# hard voting

对各个子模型进行结果投票,投票占比多的结果为最终结果.

# soft voting:

只计算投票的多少来决定并不是最准确的,加入每个决策者的决定权重.要求每个模型都能计算得出结果的概率.

各模型的概率计算方式:

1.knn:得出结果时,分类中最多的类别所占个数的比例

2.决策树: 得出结果时,分类中最多的类别所占个数的比例(分类完后信息熵不为0,说明有多个分类)

3.svm:sklearn的svc实现中有个probability参数,默认为false,设置成true 可以输出概率.

# 集成学习的扩展:

继承学习不能只靠机器学习算法类型来进行,这样的集成学习个数总是固定的,当我们想要成千上万的投票者就无法满足条件,此时我们需要将一个学习算法通过给予不同的数据(只看部分数据)来训练出不同的子模型,只要给出的数据不同,训练出来的子模型有差异性,这样的投票就是有效果的.虽然每单个模型的准确率会下降,但是投票后整体准确率还是会上升.

# 部分数据取样方法:

**放回取样 -Bagging(统计学中称为bootstrap)** 更常用,可以生成更多的子模型,不依赖随机.

**不放回取样 -pasting** 等于将数据分成多少份,这个数据怎么分是有很大的随机性的.而这个随机性和最终的预测结果息息相关.

放回取样经过数据统计,大概有37%的数据是不会被取到的(oob out of bag),这部分数据我们可以用来做测试数据.sklearn的bagging分类器中的 oob\_score参数指定为true,再调用oob\_score\_来查看结果,该方案就是据此部分数据计算而来的.

除此之外还可以对特征进行随机采样,特征的放回及不放回采样.还可以同时对样本及特征同时进行随机采样来生成子模型.

集成学习一般比较慢,但是可以方便的并行计算,sklearn可以对集成学习类中的n\_jobs =-1 来设置自动根据cpu核数进行并行计算.

# 随机森林:

在集成学习中,如果使用的子模型为决策树模型,则该集成学习方法为随机森林,决策树是一个比较好的集成学习子模型,因为其可调节的超参数较多,可以很方便的生成很多子模型.

随机森林拥有决策树的大部分参数可以调节.

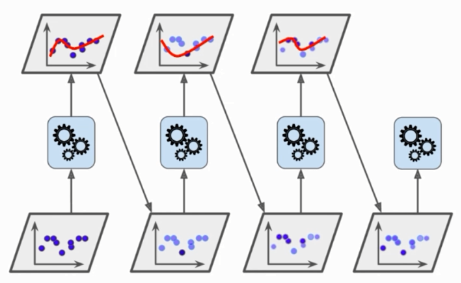
# Extra-Trees(一种特殊的随机森林):

随机选取一个特征,随机选取一个值来进行决策树的划分.训练速度快,bias偏差大,方差小.

# AdaBoosting

adaboosting其集成原理为

对一个模型的错误进行权重的加重训练下一个模型,依次迭代,最终结果为各模型的加和.



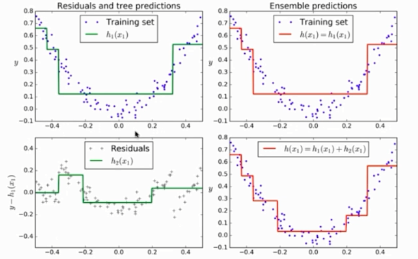
# Gradient Boosting

训练一个模型m1,产生错误e1

针对e1训练第二个模型m2,产生错误e2

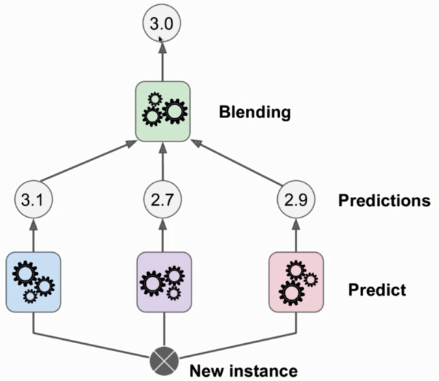
针对e2训练第二个模型m3,产生错误e3 ...

最终预测结果为 m1+m2+m3....



# stacking

将voting方式进行改版,不是直接投票或者根据权重投票,而是加一个训练模型,这个模型用其他模型的预测结果进行预测.将数据分成好几份一份训练下面的模型,一份训练上面的模型.

,

当然也可以多加一层stacking这个就有神经网络的味道了!

