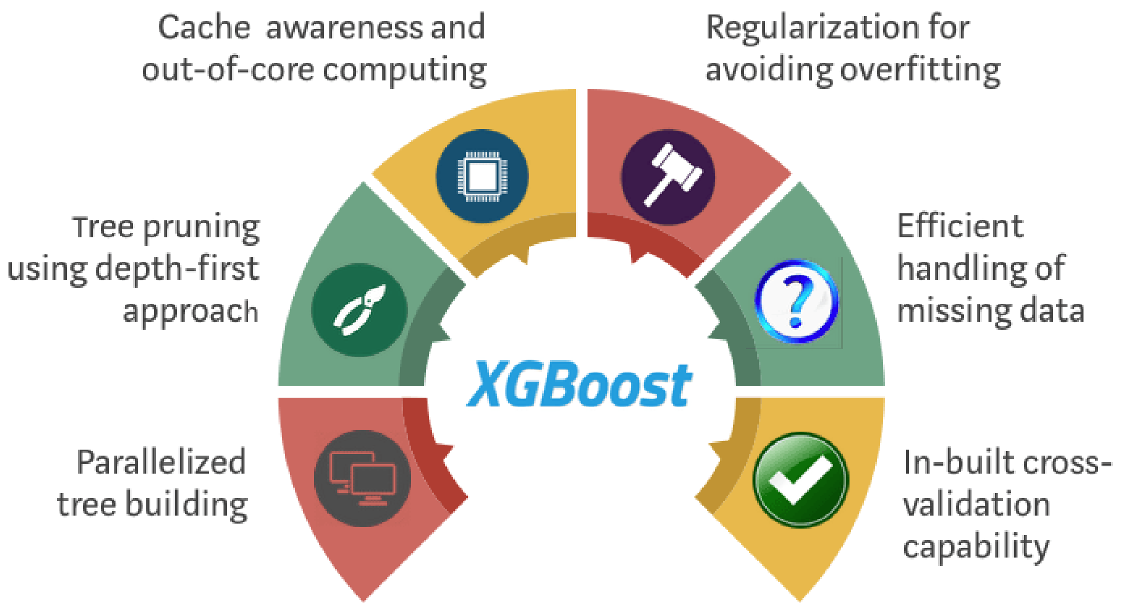
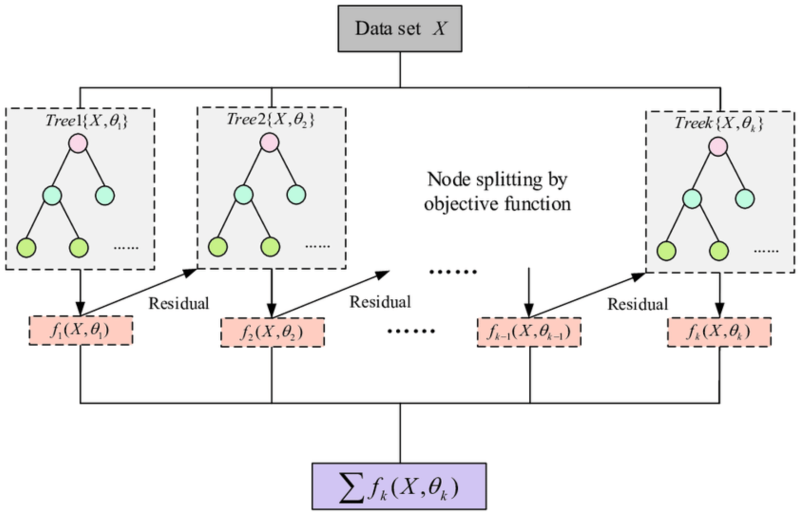
AI VIET NAM
@aivietnam.edu.vn

XGBoost

(Basic, Advanced Concepts and Its Applications)





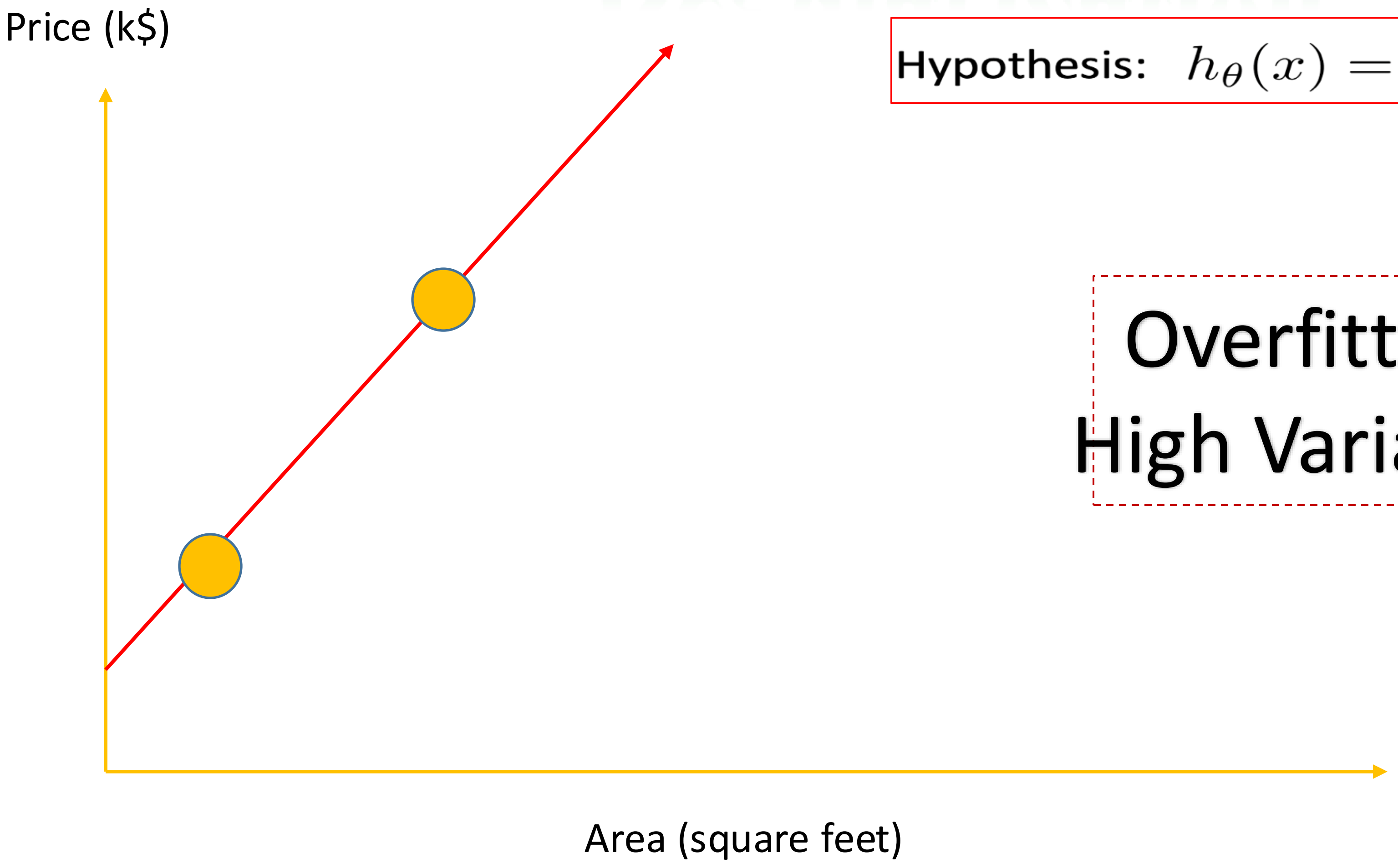
Vinh Dinh Nguyen
PhD in Computer Science

Outline

- Regularization
- Regression XGBoost
- Classification XGBoost
- XGBoost: Clearly Explain
- Time Series Example
- Summary

AI VIET NAM
@aivietnam.edu.vn

Regularization

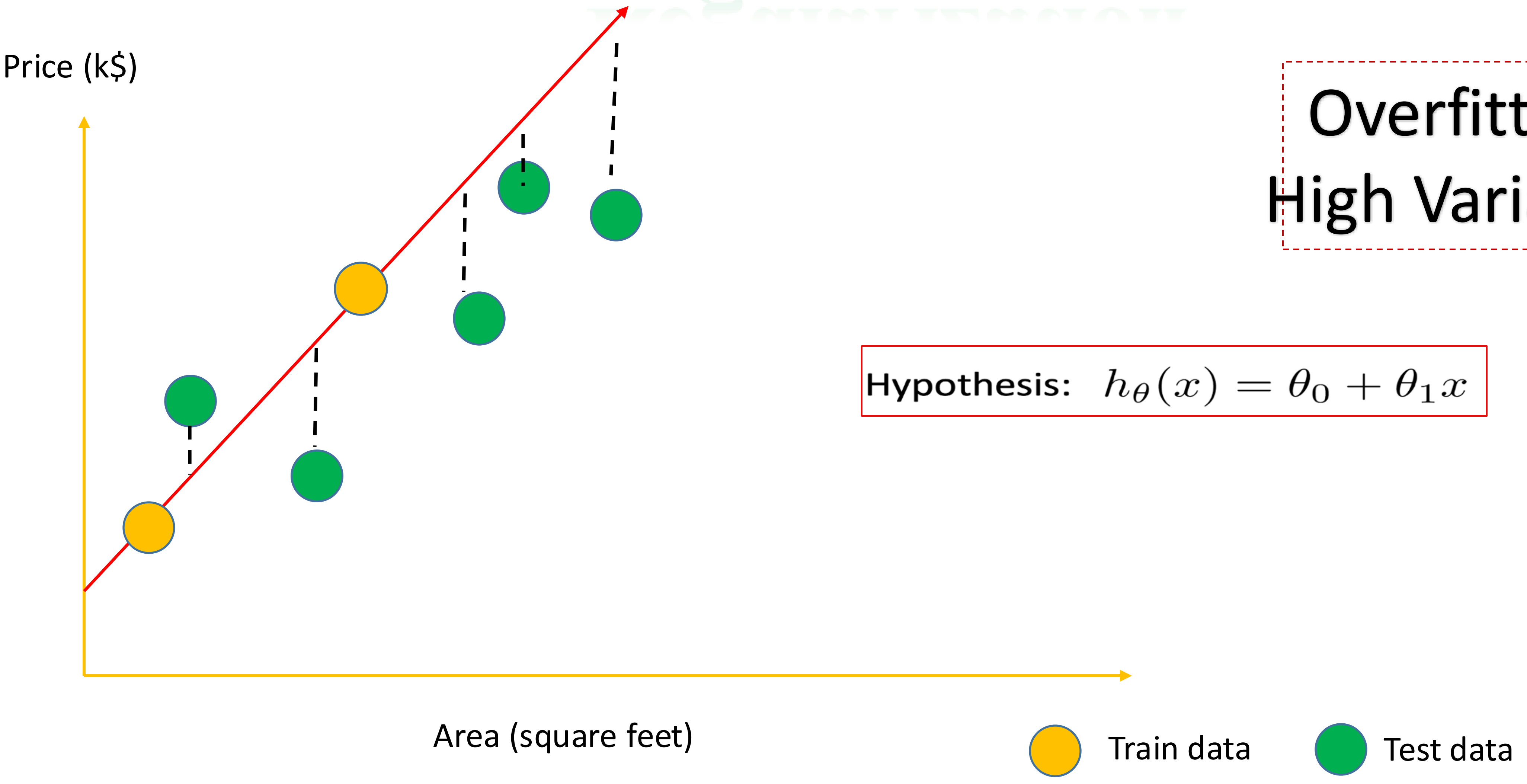


Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Overfitting
High Variance

AI VIET NAM
@aivietnam.edu.vn

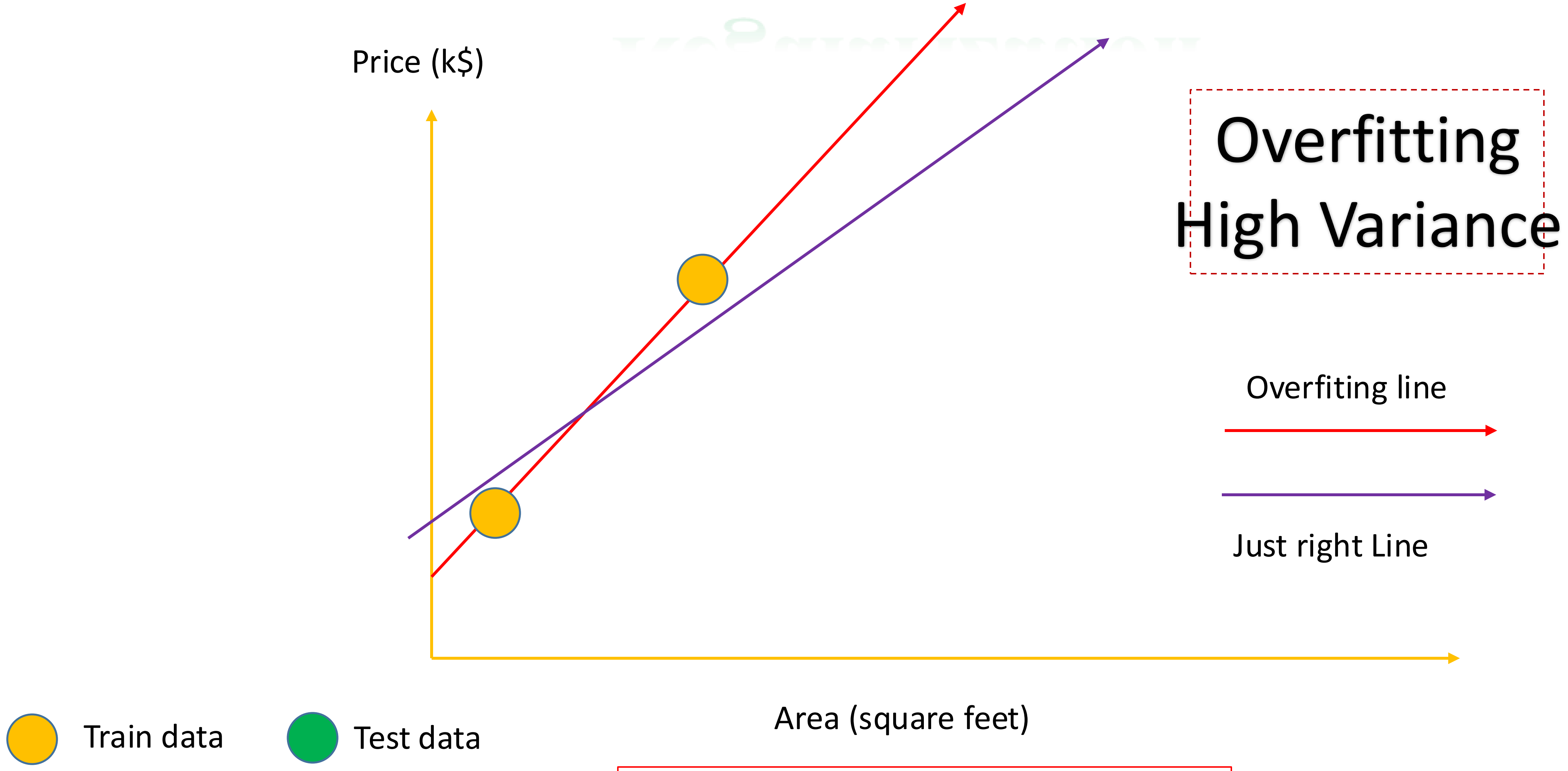
Regularization



Overfitting
High Variance

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

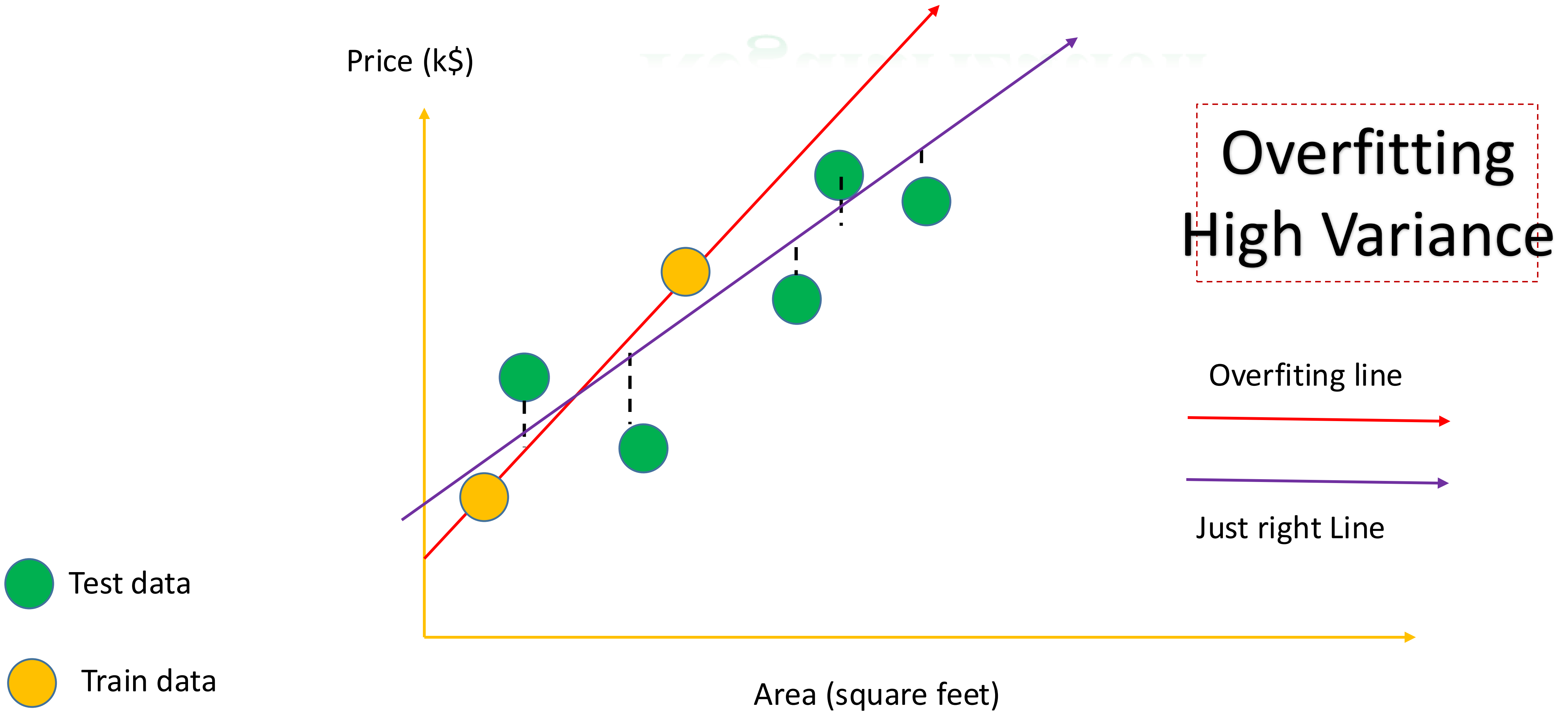
Regularization



$\boxed{\textbf{Hypothesis:} \quad h_\theta(x) = \theta_0 + \theta_1 x}$

AI VIET NAM
@aivietnam.edu.vn

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.

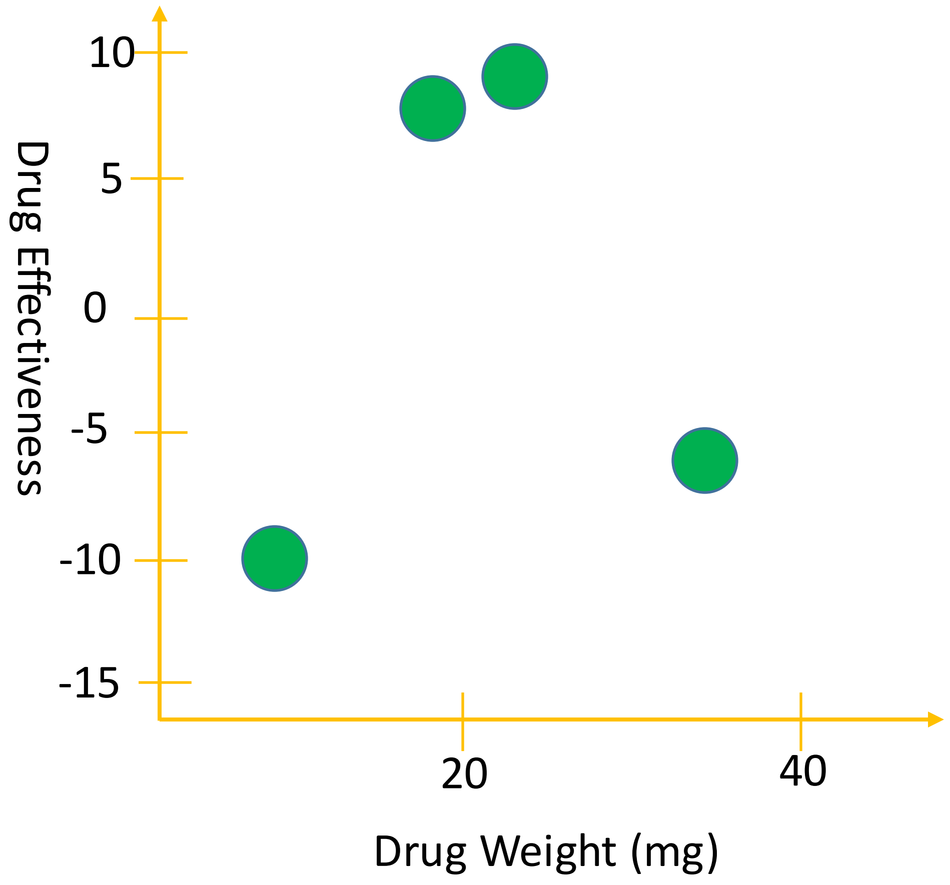
$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$

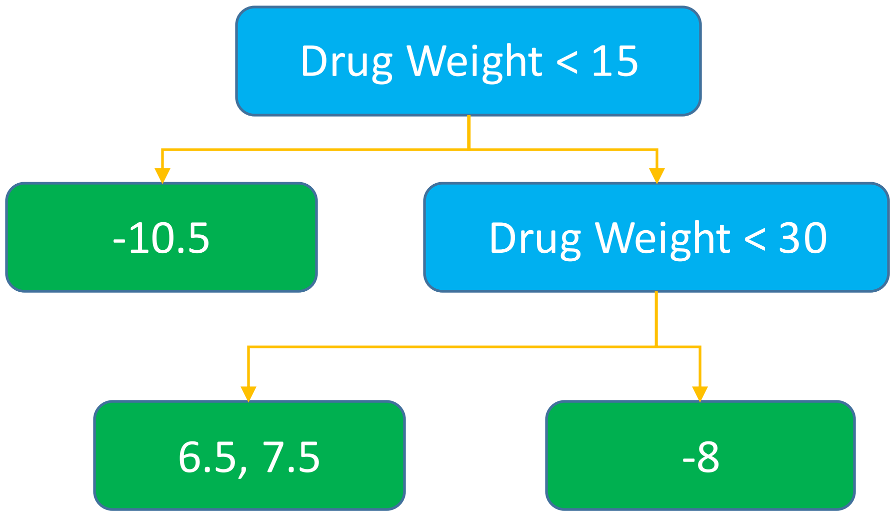
Price = [Intercept + slope * area] + λ*slope^2

Outline

- Regularization
- Regression XGBoost
- Classification XGBoost
- XGBoost: Clearly Explain
- Time Series Example
- Summary

XGBoost For Regression



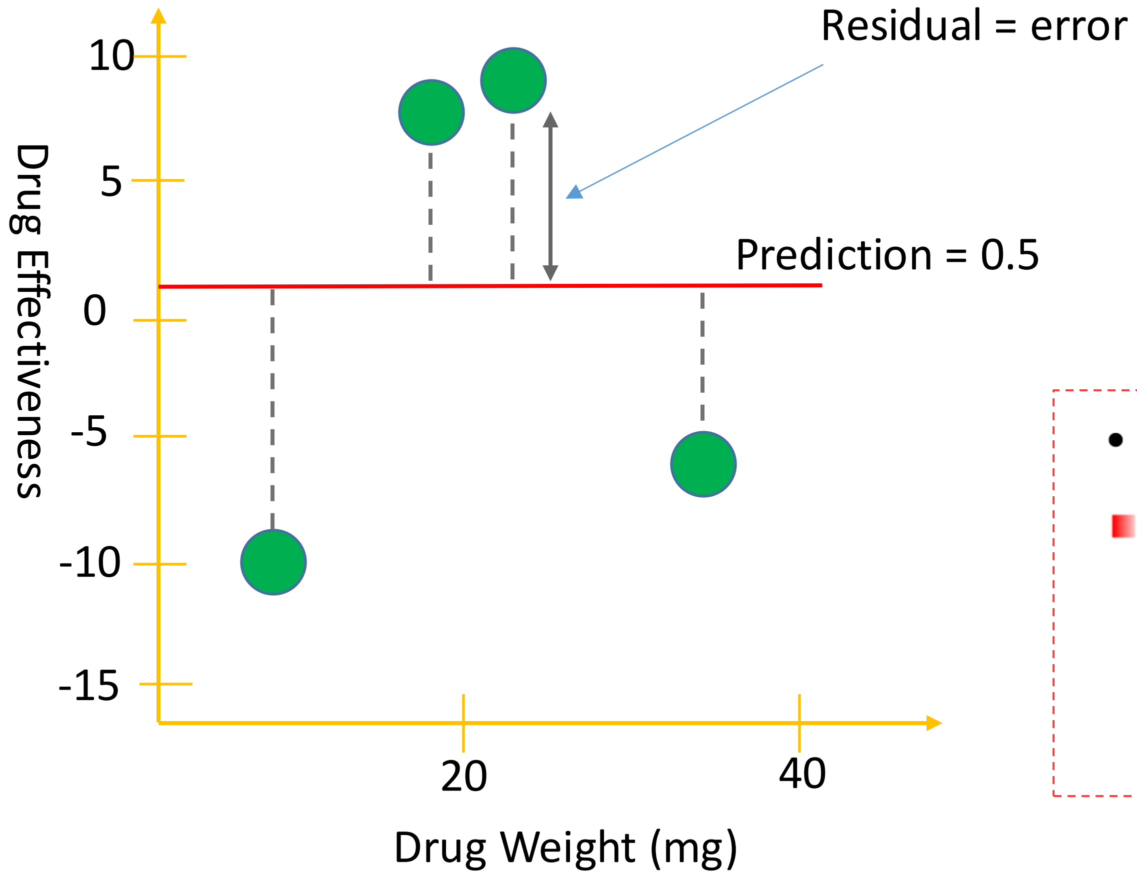


AI VIET NAM
@aivietnam.edu.vn

XGBoost For Regression

Step 1

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



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

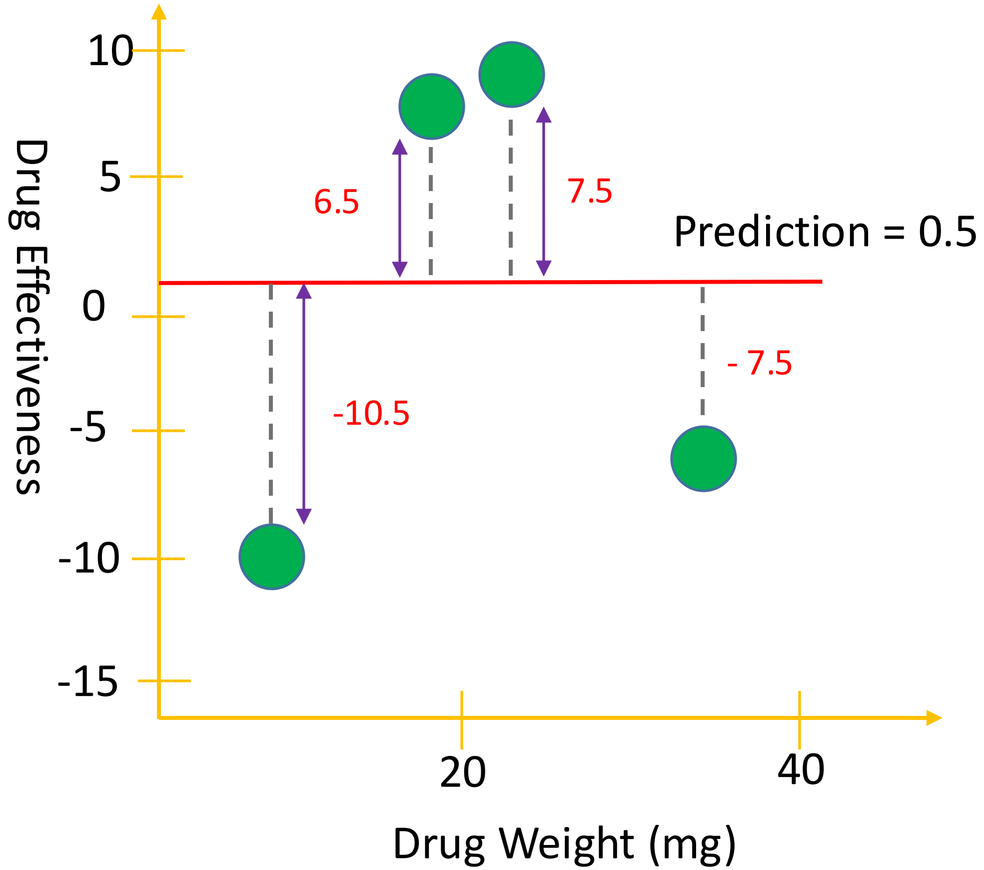
- $\{(x_i, y_i)\}^n_1$
- Loss function = $L(y_i, F(x))$ = 1/2* (Output - Predicted)^2

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

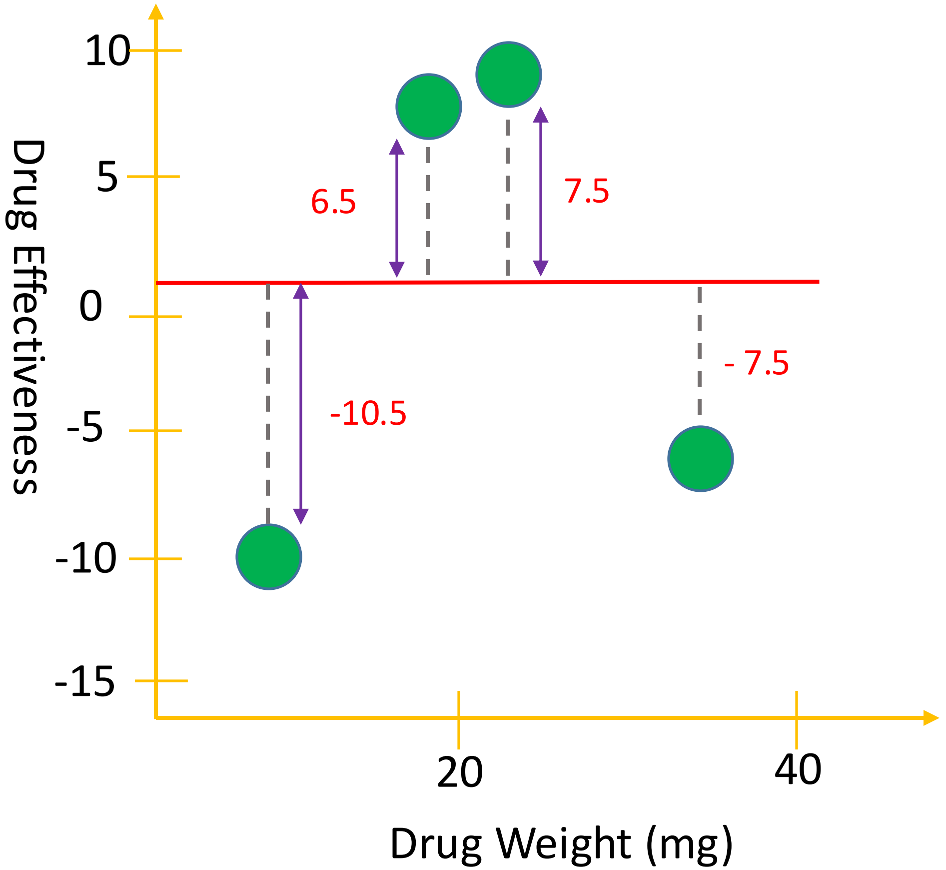
$$SC = \frac{[\sum (output - predicted)]^2}{m + \lambda}$$

