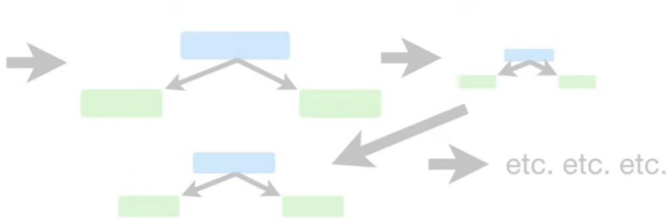
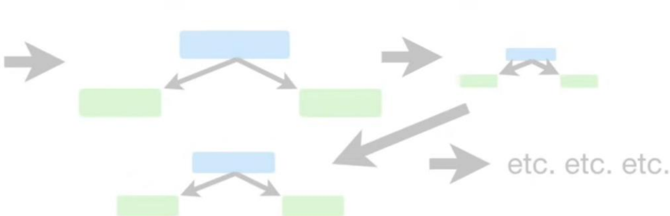
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
etc…	etc…	etc…	etc…



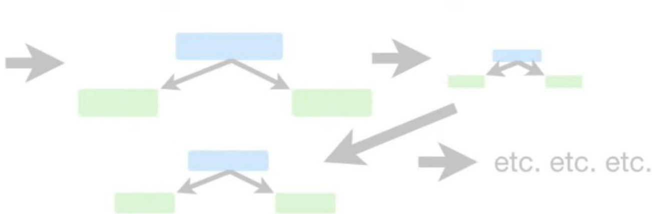
In contrast, **Gradient Boost** starts by making a single leaf, instead of a tree or stump.

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
etc…	etc…	etc…	etc…

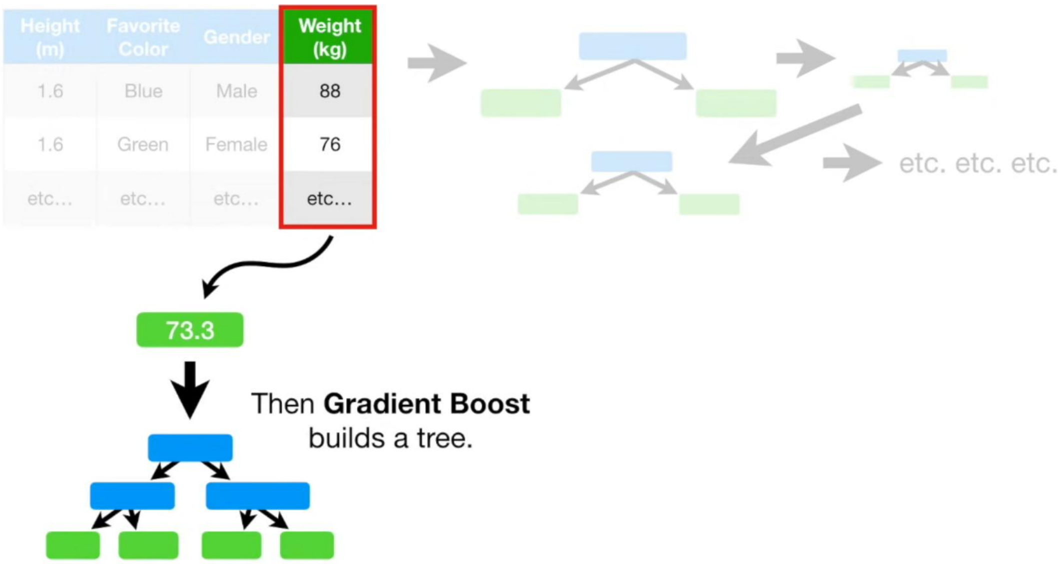


This leaf represents an initial guess for the **Weights** of all of the samples.

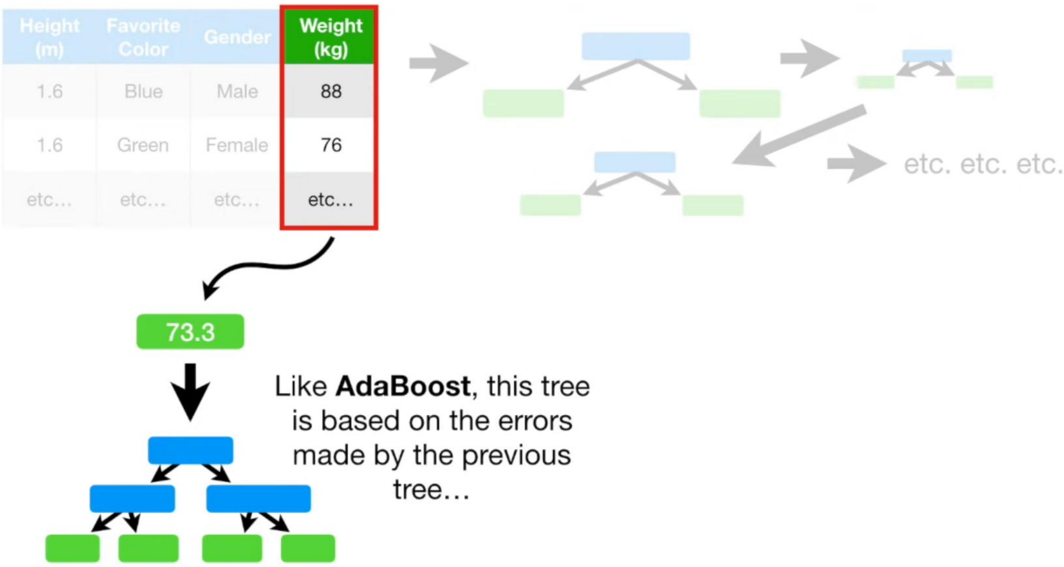
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
etc…	etc…	etc…	etc…



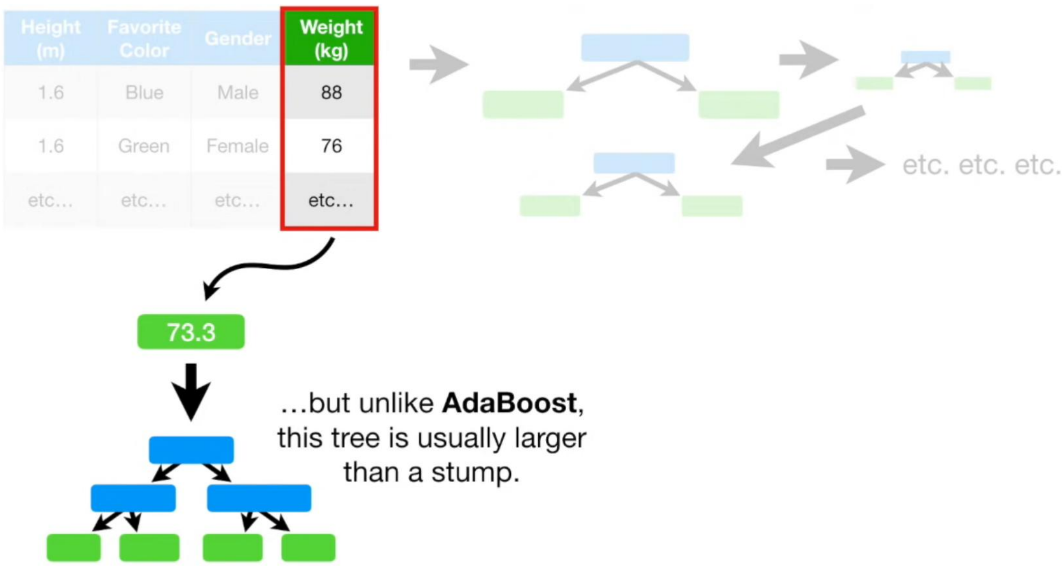
When trying to **Predict** a continuous value like **Weight**, the first guess is the the average value.



