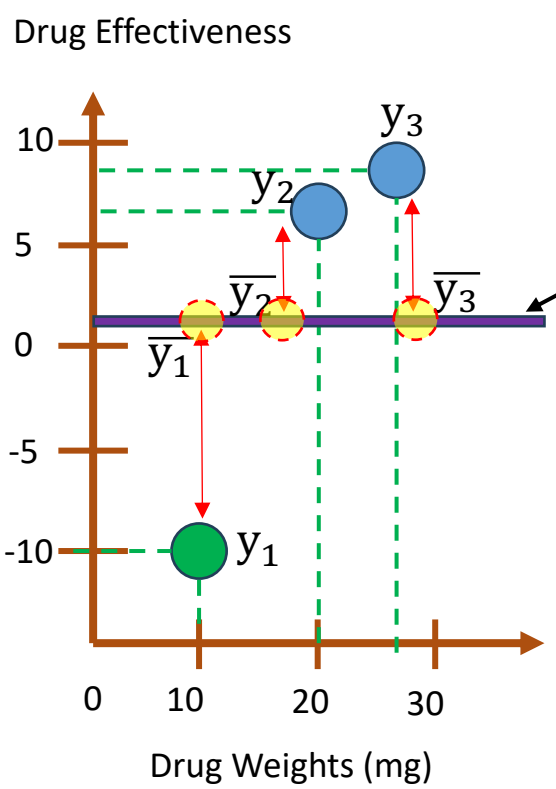
XGBoost: Behind The Scenes

Regression

Dự đoán ban đầu hiệu quả thuốc

0.5

Classification



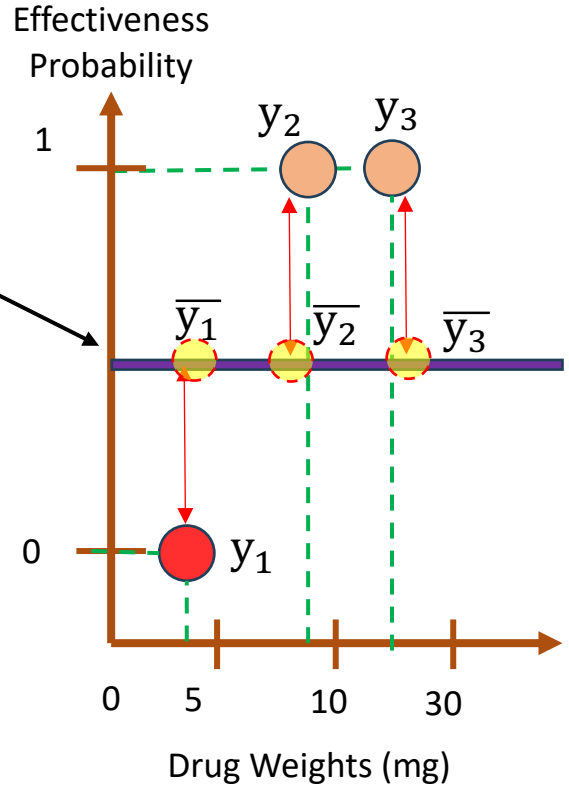
Bỏ qua γ

$$\sum_{i=1}^{n} \mathcal{L}(y_i, \overline{y_i}) + \gamma T + \lambda P^2$$

γ is a user definable penalty to encourse pruning

XGBoost can prune even when $\gamma = 0$

Pruning is excuted after the full tree built => It plays no role in deriving the Optimal Output Values



XGBoost: Behind The Scenes

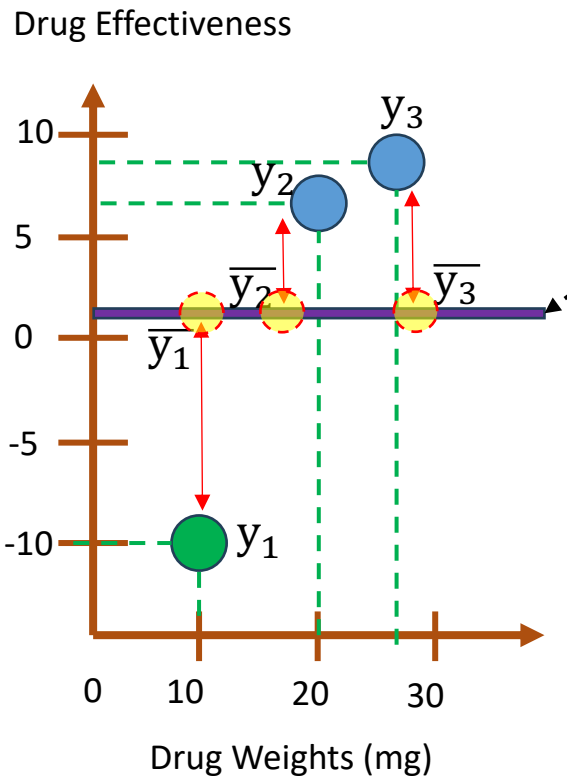
Regression

Dự đoán ban đầu hiệu quả thuốc

0.5

1st New Tree

Classification

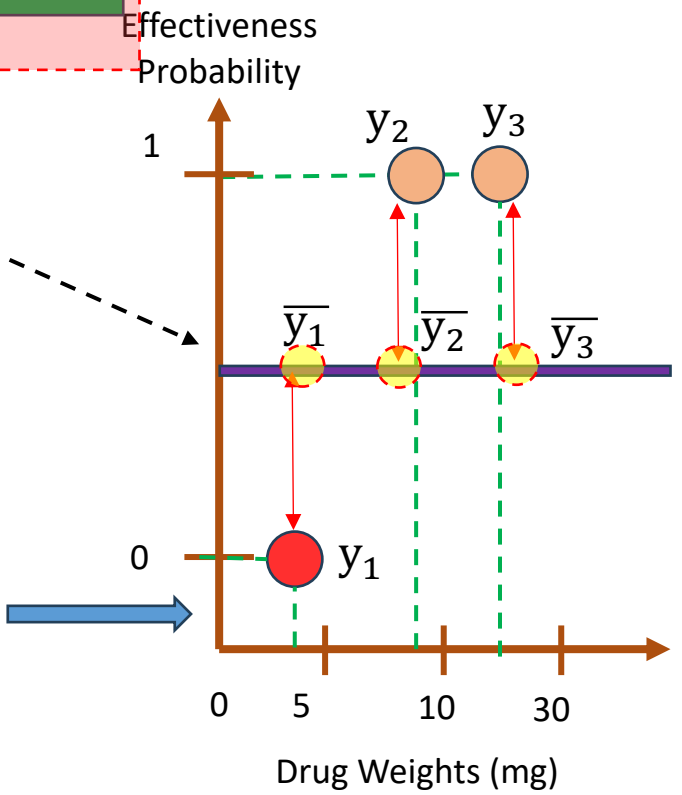


XGBoost builds the new tree based on the loss function:

$$\sum_{i=1}^{n} \mathcal{L}\left(y_i, \overline{y_i^0} + P\right) + \frac{1}{2}\lambda P^2$$

Rigde Regression
Regularization term

Mục tiêu: tìm giá trị dự đoán cho mỗi leaf (P) của cây mới nhằm minimize hàm loss.



