**随机森林算法梳理**

1. 集成学习的概念

通过构建并结合多个学习器来完成学习任务，集成学习的一般结构为:先产生一组"个体学习器" (individual learner) ， 再用某种策略将它们结合起来。

1. 个体学习器的概念

个体学习器通常由一个现有的学习算法从训练数据产生，如"决策树集成"中全是决策树"；神经网络集成"中全是神经网络，这样的集成是"同质"的(homogeneous) ，也可包含不同类型的个体学习器， 例如同时包含决策树和神经网络，这样的集成是"异质"的(heterogenous)

1. boosting bagging的概念、异同点

Bagging算法(套袋发)

* bagging的算法过程如下：
  1. 从原始样本集中使用Bootstraping 方法随机抽取n个训练样本，共进行k轮抽取，得到k个训练集（k个训练集之间相互独立，元素可以有重复）。
  2. 对于n个训练集，我们训练k个模型，（这个模型可根据具体的情况而定，可以是决策树，knn等）
  3. 对于分类问题：由投票表决产生的分类结果；对于回归问题，由k个模型预测结果的均值作为最后预测的结果（所有模型的重要性相同）。

Boosting（提升法）

* boosting的算法过程如下：
  1. 对于训练集中的每个样本建立权值wi，表示对每个样本的权重， 其关键在与对于被错误分类的样本权重会在下一轮的分类中获得更大的权重（错误分类的样本的权重增加）。
  2. 同时加大分类 误差概率小的弱分类器的权值，使其在表决中起到更大的作用，减小分类误差率较大弱分类器的权值，使其在表决中起到较小的作用。每一次迭代都得到一个弱分类器，需要使用某种策略将其组合，最为最终模型，(adaboost给每个迭代之后的弱分类器一个权值，将其线性组合作为最终的分类器,误差小的分类器权值越大。)

Bagging和Boosting 的主要区别

* 样本选择上: Bagging采取Bootstraping的是随机有放回的取样，Boosting的每一轮训练的样本是固定的，改变的是买个样的权重。
* 样本权重上：Bagging采取的是均匀取样，且每个样本的权重相同，Boosting根据错误率调整样本权重，错误率越大的样本权重会变大
* 预测函数上：Bagging所以的预测函数权值相同，Boosting中误差越小的预测函数其权值越大。
* 并行计算: Bagging 的各个预测函数可以并行生成;Boosting的各个预测函数必须按照顺序迭代生成.

将决策树与以上框架组合成新的算法

* Bagging + 决策树 = 随机森林
* AdaBoost + 决策树 = 提升树
* gradient + 决策树 = GDBT

1. 理解不同的结合策略(平均法，投票法，学习法)

平均法：简单平均、加权平均

适用范围：

+规模大的集成，学习的权重较多，加权平均法易导致过拟合

+个体学习器性能相差较大时宜使用加权平均法，相近用简单平均法。

投票法：

1.绝对多数投票法：某标记超过半数；

2.相对多数投票法：预测为得票最多的标记，若同时有多个标记的票最高，则从中随机选取一个。

3.加权投票法：提供了预测结果，与加权平均法类似。

学习法

Stacking描述：先从初始数据集中训练出初级学习器，然后“生成”一个新数据集用于训练次级学习器。在新数据集中，初级学习器的输出被当做样例输入特征，初始样本的标记仍被当做样例标记。

1. 随机森林的思想

随机森林利用随机的方式将许多决策树组合成一个森林，每个决策树在分类的时候投票决定测试样本的最终类别。随机森林是一种有监督学习算法。 就像你所看到的它的名字一样，它创建了一个森林，并使它拥有某种方式随机性。 所构建的“森林”是决策树的集成，大部分时候都是用“bagging”方法训练的。 bagging方法，即bootstrap aggregating，采用的是随机有放回的选择训练数据然后构造分类器，最后组合学习到的模型来增加整体的效果。

