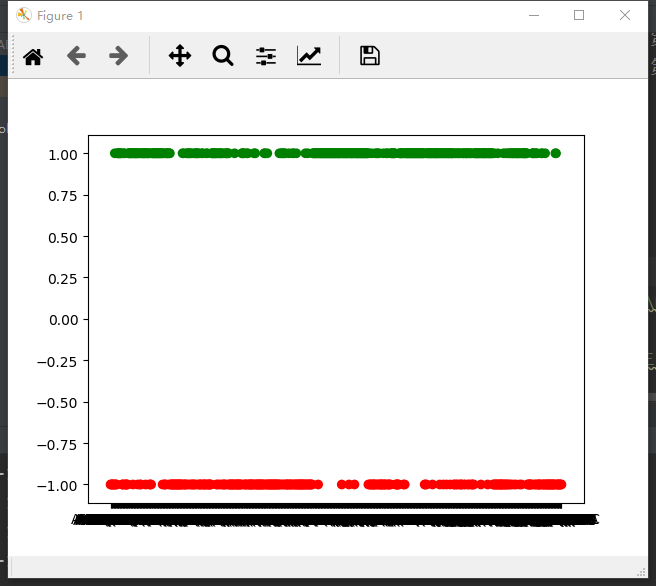
第3次训练——决策树

要求：利用决策树完成数据集[https://archive.ics.uci.edu/ml/datasets/HIV-1+protease+cleavage的二分类任务。 优先做出ID3](https://archive.ics.uci.edu/ml/datasets/HIV-1+protease+cleavage的二分类任务。%20优先做出ID3)、C4.5算法，对比其差异。争取做出CART算法。思考树与逻辑回归做分类的区别。

1. data analysis

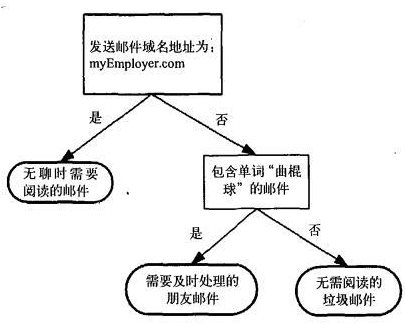
该数据共包含四个数据集，内容为：八聚体(Octamers)、label，该标签指示该字符串是否代表肽(或蛋白质)中HIV-1蛋白酶切割它的位点(是则为+1，不是为-1)；代表氨基酸的字母有：AR ND CQ EG HI LK MF PS TW YV；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| dataset | Octamers数量 | 裂解数 | 未裂解数 | 是否线性可分 |
| 746 | 746 | 401 | 345 | Y |
| 1625 | 1625 | 374 | 1251 | Y |
| Schilling | 3272 | 434 | 2838 | Y |
| Impens | 947 | 149 | 798 | Y |



将数据进行可视化，四个数据集的结果图都类似上图所示，其中绿色点为label=1的点，红色为-1的点，这四个数据集都是明显线性可分的。

实验方法原理：决策树是一种自顶向下的递归方法，其基本思想是：以信息熵()为度量构造一棵熵值下降最快的树，到叶子节点处熵为零。内部节点表示一个特征or属性(即条件)，叶子结点表示所属类别(即结论)。例如下图为决策树可视化后的结果：



|  |  |
| --- | --- |
| 构造步骤 | 特征选择、决策树的生成、修剪(自下而上)，决策树学习常用算法有：ID3、C4.5、CART |
| 输入 | 训练集：D={(x1,y1), (x2,y2), …, (xn,yn),},其中n为特征个数，y为类别(1、-1) |
| 输出 | 决策树模型 |
| 损失函数 | 正则化的极大似然函数 |
| 优化方法 | 自下而上的剪枝 |
| 可视化 | Sklearn.Graphviz |

