# 西瓜挑选问题描述

夏天买西瓜时，一般先选瓜皮有光泽的（新鲜），再拍一拍选声音清脆的（成熟），这样挑出来的好瓜的可能就比较大了。那么如何对西瓜进行挑选呢？

# 决策树

决策树是一种基于树结构来进行决策的分类算法，我们希望从给定的训练数据集学得一个模型（即决策树），用该模型对新样本分类。决策树可以非常直观展现分类的过程和结果，一旦模型构建成功，对新样本的分类效率也相当高。最经典的决策树算法有ID3、C4.5、CART，其中ID3算法是最早被提出的，它可以处理离散属性样本的分类，C4.5和CART算法则可以处理更加复杂的分类问题。

# 决策树训练西瓜数据实验：

## Step1:收集数据集

编号,色泽,根蒂,敲声,纹理,脐部,触感,好瓜  
1,青绿,蜷缩,浊响,清晰,凹陷,硬滑,是  
2,乌黑,蜷缩,沉闷,清晰,凹陷,硬滑,是  
3,乌黑,蜷缩,浊响,清晰,凹陷,硬滑,是  
4,青绿,蜷缩,沉闷,清晰,凹陷,硬滑,是  
5,浅白,蜷缩,浊响,清晰,凹陷,硬滑,是  
6,青绿,稍蜷,浊响,清晰,稍凹,软粘,是  
7,乌黑,稍蜷,浊响,稍糊,稍凹,软粘,是  
8,乌黑,稍蜷,浊响,清晰,稍凹,硬滑,是  
9,乌黑,稍蜷,沉闷,稍糊,稍凹,硬滑,否  
10,青绿,硬挺,清脆,清晰,平坦,软粘,否  
11,浅白,硬挺,清脆,模糊,平坦,硬滑,否  
12,浅白,蜷缩,浊响,模糊,平坦,软粘,否  
13,青绿,稍蜷,浊响,稍糊,凹陷,硬滑,否  
14,浅白,稍蜷,沉闷,稍糊,凹陷,硬滑,否  
15,乌黑,稍蜷,浊响,清晰,稍凹,软粘,否  
16,浅白,蜷缩,浊响,模糊,平坦,硬滑,否  
17,青绿,蜷缩,沉闷,稍糊,稍凹,硬滑,否

## **Step2：预处理数据**

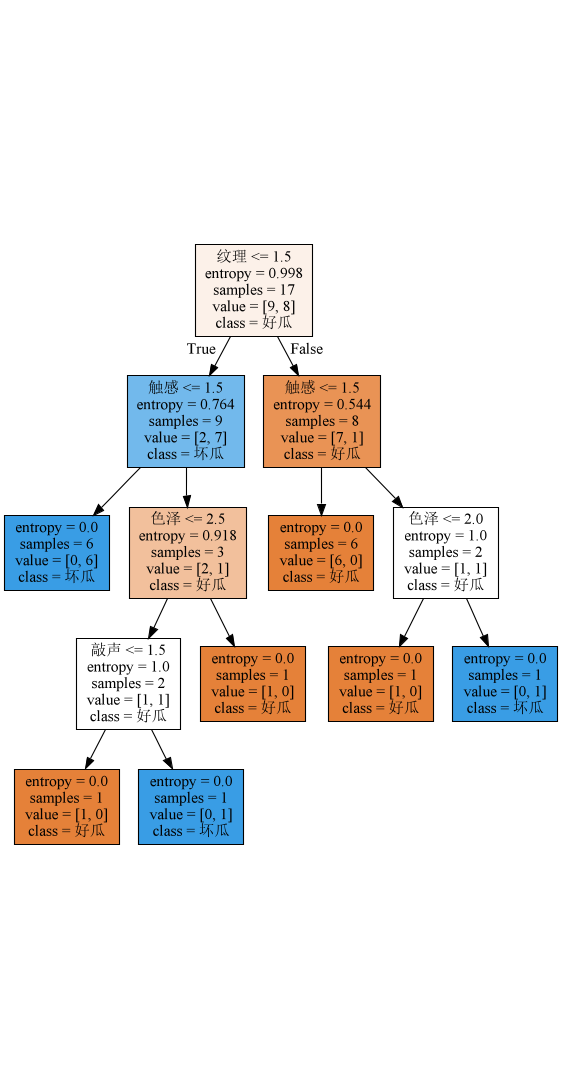
# 读取西瓜数据集  
df = pd.read\_table(r'DATA.txt', encoding='utf8', delimiter=',', index\_col=0)  
# 处理汉字问题  
'''  
属性：  
色泽 1-3代表 浅白 青绿 乌黑 根蒂 1-3代表 稍蜷 蜷缩 硬挺  
敲声 1-3代表 清脆 浊响 沉闷 纹理 1-3代表 清晰 稍糊 模糊  
脐部 1-3代表 平坦 稍凹 凹陷 触感 1-2代表 硬滑 软粘  
标签：  
好瓜 1代表 是 0 代表 不是  
'''  
df['色泽'] = df['色泽'].map({'浅白': 1, '青绿': 2, '乌黑': 3})  
df['根蒂'] = df['根蒂'].map({'稍蜷': 1, '蜷缩': 2, '硬挺': 3})  
df['敲声'] = df['敲声'].map({'清脆': 1, '浊响': 2, '沉闷': 3})  
df['纹理'] = df['纹理'].map({'清晰': 1, '稍糊': 2, '模糊': 3})  
df['脐部'] = df['脐部'].map({'平坦': 1, '稍凹': 2, '凹陷': 3})  
df['触感'] = np.where(df['触感'] == "硬滑", 1, 2)  
df['好瓜'] = np.where(df['好瓜'] == "是", 1, 0)  
# 由于西瓜数据集样本比较少，所以不划分数据集，将所有的西瓜数据用来训练模型  
Xtrain = df.iloc[:, :-1]  
Xtrain = np.array(Xtrain)  
Ytrain = df.iloc[:, -1]

## **Step3：训练决策树模型**

# 采用ID3算法，利用信息熵构建决策树模型  
# clf = tree.DecisionTreeClassifier(criterion="gini") 采用CART算法利用GINI来构建决策树模型  
clf = tree.DecisionTreeClassifier(criterion="entropy")  
  
# 训练模型  
clf = clf.fit(Xtrain, Ytrain)

## **Step4：画出决策树**

# 绘制决策树的图形  
feature\_names = ["色泽", "根蒂", "敲声", "纹理", "脐部", "触感"]  
dot\_data = tree.export\_graphviz(clf, feature\_names=feature\_names, class\_names=["好瓜", "坏瓜"], filled=True, rounded=False)  
# 保存图片  
graph = graphviz.Source(dot\_data).render(view=True)

此图为ID3算法实现的决策树：

此图为CART算法实现的决策树：

