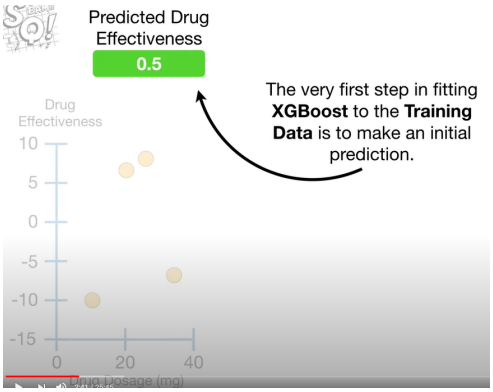
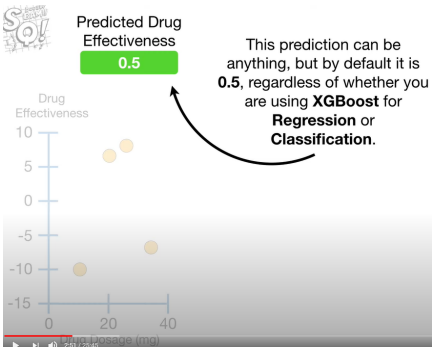
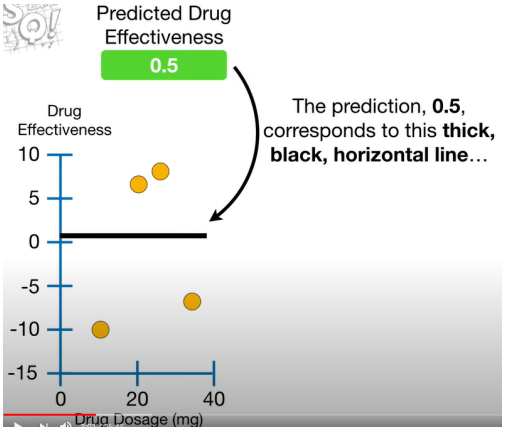
XGBoost for Regression

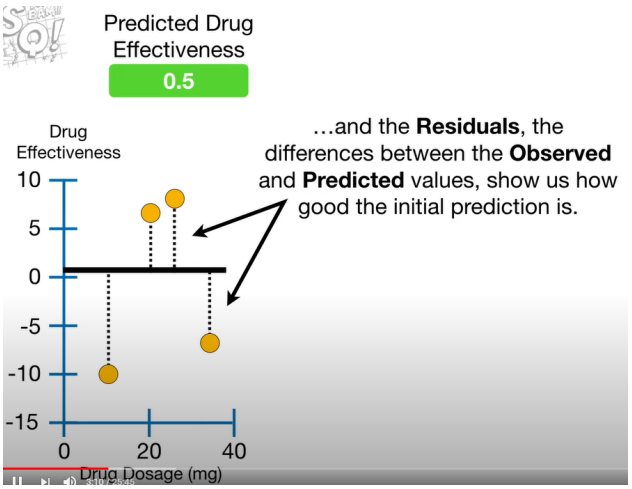
Step 1:做最初的预测



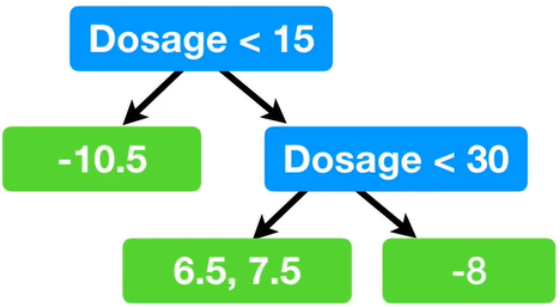


一般使用XGB（无论做regression还是分类）的最初预测值都是0.5



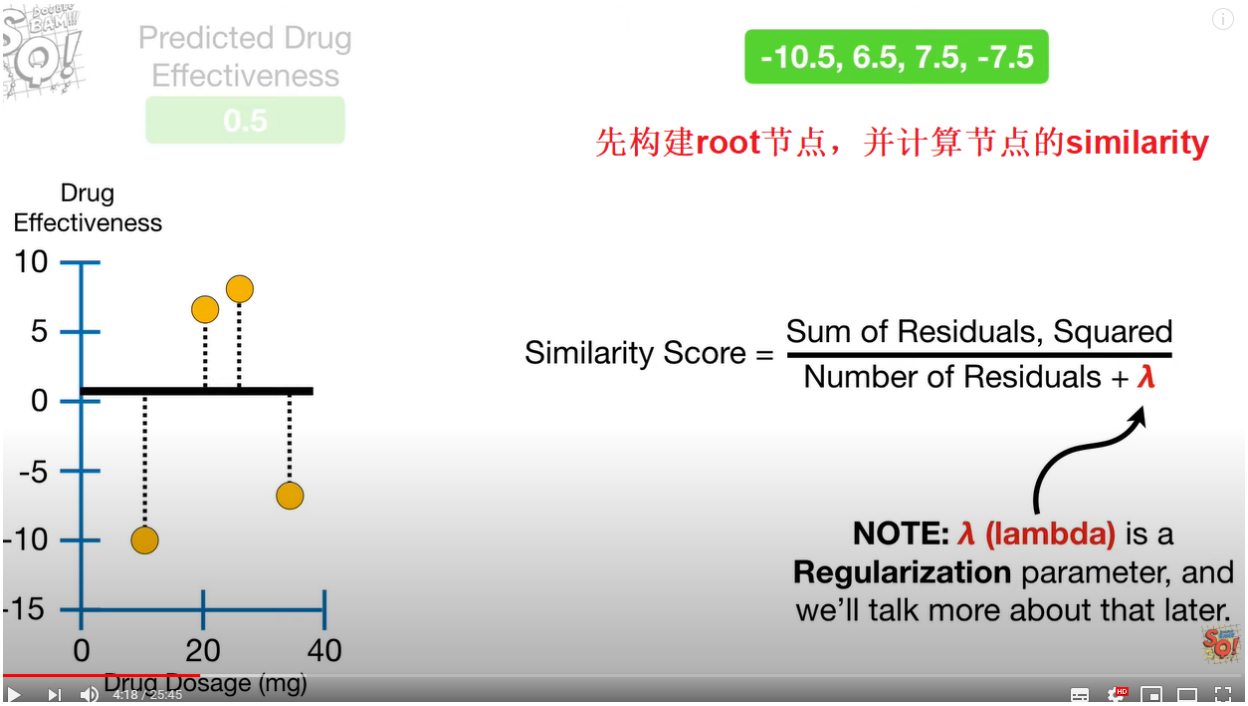


Step 2:和其他Gradient Boost算法一样，构建树结构

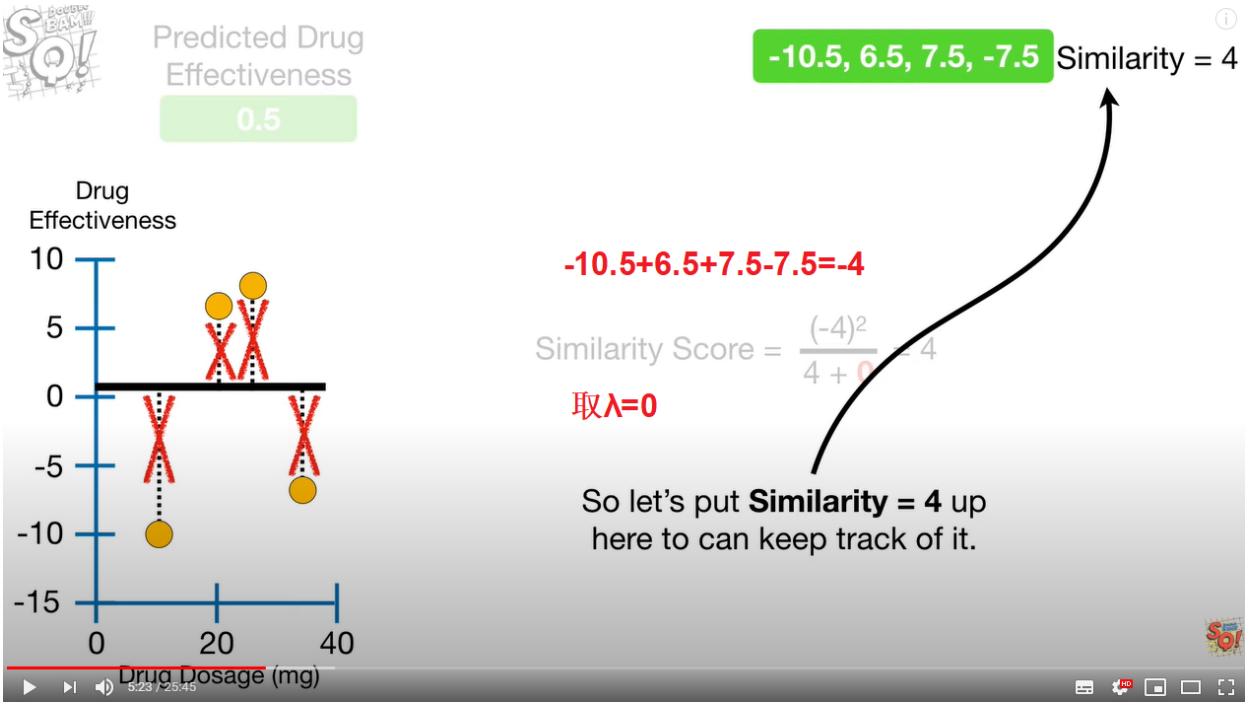


NOTE: There are many ways to build **XGBoost Trees**. This video focuses on the most common way to build them for **Regression**.

