https://www.youtube.com/watch?v=3CC4N4z3GJc

Gradient Boosting Moodel.

① GBM for regression

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

original data .

Weight (kg)
88
76
56
73
77
57

NOTE: When **Gradient Boost** is used to **Predict** a continuous value, like **Weight**, we say that we are using **Gradient Boost** for **Regression**.

Using **Gradient Boost** for **Regression** is different from doing **Linear Regression**, so while the two methods are related, don't get them confused with each other.

for continues model, it's gradient boost for regression

Input: Data $\{(x_i, y_i)\}_{i=1}^{n}$, and a differentiable **Loss Function** $L(y_i, F(x))$

Step 1: Initialize model with a constant value: $F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma)$

Step 2: for m = 1 to M:

(A) Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for

(B) Fit a regression tree to the r_{im} values and (regions R_{jm}, for $j = 1 \ldots J_m$

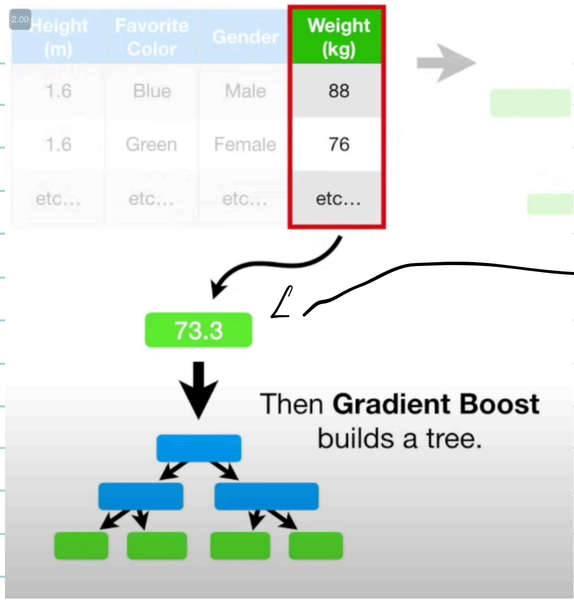
Part 2 in this series will dive deep into the math behind the **Gradient Boost** algorithm for **Regression**, walking through it step-by-step and proving that what we cover to day is correct.

(C) For $j = 1 \ldots J_m$ compute $\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

(D) Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_m I(x \in R_{jm})$

Step 3: Output $F_M(x)$

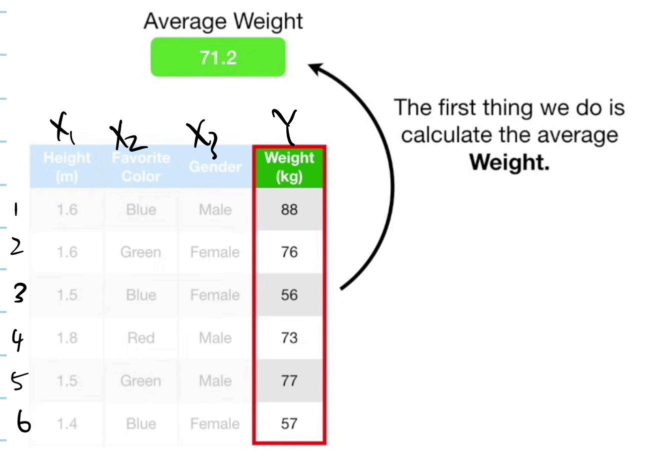
GB Algo



① GB starts with a single leaf

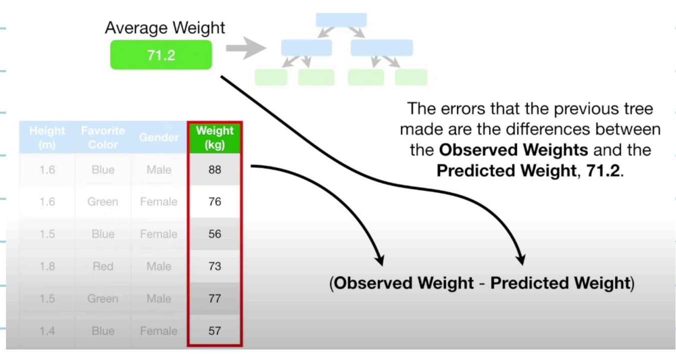
which is the average.

