# Decision Tree

## 实验要求

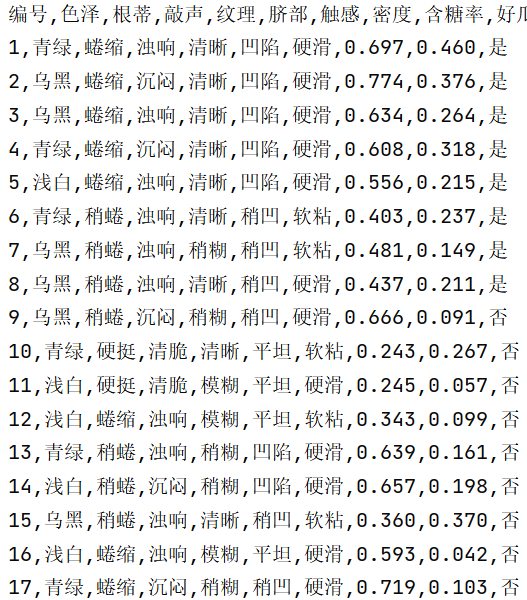
《机器学习（周志华）》习题4.3：**试编程实现基于信息熵进行划分选择的决策树算法，并为表4.3中数据生成一棵决策树**。

## 实验环境

* Windows 10
* Conda 4.10.3
* Python 3.9
* DataSpell 2021.3 EAP
* pandas 1.3.3

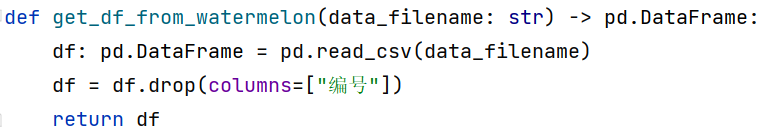
## 实验数据

实验数据源于《机器学习（周志华）》表4.3，存储于data文件夹下的watermelon.csv文件中：



*图：watermelon.csv中的实验数据*

程序使用pandas库的DataFrame数据结构存储watermelon.csv中的数据，并删除（drop）“编号”列：



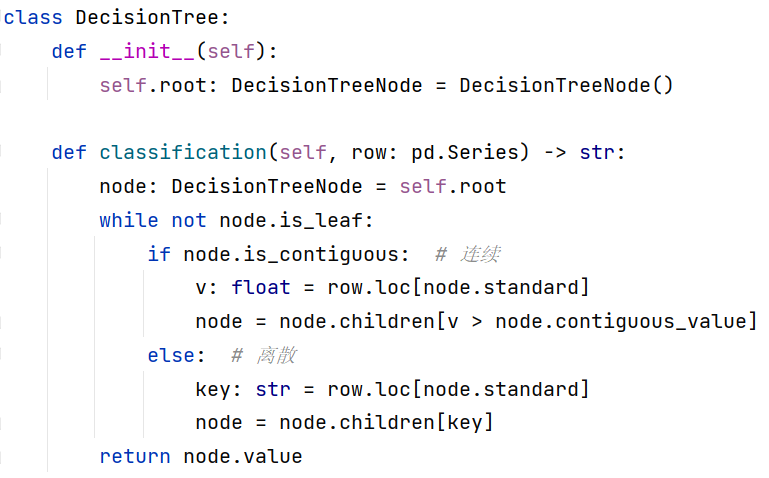
*图：数据预处理部分代码*

## 实验模型

实验共构建四个类（class）完成决策树的构建：DecisionTree、DecisionTreeNode、DecisionTreeTrainer与Accuracy。

### DecisionTree

每个**DecisionTree**实例持有一个**决策树根节点（root）**的引用变量，并提供**classification**方法作为对数据分类的API：

