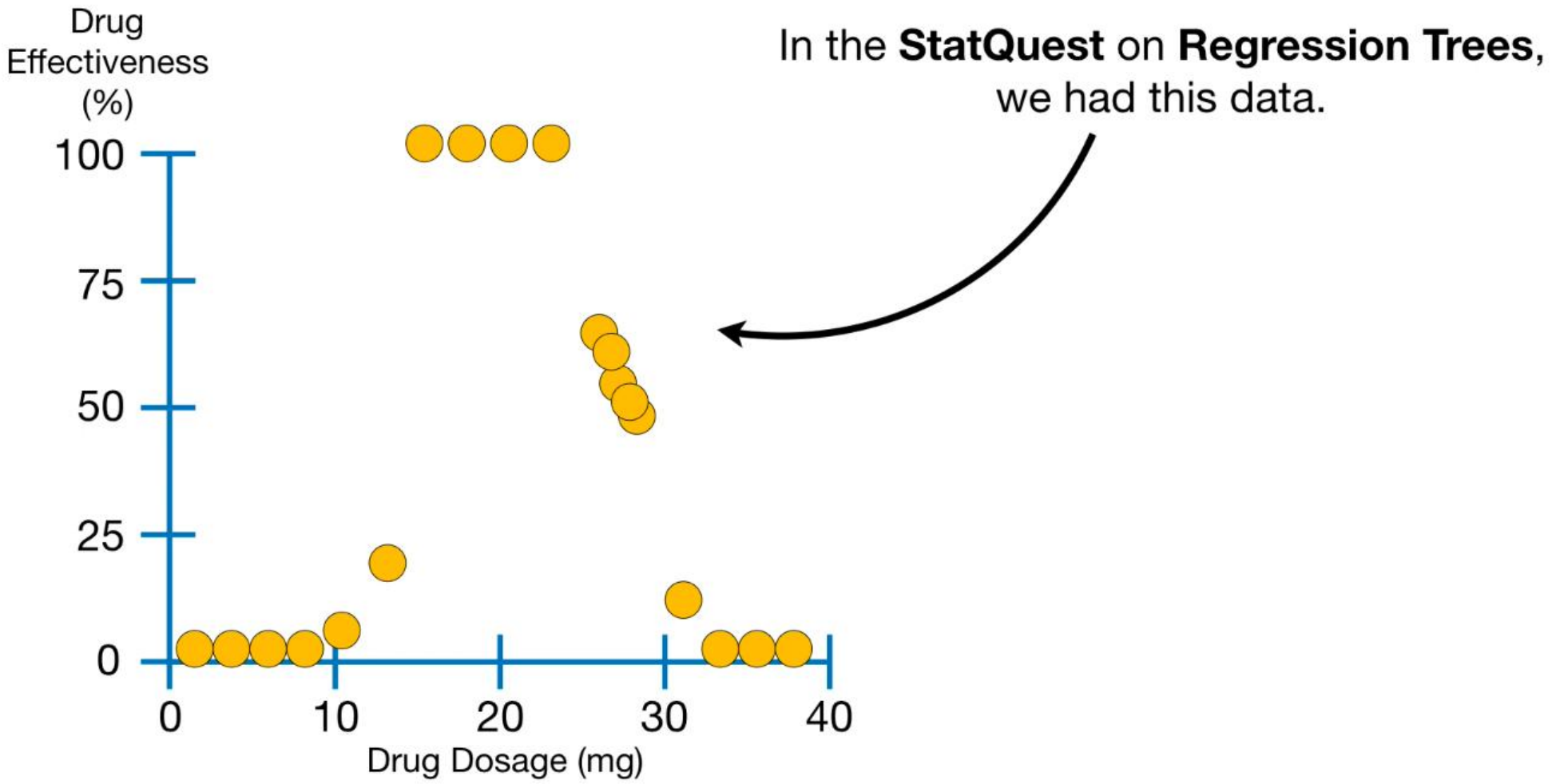
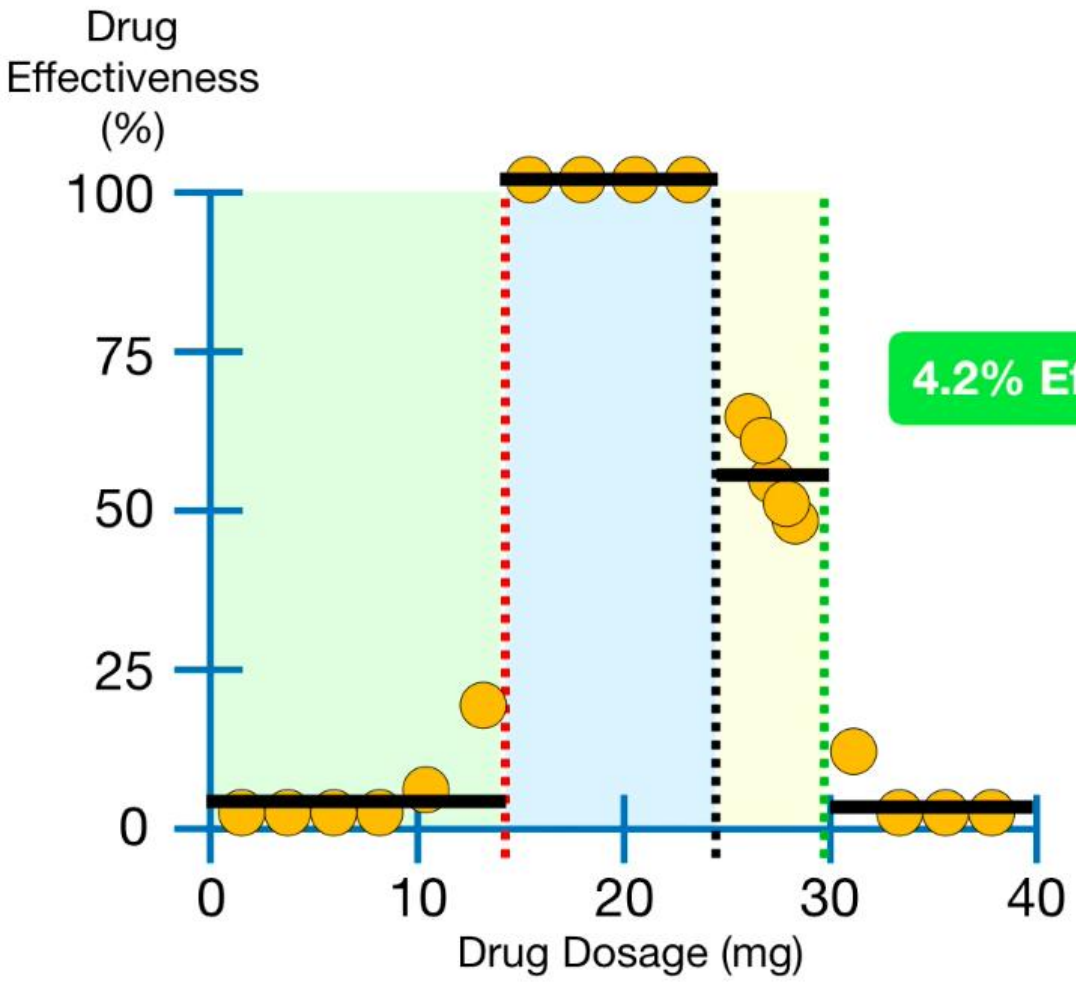
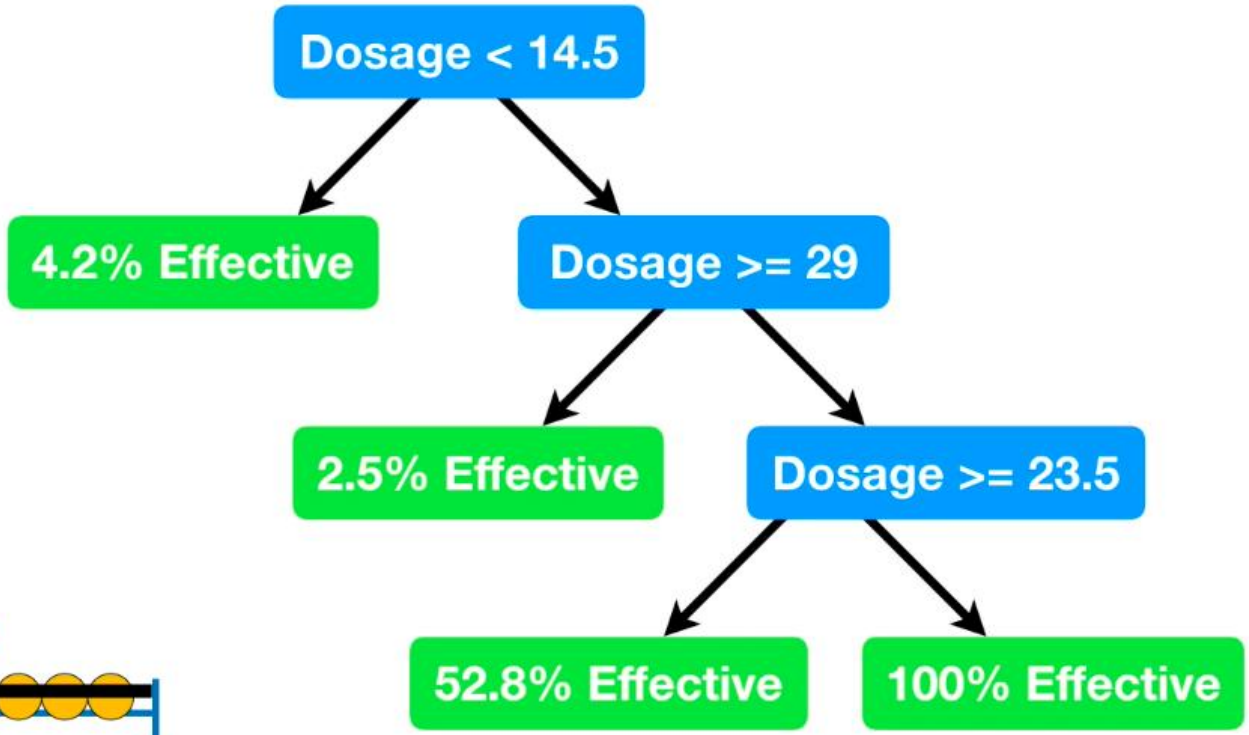
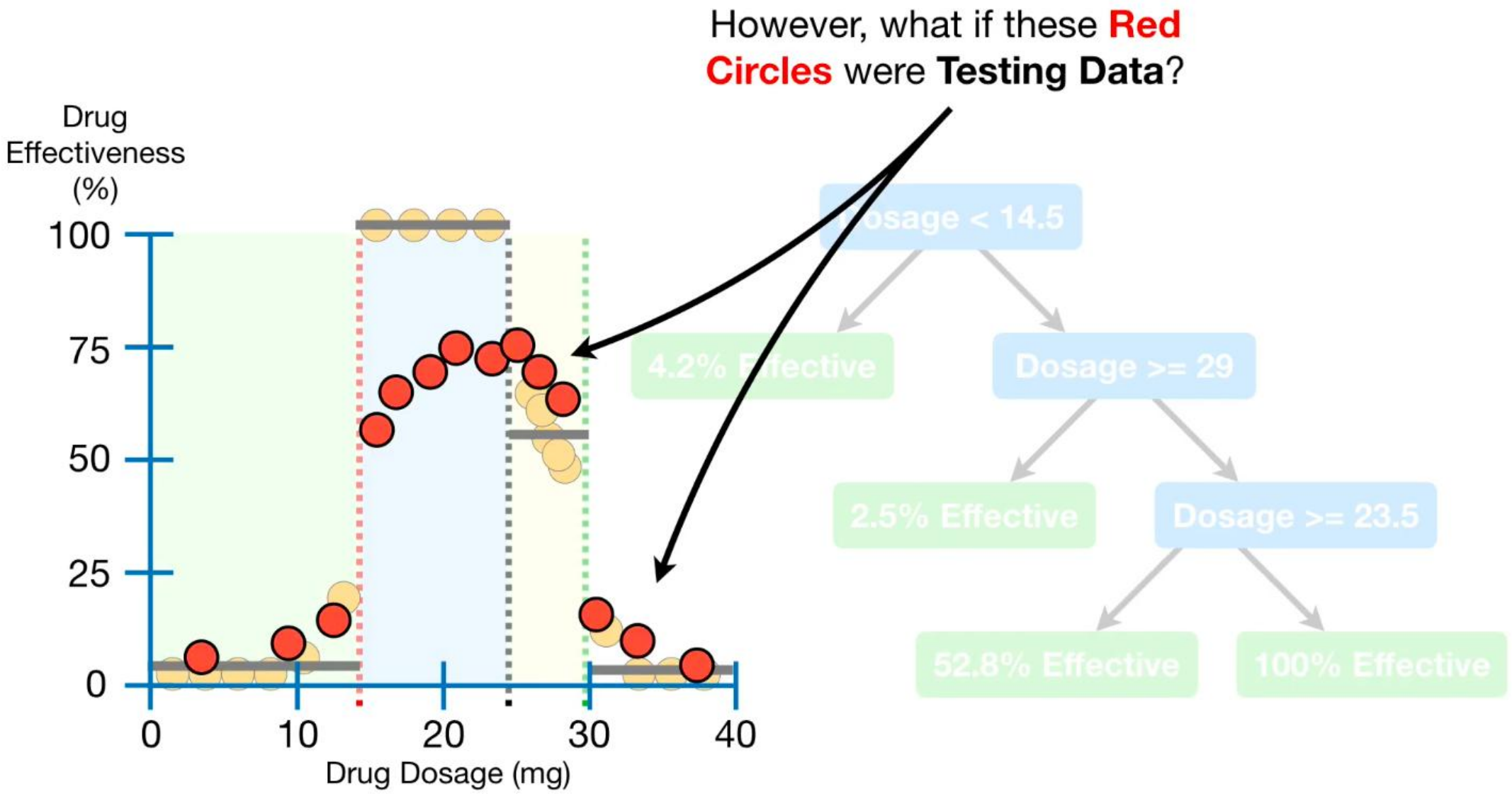
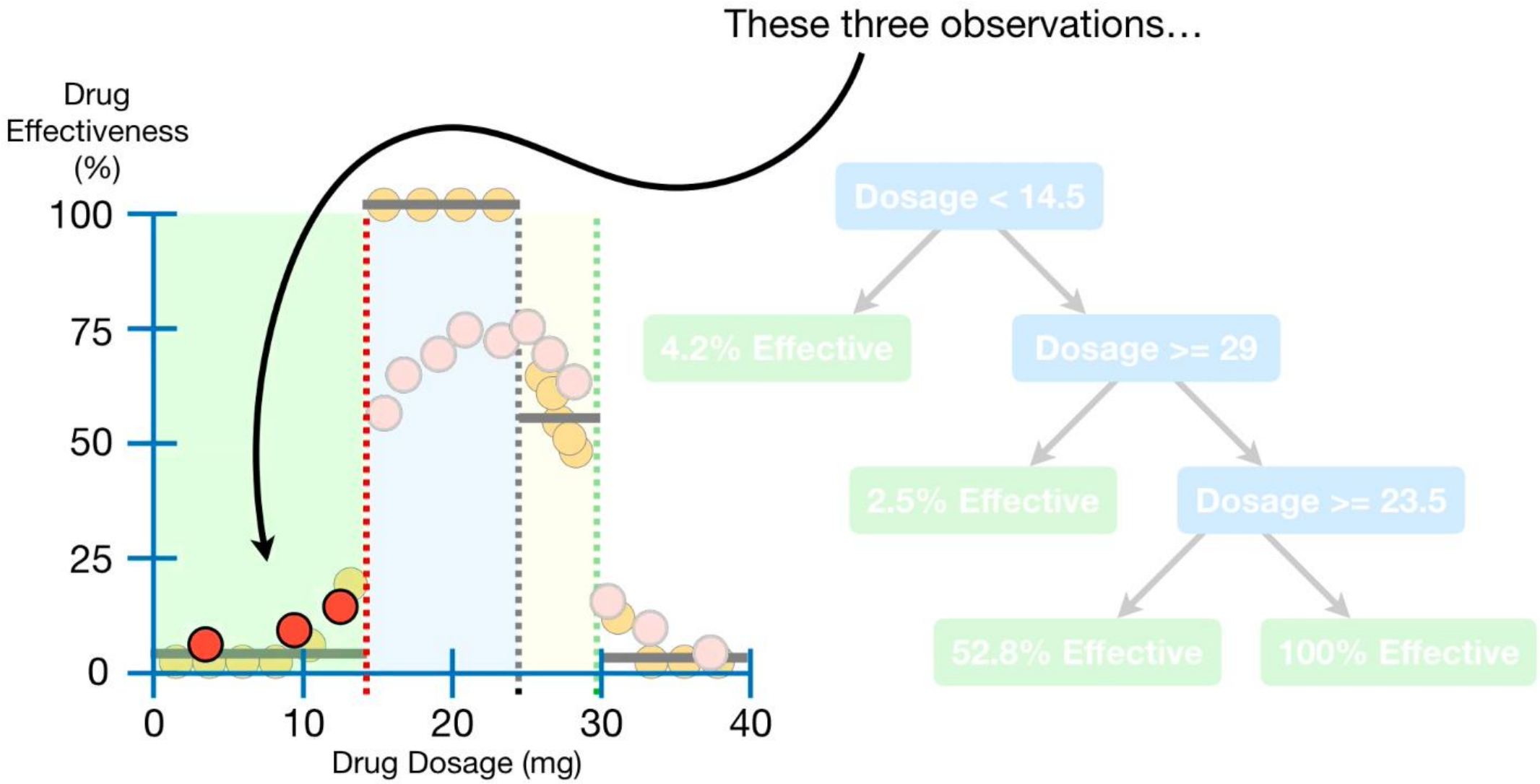

Drug
Effectiveness
(%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
In the **StatQuest** on **Regression Trees**,
we had this data.

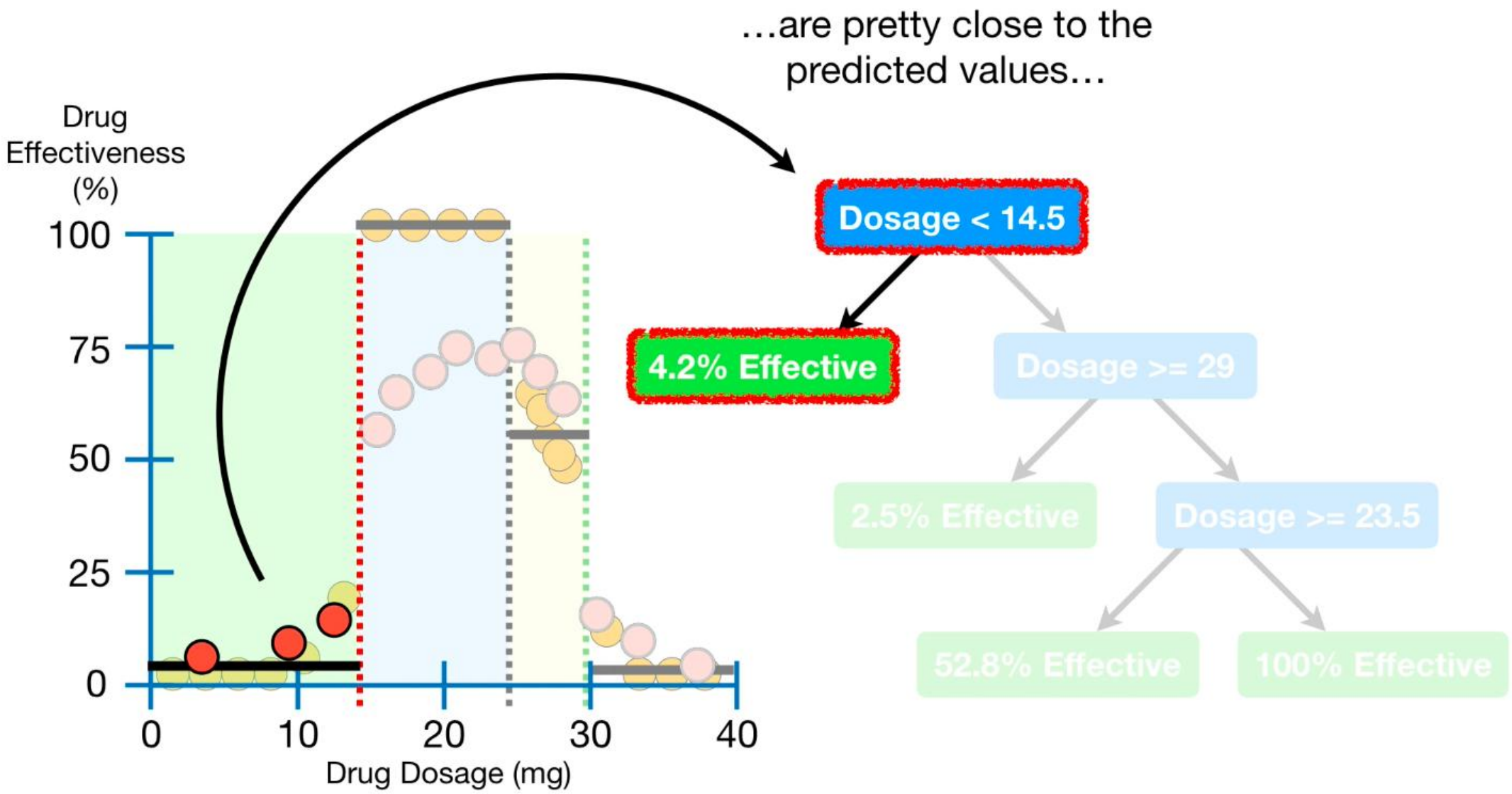
We then fit a **Regression Tree** to the data…



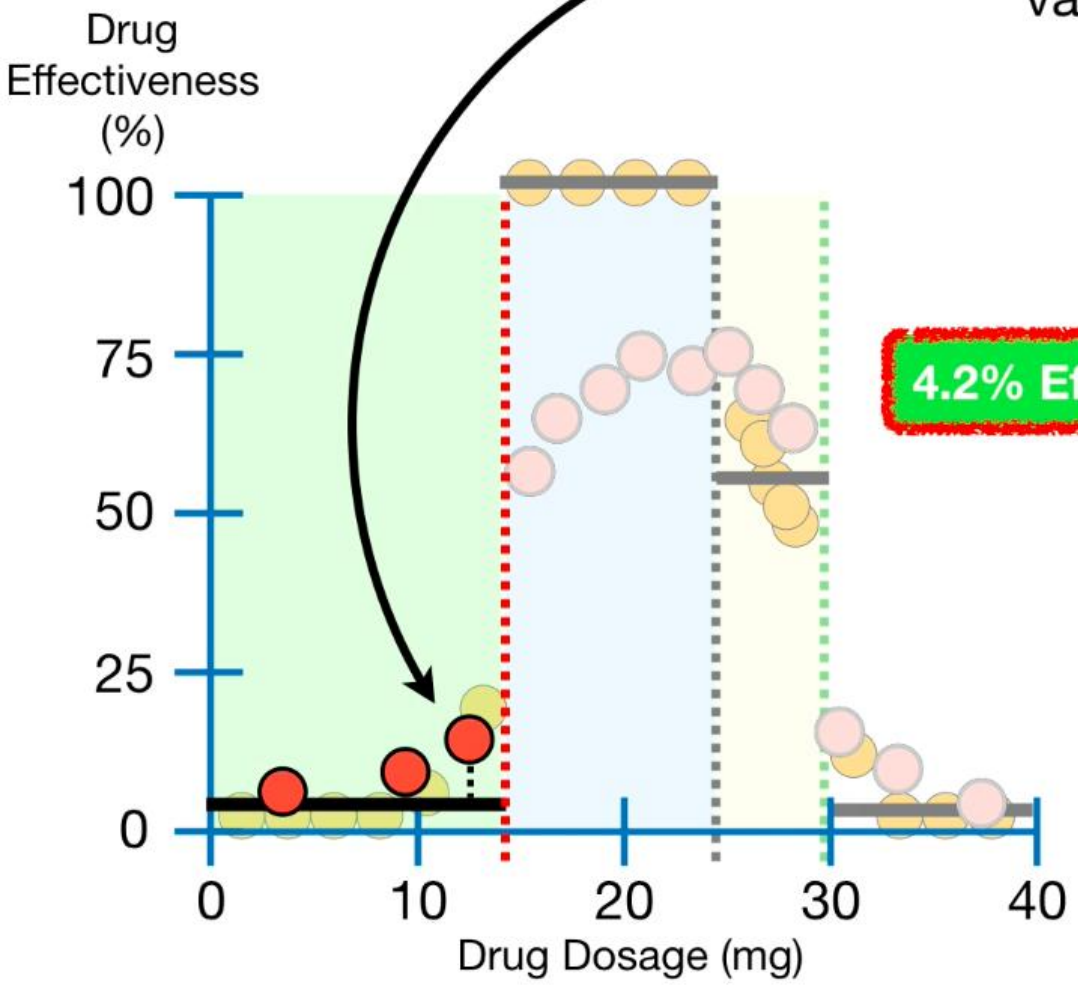


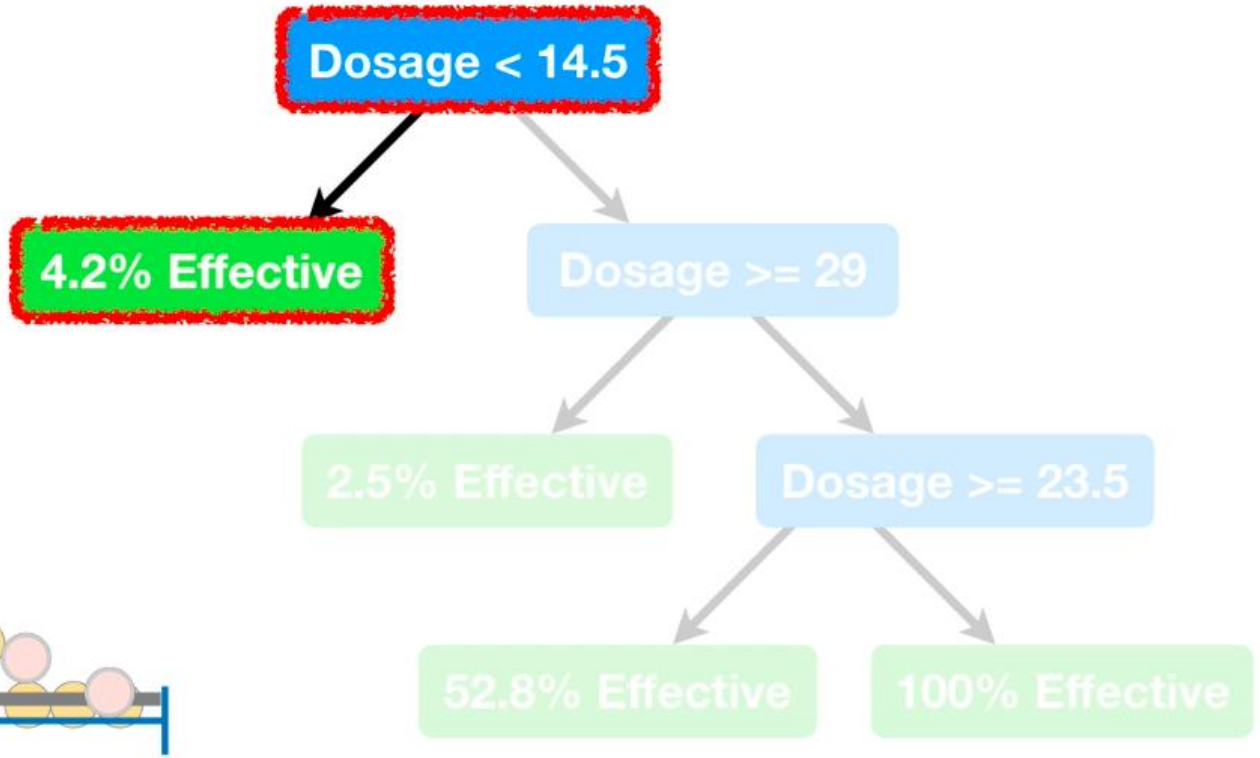

However, what if these Red Circles were Testing Data?
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
Dosage >= 29
2.5% Effective
Dosage >= 23.5
52.8% Effective
100% Effective


These three observations…
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
Dosage >= 29
2.5% Effective
Dosage >= 23.5
52.8% Effective
100% Effective

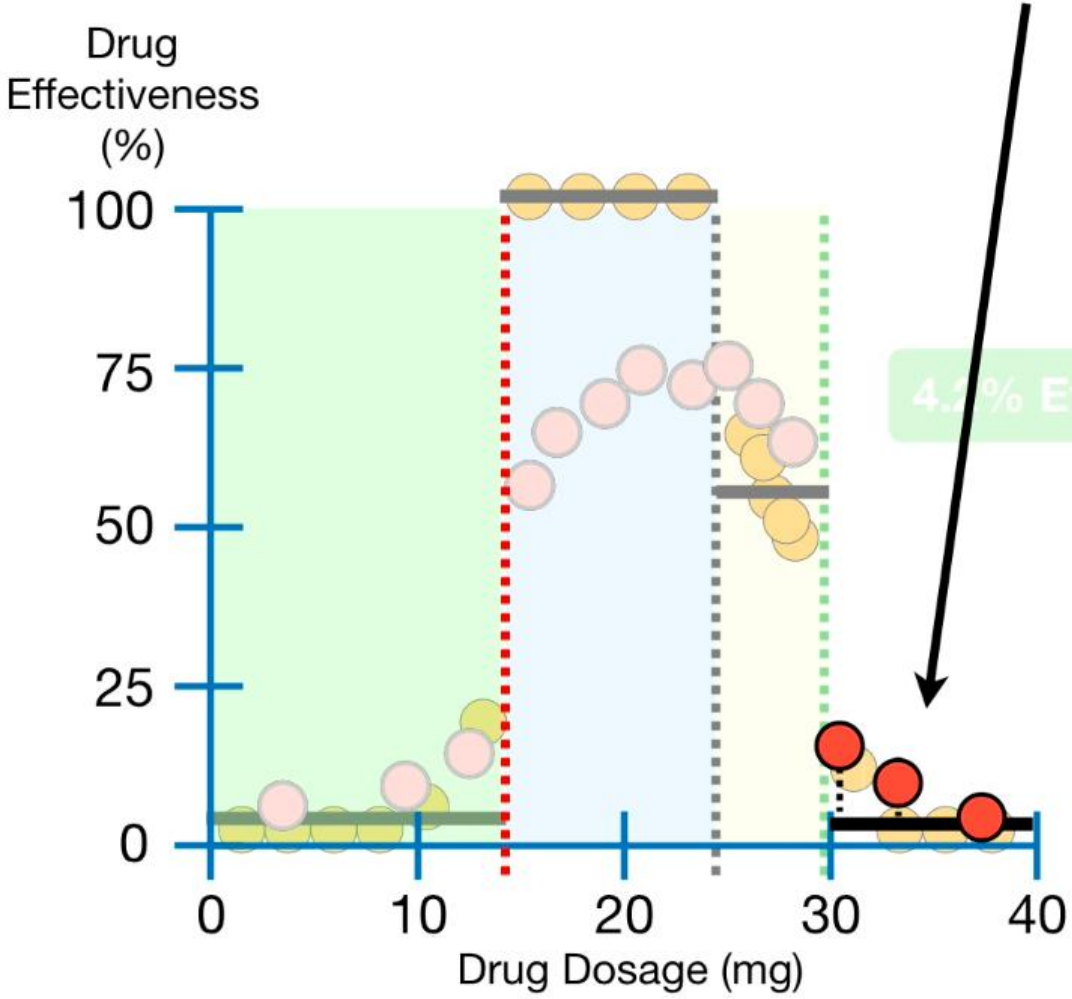

…are pretty close to the predicted values…
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
Dosage >= 29
2.5% Effective
Dosage >= 23.5
52.8% Effective
100% Effective

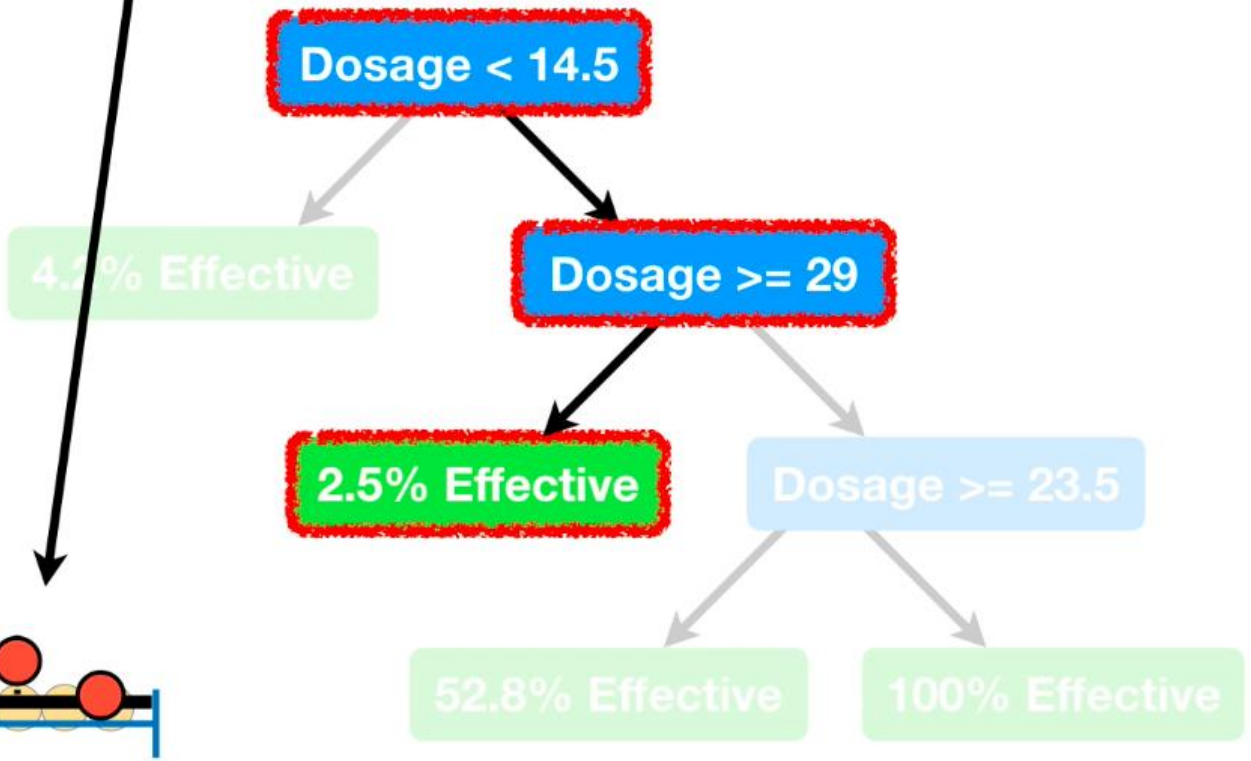
…so their **Residuals**, the difference between the **Observed** and **Predicted** values, are not very large.



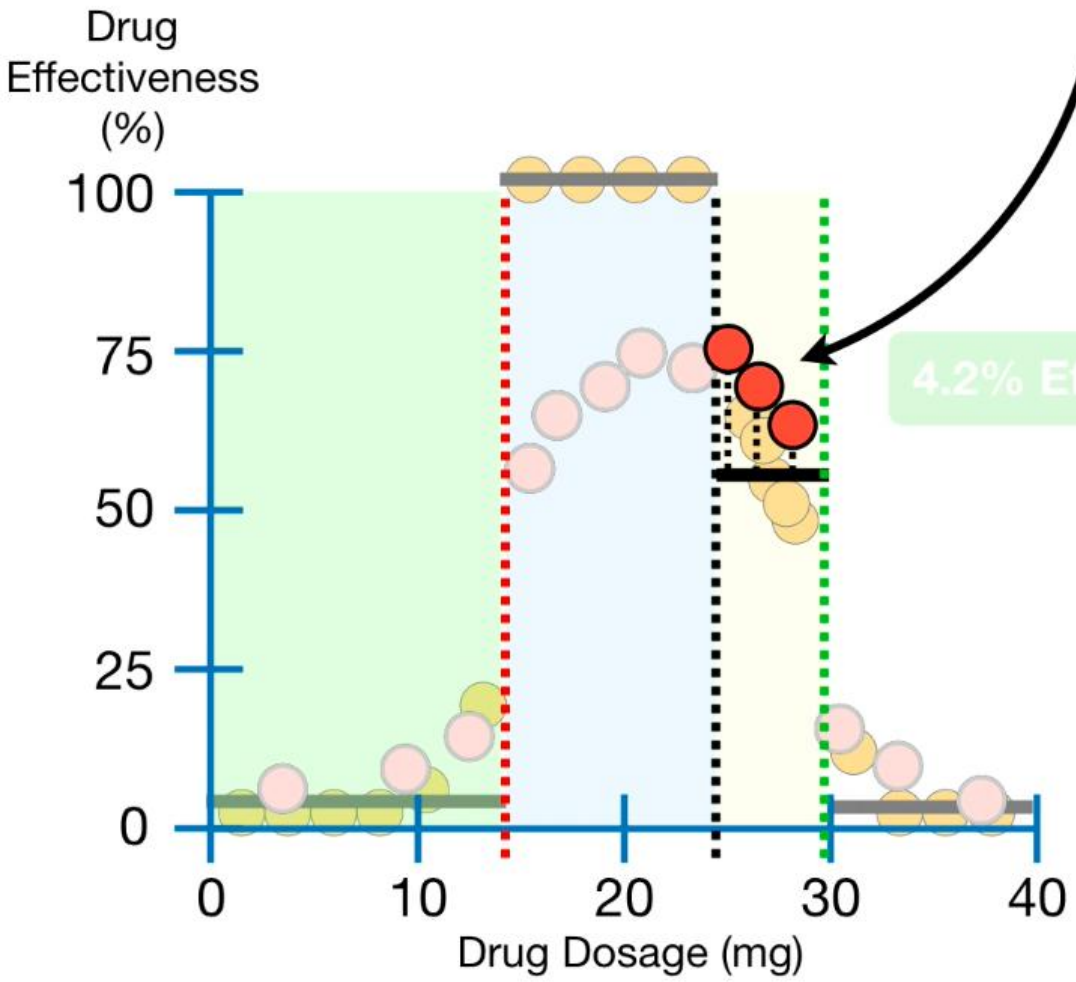


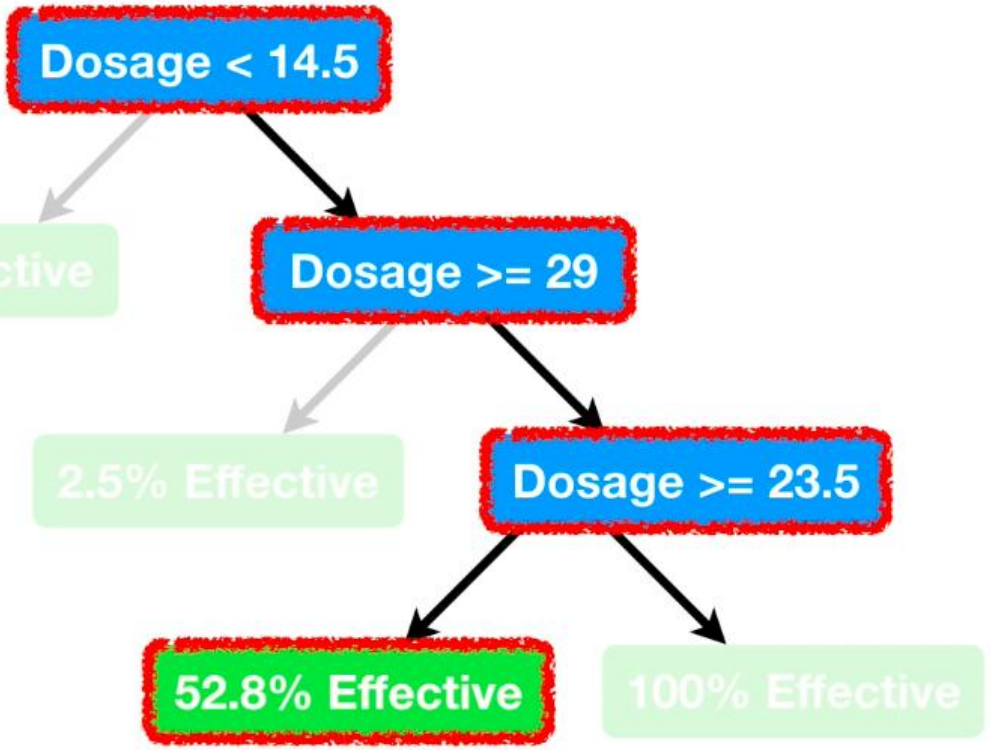
Similarly, the **Residuals** for these observations in the **Testing Data** are relatively small.



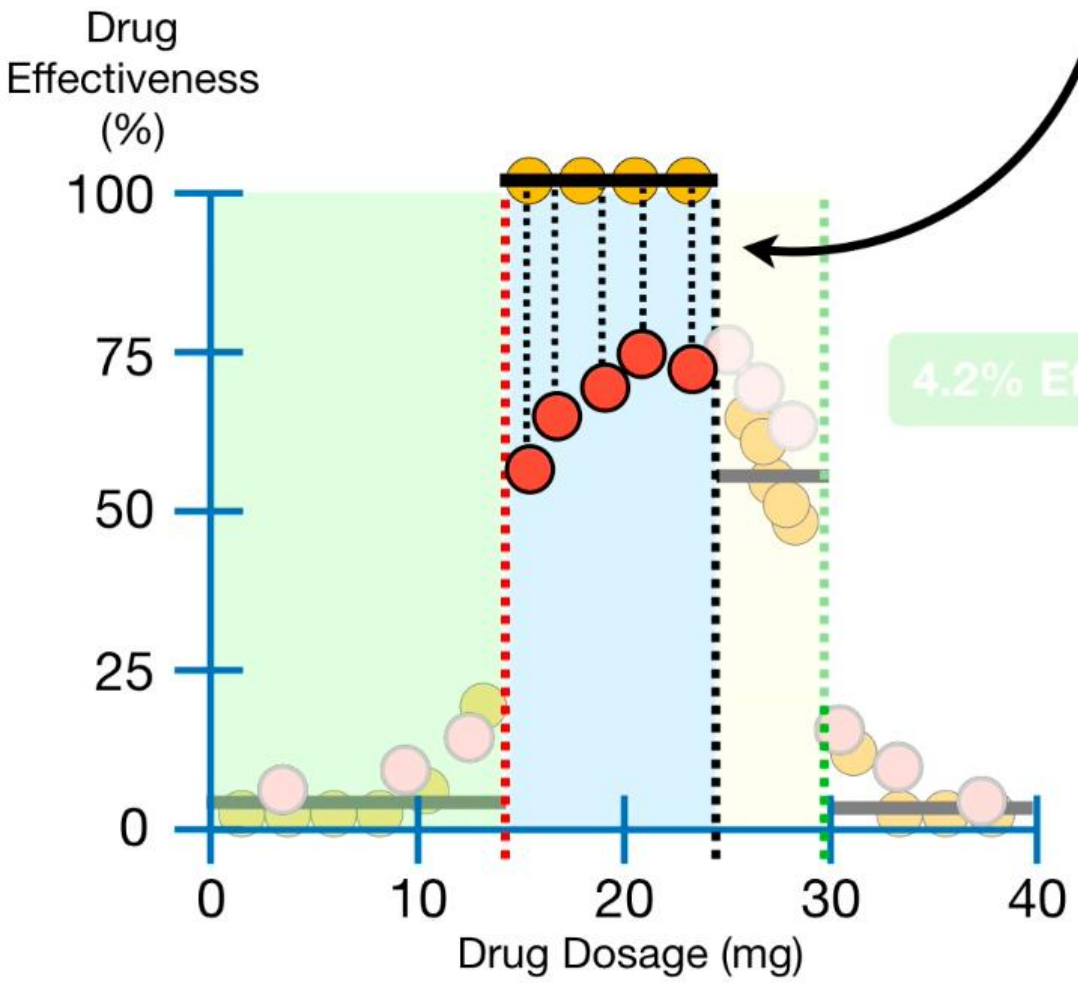


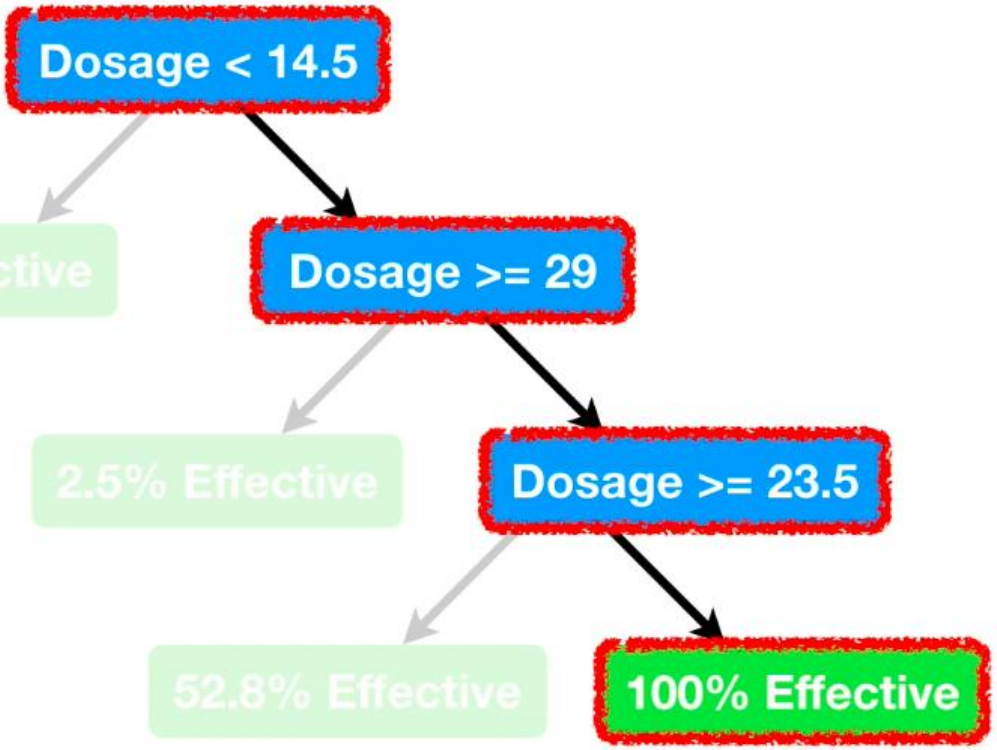
However, the **Residuals** for these **Observations** are larger than before.



