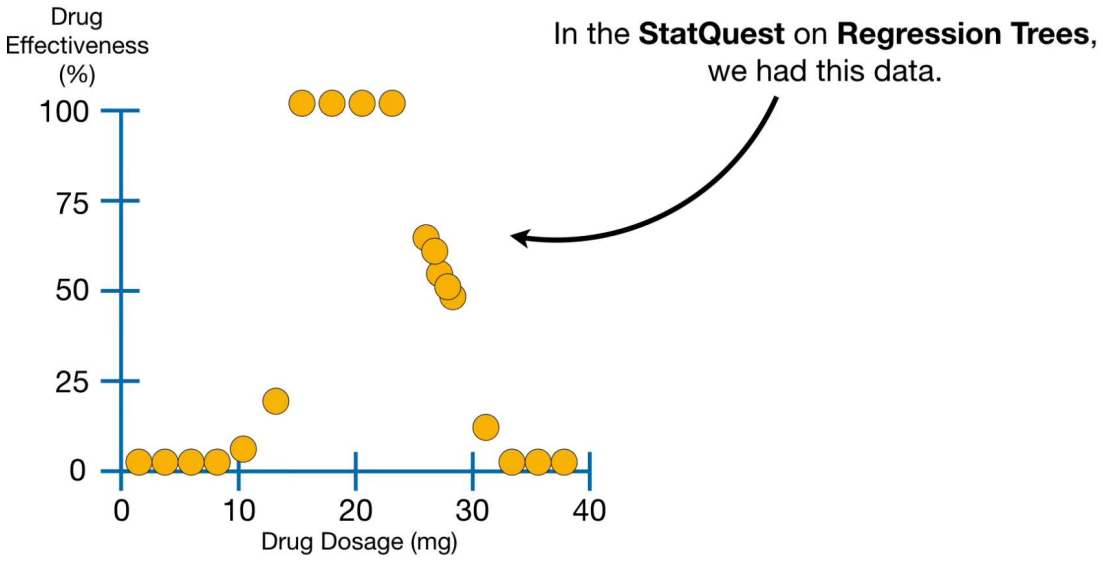
How to Prune Regression Trees

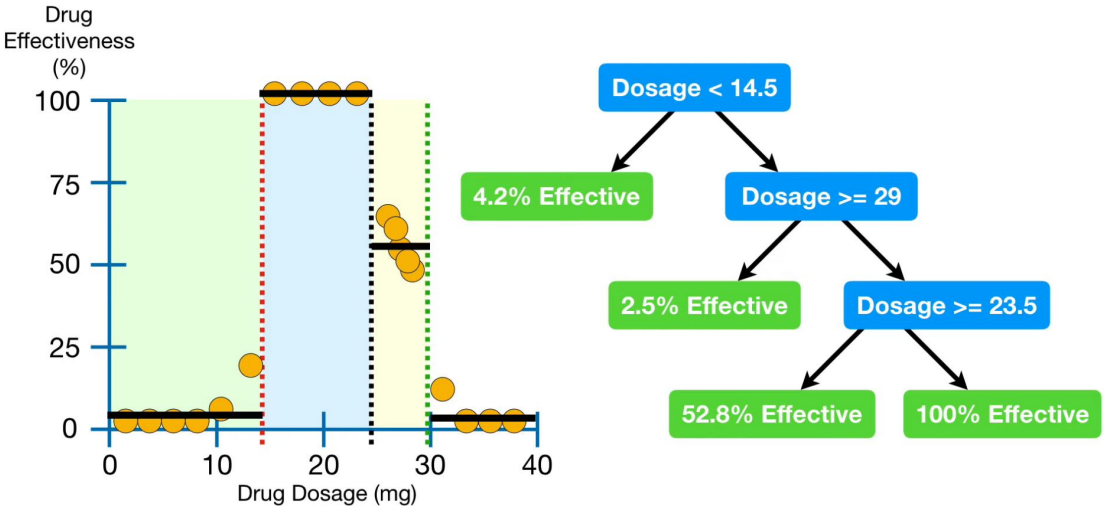
There are several methods for pruning **Regression Trees**.

The one we'll talk about in this **'Quest** is called **Cost Complexity Pruning** aka **Weakest Link Pruning**.

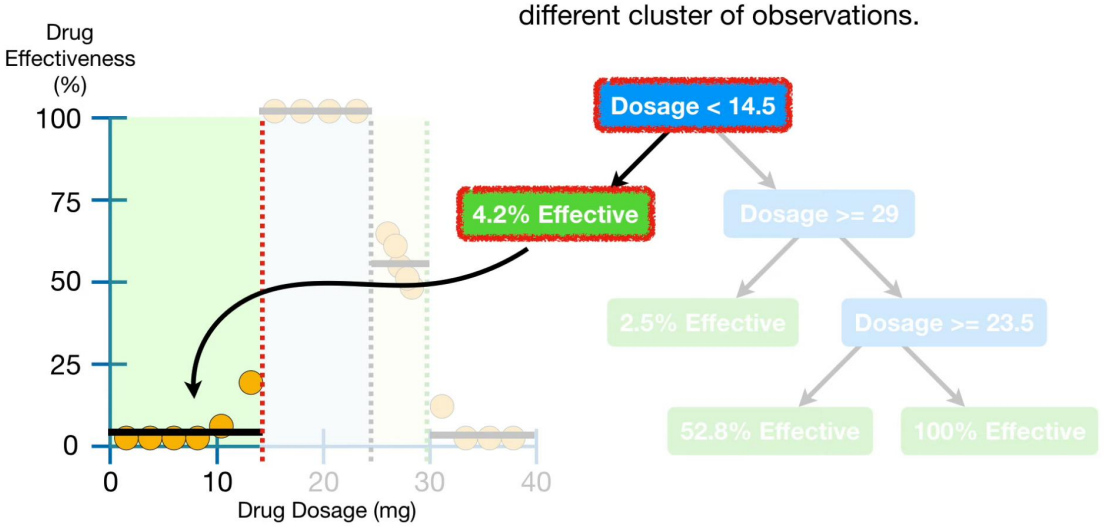
We'll start by giving a general overview of how **Cost Complexity Pruning** works, and then we'll describe how it is used to build **Regression Trees**.



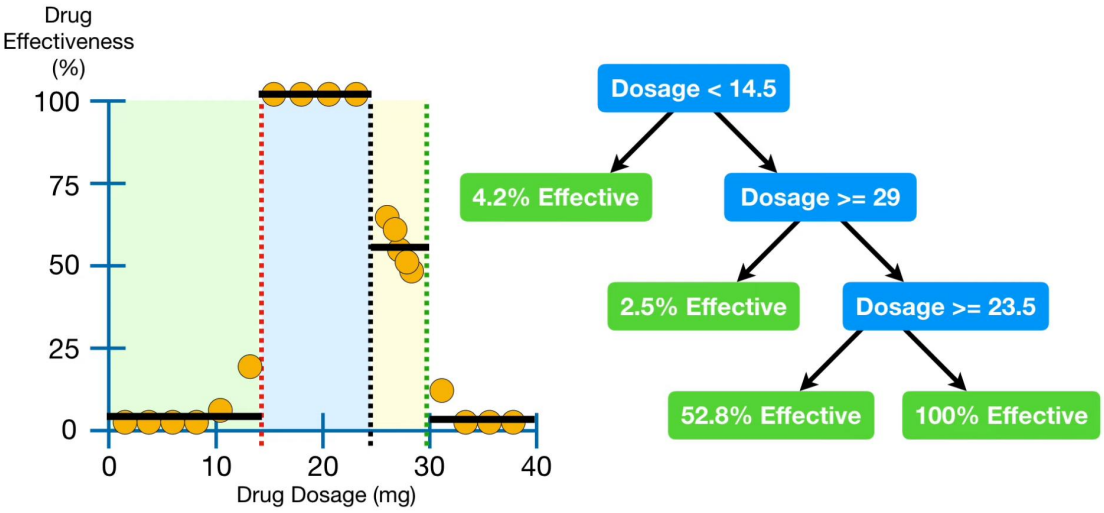
We then fit a **Regression Tree** to the data…



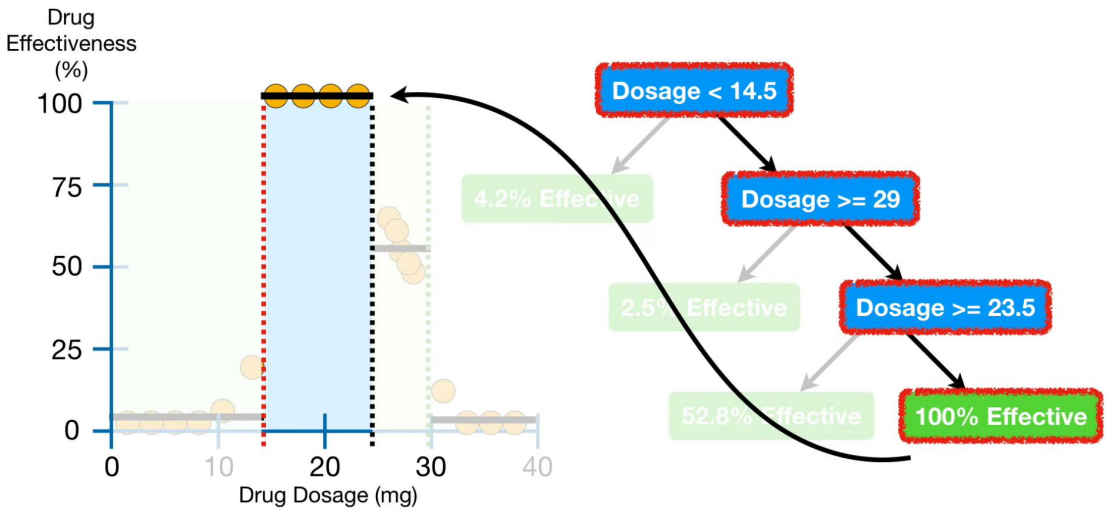
…and each leaf corresponded to the average **Drug Effectiveness** from a different cluster of observations.



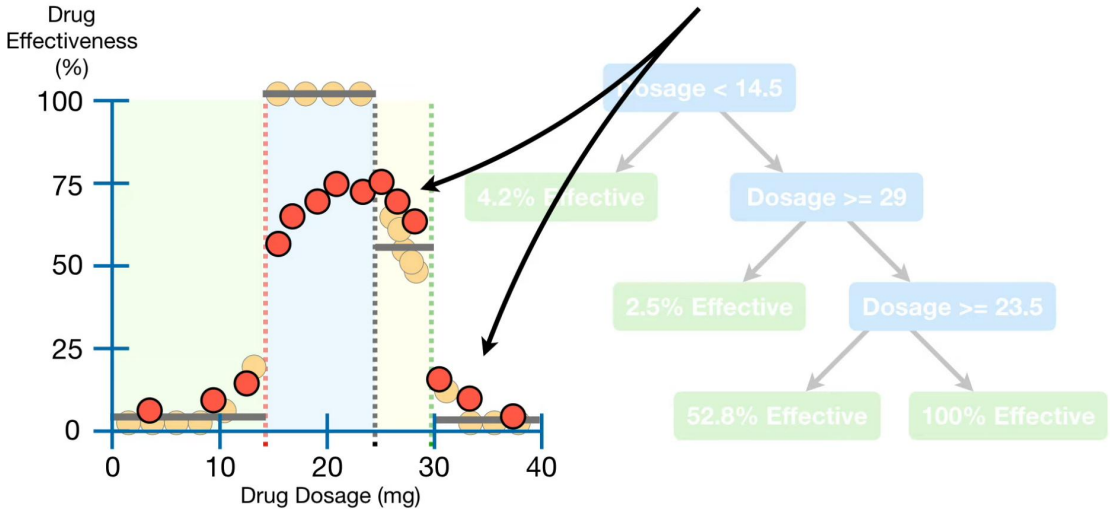
This tree does a pretty good job reflecting the **Training Data**…



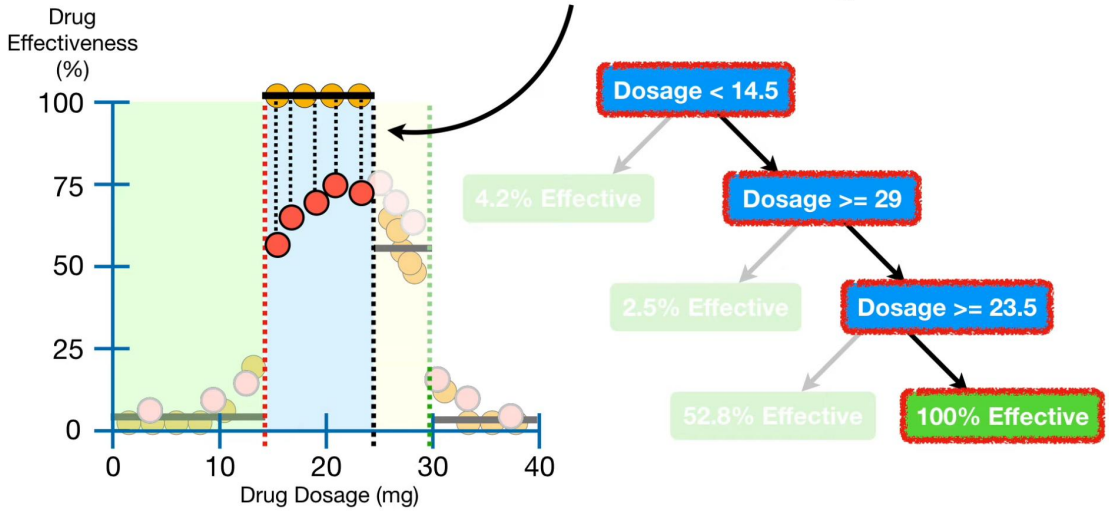
…because each leaf represents a value that is close to the data.