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
λ = 0

Compute Similarity Score

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

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

AI VIET NAM
@aivietnam.edu.vn

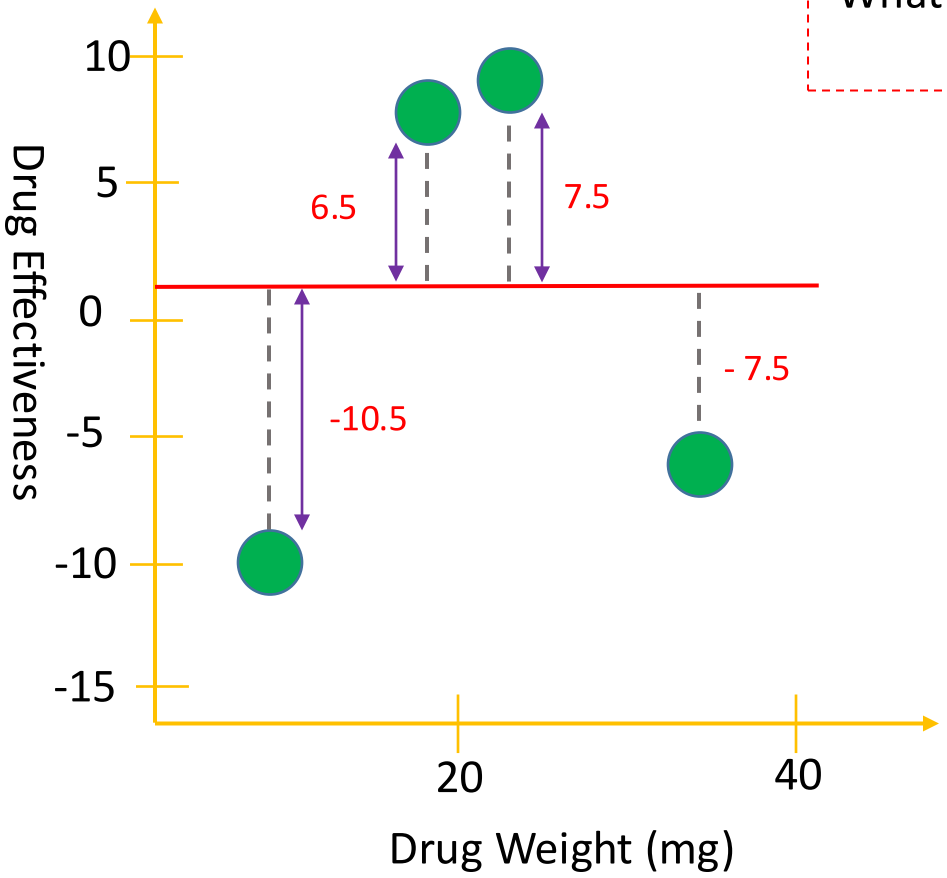
XGBoost For Regression

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



AI VIET NAM
@aivietnam.edu.vn

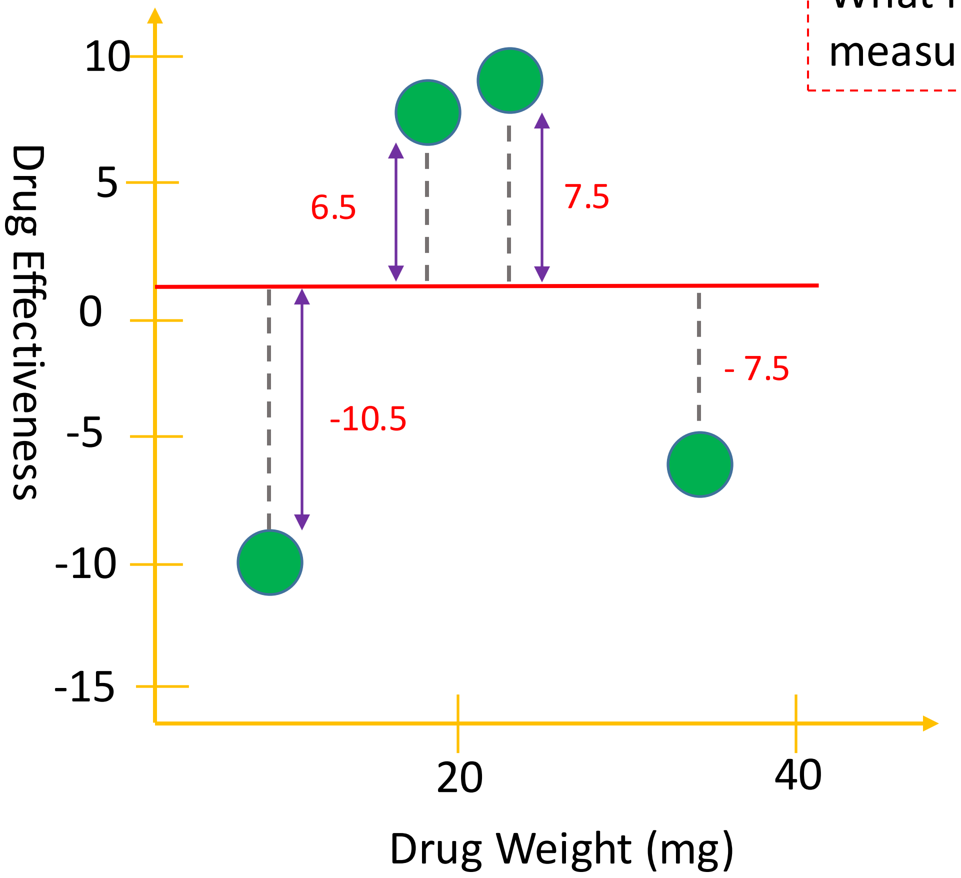
XGBoost For Regression

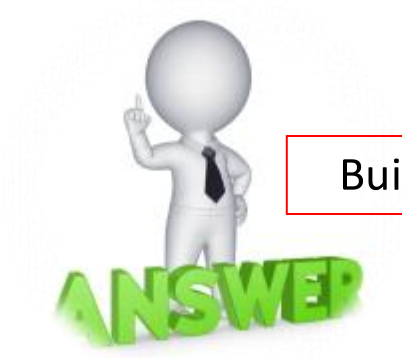
Step 1

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

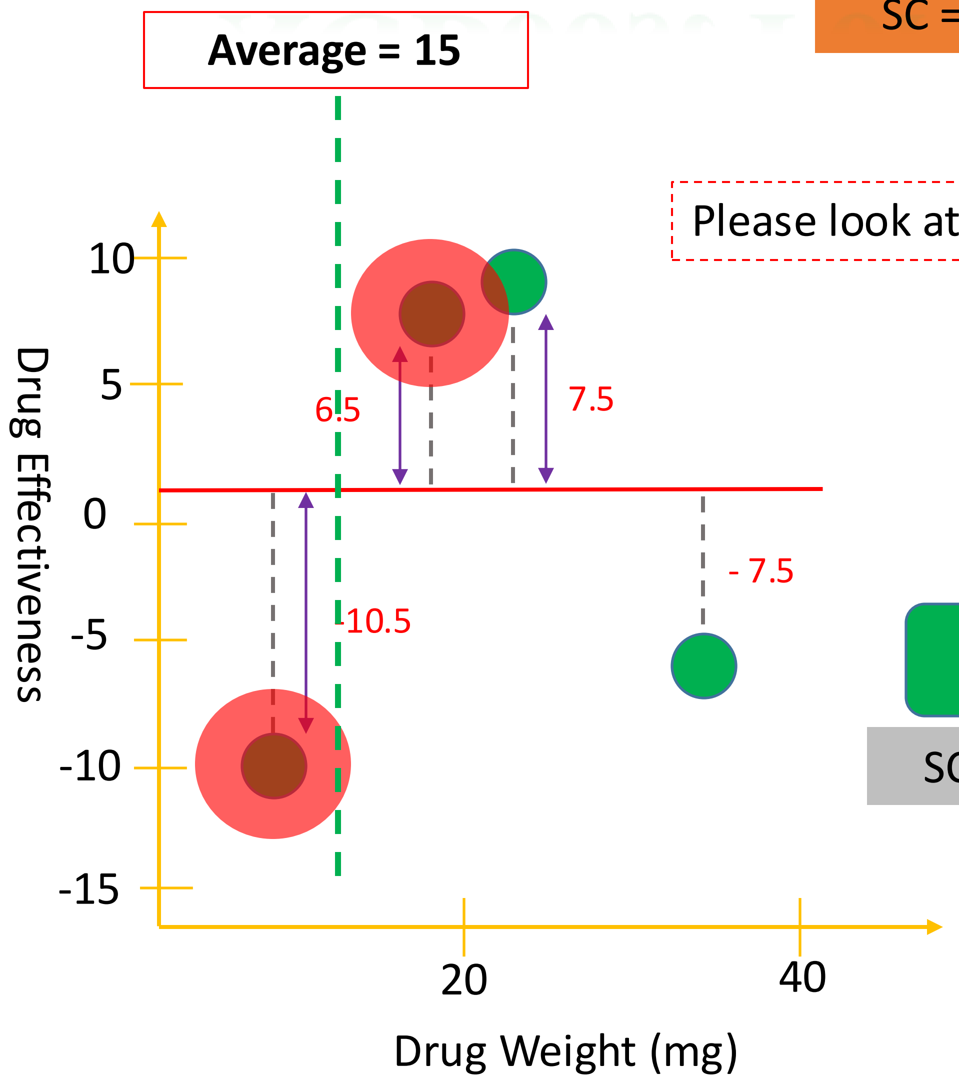


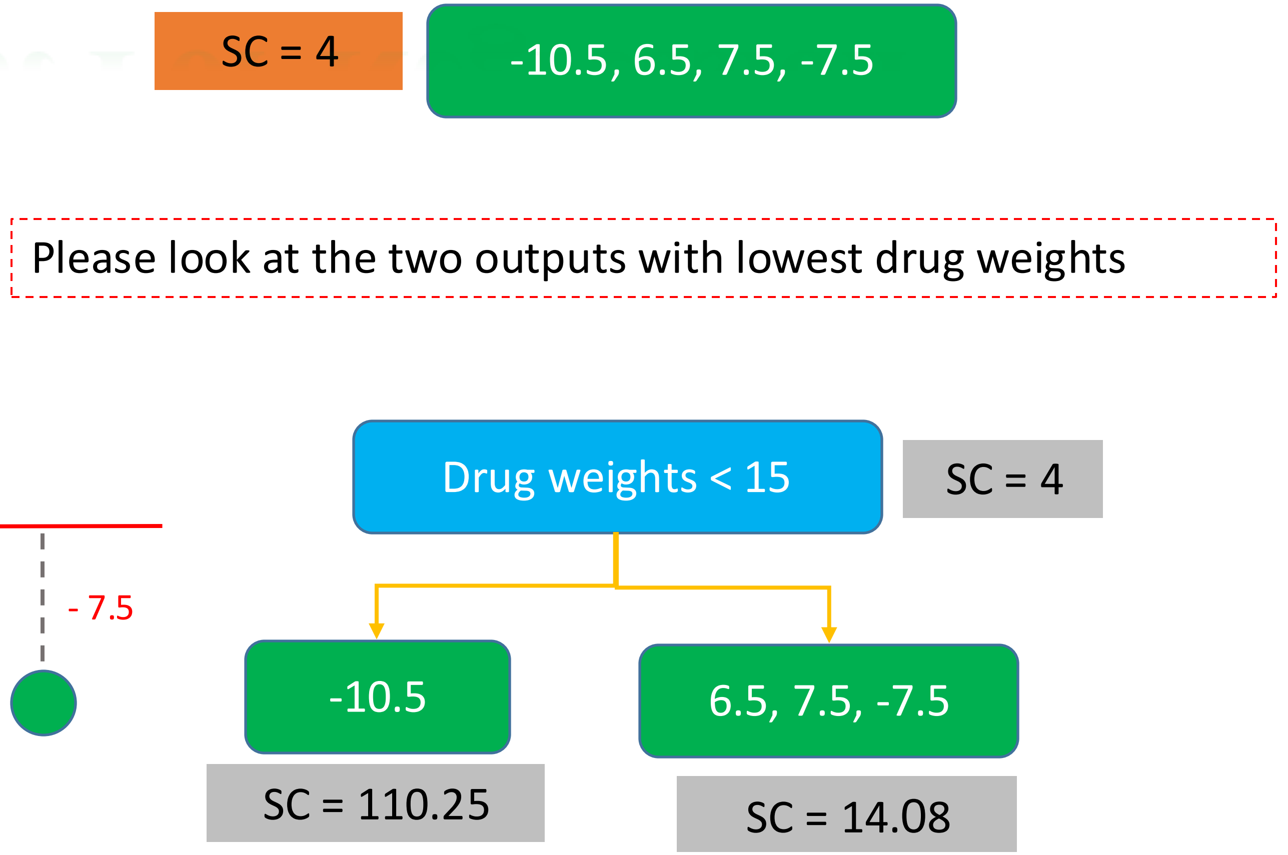


Build a tree on it

XGBoost For Regression

Step 1

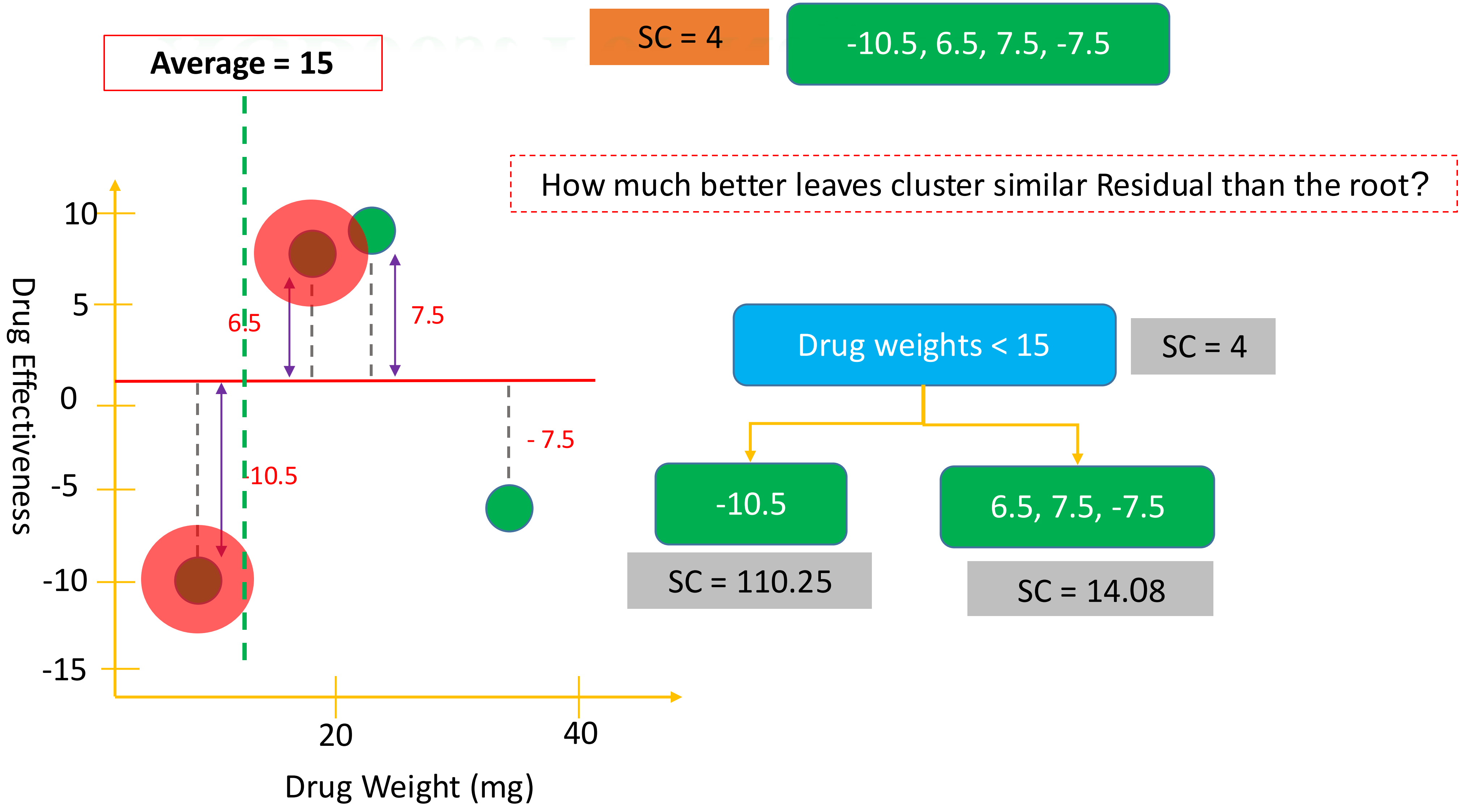




AI VIET NAM
@aivietnam.edu.vn

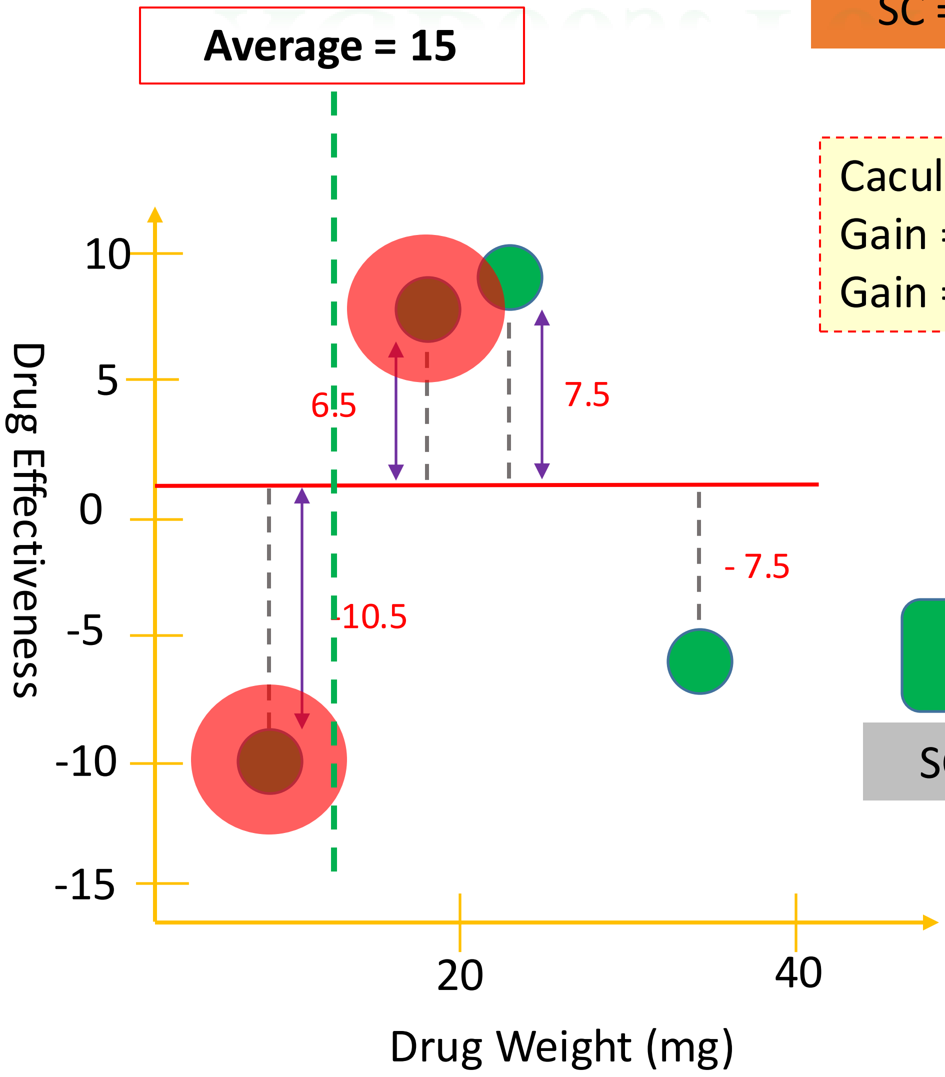
XGBoost For Regression

Step 1



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

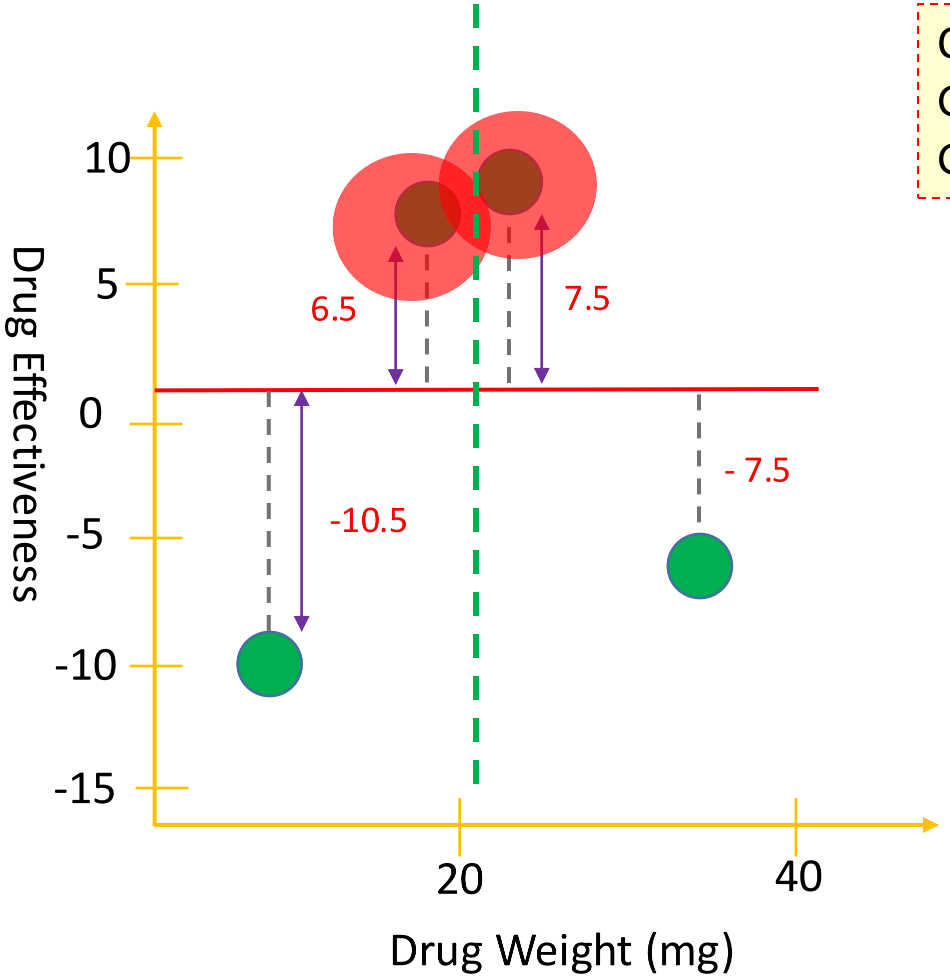
6.5, 7.5, -7.5 | SC = 14.08

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

XGBoost For Regression

❑ *Step 1*

Average = 22.5



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

Drug weights < 22.5

SC = 4

-10.5, 6.5

SC = 8

7.5, -7.5

SC = 0

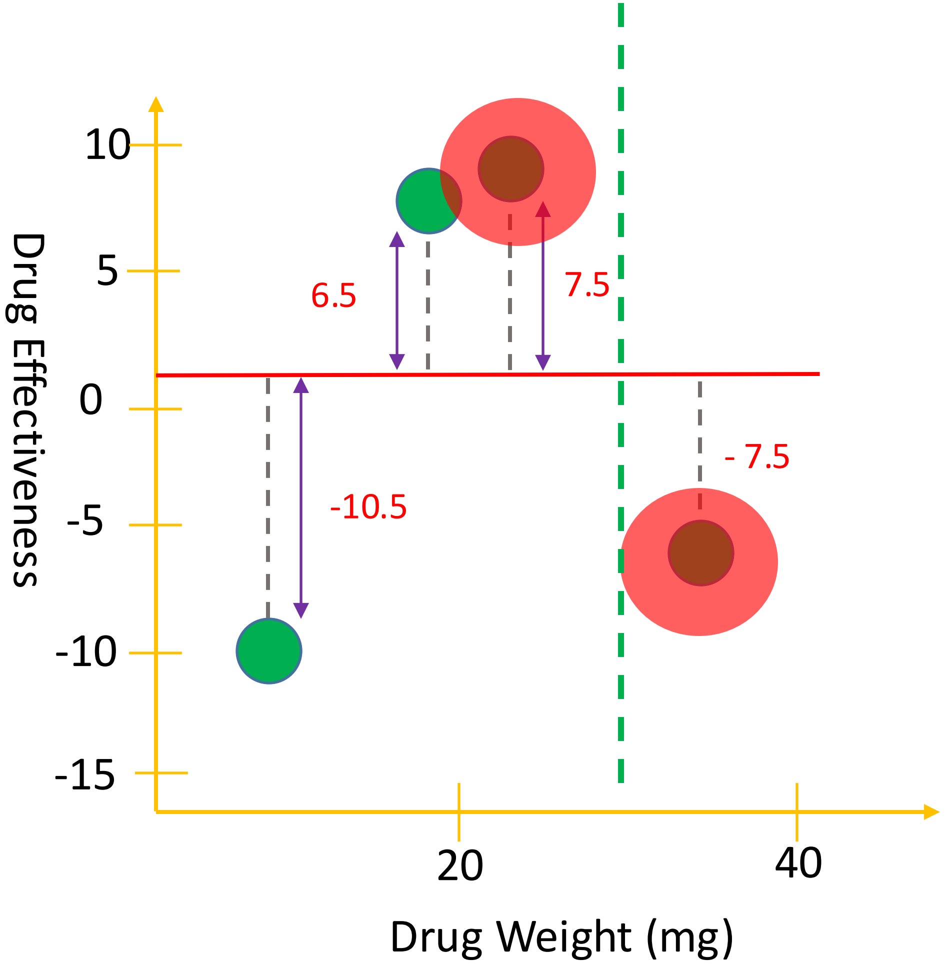
Residual giống nhau hoặc không triệt tiêu lẫn nhau, nên SC khá lơn