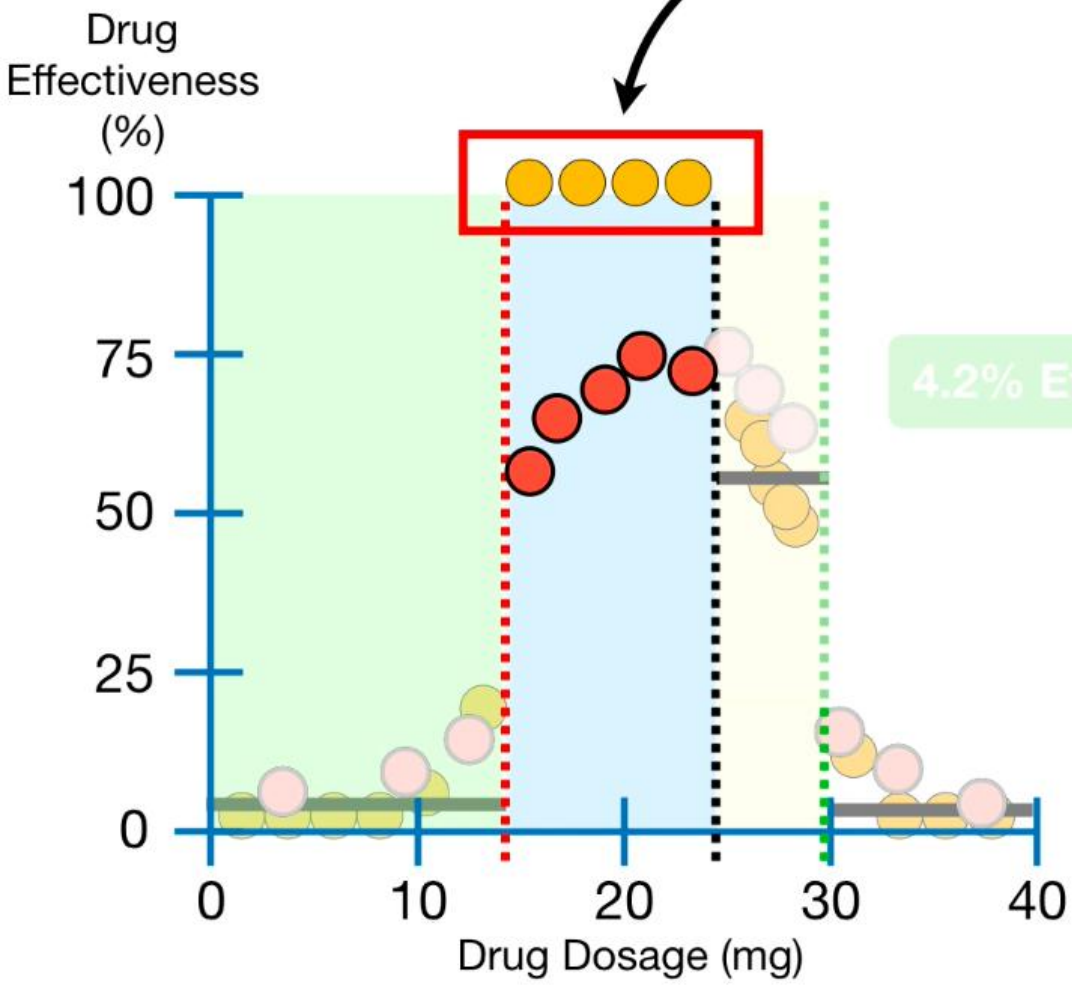


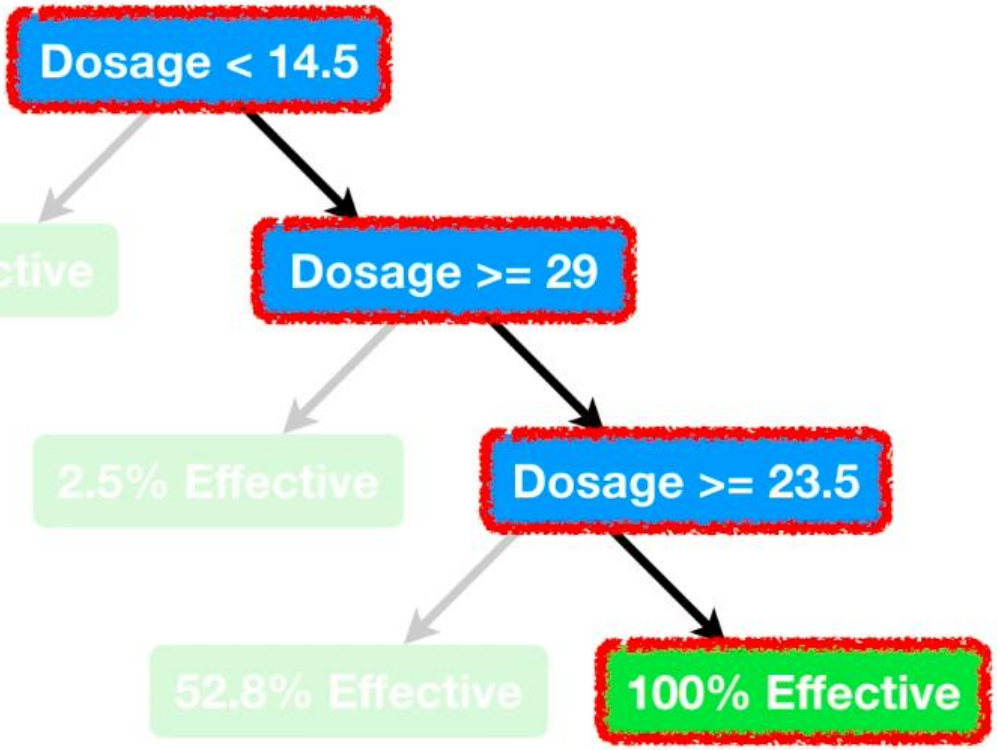
And the **Residuals** for these **Observations** are much larger.



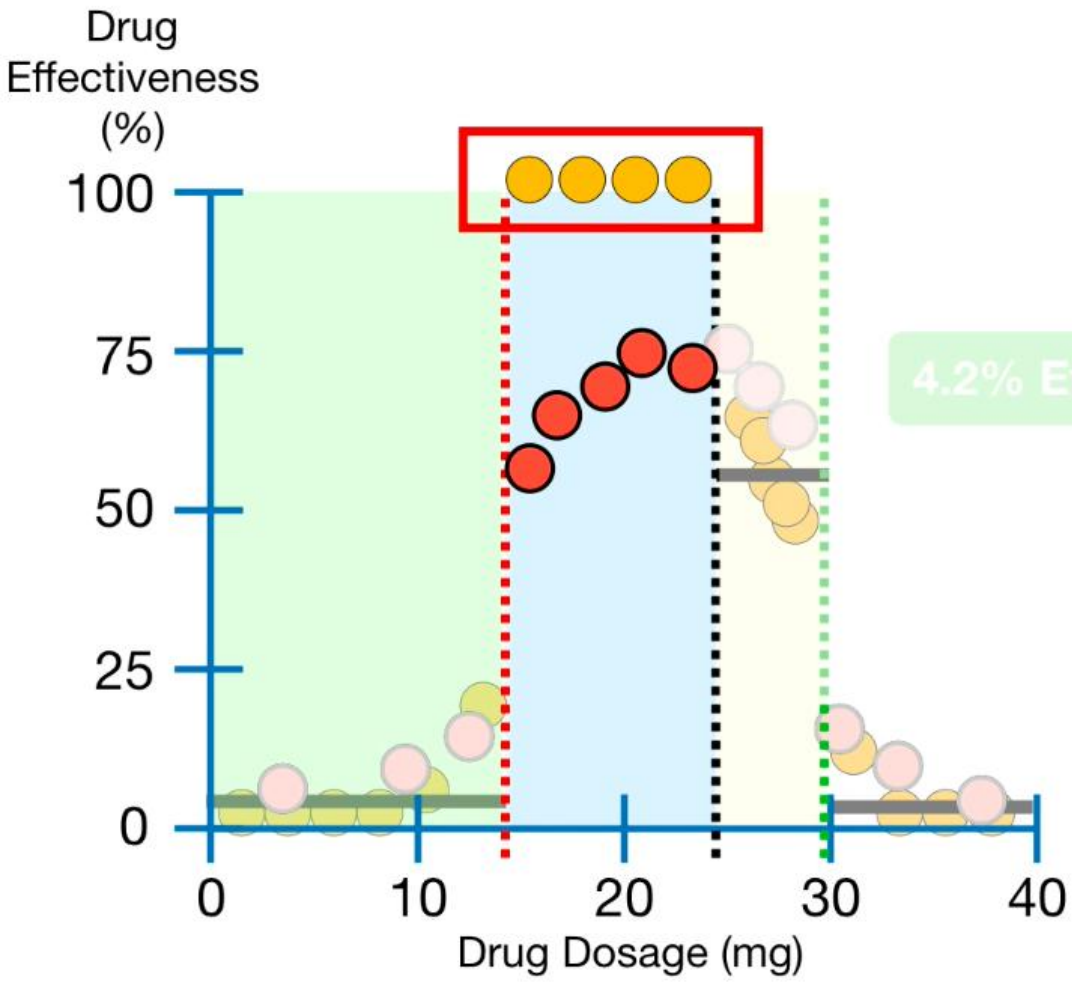


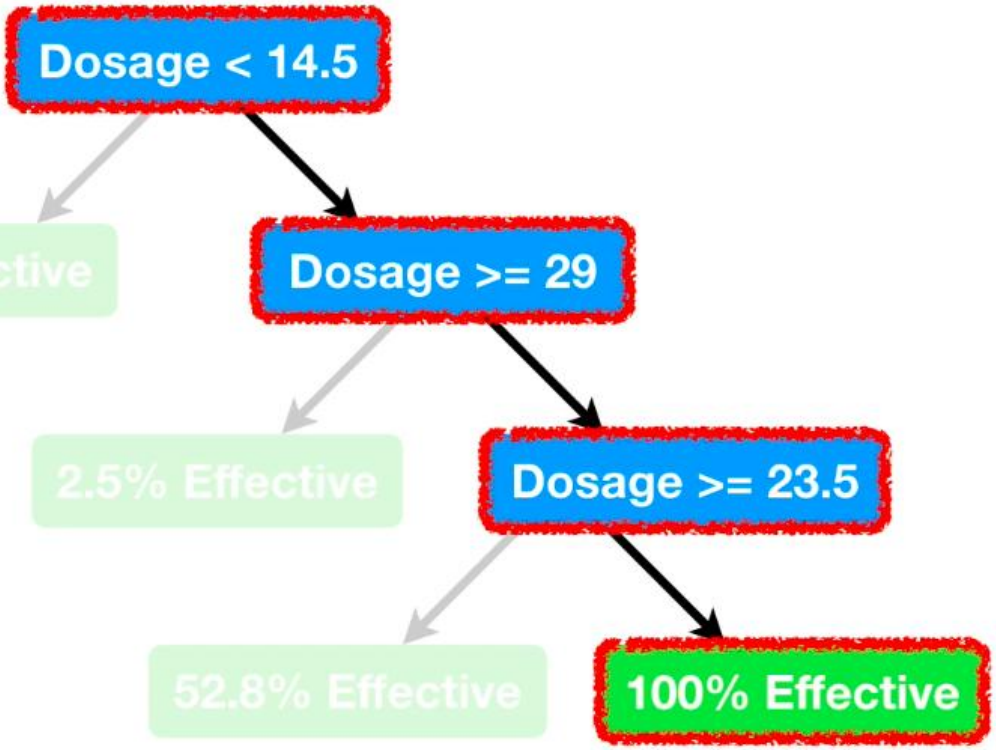
These **4** observations from the **Training Data** with **100% Drug Effectiveness** now look a little bit like outliers…



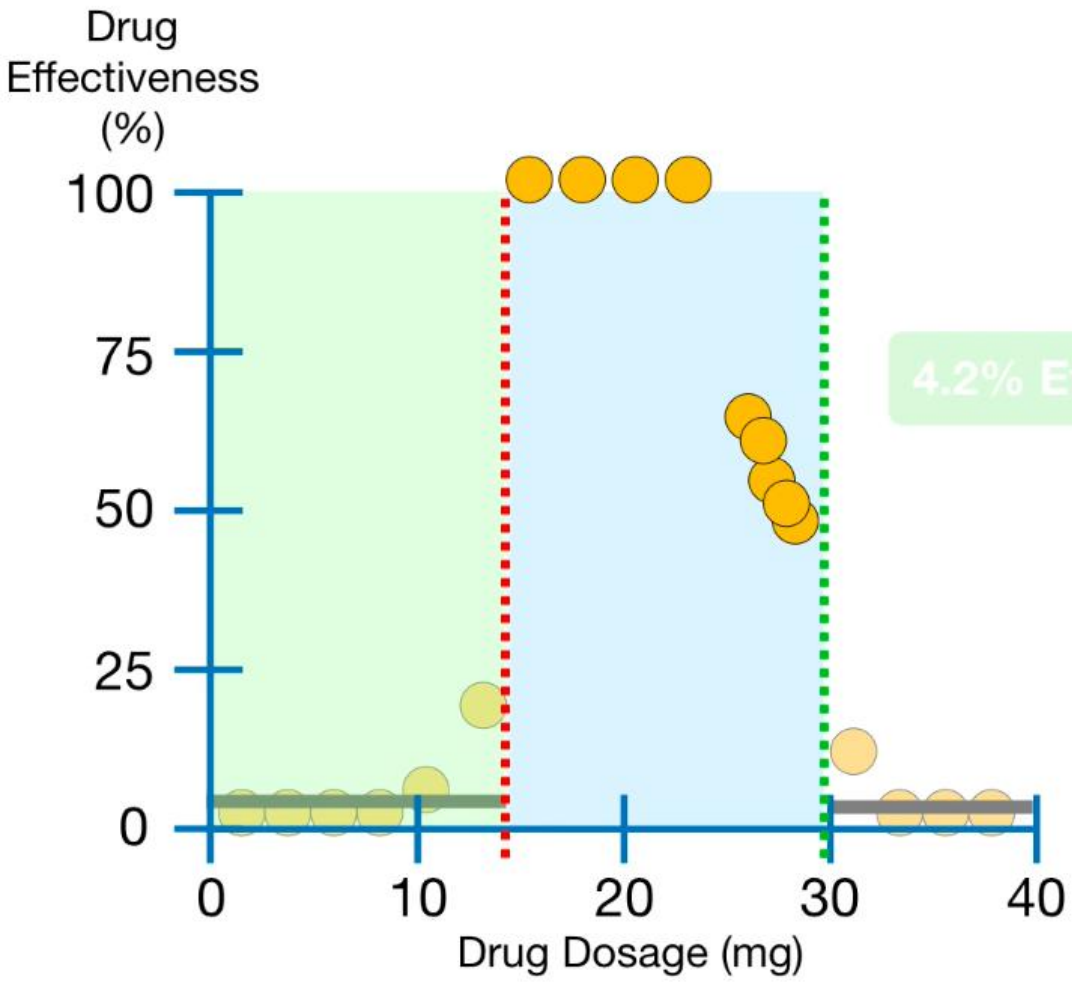


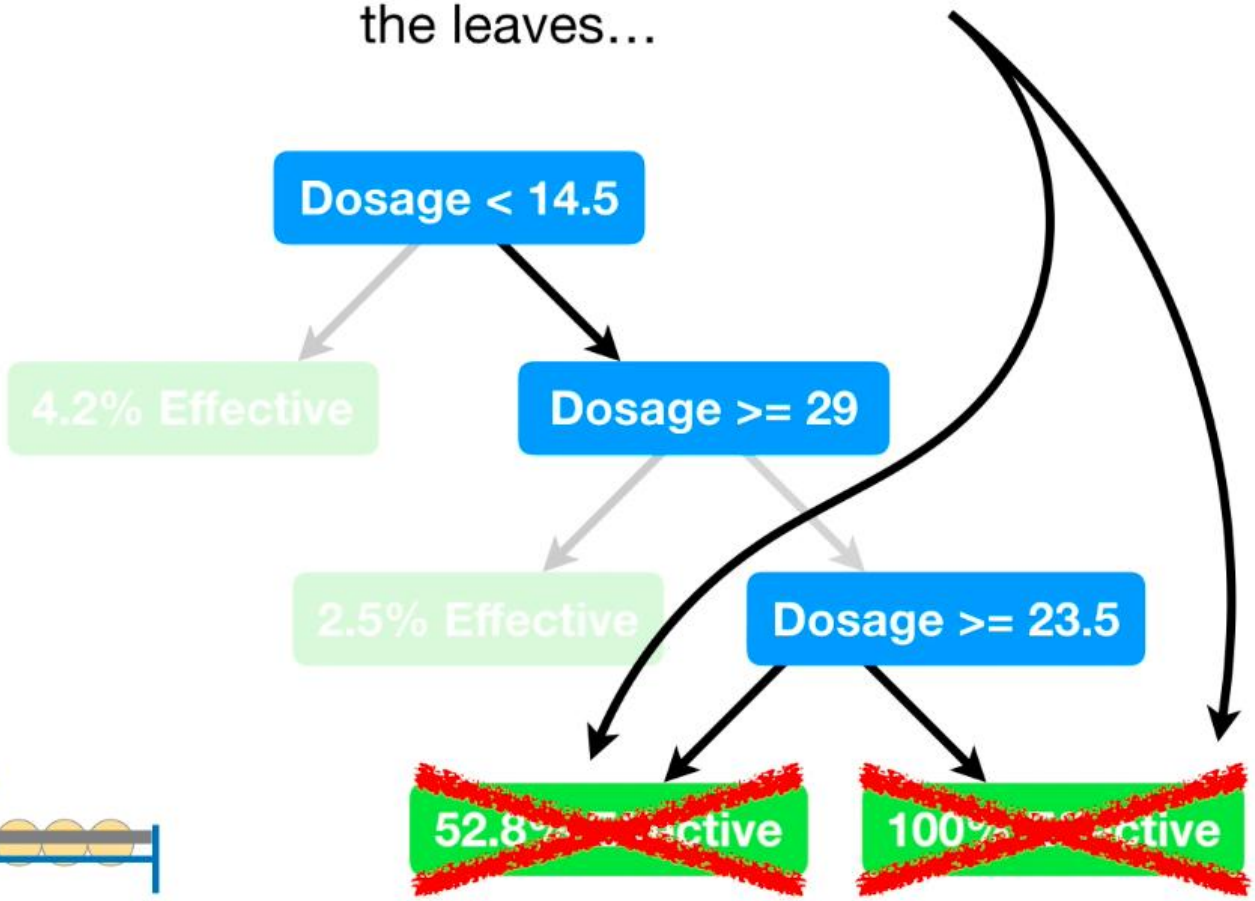
…and that means that maybe we **Overfit** the **Regression Tree** to the **Training Data**.



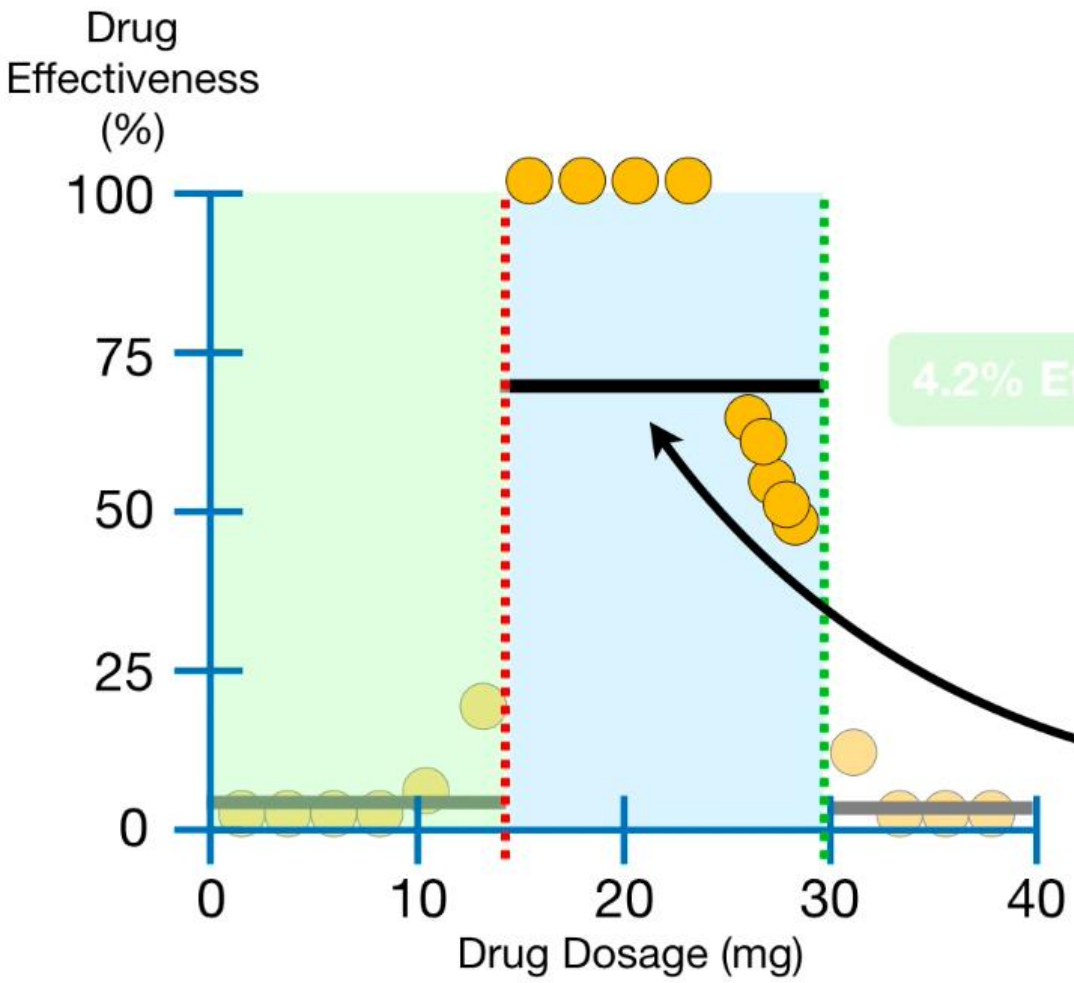


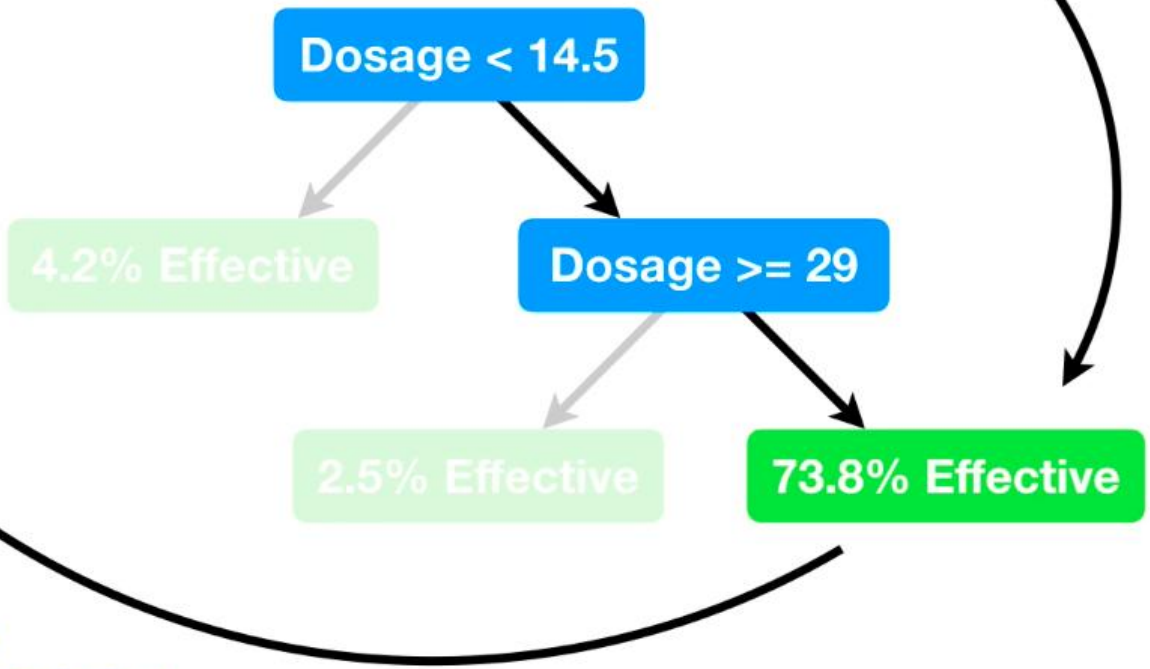
One way to prevent **Overfitting** a **Regression Tree** to the **Training Data** is to remove some of the leaves…

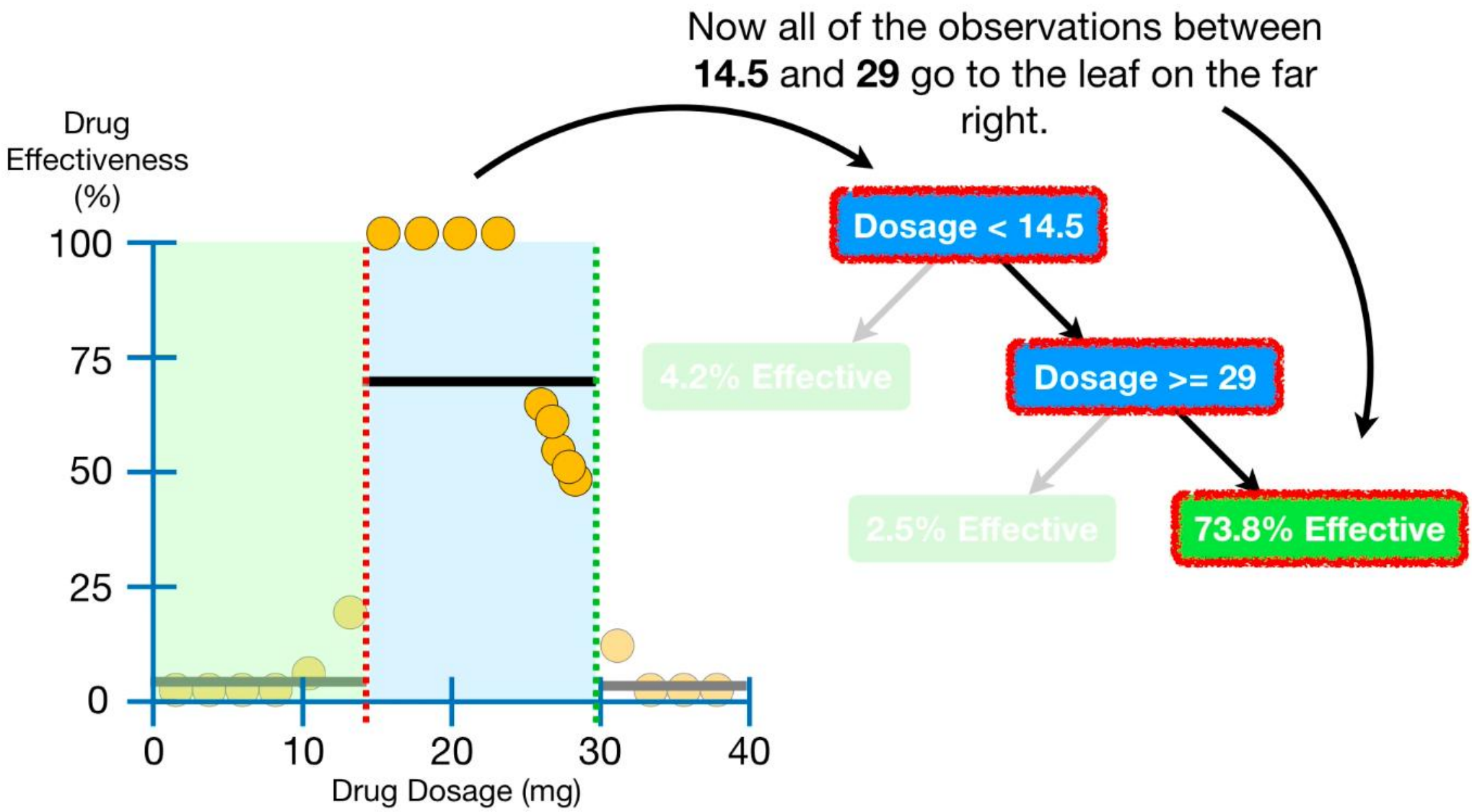




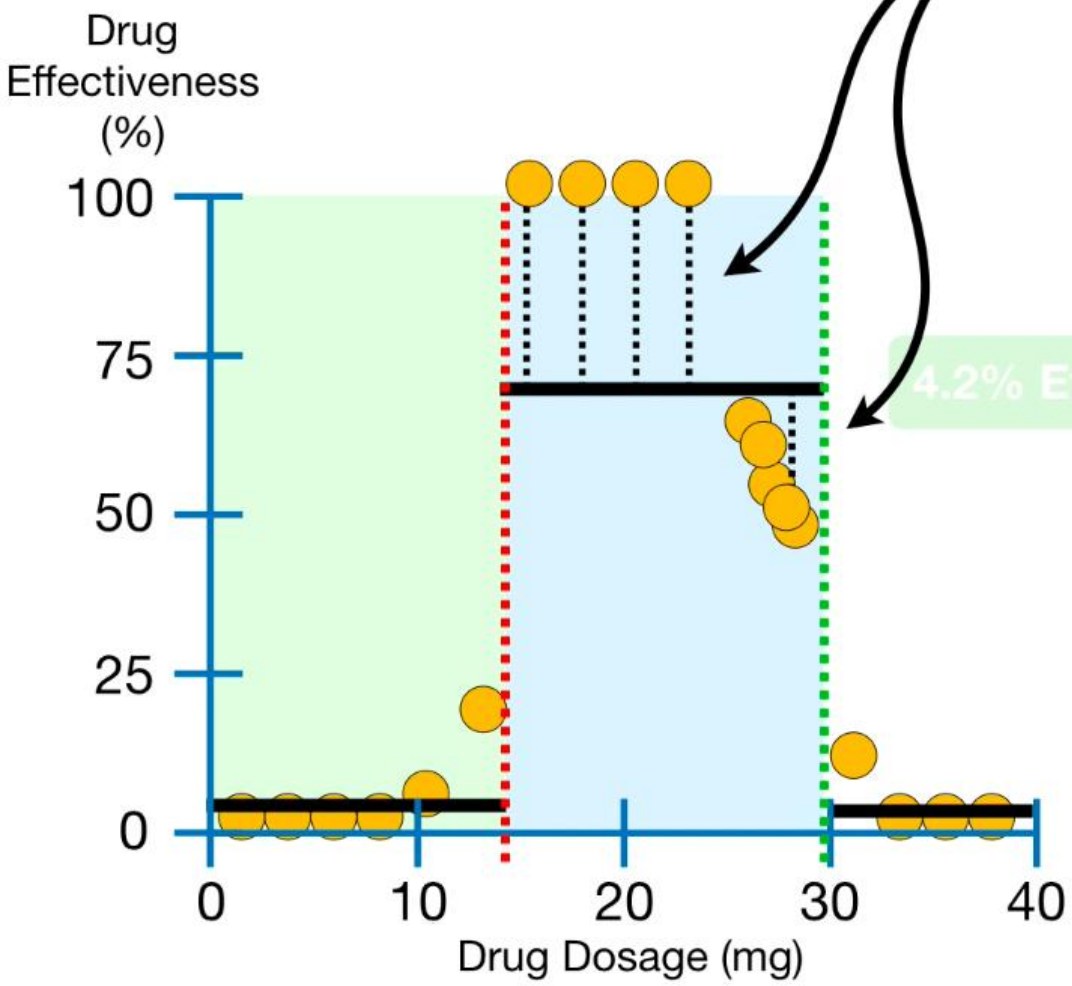
…and replace the split with a leaf that is the average of a larger number of observations.

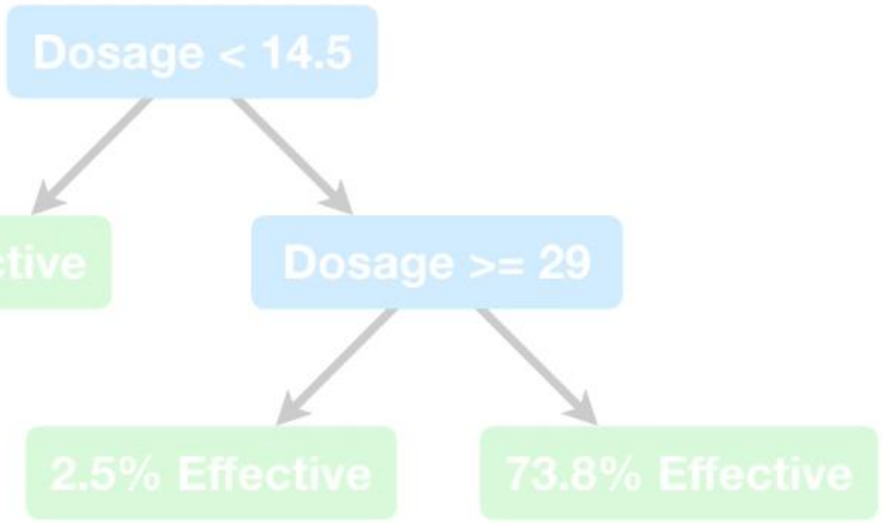




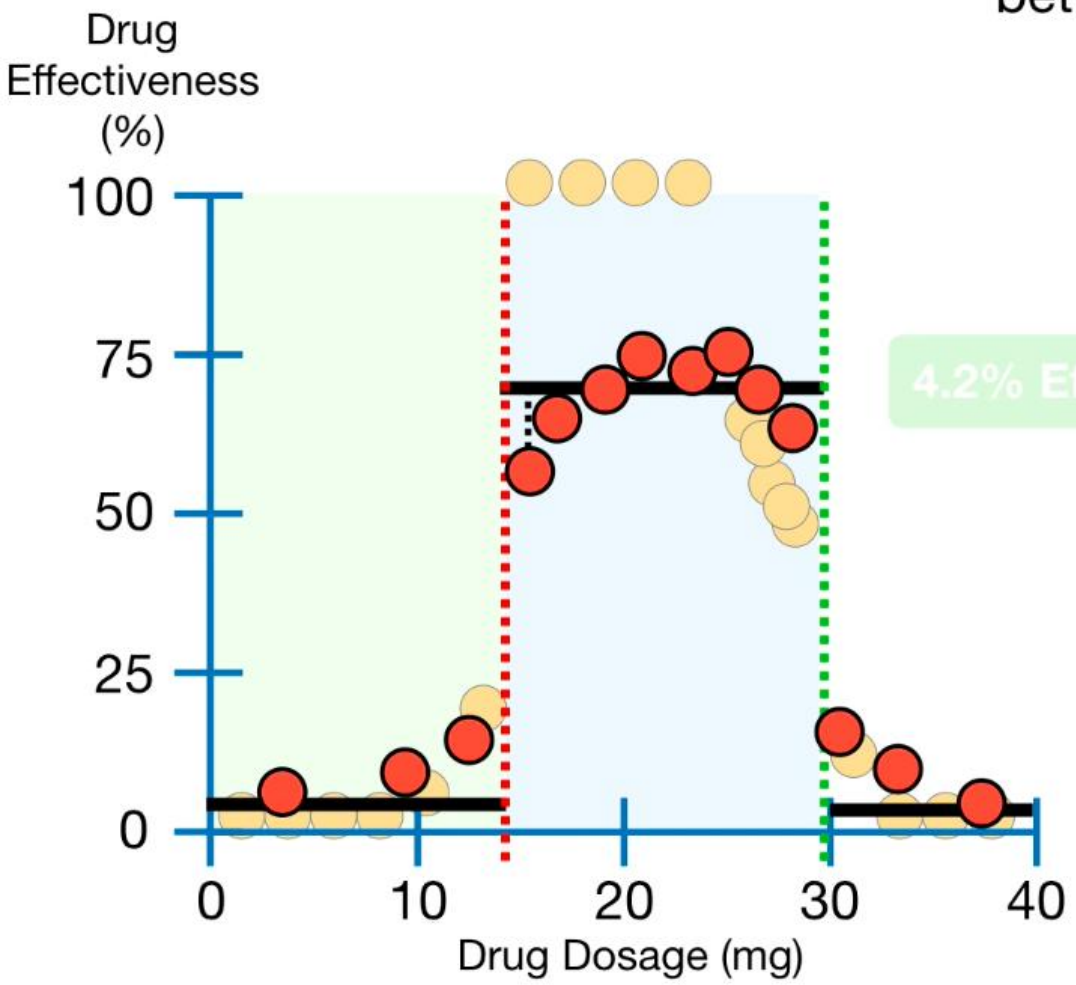

Now all of the observations between **14.5** and **29** go to the leaf on the far right.
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
Dosage >= 29
2.5% Effective
73.8% Effective

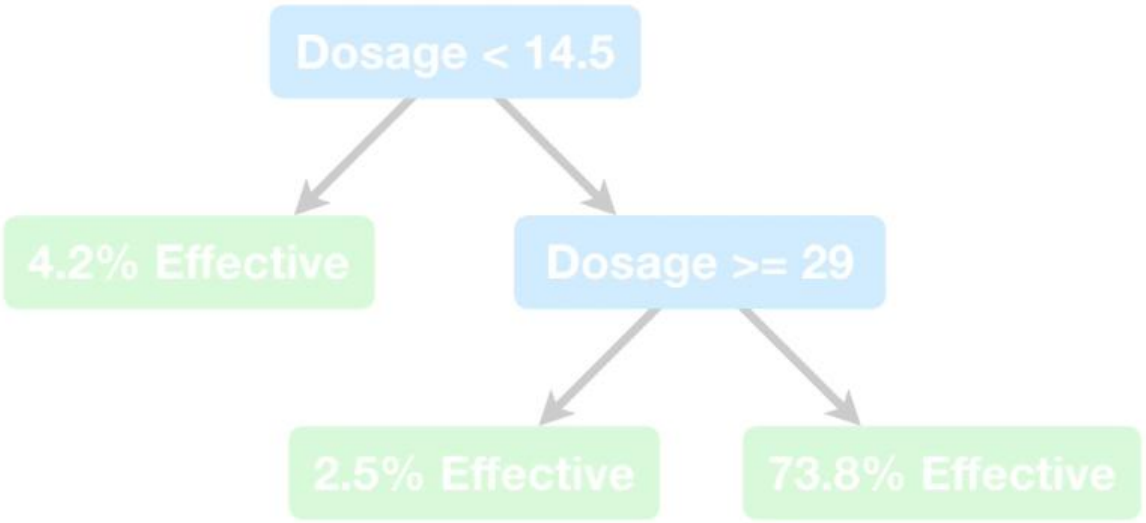
The large **Residuals** tell us that the new tree doesn't fit the **Training Data** as well as before…



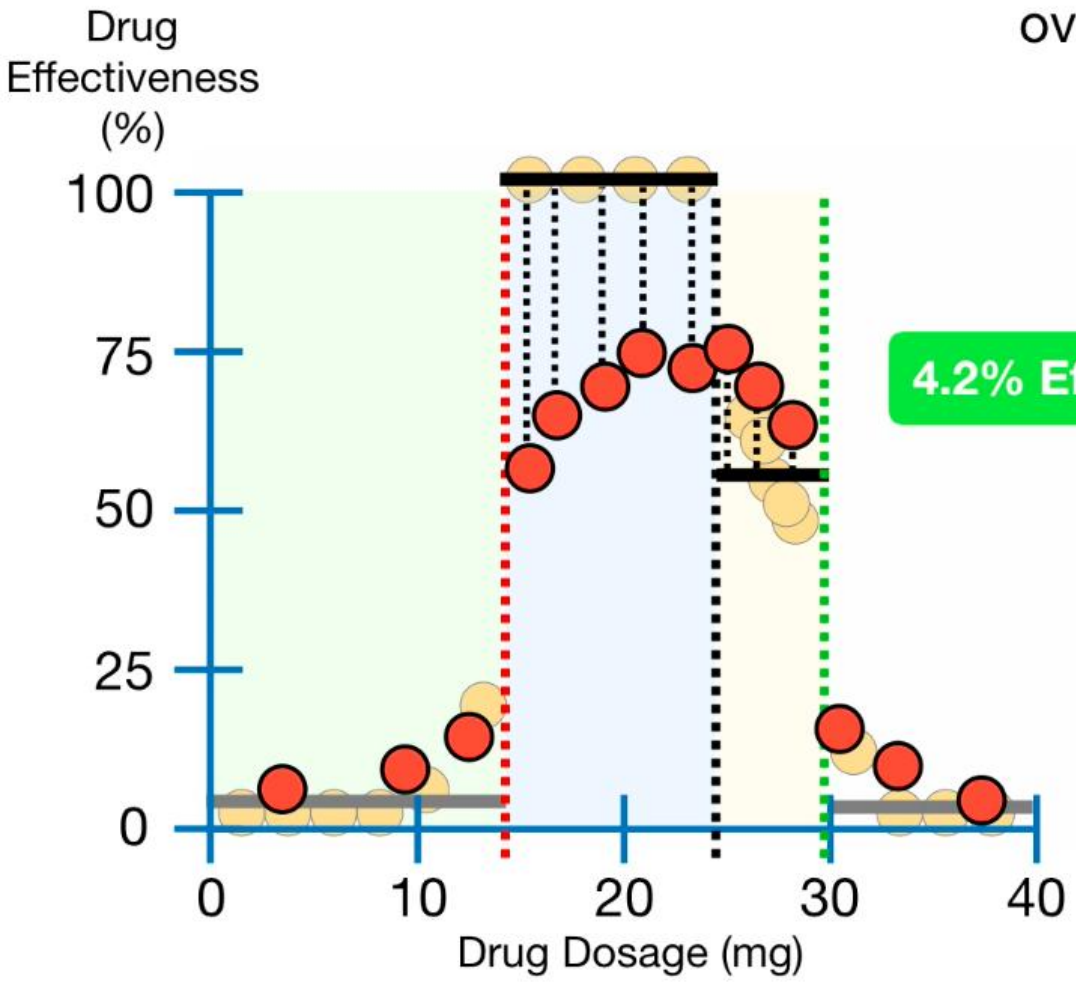


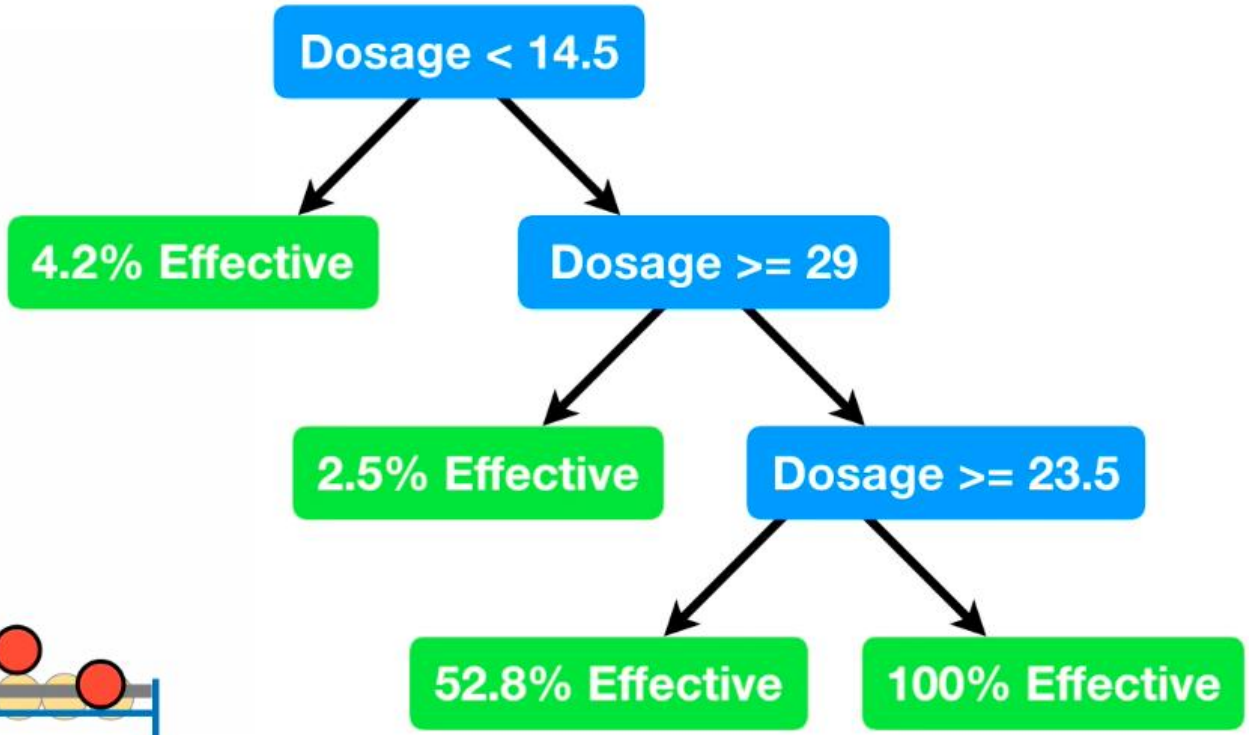
…but the new sub-tree does a much better job with the **Testing Data**.