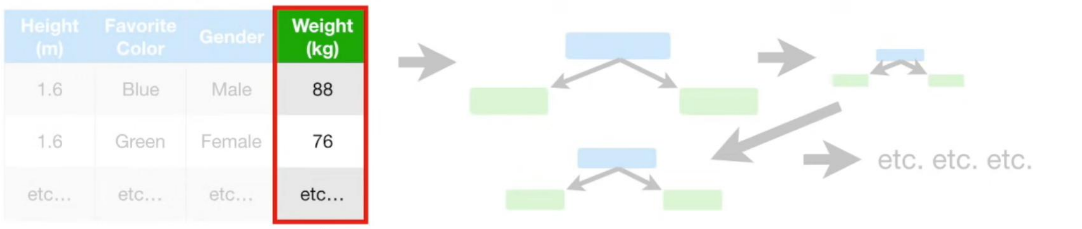
Then **Gradient Boost** builds a tree.

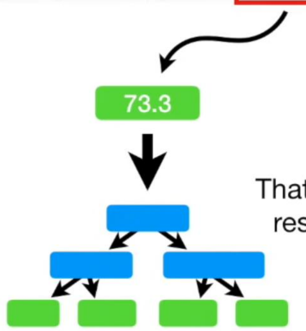


Like **AdaBoost**, this tree is based on the errors made by the previous tree…

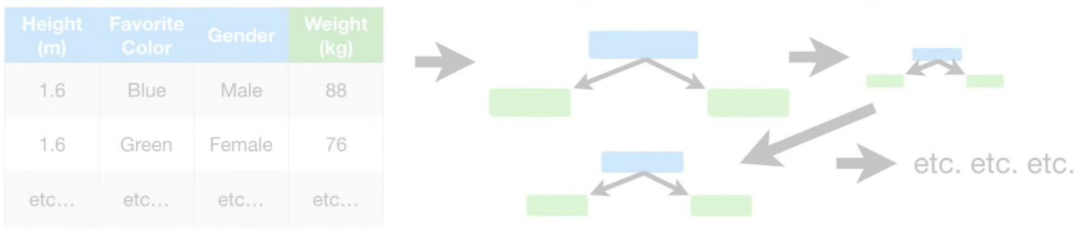


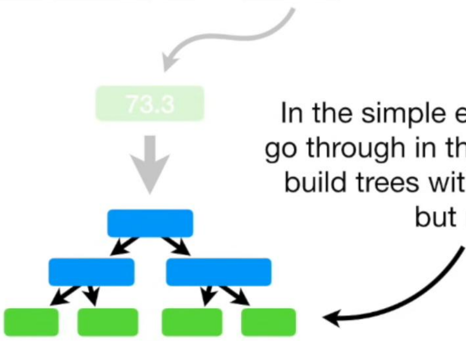
…but unlike **AdaBoost**, this tree is usually larger than a stump.



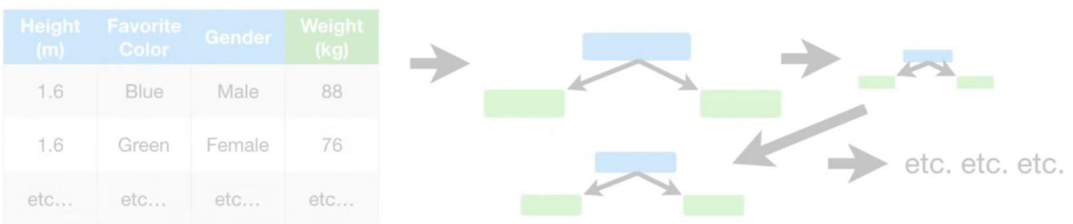


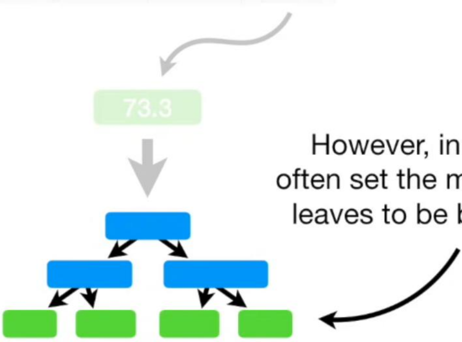
That said, **Gradient Boost** still restricts the size of the tree.



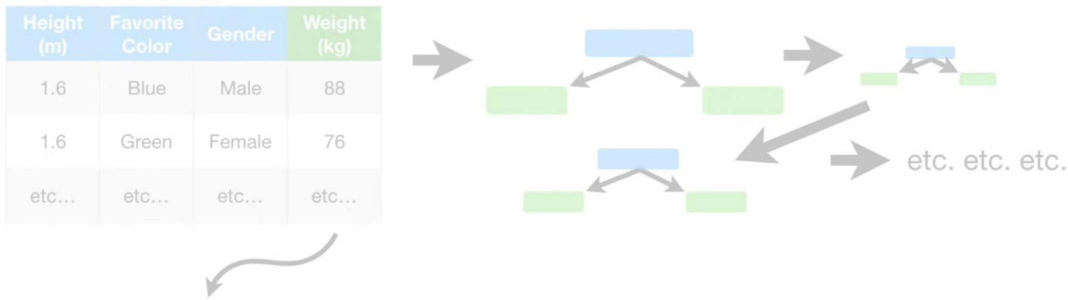


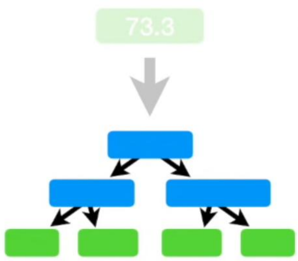
In the simple example that we will go through in this StatQuest, we will build trees with up to four leaves, but no larger.



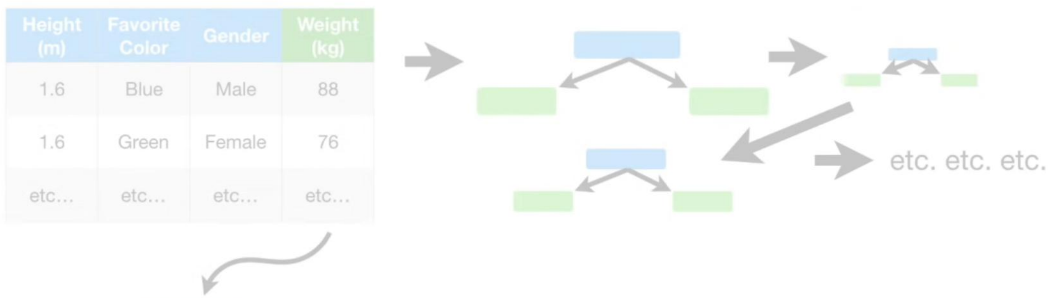


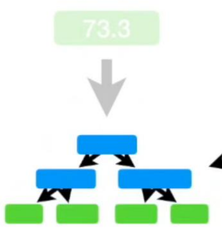
However, in practice, people often set the maximum number of leaves to be between **8** and **32**.



