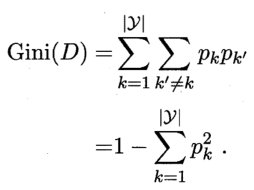
第三次-CART决策树

数据集：<https://archive.ics.uci.edu/ml/datasets/HIV-1+protease+cleavage>

要求：对树进行剪枝，并与逻辑回归做分类进行对比

1. CART原理简介
2. 数据预处理
3. 调包看效果
4. 手写
5. 实验结果分析
6. 结论
7. CART原理简介

CART即可实现分类，也可实现回归，这次要实现的任务是分类，对比之前的ID3、C4.5算法, CART算法运用的是基尼指数来衡量“纯净度”：



其中，表示某个事件第K个可能值的发生概率，其可用经验概率来表示，因此有：



其中，表示所有样本点，表示事件的第K个可能值的出现的次数，所以：

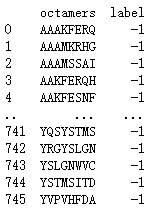
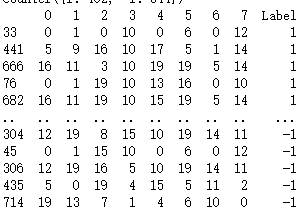
Gini(D)反映了从数据集D中随机抽取两个样本，其类别标记不一样的概率，故Gini值越小，则表示数据集D的纯度越高。

树的剪枝——

预剪枝：在树的生成过程中，对每个节点划分前，先进行估计，可以降低过拟合的风险。

后剪枝：得到树后，自下向上对非叶节点进行节点考察，先用训练集生成尽可能大的树，然后使用测试集进行剪枝操作。

1. 数据预处理
2. 数据是由一个个的字符串组成的，因此这里为了避免决策树将数据当作连续的，就把字符串拆开来一个个的单个字母对应的0-19的数字的离散型：

Labels:



1. 数据类别不平衡

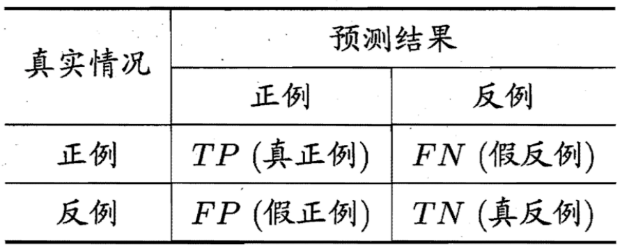
|  |  |
| --- | --- |
| 数据集 | Y原始分布比例(-1：1) |
| 746 | 344：402 |
| 1625 | 1250：375 |
| impens | 798：149 |
| Schilling | 2838：434 |

最开始采用了SMOTE的方法：对少数样本进行KNN算法插值模拟生成新样本，从而使得正负样本的类别数据个数保持平衡，结果见下表的斜线后的数据，可见平衡后的数据使得模型的评价分数均下降了许多，这说明SMOTE算法不适用于离散数据，于是才采用其他的评价指标来解决类别不平衡的问题。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试集  训练集 | 746 | 1625 | impens | schilling |
| 746 | 1.0 | 0.855/0.657 | 0.770/0.566 | 0.763/0.573 |
| 1625 | 0.829/0.958 | 1.0 | 0.765/0.6021 | 0.759/0.623 |
| impens | 0.614/0.515 | 0.586/0.533 | 1.0 | 0.789/0.608 |
| schilling | 0.595/0.856 | 0.607/0.554 | 0.818/0.632 | 1.0 |

注: 上表的score是直接用的模型自带的clf.score

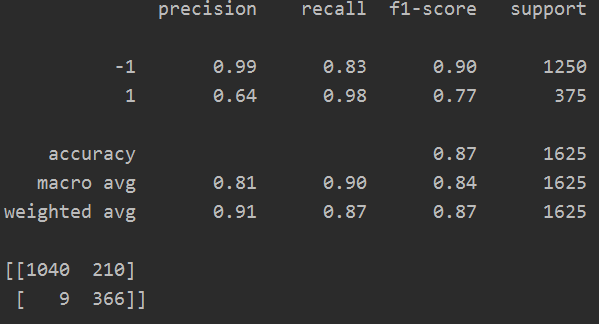
数据类别不平衡就不采用准确度来做评价指标了，并且鉴于上采样易造成过拟合，下采样易造成数据信息缺失，因此这里采用更具有说服力的评价指标——混淆矩阵，精确度，召回率,F1值，或者ROC曲线。