AI VIET NAM
@aivietnam.edu.vn

XGBoost For Regression

❑ *Step 1*

Average = 30

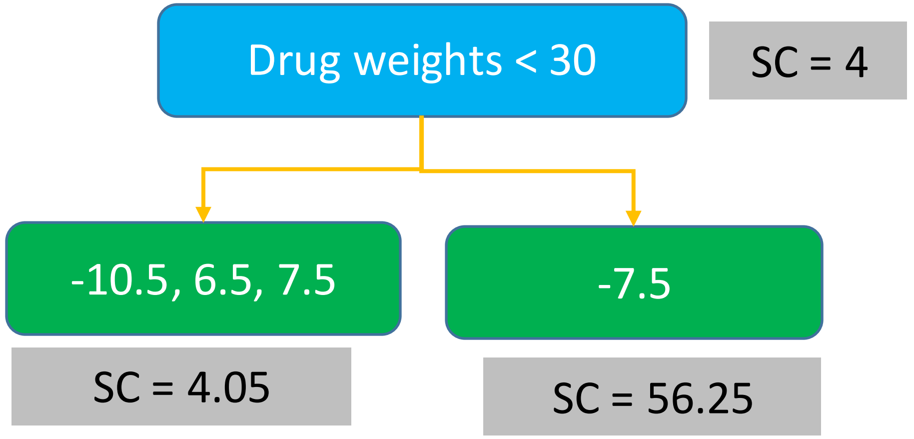


SC = 4

-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

Drug weights < 22.5 | SC = 4

- −10.5, 6.5 | SC = 8
- 7.5, −7.5 | SC = 0

Drug weights < 30 | SC = 4

- -10.5, 6.5, 7.5 | SC = 4.05
- -7.5 | SC = 56.25

Gain = 4 ≤ Gain = 56.25 ≤ Gain = 120.33

Drug weights < 15 | SC = 4

- -10.5 | SC = 110.25
- 6.5, 7.5, -7.5 | SC = 14.08

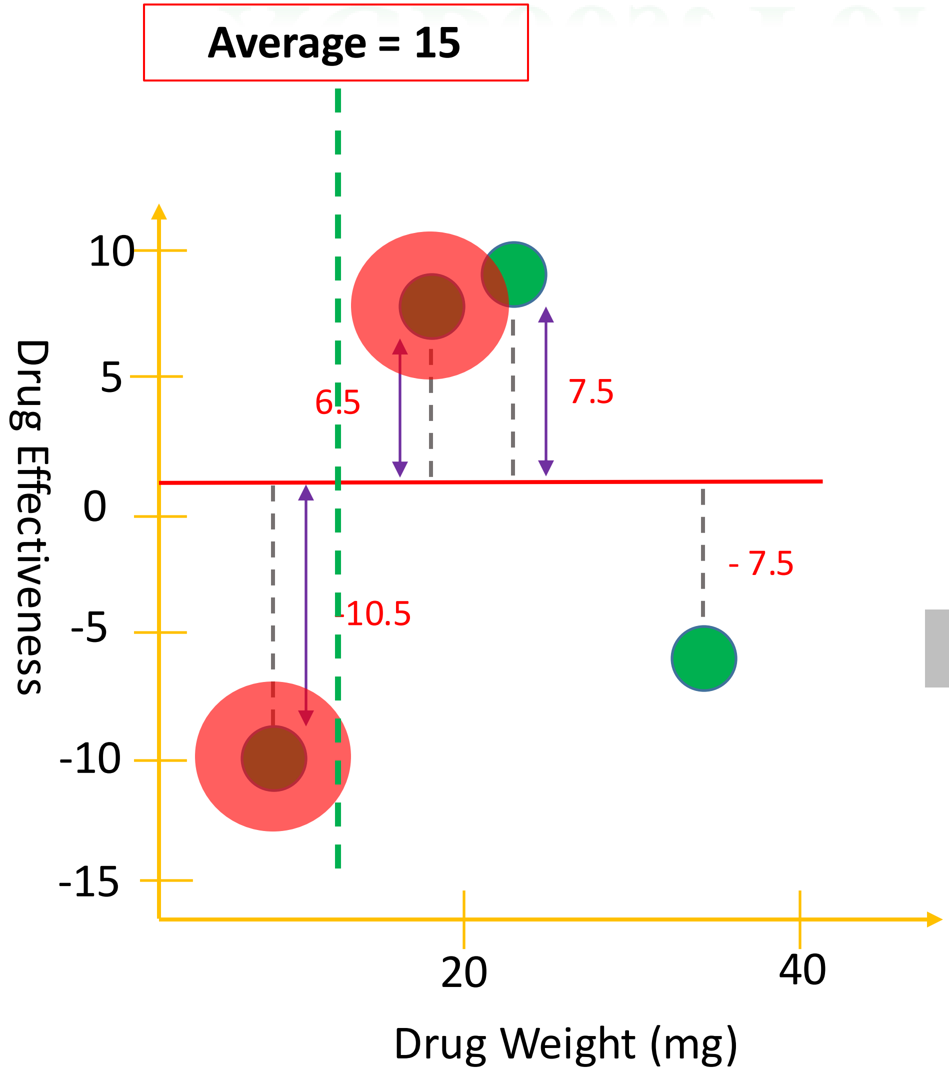
We select

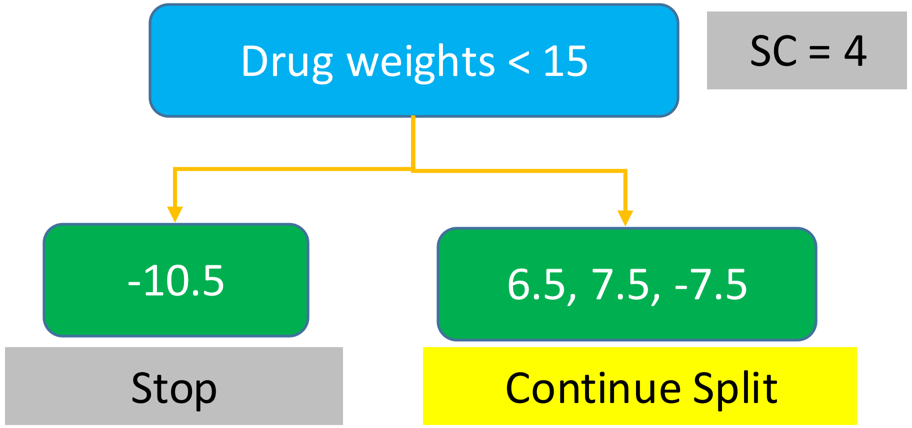
Drug weights < 15

AI VIET NAM
@aivietnam.edu.vn

XGBoost For Regression

❑ *Step 1*

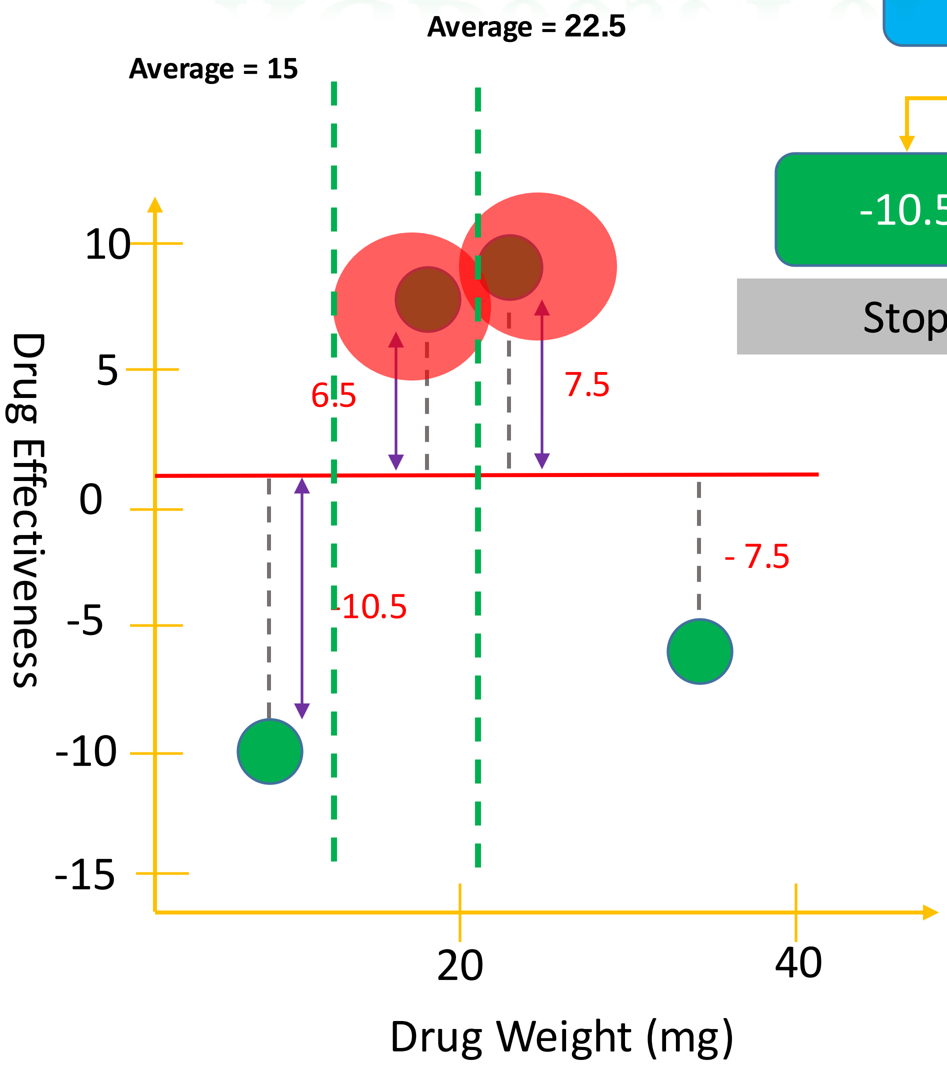


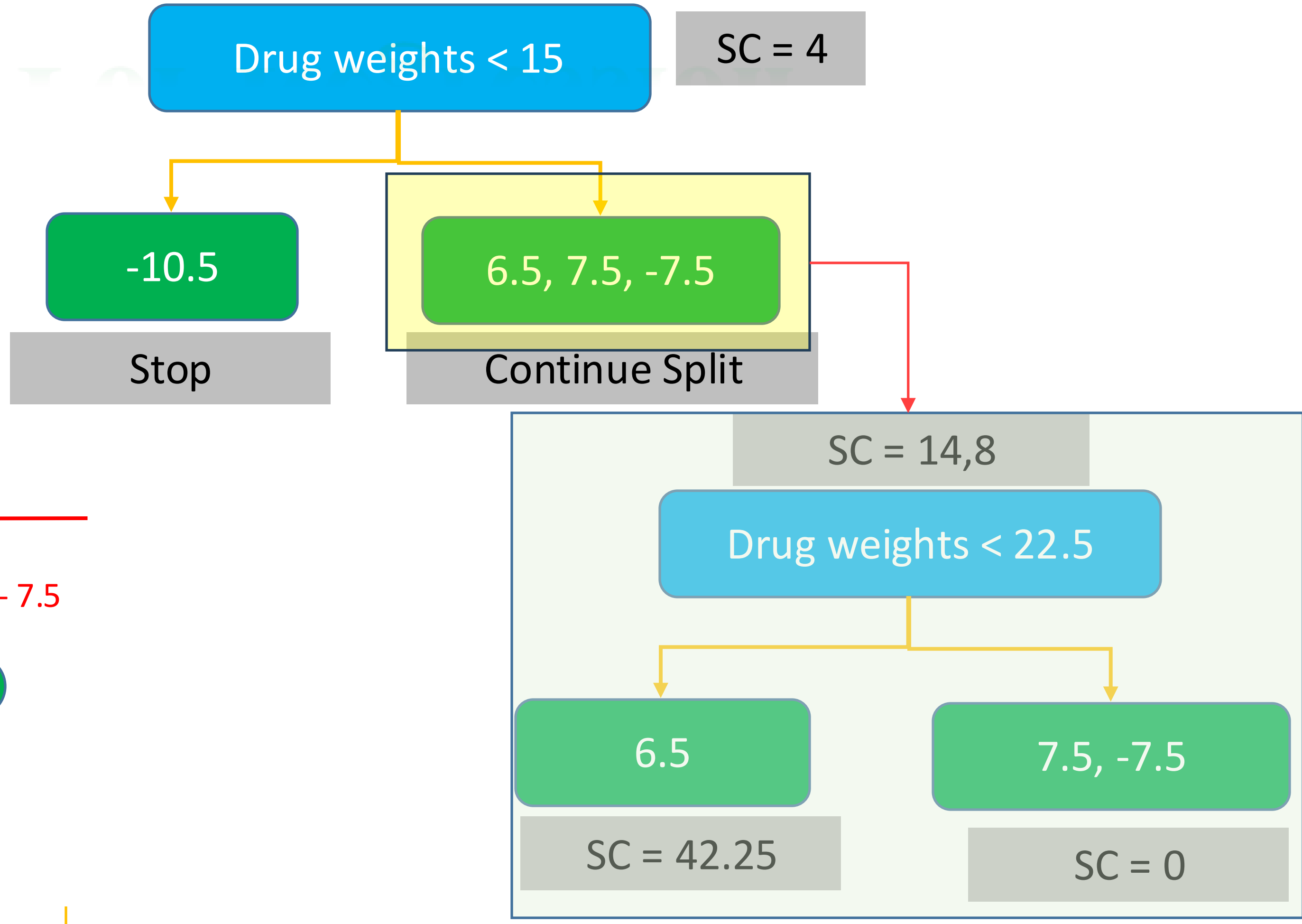


AI VIET NAM
@aivietnam.edu.vn

XGBoost For Regression

❑ *Step 1*

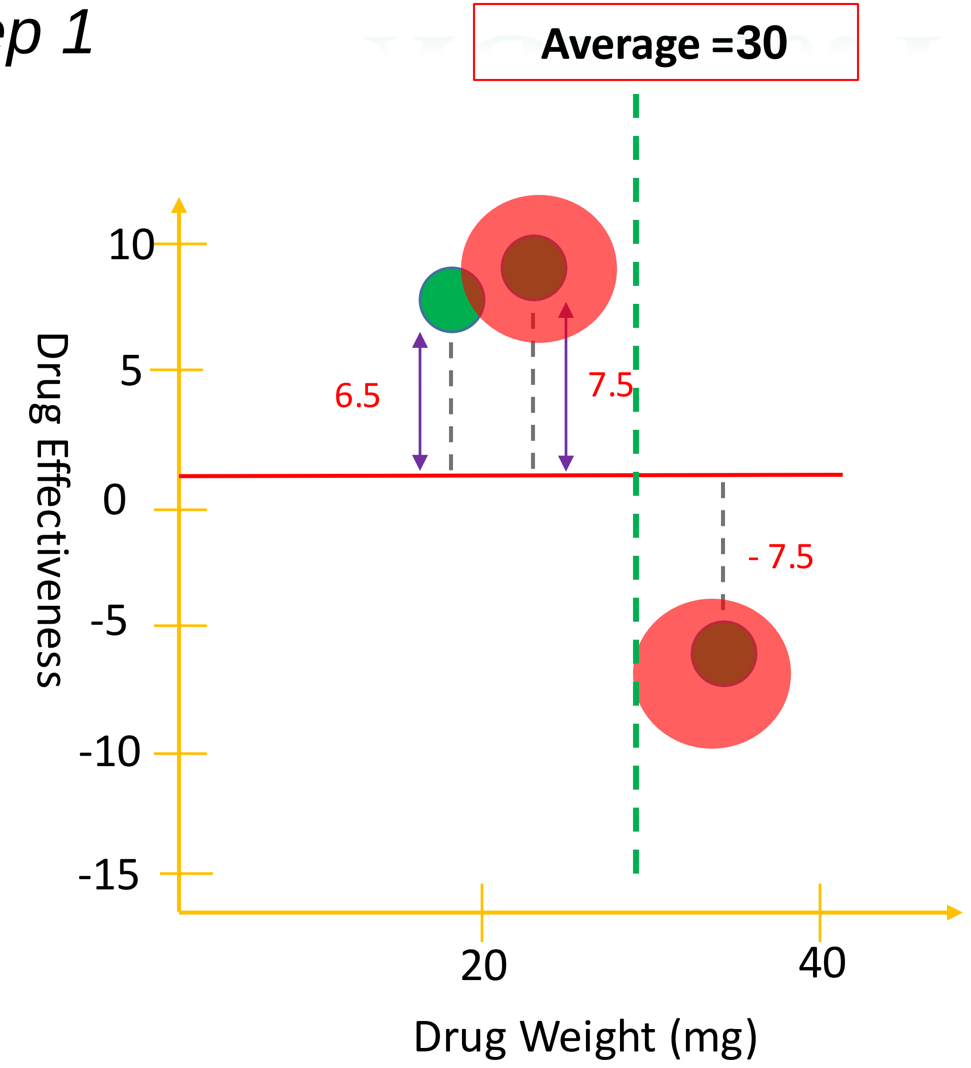


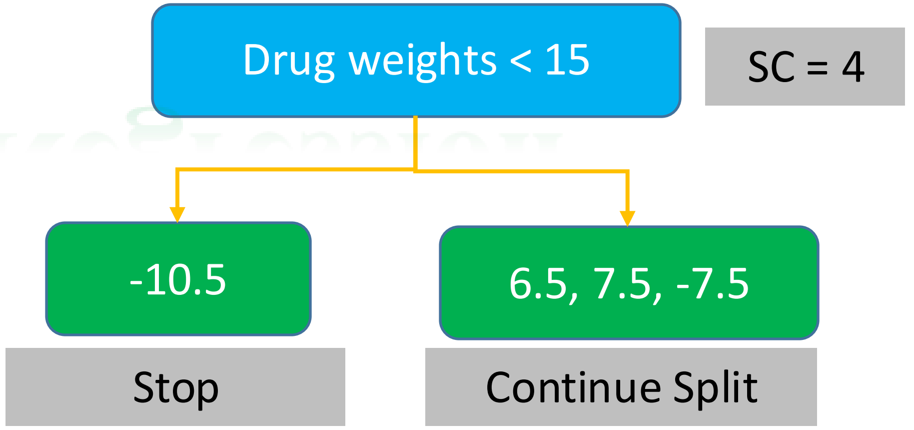


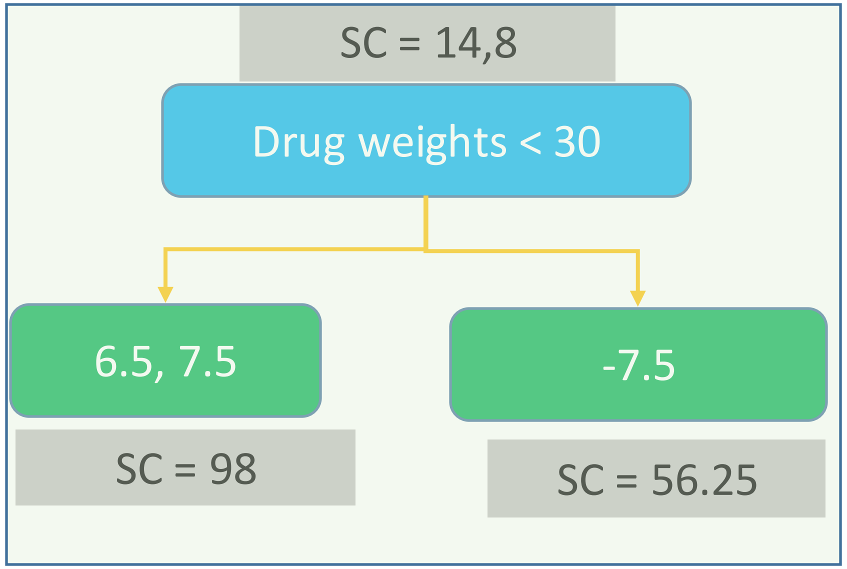
Caculate the Gain
Gain = Left SC + Right SC - Root SC
Gain = 28.17

XGBoost For Regression

❑*Step 1*







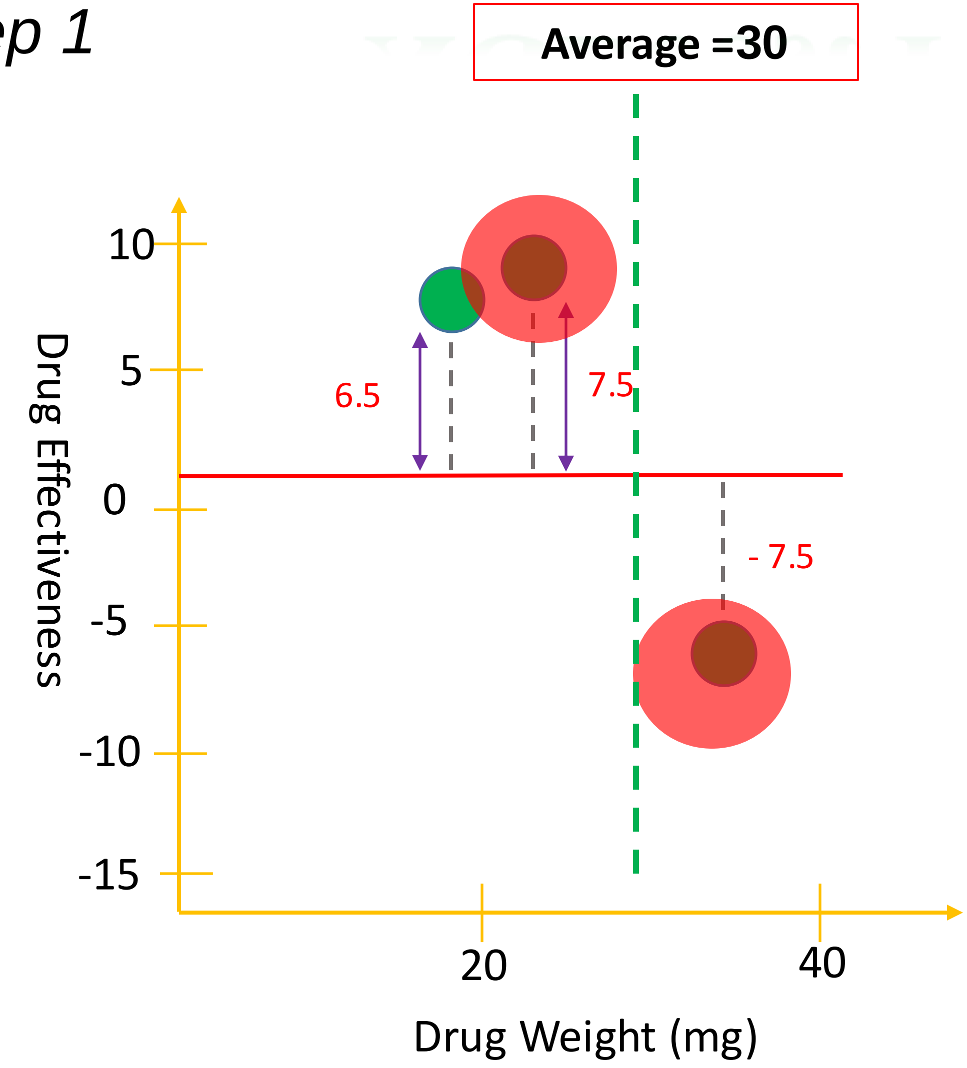
Caculate the Gain

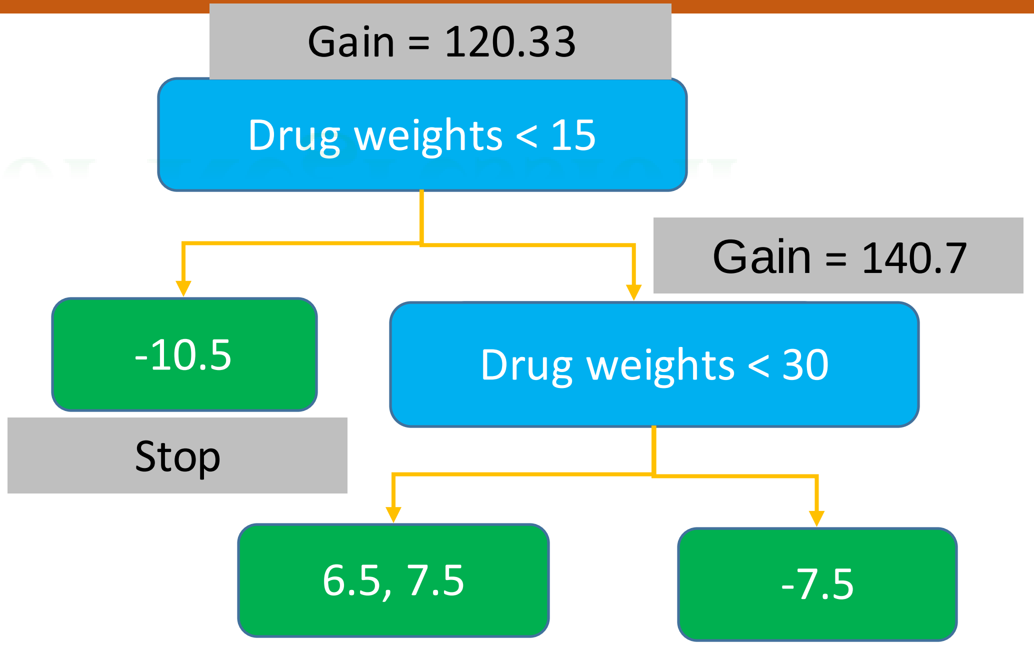
Gain = Left SC + Right SC - Root SC

Gain = 140.17

XGBoost For Regression

❑*Step 1*



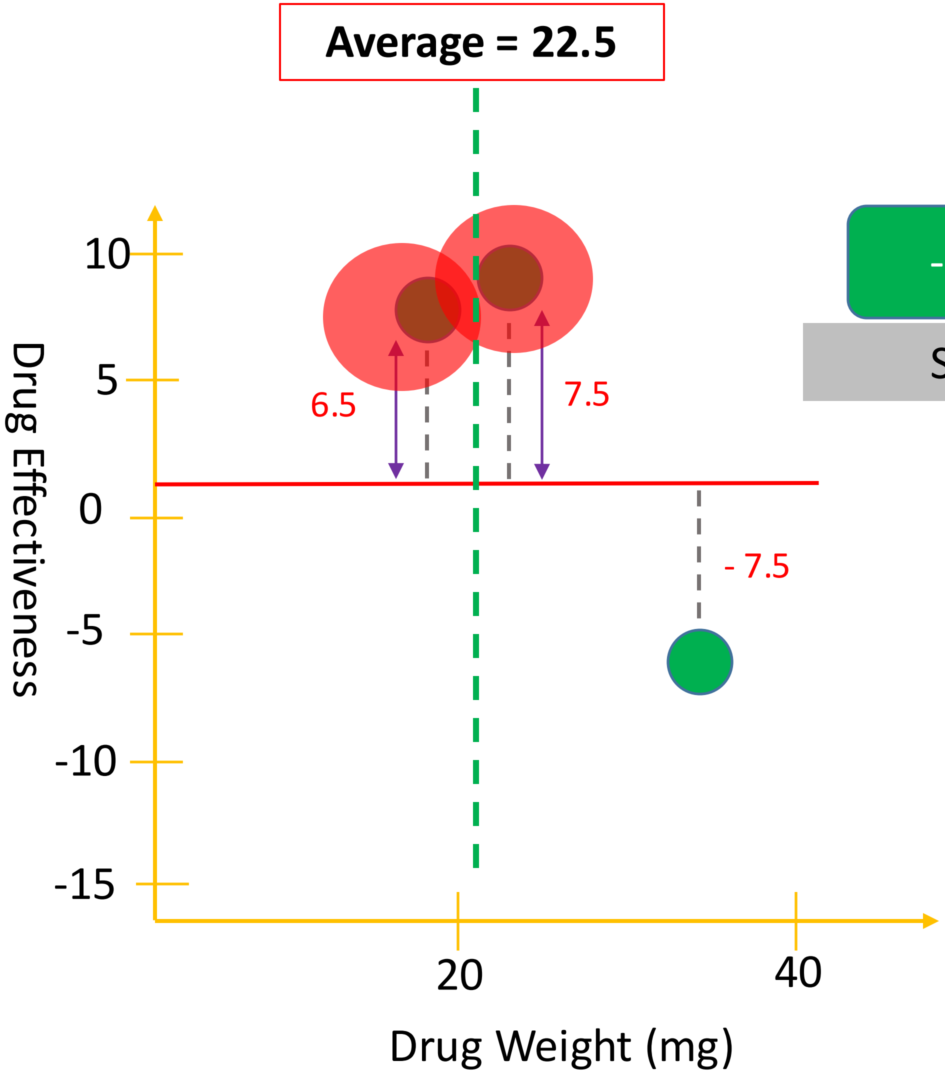


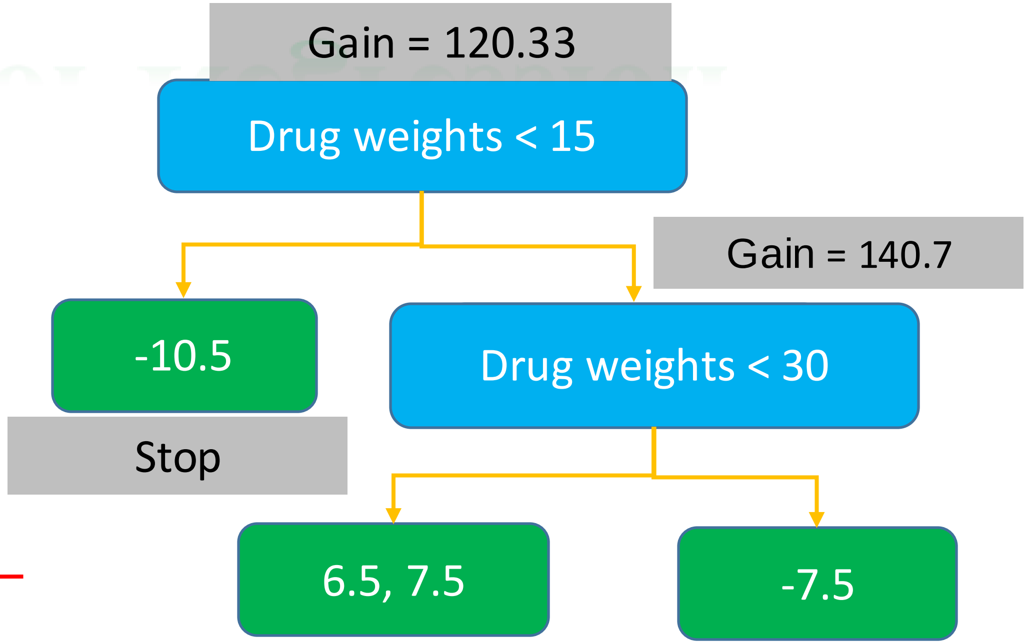
How to prune the tree to prevent Overfitting ? Gain information

$\gamma = 130$

XGBoost For Regression

- Step 1





How to prune the tree to prevent Overfitting ? Gain information

$$\gamma = 130$$

Difference = Gain - γ

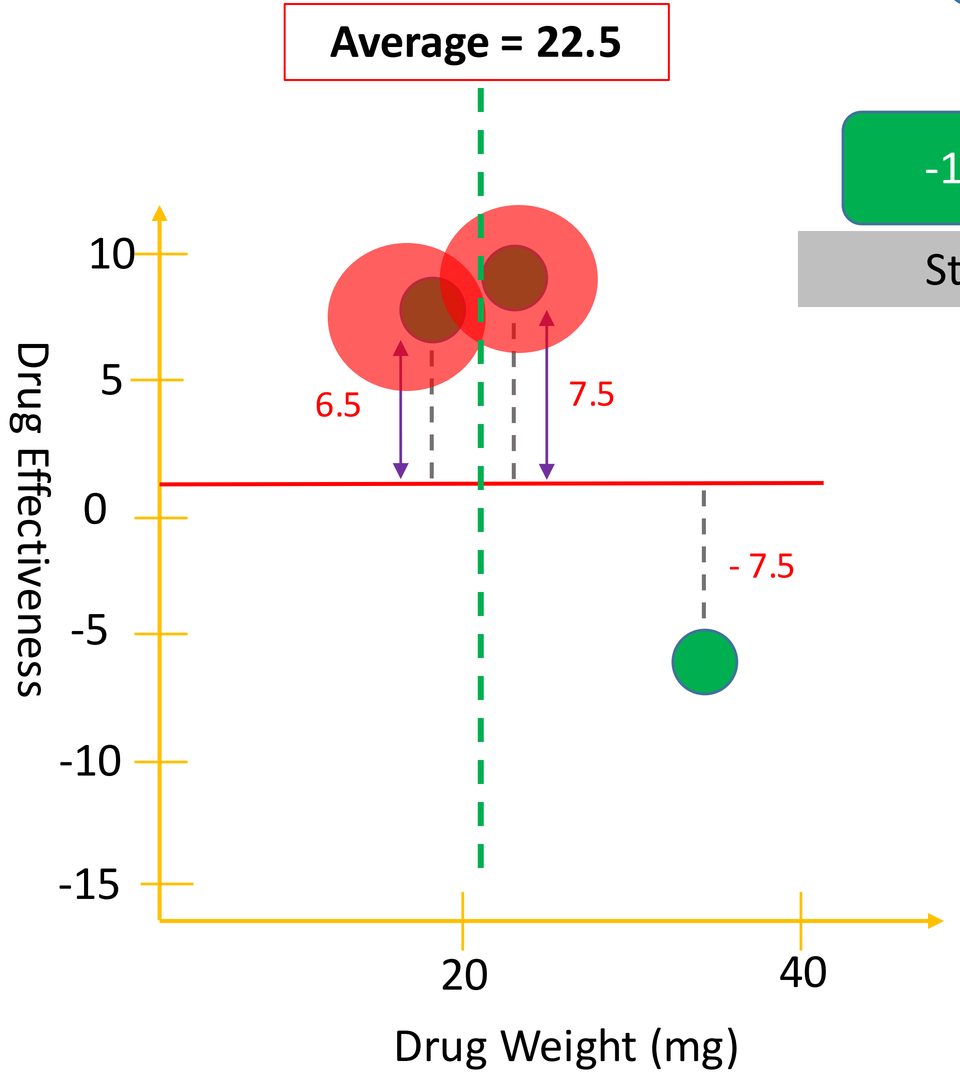
If difference > 0, do not remove branch

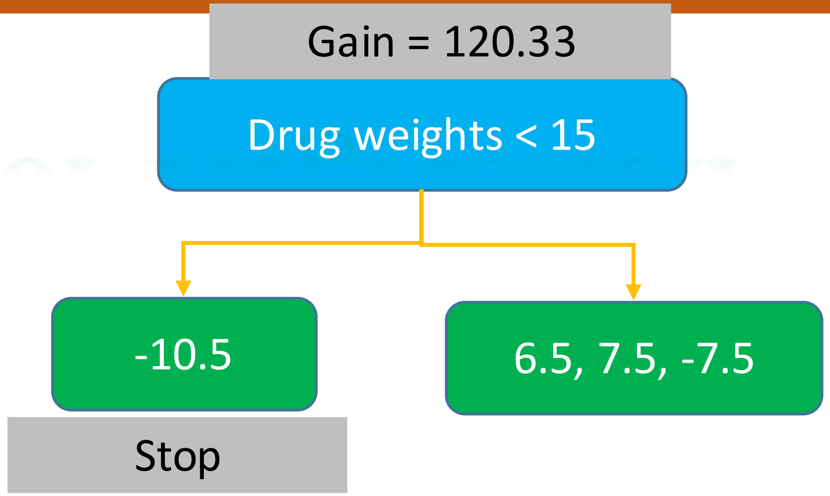
If difference < 0, remove branch

AI VIET NAM
@aivietnam.edu.vn

XGBoost For Regression

❑ *Step 1*





How to prune the tree to prevent Overfitting ? Gain information

$\gamma = 150$

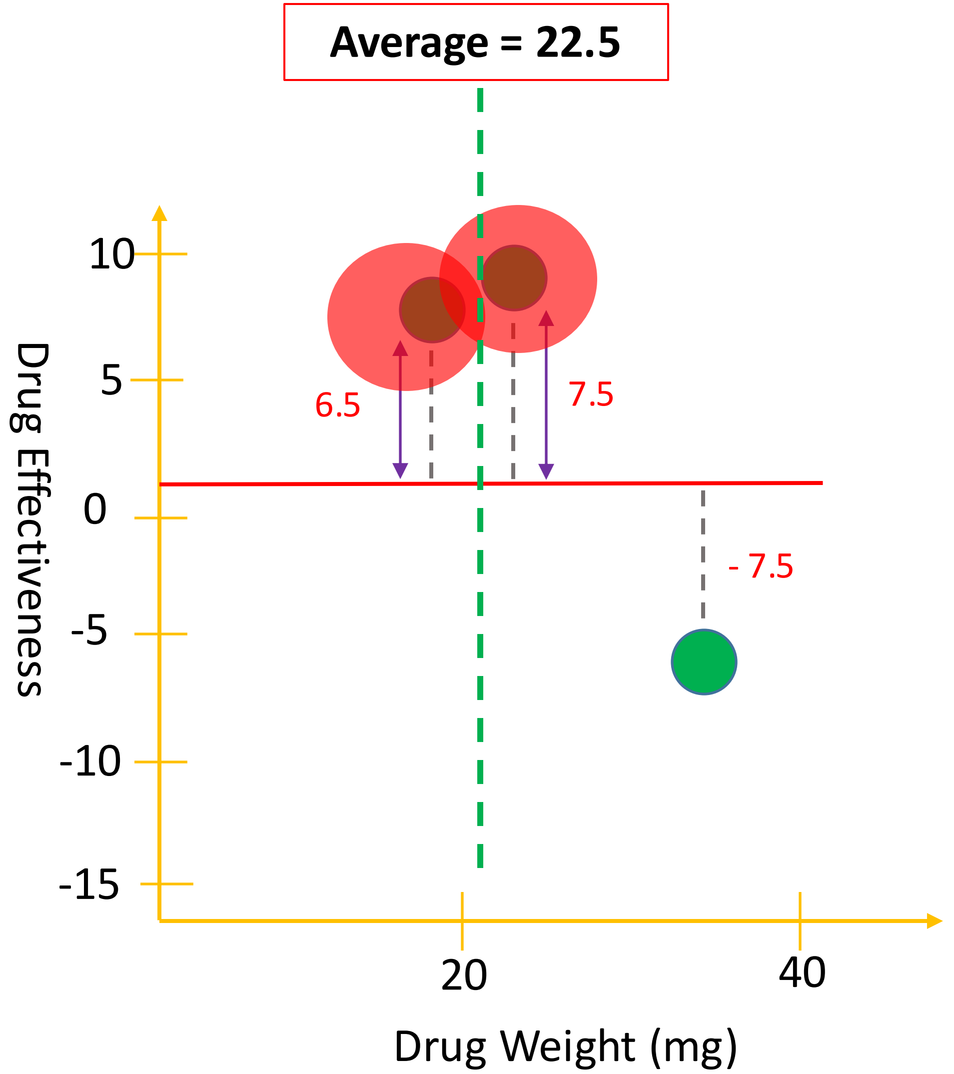
Difference = Gain - γ

If difference > 0, do not remove branch

If difference < 0, remove branch

XGBoost For Regression

- Step 1



o.5

How to prune the tree to prevent Overfitting ? Gain information

$\gamma = 150$

Difference = Gain - γ

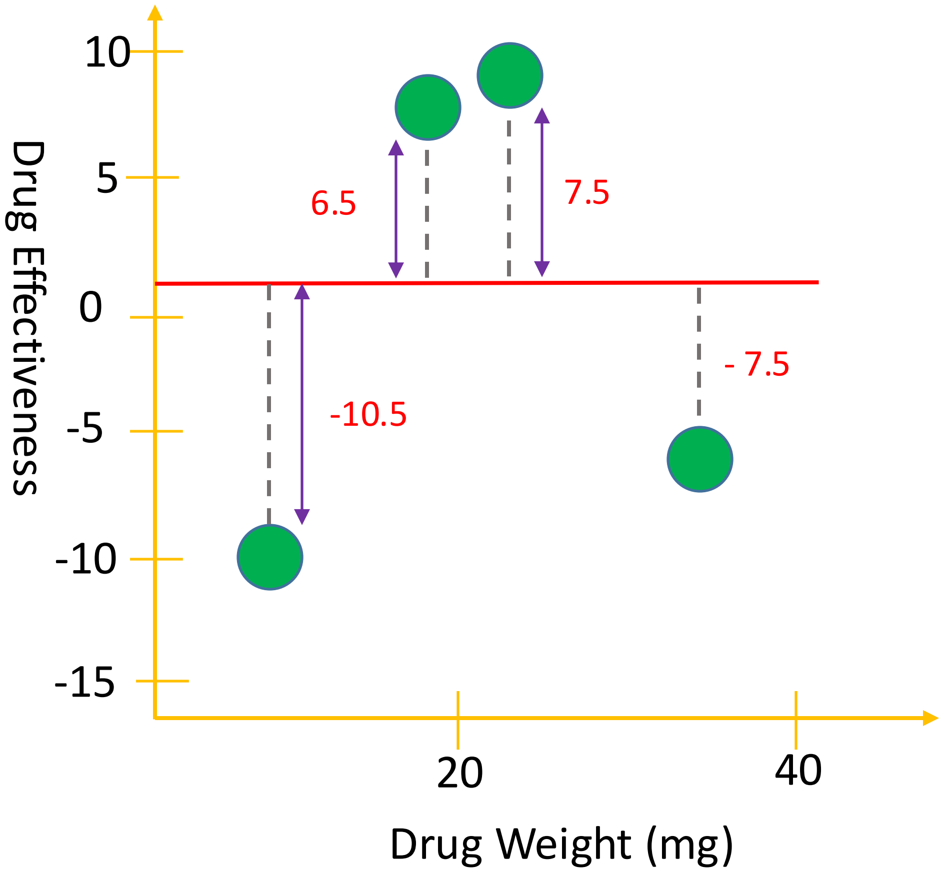
If difference > 0, do not remove branch

If difference < 0, remove branch

AI VIET NAM
@aivietnam.edu.vn

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

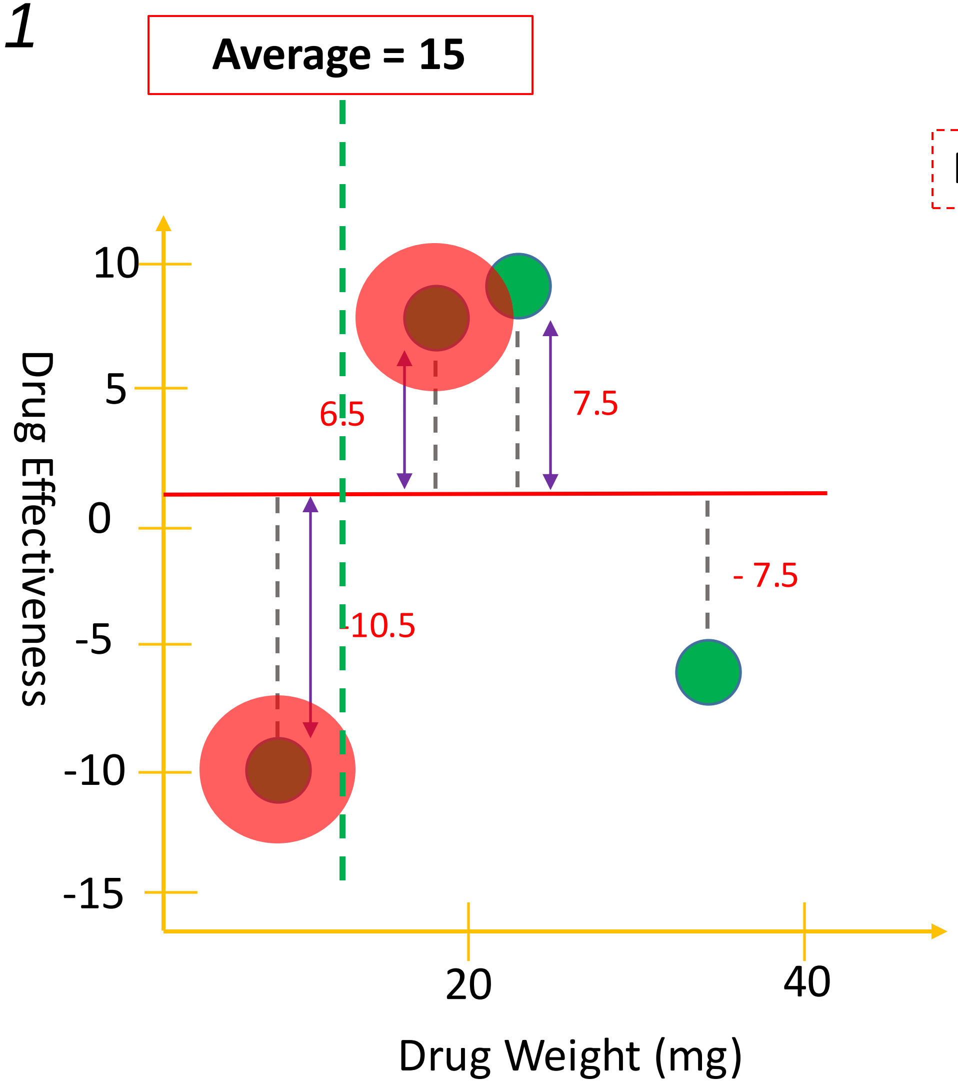
$$SC = \frac{\sum\ (output - predicted)^2}{m + \lambda}$$

