XGBoost

Regression

XGBoost is “extreme” gradient boost, this means that it contains so many components

- Gradient Boost
- Regularization
- A unique Regression tree
- Approximate Greedy Algorithm
- Weighted Quantile Sketch
- Sparsity-Aware Split Finding
- Parallel Learning
- Cache-Aware Access
- Blocks for Out-of-Core Computation

XGBoost is “extreme” gradient boost, this means that it contains so many components

- Gradient Boost
- Regularization
- A unique Regression tree
- Approximate Greedy Algorithm
- Weighted Quantile Sketch
- Sparsity-Aware Split Finding
- Parallel Learning
- Cache-Aware Access
- Blocks for Out-of-Core Computation

Luckily, each component is fairly simple

XGBoost is “extreme” gradient boost, this means that it contains so many components

- Gradient Boost
- Regularization

We’ve covered these two in previous tutorials

- A unique Regression tree

So let’s start from here

- Approximate Greedy Algorithm
- Weighted Quantile Sketch
- Sparsity-Aware Split Finding
- Parallel Learning
- Cache-Aware Access
- Blocks for Out-of-Core Computation

Luckily, each component is fairly simple

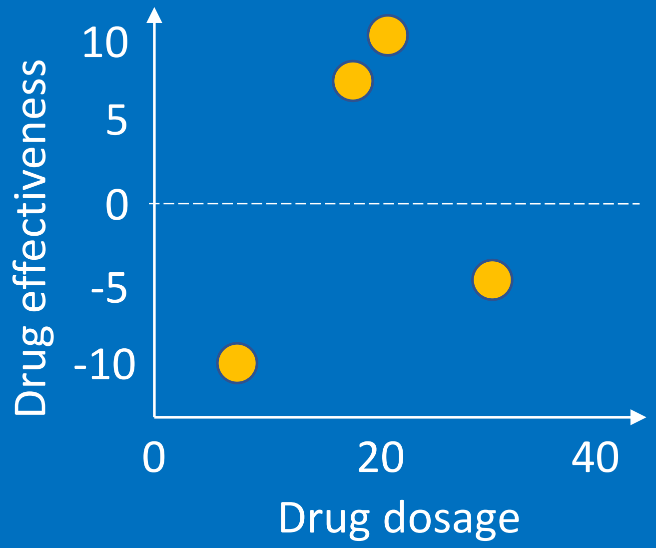
Drug dosage	Drug effectiveness
9	-10
20	7
24	8
36	-7.5

Let’s assume that dataset to be used

Drug dosage	Drug effectiveness
9	-10
20	7
24	8
36	-7.5

Let's assume that dataset to be used

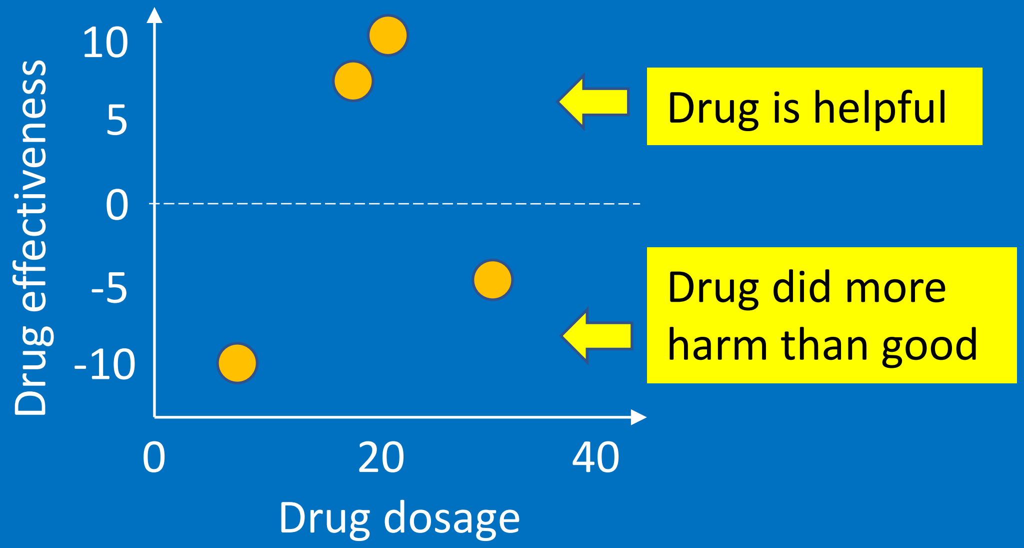
Let's plot it out



Drug dosage	Drug effectiveness
9	-10
20	7
24	8
36	-7.5

Let's assume that dataset to be used

Let's plot it out

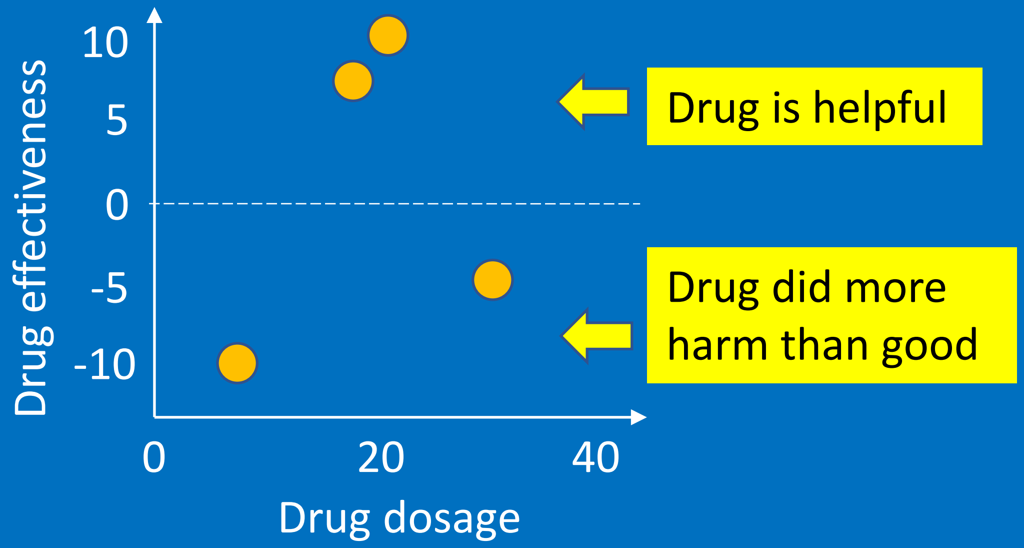


Step 1: make an initial prediction

Drug dosage	Drug effectiveness
9	-10
20	7
24	8
36	-7.5

Let's assume that dataset to be used

Let's plot it out

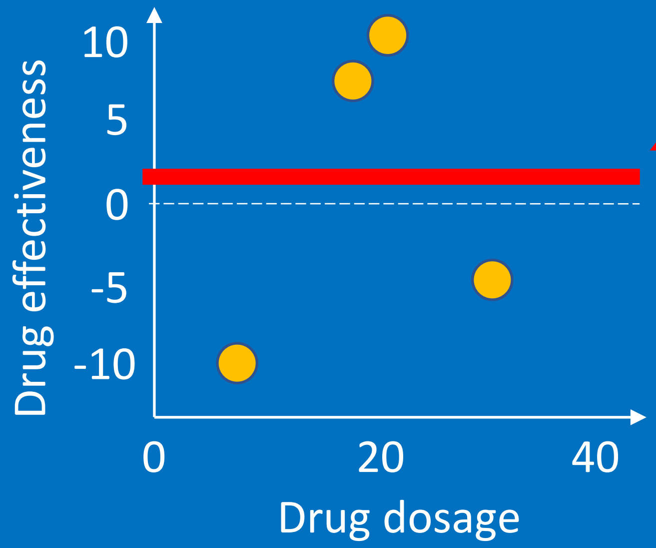


Step 1: make an initial prediction

Drug dosage	Drug effectiveness
9	-10
20	7
24	8
36	-7.5

Let's assume that dataset to be used

Let's plot it out



0.5

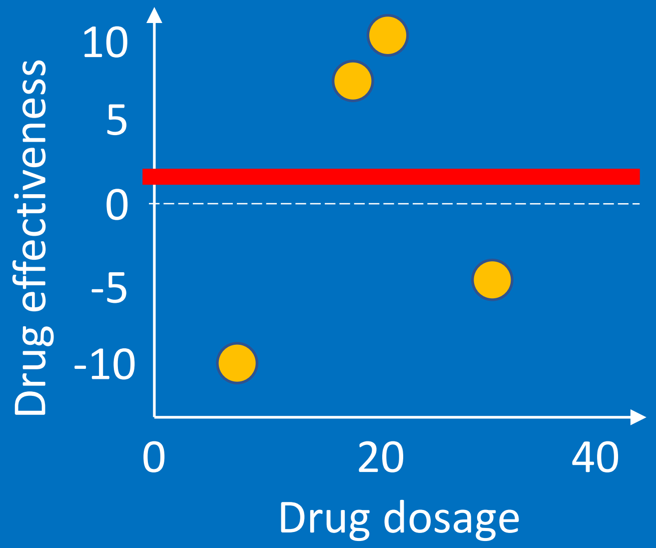
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

The prediction, 0.5, corresponds to this thick red horizontal line

Drug dosage	Drug effectiveness
9	-10
20	7
24	8
36	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

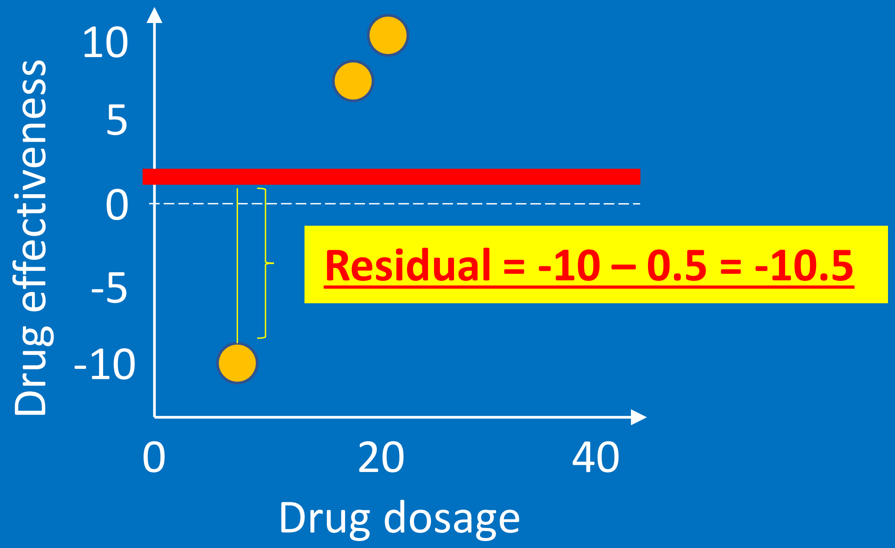
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Drug dosage	Drug effectiveness
9	-10
20	7
24	8
36	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

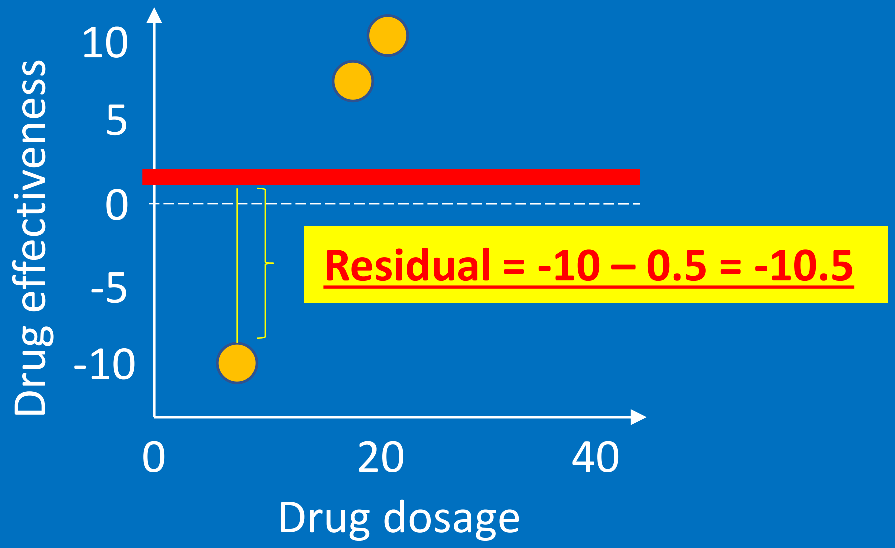
Step 2: Obtain the residuals

For example, when the dosage is "9", the prediction is "0.5", while the actual drug effectiveness is "-10". So the residual is **-10 – 0.5 = -10.5**

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

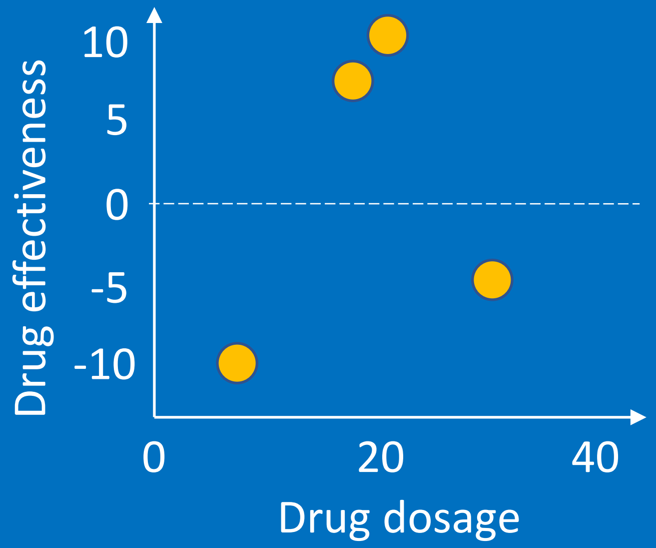
For example, when the dosage is "9", the prediction is "0.5", while the actual drug effectiveness is "-10". So the residual is **-10 – 0.5 = -10.5**

We do this for all the samples, and use the residuals as the target for growing trees (like gradient boost)

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let’s assume that dataset to be used

Let’s plot it out



Step 1: make an initial prediction

0.5

Let’s assume that the “initial guess” of “predicted drug effectiveness” is 0.5 (so for whatever testing data, the prediction is always 0.5)

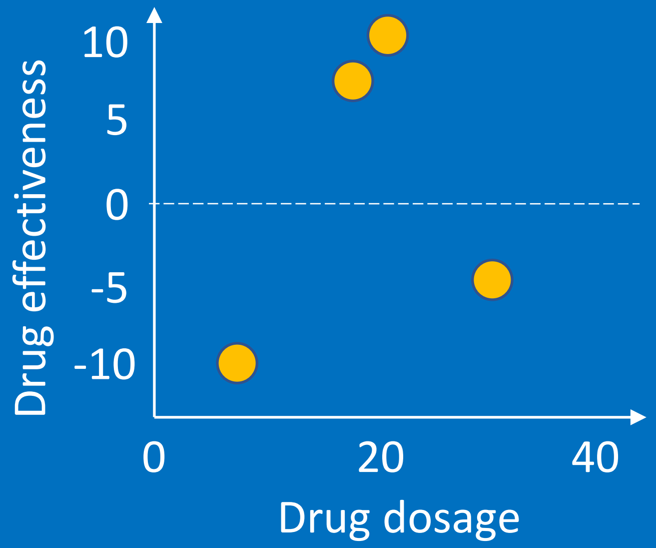
Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

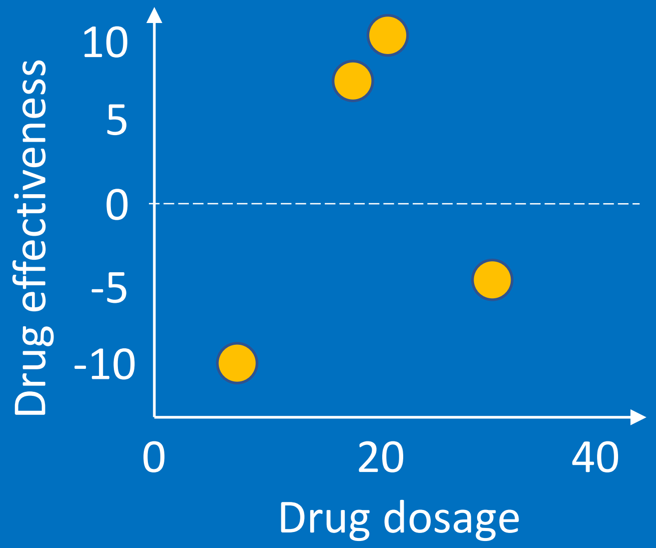
3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

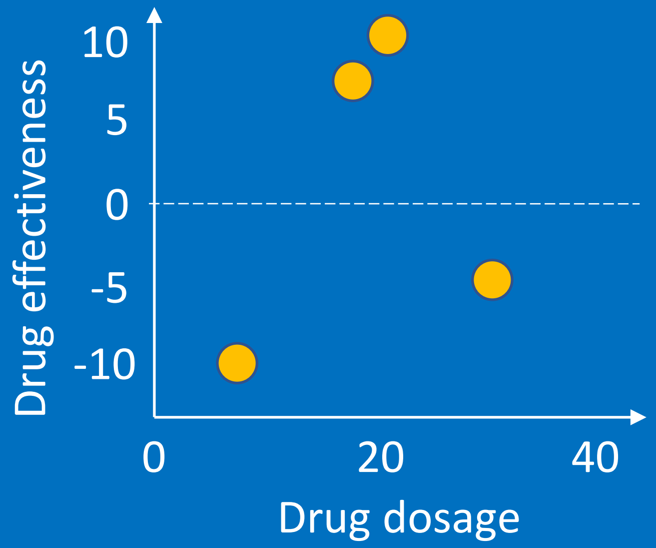
-10.5,6.5,7.5,-7.5

3.2: Calculate a "quality score" (or "similarity score") for the residuals

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5

3.2: Calculate a "quality score" (or "similarity score") for the residuals

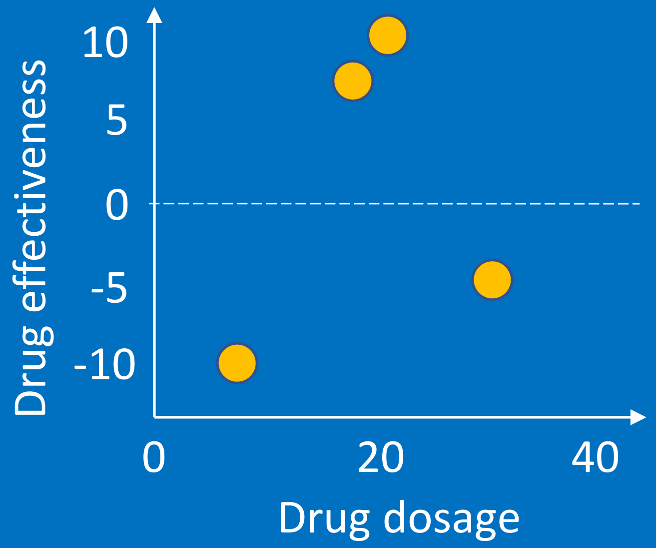
$$Similarity\ score = \frac{(\sum residuals)^2}{number\ of\ residuals + \lambda}$$

A regularization parameter

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5

3.2: Calculate a "quality score" (or "similarity score") for the residuals

$$Similarity\ score = \frac{(\sum residuals)^2}{number\ of\ residuals + \lambda}$$