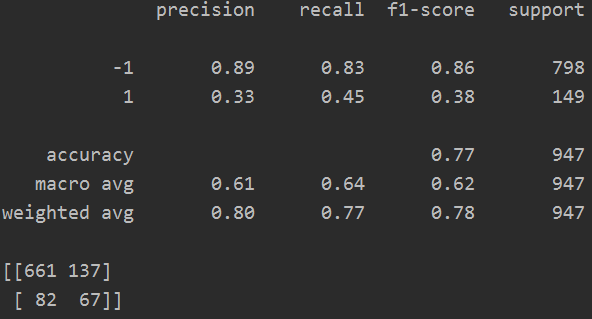


|  |  |  |
| --- | --- | --- |
| 精准率(Precision) |  | P、R相互制约，一般P高则R会降低，F1值则用来作为P与R的一个近似平衡点。 |
| 召回率(Recall) |  |
| F1-score |  | 是精准率与召回率的调和平均值，F1高的话，则意味着P、R都很高。 |
| ROC曲线 | ( False positive Rate): x轴  (True positive Rate): y轴 | AUC是ROC曲线与坐标轴围成是面积大小；  其评价标准是：越靠近左上角，模型效果越好，即AUC越大越好 |

1. 调包决策树模型，寻找模型的最优参数；

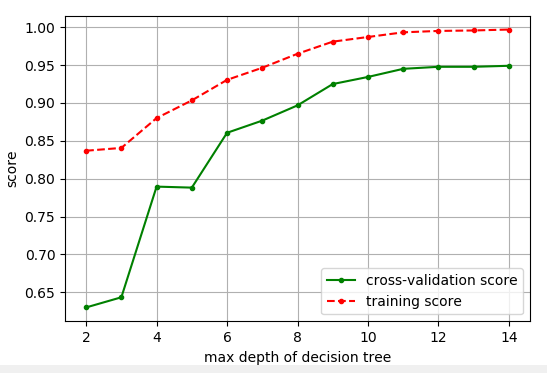
此时746为训练集，测试集为：

1625：

Impens:

模型调优，class\_weight=’balanced’来分配类别不平衡的数据的权重，多的数据则权重小些，少的则权重大些。开始是调节单个参数max\_depth,这里训练集为1625，测试集为746，网格搜索法并可视化，得到的结果如下：

best param: 14; best score: 0.9490616621983914



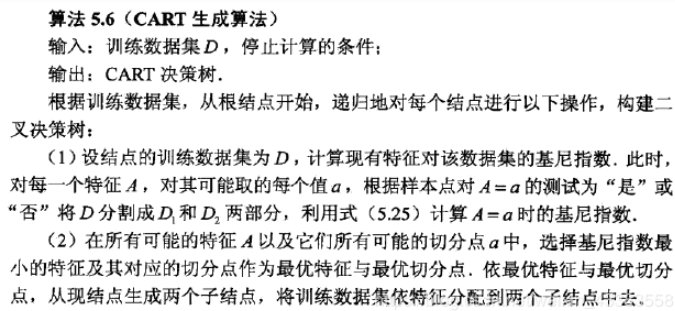
于是接着考虑为了模型的可靠性与有效性，便多调节了几个参数：

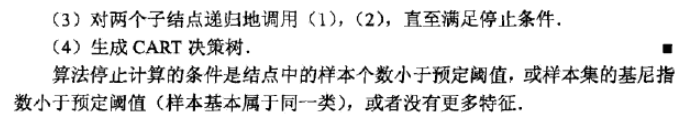
best param: {'criterion': 'gini', 'max\_depth': 5, 'min\_impurity\_decrease': 0.0}

best score: 0.8375384615384615

1. Cart手写

作为分类决策树时，需要做到基尼指数最小化。





所以代码中的步骤是——将数据数值化并离散化🡪计算Gini指数值，选择基尼指数小的作为最佳特征分支点 🡪 根据CART算法二分法划分数据集🡪决策树的生成🡪剪枝(预剪枝+后剪枝)

* 将4个数据集分别拿来训练并测试，



746Data: 最优特征：x5,

1625Data: 最优特征：x5,

impenseData: 最优特征：x5,

schillingData: 最优特征：x1,

1. 总结：
2. 对比ID3,c4.5的决策树算法，CART算法更加全面性，因为它既可以用来做回归，也可以用来做分类，在针对离散型、连续型数据时采取不同的处理方式;
3. 对比logistic回归，在这个氨基酸的数据集上，决策树的效果的总体效果要优于Logistic Re,说明决策树针对离散型的数据处理的效果更好。