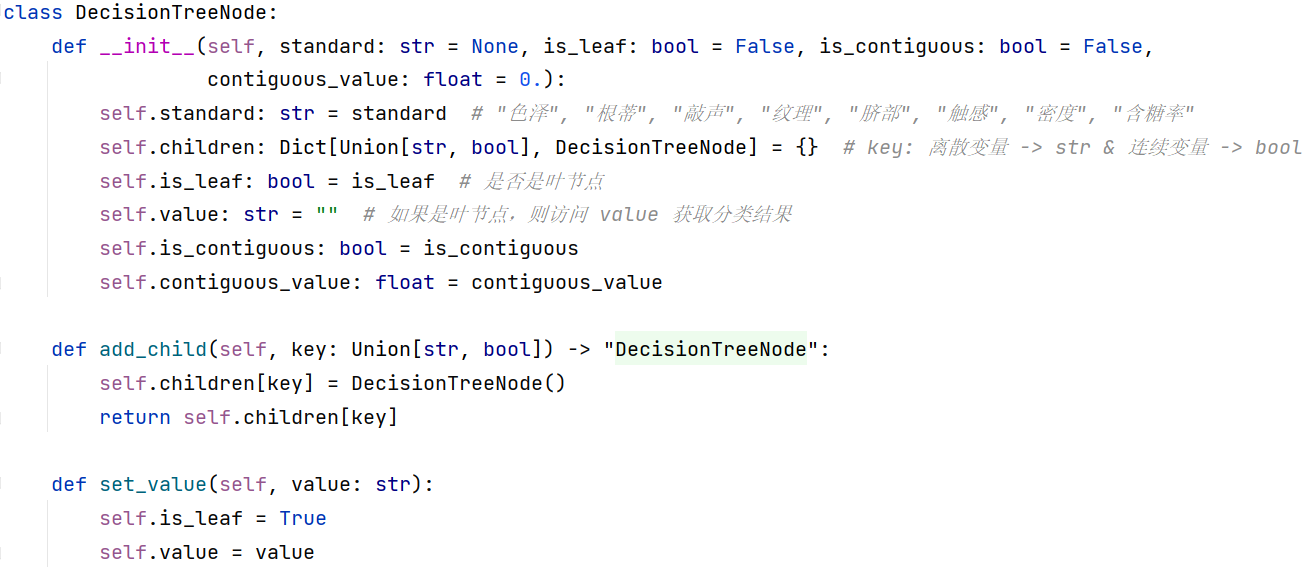


*图：DecisionTree代码*

### DecisionTreeNode

每个**DecisionTreeNode**实例都维护以下属性：

* **决策标签**（standard，例如“色泽”）
* **存储所有子节点的字典**（Dict，例如，决策标签为“色泽”的字典的键的可能取值为“青绿”“乌黑”“浅白”，键为对应的DecisionTreeNode子节点的引用）
* 当前的决策标签是否是**连续值**
* 用于**对连续值二分判断的决策值**（若当前的决策标签是连续值）
* **是否为叶节点**的布尔变量
* 当前节点所代表的**分类结果**（若当前节点是叶节点）