By assuming the regularization parameter $\lambda = 0$

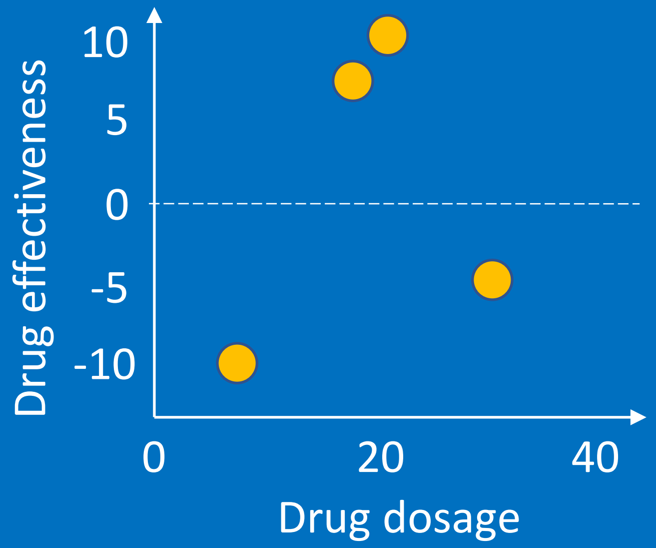
So for this case, we have

$$Similarity\ score = \frac{(-10.5+6.5+7.5-7.5)^2}{4+0} = 4$$

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

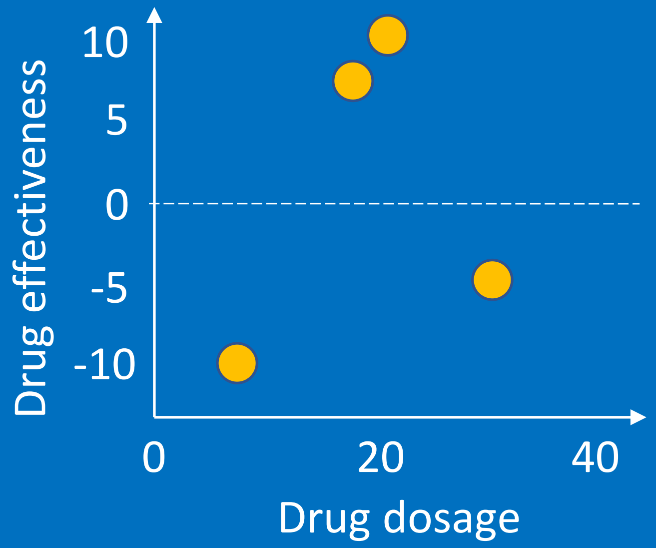
-10.5,6.5,7.5,-7.5

4 *Similarity score*

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

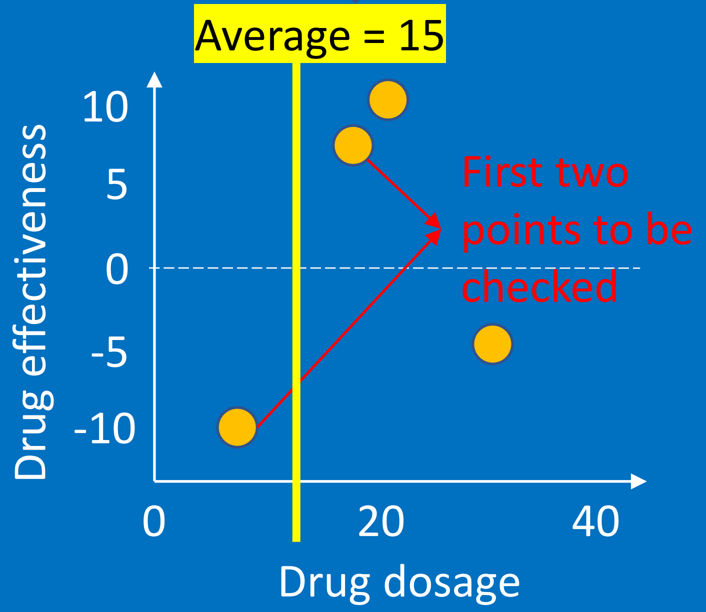
-10.5,6.5,7.5,-7.5 4 *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 | 4 *Similarity score*

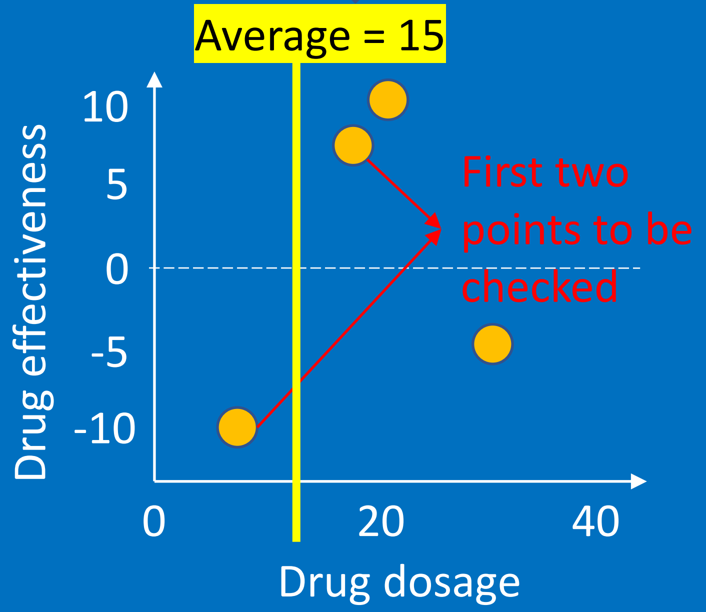
3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

To answer this, we first look at the first two points, there average is 15

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

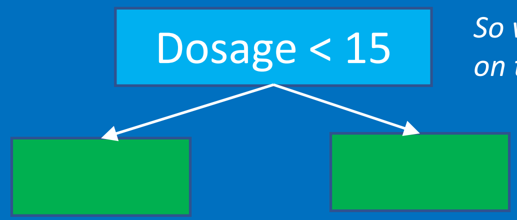
Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 4 *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

To answer this, we first look at the first two points, there average is 15

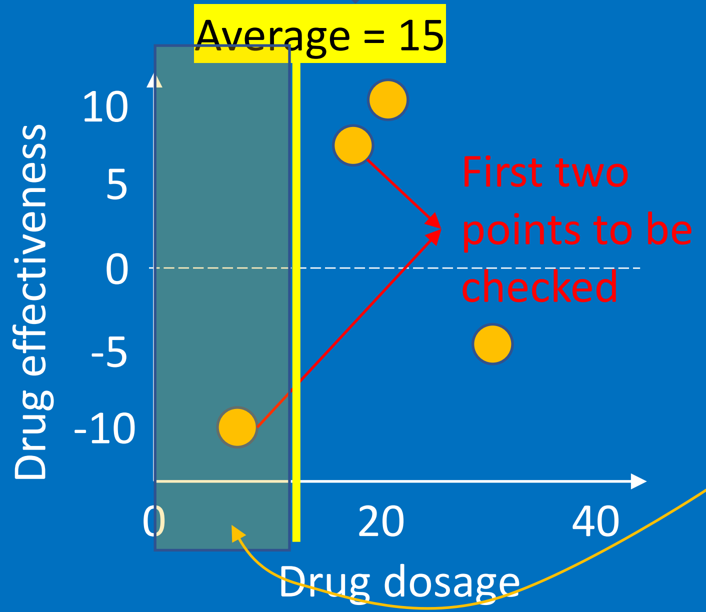


So we build a tree based on the dosage < 15

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



All the residuals go to the leaf on the left

Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 | 4 *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

To answer this, we first look at the first two points, there average is 15

Dosage < 15

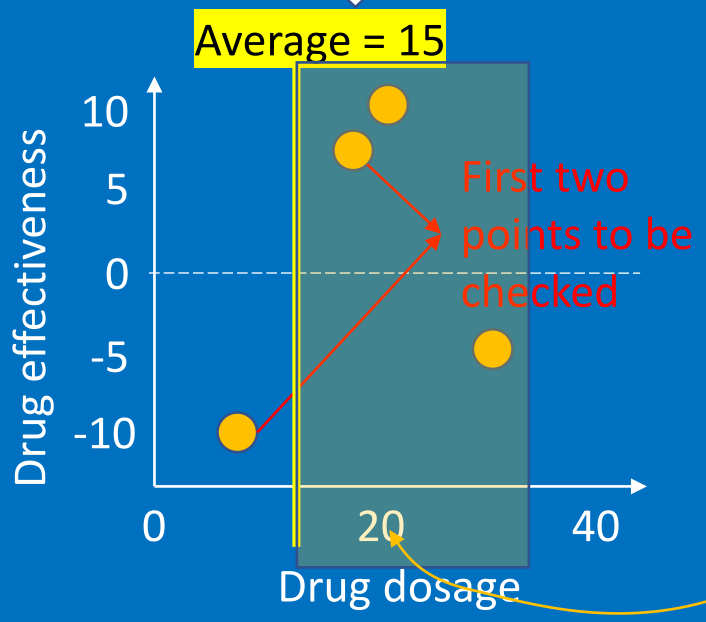
-10.5

So we build a tree based on the dosage < 15

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



All the residuals go to the leaf on the right

Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

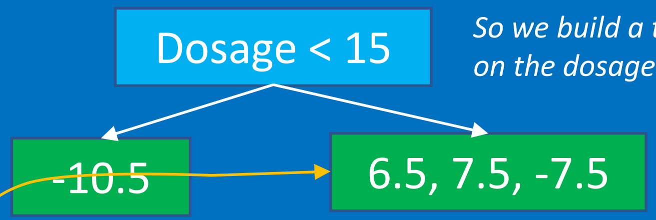
Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 4 *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

To answer this, we first look at the first two points, there average is 15

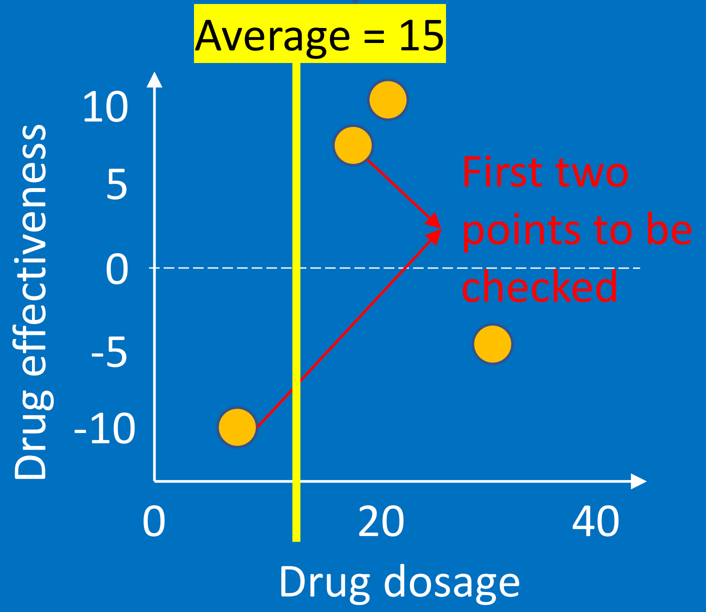


So we build a tree based on the dosage < 15

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

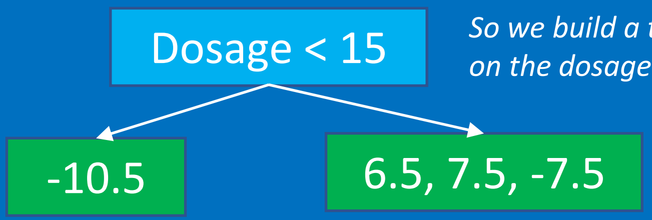
Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 4 *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

To answer this, we first look at the first two points, there average is 15



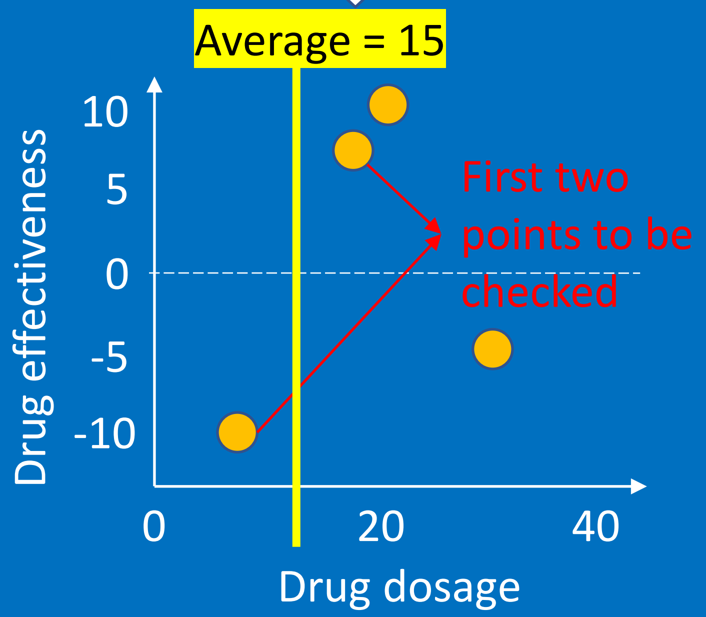
So we build a tree based on the dosage < 15

Similarly, we can calculate the "similarity score" for the leafs

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 | 4 | *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

To answer this, we first look at the first two points, there average is 15

Dosage < 15

-10.5 | 110.25

6.5, 7.5, -7.5 | 14.08

So we build a tree b... on the dosage < 15

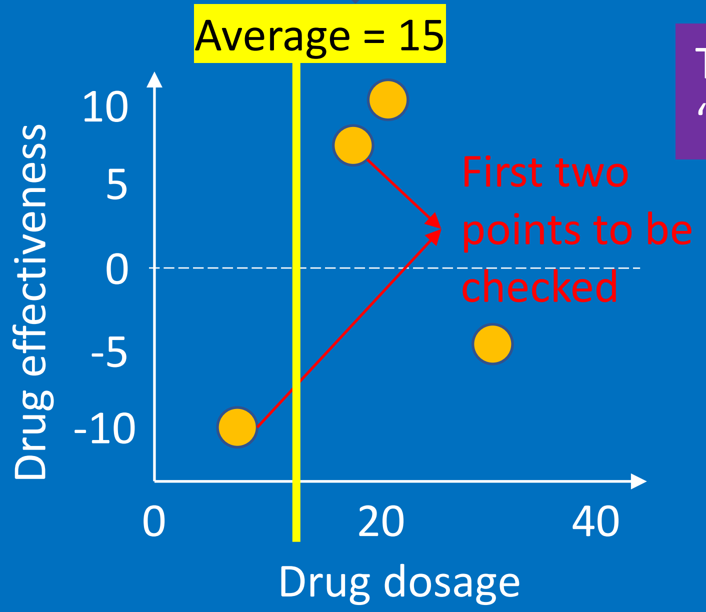
$$\text{Similarity score} = \frac{(\sum residuals)^2}{\text{number of residuals} + \lambda}$$

Similarly, we can calculate the "similarity score" for the leafs

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 | 4 | *Similarity score*

This is called "Root" Node

3.2: whether or not we can do a better job clustering similar Residuals if we split them further

To answer this, we first look at the first two points, there average is 15

These are called "Leaf" Node

Dosage < 15

So we build a tree based on the dosage < 15

-10.5

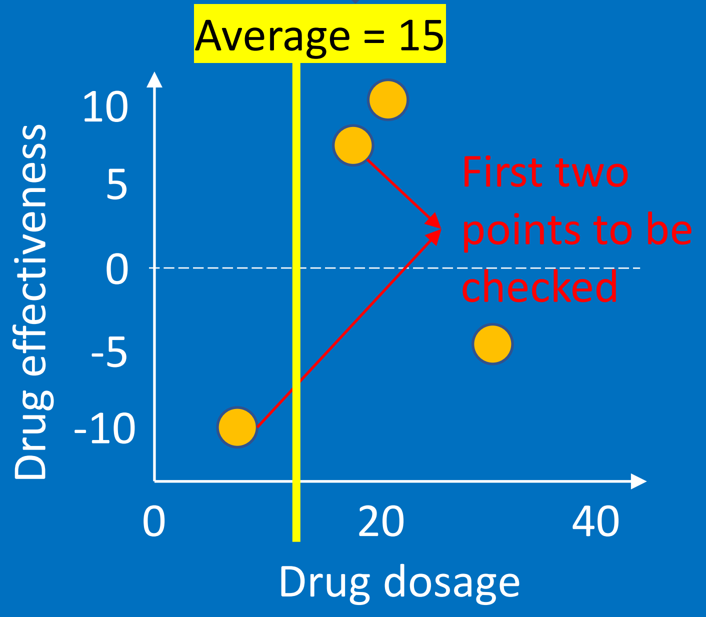
6.5, 7.5, -7.5 | 14.08

Similarly, we can calculate the "similarity score" for the leafs

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

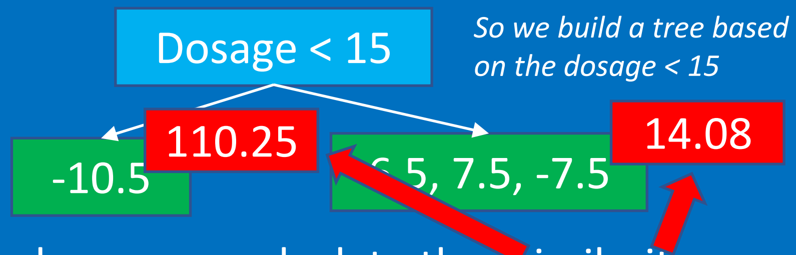
Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 **4** *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

To answer this, we first look at the first two points, there average is 15



So we build a tree based on the dosage < 15

Similarly, we can calculate the "similarity score" for the leafs

Calculate the "Gain" from this particular split

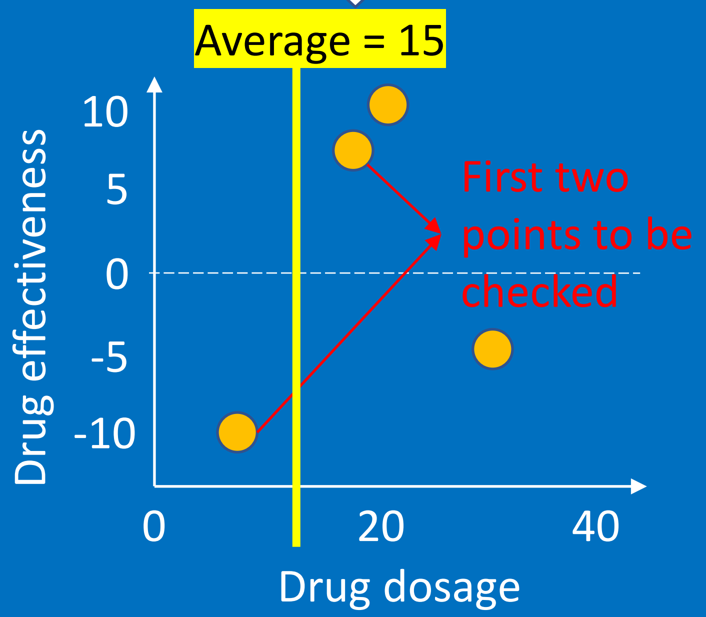
$$Gain = Leaf_{left,similarity} + Leaf_{right,similarity} - Root_{similarity}$$

110.25 14.08 4

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 | 4 *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

To answer this, we first look at the first two points, there average is 15

Dosage < 15

So we build a tree based on the dosage < 15

-10.5 | 110.25

6.5, 7.5, -7.5 | 14.08

Similarly, we can calculate the "similarity score" for the leafs

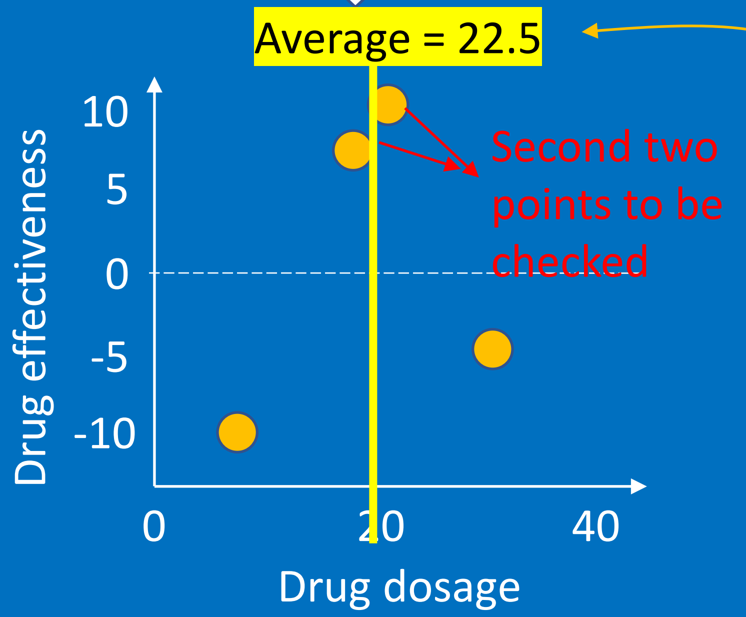
Calculate the "Gain" from this particular split

$$Gain = 120.33$$

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 4 *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

Dosage < 15

-10.5

6.5, 7.5, -7.5

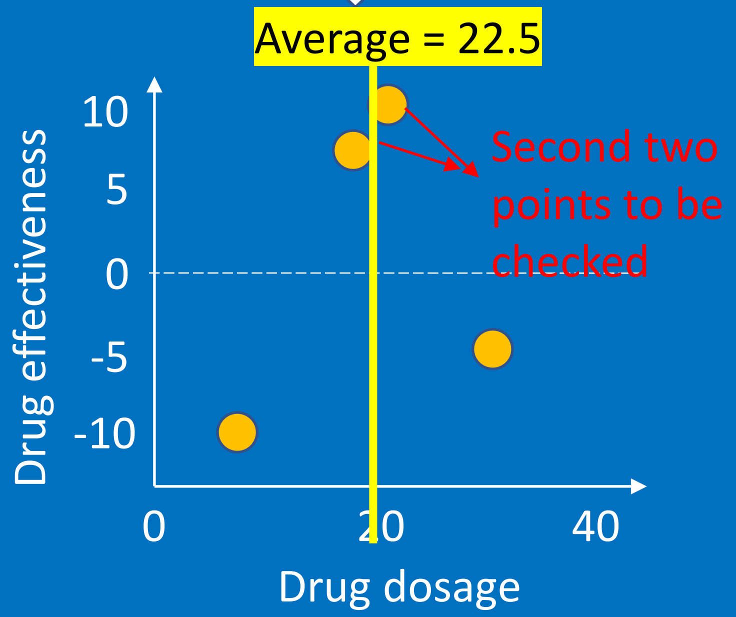
$Gain = 120.33$

Then let's move to the next point sets

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 | 4 *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