(usually maximum number of tree between 8~32)



	X_1 Height (m)	X_2 Favorite Color	X_3 Gender	Y Weight (kg)
1	1.6	Blue	Male	88
2	1.6	Green	Female	76
3	1.5	Blue	Female	56
4	1.8	Red	Male	73
5	1.5	Green	Male	77
6	1.4	Blue	Female	57

1. Calculate the Avg. $\bar{Y}=71.2$



2. we calculate the errors.

took mean as $\hat{Y}$.

then error is $Y_i - \hat{Y}$

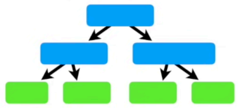
errors are the psedual residuals.

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

3. then we built a new tree with X_1, X_2, X_3 to predict the $\underset{\sim}{Y_R}$.

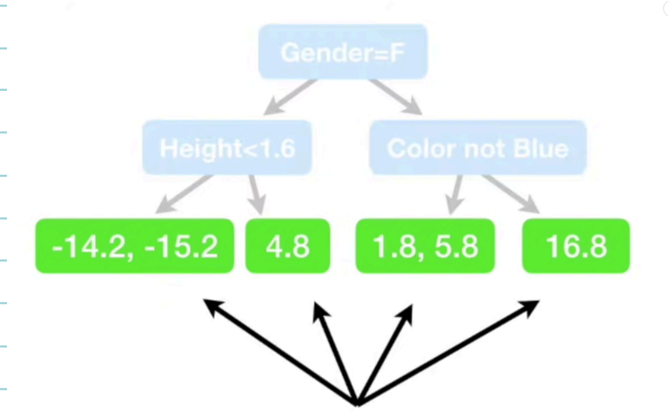
(distance)

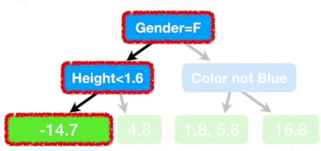
then we get a tree



Height (m) X_1	Favorite Color X_2	Gender X_3	Weight (kg)	Residual Y_R
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

If it seems strange to **Predict** the **Residuals** instead of the original **Weights**, just bear with me and soon all will become clear.



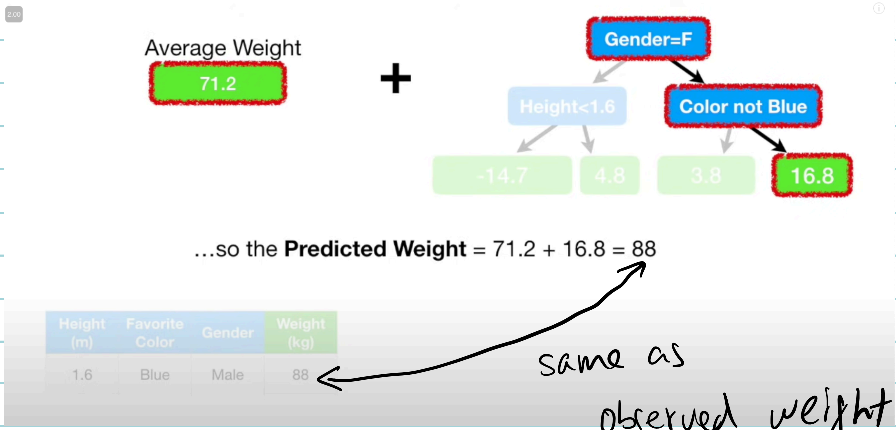


Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

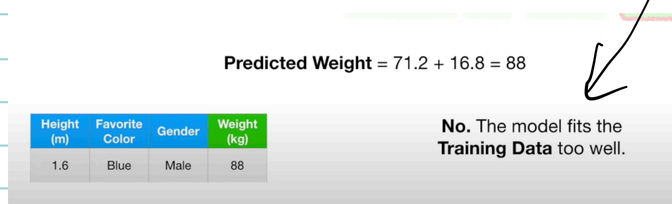
So we replace these residuals with their average.

