集成学习算法：

1. Bagging（对原数据集进行放回抽样，总共形成s个数据集，将这s个数据集作用于同一函数，根据函数的结果进行投票）
2. boosting{介绍了AdaptiveBoosting}，是顺序处理一个数据集，每次着重处理上次分类错误的样本

算法要点：

1. 注意样本的分类标签为【-1，1】，不是【0，1】
2. 注意每个训练样本是就有权重的，权重通过上一次的迭代获得，上一次分类正确的样本下一次权重会变小，上一次错误的样本下一次权重会变大，第一次迭代每个样本的权重相同
3. 每一次迭代都是找到一个在当前权重下，找到一个加权误差最小的弱分类器（本文使用单层决策树），加权误差具体计算为：每个样本的权重×该样本是否分对，分对则为0，分错则为1，例：【0.5，0.125，0.125，0.125，0.125】\*【1，0，0，1，0】，前者是每个样本权重（注意其和要为1），后者是是否分对，可知，0，4号样本分错。
4. 找到当前迭代下的最好的弱分类器后，对alpha，D，进行更新，具体公式为： alpha=0.5\*ln((1-error)/error)，其中error为当前最好的弱分类器在训练集上的误差，对于分对样本Di=Di\*e(-alpha)/sum(D)，对于分错的样本Dj=Dj\*e(alpha)/sum(D)，注意要到最后全部更新完Di后再除sum(D)保证D所有元素之和为1
5. 算法直到达到迭代次数或在训练集上的误差为0
6. 如何判断一个样本的取值是【-1，1】，求出每个分类器的alpha后，将一个样本分别带入所有分类器，分类器给出-1或1的结果，然后这个结果×alpha，并对所有分类器求和，sign（∑分类器i）

在第七张，同时学习了precession，recall

**注意第七章同时讲了roc曲线，本人并没有去看**

**倾斜类的处理并没有认真看（主要是对样本少的类过抽样（即复制一些相同的类），对样本多的类欠抽样（即删除一些离分类边界较远的样本）**