Dosage < 15

-10.5 | 6.5, 7.5, -7.5

$Gain = 120.33$

Dosage < 22.5

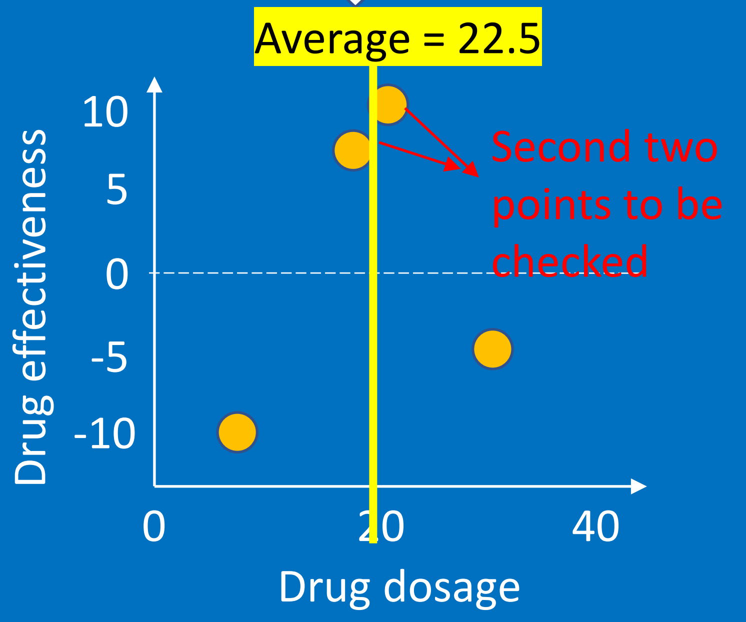
-10.5, 6.5 | 7.5, -7.5

$Gain = 4$

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 | 4 | *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

Dosage < 15

-10.5 | 6.5, 7.5, -7.5

$Gain = 120.33$

Dosage < 22.5

-10.5, 6.5 | 7.5, -7.5

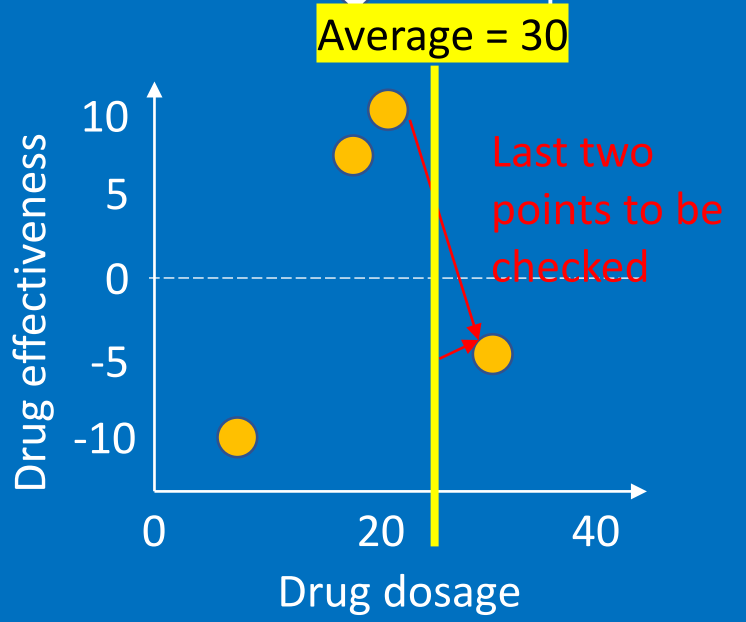
$Gain = 4$

Since Gain=120.33 is larger than Gain=4, so "Dosage < 15" is better at splitting the Residuals into clusters of similar values

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 4 *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

Dosage < 15

-10.5 | 6.5, 7.5, -7.5

$Gain = 120.33$

Dosage < 22.5

-10.5, 6.5 | 7.5, -7.5

$Gain = 4$

Dosage < 30

-10.5, 6.5,7.5 | -7.5

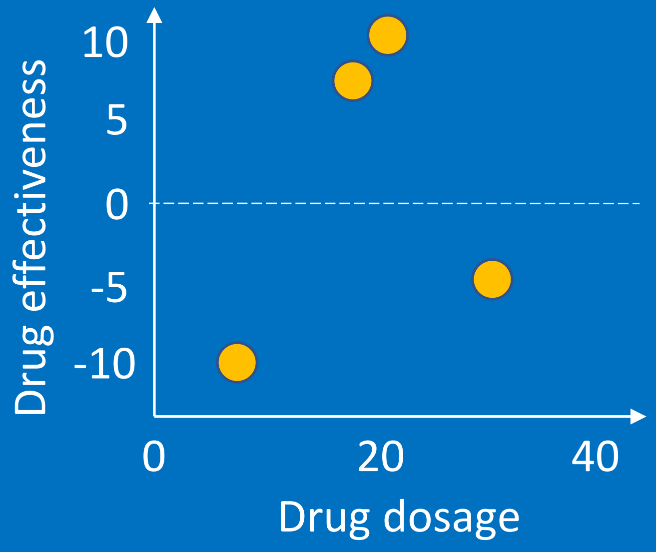
$Gain = 56.33$

Similarly, we can get the gain for the last two data points

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 | 4 *Similarity score*

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

Dosage < 15

-10.5 | 6.5, 7.5, -7.5

$Gain = 120.33$

Dosage < 22.5

-10.5, 6.5 | 7.5, -7.5

$Gain = 4$

Dosage < 30

-10.5, 6.5,7.5 | -7.5

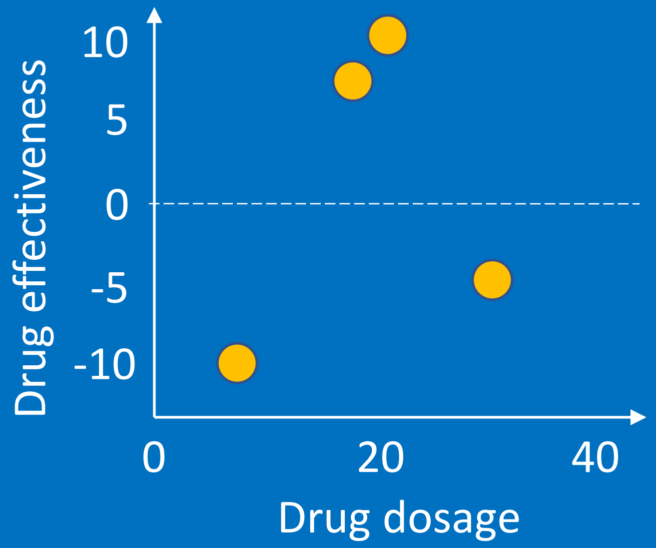
$Gain = 56.33$

Therefore, "Dosage < 15" is better at splitting the observations into clusters with similar values

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the resid

-10.5,6.5,7.5,-7.5 | 4 | *Similarity score*

For the original "root", Gain = Similarity score

3.2: Whether or not we can do a better job clus

$$Gain = Leaf_{left,similarity} + Leaf_{right,similarity} - Root_{similarity}$$

Dosage < 15

-10.5 | 6.5, 7.5, -7.5

$Gain = 120.33$

Dosage < 22.5

-10.5, 6.5 | 7.5, -7.5

$Gain = 4$

Dosage < 30

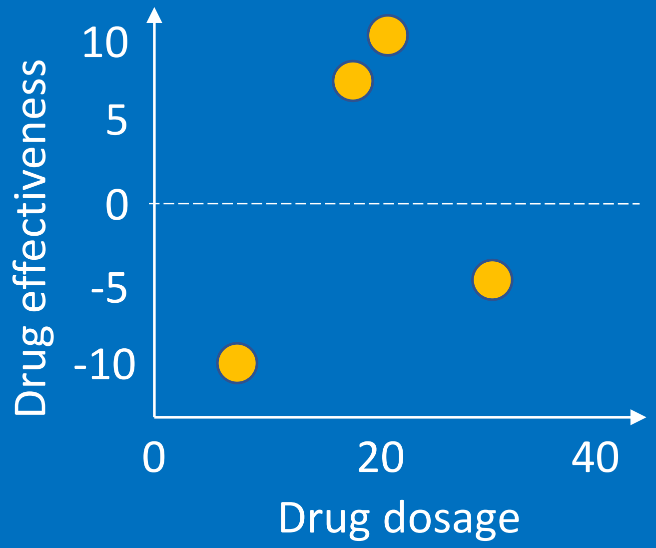
-10.5, 6.5,7.5 | -7.5

$Gain = 56.33$

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

3.1: Start from a single leaf, and all of the residuals go to the leaf

-10.5,6.5,7.5,-7.5 $Gain = 4$

3.2: Whether or not we can do a better job clustering similar Residuals if we split them further

Dosage < 15

-10.5 | 6.5, 7.5, -7.5

$Gain = 120.33$

Dosage < 22.5

-10.5, 6.5 | 7.5, -7.5

$Gain = 4$

Dosage < 30

-10.5, 6.5,7.5 | -7.5

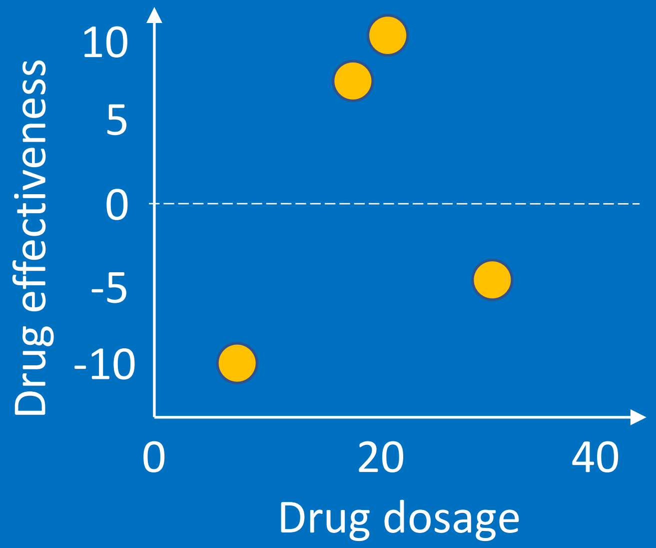
$Gain = 56.33$

So for all of the available trees, Dosage < 15 has the largest Gain, therefore it will be used as the first tree

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Dosage < 15

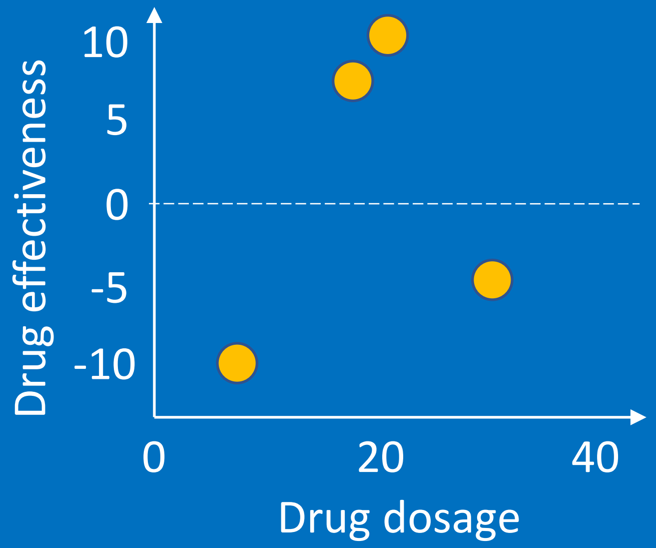
-10.5

6.5, 7.5, -7.5

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



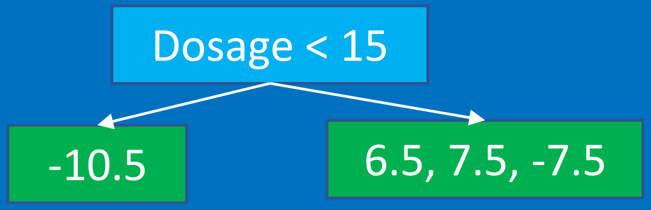
Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

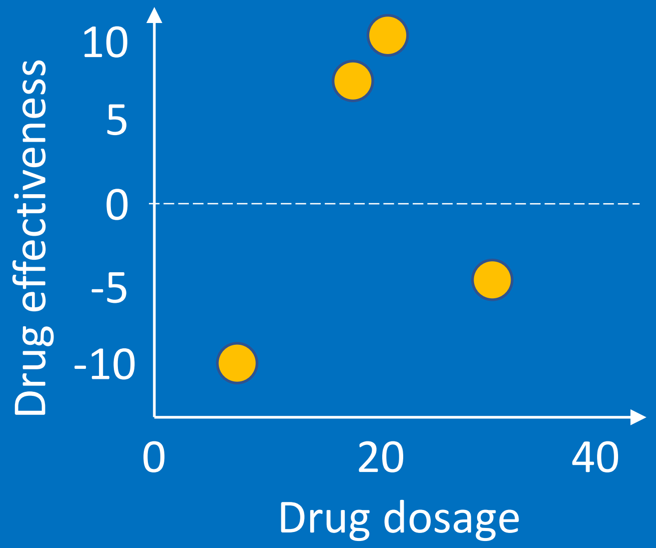


There is only one residual on the left, we can't split any further

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



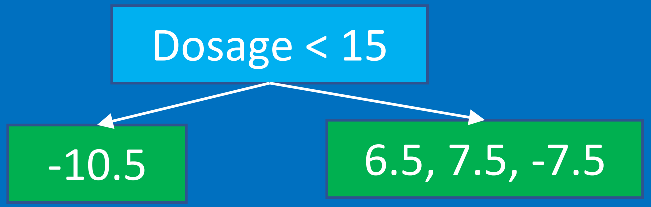
Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

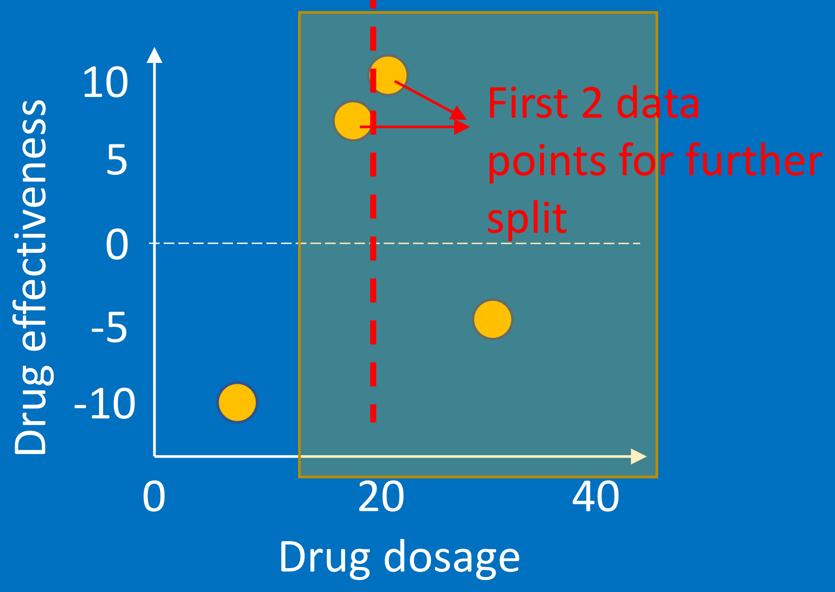


Three residuals on the right, so we can further split it

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Dosage < 15

-10.5

6.5, 7.5, -7.5

Three residuals on the right, so we can further split it

Using the first 2 data points on the right side

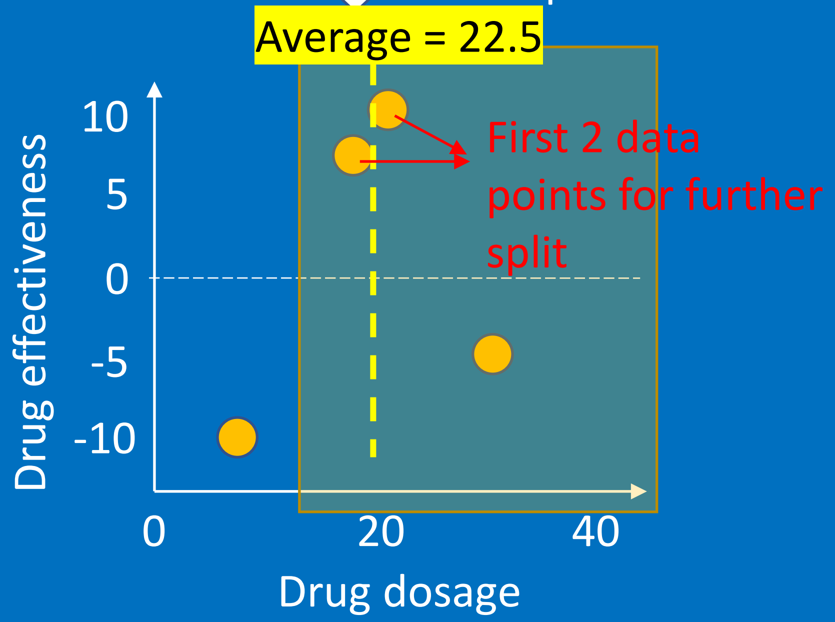
Note that this leaf is just like the original leaf we get from the Step 2, but with less elements

-10.5,6.5,7.5,-7.5

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



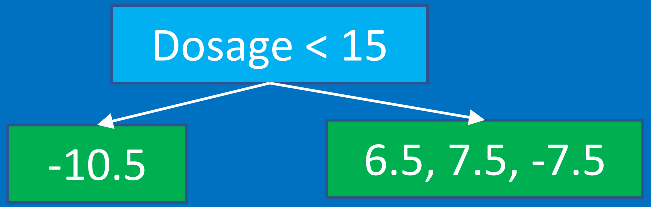
Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

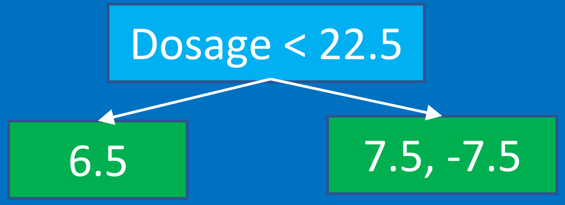
Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree



Three residuals on the right, so we can further split it

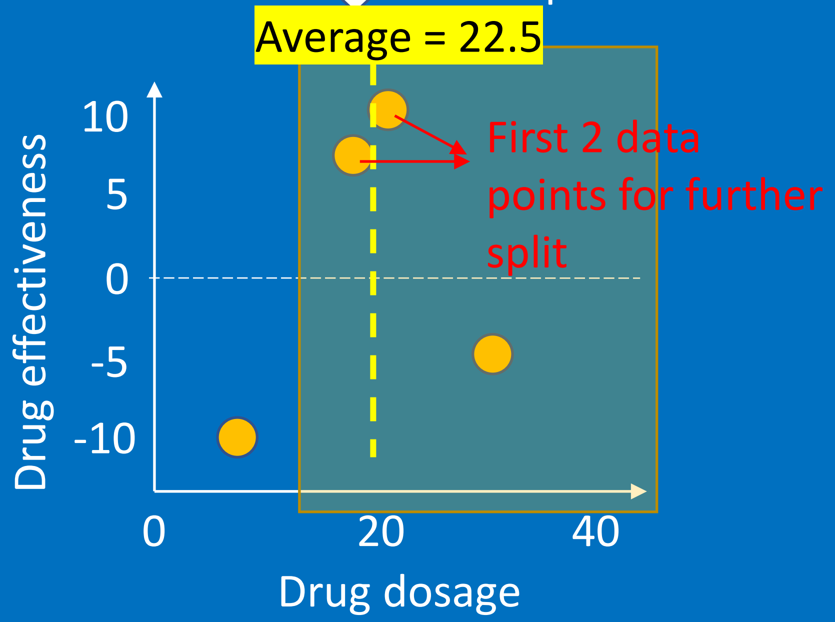
Using the first 2 data points on the right side



Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



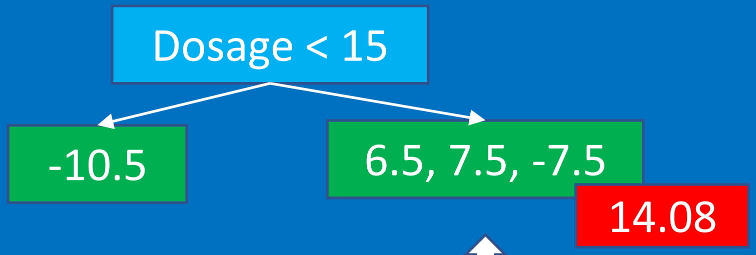
Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