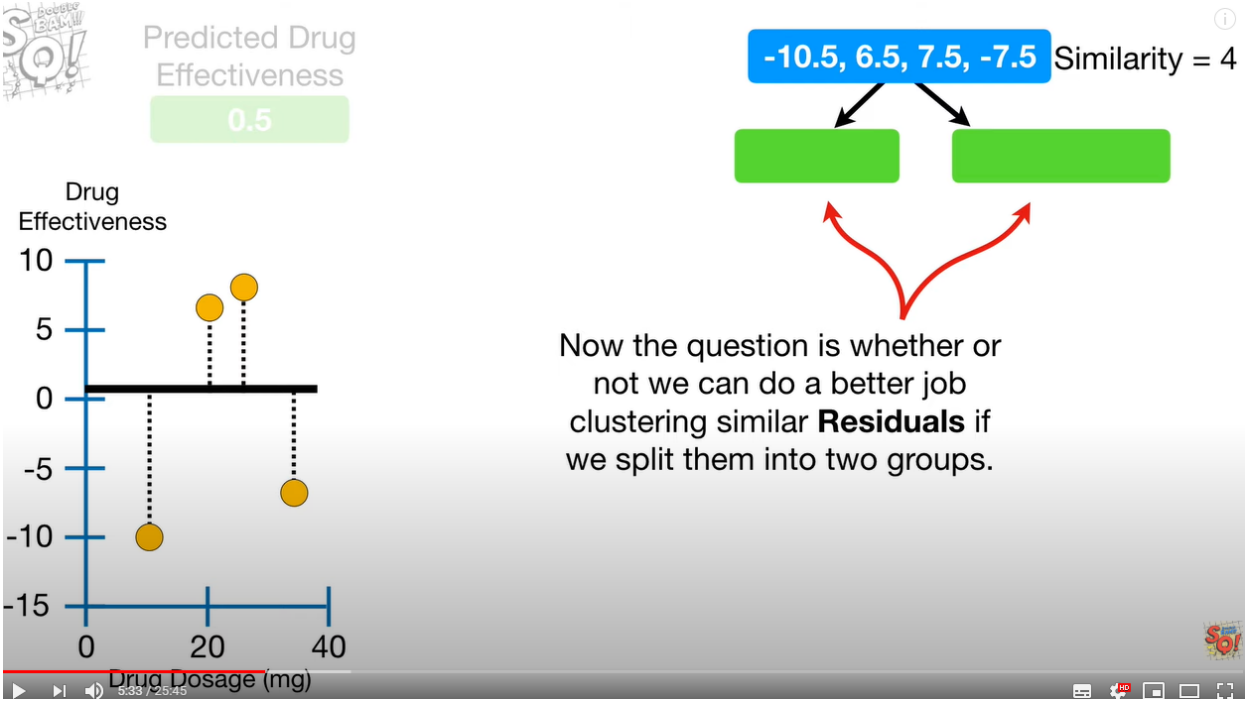
但和Gradient Boost使用现成的regression tree不同，XGB使用特有的树结构



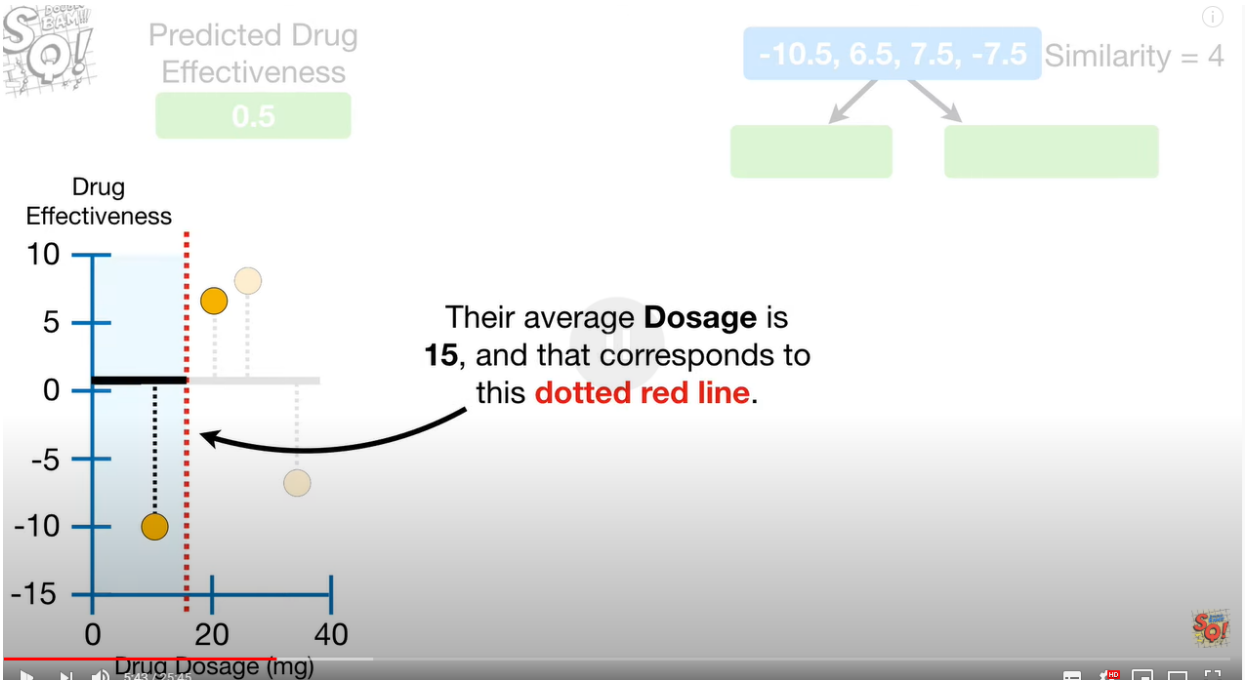
把所有样本的resudial放到Root节点中，计算该节点similarity。

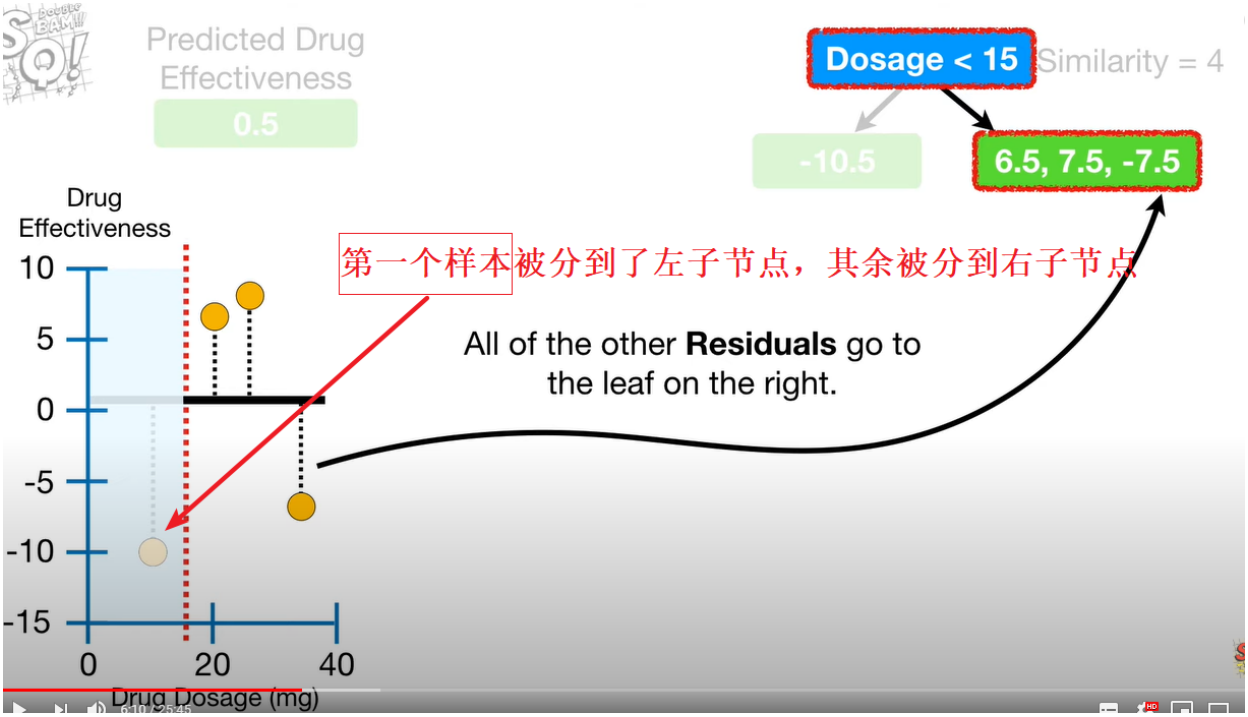


接着将root节点中的residuals再细分，看能不能得到更好的聚类效果（相似的样本分到同一边）。表面上是对residuals进行聚类，实质上是将样本进行聚类分到不同的叶节点。