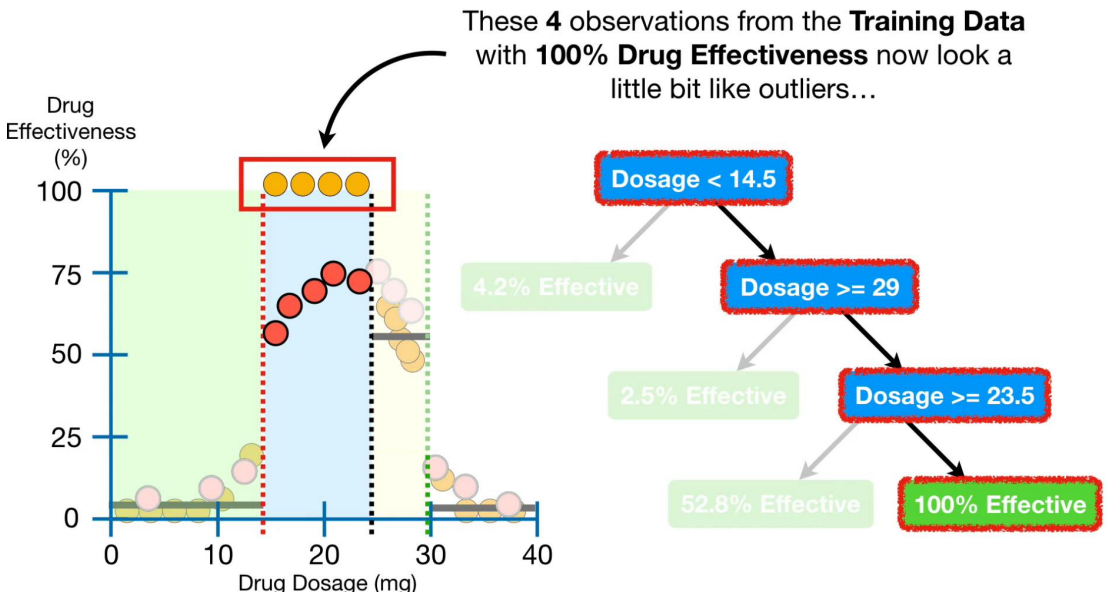


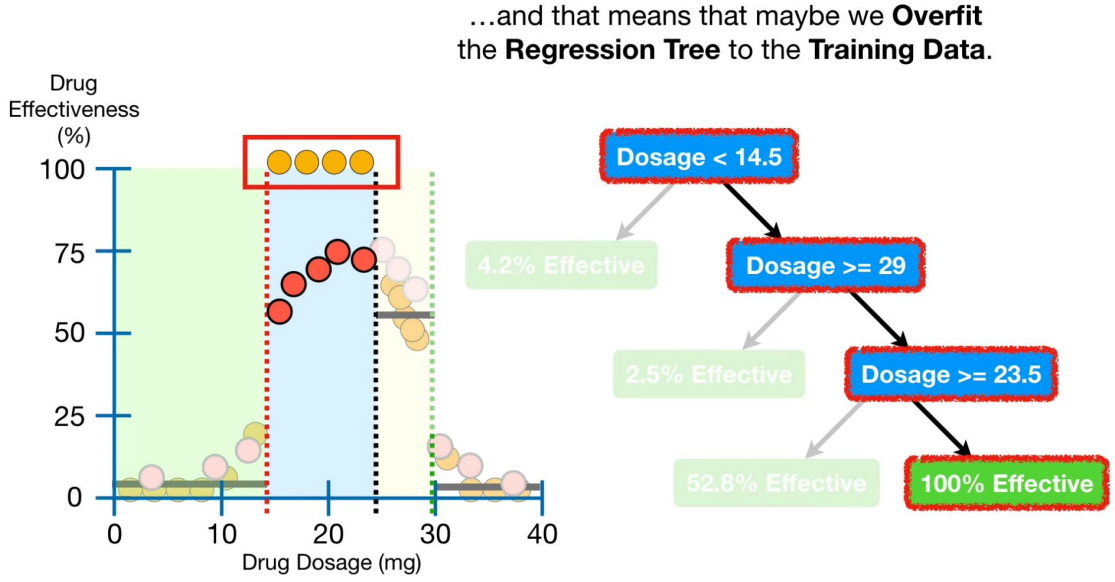
However, what if these **Red Circles** were **Testing Data**?



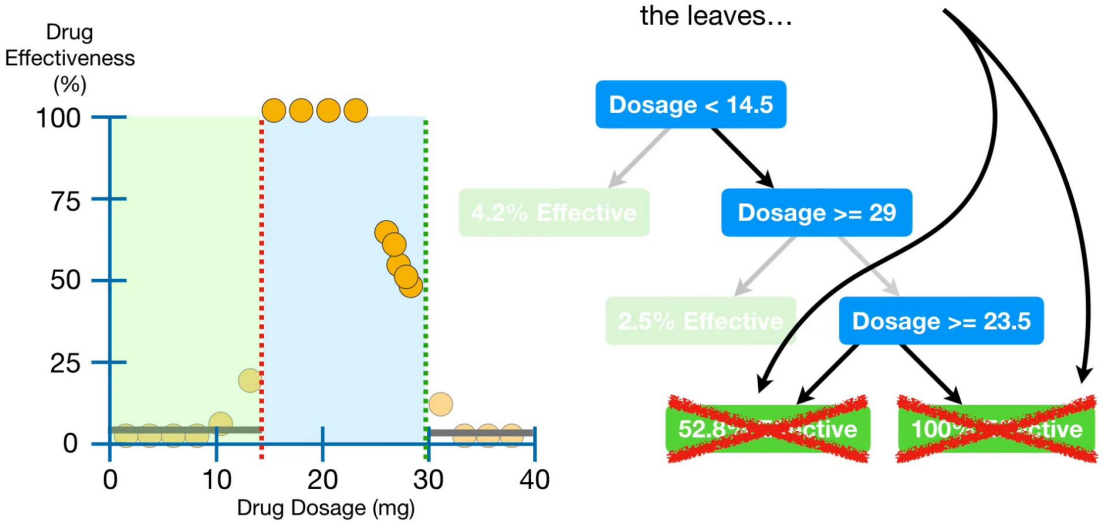
And the **Residuals** for these **Observations** are much larger.



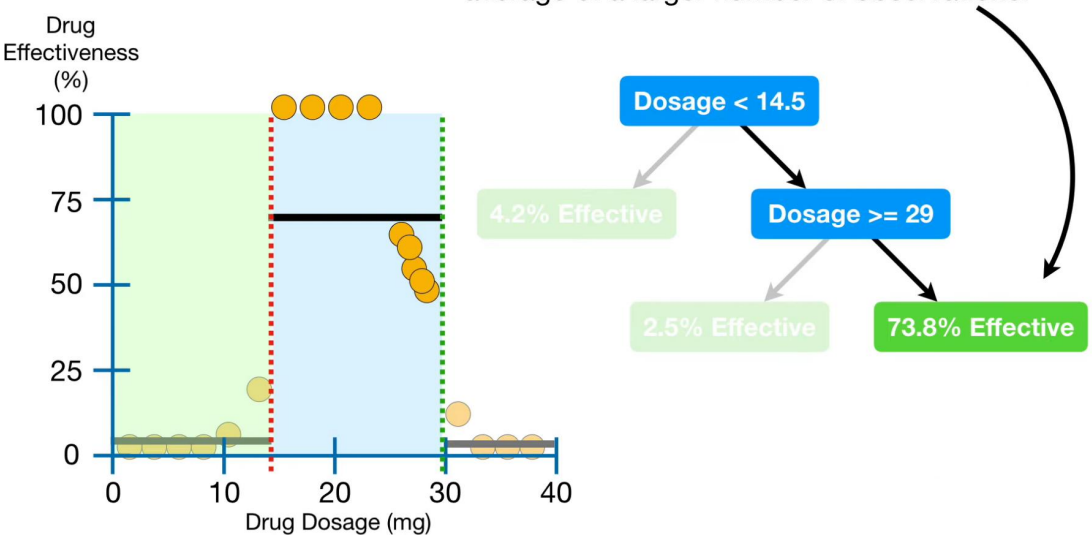

These 4 observations from the Training Data with 100% Drug Effectiveness now look a little bit like outliers…
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
Dosage >= 29
2.5% Effective
Dosage >= 23.5
52.8% Effective
100% Effective

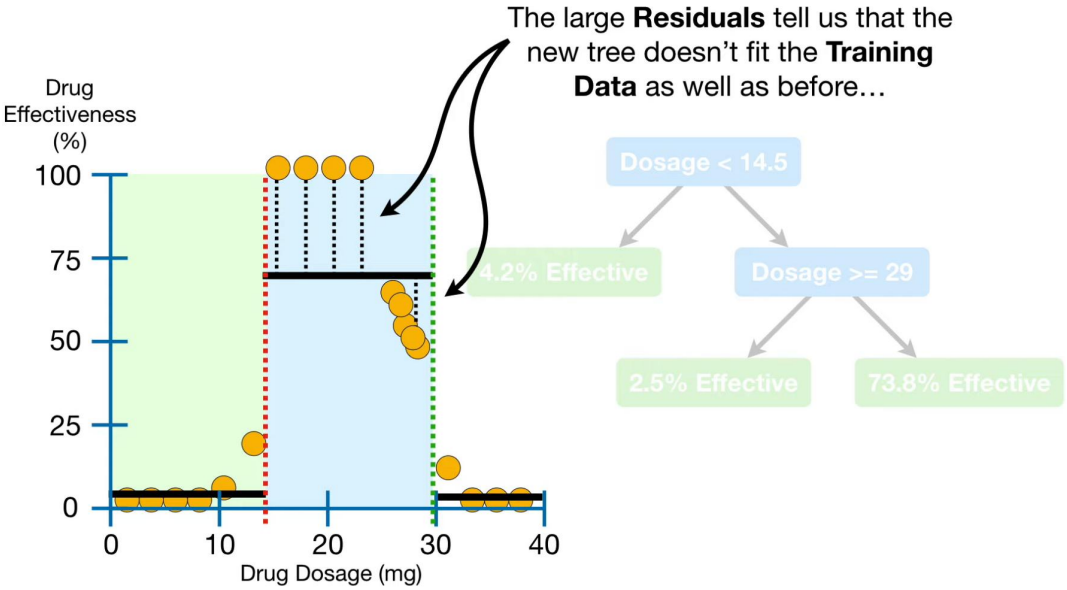

…and that means that maybe we Overfit the Regression Tree to the Training Data.
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
Dosage >= 29
2.5% Effective
Dosage >= 23.5
52.8% Effective
100% Effective

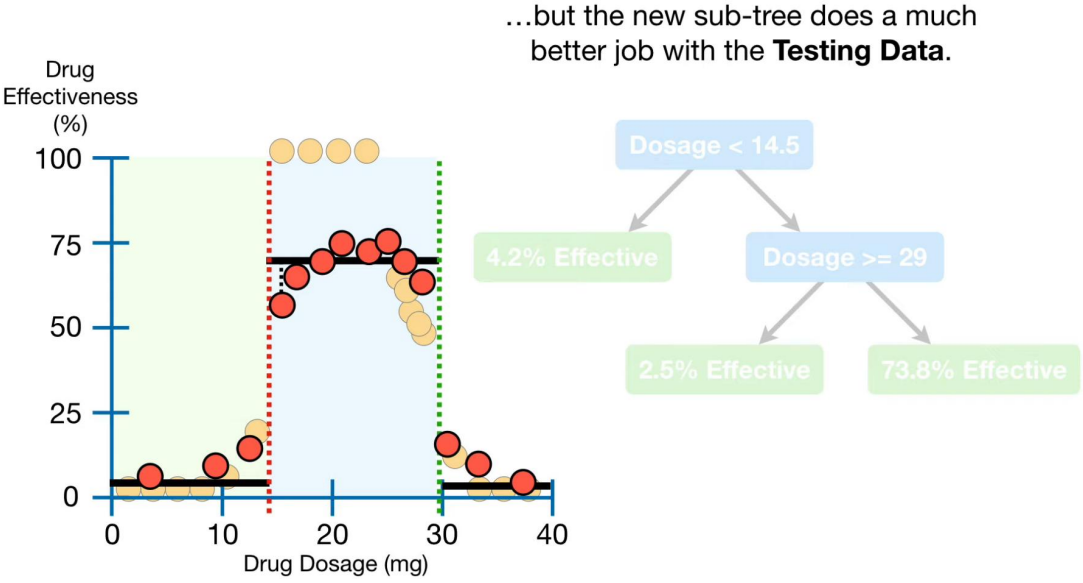
One way to prevent **Overfitting** a **Regression Tree** to the **Training Data** is to remove some of the leaves…



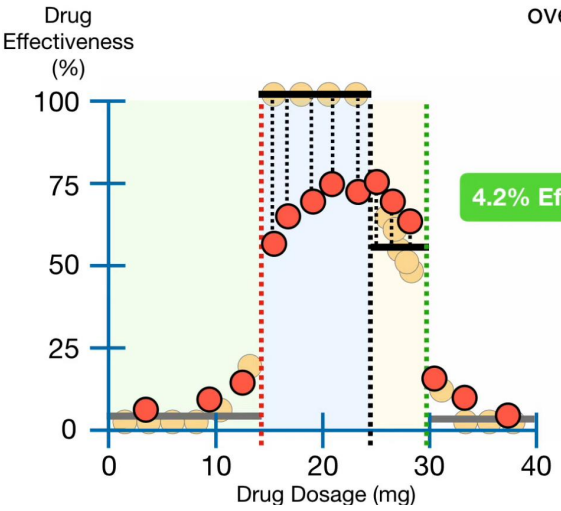
…and replace the split with a leaf that is the average of a larger number of observations.