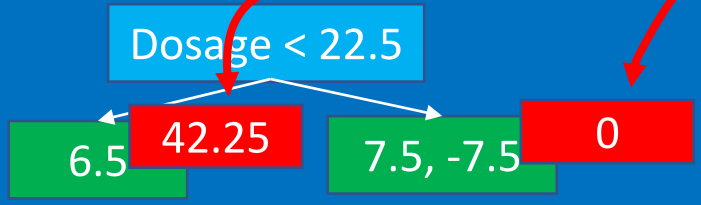
Step 3: Grow a XGBoost tree



Three residuals on the right, so we can further split it

Similarity Score

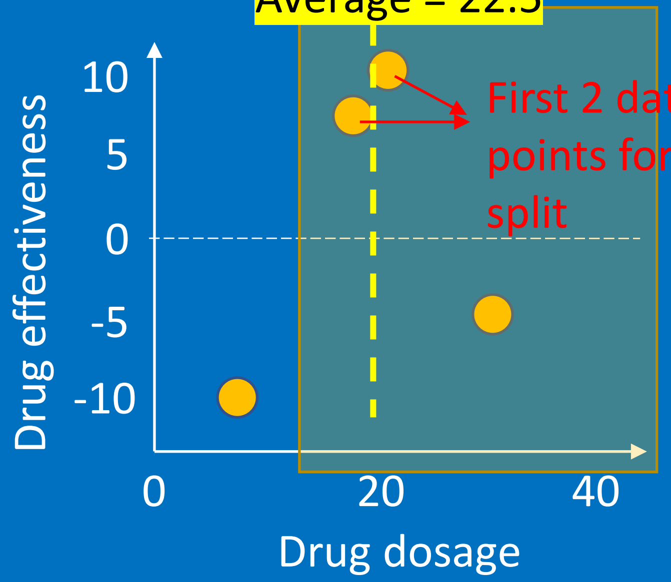
Using the first 2 data points on the right side



Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Dosage < 15

-10.5

6.5, 7.5, -7.5

14.08

Three residuals on the right, so we can further split it

Similarity Score

Using the first 2 data points on the right side

Dosage < 22.5

6.5

42.25

7.5, -7.5

0

$$Gain = Leaf_{left,similarity} + Leaf_{right,similarity} - Root_{similarity} = 28.17$$

42.25

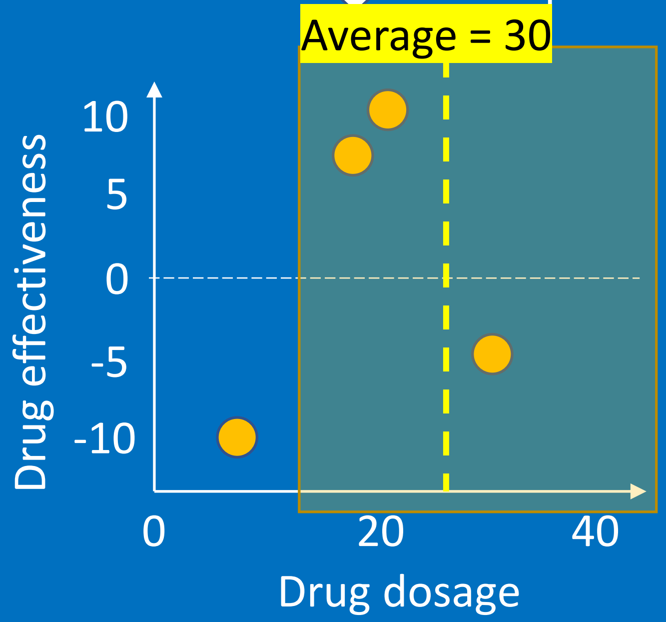
0

-14.08

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



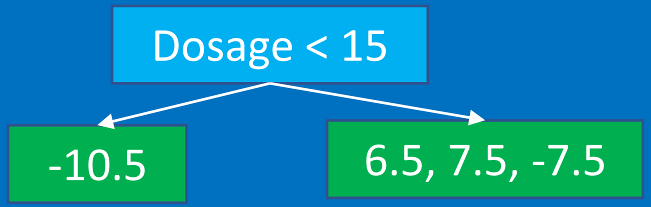
Step 1: make an initial prediction

0.5

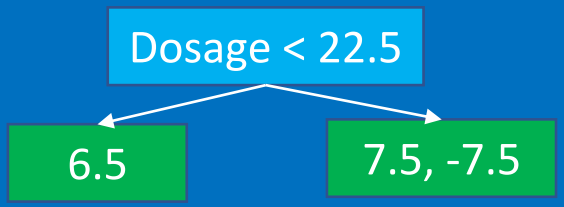
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

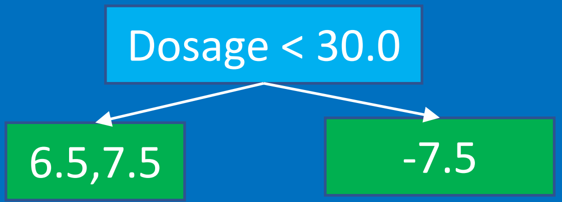
Step 3: Grow a XGBoost tree



Then we move to next points, we get



$Gain = 28.17$

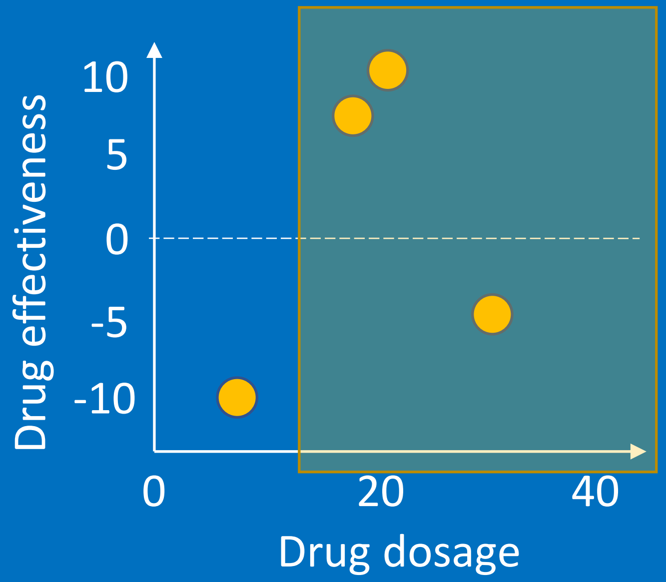


$Gain = 140.17$

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



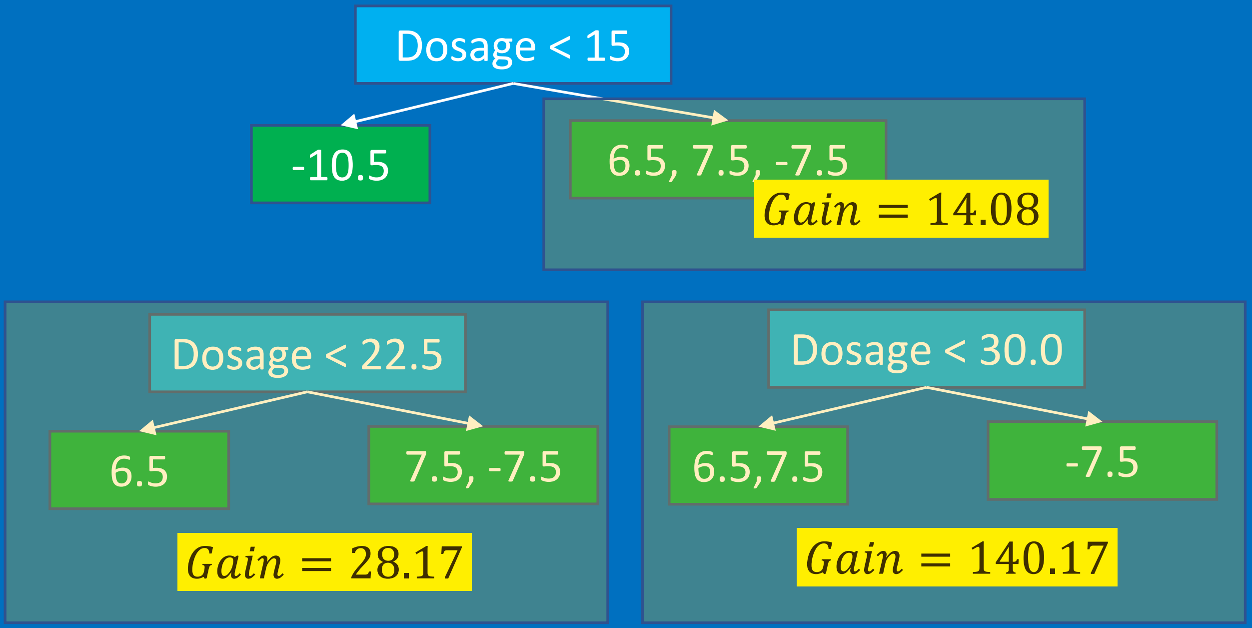
Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

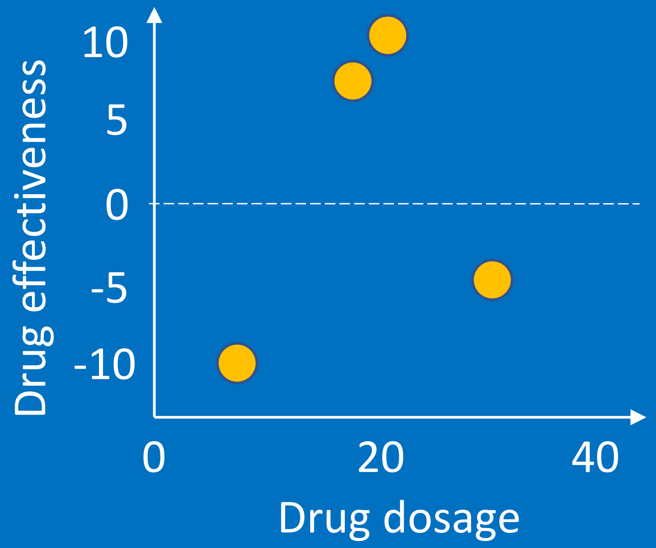


So for all of the available trees, Dosage < 30.0 has the largest Gain, therefore it will be used for the further split

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



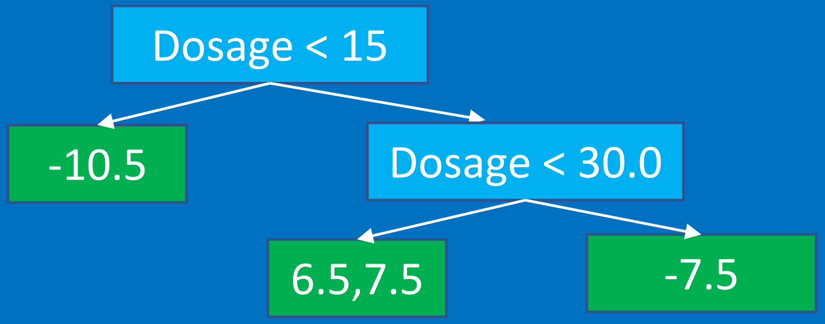
Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

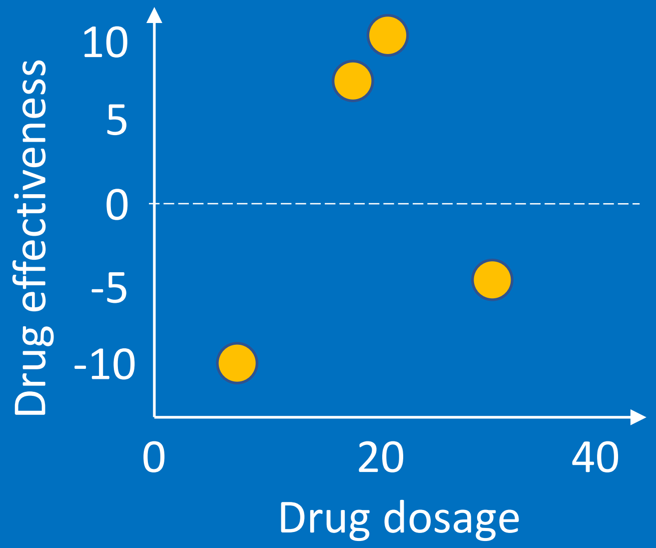
Step 3: Grow a XGBoost tree



Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



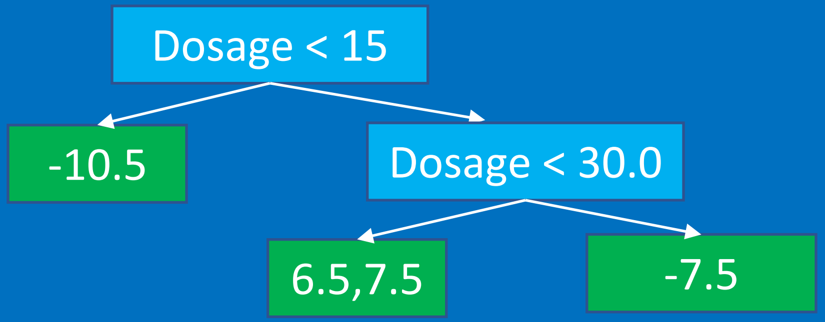
Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

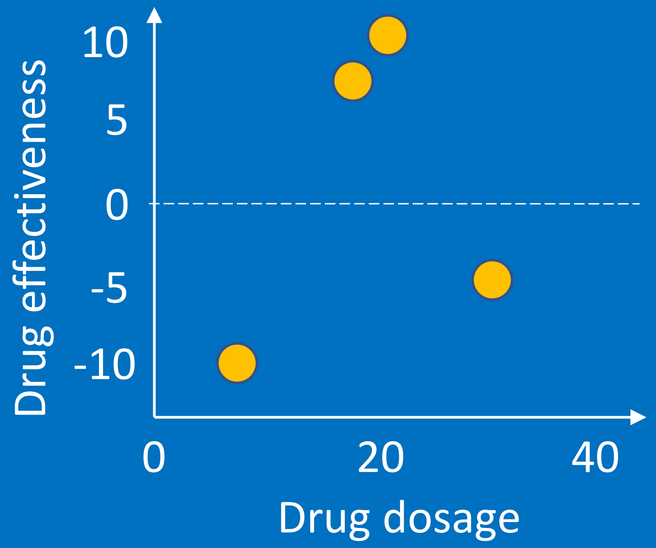


We can continue the further split further if we want …

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



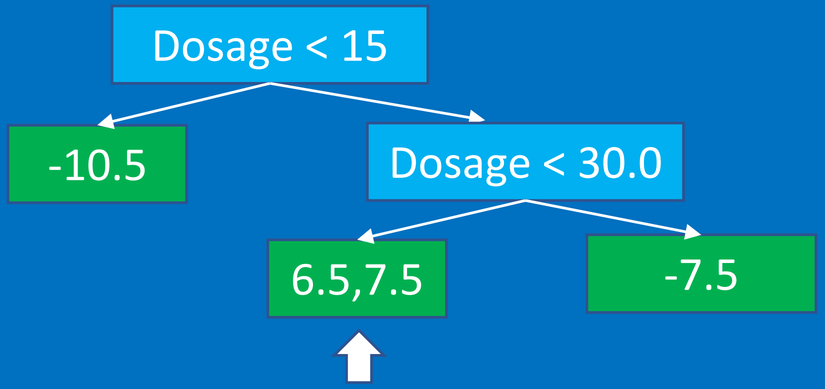
Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

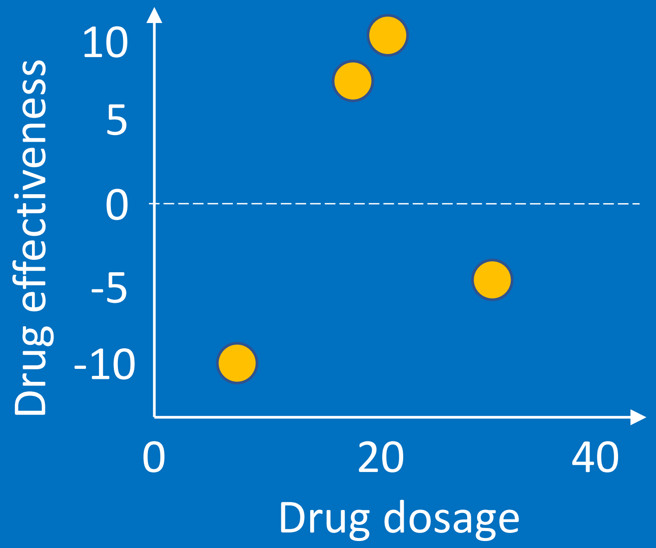


For now, we won't split this leaf further, so we are done for building this tree

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

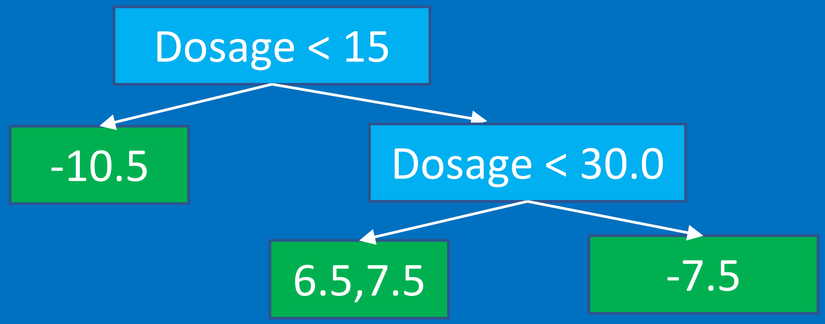
0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

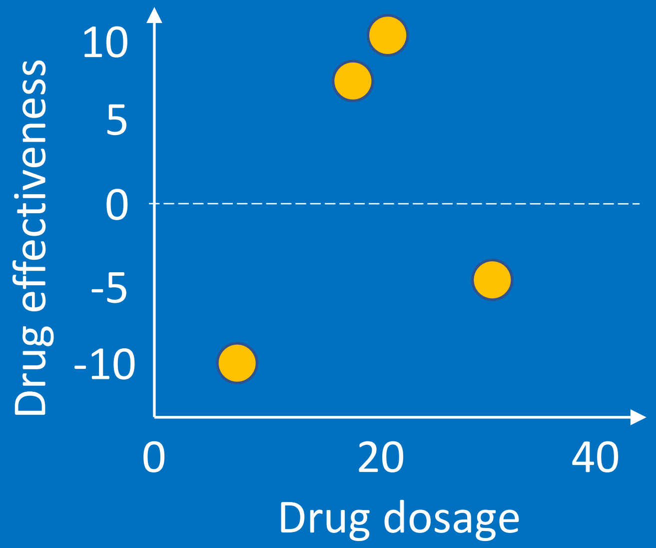


We prune the tree based on its Gain value

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

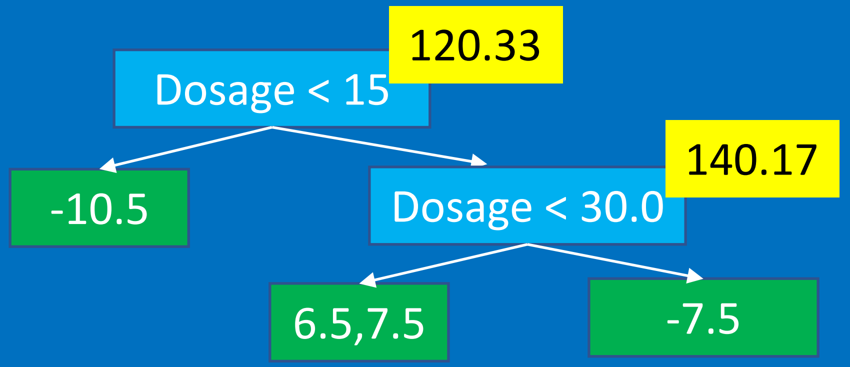
0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree