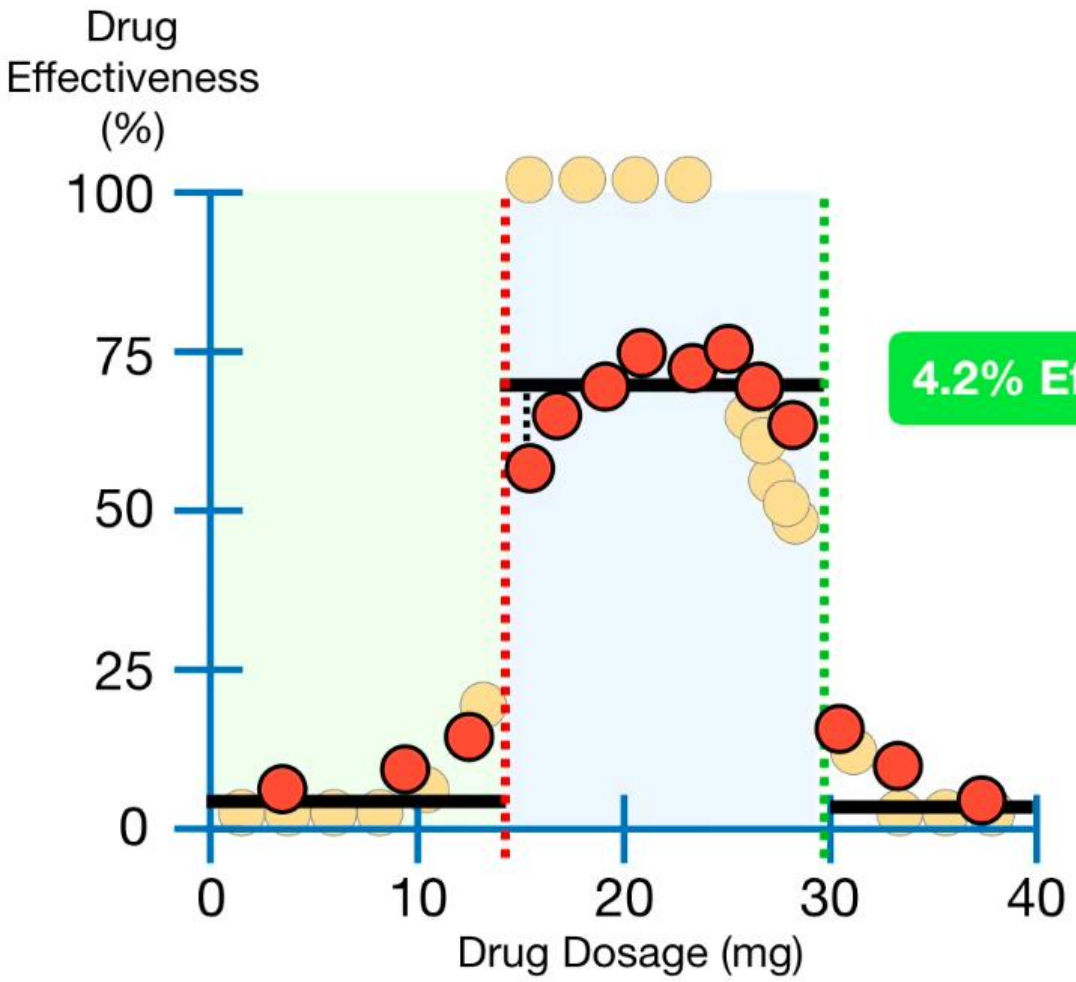


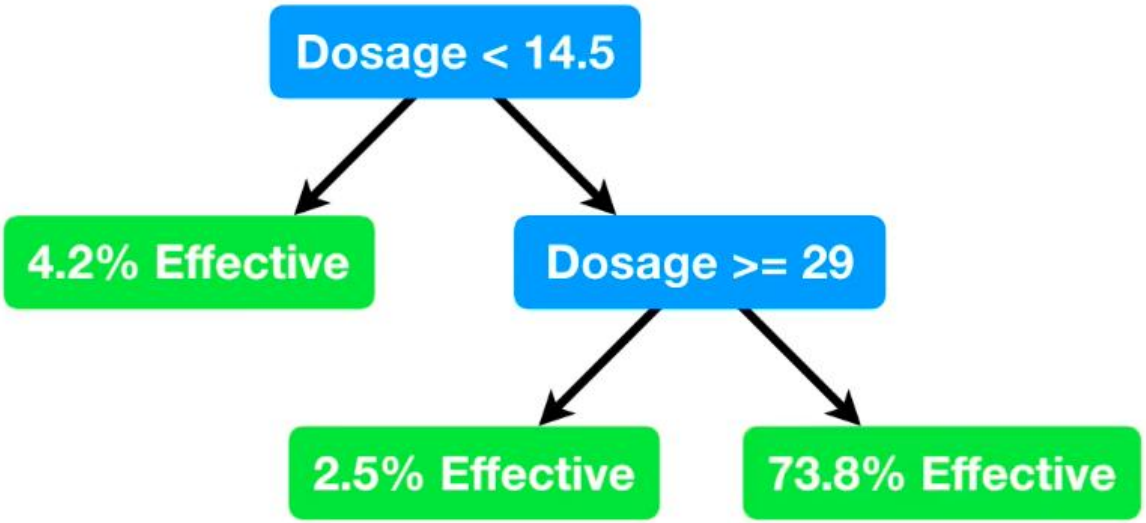
Thus, the main idea behind pruning a **Regression Tree** is to prevent overfitting the **Training Data**…

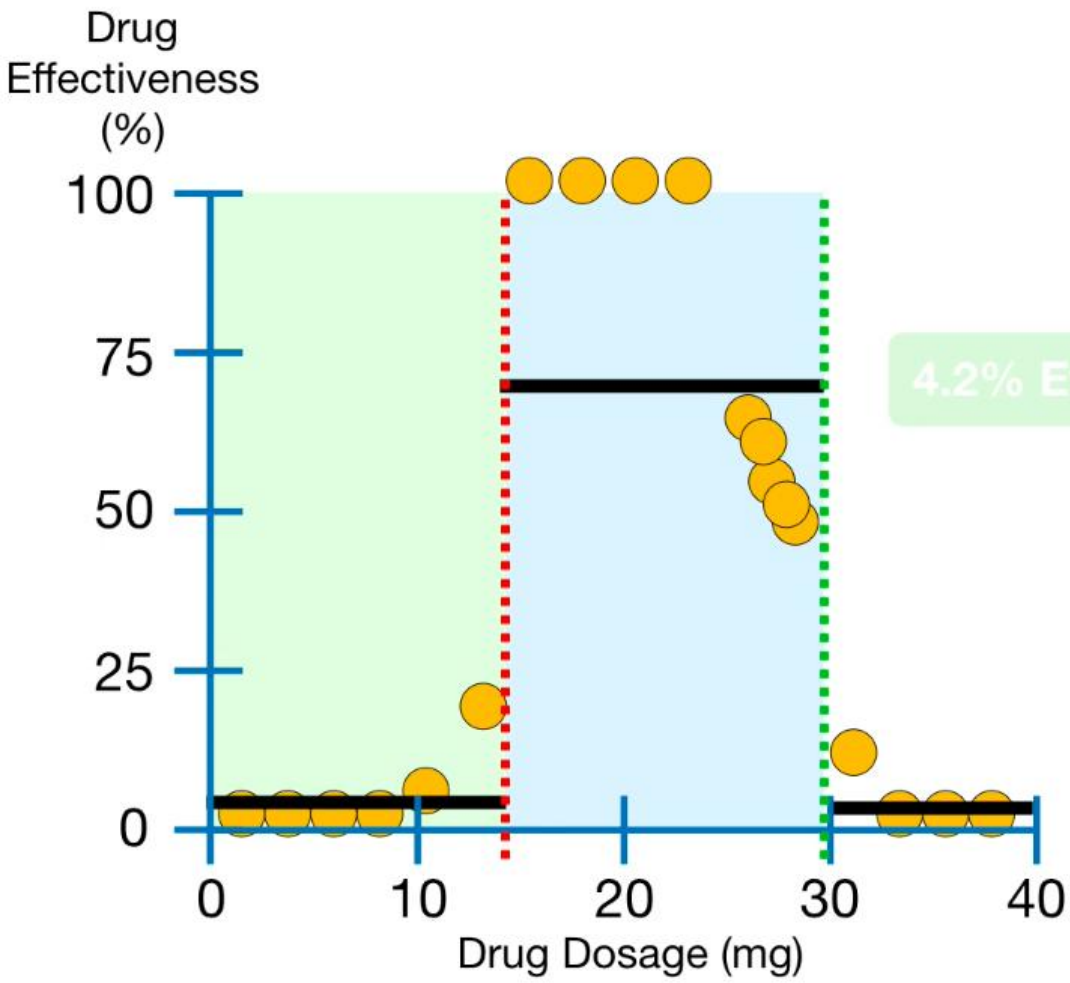




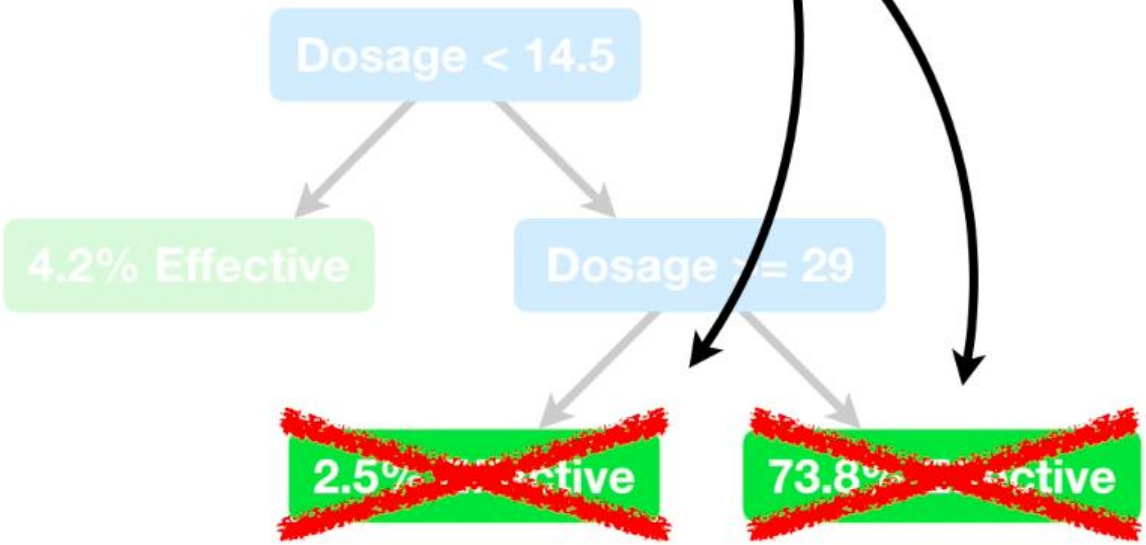
…so that the tree will do a better job with the **Testing Data**.



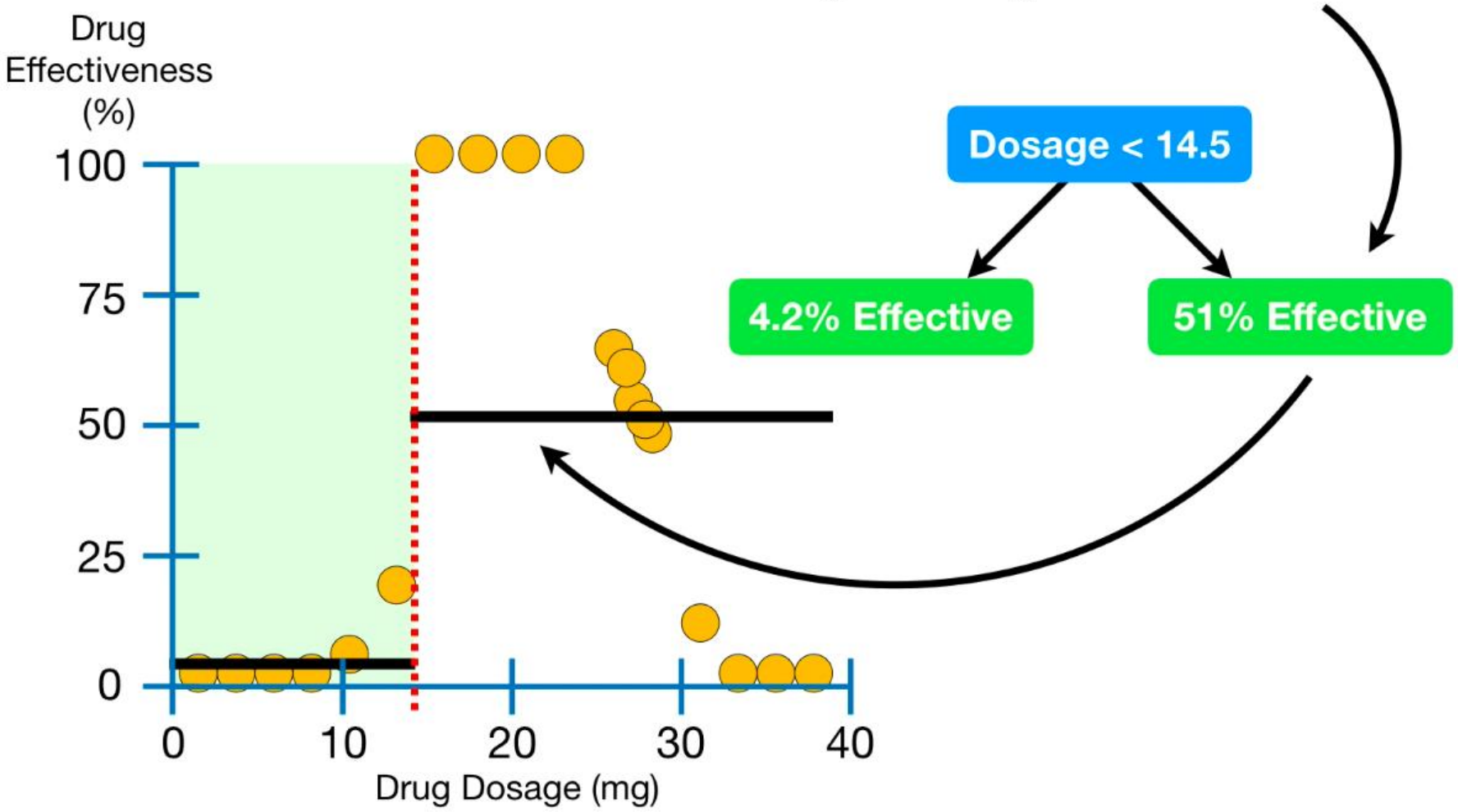


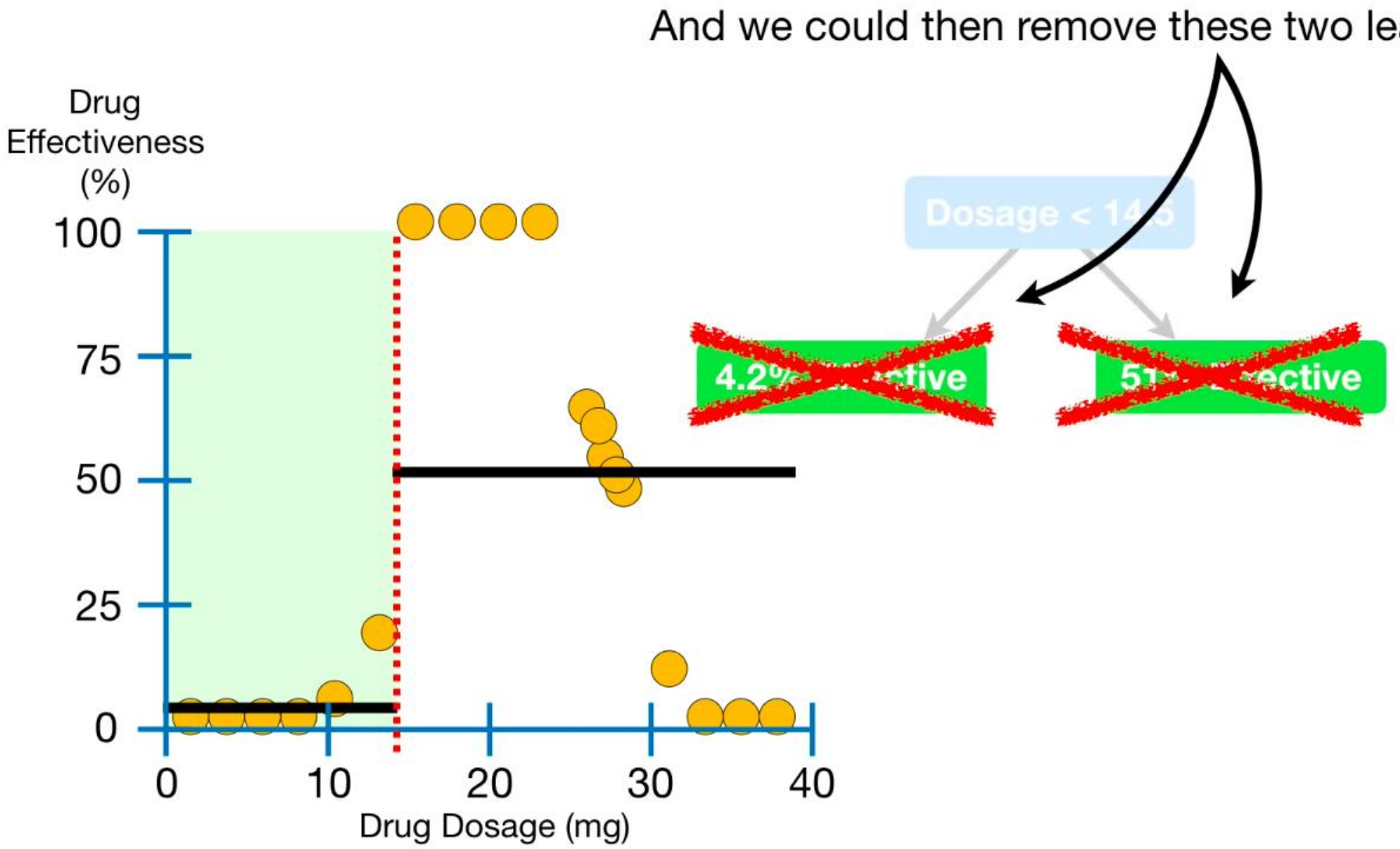


NOTE: If we wanted to prune the tree more, we could remove these two leaves…

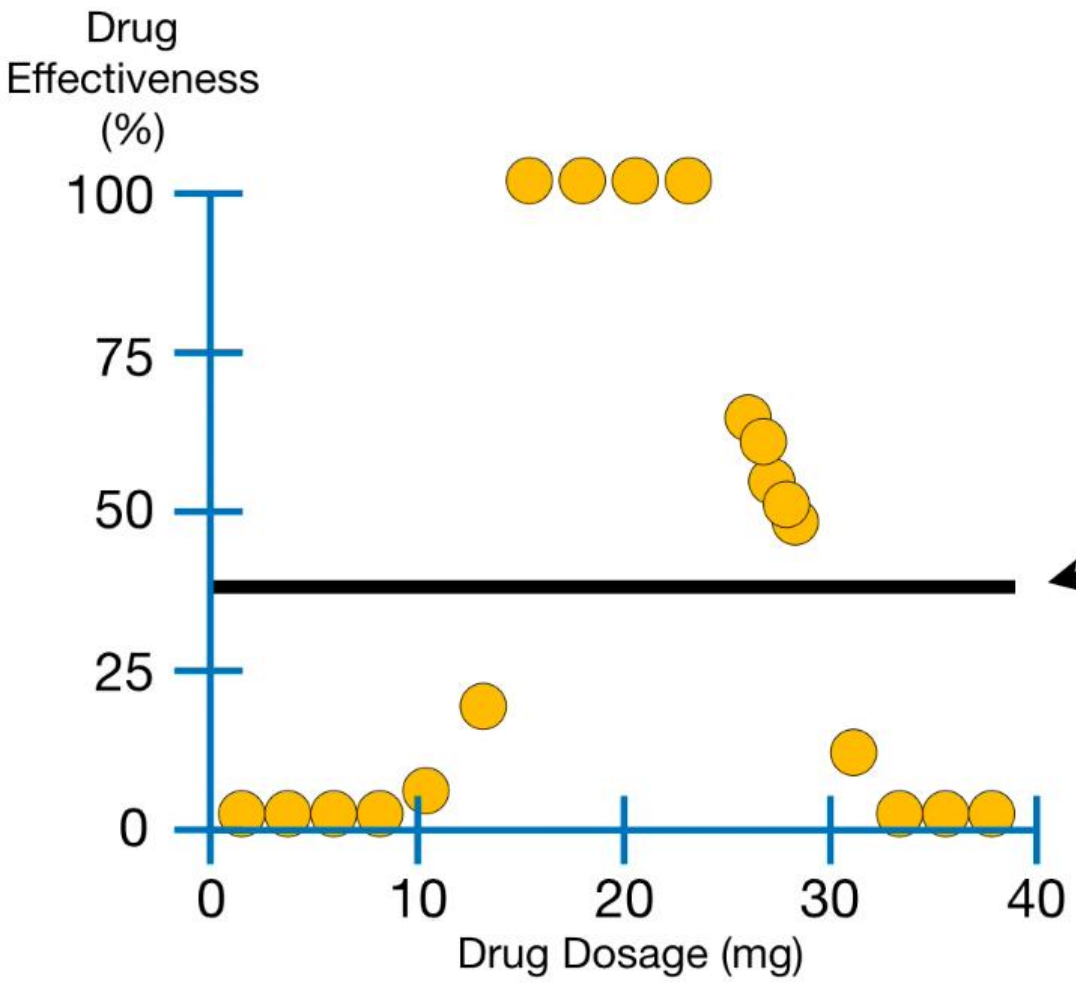


…and replace the split with a leaf that is the average of a larger number of observations.



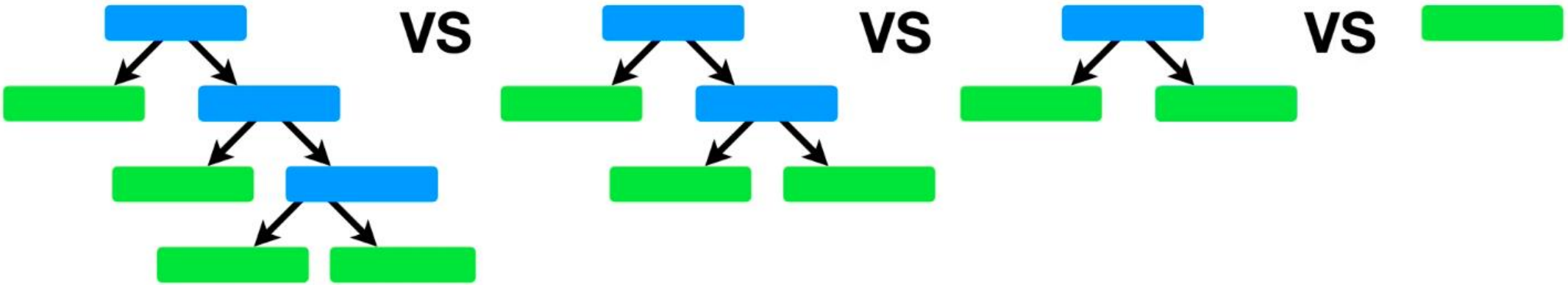

And we could then remove these two leaves…
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.6
4.2%
51

…and replace the split with a leaf that is the average of all of the observations.

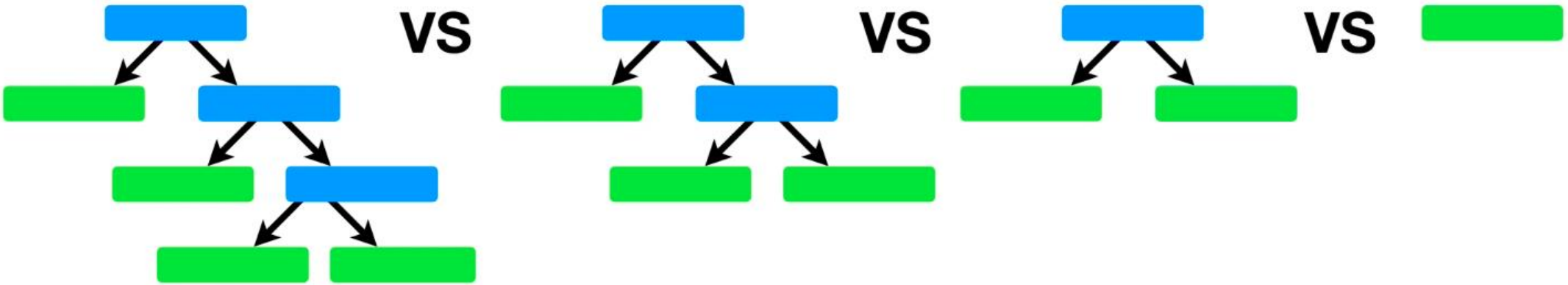


So the question is:

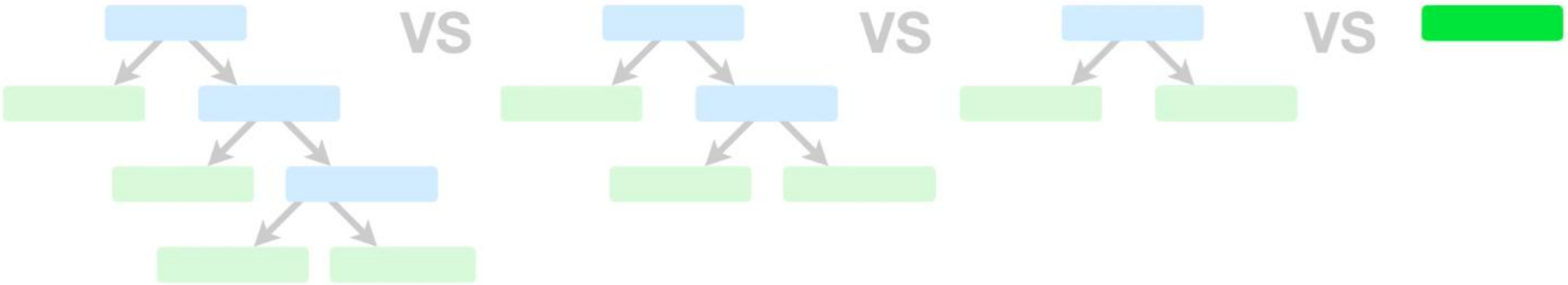
“How do we decide which tree to use?”



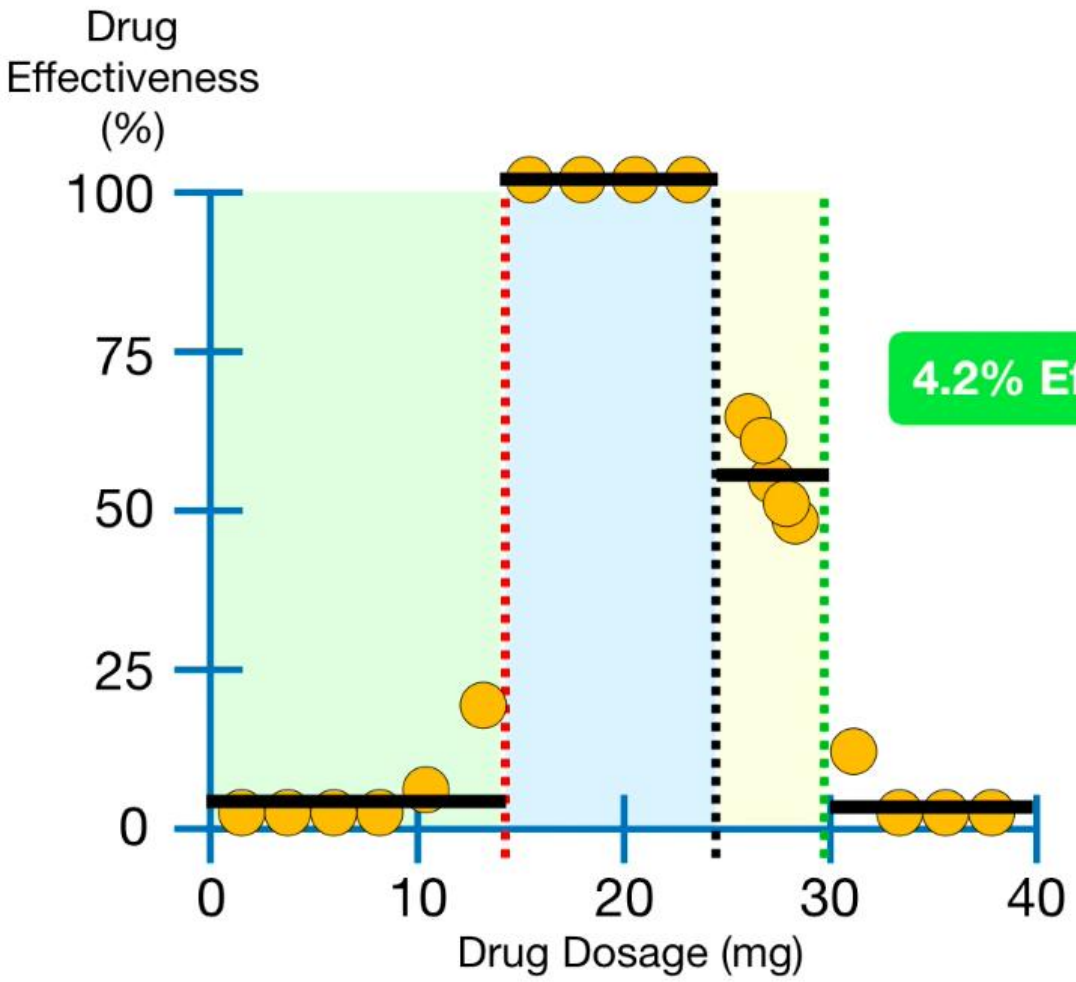
In this **StatQuest**, we will answer that question with **Cost Complexity Pruning**.

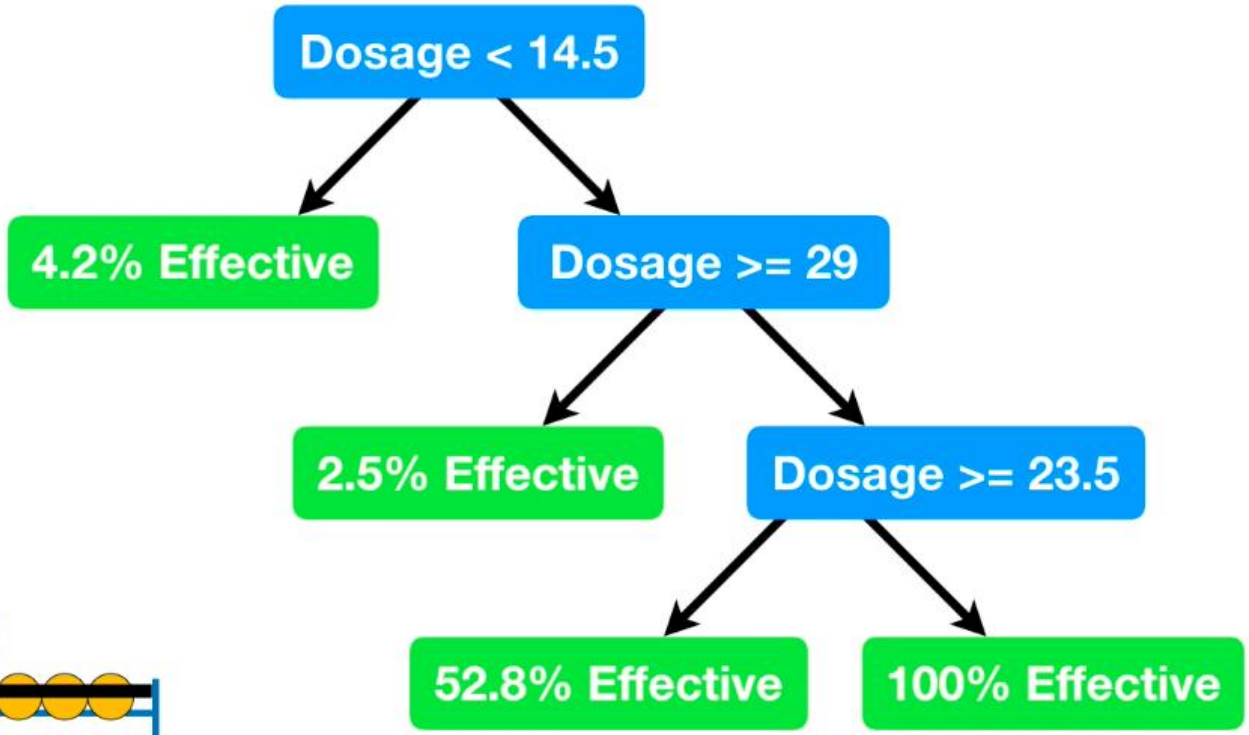


The first step in **Cost Complexity Pruning** is to calculate the **Sum of Squared Residuals** for each tree.

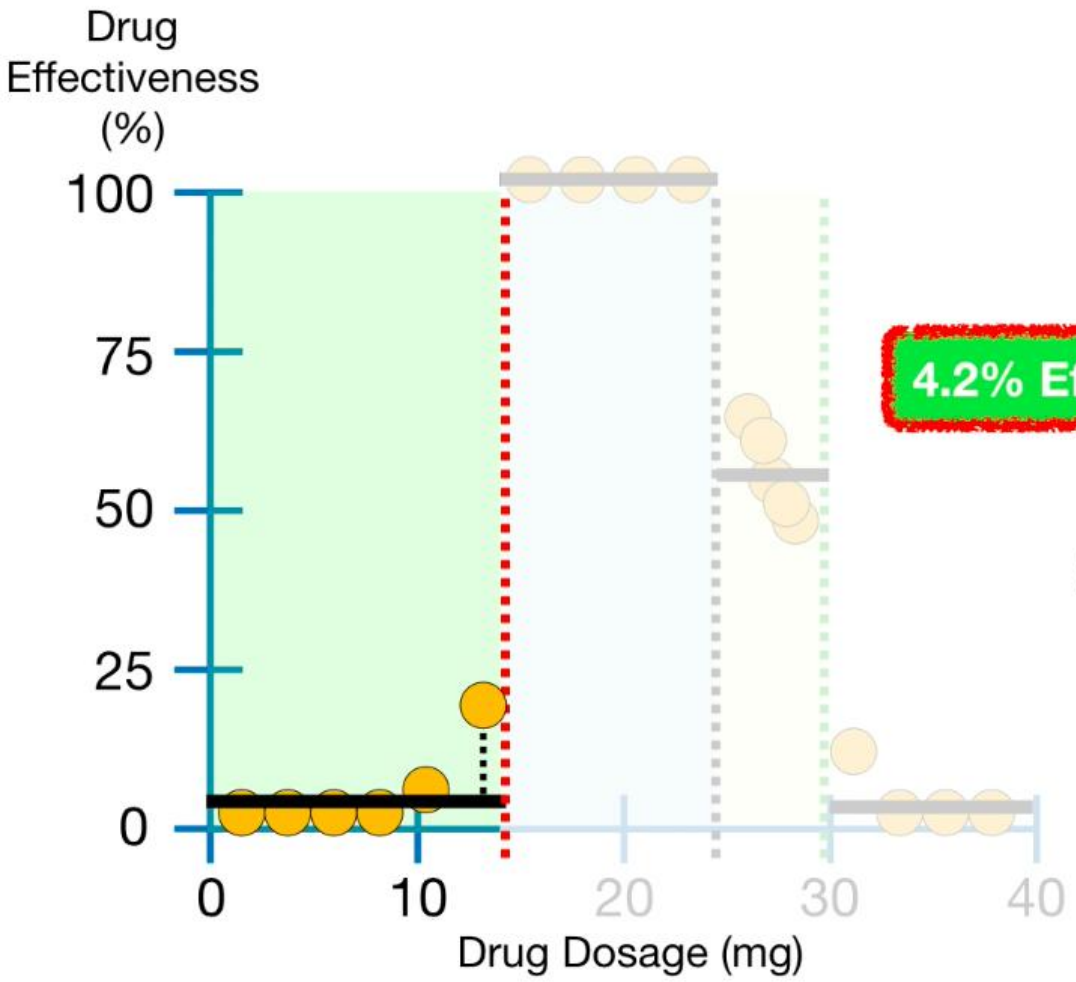


Here is the original, full sized tree.





The **Sum of Squared Residuals** for the **Observations** with **Dosages < 14.5** is…



Dosage < 14.5

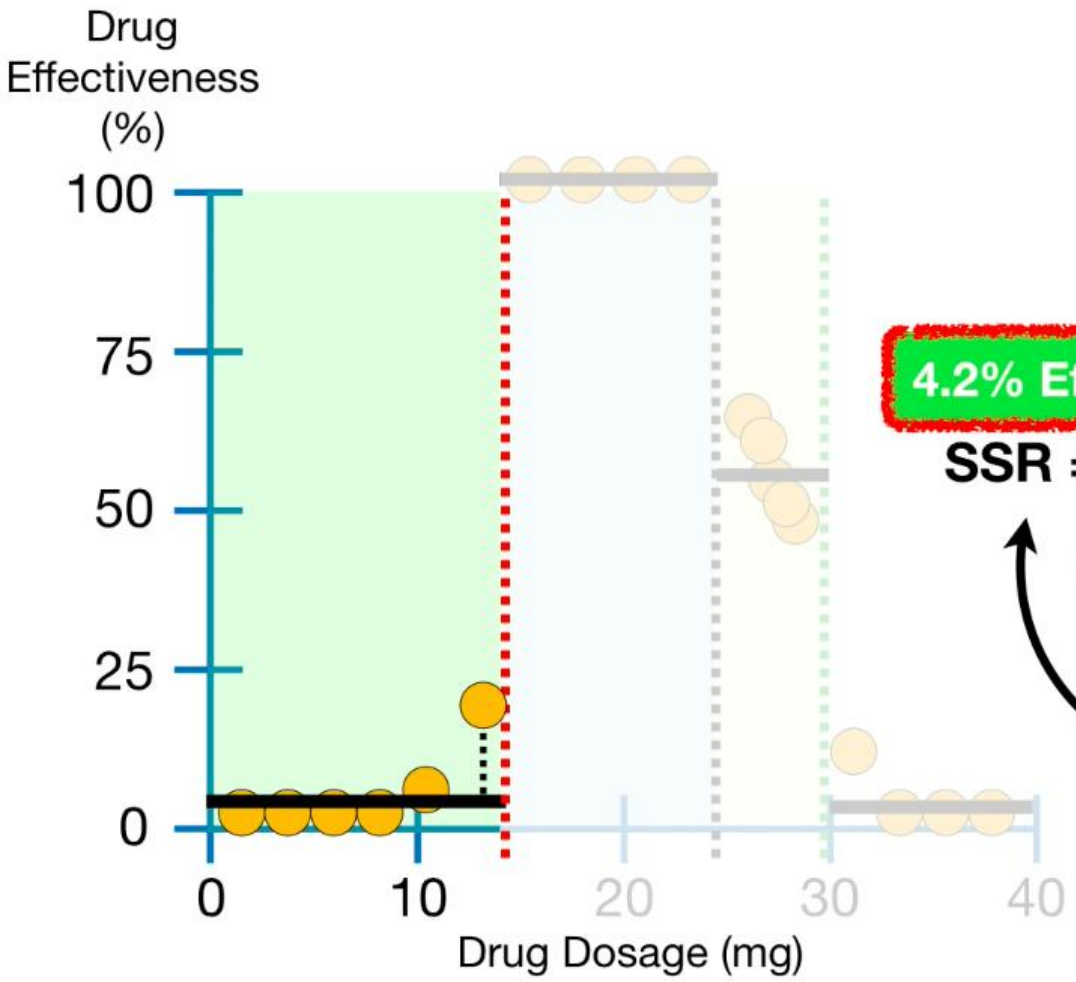
4.2% Effective

$(0 - 4.2)^2 + (0 - 4.2)^2 + (0 - 4.2)^2 + (0 - 4.2)^2$

$+ (5 - 4.2)^2 + (20 - 4.2)^2$

= 320.8

So we'll save that **Sum of Squared Residuals** (**SSR**) underneath the corresponding leaf.