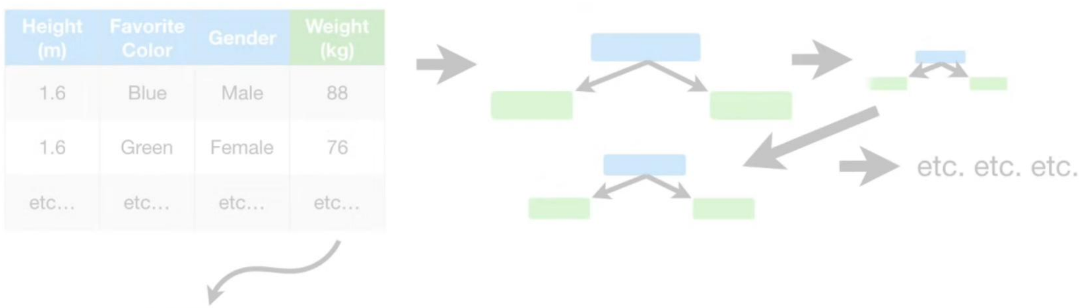


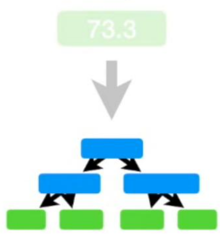
Thus, like **AdaBoost**, **Gradient Boost** builds fixed sized trees based on the previous tree's errors, but unlike **AdaBoost**, each tree can be larger than a stump.



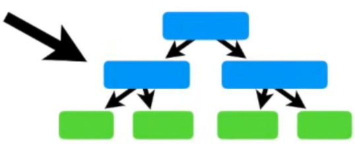


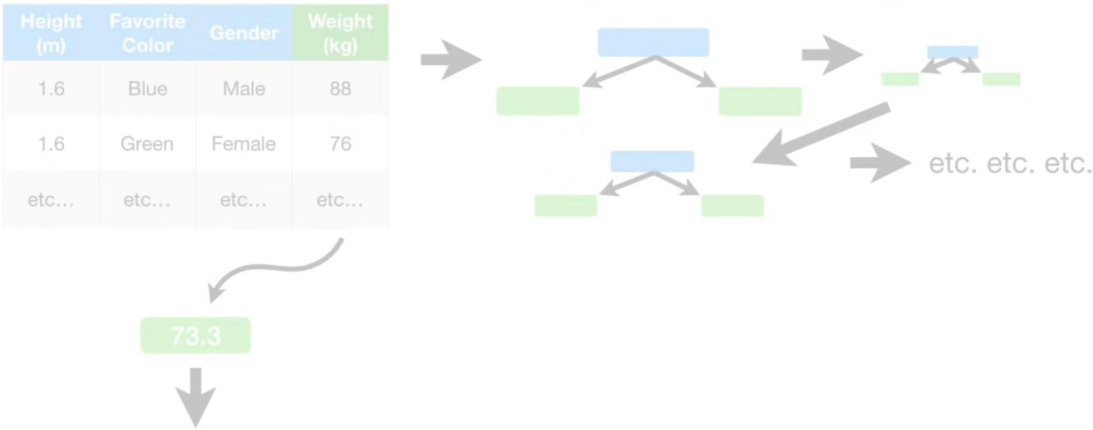
Also like **AdaBoost**, **Gradient Boost** scales the trees. However, **Gradient Boost** scales all trees by the same amount.



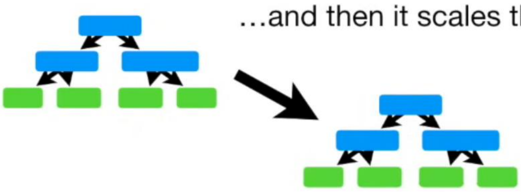


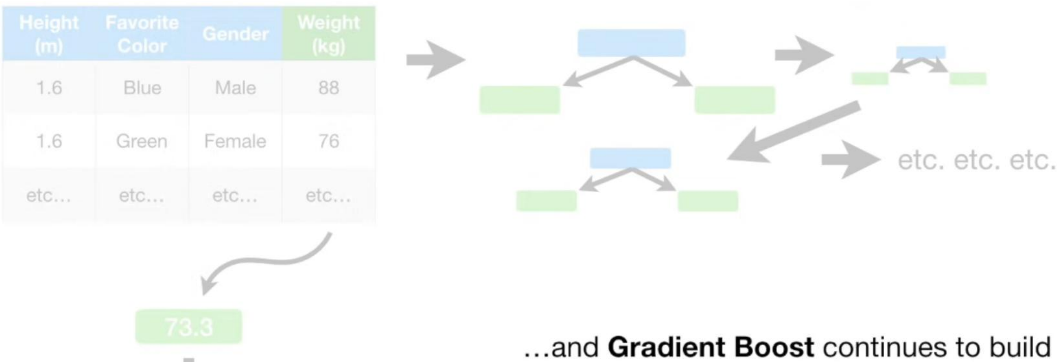
Then **Gradient Boost** builds another tree based on the errors made by the previous tree…



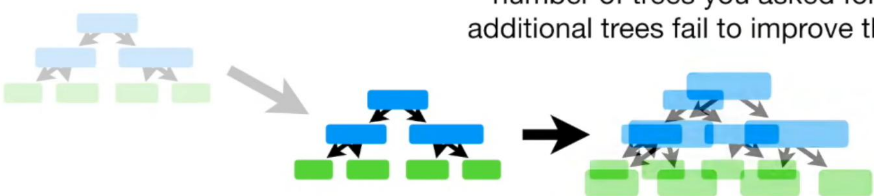


…and then it scales the tree…





…and **Gradient Boost** continues to build trees in this fashion until it has made the number of trees you asked for, or additional trees fail to improve the fit.



…let's see how the most common **Gradient Boost** configuration would use this **Training Data** to **Predict Weight**.

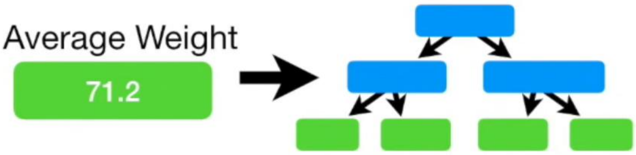
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

Average Weight

71.2

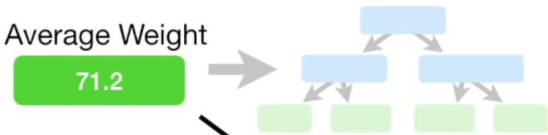
The first thing we do is calculate the average **Weight.**

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57



Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

The next thing we do is build a tree based on the errors from the first tree.

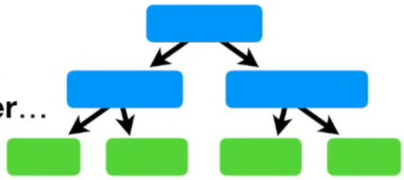


Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

The errors that the previous tree made are the differences between the **Observed Weights** and the **Predicted Weight**, **71.2**.

