Gradient Boost

Gradient Boost 是其中一种 Boosting 的方法，Gradient Boost 能够解决 Regression 和 Classification 的问题。这里需要注意，Gradient Boost for Regression 和 Linear Regression 是不同的方法。Gradient Boost 使用了 Decision Tree 的原理还有 AdaBoost 的原理。

Gradient Boost for Regression

在使用 Gradient Boost 前，需要准备好一组 Dataset。

$$Data = \{(x_i, y_i\}_{i=1}^{n}$$

$$x \rightarrow Input\ Data\ Variables$$

$$y \rightarrow Output\ Data\ Labels$$

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56

从上图来看，$x_1 = \{1.6, Blue, Male\}, x_2 = \{1.6, Green, Female\}, x_3 = \{1.5, Blue, Female\}$，而 $y_1 = 88, y_2 = 76, y_3 = 56$。当 $i = 1$的时候代表着 Dataset 里面的第一行数据，n 代表着最后一行数据。

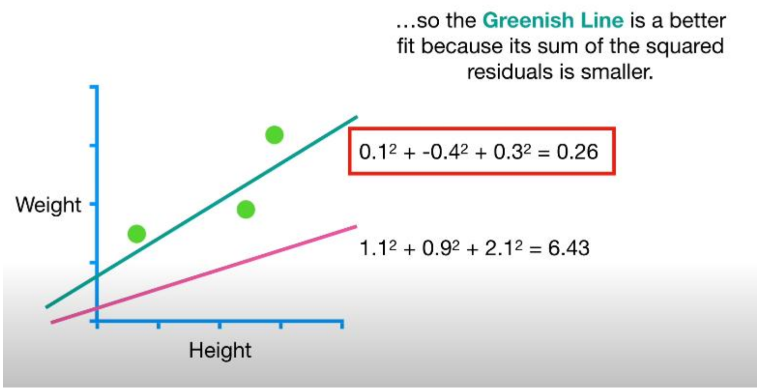
这时候需要有 Loss Function，Loss Function 需要是一个能够被 Differentiate 的 Function，用来定义 Observed Value 与 Predicted Value 的差别有多大。在 Gradient Boost for Regression 里，最长被使用到的 Loss Function 是 Residual Loss。

$$Loss\ Function = L\big(y_i, F(x)\big) = \frac{1}{2}(Observed - Predicted)^2$$

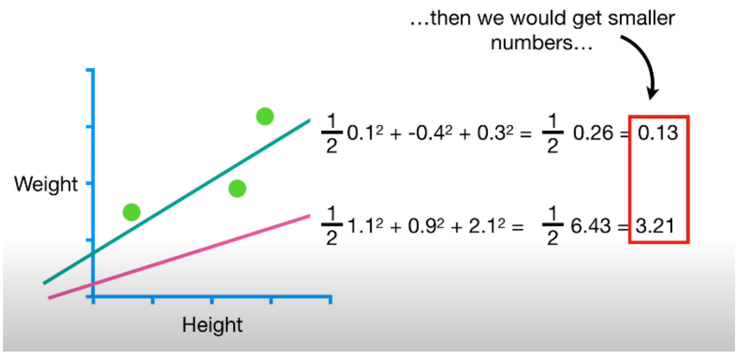
$$y_i \rightarrow Observed\ Value$$

$$F(x) \rightarrow Predicted\ Value$$

这个 Equation 前面乘上了 0.5 是为了方便 Differentiation 后的 Equation，乘上 0.5 只会让算出来的 Loss 值变小，但是不会影响计算结果。



上图是当 Loss Function 没有乘上 0.5。



上图是乘上 0.5 之后的 Loss 值，只是算出来的值变小，但是不影响选择。

Step 1

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$$

$$L(y_i, \gamma) = \frac{1}{2}(Observed - Predicted)^2$$

$\arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$ 代表着需要找出一个 Predicted 值能够让这个 Summation of Loss Function 有着最小的值。可以通过计算出 Loss Function 的 Derivative 值让后让这个 Equation 等于 0 。

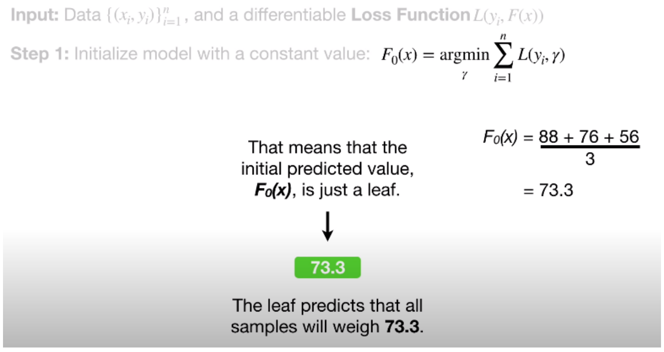
$$\frac{d\frac{1}{2}(Observed - Predicted)^2}{dPredicted} = -(Observed - Predicted)$$

Example:

$$-(88 - Predicted) + -(76 - Predicted) + -(56 - Predicted) = 0$$

$$Predicted = \frac{88 + 76 + 56}{3} = 73.33$$

这里可以发现，计算出的值是所有 y (Observed Weights) 的平均值，而这个值是能够让 Loss Function 最小的值。



$$Initialize\ Model \rightarrow F_0(x) = 73.33$$

当计算出这个值之后，也就是 Gradient Boost 的第一颗树完成。Gradient Boost 的第一棵树只有一颗叶子，而这个叶子的值就是所有 Observed Weights 加起来然后取平均。这时候模型给出的预测值都是 73.33，不管 Input 是什么。