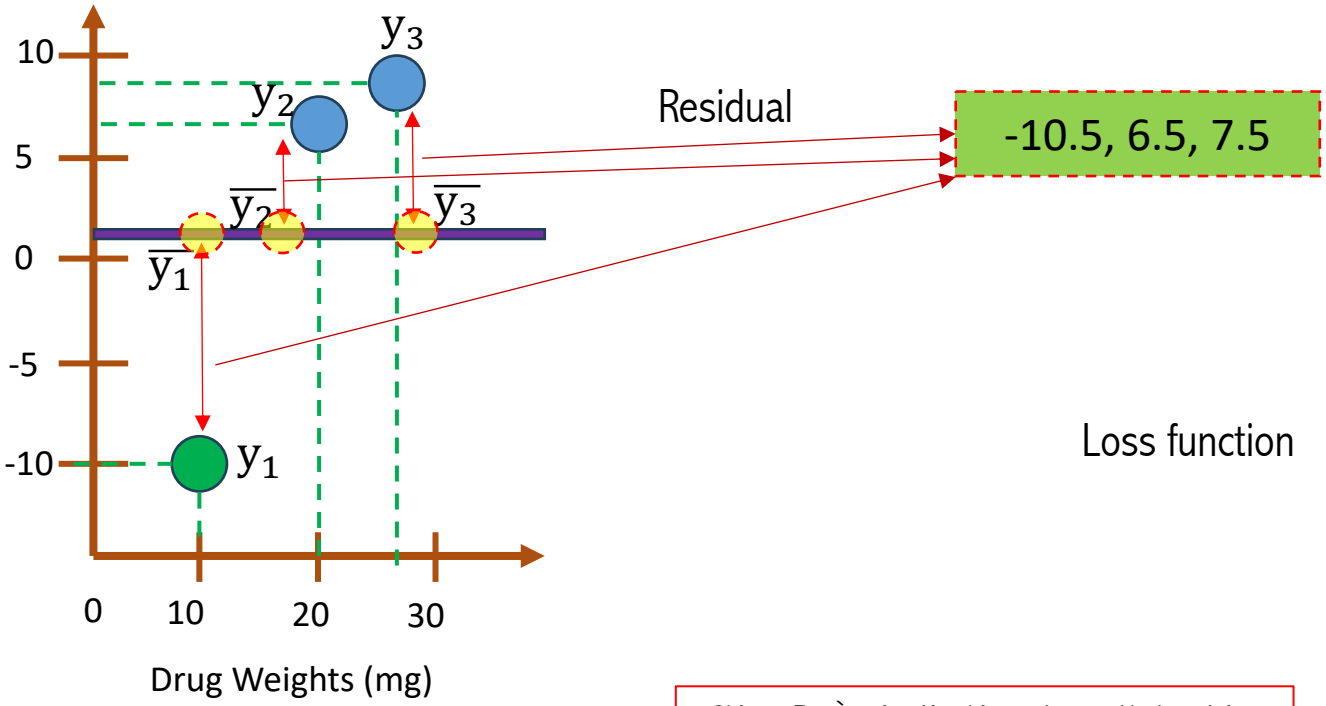
XGBoost Regression: Behind The Scenes

Regression

Dự đoán ban đầu hiệu quả thuốc

0.5

Drug Effectiveness



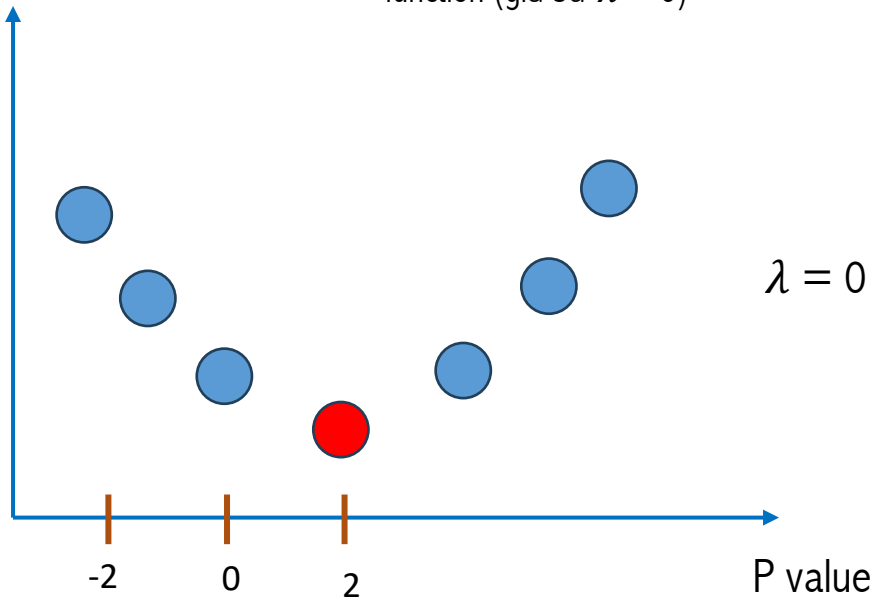
Residual

-10.5, 6.5, 7.5

Loss function

$$\sum_{i=1}^{n} \mathcal{L}\left(y_i, \overline{y_i^0} + P\right) + \frac{1}{2}\lambda P^2$$

Chúng cần tìm giá trị đầu ra của nút lá này (giá trị P) bằng cách mininize loss function (giả sử $\lambda = 0$)



Giá trị P cần tìm là giá trị ứng với đạo hàm của loss theo P bằng 0

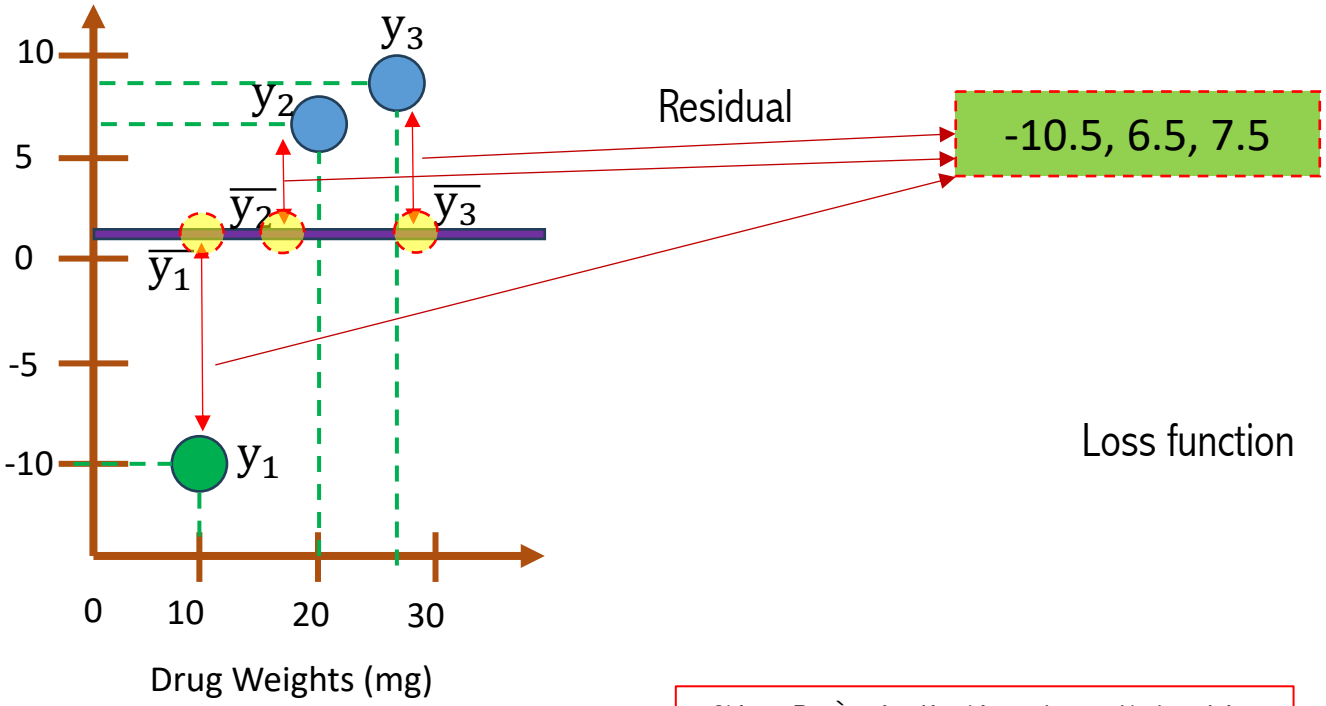
XGBoost Regression: Behind The Scenes

Regression

Dự đoán ban đầu hiệu quả thuốc

0.5

Drug Effectiveness



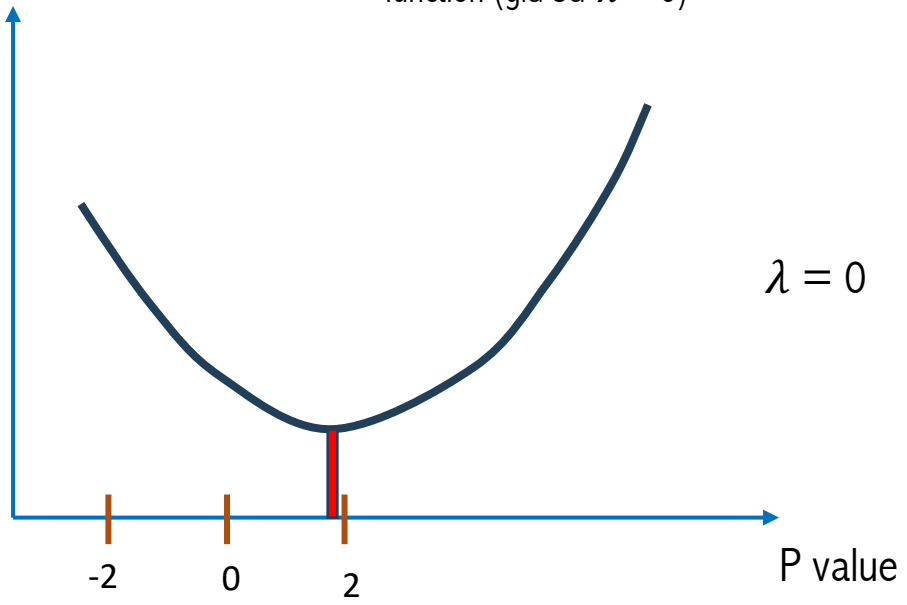
Residual

-10.5, 6.5, 7.5

$$\sum_{i=1}^{n} \mathcal{L}\left(y_i, \overline{y_i^0} + P\right) + \frac{1}{2}\lambda P^2$$

Chúng cần tìm giá trị đầu ra của nút lá này (giá trị P) bằng cách mininize loss function (giả sử $\lambda = 0$)

