Classification

Logistics regression

- It relies on a specific model relating predictor variables with an outcome (target) variable.
- Due to the presence of the logit lnk function, parameter estimation is done using **maximum likelihood estimates**
- 把classification问题，用log转换成数字，再用probability来做。

判断true/false种类

> 不仅能做classification预测，还能得出possibility on each。

Decision tree

Root, Node, leaf

怎么判断谁来做第一个node？—> Measures of **Node Impurity**

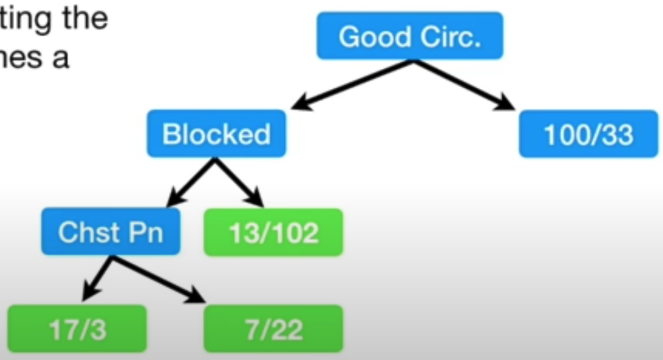
▼ Gini

- Gini impurity 选最小的
- Yes No question / numerical (先rank然后算两数平均数的Gini)
- ranking / multiple choices (最后一种可能性选项不用算)

1) Calculate all of the Gini impurity scores.

2) If the node itself has the lowest score, than there is no point in separating the patients any more and it becomes a leaf node.

3) If separating the data results in an improvement, than pick the separation with the lowest impurity value.



▼ Entropy

- Entropy is a measure of disorder or uncertainty and the goal of machine learning models and Data Scientists in general is to reduce uncertainty.

Let's calculate the entropy for the parent node and see how much uncertainty the tree can reduce by splitting on Balance.

$$E(Parent) = -\frac{16}{30}\log_2\left(\frac{16}{30}\right) - \frac{14}{30}\log_2\left(\frac{14}{30}\right) \approx 0.99$$

$$E(Balance < 50K) = -\frac{12}{13}\log_2\left(\frac{12}{13}\right) - \frac{1}{13}\log_2\left(\frac{1}{13}\right) \approx 0.39$$

$$E(Balance > 50K) = -\frac{4}{17}\log_2\left(\frac{4}{17}\right) - \frac{13}{17}\log_2\left(\frac{13}{17}\right) \approx 0.79$$

Weighted Average of entropy for each node:

$$E(Balance) = \frac{13}{30} \times 0.39 + \frac{17}{30} \times 0.79$$
$$= 0.62$$

Information Gain:

$$IG(Parent, Balance) = E(Parent) - E(Balance)$$
$$= 0.99 - 0.62$$
$$= 0.37$$

Splitting on feature ,"Balance" leads to an information gain of 0.37 on our target variable. Let's do the same thing for feature, "Residence" to see how it compares.

- Note：两种放都要记得weighted average算分数

> 💡 Regressions offer a different approach to prediction compared to decision trees. **Regressions**, as **parametric models**. **assume a specific association structure between inputs and target**. By contrast, **trees**, as **predictive algorithms**, **do not assume any association structure**; they simply seek to isolate concentrations of cases with like-valued target measurements.