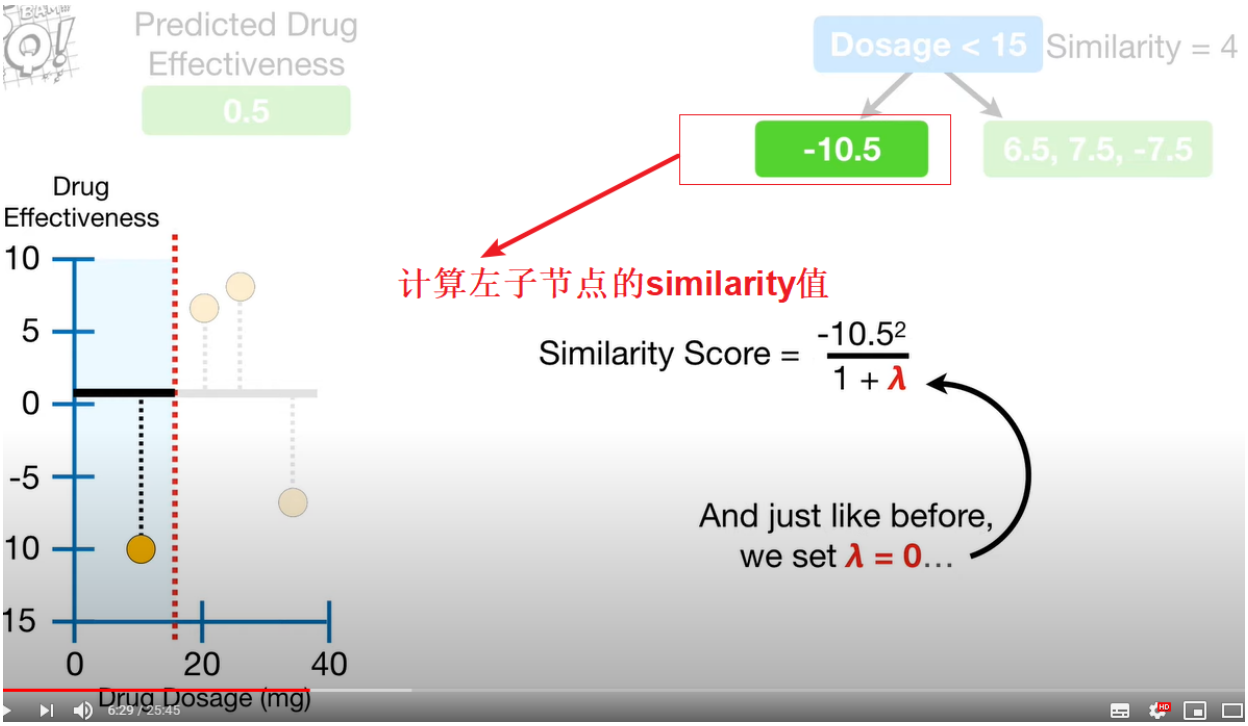


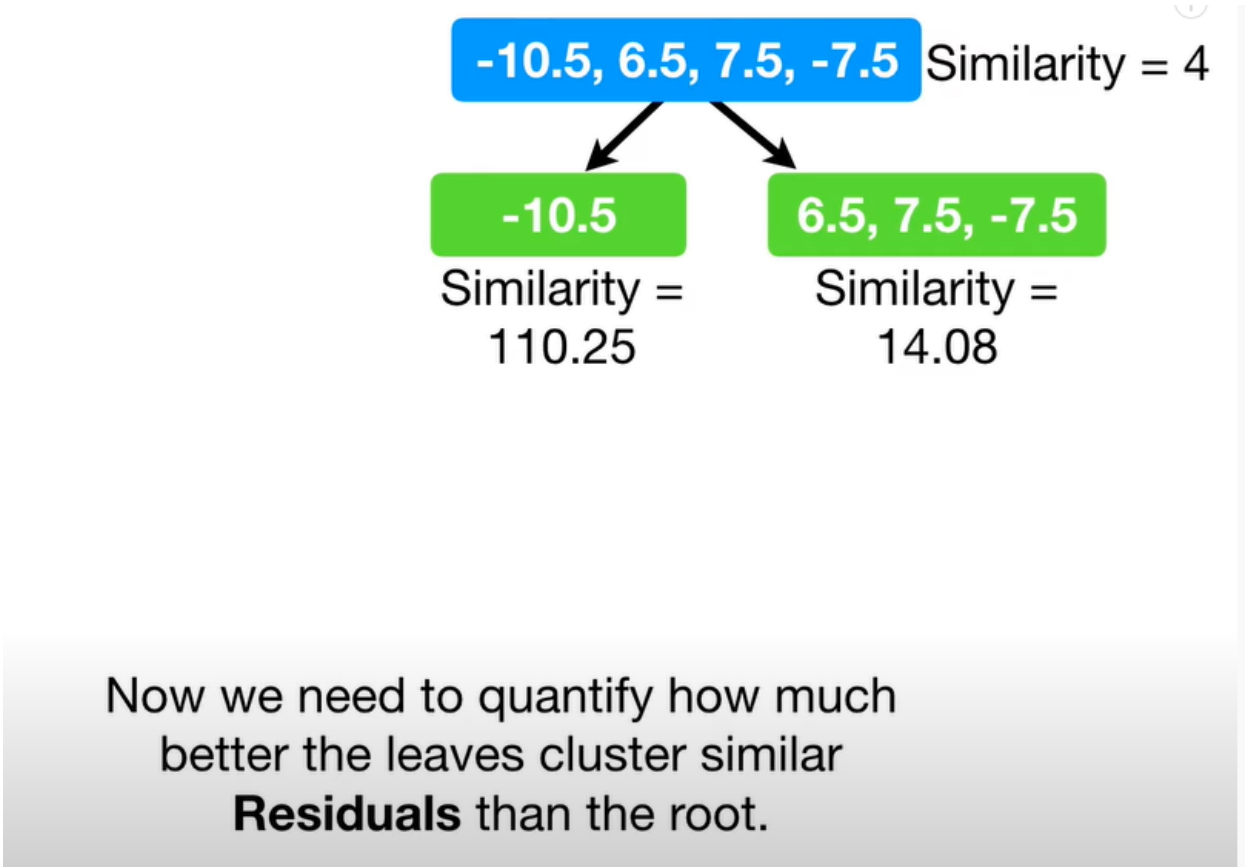
取不同的dosage值尝试将样本进行分离。注意前面一步是计算root节点的similarity值，但还没确定root节点将sample进行分离的标准。





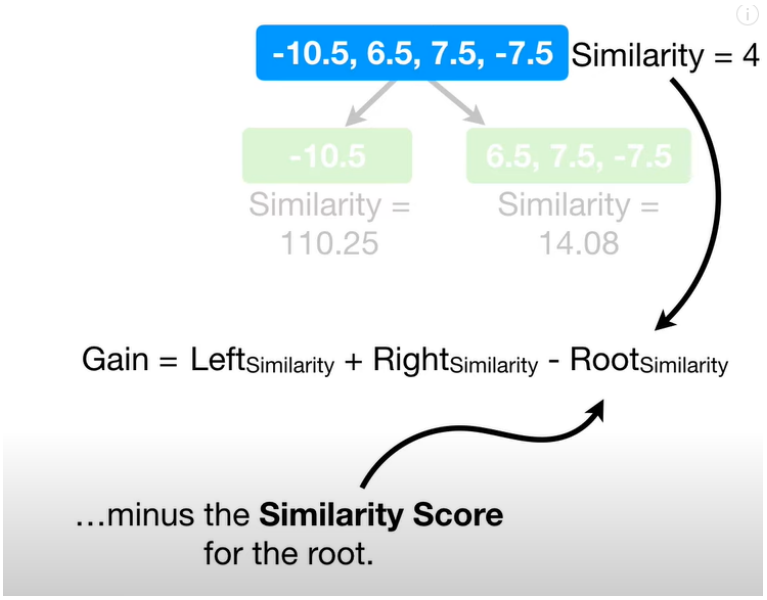
然后使用上面提到的similarity计算公式，计算各个子节点的similairty。