Loss function



Giá trị P cần tìm là giá trị ứng với đạo hàm của loss theo P bằng 0

XGBoost: Behind The Scenes

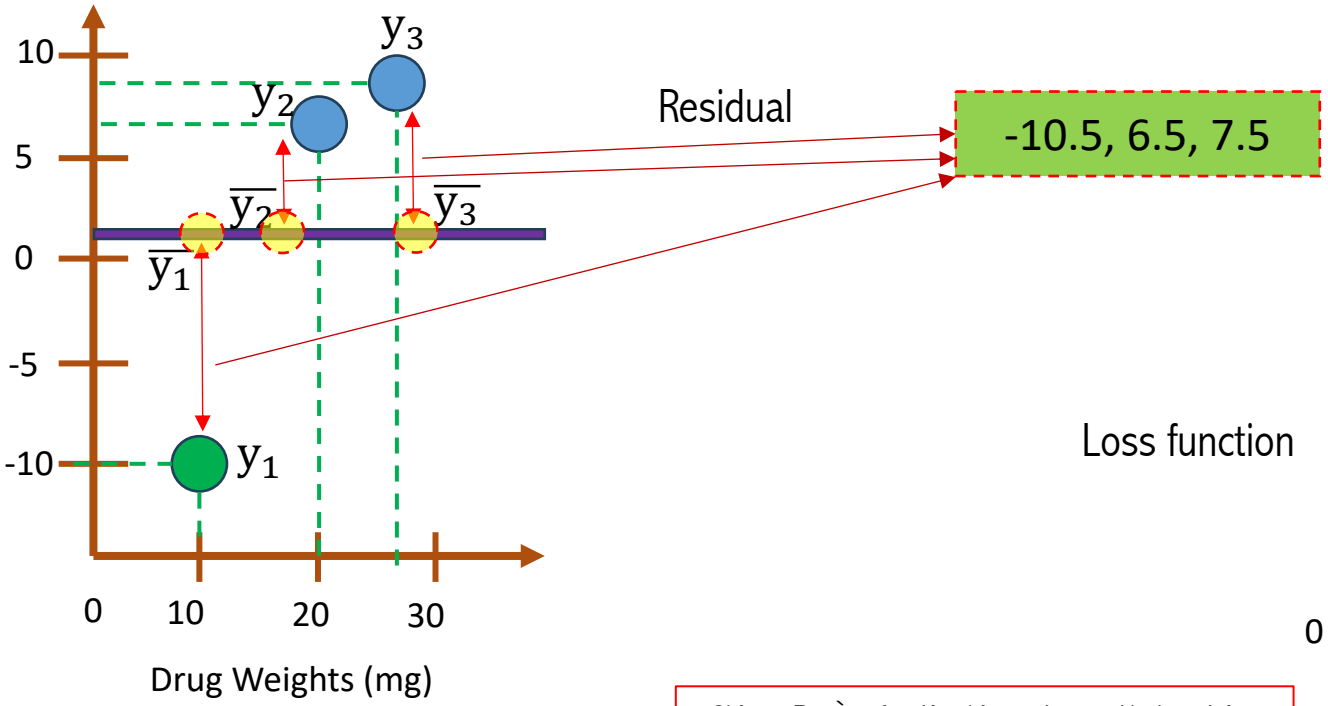
Regression

Dự đoán ban đầu hiệu quả thuốc

0.5

Whats happen if λ is very large?

Drug Effectiveness



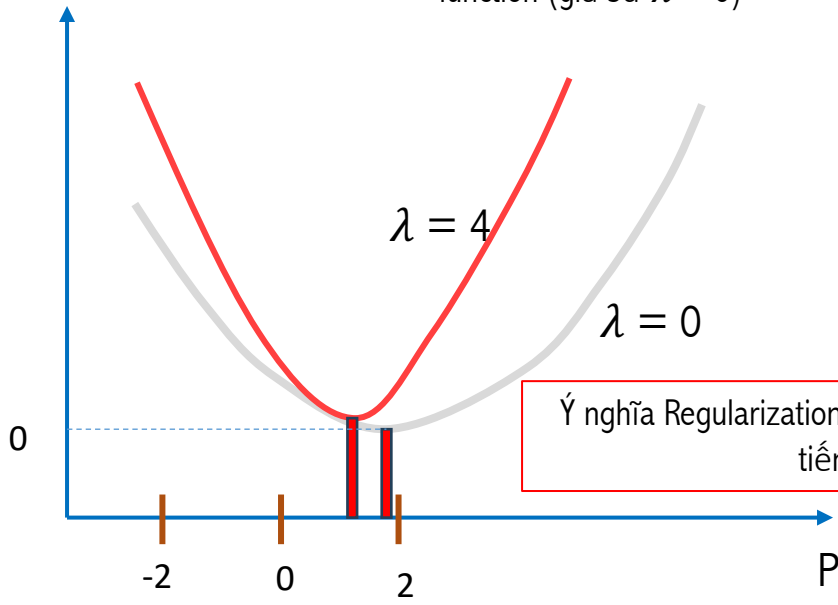
Residual

-10.5, 6.5, 7.5

$$\sum_{i=1}^{n} \mathcal{L}\left(y_i, \overline{y_i^0} + P\right) + \frac{1}{2}\lambda P^2$$

Chúng cần tìm giá trị đầu ra của nút lá này (giá trị P) bằng cách mininize loss function (giả sử $\lambda = 0$)

Loss function



Giá trị P cần tìm là giá trị ứng với đạo hàm của loss theo P bằng 0

Ý nghĩa Regularization: Tăng giá trị λ, giá trị P tiến về 0

XGBoost Regression: Behind The Scenes

$$\sum_{i=1}^{n} \mathcal{L}\left(y_i, \overline{y_i^0} + P\right) + \frac{1}{2}\lambda P^2$$

Rất khó để tìm optimalization, nên cũng ta sẽ sắp sỉ hàm loss bằng Second Order Tayler Appriximation

$$\mathcal{L}(y_i, \overline{y_i} + P) \approx \mathcal{L}(y_i, \overline{y_i}) + \left[\frac{d}{d\overline{y_i}}\mathcal{L}(y_i, \overline{y_i})\right] P + \frac{1}{2}\left[\frac{d}{d\overline{y_i^2}}\mathcal{L}(y_i, \overline{y_i})\right] P^2$$

$$\mathcal{L}(y_i, \overline{y_i} + P) \approx \mathcal{L}(y_i, \overline{y_i}) + gP + \frac{1}{2}hP^2$$

g (gradient) presents the first derivative of the loss function
h (hessian) presents the second derivative of the loss function

$$\mathcal{L}\left(y_1, \overline{y_1^0}\right) + g_1 P + \frac{1}{2}h_1 P^2 + \mathcal{L}\left(y_2, \overline{y_2^0}\right) + g_2 P + \frac{1}{2}h_2 P^2 + \ldots + \mathcal{L}\left(y_n, \overline{y_n^0}\right) + g_n P + \frac{1}{2}h_n P^2 + \frac{1}{2}\lambda P^2$$

Tìm giá trị P cần tìm sao cho đạo hàm của loss function theo P bằng 0

$$\frac{d}{dP}\left[(g_1 + g_2 + \cdots + g_n)P + \frac{1}{2}(h_1 + h + \cdots + h_n + \lambda)P^2\right] = 0$$

XGBoost Regression: Output Value

$$\frac{d}{dP}\left[(g_1 + g_2 + \cdots + g_n)P + \frac{1}{2}(h_1 + h + \cdots + h_n + \lambda)P^2\right] = 0$$

$$(g_1 + g_2 + \cdots + g_n) + (h_1 + h + \cdots + h_n\lambda)P \quad = 0$$

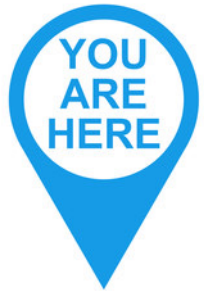
1

$$g_i = \frac{d}{d\bar{y}_i}\frac{1}{2}(y_i - \bar{y}_i)^2 = -(y_i - \bar{y}_i)$$

2

$$h_i = \frac{d^2}{d\bar{y}_i^2}\frac{1}{2}(y_i - \bar{y}_i)^2 = 1$$

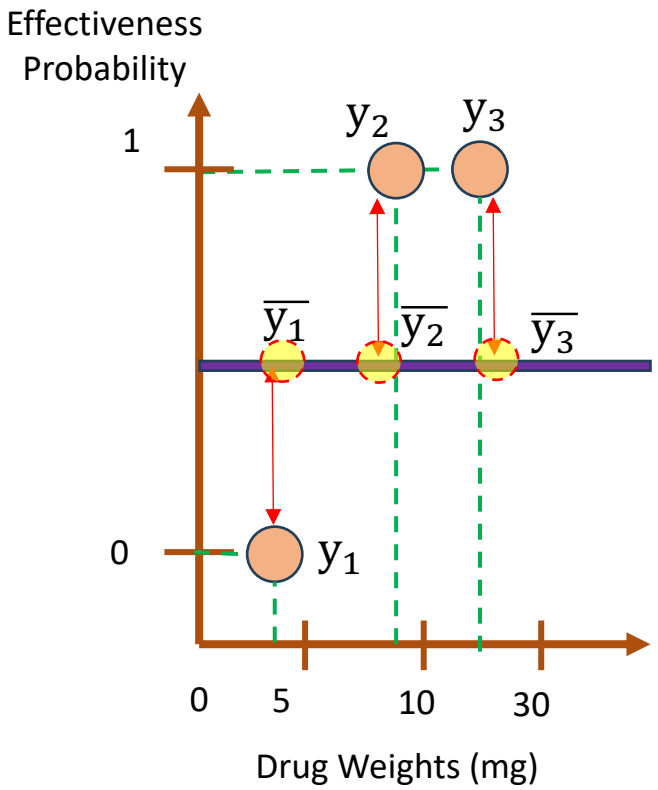
$$P = \frac{-(g_1+g_2+\cdots+g_n)}{h_1+h_2+\cdots+h_n+\lambda} = \frac{-(-(y_1-\bar{y_1})+-(y_2-\bar{y_2})+\cdots+-(y_n-\bar{y_n}))}{1+1+\cdots+1+\lambda} = \frac{\text{sum of residual}}{\text{numer of sum residual}+\lambda}$$



Output value af the leaf (or terminal node)

