决策树

一．简介

决策树(Decision Tree）是在已知各种情况发生概率的[基础](https://baike.baidu.com/item/%E5%9F%BA%E7%A1%80/32794)上，通过构成决策树来求取净现值的[期望](https://baike.baidu.com/item/%E6%9C%9F%E6%9C%9B/35704)值大于等于零的概率，评价项目风险，判断其可行性的决策分析方法，是直观运用概率分析的一种图解法。由于这种决策分支画成图形很像一棵树的枝干，故称决策树。在机器学习中，决策树是一个预测模型，他代表的是对象属性与对象值之间的一种映射关系。Entropy = 系统的凌乱程度，使用算法[ID3](https://baike.baidu.com/item/ID3), [C4.5](https://baike.baidu.com/item/C4.5)和C5.0生成树算法使用熵。这一度量是基于信息学理论中熵的概念。

决策树是一种树形结构，其中每个内部节点表示一个属性上的测试，每个分支代表一个测试输出，每个叶节点代表一种类别。

分类树（决策树）是一种十分常用的分类方法。他是一种监管学习，所谓监管学习就是给定一堆样本，每个样本都有一组属性和一个类别，这些类别是事先确定的，那么通过学习得到一个分类器，这个分类器能够对新出现的对象给出正确的分类。这样的机器学习就被称之为监督学习。

二．决策树（Decision Tree）特性和使用

1.决策树的特性

决策树（Decision Tree）是一种简单但是广泛使用的分类器。通过训练数据构建决策树，可以高效地对未知的数据进行分类。决策数有两大优点：

1）决策树模型可以读性好，具有描述性，有助于人工分析；

2）效率高，决策树只需要一次构建，反复使用，每一次预测的最大计算次数不超过决

策树的深度。

决策树优点：计算复杂度不高，输出结果易于理解，对中间值的缺失不敏感，可以

处理不相关的特征数据。缺点：可能产生过度匹配问题（过拟合）。

整体思路：大原则是将无序的数据变得更加有序。从当前可供学习的数据集中，

选择一个特性，根据这个特性划分出来的数据分类，可以获得最高的信息增益（在

划分数据集前后信息发生的变化）。信息增益是熵的减少，或者是数据无序度的减

少。在此划分之后，对划分出的各个分类再次进行算法，直到所有分类中均为同一

类元素，或所有特性均已使用。

2.sk-learn 中决策树的使用

sklearn 中提供了决策树的相关方法，即 DecisonTreeClasifier 分类器，它能够对

数据进行多分类，具体定义及部分参数详细含义如下表所示，详细可查看项目主页:

http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

|  |  |  |
| --- | --- | --- |
| class\**sklearn.tree.DecisionTreeClassifier**(criterion=’gini’,splitter=’best’,max\_depth=None,min\_samples\_split=2,min\_samples\_leaf=1,min\_weight\_fraction\_leaf=0.0,  max\_features=None,random\_state=None,max\_leaf\_nodes=None,class\_weight=None,presort=False) | | |
| 参数  说明 | criterion:string | 衡量分类的质量。支持的标准有”gini”(默认)代表的是Gini impurity与”entropy”代表的是information gain |
| splitter:string | 一种用来在节点中选择分类的策略。支持的策略有”best”(默认)，选择最好的分类；”random”选择最好的随机分类。 |
| max\_depth:int or None | 树的最大深度。如果是”None”(默认)，则节点会一直扩展直到所有的叶子都是纯的或者所有的叶子节点都包含少于main\_samples\_split个样本点。忽视max\_leaf\_nodes是不是为None。 |
| persort:bool | 是否预分类数据以加速训练时最好分类的查找。在有大数据集的决策树中，如果设为true可能会减慢训练的过程。当使用一个小数据集或者一个深度受限的决策树中，可以减速训练的过程。默认False |

和其他分类器一样，DecisionTreeClassifier 有两个向量输入：X，稀疏或密集，大小为[n\_sample,n\_fearure],存放训练样本； Y，值为整型，大小为[n\_sample],存放训练样本的分类标签。但由于 DecisionTreeClassifier 不支持输入文本属性和文本标签，因此需要将原始数据集中的文本标签转化为数字标签，及 X、Y 应为数字矩阵。接着将 X、Y传给 fit()函数进行训练，得到的模型即可对样本进行预测。

实验：决策树的简单实用

实验环境：python3

实验步骤：

1. 安装python3（此处省略，去官网下载即可）
2. 安装python3后，打开cmd，输入pip install sklearn安装此次实验所需库
3. 输入代码，如下

from sklearn import tree

X = [[0,0],[1,1]]

Y = [0,1]

clf = tree.DecisionTreeClassifier() #初始化

clf = clf.fit(X,Y) #根据XY数据训练决策树

clf.predict([[2.,2.]]) #根据训练出的模型预测样本

运行结果为：

[1]