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

AI VIET NAM
@aivietnam.edu.vn

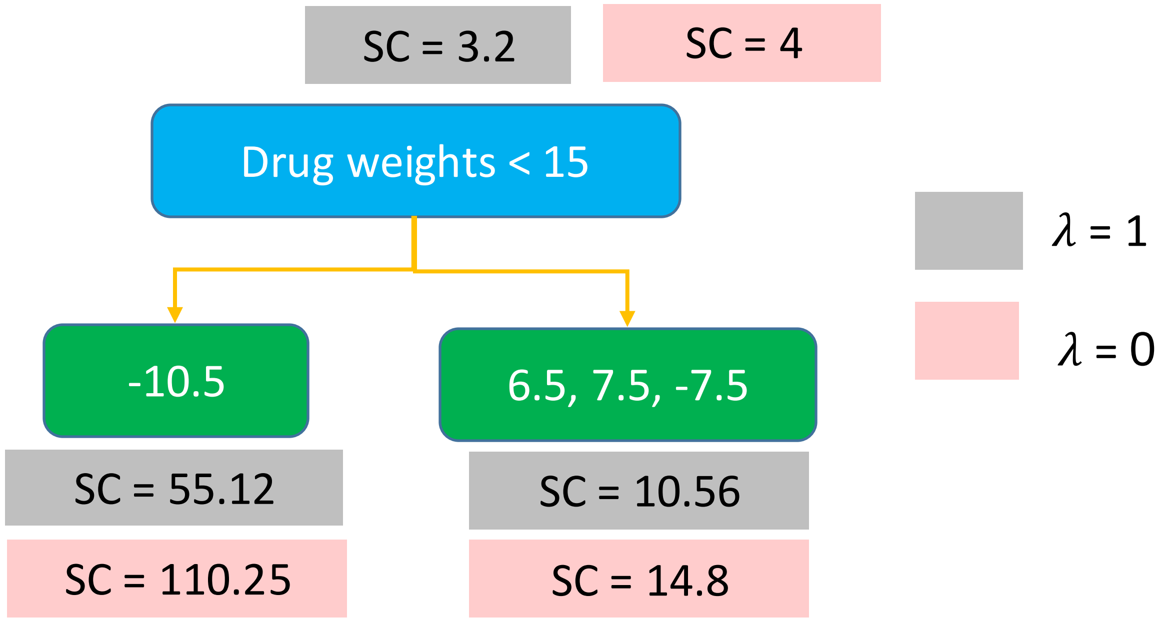
XGBoost For Regression

❑ *Step 1*



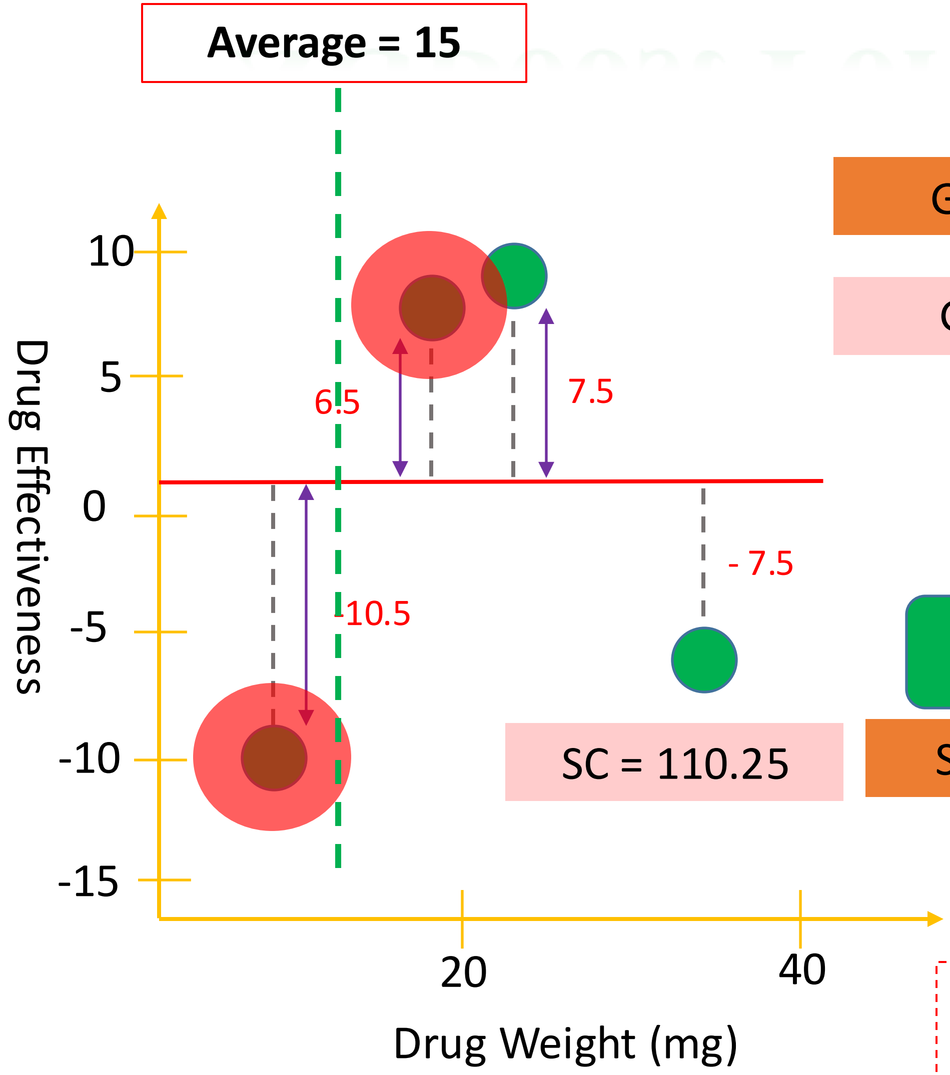
SC = 3.2 | -10.5, 6.5, 7.5, -7.5

Please look at the two outputs with lowest drug weights



When λ > 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

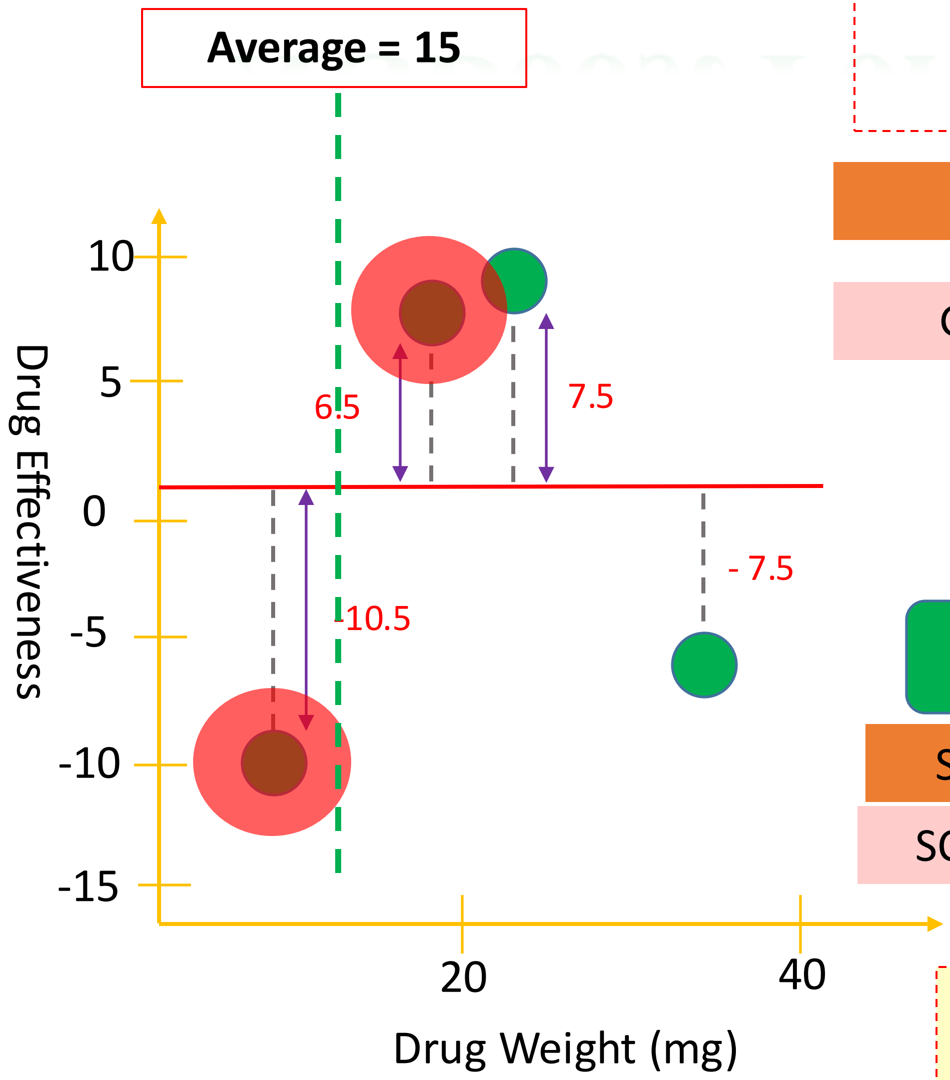
Drug weights < 15

Gain = 82.9 Gain = 140.17

-10.5	6.5, 7.5, -7.5
SC = 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



$\lambda > 0$: 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: $\gamma = 130$

Drug weights < 15

Gain = 82.9

Gain = 140.17

-10.5

SC = 55.12

SC = 110.25

6.5, 7.5, -7.5

SC = 10.56

SC = 14.8

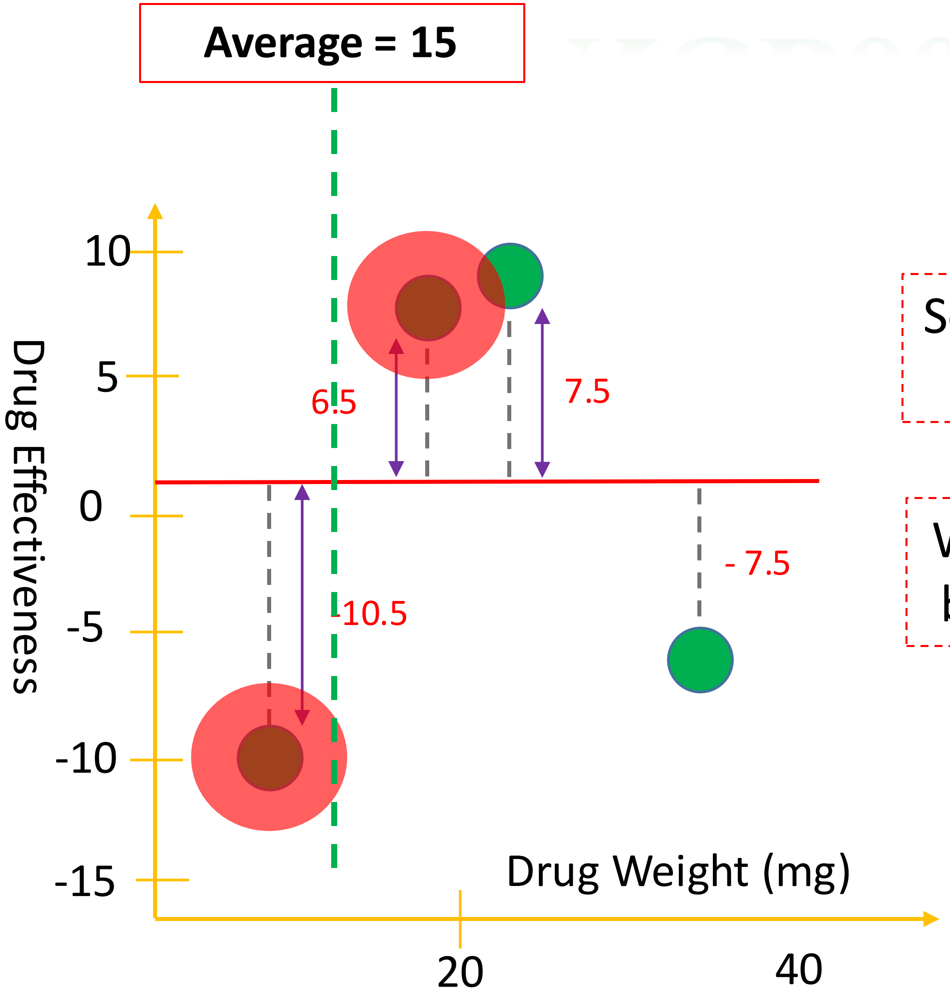
$\lambda = 0$

$\lambda = 1$

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

AI VIET NAM
@aivietnam.edu.vn

XGBoost For Regression



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

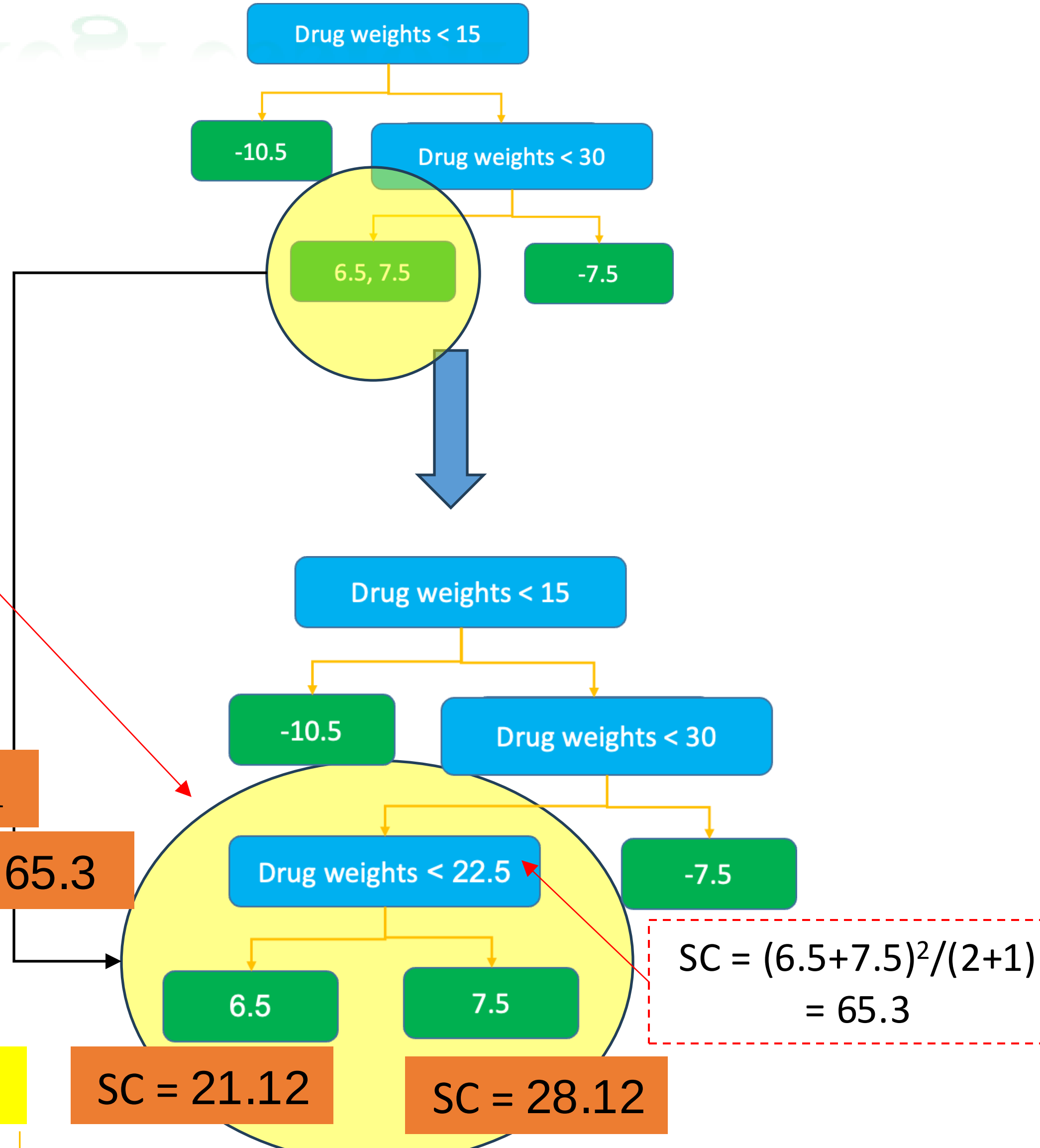
We will remove this branch: -16.06- γ < 0