XGBoost For Classification: Output Value

Classification



$$\mathcal{L}(y_i, \overline{y_i}) = -[y_i \log(\overline{y_i}) + (1 - y_i)\log(1 - \overline{y_i})]$$

Convert probability to Log(odds)

$$\mathcal{L}(y_i, \log(\text{odds})) = -y_i \log(odds) + log(1 + e^{\log(odds)})$$

$$g_i = \frac{d}{dlog(odds)} \mathcal{L}(y_i, \log(\text{odds})) = -y_i + \frac{e^{\log(odds)})}{1+e^{\log(odds)}} = -(y - \overline{y_i})$$

$$h_i = \frac{d^2}{dlog(odds)^2} \mathcal{L}(y_i, \log(\text{odds})) = \frac{e^{\log(odds)})}{1 + e^{\log(odds)}} \times \frac{1}{1 + e^{\log(odds)}} = \overline{y_i} \times (1 - \overline{y_i})$$

$$P = \frac{-(g_1+g_2+\cdots+g_n)}{h_1+h_2+\cdots+h_n+\lambda} = \frac{\text{sum of residual}}{\overline{y_1}\times(1-\overline{y_1})+\overline{y_2}\times(1-\overline{y_2})+\cdots+\overline{y_n}\times(1-\overline{y_n})+\lambda} = \frac{(\sum \text{Residual})}{\sum \overline{y_i}\times(1-\overline{y_i})+\lambda}$$

YOU ARE HERE

XGBoost: Similarity Score

1

$$\sum_{i=1}^{n} \mathcal{L}(y_i, \overline{y_i^0} + P) + \frac{1}{2}\lambda P^2$$

Cả (1) và (2) đều có cùng optimization point P

Rất khó để tìm optimalization, nên cũng ta sẽ sắp sỉ hàm loss bằng Second Order Tayler Appriximation

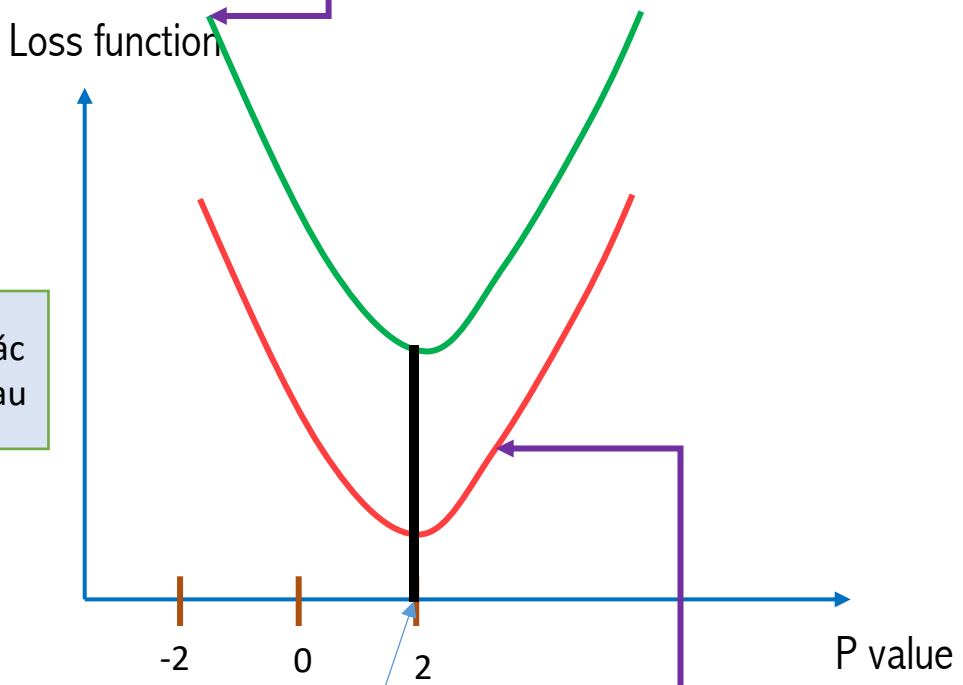
$$\mathcal{L}(y_i, \overline{y}_i + P) \approx \mathcal{L}(y_i, \overline{y}_i) + \left[\frac{d}{d\overline{y}_i}\mathcal{L}(y_i, \overline{y}_i)\right] P + \frac{1}{2}\left[\frac{d}{d\overline{y}_i^2}\mathcal{L}(y_i, \overline{y}_i)\right] P^2$$

$$\mathcal{L}(y_i, \overline{y}_i + P) \approx \mathcal{L}(y_i, \overline{y}_i) + gP + \frac{1}{2}hP^2$$

2

$$(g_1 + g_2 + \cdots + g_n)P + \frac{1}{2}(h_1 + h + \cdots + h_n + \lambda)P^2$$

Khác nhau



$$P = \frac{-(g_1 + g_2 + \cdots + g_n)}{h_1 + h_2 + \cdots + h_n + \lambda}$$

XGBoost: Similarity Score

1

$$\sum_{i=1}^{n} \mathcal{L}(y_i, \overline{y_i^0} + P) + \frac{1}{2}\lambda P^2$$

Cả (1) và (2) đều có cùng optimization point P

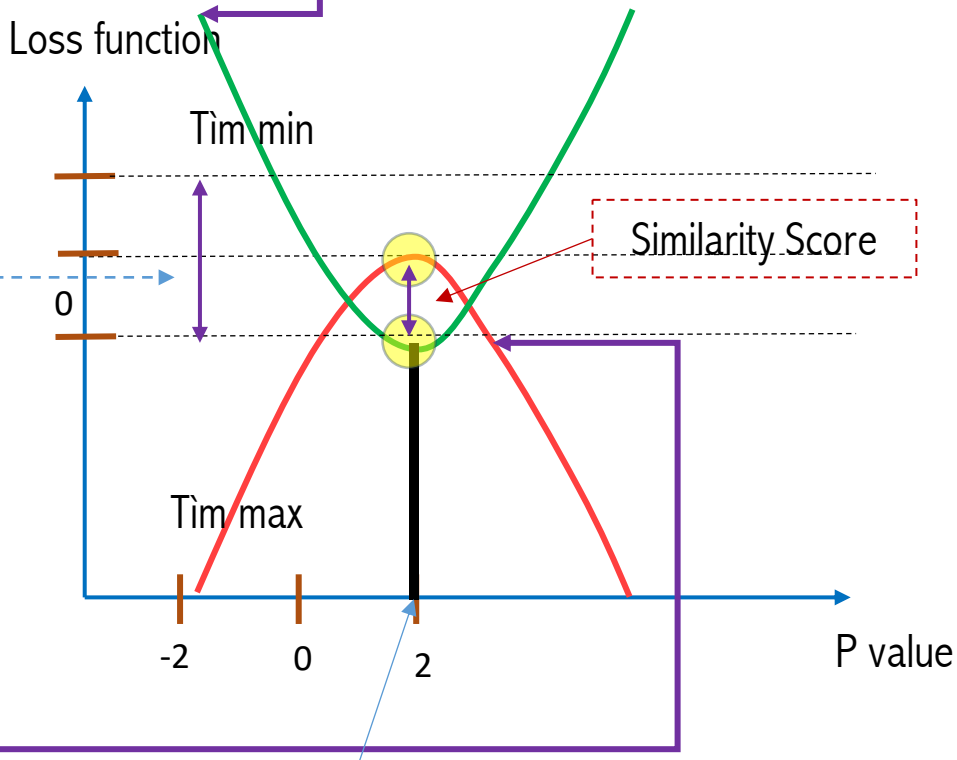
Rất khó để tìm optimalization, nên cũng ta sẽ sắp sỉ hàm loss bằng Second Order Tayler Appriximation

$$\mathcal{L}(y_i, \overline{y_i} + P) \approx \mathcal{L}(y_i, \overline{y_i}) + \left[\frac{d}{d\overline{y_i}}\mathcal{L}(y_i, \overline{y_i})\right] P + \frac{1}{2}\left[\frac{d}{d\overline{y_i^2}}\mathcal{L}(y_i, \overline{y_i})\right] P^2$$

$$\mathcal{L}(y_i, \overline{y_i} + P) \approx \mathcal{L}(y_i, \overline{y_i}) + gP + \frac{1}{2}hP^2$$

2

$$\text{-1 X } (g_1 + g_2 + \cdots + g_n)P + \text{-1 X } \frac{1}{2}(h_1 + h + \cdots + h_n + \lambda)P^2$$



Implementation Similarity Score

$$P = \frac{-(g_1 + g_2 + \cdots + g_n)}{h_1 + h_2 + \cdots + h_n + \lambda}$$

XGBoost: Similarity Score

1

$$\sum_{i=1}^{n} \mathcal{L}(y_i, \overline{y_i^0} + P) + \frac{1}{2}\lambda P^2$$