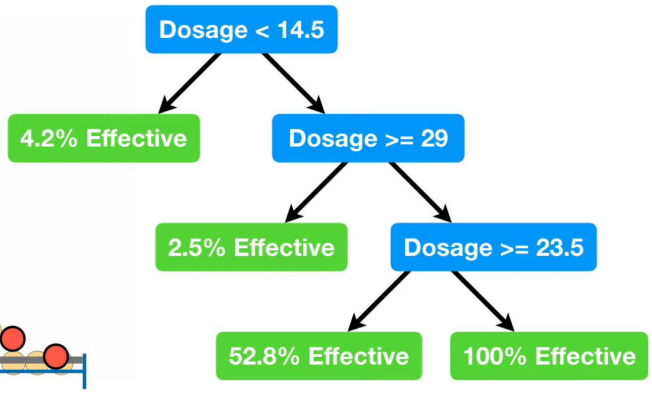



The large **Residuals** tell us that the new tree doesn't fit the **Training Data** as well as before…
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
Dosage >= 29
2.5% Effective
73.8% Effective

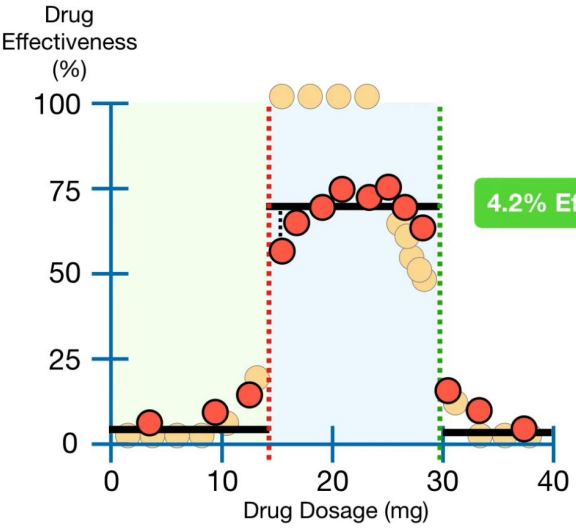

…but the new sub-tree does a much better job with the **Testing Data**.
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
Dosage >= 29
2.5% Effective
73.8% Effective

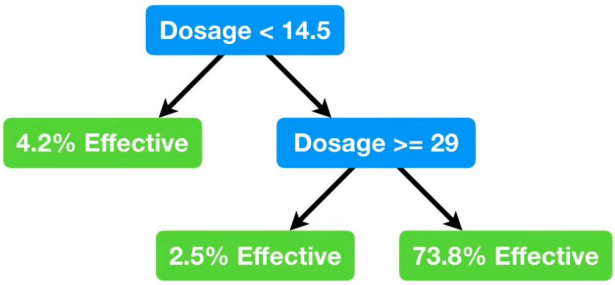
Thus, the main idea behind pruning a **Regression Tree** is to prevent overfitting the **Training Data**…

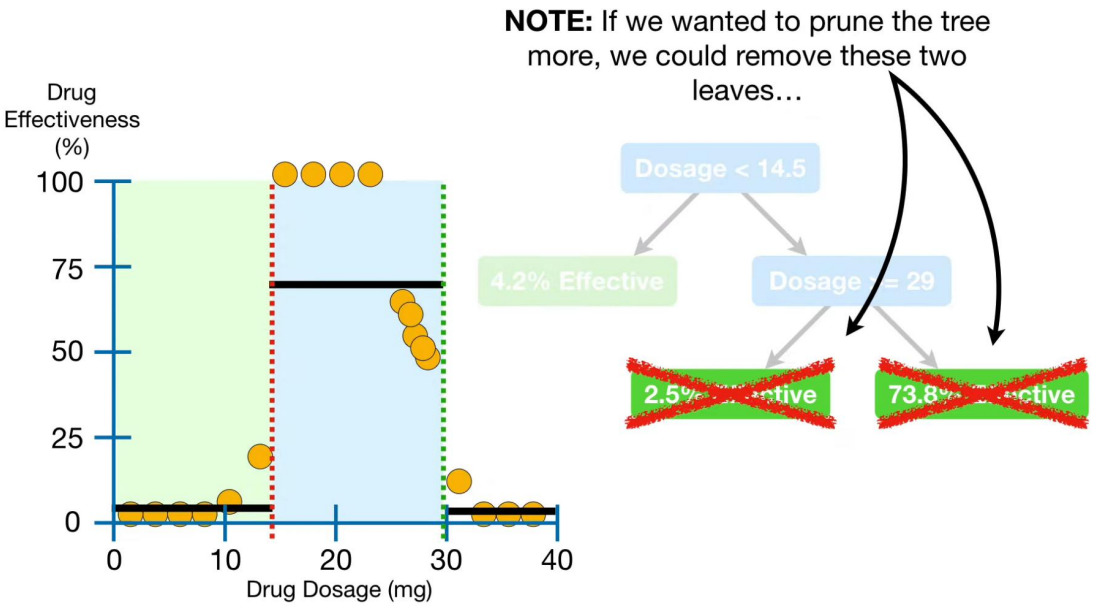


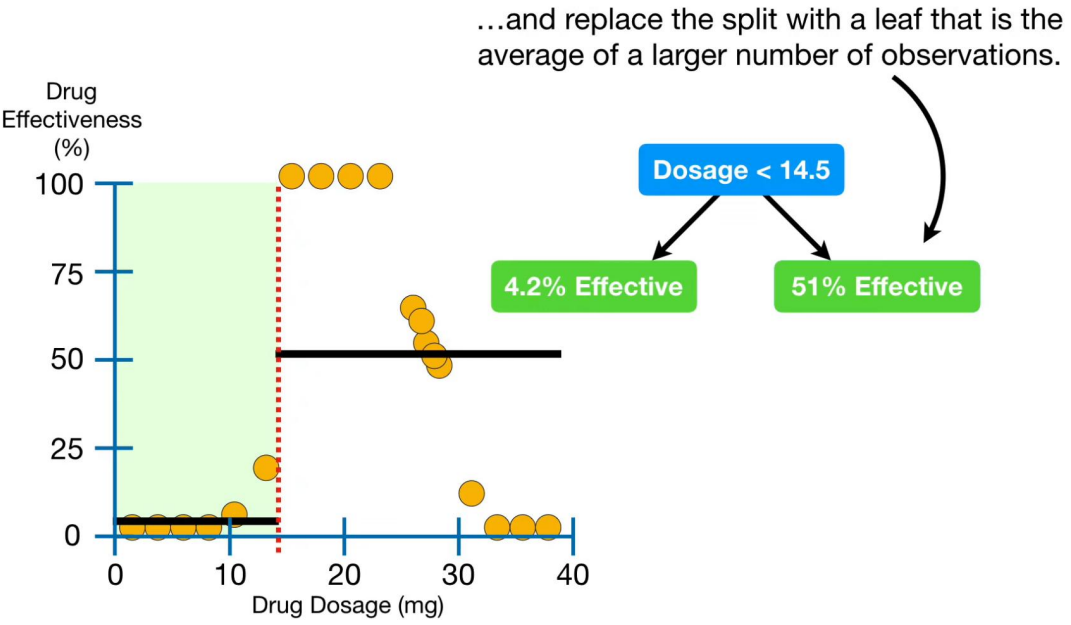


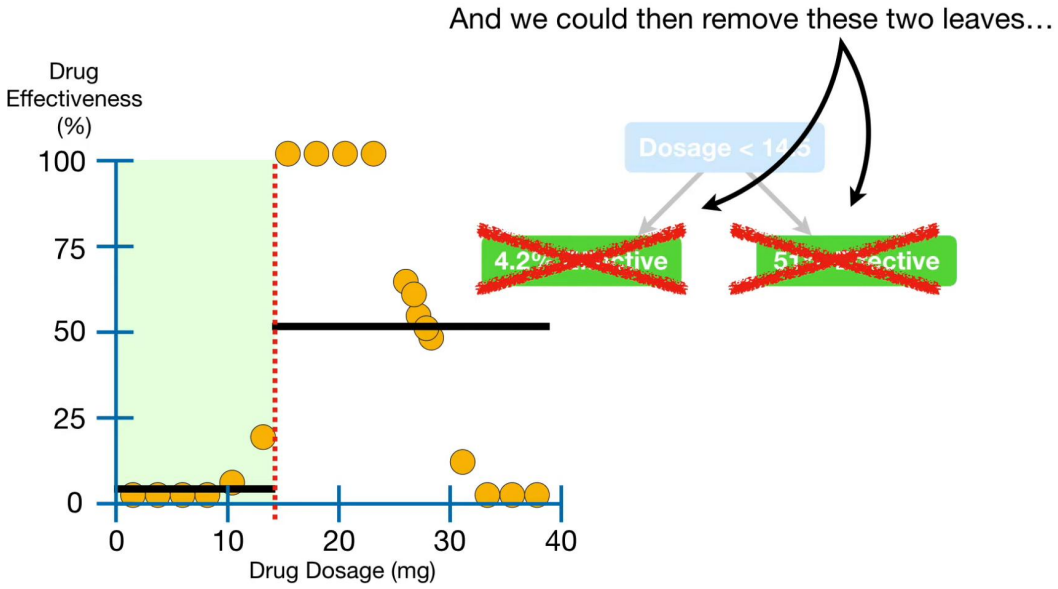
…so that the tree will do a better job with the **Testing Data**.

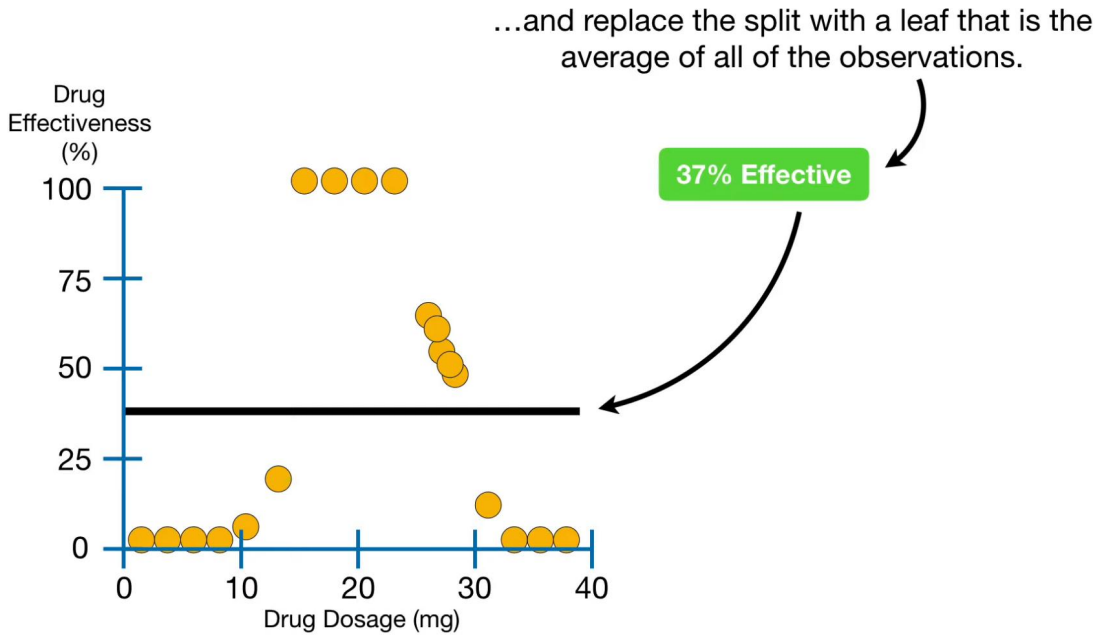





NOTE: If we wanted to prune the tree more, we could remove these two leaves…
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
Dosage < 29
2.5% Effective
73.8% Effective

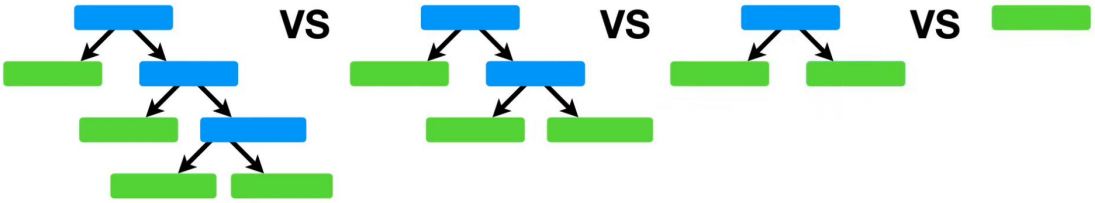

…and replace the split with a leaf that is the average of a larger number of observations.
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
51% Effective


And we could then remove these two leaves…
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
4.2% Effective
51% Effective


…and replace the split with a leaf that is the average of all of the observations.
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
37% Effective

So the question is:

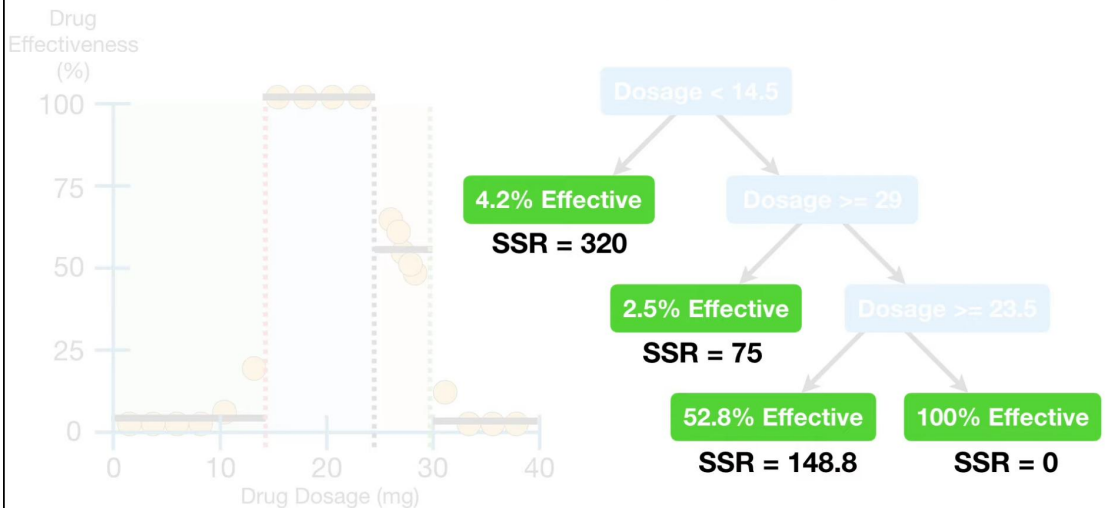
“How do we decide which tree to use?”