$$\frac{(-14.2 + -15.2)}{2} = -14.7$$

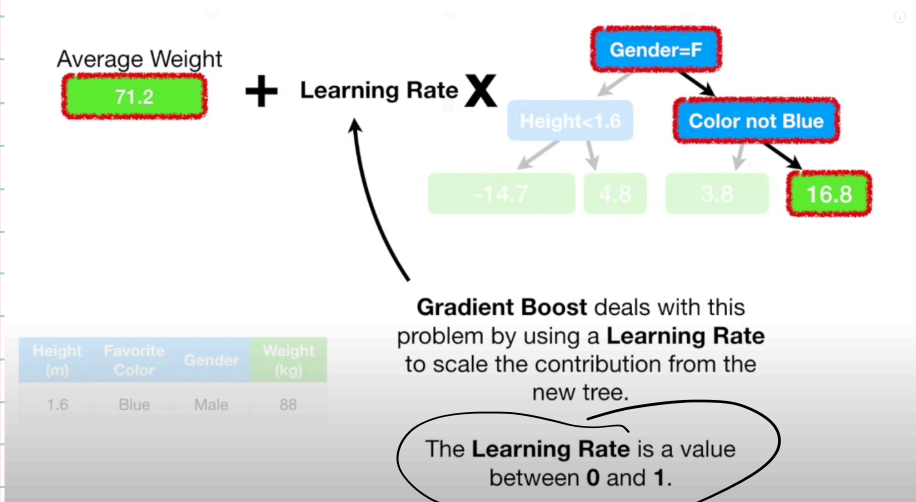
④ now combine original leaf with new tree to make prediction for individual obs from training data



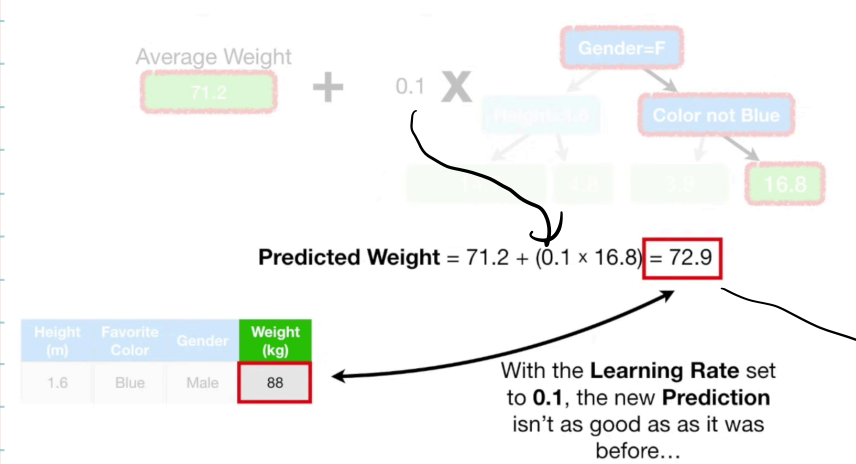
same as observed weight.



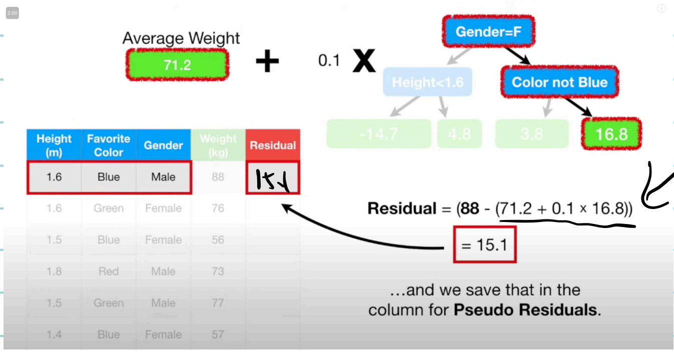
⑤ So we will scale the tree with learning rate.



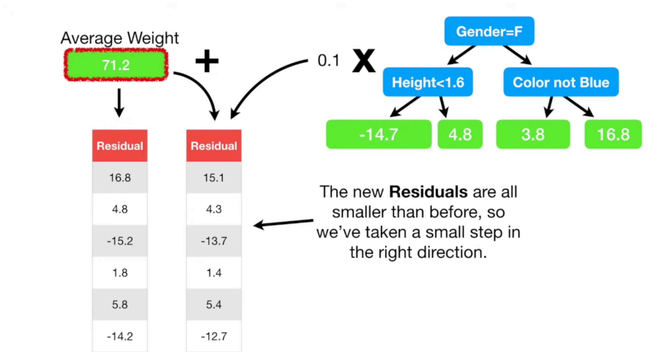
(with learning rate, it means a small step to the right direction)



So we calculate another tree :



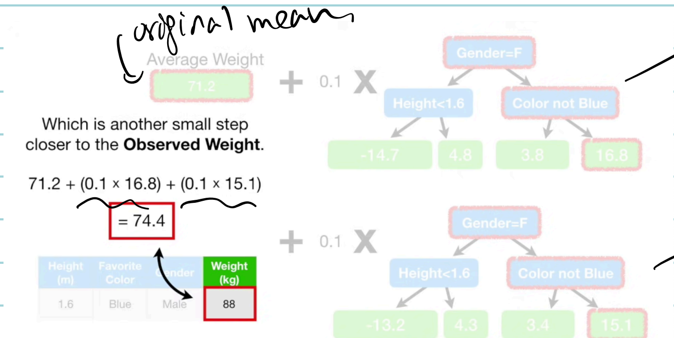
72.9 . 88 - 72.9 = 15.1



we repeat the process.

