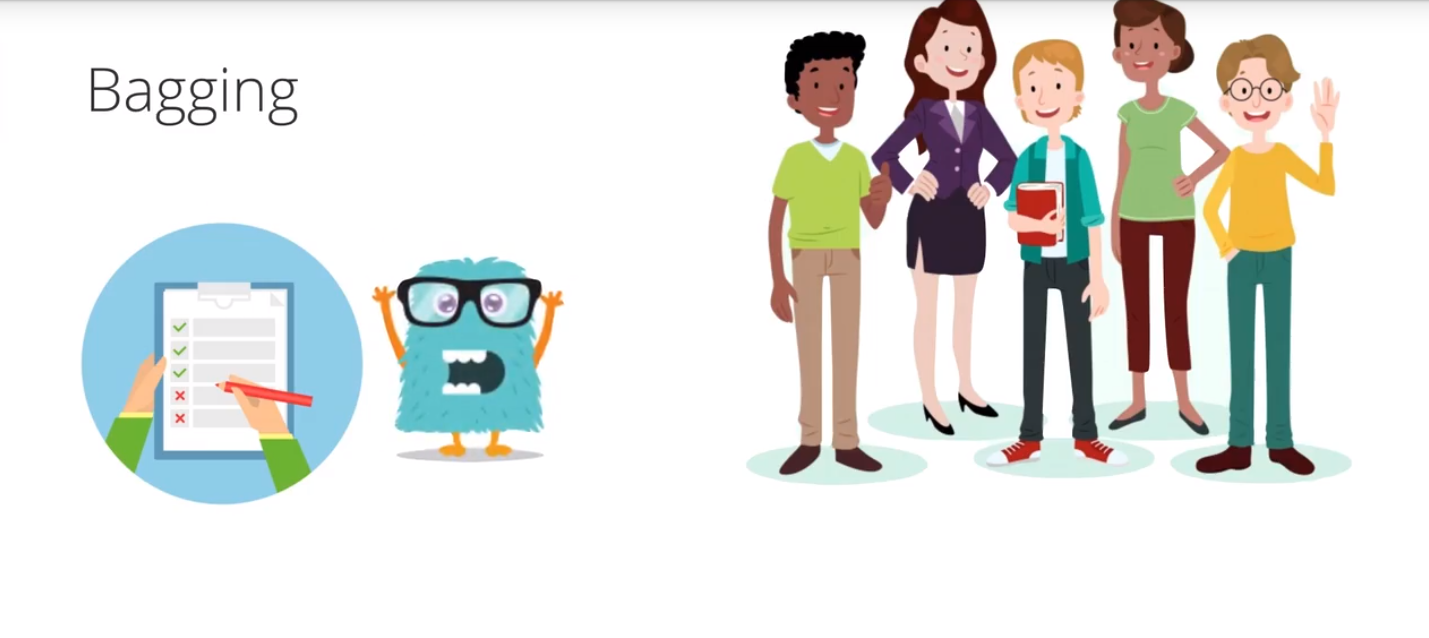
1. 简介

最普遍的两种集成方法,分别是 bagging 和 boosting

例如：我们要做判断题测试，而我们不是很有信心,所以需要向朋友求助

## 1-bagging



我们要做的事情是，让每个朋友分别做这项测试，最后把结果集成到一起。要怎么把结果集成到一起呢？

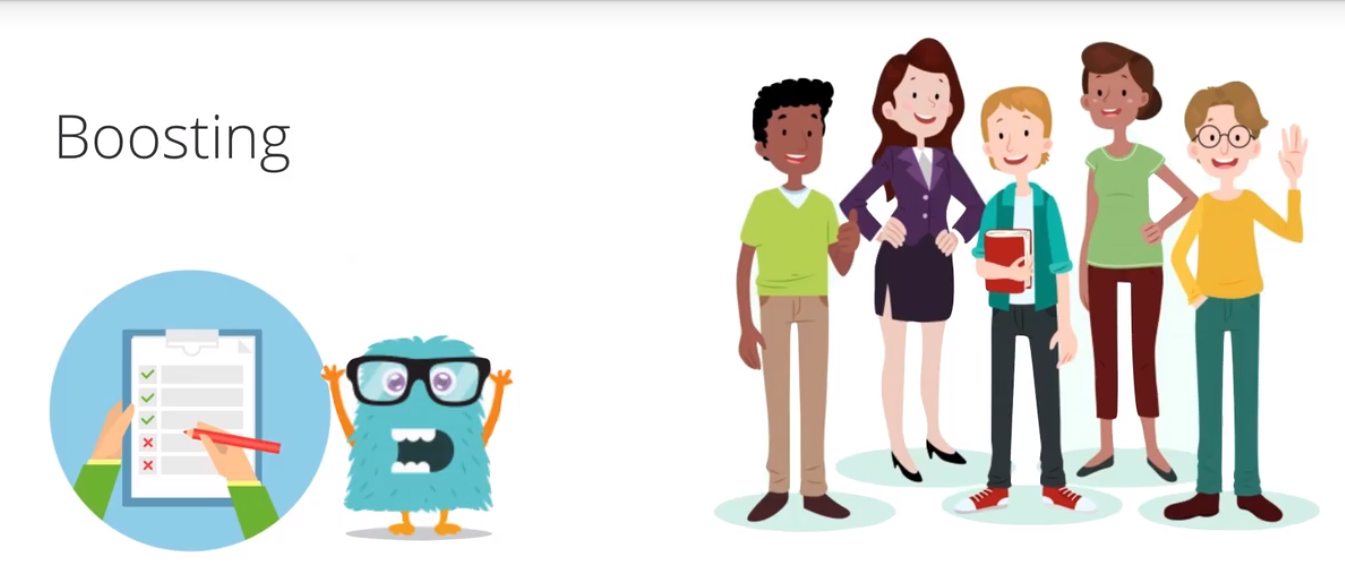
例如 如果问题的答案与数值有关，那我们可以取平均值

如果测试都是判断题 那么我们可以用计票的方式

对于每个问题，主要看 是与否 两个选项中 的得票多少，只选得票多的答案。

这样 我们就把朋友们集合成了一个天才，帮助我们在测试中更好地表现

2-boosting



假设先选了一位朋友 让他来做测试，这位朋友是个哲学家,所以 他答对了所有关于哲学的问题,但是关于科学问题 他表现并不好。

所以我们再考虑其他朋友，看谁能在这方面帮上忙, 我们发现朋友中，有一个科学家，所以再让她来做这项测试，然后重点关注科学问题。

假设她对运动一无所知,那位哲学家朋友也不了解运动,所以他们关于运动的问题都答错了，然后我们再找擅长运动的朋友来回答这些问题

同样的道理 我们再找擅长数学的朋友,音乐家朋友 等等

最后 把他们集合到一起,形成一个答题非常厉害的智慧群体,这种方法就叫做 boosting

总结

把每位朋友称作 “弱学习器”，把他们集合成的小群体称作 “强学习器”