Step 2

$$For\ m = 1\ to\ M$$

$$m \rightarrow Individual\ Tree$$

$$M \rightarrow Number\ of\ Tree\ Needed$$

在 Step 2 里，需要设定要创建多少颗树，然后就一直重复计算后创建不同的树，直到所有树创建完成，一般的情况会创建至少 100 课树。

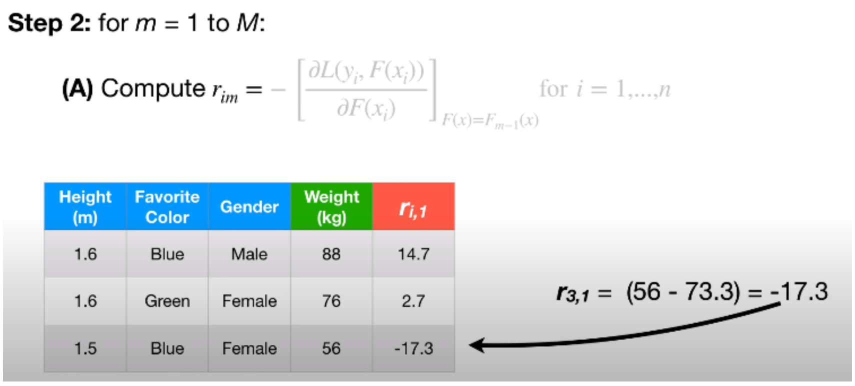
$$Pseudo\ Residuals \rightarrow r_{im} = -\left[\frac{\partial \mathrm{L}\left(y_i, F(x_i)\right)}{\partial \mathrm{F}(\mathrm{x_i})}\right]_{F(x)=F_{m-1}(x)} \quad for\ i = 1 \ldots n$$

$$r_{im} \rightarrow r = \ Pseudo\ Residual, i = Sample\ Number, m = Tree\ Trying\ to\ Build$$

$$\frac{\partial \mathrm{L}\left(y_i, F(x_i)\right)}{\partial \mathrm{F}(\mathrm{x_i})} \rightarrow Derivative\ of\ Loss\ Function = \ -(Observed - Predicted)$$

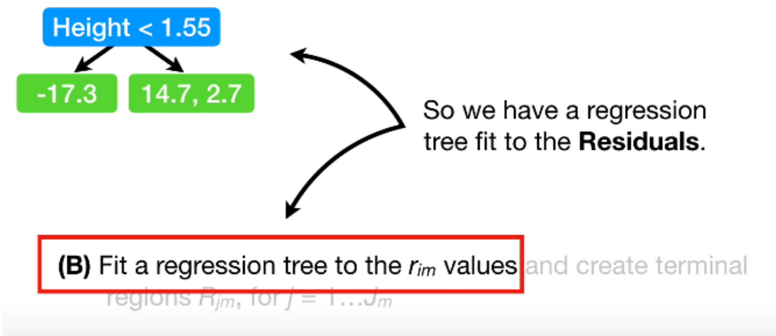
$$(Observed - Predicted) \rightarrow \left(Observed - F_{m-1}(x)\right)$$

当 $m = 1$ 的时候，计算时使用 $F_{m-1}(x) = F_0(x) = 73.33$



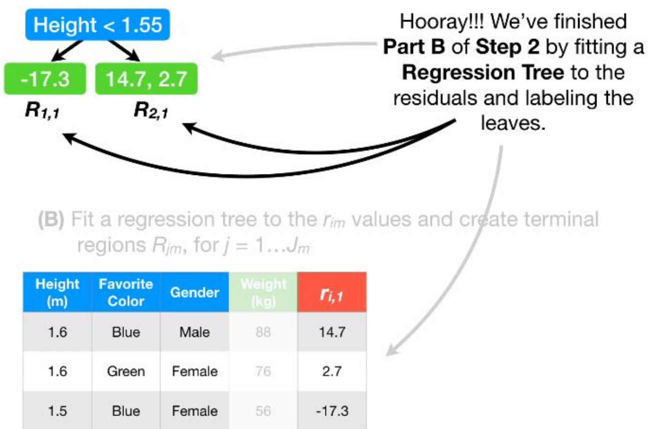
Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$
1.6	Blue	Male	88	14.7
1.6	Green	Female	76	2.7
1.5	Blue	Female	56	-17.3

当计算完 $r_{i,1}$ 之后，就能创建一棵树来预测 Pseudo Residuals 的值，同样是使用了 Dataset 里面的所有 Parameters (Height, Favorite Color, Gender)。在 Gradient Boost 里创建的树一般都不是 Stump，而是能有 8 到 32 个叶子的树。



在这里因为 Dataset 太小所以使用了 Stump 来解释。当树创建好后，需要对每个叶子(Terminal Region 终端区) 进行 Label $R_{j,m}$。

$$Terminal\ Region \rightarrow R_{j,m}\ for\ j = 1 \ldots J_m$$



当 Label 好每一个叶子之后，就能进行计算该叶子的 Predicted 值，如上图，在一个叶子出现多过一个 Predicted 值的时候，需要对最终的 Predicted 值进行计算。

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)\ \ for\ j = 1 \ldots J_m$$

这里要计算的是每个叶子需要 Predicted 的值是有着最低 Loss 的值，这个 Equation 与 Step 1 的很相似，但是在这里需要加上一个 γ (Previous Prediction) 值，而 $x_i \in R_{ij}$ 代表着每一次只对该叶子里面的所有值进行计算，不考虑其他叶子的值。

Example:

要对上图右边的叶子进行计算。

$$\gamma_{2,1} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} \frac{1}{2}\left(y_i - (F_{m-1}(x_i) + \gamma)\right)^2$$

$$\gamma_{2,1} = \arg\min_{\gamma} \left[\frac{1}{2}\left(88 - (F_{m-1}(x_1) + \gamma)\right)^2 + \frac{1}{2}\left(76 - (F_{m-1}(x_2) + \gamma)\right)^2\right]$$

这时候的$F_{m-1}(x)$ 代表着上一棵树的 Output 值。

$$\gamma_{2,1} = \arg\min_{\gamma} \left[\frac{1}{2}\left(88 - (73.33 + \gamma)\right)^2 + \frac{1}{2}\left(76 - (73.33 + \gamma)\right)^2\right]$$

$$\gamma_{2,1} = \arg\min_{\gamma} \left[\frac{1}{2}(14.7 - \gamma)^2 + \frac{1}{2}(2.7 - \gamma)^2\right]$$

