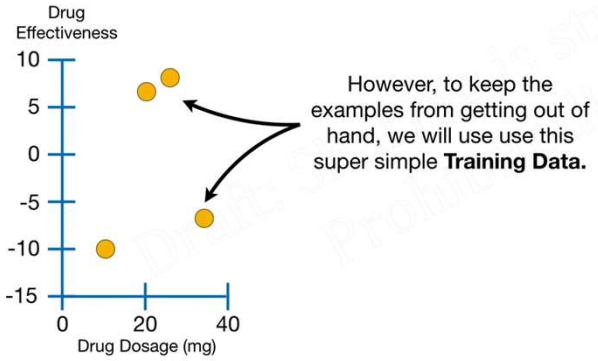
BOOSTING

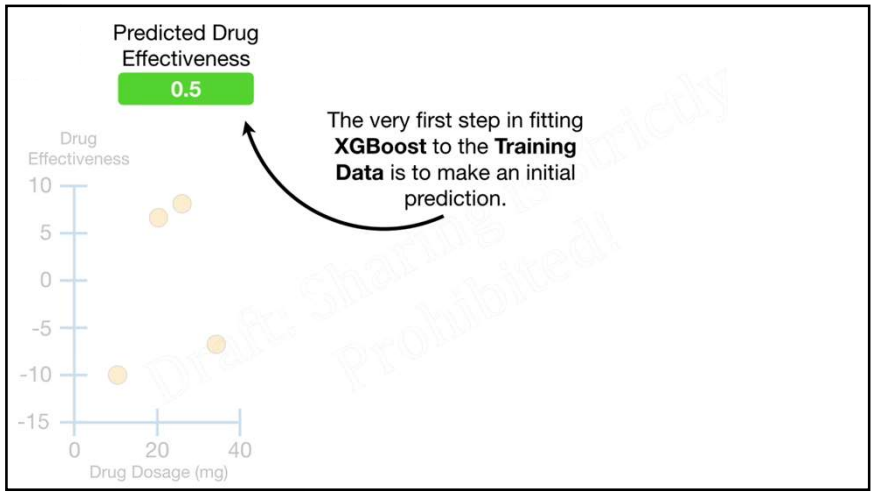
- AdaBoost
- Gradient Boost
- XGBoost

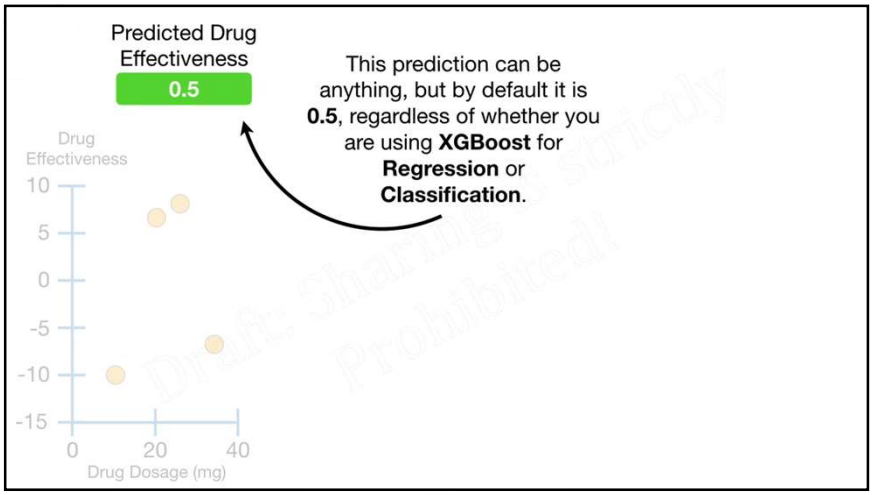
XGBOOST

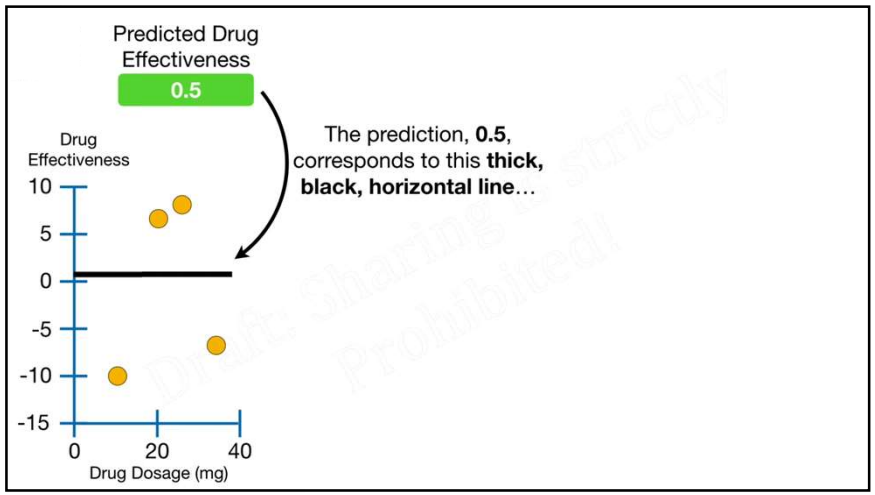
XGBoost was designed to be used with large, complicated data sets.

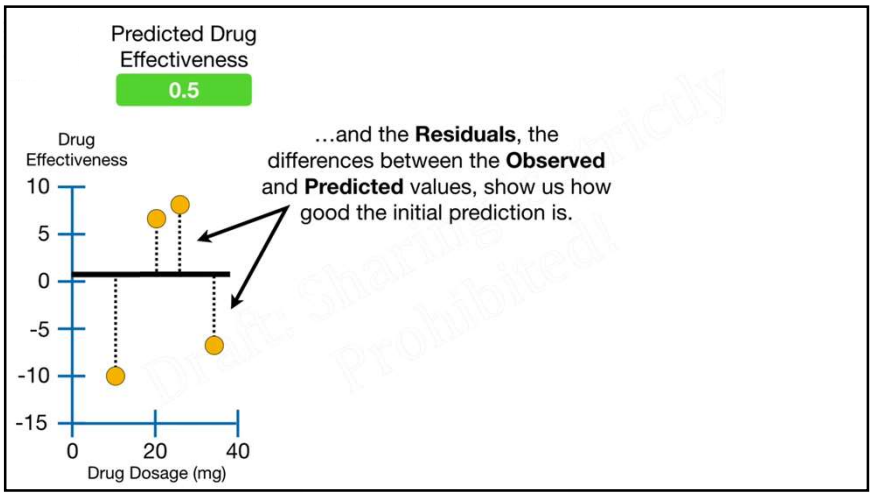
Mass	Age	BP	Color	Movie	Car	Hair	etc.
120	23	102	Brown	T2	Ford	Long	...
150	25	98	Brown	Frozen	Kia	Short	...
165	22	130	Black	Spiderman	Ford	Short	...
123	45	98	Red	T2	Kia	Long	...
156	33`	78	Brown	Frozen 2	Ford	Long	...
...	...	...	...	...	...	...	...

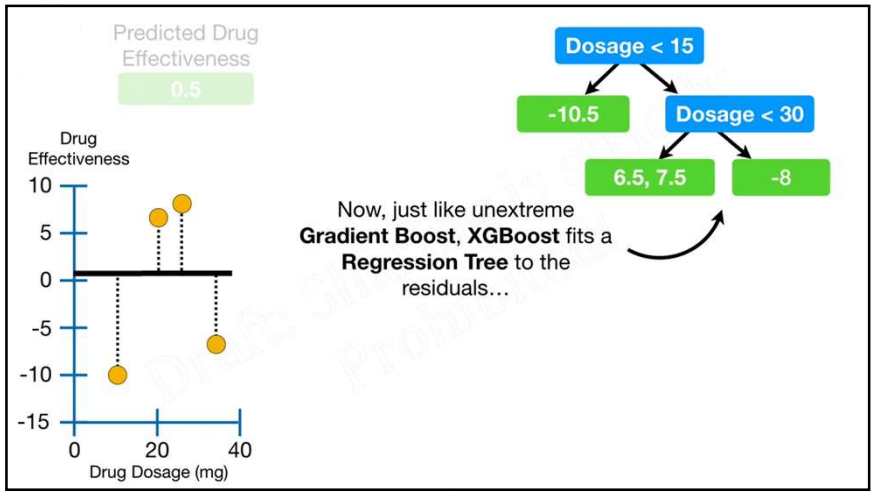


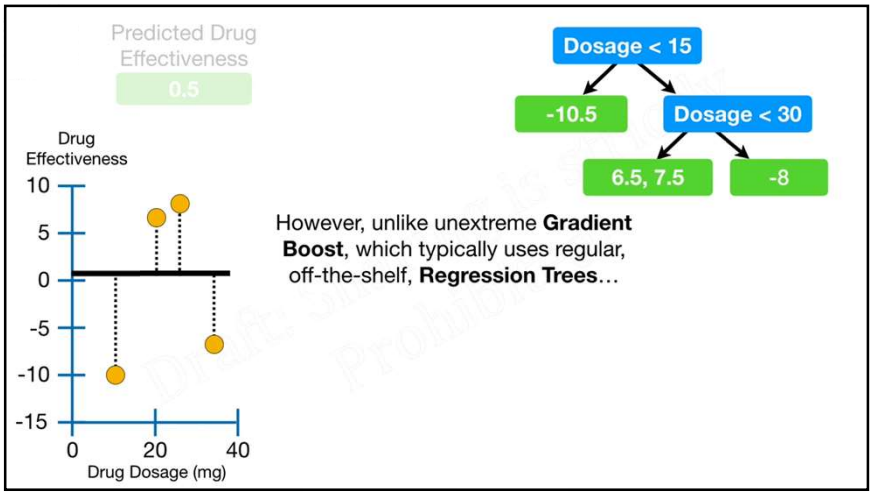

Predicted Drug Effectiveness
0.5
The very first step in fitting **XGBoost** to the **Training Data** is to make an initial prediction.
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)

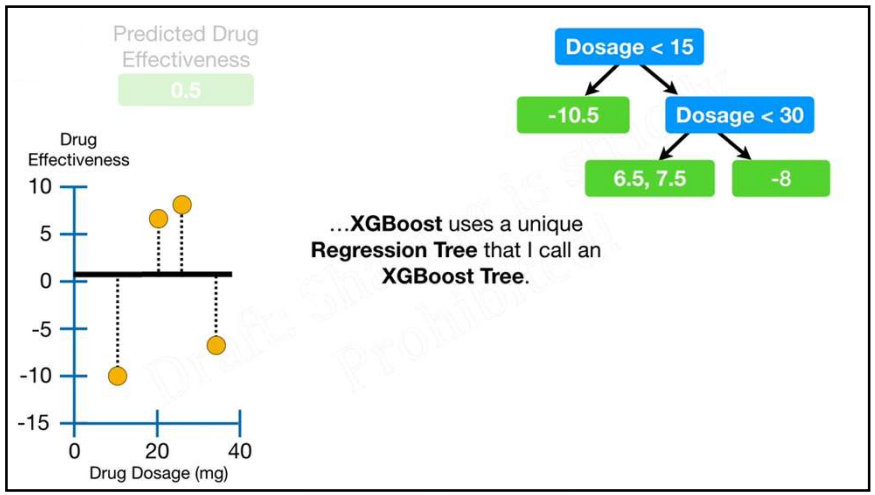

Predicted Drug Effectiveness
0.5
This prediction can be anything, but by default it is **0.5**, regardless of whether you are using **XGBoost** for **Regression** or **Classification**.
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)

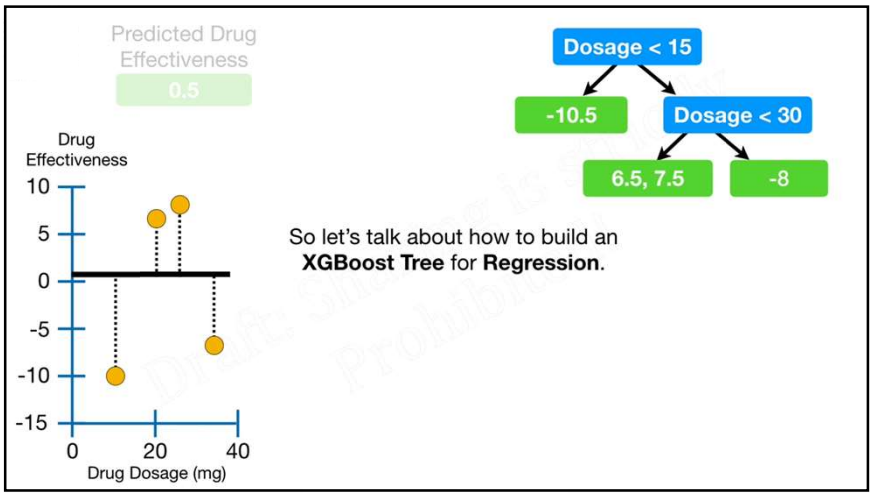

Predicted Drug Effectiveness
0.5
The prediction, **0.5**, corresponds to this **thick, black, horizontal line**…
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)

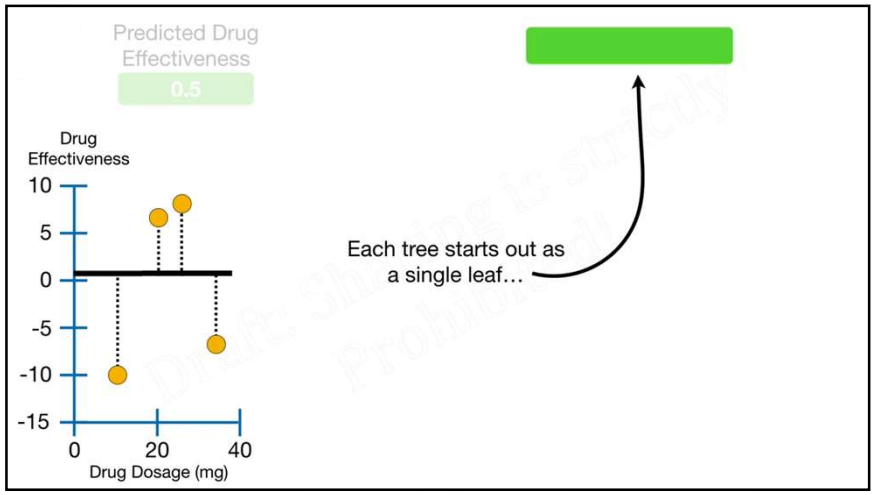

Predicted Drug Effectiveness
0.5
…and the **Residuals**, the differences between the **Observed** and **Predicted** values, show us how good the initial prediction is.
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)

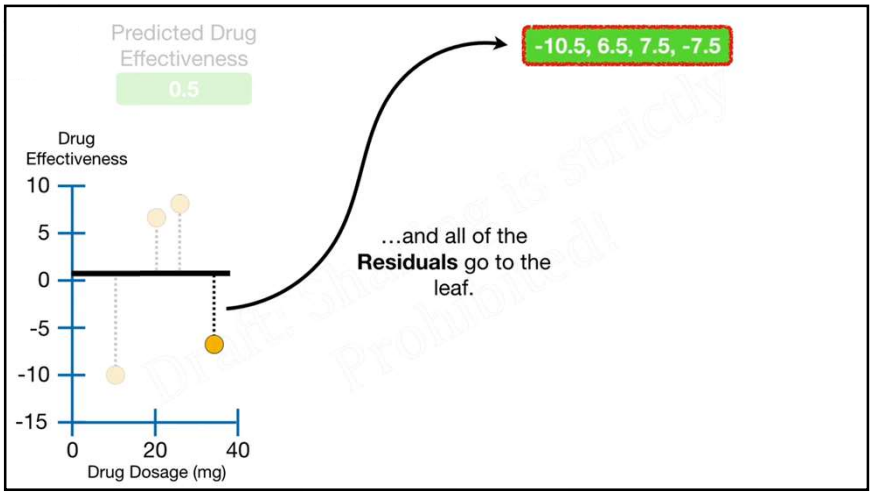

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-8
Now, just like unextreme **Gradient Boost**, **XGBoost** fits a **Regression Tree** to the residuals…

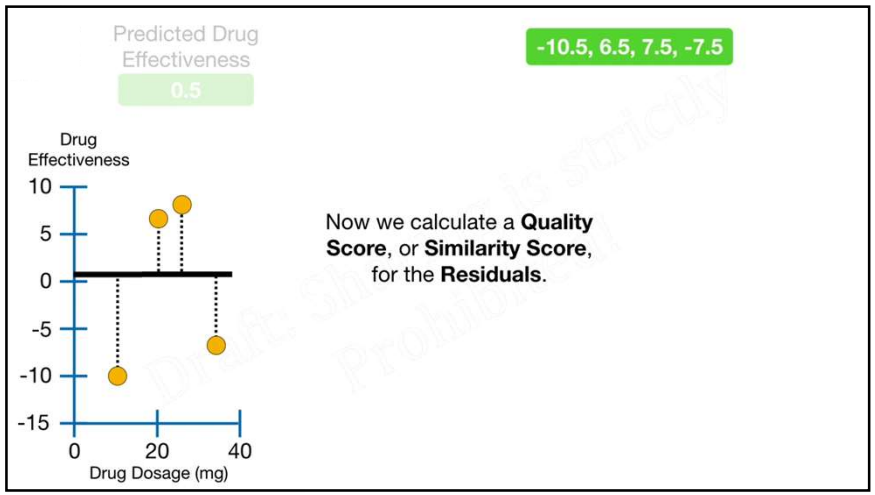

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-8
However, unlike unextreme **Gradient Boost**, which typically uses regular, off-the-shelf, **Regression Trees**…

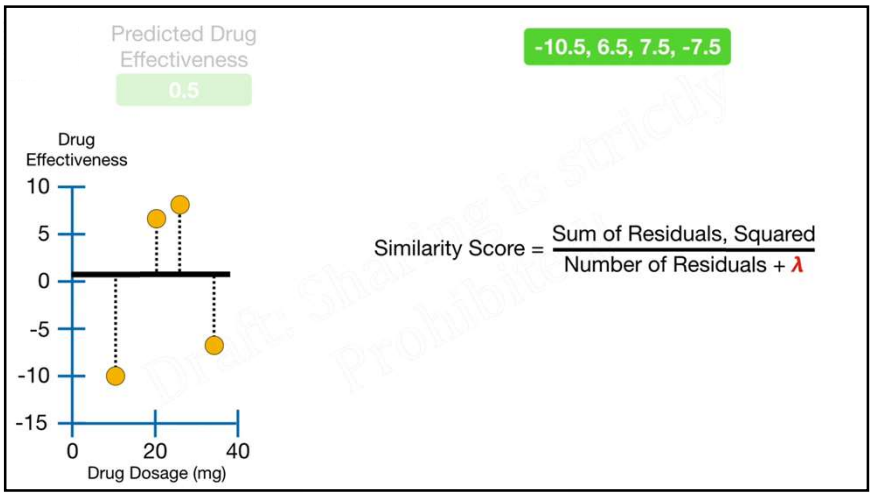

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-8
…**XGBoost** uses a unique **Regression Tree** that I call an **XGBoost Tree**.