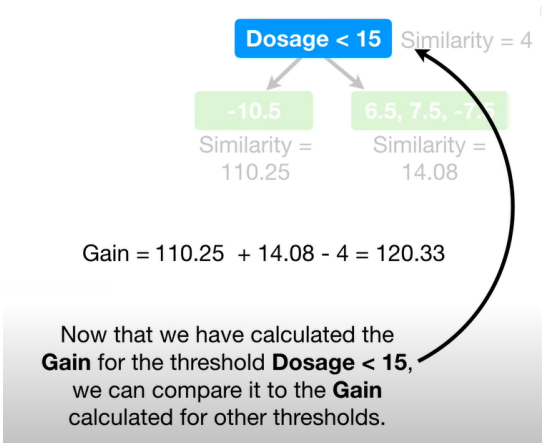




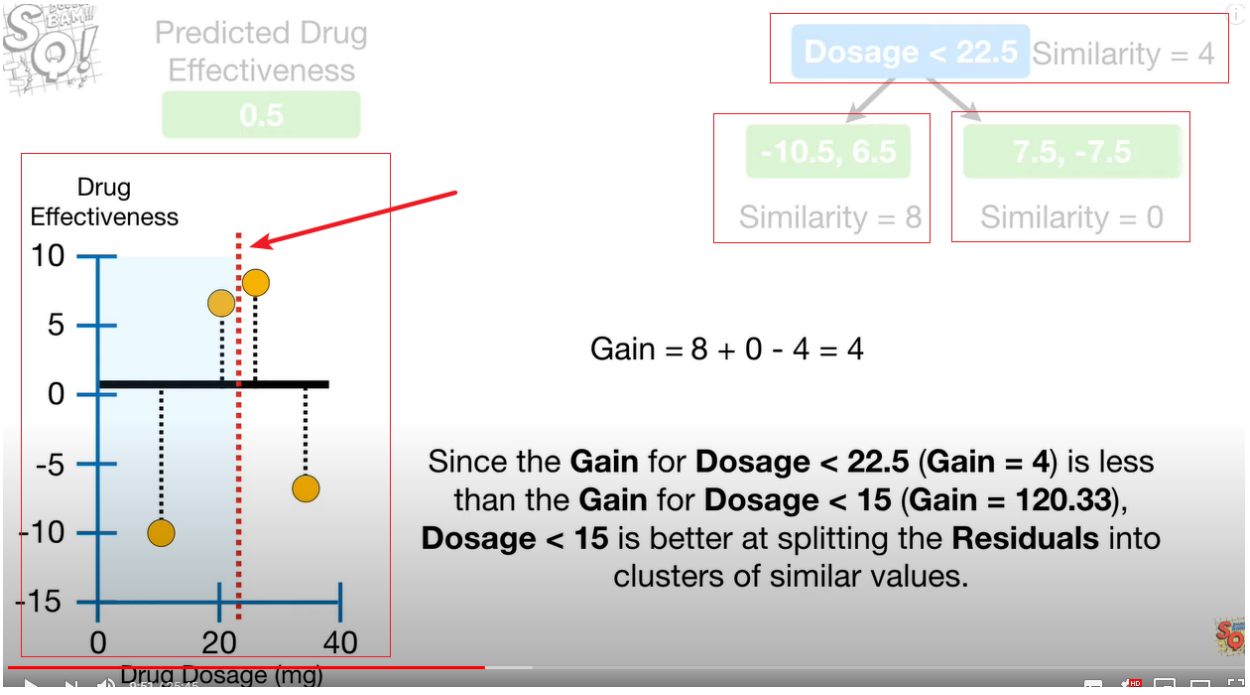
可以看到叶节点的similarity值都比Root节点的要大。是因为在root节点中，有些residuals相互抵消了，比如7.5和-7.5。Similarity值一定程度越大说明样本之间更相似。