(**Observed Weight** - **Predicted Weight**)

Now we will build a **Tree**, using **Height**, **Favorite Color** and **Gender**…



Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

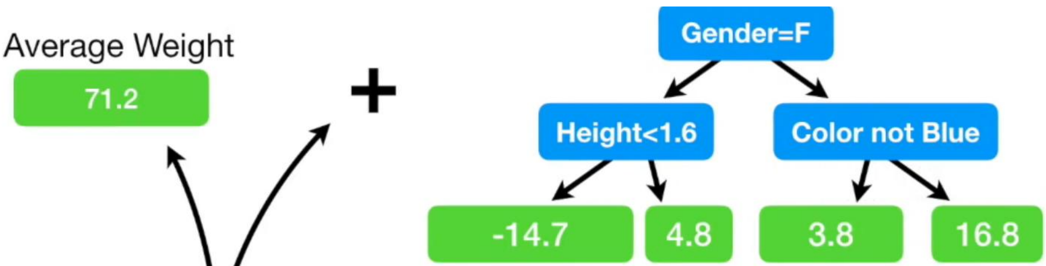
…to **Predict** the **Residuals**.

Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

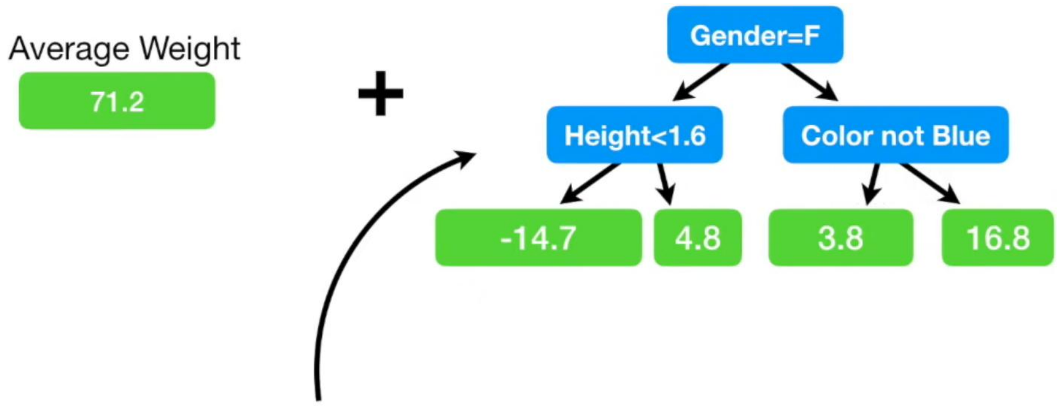
Remember, in this example we are only allowing up to four leaves…

…but when using a larger dataset, it is common to allow anywhere from **8** to **32**.

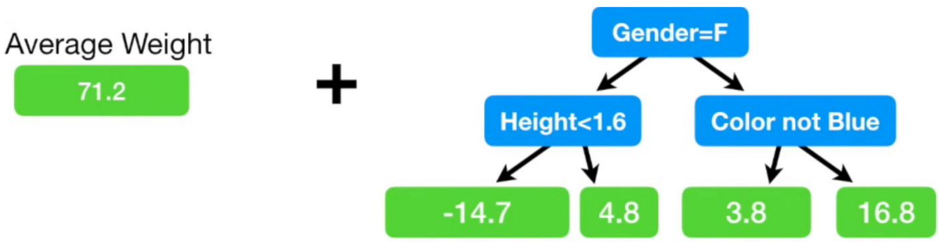
Average Weight



Now we can now combine the original leaf…



...with the new tree...

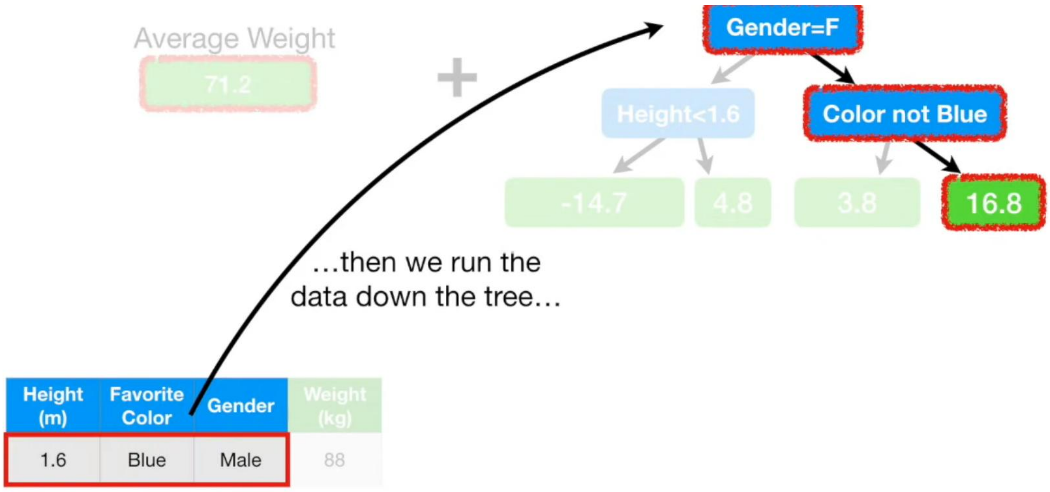


Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88

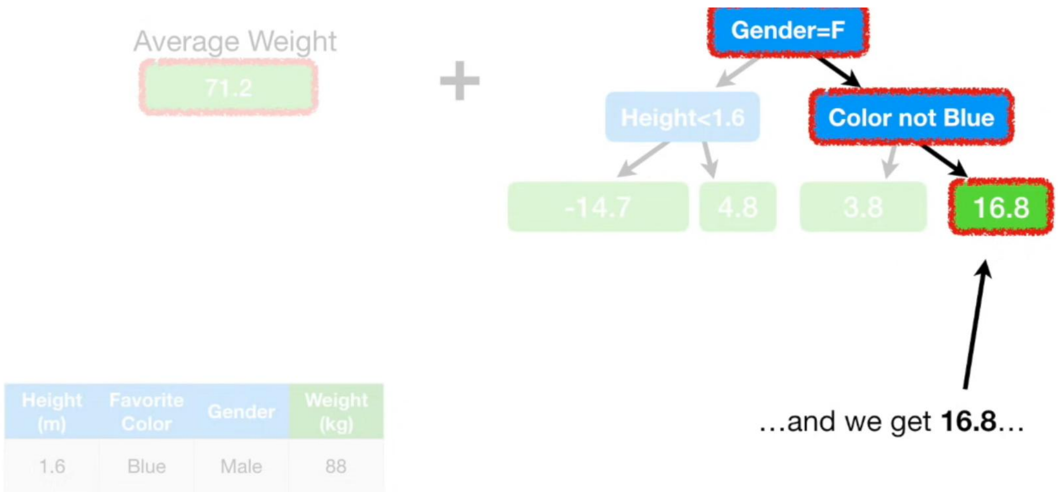
...to make a new **Prediction** of an individual's **Weight** from the **Training Data**.

