随机森林

1. 一些我对随机森林的理解

鉴于决策树容易过拟合的缺点，随机森林采用多个决策树的投票机制来改善决策树，假设随机森林使用了m棵决策树，那么就需要产生m个一定数量的样本集来训练每一棵树，如果用全样本去训练m棵决策树显然是不可取的，全样本训练忽视了局部样本的规律，对于模型的泛化能力是有害的。

随机森林采用多个决策树的投票机制，最终结果采用Bagging的策略来获得，即多数投票机制。Bagging可以简单的理解为：放回抽样，多数表决（分类）或简单平均（回归）,同时Bagging的基学习器之间属于并列生成，不存在强依赖关系。

Bagging的生成方法：

1. 从样本集中通过重采样的方式产生n个样本
2. 假设样本特征数目为a，对n个样本选择a中的k个特征，用建立决策树的方式获得最佳分割点
3. 重复m次，产生m棵决策树
4. 多数投票机制来进行预测

Random Forest（随机森林）是Bagging的扩展变体，它在以决策树为基学习器构建Bagging集成的基础上，进一步在决策树的训练过程中引入了随机特征选择，因此可以概括RF包括四个部分：

1. 随机选择样本（放回抽样, 行抽样）；
2. 构建决策树（CART树）；
3. 随机选择特征(列抽样)；
4. 随机森林投票（平均）。
5. 代码处理

不掉包实现

定义了两个类，一个是决策树类，一个是随机森林分类器的类。

决策树类包括两个函数，一是通过递归决策树找样本所述叶子节点，二是用字典描述树的结构。

随机森林类包括，静态方法计算叶子值，即选择样本中出现次数最多的类别作为叶子节点取值，静态方法计算基尼值，即分类树采用基尼指数作为指标来选择最优分裂点，静态方法分裂数据集，即根据特征和阈值将样本划分成左右两份，左边小于等于阈值，右边大于阈值，选择最佳特征函数，即寻找最好的数据集划分方式，找到最优分裂特征、分裂阈值、分裂增益，递归建立决策树函数，模型训练函数，预测函数。

调包实现：

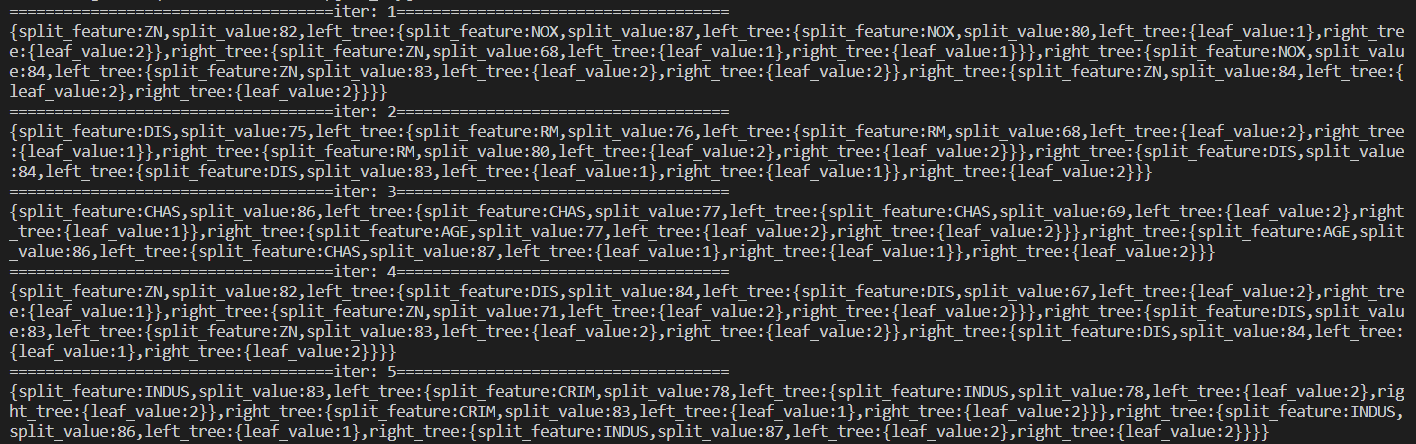
1. 实验结果与结论

处理氨基酸数据，操作仍然是ASCII码，前面8个为属性，最后一列为标签，标签由原来的-1和1改为2和1。

处理数据后，部分如下：



实验结果为：构建树



实验结果：准确率



1. 总结

关于决策树与其他几种模型的关系：

1. Bagging + 决策树 = 随机森林
2. AdaBoost + 决策树 = 提升树
3. Gradient Boosting + 决策树 = GBDT