Dosage < 14.5

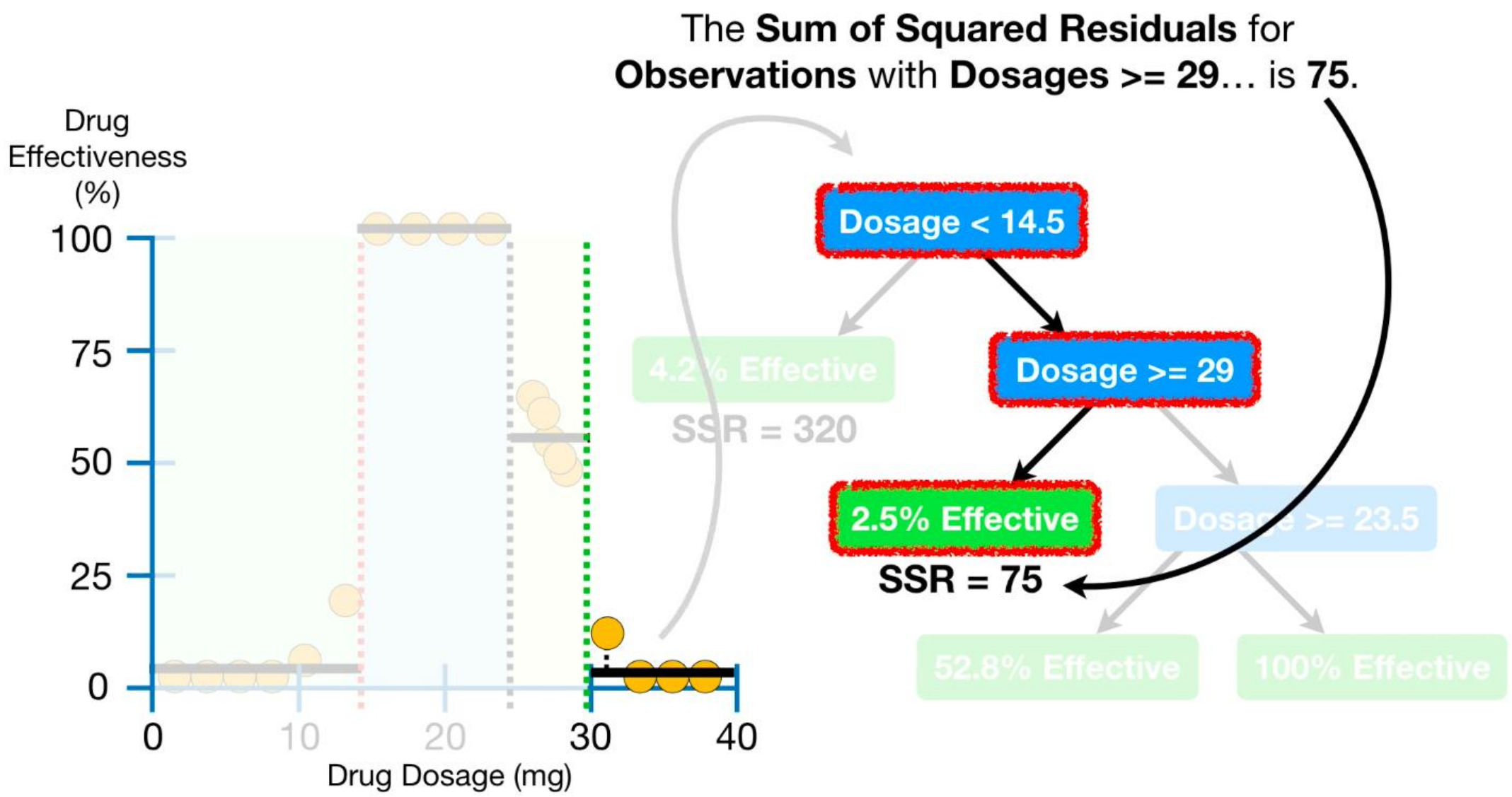
4.2% Effective

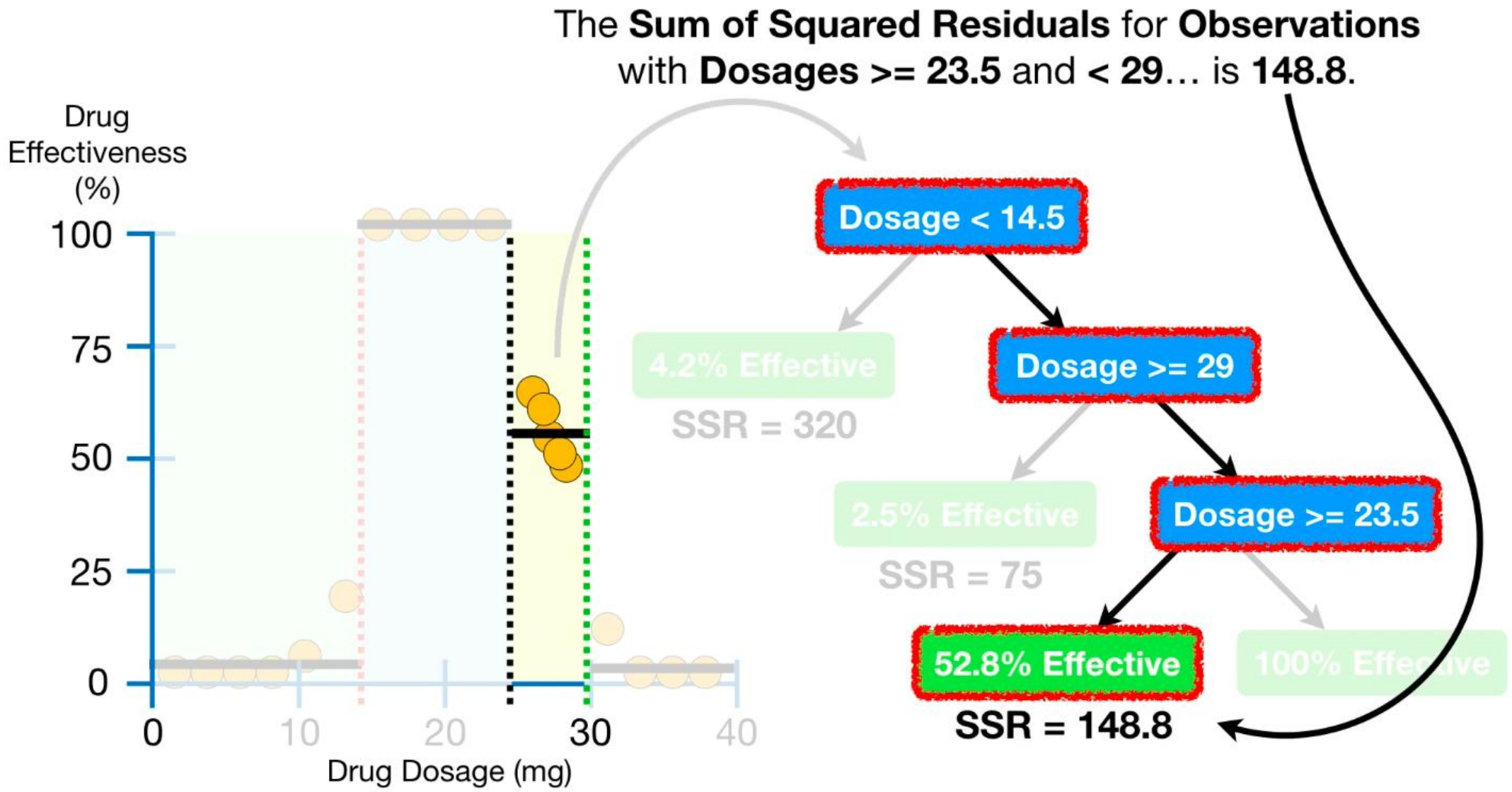
SSR = 320

$(0 - 4.2)^2 + (0 - 4.2)^2 + (0 - 4.2)^2 + (0 - 4.2)^2$

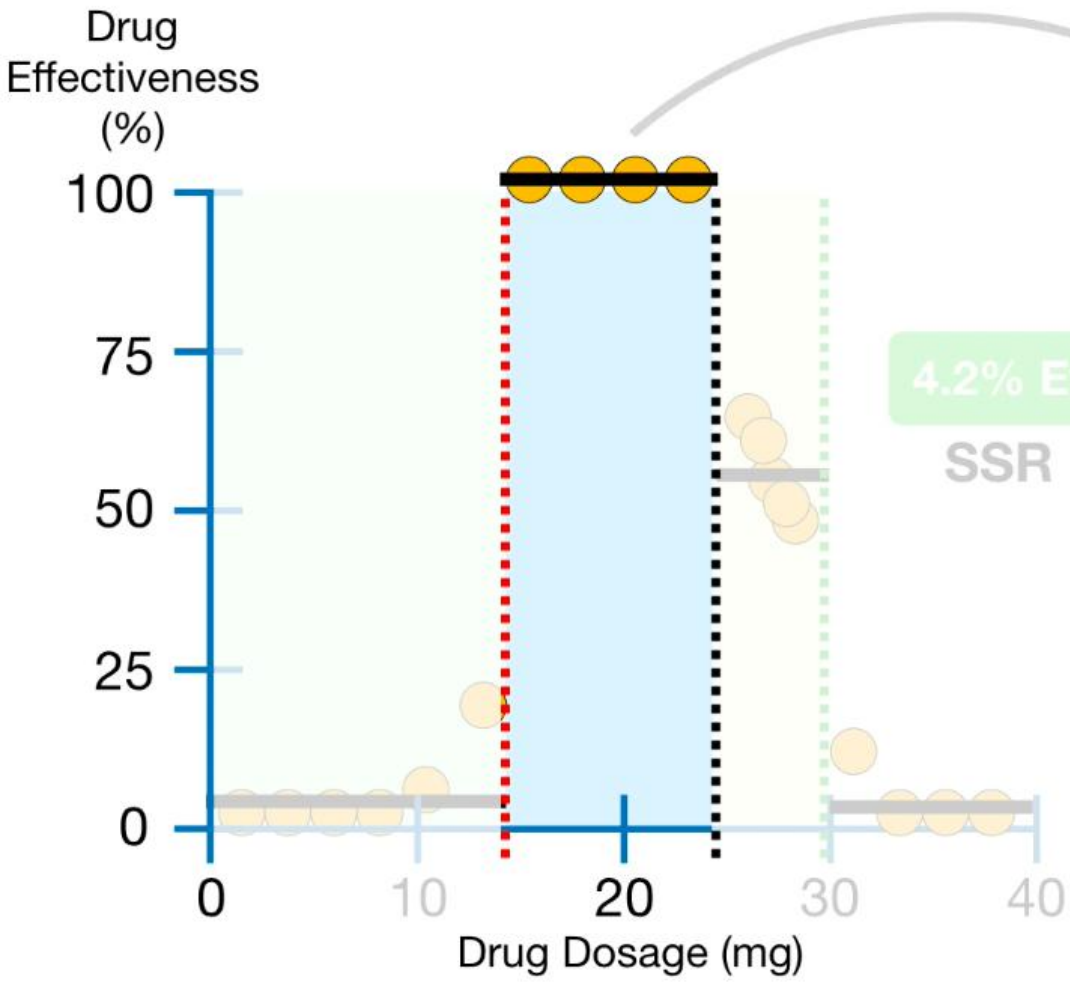
$+ (5 - 4.2)^2 + (20 - 4.2)^2$

= 320.8


The Sum of Squared Residuals for Observations with Dosages >= 29... is 75.
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
SSR = 320
Dosage >= 29
2.5% Effective
SSR = 75
Dosage >= 23.5
52.8% Effective
100% Effective


The Sum of Squared Residuals for Observations with Dosages >= 23.5 and < 29... is 148.8.
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
SSR = 320
Dosage >= 29
2.5% Effective
SSR = 75
Dosage >= 23.5
52.8% Effective
100% Effective
SSR = 148.8

And the **Sum of Squared Residuals** for **Observations** with **Dosages >= 14.5** and **< 23.5**… is **0**.



Dosage < 14.5

4.2% Effective

SSR = 320

Dosage >= 29

2.5% Effective

SSR = 75

Dosage >= 23.5

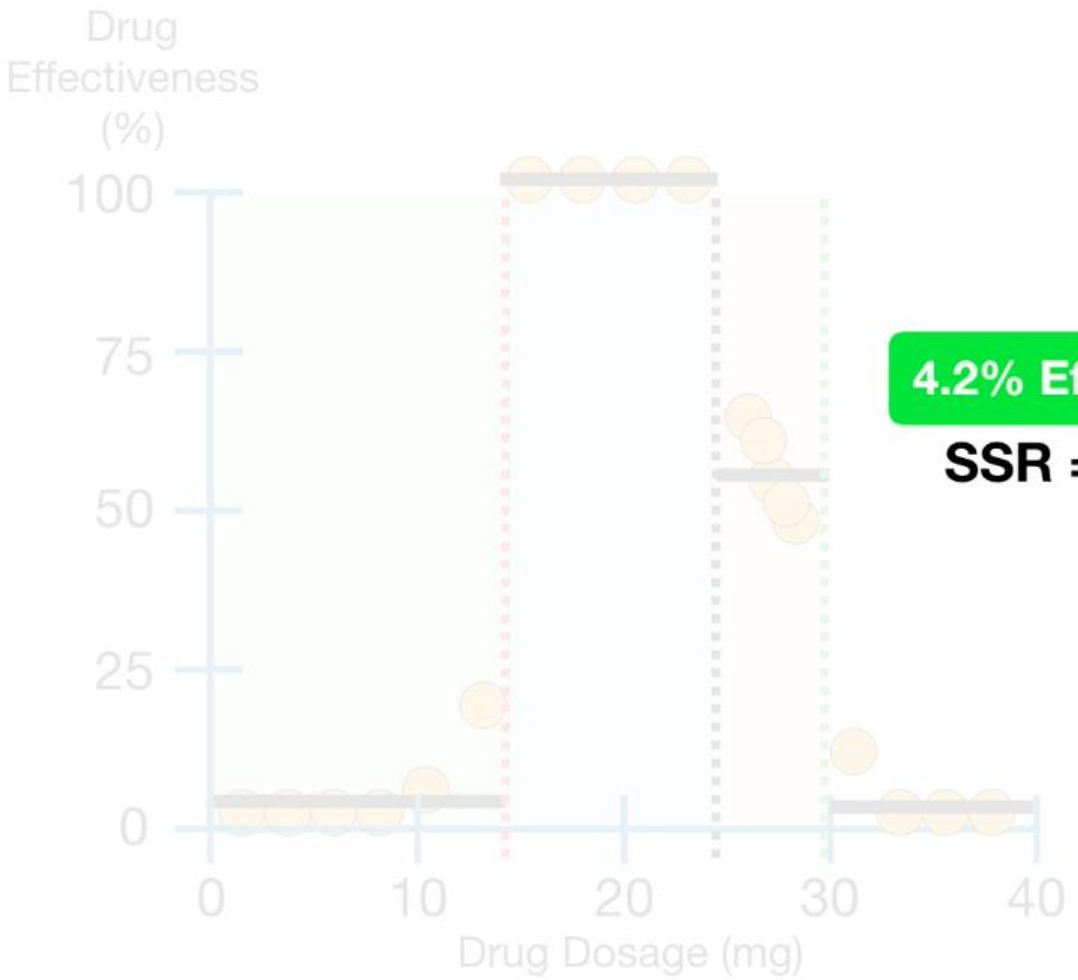
52.8% Effective

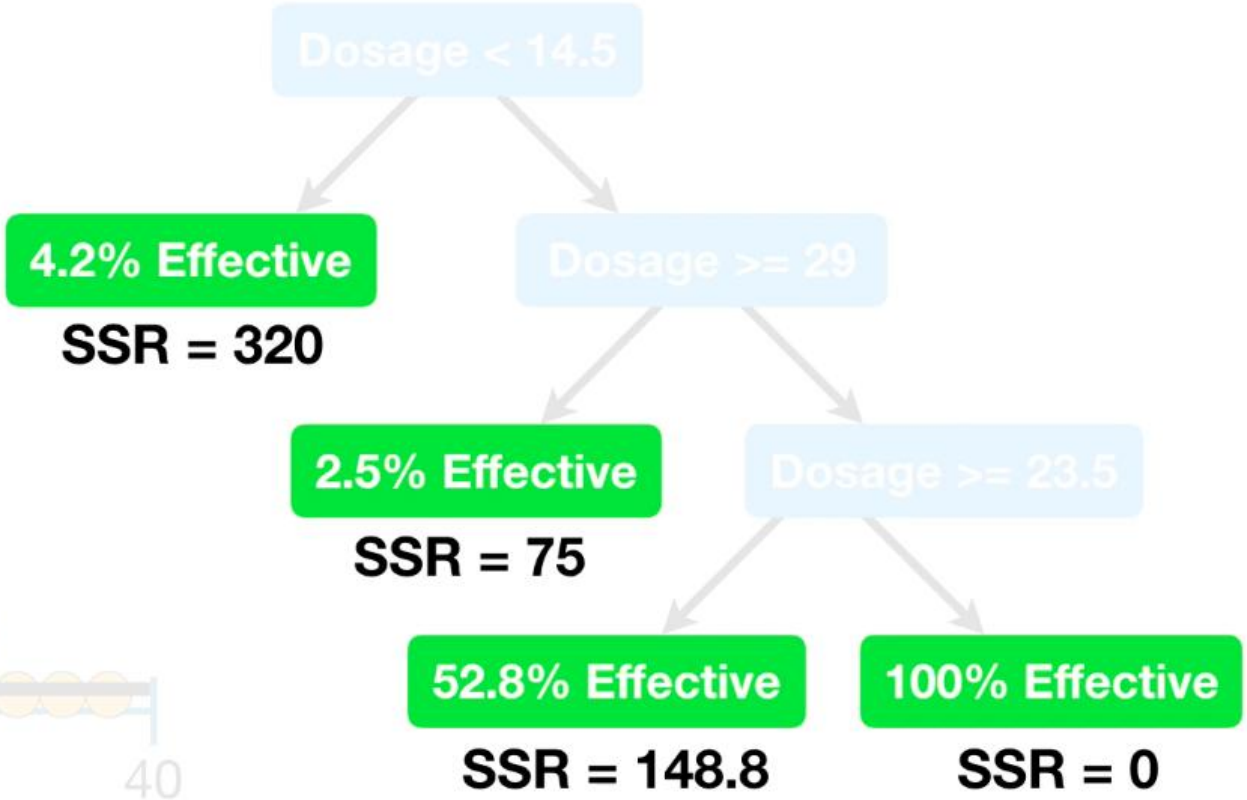
SSR = 148.8

100% Effective

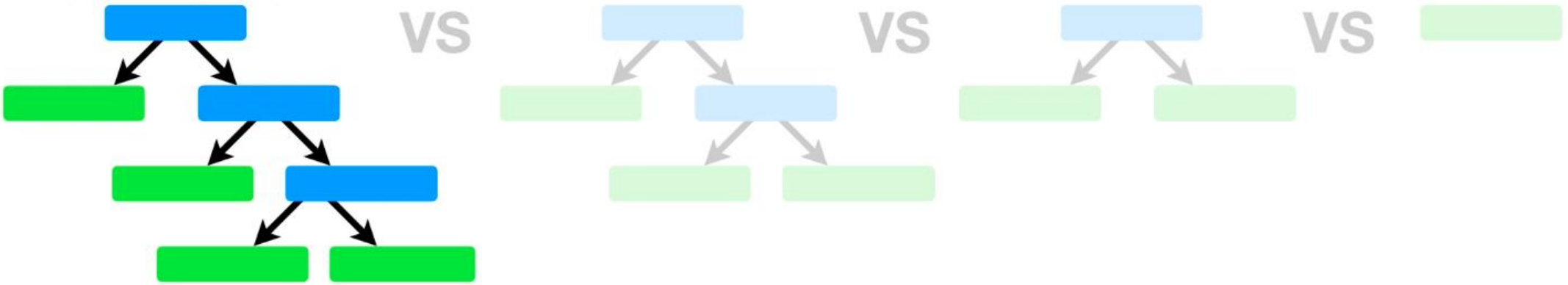
SSR = 0

Thus, the total **Sum of Squared Residuals** (SSR) for the whole tree is 320 + 75 + 148.8 + 0 = **543.8**

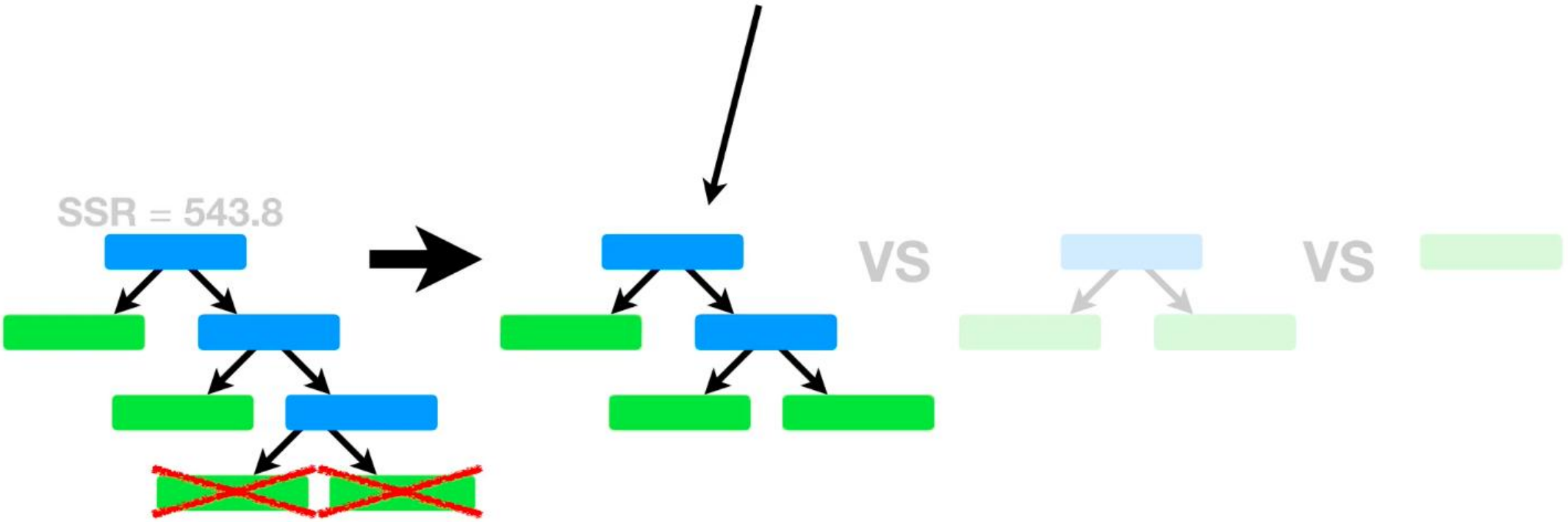




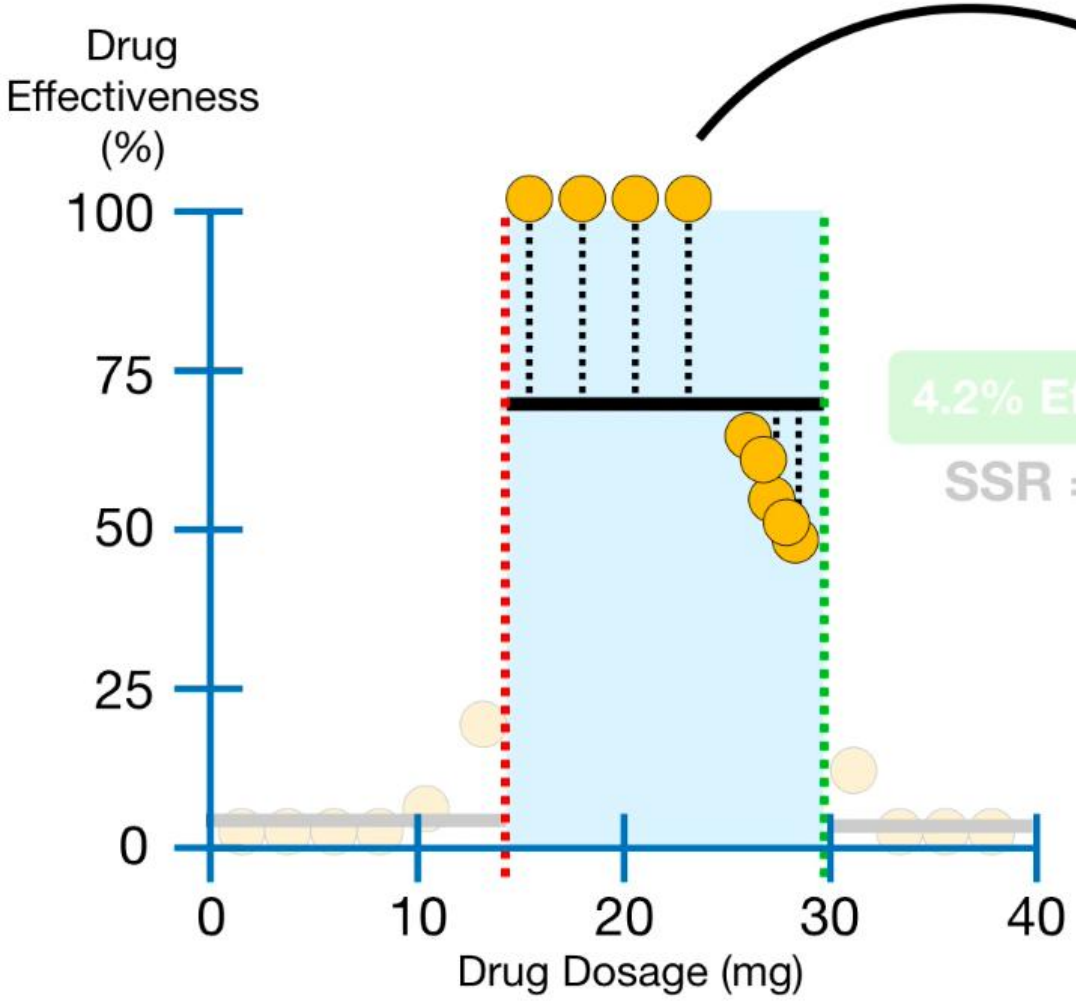
So let's put **SSR = 543.8** on top of the original, full sized tree.

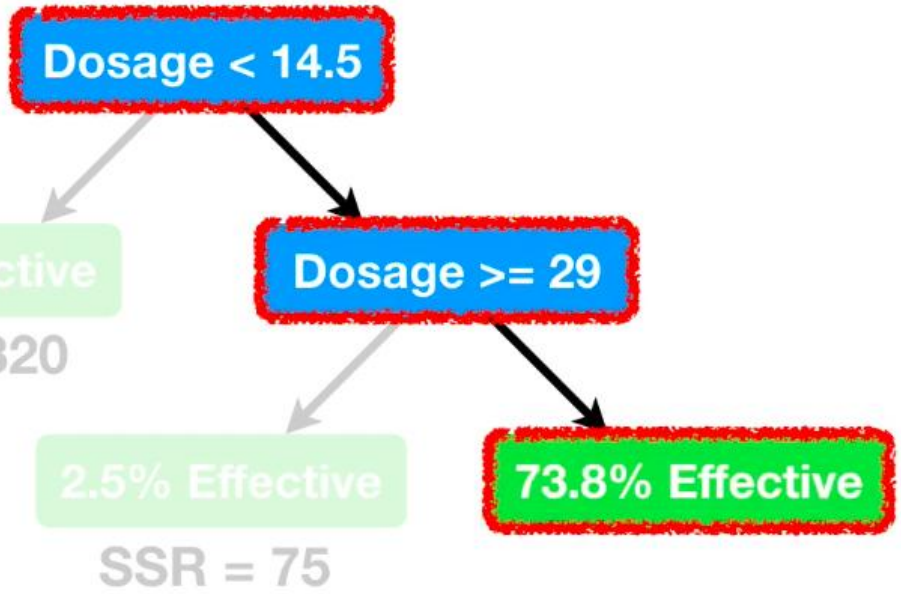


Now let's calculate the **Sum of Squared Residuals** for the sub-tree with one fewer leaf.

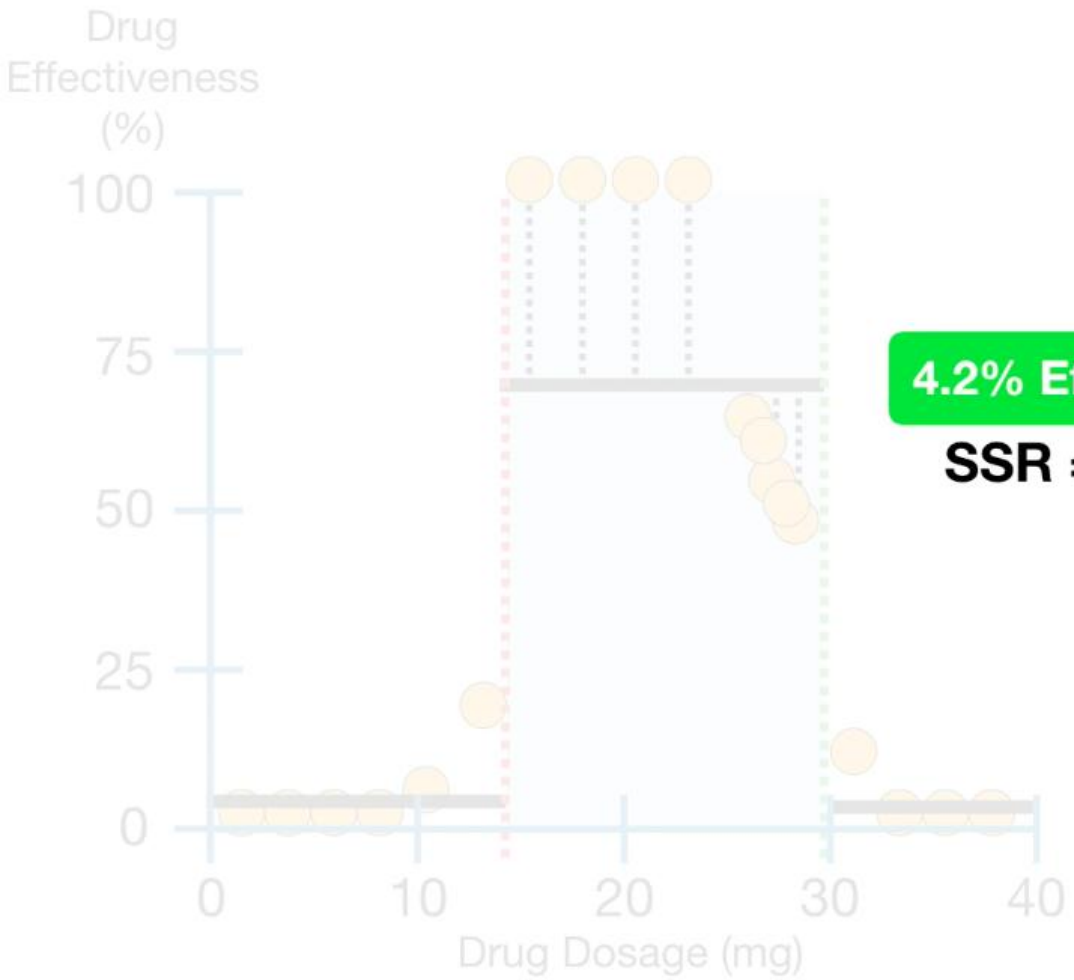


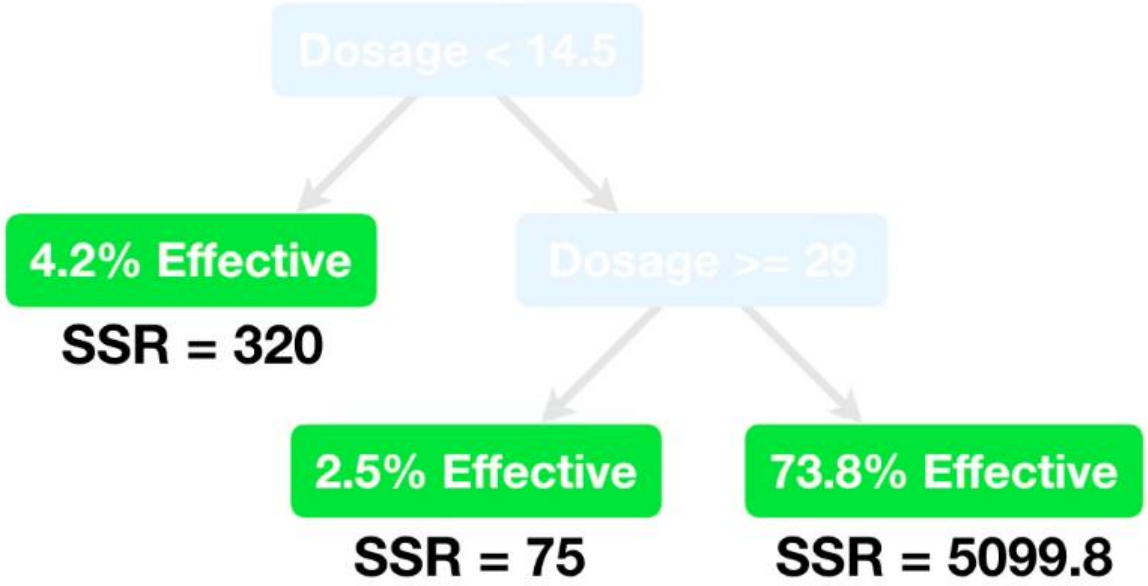
…but we have to calculate a new **Sum of Squared Residuals** for when the **Dosage** is between **14.5** and **29**.



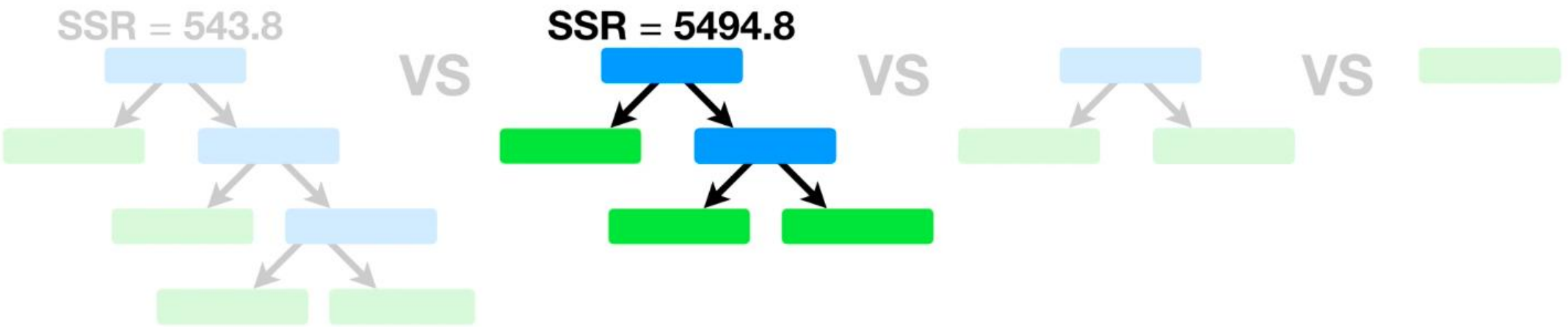


Thus, the total **Sum of Squared Residuals** for this tree is 320 + 75 + 5099.8 = **5494.8**

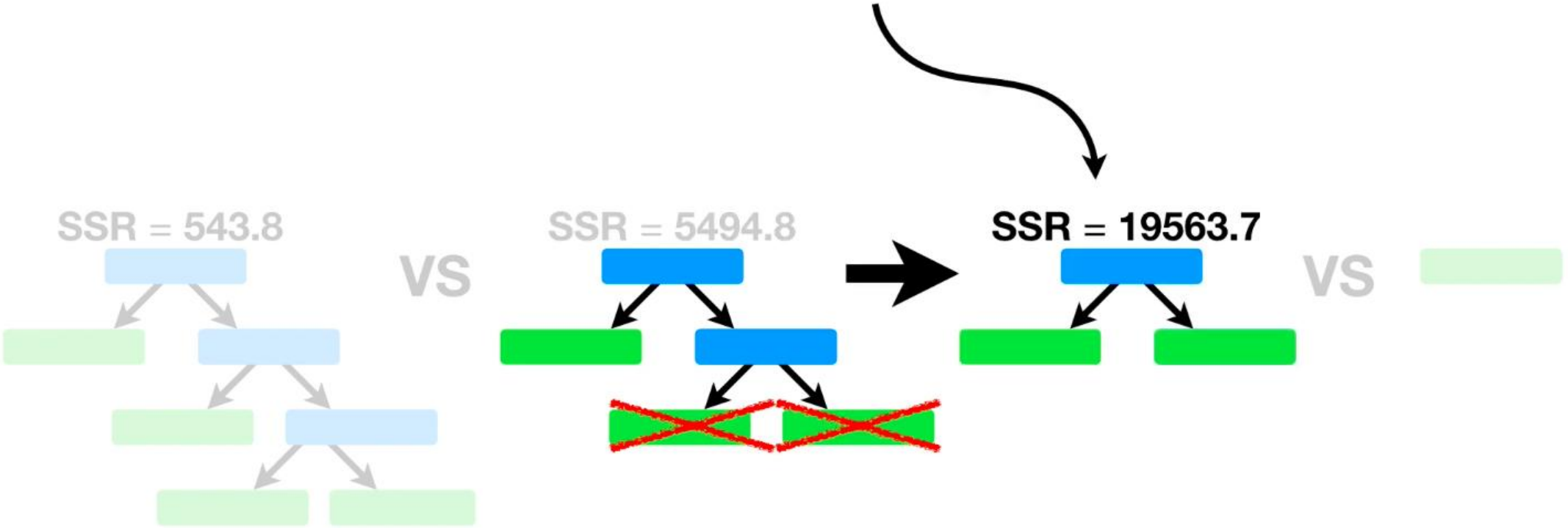




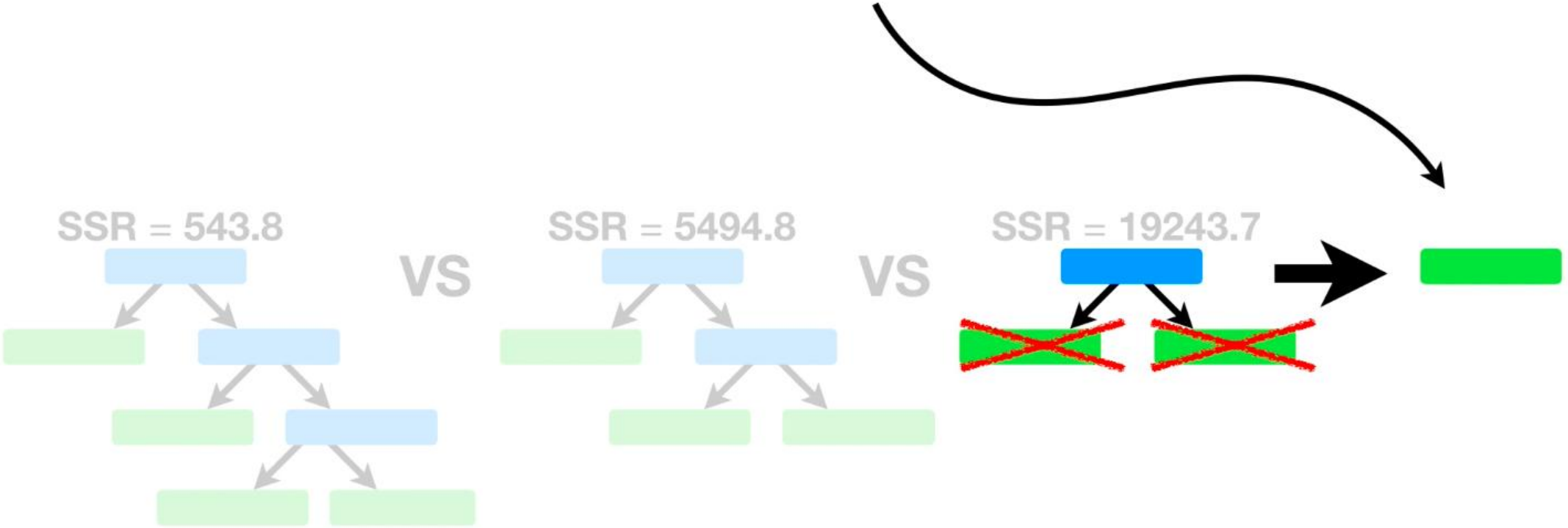
So let's put **SSR** = **5494.8** on top of the sub-tree with **3** leaves.



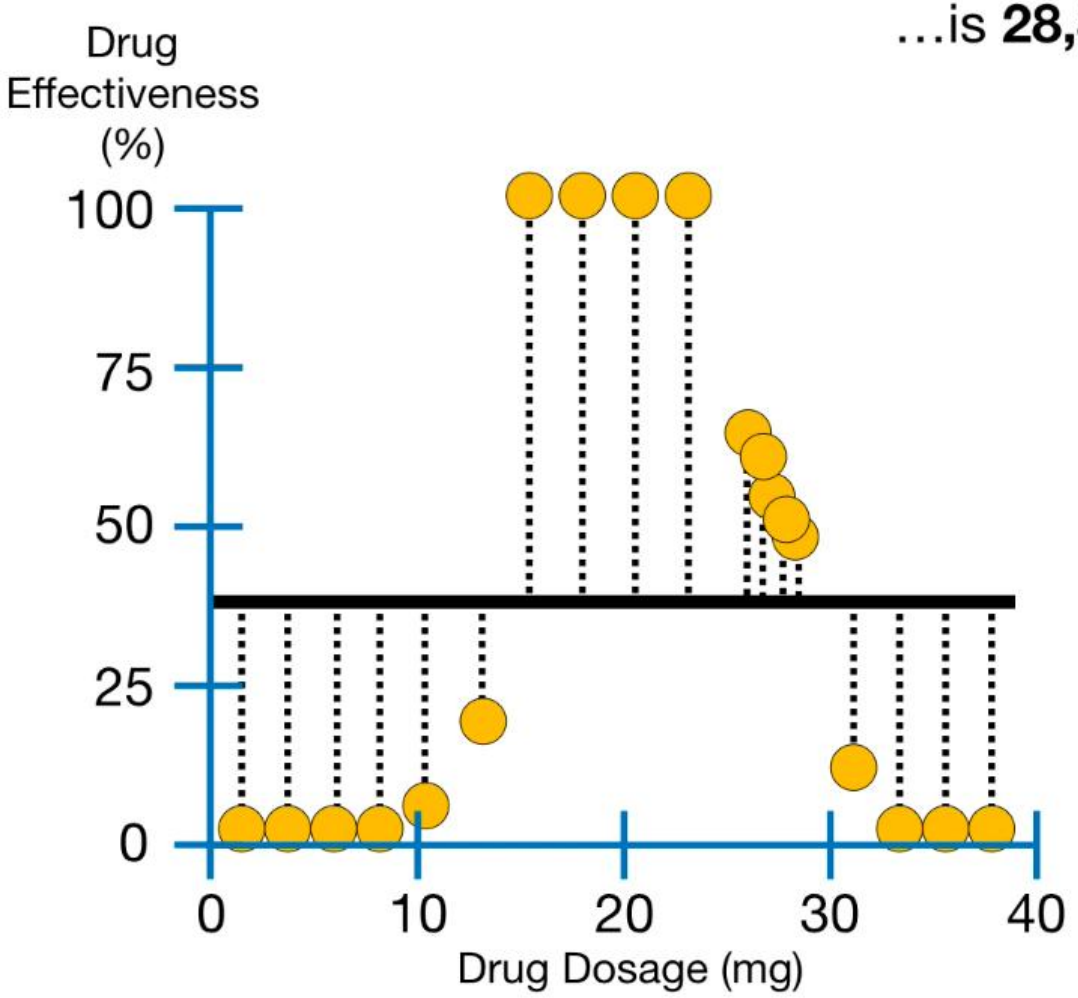
Similarly, the **Sum of Squared Residuals** for the sub-tree with **2** leaves…



Lastly, the **Sum of Squared Residuals** for the sub-tree with only one leaf…

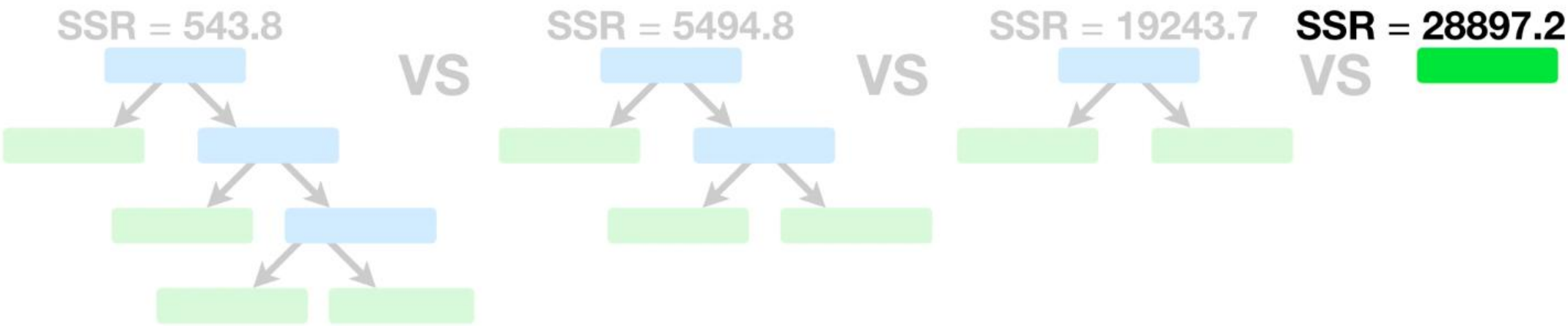


…is **28,897.2**.

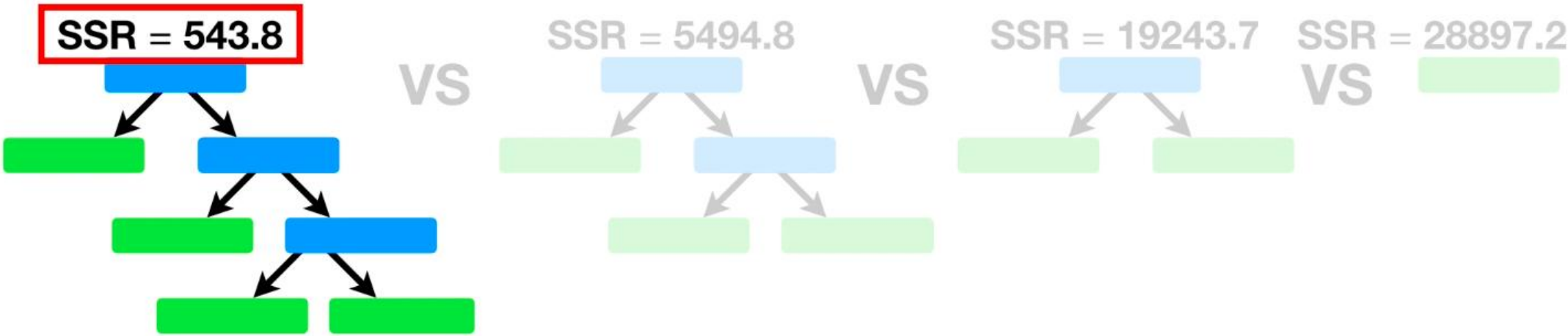


37% Effective

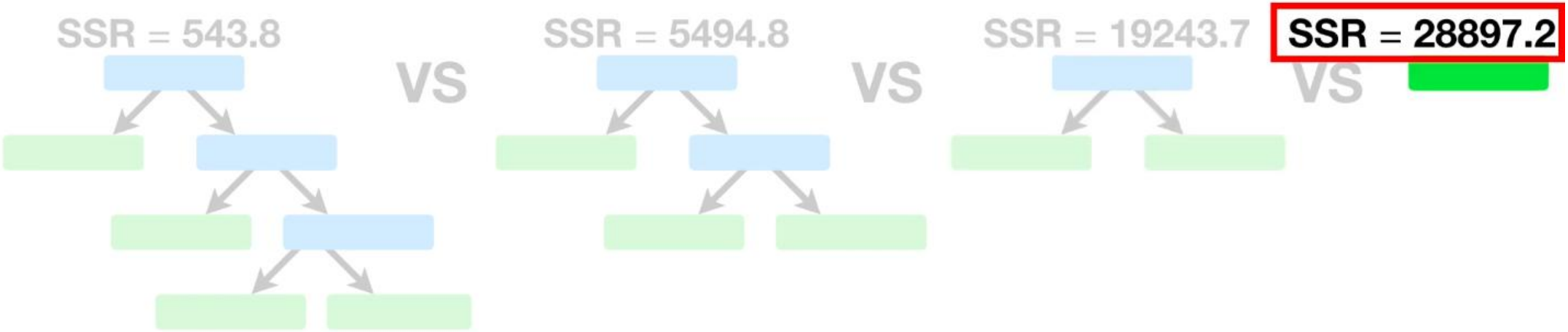
…so let’s put **SSR = 28897.2** on top of the sub-tree with **1** leaf.



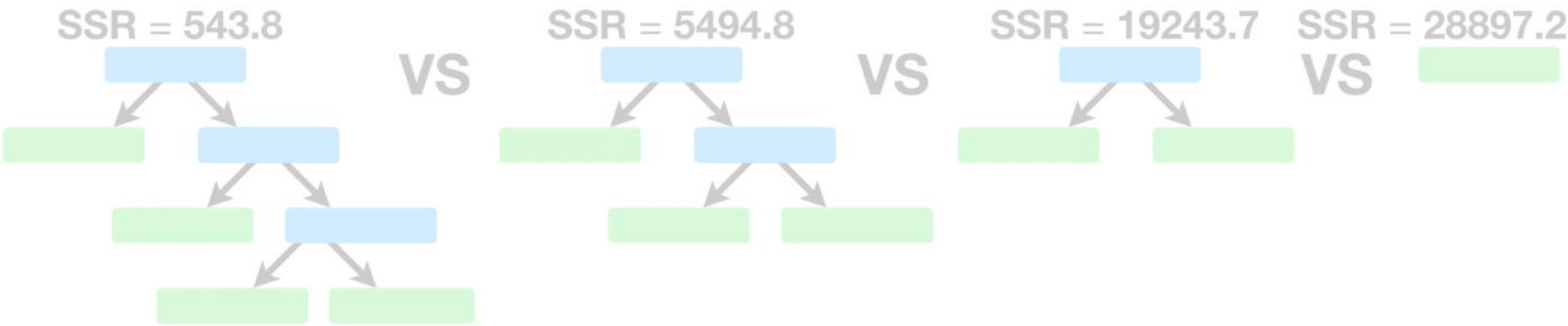
NOTE: The **Sum of Squared Residuals** is relatively small for the original, full sized tree…



…but each time we remove a leaf, the **Sum of Squared Residuals** gets larger and larger.