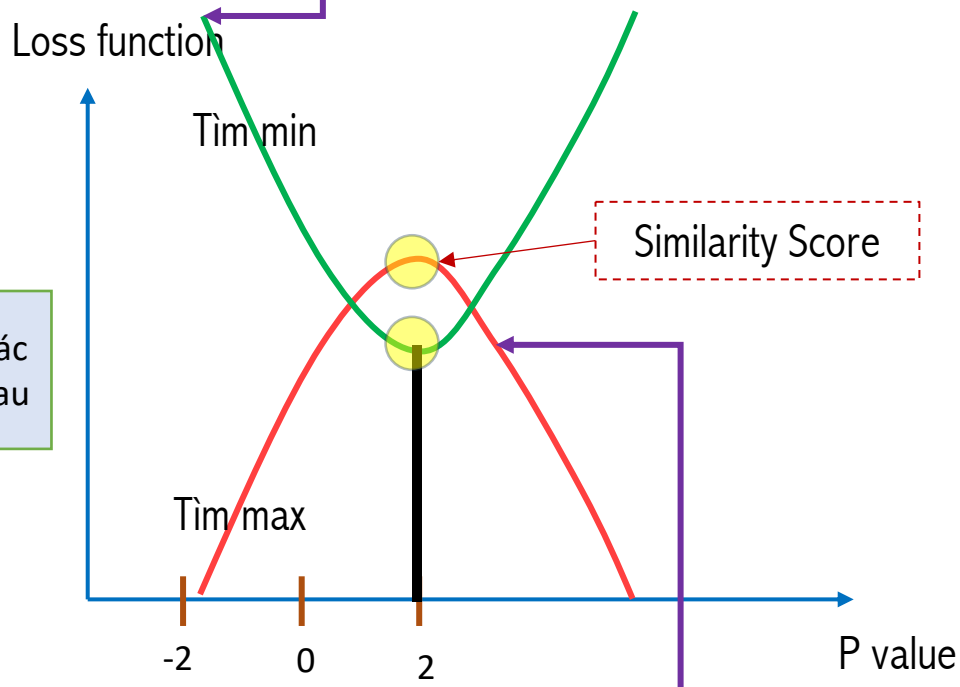
Cả (1) và (2) đều có cùng optimization point P

Rất khó để tìm optimalization, nên cũng ta sẽ sắp sỉ hàm loss bằng Second Order Tayler Appriximation

$$\mathcal{L}(y_i, \overline{y}_i + P) \approx \mathcal{L}(y_i, \overline{y}_i) + \left[\frac{d}{d\overline{y}_i}\mathcal{L}(y_i, \overline{y}_i)\right] P + \frac{1}{2}\left[\frac{d}{d\overline{y}_i^2}\mathcal{L}(y_i, \overline{y}_i)\right] P^2$$

$$\mathcal{L}(y_i, \overline{y}_i + P) \approx \mathcal{L}(y_i, \overline{y}_i) + gP + \frac{1}{2}hP^2$$

Khác nhau



2

$$\text{-1 X } (g_1 + g_2 + \cdots + g_n)P + \text{-1 X } \frac{1}{2}(h_1 + h + \cdots + h_n + \lambda)P^2$$

$$\text{Similarity Score} = \frac{1}{2}\frac{(g_1 + g_2 + \cdots + g_n)^2}{(h_1 + h_2 + \cdots + h_n + \lambda)}$$

$$P = \frac{-(g_1 + g_2 + \cdots + g_n)}{h_1 + h_2 + \cdots + h_n + \lambda}$$

XGBoost Regression: Similarity Score

1

$$\sum_{i=1}^{n} \mathcal{L}(y_i, \overline{y_i^0} + P) + \frac{1}{2}\lambda P^2$$

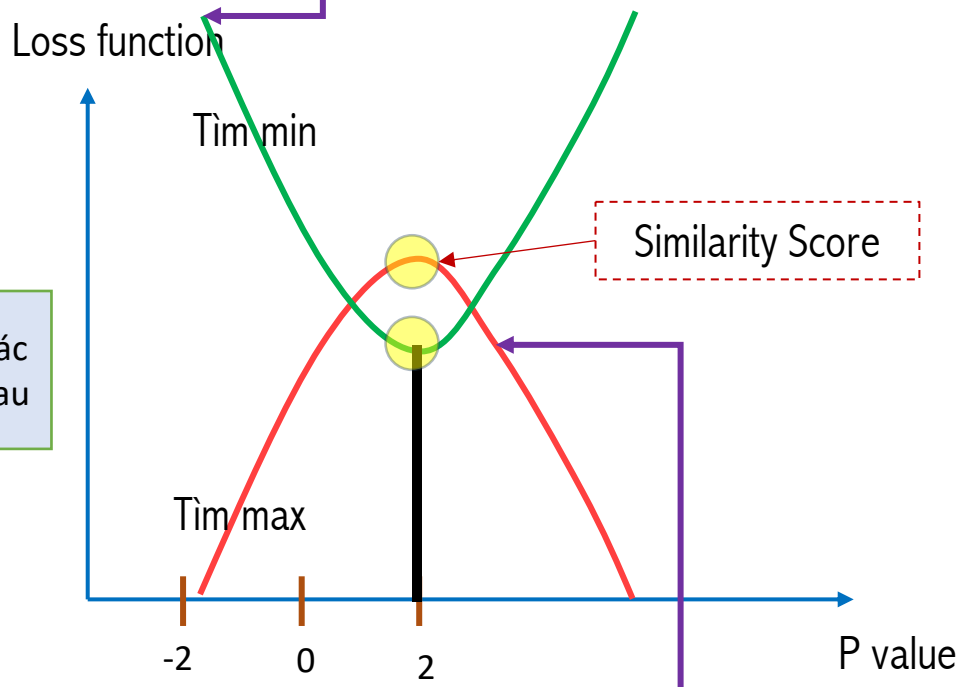
Cả (1) và (2) đều có cùng optimization point P

Rất khó để tìm optimalization, nên cũng ta sẽ sắp sỉ hàm loss bằng Second Order Tayler Appriximation

$$\mathcal{L}(y_i, \overline{y_i} + P) \approx \mathcal{L}(y_i, \overline{y_i}) + \left[\frac{d}{d\overline{y_i}}\mathcal{L}(y_i, \overline{y_i})\right] P + \frac{1}{2}\left[\frac{d}{d\overline{y_i^2}}\mathcal{L}(y_i, \overline{y_i})\right] P^2$$

$$\mathcal{L}(y_i, \overline{y_i} + P) \approx \mathcal{L}(y_i, \overline{y_i}) + gP + \frac{1}{2}hP^2$$

Khác nhau



2

$$\text{-1 X } (g_1 + g_2 + \cdots + g_n)P + \text{-1 X } \frac{1}{2}(h_1 + h + \cdots + h_n + \lambda)P^2$$

$h_i = \frac{d}{d\overline{y_i^2}}\frac{1}{2}(y_i - \overline{y_i})^2 = 1$

$g_i = \frac{d}{dy_i}\frac{1}{2}(y_i - \overline{y_i})^2 = -(y_i - \overline{y_i})$

$$\text{Similarity Score} = \frac{1}{2}\frac{(g_1 + g_2 + \cdots + g_n)^2}{(h_1 + h_2 + \cdots + h_n + \lambda)}$$

$$\text{Similarity Score} = \frac{(\sum \text{Residual})^2}{\text{Number of Residual} + \lambda}$$

AI VIETNAM
All-in-One Course

XGBoost Classification: Similarity Score

1

$$\sum_{i=1}^{n} \mathcal{L}\left(y_i, \overline{y_i^0} + P\right) + \frac{1}{2}\lambda P^2$$

Rất khó để tìm optimalization, nên cũng ta sẽ sắp sỉ hàm loss bằng Second Order Tayler Appriximation

$$\mathcal{L}(y_i, \overline{y_i} + P) \approx \mathcal{L}(y_i, \overline{y_i}) + \left[\frac{d}{d\overline{y_i}}\mathcal{L}(y_i, \overline{y_i})\right] P + \frac{1}{2}\left[\frac{d}{d\overline{y_i^2}}\mathcal{L}(y_i, \overline{y_i})\right] P^2$$

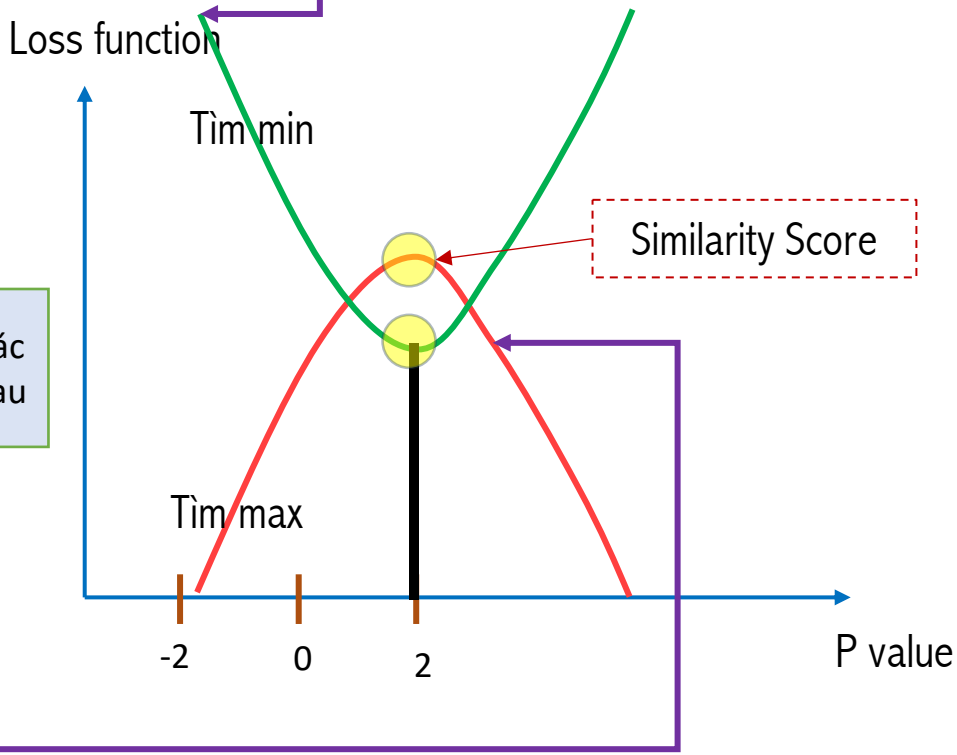
$$\mathcal{L}\left(y_i, \overline{y_i^0} + P\right) \approx \mathcal{L}(y_i, \overline{y_i}) + gP + \frac{1}{2}hP^2$$