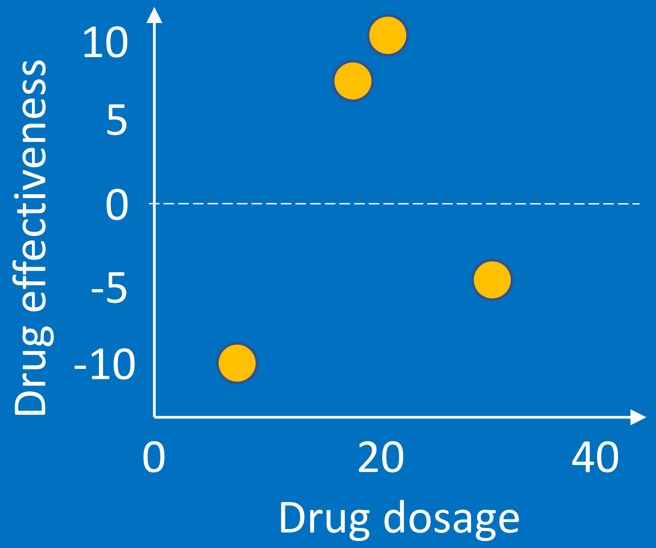
Previously calculated Gain values

We prune the tree based on its Gain value

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

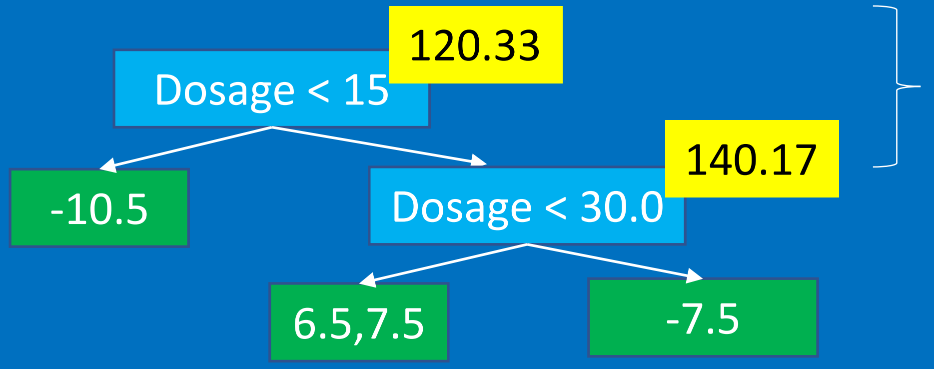
0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree



Previously calculated Gain values

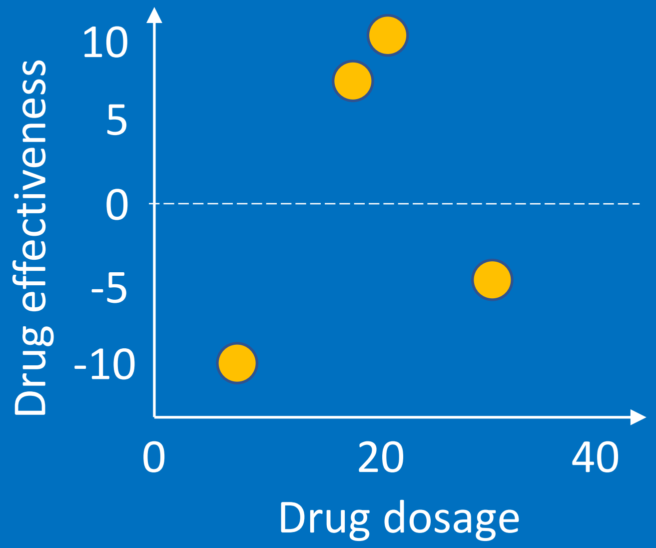
We prune the tree based on its Gain value

- We start by picking up a "random" number, γ (gamma) ~ in this case $\gamma = 130$

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

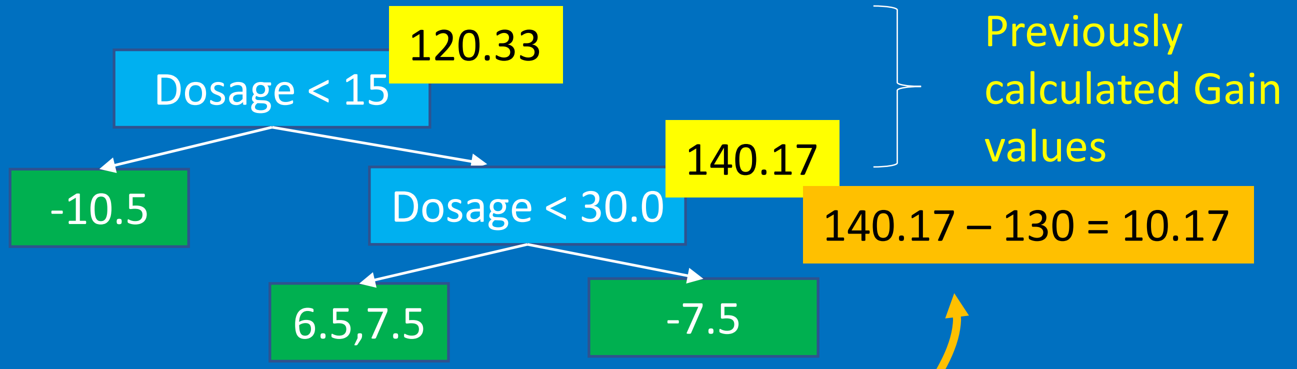
0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree



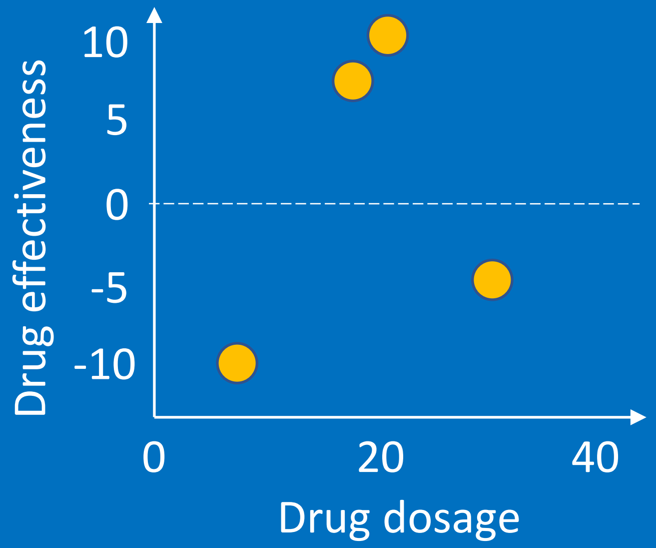
We prune the tree based on its Gain value

- We start by picking up a "random" number, γ (gamma) ~ in this case $\gamma = 130$
- Calculate "Gain - γ" from the lowest level of the tree

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

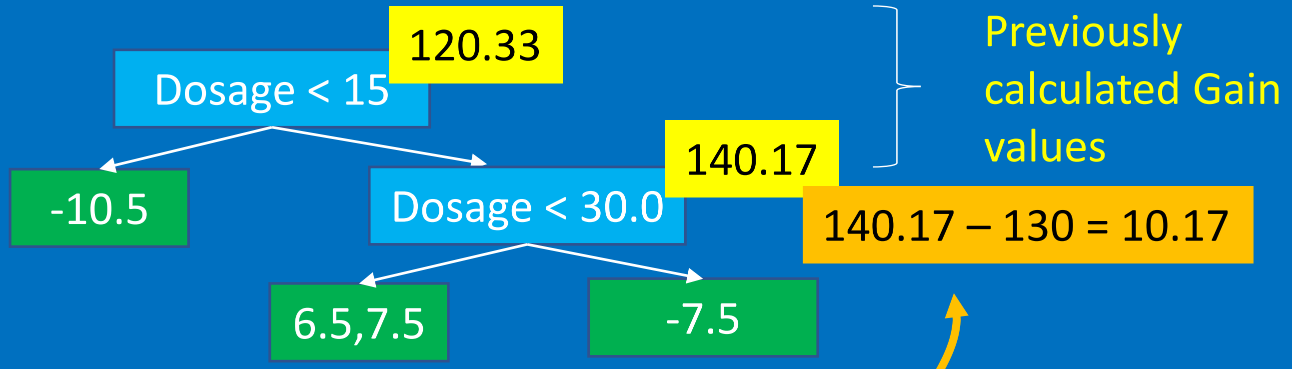
0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree



We prune the tree based on its Gain value

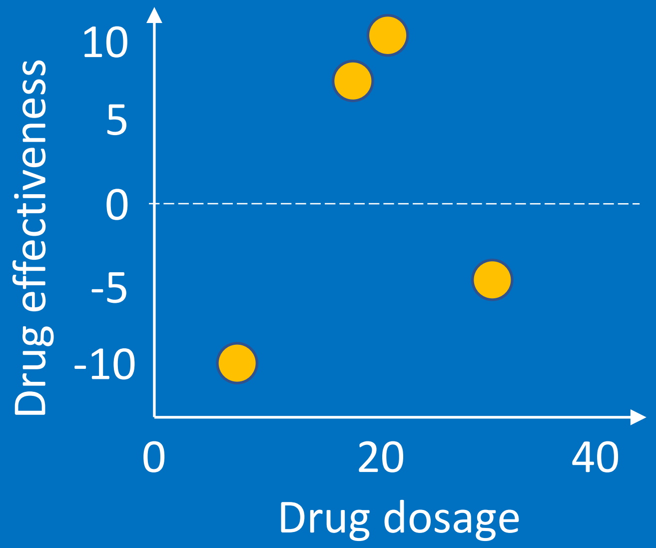
- We start by picking up a "random" number, γ (gamma) ~ in this case $\gamma = 130$
- Calculate "Gain - γ" from the lowest branch of the tree

If "Gain - γ" < 0: then the branch will be removed. Otherwise the branch pruning is done (so we don't continue …)

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

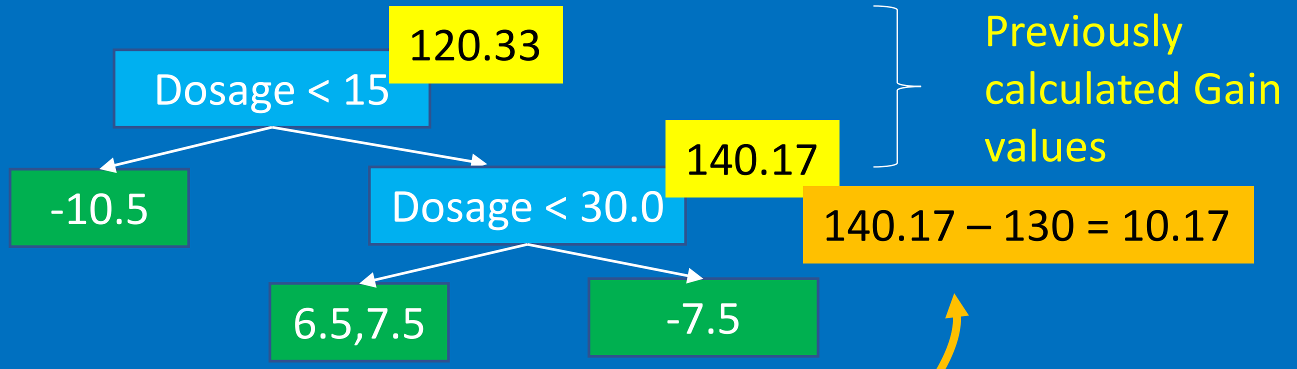
0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree



We prune the tree based on its Gain value

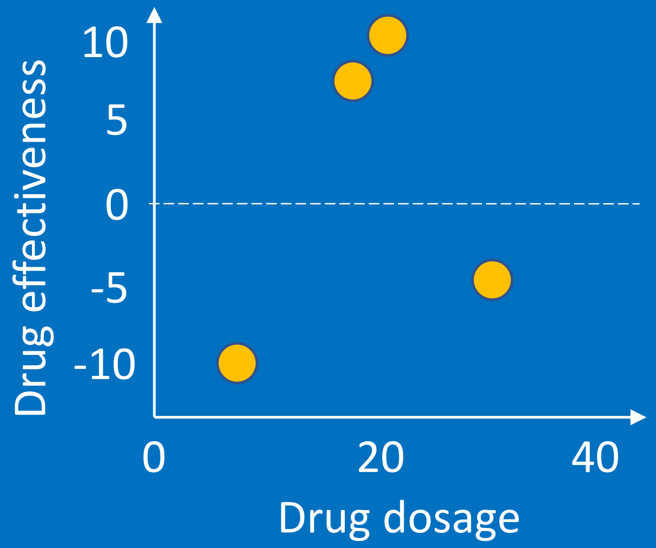
- We start by picking up a "random" number, γ (gamma) ~ in this case $\gamma = 130$
- Calculate "Gain - γ" from the lowest branch of the tree

  If "Gain - γ" < 0: then the branch will be removed. Otherwise the branch pruning is done (so we don't continue …)
- So in this case we don't have to prune the tree

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let’s assume that dataset to be used

Let’s plot it out



Step 1: make an initial prediction

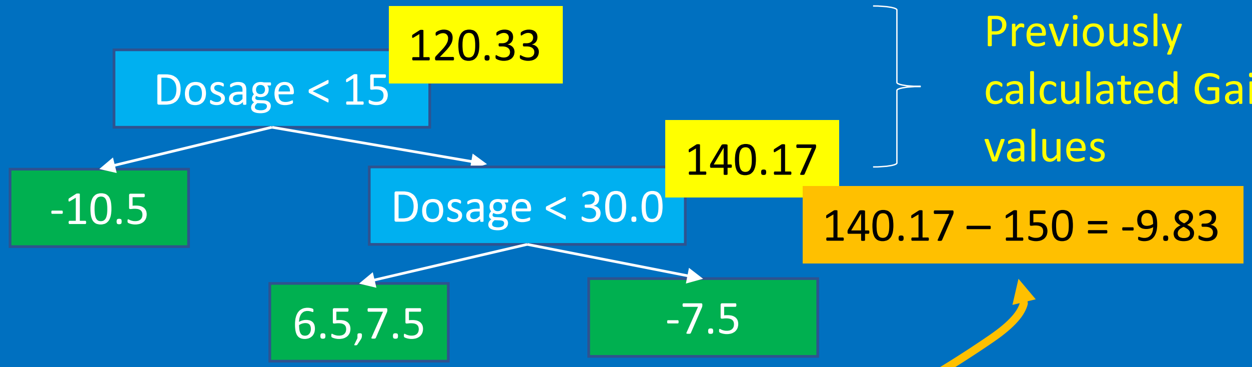
0.5

Let’s assume that the “initial guess” of “predicted drug effectiveness” is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree



We prune the tree based on its Gain value

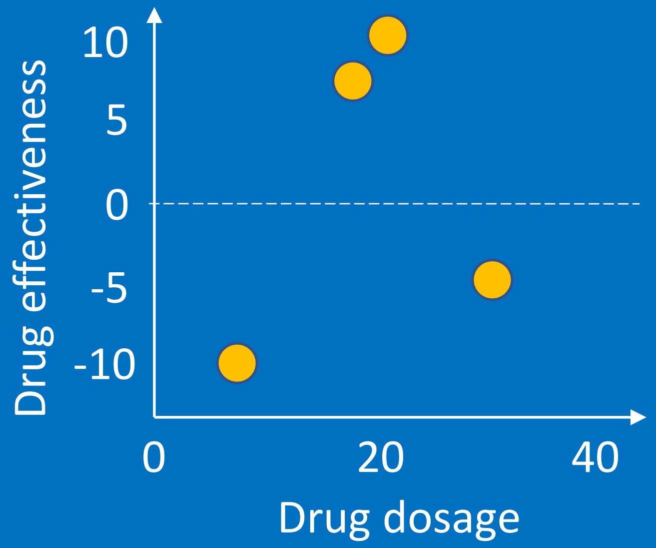
If we set $\gamma = 150$

Gain - γ = -9.83, so this branch will be removed

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

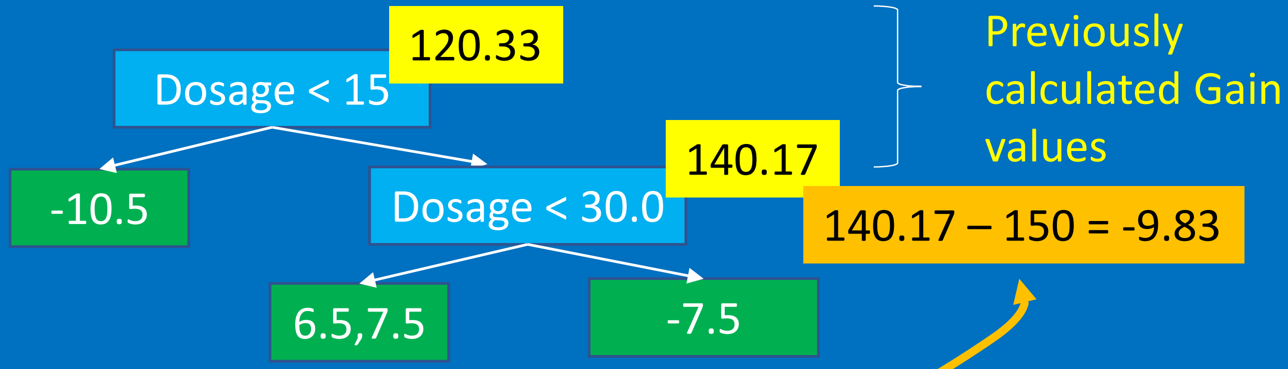
0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree



We prune the tree based on its Gain value

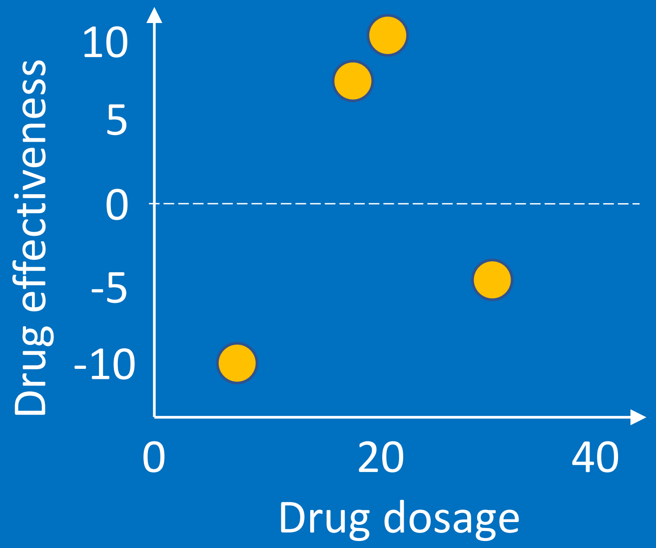
If we set $\gamma = 150$

Gain - γ = -9.83, so this branch will be removed

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

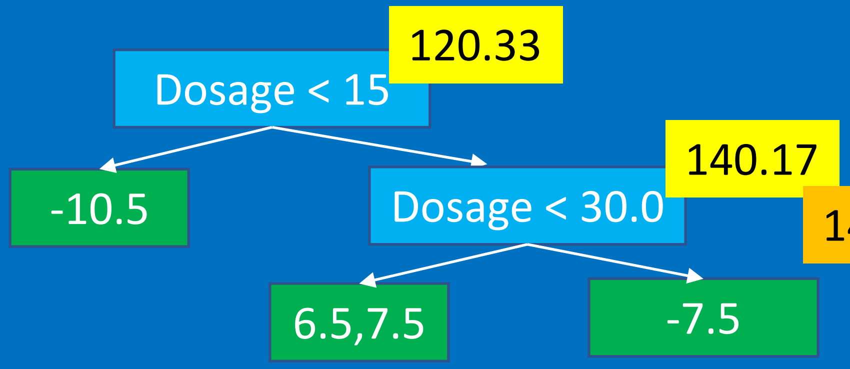
0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree



Previously calculated Gain values

140.17 – 150 = -9.83

We prune the tree based on its Gain value

If we set $\gamma = 150$

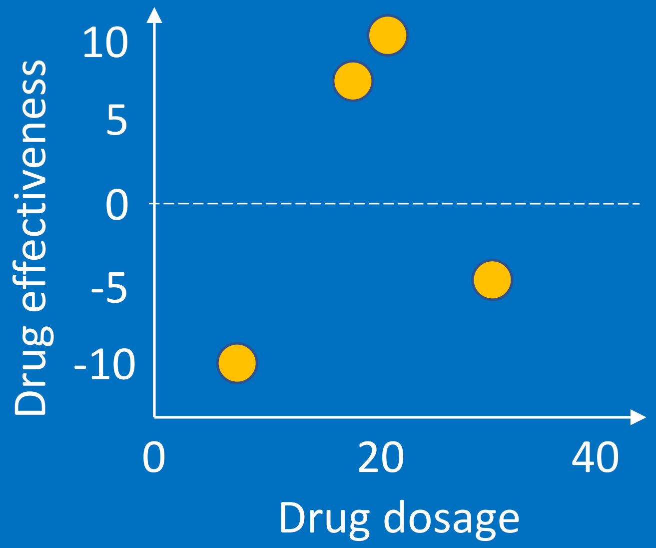
Gain - γ = -9.83, so this branch will be removed

Actually if we want to, we can continue to apply the same γ to this branch and see we want to keep it. But we are not doing it here

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



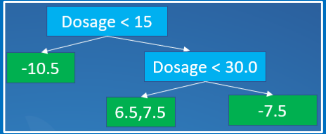
Step 1: make an initial prediction

0.5

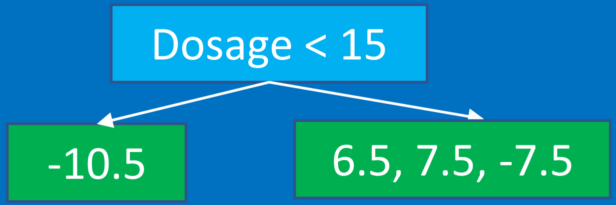
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree



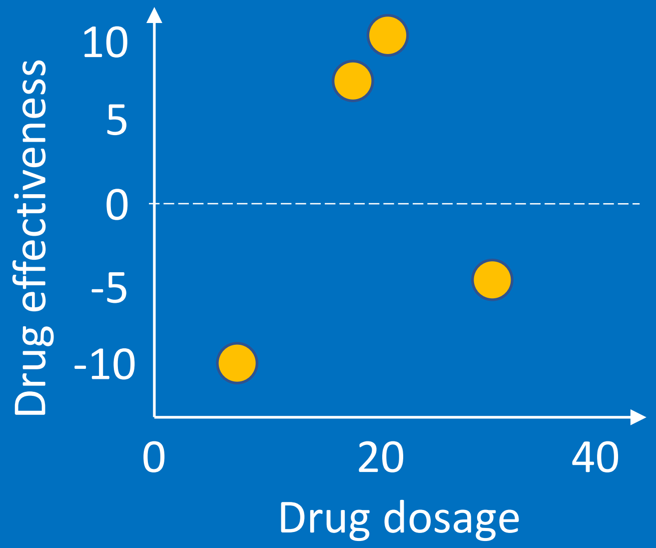
Step 4: Prune the tree



Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



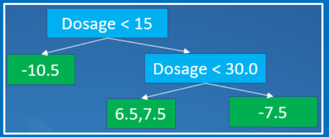
Step 1: make an initial prediction

0.5

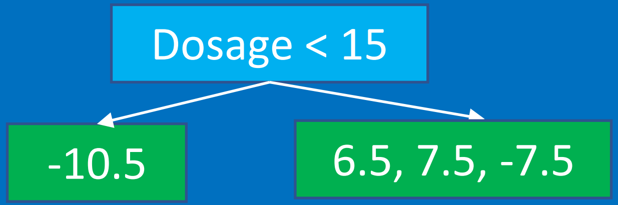
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree



Step 4: Prune the tree

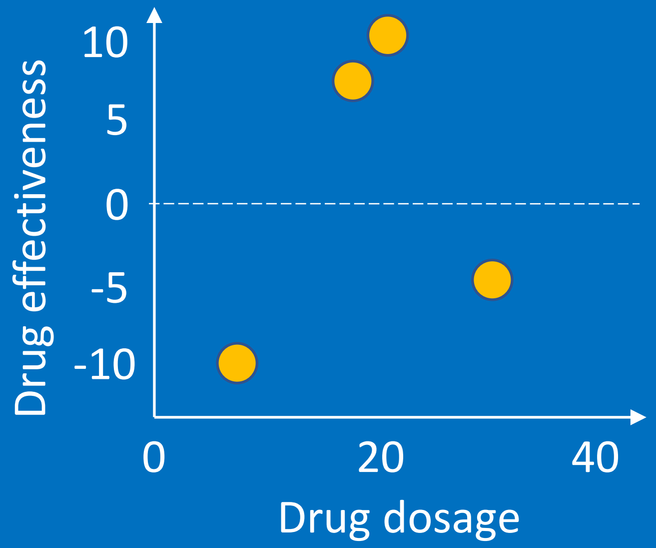


Now, as a related subject, let's look at how the regularization parameter λ would affect the tree build

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



For example, when we just start splitting the tree

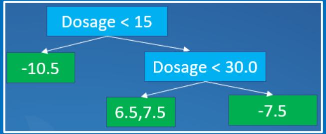
Step 1: make an initial prediction

0.5

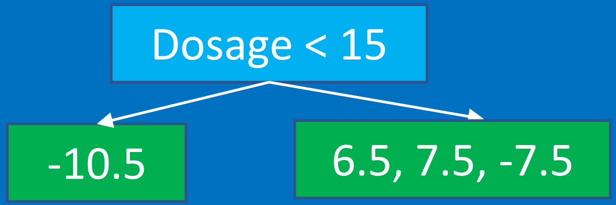
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

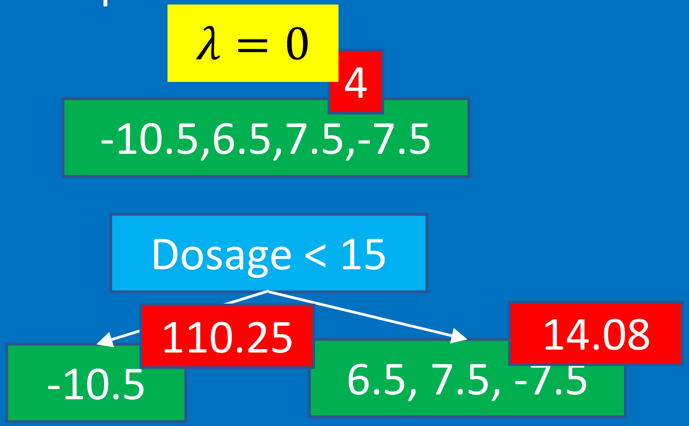
Step 3: Grow a XGBoost tree



Step 4: Prune the tree



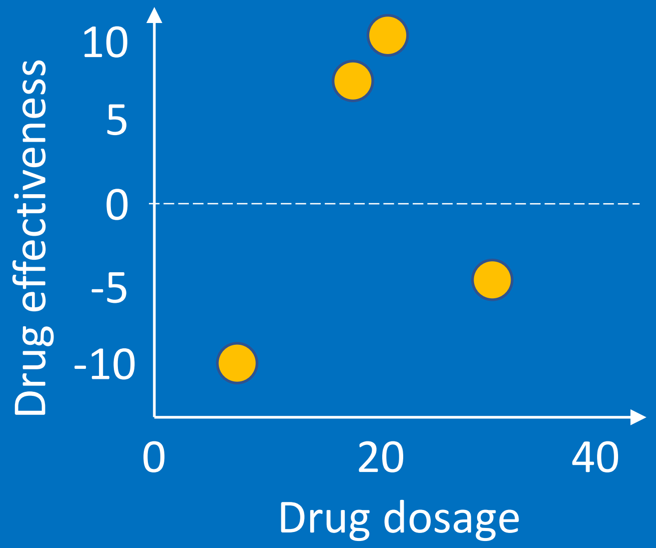
Now, as a related subject, let's look at how the regularization parameter λ would affect the tree build



Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



For example, when we just start splitting the tree

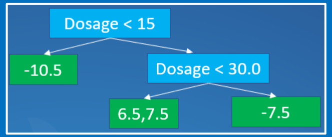
Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree



Step 4: Prune the tree

Dosage < 15

-10.5

6.5, 7.5, -7.5

Now, as a related subject, let's look at how the regularization parameter λ would affect the tree build

$\lambda = 0$

4

-10.5,6.5,7.5,-7.5

Dosage < 15

110.25

-10.5

14.08

6.5, 7.5, -7.5

$\lambda = 1$

3.2

-10.5,6.5,7.5,-7.5

Dosage < 15

55.12

-10.5

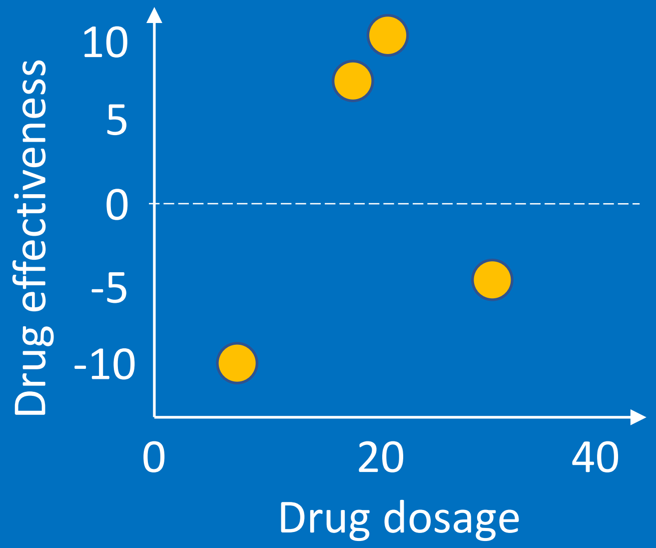
10.56

6.5, 7.5, -7.5

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



For example, when we just start splitting the tree

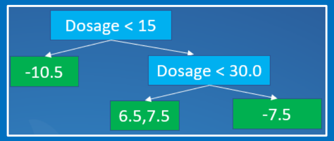
Step 1: make an initial prediction

0.5

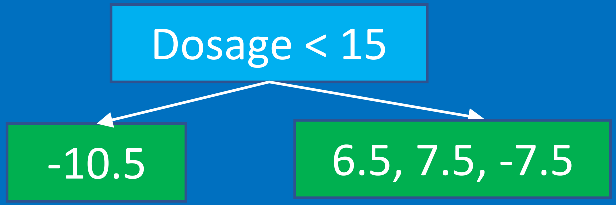
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

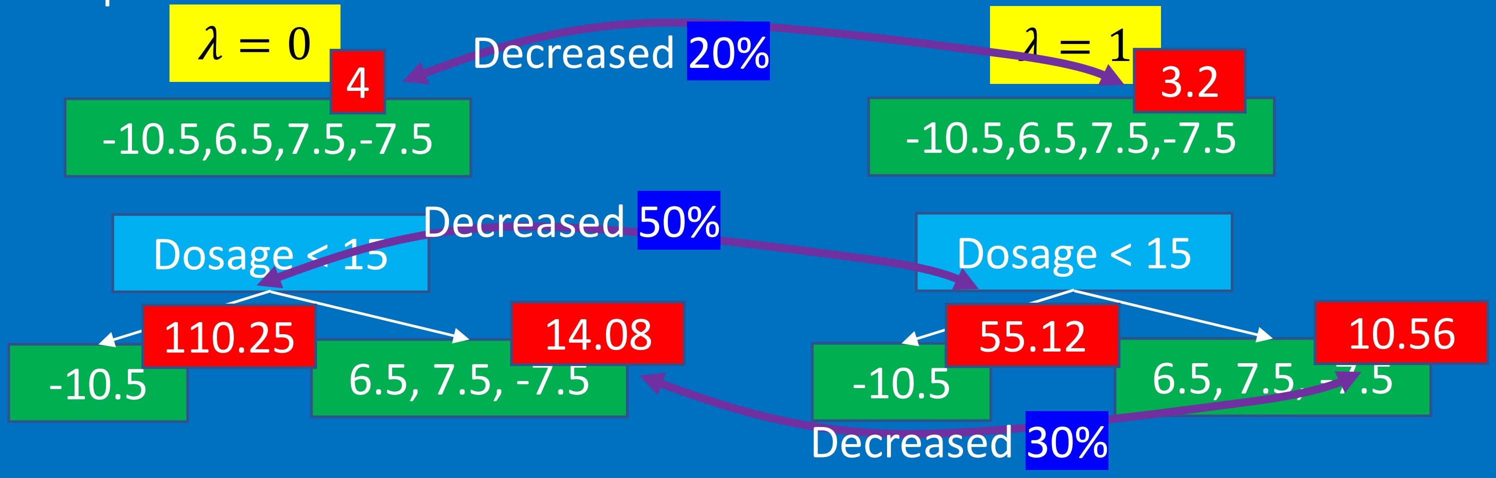
Step 3: Grow a XGBoost tree



Step 4: Prune the tree



Now, as a related subject, let's look at how the regularization parameter λ would affect the tree build



Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out

For

So, we can tell that,

(1) With increased λ, the similarity score is getting smaller
(2) The more residuals in the leaf, the less changes on the similarity score

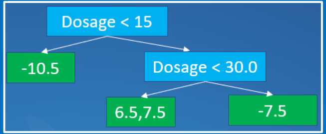
Step 1: make an initial prediction

0.5

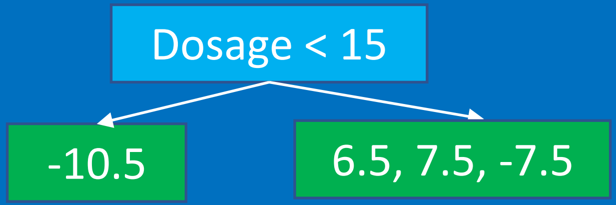
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

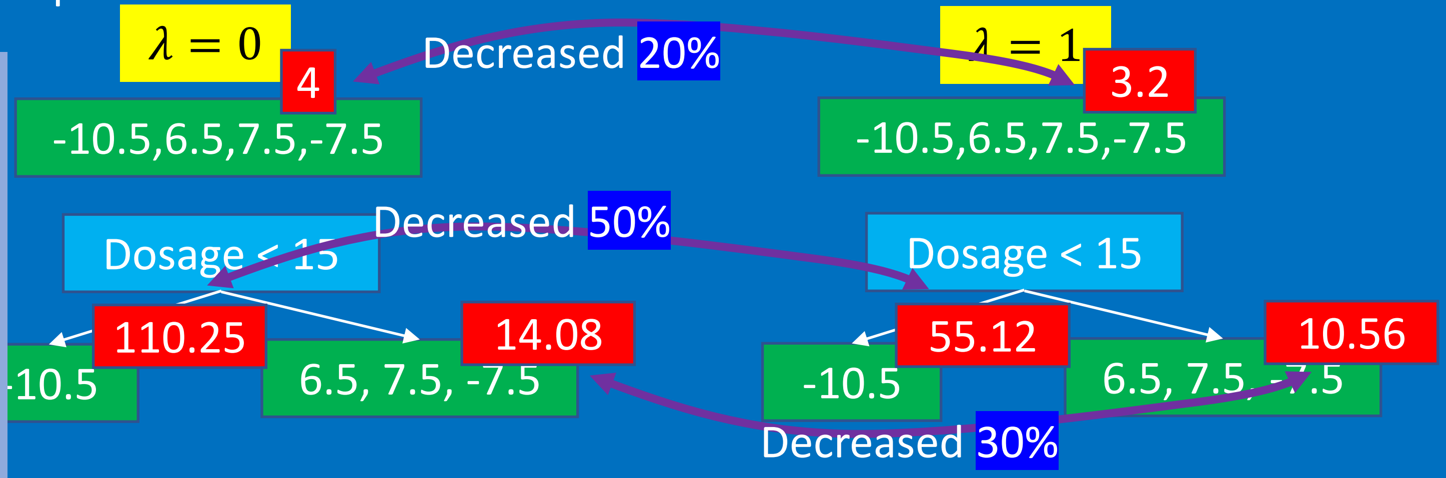
Step 3: Grow a XGBoost tree



Step 4: Prune the tree



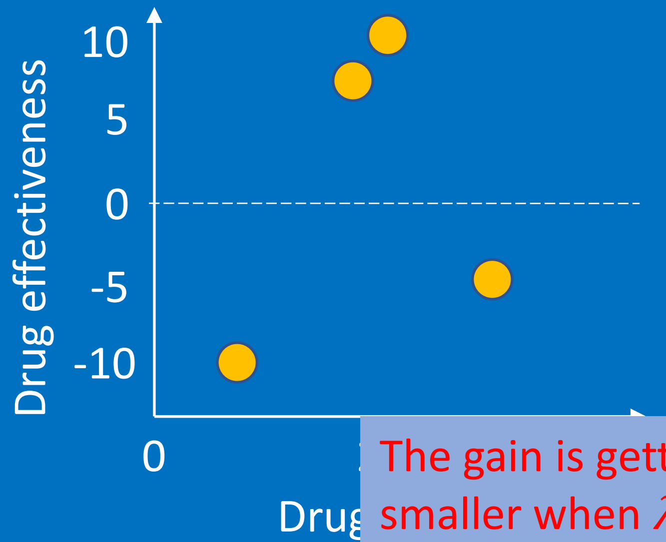
Now, as a related subject, let's look at how the regularization parameter λ would affect the tree build



Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



For example, when we just start splitting the tree

The gain is getting much smaller when λ increases

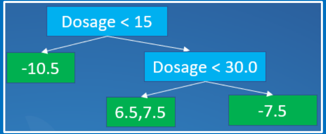
Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree



Step 4: Prune the tree

Dosage < 15

-10.5

6.5, 7.5, -7.5

Now, as a related subject, let's look at how the regularization parameter λ would affect the tree build

$\lambda = 0$

4

-10.5,6.5,7.5,-7.5

Decreased 20%

$\lambda = 1$

3.2

-10.5,6.5,7.5,-7.5

Dosage < 15

110.25

-10.5

14.08

6.5, 7.5, -7.5

Decreased 50%

Dosage < 15

55.12

-10.5

10.56

6.5, 7.5, -7.5

Decreased 30%

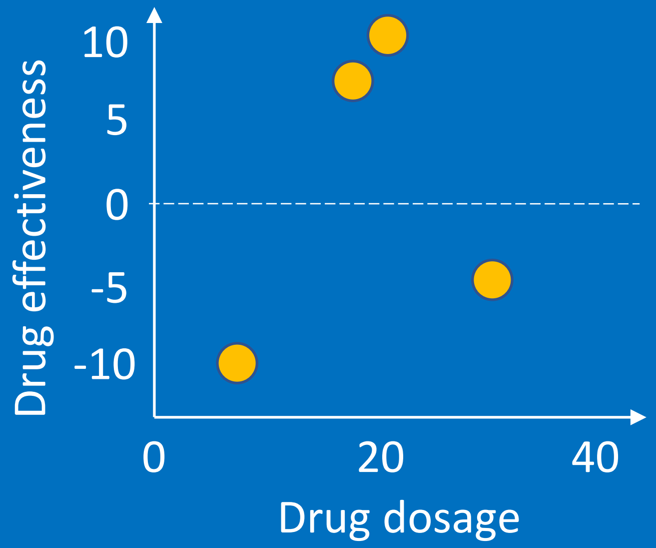
Gain = 120.33

Gain = 62.48

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



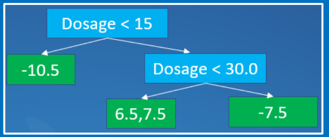
Step 1: make an initial prediction

0.5

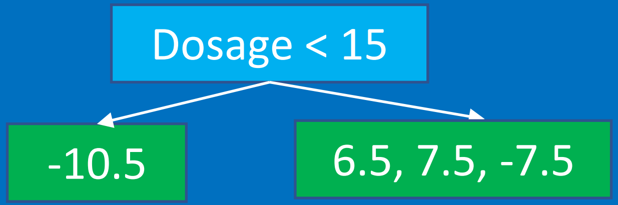
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

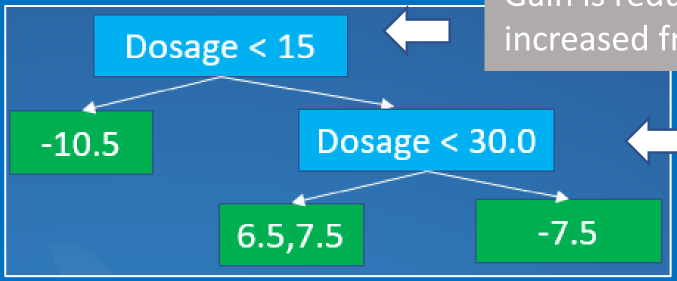
Step 3: Grow a XGBoost tree



Step 4: Prune the tree



Now, as a related subject, let's look at how the regularization parameter λ would affect the tree build



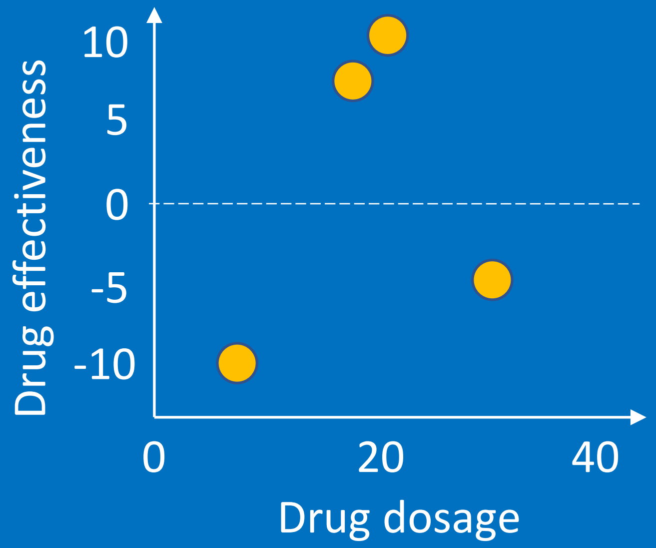
Gain is reduced from 120.33 to 62.48 when λ is increased from 0 to 1

Similarly, The "Gain" for the subsequent branch will be smaller

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



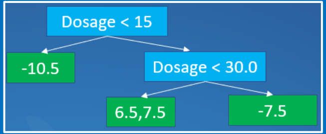
Step 1: make an initial prediction

0.5

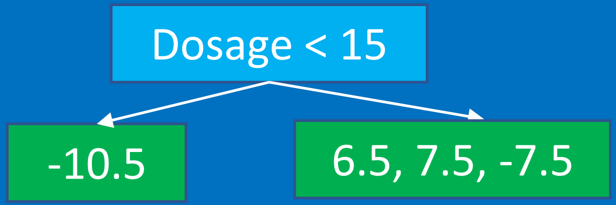
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

