# 决策树

## 数据集

数据集来源：

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

依然是上次逻辑回归的数据集，分类为二分类

## 数据集的处理

考虑到数据分类样本不均，处理方式为：过采样小样本，欠采样大样本。本次实验采用过采样小样本（采用直接复制小类样本，形成数量上的均衡），经统计如下：

|  |  |  |
| --- | --- | --- |
| 数据 | 正类（1） | 负类（0） |
| X1 | 402 | 344 |
| X2 | 375 | 1250 |
| X3 | 149 | 798 |
| X4 | 434 | 2838 |

处理后如下：

|  |  |  |
| --- | --- | --- |
| 数据 | 正类（1） | 负类（0） |
| X1 | 402 | 344 |
| X2 | 375\*3=1125 | 1250 |
| X3 | 149\*5=745 | 798 |
| X4 | 434\*6=2604 | 2838 |

这种方法实现简单，但会由于数据较为单一而容易造成过拟合。

其他方法如下：

（1）、对原数据的权值进行改变：

为少数类样本赋予更大的权值，为多数类样本赋予较小的权值。例如scikit-learn中的svm算法（penalized-SVM），可以手动设置权重。

（2）、通过组合集成的方法：

将多数类数据随机分成少数类数据的量N份，每一份与全部的少数类数据一起训练成一个分类器，经过反复训练会产生多个分类器，最后再用组合的方式（bagging或者boosting）对分类器进行组合，得到更好的预测效果。

依然采用对每个氨基酸字母进行one-hot的编码，然后拼接每个氨基酸的方式得到数据的表示。数据特征有160维。

## 实验过程

#### 调包实现

需要用到的是from **sklearn.tree** import **DecisionTreeClassifier**

过程为实例化决策树，对数据进行拟合，最后预测，如下：

dec=DecisionTreeClassifier()

dec.fit(X1,Y1)

y\_pred=dec.predict(X3)

调包结果如下(准确率(预测对的样本和总样本之比))：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train  Test | X1 | X2 | X3 | X4 |
| X1 | 1 | 0.97 | 0.67 | 0.68 |
| X2 | 0.9 | 1 | 0.72 | 0.71 |
| X3 | 0.71 | 0.74 | 1 | 0.69 |
| X4 | 0.70 | 0.66 | 0.69 | 1 |

结论：用X1、X2训练出来的结果比X3和X4训练出来的结果要好。

#### 用numpy实现

##### ID3算法

1. 对当前训练集，计算各个属性的信息增益。
2. 选择信息增益最大的属性Ak，作为根结点。
3. 把在Ak处取值相同的例子归于同一子集，作为该结点的一个树枝，Ak取几个值就得几个子集。
4. 若在某个子集中的所有样本都属于同一个类型，则该分支上类型号作为叶子结点。
5. 对于同时含有多种类型的子集，则递归调用该算法来完成树的构造。

具体实现：

def calcShannonEnt(dataSet) 计算给定数据集的经验熵

def splitDataSet(dataSet, axis, value): 根据数据集dataSet中的axis特征的值等于value来划分数据

def chooseBestFeatureToSplit(dataSet):根据数据集dataSet得到最优的那个特征

def majorityCnt(classList): 根据classlist统计出现最多的类标签

def createTree(dataSet, labels, featLabels):递归构建决策树

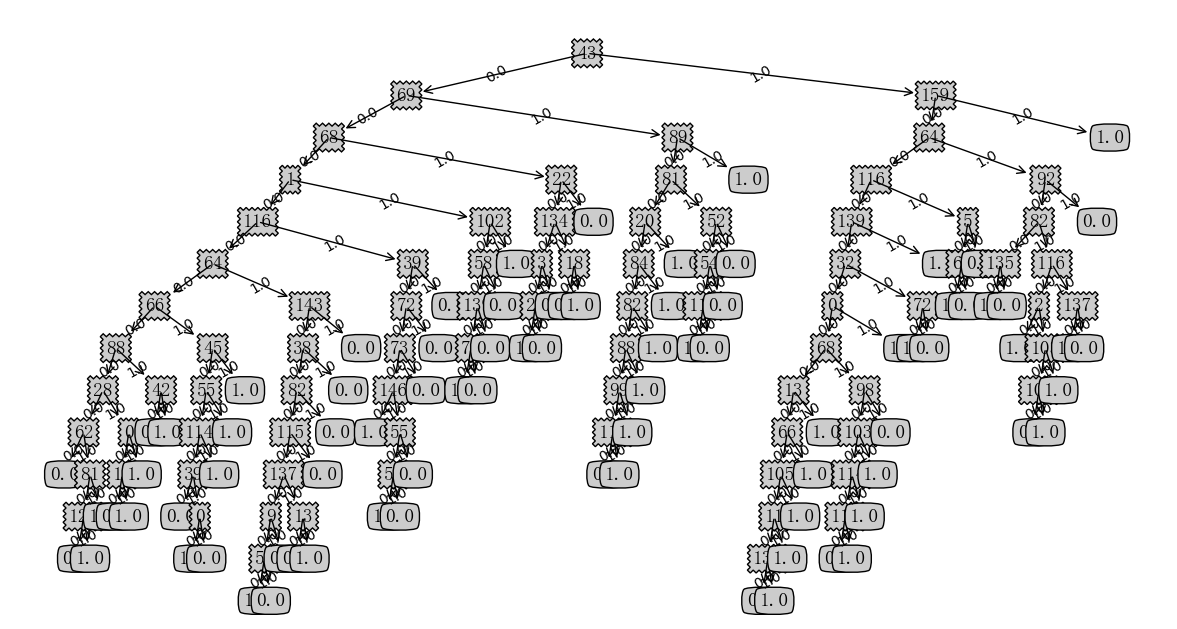
1. 先取出dataset中所有的分类标签
2. 判断，如果类别完全相同就停止划分数据
3. 判断，遍历完所有特征时返回出现次数最多的类标签
4. 用字典构建最好的特征的字典来作为树
5. 删除已经使用过的特征标签
6. 得到dataSet中的最优的特征的属性值
7. 去重这个属性值
8. 遍历每个属性值
9. 递归调用createTree
10. 返回mytree

实验结果：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train  Test | X1 | X2 | X3 | X4 |
| X1 | 1 | 0.97 | 0.62 | 0.68 |
| X2 | 0.91 | 1 | 0.66 | 0.71 |
| X3 | 0.70 | 0.64 | 1 | 0.67 |
| X4 | 0.69 | 0.64 | 0.67 | 1 |

总结：与调包相比，测试准确率最大相差0.1。

可视化决策树：



##### C4.5

在生成树的过程中，用信息增益比来选择特征。

将信息增益与训练数据集D关于特征A的值的熵Ha(D)之比。

主要问题是：以信息增益作为划分训练数据集的特征，存在偏向于选择取值较多的特征。

#### Cart(分类与回归树)

实验结果：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train  Test | X1 | X2 | X3 | X4 |
| X1 | 1 | 0.96 | 0.61 | 0.71 |
| X2 | 0.91 | 1 | 0.65 | 0.74 |
| X3 | 0.71 | 0.65 | 1 | 0.66 |
| X4 | 0.70 | 0.65 | 0.67 | 1 |

## 4．总结

与逻辑回归需要初始化W，设置学习率与遍历次数这样的超参相比，决策树的分类更注重于训练数据，同时也更容易过拟合。从整体的预测准确率分析，逻辑回归得到的预测效果要比决策树的准确率高。