接下来要计算Gain叶节点对residuals的聚类想过俾root节点好多少。

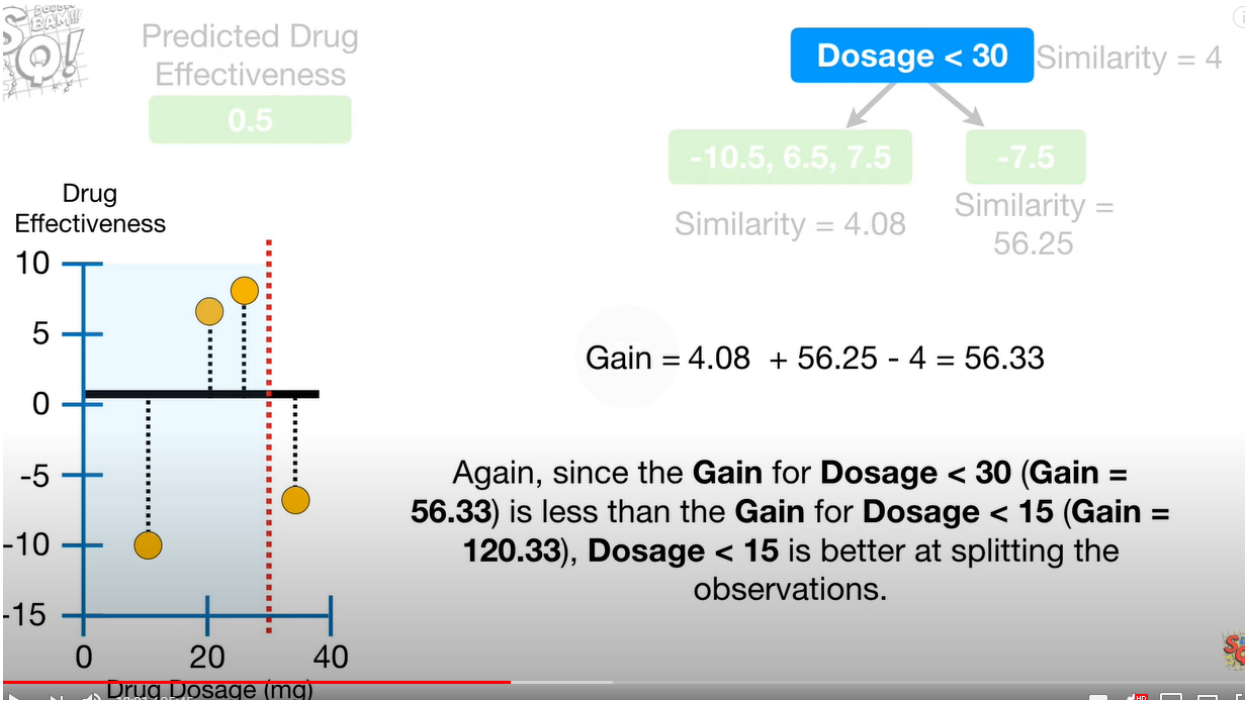




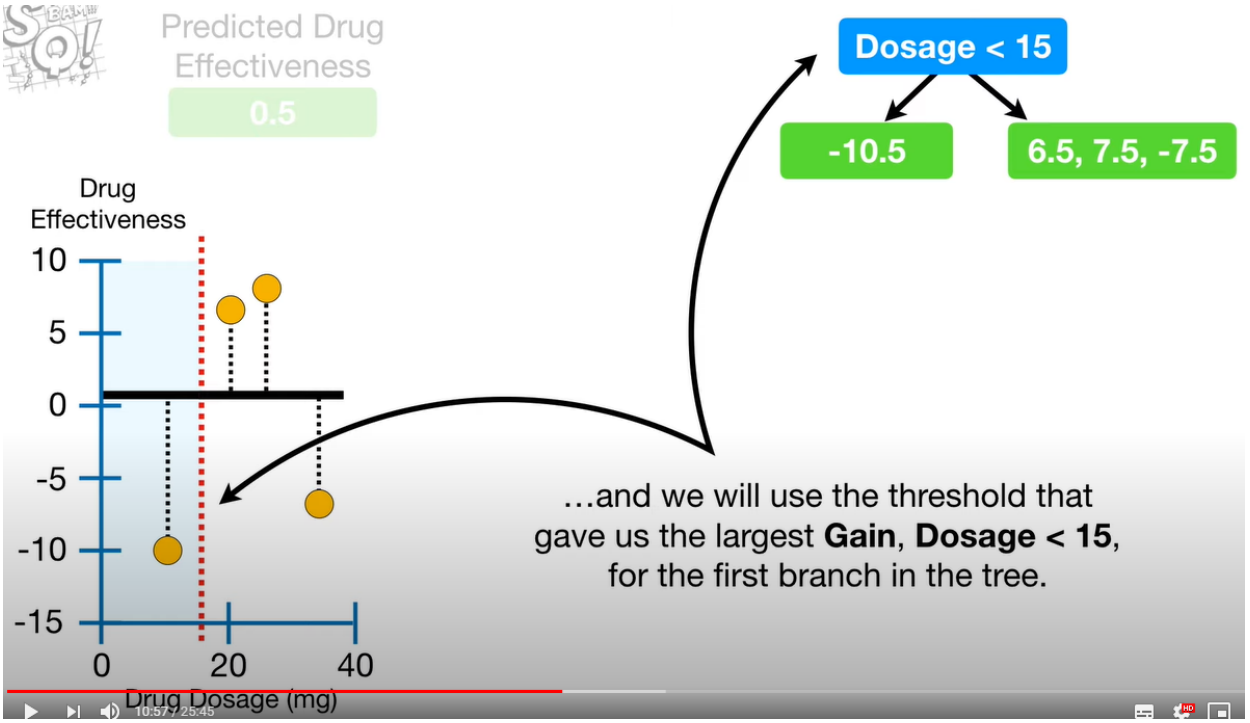
同样地，尝试取第二和第三个样本之间的平均值作为划分标准，计算Gain值



如此类推，取第三个和第四个样本之间的平均值作为划分标准，计算Gain值

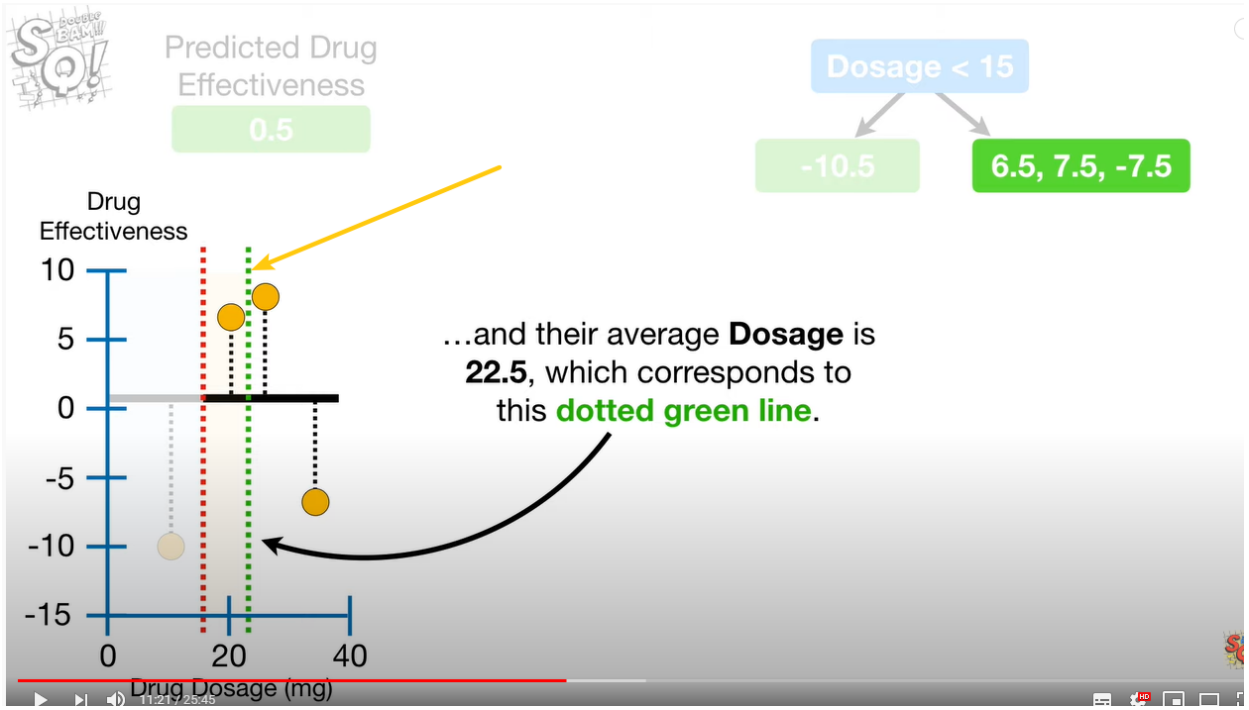


可以看出当Dosage<15作为划分阈值时，Gain的值是最大的，效果最好

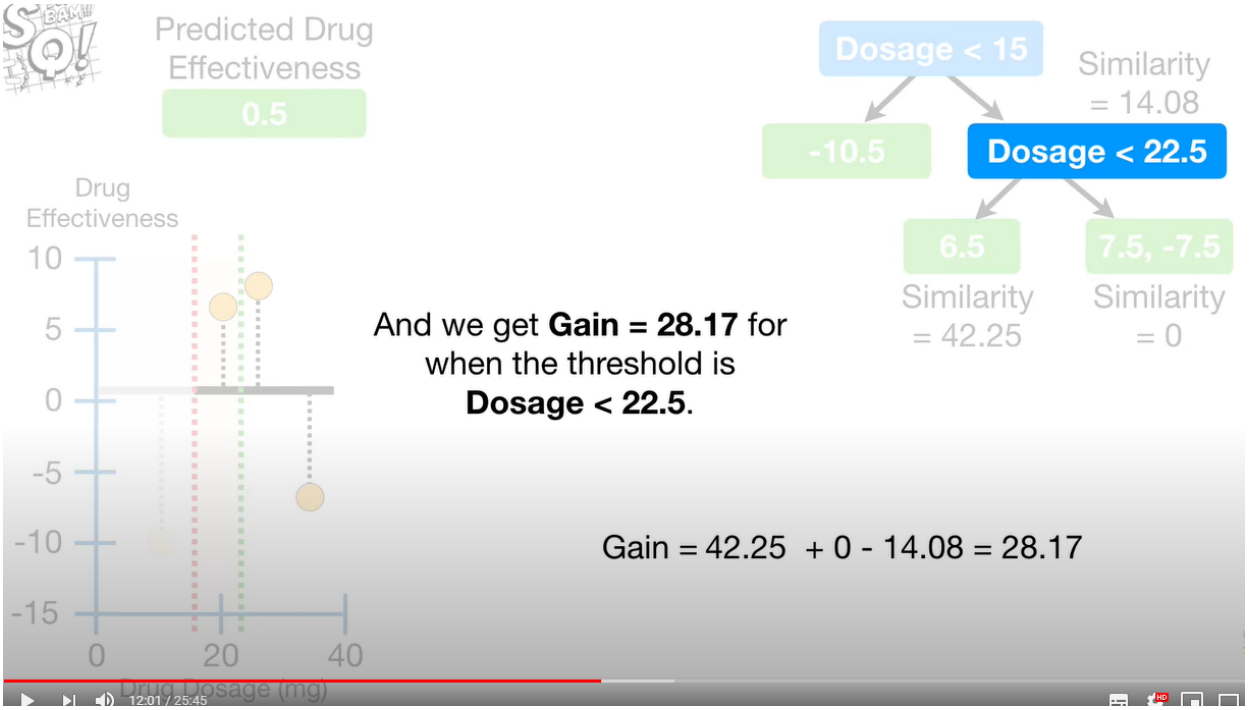


接下来对右子节点的residuals继续进行划分

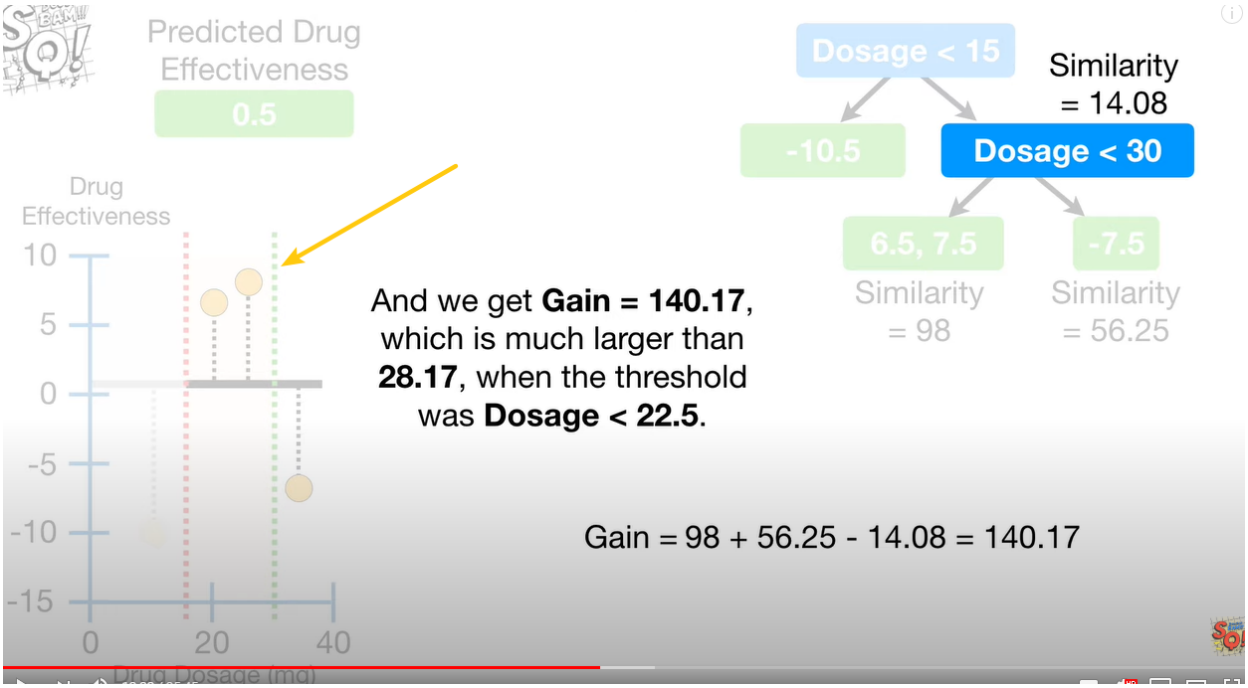
第一次尝试用绿色箭头标记的线作为threshould划分，即dosage的值为第二和第三个样本的平均值



