浙江工业大学

数据挖掘实验

计算机科学与技术学院

决策树和朴素贝叶斯

一、实验目的

熟悉决策树构造以及剪枝的基本方法。

二、实验内容

本实验数据集：breast_cancer2。注意，该数据集与之前从 sklearn 中导入的 breast_cancer **是两个不同的数据集。breast_cancer2 包含 9 个属性，286 个样本，两个类别。第一列为类别标签，分别是乳腺癌复发（recurrence-events）和未复发（no-recurrence-events）。**9 个属性的取值范围如下：

```
age: 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99.
menopause: lt40, ge40, premeno.
tumor-size: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44,
            45-49, 50-54, 55-59.
inv-nodes: 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29,
           30-32, 33-35, 36-39.
node-caps: yes, no.
deg-malig: 1, 2, 3.
breast: left, right.
breast-quad: left-up, left-low, right-up, right-low, central.
irradiat:  yes, no.
```

1.数据集准备

1.1 从文件导入 breast_cancer2 数据集:x=pd.read_csv('D:/数据集/breast_cancer2',*header=None)*，查看数据集基本信息、缺失情况、以及每个类别大小。

1.2 用实验 4 中的分层留出法 StrateSplit（20%测试集，设随机种子为 42）把数据集划分为训练集(X_train, y_train)和测试集(X_test,y_test)。

1.3 实际上该数据集包含少量用“？”标记的缺失值，用 na 代替’?’后，重新查看训练集缺失情况，并用对应属性的众数替换缺失值，得到 X_train_fill。

1.4 对 X_train_fill 进行标准化(均值为 0,标准差为 1)得到 X_train_fill_std。

1.5 对测试集做同样的预处理：用 na 代替’?’,基于**训练集**各属性的众数对测试集进行缺失值填充得到 X_test_fill，以及基于**训练集**的均值和标准差对测试

集进行标准化得到 X_test_fill_std。

1.6 把以上 2 个版本的训练和测试集的数据和标签拼接并保存分别保存到文件， 如 train=pd.concat([X_train_fill, y_train],axis=1) ，
train.to_csv("train_filled.csv")

1.7 **思考与分析**：为什么要先分割然后对测试集用训练集同样方法（用训练集各个特征的众数以及均值和标准差）进行预处理，而不是对整个数据集预处理后进行分割？

2.决策树构造与测试

2.1 以基尼指数(gini-index)为属性划分标准用 X_train_fill 构建决策树，并画出决策树。

2.2 用以上决策树得到测试集 X_test_fill 上的 F1。

2.3 以信息增益(information gain)为划分标准，重复 2.1 和 2.2。比较结果是否有变化。

2.4 以基尼指数(gini-index)为属性划分标准用 X_train_fill_std 构建决策树，用以上决策树得到测试集 X_test_fill_std 上的 F1，对比 2.2 中结果讨论决策树对数据标准化是否敏感。

3.剪枝（选做）

3.1 通过以下两种方式进行简单预剪枝：a. 设置 **min_samples_split(默认为 2，一个中间节点被划分需要包含的最小样本数目，如果小于该值，则即便不纯也不再划分)**。b. 设置 **min_impurity_split(默认为 0，一个中间节点的不纯程度低于该值就不再生长，即用最多类标记该叶子节点)**。

3.2 画出剪枝后的决策树，对比剪枝前后的效果，讨论剪枝的用途。

3.3 用剪枝后的决策树对上面同一测试集标签进行预测并计算 F1，并分析什么时候剪枝会降低模型准确率。

4.朴素贝叶斯分类器以及拉普拉斯修正

4.1 从文件中导入划分以及处理好的 2 个版本的 breast_cancer2 的训练集和测试集,即填充但没有标准化的，以及填充且标准化的。

4.2 用未标准化版本，编程实现朴素贝叶斯分类（先建模再预测的模式），预测测试集结果并计算 F1。

4.3 进行拉普拉斯修正后，预测测试集结果并计算 F1,比较两次结果，分析拉普拉斯修正的作用。

4.4 用标准化版本，重复 1.2 实验内容，对比结果，讨论朴素贝叶斯是否对标准化敏感。

==========

提示和参考信息：

大家可以参考课件上的决策树构造基本算法（ID3）编写以上决策树构造和剪枝函数。也可以调用 **sklearn.tree.DecisionTreeClassifier 进行决策树构建。**

调用该算法要注意：DecisionTreeClassifier 为 CART 算法，默认处理连续型属性，如果给定数据包含类别型属性，需要转换；DecisionTreeClassifier 默认把所有属性预处理成二值属性后再构建决策树，即每个中间节点只有两个分支。

sklearn 中决策树分类 DecisionTreeClassifier 例子, 包括决策树构建\画图\测试，**注意各参数的作用和默认设置。**

https://scikit-learn.org/stable/modules/tree.html

例子：

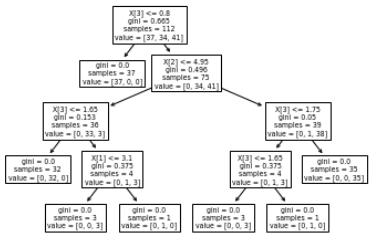
In [28]:
```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn import tree
from matplotlib import pyplot as plt

iris = datasets.load_iris()
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

clf = tree.DecisionTreeClassifier( random_state=0)
clf.fit(X_train, y_train)
```

Out[28]:
```
DecisionTreeClassifier(random_state=0)
```

In [30]:
```
tree.plot_tree(clf)
plt.show()
```



更多决策树可视化用：graphviz