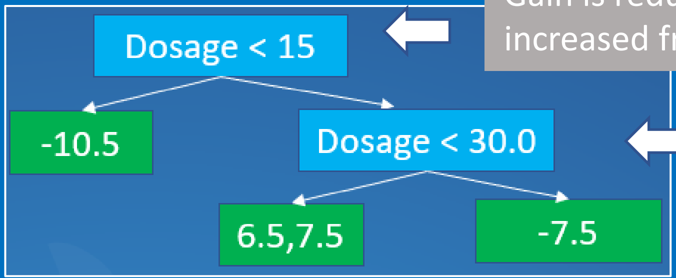
Step 3: Grow a XGBoost tree



Step 4: Prune the tree



Now, as a related subject, let's look at how the regularization parameter λ would affect the tree build



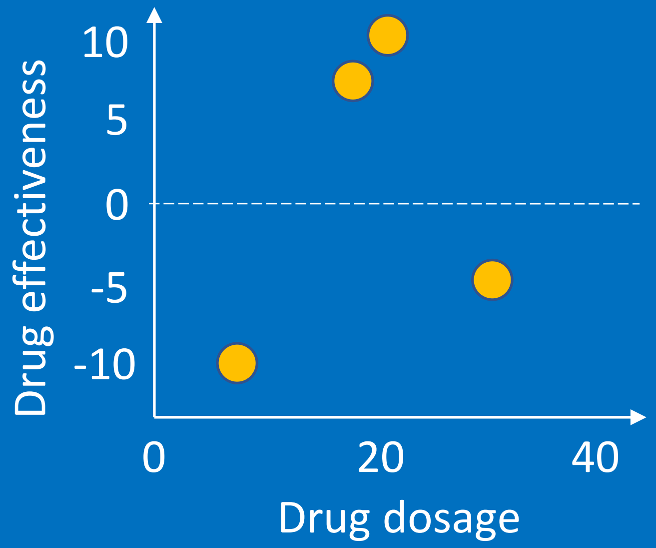
Gain is reduced from 120.33 to 62.48 when λ is increased from 0 to 1

In this case, Gain is reduced from 140.17 to 82.9

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



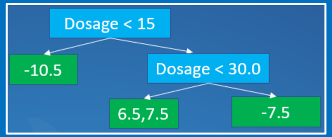
Step 1: make an initial prediction

0.5

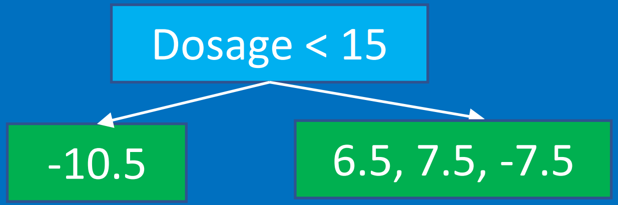
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

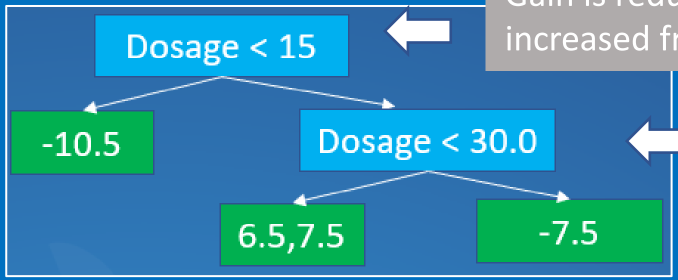
Step 3: Grow a XGBoost tree



Step 4: Prune the tree



Now, as a related subject, let's look at how the regularization parameter λ would affect the tree build



Gain is reduced from 120.33 to 62.48 when λ is increased from 0 to 1

In this case, Gain is reduced from 140.17 to 82.9

140.17 – 130 > 0

82.9 – 130 < 0

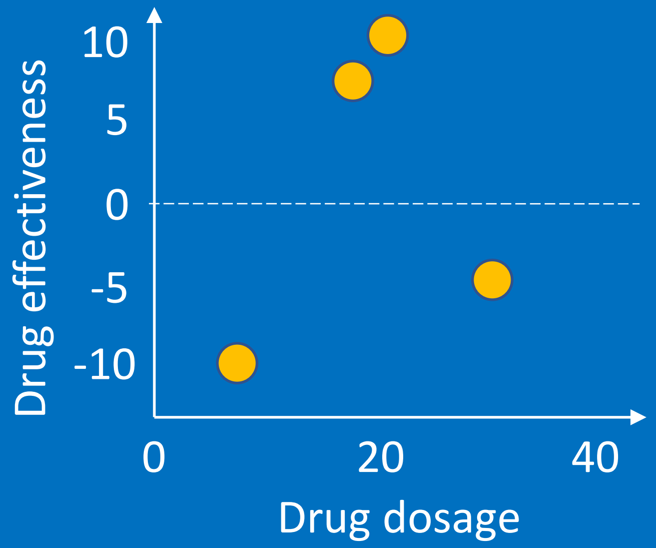
So when $\gamma = 130$:

- When λ=0: we don't prune the tree at all (the last branch has the gain of 140.0)
- When λ=1: we need to prune the tree (so the predictions is less dependant on the variables)

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



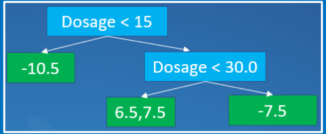
Step 1: make an initial prediction

0.5

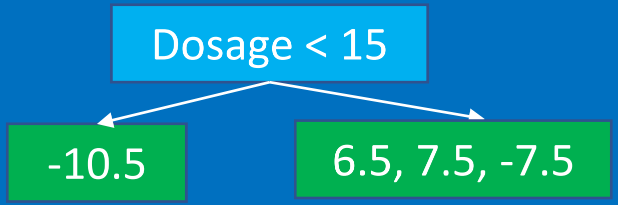
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree



Step 4: Prune the tree



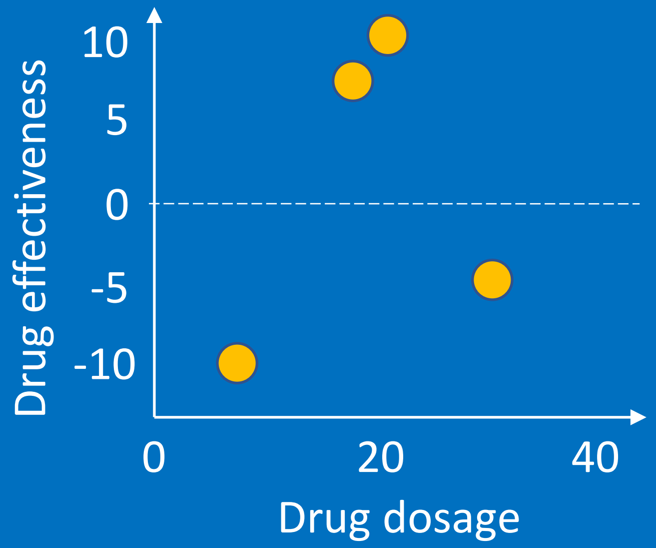
Now, as a related subject, let's look at how the regularization parameter λ would affect the tree build

Conclusion, if $\lambda > 0$, it's easier to prune leaves because the values for Gain are smaller, and therefore prevent overfitting data

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

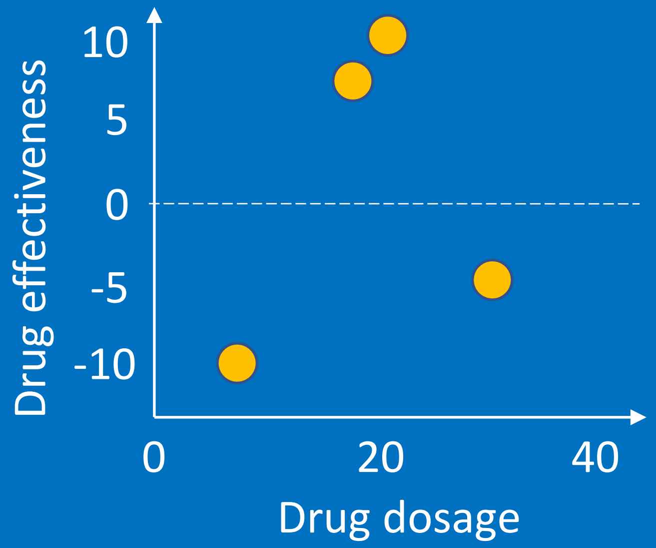
Step 4: Prune the tree

Step 5: Get the tree output

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

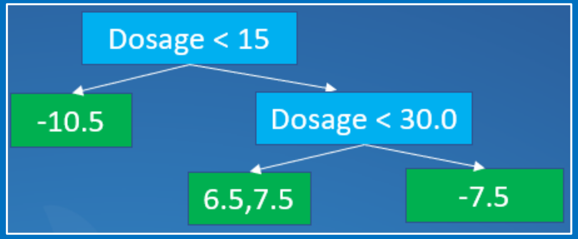
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

Step 5: Get the tree output

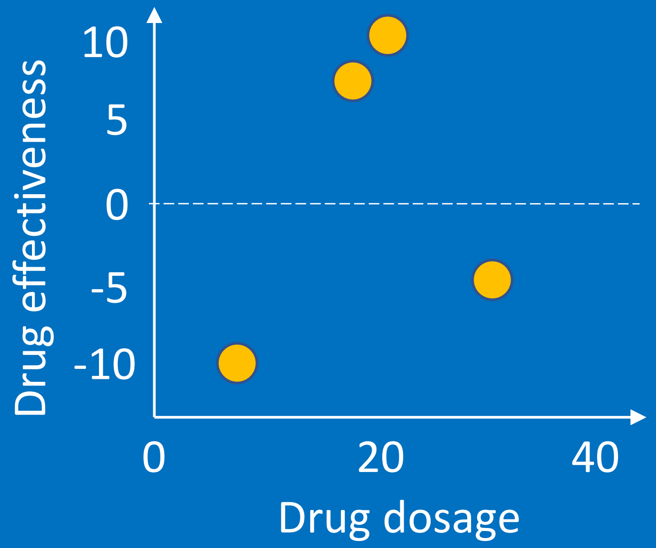


Assuming that after all the pruning, this is the tree we get

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

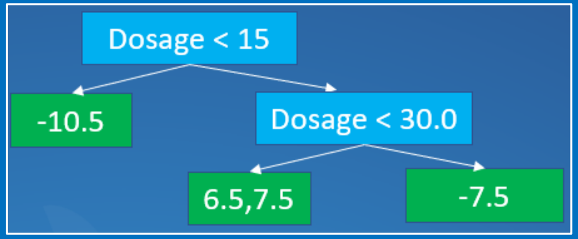
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

Step 5: Get the tree output



Assuming that after all the pruning, this is the tree we get

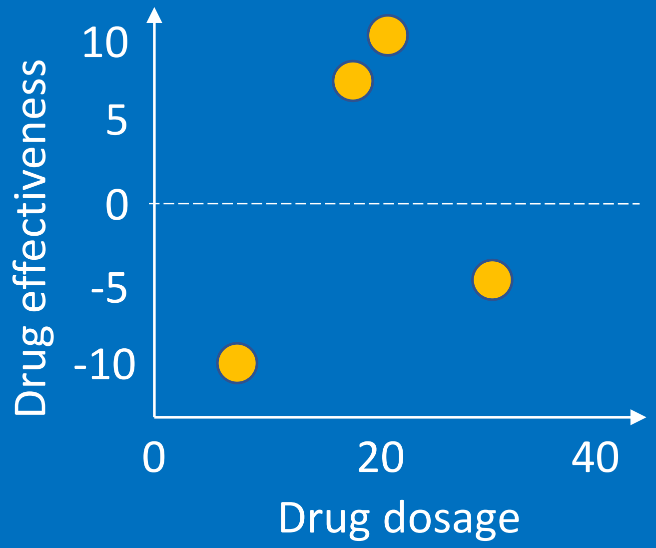
$$output = \frac{sum\ of\ residuals}{number\ of\ residuals + \lambda}$$

Regularization parameter

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

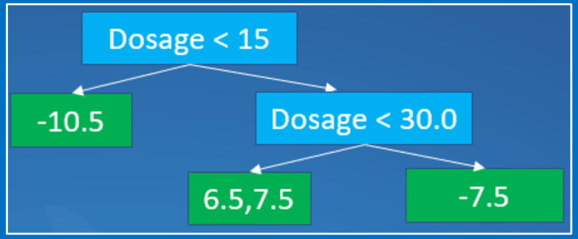
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

Step 5: Get the tree output



Assuming that after all the pruning, this is the tree we get

$$output = \frac{sum\ of\ residuals}{number\ of\ residuals +\ \lambda}$$

Regularization parameter

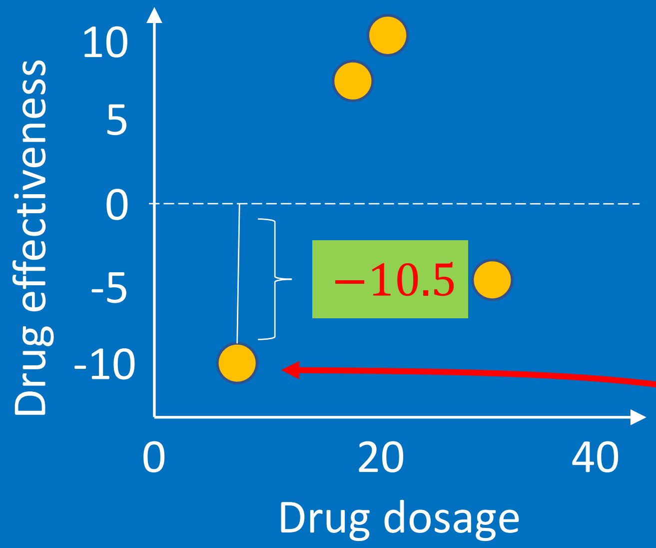
Therefore we have

$$output_{\lambda=0} = \frac{-10.5}{1+0} = -10.5$$
$$output_{\lambda=1} = \frac{-10.5}{1+0} = -5.25$$

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

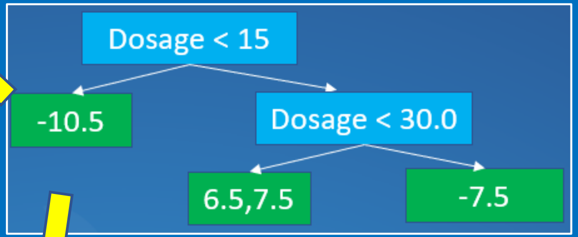
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

Step 5: Get the tree output



Assuming that after all the pruning, this is the tree we get

$$output = \frac{sum\ of\ residuals}{number\ of\ residuals + \lambda}$$

Therefore we have

$$output_{\lambda=0} = \frac{-10.5}{1+0} = -10.5$$

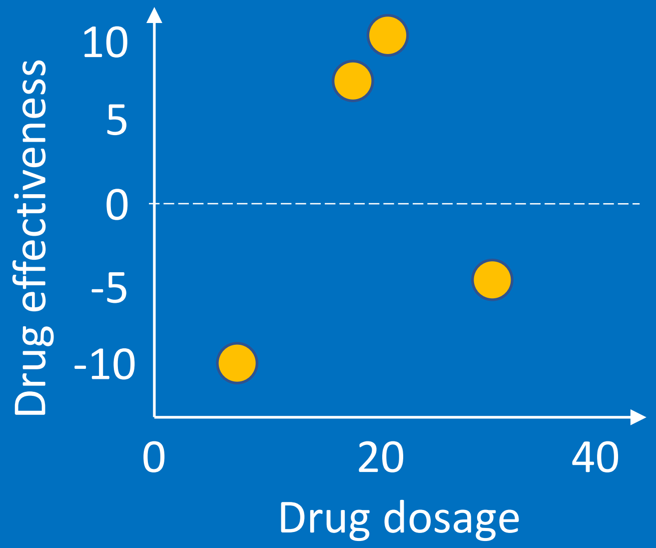
$$output_{\lambda=1} = \frac{-10.5}{1+0} = -5.25$$

The output actually reflects the residual we want to predict from the xgboost tree, so by increasing the "lambda", we actually makes the residuals smaller ~ this means the reduction of prediction's sensitivity to <u>this particular observation</u>

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



For now we just keep it sample, and let $\lambda = 0$

Step 1: make an initial prediction

0.5

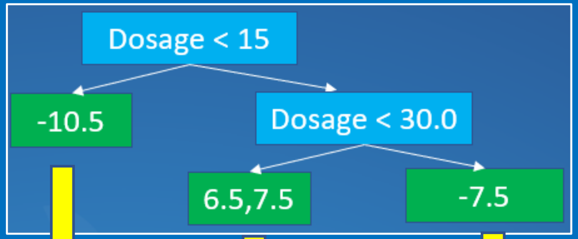
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

Step 5: Get the tree output



Assuming that after all the pruning, this is the tree we get

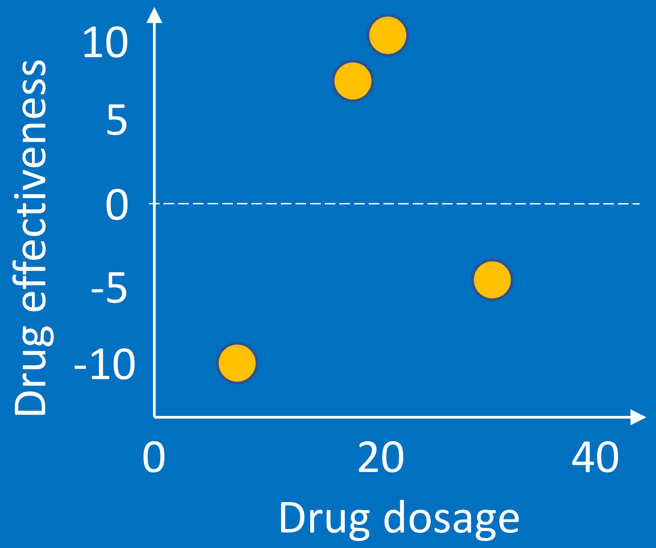
$$output = \frac{sum\ of\ residuals}{number\ of\ residuals + \lambda}$$

-10.5 7 -7.5

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

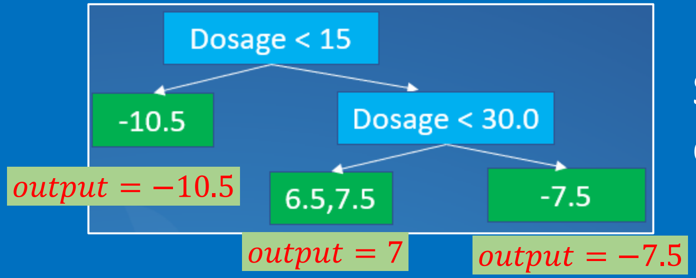
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

Step 5: Get the tree output

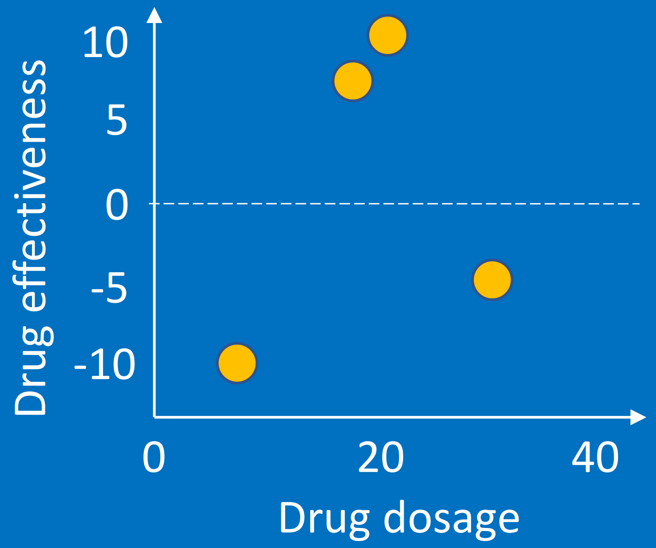


So up to now, the first tree is completed

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

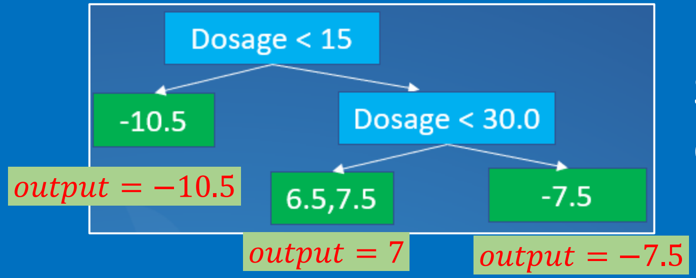
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