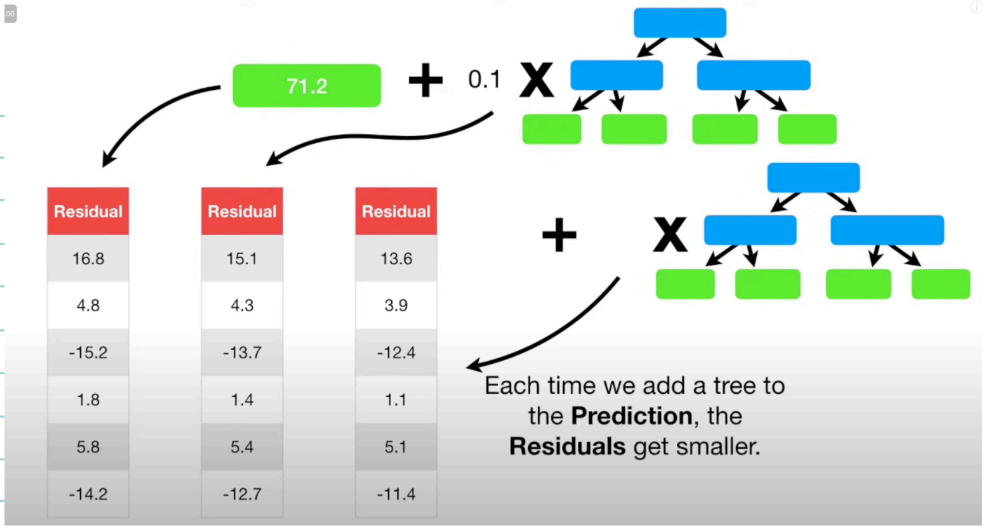
original mean

from first tree.