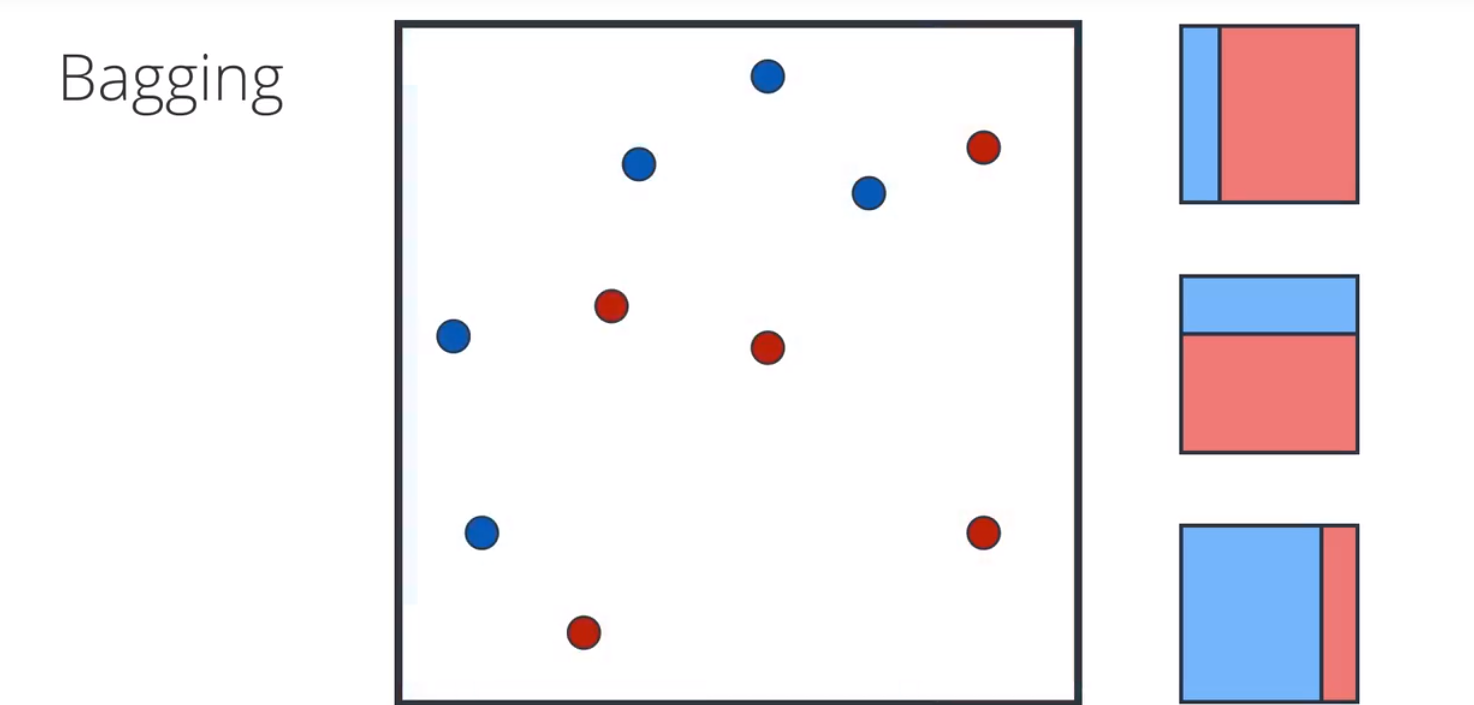
我们要做的事情，就是有一堆模型 然后把它们整合成一个更好的模型，所以 原模型不一定非要是很棒的模型，我们只需要它们的性能比随机选择的效能略好就够了。

1. Bagging

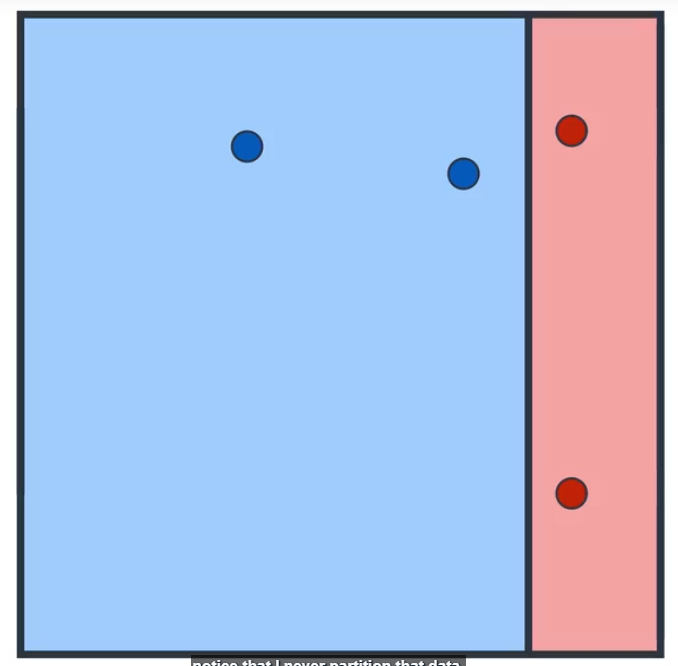
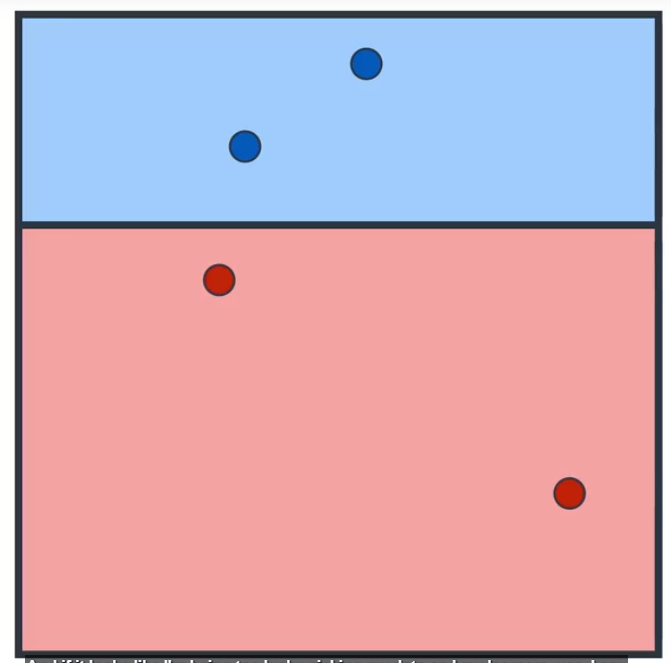
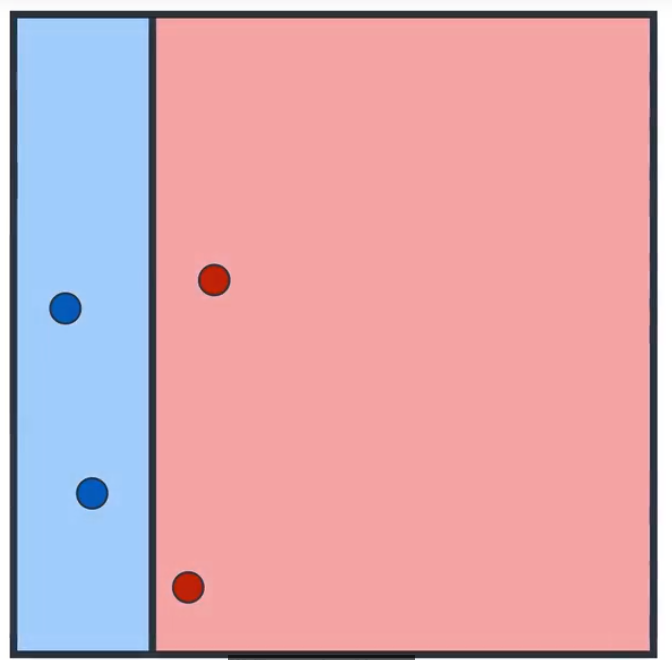
简介