简而言之：随机森林建立了多个决策树，并将它们合并在一起以获得更准确和稳定的预测。随机森林的一大优势在于它既可用于分类，也可用于回归问题，这两类问题恰好构成了当前的大多数机器学习系统所需要面对的。 接下来，将探讨随机森林如何用于分类问题，因为分类有时被认为是机器学习的基石。

1. 随机森林的推广

extra trees

　　　　extra trees是RF的一个变种, 原理几乎和RF一模一样，仅有区别有：

　　　　1） 对于每个决策树的训练集，RF采用的是随机采样bootstrap来选择采样集作为每个决策树的训练集，而extra trees一般不采用随机采样，即每个决策树采用原始训练集。

　　　　2） 在选定了划分特征后，RF的决策树会基于基尼系数，均方差之类的原则，选择一个最优的特征值划分点，这和传统的决策树相同。但是extra trees比较的激进，他会随机的选择一个特征值来划分决策树。

　　　　从第二点可以看出，由于随机选择了特征值的划分点位，而不是最优点位，这样会导致生成的决策树的规模一般会大于RF所生成的决策树。也就是说，模型的方差相对于RF进一步减少，但是偏倚相对于RF进一步增大。在某些时候，extra trees的泛化能力比RF更好。

Totally Random Trees Embedding

　　　　Totally Random Trees Embedding(以下简称 TRTE)是一种非监督学习的数据转化方法。它将低维的数据集映射到高维，从而让映射到高维的数据更好的运用于分类回归模型。我们知道，在支持向量机中运用了核方法来将低维的数据集映射到高维，此处TRTE提供了另外一种方法。

　　　　TRTE在数据转化的过程也使用了类似于RF的方法，建立T个决策树来拟合数据。当决策树建立完毕以后，数据集里的每个数据在T个决策树中叶子节点的位置也定下来了。比如我们有3颗决策树，每个决策树有5个叶子节点，某个数据特征xx 划分到第一个决策树的第2个叶子节点，第二个决策树的第3个叶子节点，第三个决策树的第5个叶子节点。则x映射后的特征编码为(0,1,0,0,0,     0,0,1,0,0,     0,0,0,0,1), 有15维的高维特征。这里特征维度之间加上空格是为了强调三颗决策树各自的子编码。

　　　　映射到高维特征后，可以继续使用监督学习的各种分类回归算法了

1. 随机森林的优缺点

 优点：

* 它能够处理很高维度（feature很多）的数据，并且不用做特征选择；
* 由于随机选择样本导致的每次学习决策树使用不同训练集，所以可以一定程度上避免过拟合；

   缺点：

* 随机森林已经被证明在某些噪音较大的分类或回归问题上会过拟合；
* 对于有不同级别的属性的数据，级别划分较多的属性会对随机森林产生更大的影响，所以随机森林在这种数据上产出的属性权值是不可信的

1. 随机森林在sklearn中的参数解释

ensemble.AdaBoostClassifier ： AdaBoost分类

ensemble.AdaBoostRegressor ：Adaboost回归

ensemble.BaggingClassifier ：装袋分类器

ensemble.BaggingRegressor ：装袋回归器

ensemble.ExtraTreesClassifier ：Extra-trees分类（超树，极端随机树）

ensemble.ExtraTreesRegressor ： Extra-trees回归

ensemble.GradientBoostingClassifier ： 梯度提升分类

ensemble.GradientBoostingRegressor ：梯度提升回归

ensemble.IsolationForest ：隔离森林

\*\*ensemble.RandomForestClassifier ：随机森林分类

ensemble.RandomForestRegressor ： 随机森林回归\*\*

ensemble.RandomTreesEmbedding ：完全随机树的集成

ensemble.VotingClassifier ：用于不合适估算器的软投票/多数规则分类

1. 随机森林的应用场景

把多维度的数据沿着各个维度上的具体的值进行切割，比如按照性别，或者按照收入的高低