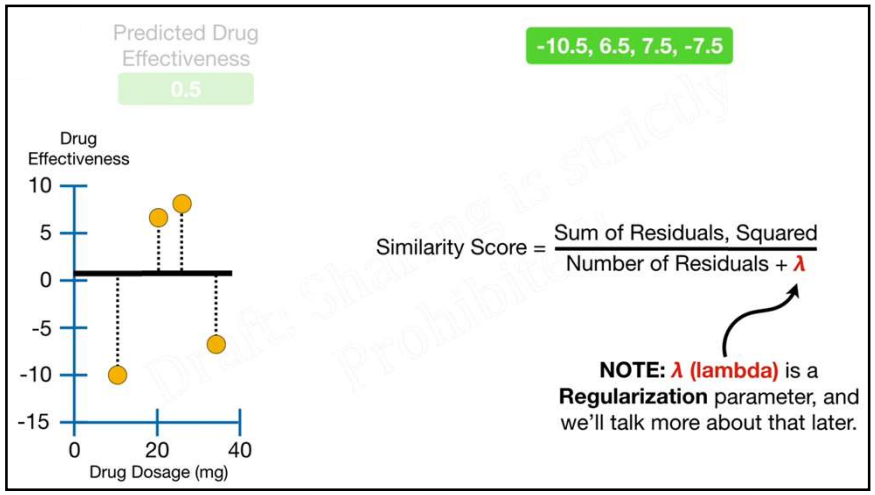

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-8
So let's talk about how to build an **XGBoost Tree** for **Regression**.

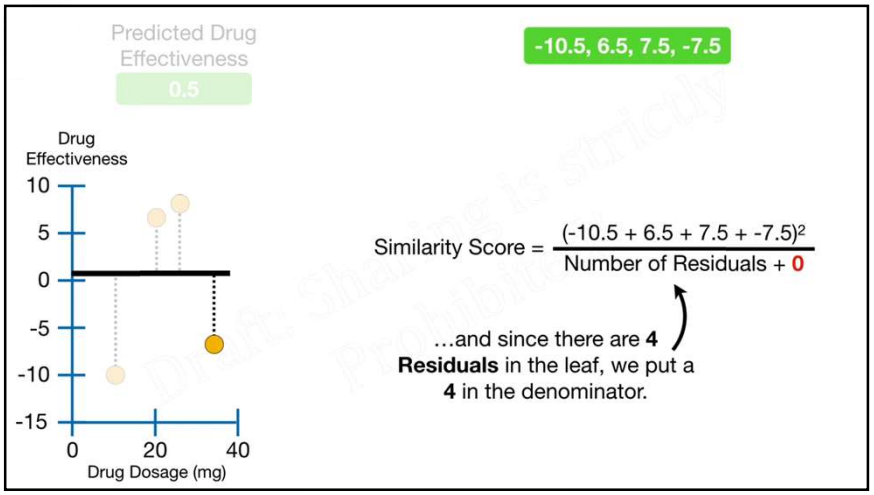

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Each tree starts out as a single leaf…

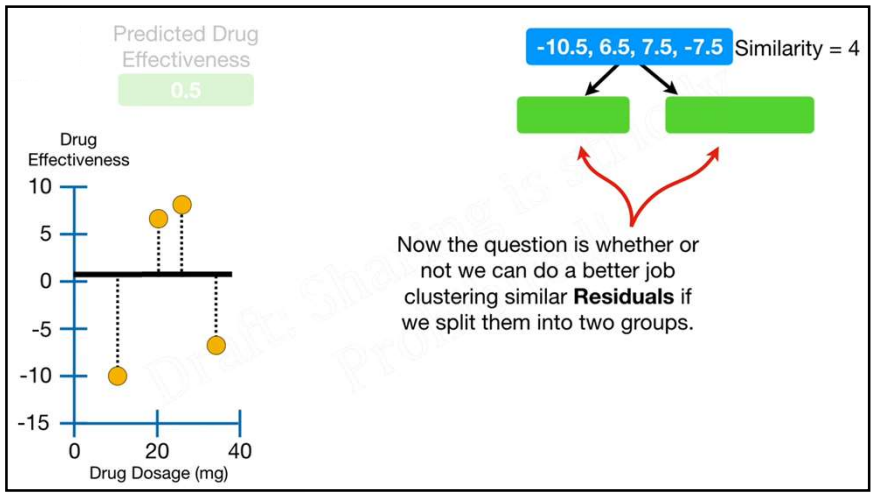

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…and all of the **Residuals** go to the leaf.

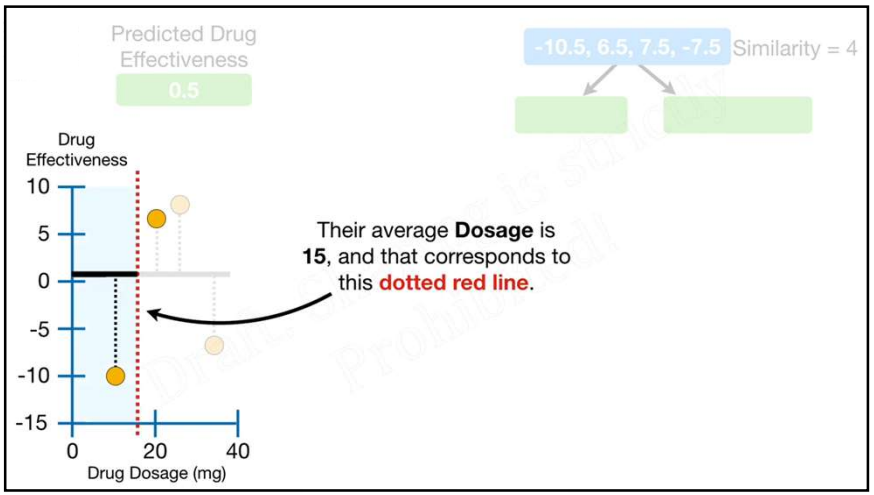

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Now we calculate a **Quality Score**, or **Similarity Score**, for the **Residuals**.

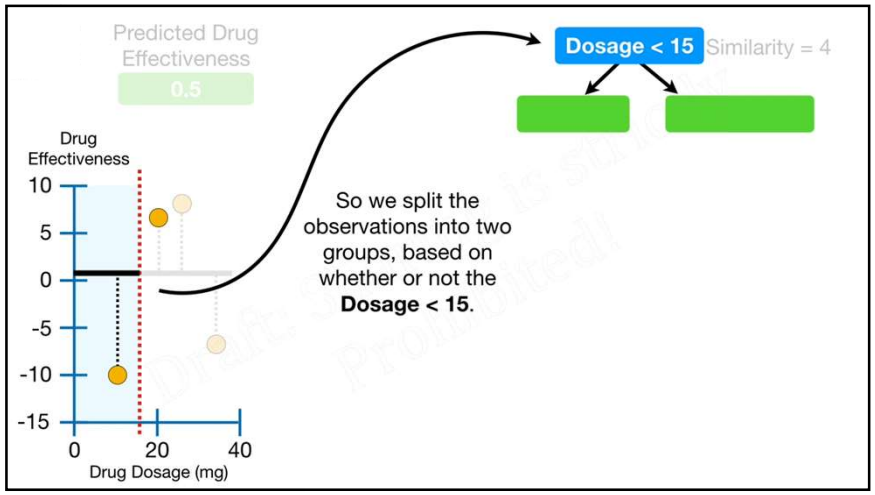

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Similarity Score = Sum of Residuals, Squared / Number of Residuals + λ

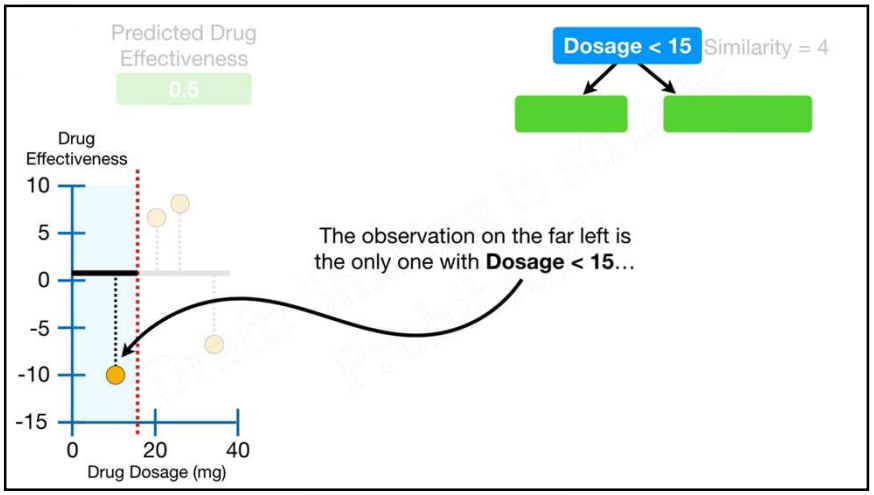

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Similarity Score = Sum of Residuals, Squared / Number of Residuals + λ
NOTE: λ (lambda) is a Regularization parameter, and we'll talk more about that later.

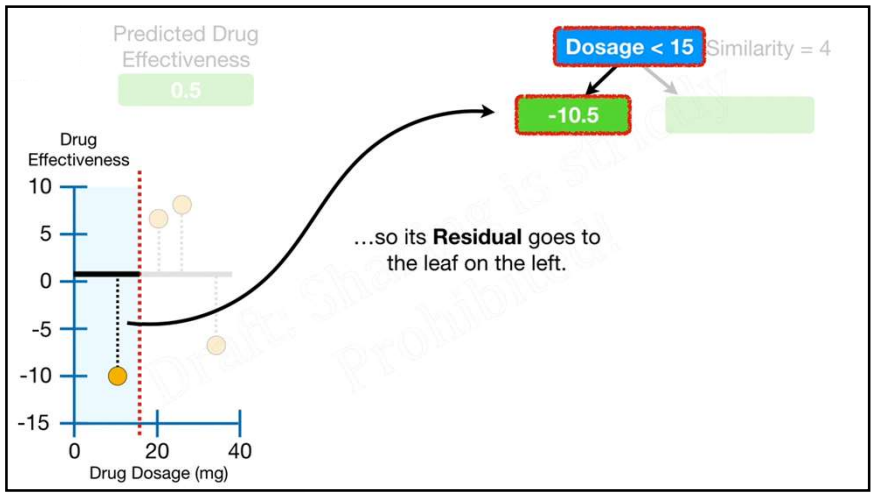

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Similarity Score = (-10.5 + 6.5 + 7.5 + -7.5)² / Number of Residuals + 0
…and since there are 4 Residuals in the leaf, we put a 4 in the denominator.

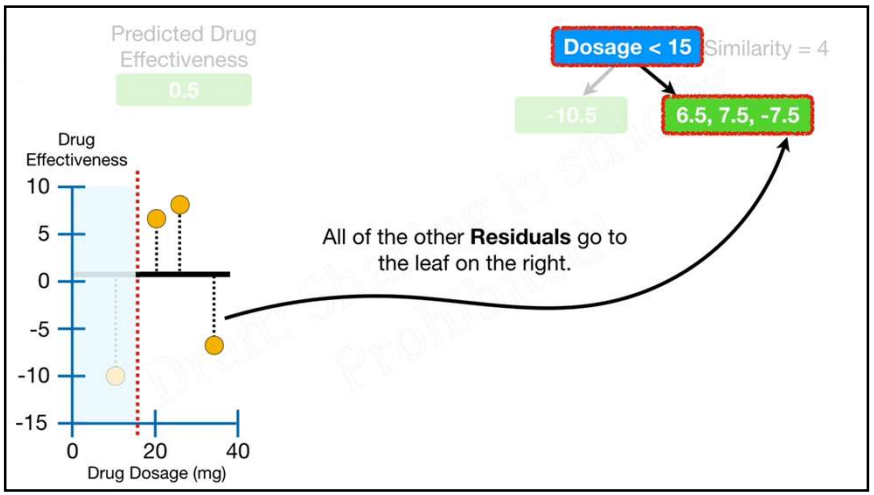

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Similarity = 4
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Now the question is whether or not we can do a better job clustering similar Residuals if we split them into two groups.

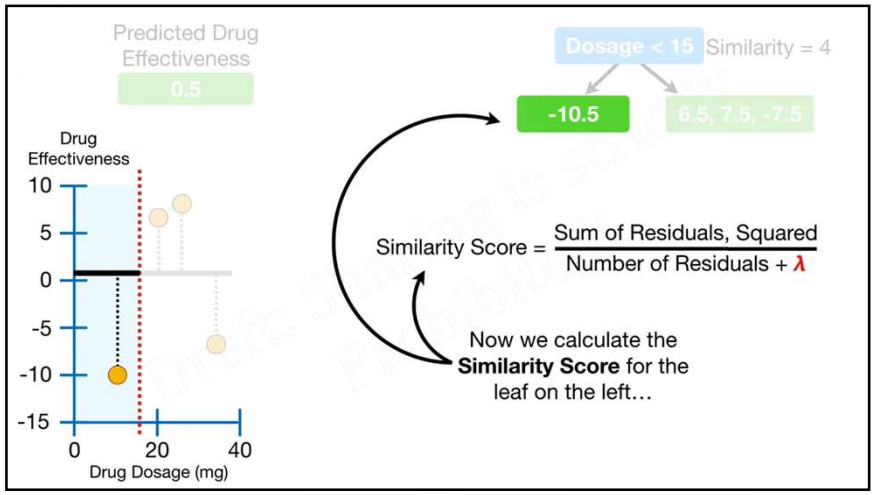

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Similarity = 4
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Their average Dosage is 15, and that corresponds to this dotted red line.

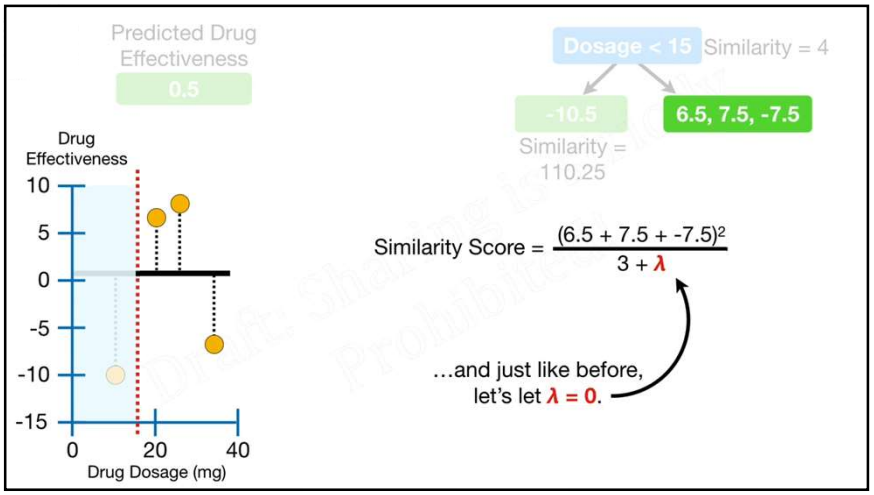

Predicted Drug Effectiveness
0.5
Dosage < 15
Similarity = 4
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
So we split the observations into two groups, based on whether or not the Dosage < 15.