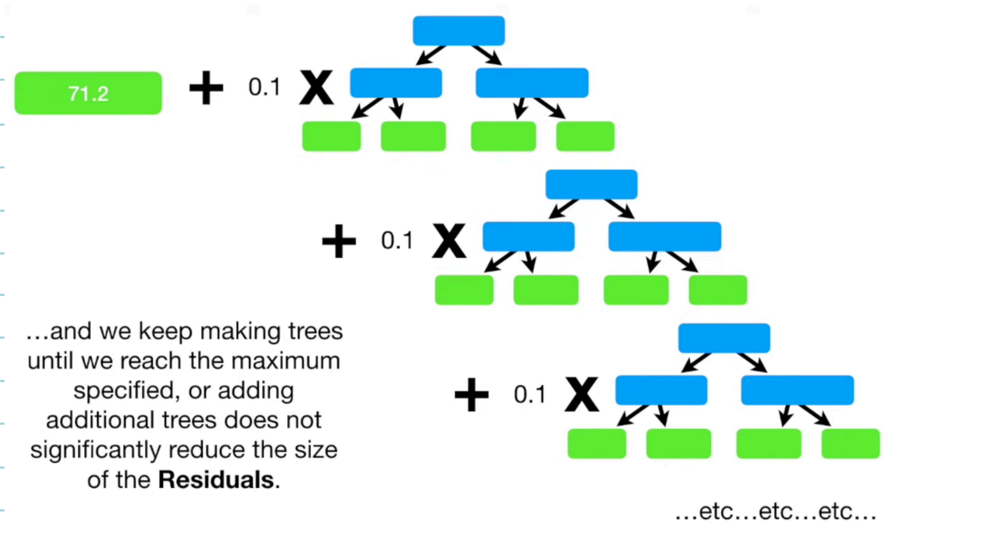


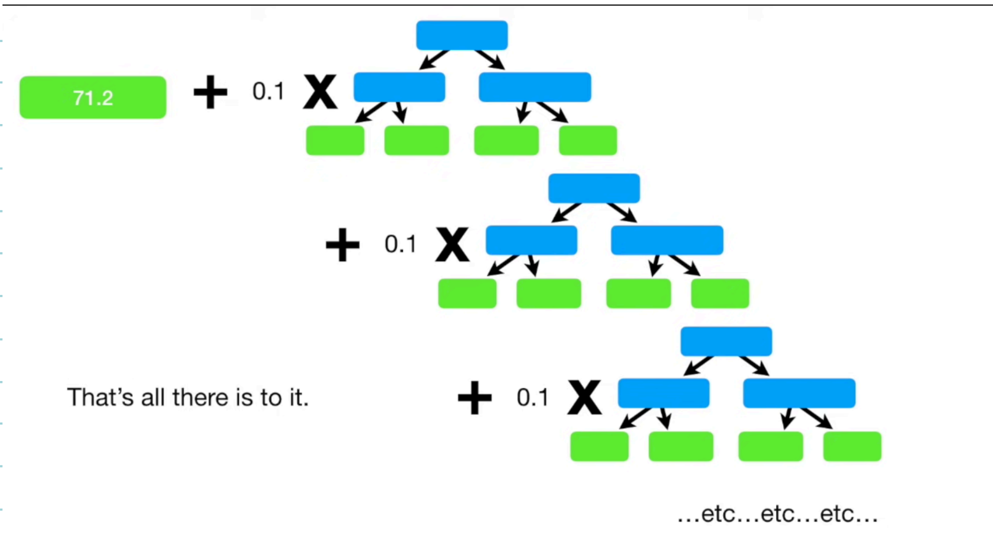
from second tree

another small step.



Residual	Residual	Residual
16.8	15.1	13.6
4.8	4.3	3.9
-15.2	-13.7	-12.4
1.8	1.4	1.1
5.8	5.4	5.1
-14.2	-12.7	-11.4

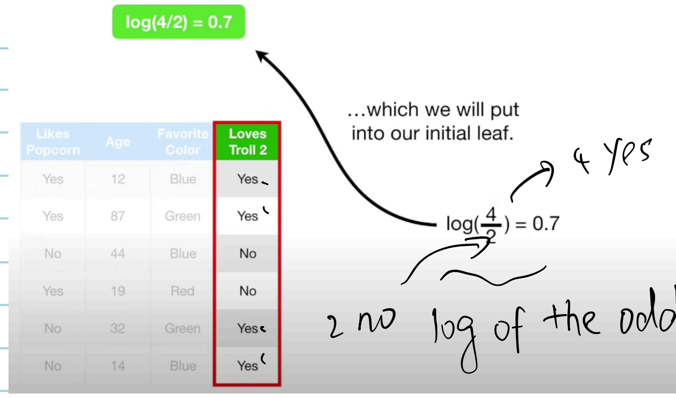




GB for classification.

Likes Popcorn	Age	Favorite Color	Loves Troll 2
Yes	12	Blue	Yes
Yes	87	Green	Yes
No	44	Blue	No
Yes	19	Red	No
No	32	Green	Yes
No	14	Blue	Yes

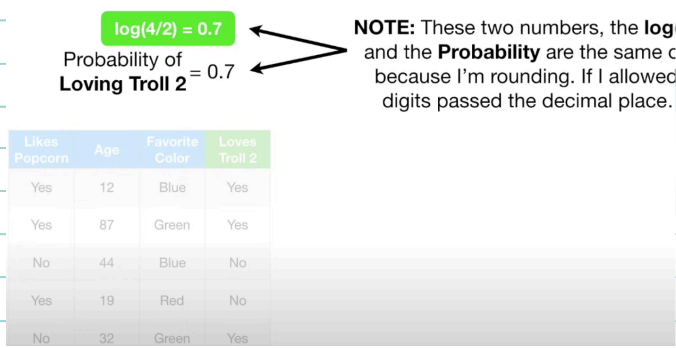
initial Prediction

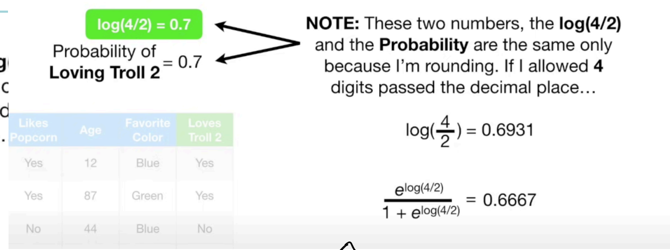


4 yes

2 no

log of the odds.