使用最简单的弱学习器-单结点决策树模型，既分类结果 一侧是正例样本,另一侧是反例样本

由于数据可能过于庞大，我们不想针对同样的数据采用多种模型，这样做代价太大，相反 我们选取部分子集，每个子集使用一种弱学习器，这样我们就知道怎么把这些学习器集合到一起。



## 三个子集学习器



总体来说 这些学习器性能都不好，但是 如果我们的数据足够多，随机选择一个子集通常就能让我们了解一定情况，而且能让整个流程进展更快。

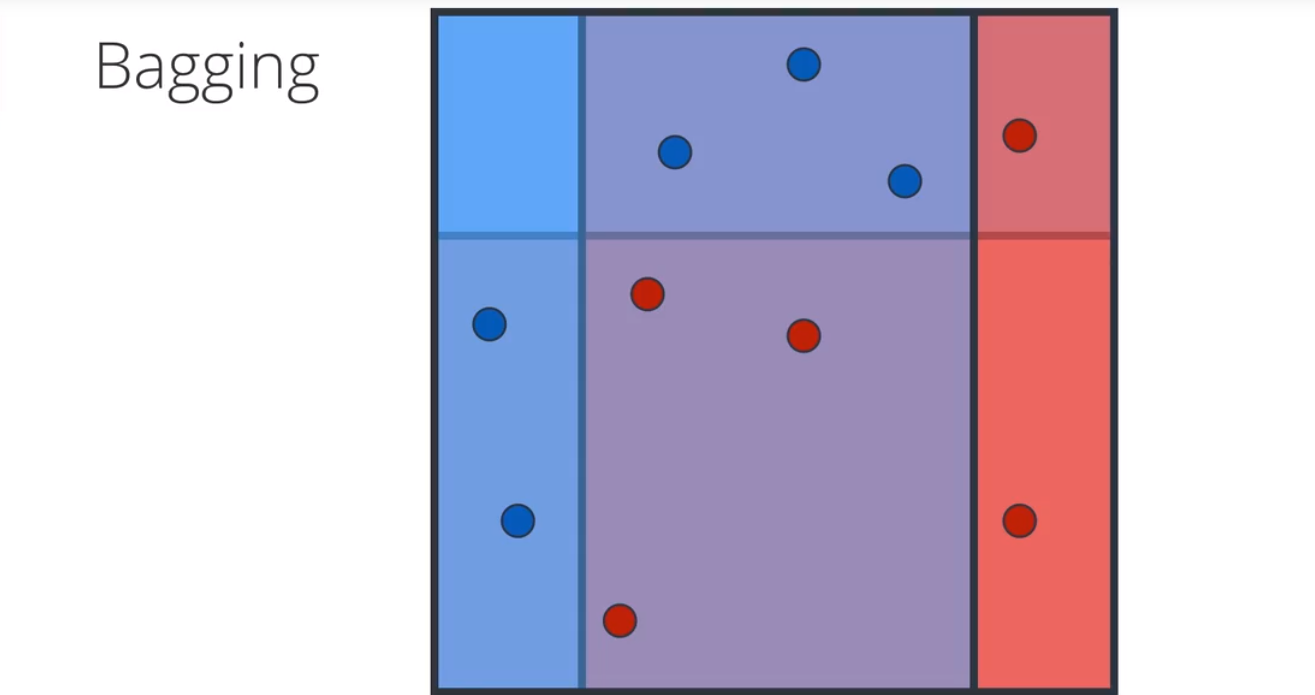
注意 这些数据 我从未做过分区处理，每一步 我们都随机选择一个子集。

## 集合在一起

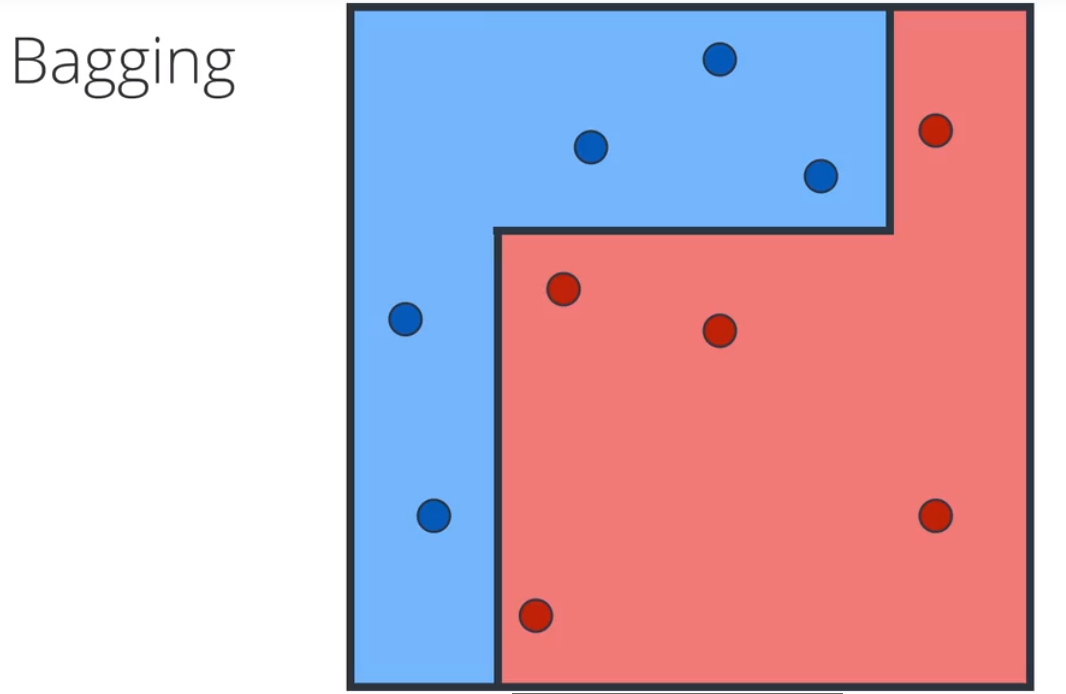
现在有三个弱学习器，我们怎么把它们集合到一起？ 不如通过计票，把这些学习器和点重叠在一起