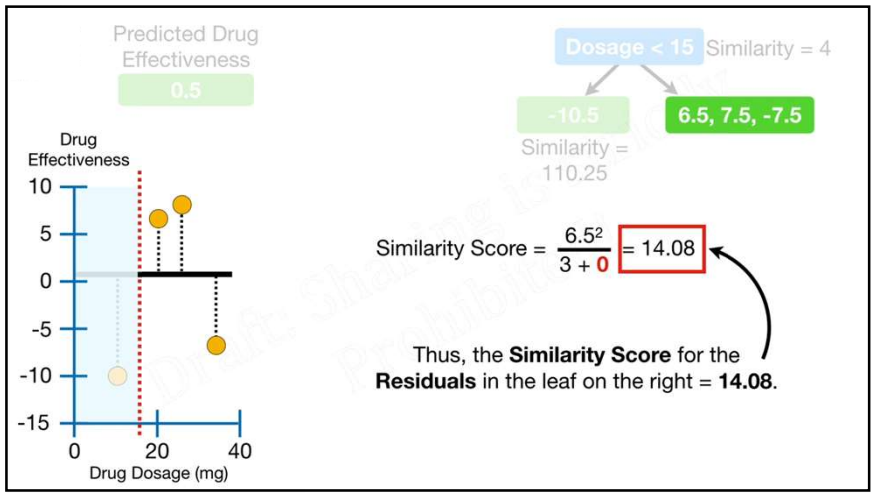

Predicted Drug Effectiveness
0.5
Dosage < 15
Similarity = 4
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
The observation on the far left is the only one with Dosage < 15…

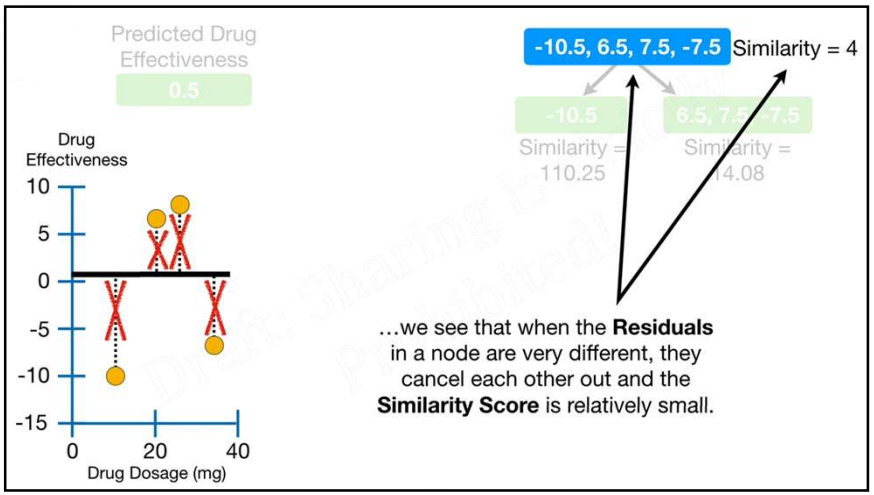

Predicted Drug Effectiveness
0.5
Dosage < 15
Similarity = 4
-10.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…so its Residual goes to the leaf on the left.

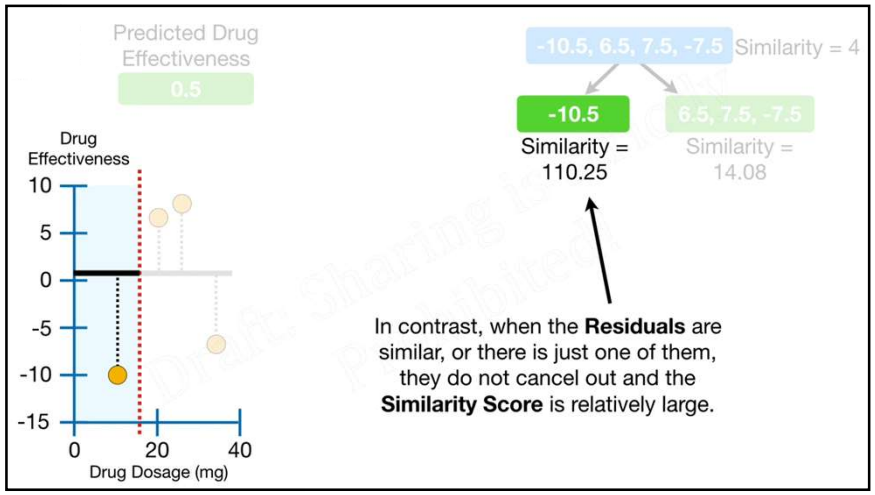

Predicted Drug Effectiveness
0.5
Dosage < 15
Similarity = 4
-10.5
6.5, 7.5, -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
All of the other Residuals go to the leaf on the right.

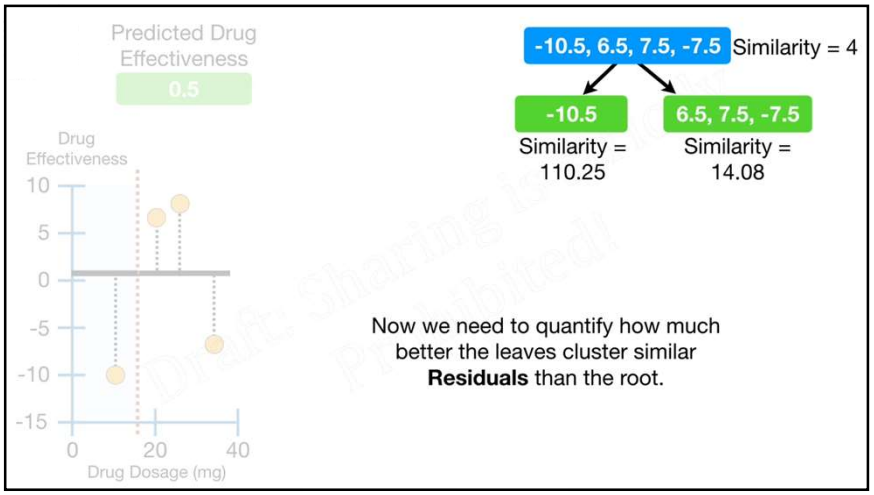

Predicted Drug Effectiveness
0.5
Dosage < 15
Similarity = 4
-10.5
6.5, 7.5, -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Similarity Score = Sum of Residuals, Squared / Number of Residuals + λ
Now we calculate the Similarity Score for the leaf on the left…

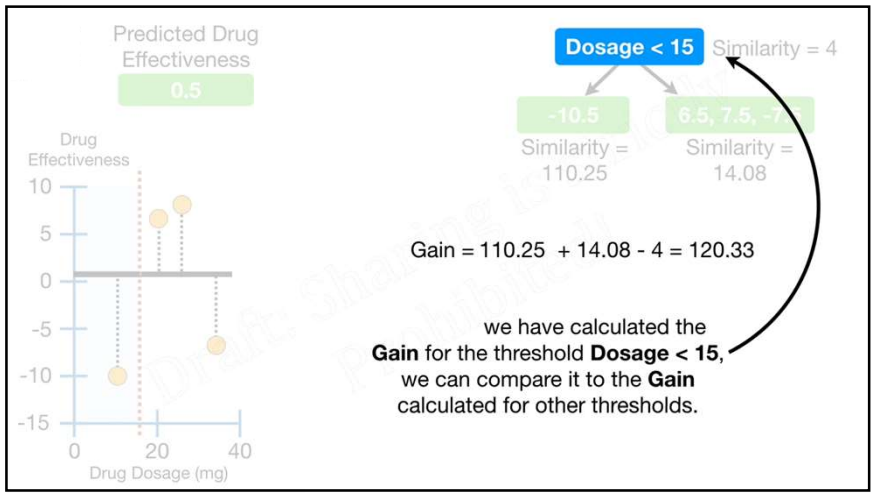

Predicted Drug Effectiveness
0.5
Dosage < 15
Similarity = 4
-10.5
6.5, 7.5, -7.5
Similarity = 110.25
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Similarity Score = (6.5 + 7.5 + -7.5)² / 3 + λ
…and just like before, let's let λ = 0.

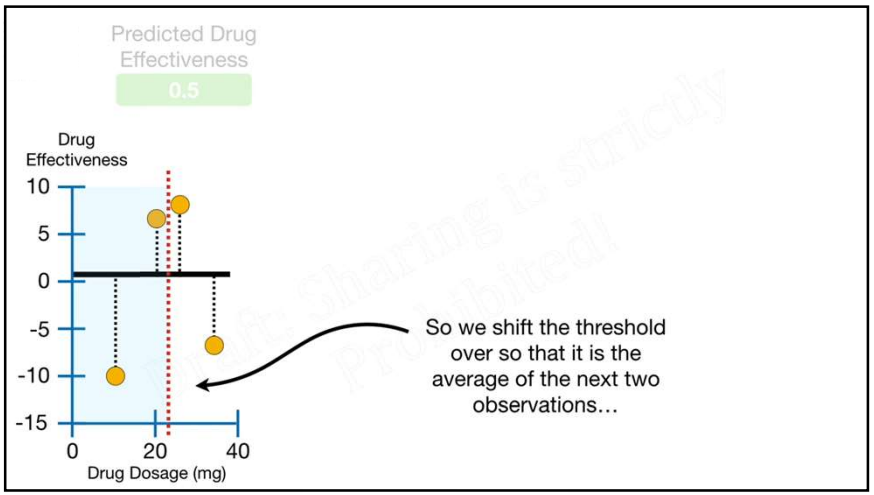

Predicted Drug Effectiveness
0.5
Dosage < 15
Similarity = 4
-10.5
6.5, 7.5, -7.5
Similarity = 110.25
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Similarity Score = 6.5² / 3 + 0 = 14.08
Thus, the Similarity Score for the Residuals in the leaf on the right = 14.08.

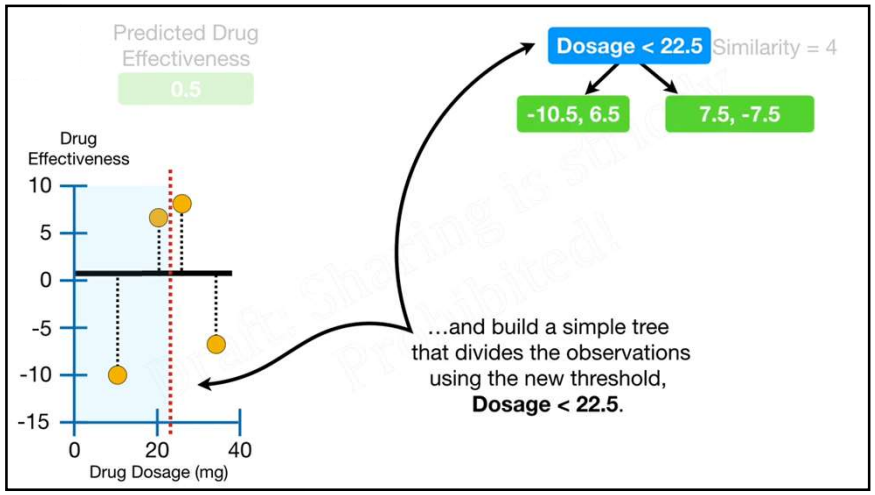

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Similarity = 4
-10.5
Similarity = 110.25
Similarity = 14.08
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…we see that when the Residuals in a node are very different, they cancel each other out and the Similarity Score is relatively small.

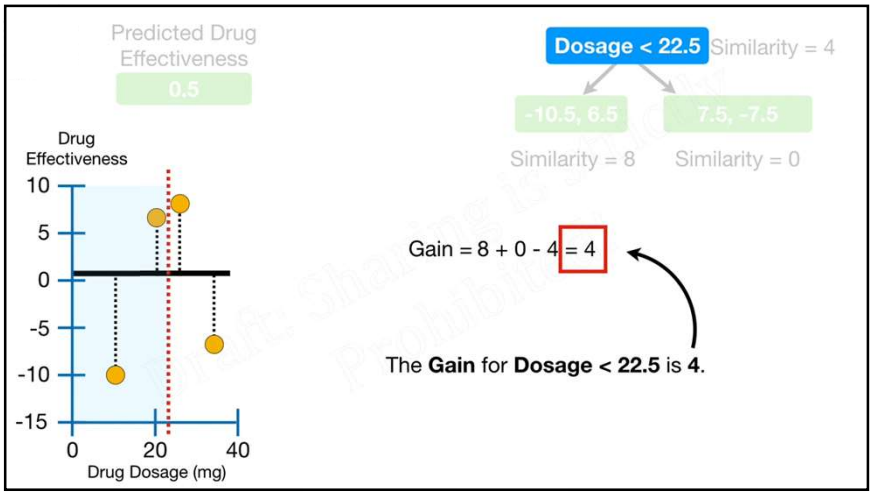

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
-10.5, 6.5, 7.5, -7.5
Similarity = 4
-10.5
Similarity = 110.25
6.5, 7.5, -7.5
Similarity = 14.08
In contrast, when the **Residuals** are similar, or there is just one of them, they do not cancel out and the **Similarity Score** is relatively large.

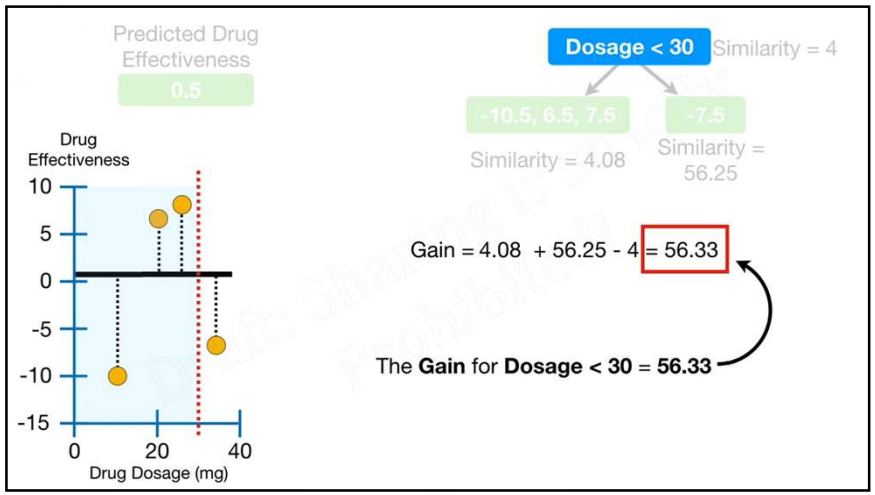

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
-10.5, 6.5, 7.5, -7.5
Similarity = 4
-10.5
Similarity = 110.25
6.5, 7.5, -7.5
Similarity = 14.08
Now we need to quantify how much better the leaves cluster similar **Residuals** than the root.

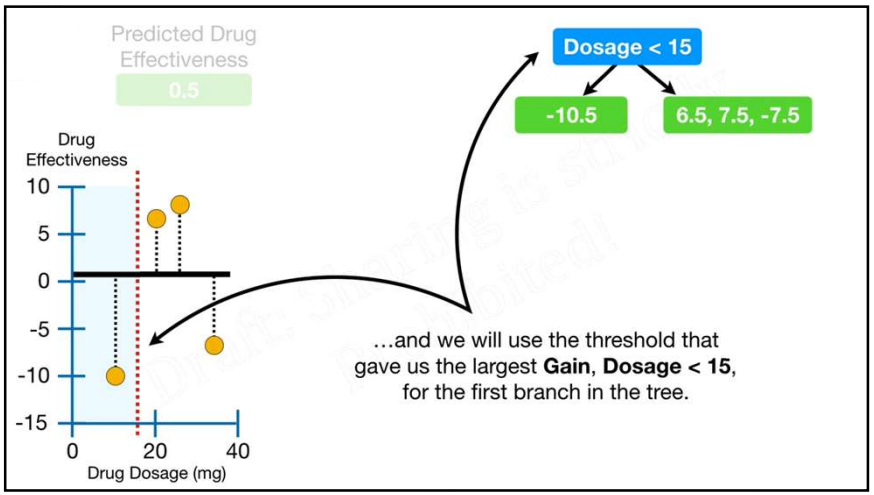

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
Similarity = 4
-10.5
Similarity = 110.25
6.5, 7.5, -7.5
Similarity = 14.08
Gain = 110.25 + 14.08 - 4 = 120.33
we have calculated the **Gain** for the threshold **Dosage < 15**, we can compare it to the **Gain** calculated for other thresholds.

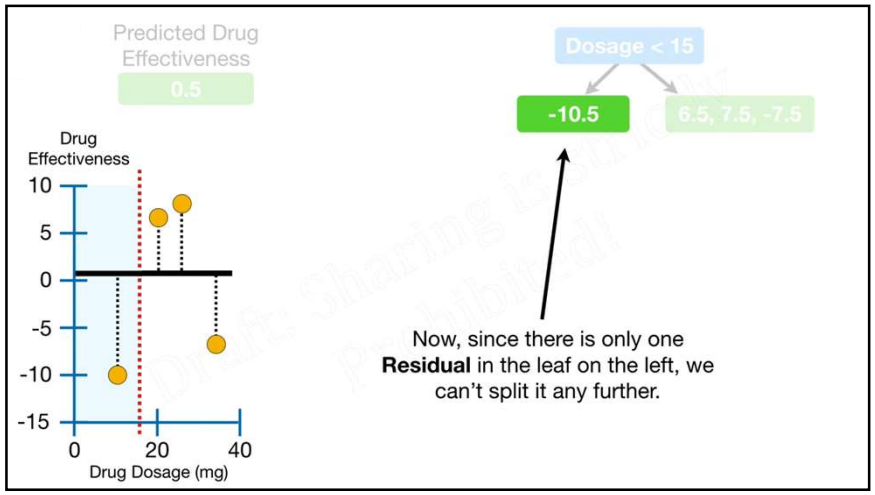

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
So we shift the threshold over so that it is the average of the next two observations…

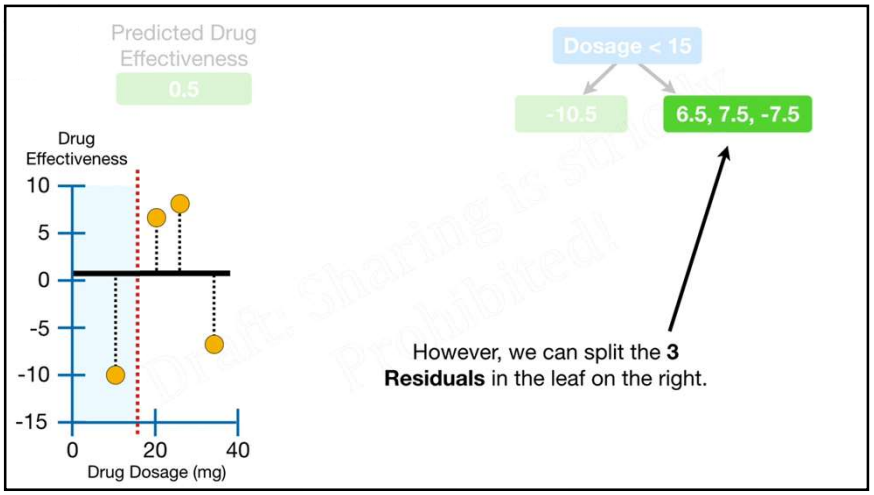

Predicted Drug Effectiveness
0.5
Dosage < 22.5
Similarity = 4
-10.5, 6.5
7.5, -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
...and build a simple tree that divides the observations using the new threshold, Dosage < 22.5.