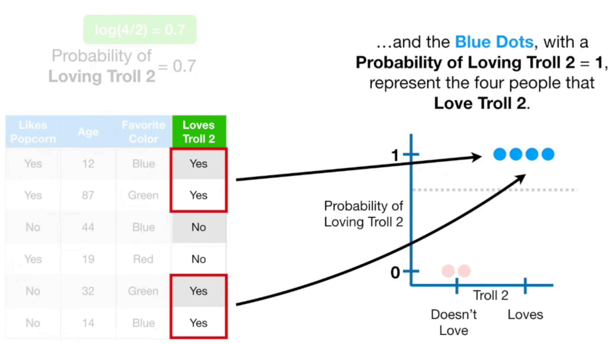


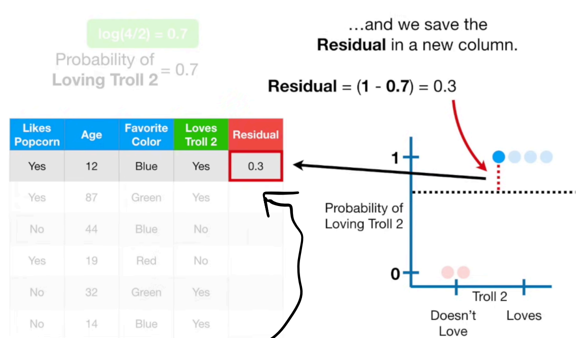


way of turning it to prob.

Since majority agree. call is yes.

so initial is yes

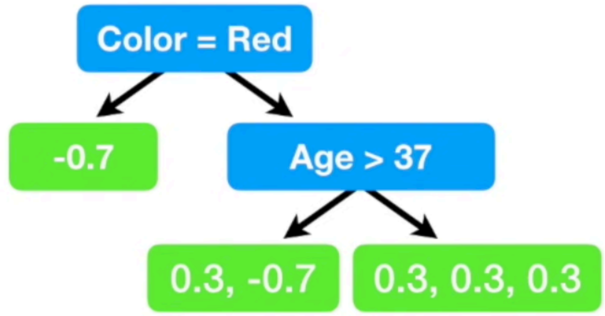




Residual = (Observed - Predicted)

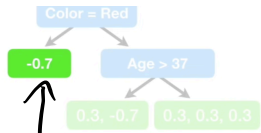
Likes Popcorn	Age	Favorite Color	Loves Troll 2	Residual
Yes	12	Blue	Yes	0.3
Yes	87	Green	Yes	0.3
No	44	Blue	No	-0.7
Yes	19	Red	No	-0.7
No	32	Green	Yes	0.3
No	14	Blue	Yes	0.3

then we build the tree.



NOTE: The derivation of this formula is quite technical, so I'm saving it for **Part 4** of this series when we get into the nitty gritty details of **Gradient Boost for Classification**.

$$\frac{\sum \text{Residual}_i}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)]}$$



$\log(\frac{4}{2}) = 0,7$

log(4/2) = 0.7

Probability of **Loving Troll 2** = 0.7

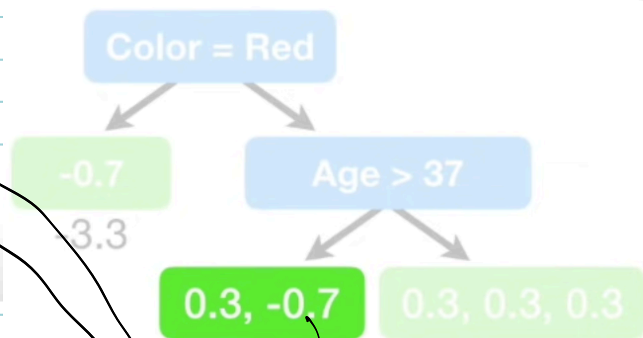
NOTE: These
and the **Prob**
because I'
digits pass

$$\frac{-0,7}{0,7 \times (1-0,7)} = -3,3$$

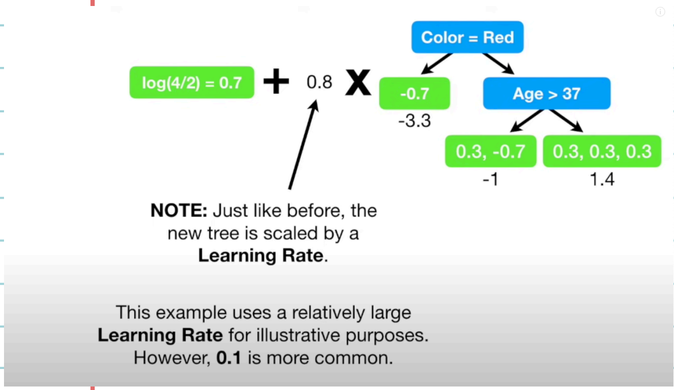
Likes Popcorn	Age	Favorite Color	Loves Troll 2
Yes	12	Blue	Yes
Yes	87	Green	Yes
No	44	Blue	No
Yes	19	Red	No
No	32	Green	Yes