2

$$\text{-1 X } (g_1 + g_2 + \cdots + g_n)P + \text{-1 X } \frac{1}{2}(h_1 + h + \cdots + h_n + \lambda)P^2$$

Khác nhau

Cả (1) và (2) đều có cùng optimization point P



$g_i = -(y_i - \overline{y_i})$

$h_i = \overline{y_i} \times (1 - \overline{y_i})$

$$\text{Similarity Score} = \frac{1}{2}\frac{-(g_1 + g_2 + \cdots + g_n)^2}{(h_1 + h_2 + \cdots + h_n + \lambda)}$$

$$\text{Similarity Score} = \frac{(\sum \text{Residual})^2}{\sum \overline{y_i} \times (1 - \overline{y_i}) + \lambda}$$

Science

Outline

- Regularization
- XGBoost For Regression
- XGBoost For Classification
- XGBoost: Mathematical Explanation
- XGBoost: Optimization
- Example

How to Handle Missing Value

Sparse-Aware Split Finding

Initial Prediction for Drug Effectiveness

0.5

Dosage	Drug Effectiveness
10	-7
???	-3
21	7
25	8
5	-5
???	-2

Dosage	Drug Effectiveness	Residual
10	-7	-7.5
???	-3	-3.5
21	7	6.5
25	8	7.5
5	-5	-5.5
???	-2	-2.5

How to Handle Missing Value

Table without missing values

Dosage	Drug Effectiveness	Residual
5	-5	-5.5
10	-7	-7.5
21	7	6.5
25	8	7.5

Build tree

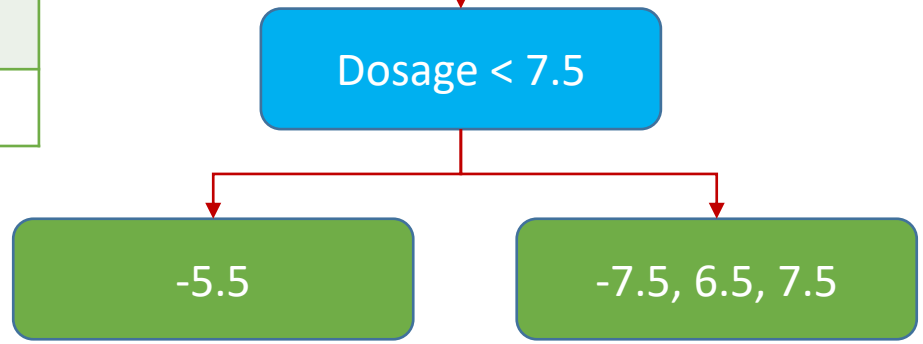


Table with missing values

Dosage	Drug Effectiveness	Residual
???	-3	-3.5
???	-2	-2.5

How to Handle Missing Value

Table without missing values

Dosage	Drug Effectiveness	Residual
5	-5	-5.5
10	-7	-7.5
21	7	6.5
25	8	7.5

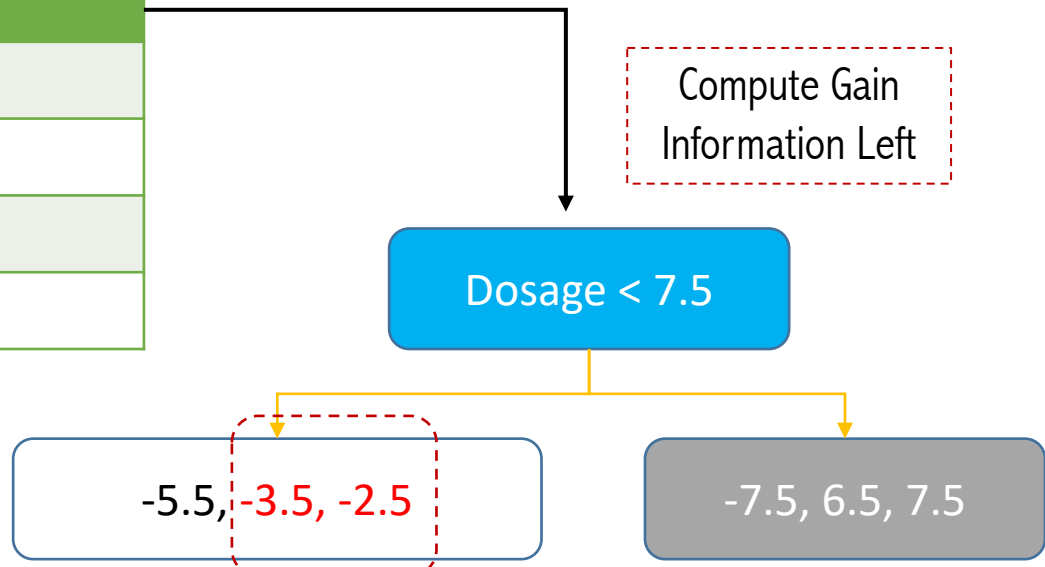
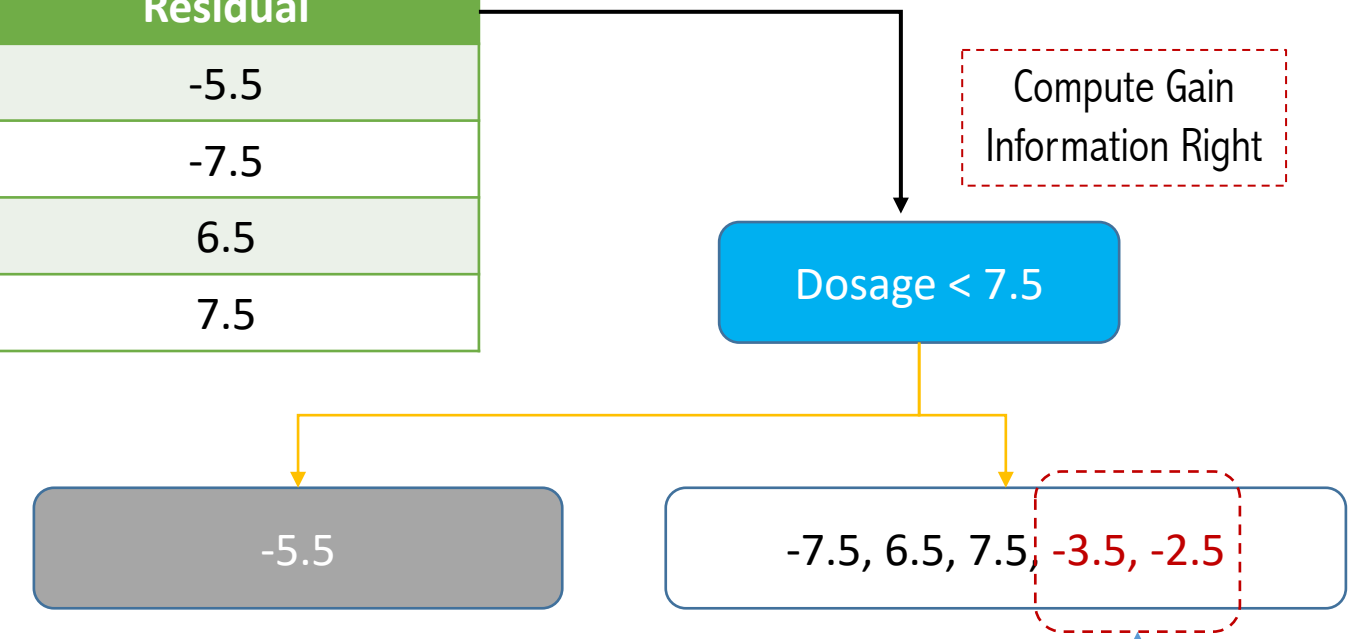


Table with missing values

Dosage	Drug Effectiveness	Residual
???	-3	-3.5
???	-2	-2.5

How to Handle Missing Value

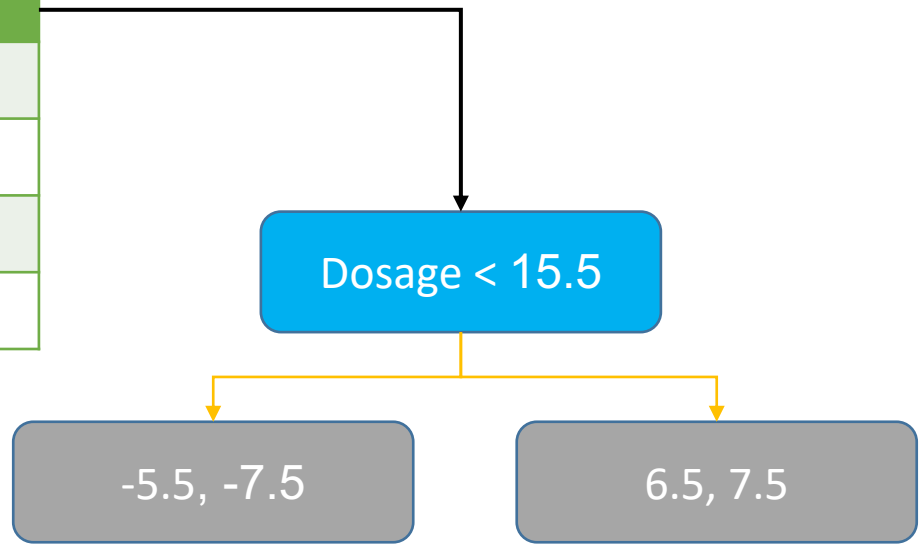
Dosage	Drug Effectiveness	Residual
5	-5	-5.5
10	-7	-7.5
21	7	6.5
25	8	7.5