如果数量更多的是蓝点 那结果就是蓝点

如果数量更多的是红点 那结果就是红点



采用计票方式我们就得到了这样的结果



这就是 bagging 算法

# 3.AdaBoost- boosting

有多种方法可以实现 boosting，最普遍的一种算法是 AdaBoost（自适应增强算法）

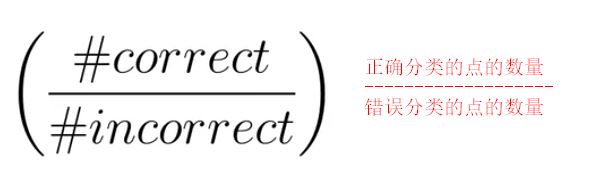
1. **数据权重计算**

## 目标

1. 最大程度地减少错误数量，以保证准确率最高

2. 把误分类的点的权重 累加的和 降到最小

## 数据权重公式

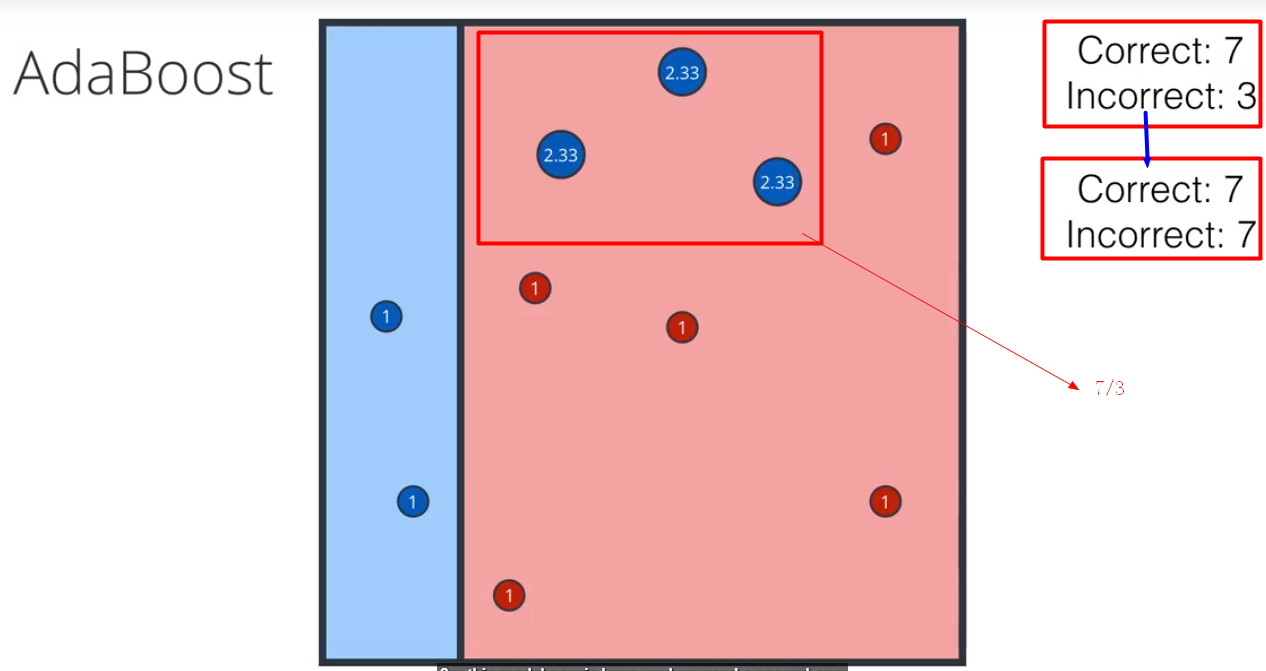


## 条件

假设每个数据点的初始权重相同为 1

## 案例

### 第一个模型



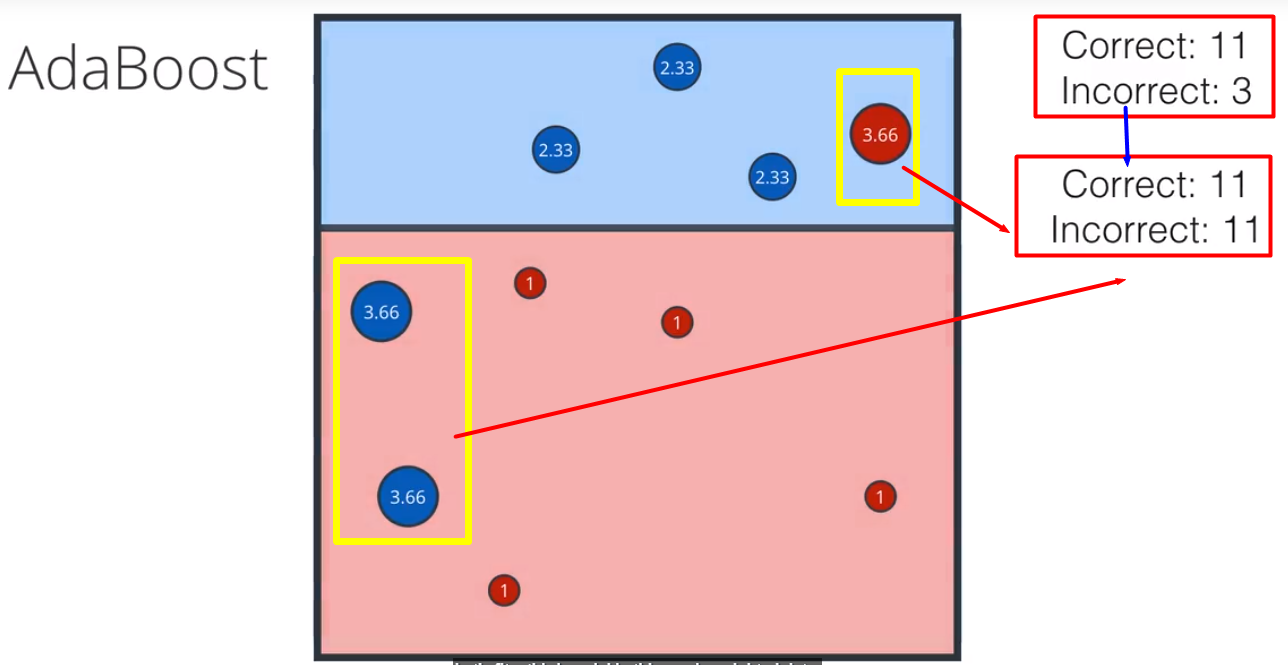
把正确分类的点的权重相加结果是 7,而误分类点的权重和是 3

重新为 误分类的点 赋予权重，条件是使得该模型成为一个 50:50 的模型，筛选出这三个点，给他们赋予7/3的权重（也就是 2.33）

既：模型现在正确分类的点的权重和，即为 7，误分类的点的权重和同样是 7

### 第二个模型

目标：要最大化的匹配 这些新的权重值

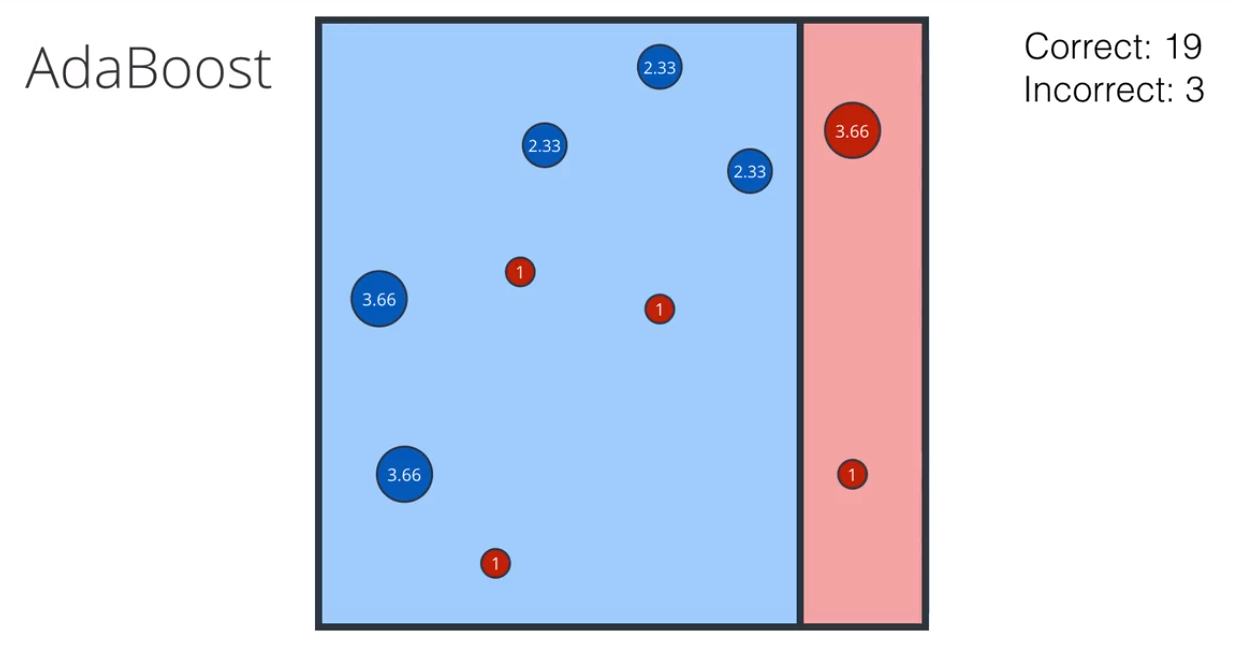


在这个模型中 把正确分类的点的权重相加，所得结果为 11 ，误分类的点的权重和是 3

再一次为误分类的点赋予权重，

同样 条件是要使得该模型成为 50:50 的模型，我赋予这三个错误，11/3的权重 (也就是 3.66)，使得这些误分类的点的权重和就是11 = 11/3 \* 3