Prunning parameter: $\gamma = 0$

$\lambda = 1$

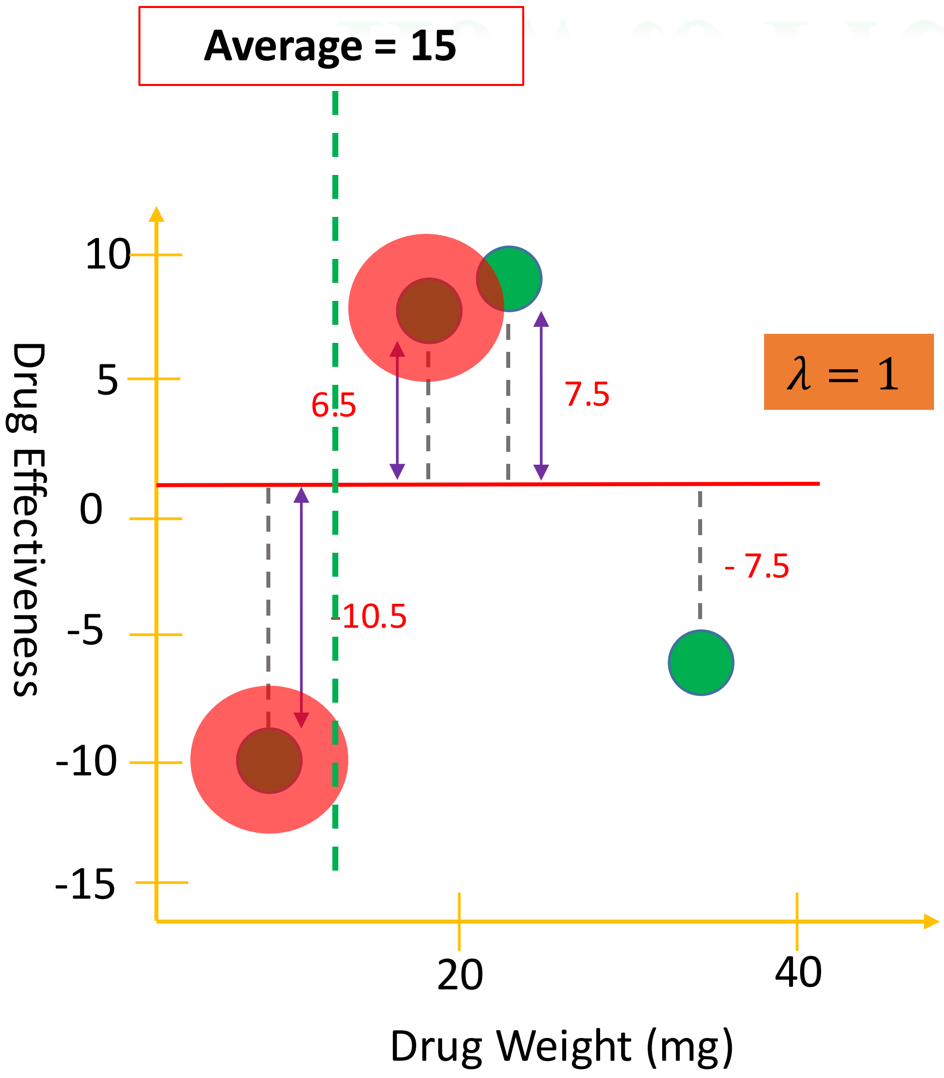
SC = 65.3

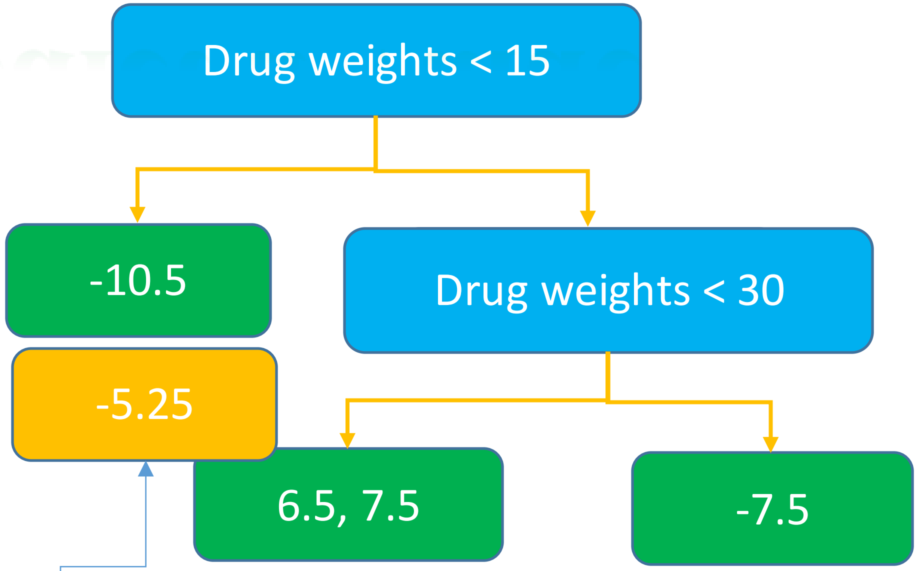
Gain = 21.12 + 28.12 – 65.3 = -16.06



AI VIET NAM
@aivietnam.edu.vn

How to Predict a Value



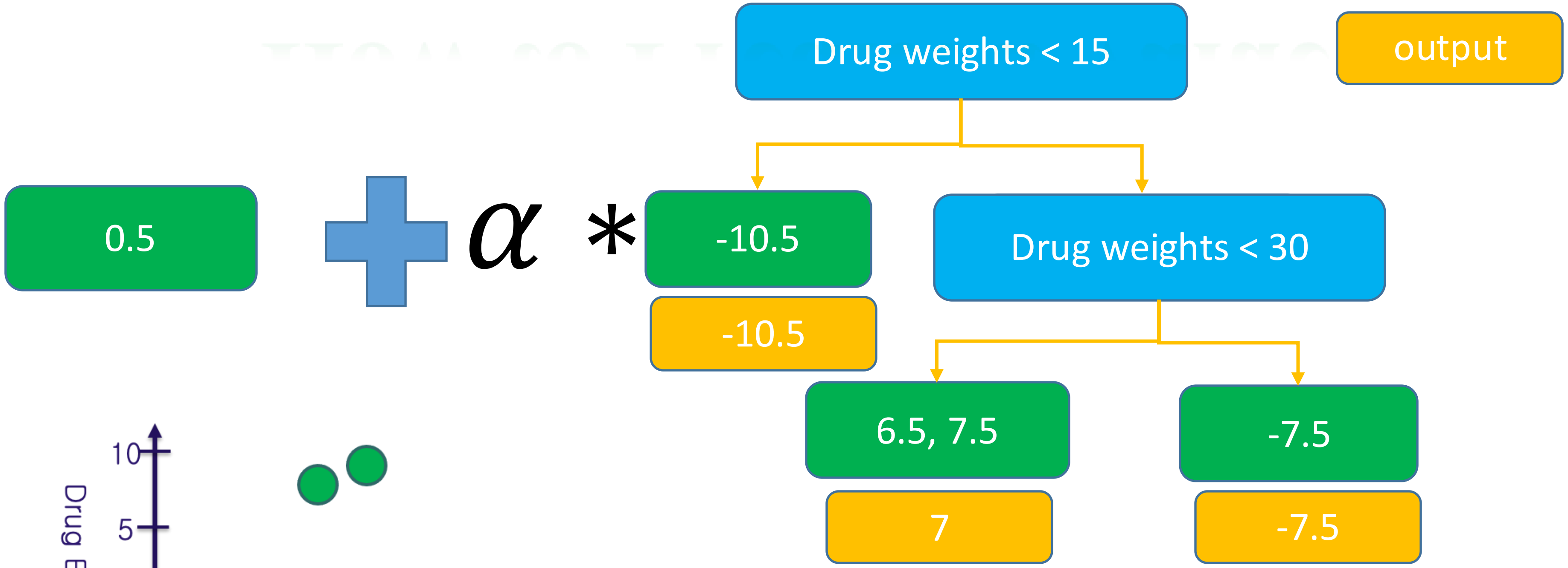


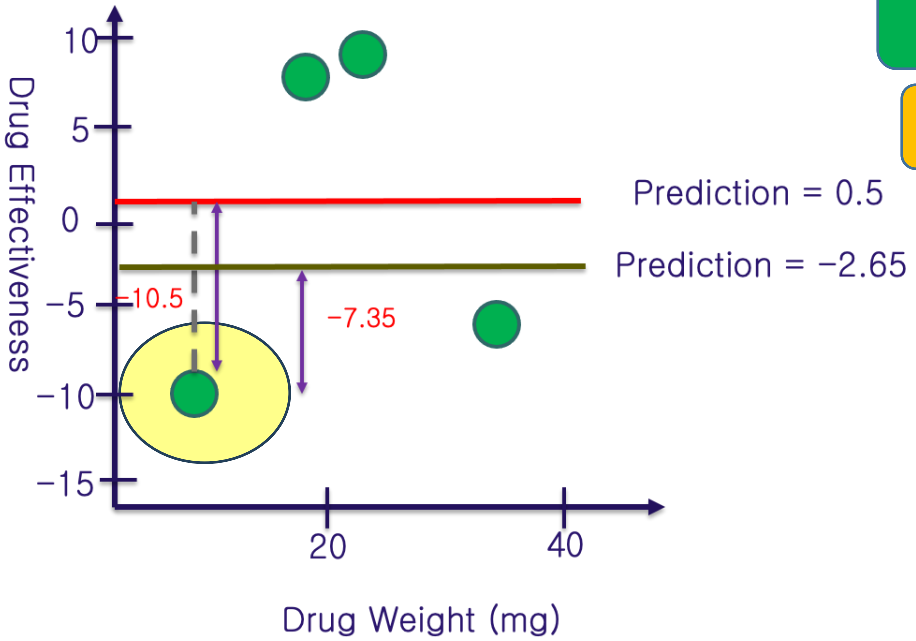
$\lambda = 1$

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

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

How to Predict a Value

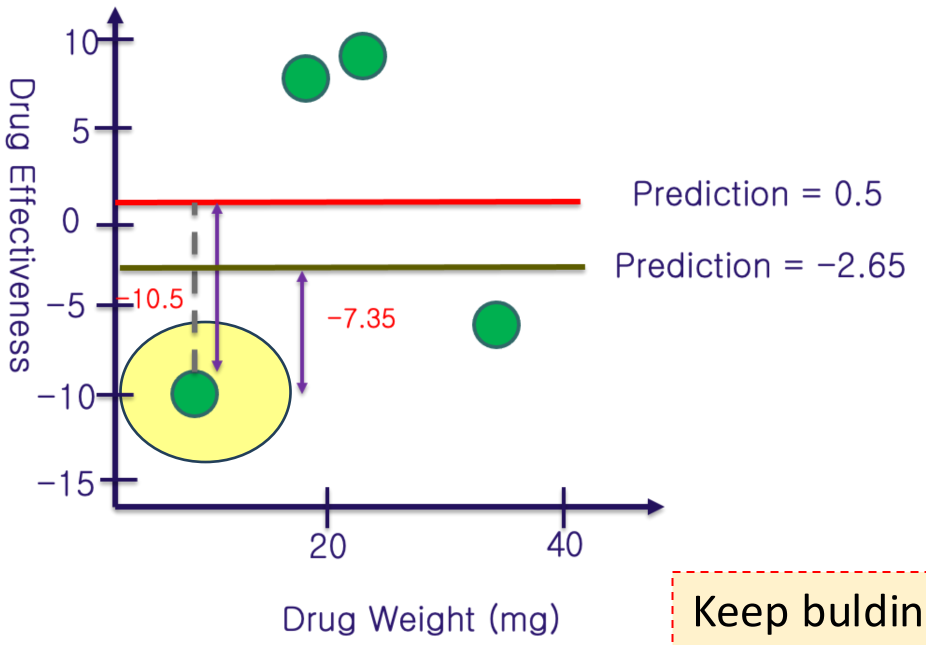


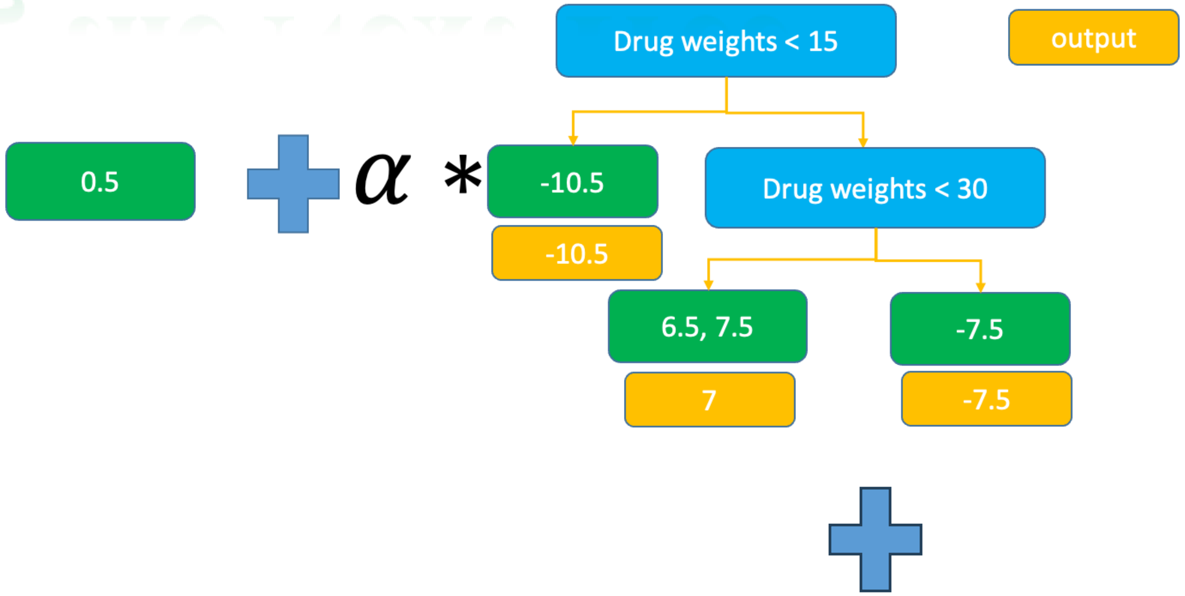


$\lambda = 0$ $\alpha = 0.3$

AI VIET NAM
@aivietnam.edu.vn

Building the Next Tree

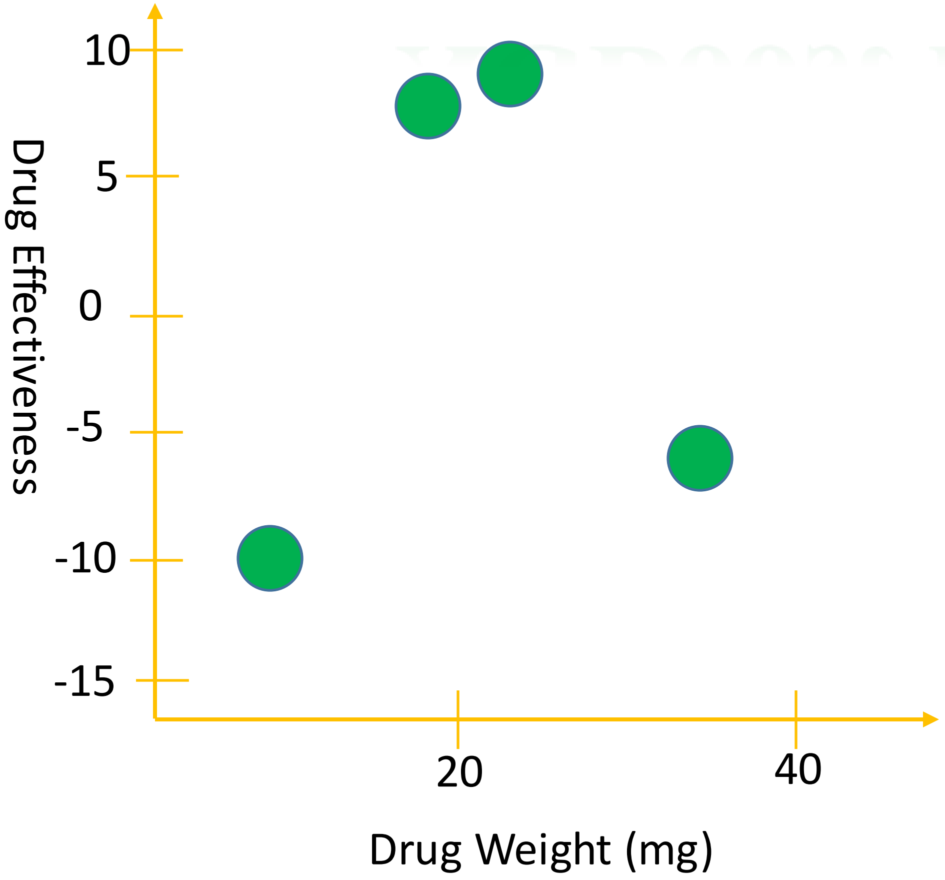


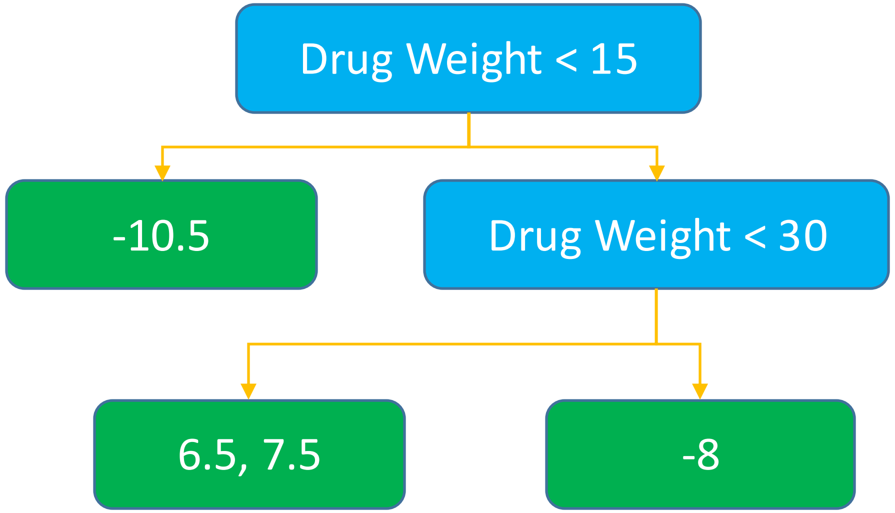


$\alpha *$ Next Tree Result

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

XGBoost for Regression



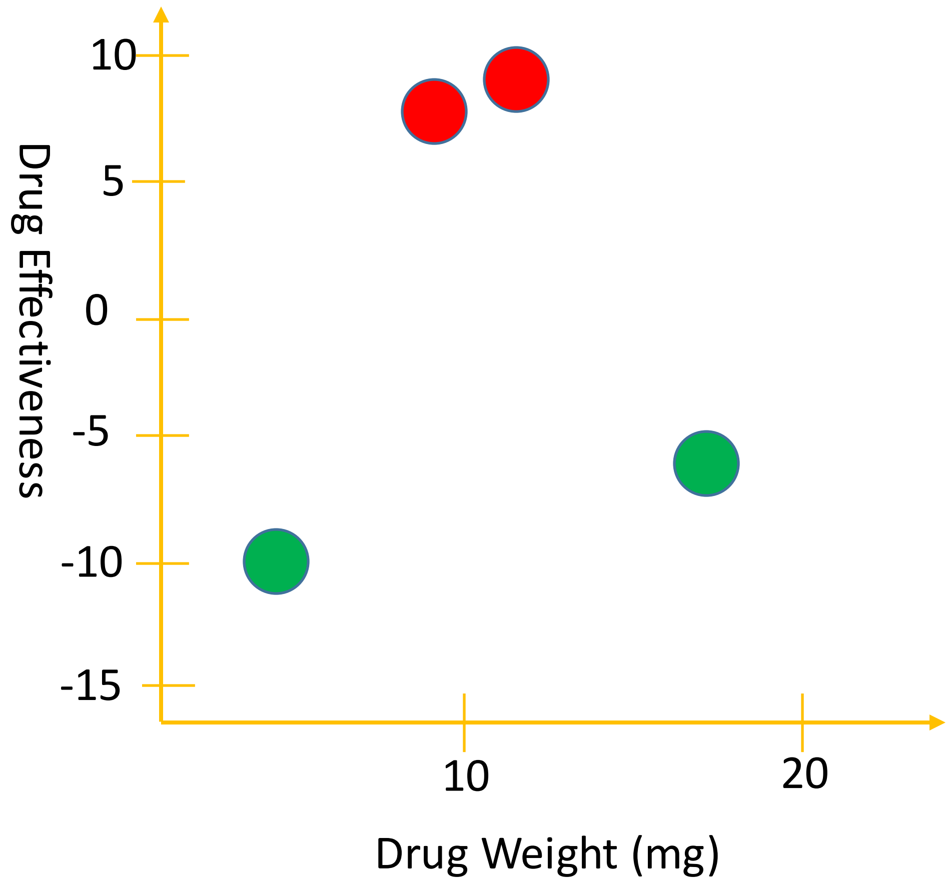


HOW TO FIND QUANTILES? => QUANTILE SKETCH APPROXIMATE SOLUTION

Outline

- Regularization
- Regression XGBoost
- Classification XGBoost
- XGBoost: Clearly Explain
- Time Series Example
- Summary

XGBoost For Classification



Not effectiviness

Effectiviness

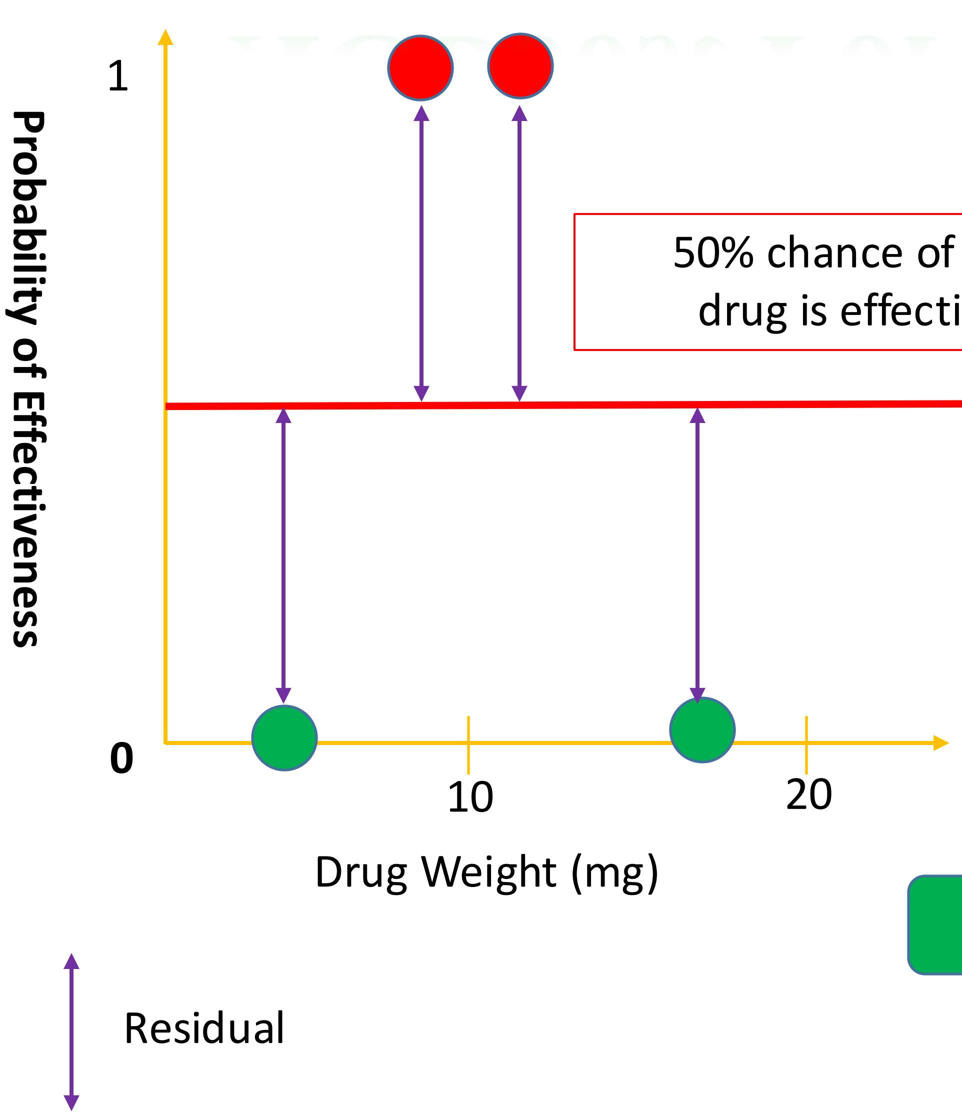
First prediction

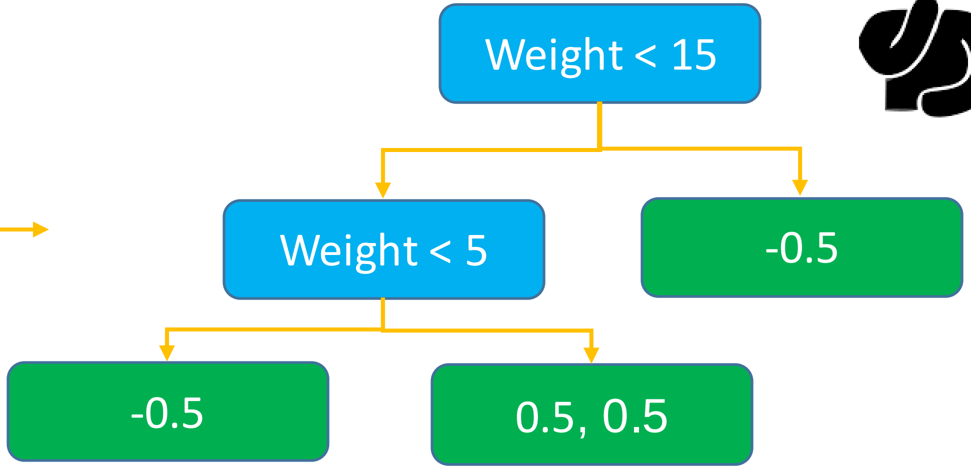
0.5

50% chance of the drug is effective

AI VIET NAM
@aivietnam.edu.vn

XGBoost For Classification





AI VIET NAM
@aivietnam.edu.vn

XGBoost For Classification

Similarity Score for Classification:

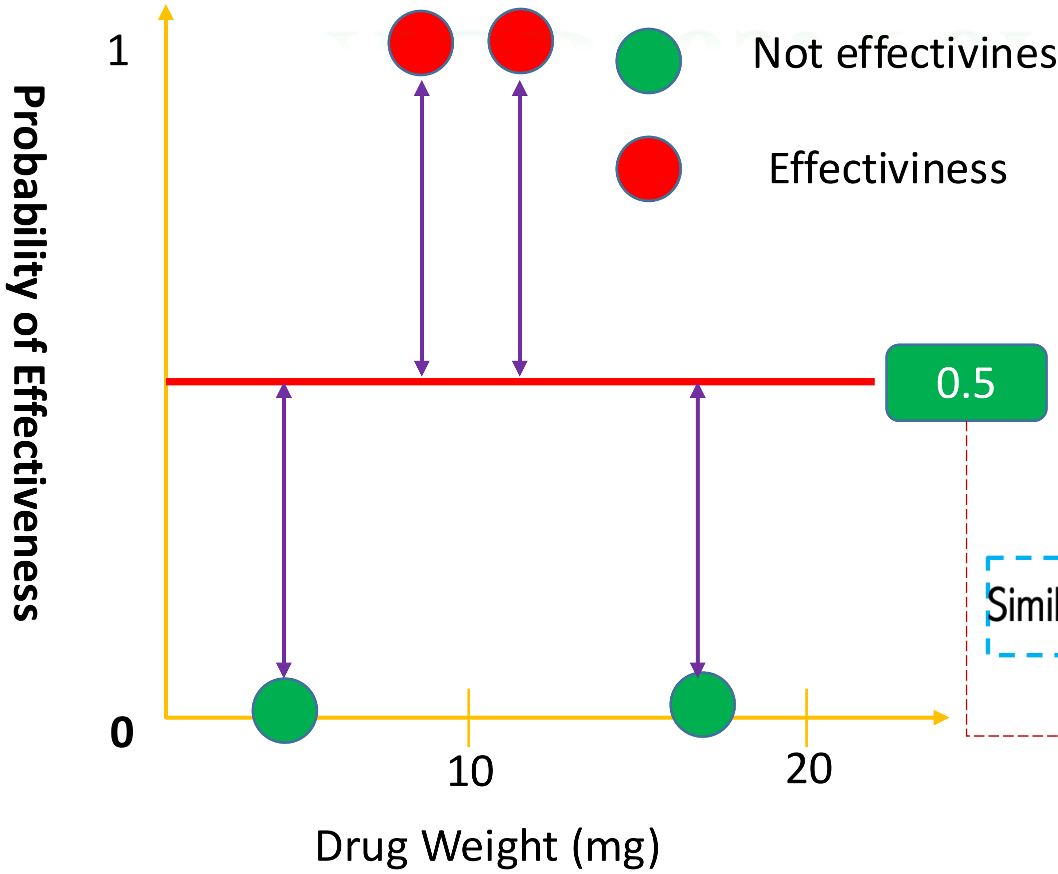
$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum[\text{previous probability}_i \times (1 - \text{previous probability}_i)] + \lambda}$$

Similarity Score for Prediction (regression):

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

AI VIET NAM
@aivietnam.edu.vn

XGBoost For Classification



-0.5, 0.5, 0.5, -0.5

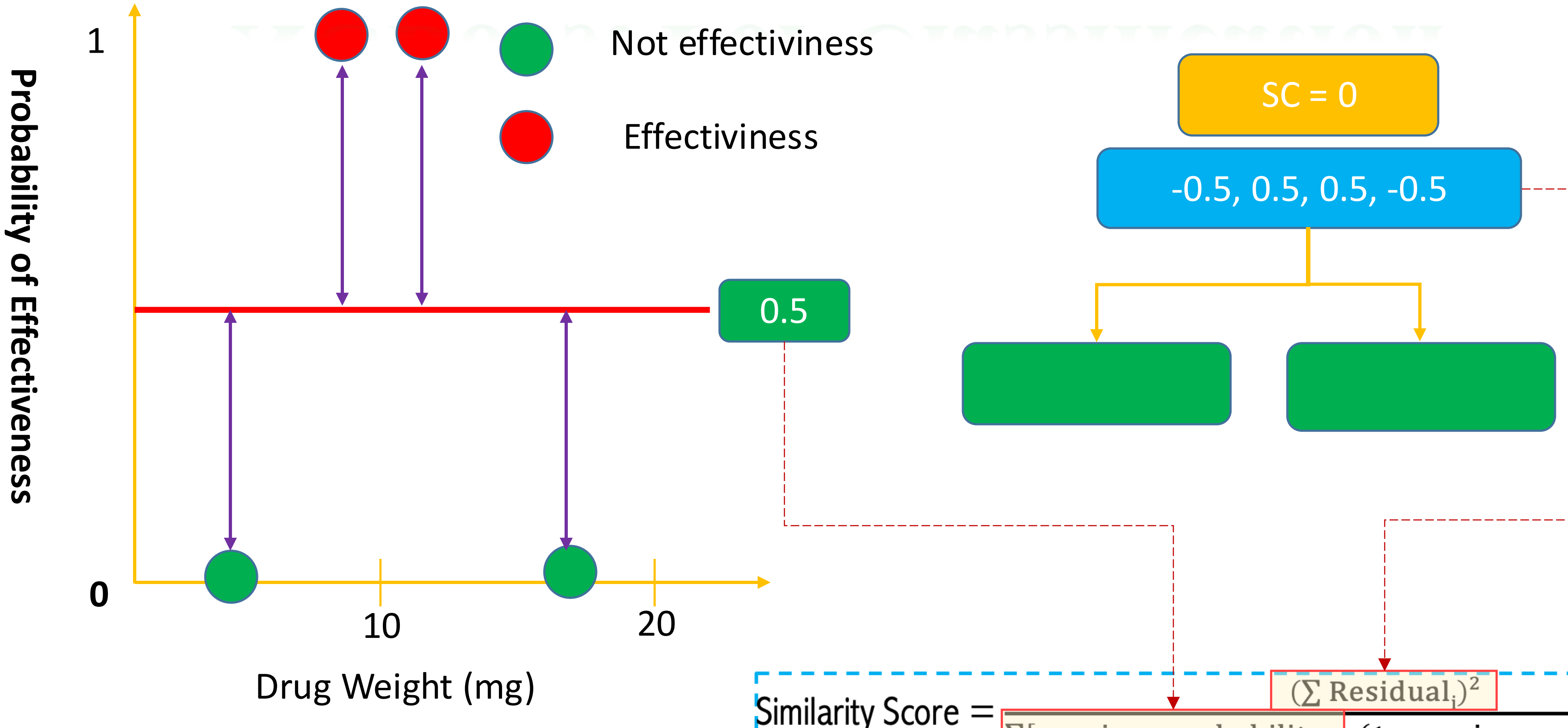
First Tree

SC = 0

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

$\lambda = 0$

XGBoost For Classification

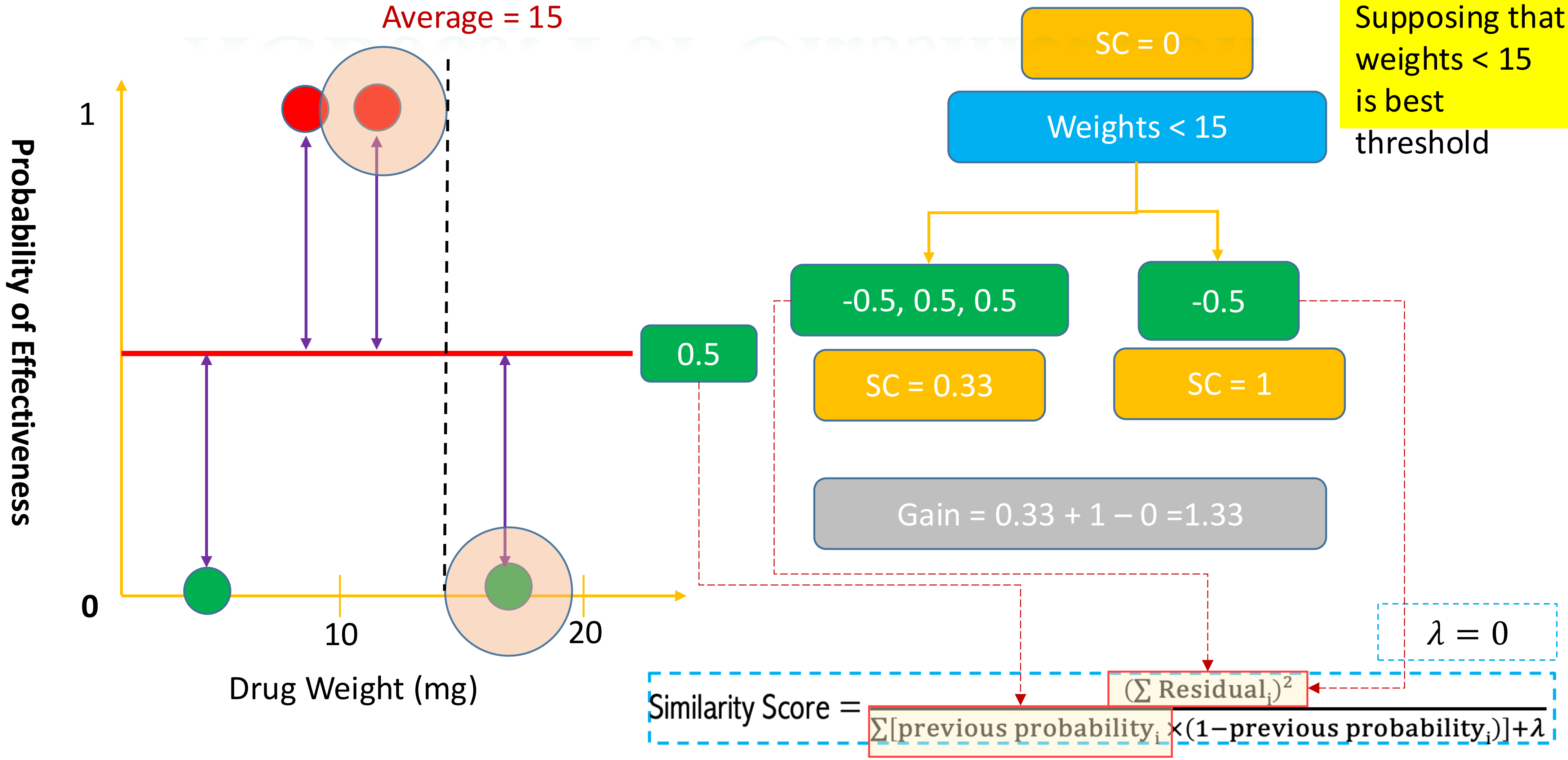


$\lambda = 0$

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

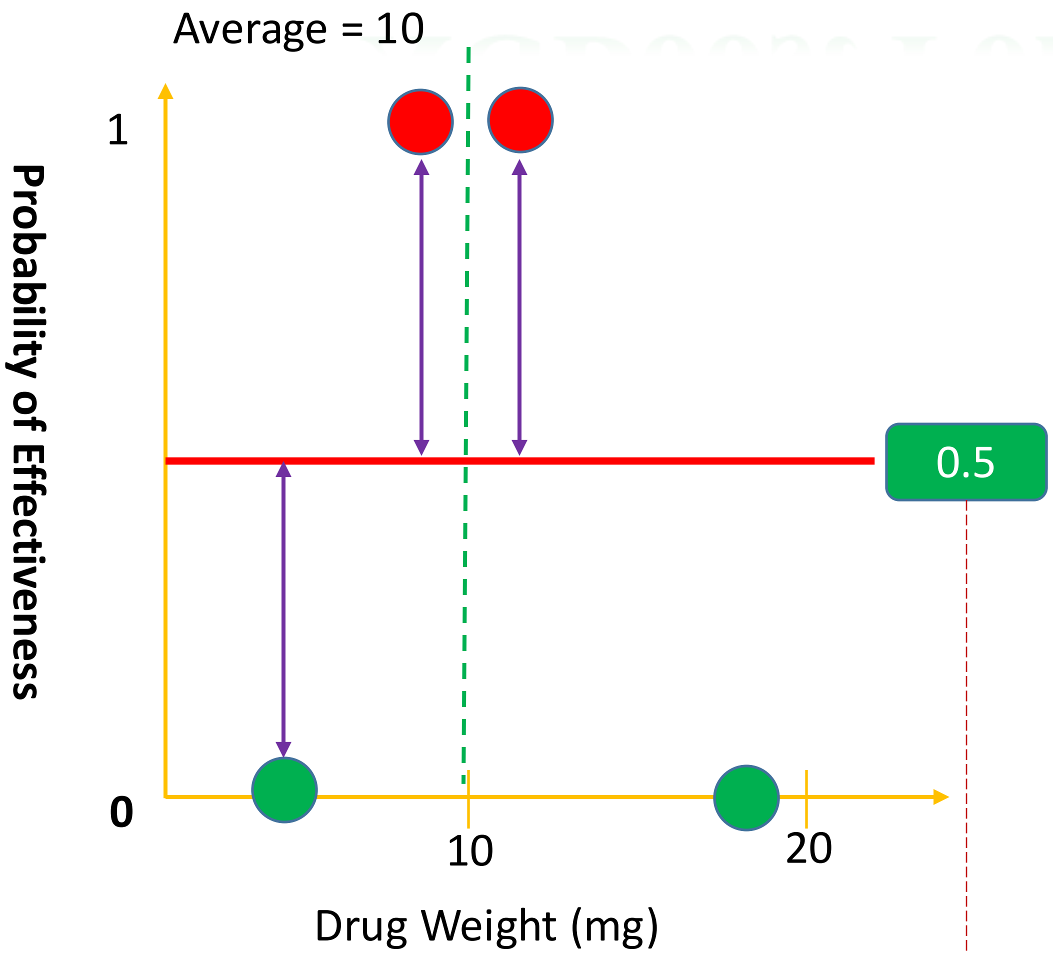
AI VIET NAM
@aivietnam.edu.vn

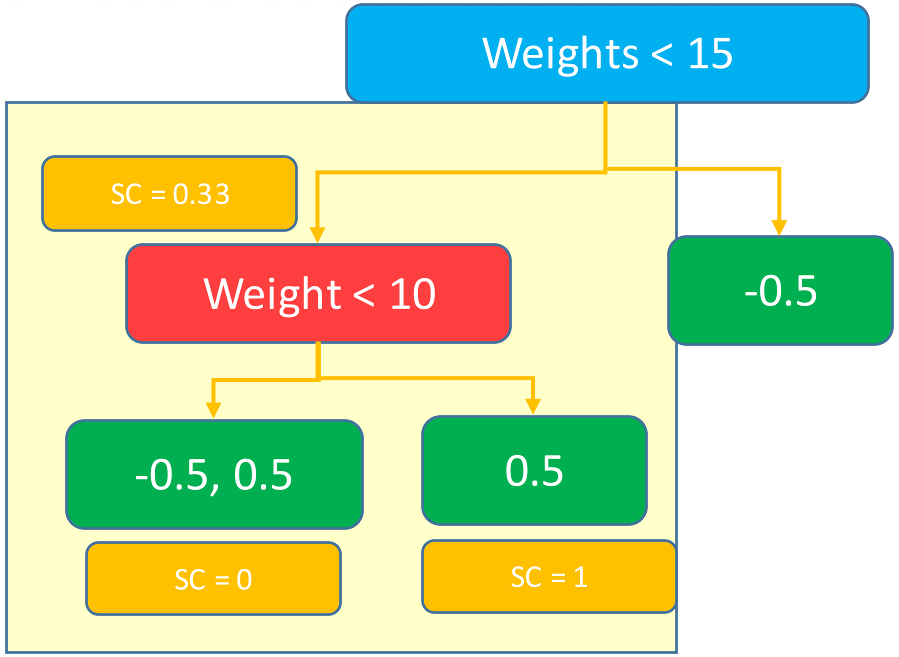
XGBoost For Classification



Supposing that weights < 15 is best threshold

XGBoost For Classification





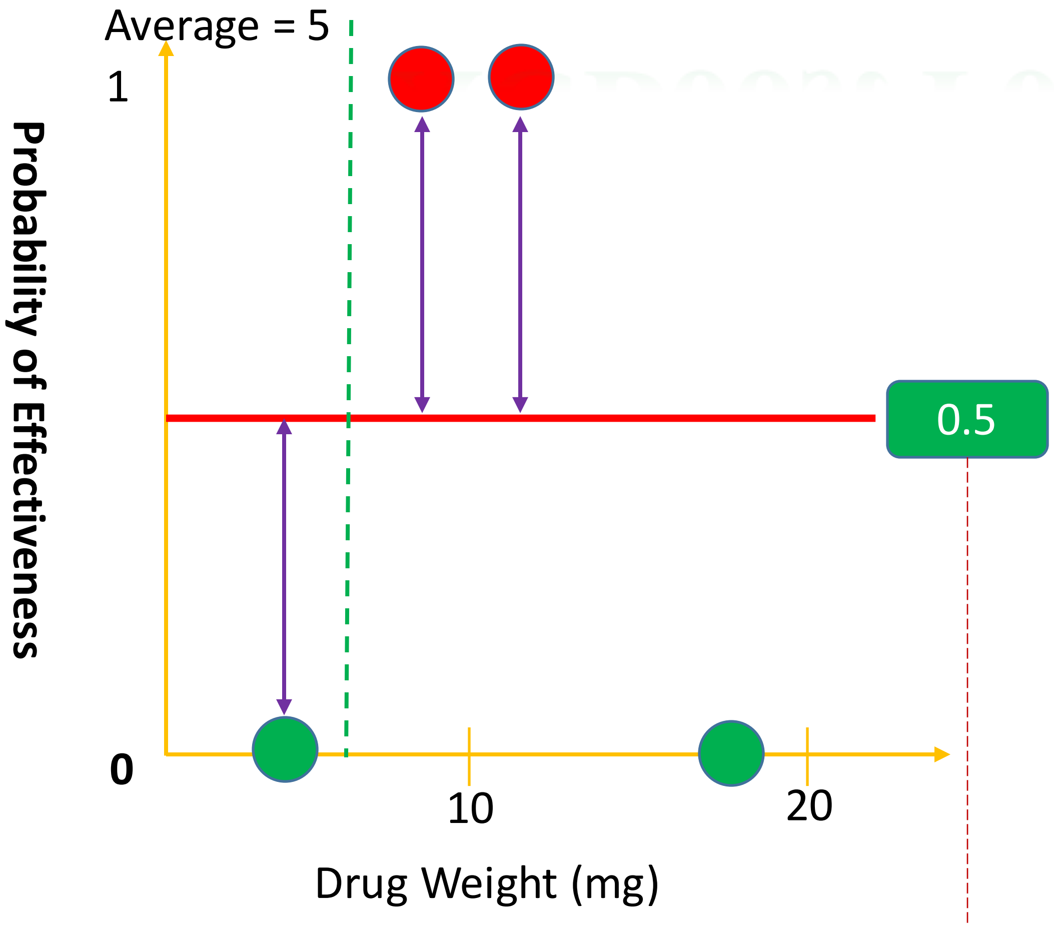
Gain = 0 + 1 – 0.33 = 0.66

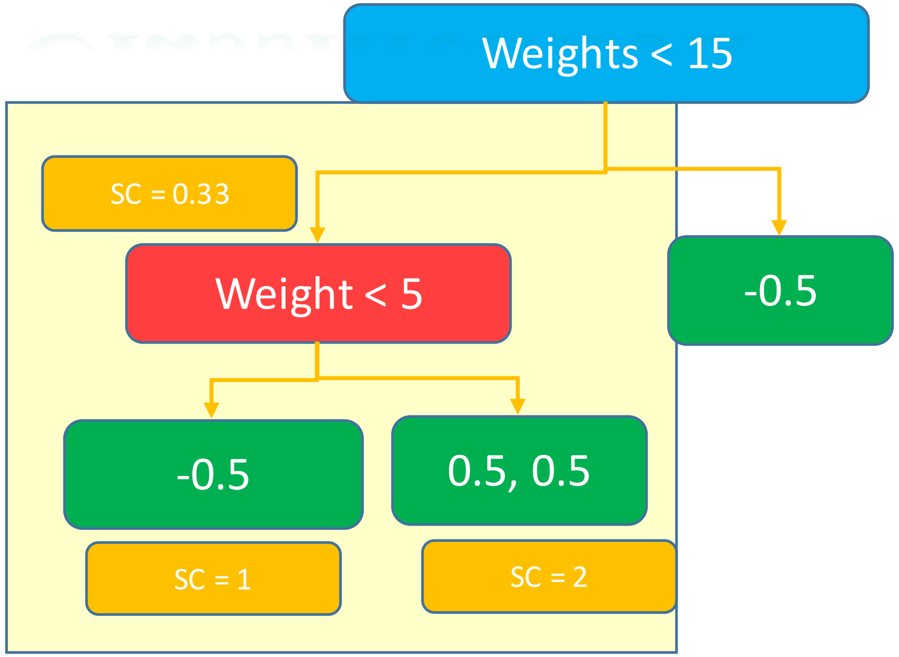
$\lambda = 0$

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

AI VIET NAM
@aivietnam.edu.vn

XGBoost For Classification

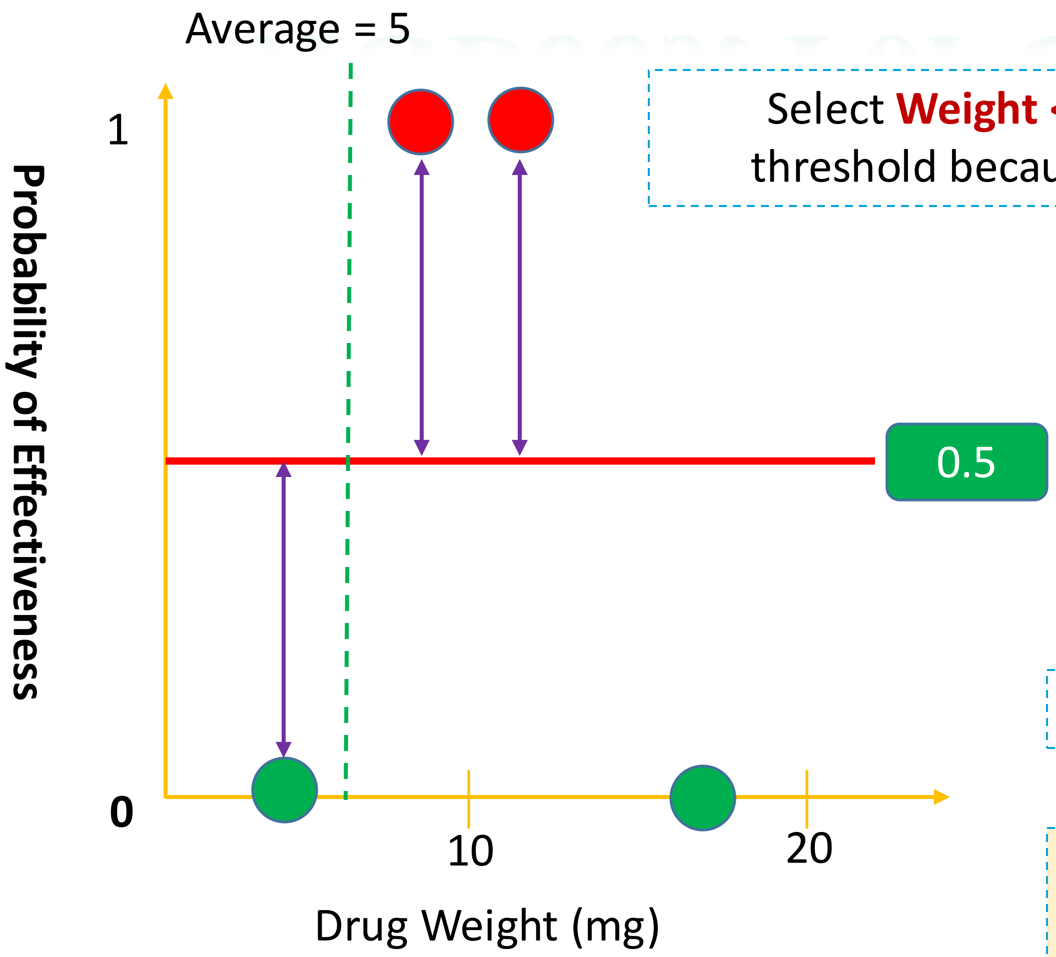




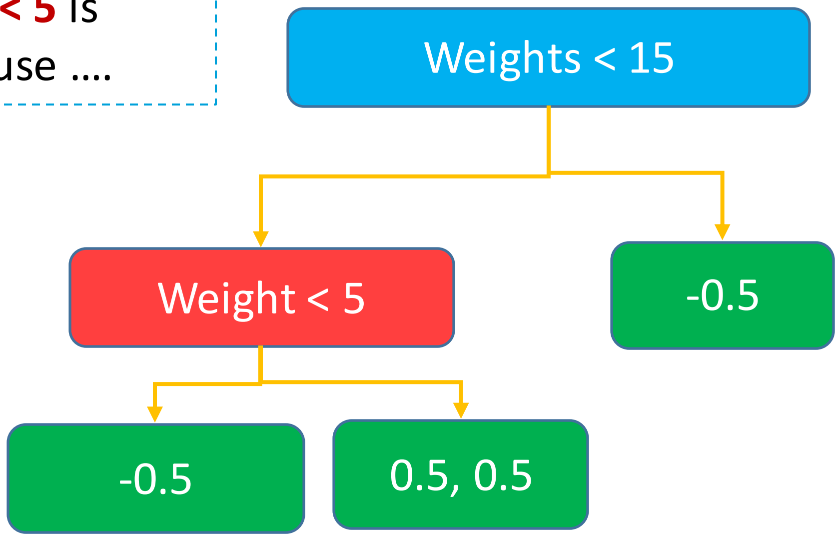
Gain = 1 + 2 – 0.33 = 2.66

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

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

AI VIET NAM
@aivietnam.edu.vn

What is a Cover

Similarity Score for Classification:

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

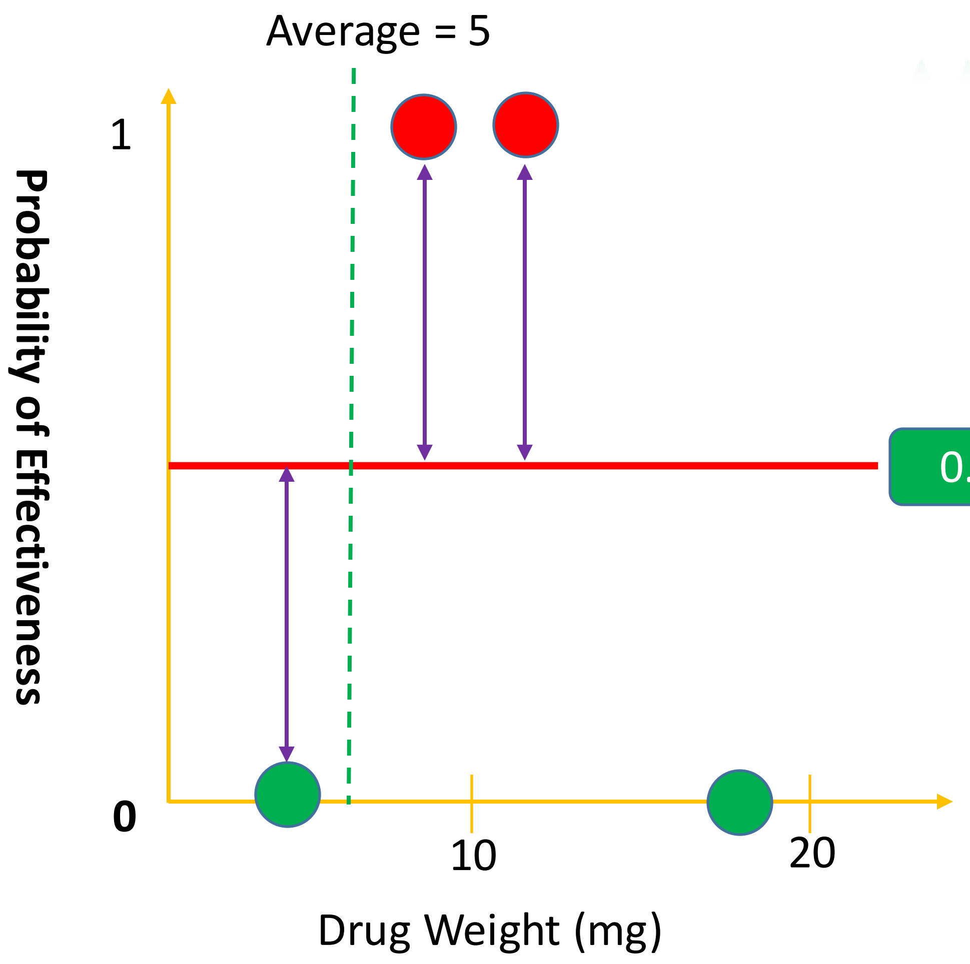
Cover

Similarity Score for Prediction:

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

AI VIET NAM
@aivietnam.edu.vn

What is a Cover



Weights < 15

Weight < 5

-0.5

-0.5

0.5, 0.5

Delete

Delete

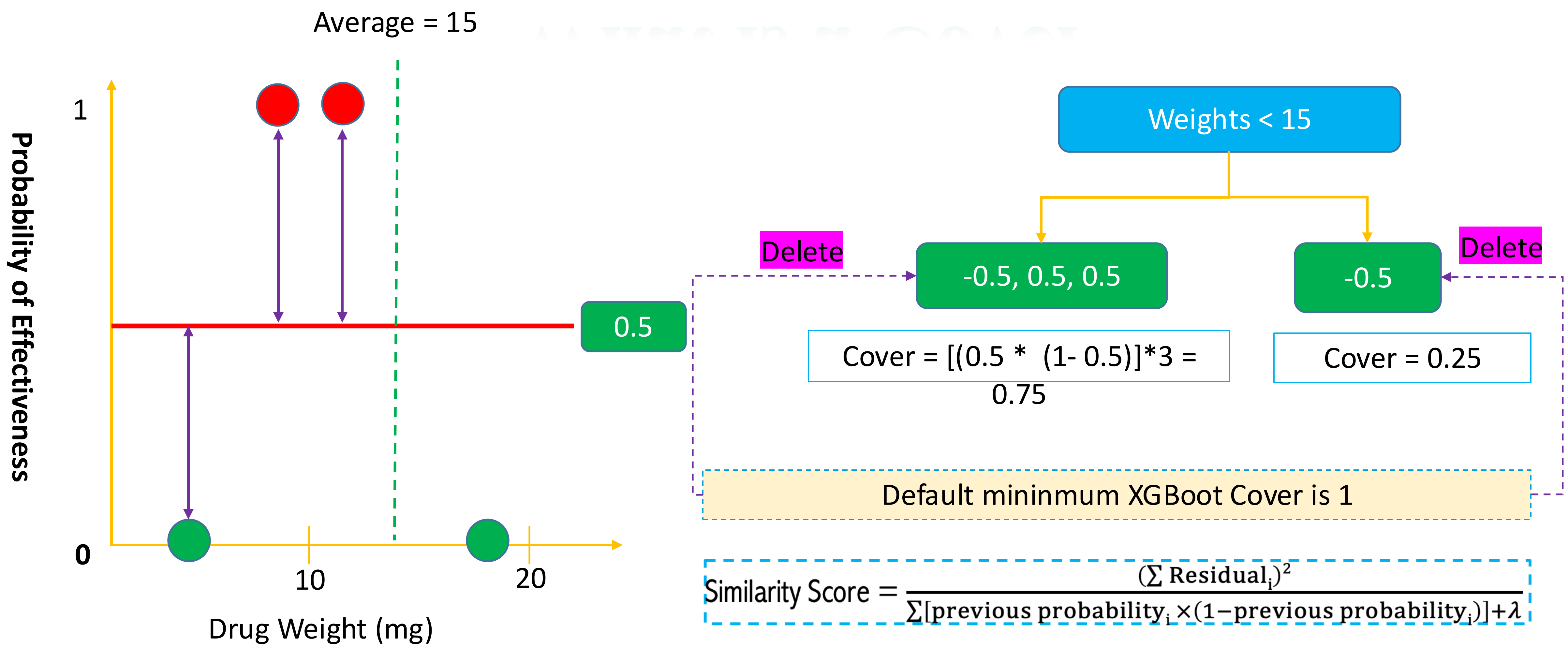
Cover = 0.5 * (1- 0.5) = 0.25

Cover = (0.5 * (1- 0.5))*2 = 0.5

Mininmum XGBoot Cover is 1

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

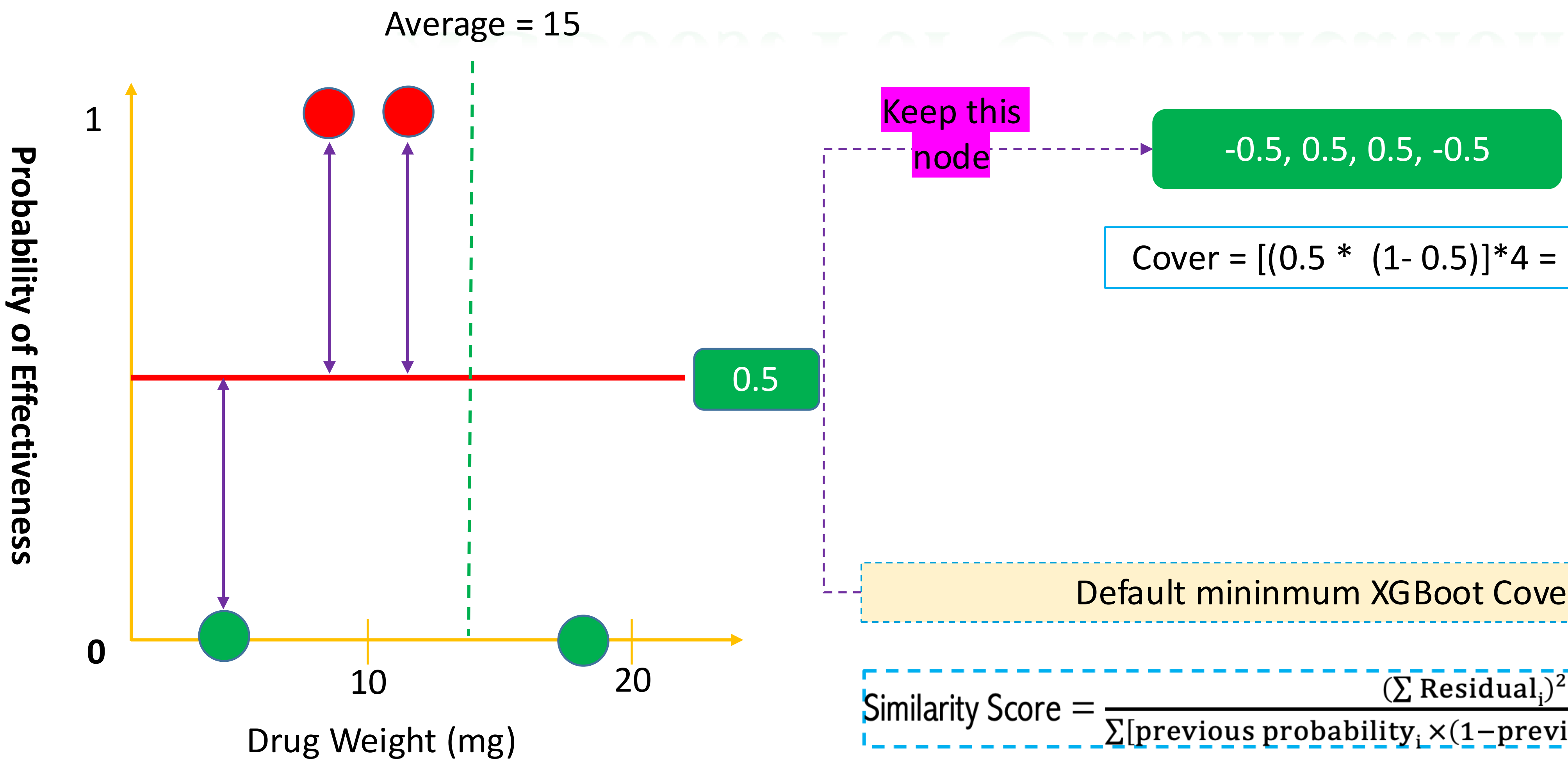
What is a Cover



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

AI VIET NAM
@aivietnam.edu.vn

XGBoost For Classification



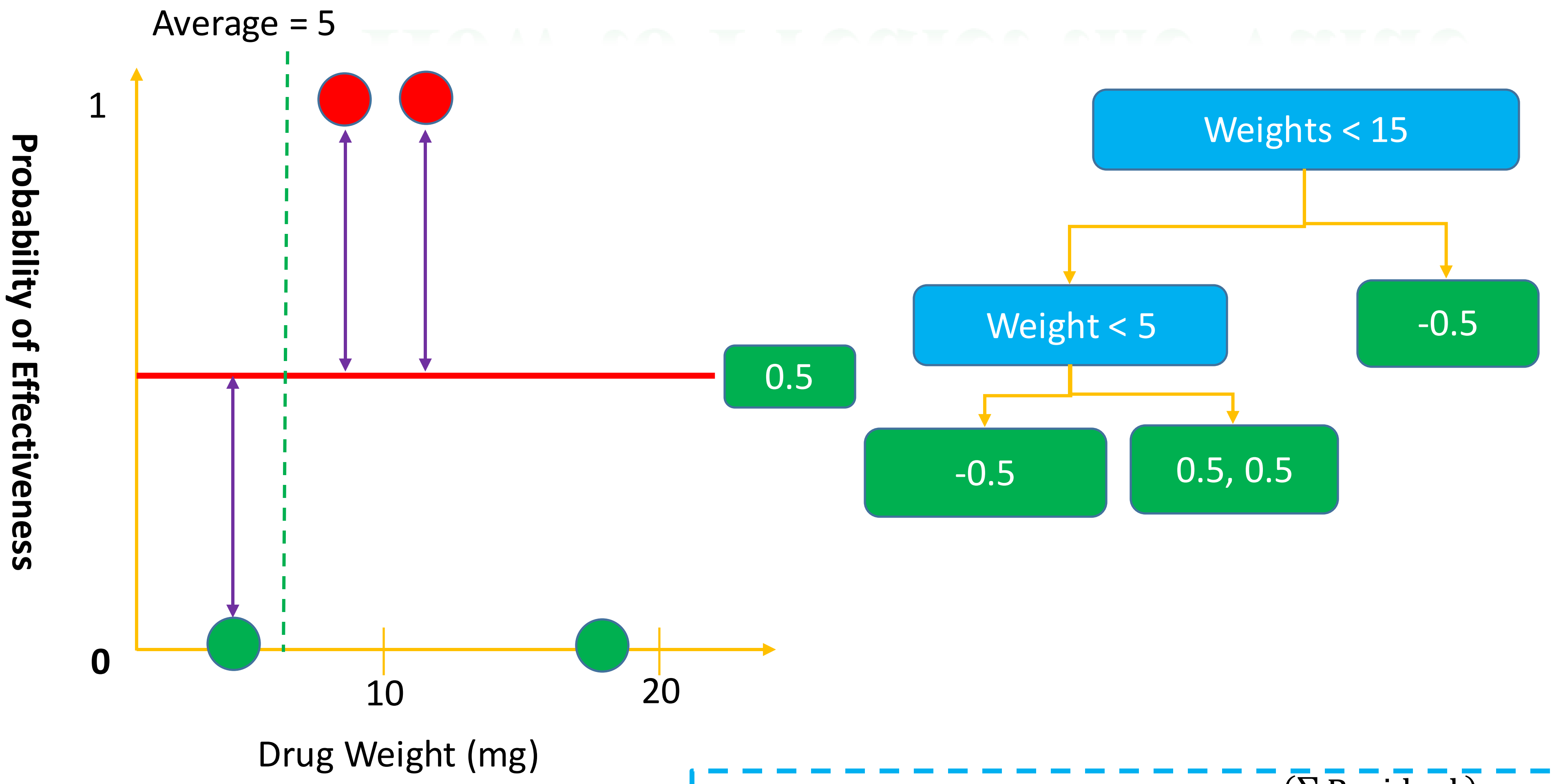
Cover = [(0.5 * (1- 0.5)]*4 = 1

Default mininmum XGBoot Cover is 1

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

AI VIET NAM
@aivietnam.edu.vn

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
● (green)	No
● (red)	Yes
● (red)	Yes
● (green)	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

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

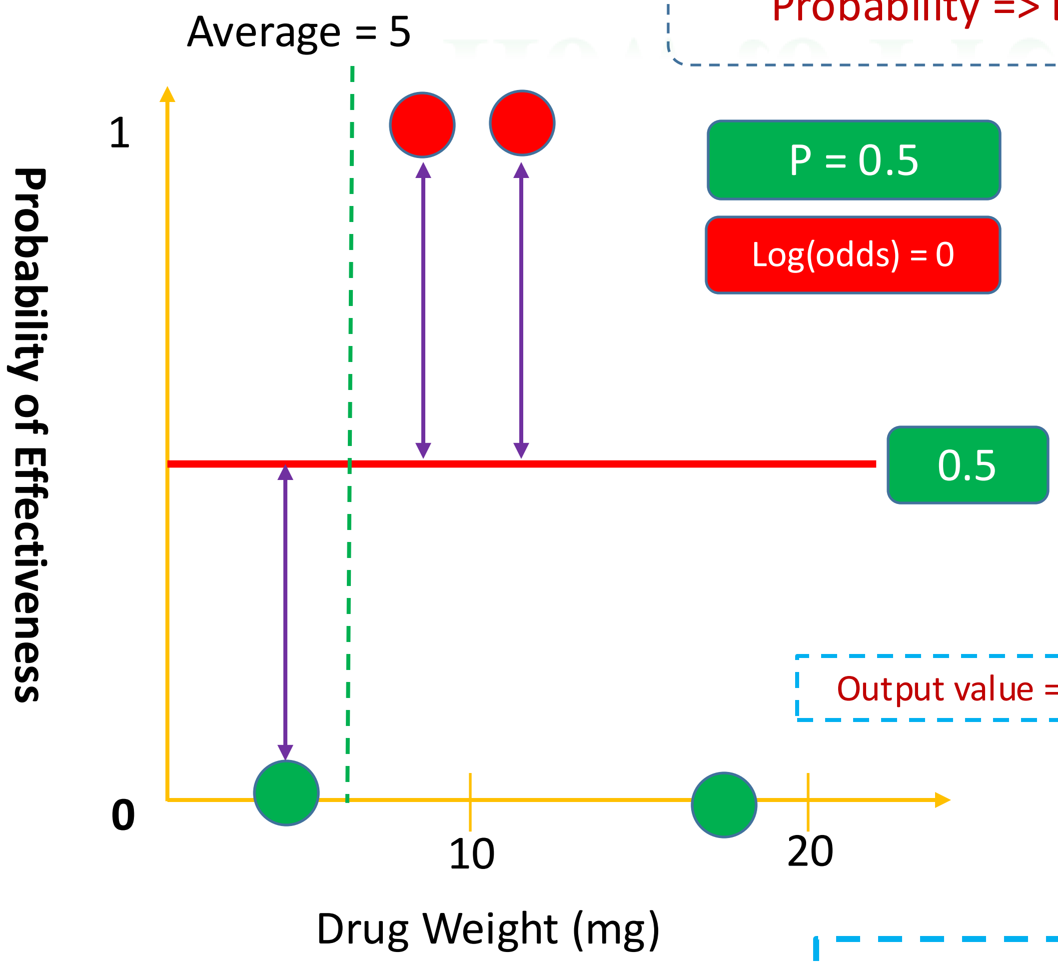
In XGBoost (or Gradient Boost), the initial prediction is that the log(odds)

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

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

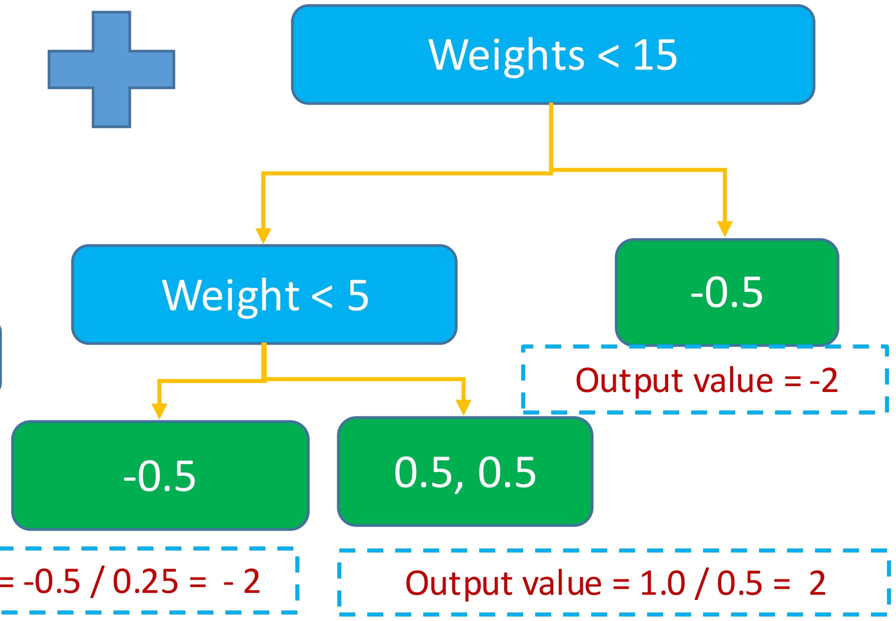
How to Predict the Value

Probability => Log(odds)



P = 0.5

Log(odds) = 0



$\lambda = 0$

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

Tranformation formular for getting value at a leaf.

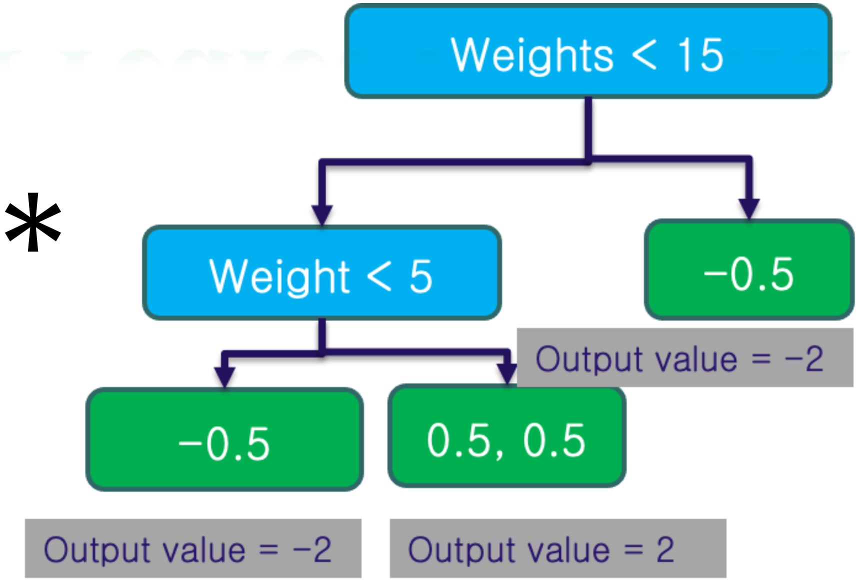
How to Predict the Value

Probability => Log(odds)

P = 0.5

Log(odds) = 0

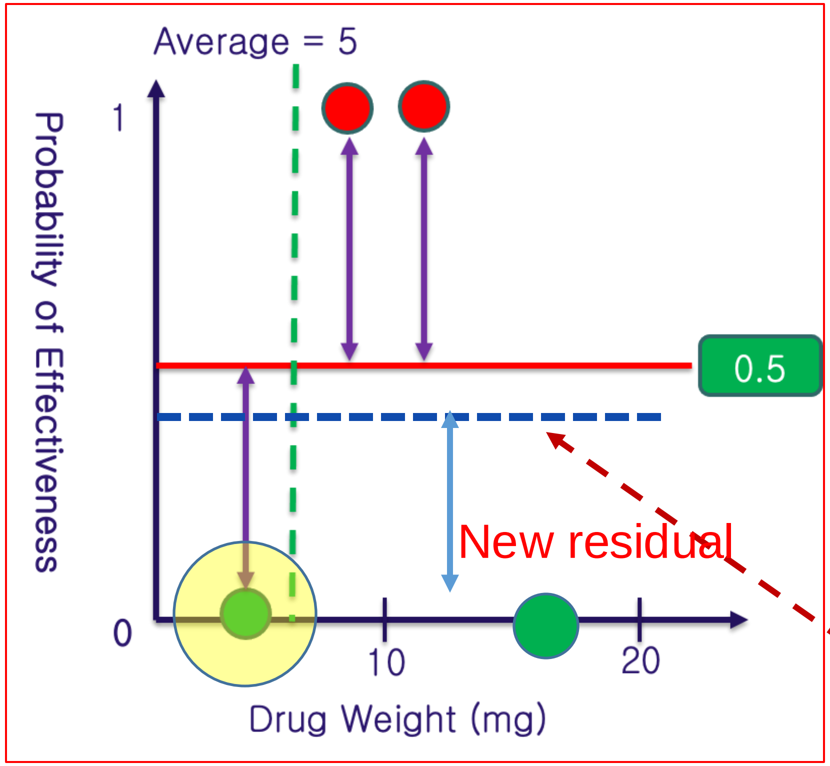
+ α *



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

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

α = 0.3



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

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

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

AI VIET NAM
@aivietnam.edu.vn

How to Predict the Value

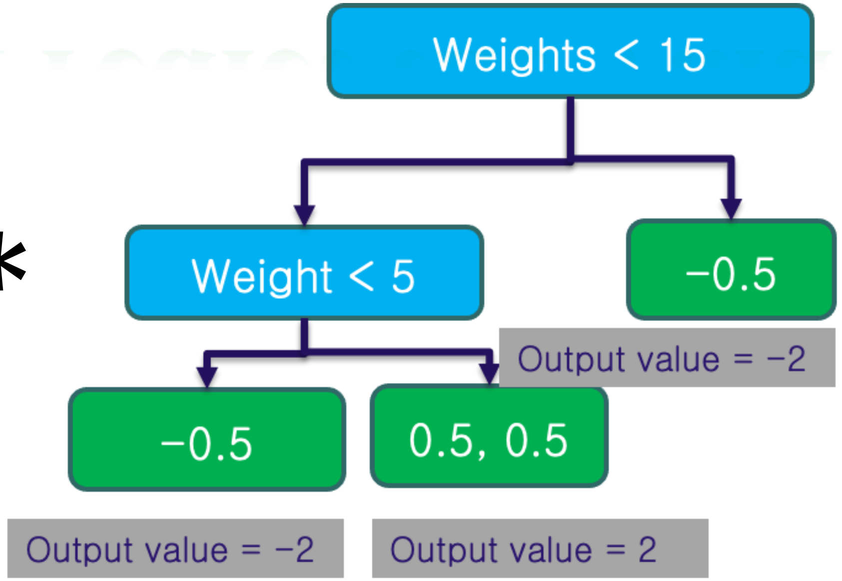
Probability => Log(odds)

Can we change P?

P = 0.5

Log(odds) = 0

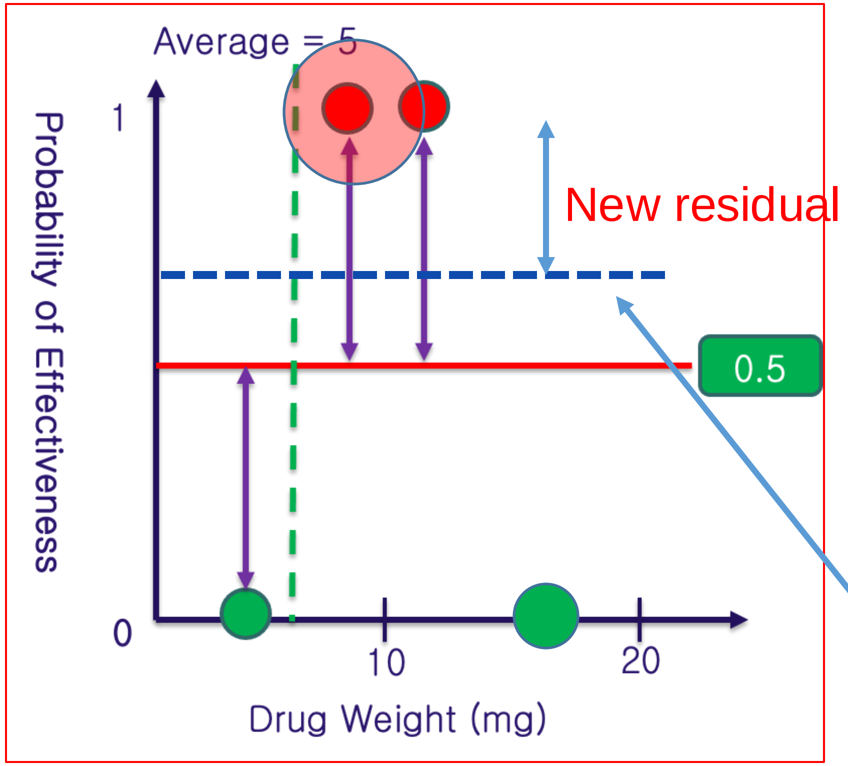
α *



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

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

$\alpha = 0.3$



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

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

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

AI VIET NAM
@aivietnam.edu.vn

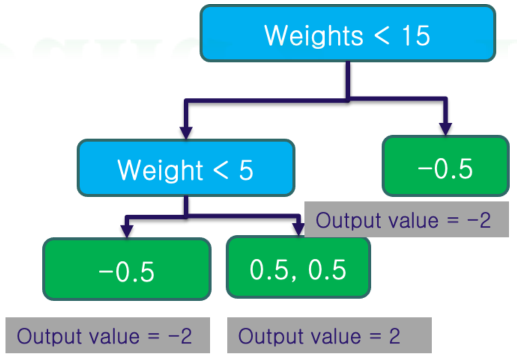
Build 2nd Tree

Probability => Log(odds)

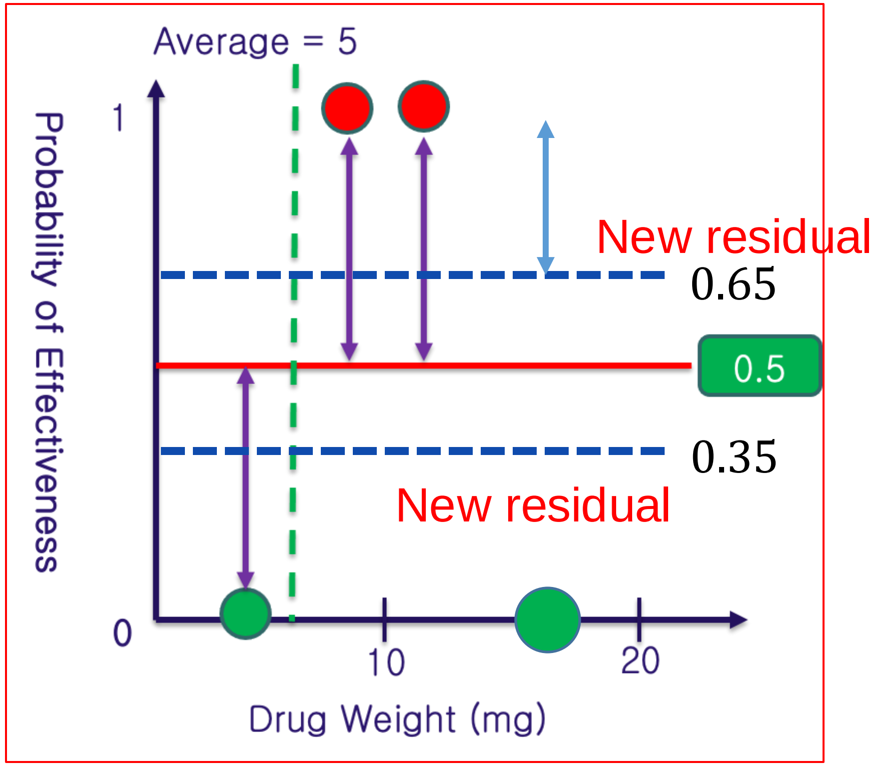
P = 0.5

Log(odds) = 0

+ ∝ *



+ ∝ *



-0.35, 0.35, 0.35, -0.35

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

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

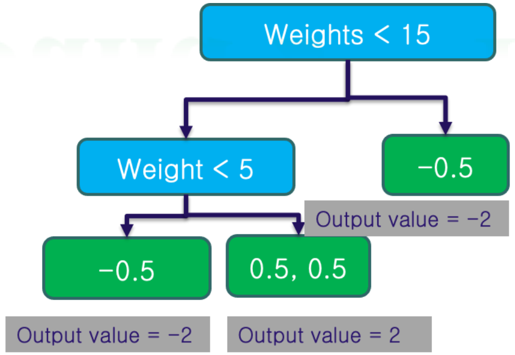
Build 2nd Tree

Probability => Log(odds)