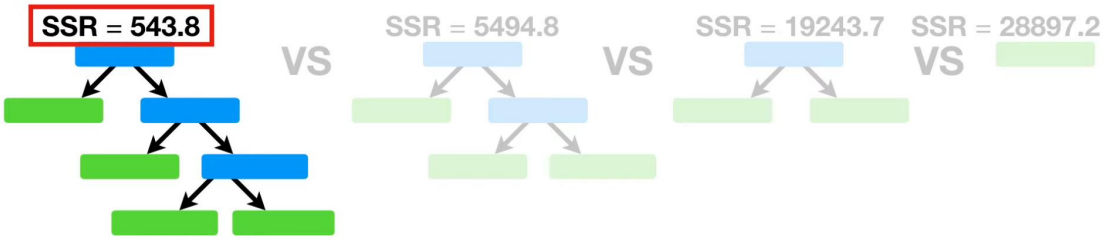
In this **StatQuest**, we will answer that question with **Cost Complexity Pruning**.

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

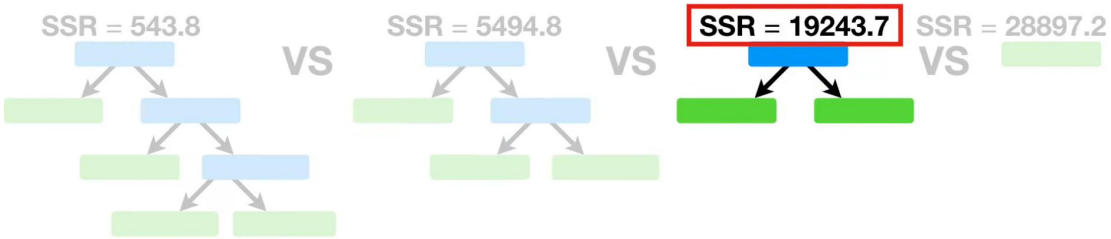
Thus, the total **Sum of Squared Residuals** (SSR) for the whole tree is 320 + 75 + 148.8 + 0 = **543.8**



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.

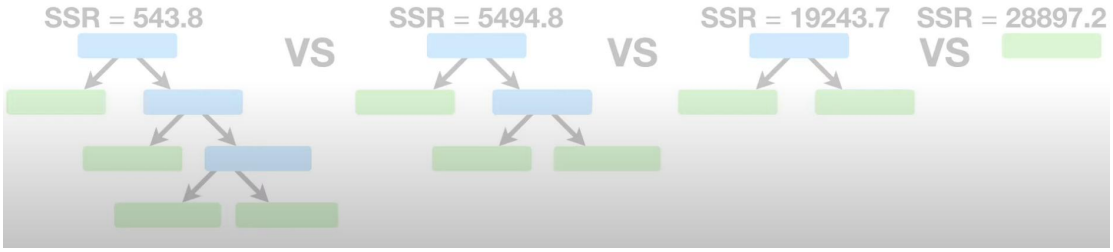
So how do we compare these trees?

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

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

…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***

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***

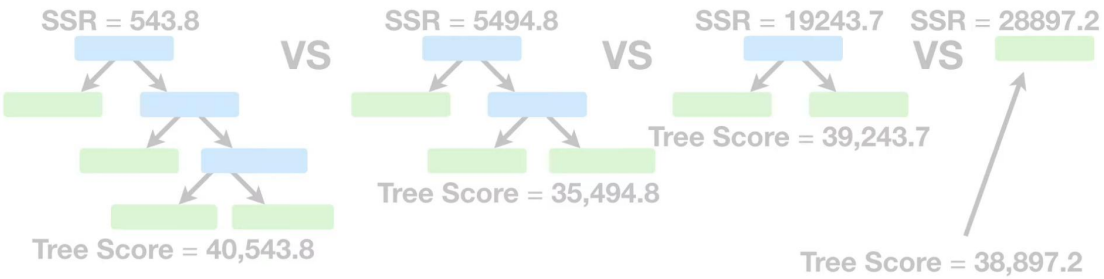
For now, let's let $\alpha = 10{,}000$.

Tree Score = SSR + 10,000 × *T*

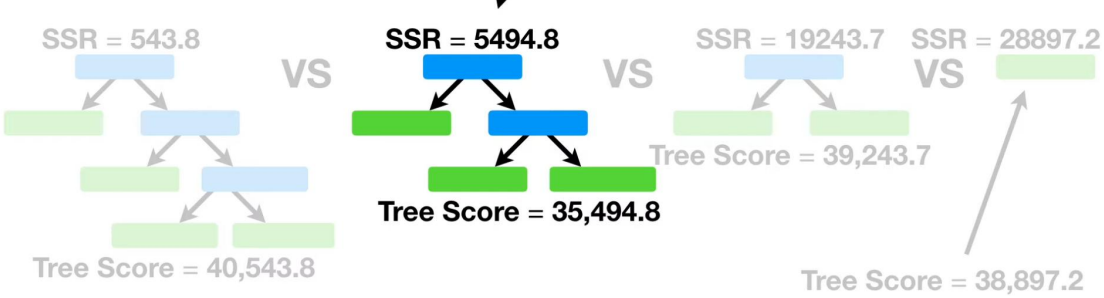
Now let's calculate the **Tree Score** for each tree.

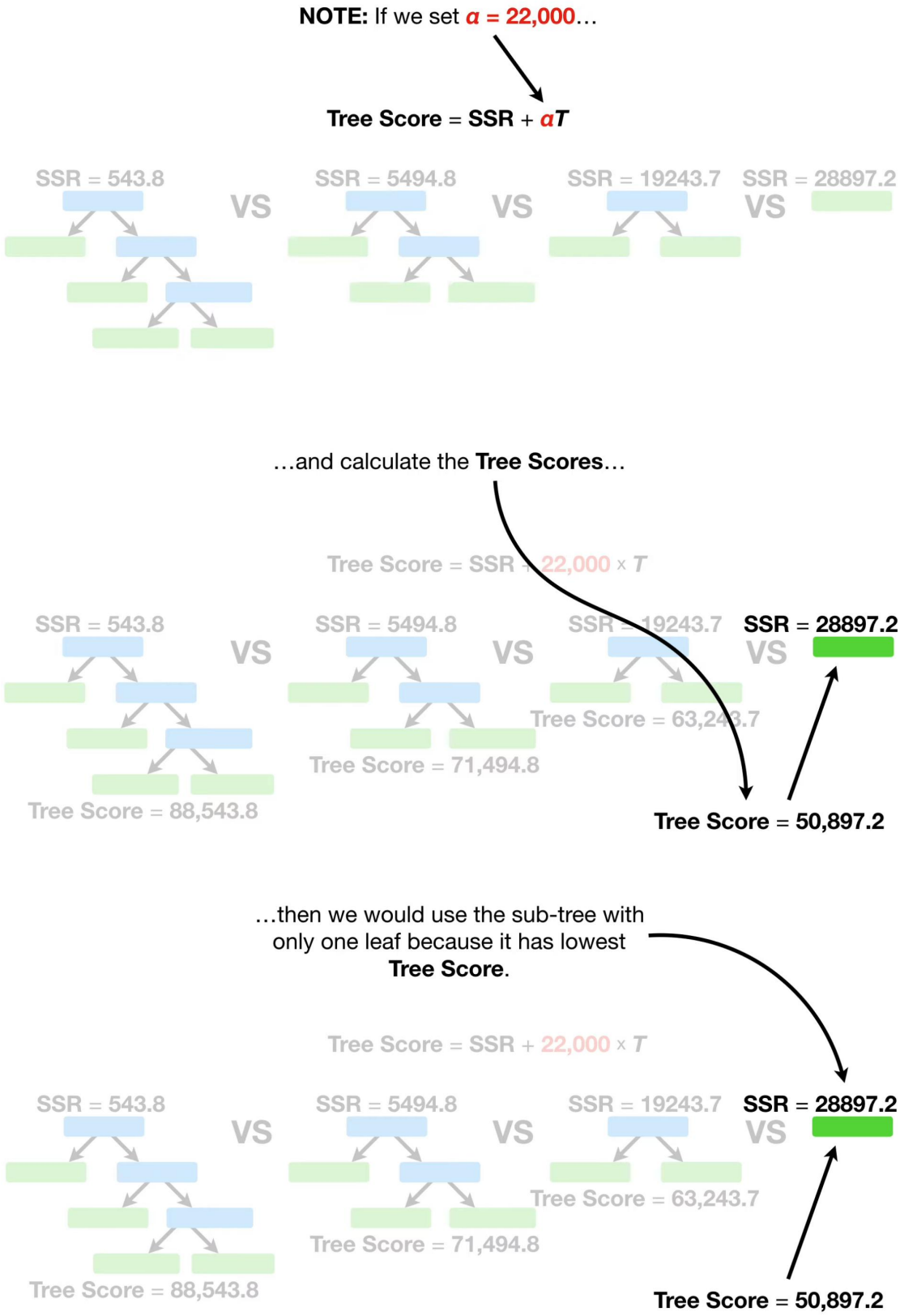
Thus, the the more leaves, the larger the penalty.

Tree Score = SSR + 10,000 × *T*



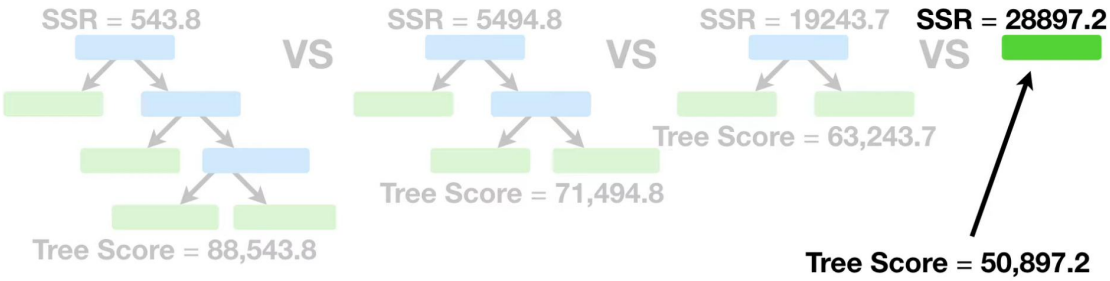
…we pick this sub-tree because it has the lowest **Tree Score**.




NOTE: If we set α = 22,000…
Tree Score = SSR + αT
SSR = 543.8
VS
SSR = 5494.8
VS
SSR = 19243.7
VS
SSR = 28897.2
…and calculate the Tree Scores…
Tree Score = SSR + 22,000 x T
SSR = 543.8
VS
SSR = 5494.8
VS
SSR = 19243.7
VS
SSR = 28897.2
Tree Score = 63,243.7
Tree Score = 71,494.8
Tree Score = 88,543.8
Tree Score = 50,897.2
…then we would use the sub-tree with only one leaf because it has lowest Tree Score.
Tree Score = SSR + 22,000 x T
SSR = 543.8
VS
SSR = 5494.8
VS
SSR = 19243.7
VS
SSR = 28897.2
Tree Score = 63,243.7
Tree Score = 71,494.8
Tree Score = 88,543.8
Tree Score = 50,897.2

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

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

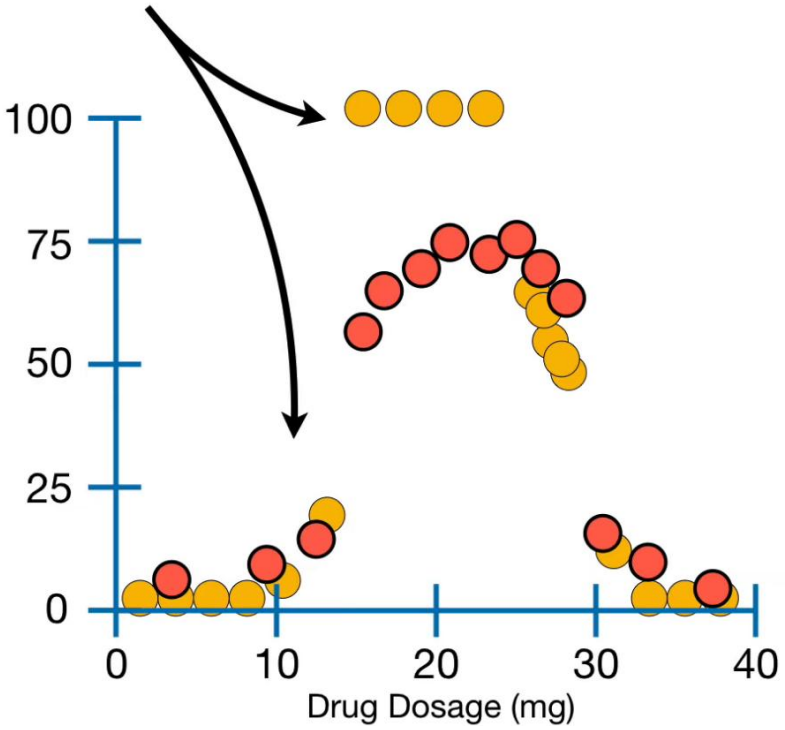


So let's talk about how build a pruned regression tree…

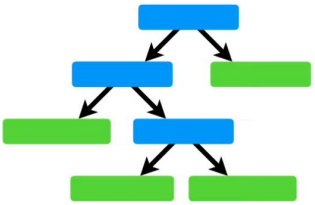
…and how to find the best value for α.