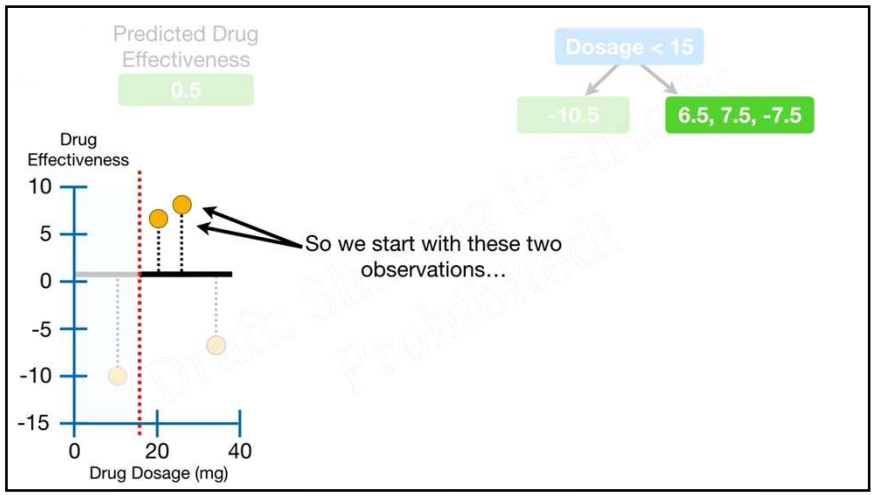

Predicted Drug Effectiveness
0.5
Dosage < 22.5
Similarity = 4
-10.5, 6.5
7.5, -7.5
Similarity = 8
Similarity = 0
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Gain = 8 + 0 - 4 = 4
The Gain for Dosage < 22.5 is 4.

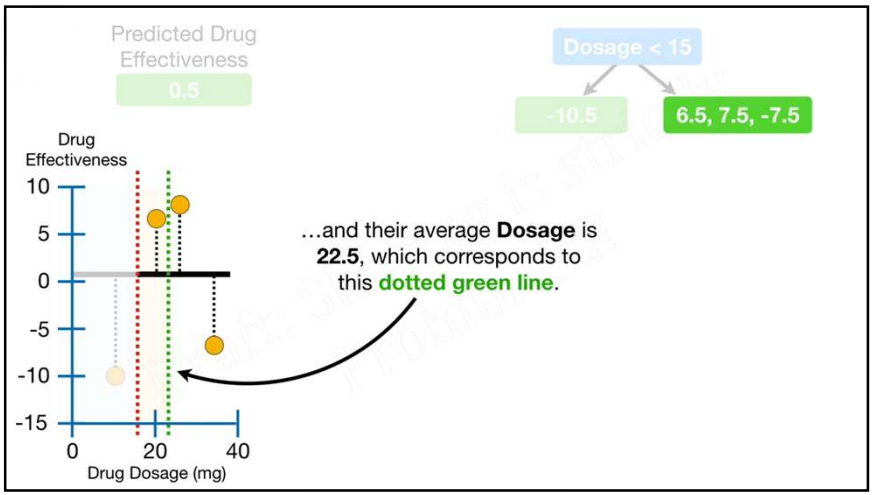

Predicted Drug Effectiveness
0.5
Dosage < 30
Similarity = 4
-10.5, 6.5, 7.5
-7.5
Similarity = 4.08
Similarity = 56.25
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Gain = 4.08 + 56.25 - 4 = 56.33
The Gain for Dosage < 30 = 56.33

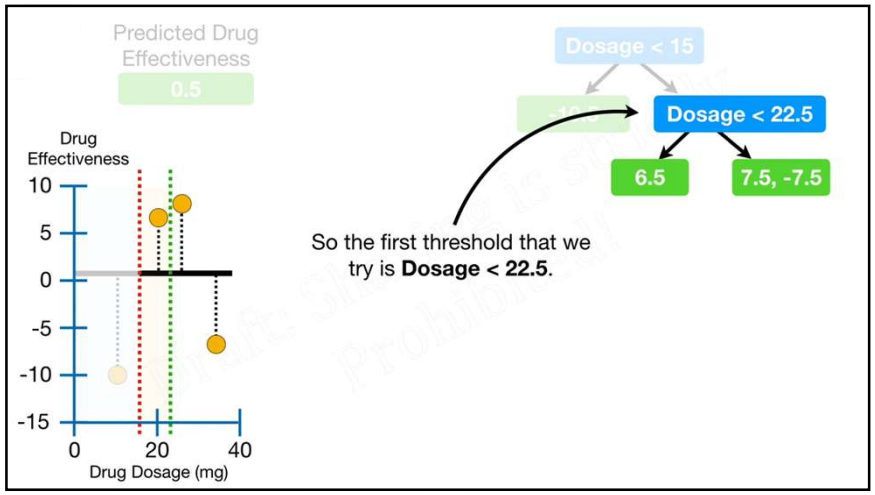

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
6.5, 7.5, -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
...and we will use the threshold that gave us the largest Gain, Dosage < 15, for the first branch in the tree.

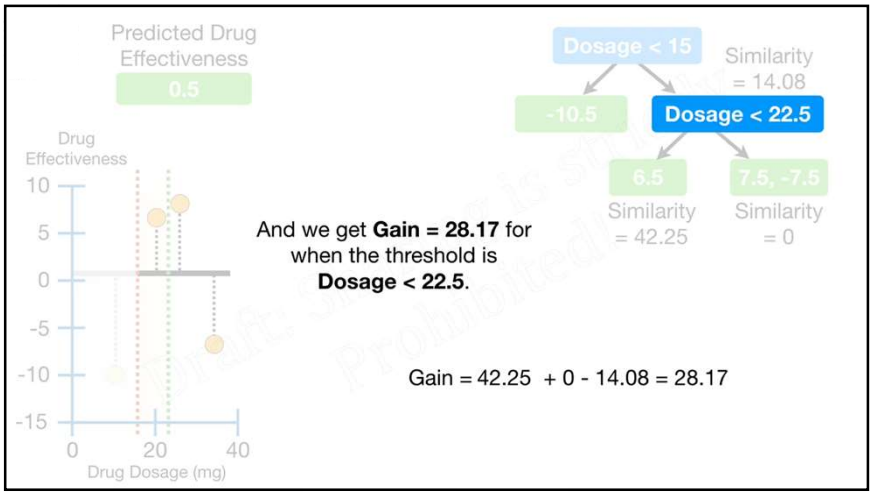

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
6.5, 7.5, -7.5
Now, since there is only one **Residual** in the leaf on the left, we can't split it any further.

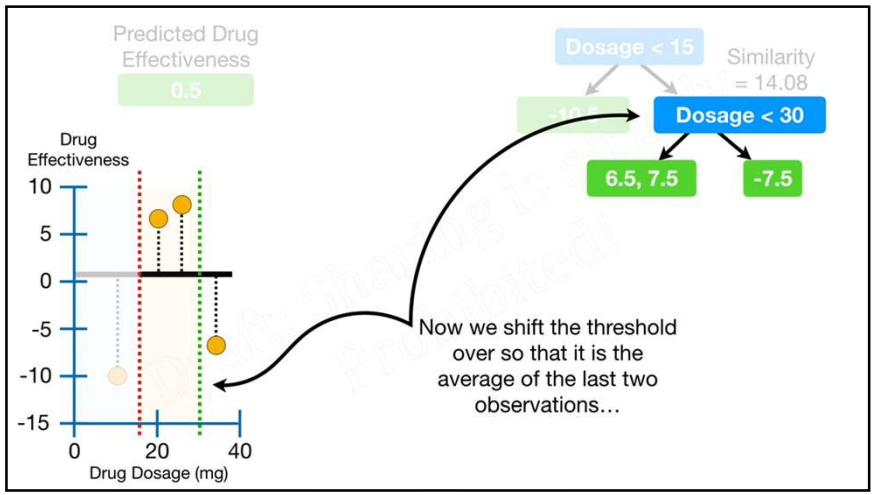

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
6.5, 7.5, -7.5
However, we can split the **3 Residuals** in the leaf on the right.

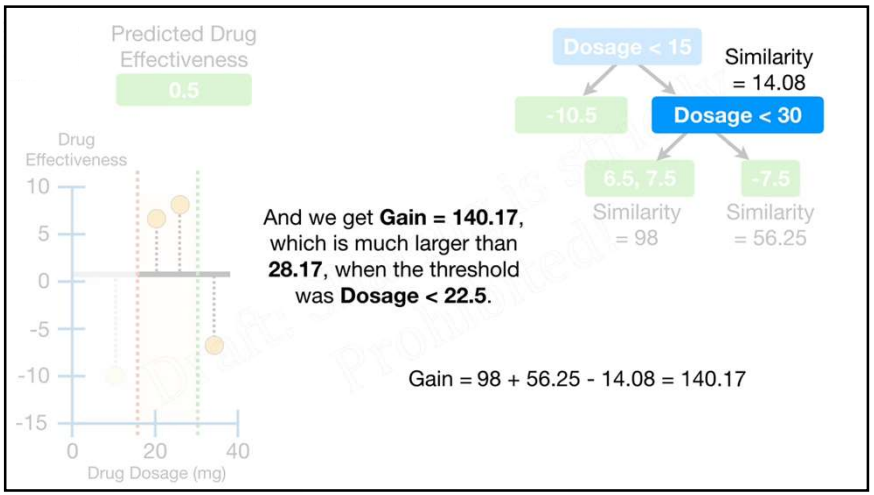

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
6.5, 7.5, -7.5
So we start with these two observations…

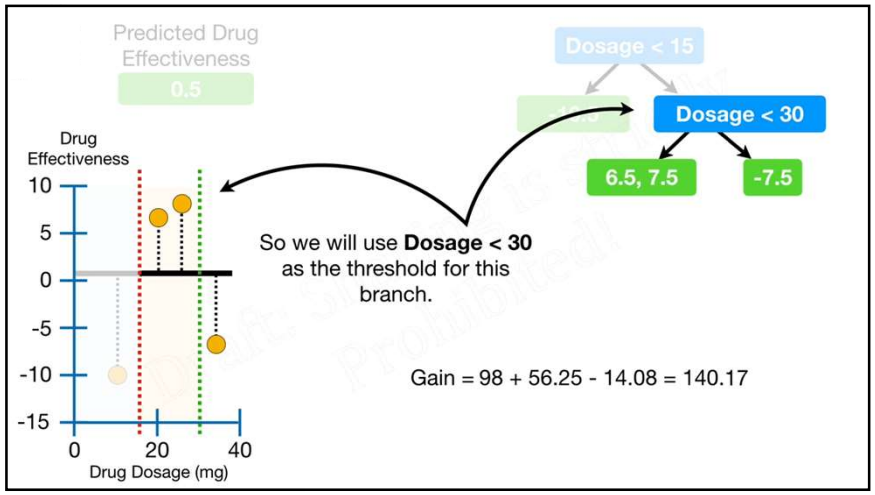

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
6.5, 7.5, -7.5
…and their average **Dosage** is **22.5**, which corresponds to this **dotted green line**.


Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
Dosage < 22.5
6.5
7.5, -7.5
So the first threshold that we try is Dosage < 22.5.

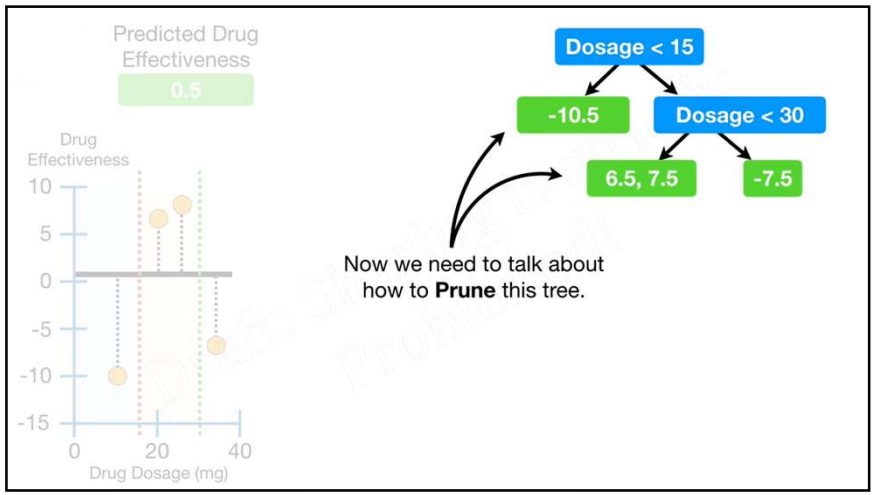

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
Similarity = 14.08
-10.5
Dosage < 22.5
6.5
7.5, -7.5
Similarity = 42.25
Similarity = 0
And we get Gain = 28.17 for when the threshold is Dosage < 22.5.
Gain = 42.25 + 0 - 14.08 = 28.17

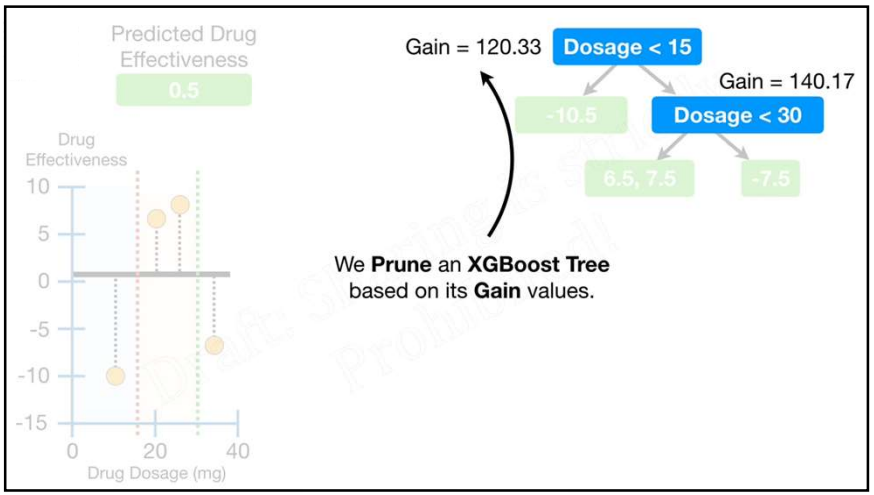

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
Similarity = 14.08
Dosage < 30
6.5, 7.5
-7.5
Now we shift the threshold over so that it is the average of the last two observations…

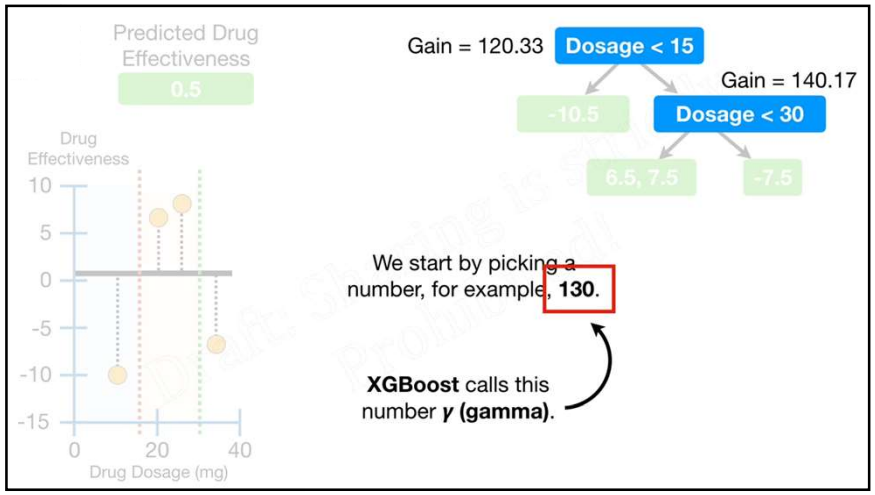

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
Similarity = 14.08
-10.5
Dosage < 30
6.5, 7.5
-7.5
Similarity = 98
Similarity = 56.25
And we get Gain = 140.17, which is much larger than 28.17, when the threshold was Dosage < 22.5.
Gain = 98 + 56.25 - 14.08 = 140.17

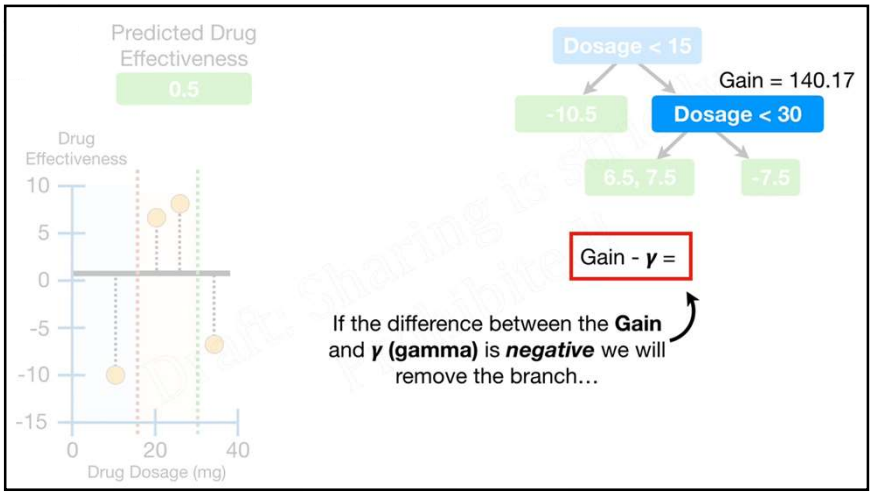

Predicted Drug Effectiveness
0.5
Dosage < 15
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
So we will use **Dosage < 30** as the threshold for this branch.
Gain = 98 + 56.25 - 14.08 = 140.17

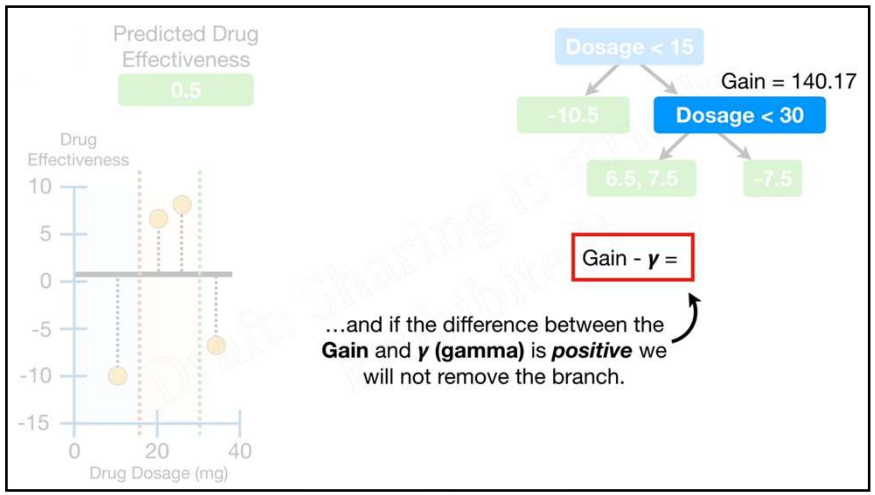

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
We stop here,
However, the default is to allow up to **6** levels.

