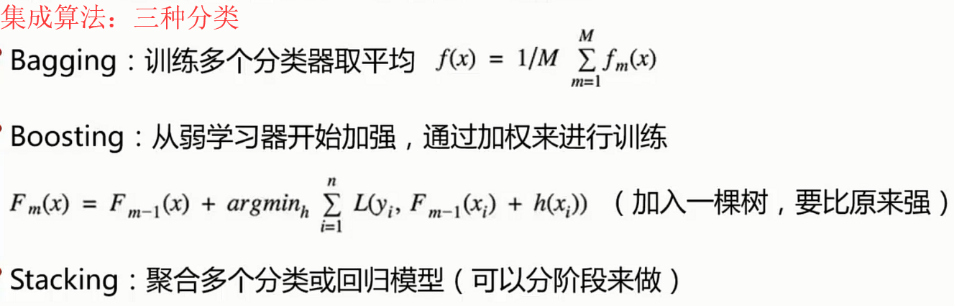
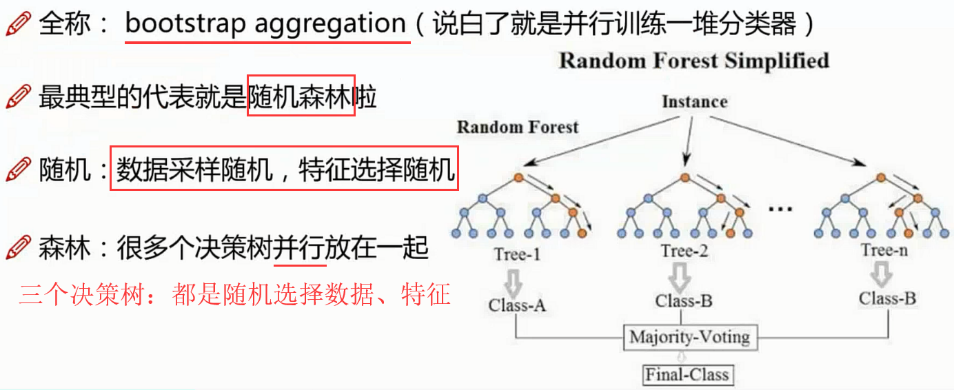
1. 集成算法
2. **本质：**就是多个决策树叠加，或者说多种算法叠加使用
3. **分类：**



1. **Bagging模型（bootstrap aggregating，自举汇聚法）：并行算法**

**其所含的分类器，都是等权重的。**



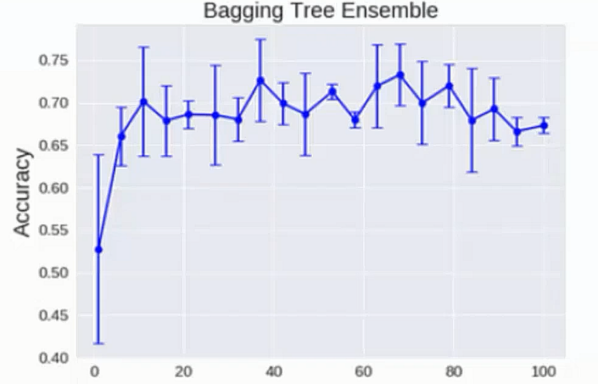
**1）典型代表**：随机森林，（将很多决策树，并行放在一起）

**2）随机森林：**每棵树都时不一样的。

**特点：**可以处理高维度数据、不用进行特征选择，从结果可判别哪些特征更

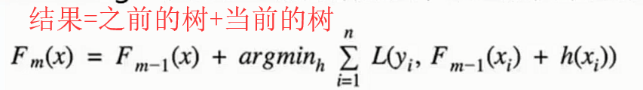
重要。

**3）树模型数量选择：**理论上，树越多，精度越高。



1. **Boosting模型：串行算法**

其所含的分类器，不是等权重的，每个权重代表对应分类器在上一轮训练中的成功度。权重越大，分类效果越好的。



1. **典型算法：**AdaBoost、XgBoost
2. **AdaBoost：**需要通过根据上一个结果来调整权重。(对离群点敏感)

eg：上次的某个数据分类错误，这次需要加大该数据的权重（重视该数据）

最终结果 = 所有分类器的加权求和的结果

1. **XgBoot：**每叠加一次，效果都会更好。
3. Stacking模型：堆叠算法
4. 直接使用算法堆叠：例如：SVM、KNN、RF等
5. 选择阶段1）结果中的特征，进行阶段2）的训练。以此，达到最佳效果。

1. AdaBoost（adaptive boosting）详解：

（以弱学习器作为基分类器，eg：单层决策树）

1. 思想：（每个训练数据的**样本**都有一个权重参数，每个**分类器**也有对应权重）
2. 初始化不同分类器的权重，对每一个分类器进行迭代
3. 计算错误率

更新**分类器**权重参数

1. 更新**对应样本**权重参数
2. 一直重复训练2-3步骤，直至错误率=0，or分类器数目达到制定指标。
3. AdaBoost实现伪代码：

每次迭代：

{ 找最佳单层决策树

将最佳单层决策树加入单层决策树数组

计算alpha

更新权重向量D

更新累计类别估计值

if 错误率==0.0

退出循环

}