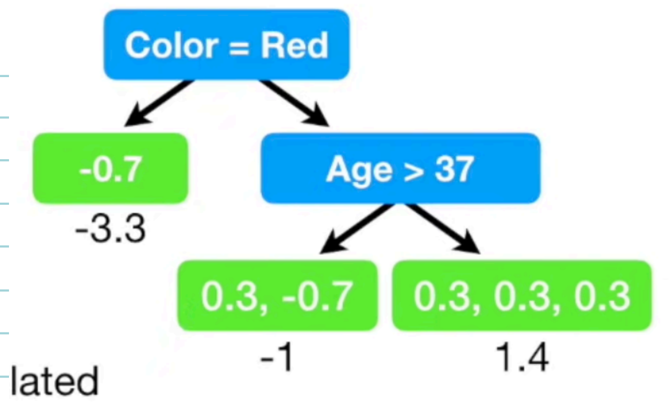
$\log(\frac{4}{2}) = 0,7$

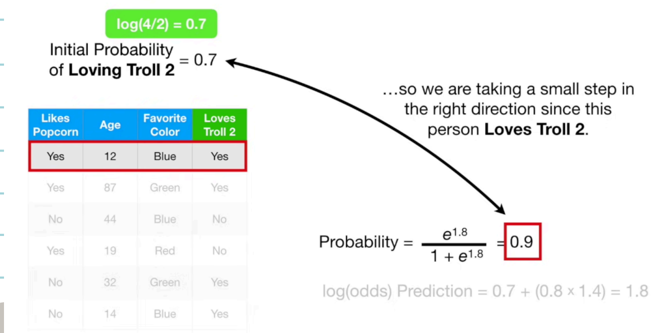
$$\frac{\sum \text{Residual}_i}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)]}$$

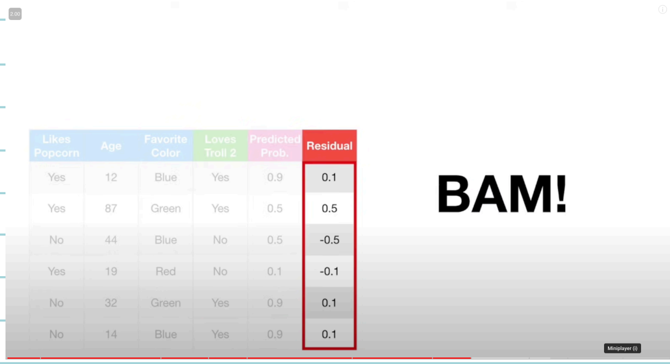


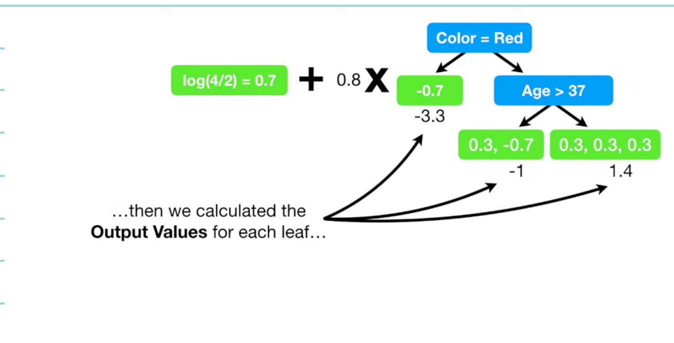
$$\frac{(0,3-0,7)}{(0,7 \times (1-0,7)) + (0,7 \times (1-0,7))} = -1$$

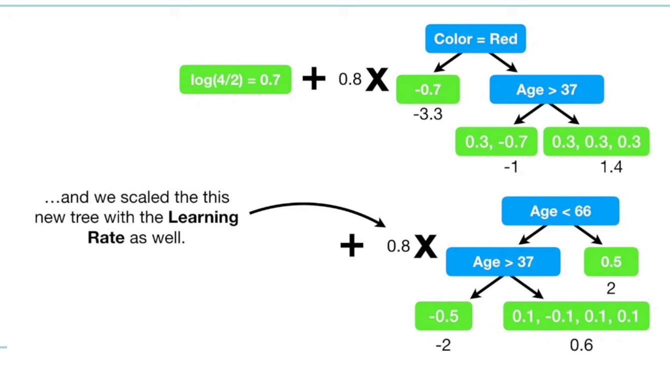

log(4/2) = 0.7
0.8
Color = Red
-0.7
-3.3
Age > 37
0.3, -0.7
-1
0.3, 0.3, 0.3
1.4
NOTE: Just like before, the new tree is scaled by a Learning Rate.
This example uses a relatively large Learning Rate for illustrative purposes. However, 0.1 is more common.