*图：DecisionTreeNode代码*

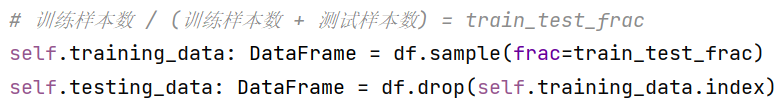
### DecisionTreeTrainer

DecisionTreeTrainer主要有三部分组成：**初始化**、**训练**与**测试**。

#### 初始化

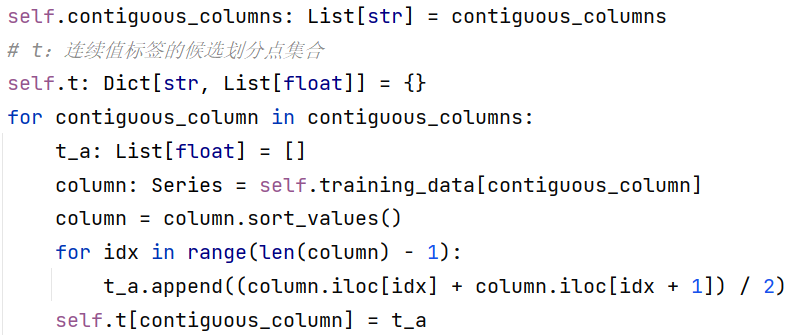
在初始化的部分，DecisionTreeTrainer负责以下工作：

* 根据train\_test\_frac比例，通过**随机采样**来**划分训练数据与测试数据**



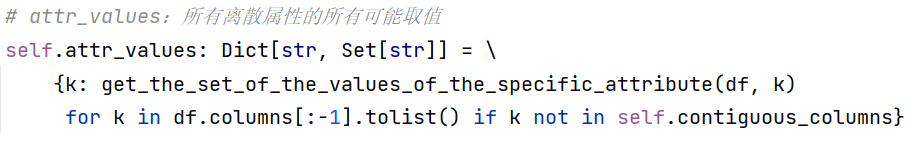
*图：划分数据集部分的代码*

* **处理连续值**：对每个连续属性，将其进行离散化处理，本实验使用二分法（bi-partition）对连续值进行处理（对每种连续属性进行排序，并取中间值，生成候选划分点集合）