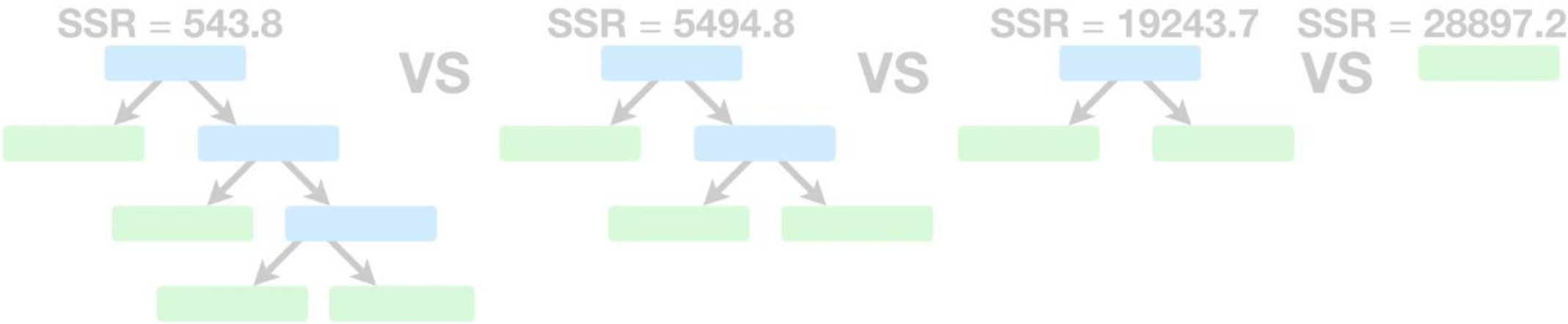


However, we knew that was going to happen because the whole idea was for the pruned trees to ***not*** fit the **Training Data** as well as the full sized tree.



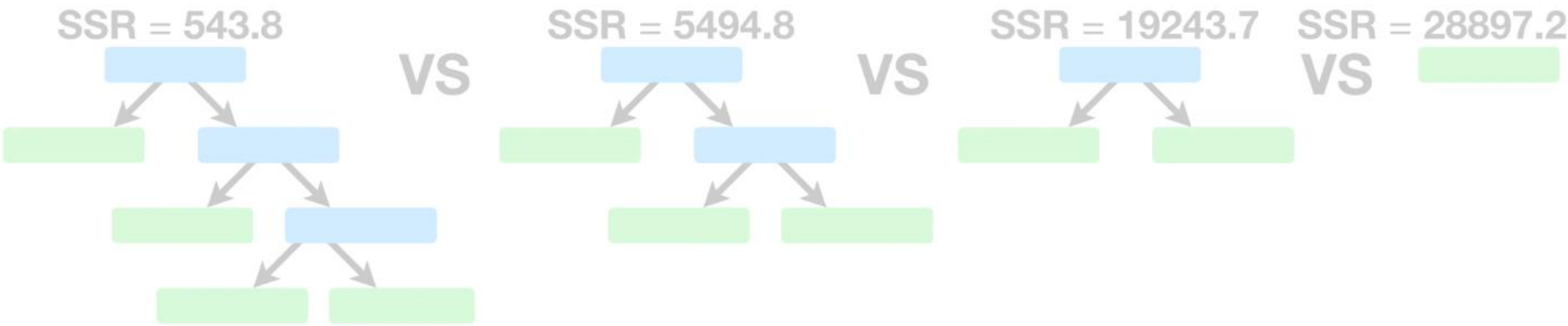
Weakest Link Pruning works by calculating a **Tree Score**…

Tree Score



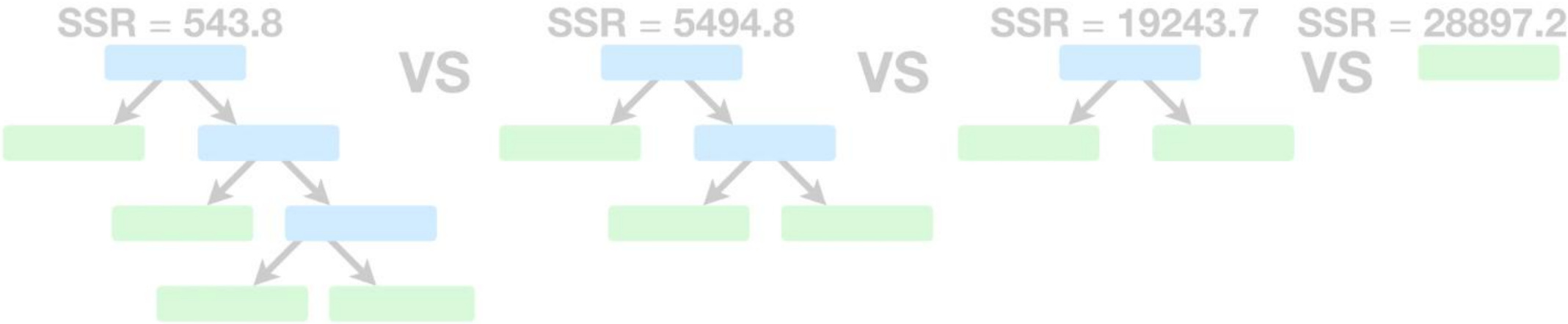
…that is based on the **Sum of Squared Residuals** (**SSR**) for the tree or sub-tree…

Tree Score = **SSR**



…and a **Tree Complexity Penalty** that is a function of the number of leaves, or ***T***erminal nodes, in the tree or sub-tree.

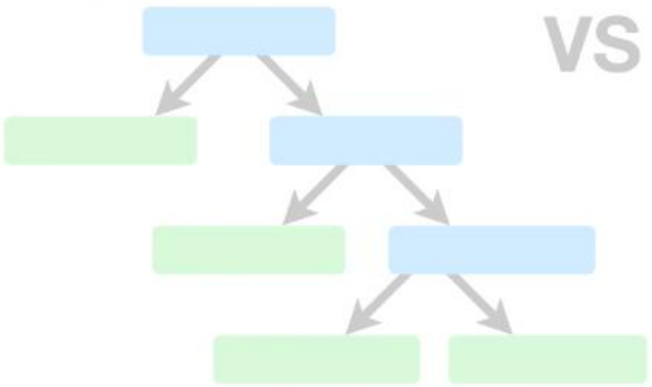
Tree Score = SSR + αT



The **Tree Complexity Penalty** compensates for the difference in the number of leaves.

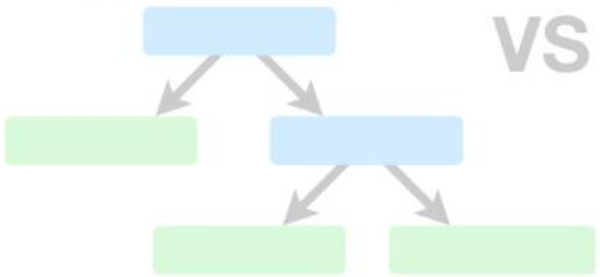
Tree Score = **SSR** + αT

SSR = 543.8



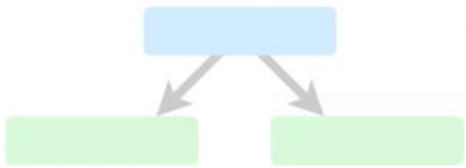
VS

SSR = 5494.8



VS

SSR = 19243.7



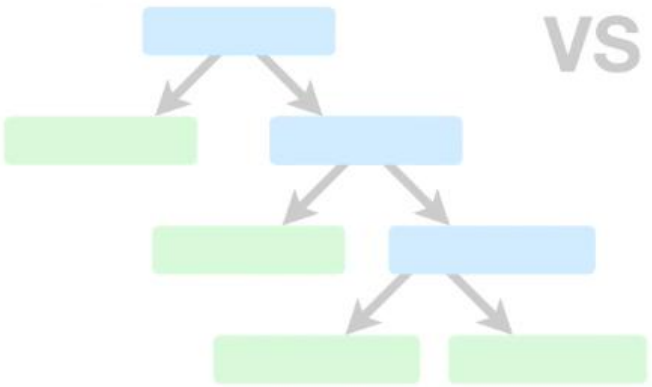
VS

SSR = 28897.2

NOTE: α (alpha) is a tuning parameter that we finding using **Cross Validation** and we'll talk more about it in a bit.

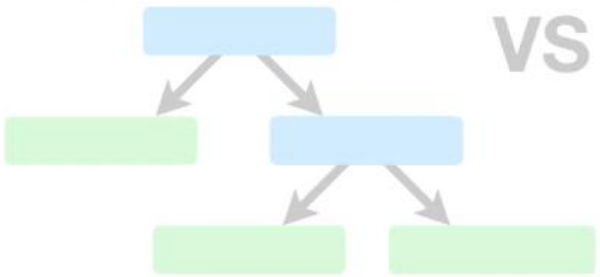
Tree Score = SSR + αT

SSR = 543.8



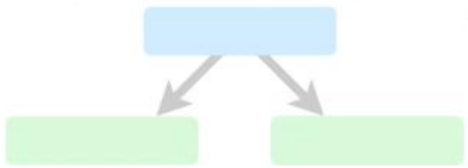
VS

SSR = 5494.8



VS

SSR = 19243.7

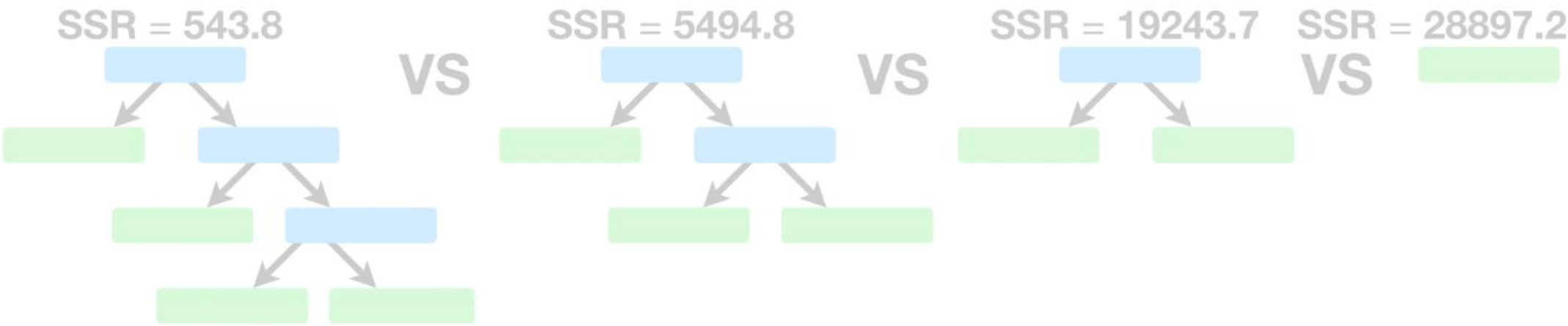


VS

SSR = 28897.2

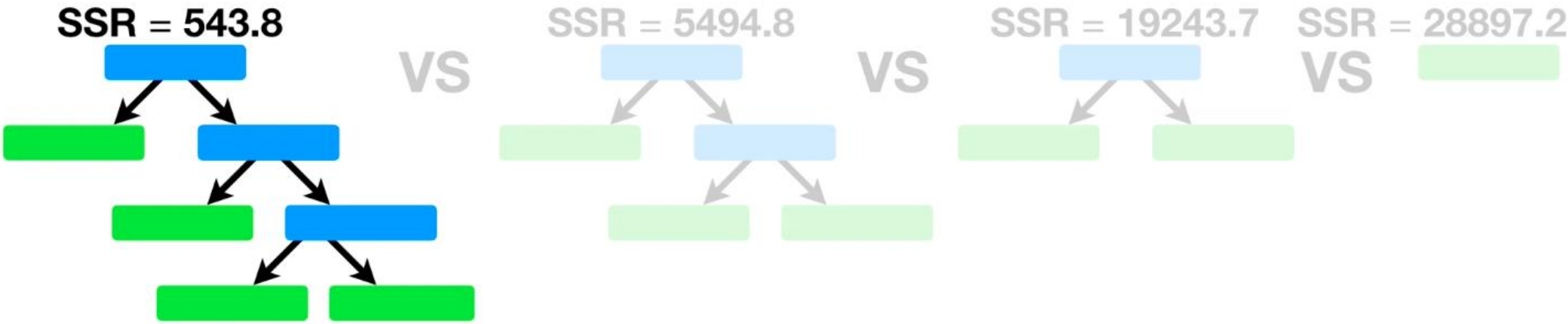
For now, let's let $\alpha = 10{,}000$.

Tree Score = SSR + 10,000 × *T*



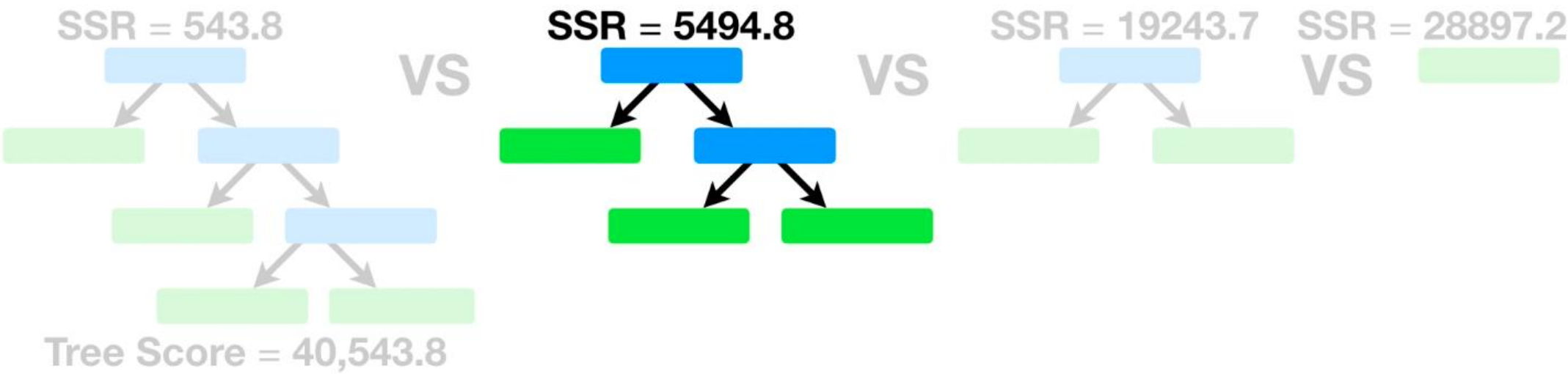
So the **Tree Score** for the original,
full sized tree is **40,543.8**.

Tree Score = **543.8** + **10,000** × **4** = 40,543.8



…and the total
Tree Score = 35,494.8.

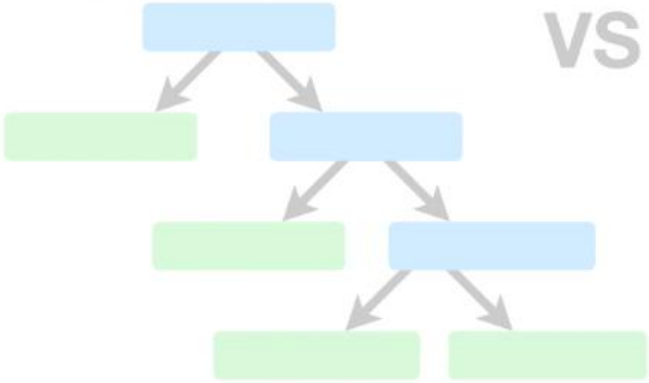
Tree Score = 5494.8 + 10,000 x 3 = 35494.8



…39,243.7.

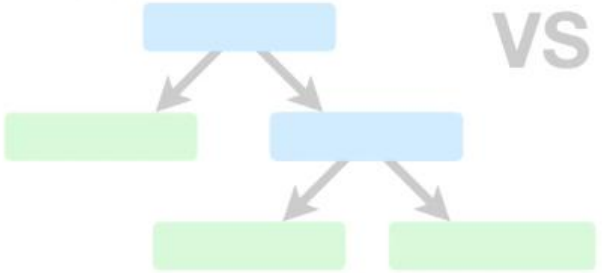
Tree Score = 19243.7 + 10,000 x 2 = 39243.7

SSR = 543.8



VS

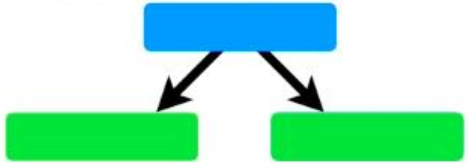
SSR = 5494.8



Tree Score = 35,494.8

VS

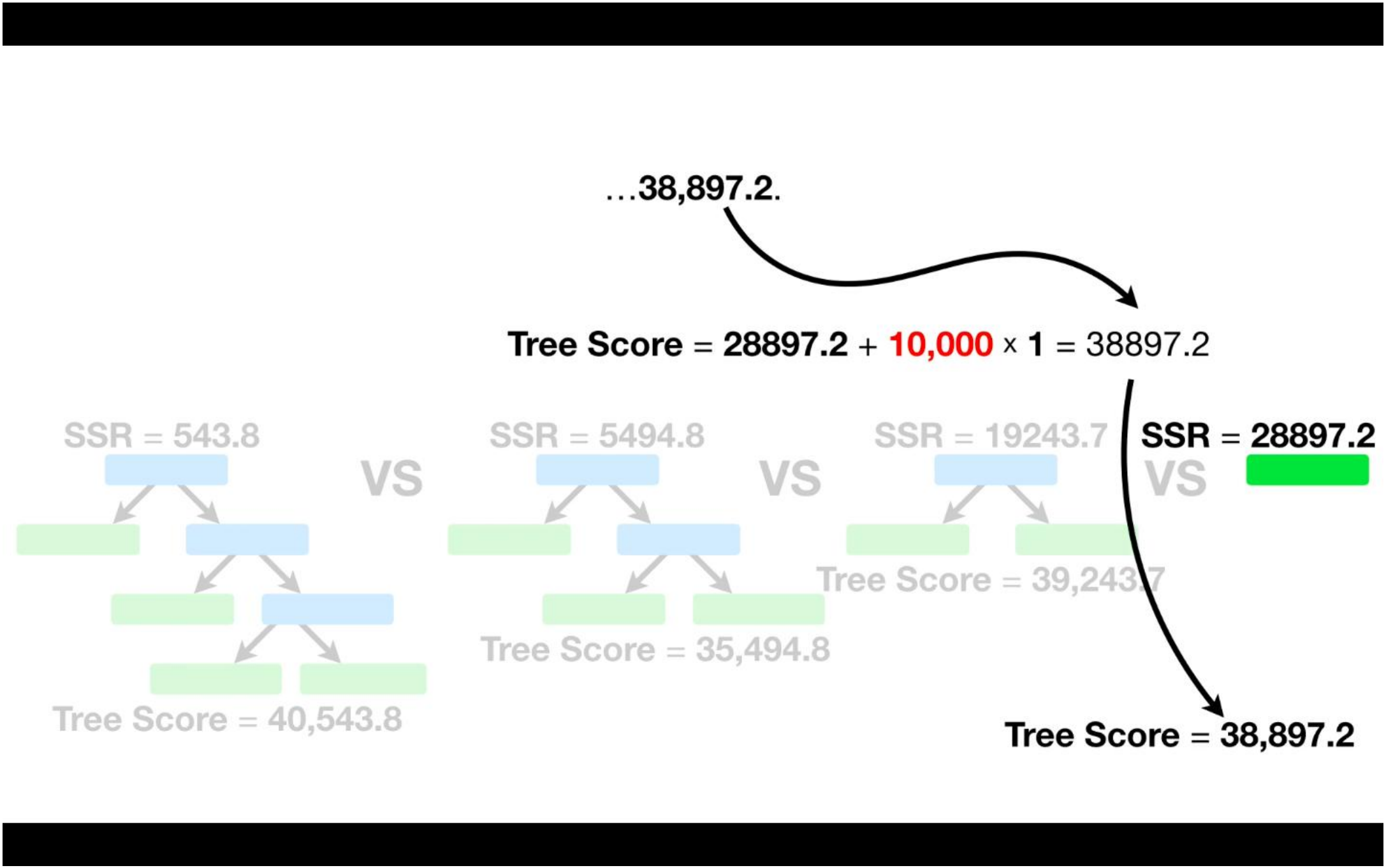
SSR = 19243.7



VS

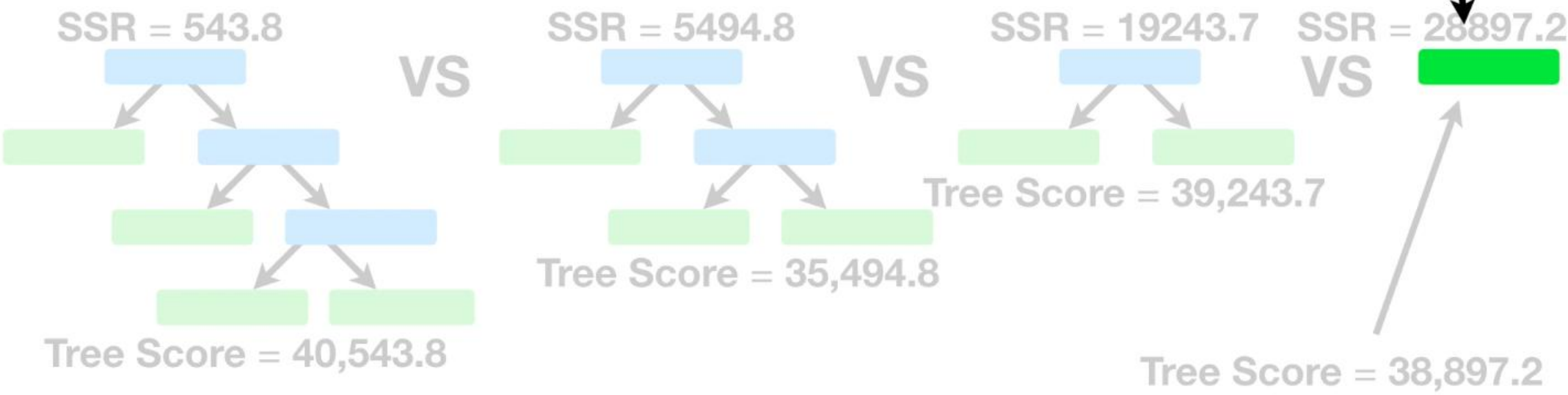
SSR = 28897.2

Tree Score = 40,543.8


…38,897.2.
Tree Score = 28897.2 + 10,000 × 1 = 38897.2
SSR = 543.8
VS
SSR = 5494.8
VS
SSR = 19243.7
SSR = 28897.2
VS
Tree Score = 39,243.7
Tree Score = 35,494.8
Tree Score = 40,543.8
Tree Score = 38,897.2

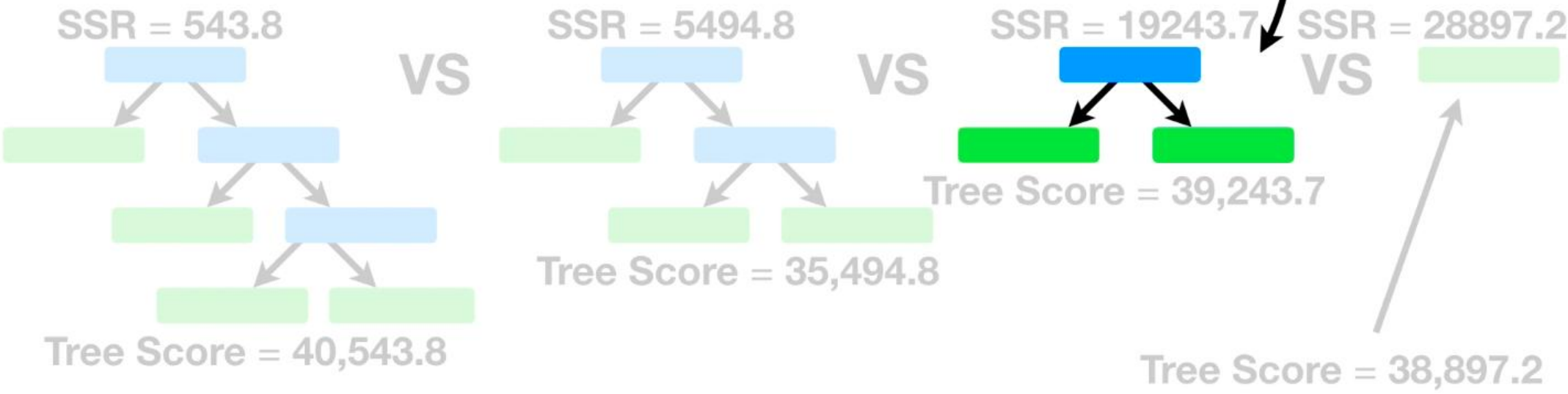
NOTE: Because $\alpha = 10{,}000$, the **Tree Complexity Penalty** for the tree with **1** leaf was **10,000**....

Tree Score = **SSR** + **10,000** x ***T***



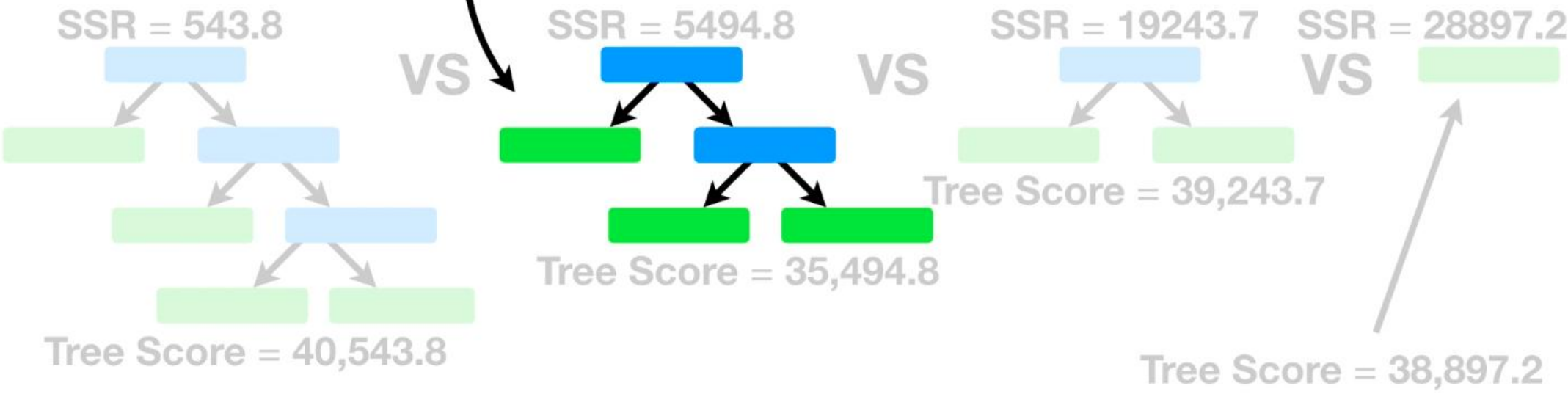
…and the **Tree Complexity Penalty** for the tree with **2** leaves was **20,000**…

Tree Score = SSR + 10,000 x *T*



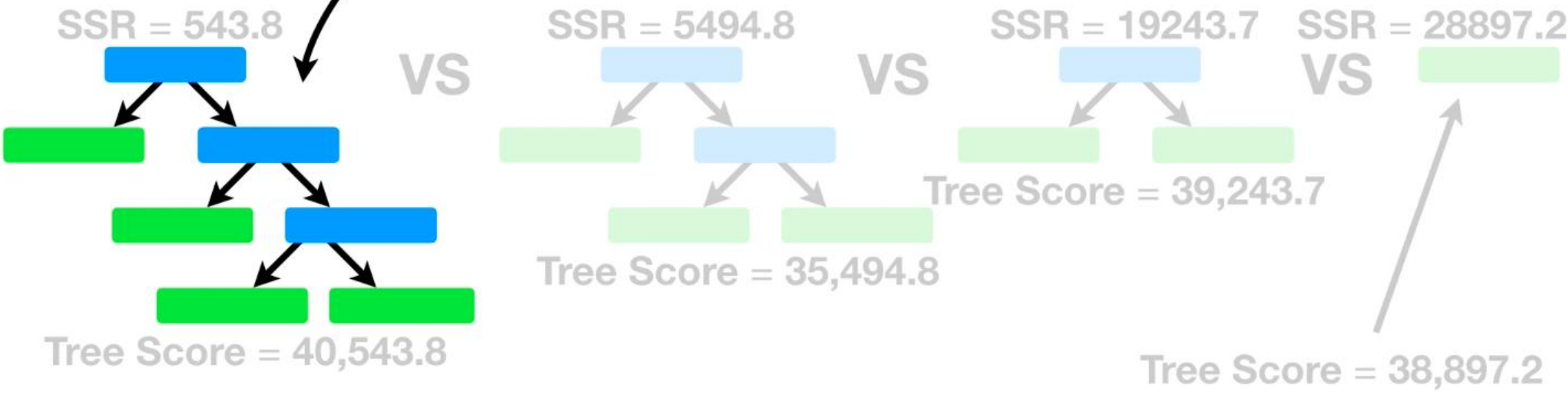
…and the **Tree Complexity Penalty** for the tree with **3** leaves was **30,000**…

Tree Score = SSR + 10,000 x *T*



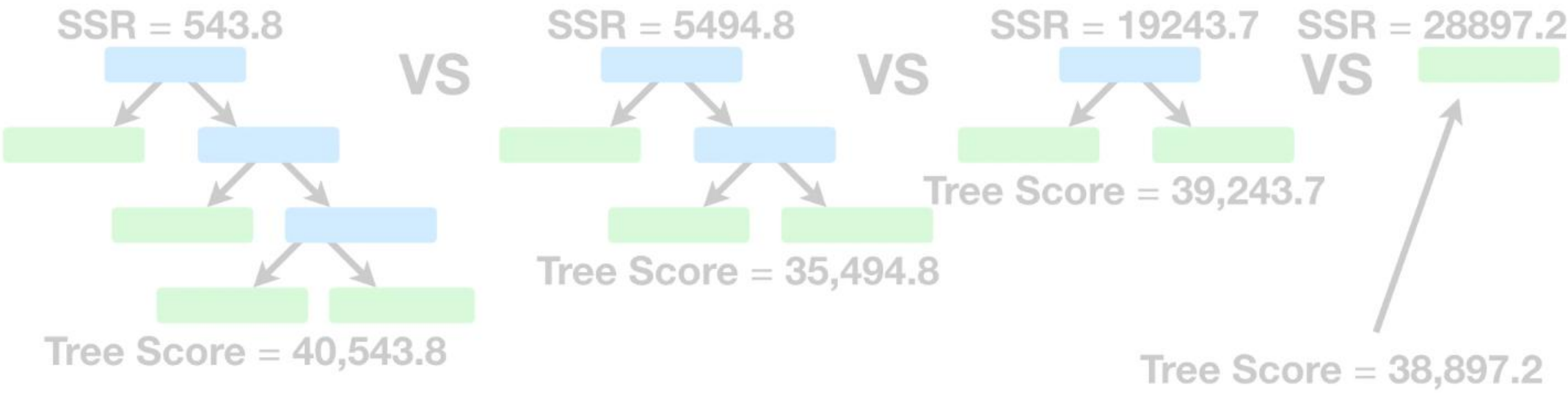
…and the **Tree Complexity Penalty** for the original, full sized tree with **4** leaves was **40,000**.

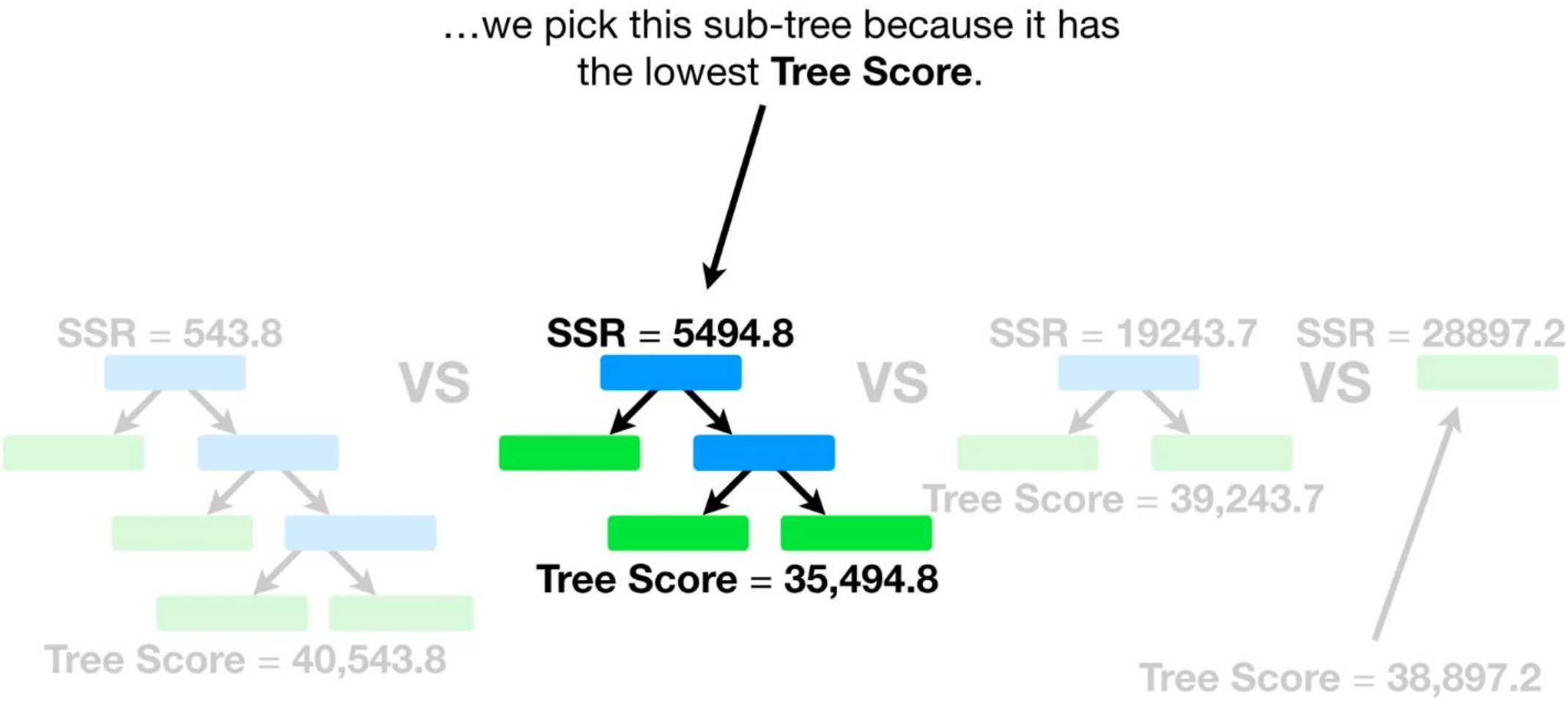
Tree Score = SSR + 10,000 x *T*



Thus, the the more leaves,
the larger the penalty.

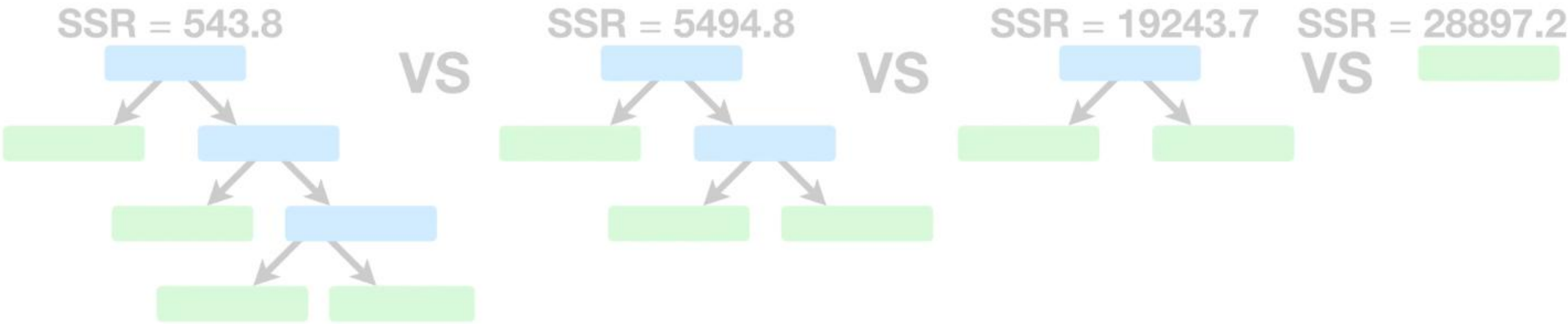
Tree Score = SSR + **10,000 x *T***




…we pick this sub-tree because it has the lowest **Tree Score**.
SSR = 543.8
VS
SSR = 5494.8
VS
SSR = 19243.7
VS
SSR = 28897.2
Tree Score = 39,243.7
Tree Score = 35,494.8
Tree Score = 40,543.8
Tree Score = 38,897.2

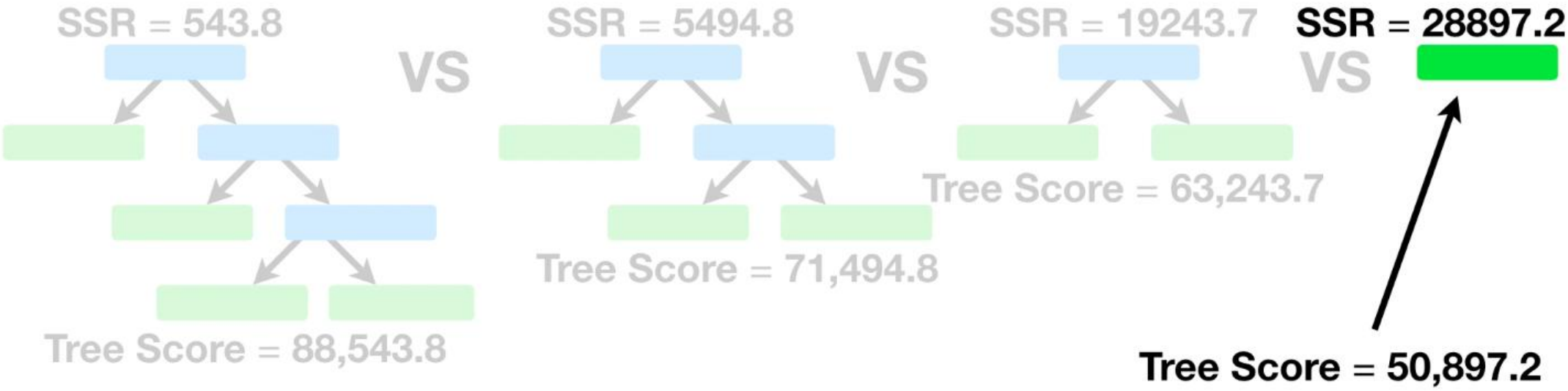
NOTE: If we set $\alpha = 22{,}000$…

Tree Score = SSR + 22,000 × *T*



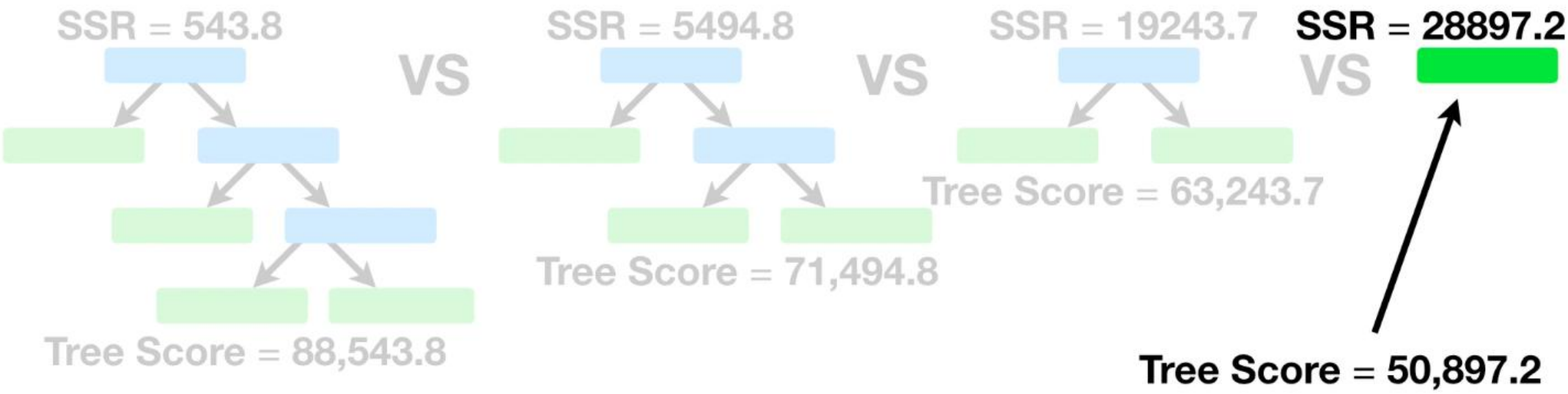
…then we would use the sub-tree with only one leaf because it has lowest **Tree Score**.

Tree Score = SSR + 22,000 x *T*



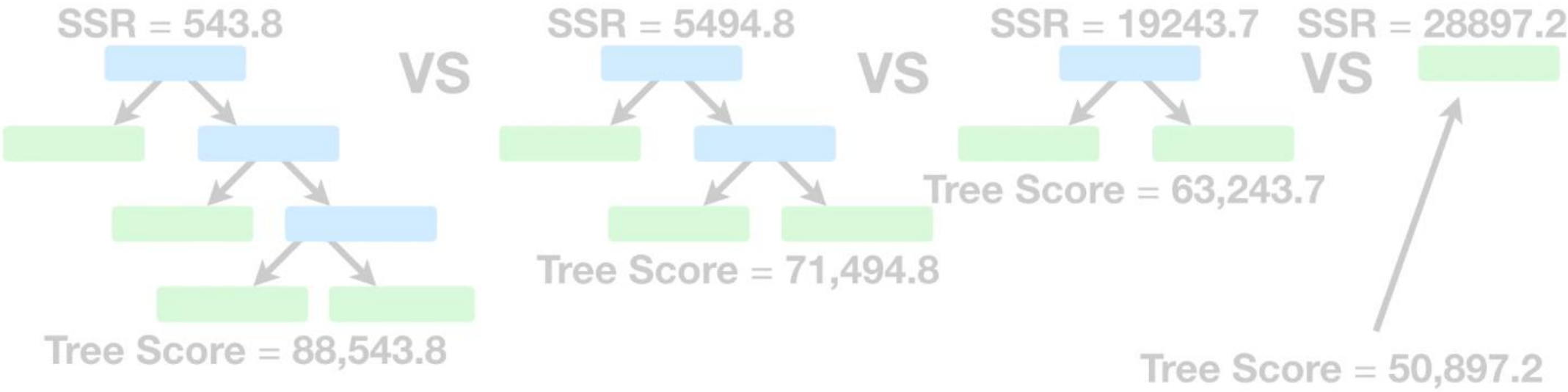
Thus, the value for α makes a difference in our choice of sub-tree.

Tree Score = SSR + 22,000 x *T*



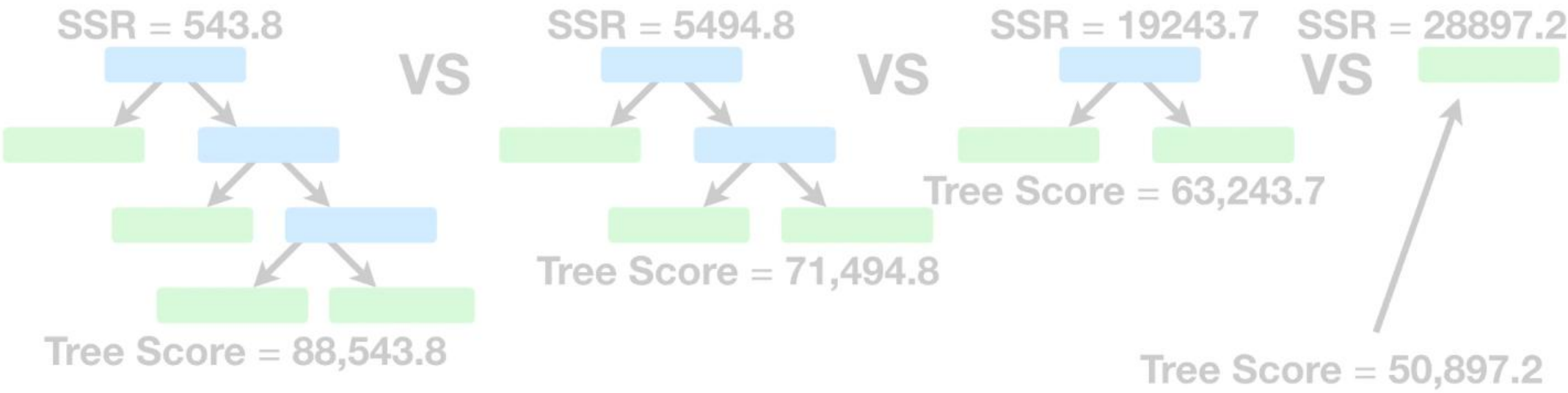
So let's talk about how build a pruned regression tree…

Tree Score = SSR + αT

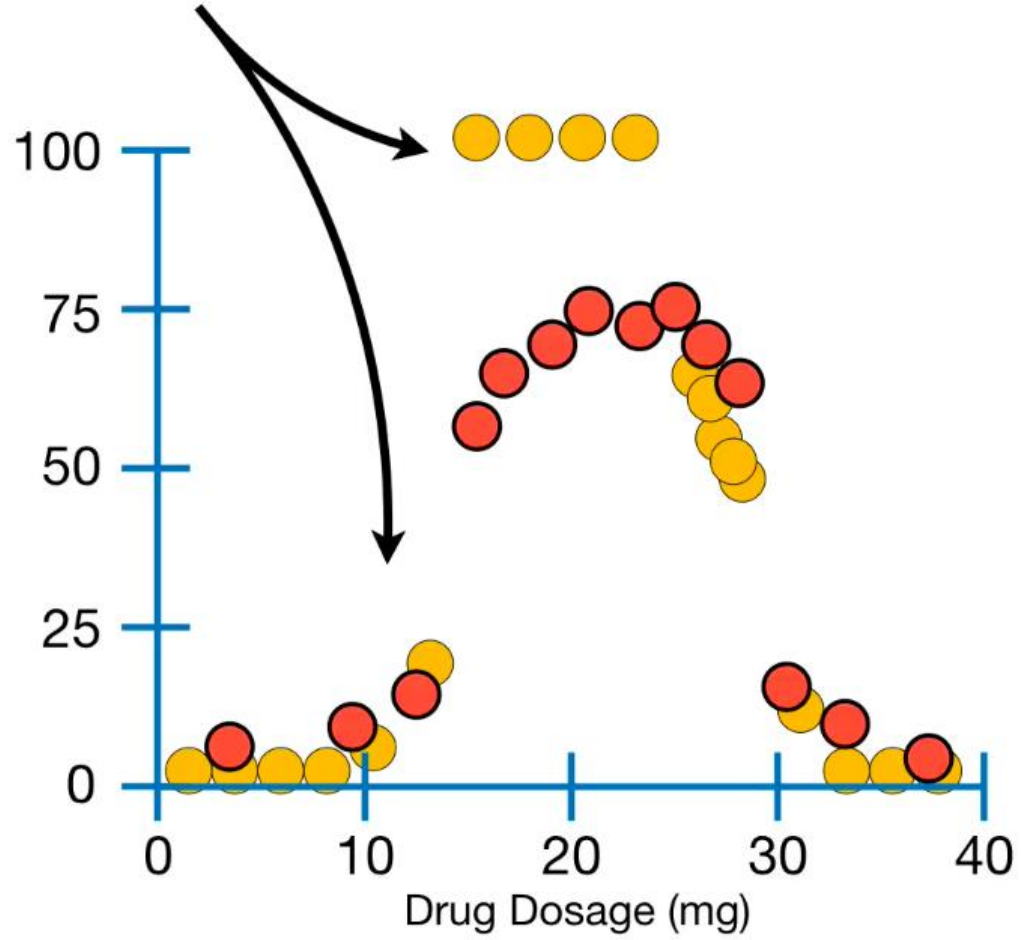


…and how to find the best value for α.