P = 0.5

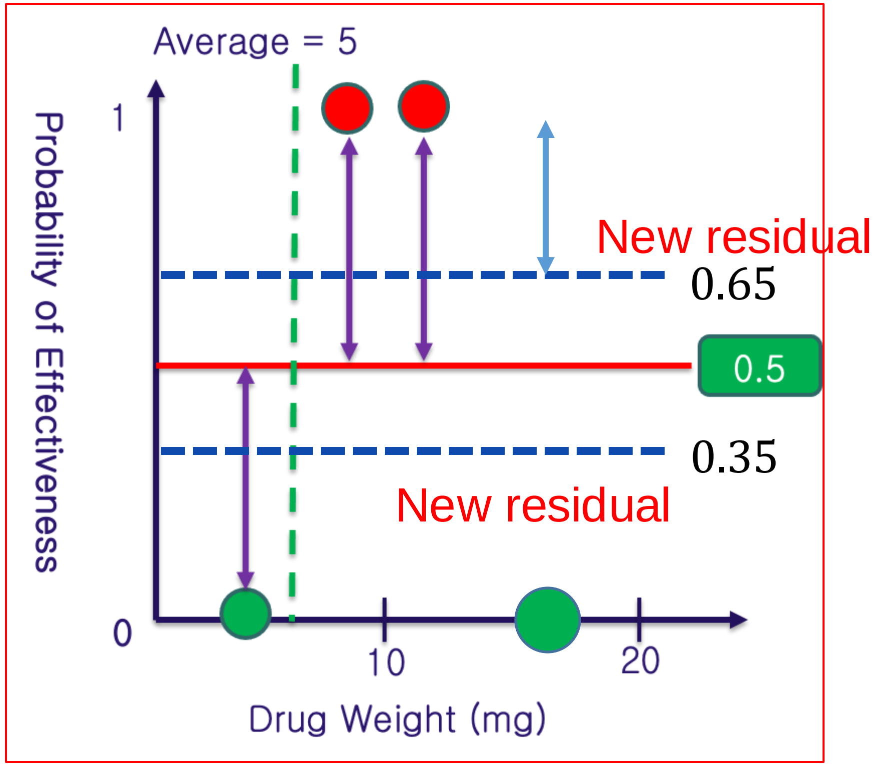
Log(odds) = 0

+ ∝ *



+ ∝ *

-0.35, 0.35, 0.35, -0.35



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

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

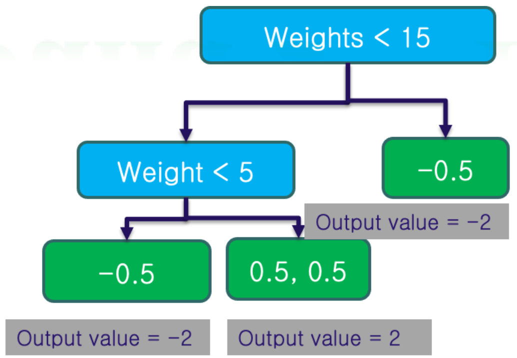
Build 2nd Tree

Probability => Log(odds)

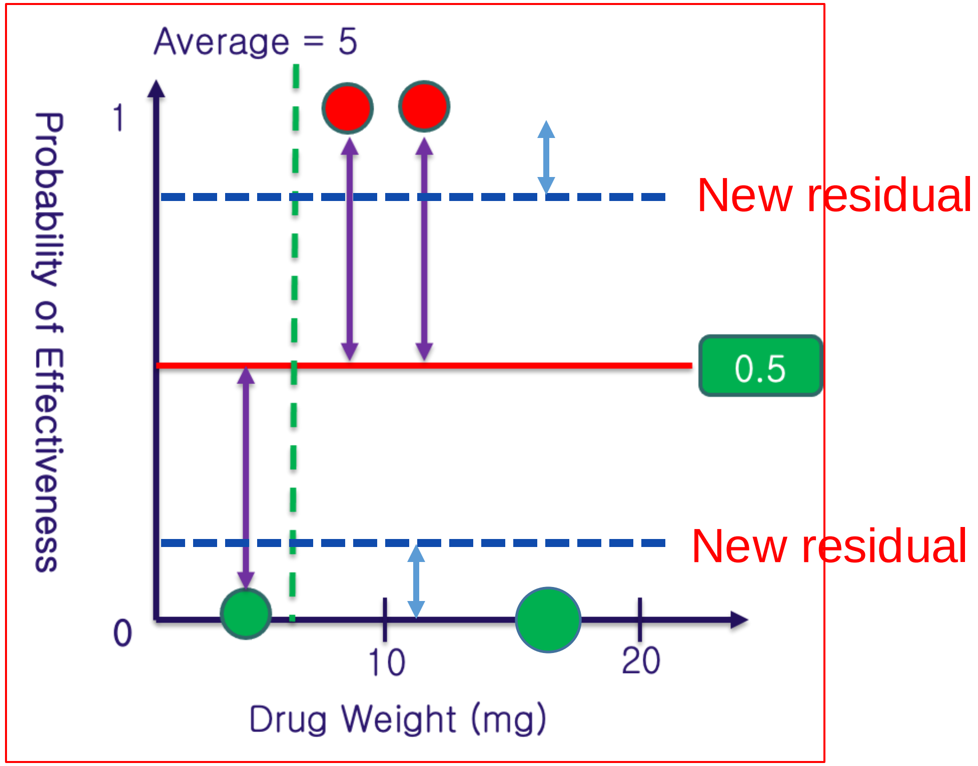
P = 0.5

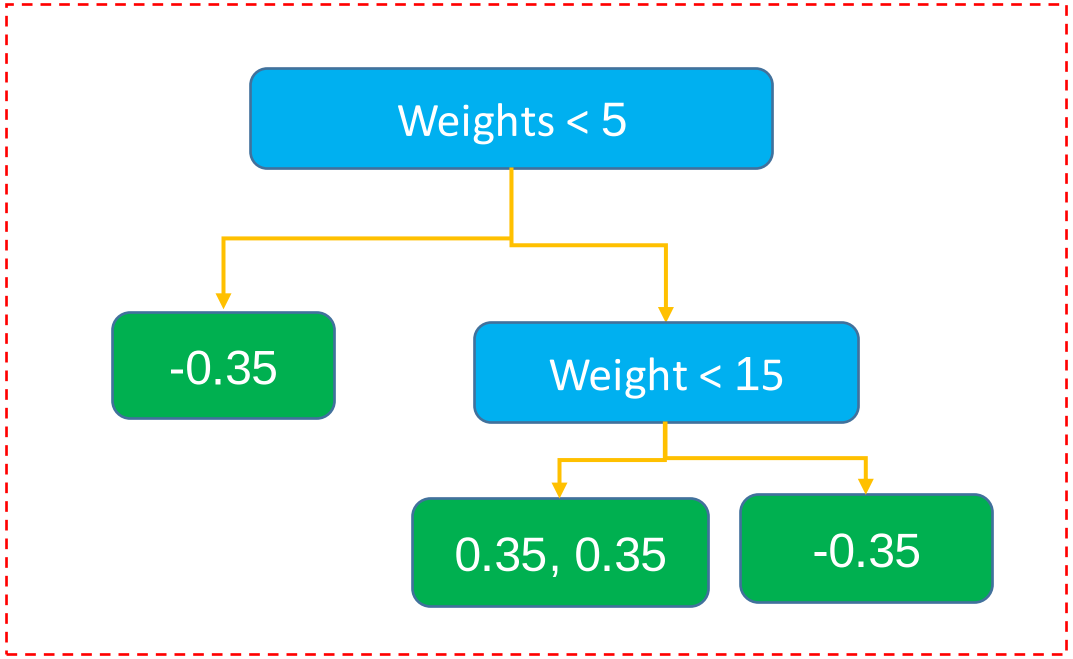
Log(odds) = 0

+ ∝ *



+ ∝ *





Q & A

1. When do you stop to build the Tree

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

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

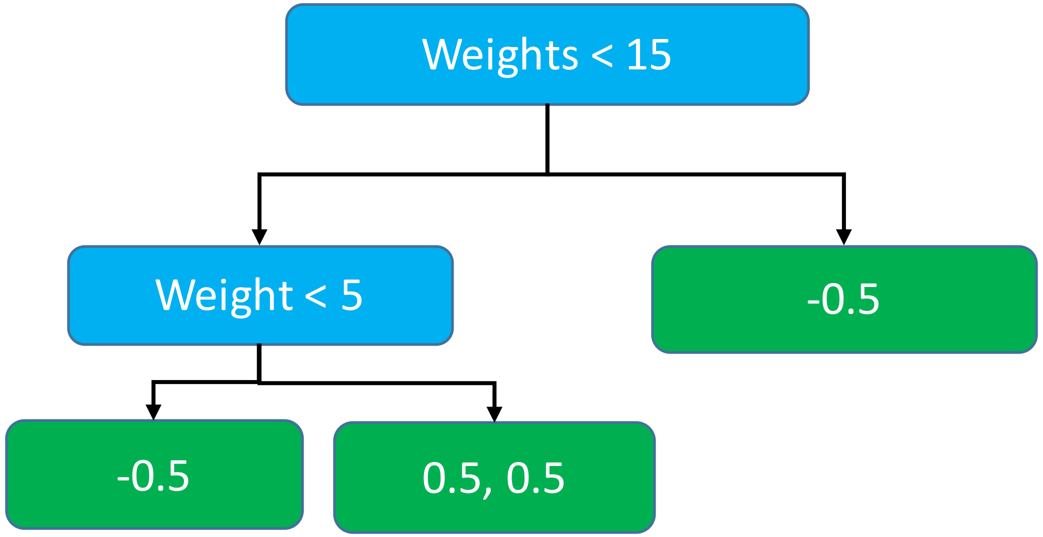
Outline

- Regularization
- Regression XGBoost
- Classification XGBoost
- XGBoost: Clearly Explain
- Time Series Example
- Summary

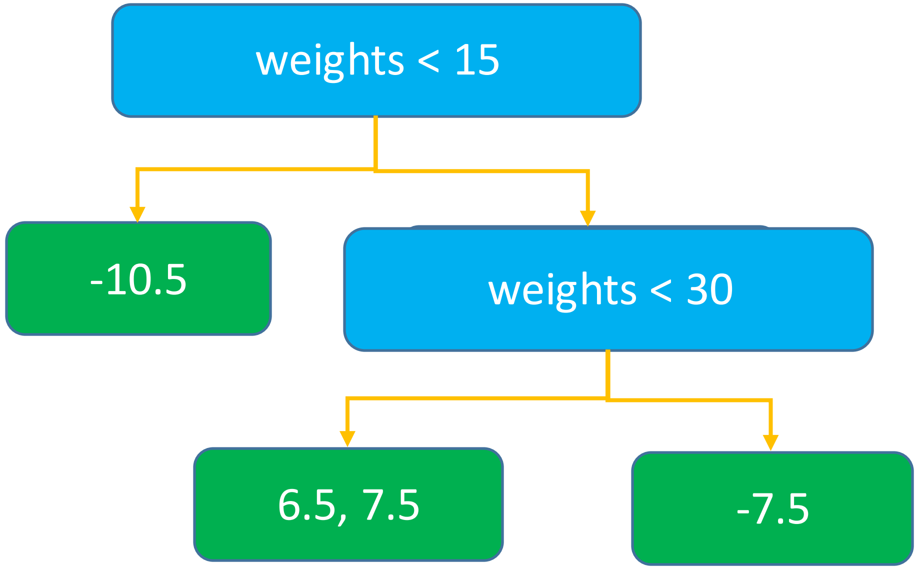
AI VIET NAM
@aivietnam.edu.vn

XGBoost: Behind The Scenes

Classification



Regression



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

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

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

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

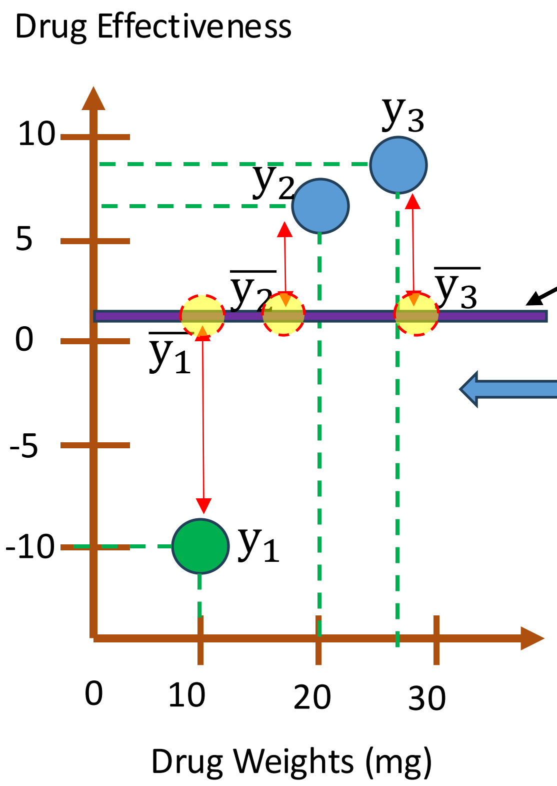
XGBoost: Behind The Scenes

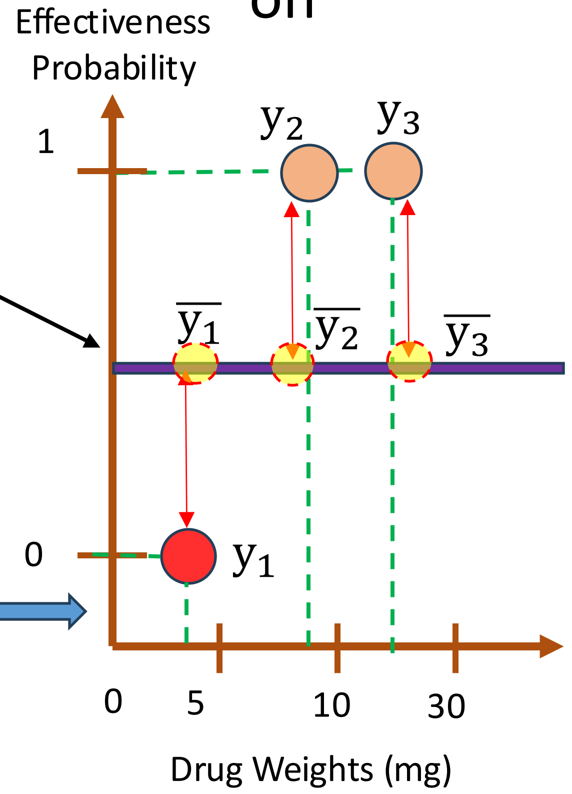
Regression

Dự đoán ban đầu hiệu quả thuốc

0.5

Classification





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

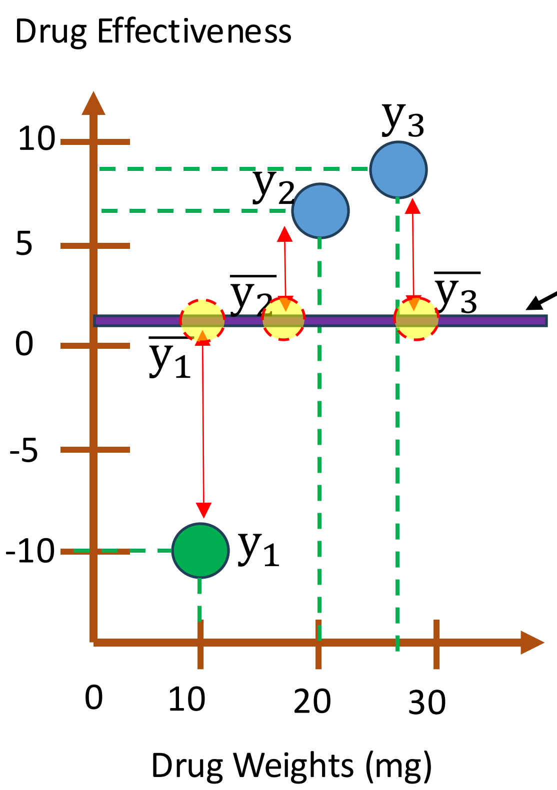
$$\sum_{i=1}^{3} \mathcal{L}(y_i, \overline{y_i}) \text{ Where}$$

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

Sử dụng loss functions xây dựng cây

XGBoost: Behind The Scenes

Regression



Dự đoán ban đầu hiệu quả thuốc

0.5

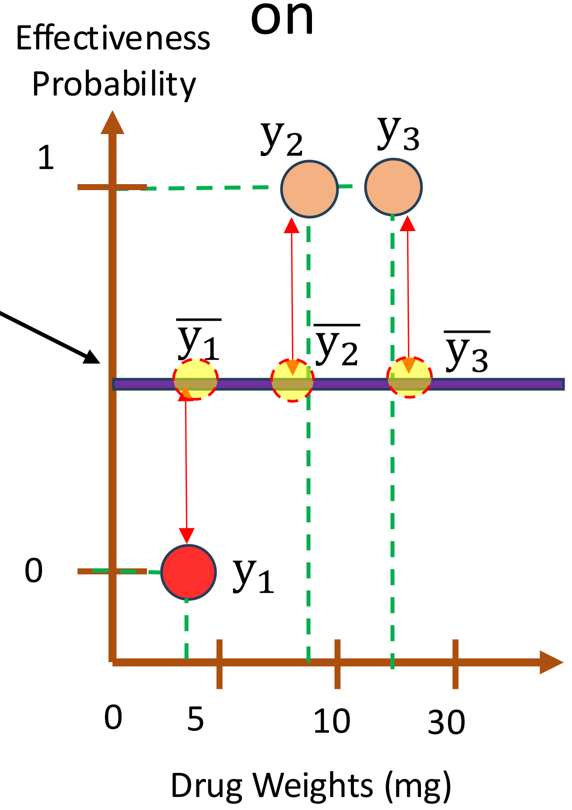
$$\sum_{i=1}^{n} \mathcal{L}(y_i, \overline{y_i}) + \gamma T + \lambda P^2$$

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

Classification

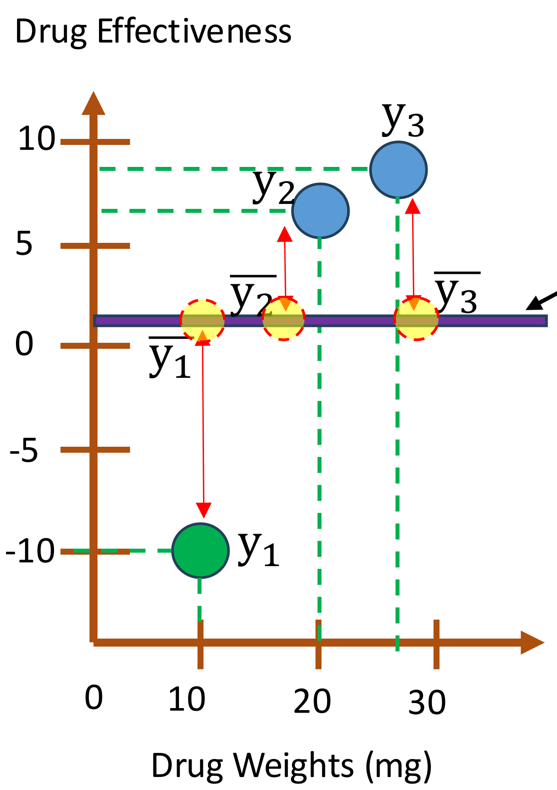


XGBoost: Behind The Scenes

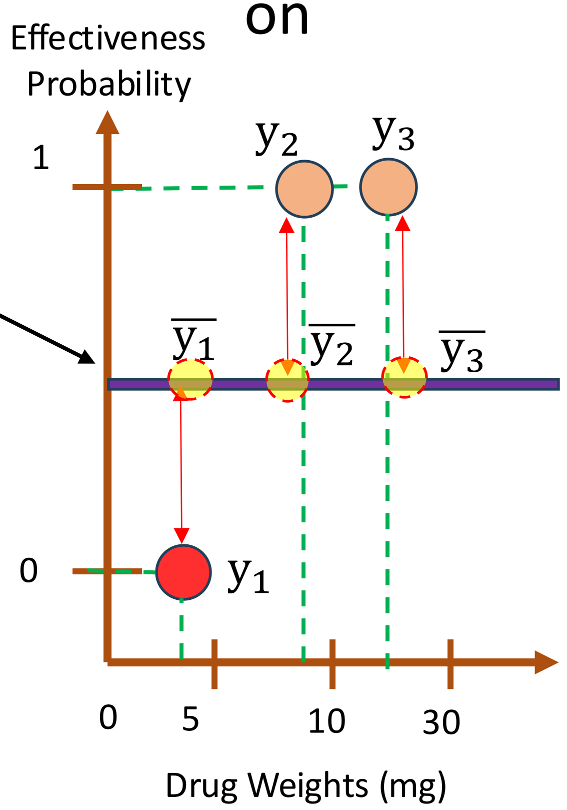
Dự đoán ban đầu hiệu quả thuốc

0.5

Regression



Classification



Bỏ qua γ

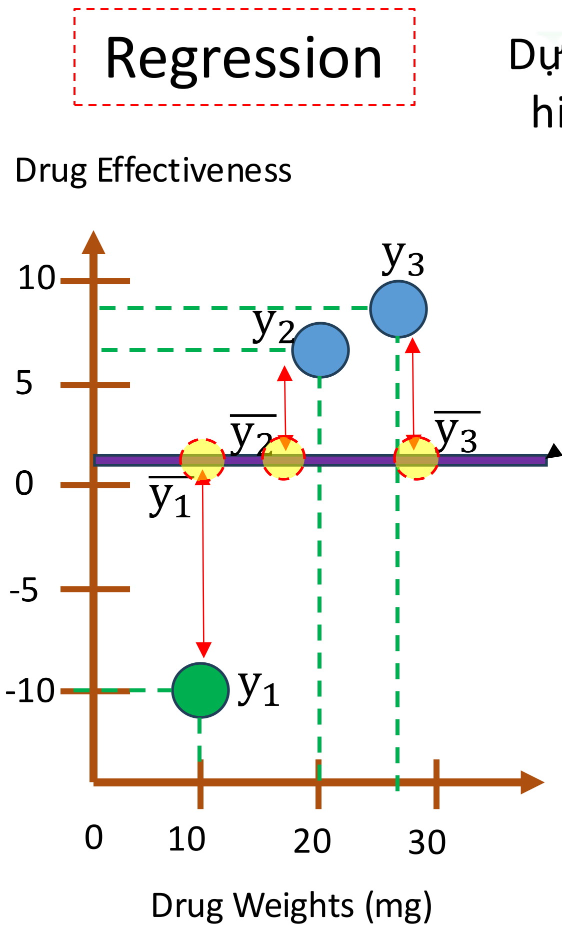
$$\sum_{i=1}^{n} \mathcal{L}(y_i, \overline{y_i}) + \cancel{\gamma T} + \lambda P^2$$

γ is a user definable penalty to encourse pruning

XGBoost can prune even when $\gamma = 0$

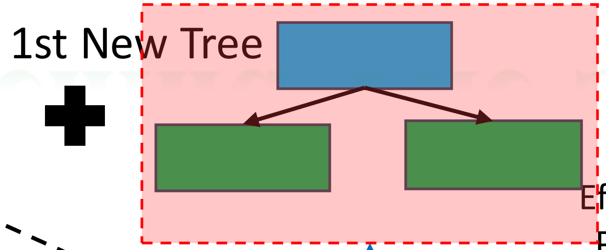
Pruning is excuted after the full tree built
=> It plays no role in deriving the Optimal

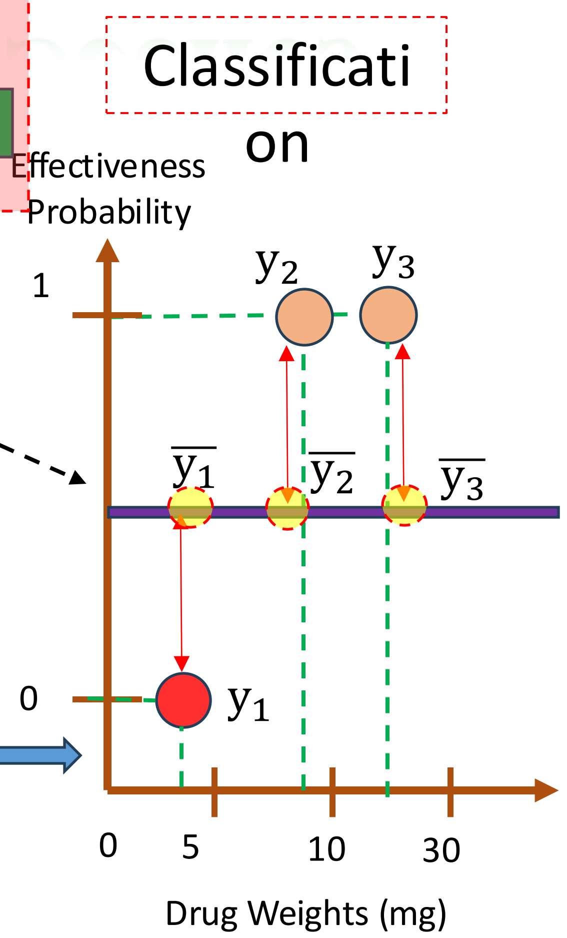
XGBoost: Behind The Scenes



Dự đoán ban đầu hiệu quả thuốc

0.5





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.

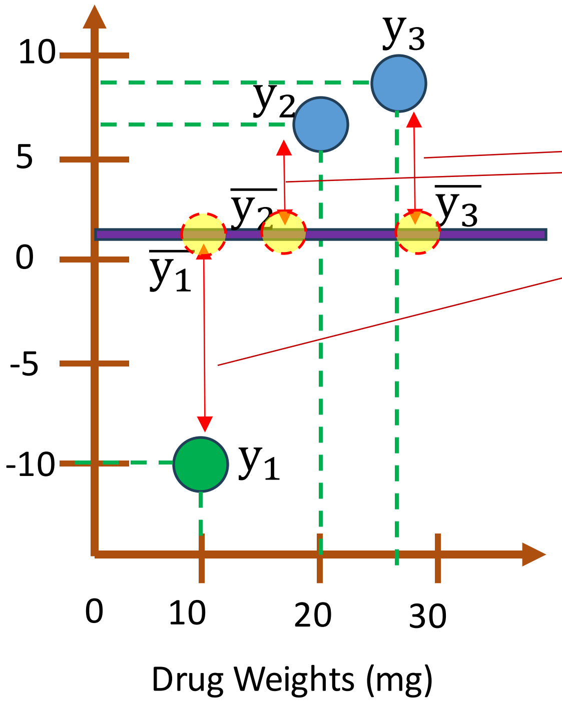
AI VIET NAM
@aivietnam.edu.vn

XGBoost: Behind The Scenes

Regression

Dự đoán ban đầu hiệu quả thuốc: 0.5

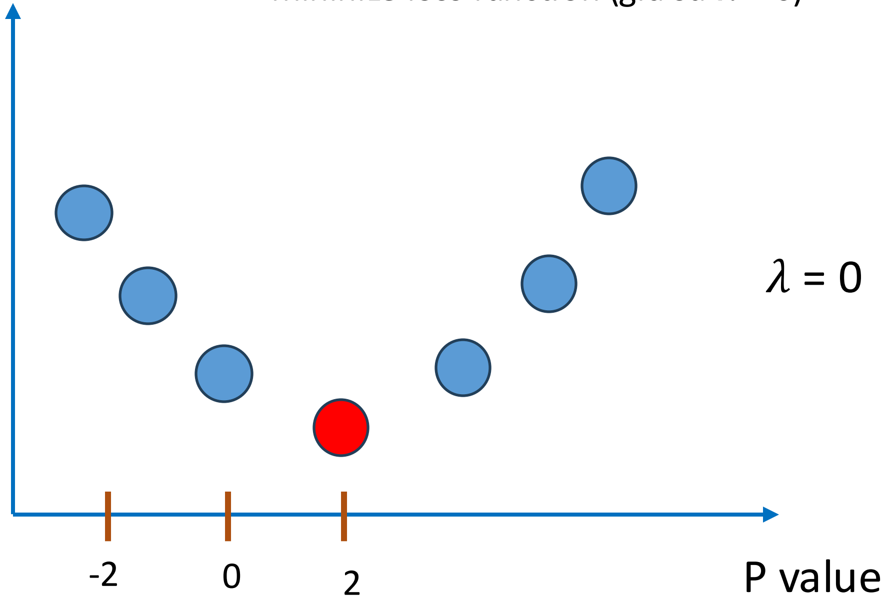
Residual: -10.5, 6.5, 7.5



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

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



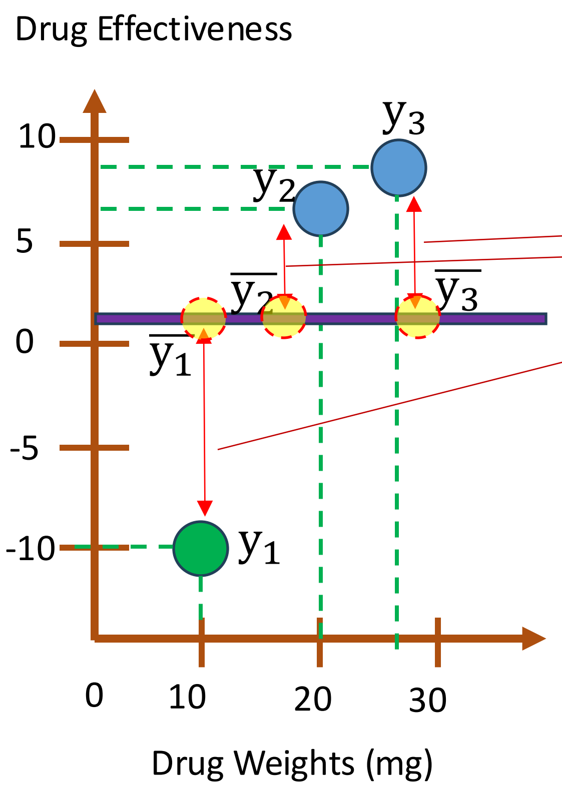
Giá trị P cần tìm là giá trị ứng với đạo hàm của loss theo P bằng 0

XGBoost: Behind The Scenes

Regression

Dự đoán ban đầu hiệu quả thuốc: 0.5

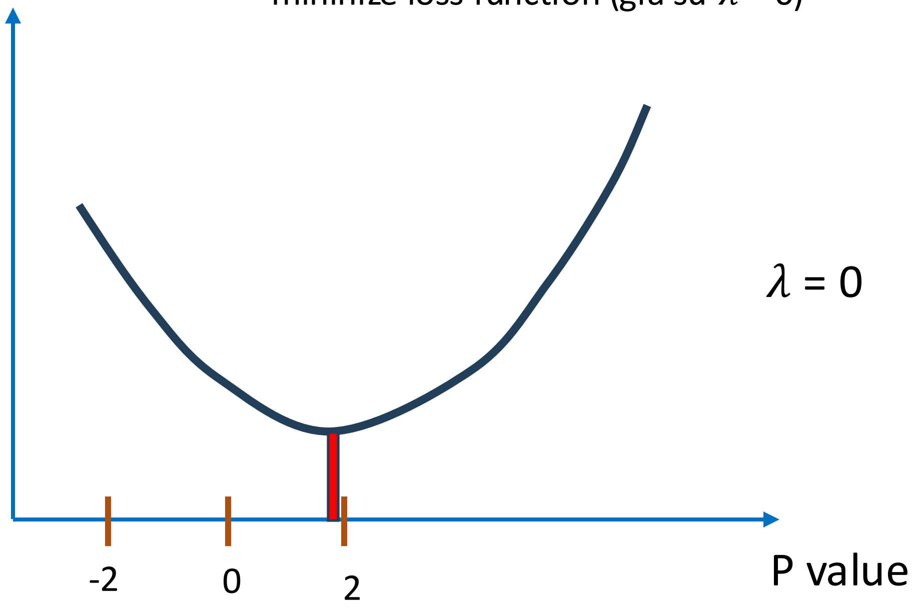
Residual: -10.5, 6.5, 7.5



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

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

Loss function



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

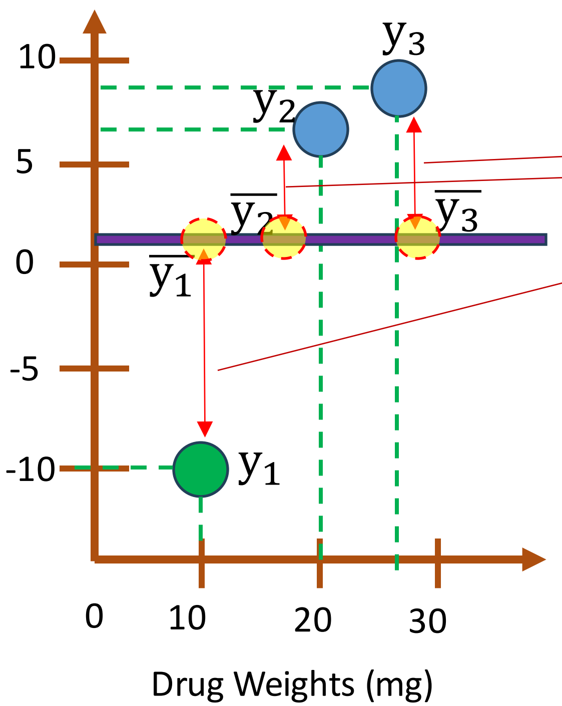
AI VIET NAM
@aivietnam.edu.vn

XGBoost: Behind The Scenes

Regression

Dự đoán ban đầu hiệu quả thuốc: 0.5

Whats happen if λ is very large?

