决策树python代码详解

from math import log

import operator

def calcShannonEnt(dataSet): # 计算数据的熵(entropy)

numEntries=len(dataSet) # 数据条数

labelCounts={}

for featVec in dataSet:

currentLabel=featVec[-1] # 每行数据的最后一个字（类别）

if currentLabel not in labelCounts.keys():

labelCounts[currentLabel]=0

labelCounts[currentLabel]+=1 # 统计有多少个类以及每个类的数量

shannonEnt=0

for key in labelCounts:

prob=float(labelCounts[key])/numEntries # 计算单个类的熵值

shannonEnt-=prob\*log(prob,2) # 累加每个类的熵值

return shannonEnt

def createDataSet1(): # 创造示例数据

dataSet = [['长', '粗', '男'],

['短', '粗', '男'],

['短', '粗', '男'],

['长', '细', '女'],

['短', '细', '女'],

['短', '粗', '女'],

['长', '粗', '女'],

['长', '粗', '女']]

labels = ['头发','声音'] #两个特征

return dataSet,labels

def splitDataSet(dataSet,axis,value): # 按某个特征分类后的数据

retDataSet=[]

for featVec in dataSet:

if featVec[axis]==value: #若某个特征对应的值等于指定值

reducedFeatVec =featVec[:axis] #返回该特征之前的特征

reducedFeatVec.extend(featVec[axis+1:]) # 返回该特征之后的数据，并添加到之前获取的特征之后；相当于按照某个特征进行划分，划分后的样本（除去了作为划分的特征）

retDataSet.append(reducedFeatVec) #得到按特征值划分的子集

return retDataSet

def chooseBestFeatureToSplit(dataSet): # 选择最优的分类特征

numFeatures = len(dataSet[0])-1 #特征数

baseEntropy = calcShannonEnt(dataSet) # 原始的熵

bestInfoGain = 0

bestFeature = -1

for i in range(numFeatures):

featList = [example[i] for example in dataSet]

uniqueVals = set(featList) # set() 函数创建一个无序不重复元素集

newEntropy = 0

for value in uniqueVals:

subDataSet = splitDataSet(dataSet,i,value) #按特征值划分后的子集给subDataSet

prob =len(subDataSet)/float(len(dataSet))

newEntropy +=prob\*calcShannonEnt(subDataSet) # 由calcShannonEnt(subDataSet)得到按特征值分类后的信息熵，newEntropy得到该特征的信息熵

infoGain = baseEntropy - newEntropy # 原始熵与按特征分类后的熵的差值

if (infoGain>bestInfoGain): # 若按某特征划分后，信息增益最大的，则该特征为最优分类特征

bestInfoGain=infoGain

bestFeature = i

return bestFeature

# 如果数据集已经处理了所有属性，而此时类标签依然不是唯一的，需要决定如何定义该叶子节点，下面采用多数表决的方法决定该叶子节点分类

def majorityCnt(classList): #按分类后类别数量排序，比如：最后分类为2男1女，则判定为男；

classCount={}

for vote in classList:

if vote not in classCount.keys():#这里面主要是进行计数，分别为“男”和“女”两种类别的次数

classCount[vote]=0

classCount[vote]+=1

sortedClassCount = sorted(classCount.items(),key=operator.itemgetter(1),reverse=True)

return sortedClassCount[0][0]

def createTree(dataSet,labels):

classList=[example[-1] for example in dataSet] # 类别：男或女

if classList.count(classList[0])==len(classList): # 如果类别完全相同就停止划分

return classList[0]

if len(dataSet[0])==1:

return majorityCnt(classList) #若所有特征用完，剩下的类别不纯，则根据多数表决

bestFeat=chooseBestFeatureToSplit(dataSet) #选择最优特征，即第几列

bestFeatLabel=labels[bestFeat]

myTree={bestFeatLabel:{}} #分类结果以字典形式保存

del(labels[bestFeat]) #将对应的标签从标签列表中删除

featValues=[example[bestFeat] for example in dataSet]

uniqueVals=set(featValues)

for value in uniqueVals:

subLabels=labels[:] #去掉选择特征后的特征标签给subLabels

myTree[bestFeatLabel][value]=createTree(splitDataSet\

(dataSet,bestFeat,value),subLabels)

#这里的\相当于连接符，是为了换行编辑。

#利用递归实现树构建；splitDataSet(dataSet,bestFeat,value)去掉选择特征后留下的子数据集

return myTree

if \_\_name\_\_=='\_\_main\_\_':

dataSet, labels=createDataSet1() # 创造示列数据

print(createTree(dataSet, labels)) # 输出决策树模型结果

补充知识：

1. # \_\_name\_\_ 是当前模块名，当模块被直接运行时模块名为 \_\_main\_\_ 。这句话的意思就是，当模块被直接运行时，以下代码块将被运行，当模块是被导入时，代码块不被运行。

通过\_\_name\_\_的值，我们可以判断出该模块是作为脚本正在执行还是被其他模块导入，而根据这个判断，我们就可以选择性地执行代码。将只有模块作为脚本执行时的代码放入到if \_\_name\_\_ == '\_\_main\_\_':