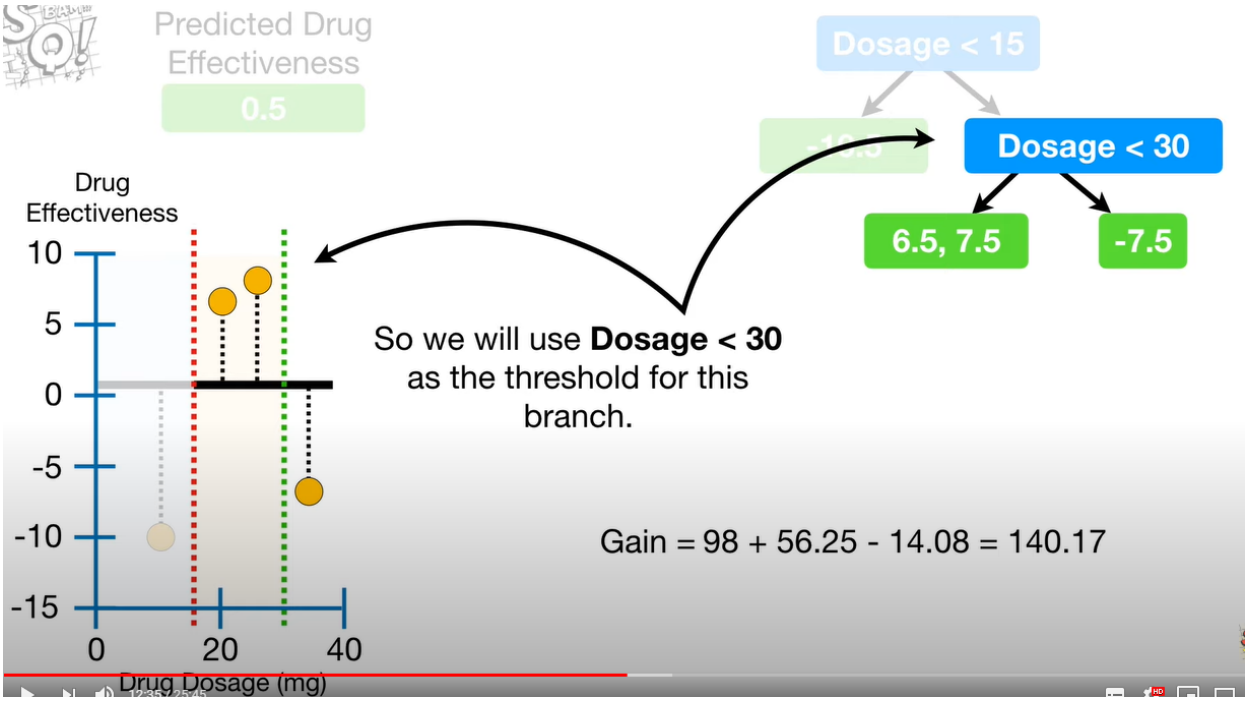
计算得到threshold为22.5时的Gain值



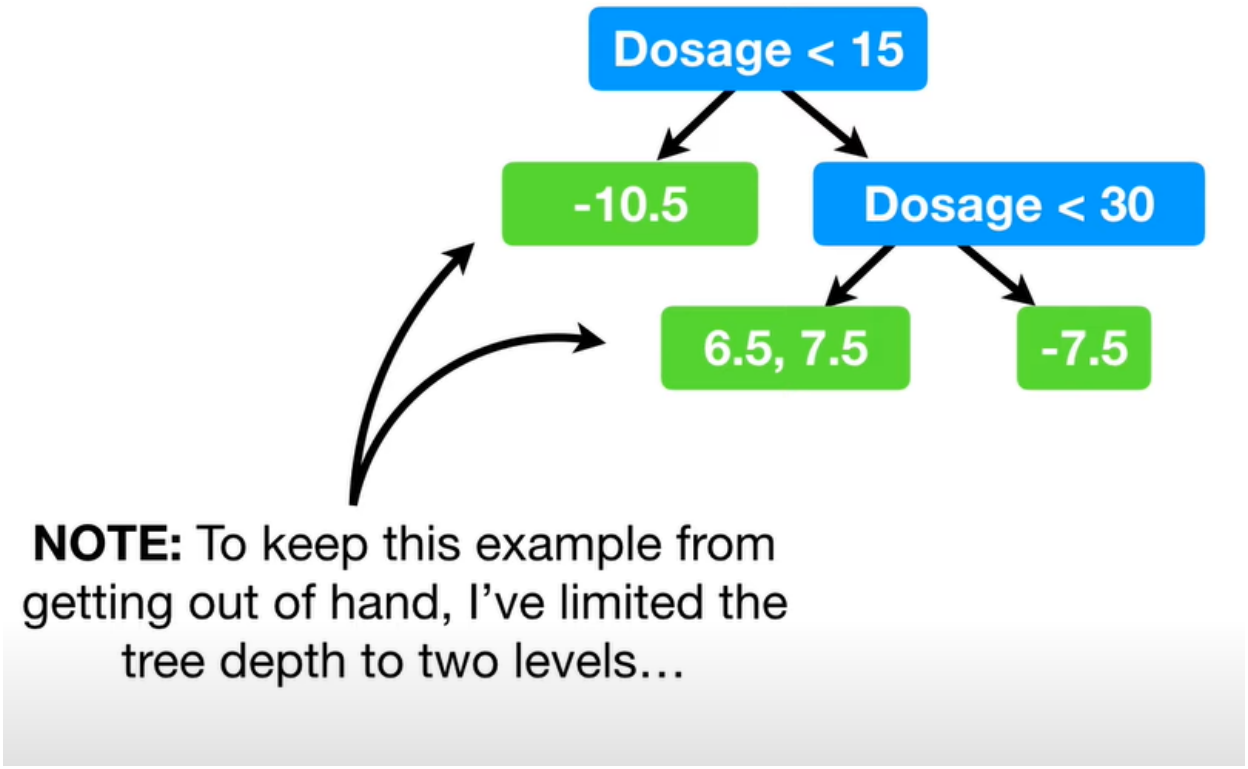
将threshold改为30，即绿色箭头标志的线，为第三和第四个样本的平均值



Gain(30)=140,17>Gain(22.5)=28.17，因此选dosage=30作为划分标准



还可以对（6.5, 7.5）的节点继续进行划分

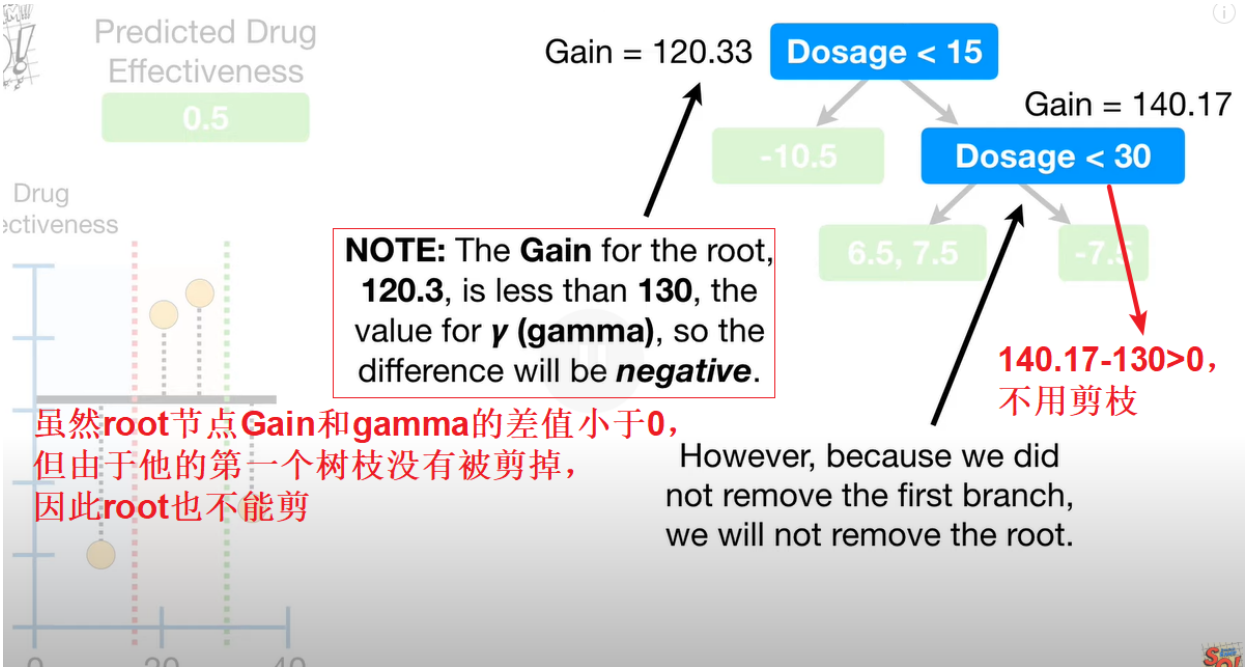


接下来要对树进行剪枝（Prune）

注意剪枝从最深层的分支开始(lowest branch)。

先随机选取一个值，比如130，这个值在XGB中称为γ（Gamma）。

计算Gain和γ的差值。如果差值>0，则保留该分支。如果差值小于0，则剪掉该分支。



但如果γ=150，则整棵树都被减掉。（两个分支的gain-gamma都<0）

到这里，树的构建就完成了。包括选择threshold（划分标准），计算similarity，计算gain，剪枝操作。

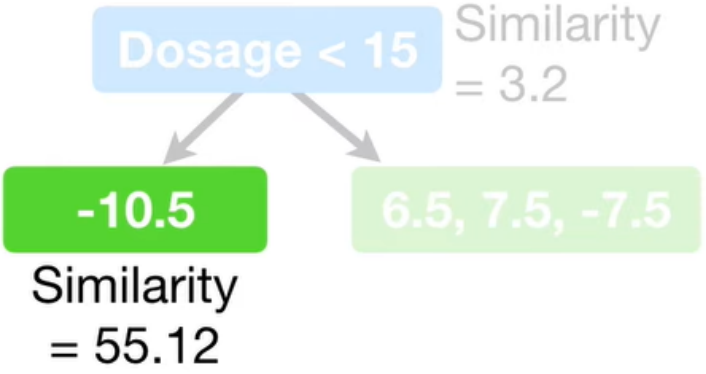
如果计算similarity的公式中，λ的值改变，会有哪些影响？

$$\text{Similarity Score} = \frac{\text{Sum of Residuals, Squared}}{\text{Number of Residuals} + 1}$$

Remember **λ (lambda)** is a **Regularization Parameter**, which means that it is intended to reduce the prediction's sensitivity to individual observations.

