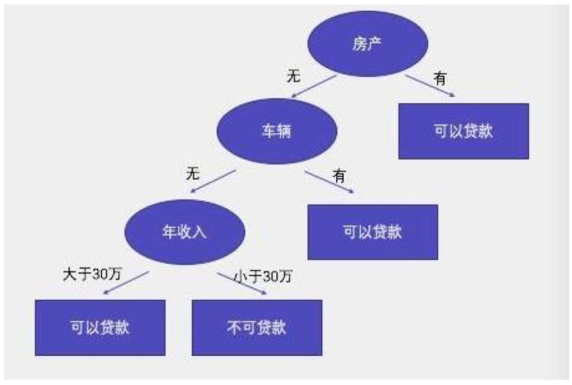
**对决策树的一些心得**

决策树是通过对训练样本的学习，并建立分类规则，然后依据分类规则，对新样本数据进行分类预测。

可视化如下：



其中，第一个节点叫跟节点，各个分支的最终结果节点叫叶子节点，其余的都归类于非叶子节点

关于分类规则，共有三类标准：

1. ID3算法

利用熵值来决定特征的混乱程度，公式为：

A picture containing text

Description automatically generated

举个例子：

Table

Description automatically generated with medium confidence

14天中，9天打球，5天不打球，熵为：

-9/14 〖log〗*2 9/14-5/14 〖log〗*2 5/14=0.94

再对天气进行举例：

Diagram

Description automatically generated

晴天【5天】的熵：

Overcast（阴天）【4天】的熵：

雨天【5天】的熵：

那么，天气的熵为：

5/14\**0.971+4/14\**0+5/14\*0.971=0.693

**则信息增益为：0.940-0.693=0.247，信息增益越大，表示该特征纯度越高**

1. C4.5算法

特征对训练数据集的信息增益比定义为其信息增益与训练数据集的经验熵之比,公式如下：

Text, letter

Description automatically generated

利用前面的数据，天气的自身熵值为

信息增益率为：0.247/1.577 = 0.1566‘

**信息增益率越大，表示该特征纯度越高**

1. cart 算法

这个算法是利用基尼系数来计算特征的混乱程度

A picture containing text, watch, clock, gauge

Description automatically generated

例：

A picture containing text

Description automatically generatedTable

Description automatically generatedText, letter

Description automatically generated

青年(5人，2人贷款)的基尼系数：

2/5\**(1-2/5)+3/5\**(1-3/5)=0.48 2\**2/5\**(1-2/5)=0.48

非青年（10人，7人贷款）的基尼系数： 2*7/10\**(1-7/10)=0.42 在A1=1条件下，D的基尼系数： 5/15\**0.48+ 10/15\**0.42=0.44 总公式： 5/15\**[2\**2/5\**(1-2/5)]+ 10/15\**[2\**7/10\**(1-7/10)]=0.44

**基尼系数越小表示纯度越高**

代码如下：

datas = pd.read\_excel(r'C:\Users\14272\Documents\Tencent Files\1427236256\FileRecv\王老师Python数据分析机器学习实战课程资料\Python数据分析与机器学习\8.决策树之分类树\决策树算法\决策树\电信客户流失数据.xlsx')  
​  
#将变量与结果分开  
data = datas.ix[:,:-1]  
target = datas.ix[:,-1]  
​  
"""  
对导入模块进行划分  
"""  
from sklearn.model\_selection import train\_test\_split  
​  
data\_train, data\_test, target\_train, target\_test = train\_test\_split(data,target,test\_size=0.2,random\_state=42)  
​  
#定义决策树  
from sklearn import tree  
dtr = tree.DecisionTreeClassifier(criterion='gini', max\_depth=6, random\_state=42)  
dtr.fit(data\_train, target\_train)

`\***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*, *min\_impurity\_decrease=0.0*, *min\_impurity\_split=None*, *class\_weight=None*, *presort=False*``)```

**树模型参数\***\*:\*\*

**1.criterion** :gini or entropy **【采用基尼系数还是熵值衡量\***\*,默认基尼系数】 **2.splitter:** best or random 前者是在所有特征中找最好的切分点 后者是在部分特征中（数据量大的时候）**【默认\***\*best，】

**3.max\_features:**（所有），log2，sqrt，N 特征小于50的时候一般使用所有的 **【默认取所有特征，】**

**4.max\_depth:** 数据少或者特征少的时候可以不管这个值，如果模型样本量多，特征也多的情况下，可以尝试限制下。如果没有设置，那么将会把节点完全展开，直到所有的叶子节点都是纯的，或者达到最小叶子节点的个数阈值设置。

**5.min\_samples\_split** **：**（内部节点再划分所需最小样本数），如果某节点的样本数少于min\_samples\_split，则不会继续再尝试选择最优特征来进行划分，如果样本量不大，不需要管这个值。如果样本量数量级非常大，则推荐增大这个值。**【控制内部节点分裂的情况；假设\***\*<10，那么分裂的数量小于10就不会再次分裂了，默认2个】

**6.min\_samples\_leaf** **：**（叶子节点最少样本数），这个值限制了叶子节点最少的样本数，如果某叶子节点数目小于样本数，则会和兄弟节点一起被剪枝，如果样本量不大，不需要管这个值**【先构成决策树，再剪枝，当小于某个设定值后，删除此节点以及此节点的分支节点】**

**7.min\_weight\_fraction\_leaf\***\*：\*\*（叶子节点最小的样本权重和），这个值限制了叶子节点所有样本权重和的最小值，如果小于这个值，则会和兄弟节点一起被剪枝，默认是0，就是不考虑权重问题。一般来说，如果我们有较多样本有缺失值，或者分类树样本的分布类别偏差很大，就会引入样本权重，这时我们就要注意这个值了。

**8.max\_leaf\_nodes** **：**（最大叶子节点数），通过限制最大叶子节点数，可以防止过拟合，默认是"None”，即不限制最大的叶子节点数。如果加了限制，算法会建立在最大叶子节点数内最优的决策树。如果特征不多，可以不考虑这个值，但是如果特征分成多的话，可以加以限制具体的值可以通过交叉验证得到。**【假设限制最大的叶子节点数为\***\*10个，那么就不会再次分裂了】\*\*

**9. min\_impurity\_decrease** **：这个参数有点观察**

**10.min\_impurity\_split\***\*：这个值限制了决策树的增长，如果某节点的不纯度(基尼系数，信息增益，均方差，绝对差)小于这个阈值则该节点不再生成子节点。即为叶子节点 。

**11.class\_weight** 指定样本各类别的的权重，主要是为了防止训练集某些类别的样本过多导致训练的决策树过于偏向这些类别。这里可以自己指定各个样本的权重如果使用“balanced”，则算法会自己计算权重，样本量少的类别所对应的样本权重会高。