### 第三个模型

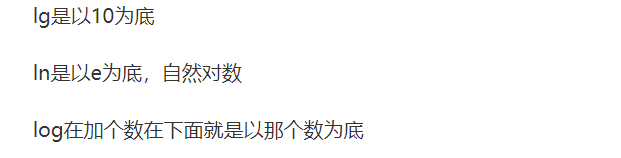


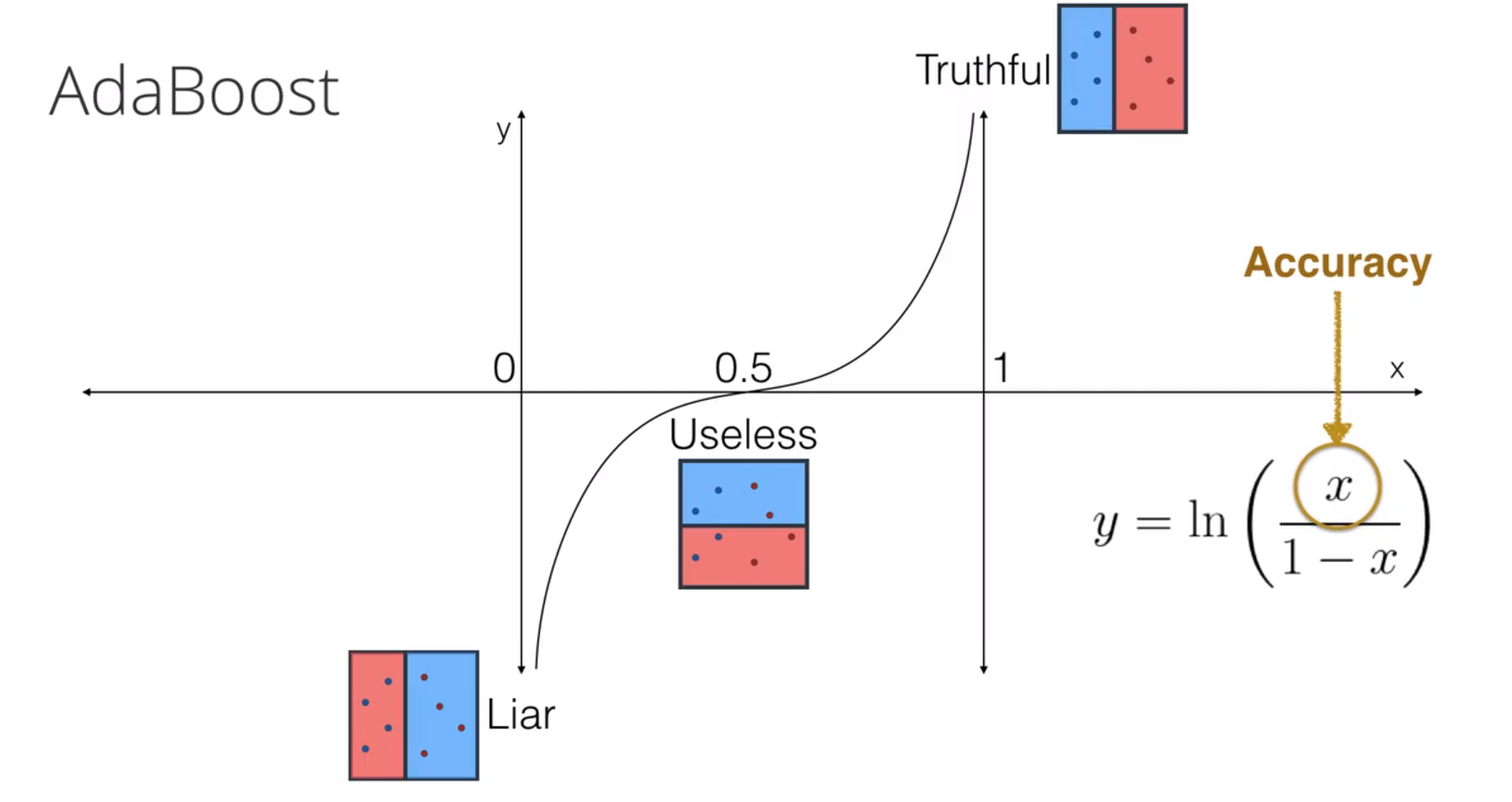
那么正确分类的点的权重和是多少呢？答案是 19 而误分类的点的权重和依然是3

这个流程可以继续进行下去 但我们先停在这里

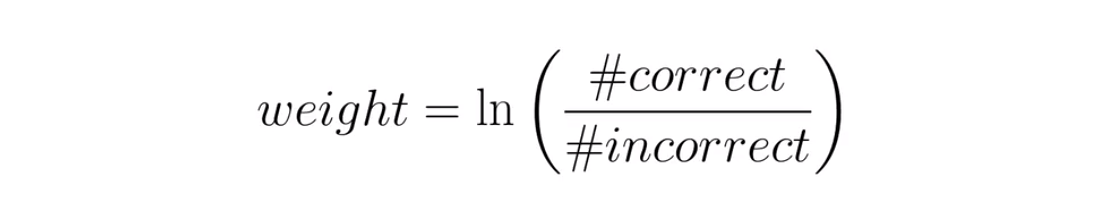
已经建立了三个模型或者叫三个弱学习器

# 5 模型权重计算

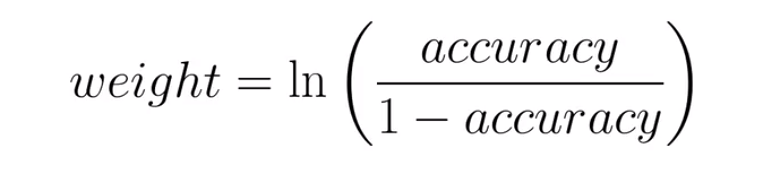




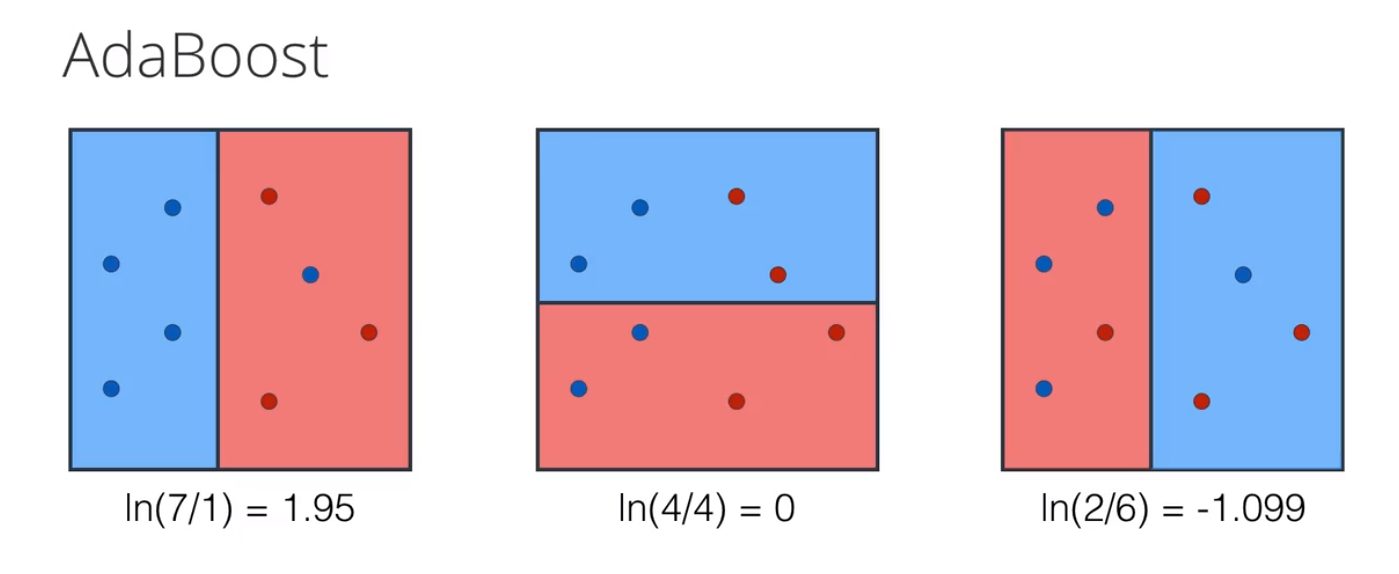
## 权重公式



权重公式也可以写成: 正确分类的点的数量 除以 错误分类的点的数量 所得 商的 自然对数