*图：处理连续值部分的代码*

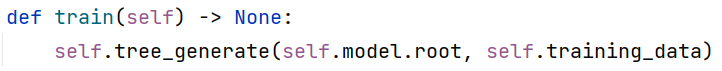
* **获取所有离散属性的所有可能取值**



*图：获取所有离散属性的所有可能取值部分的代码*

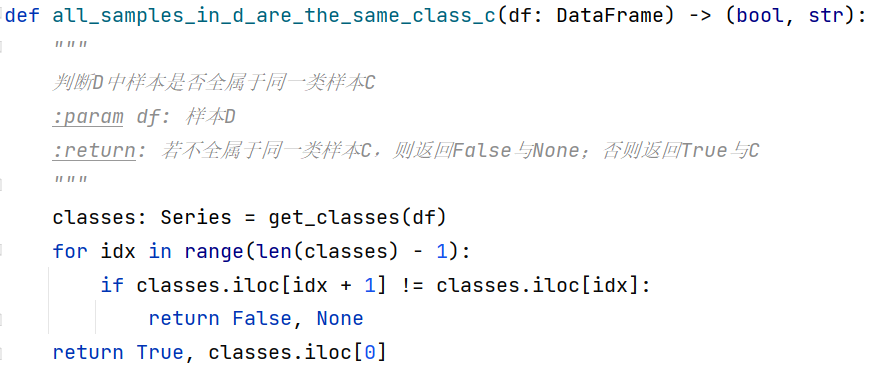
#### 训练

训练过程主要由**递归函数tree\_generate**实现：



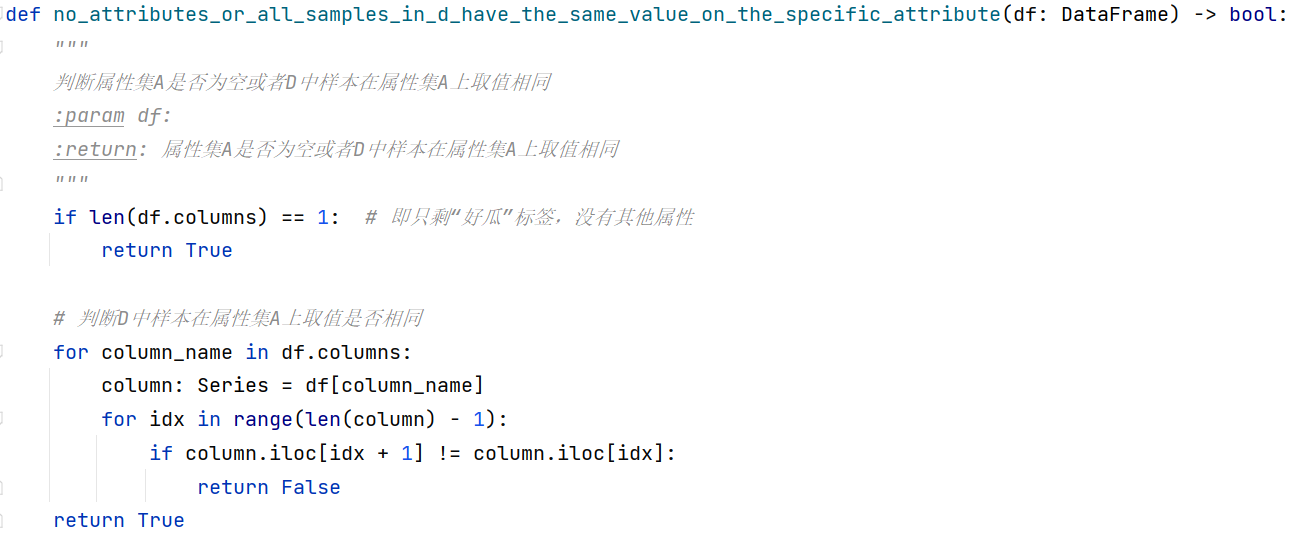
*图：train函数的代码*

在**递归函数tree\_generate**中，首先**判断中样本是否全属于同一类样本**：



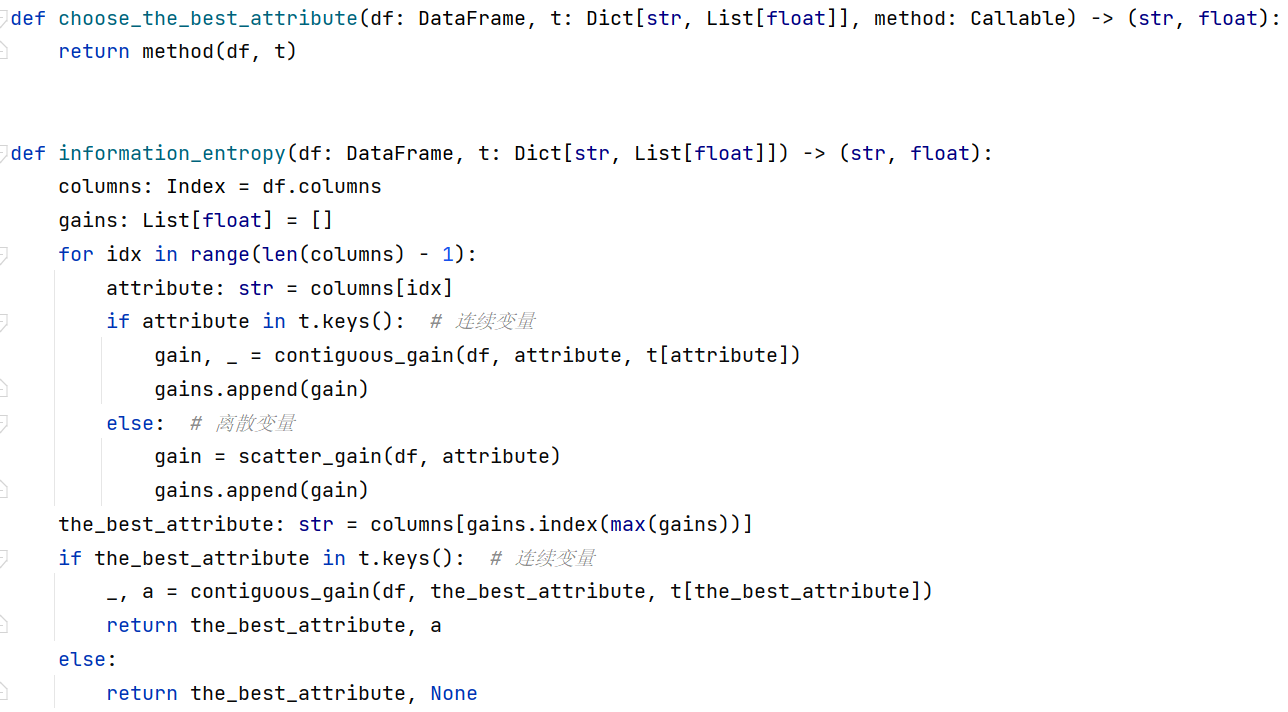
*图：判断D中样本是否全属于同一类样本C*

然后**判断属性集是否为空**或者**样本在属性集上取值是否相同**：



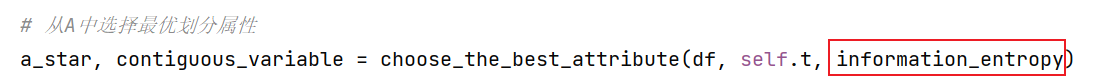
*图：判断属性集A是否为空或者D中样本在属性集A上取值是否相同*

接着，对数据集**选择一个最优划分属性**，本实验使用**信息增益**（information gain）作为度量标准，程序寻找能取到最大的信息增益的属性作为最有划分属性：