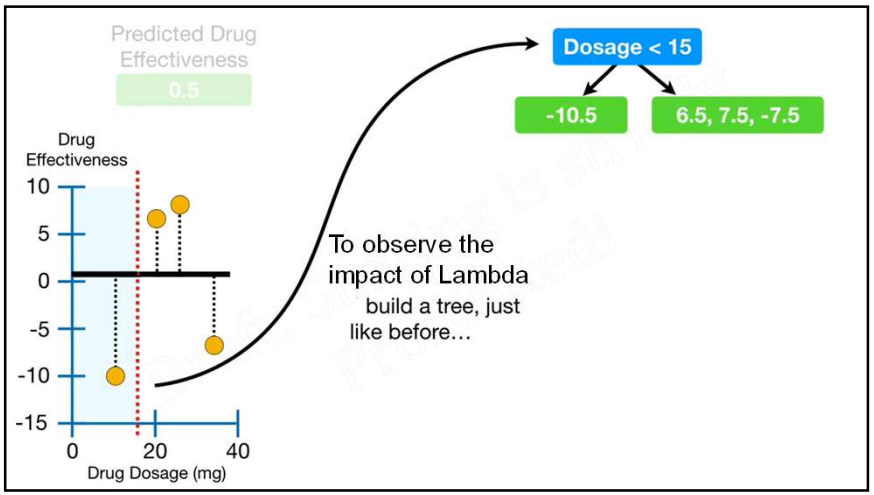

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Now we need to talk about how to **Prune** this tree.

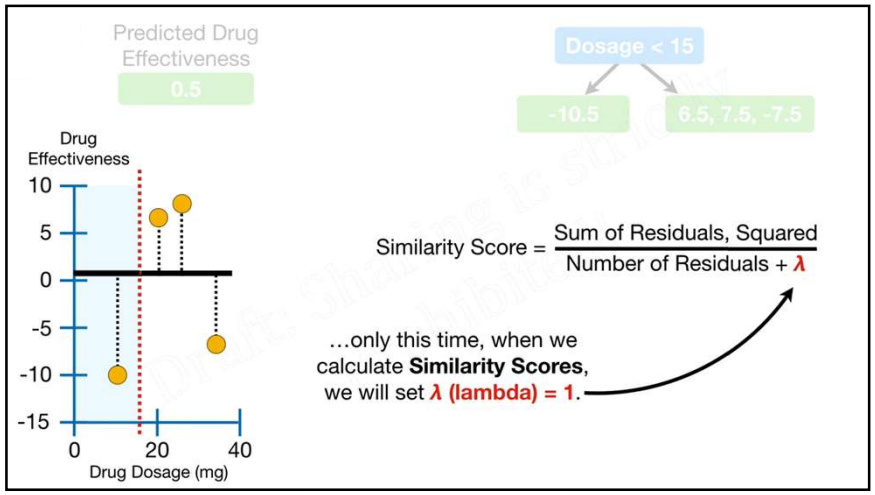

Predicted Drug Effectiveness
0.5
Gain = 120.33
Dosage < 15
Gain = 140.17
-10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
We **Prune** an **XGBoost Tree** based on its **Gain** values.

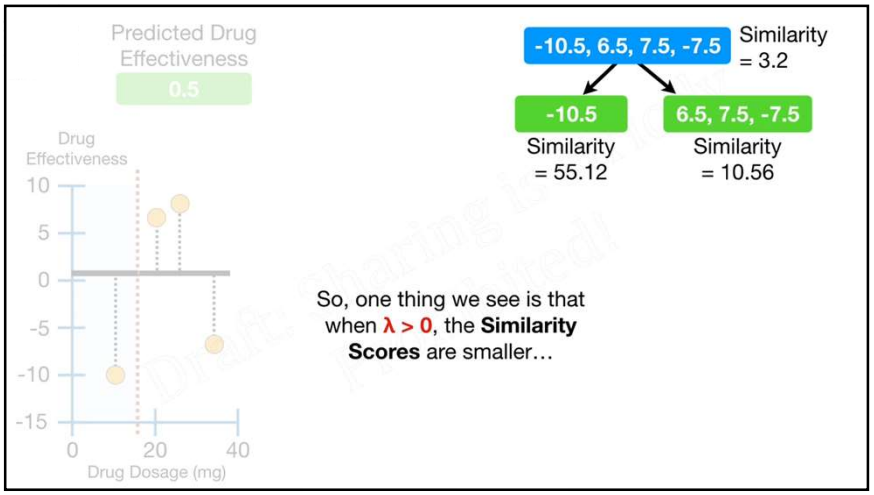

Predicted Drug Effectiveness
0.5
Gain = 120.33
Dosage < 15
-10.5
Gain = 140.17
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
We start by picking a number, for example, 130.
XGBoost calls this number γ (gamma).

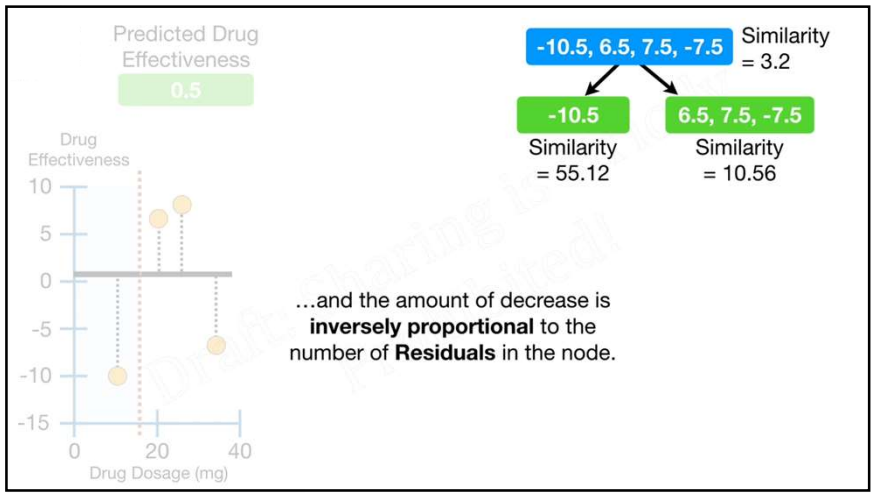

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Gain = 140.17
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Gain - γ =
If the difference between the Gain and γ (gamma) is negative we will remove the branch…

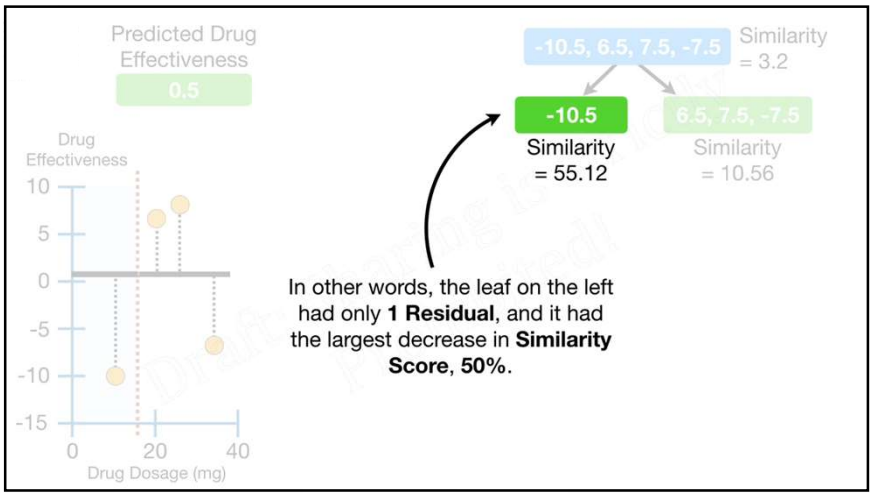

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Gain = 140.17
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Gain - γ =
…and if the difference between the Gain and γ (gamma) is positive we will not remove the branch.

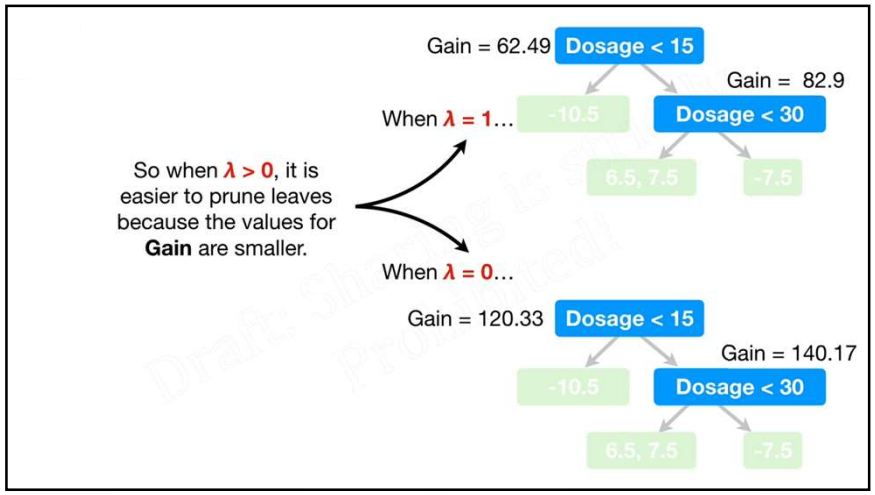

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
6.5, 7.5, -7.5
To observe the impact of Lambda build a tree, just like before…

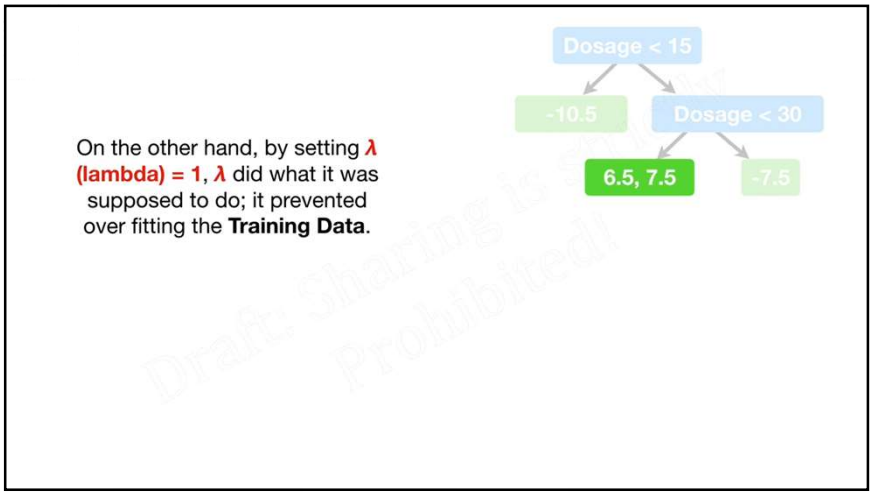

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
6.5, 7.5, -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Similarity Score = Sum of Residuals, Squared / Number of Residuals + λ
…only this time, when we calculate Similarity Scores, we will set λ (lambda) = 1.

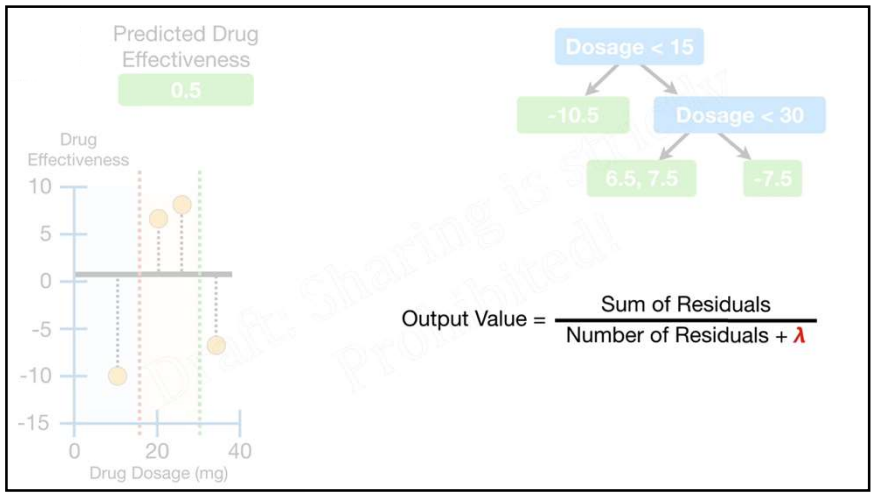

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Similarity = 3.2
-10.5
Similarity = 55.12
6.5, 7.5, -7.5
Similarity = 10.56
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
So, one thing we see is that when λ > 0, the Similarity Scores are smaller…

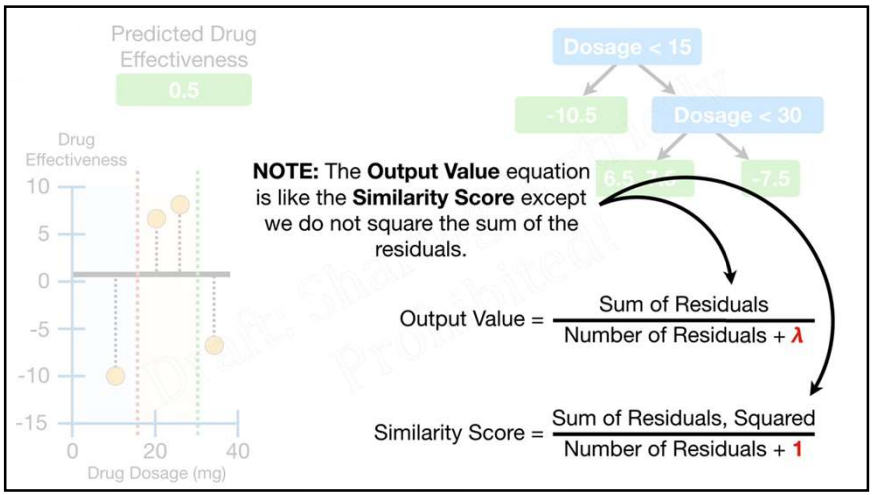

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Similarity = 3.2
-10.5
Similarity = 55.12
6.5, 7.5, -7.5
Similarity = 10.56
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…and the amount of decrease is inversely proportional to the number of Residuals in the node.

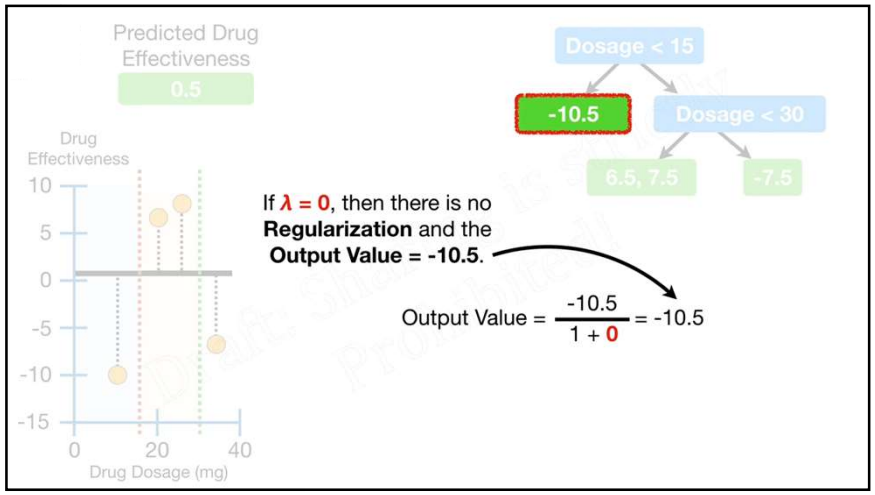

Predicted Drug Effectiveness
0.5
-10.5, 6.5, 7.5, -7.5
Similarity = 3.2
-10.5
Similarity = 55.12
6.5, 7.5, -7.5
Similarity = 10.56
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
In other words, the leaf on the left had only 1 Residual, and it had the largest decrease in Similarity Score, 50%.