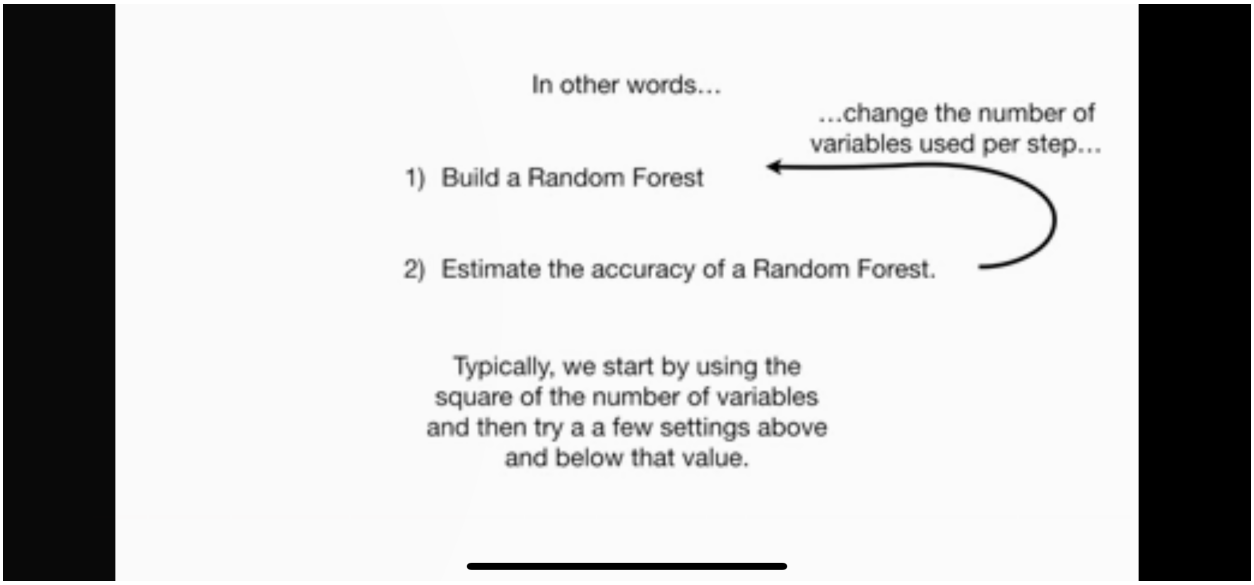
Random forest

用单个decision tree对于新的不同种类不一定准确（inaccurate），所以多用sample做好多random tree。

怎么选用几个variable？-试！

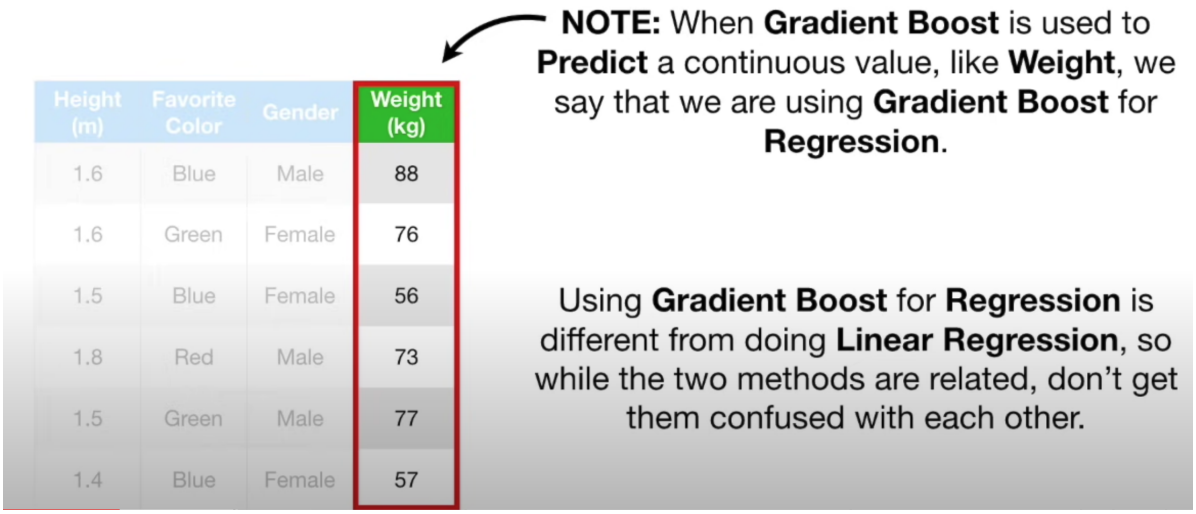
用什么验证？-没被pick的sample（一般1/3的sample）

如果有missing data？-分在sample里missing还是要predict的data里missing。但基本原理都是用结果反推。

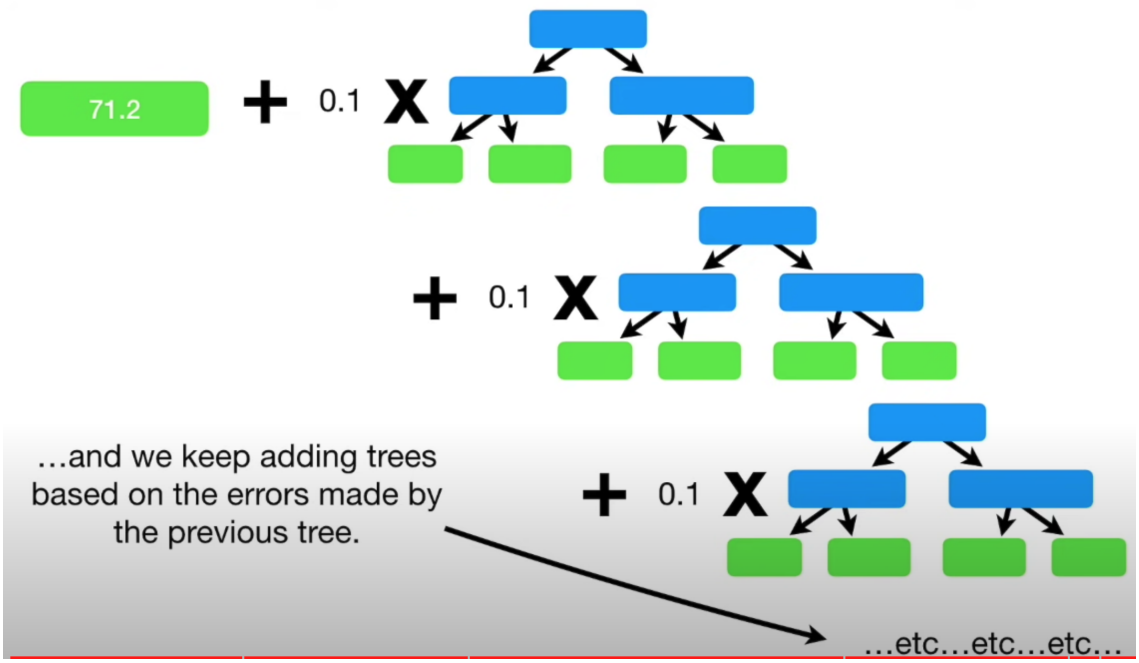


Boosting tree

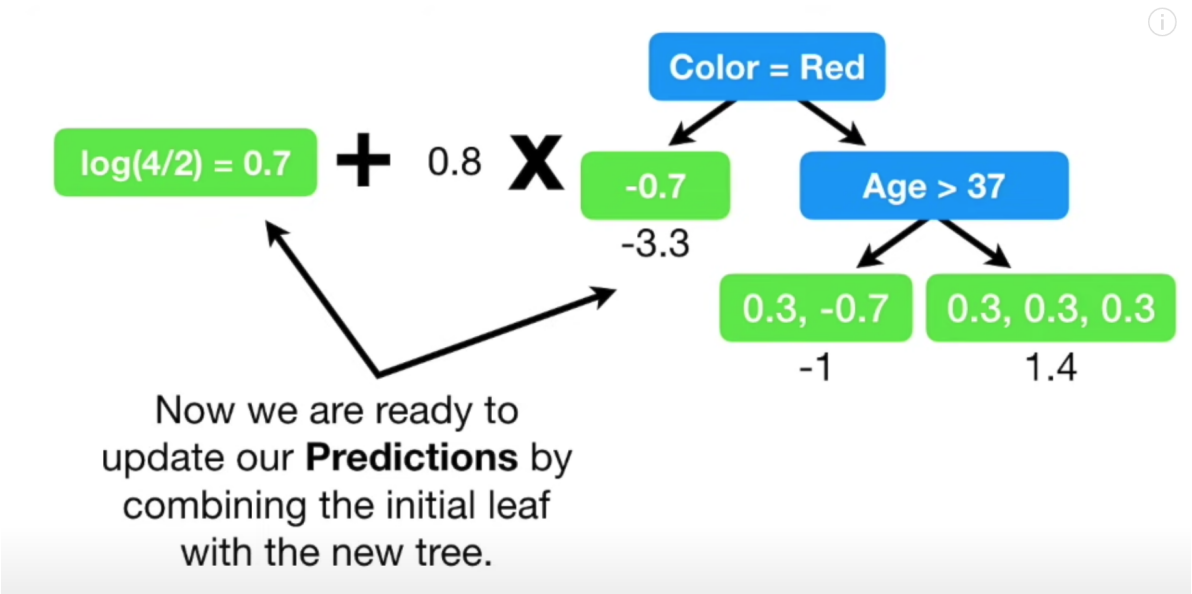
- When Gradient Boost is used to **Predict** a **continuous** value, like **Weight**, we say that we're using Gradient Boost for **Regression**(跟liner Regression不一样，注意区别）.



- 跟adaBoost 相比，Gradient Boost是**反着推**
  1. average value of the variable we want to **predict**
  2. 建一个tree based on the residuals
  3. scale the tree's contribution to the final Prediction with a Learning Rate(0-1)
  4. Add another tree based on the residuals
  5. 理论就是每次one small step to the right direction，加在一起就是非常accurate了



- When Gradient Boost is used for **Classification**, it has a lot in common with Logistic Regression.
- 不能直接加，需要一些math


Color = Red
log(4/2) = 0.7
+
0.8
X
-0.7
-3.3
Age > 37
0.3, -0.7
-1
0.3, 0.3, 0.3
1.4
Now we are ready to update our **Predictions** by combining the initial leaf with the new tree.