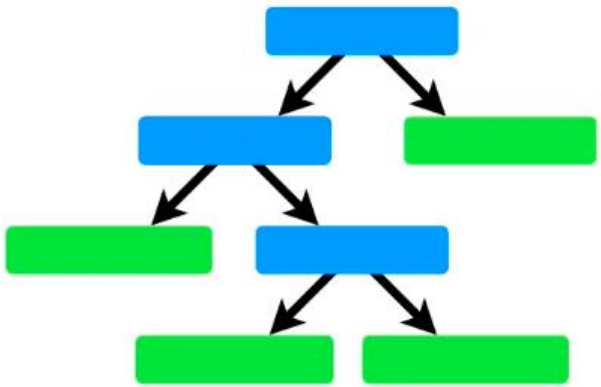
Tree Score = SSR + αT

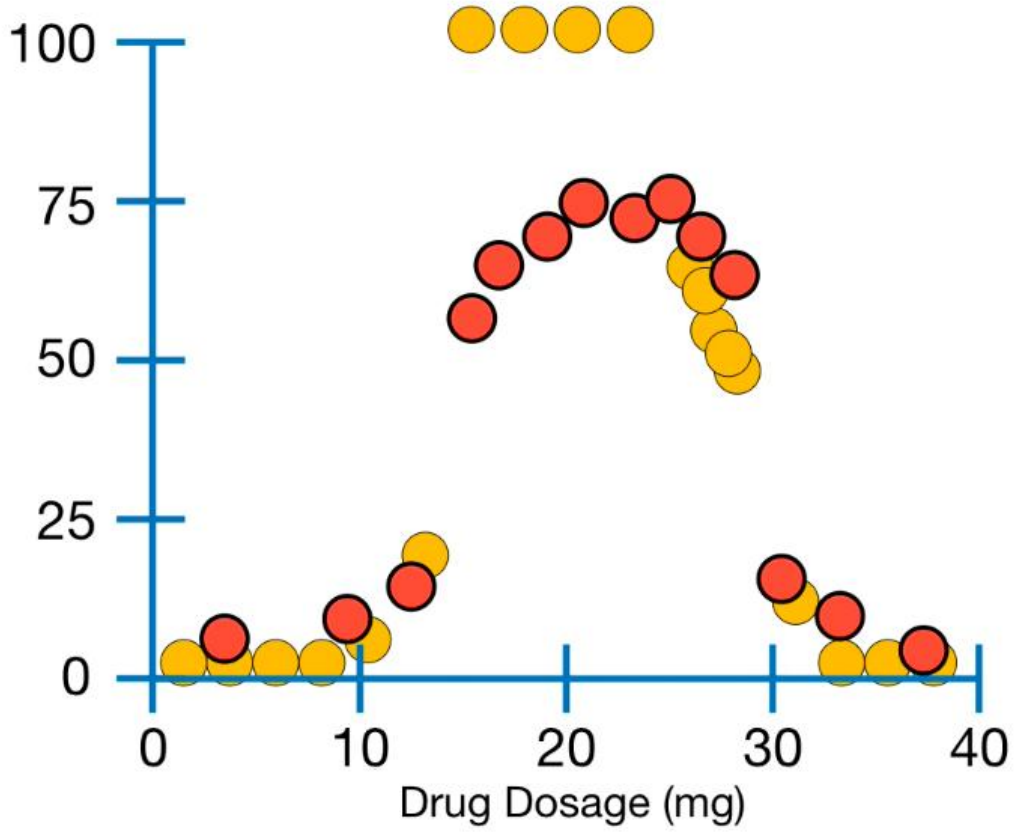


First, using ***all*** of the data…

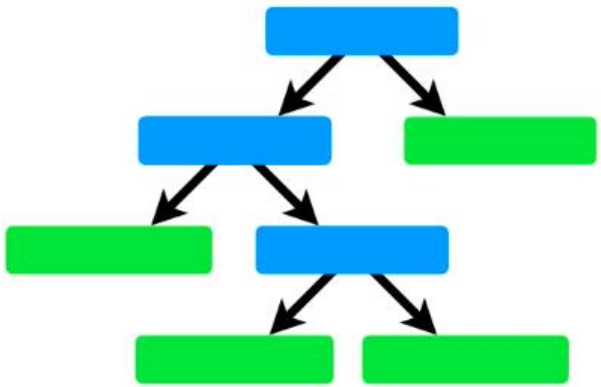


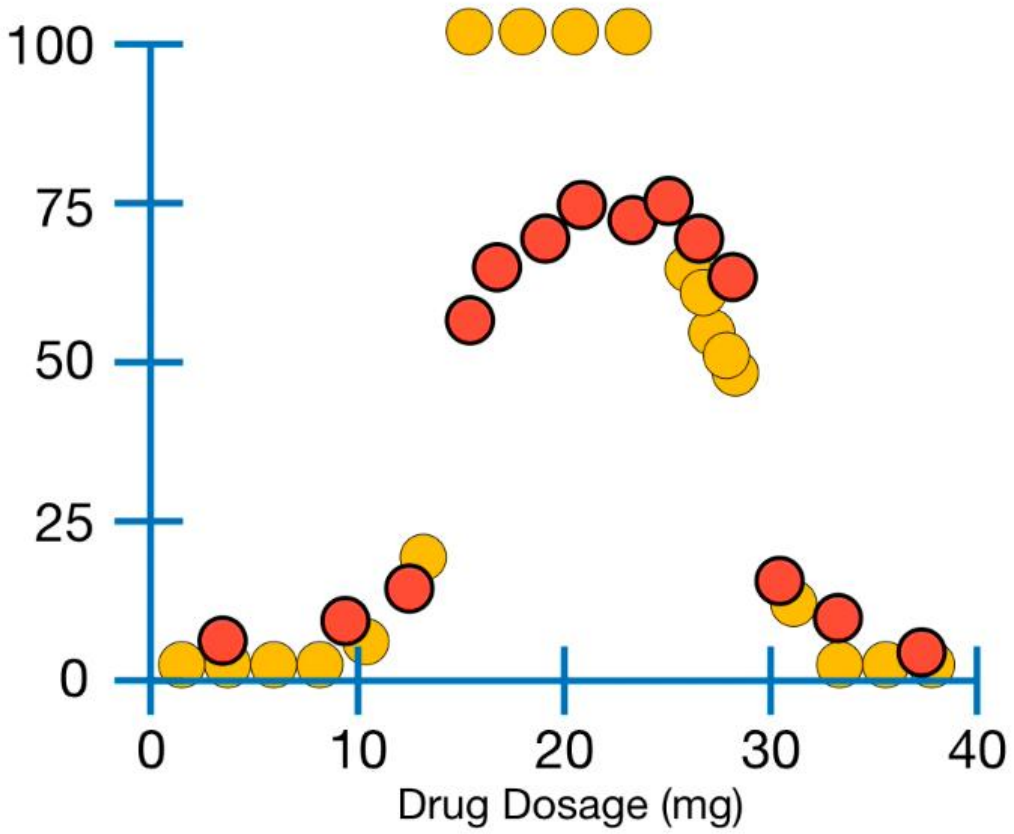
…build a full sized **Regression Tree**.





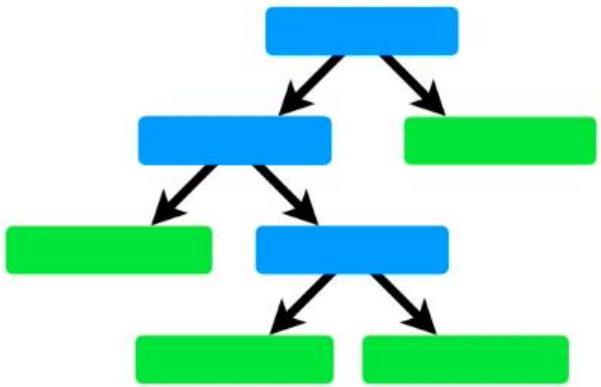
NOTE: This full sized tree is different than before because it was fit to ***all*** of the data, not just the **Training Data**.





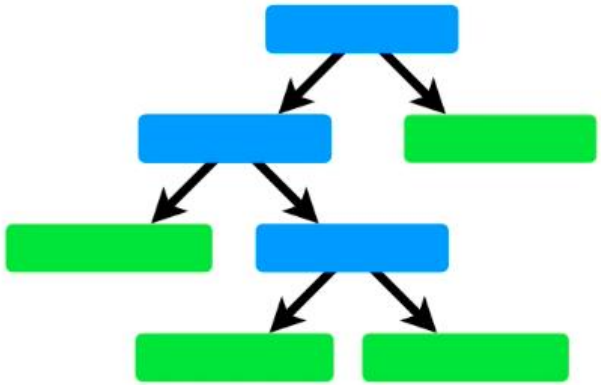
ALSO NOTE: This full sized tree has the lowest **Tree Score** when $\alpha = 0$.

Tree Score = **SSR** + αT



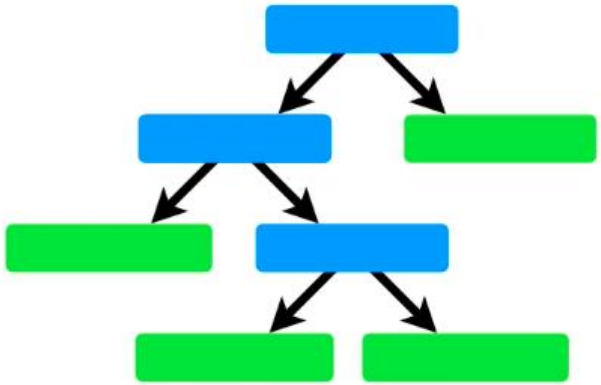
…and the **Tree Score** is just the **Sum of the Squared Residuals**…

Tree Score = SSR



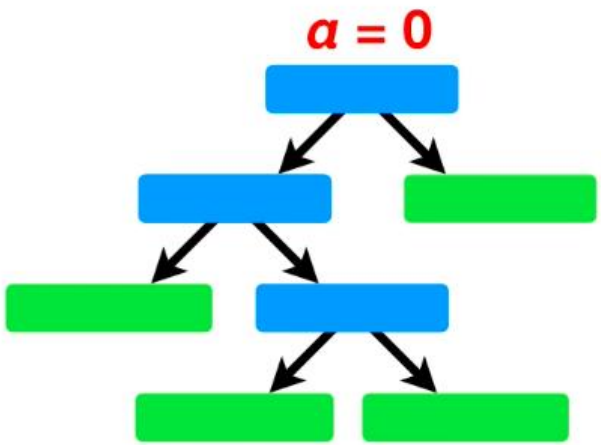
…and, as we saw earlier, all of the sub-trees will have larger **Sum of Squared Residuals**.

Tree Score = SSR



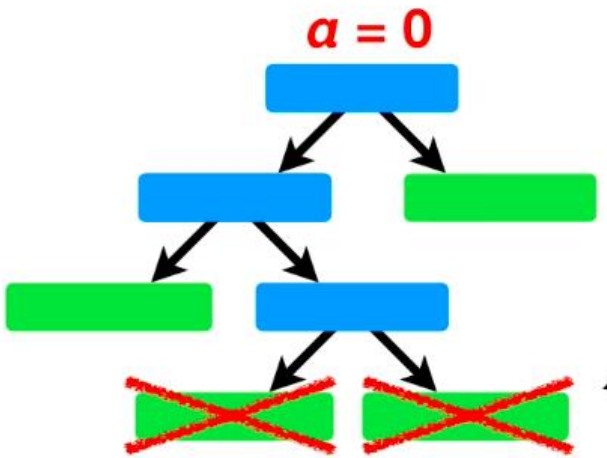
So let's put $\alpha = 0$ here, to remind us that this tree has the lowest **Tree Score** when $\alpha = 0$.

Tree Score = SSR



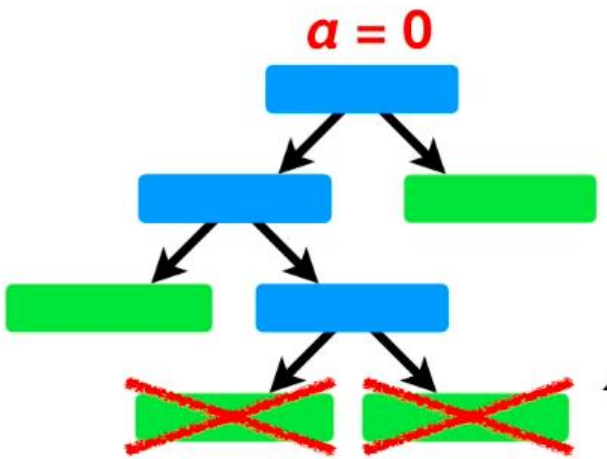
Now we increase α until pruning leaves will give us a lower **Tree Score**.

Tree Score = **SSR** + αT



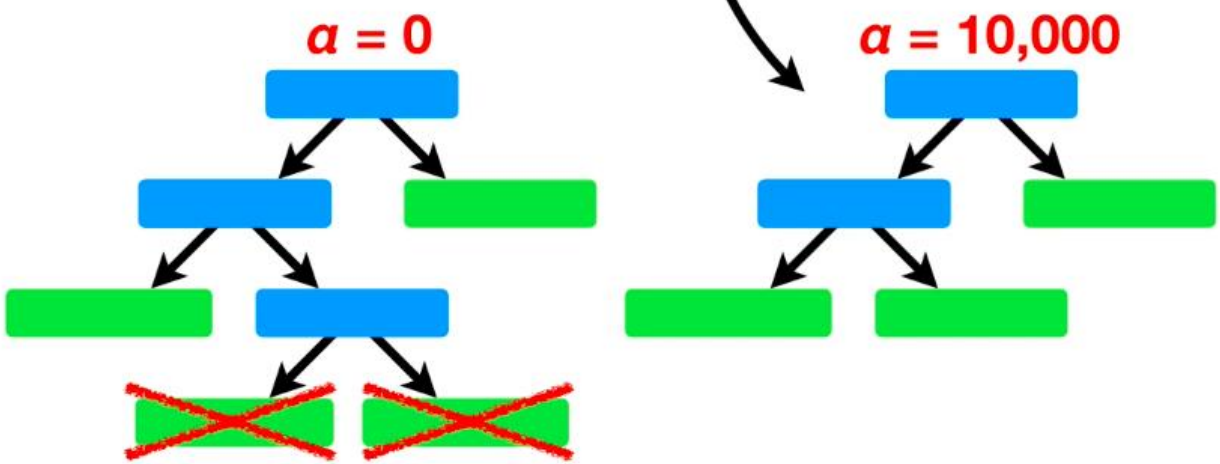
In this case, when $\alpha = 10{,}000$, we'll get a lower **Tree Score** if we remove these leaves…

Tree Score = **SSR** + αT



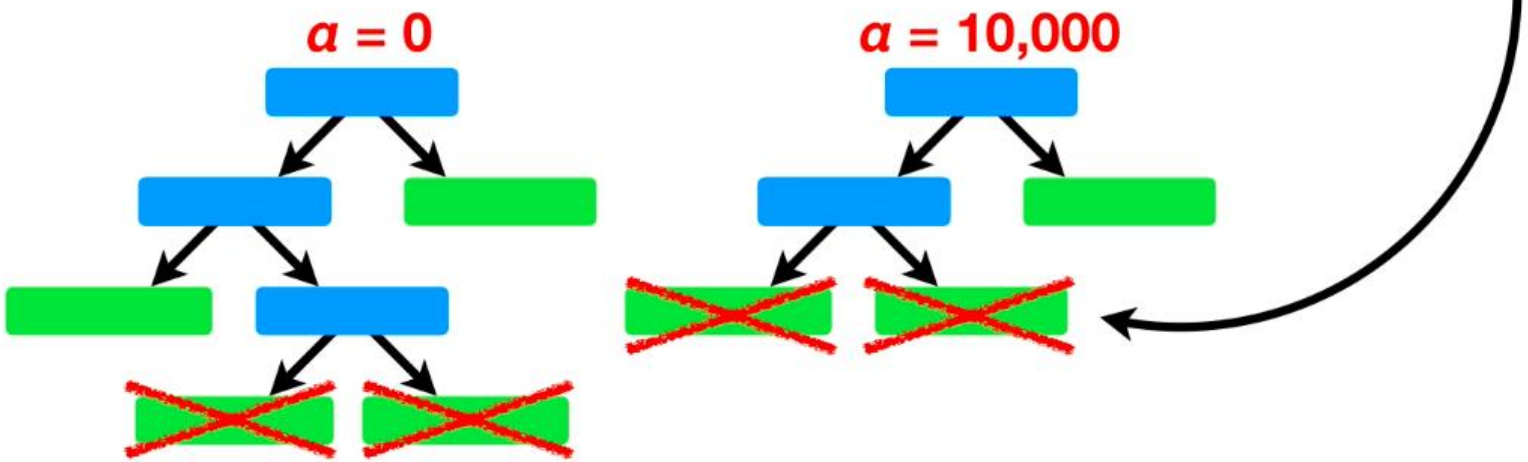
…and use this sub-tree.

Tree Score = SSR + αT



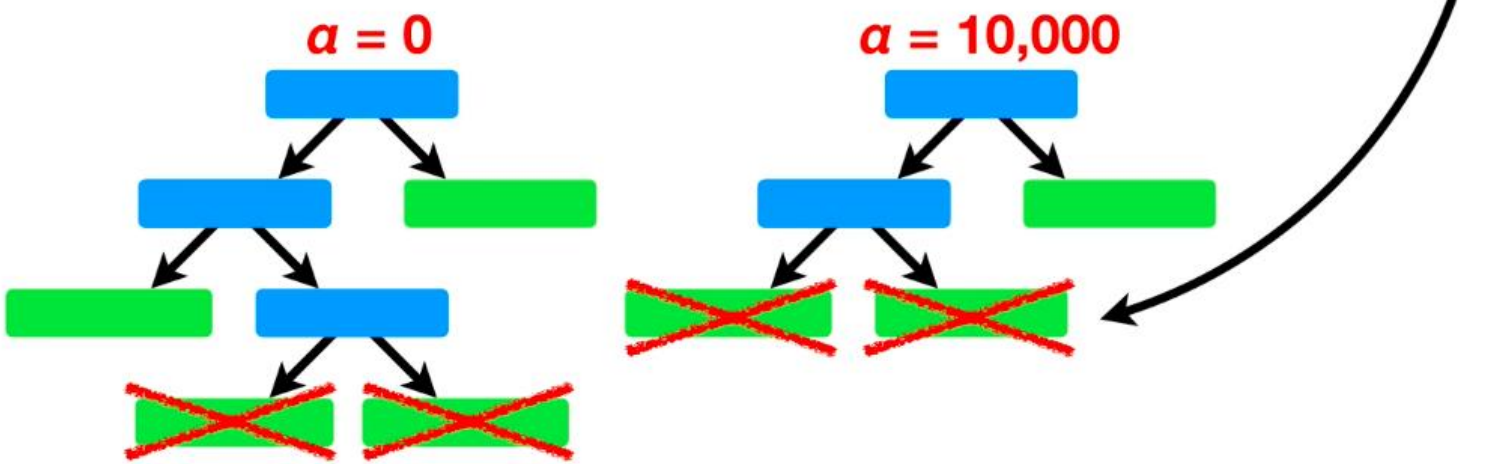
Now we increase α again until pruning leaves will give us a lower **Tree Score**.

Tree Score = **SSR** + αT



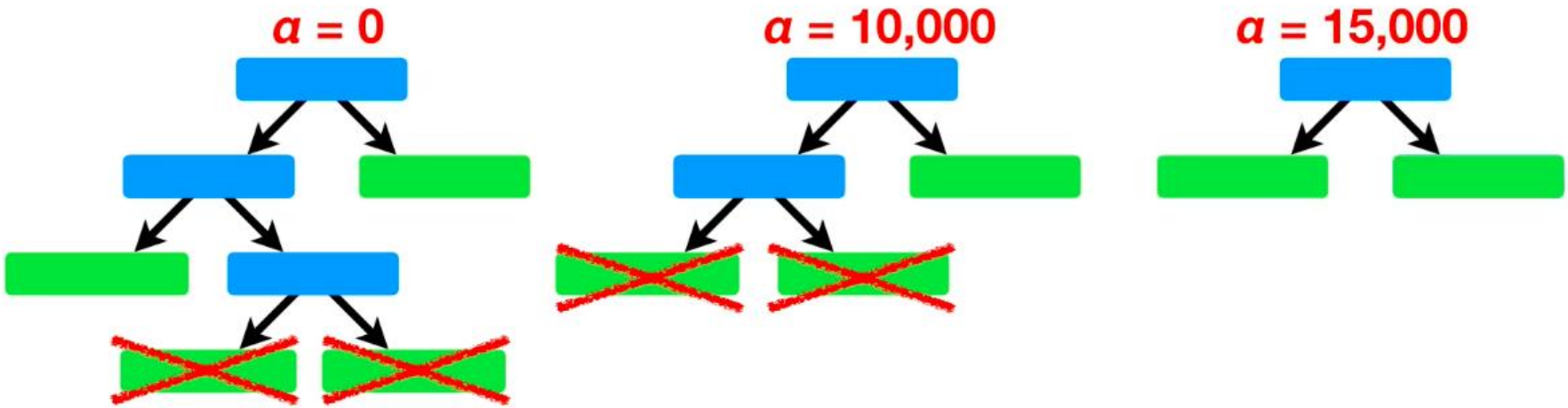
In this case when $\alpha = 15{,}000$,
we will get a lower **Tree Score**
if we remove these leaves…

Tree Score = **SSR** + αT



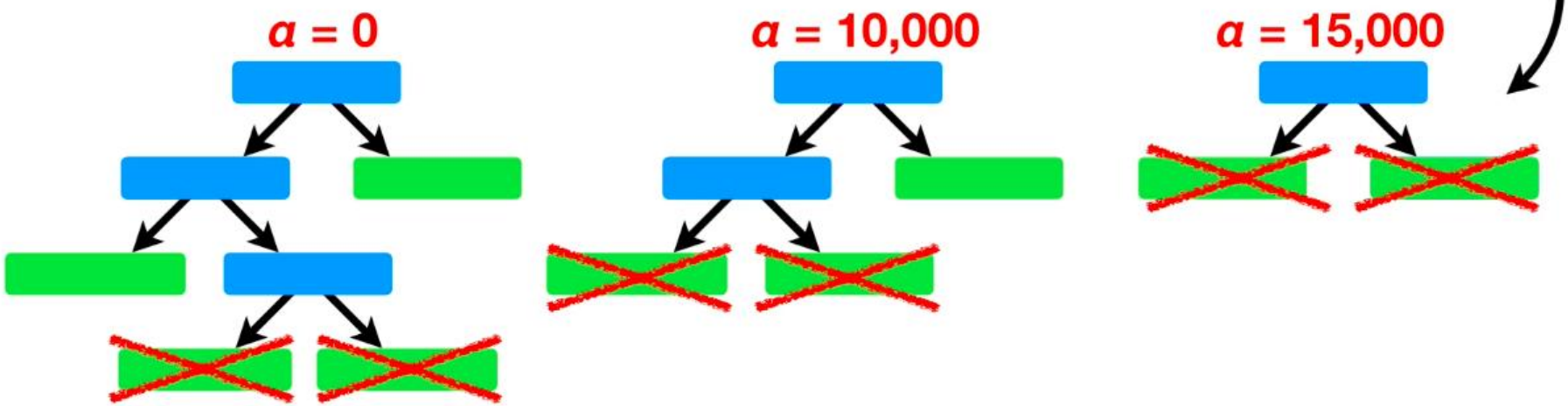
…and use this sub-tree instead.

Tree Score = SSR + αT



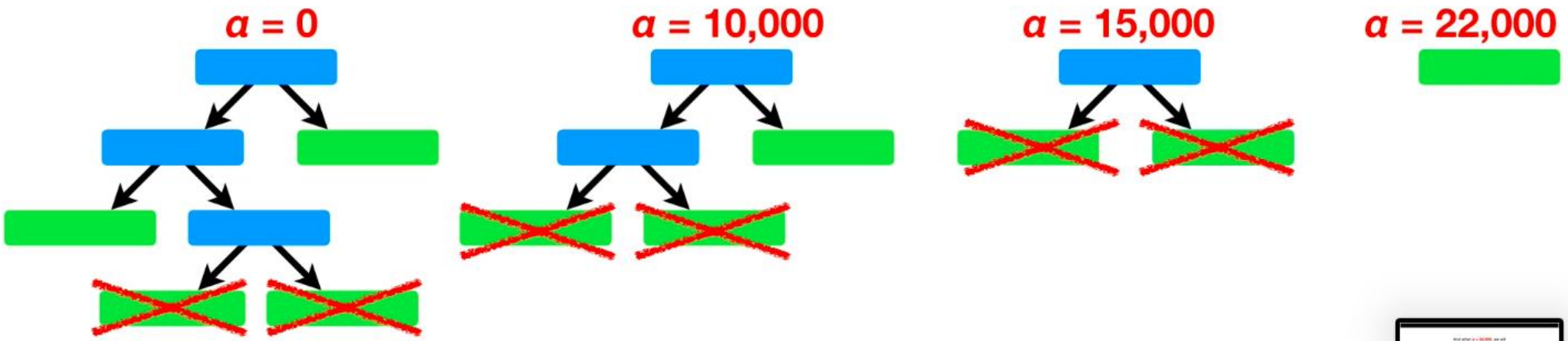
And when α = 22,000, we will get a lower **Tree Score** if we remove these leaves…

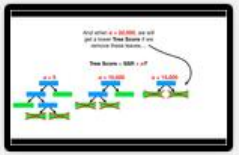
Tree Score = **SSR** + αT



…and use this sub-tree instead.

Tree Score = SSR + αT

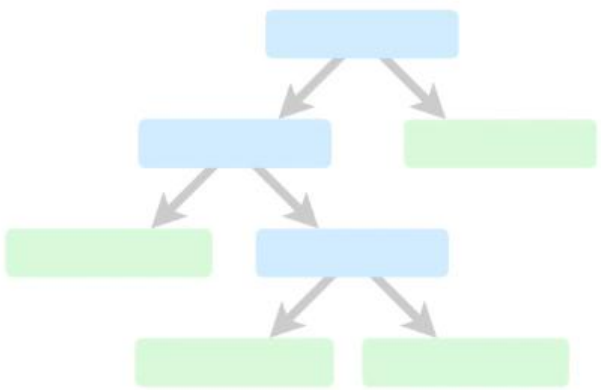




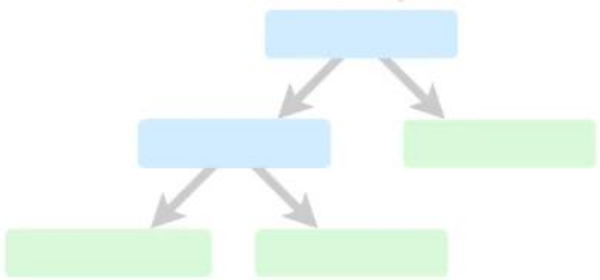
In the end, different values for
α give us a sequence of trees,
from full sized to just a leaf.

Tree Score = SSR + αT

$\alpha = 0$



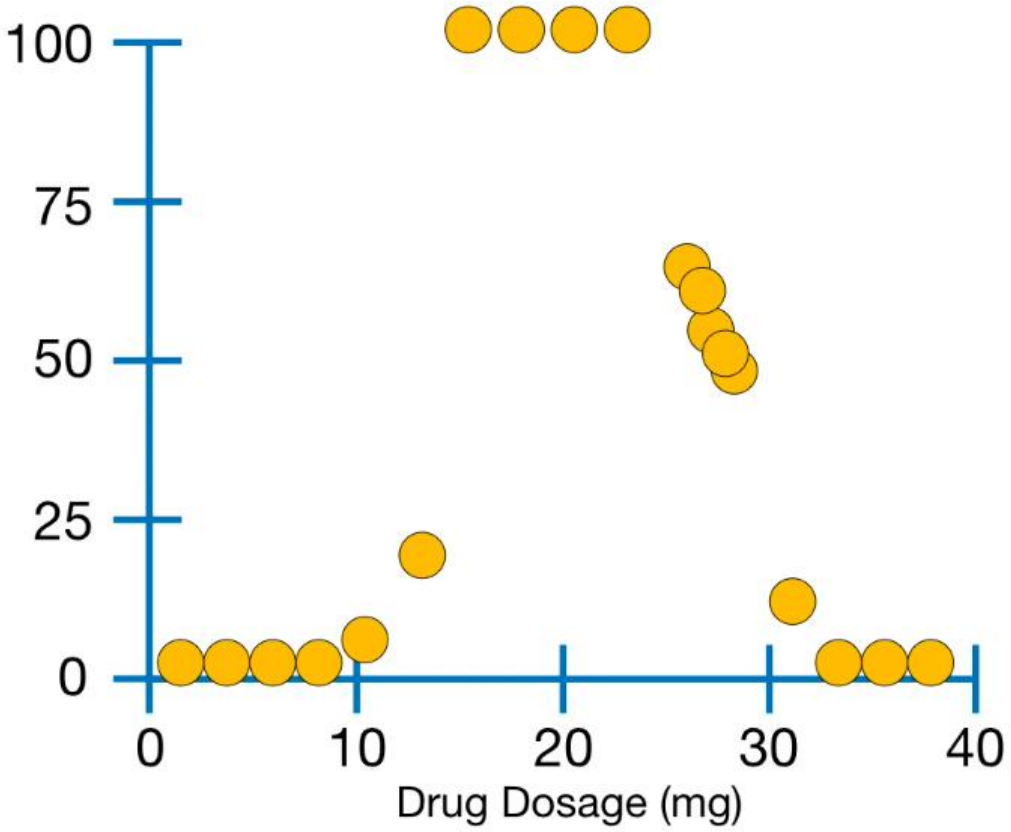
$\alpha = 10{,}000$



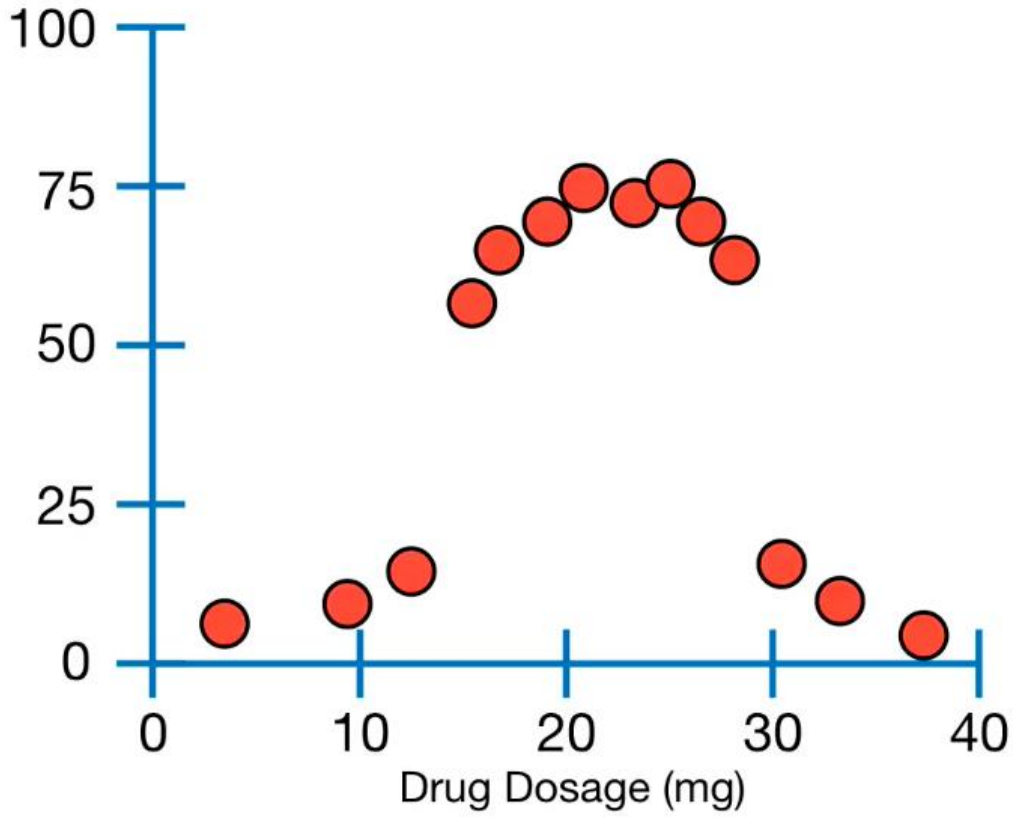
$\alpha = 15{,}000$

$\alpha = 22{,}000$

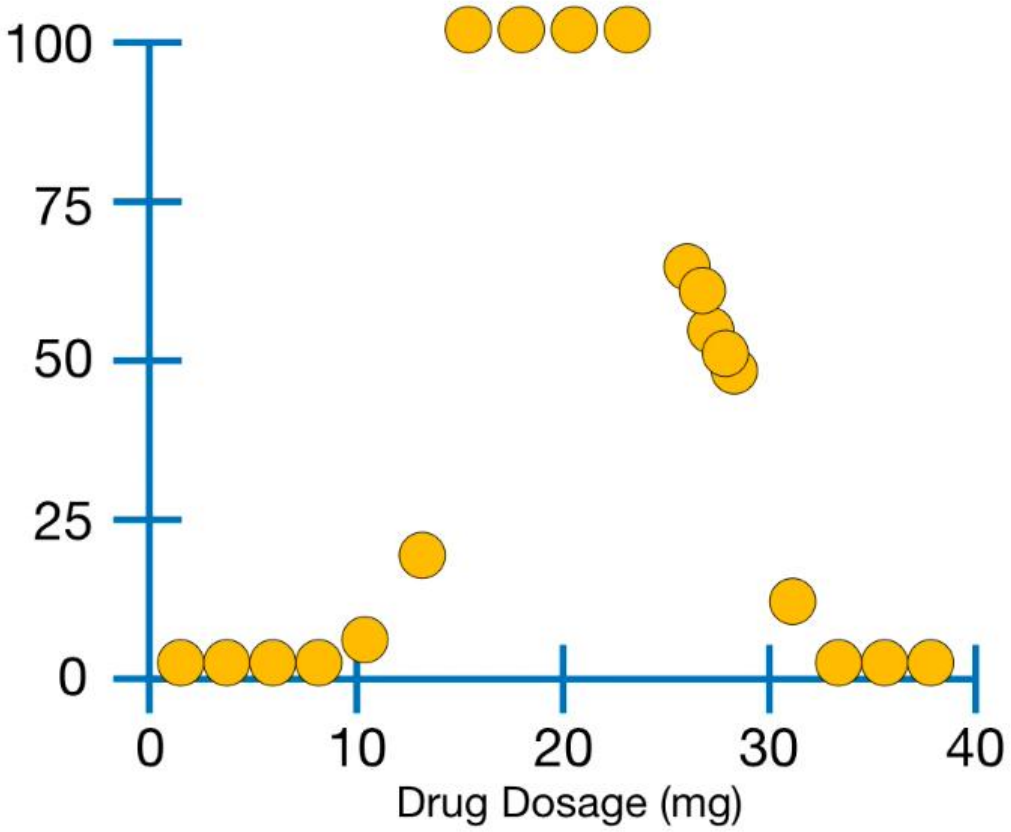
…and divide it into **Training**…



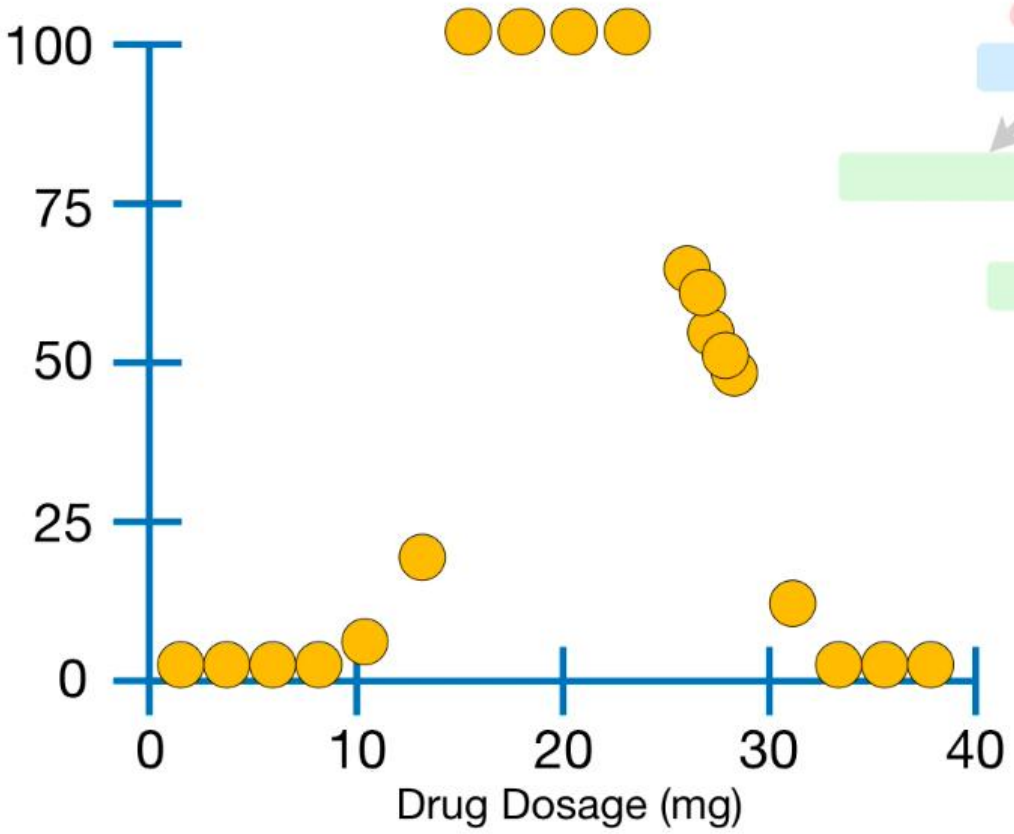
…and **Testing Datasets**.



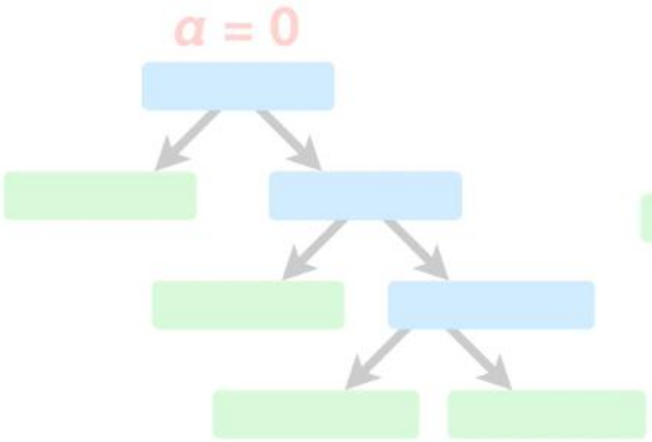
And just using the
Training Data…

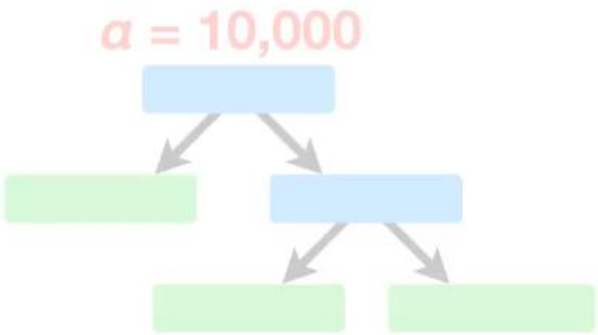


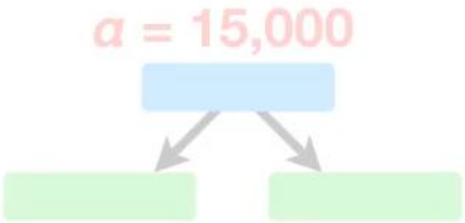
And just using the **Training Data**…



…use the ***α*** values we found before to build a full tree and a sequence of sub-trees that minimize the **Tree Score**.



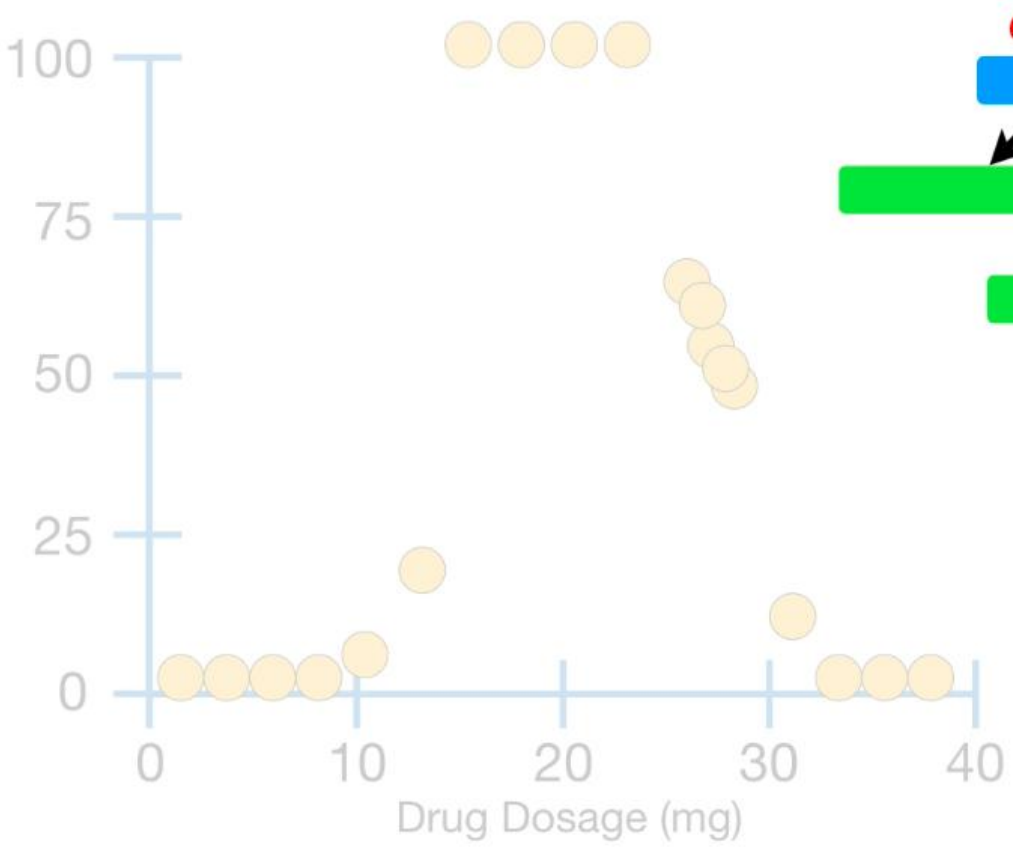


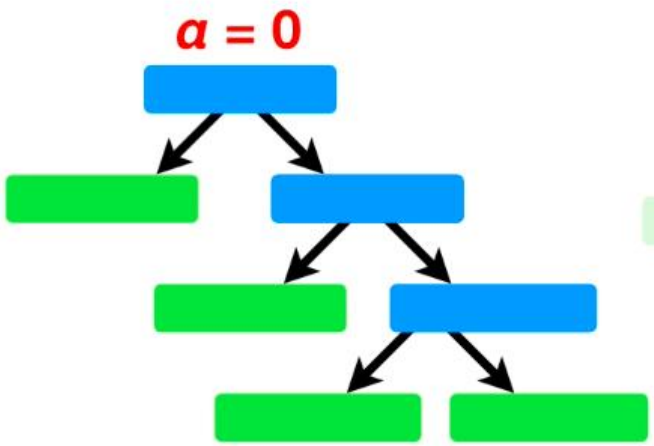


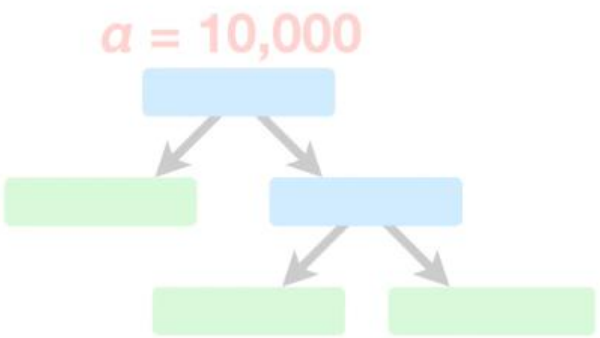
***α* = 22,000**

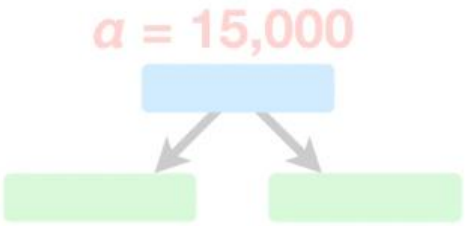
Tree Score = **SSR** + ***αT***

In other words, when ***α* = 0**, we build a full sized tree, since it will have the lowest **Tree Score**.