Dosage	Drug Effectiveness	Residual
???	-3	-3.5
???	-2	-2.5

How to Handle Missing Value

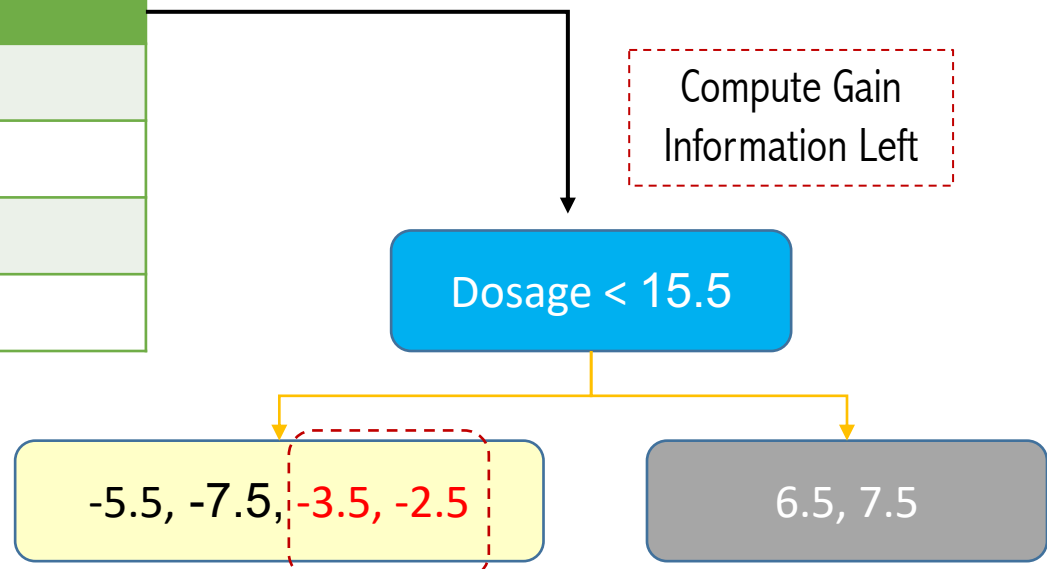
Dosage	Drug Effectiveness	Residual
5	-5	-5.5
10	-7	-7.5
21	7	6.5
25	8	7.5



Dosage	Drug Effectiveness	Residual
???	-3	-3.5
???	-2	-2.5

How to Handle Missing Value

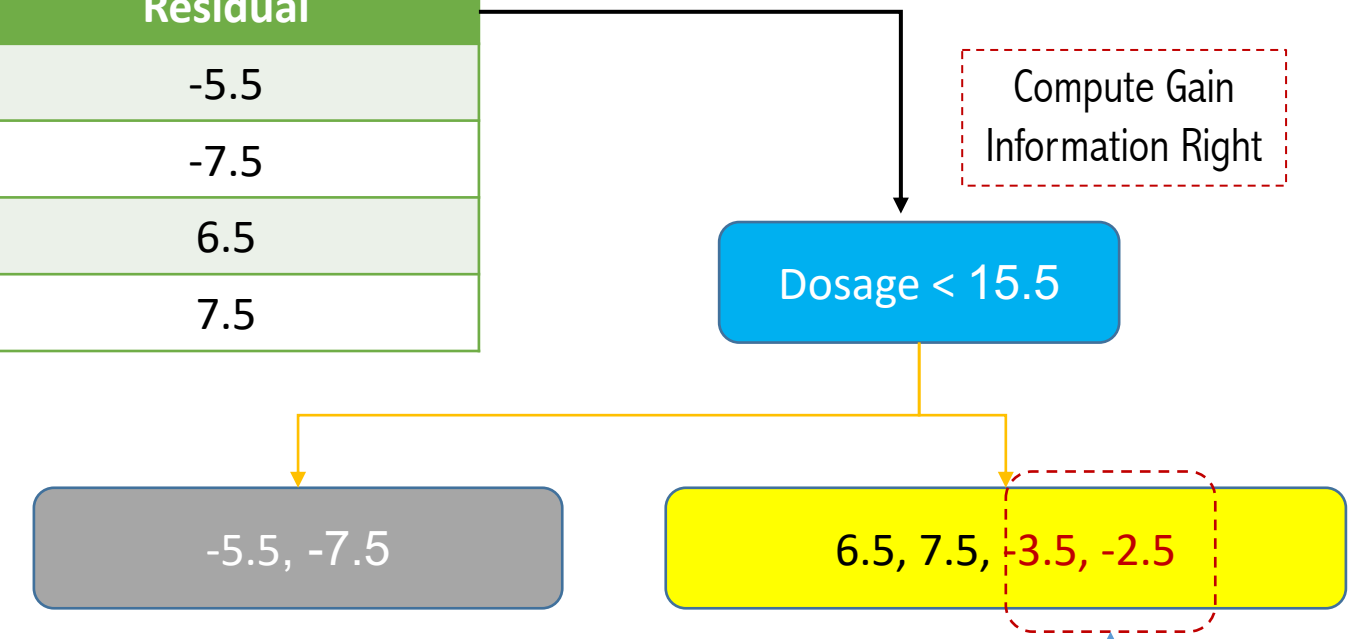
Dosage	Drug Effectiveness	Residual
5	-5	-5.5
10	-7	-7.5
21	7	6.5
25	8	7.5



Dosage	Drug Effectiveness	Residual
???	-3	-3.5
???	-2	-2.5

How to Handle Missing Value

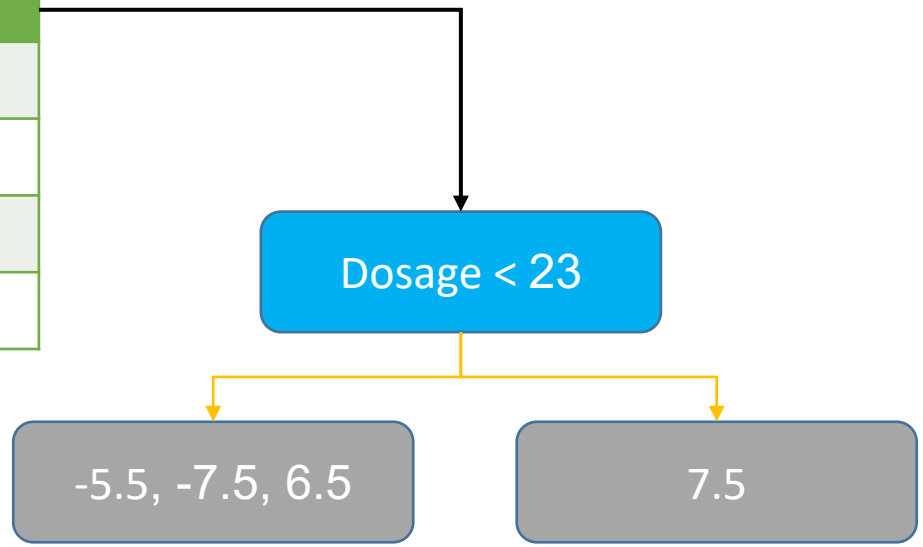
Dosage	Drug Effectiveness	Residual
5	-5	-5.5
10	-7	-7.5
21	7	6.5
25	8	7.5



Dosage	Drug Effectiveness	Residual
???	-3	-3.5
???	-2	-2.5

How to Handle Missing Value

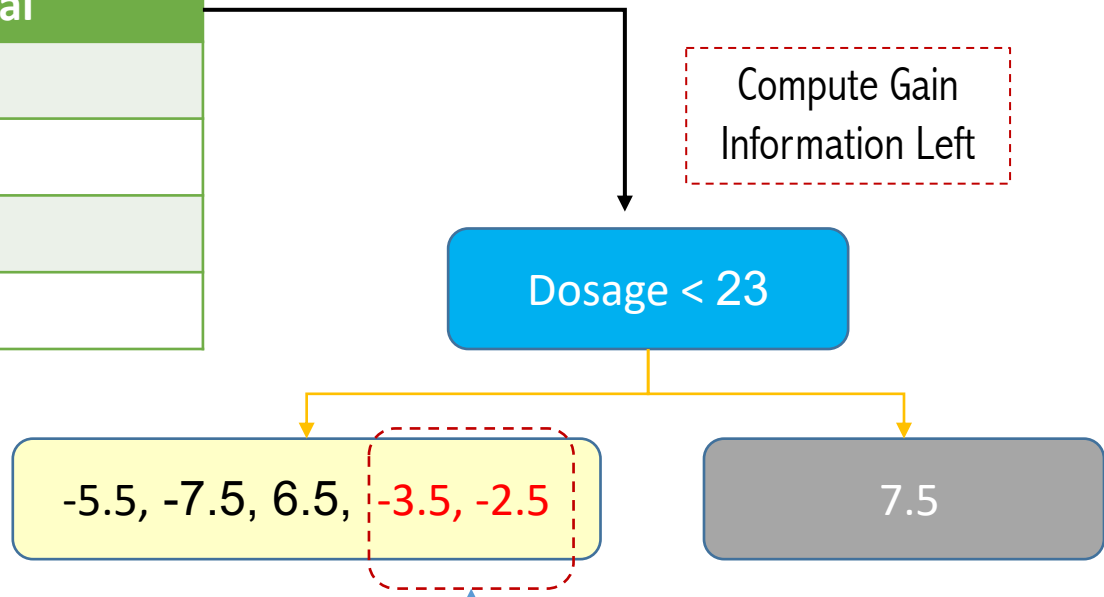
Dosage	Drug Effectiveness	Residual
5	-5	-5.5
10	-7	-7.5
21	7	6.5
25	8	7.5