要计算当前这个叶子的值能够让 Loss Function 有着最低值的值，就需要对其 Equation 进行 Derivative 的计算然后让这个 Equation 等于 0，来找出最低的 γ 值。

$$\frac{\partial}{\partial \gamma}\left[\frac{1}{2}(14.7 - \gamma)^2 + \frac{1}{2}(2.7 - \gamma)^2\right] = -14.7 + \gamma + -2.7 + \gamma$$

$$-14.7 + \gamma + -2.7 + \gamma = 0$$

$$\gamma_{2,1} = \frac{14.7 + 2.7}{2}$$

这里可以发现就是求这个叶子里所有值的平均。当计算好第二棵树之后，需要对 Prediction Equation 进行 Update。

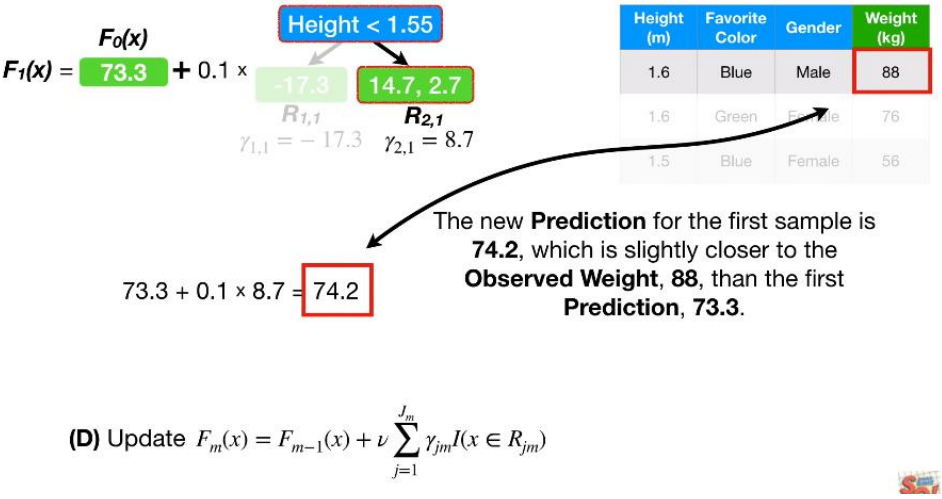
$$F_m(x) = F_{m-1}(x) + \upsilon \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

$$F_{m-1}(x) \rightarrow Last\ Prediction$$

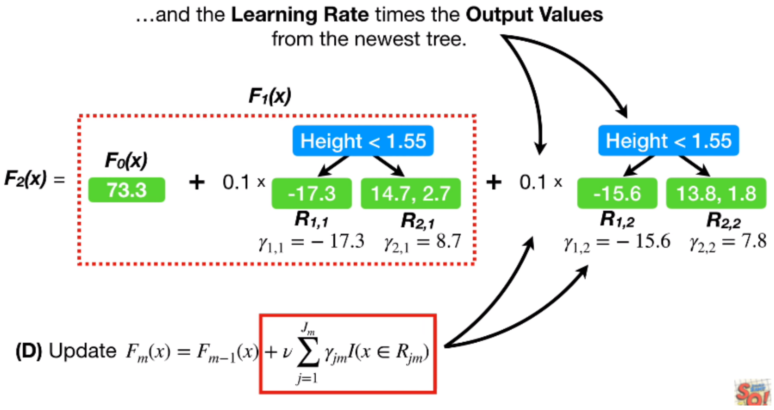
$$\upsilon \rightarrow Learning\ Rate\ (between\ 0\ to\ 1)$$

$$\sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm}) \rightarrow Prediction\ on\ Current\ Tree$$

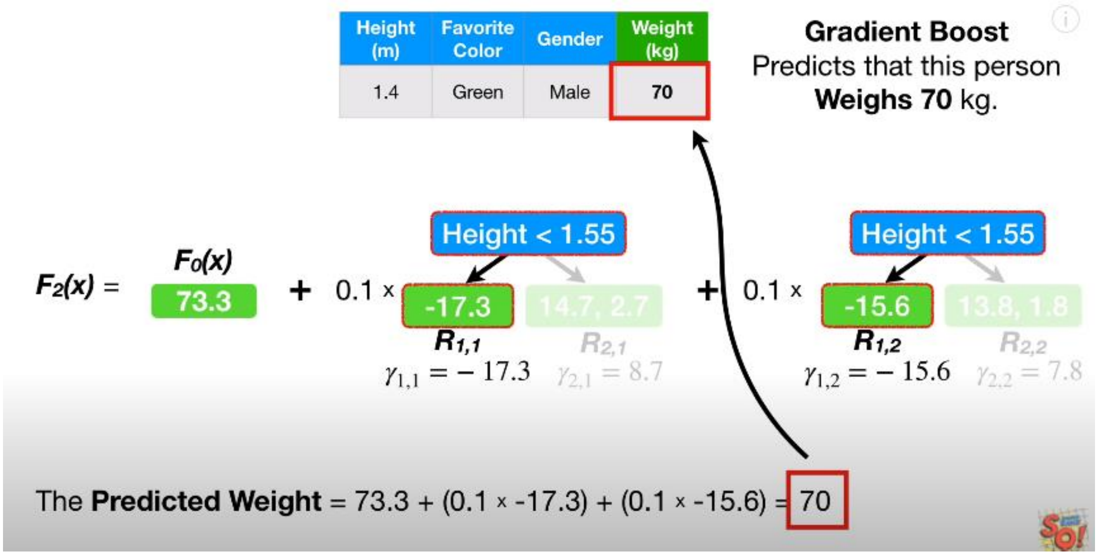
Prediction on Current Tree 有着 Summation 的符号是为了避免当 Single Sample End Up in Multiple Leaves 的情况。



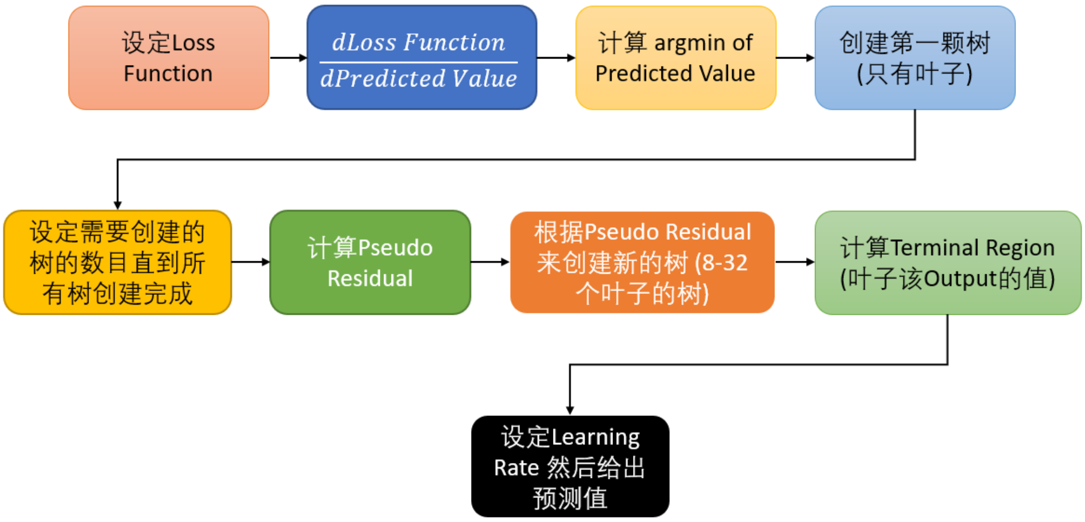
当设定好了 Learning Rate 之后就能进行预测，Gradient Boost 的预测将是每一颗树的总和。



