作业四

要求：天气因素有温度、湿度和刮风等，通过给出数据，使用决策树算法学习分类，输出一个人是运动和不运动与天气之间的规则树。

训练集和测试集可以自由定义，另外需要对温度和湿度进行概化，将数值变为概括性表述，比如温度热，温，凉爽，湿度变为高，中。

In [74]:

```
from sklearn import tree
from sklearn.model_selection import train_test_split
import pandas as pd
import graphviz
import numpy as np
```

数据预处理

数据读取

In [75]:

```
df = pd.read_excel('data.xlsx', index_col=None)
df
```

Out[75]:

	天气	温度	湿度	风况	运动
0	晴	85	85	无	不适合
1	晴	80	90	有	不适合
2	多云	83	78	无	适合
3	有雨	70	96	无	适合
4	有雨	68	80	无	适合
5	有雨	65	70	有	不适合
6	多云	64	65	有	适合
7	晴	72	95	无	不适合
8	晴	69	70	无	适合
9	有雨	75	80	无	适合
10	晴	75	70	有	适合
11	多云	72	90	有	适合
12	多云	81	75	无	适合
13	有雨	71	80	有	不适合

文字指标量化

为了后续决策树的计算，需要把文字指标进行量化，下面进行转换：

天气——晴-0，多元-1，有雨-2

风况——无-0，有-1

运动——不适合-0，适合-1

In [76]:

```
df['天气'] = df['天气'].replace("晴", 0)
df['天气'] = df['天气'].replace("多云", 1)
df['天气'] = df['天气'].replace("有雨", 2)
df['风况'] = df['风况'].replace("无", 0)
df['风况'] = df['风况'].replace("有", 1)
df['运动'] = df['运动'].replace("不适合", 0)
df['运动'] = df['运动'].replace("适合", 1)
```

温湿度概化

题目要求，将温湿度数值变为概括性表述。这里将温湿度进行概述并转化为数值，具体规则如下：

温度：<70-凉爽-0，70~80-温-1，>80-热-2

湿度：>80-高-1，<=80-中-0

In [77]:

```
df['温度'] = np.where(df['温度'] < 70, 0, df['温度'])
df['温度'] = np.where((df['温度'] < 80) & (df['温度'] >= 70), 1, df['温度'])
df['温度'] = np.where(df['温度'] >= 80, 2, df['温度'])
df['湿度'] = np.where(df['湿度'] > 80, 1, 0)
```

转换后的数据如下表所示：

In [78]:

```
df
```

Out[78]:

	天气	温度	湿度	风况	运动
0	0	2	1	0	0
1	0	2	1	1	0
2	1	2	0	0	1
3	2	1	1	0	1
4	2	0	0	0	1
5	2	0	0	1	0
6	1	0	0	1	1
7	0	1	1	0	0
8	0	0	0	0	1
9	2	1	0	0	1
10	0	1	0	1	1
11	1	1	1	1	1
12	1	2	0	0	1
13	2	1	0	1	0

数据集划分

根据7/3的比例划分训练集和测试集

In [128]:

```
data = df[['天气', '温度', '湿度', '风况']]
target = df['运动']
data = np.array(data)
target = np.array(target)
Xtrain, Xtest, Ytrain, Ytest = train_test_split(data, target, test_size=0.3)
```

决策树构建

这里决策树的标准选择基尼指数，最终得到分类准确率为60%

In [129]:

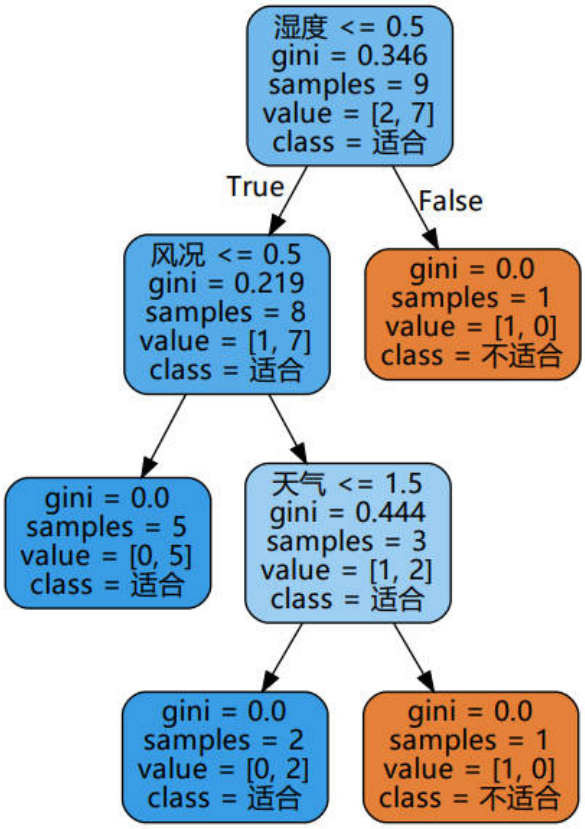
```
clf = tree.DecisionTreeClassifier(criterion="gini")
clf = clf.fit(Xtrain, Ytrain)
score = clf.score(Xtest, Ytest)
print(score)
```

```
0.6
```

可视化结果

In [130]:

```
feature_name = ['天气', '温度', '湿度', '风况']
dot_data = tree.export_graphviz(clf, feature_names=feature_name, class_names=["不适合", "适合"],
filled=True, rounded=True
                                )
graph = graphviz.Source(dot_data.replace(
    'helvetica', '"Microsoft YaHei"'), encoding='utf-8')
graph.view()
```



如图所示，湿度低于0.5(湿度中)，风况低于0.5(风况无)，天气低于1.5（天气晴或多云）适合运动。