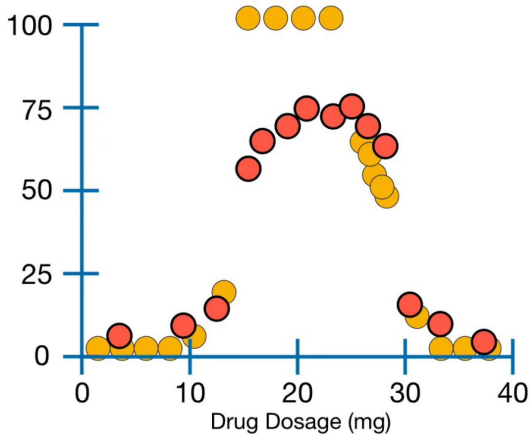
Tree Score = SSR + αT

First, using ***all*** of the data…

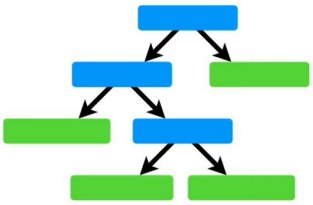


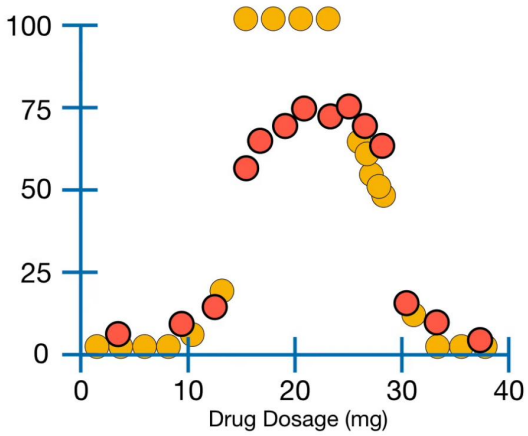
…build a full sized **Regression Tree**.





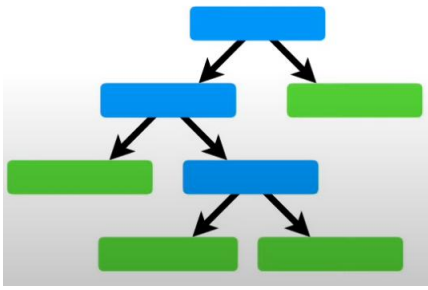
NOTE: This full sized tree is different than before because it was fit to ***all*** of the data, not just the **Training Data**.





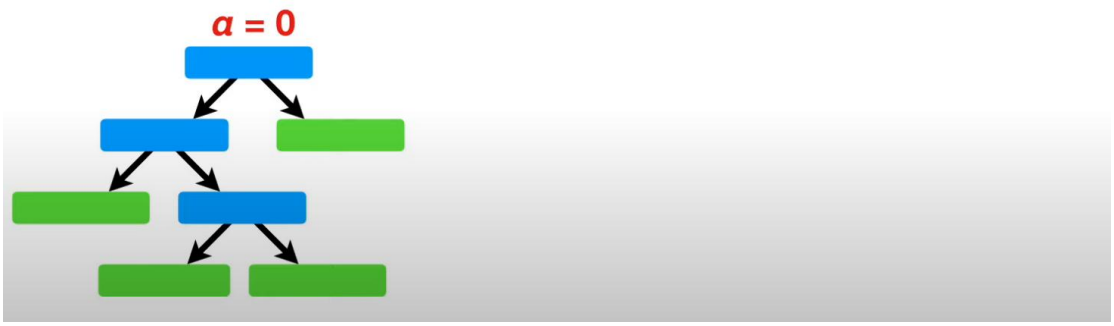
ALSO NOTE: This full sized tree has the lowest **Tree Score** when $\alpha = 0$.

Tree Score = SSR + αT



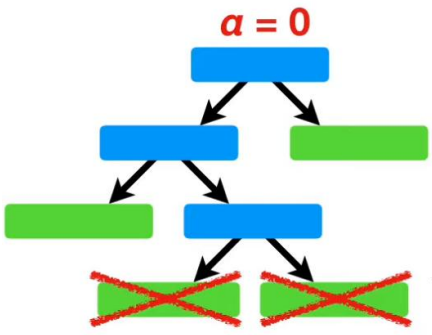
So let's put $\alpha = 0$ here, to remind us that this tree has the lowest **Tree Score** when $\alpha = 0$.

Tree Score = SSR



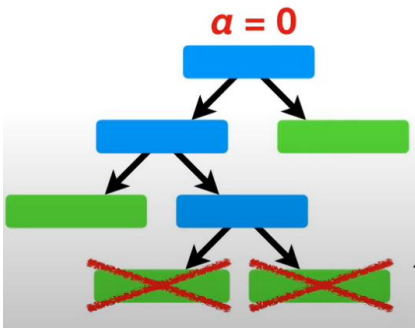
Now we increase α until pruning leaves will give us a lower **Tree Score**.

Tree Score = SSR + αT



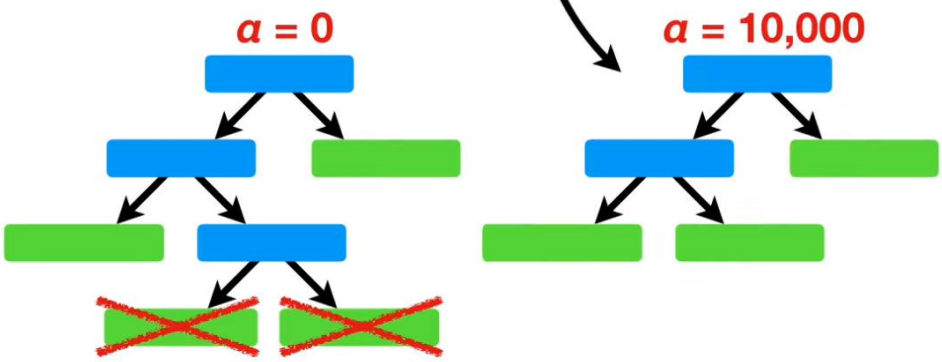
In this case, when $\alpha = 10{,}000$, we'll get a lower **Tree Score** if we remove these leaves…

Tree Score = **SSR** + αT



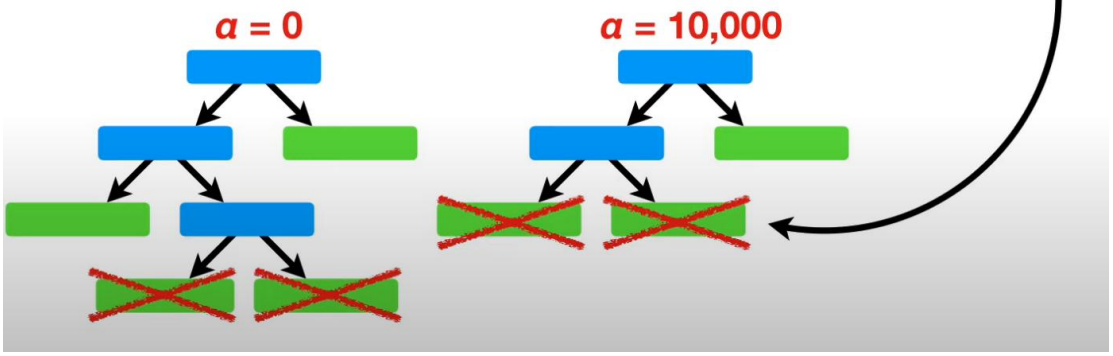
…and use this sub-tree.

Tree Score = **SSR** + αT



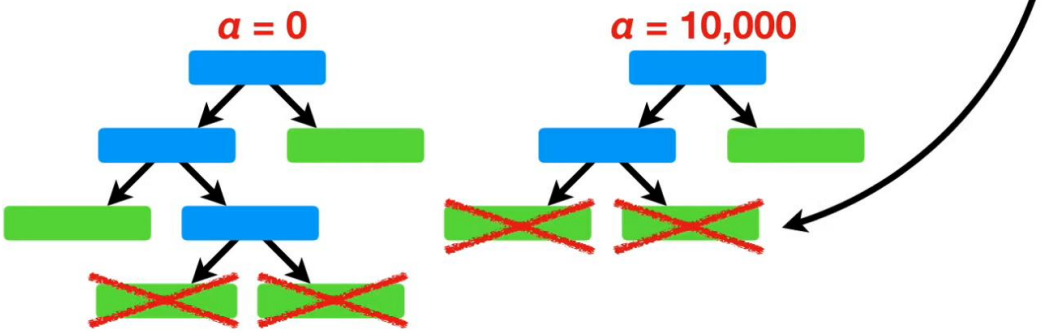
Now we increase α again until pruning leaves will give us a lower **Tree Score**.

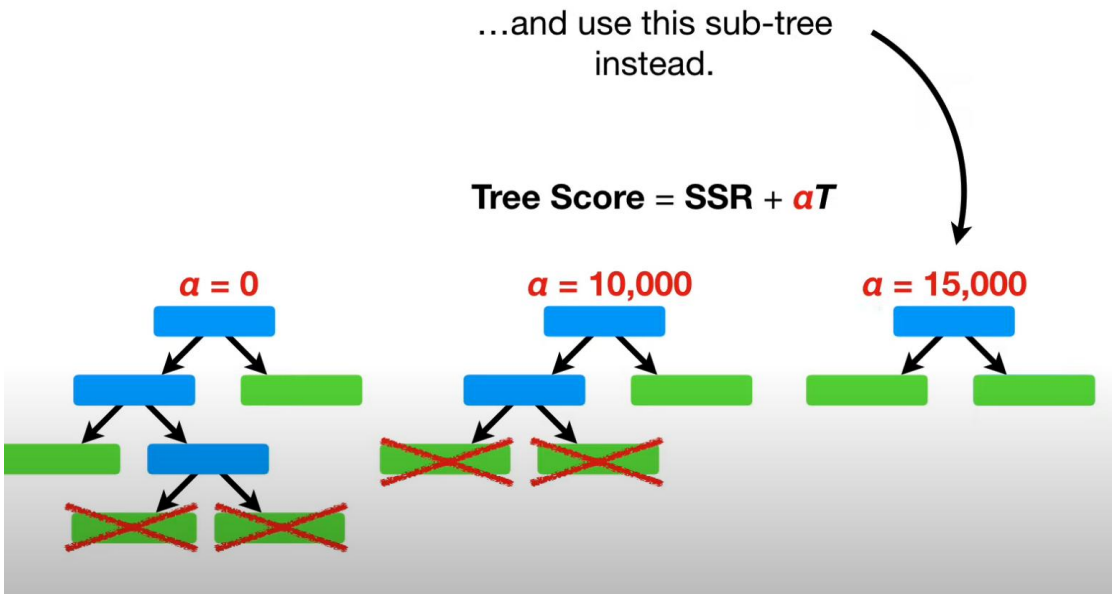
Tree Score = **SSR** + αT

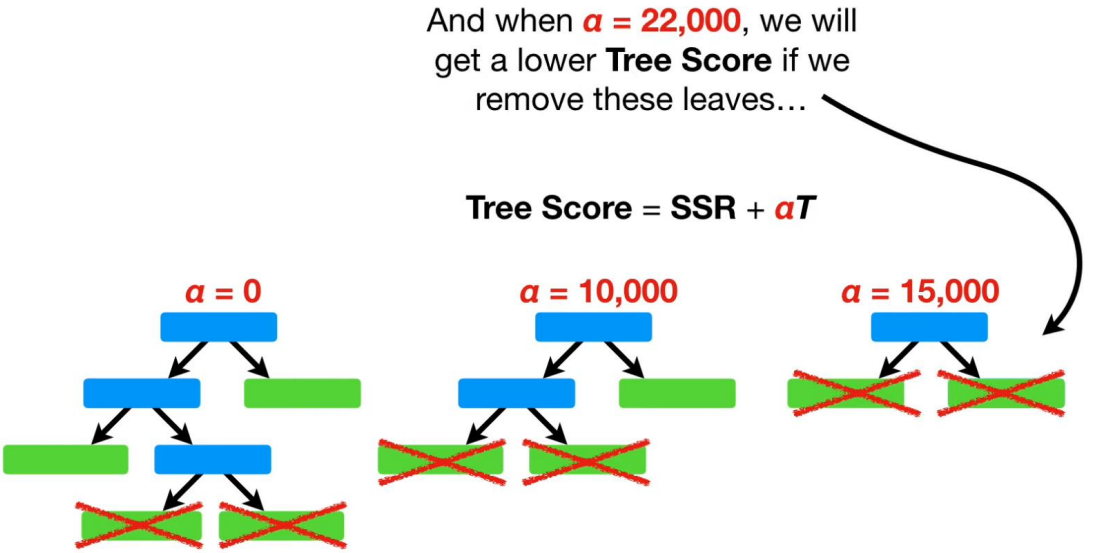


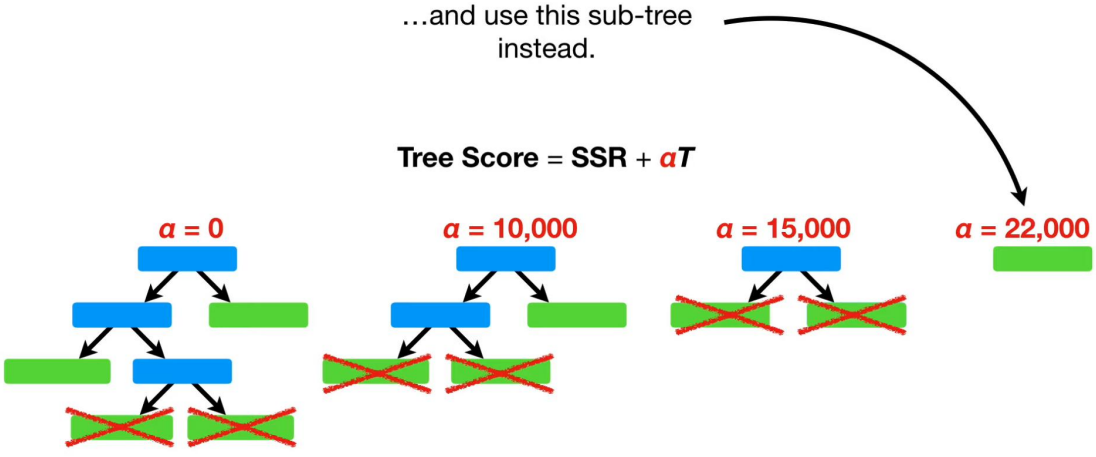
In this case when $\alpha = 15{,}000$, we will get a lower **Tree Score** if we remove these leaves…