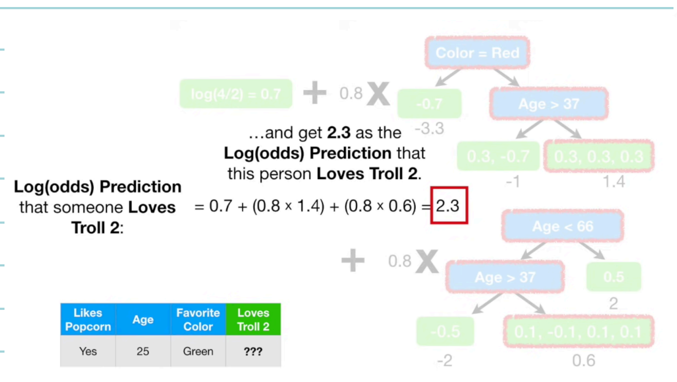

Color = Red
-0.7
-3.3
Age > 37
0.3, -0.7
-1
0.3, 0.3, 0.3
1.4
lated

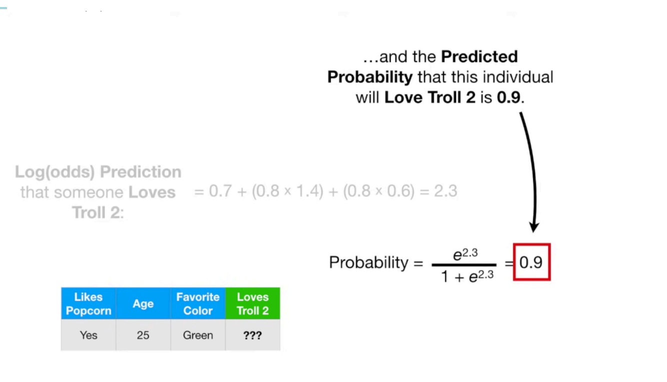

log(4/2) = 0.7
Initial Probability of Loving Troll 2 = 0.7
…so we are taking a small step in the right direction since this person Loves Troll 2.
Likes Popcorn | Age | Favorite Color | Loves Troll 2
Yes | 12 | Blue | Yes
Yes | 87 | Green | Yes
No | 44 | Blue | No
Yes | 19 | Red | No
No | 32 | Green | Yes
No | 14 | Blue | Yes
Probability = e^1.8 / (1 + e^1.8) = 0.9
log(odds) Prediction = 0.7 + (0.8 × 1.4) = 1.8


Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Predicted Prob. | Residual
Yes | 12 | Blue | Yes | 0.9 | 0.1
Yes | 87 | Green | Yes | 0.5 | 0.5
No | 44 | Blue | No | 0.5 | -0.5
Yes | 19 | Red | No | 0.1 | -0.1
No | 32 | Green | Yes | 0.9 | 0.1
No | 14 | Blue | Yes | 0.9 | 0.1
BAM!


log(4/2) = 0.7
0.8
Color = Red
-0.7
-3.3
Age > 37
0.3, -0.7
-1
0.3, 0.3, 0.3
1.4
…then we calculated the Output Values for each leaf…


log(4/2) = 0.7
0.8
Color = Red
-0.7
-3.3
Age > 37
0.3, -0.7
-1
0.3, 0.3, 0.3
1.4
…and we scaled the this new tree with the Learning Rate as well.
0.8
Age < 66
Age > 37
0.5
2
-0.5
-2
0.1, -0.1, 0.1, 0.1
0.6


…and get 2.3 as the Log(odds) Prediction that this person Loves Troll 2.
Log(odds) Prediction that someone Loves Troll 2:
= 0.7 + (0.8 × 1.4) + (0.8 × 0.6) = 2.3
Likes Popcorn | Age | Favorite Color | Loves Troll 2
Yes | 25 | Green | ???


…and the Predicted Probability that this individual will Love Troll 2 is 0.9.
Log(odds) Prediction that someone Loves Troll 2:
= 0.7 + (0.8 × 1.4) + (0.8 × 0.6) = 2.3
Probability = e^2.3 / (1 + e^2.3) = 0.9
Likes Popcorn | Age | Favorite Color | Loves Troll 2
Yes | 25 | Green | ???