上图是当 $m = 2$ 的列子，就是先前算过的预测值再加上当前预测值的总和。



小总结


设定Loss Function
dLoss Function / dPredicted Value
计算 argmin of Predicted Value
创建第一颗树 (只有叶子)
设定需要创建的树的数目直到所有树创建完成
计算Pseudo Residual
根据Pseudo Residual 来创建新的树 (8-32 个叶子的树)
计算Terminal Region (叶子该Output的值)
设定Learning Rate 然后给出预测值

Gradient Boost for Classification

再 Gradient Boost for Classification 里，需要先准备一组 Dataset。

$$Data = \{(x_i, y_i)\}_{i=1}^{n}$$

$$x \rightarrow Input\ Data\ Variables$$

$$y \rightarrow Output\ Data\ Labels$$

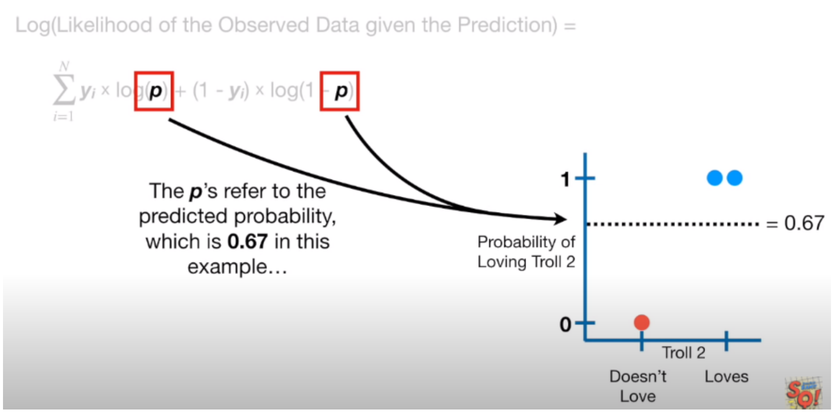
Likes Popcorn	Age	Favorite Color	Loves Troll 2
Yes	12	Blue	Yes
No	87	Green	Yes
No	44	Blue	No

从上图来看，$x_1 = \{Yes, 12, Blue\}, x_2 = \{No, 87, Green\}, x_3 = \{No, 44, Blue\}$，而$y_1 = Yes, y_2 = Yes, y_3 = No$。当 $i = 1$的时候代表着 Dataset 里面的第一行数据，n 代表着最后一行数据。

这时候需要有 Loss Function，Loss Function 需要是一个能够被 Differentiate 的 Function，用来定义 Observed Value 与 Predicted Value 的差别有多大。在 Gradient Boost for Classification 里，最长被使用到的 Loss Function 是 Log Likelihood。这个 Log likelihood 的值越小越好，越小就代表 Predicted Value 与 Actual Value 越接近。

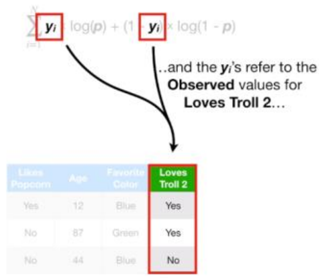
$$\boldsymbol{Log\ Likelihood = -\sum_{i=1}^{n} y_i \log(p) + (1 - y_i) \log(1 - p)}$$

$$p \rightarrow Predicted\ Probability$$



从上图的计算 $p = 0.67$，将仅有的 3 笔 Data 的 y 加起来然后取平均，

$$p = \frac{1 + 1 + 0}{3} = 0.67$$

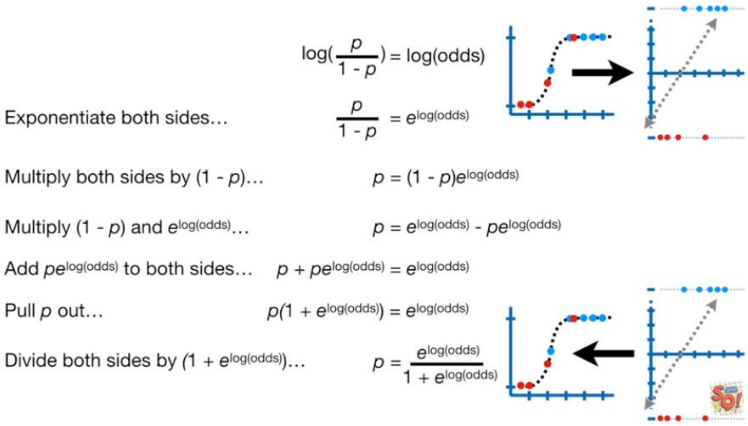


而 y_i 是指 Dataset 的 Output，在 Binary Classification 里，Output 只有 0 和 1。当 Output 是 Yes 的时候就代表着$y_i = 1$，而 Loss Function 就是 $Loss\ Function = \log(p)$。当 Output 是 No 的时候就代表着$y_i = 0$ 而 Loss Function 就是 $Loss\ Function = \log(1-p)$。