Tree Score = **SSR** + αT



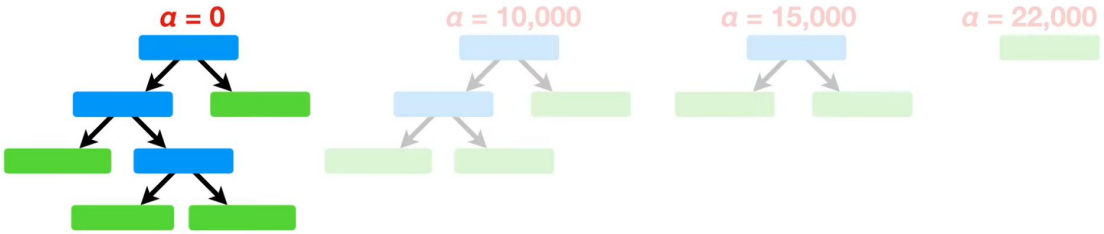

…and use this sub-tree instead.
Tree Score = SSR + αT
α = 0
α = 10,000
α = 15,000


And when α = 22,000, we will get a lower Tree Score if we remove these leaves…
Tree Score = SSR + αT
α = 0
α = 10,000
α = 15,000

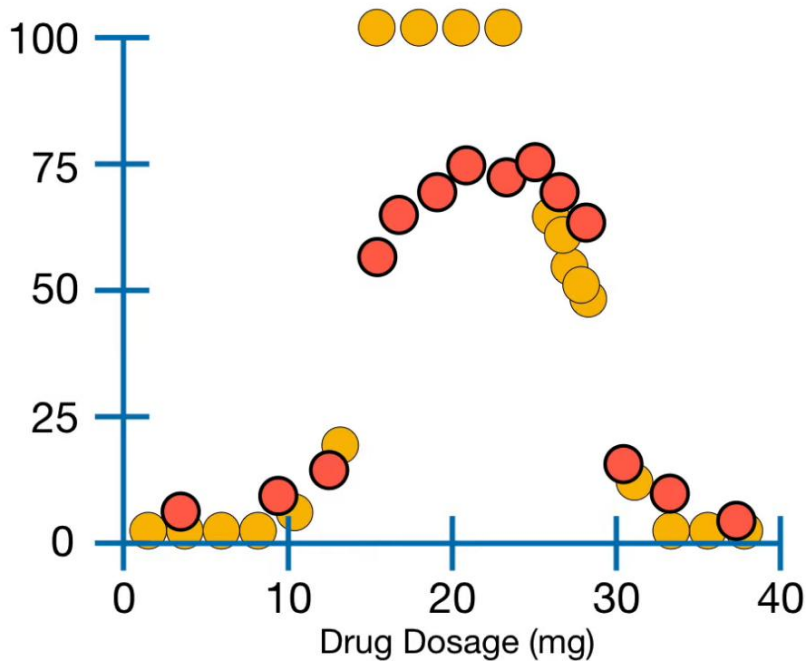

…and use this sub-tree instead.
Tree Score = SSR + αT
α = 0
α = 10,000
α = 15,000
α = 22,000

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

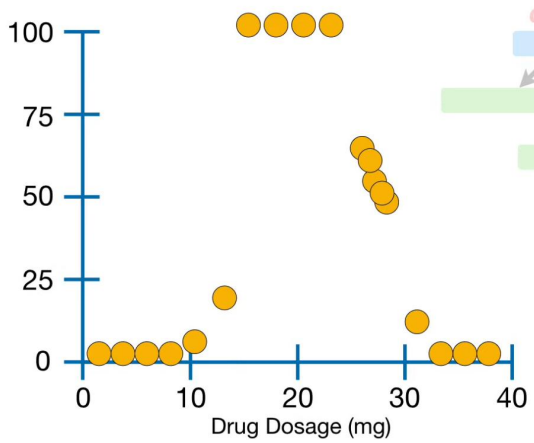
Tree Score = SSR + αT



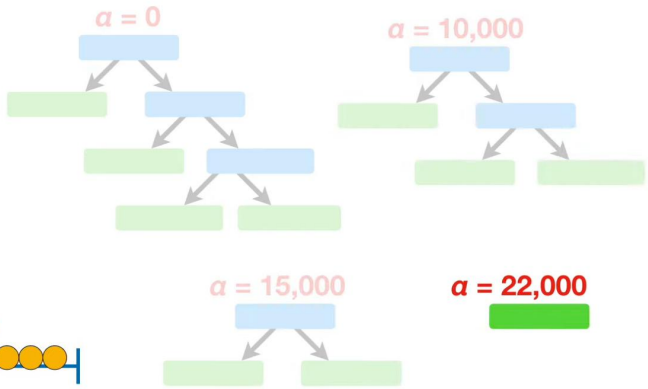
Now go back to the full dataset…



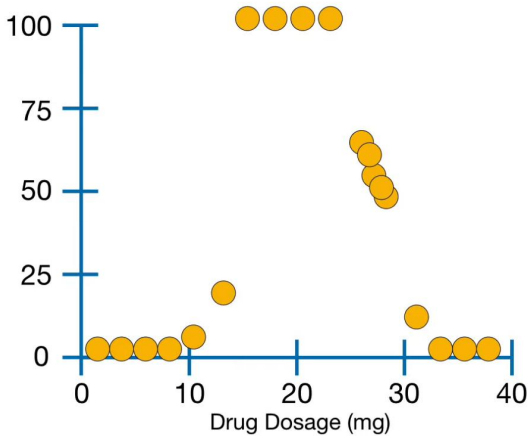
And just using the **Training Data**…



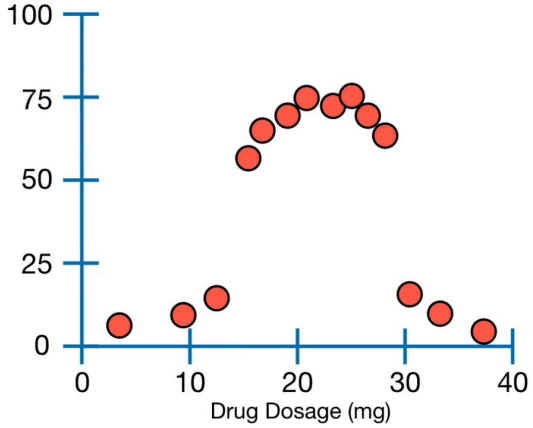
…use the α values we found before to build a full tree and a sequence of sub-trees that minimize the **Tree Score**.



…and divide it into **Training**…

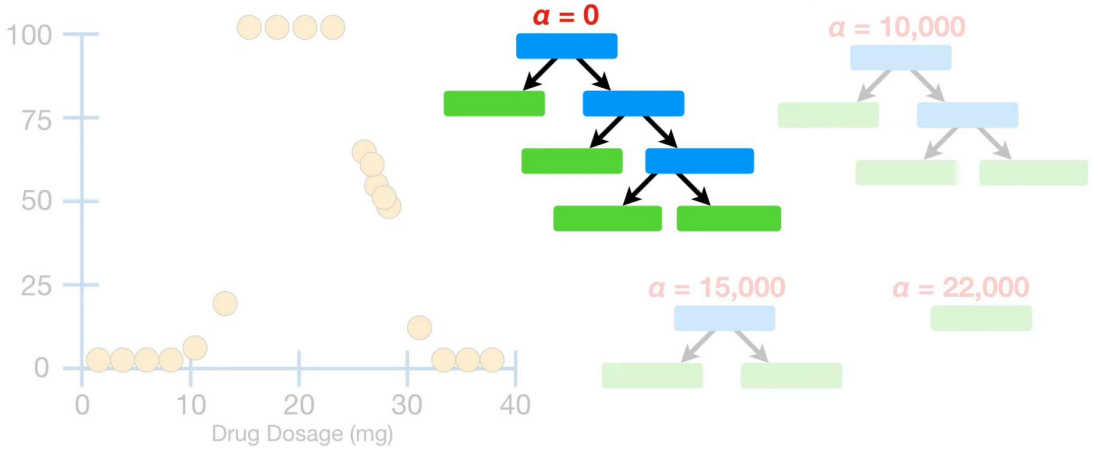


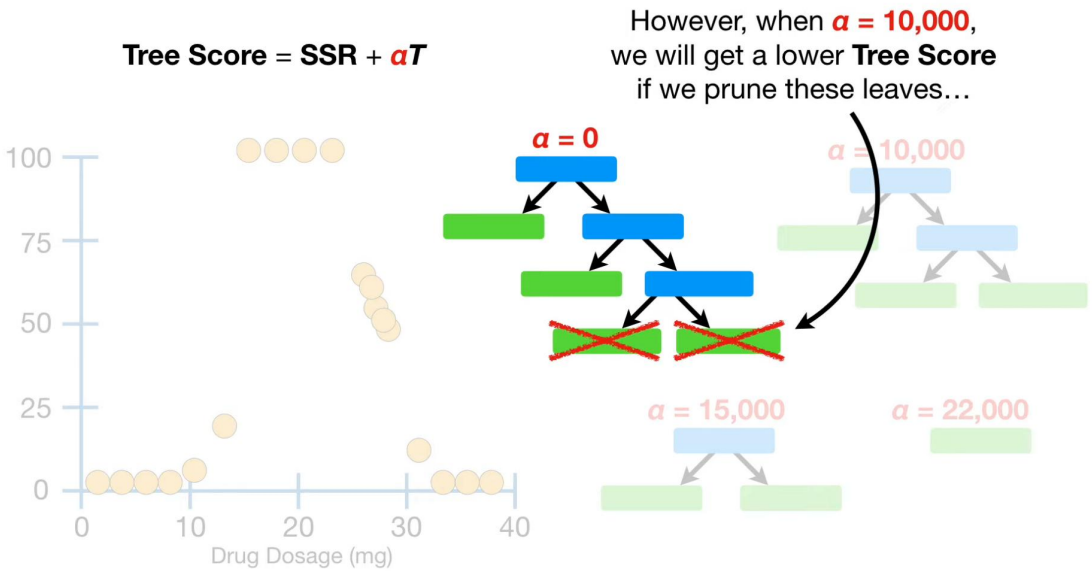
…and **Testing Datasets**.

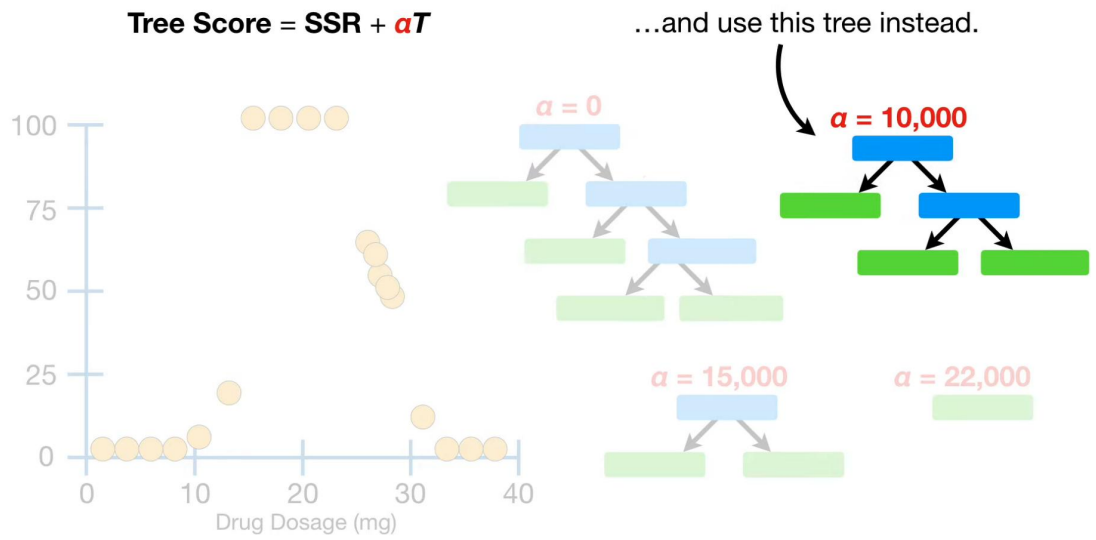


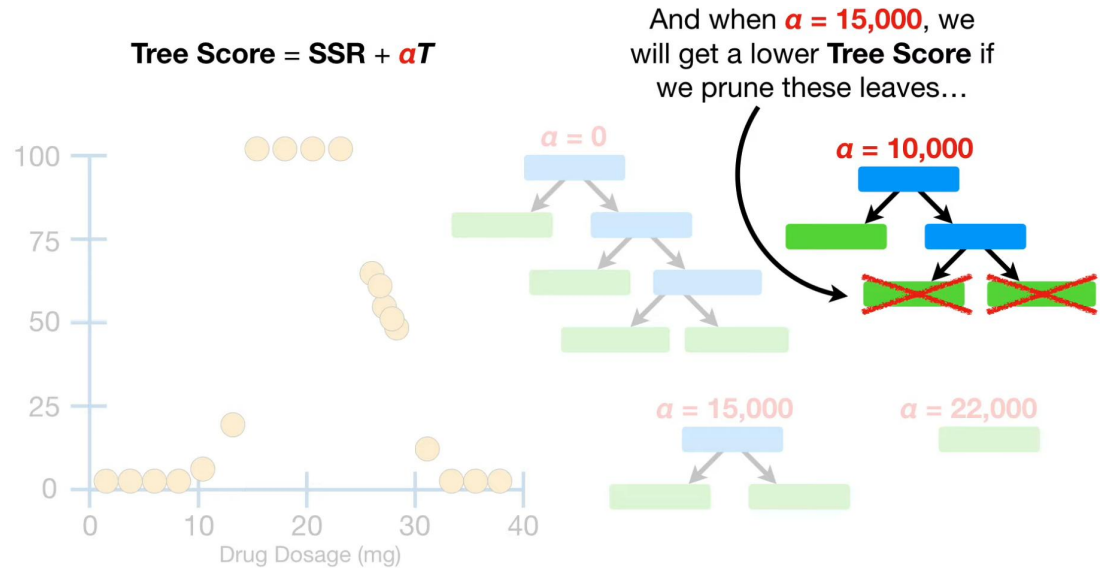
Tree Score = **SSR** + αT

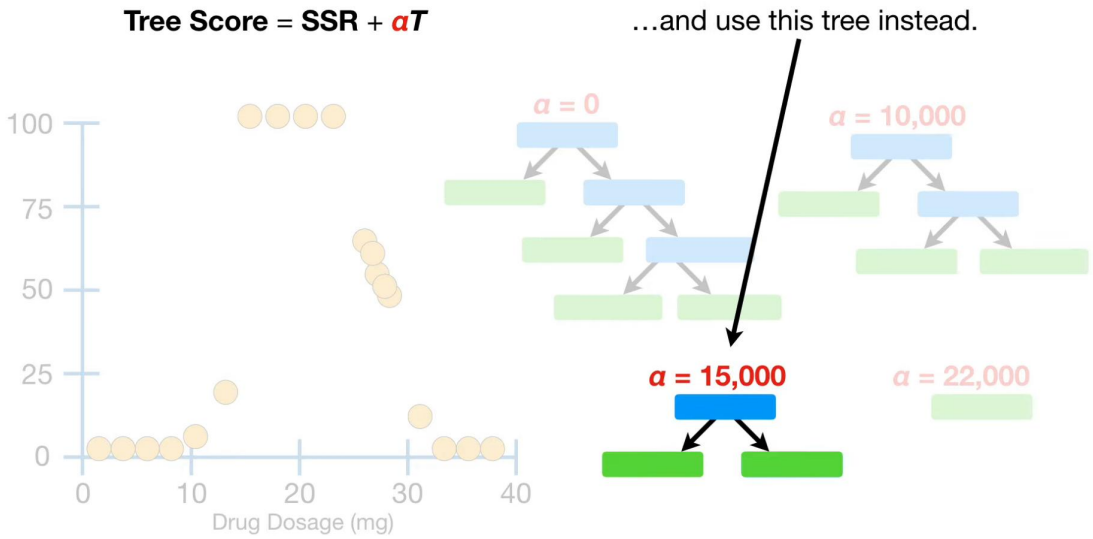
In other words, when $\alpha = 0$, we build a full sized tree, since it will have the lowest **Tree Score**.