We start with the initial **Prediction**, **71.2**...

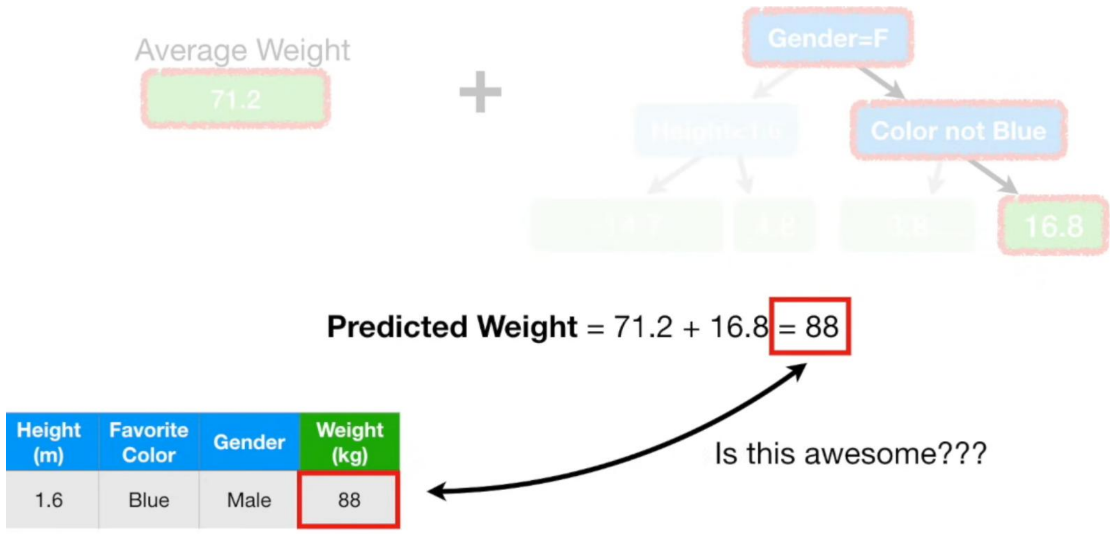
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88



Average Weight
71.2
+
Gender=F
Height<1.6
Color not Blue
-14.7
4.8
3.8
16.8
…then we run the data down the tree…
Height (m)
Favorite Color
Gender
Weight (kg)
1.6
Blue
Male
88



Average Weight
71.2
+
Gender=F
Height<1.6
Color not Blue
-14.7
4.8
3.8
16.8
…and we get **16.8**…
Height (m)
Favorite Color
Gender
Weight (kg)
1.6
Blue
Male
88



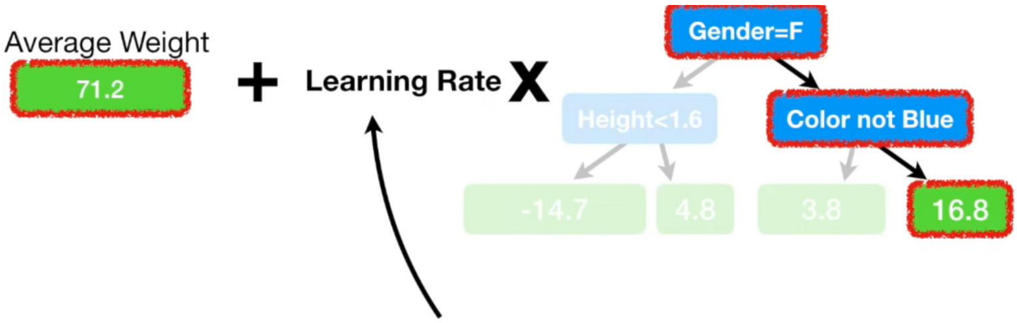
Average Weight
71.2
+
Gender=F
Color not Blue
16.8
Predicted Weight = 71.2 + 16.8 = 88
Is this awesome???
Height (m)
Favorite Color
Gender
Weight (kg)
1.6
Blue
Male
88

Predicted Weight = 71.2 + 16.8 = 88

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88

No. The model fits the **Training Data** too well.

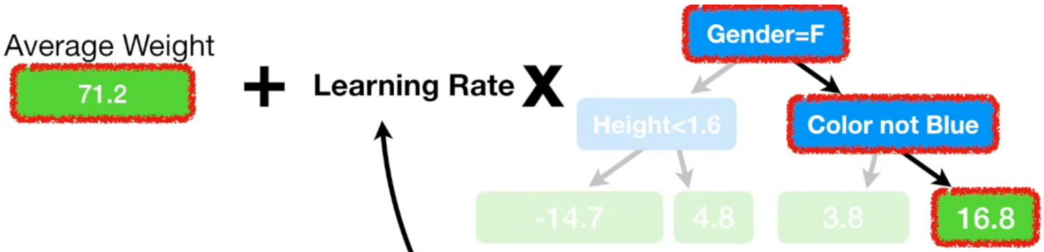
In other words, we have low **Bias**, but probably very high **Variance**.



Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88

Gradient Boost deals with this problem by using a **Learning Rate** to scale the contribution from the new tree.

The **Learning Rate** is a value between **0** and **1**.



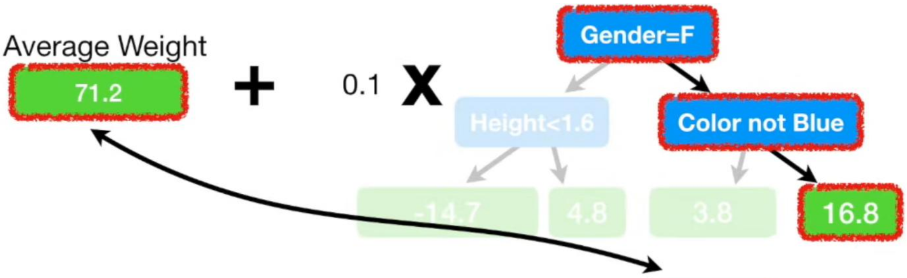
In this case, we'll set the **Learning Rate** to **0.1**.

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88

Predicted Weight = 71.2 + (0.1 × 16.8) = 72.9

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88

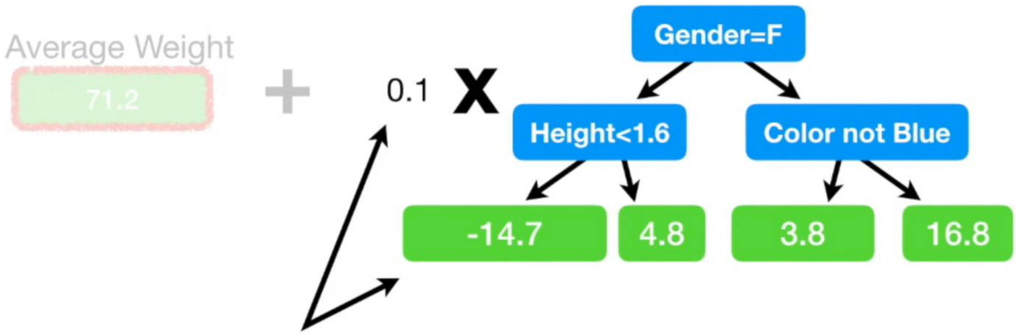
With the **Learning Rate** set to **0.1**, the new **Prediction** isn't as good as as it was before…