λ是正则项，用来解决over fit的问题。

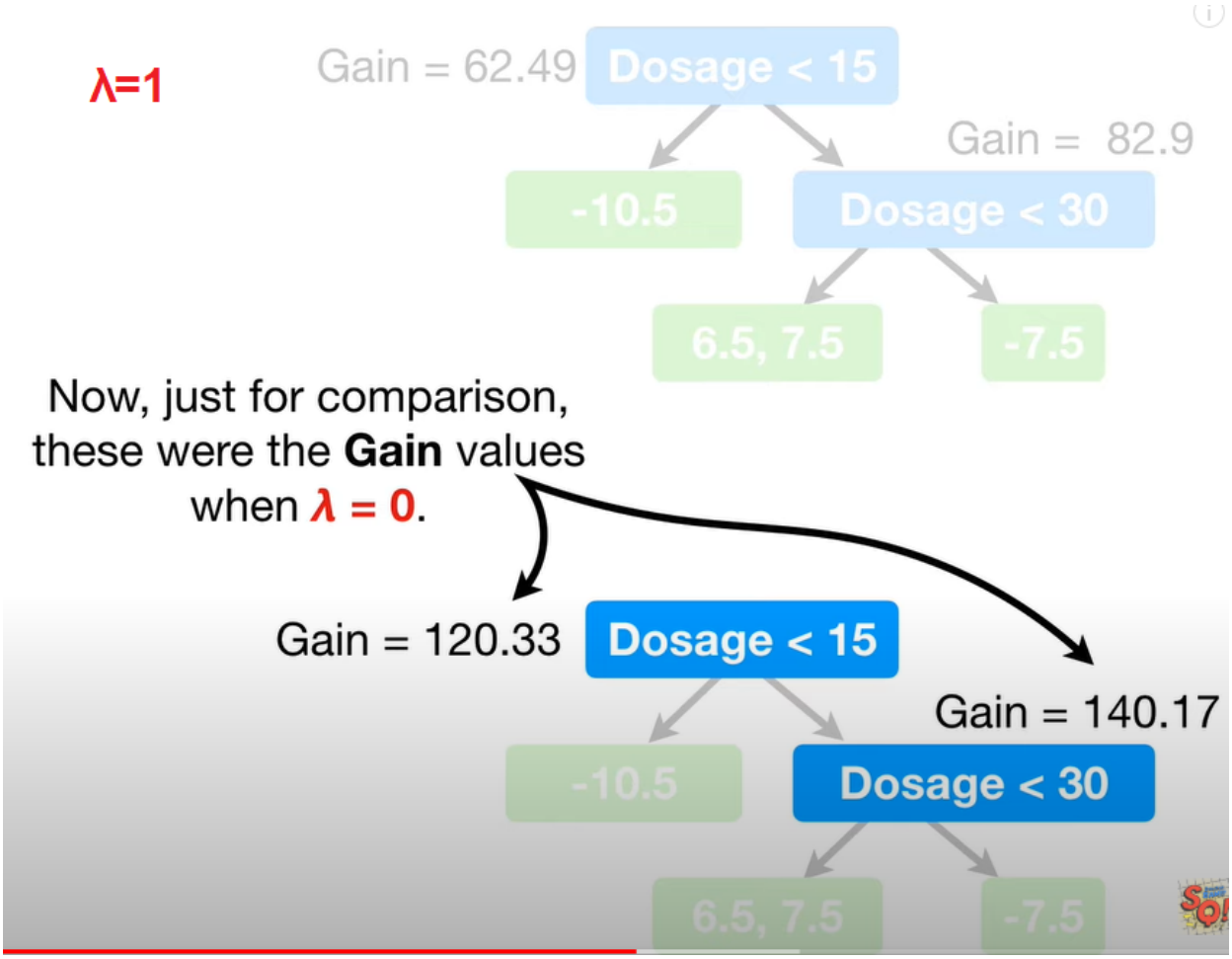
假设当λ=1，和λ=0的时候作对比，左子节点的similarity只有原来的一半。右子节点的similarity只有原来的3/4。



$$\text{Similarity Score} = \frac{-10.5^2}{1 + 1} = 55.12$$

…we get **55.12**, which is half of what we got when **λ = 0**.

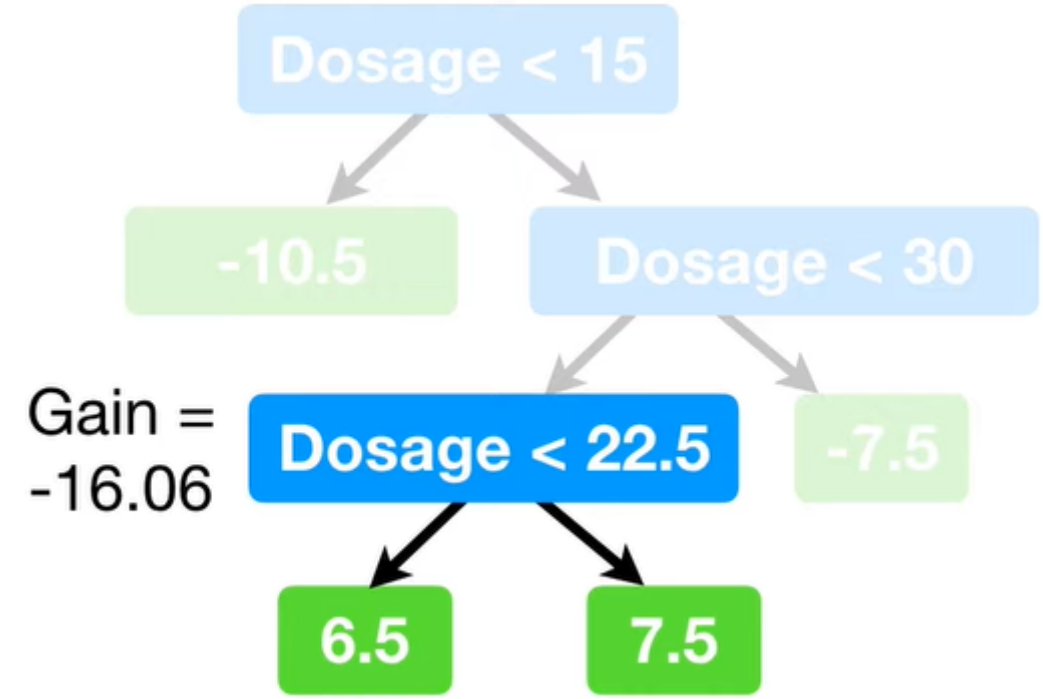
λ=0和λ=1的similarity和gain值对比：



当取γ=130时，λ=0的树会完全保留，而λ=1的树会全部被减掉。

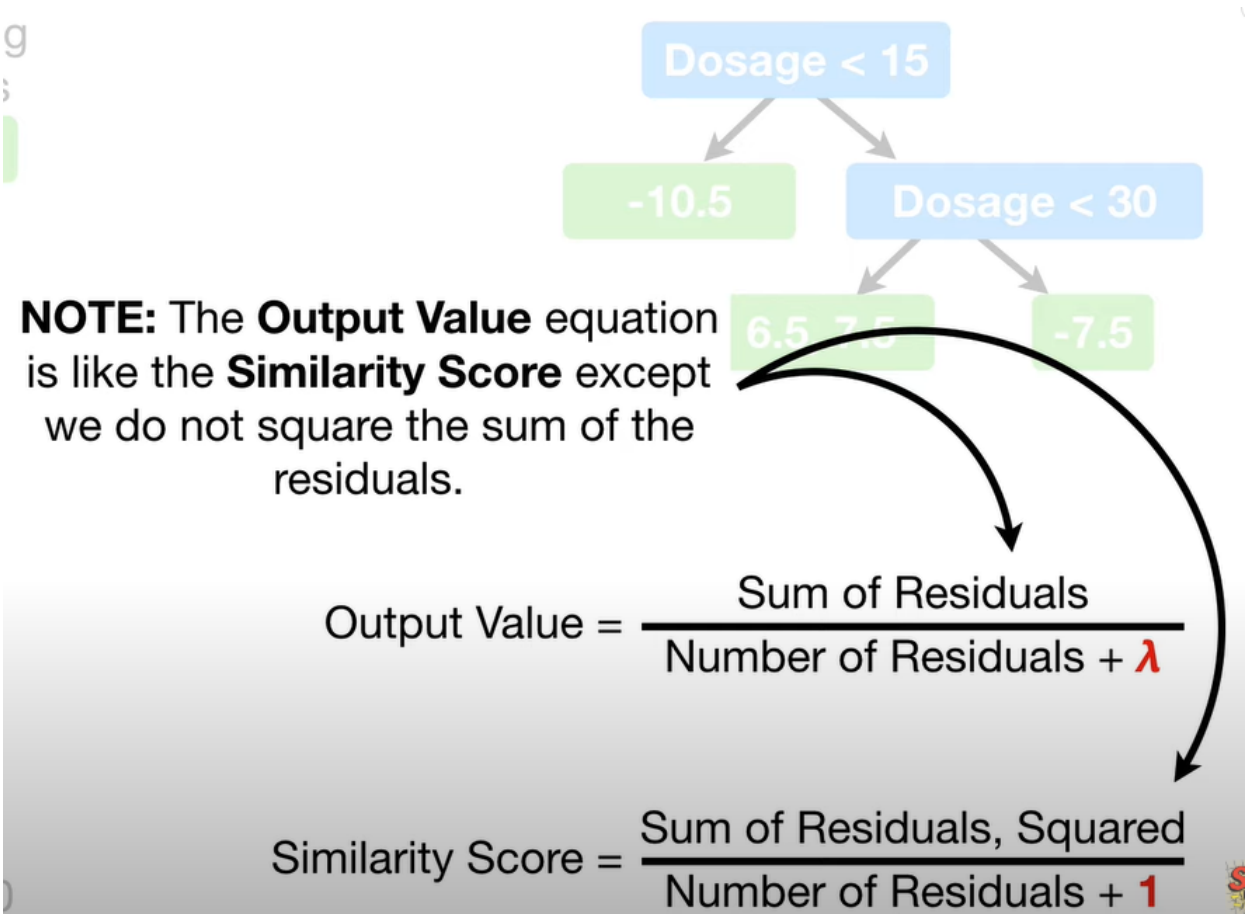
因此，当λ>0时，触发剪枝操作会更容易（因为gain值会因为similarity变小而变小），从而解决过拟合的问题。