从上图的 Dataset 来看，算出来的**第一次**的 Loss 就是:

$$Sum\ of\ Loss = -[\log(0.67) + \log(0.67) + \log(1 - 0.67)]$$

简单化 Loss Function 是为了容易做 Derivative 与计算。



$$odds \rightarrow Ratio\ of\ \frac{Something\ Happening}{Something\ Not\ Happening}$$

$$Probability \rightarrow \frac{Something\ Happening}{Something\ Happening + Something\ Not\ Happening}$$

$$Odds = \frac{p}{1-p} \rightarrow A\ Method\ To\ Calculate\ Odds\ From\ Probability$$

$$\log(odds) = \log\left(\frac{p}{1-p}\right) \rightarrow Makes\ Thing\ Symmetrically\ (对称)\ with\ Log$$

$$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

$$\log(1 - p) = \log\left(1 - \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}\right) = \log\left(\frac{1 + e^{\log(odds)}}{1 + e^{\log(odds)}} - \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}\right) = \log\left(\frac{1}{1 + e^{\log(odds)}}\right)$$

$$= \log(1) - \log(1 + e^{\log(odds)}) = -\log(1 + e^{\log(odds)})$$

$$\log(1-p) = \log\left(\frac{1}{1 + e^{\log(odds)}}\right) = -\log\left(1 + e^{\log(odds)}\right)$$

$$Log\ Likelihood = -y_i \log(p) - (1 - y_i)\log(1-p) = -y_i \log(p) - log(1-p) + y_i log(1-p)$$

$$-y_i(\log(p) - \log(1-p)) - \log(1-p)$$

$$-y_i \log\left(\frac{p}{1-p}\right) + \log\left(1 + e^{\log(odds)}\right)$$

$$y_i \rightarrow Observed\ Value$$

$$\boldsymbol{Simplified\ Log\ Likelihood = -Observed * \log(odds) + \log\left(1 + e^{\log(odds)}\right)}$$

Derivative of Loss Function

$$\frac{d - \boldsymbol{Observed * \log(odds) + \log\left(1 + e^{\log(odds)}\right)}}{dlog(odds)} = -Observed + \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

$$\boldsymbol{1st\ Loss\ Derivative \rightarrow -Observed + \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} = -Observed + p}$$

Second Derivative of Loss Function

$$\frac{d}{dlog(odds)} \frac{d}{dlog(odds)} - Observed * \log(odds) + \log\left(1 + e^{log(odds)}\right)$$

$$\frac{d}{dlog(odds)} - Observed + \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

$$\boldsymbol{\frac{d}{dlog(odds)} - Observed + \left(1 + e^{\log(odds)}\right)^{-1} * e^{\log(odds)}}$$

Apply Product Rule & Chain Rule

$$Product\ Rule \rightarrow a * b = a' * b + a * b'$$

$$\frac{d^2L'}{dlog(odds)^2} = -\left(1 + e^{\log(odds)}\right)^{-2} e^{\log(odds)} * e^{\log(odds)} + \left(1 + e^{\log(odds)}\right)^{-1} * e^{\log(odds)}$$

$$\frac{d^2L}{dlog(odds)^2} = -\frac{e^{\log(odds)^2}}{(1 + e^{\log(odds)})^2} + \frac{e^{\log(odds)}}{(1 + e^{\log(odds)})} * \frac{\left(1 + e^{\log(odds)}\right)}{\left(1 + e^{\log(odds)}\right)}$$

$$\frac{d^2L}{dlog(odds)^2} = \frac{e^{\log(odds)}}{(1 + e^{\log(odds)})^2} = \frac{e^{\log(odds)}}{(1 + e^{log(odds)})(1 + e^{log(odds)})}$$

$$\boldsymbol{Second\ Loss\ Derivative \rightarrow \frac{d^2L}{dlog(odds)^2} = \frac{e^{log(odds)}}{(1 + e^{log(odds)})} * \frac{1}{(1 + e^{\log(odds)})} = p * (1 - p)}$$

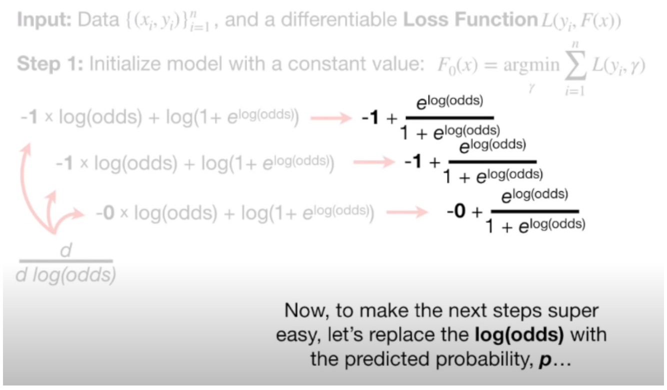
Step 1

当了解了 Gradient Boost for Classification 的 Data 和 Loss Function 之后，就能开始建立 Gradient Boost for Classification。在 Step 1 需要 Initialize Model with a Constant Value $F_0(x)$，也就是第一个 Leaf。

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$$

首先需要计算能给出最低的 Sum 的 $\log(odds)$ 的值，就需要使用到 First Derivative Equation，然后让这个 Equation 等于 0，之后求 log(odds)。

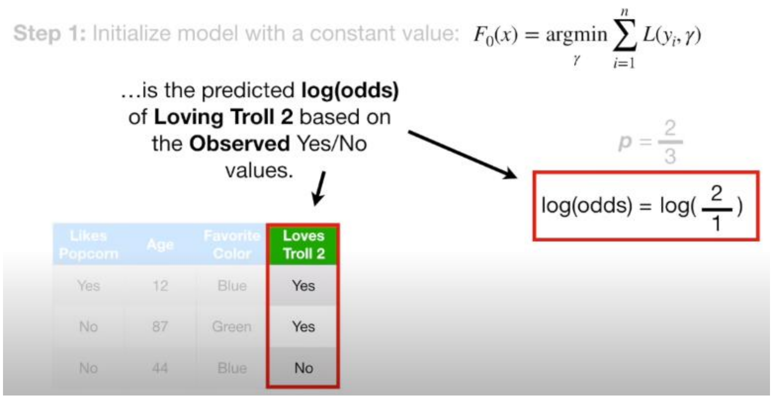
$$-Observed + \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} = -Observed + p$$



$$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

$$-1 + p - 1 + p - 0 + p = 0$$

$$p = \frac{2}{3} \rightarrow \frac{2\ Yes}{Total\ 3\ Data}$$



$$\log(odds) = \log\left(\frac{\frac{2}{3}}{1 - \frac{2}{3}}\right) = \log\left(\frac{2}{1}\right) \rightarrow \frac{2\ Yes}{1\ No}$$