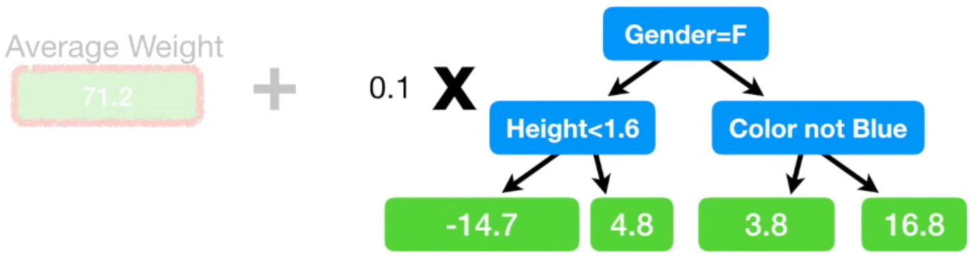
Predicted Weight = 71.2 + (0.1 × 16.8) = 72.9

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88

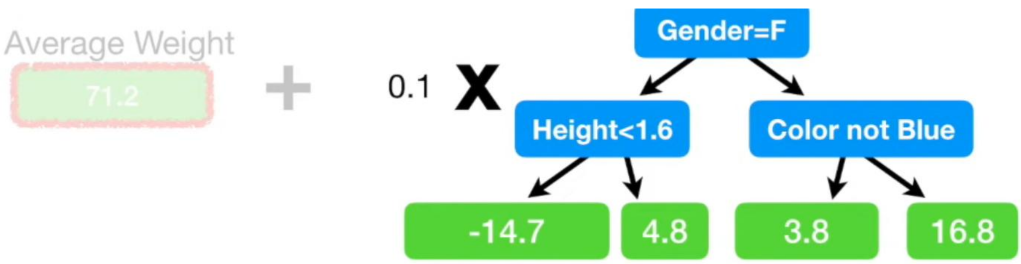
…but it's a little bit better than the **Prediction** made with just the original leaf, which predicted that all samples would weigh **71.2**.



In other words, scaling the tree by the **Learning Rate** results in a small step in the right direction.



According to the dude that invented **Gradient Boost**, Jerome Friedman, empirical evidence shows that taking lots of small steps in the right direction results in better **Predictions** with a **Testing Dataset**, i.e. lower **Variance**.



So let's build another tree so we can take another small step in the right direction.

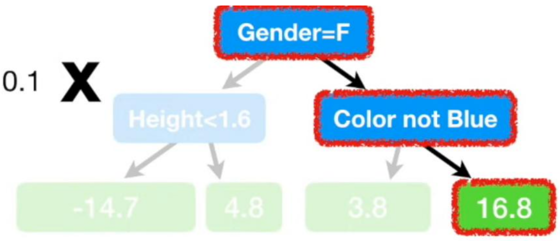
Just like before, we calculate the **Pseudo Residuals**, the difference between the **Observed Weights** and our latest **Predictions**.

Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	
1.6	Green	Female	76	
1.5	Blue	Female	56	
1.8	Red	Male	73	
1.5	Green	Male	77	
1.4	Blue	Female	57	

Residual = (**Observed** - **Predicted**)

Average Weight

71.2 + 0.1 X



Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	
1.6	Green	Female	76	
1.5	Blue	Female	56	
1.8	Red	Male	73	
1.5	Green	Male	77	
1.4	Blue	Female	57	

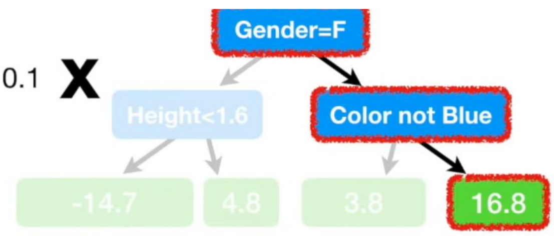
Residual = (**88** - (71.2 + 0.1 × 16.8))

= 15.1

…and we get **15.1**…

Average Weight

71.2 + 0.1 X



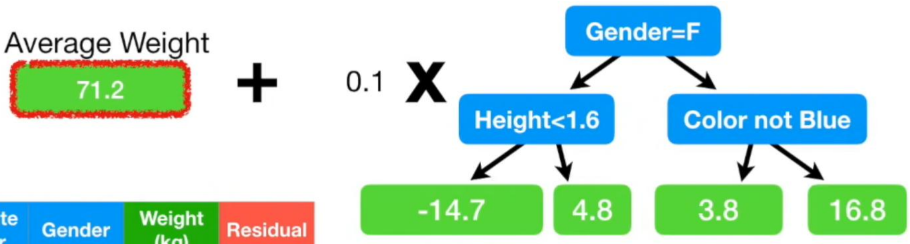
Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	
1.6	Green	Female	76	
1.5	Blue	Female	56	
1.8	Red	Male	73	
1.5	Green	Male	77	
1.4	Blue	Female	57	

Residual = (**88** - (71.2 + 0.1 × 16.8))

$\boxed{= 15.1}$

…and we save that in the column for **Pseudo Residuals**.

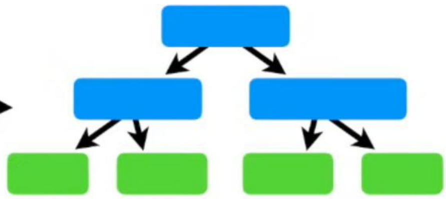
Then we repeat for the all of the other individuals in the **Training Dataset**.



Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	15.1
1.6	Green	Female	76	4.3
1.5	Blue	Female	56	-13.7
1.8	Red	Male	73	1.4
1.5	Green	Male	77	5.4
1.4	Blue	Female	57	-12.7

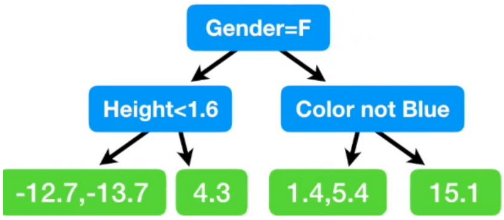
Small bam.

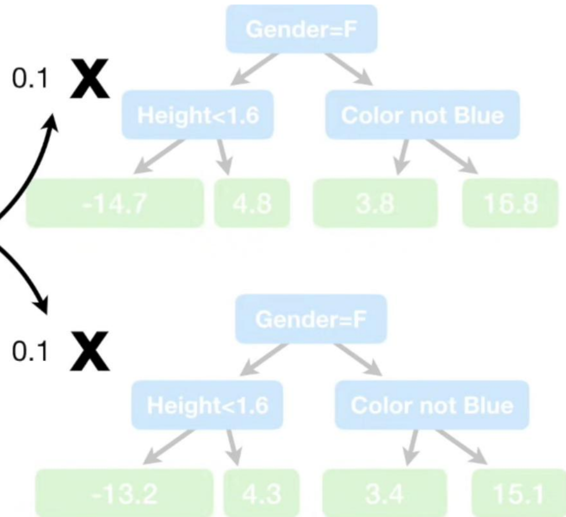
Now let's build a new tree to predict the new **Residuals**.