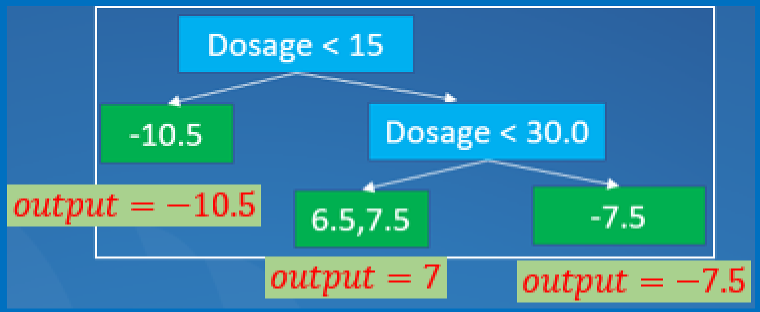
Step 5: Get the tree output



So up to now, the first tree is completed

Step 6: Make predictions

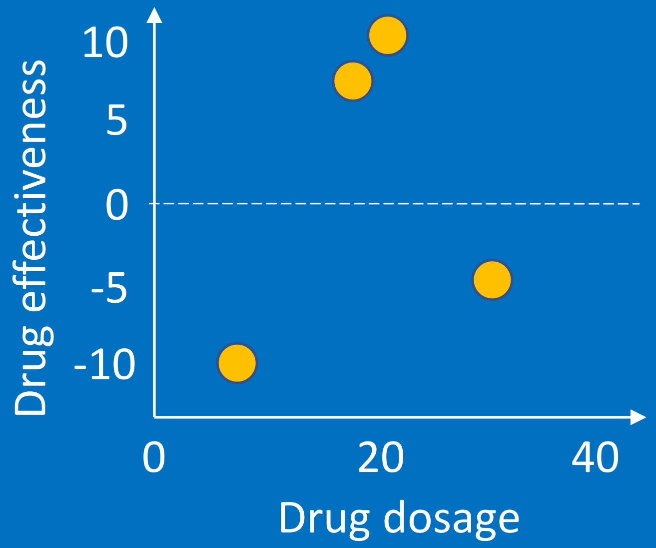
0.5 + Learning rate x



Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

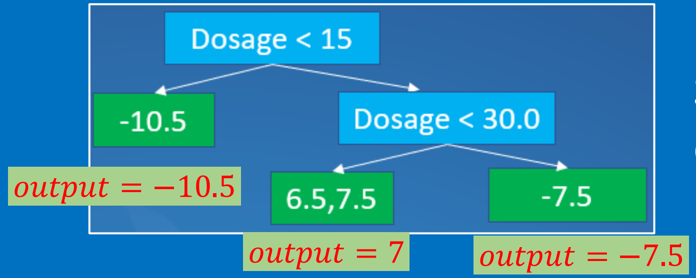
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

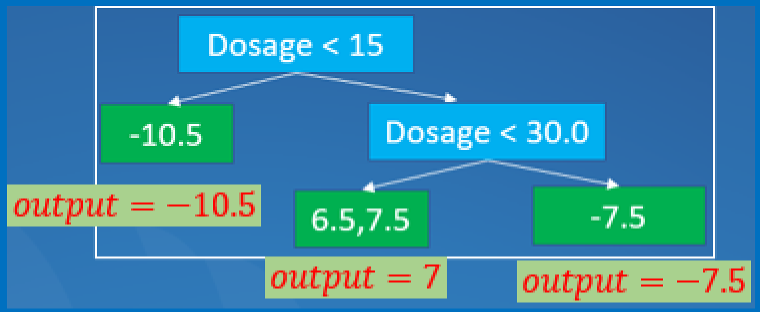
Step 5: Get the tree output



So up to now, the first tree is completed

Step 6: Make predictions

0.5 + Learning rate X

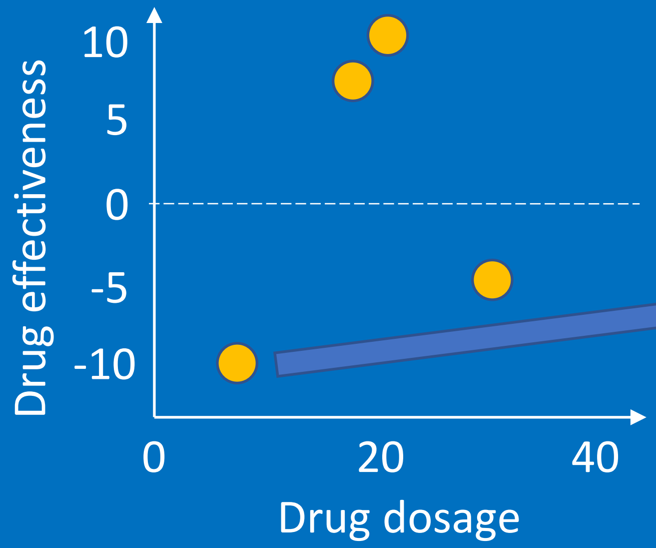


Let's assume it is 0.3

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

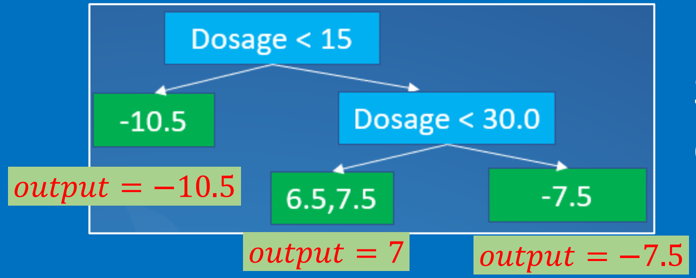
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

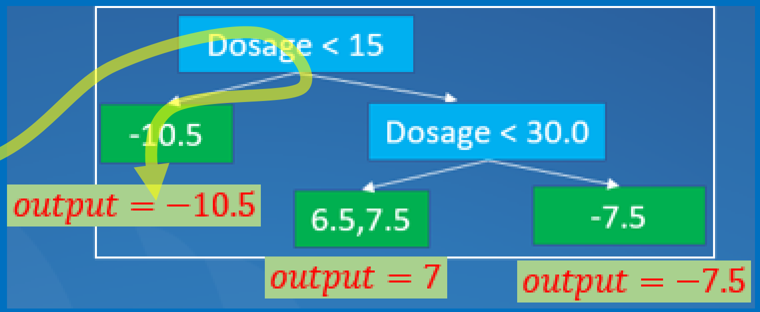
Step 5: Get the tree output



So up to now, the first tree is completed

Step 6: Make predictions

0.5 + 0.3 x

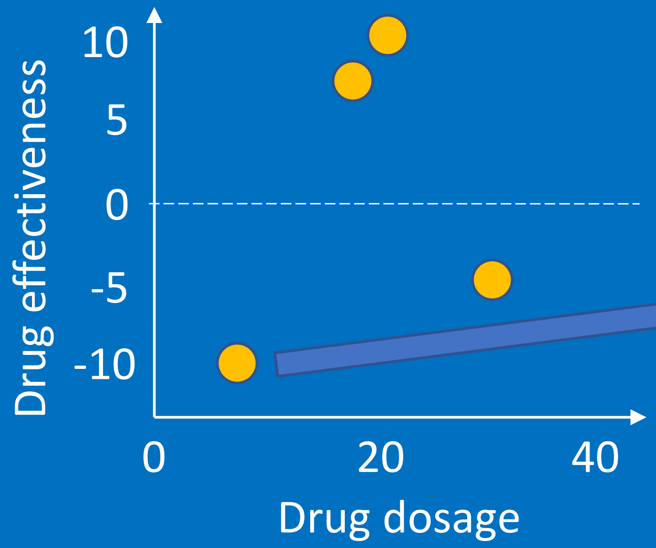


So the prediction for the first sample is:
0.5 + 0.3 x (-10.5) = -2.65

Drug dosage	Drug effectiveness	residuals
9	-10	-10.5
20	7	6.5
24	8	7.5
36	-7	-7.5

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

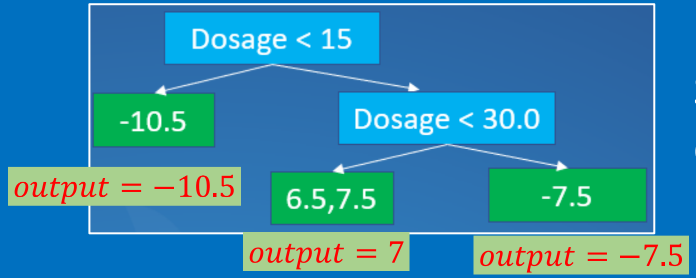
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

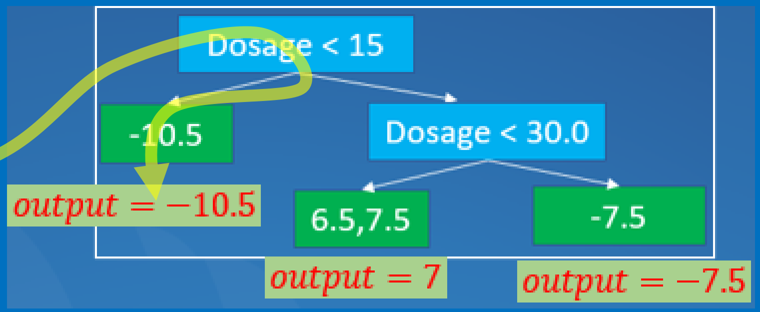
Step 5: Get the tree output



So up to now, the first tree is completed

Step 6: Make predictions

0.5 + 0.3 x



So the prediction for the first sample is:

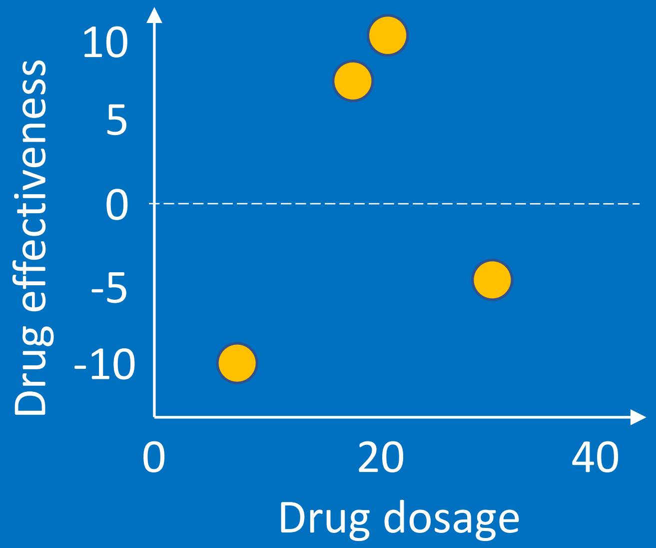
0.5 + 0.3 x (-10.5) = -2.65

This better than the initial prediction, which is 0.5

Drug dosage	Drug effectiveness	residuals	Residuals (+1)
9	-10	-10.5	……
20	7	6.5	
24	8	7.5	
36	-7	-7.5	

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

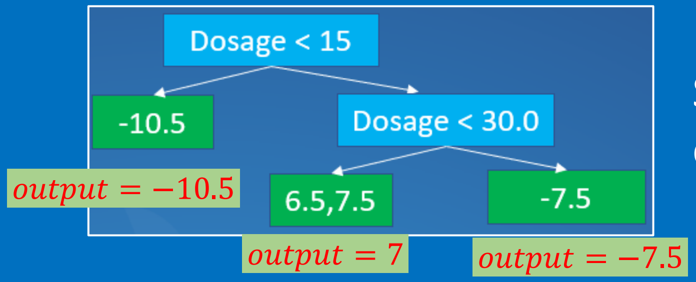
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

Step 5: Get the tree output



So up to now, the first tree is completed

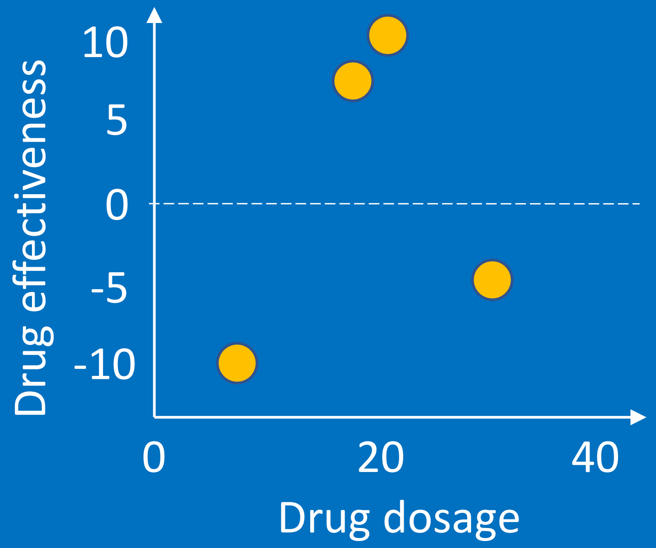
Step 6: Make predictions

We can go through this process for each samples, and we can get a new set of residuals

Drug dosage	Drug effectiveness	residuals	Residuals (+1)
9	-10	-10.5	……
20	7	6.5	
24	8	7.5	
36	-7	-7.5	

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

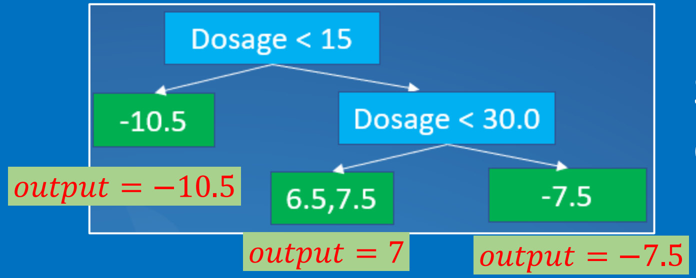
Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

Step 3: Grow a XGBoost tree

Step 4: Prune the tree

Step 5: Get the tree output



So up to now, the first tree is completed

Step 6: Make predictions

We can go through this process for each samples, and we can get a new set of residuals

Step 7: Iteratively making many trees based on updated residuals

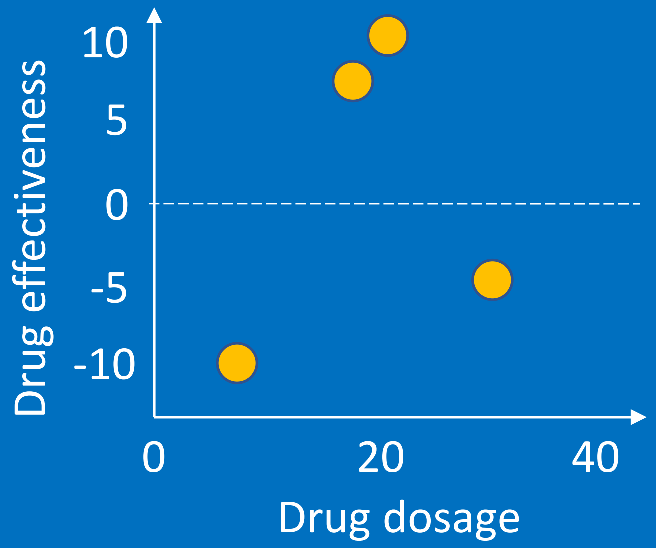
Then we can go back to step 3 and grow another tree

By going this process iteratively, we can have many trees

Drug dosage	Drug effectiveness	residuals	Residuals (+1)
9	-10	-10.5	……
20	7	6.5	
24	8	7.5	
36	-7	-7.5	

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

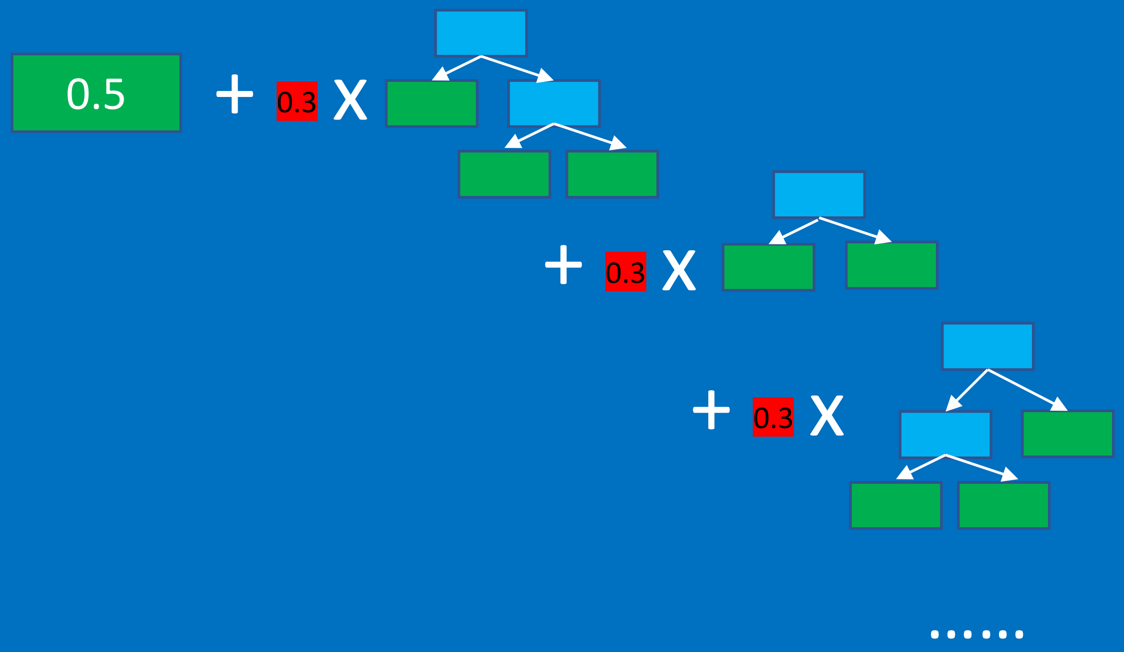
Step 3: Grow a XGBoost tree

Step 4: Prune the tree

Step 5: Get the tree output

Step 6: Make predictions

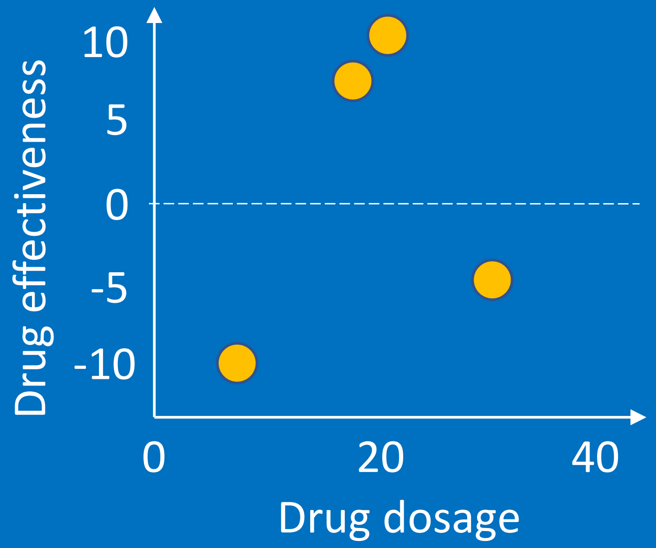
Step 7: Iteratively making many trees based on updated residuals



Drug dosage	Drug effectiveness	residuals	Residuals (+1)
9	-10	-10.5	……
20	7	6.5	
24	8	7.5	
36	-7	-7.5	

Let's assume that dataset to be used

Let's plot it out



Step 1: make an initial prediction

0.5

Let's assume that the "initial guess" of "predicted drug effectiveness" is 0.5 (so for whatever testing data, the prediction is always 0.5)

Step 2: Obtain the residuals

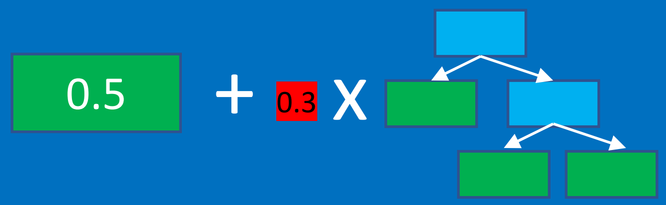
Step 3: Grow a XGBoost tree

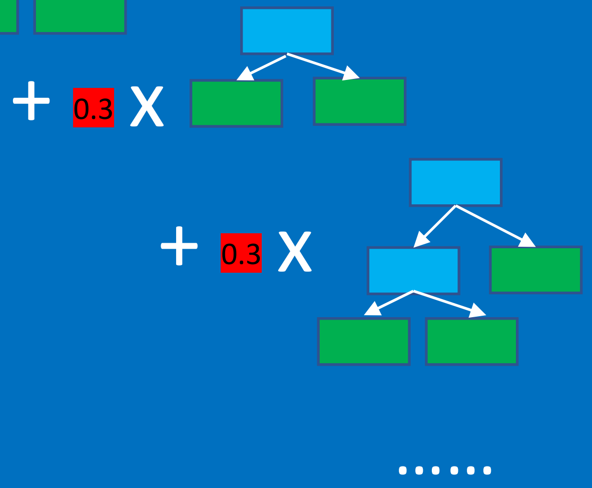
Step 4: Prune the tree

Step 5: Get the tree output

Step 6: Make predictions

Step 7: Iteratively making many trees based on updated residuals





We keep building trees until the residuals are super small, or we have reached the maximum number