$$F_0(x) = \log\left(\frac{2}{1}\right) = 0.69$$

Step 2

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 $i = 1,\ldots,n$

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

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

上图是在 Gradient Boost for Classification 的 Step 2 里需要做的步骤。首先先要计算 Loss Function 的 Derivative。

$$r_{im} = -\left[\frac{\partial \mathrm{L}\left(y_i, F(x_i)\right)}{\partial \mathrm{F}(\mathrm{x_i})}\right]_{F(x)=F_{m-1}(x)} = -\frac{d - Observed * \log(odds) + \log\left(1 + e^{\log(odds)}\right)}{d\log(odds)}$$

$$\boldsymbol{r_{im} = -\left[\frac{\partial \mathrm{L}\left(y_i, F(x_i)\right)}{\partial \mathrm{F}(\mathrm{x_i})}\right]_{F(x)=F_{m-1}(x)} = Observed - \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} = Observed - p}$$

$$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

$$r_{im} \rightarrow r = Pseudo\ Residual, i = Sample\ Number, m = Tree\ Trying\ to\ Build$$

继续上面的 Example

$$F_0(x) = \log\left(\frac{2}{1}\right) = 0.69$$

$$r_{im} = Observed - \frac{e^{\log\left(\frac{2}{1}\right)}}{1 + e^{\log\left(\frac{2}{1}\right)}} = Observed - \frac{2}{3}$$

$$Residual\ for\ 1st\ \&\ 2nd\ Sample\ \&\ 1st\ Tree\ R_{1,1} = (Observed - 0.67) = 1 - 0.67 = 0.33$$

$$Residual\ for\ 3rd\ Sample\ \&\ 1st\ Tree\ R_{1,1} = (Observed - 0.67) = 0 - 0.67 = -0.67$$

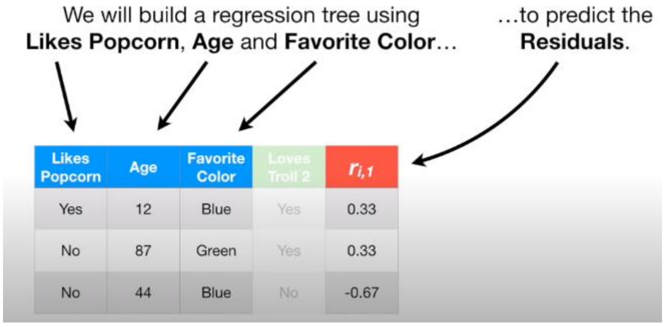
Step 2: for $m = 1$ to M:

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

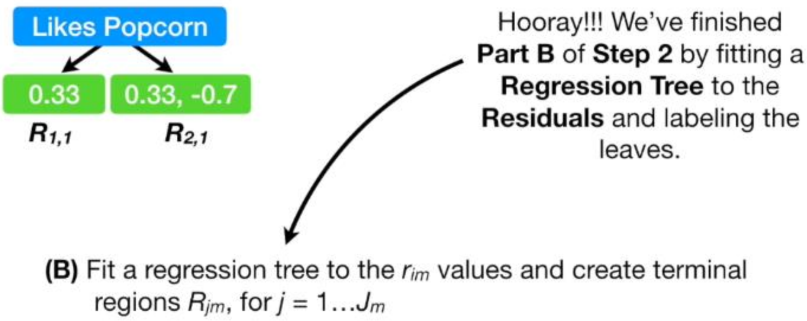
Likes Popcorn	Age	Favorite Color	Loves Troll 2	$r_{i,1}$
Yes	12	Blue	Yes	0.33
No	87	Green	Yes	0.33
No	44	Blue	No	-0.67

Hooray! We've finished **Part A** of **Step 2** by calculating a **Residual** for each sample.

计算完所有 r_{i1} 之后，就会得到如上图一样的数据，就是第一颗树的 Pseudo Residual (Actual 与 $log(odds)$ 之间的差别)。这时候就能创建第一颗树。

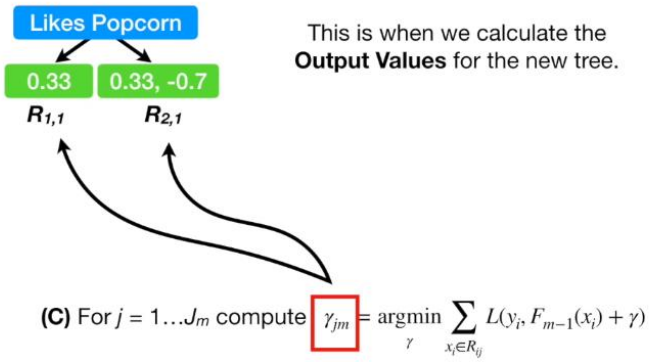


这时候就使用 Dataset 里面的所有 Variables (Parameters) 与刚算出的第一颗树的 Pseudo Residual 来创建第一颗树。在 Gradient Boost 里，创建的树通常是在 8 到 32 片叶子，由于这里的 Example Dataset 太小，所以创建出的树只有 2 个叶子 (Stump)。



创建出第一棵树之后，将每一个叶子命名为R_{jm}，如上图。

$$Creates\ Terminal \rightarrow R \rightarrow Residual, j \rightarrow Number\ of\ leaves, m \rightarrow Number\ of\ Tree$$



$$Output\ Value\ for\ New\ Tree \rightarrow \gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, \boldsymbol{F_{m-1}(x_i) + \gamma})$$

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} -y_i * [F_{m-1}(x_i) + \gamma] + \log(1 + e^{F_{m-1}(x_i) + \gamma})$$

在这里计算的 γ_{jm} 会考虑到之前的 $F_{m-1}(x_i)$ 值。γ 代表着当前树的 Output 值，$F_{m-1}(x_i)$ 代表着之前所有树与首个叶子总和的 Output 值。