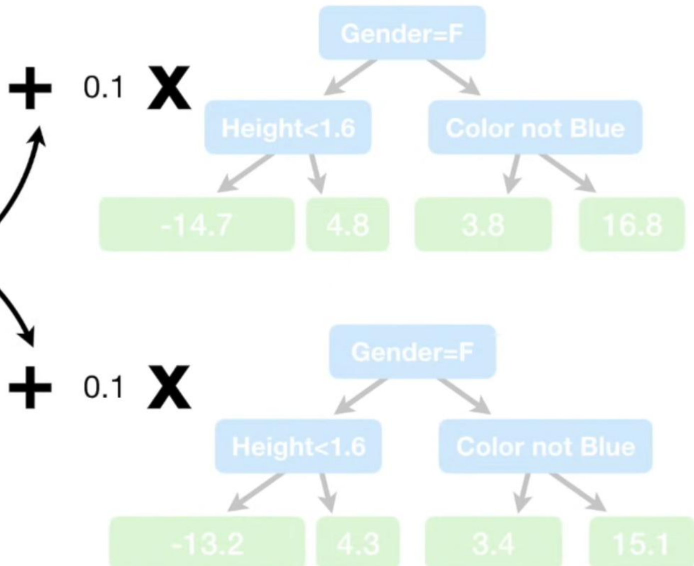
Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	15.1
1.6	Green	Female	76	4.3
1.5	Blue	Female	56	-13.7
1.8	Red	Male	73	1.4
1.5	Green	Male	77	5.4
1.4	Blue	Female	57	-12.7

And here's the new tree!

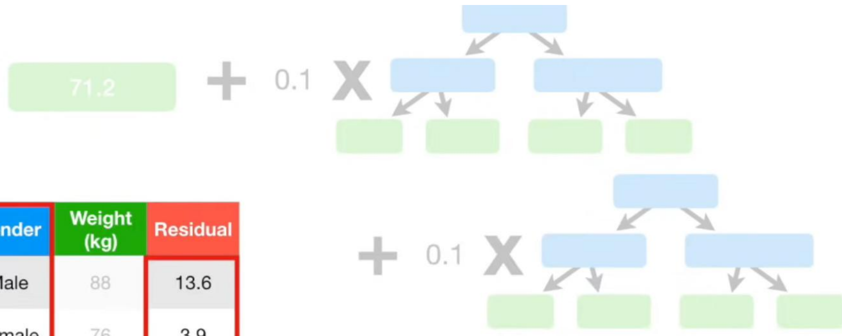




NOTE: We scale all of the **Trees** by the **Learning Rate**, which we set to **0.1**…

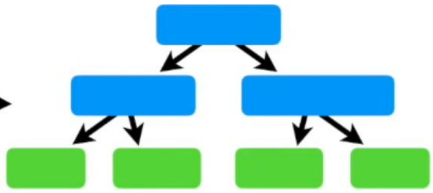


…and add everything together.



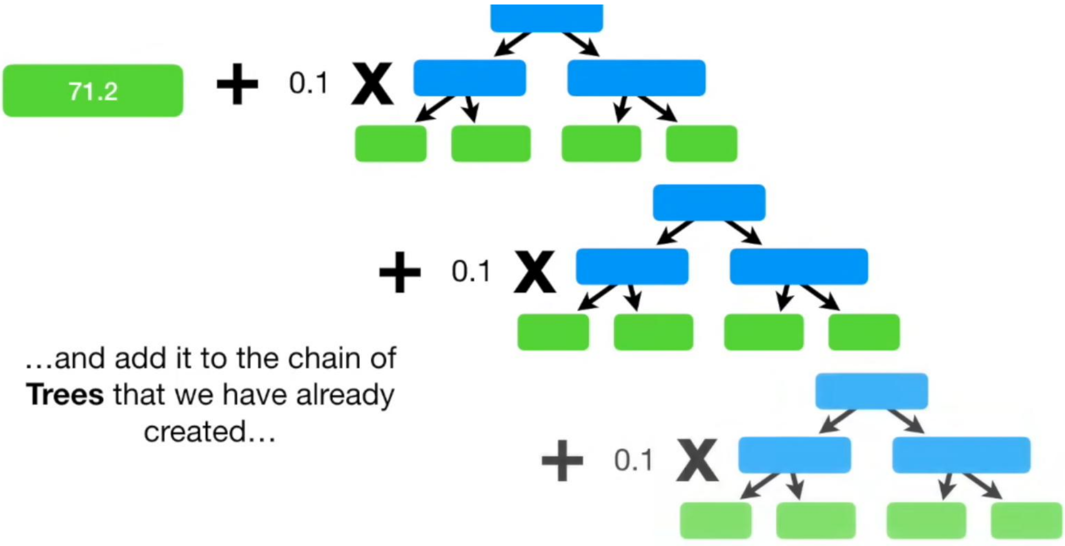
Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	13.6
1.6	Green	Female	76	3.9
1.5	Blue	Female	56	-12.4
1.8	Red	Male	73	1.1
1.5	Green	Male	77	5.1
1.4	Blue	Female	57	-11.4

…to calculate new **Residuals**.

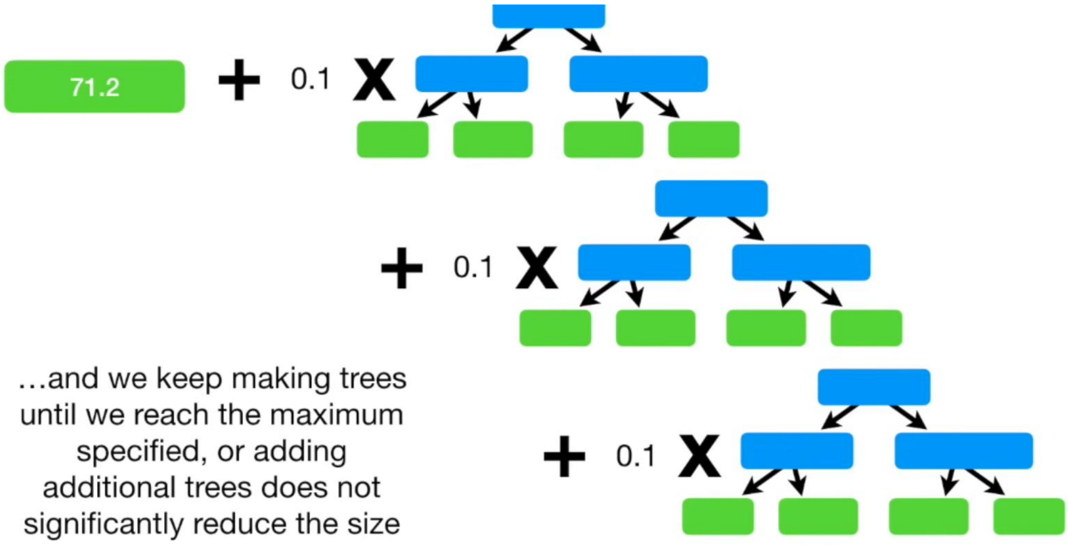


Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	13.6
1.6	Green	Female	76	3.9
1.5	Blue	Female	56	-12.4
1.8	Red	Male	73	1.1
1.5	Green	Male	77	5.1
1.4	Blue	Female	57	-11.4

Now we build a another tree to predict the new **Residuals**…



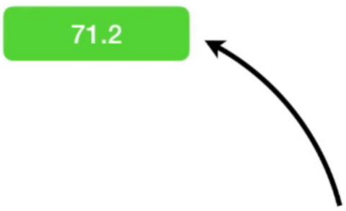
…and add it to the chain of **Trees** that we have already created…



…and we keep making trees until we reach the maximum specified, or adding additional trees does not significantly reduce the size of the **Residuals**.