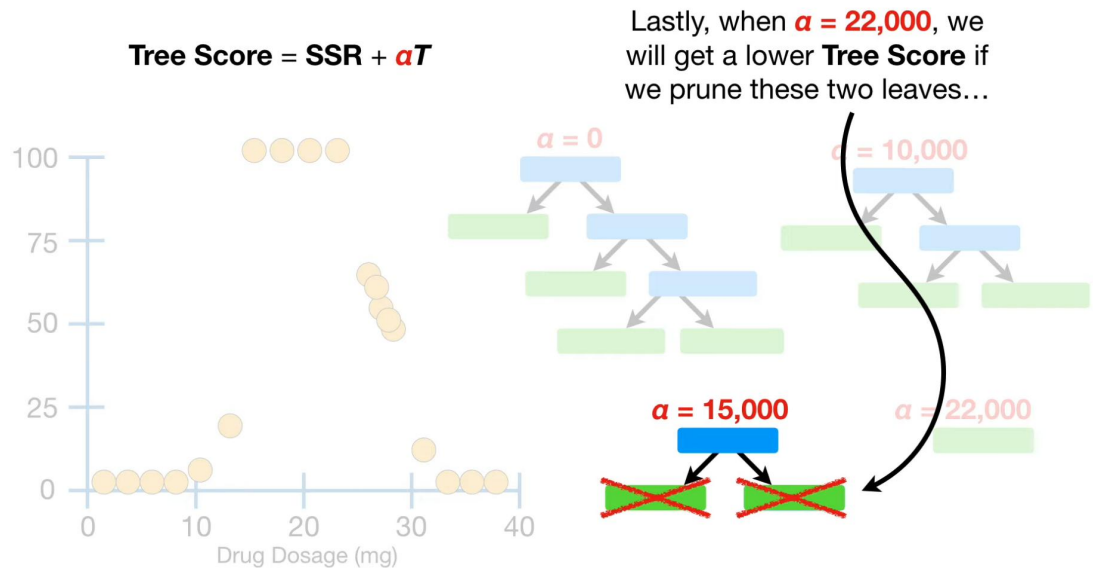


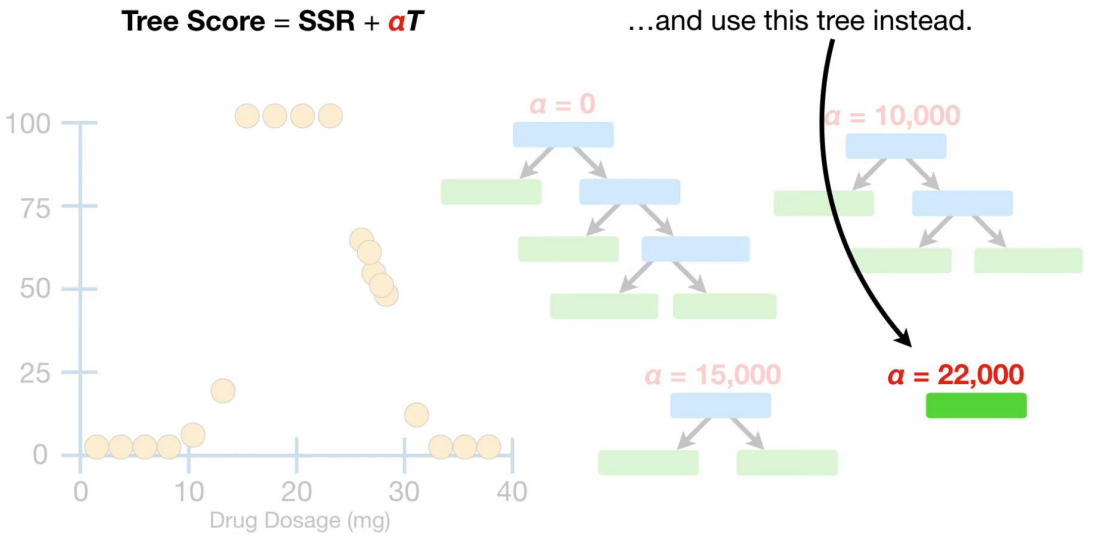

Tree Score = SSR + αT
However, when α = 10,000, we will get a lower Tree Score if we prune these leaves…
α = 0
α = 10,000
α = 15,000
α = 22,000
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)


Tree Score = SSR + αT
…and use this tree instead.
α = 0
α = 10,000
α = 15,000
α = 22,000
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)

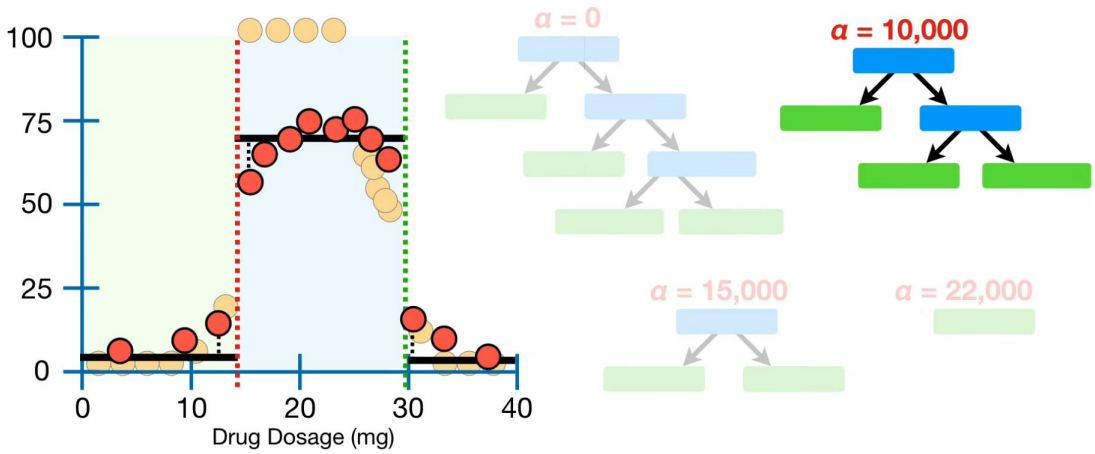

Tree Score = SSR + αT
And when α = 15,000, we will get a lower Tree Score if we prune these leaves…
α = 0
α = 10,000
α = 15,000
α = 22,000
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)


Tree Score = SSR + αT
…and use this tree instead.
α = 0
α = 10,000
α = 15,000
α = 22,000
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)

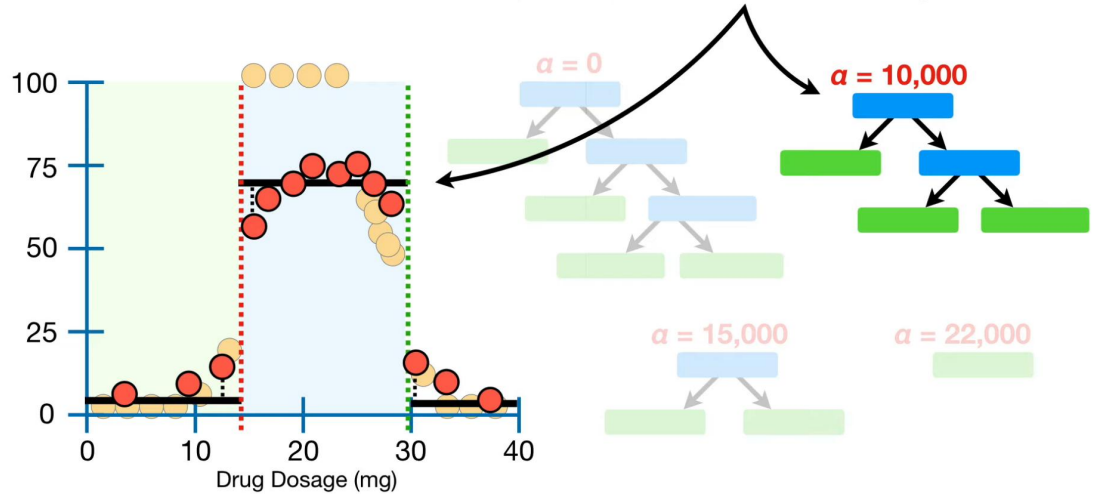

Tree Score = SSR + αT
Lastly, when α = 22,000, we will get a lower Tree Score if we prune these two leaves…
α = 0
α = 10,000
α = 15,000
α = 22,000
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)


Tree Score = SSR + αT
…and use this tree instead.
α = 0
α = 10,000
α = 15,000
α = 22,000
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)

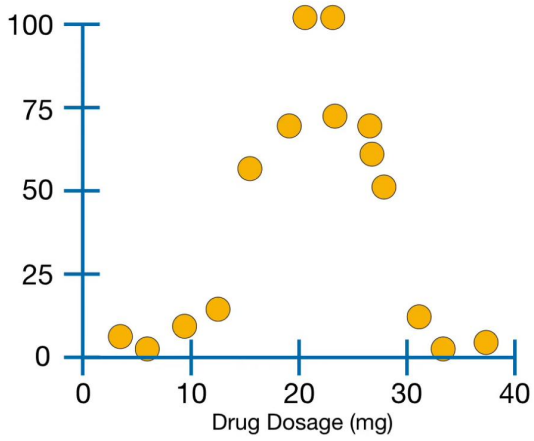
Now calculate the **Sum of Squared Residuals** for each new tree using only the **Testing Data**.



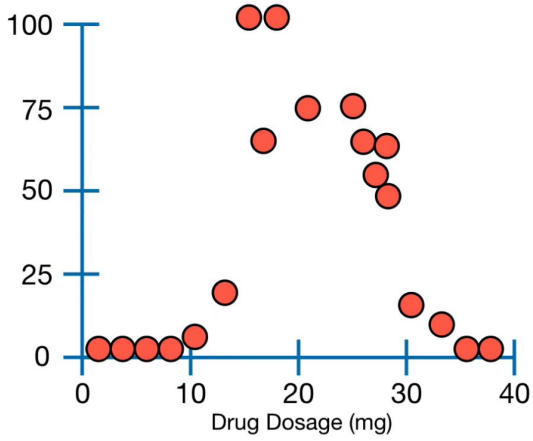
In this case, the tree with **α = 10,000** had the smallest **Sum of Squared Residuals** for the **Testing Data**.



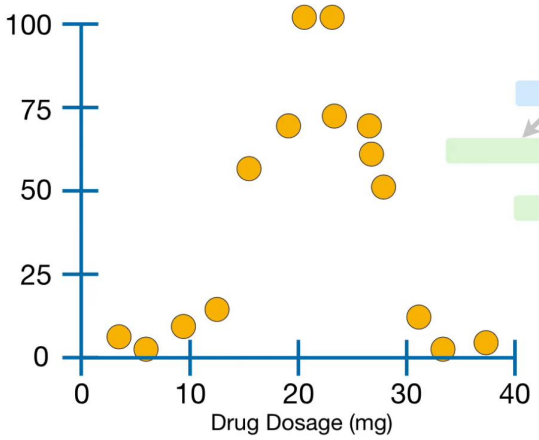
Now we go back and create new **Training Data**…



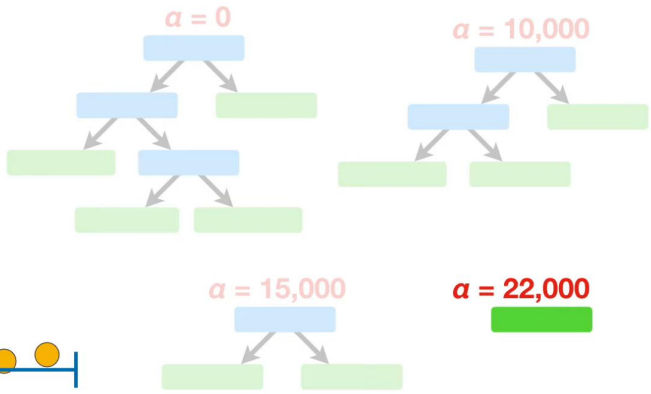
…and new **Testing Data**.