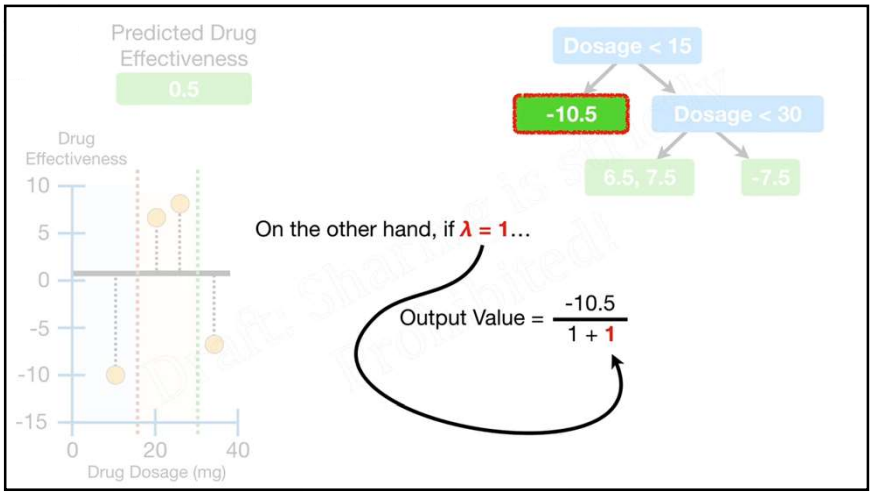

Gain = 62.49
Dosage < 15
Gain = 82.9
When λ = 1…
-10.5
Dosage < 30
6.5, 7.5
-7.5
So when λ > 0, it is easier to prune leaves because the values for Gain are smaller.
When λ = 0…
Gain = 120.33
Dosage < 15
Gain = 140.17
-10.5
Dosage < 30
6.5, 7.5
-7.5

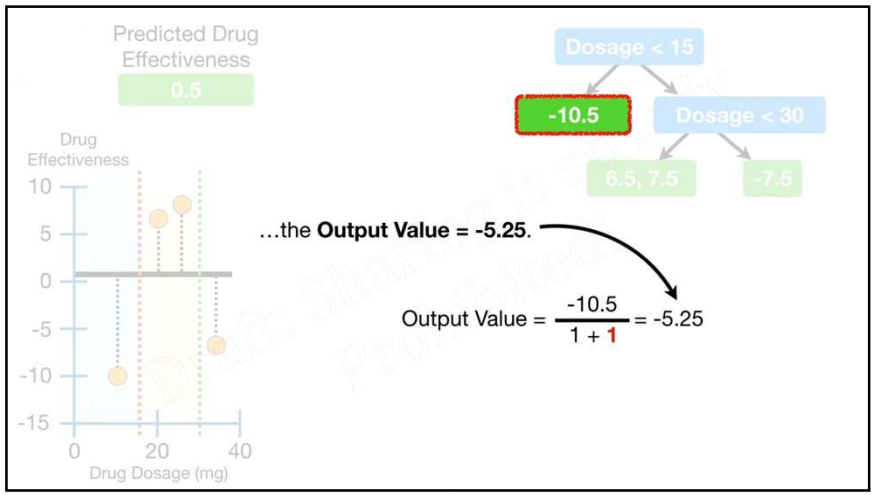

On the other hand, by setting λ (lambda) = 1, λ did what it was supposed to do; it prevented over fitting the Training Data.
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5

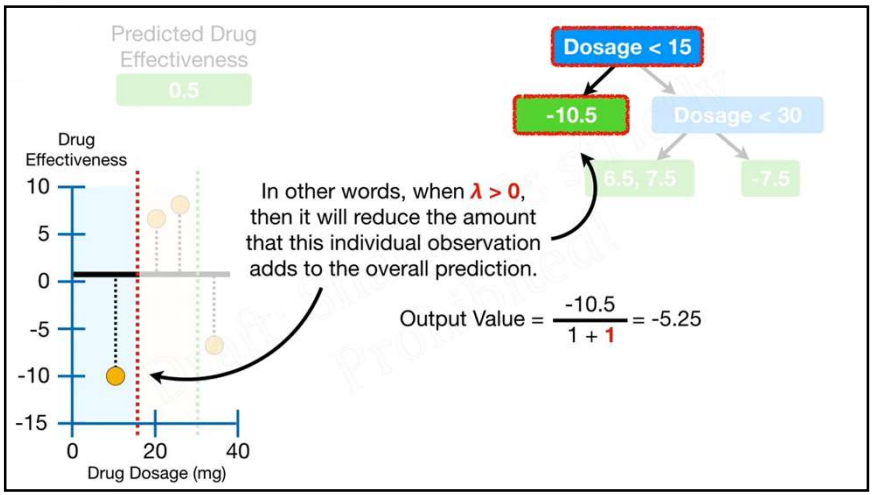

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5
Output Value = Sum of Residuals / Number of Residuals + λ

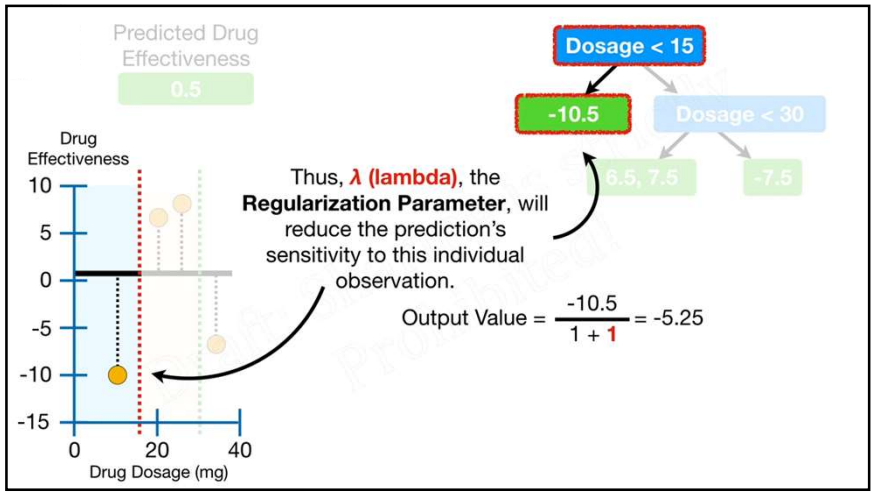

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5
NOTE: The Output Value equation is like the Similarity Score except we do not square the sum of the residuals.
Output Value = Sum of Residuals / Number of Residuals + λ
Similarity Score = Sum of Residuals, Squared / Number of Residuals + 1

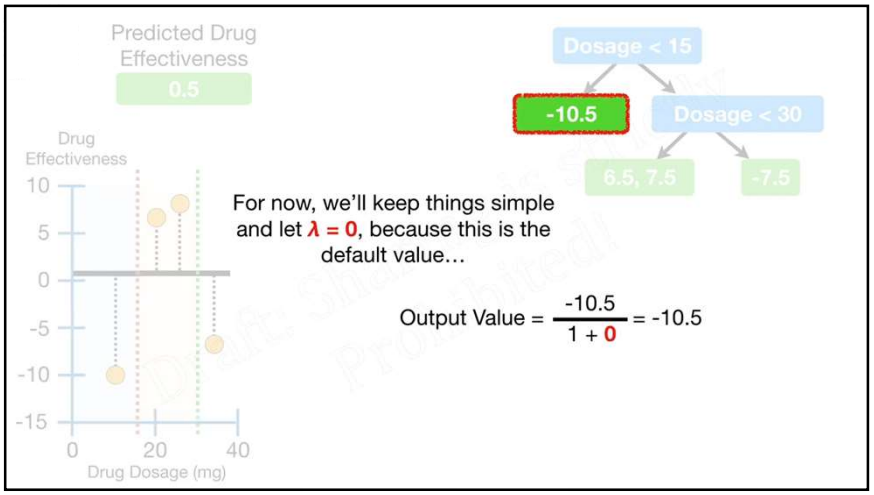

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
If λ = 0, then there is no Regularization and the Output Value = -10.5.
Output Value = -10.5 / 1 + 0 = -10.5

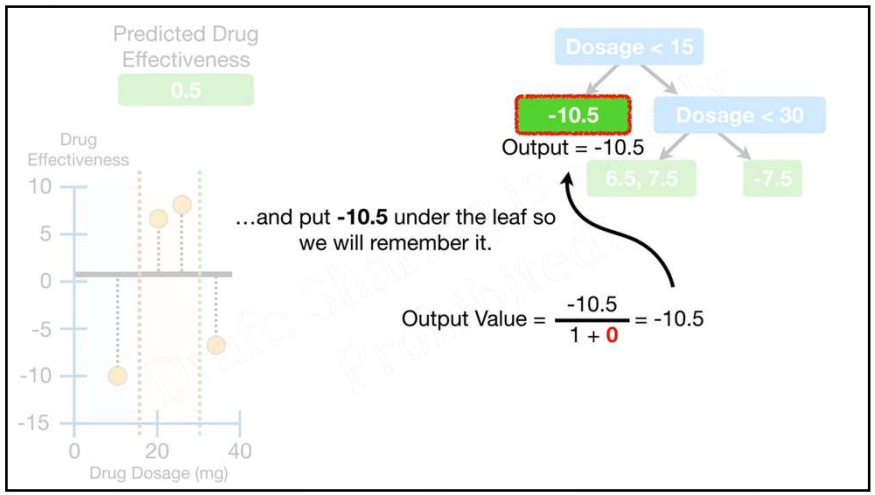

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
On the other hand, if λ = 1…
Output Value = -10.5 / 1 + 1

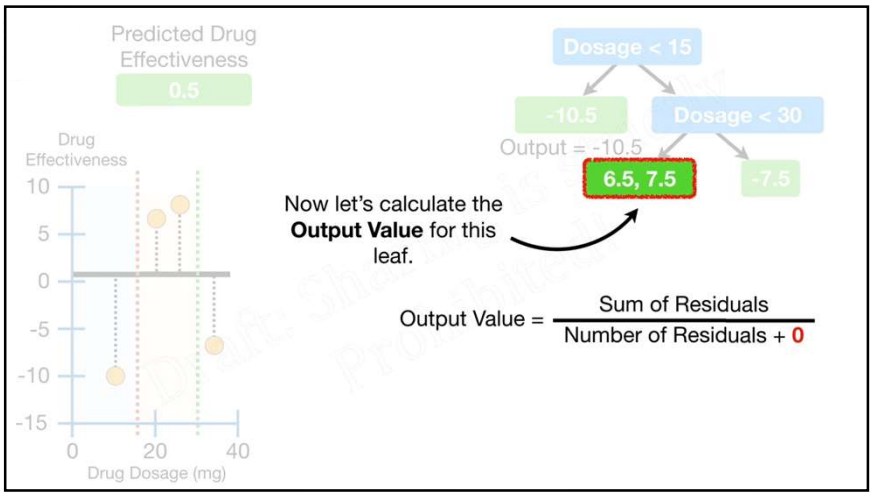

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…the Output Value = -5.25.
Output Value = -10.5 / 1 + 1 = -5.25

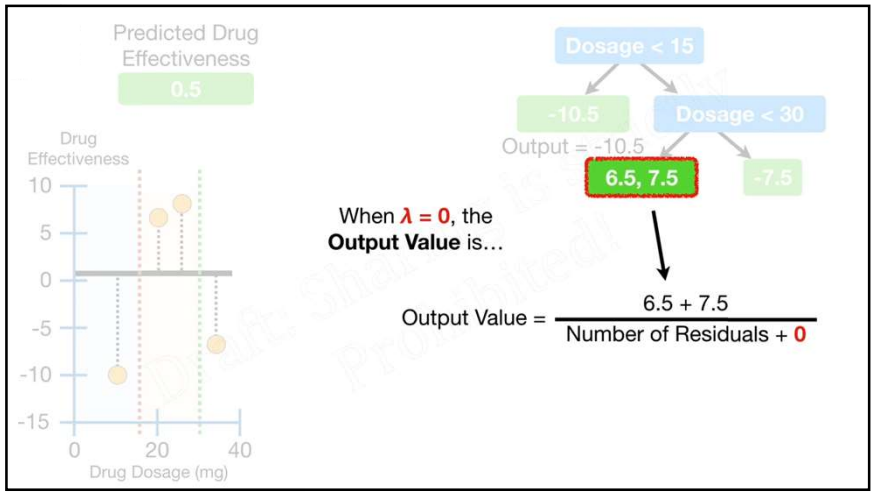

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
In other words, when λ > 0, then it will reduce the amount that this individual observation adds to the overall prediction.
Output Value = -10.5 / 1 + 1 = -5.25

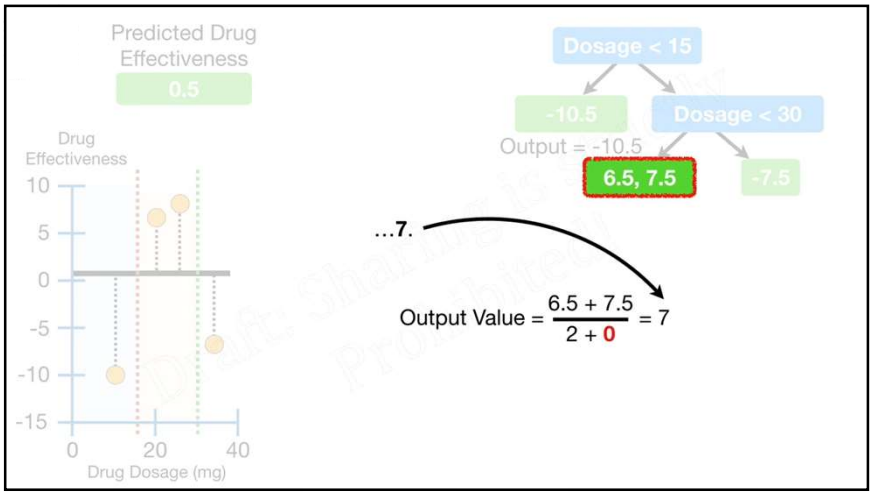

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Thus, λ (lambda), the Regularization Parameter, will reduce the prediction's sensitivity to this individual observation.
Output Value = -10.5 / 1 + 1 = -5.25

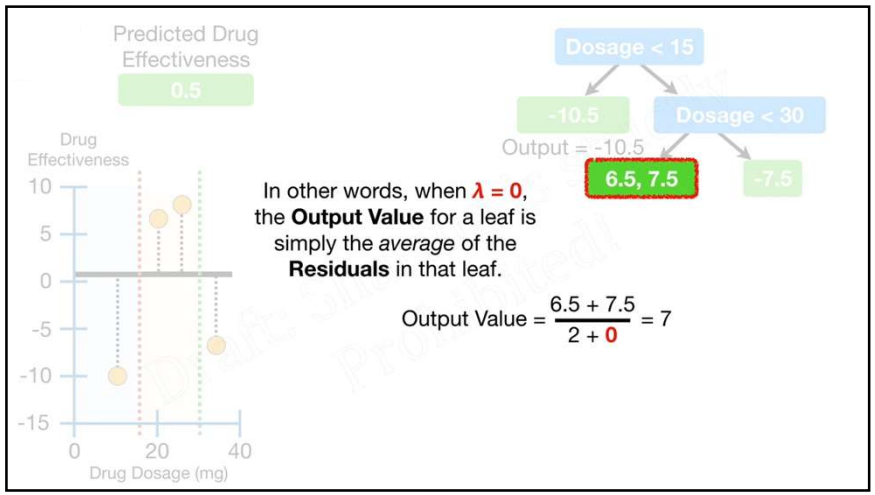

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
For now, we'll keep things simple and let λ = 0, because this is the default value…
Output Value = -10.5 / 1 + 0 = -10.5

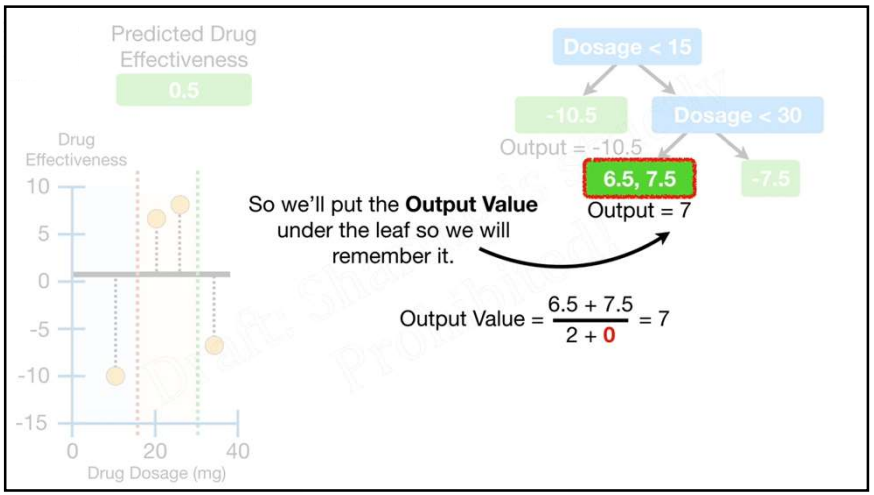

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Output = -10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…and put -10.5 under the leaf so we will remember it.
Output Value = -10.5 / 1 + 0 = -10.5