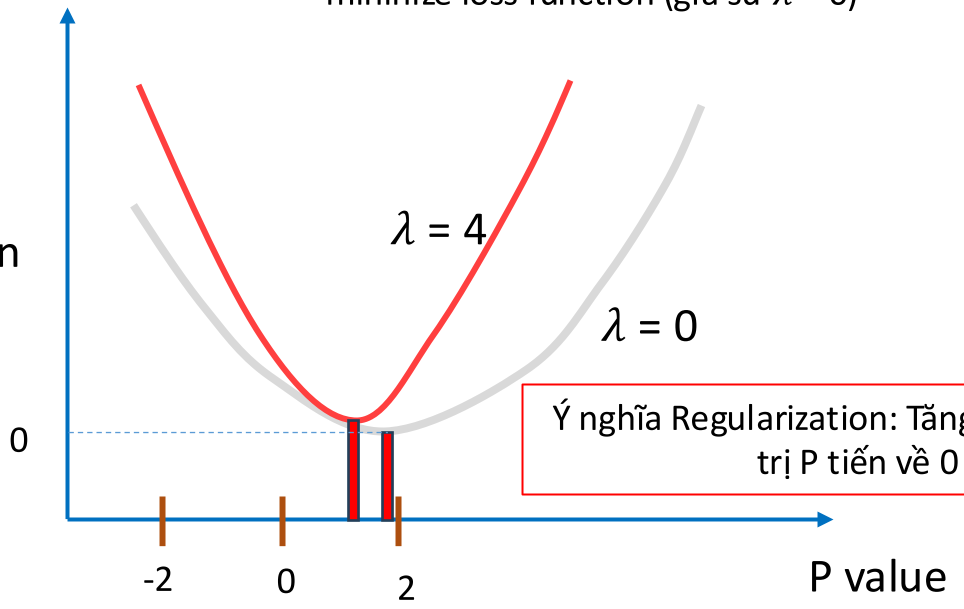


Residual: -10.5, 6.5, 7.5

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

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

Loss function



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

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

XGBoost: 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 Taylor Approximation

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

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

g (gradient) presents the first derivative of the loss function

h (hessian) presents the second derivative of the loss function

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

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

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

AI VIET NAM
@aivietnam.edu.vn

XGBoost Regression: Output Value

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

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

1

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

2

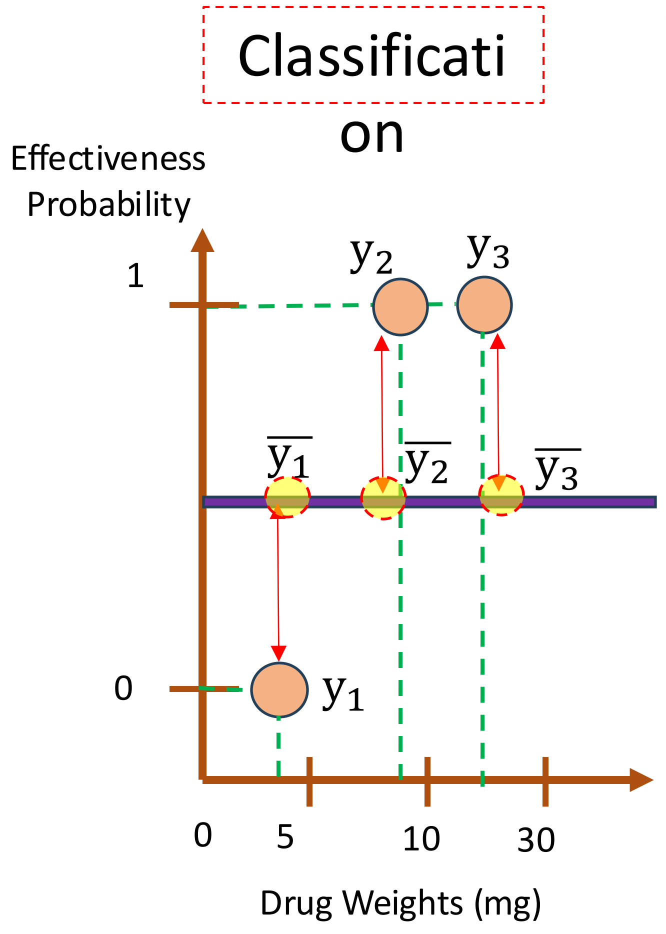
$$h_i = \frac{d^2}{d\bar{y}_i^2}\frac{1}{2}(y_i - \bar{y}_i)^2 = 1$$

$$P = \frac{-(g_1+g_2+\cdots+g_n)}{h_1+h_2+\cdots+h_n+\lambda} = \frac{-(-(y_1-\bar{y}_1)+-(y_2-\bar{y}_2)+\cdots+-(y_n-\bar{y}_n))}{1+1+\cdots+1+\lambda} = \frac{\text{sum of residual}}{\text{numer of sum residual}+\lambda}$$

Output value af the leaf (or terminal node)

AI VIET NAM
@aivietnam.edu.vn

XGBoost For Classification: Output Value



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

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

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

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

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

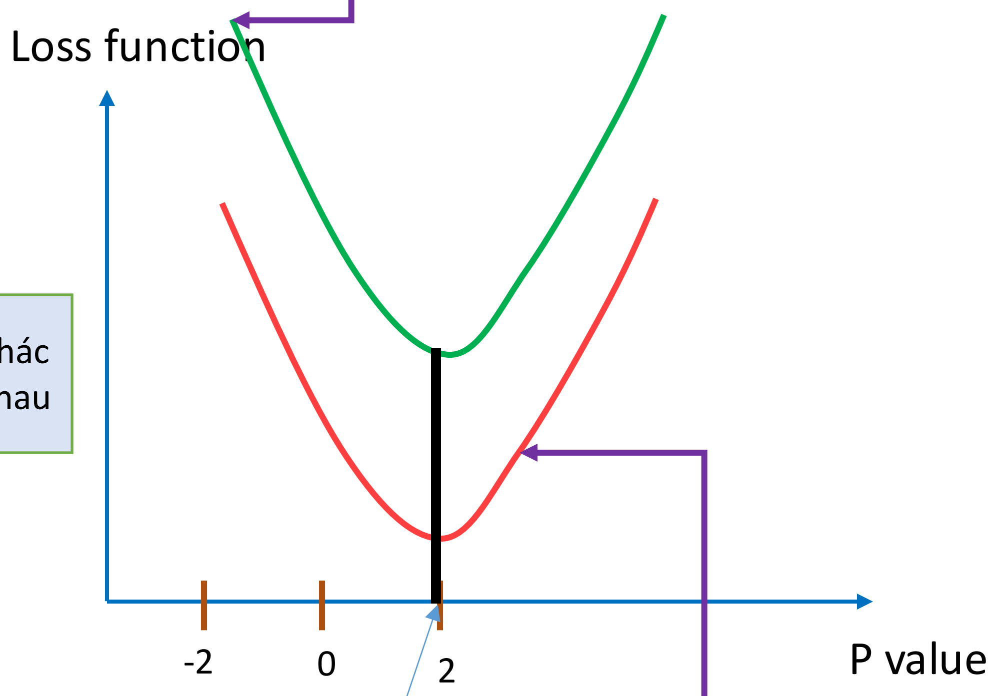
$$\mathcal{L}(y_i, \bar{y}_i + P) \approx \mathcal{L}(y_i, \bar{y}_i) + gP + \frac{1}{2} hP^2$$

2

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

Khác nhau

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



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

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

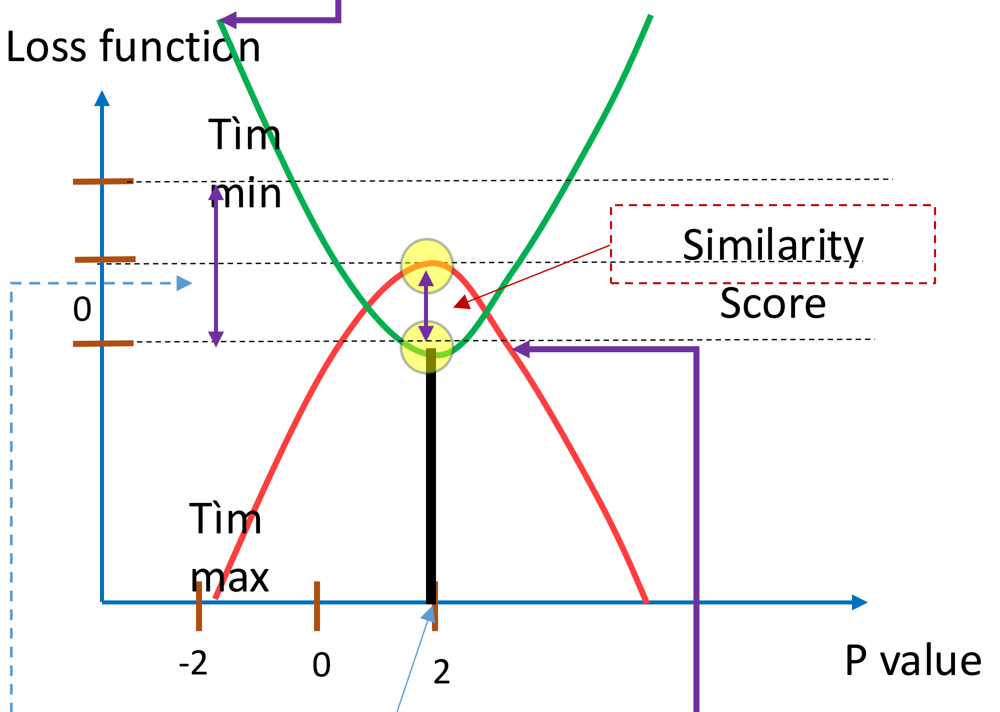
Rất khó để tìm optimalization, nên cũng ta sẽ sắp sỉ hàm loss bằng Second Order Tayler Approximation

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

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

2

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



Implementation Similarity Score

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

AI VIET NAM
@aivietnam.edu.vn

XGBoost: Similarity Score

1

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

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

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

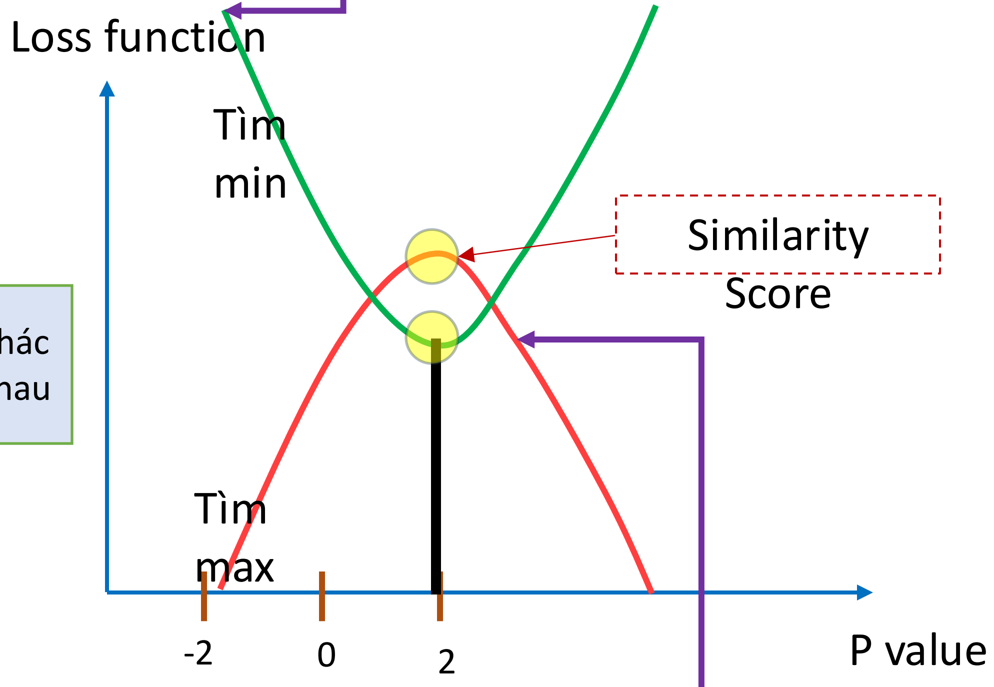
$$\mathcal{L}(y_i, \bar{y}_i + P) \approx \mathcal{L}(y_i, \bar{y}_i) + gP + \frac{1}{2}hP^2$$

2

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

Khác nhau

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



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

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

XGBoost Regression: Similarity Score

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

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

(1)

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

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

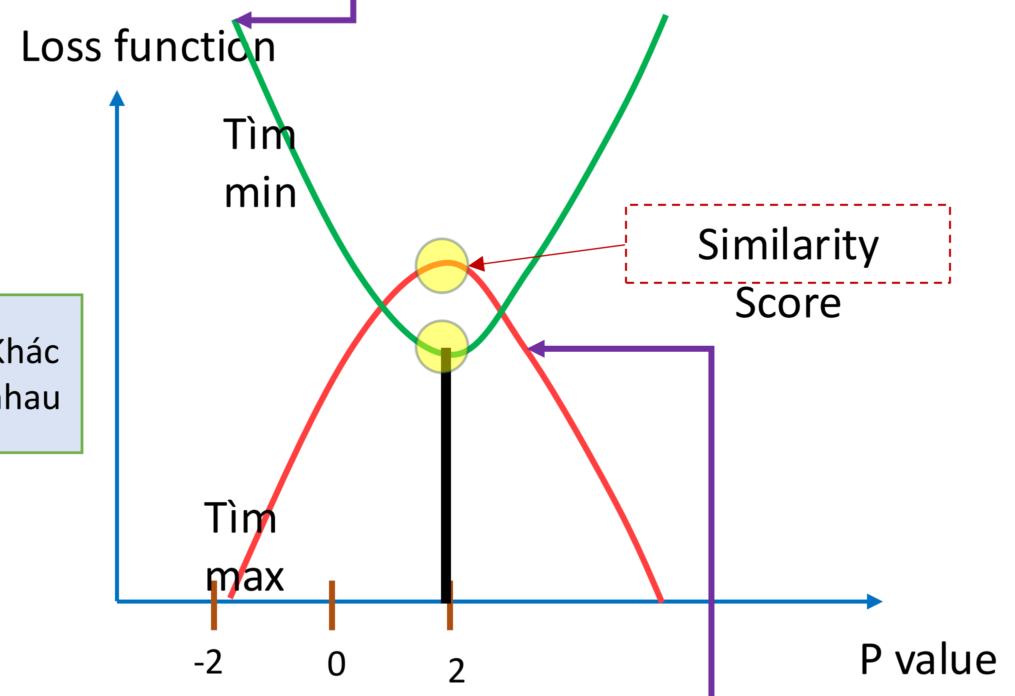
$$\mathcal{L}(y_i, \overline{y_i} + P) \approx \mathcal{L}(y_i, \overline{y_i}) + \left[\frac{d}{d\overline{y_i}} \mathcal{L}(y_i, \overline{y_i})\right] P + \frac{1}{2}\left[\frac{d}{d\overline{y}_i^2} \mathcal{L}(y_i, \overline{y_i})\right] P^2$$

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

(2)

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

Khác nhau



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

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

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

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

AI VIET NAM
@aivietnam.edu.vn

XGBoost Classification: Similarity Score

1

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

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

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

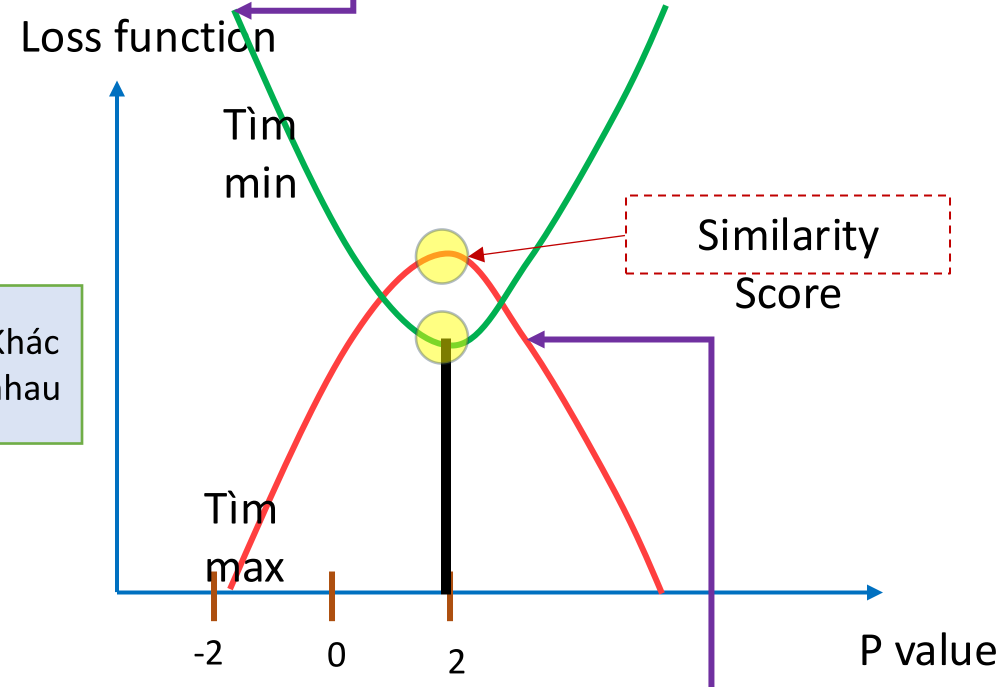
$$\mathcal{L}(y_i, \overline{y_i^0} + P) \approx \mathcal{L}(y_i, \bar{y}_i) + gP + \frac{1}{2}hP^2$$

2

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

Khác nhau

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



$g_i = -(y_i - \bar{y}_i)$

$h_i = \bar{y}_i \times (1 - \bar{y}_i)$

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

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

Outline

- Regularization
- Regression XGBoost
- Classification XGBoost
- XGBoost: Clearly Explain
- Time Series Example
- Summary

AI VIET NAM
@aivietnam.edu.vn

Time Series Forecasting

We will focus on the energy consumption problem, where given a sufficiently large dataset of the daily energy consumption of different households in a city, we are tasked to predict as accurately as possible the future energy demands.

london_energy

LCLid	Date	KWH
MAC000002	2012-10-12	7.098
MAC000002	2012-10-13	11.087
MAC000002	2012-10-14	13.223
MAC000002	2012-10-15	10.257
MAC000002	2012-10-16	9.769
MAC000002	2012-10-17	10.885
MAC000002	2012-10-18	10.751
MAC000002	2012-10-19	8.431
MAC000002	2012-10-20	17.578
MAC000002	2012-10-21	24.49
MAC000002	2012-10-22	18.885
MAC000002	2012-10-23	10.485
MAC000002	2012-10-24	15.537

Preprocessing

```python
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("/content/drive/MyDrive/AIO2024/london_energy.csv")
print(df.isna().sum())
df.head()
```

```
LCLid    0
Date     0
KWH      0
dtype: int64
```

	LCLid	Date	KWH
0	MAC000002	2012-10-12	7.098
1	MAC000002	2012-10-13	11.087
2	MAC000002	2012-10-14	13.223
3	MAC000002	2012-10-15	10.257
4	MAC000002	2012-10-16	9.769

Time Series Forecasting

XGBoost For Training

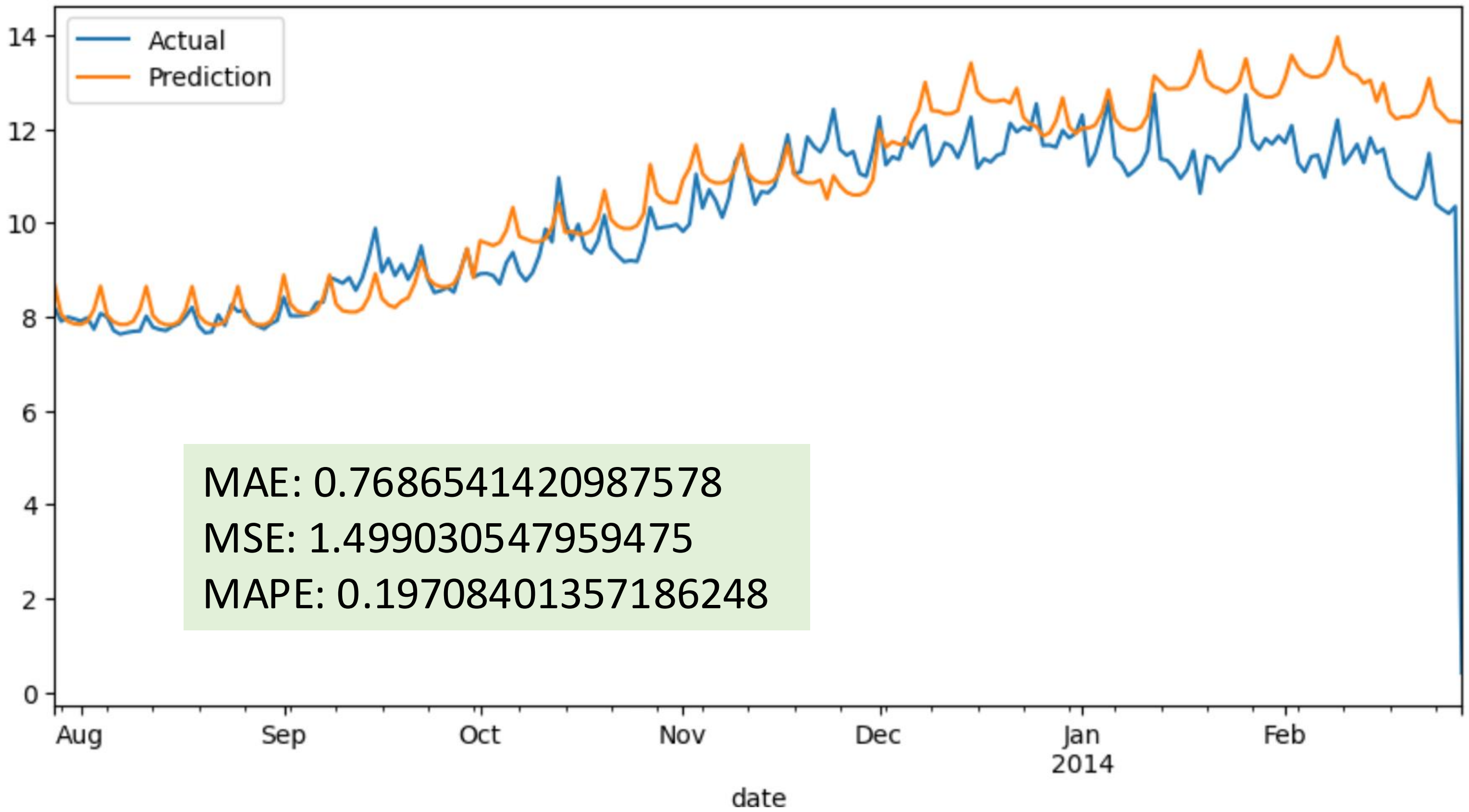
```python
from xgboost import XGBRegressor
import lightgbm as lgb
from sklearn.model_selection import TimeSeriesSplit, GridSearchCV

# XGBoost
cv_split = TimeSeriesSplit(n_splits=4, test_size=100)
model = XGBRegressor()
parameters = {
    "max_depth": [3, 4, 5],
    "learning_rate": [0.01, 0.05],
    "n_estimators": [100, 300],
    "colsample_bytree": [0.3]
}


grid_search = GridSearchCV(estimator=model, cv=cv_split, param_grid=parameters)
grid_search.fit(X_train, y_train)
```

AI VIET NAM
@aivietnam.edu.vn

Time Series Forecasting

XGBoost For Predicting

```
# Evaluating GridSearch results
prediction = grid_search.predict(X_test)
plot_predictions(testing_dates, y_test, prediction)
evaluate_model(y_test, prediction)
```



MAE: 0.7686541420987578
MSE: 1.499030547959475
MAPE: 0.19708401357186248

AI VIET NAM
@aivietnam.edu.vn

Time Series Forecasting

The model performs relatively well, but is there a way to improve it even further?
The answer is yes

Metric	XGBoost
MAE	0.768
MSE	1.499
MAPE	0.197

Enhance our dataset with weather data from the London Weather Dataset

london_weather

date	cloud_cover	sunshine	global_radiation	max_temp	mean_temp	min_temp	precipitation	pressure	snow_depth
19790101	2.0	7.0	52.0	2.3	-4.1	-7.5	0.4	101900.0	9.0
19790102	6.0	1.7	27.0	1.6	-2.6	-7.5	0.0	102530.0	8.0
19790103	5.0	0.0	13.0	1.3	-2.8	-7.2	0.0	102050.0	4.0
19790104	8.0	0.0	13.0	-0.3	-2.6	-6.5	0.0	100840.0	2.0
19790105	6.0	2.0	29.0	5.6	-0.8	-1.4	0.0	102250.0	1.0
19790106	5.0	3.8	39.0	8.3	-0.5	-6.6	0.7	102780.0	1.0
19790107	8.0	0.0	13.0	8.5	1.5	-5.3	5.2	102520.0	0.0
19790108	8.0	0.1	15.0	5.8	6.9	5.3	0.8	101870.0	0.0
19790109	4.0	5.8	50.0	5.2	3.7	1.6	7.2	101170.0	0.0
19790110	7.0	1.9	30.0	4.9	3.3	1.4	2.1	98700.0	0.0
19790111	1.0	6.8	55.0	2.9	2.6	0.3	2.3	98960.0	0.0
19790112	3.0	6.4	54.0	2.0	0.4	-2.0	0.0	100650.0	1.0

Data Analysis: Filling missing value

```
[22] df_weather = pd.read_csv("/content/drive/MyDrive/AIO2024/london_weather.csv")
     print(df_weather.isna().sum())
     df_weather.head()
```

```
date                 0
cloud_cover         19
sunshine             0
global_radiation    19
max_temp             6
mean_temp           36
min_temp             2
precipitation        6
pressure             4
snow_depth        1441
dtype: int64
```

```
# Parsing dates
df_weather["date"] = pd.to_datetime(df_weather["date"], format="%Y%m%d")

# Filling missing values through interpolation
df_weather = df_weather.interpolate(method="ffill")

# Enhancing consumption dataset with weather information
df_avg_consumption = df_avg_consumption.merge(df_weather, how="inner", on="date")
df_avg_consumption.head()
```

Time Series Forecasting

Prepare New Dataset

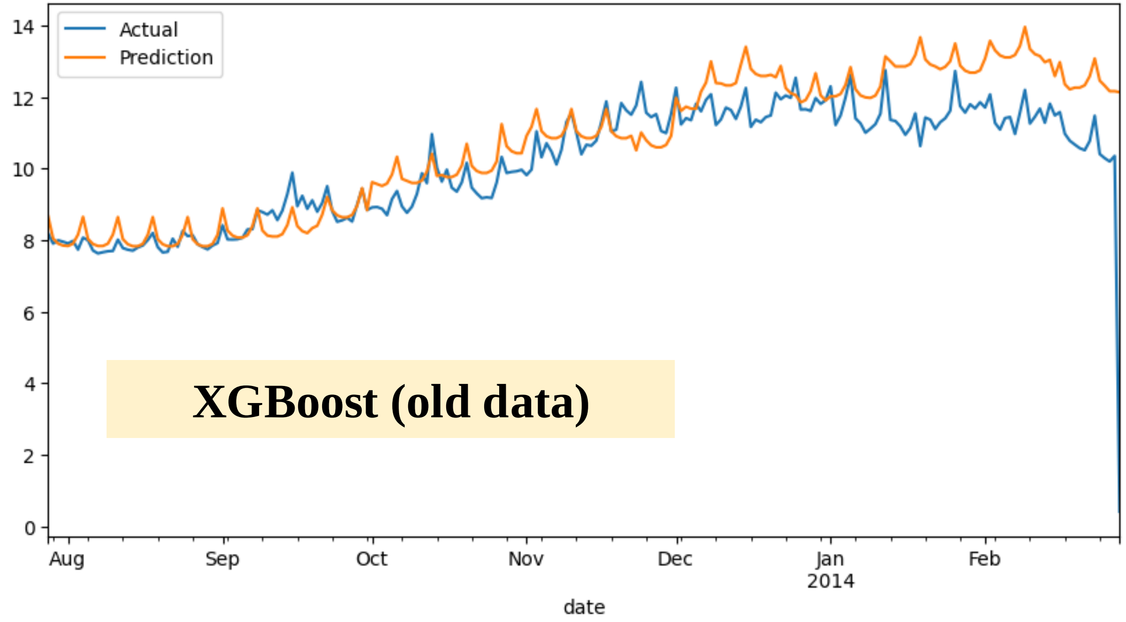
```
# Dropping unnecessary `date` column
training_data = training_data.drop(columns=["date"])
testing_dates = testing_data["date"]
testing_data = testing_data.drop(columns=["date"])

X_train = training_data[["day_of_week", "day_of_year", "month", "quarter", "year",\
                         "cloud_cover", "sunshine", "global_radiation", "max_temp",\
                         "mean_temp", "min_temp", "precipitation", "pressure",\
                         "snow_depth"]]
y_train = training_data["consumption"]


X_test = testing_data[["day_of_week", "day_of_year", "month", "quarter", "year",\
                         "cloud_cover", "sunshine", "global_radiation", "max_temp",\
                         "mean_temp", "min_temp", "precipitation", "pressure",\
                         "snow_depth"]]
y_test = testing_data["consumption"]
```

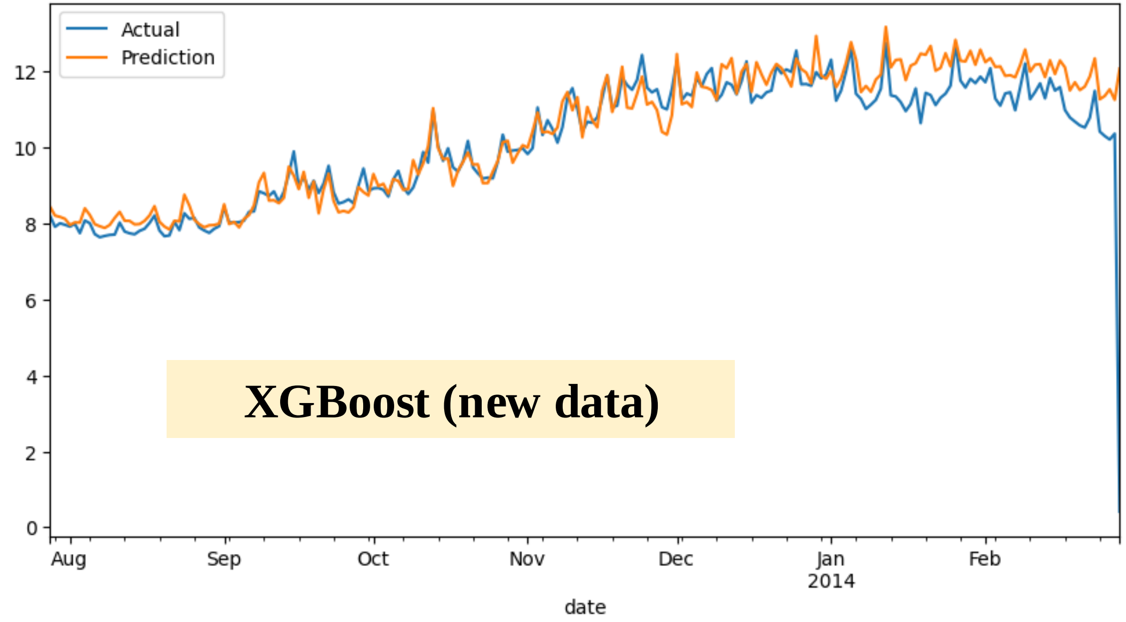
AI VIET NAM
@aivietnam.edu.vn

Time Series Forecasting

Performance Evaluation

Metric	XGBoost (old data)	XGBoost (new data)
MAE	0.768	0.423
MSE	1.499	0.864
MAPE	0.197	0.164





Outline

- Regularization
- Regression XGBoost
- Classification XGBoost
- XGBoost: Clearly Explain
- How to File Missing Values
- Time Series Example
- Summary

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

$$\mathcal{L}(y_i, \bar{y}_i) = [-y_i \log(\bar{y}_i) - (1 - y_i)\log(1 - \bar{y}_i)]$$

$$\mathcal{L}(y_i, \bar{y}_i) = -y_i \log(\bar{y}_i) - \log(1 - \bar{y}_i) + y_i \log(1 - \bar{y}_i)$$

$$\mathcal{L}(y_i, \bar{y}_i) = -y_i[\log(\bar{y}_i) - \log(1 - \bar{y}_i)] - \log(1 - \bar{y}_i))$$

$$\log(\bar{y}_i) - \log(1 - \bar{y}_i)] = \log\left(\frac{\bar{y}_i}{1-\bar{y}_i}\right) = \log(\text{odds})$$

$$\mathcal{L}(y_i, \bar{y}_i) = -y_i \log(\text{odds}) - \log(1 - \bar{y}_i)$$

$$\log(1 - \bar{y}_i) = \log\left(1 - \frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}}\right) = \log\left(\frac{1}{1+e^{\log(\text{odds})}}\right) = \log(1) - \log(1 + e^{\log(\text{odds})}) = -\log(1 + e^{\log(\text{odds})})$$

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

$$\log\left(\frac{\overline{y}_i}{1-\overline{y}_i}\right)=\log(\text{odds})$$

Exponential both sides

$$\left(\frac{\overline{y}_i}{1-\overline{y}_i}\right)=e^{\log(\text{odds})}$$

$$\overline{y}_i=(1-\overline{y}_i)e^{\log(\text{odds})}$$

Add $\overline{y}_i e^{\log(\text{odds})}$ both sides

$$\overline{y}_i = e^{\log(\text{odds})} - \overline{y}_i e^{\log(\text{odds})}$$

$$\overline{y}_i + \overline{y}_i e^{\log(\text{odds})} = e^{\log(\text{odds})}$$

$$\overline{y}_i(1+e^{\log(\text{odds})}) = e^{\log(\text{odds})}$$

$$\overline{y}_i=\frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}}$$