Dosage	Drug Effectiveness	Residual
???	-3	-3.5
???	-2	-2.5

How to Handle Missing Value

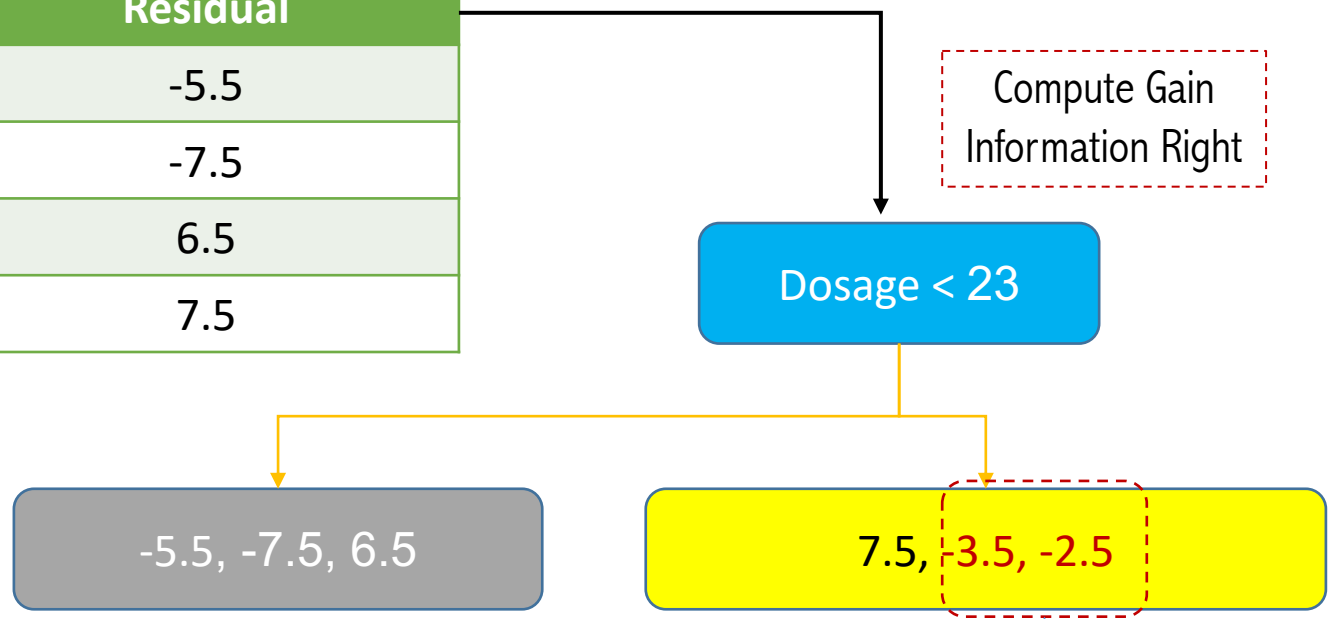
Dosage	Drug Effectiveness	Residual
5	-5	-5.5
10	-7	-7.5
21	7	6.5
25	8	7.5



Dosage	Drug Effectiveness	Residual
???	-3	-3.5
???	-2	-2.5

How to Handle Missing Value

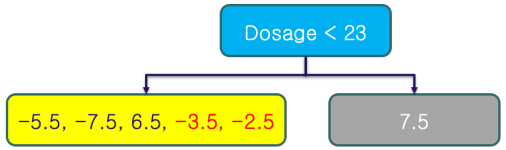
Dosage	Drug Effectiveness	Residual
5	-5	-5.5
10	-7	-7.5
21	7	6.5
25	8	7.5



Dosage	Drug Effectiveness	Residual
???	-3	-3.5
???	-2	-2.5

Dosage	Drug Effectiveness	Residual
5	−5	−5.5
10	−7	−7.5
21	7	6.5
25	8	7.5

Compute Gain value

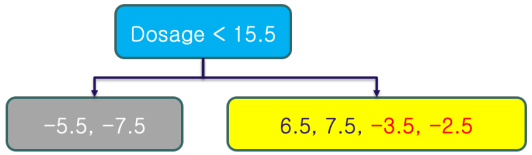


Dosage	Drug Effectiveness	Residual
5	−5	−5.5
10	−7	−7.5
21	7	6.5
25	8	7.5

Compute Gain value

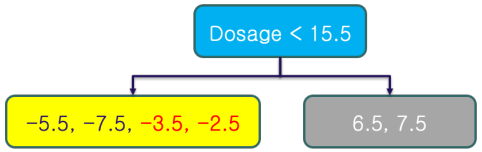
Dosage	Drug Effectiveness	Residual
5	−5	−5.5
10	−7	−7.5
21	7	6.5
25	8	7.5

Compute Gain value



Dosage	Drug Effectiveness	Residual
5	−5	−5.5
10	−7	−7.5
21	7	6.5
25	8	7.5

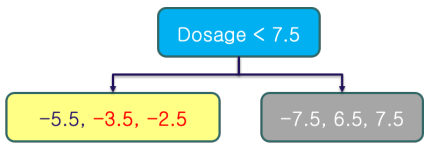
Compute Gain value



MAXIUM GAIN VALUE

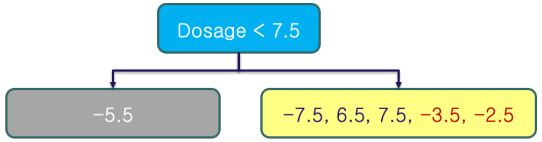
Dosage	Drug Effectiveness	Residual
10	−7	−7.5
21	7	6.5
25	8	7.5
5	−5	−5.5

Compute Gain value



Dosage	Drug Effectiveness	Residual
10	−7	−7.5
21	7	6.5
25	8	7.5
5	−5	−5.5

Compute Gain value

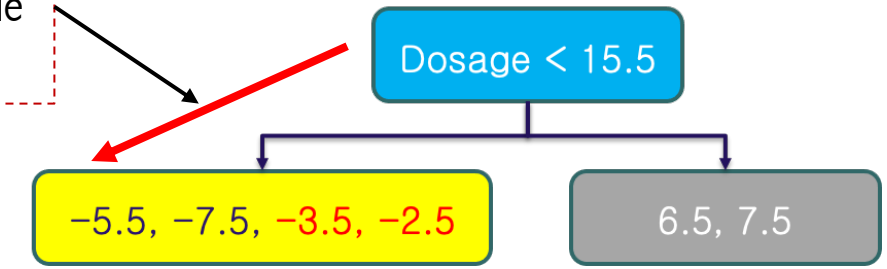


How to Handle Missing Value

Dosage	Drug Effectiveness	Residual
5	−5	−5.5
10	−7	−7.5
21	7	6.5
25	8	7.5

Compute Gain value

Rẻ nhánh mặc định cho tất cả các missing value trong Dosage



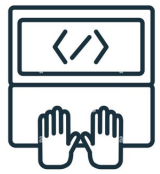
Dosage	Drug Effectiveness	Residual
???	−3	−3.5
???	−2	−2.5

Outline

- Regularization
- XGBoost For Regression
- XGBoost For Classification
- XGBoost: Mathematical Explanation
- How To Fill Missing Values
- Example

Sales Prediction

TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2
8.6	2.1	1	4.8
199.8	2.6	21.2	15.6



Sales Prediction

```
class AIVN_XGBoostRegressor():
    '''XGBoost from Scratch
    '''

    def __init__(self, params, random_seed=None):
```

```
import numpy as np

class SquaredErrorObjective():
    def loss(self, y, pred): return np.mean((y - pred)**2)
    def gradient(self, y, pred): return pred - y
    def hessian(self, y, pred): return np.ones(len(y))
```

```
params = {
    'learning_rate': 0.1,
    'max_depth': 5,
    'subsample': 0.8,
    'reg_lambda': 1.5,
    'gamma': 0.0,
    'min_child_weight': 25,
    'base_score': 0.0,
    'tree_method': 'exact',
}
num_boost_round = 50

# train the from-scratch XGBoost model
model_scratch = AIVN_XGBoostRegressor(params, random_seed=42)
model_scratch.fit(X_train, y_train, SquaredErrorObjective(), num_boost_round)
```

```
pred_scratch = model_scratch.predict(X_test)
print(f'Our implementation: {SquaredErrorObjective().loss(y_test, pred_scratch)}')
```

```
scratch score: 1.7017064684759806
```


```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

#READ DATA
data = pd.read_csv("/content/advertising.csv")

#X,y
X = data.iloc[:,:-1]
y = data.iloc[:,-1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=100)
```

```
from xgboost import XGBRegressor

model = XGBRegressor()
model.fit(X_train, y_train)

pred_scratch = model.predict(X_test)
print(f'XGBRegressor Lib: {SquaredErrorObjective().loss(y_test, pred_scratch)}')
```

```
XGBRegressor Lib: 1.7209355894486629
```


Thank
you