如何该模块是作为其他模块导入，就不会执行if \_\_name\_\_ == '\_\_main\_\_':后的代码

1. Python count() 方法用于统计字符串里某个字符出现的次数。可选参数为在字符串搜索的开始与结束位置。

count()方法语法：

str.count(sub, start= 0,end=len(string))

sub -- 搜索的子字符串

start -- 字符串开始搜索的位置。默认为第一个字符,第一个字符索引值为0。

end -- 字符串中结束搜索的位置。字符中第一个字符的索引为 0。默认为字符串的最后一个位置。

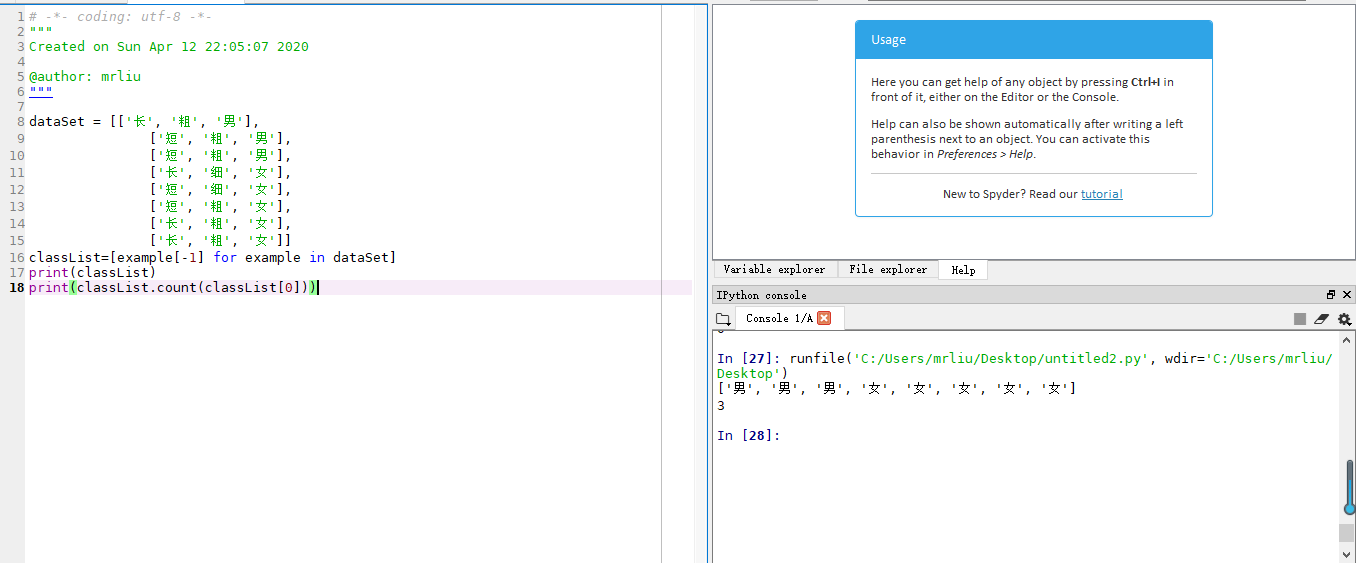
如果这段程序不懂

if classList.count(classList[0])==len(classList): # 如果类别完全相同就停止划分

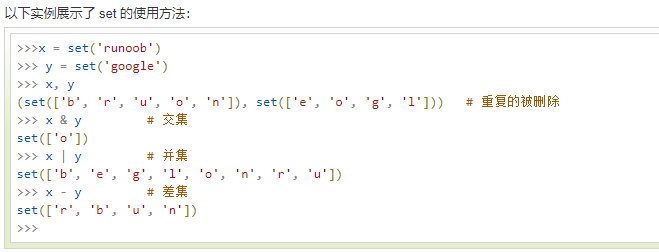
return classList[0]

即第一个类别现的次数等于整个分类标签列表元素的个数，则表明该分类列表下只有一个种类，则不需要继续划分。

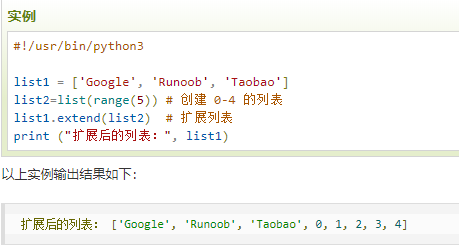
看看下面的这个程序可能比较好理解



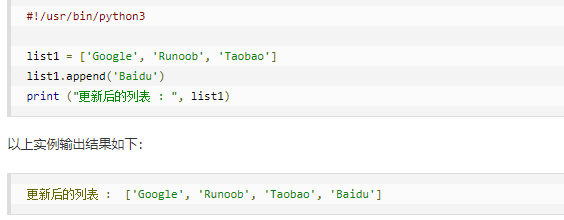
1. Set()函数用法：



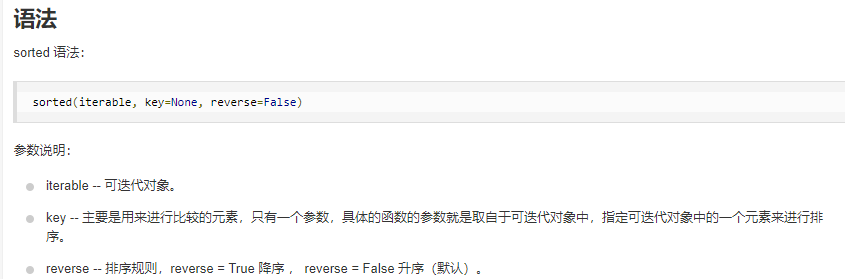
1. extend() 函数用于在列表末尾一次性追加另一个序列中的多个值（用新列表扩展原来的列表）。

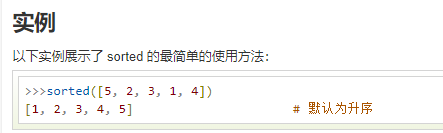


1. append() 方法用于在列表末尾添加新的对象。



1. sorted() 函数对所有可迭代的对象进行排序操作。





items()方法语法：返回可遍历的(键, 值) 元组数组。



operator.itemgetter函数，operator模块提供的itemgetter函数用于获取对象的哪些维的数据