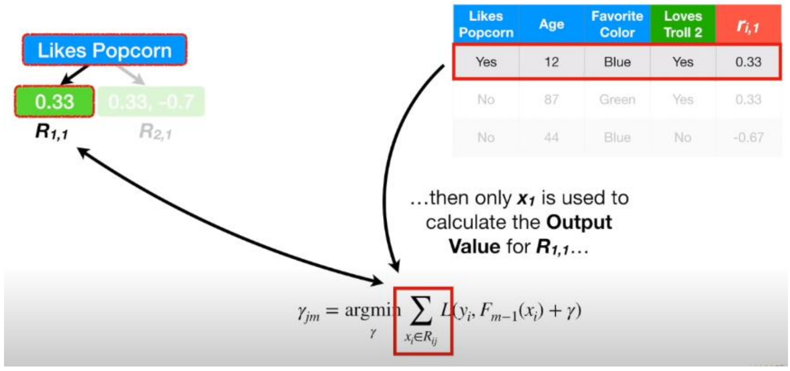
Second Order Taylor Polynomial

$$L(y_i, F_{m-1}(x_i) + \gamma) = -y_i * [F_{m-1}(x_i) + \gamma] + \log\left(1 + e^{F_{m-1}(x_i)+\gamma}\right)$$

$$\boldsymbol{L(y_i, F_{m-1}(x_i) + \gamma) \approx L(y_i, F_{m-1}(x_i)) + \frac{d}{dF()}(y_i, F_{m-1}(x_i))\gamma + \frac{1}{2}\frac{d^2}{dF()^2}(y_i, F_{m-1}(x_i))\gamma^2}$$

$$\boldsymbol{\frac{d}{d\gamma} L(y_i, F_{m-1}(x_i) + \gamma) \approx \frac{d}{dF()}(y_i, F_{m-1}(x_i)) + \frac{d^2}{dF()^2}(y_i, F_{m-1}(x_i))\gamma}$$

创建 Terminal 之后，就能计算每一个叶子该 Output 的机率也就是计算 γ_{jm}。$\sum_{x_i \in R_{ij}}$ 代表着每一个叶子里面的所有 Data。



也就是R_{jm}里面的所有值，如上图，$R_{1,1} = \{0.33\}$ 而 $R_{2,1} = \{0.33, -0.7\}$。

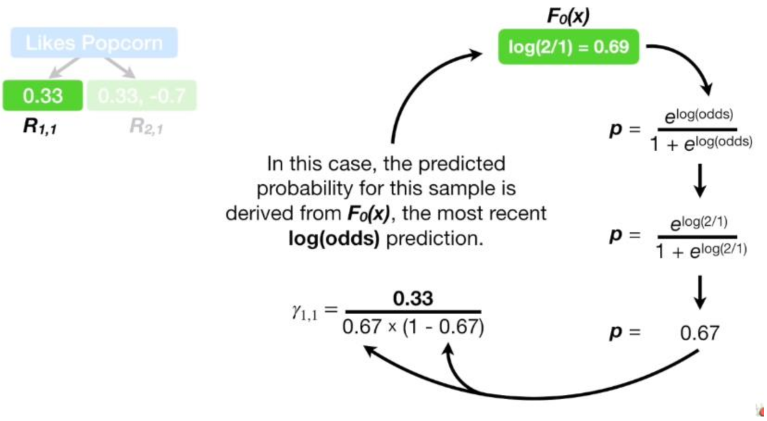
$$\frac{d}{d\gamma} L(y_1, F_{m-1}(x_1) + \gamma) \approx \frac{d}{dF()}(y_1, F_{m-1}(x_1)) + \frac{d^2}{dF()^2}(y_1, F_{m-1}(x_1))\gamma = 0$$

$$\frac{d^2}{dF()^2}(y_1, F_{m-1}(x_1))\gamma = -\frac{d}{dF()}(y_1, F_{m-1}(x_1))$$

$$\boldsymbol{\gamma = \frac{-\frac{d}{dF()}(y_1, F_{m-1}(x_1))}{\frac{d^2}{dF()^2}(y_1, F_{m-1}(x_1))}} \rightarrow \frac{\mathit{Derivative\ of\ Loss\ Function}}{\mathit{Second\ Derivative\ of\ Loss\ Function}}$$

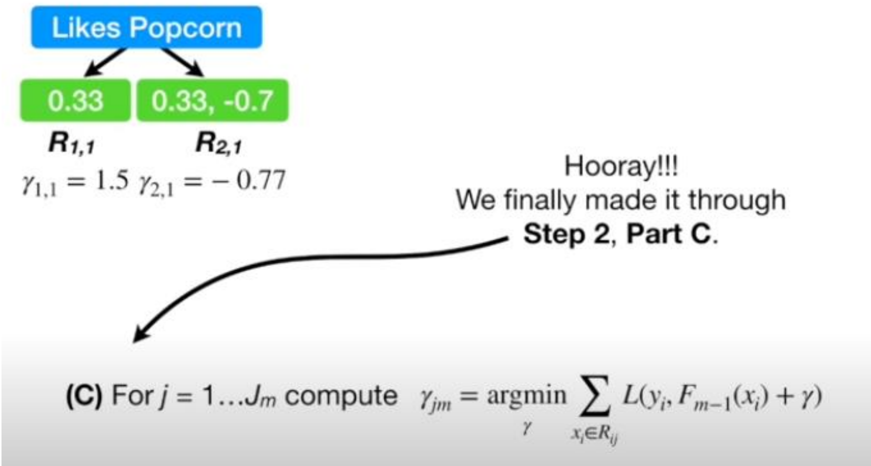
$$\boldsymbol{\gamma = \frac{Observed - \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}}{\frac{d^2}{dF()^2}(y_1, F_{m-1}(x_1))} = \frac{Observed - p}{p * (1 - p)} = \frac{Residual}{p * (1 - p)}}$$

有了这个 Equation 之后，就能计算每个 Leaves 应该要 Output 的值 γ_{jm}。



$$\gamma_{1,1} = \frac{Residual}{p*(1-p)} = \frac{0.33}{0.67*(1-0.67)} = 1.493$$

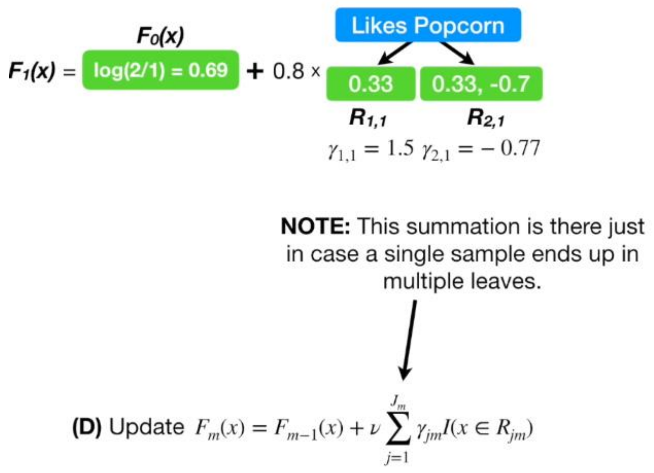
$$\gamma_{2,1} = \sum \frac{Residual}{p*(1-p)} = \frac{0.33}{0.67*(1-0.67)} + \frac{-0.67}{0.67*(1-0.67)} = -0.77$$