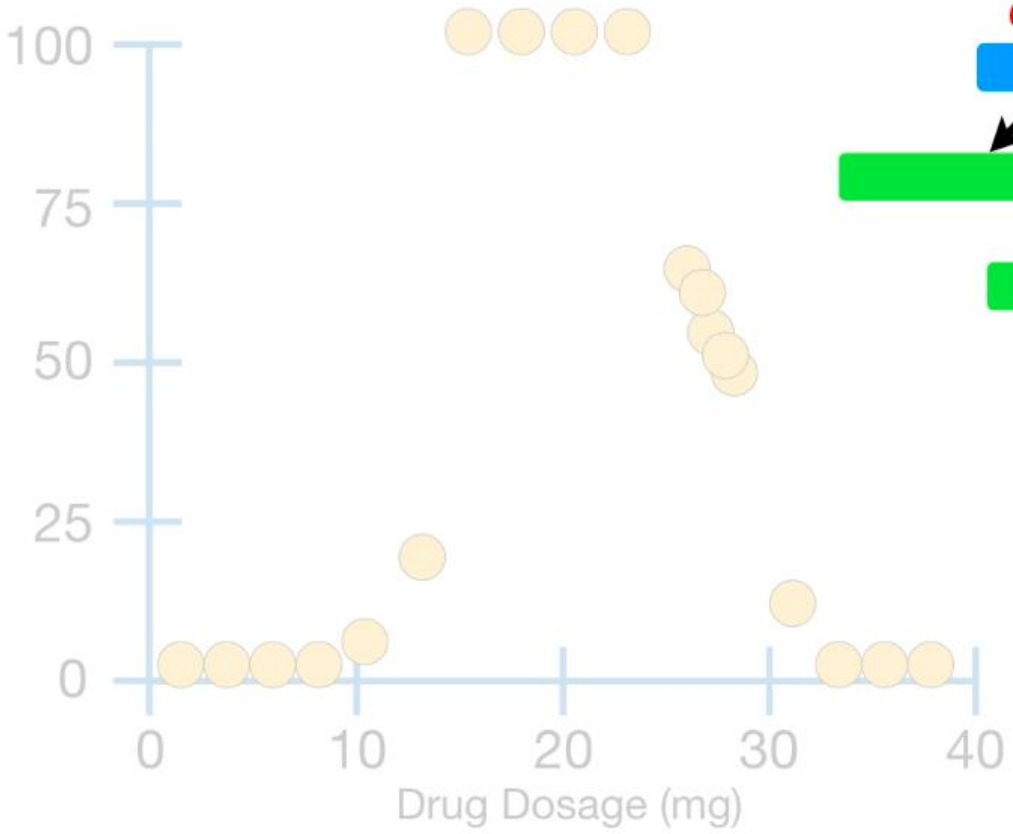


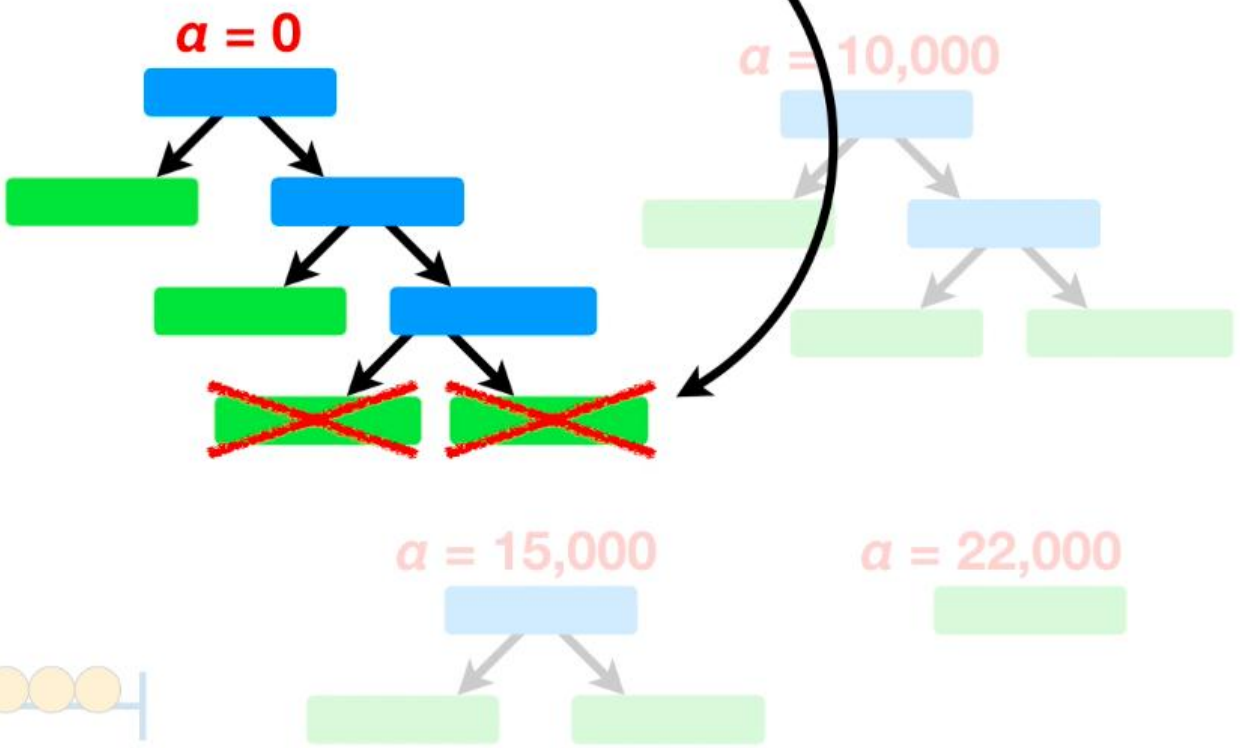




Tree Score = SSR + αT

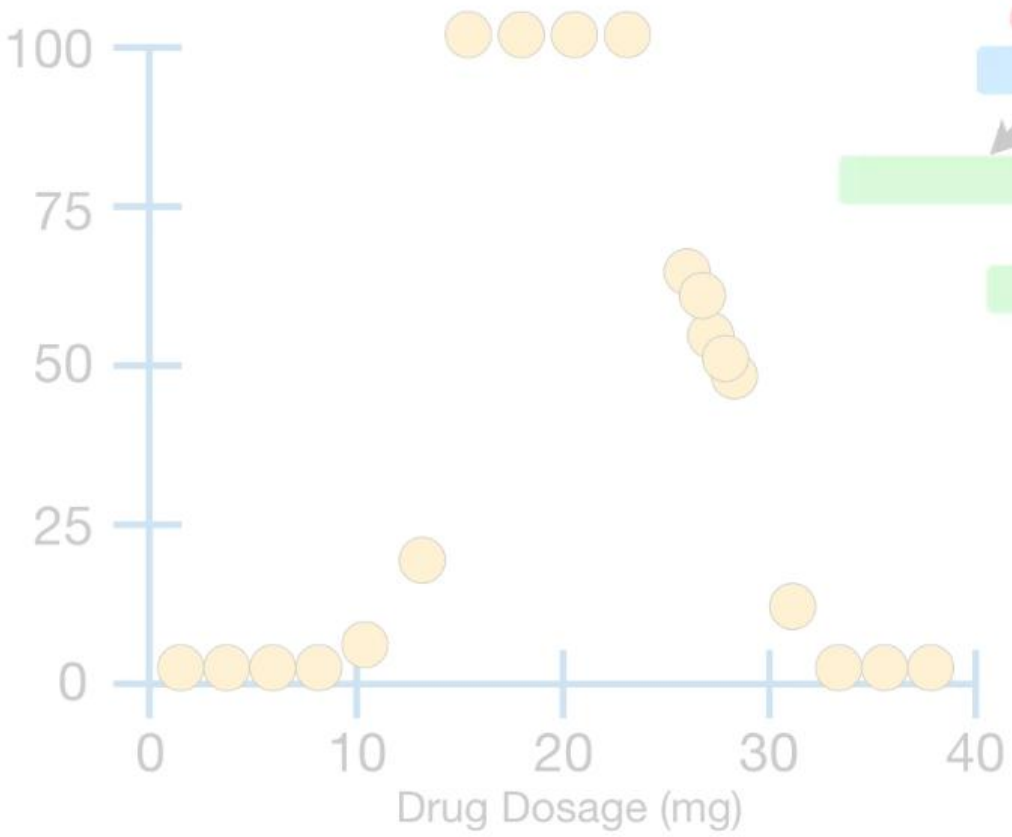
However, when α = **10,000**,
we will get a lower **Tree Score**
if we prune these leaves…

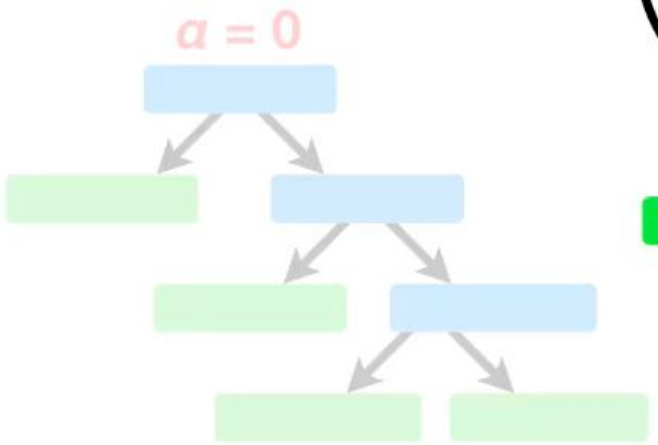


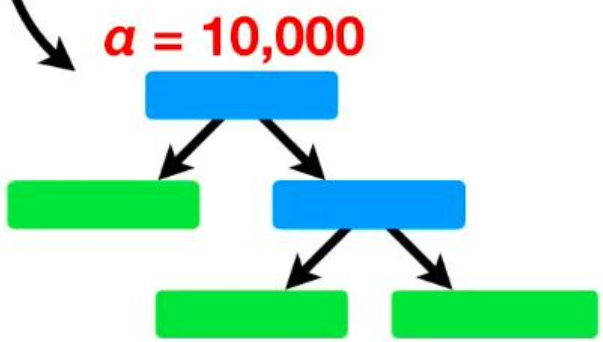


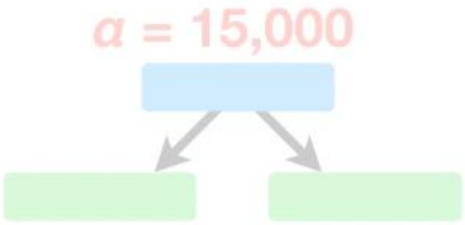
Tree Score = SSR + αT

…and use this tree instead.



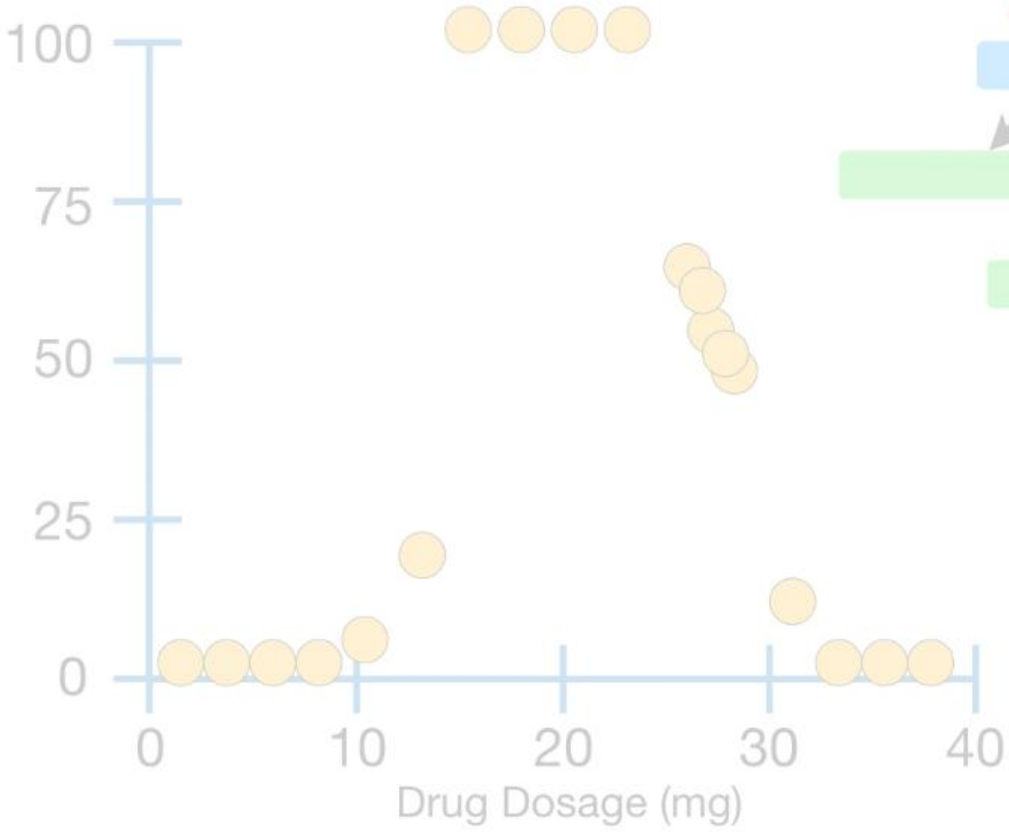


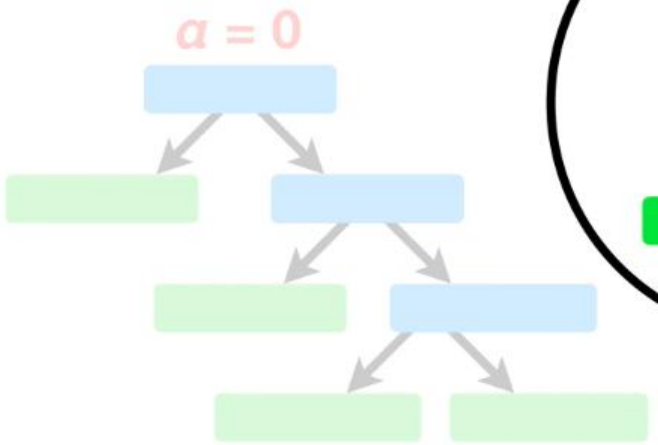


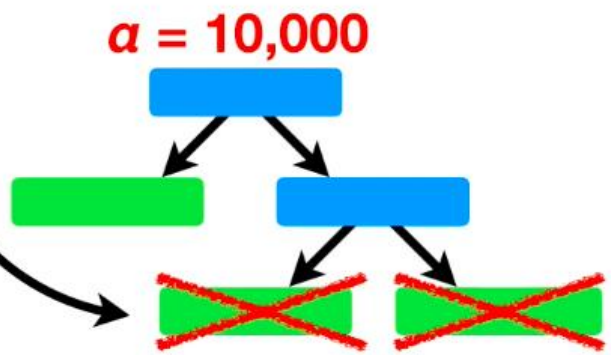


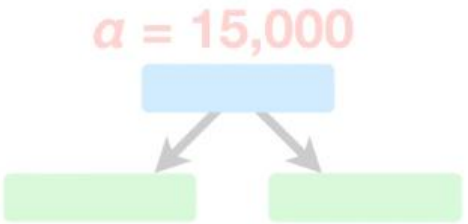
Tree Score = **SSR** + ***αT***

And when ***α* = 15,000**, we will get a lower **Tree Score** if we prune these leaves…



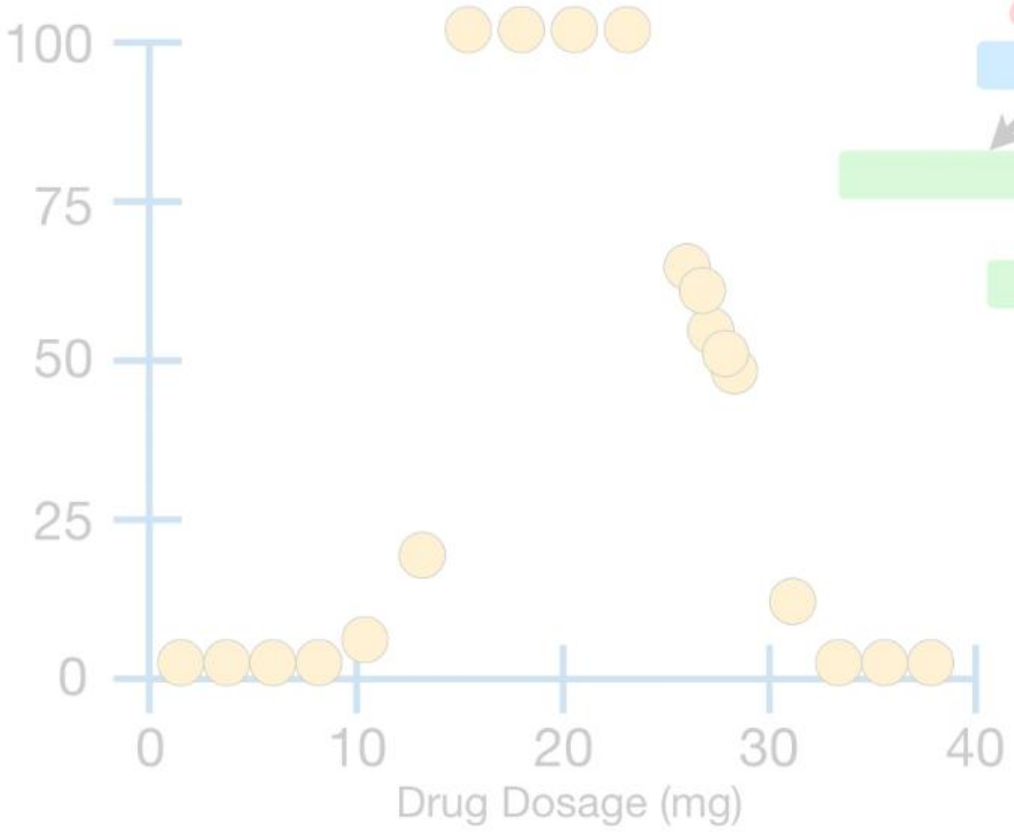


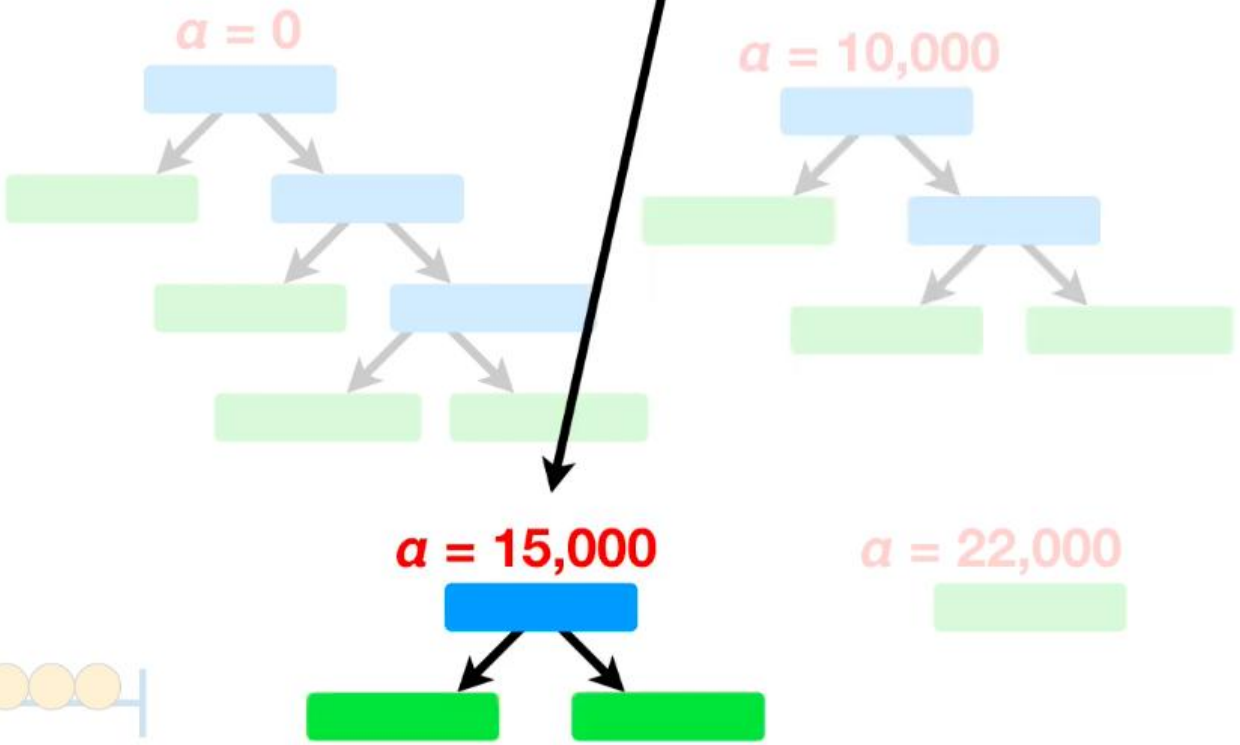




Tree Score = SSR + αT

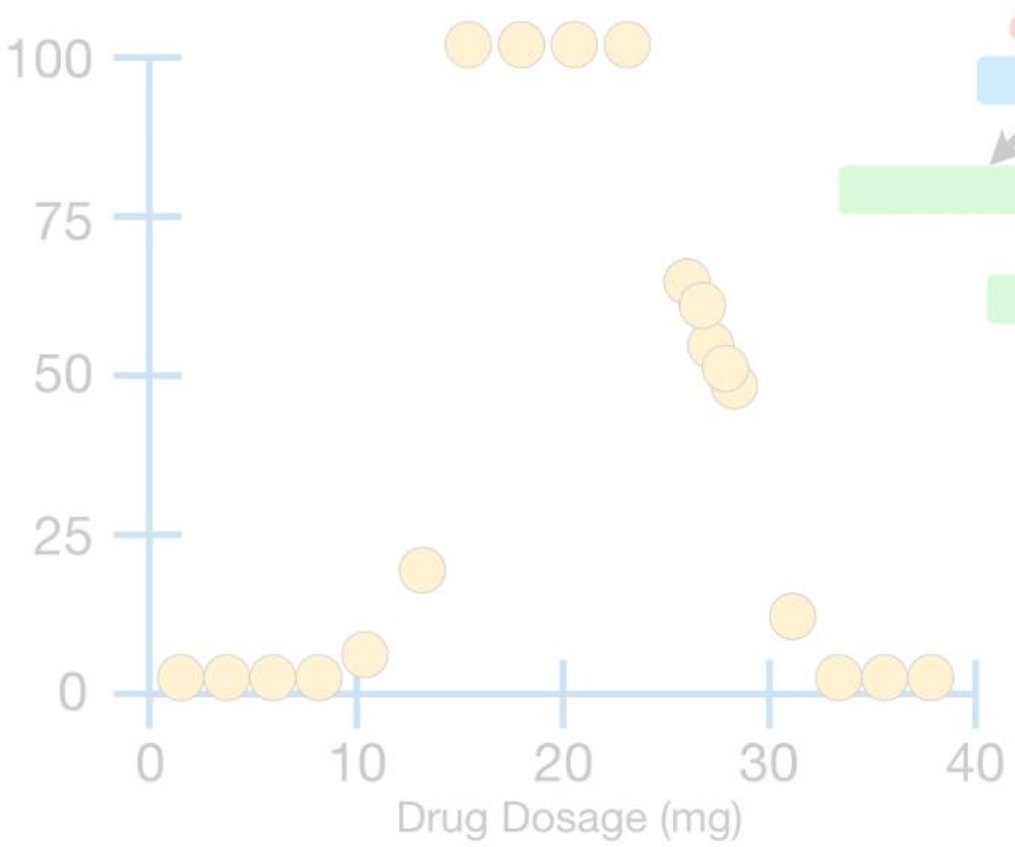
…and use this tree instead.

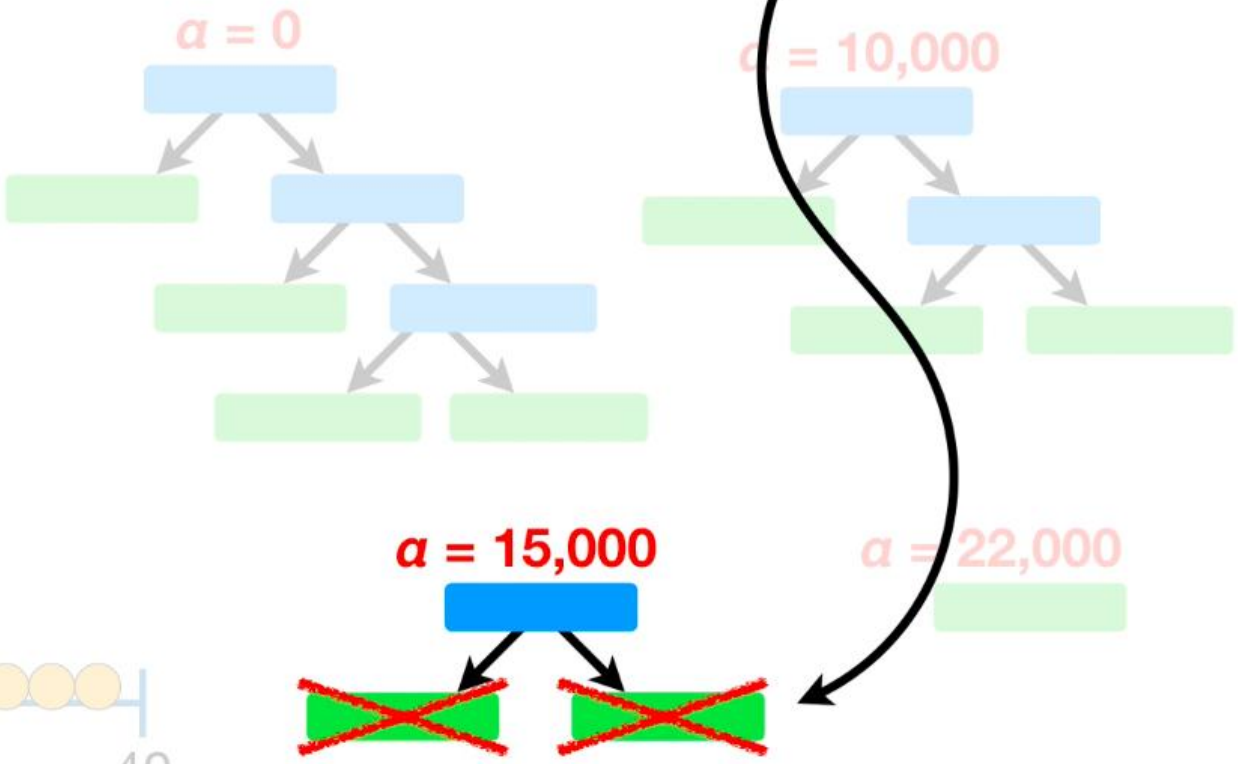




Tree Score = **SSR** + ***αT***

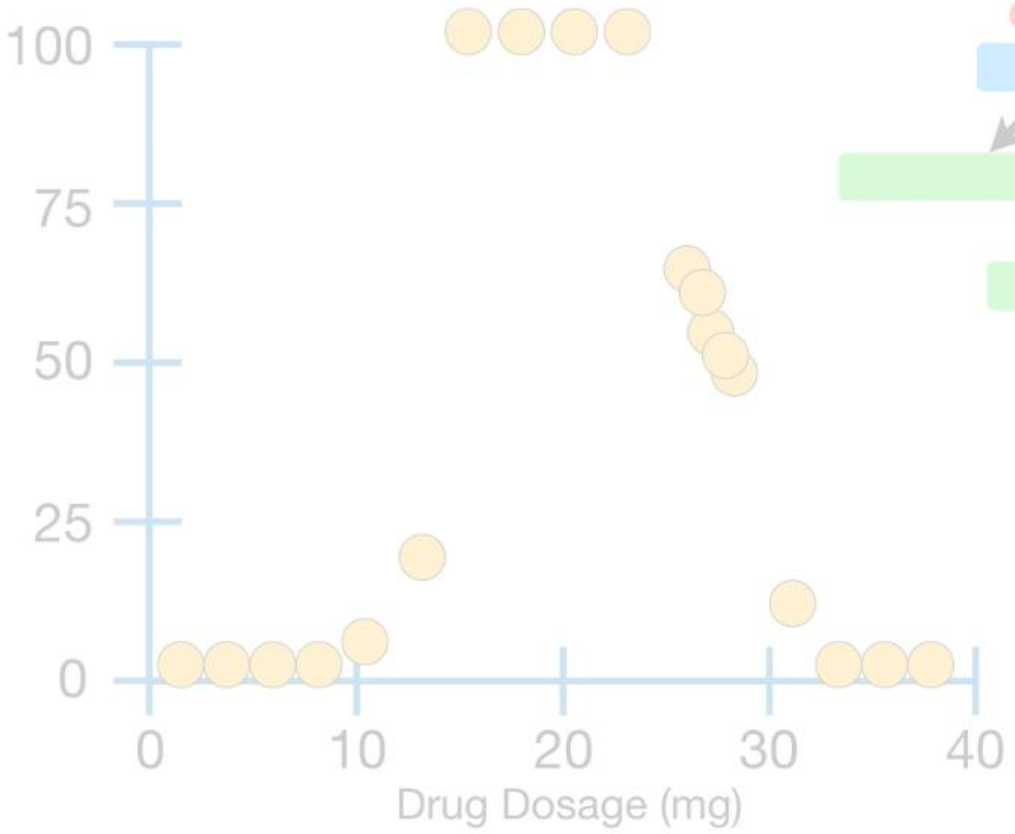
Lastly, when ***α* = 22,000**, we will get a lower **Tree Score** if we prune these two leaves…

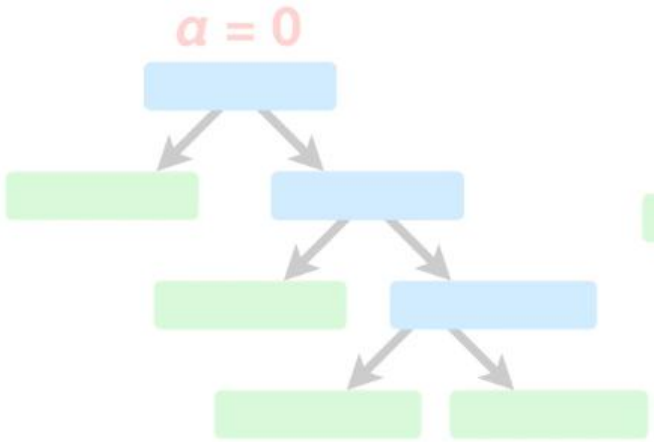


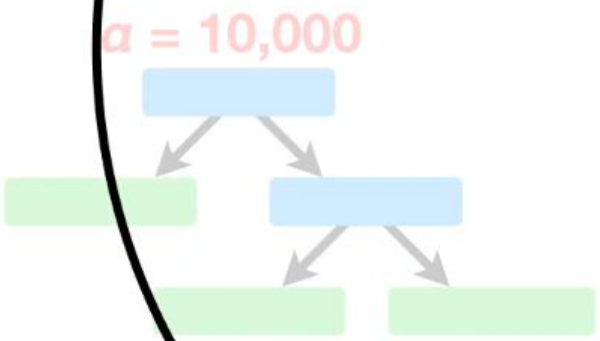


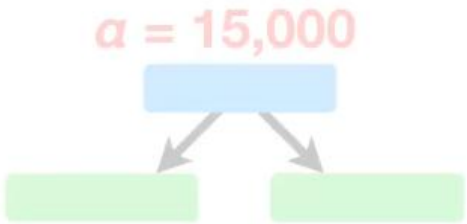
Tree Score = SSR + αT

…and use this tree instead.

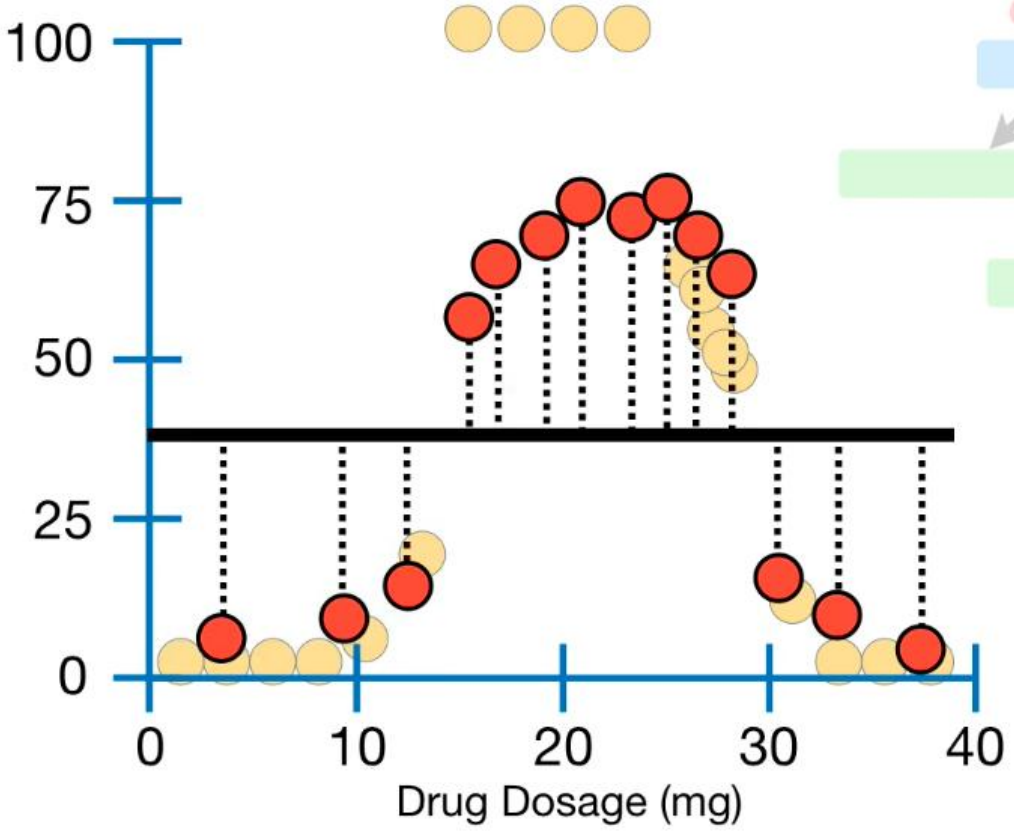


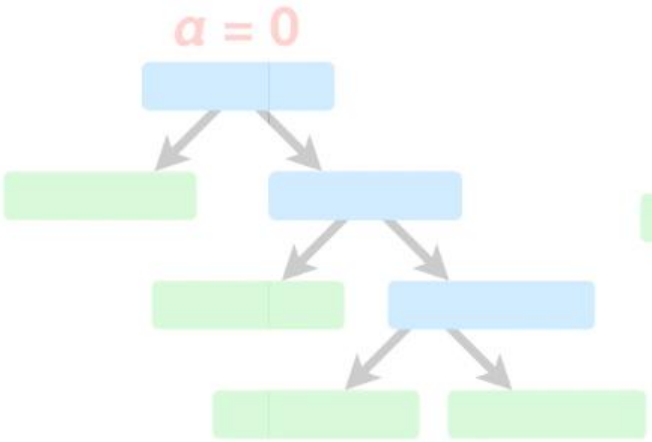


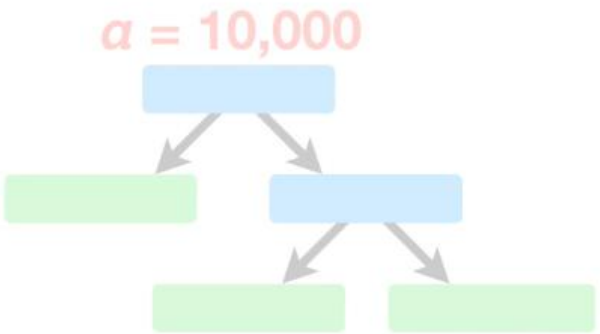


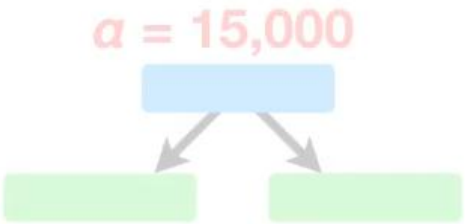


Now calculate the **Sum of Squared Residuals** for each new tree using only the **Testing Data**.



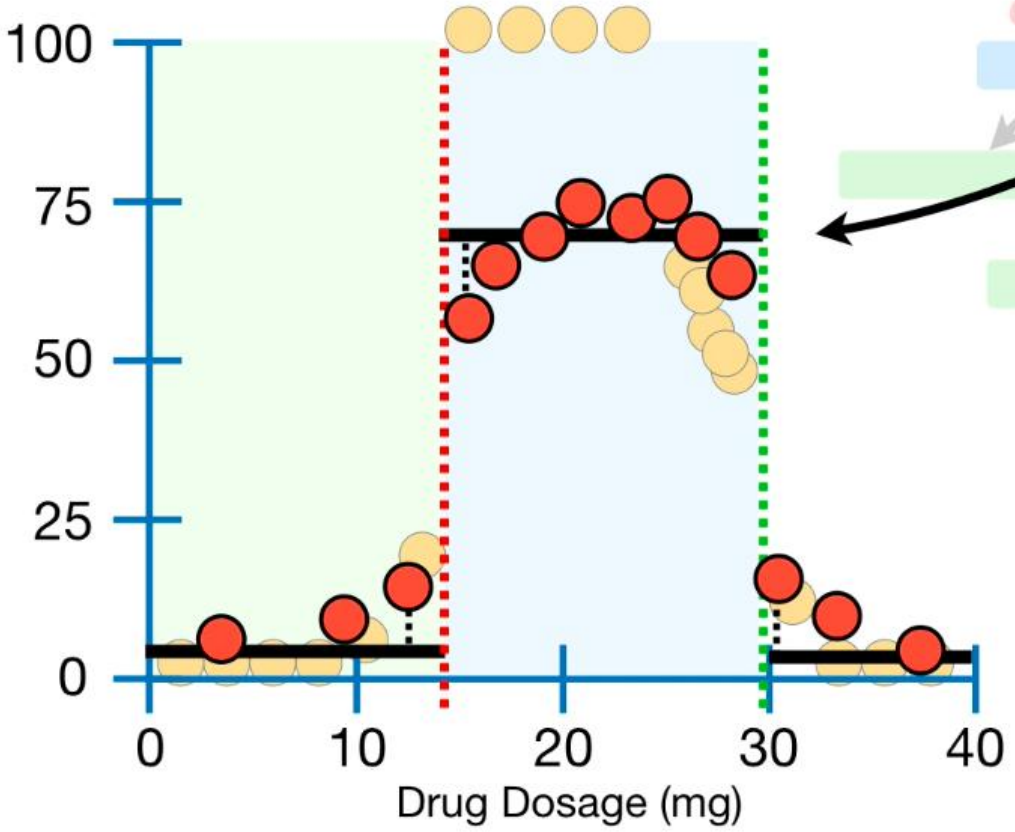


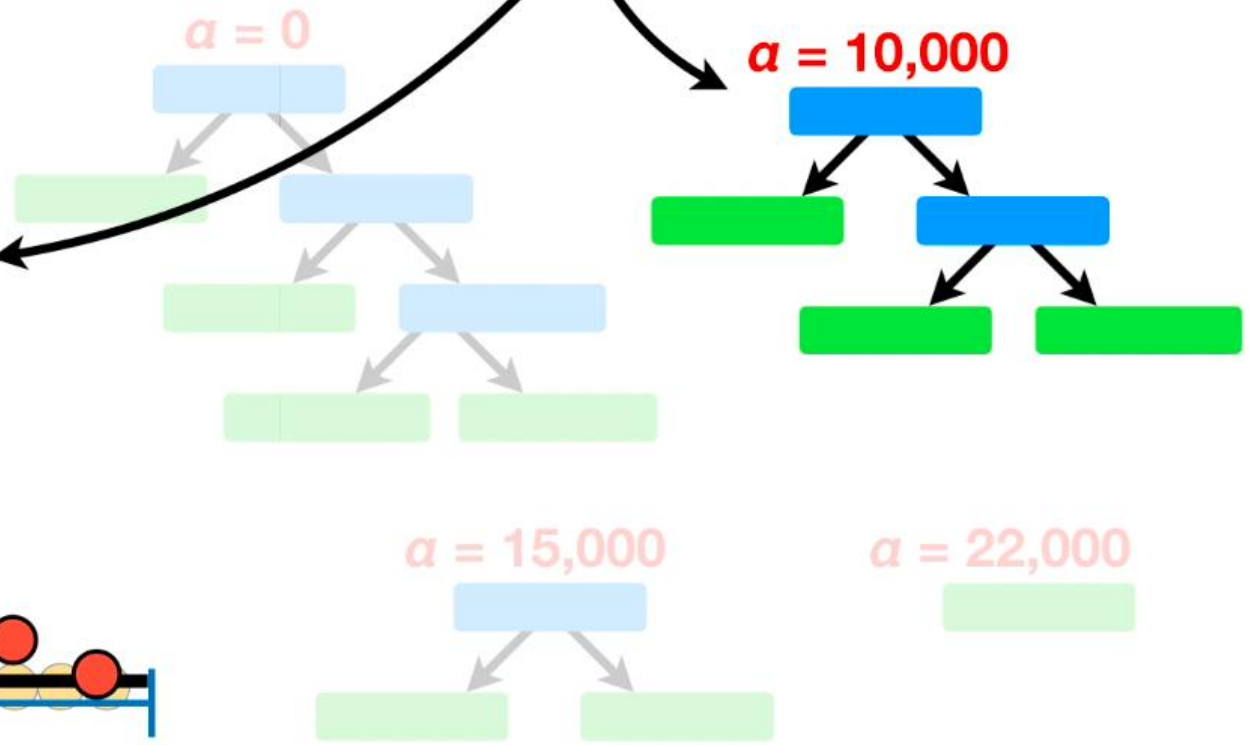




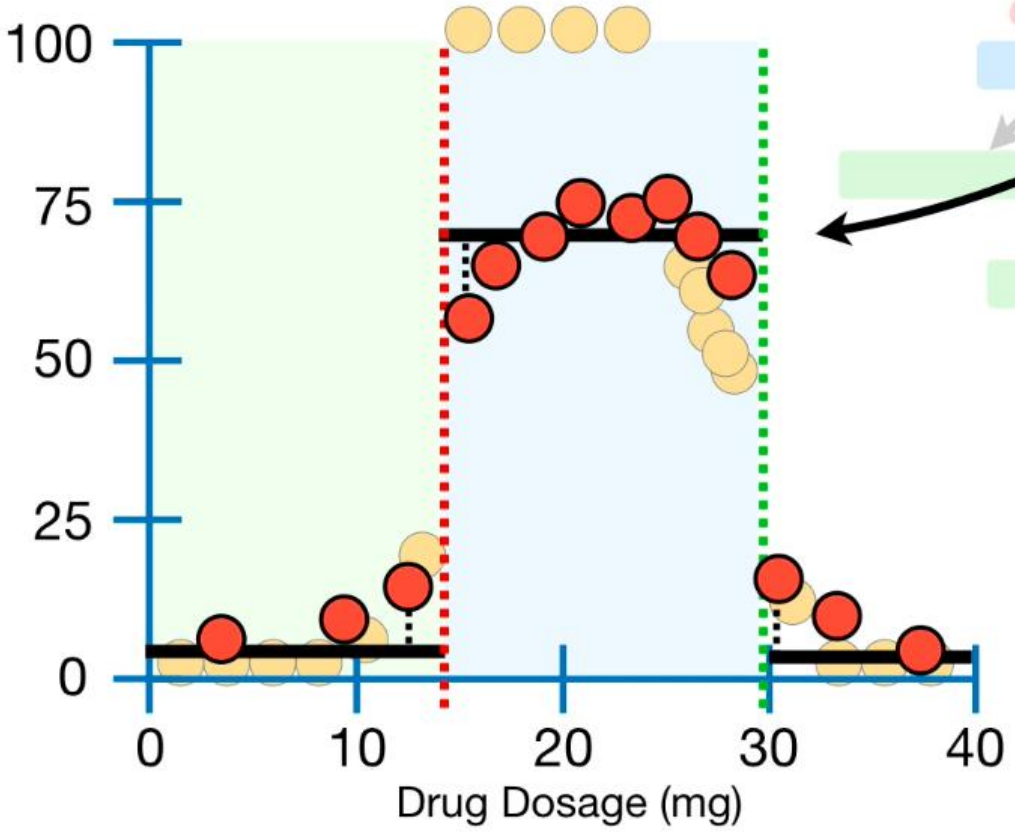
α = 22,000

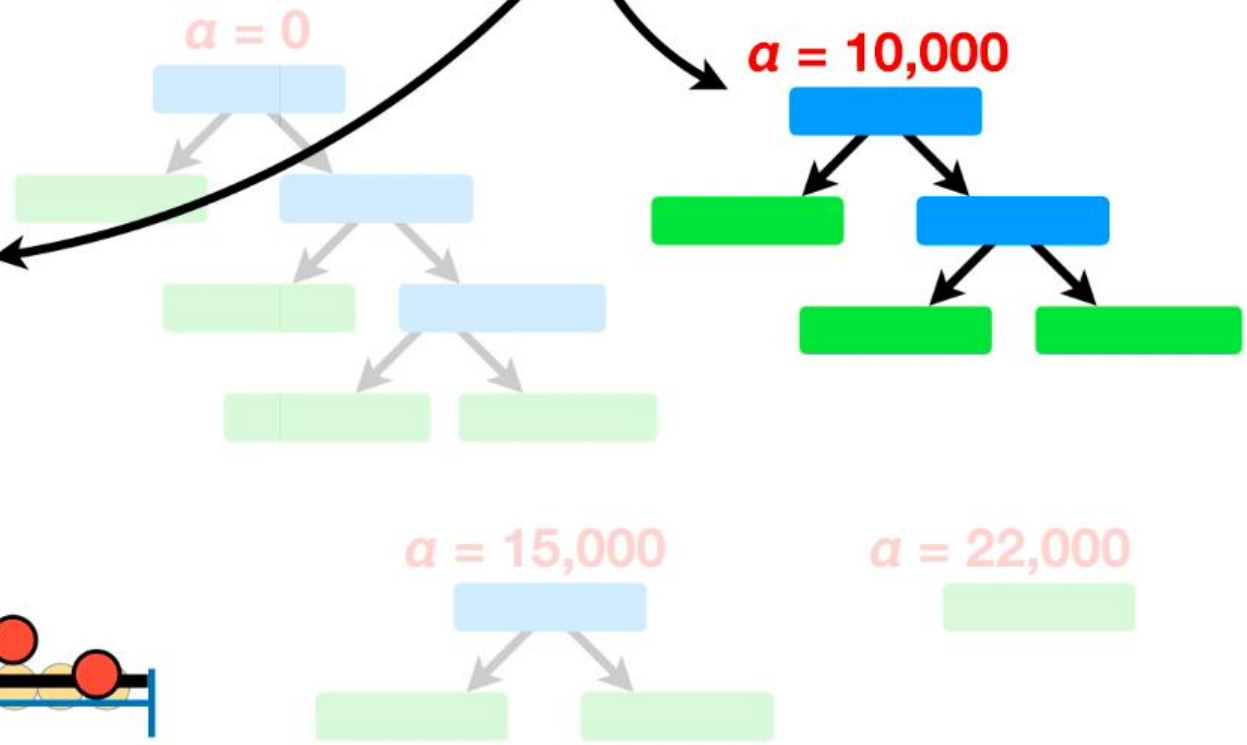
In this case, the tree with α = 10,000 had the smallest **Sum of Squared Residuals** for the **Testing Data**.