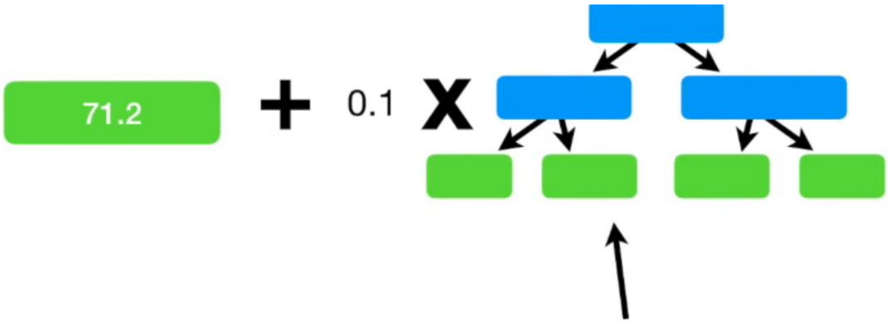
…etc…etc…etc…

In summary, when **Gradient Boost** is used for **Regression**…

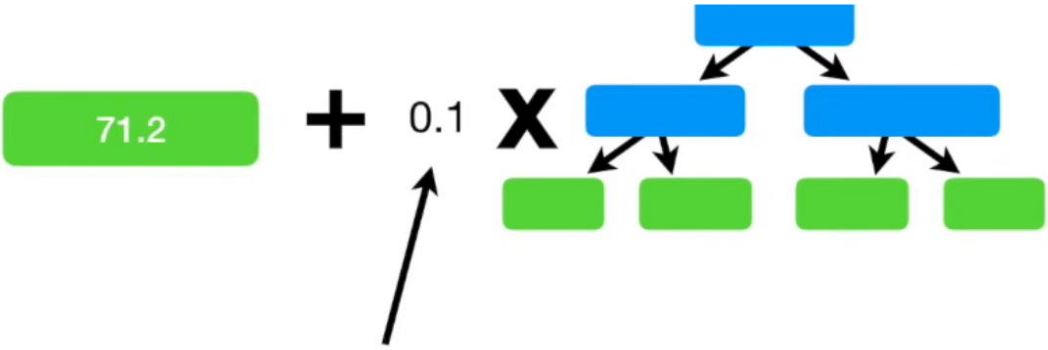


…we start with a leaf that is the average value of the variable we want to **Predict**.

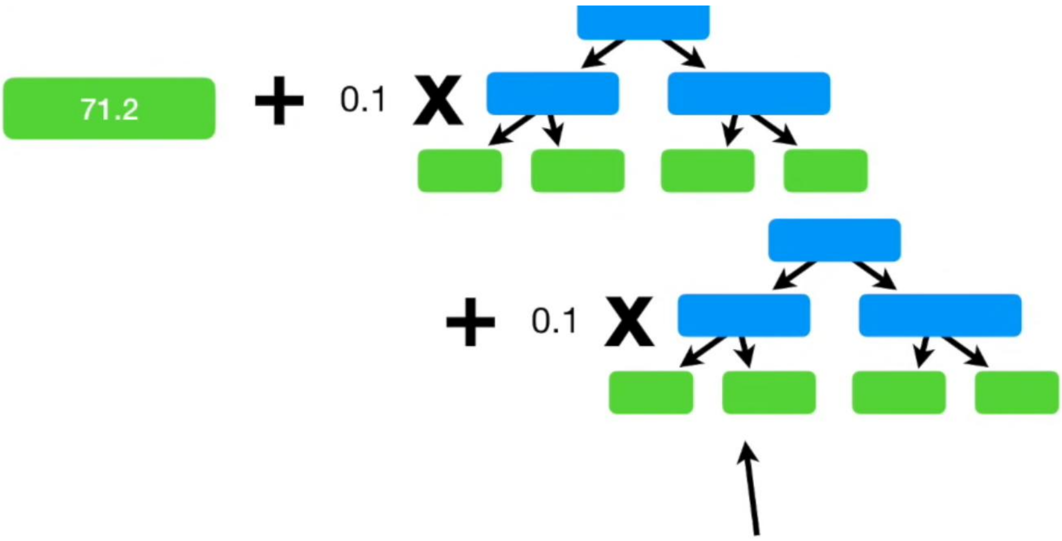
In this case, we wanted to predict **Weight**.



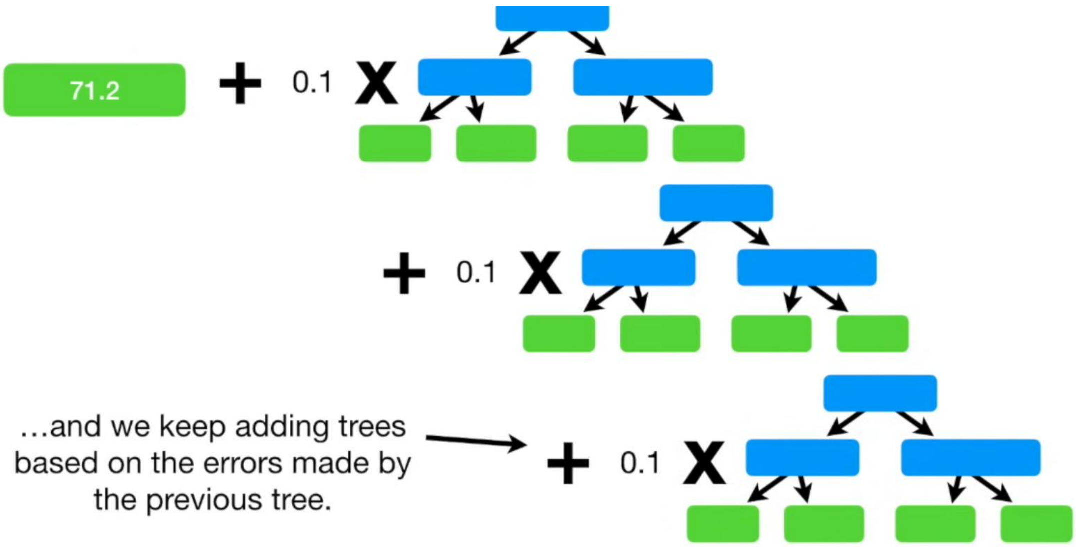
Then we add a tree based on the **Residuals**, the difference between the **Observed** values and the **Predicted** values…



…and we scale the tree's contribution to the final **Prediction** with a **Learning Rate**.



Then we add another tree based on the new **Residuals**…


71.2
+ 0.1 X
+ 0.1 X
…and we keep adding trees based on the errors made by the previous tree.
+ 0.1 X