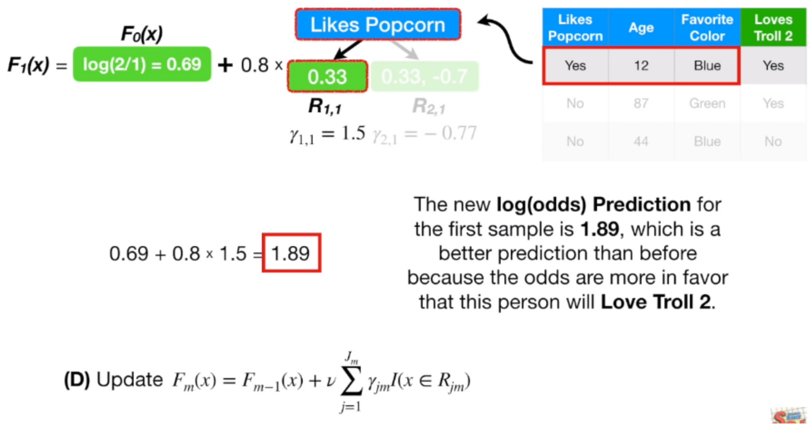
当算完所有的 γ_{jm} 之后，就需要 Update $F_m(x)$，也就是 Prediction 的值。

$$F_m(x) = F_{m-1}(x) + \upsilon \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

$$\upsilon \to Learning\ Rate$$



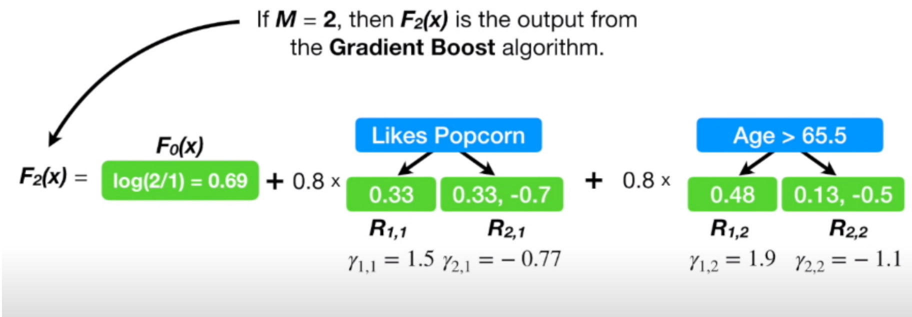
上图的 Learning Rate 设置为 0.8 只是为了给一个 Example，通常 Learning Rate 的值是在 0 到 1 之间，一般上使用 0.1。



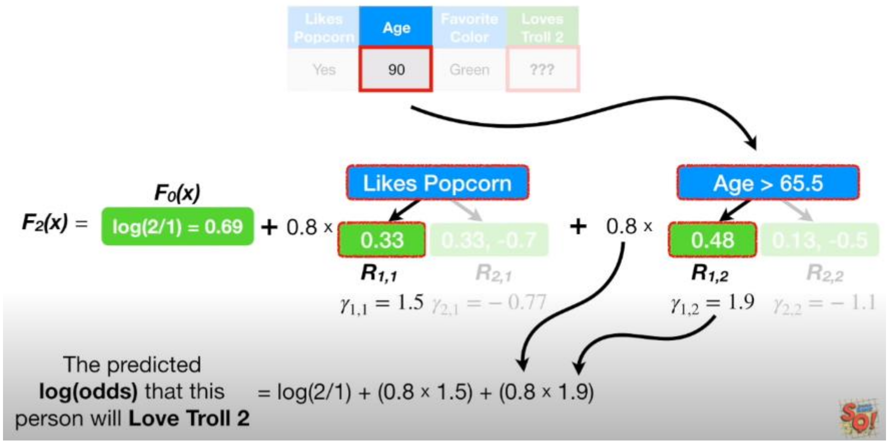
当全部值都计算完与设定完后，就能进行 Prediction 的工作，上图是只有一个叶子与一颗树，越多树，预测的值也会越准确。上图算出的 1.89 需要转换成 Probability。

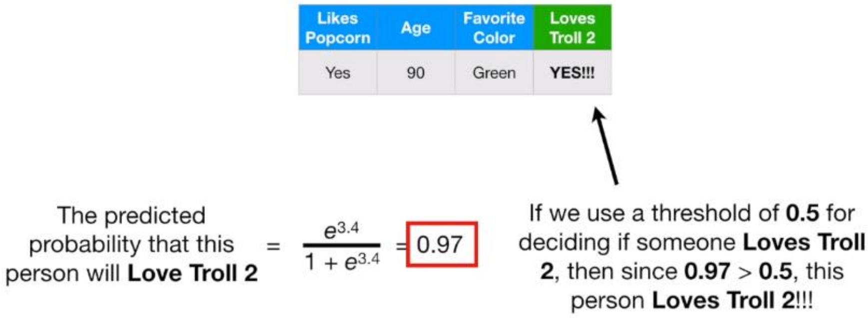
$$\frac{e^{F(x)}}{1+e^{F(x)}} = \frac{e^{1.89}}{1+e^{1.89}} = 0.869$$

当这个几率打过一个设定的 Threshold 时候，就代表这个预测是 True，少过 Threshold 就代表是 False。



上图是计算第二棵树的结果 $F_2(x)$, 而$F_2(x)$ 也就会当作最终 Output 值。





The predicted **log(odds)** that this person will **Love Troll 2** $= \log(2/1) + (0.8 \times 1.5) + (0.8 \times 1.9) = 3.4$

如上图的 Example，当计算出的几率大于 Threshold 就被设置成 True。

小总结

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 $i = 1,\ldots,n$

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

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

Step 3: Output $F_M(x)$