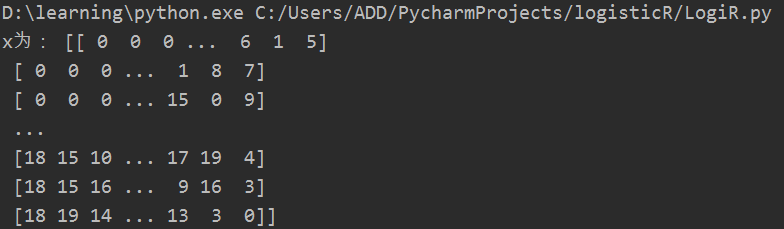
|  |  |  |  |
| --- | --- | --- | --- |
| 决策树生成算法 | 核心思想 | 公式 | 特点 |
| ID3 | 各个节点应用样本集合D对特征A的信息增益进行特征选择 | 是D中特征A取第个值的样本子集 | 1. 离散属性； 2. 熵越小，信息纯度越大； 3. 信息增益 = 划分前熵 - 划分后熵 |
| C4.5 | 样本集合D对特征A的信息增益比 |  | 1、先将特征取值排序，以连续两个值中间值作为划分标准  2、信息增益比 = 信息增益/ 划分前熵 |
| CART | 样本集合D的基尼指数 | K为类的个数，Ck是D中属于第k类的样本子集  特征A条件小集合D的基尼指数： |  |

1. 设计实验
2. 划分数据、数据处理

数据的训练组合有很多种方式，此次做法是：选择其中一个数据集进行train，剩下的三个数据集进行test。

此外，由于属性x为字符串，这里要转化为数值量，’ARNDCQEGHILKMFPSTWYV’分别对应’0、1、2、3……19’,每一个字母对应于一个one-hot编码，每个八聚体则为每个字母的数字向量拼接而成的一个8\*20的一个向量矩阵。

X的数据结果为：



|  |  |  |
| --- | --- | --- |
| 数据集 | y分布比例(-1：1) | 处理后的y |
| 746 | 344：402 | In total 746 |
| 1625 | 1250：375 | In total 750，375：375 |
| impens | 798：149 | In total 298，149：149 |
| Schilling | 2838：434 | In total 868，434：434 |

调包跑数据 (DecisionTreeClf默认的是基尼指数)

* ID3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试集  训练集 | 746 | 1625 | impens | schilling |
| 746 | 1.0 | 0.866 | 0.783/0.777 | 0.789/0.778 |
| 1625 | 0.814 | 1.0 | 0.769/0.771 | 0.779/0.785 |
| impens | 0.558 | 0.558 | 1.0 | 0.801 |
| schilling | 0.576 | 0.579 | 0.821 | 1.0 |

* CART

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试集  训练集 | 746 | 1625 | impens | schilling |
| 746 | 1.0 | 0.854 | 0.759/0.769 | 0.767/0.753 |
| 1625 | 0.829 | 1.0 | 0.765 | 0.759 |
| impens | 0.614 | 0.586 | 1.0 | 0.789 |
| schilling | 0.595 | 0.607 | 0.818 | 1.0 |

注：这里斜线后面的数字表示数据经过下采样之后的调包结果值。

1. 手写模型

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试集  训练集 | 746(epsilon=0.00001) | 1625 | impens | schilling |
| 746 | \ | 0.863 | 0.769 | 0.762 |
| 1625 | 0.835 | \ | 0.765 | 0.741 |
| impens | 0.754 | 0.586 | \ | 0.765 |
| schilling | 0.721 | 0.694 | 0.837 | \ |

ID3手写代码：首先先建立一棵树的类代码，然后计算熵，计算条件熵，计算信息增益，把信息增益作为最优特征选择，然后由下至上进行剪枝，得到一棵最优的决策树。但是调试代码debug,不知道哪里出现问题，始终无法

结论： 1、数据的预处理：将字符串转化为数值型，并且平衡分类数据。由于采用的下采样，使得数据的信息比较严重，在不同的模型中分数均会普遍降低，考虑应该还是使用加权重的方法来平衡数据。

2、手写算法ID3：建立树的过程采用递归方式，由上至下，选择信息增益最大的属性作为最优特征放在根节点的位置。最后进行剪枝的操作。剪枝，往往从已生成的树上剪掉一些叶结点或叶结点以上的子树，并将其父结点或根结点作为新的叶结点，从而简化生成的决策树。