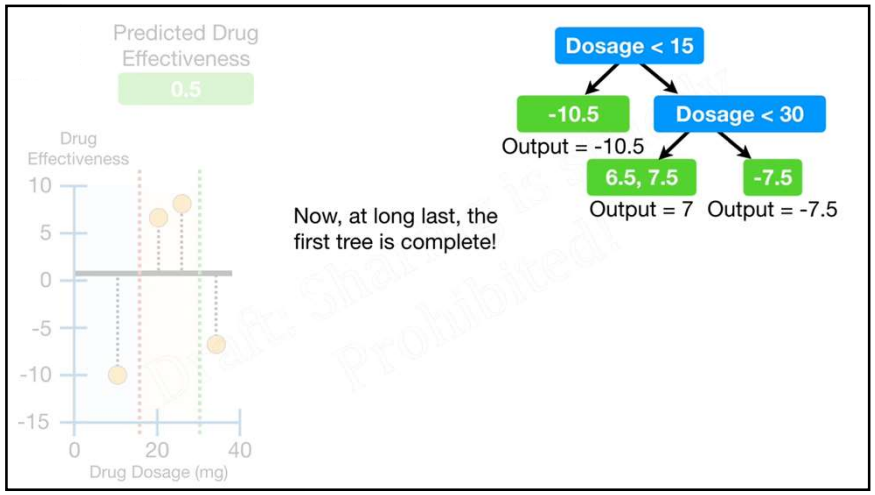

Predicted Drug Effectiveness
0.5
Dosage < 15
-10.5
Output = -10.5
Dosage < 30
6.5, 7.5
-7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Now let's calculate the Output Value for this leaf.
Output Value = Sum of Residuals / Number of Residuals + 0

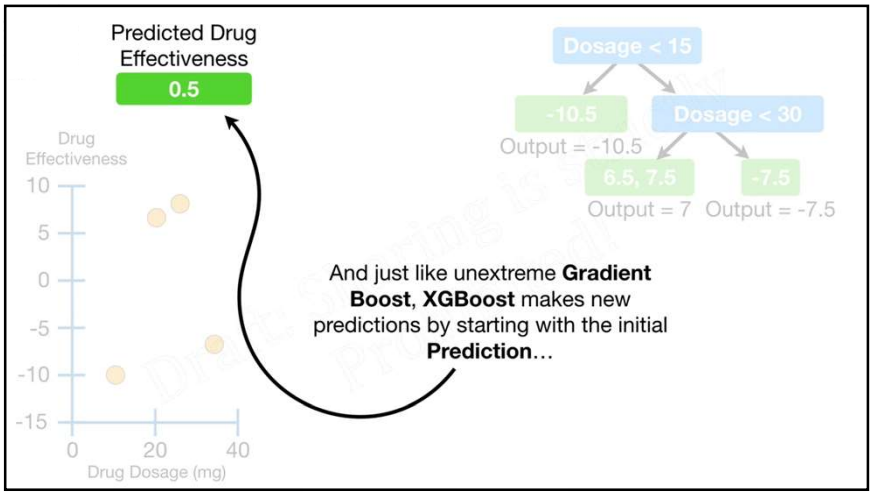

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
When λ = 0, the Output Value is…
Output Value = 6.5 + 7.5 / Number of Residuals + 0

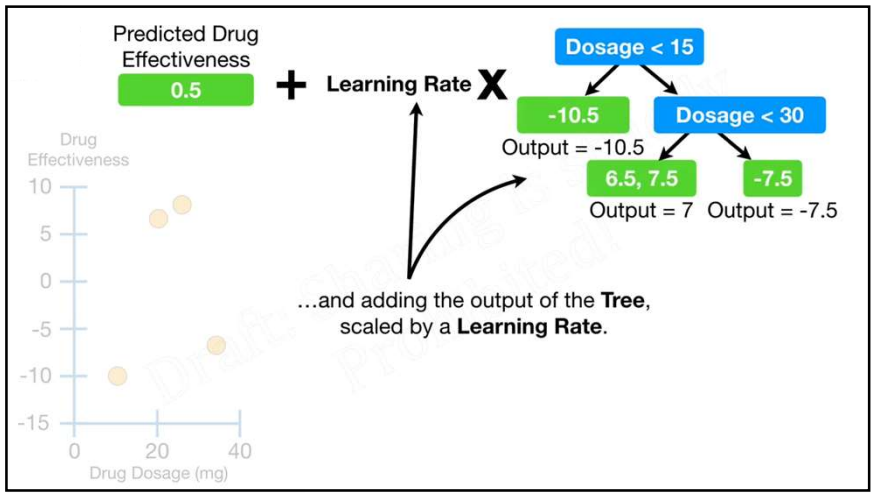

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
…7.
Output Value = 6.5 + 7.5 / 2 + 0 = 7

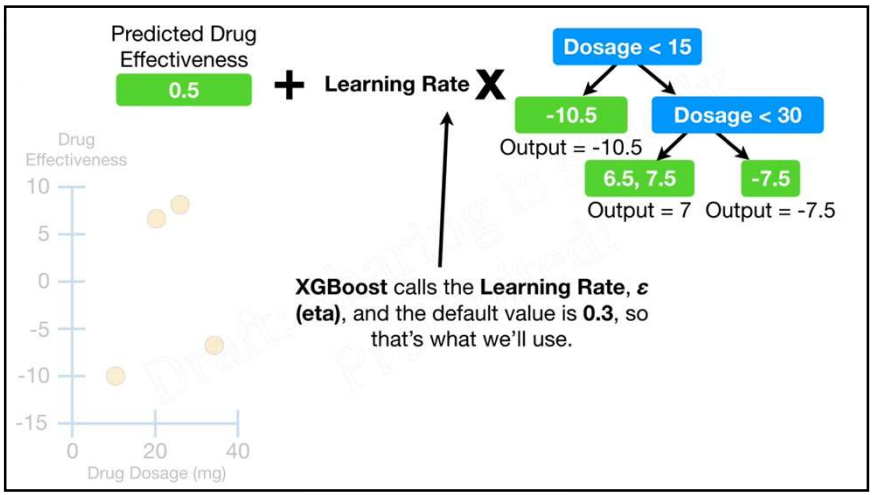

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
In other words, when λ = 0, the Output Value for a leaf is simply the average of the Residuals in that leaf.
Output Value = 6.5 + 7.5 / 2 + 0 = 7

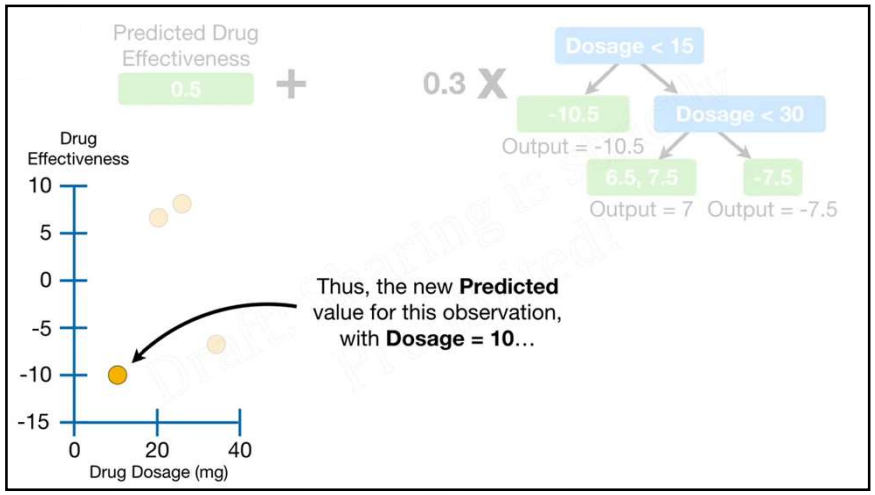

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
So we'll put the Output Value under the leaf so we will remember it.
Output = 7
Output Value = 6.5 + 7.5 / 2 + 0 = 7

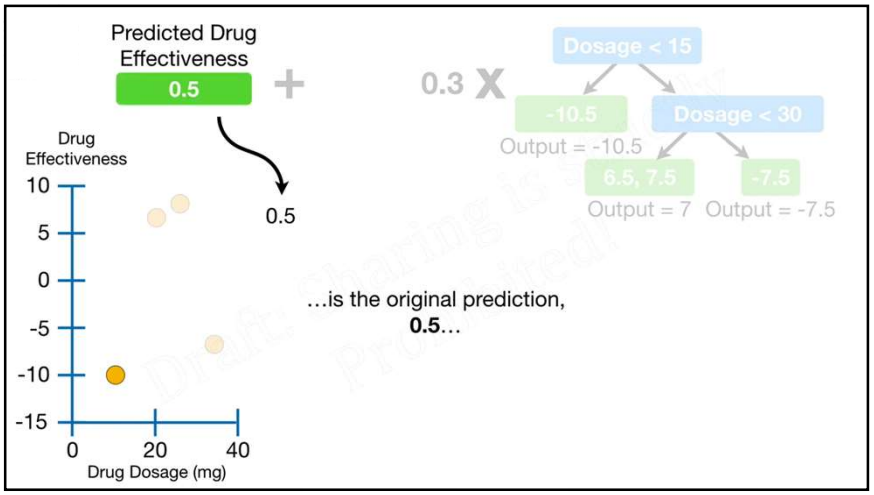

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Now, at long last, the first tree is complete!

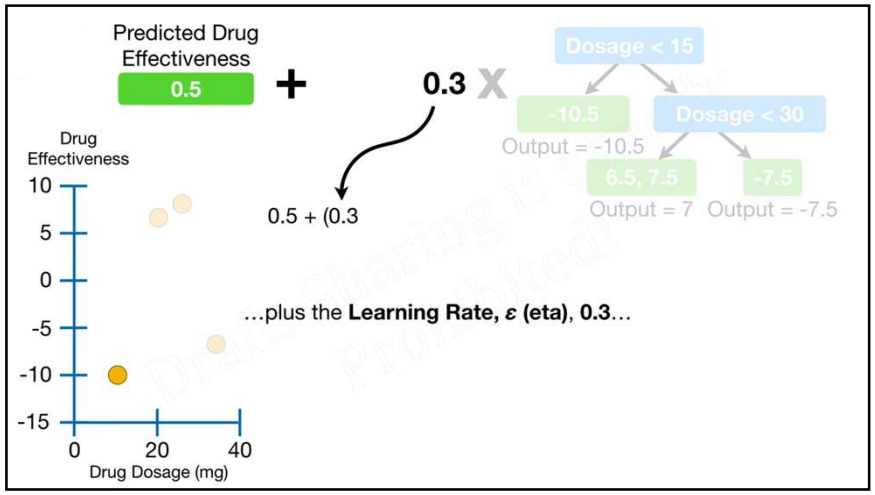

Predicted Drug Effectiveness
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
And just like unextreme **Gradient Boost**, **XGBoost** makes new predictions by starting with the initial **Prediction**…

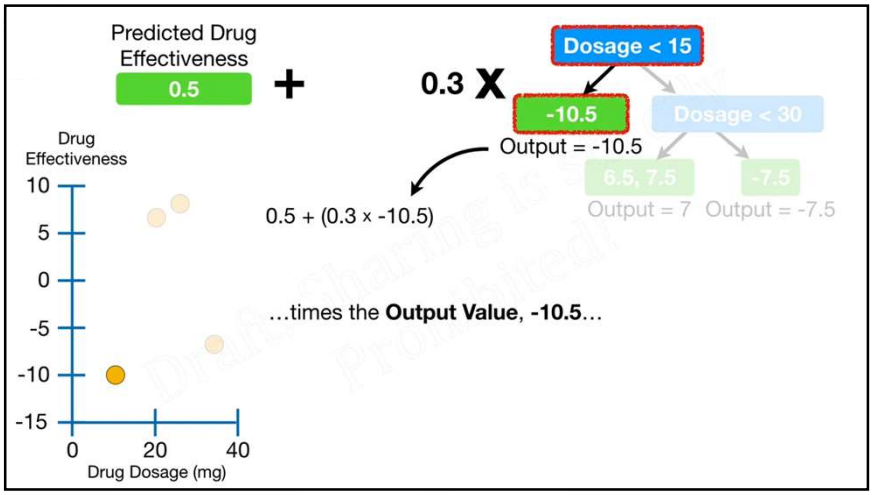

Predicted Drug Effectiveness
0.5
+
Learning Rate
X
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
…and adding the output of the **Tree**, scaled by a **Learning Rate**.

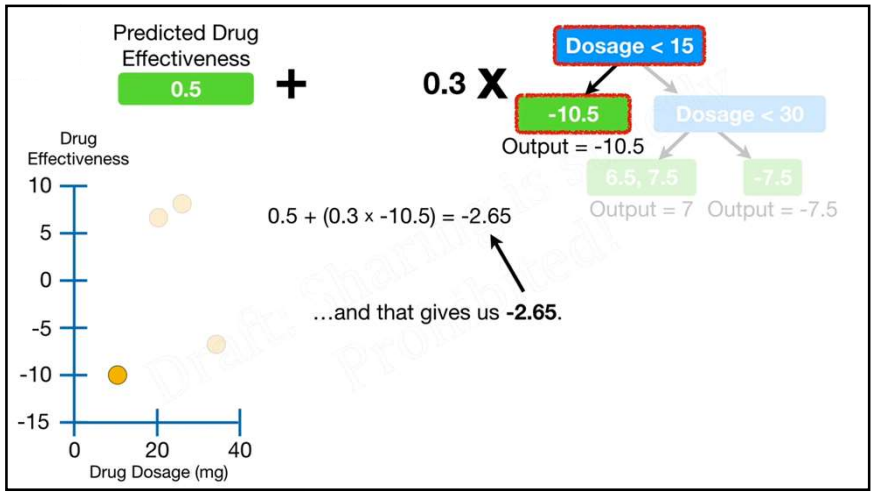

Predicted Drug Effectiveness
0.5
+
Learning Rate
X
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
XGBoost calls the **Learning Rate**, ***ε*** **(eta)**, and the default value is **0.3**, so that's what we'll use.

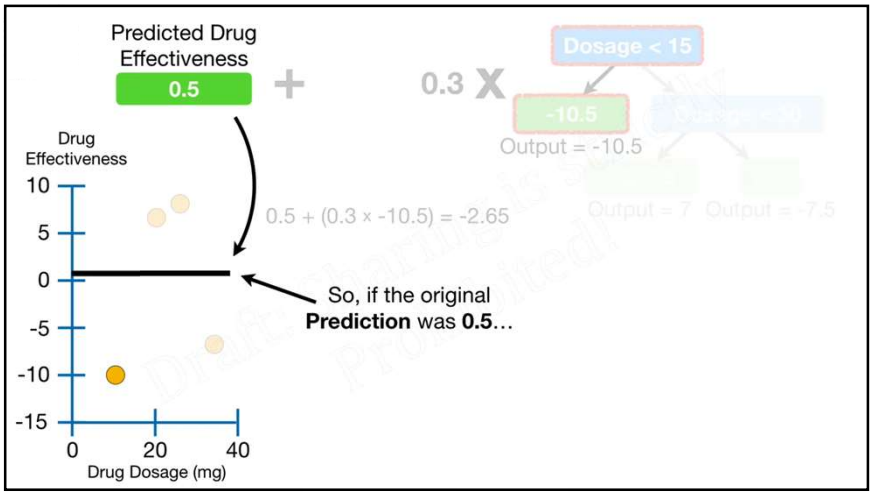

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Thus, the new **Predicted** value for this observation, with **Dosage = 10**…

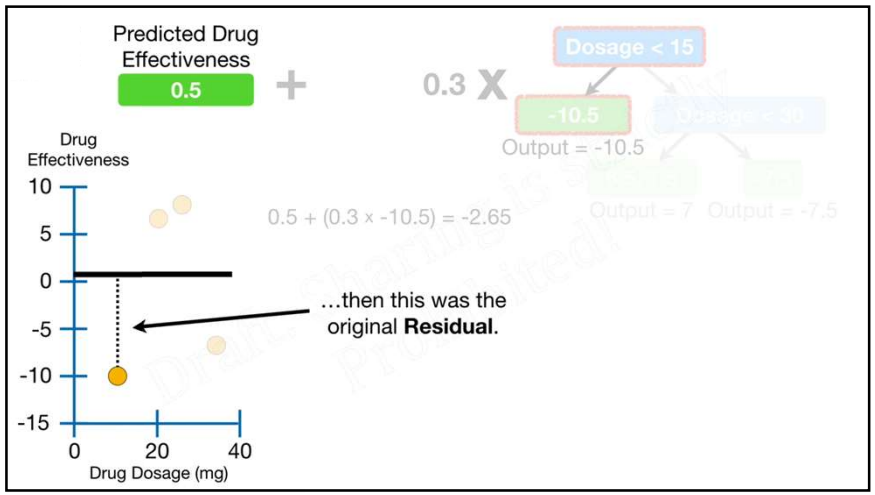

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
0.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…is the original prediction, **0.5**…