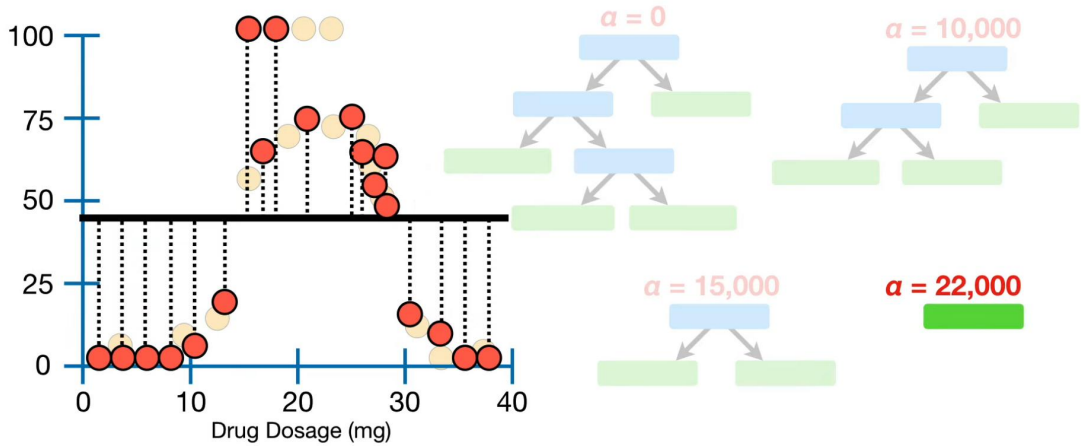
And just using the new **Training Data**…



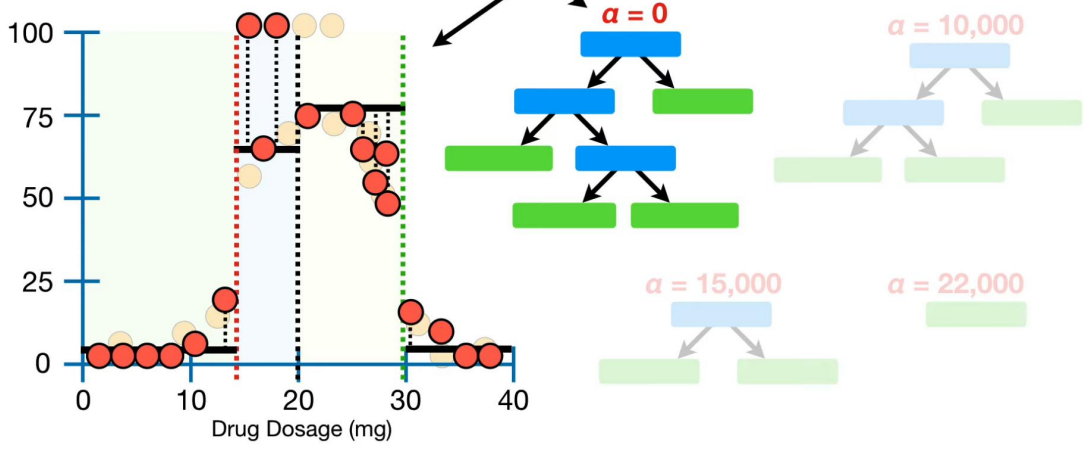
…build a new sequence of trees, from full sized to a leaf, using the α values we found before.



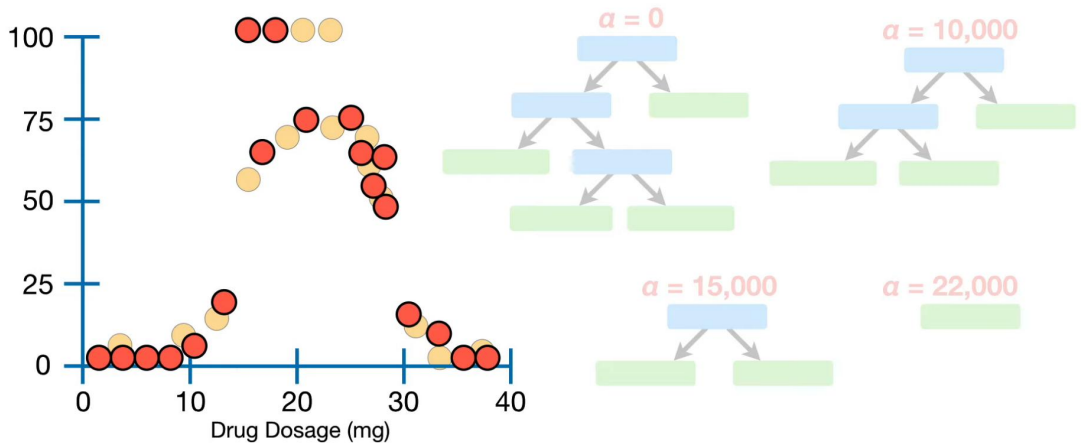
Then we calculate the **Sum of Squared Residuals** using the new **Testing Data**.



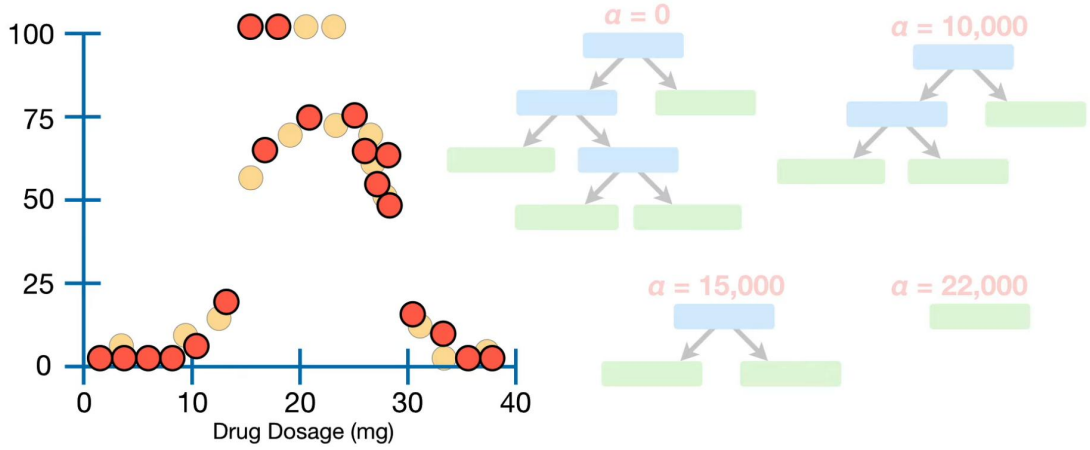
This time, the tree with $\alpha = 0$ had the lowest **Sum of Squared Residuals**.



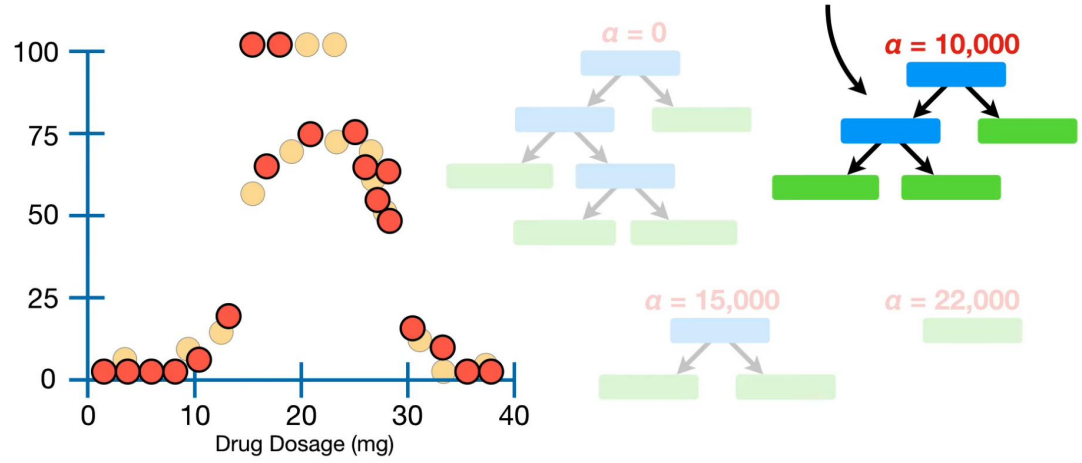
Now we just keep repeating until we have done **10-Fold Cross Validation**…



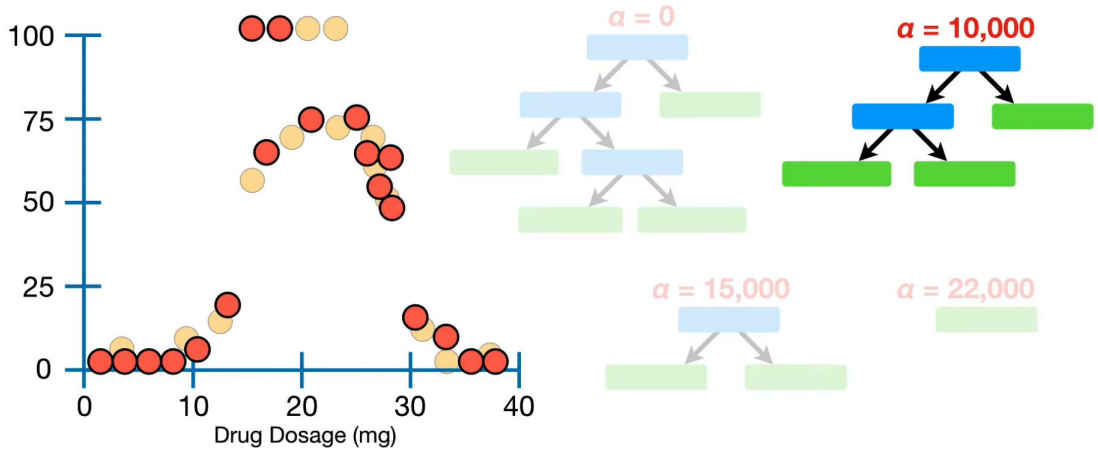
…and the value for ***α*** that, on average, gave us the lowest **Sum of Squared Residuals** with the **Testing Data**, is the final value for ***α***.



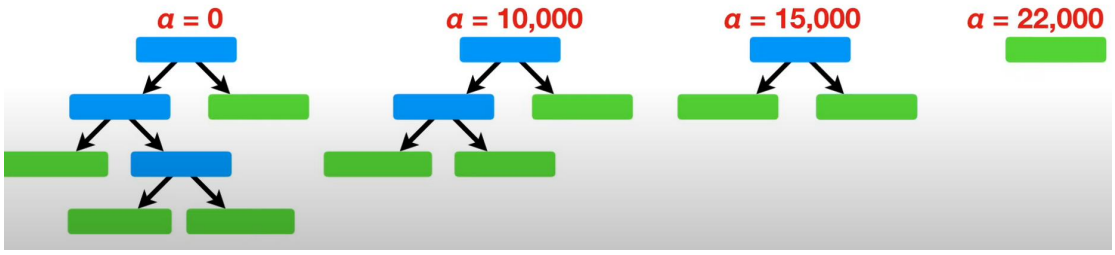
In this case, the optimal trees built with ***α* = 10,000** had, on average, the lowest **Sum of Squared Residuals**…



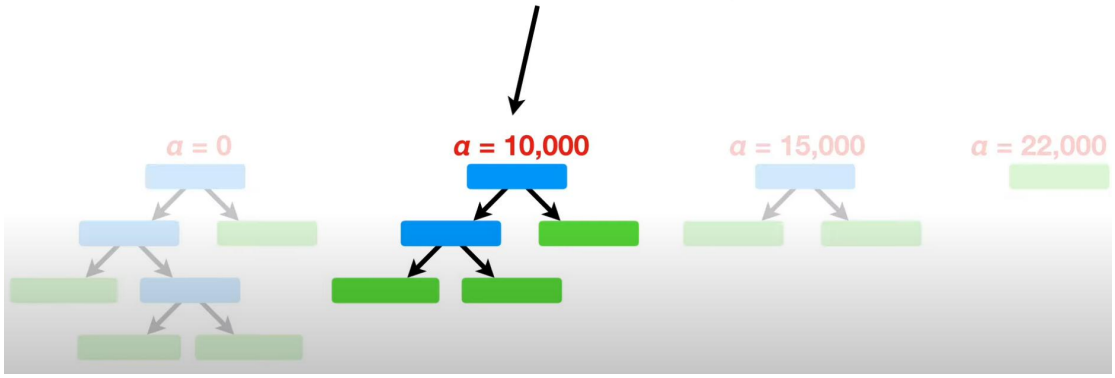
…so α = 10,000 is our final value.

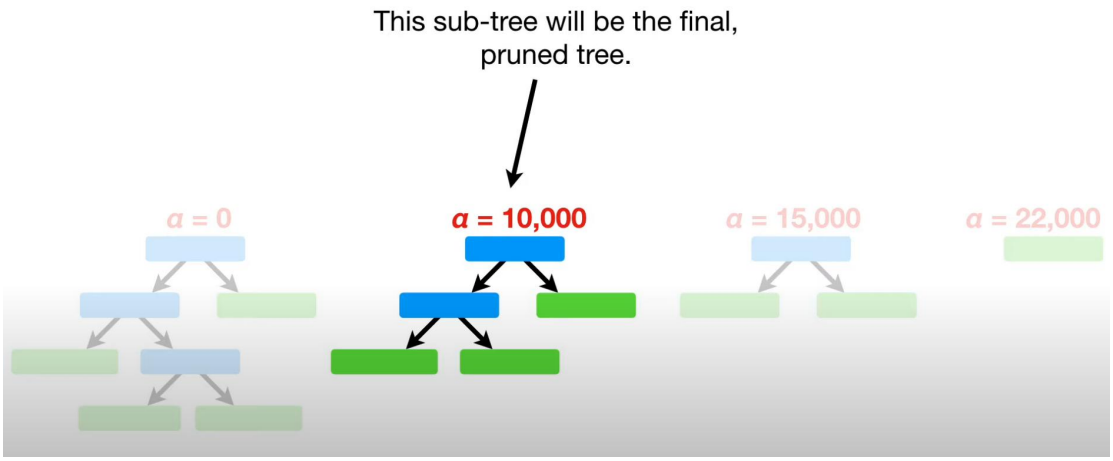


Lastly, we go back to the original trees and sub-trees made from the full data.



…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.
α = 0
α = 10,000
α = 15,000
α = 22,000