**注意： 这个公式与上边的公式是一样的，只不过，上边的公式容易计算，以上边公式为准。**



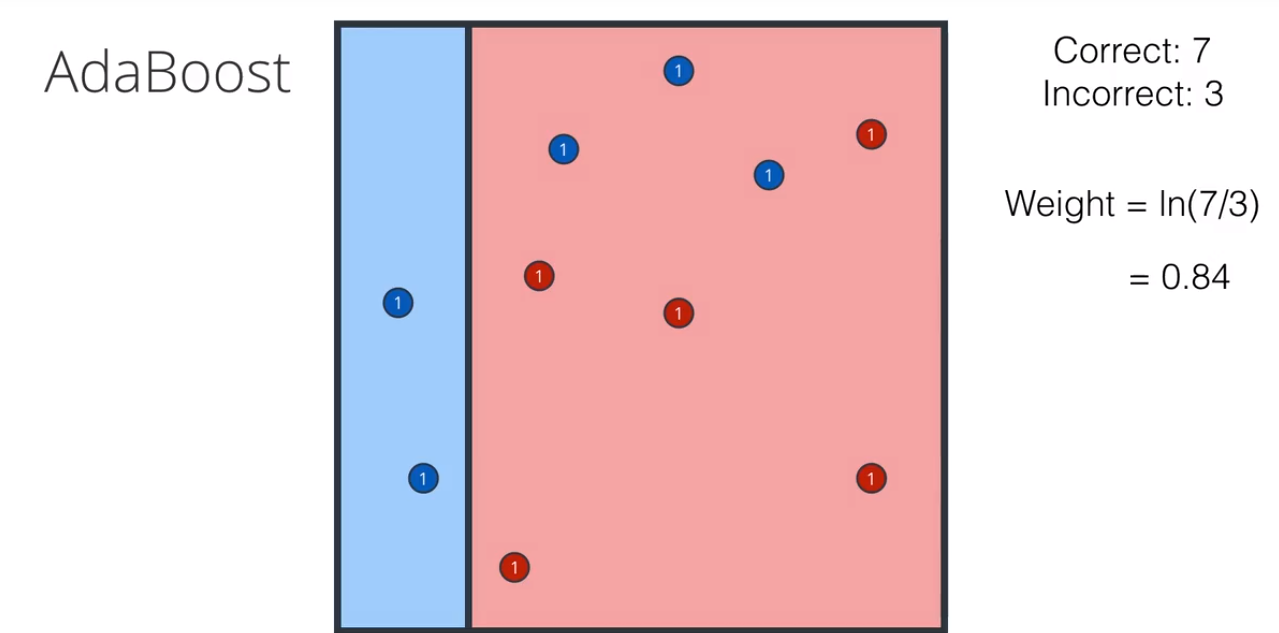
注意这个权重值是负数（-1.099）

这意味着我们不舍弃这个模型，但我们采纳与其相反的决策 因为这个模型大多数时候都是错误的

# 6 集合模型

## 1. 弱学习器-权重计算

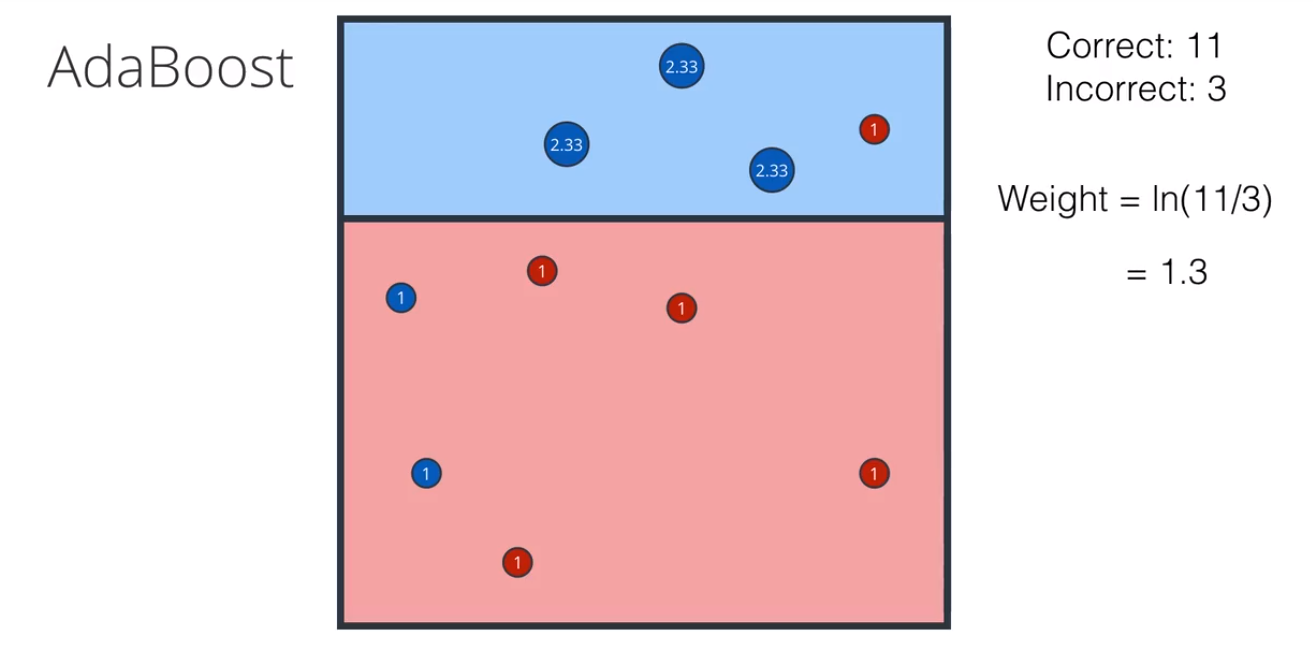
### 第一个弱学习器



正确分类的点的权重和是 7，误分类的点的权重和是 3

模型权重就是 7 除以 3 所得商的自然对数 即 0.84

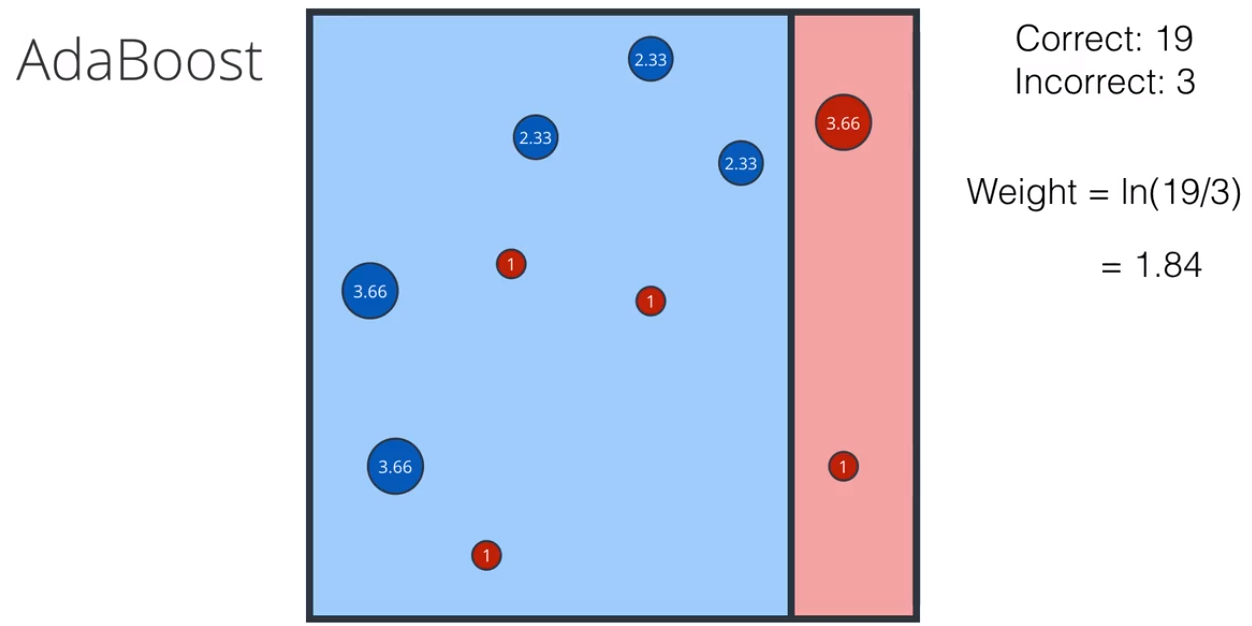
### 第二个弱学习器



正确分类的点的权重和是 11，误分类的点的权重和是 3