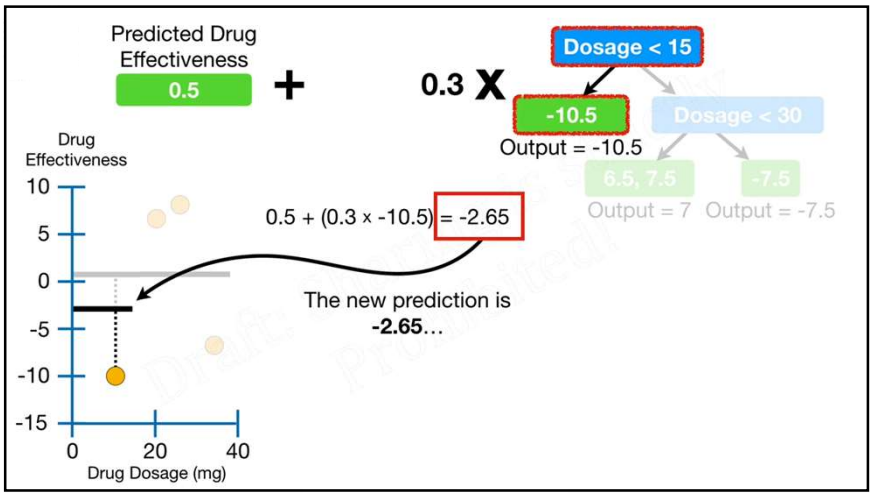

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
0.5 + (0.3
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…plus the **Learning Rate, ε (eta)**, **0.3**…

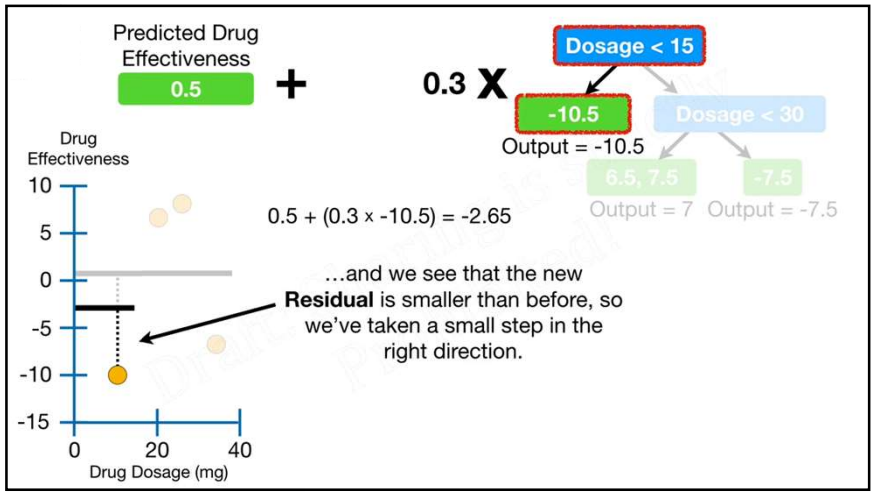

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
0.5 + (0.3 x -10.5)
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…times the **Output Value**, **-10.5**…

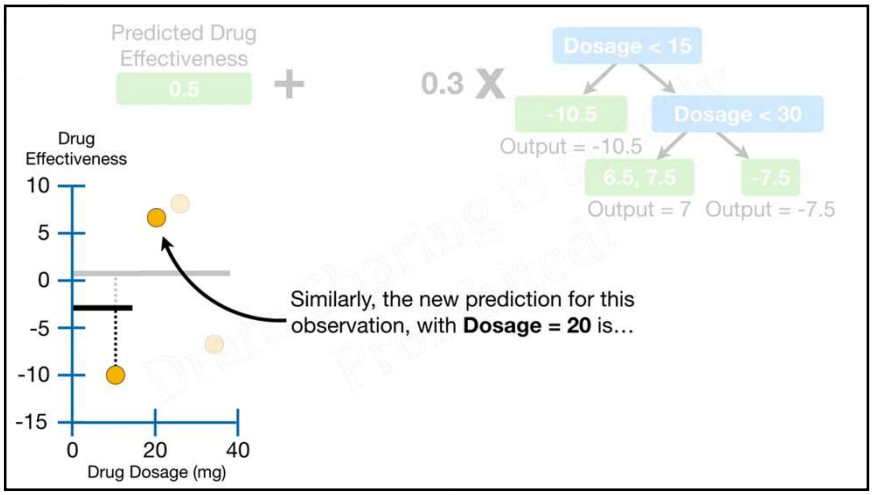

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
0.5 + (0.3 x -10.5) = -2.65
…and that gives us -2.65.

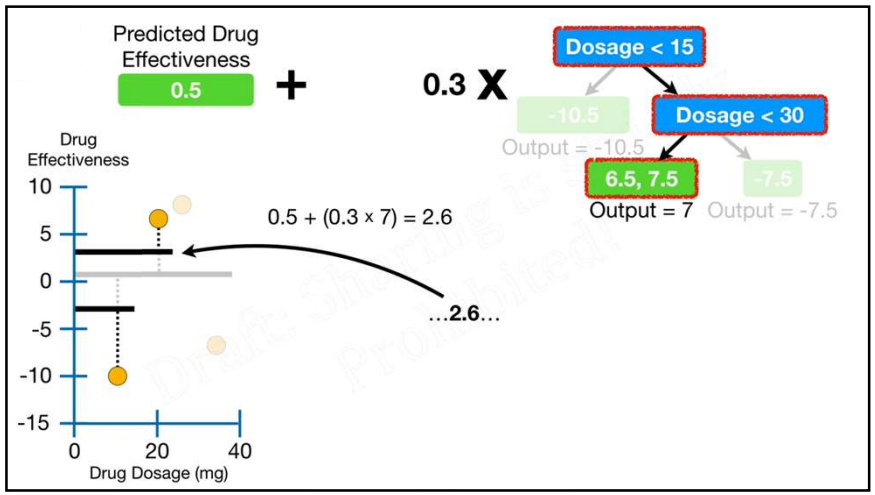

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Output = -10.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
0.5 + (0.3 x -10.5) = -2.65
So, if the original Prediction was 0.5…

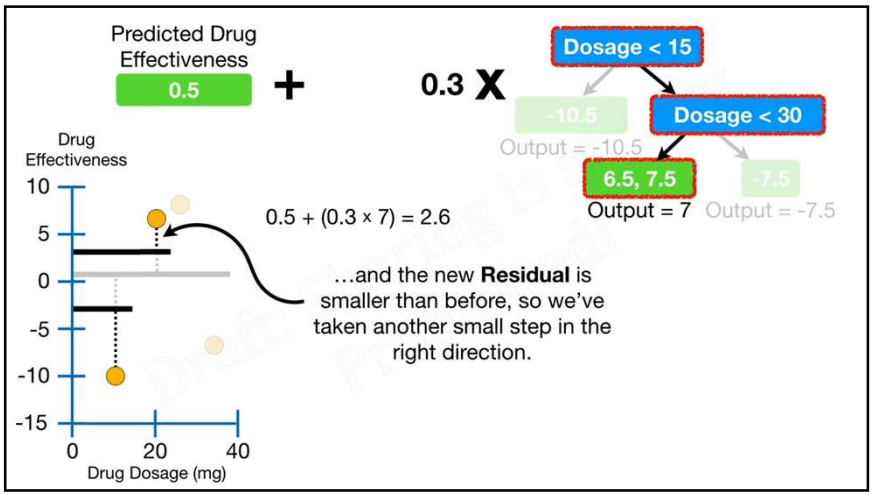

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Output = -10.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
0.5 + (0.3 x -10.5) = -2.65
…then this was the original Residual.

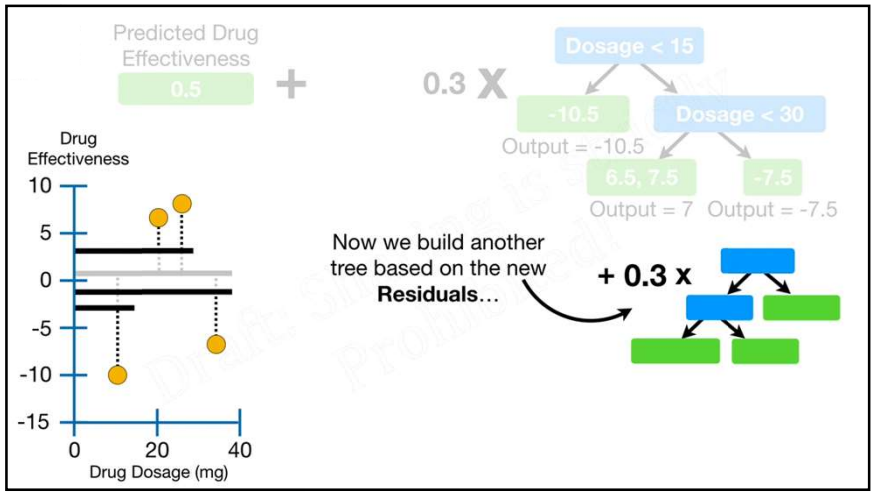

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
0.5 + (0.3 x -10.5) = -2.65
The new prediction is -2.65…

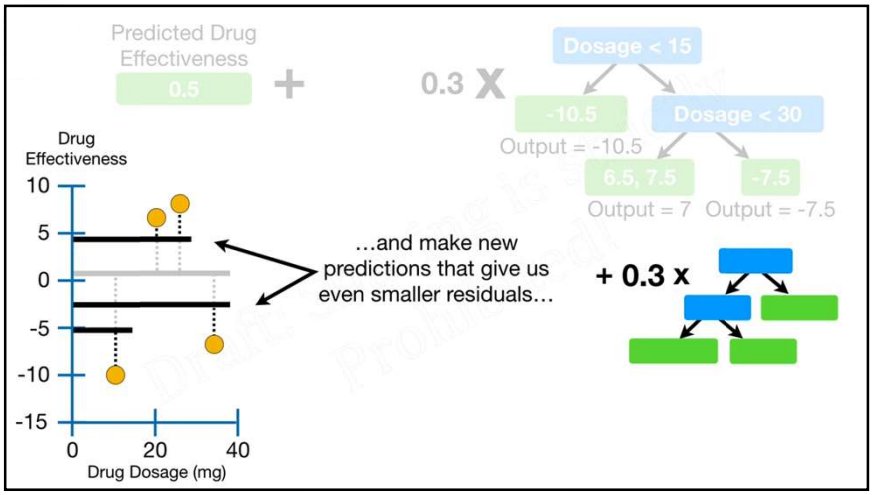

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
0.5 + (0.3 x -10.5) = -2.65
…and we see that the new **Residual** is smaller than before, so we've taken a small step in the right direction.

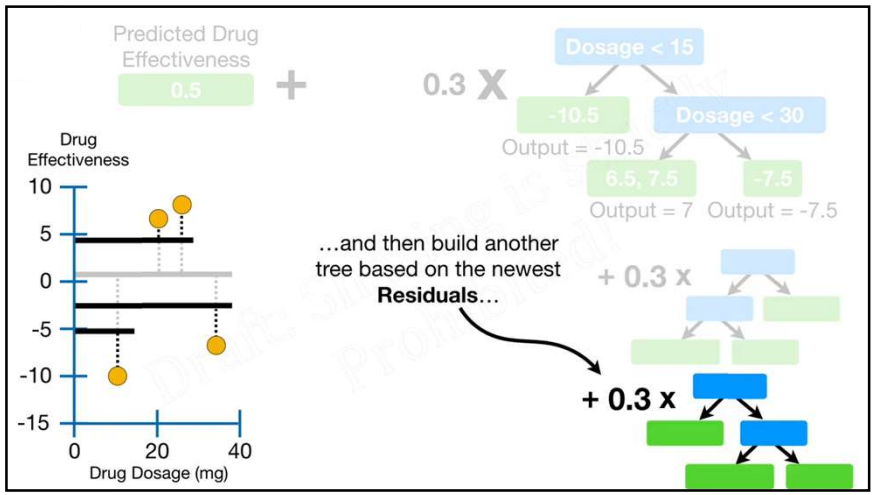

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Similarly, the new prediction for this observation, with **Dosage = 20** is…

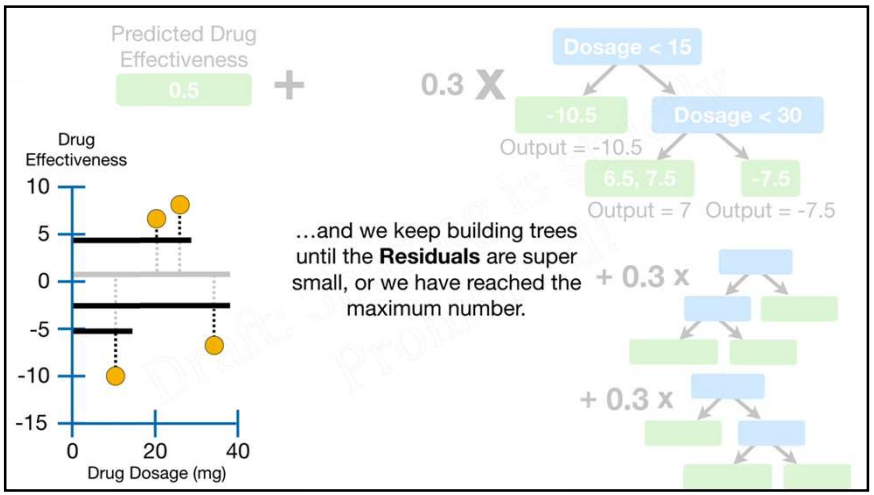

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
0.5 + (0.3 x 7) = 2.6
…**2.6**…


Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
0.5 + (0.3 x 7) = 2.6
…and the new **Residual** is smaller than before, so we've taken another small step in the right direction.


Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
Now we build another tree based on the new Residuals…
+ 0.3 x


Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…and make new predictions that give us even smaller residuals…
+ 0.3 x


Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…and then build another tree based on the newest Residuals…
+ 0.3 x
+ 0.3 x

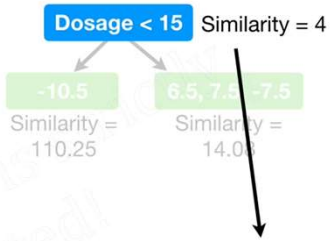

Predicted Drug Effectiveness
0.5
+
0.3 X
Dosage < 15
-10.5
Dosage < 30
Output = -10.5
6.5, 7.5
-7.5
Output = 7
Output = -7.5
Drug Effectiveness
10
5
0
-5
-10
-15
0
20
40
Drug Dosage (mg)
…and we keep building trees until the Residuals are super small, or we have reached the maximum number.
+ 0.3 x
+ 0.3 x

In summary, when building **XGBoost Trees** for **Regression**…

…we calculate **Similarity Scores**…

$$\text{Similarity Score} = \frac{\text{Sum of Residuals, Squared}}{\text{Number of Residuals} + \lambda}$$

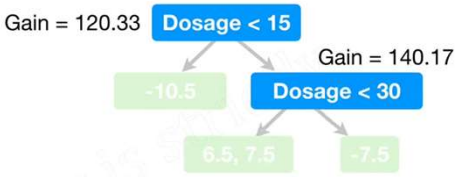
… and **Gain** to determine how to split the data…

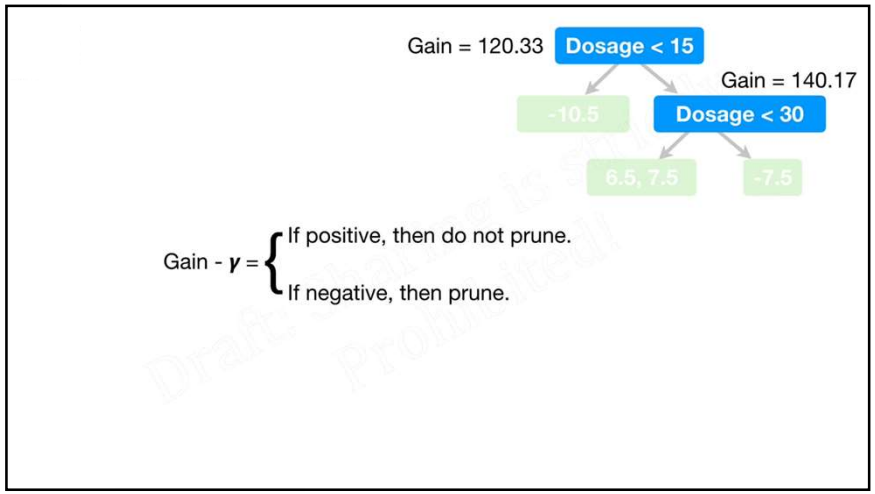


$$\text{Gain} = \text{Left}_{\text{Similarity}} + \text{Right}_{\text{Similarity}} - \text{Root}_{\text{Similarity}}$$

…and we prune the tree by calculating the differences between **Gain** values and a user defined **Tree Complexity Parameter**, ***γ*** **(gamma)**.

Gain - ***γ*** =





…and lastly, **λ (lambda)** is a **Regularization Parameter** and when **λ > 0**, it results in more pruning, by shrinking the **Similarity Scores**, and it results in smaller **Output Values** for the leaves.

…and lastly, **λ (lambda)** is a **Regularization Parameter** and when **λ > 0**, it results in more pruning, by shrinking the **Similarity Scores**, and it results in smaller **Output Values** for the leaves.

$$\text{Similarity Score} = \frac{\text{Sum of Residuals, Squared}}{\text{Number of Residuals} + \lambda}$$

…and lastly, **λ (lambda)** is a **Regularization Parameter** and when **λ > 0**, it results in more pruning, by shrinking the **Similarity Scores**, and it results in smaller **Output Values** for the leaves.

$$\text{Similarity Score} = \frac{\text{Sum of Residuals, Squared}}{\text{Number of Residuals} + \lambda}$$

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

THANK YOU

StatQuest with Josh Starmer
http://www.youtube.com/watch?v=OtD8wVaFm6E