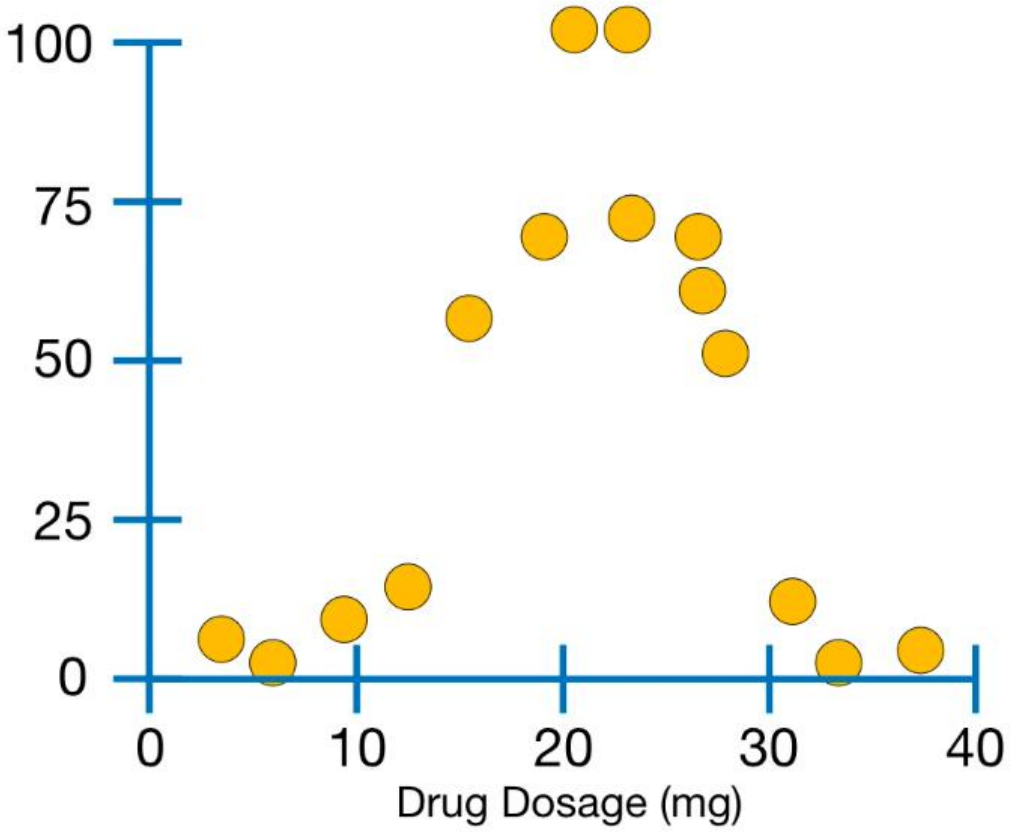


In this case, the tree with α = 10,000 had the smallest **Sum of Squared Residuals** for the **Testing Data**.

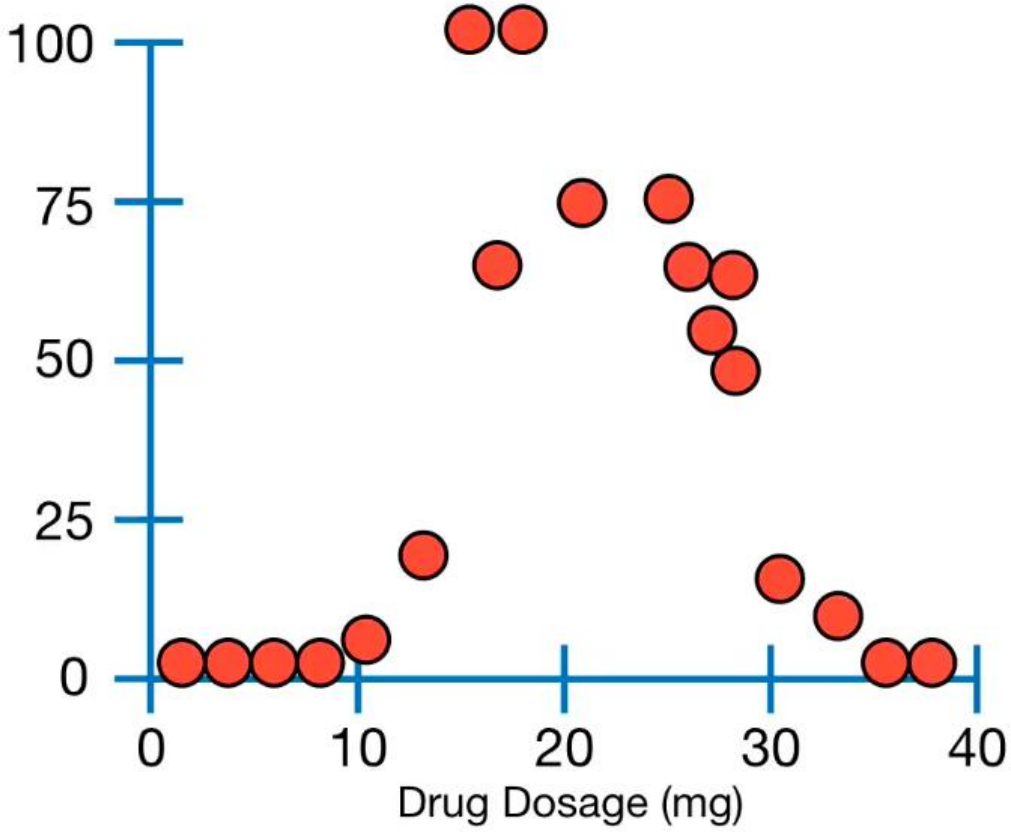




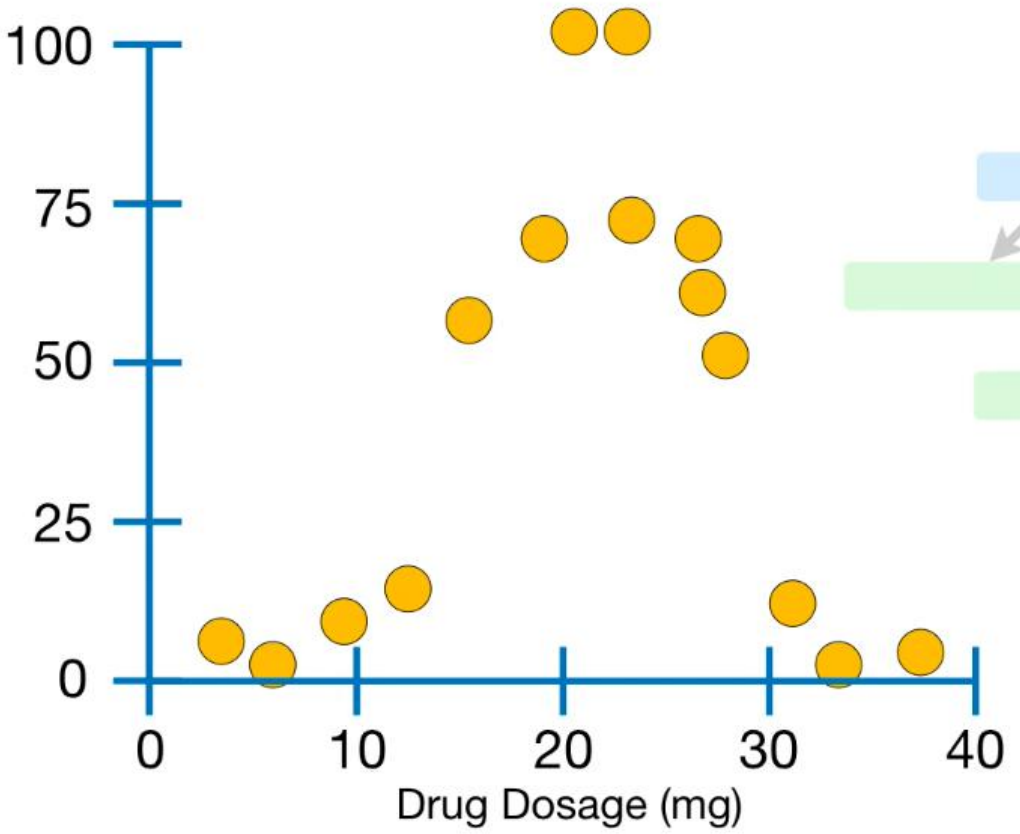
Now we go back and create new **Training Data**…



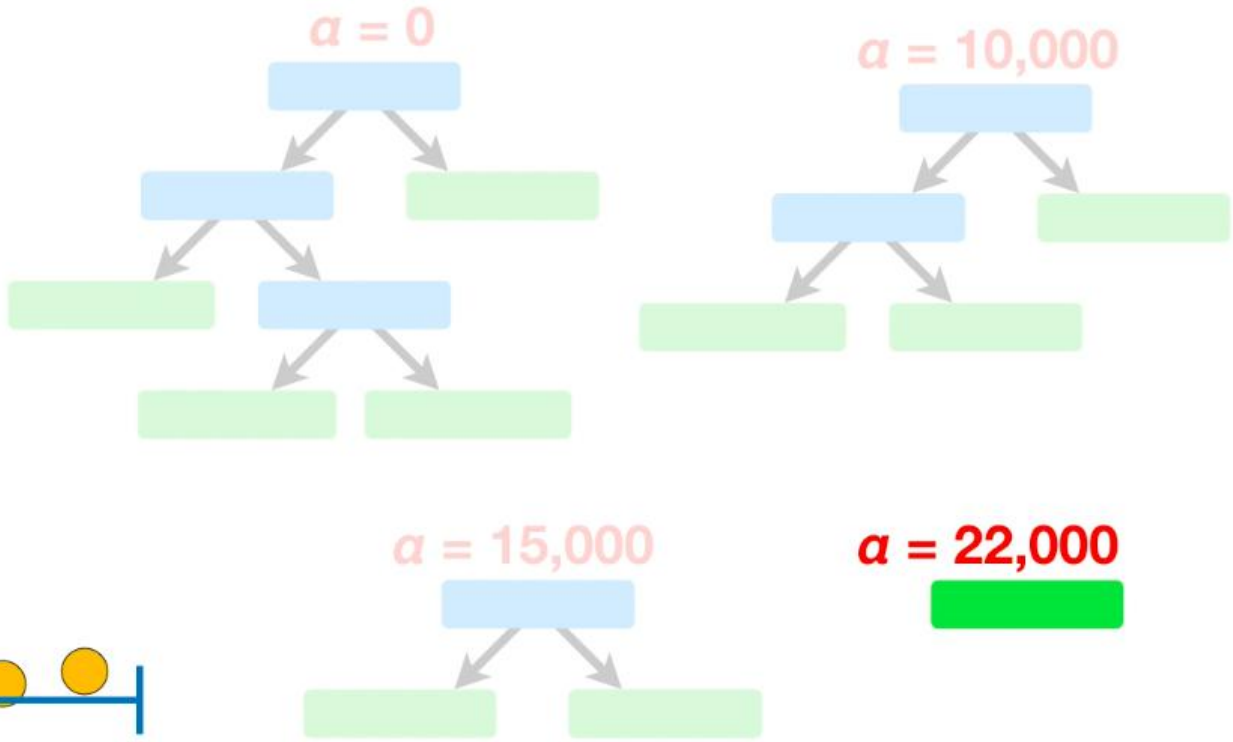
…and new **Testing Data**.



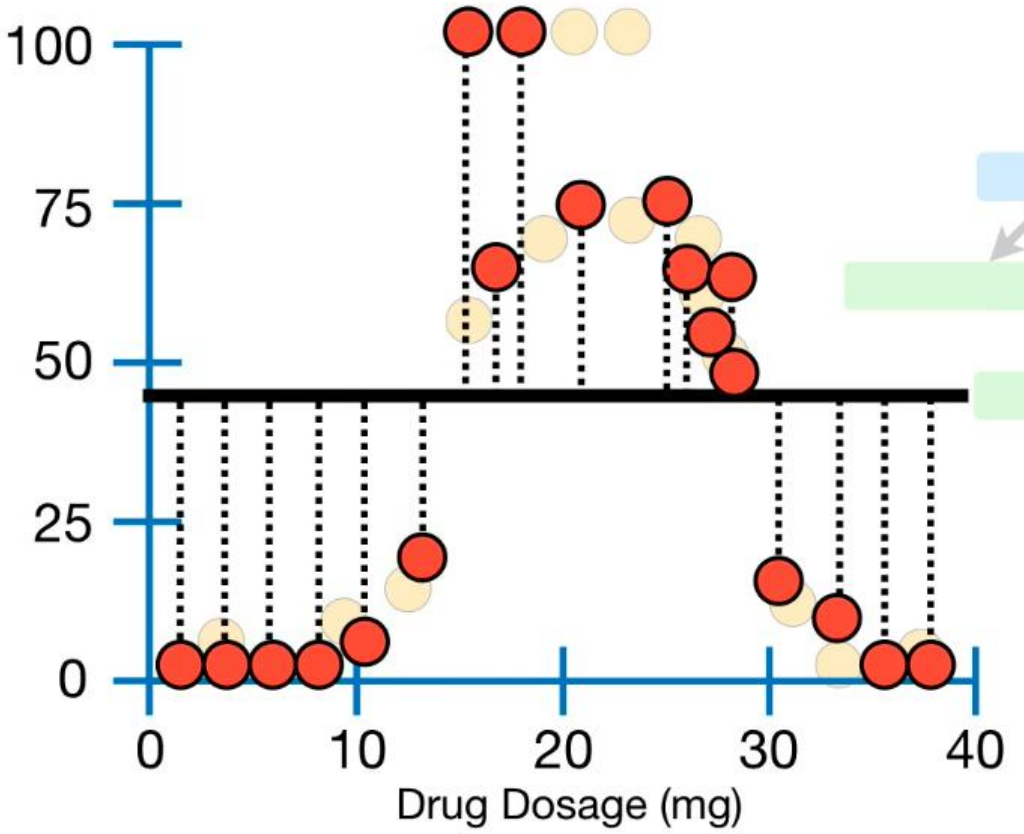
And just using the new **Training Data**…

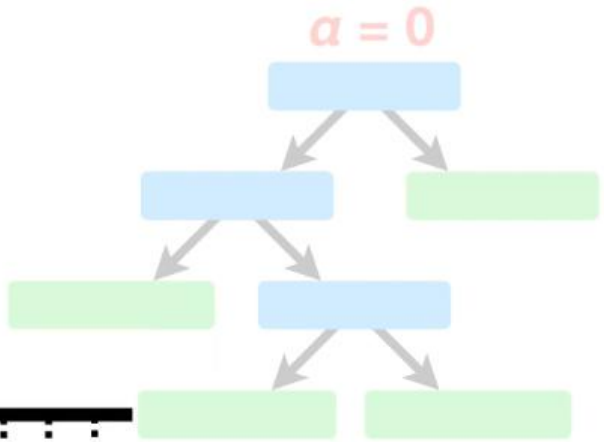


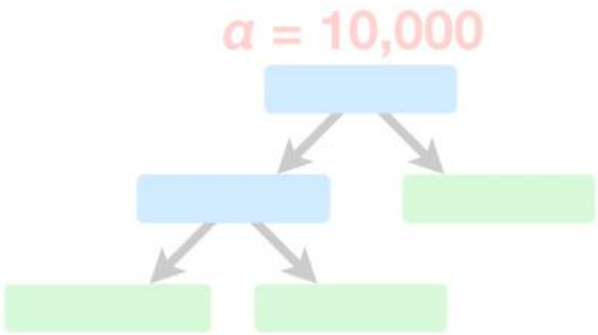
…build a new sequence of trees, from full sized to a leaf, using the ***α*** values we found before.

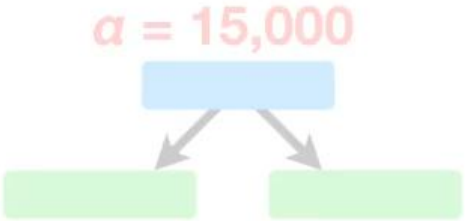


Then we calculate the **Sum of Squared Residuals** using the new **Testing Data**.

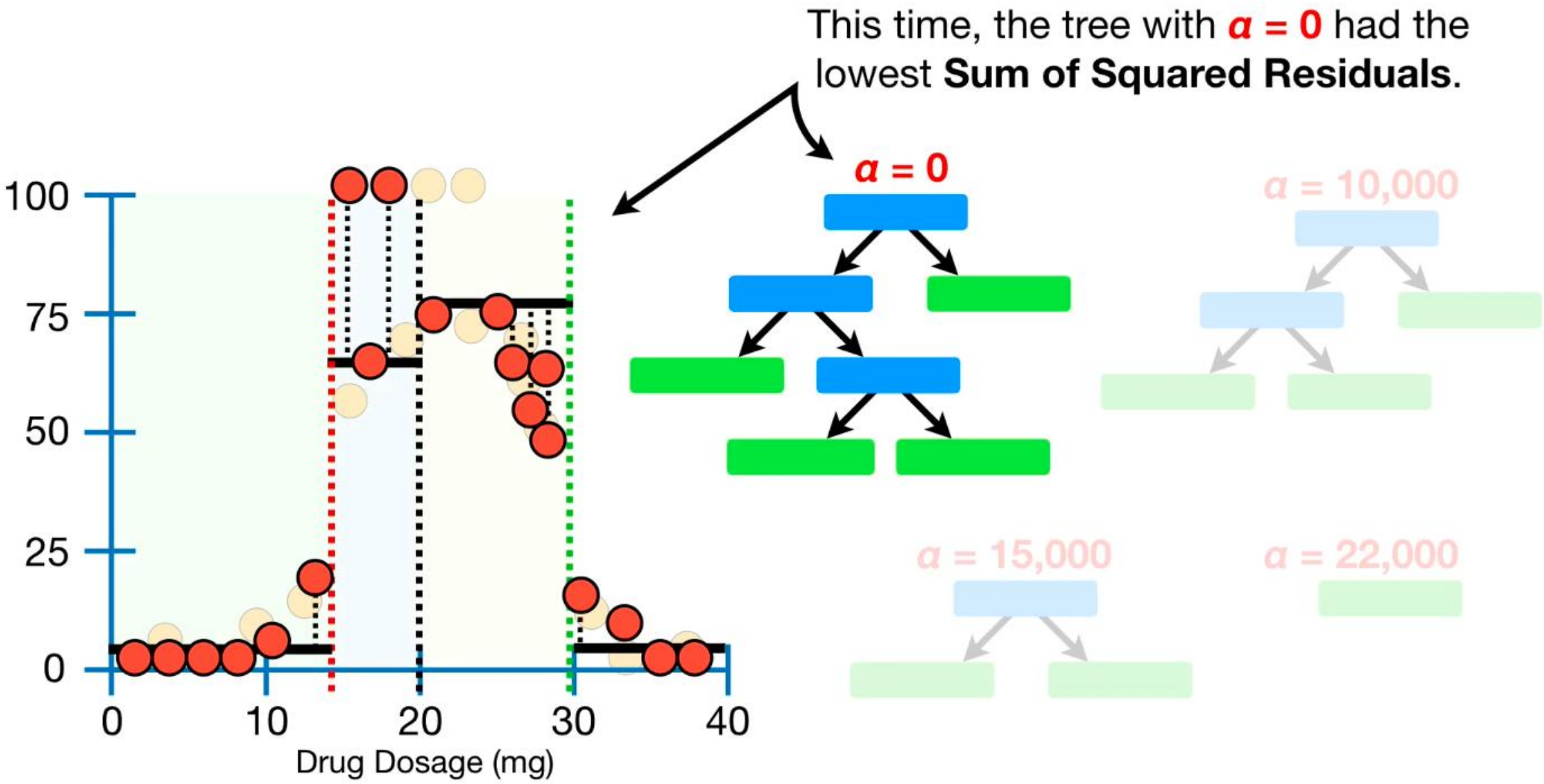




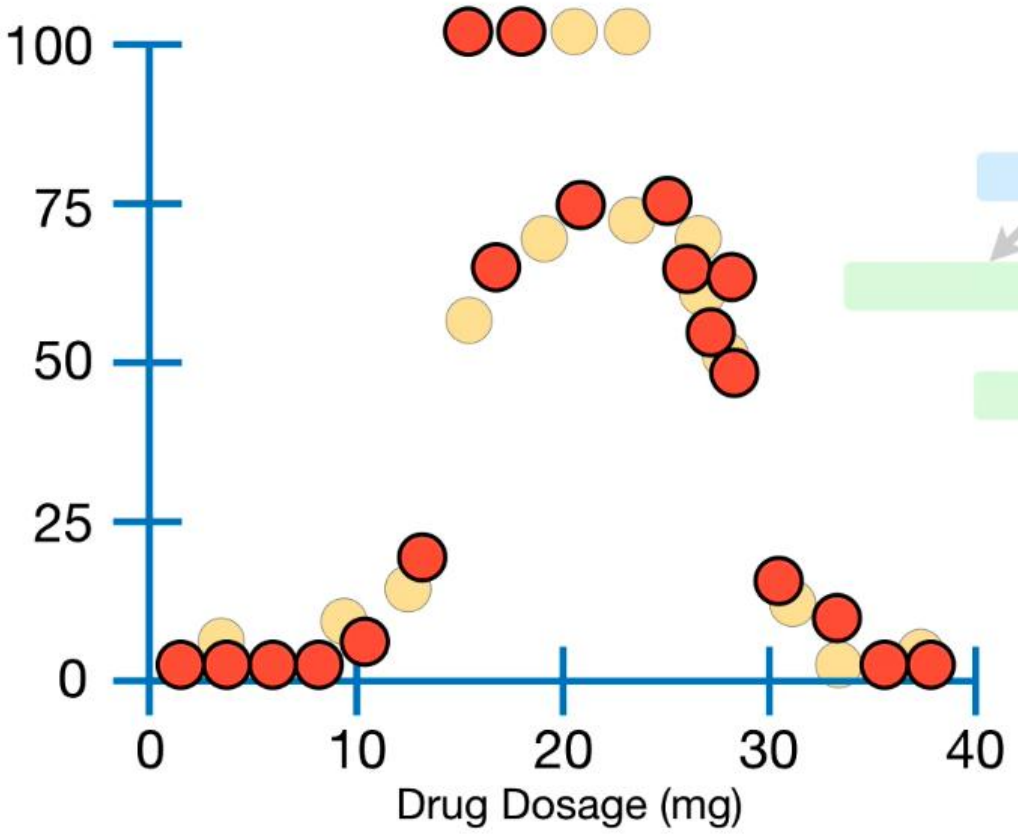


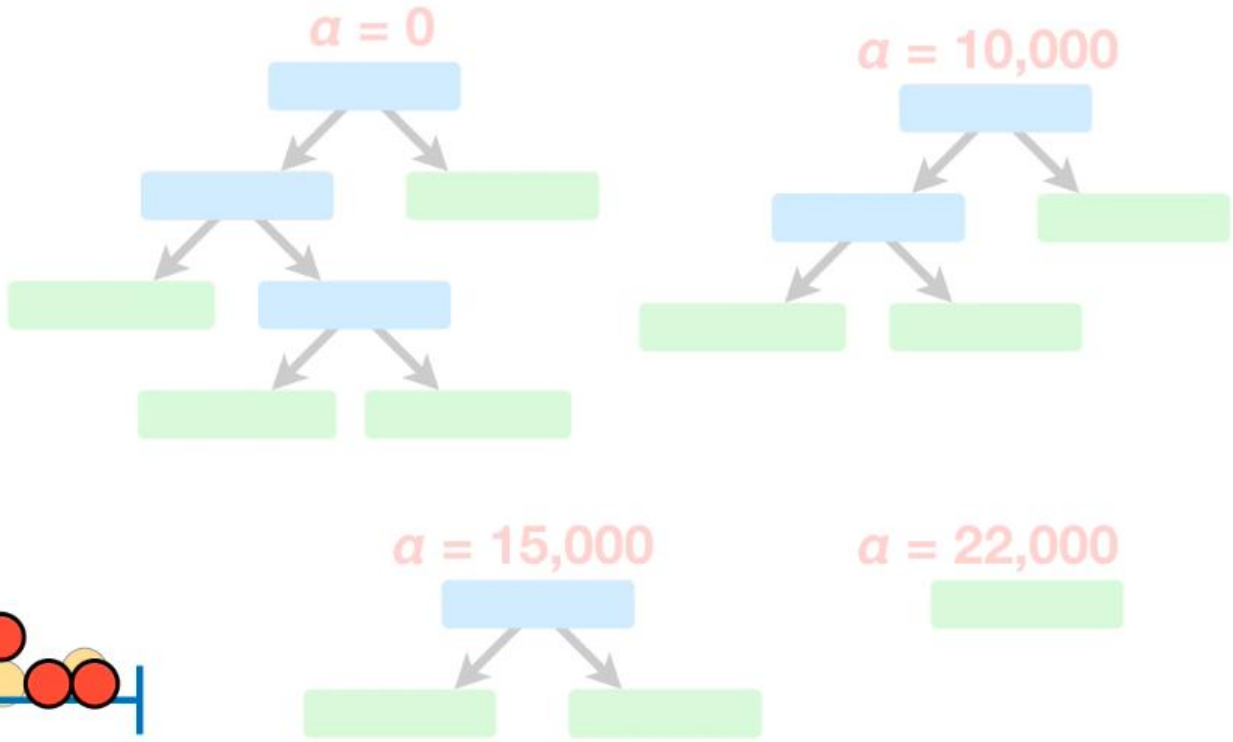


α = 22,000

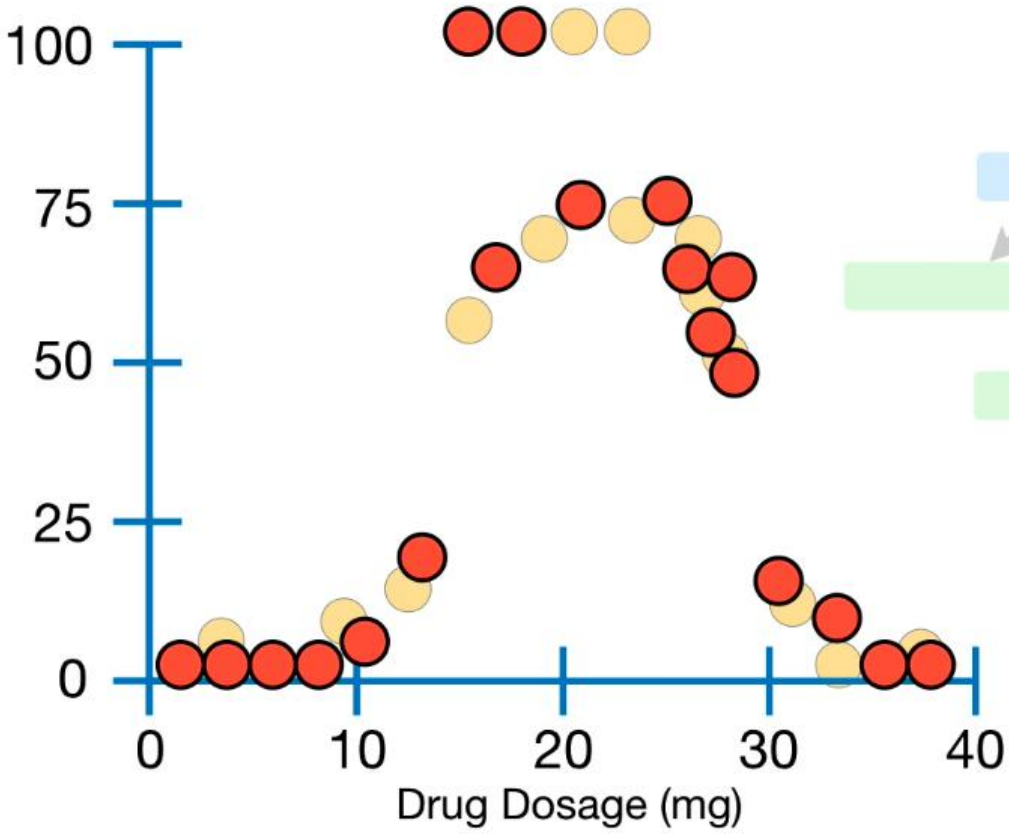

This time, the tree with α = 0 had the lowest Sum of Squared Residuals.
α = 0
α = 10,000
α = 15,000
α = 22,000
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)

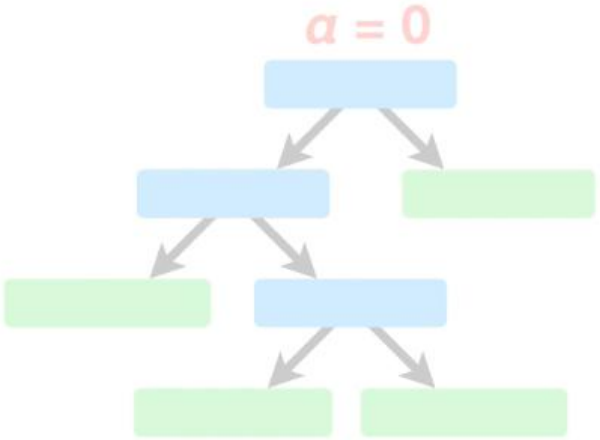
Now we just keep repeating until we have done **10-Fold Cross Validation**…

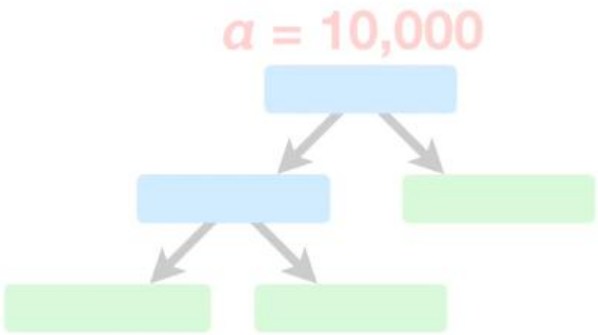


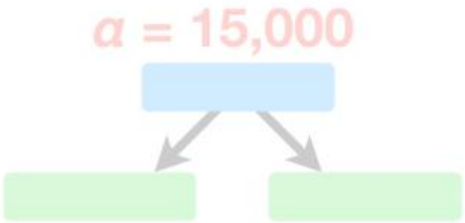


…and the value for ***α*** that, on average, gave us the lowest **Sum of Squared Residuals** with the **Testing Data**, is the final value for ***α***.

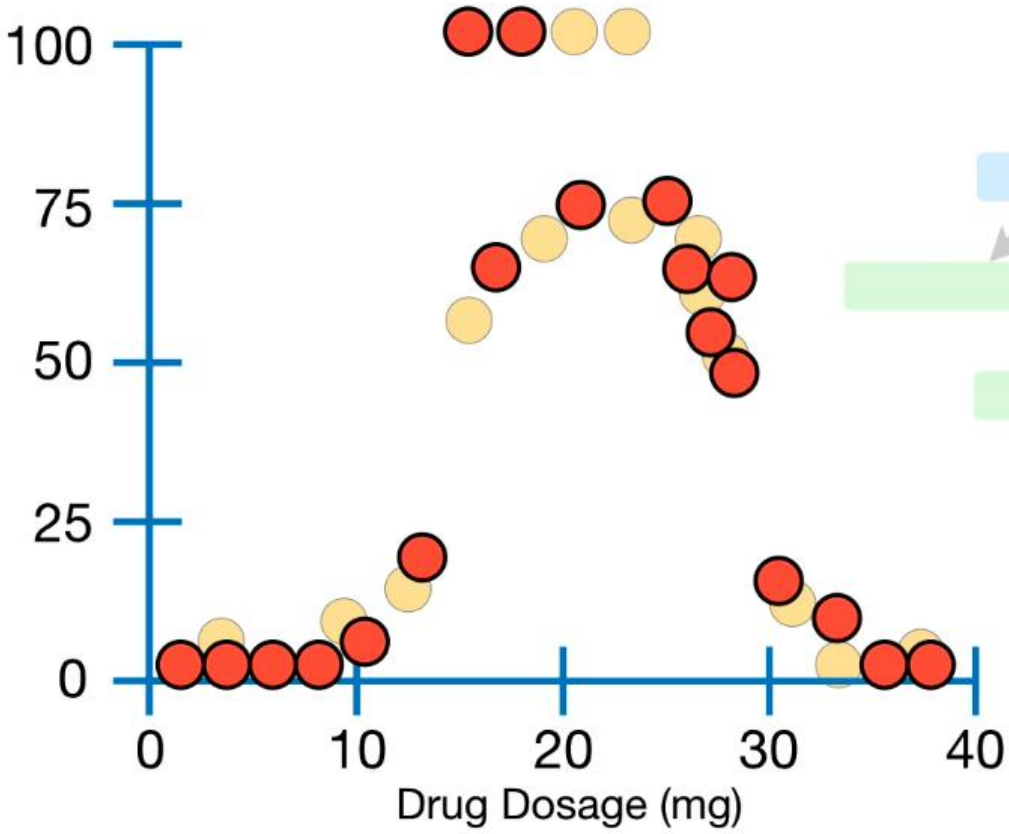


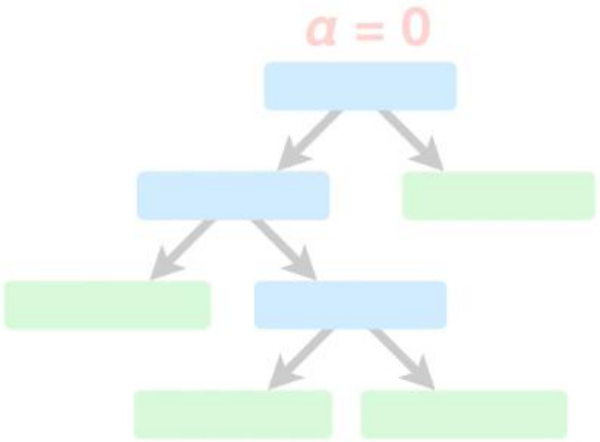


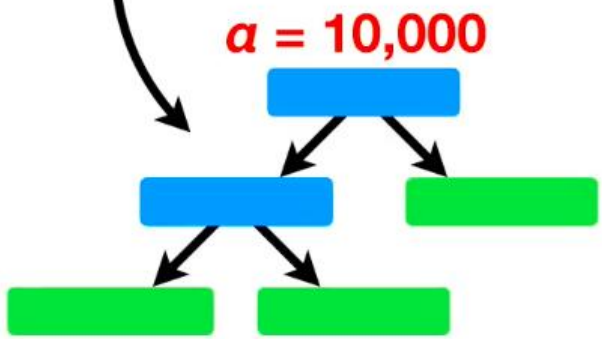


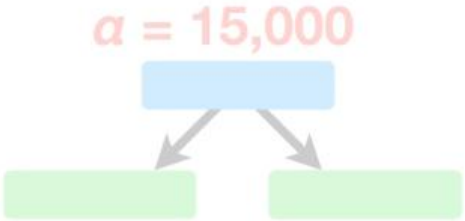


In this case, the optimal trees built with α = 10,000 had, on average, the lowest **Sum of Squared Residuals**…

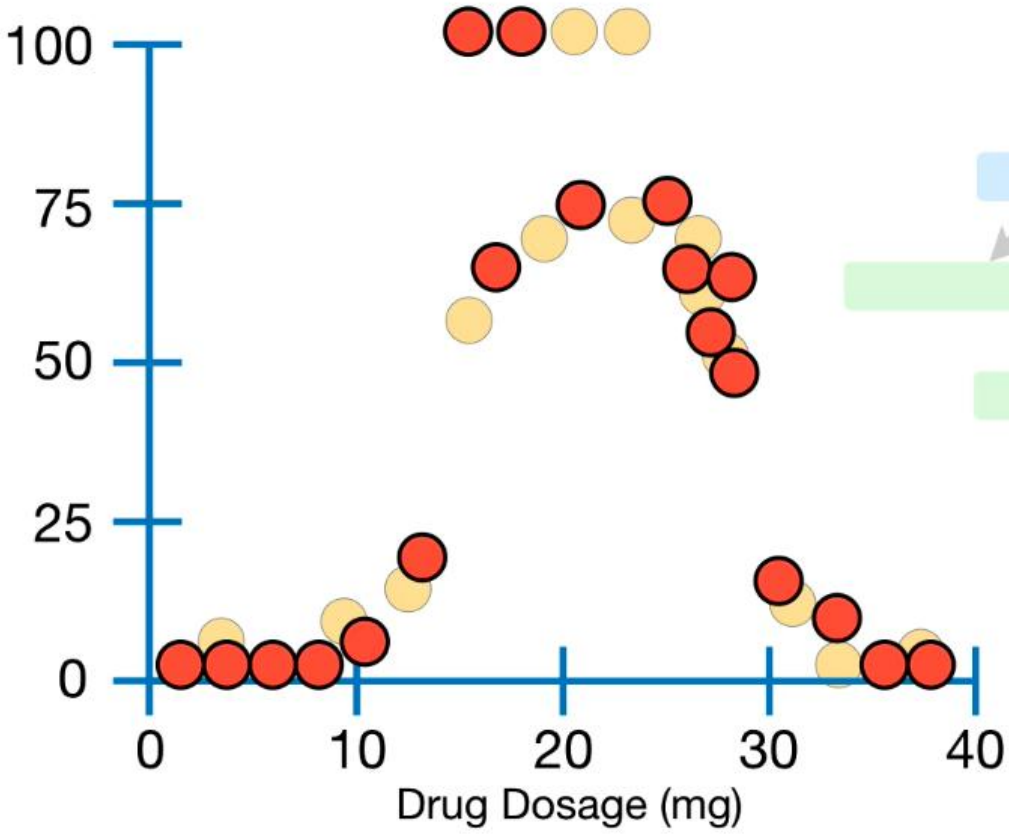


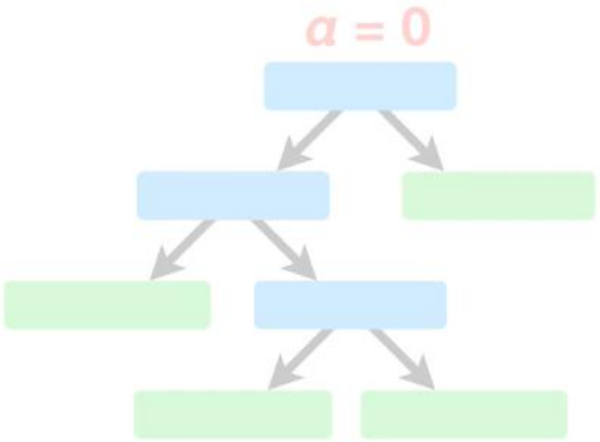


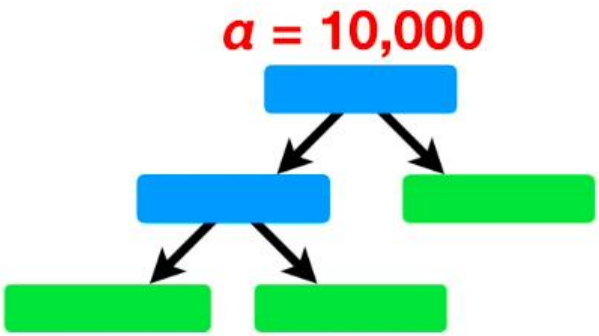


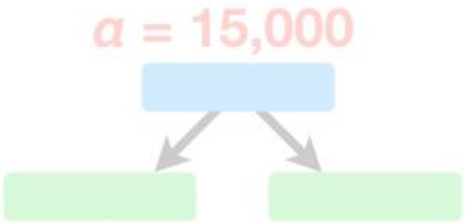


…so α = 10,000 is our final value.

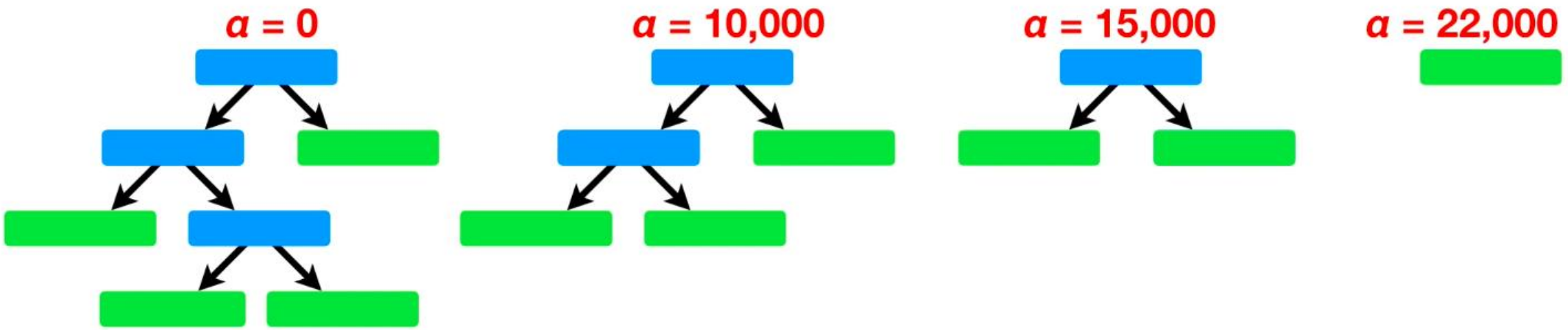




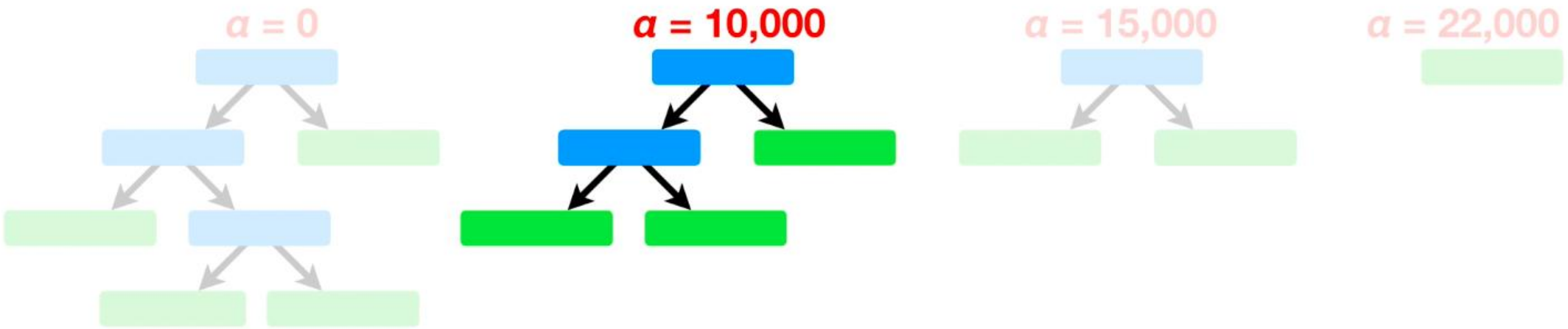




Lastly, we go back to the
original trees and sub-trees
made from the full data.



…and pick the tree that corresponds to the value for α that we selected (α = 10,000).



This sub-tree will be the final, pruned tree.