模型权重值是 11 除以 3 所得商的自然对数 即 1.3

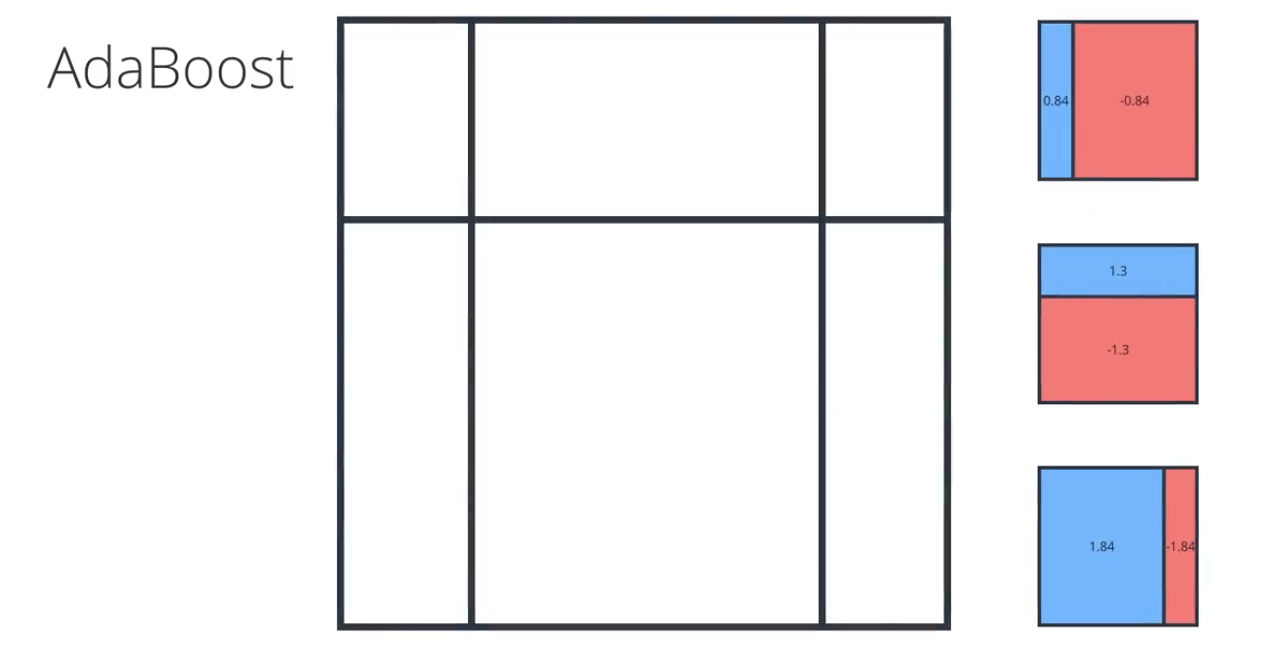
### 第三个弱学习器

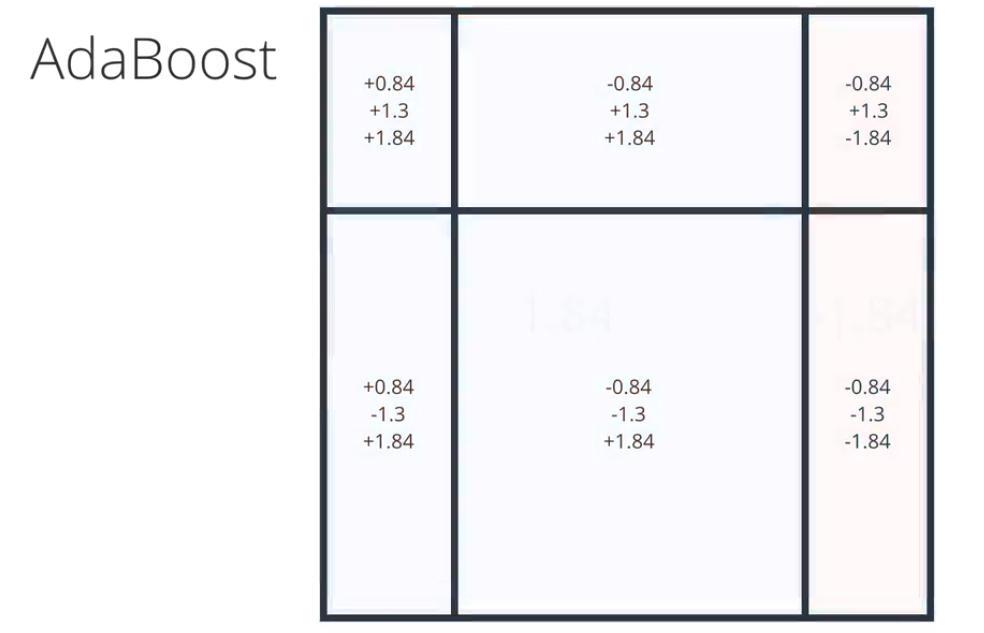


正确分类的点的权重和是 19，误分类的点的权重和是 3

模型权重值是 19 除以 3 所得商的自然对数 即 1.84

## 2.集成弱学习器

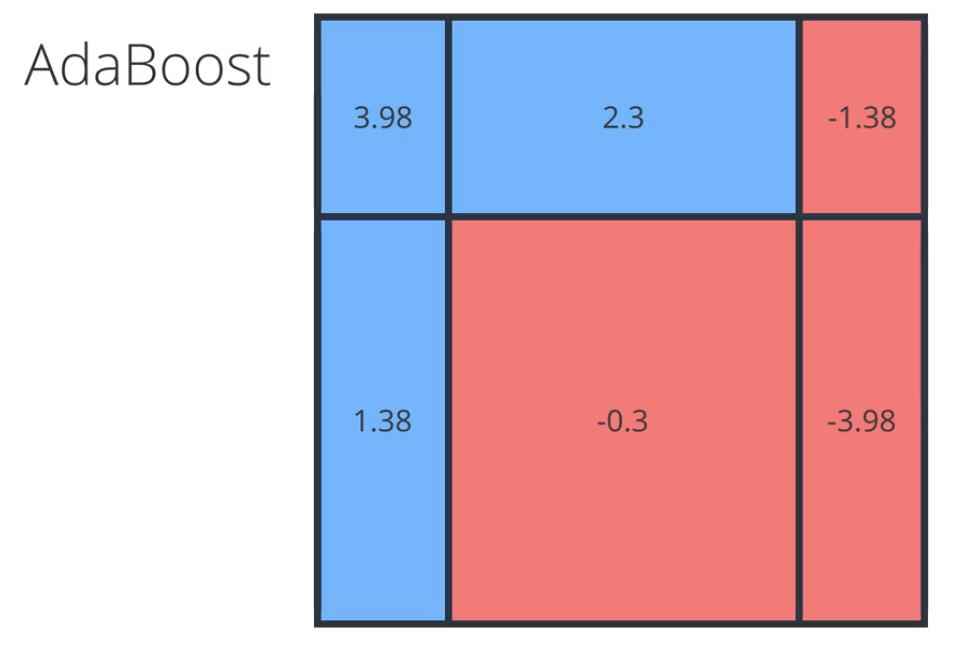




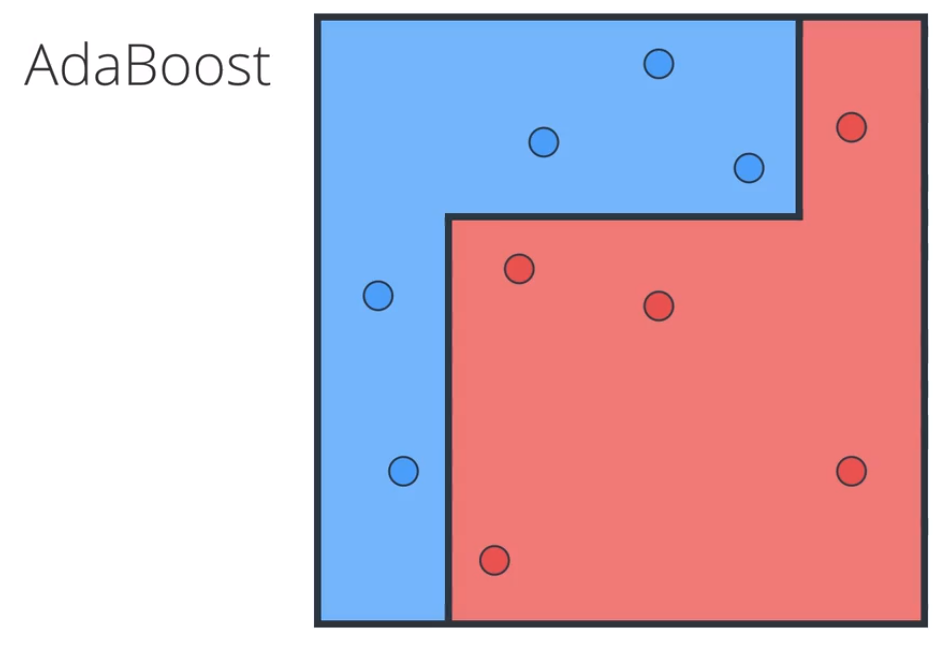
我们用计票的方式把他们集成起来,再根据它们各自的权重计算相应的票数

如果是正数 就把所有的权重相加，是负数,就减去他们的权重 进行计算

1. 结果



结果是正值 那就是蓝色，结果是负值 那就是红色



**把弱学习器的集成在一起，形成一个强学习器，这就是 AdaBoost 算法**

# 9.scikit learn中的AdaBoost



**from** sklearn.ensemble **import** AdaBoostClassifier

model = AdaBoostClassifier()

model.fit(x\_train, y\_train)

model.predict(x\_test)

**#**弱学习器使用的模型

**from** sklearn.tree **import** DecisionTreeClassifier

model = AdaBoostClassifier(

base\_estimator = DecisionTreeClassifier(max\_depth=2),

n\_estimators = 4

)