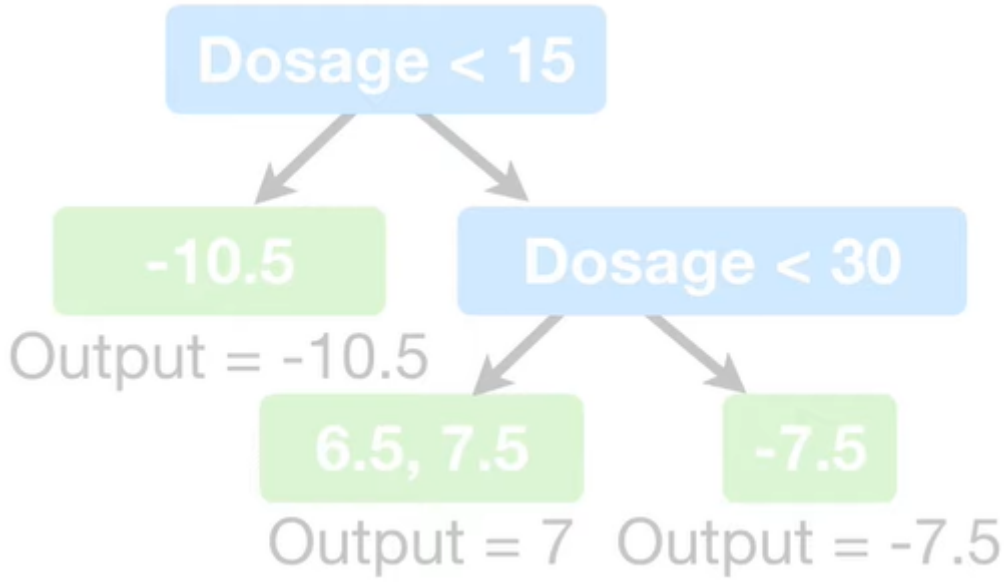
当λ=1时，如果对树继续划分，会得到下面这样的树



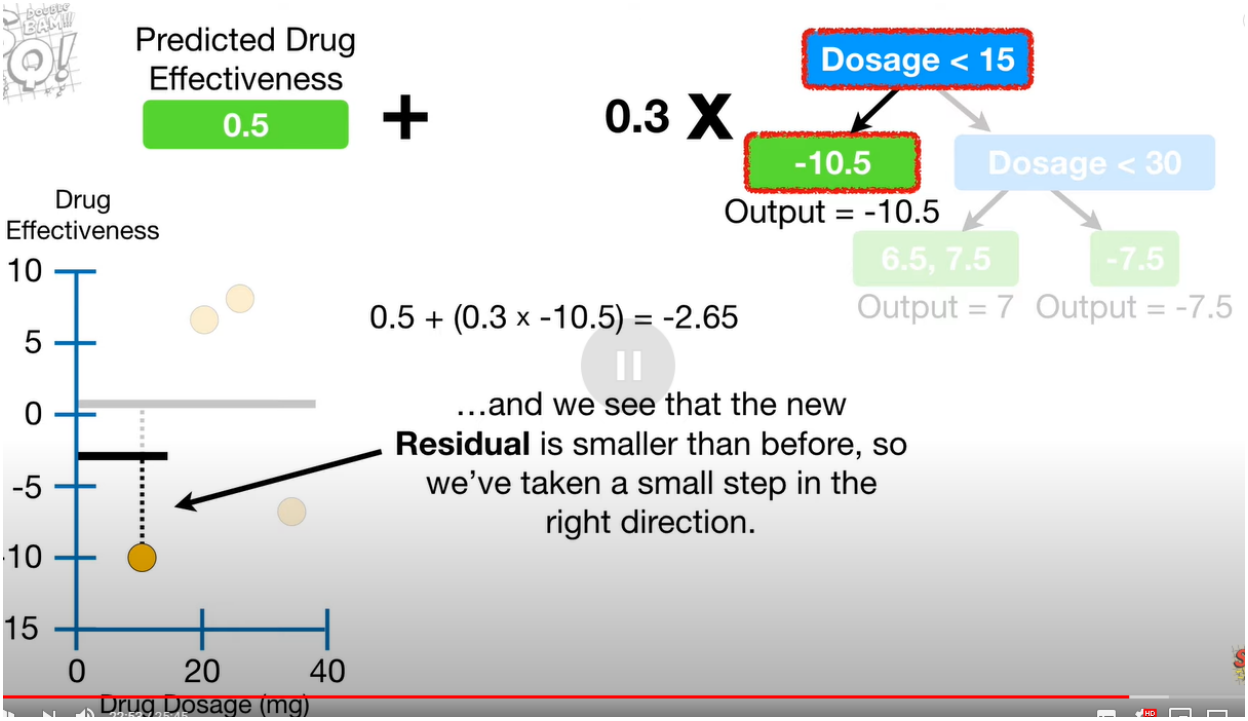
这时假设γ=0，Gain-γ还是小于0。可以得出结论，即使γ=0，剪枝还是会发生。

计算每个叶节点的输出值



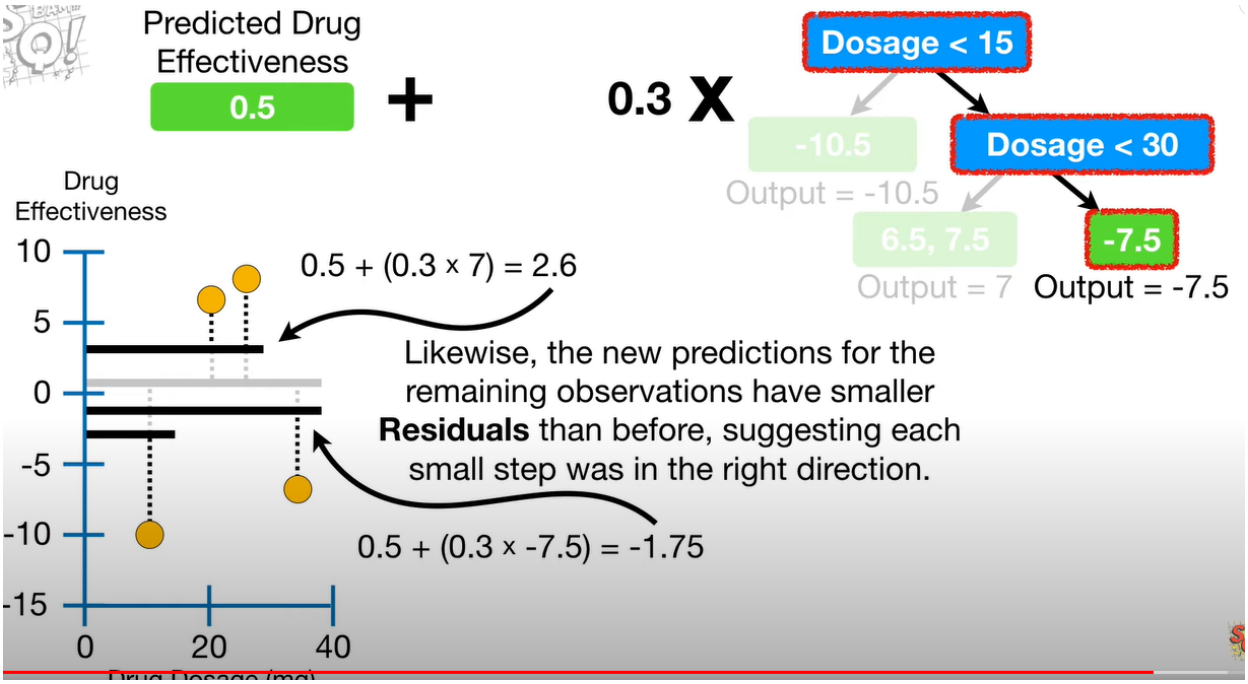


使用XGB预测



XGB中的learning rate的默认值为0.3.

以第一个样本为例，新的预测值为-2.65，可以看到与原预测值（0.5）相比，residual已经变小了。



接着使用新的预测值，得到新的residual，建造新的树。预测时将sample在所有树的output value * learning rate相加，作为预测值（不确定每个树的learning rate是否要相同）。

在XGB中，可以限制树的深度和每个叶节点residuals的个数（称为Cover）。Cover是regression问题中，similarity计算公式中分母的减掉λ，即number of residuals。默认值为1。因此当Cover=1时，对树的构造没有影响。但是在classification问题中，Cover的取值就很复杂。

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