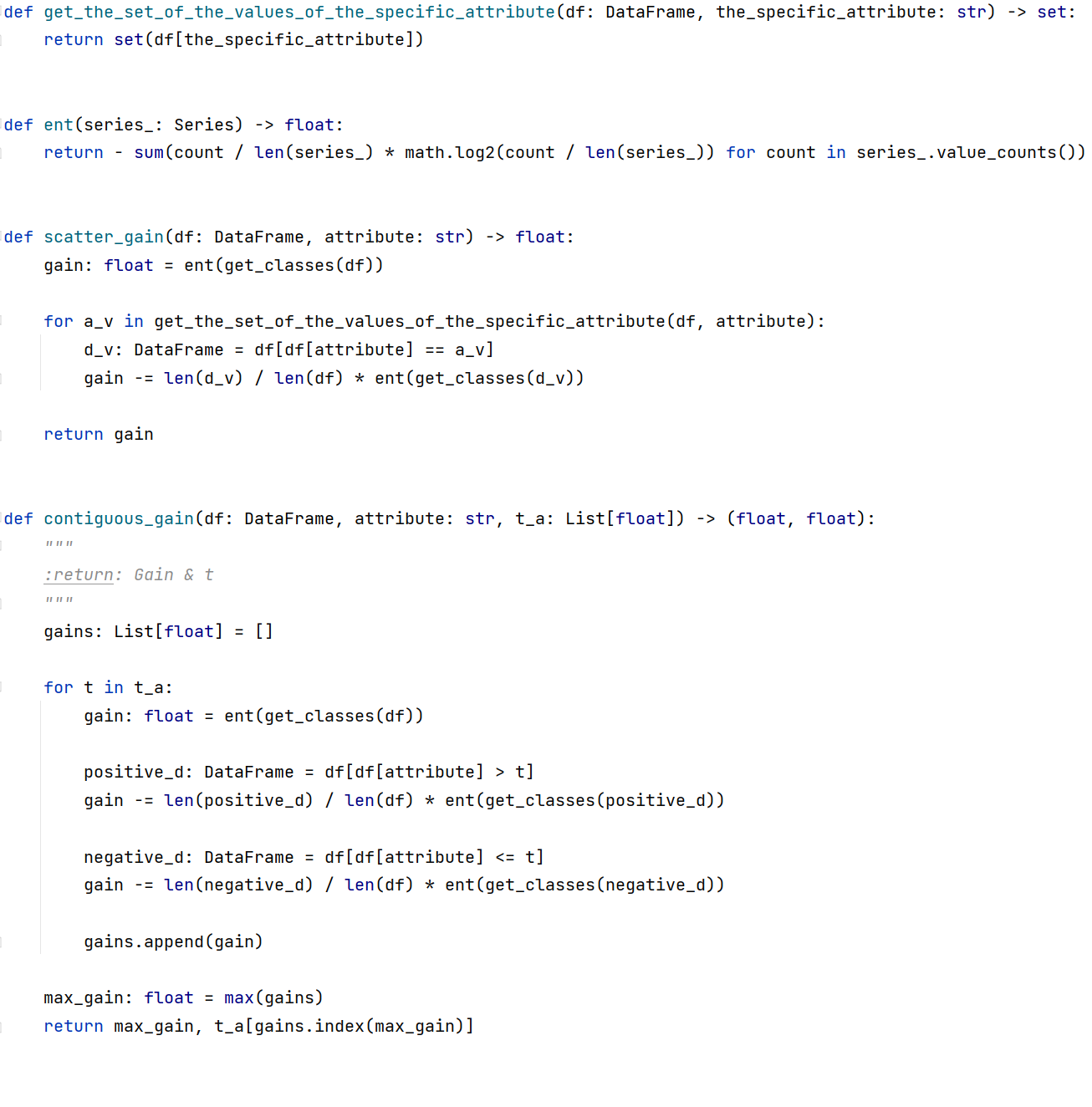
*图：选择最有划分属性*

在程序设计中，应用**设计模式中的策略模式**，将**度量函数choose\_the\_best\_attribute**进行**抽象**，通过**在函数形参中传递一个回调函数**来控制度量策略：



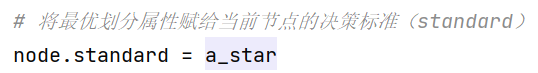
*图：information\_entropy为我们为choose\_the\_best\_attribute传递的用于度量信息增益（information gain）的回调函数*

程序实现了两个函数（scatter\_gain与contiguous\_gain）以分别计算离散属性与连续属性的信息增益：



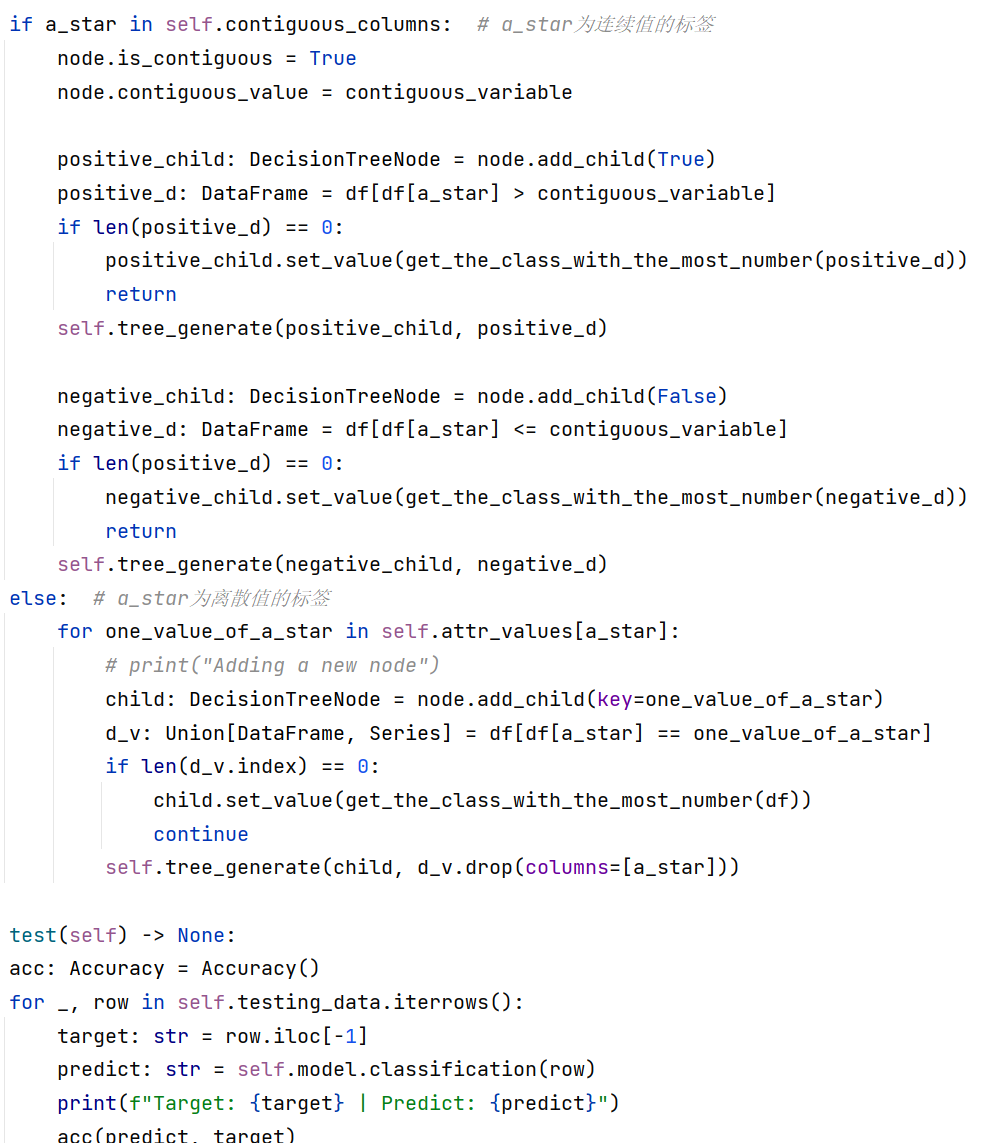
*图：信息增益计算部分的代码*

并将最优划分属性赋给当前节点的决策标准（standard）：



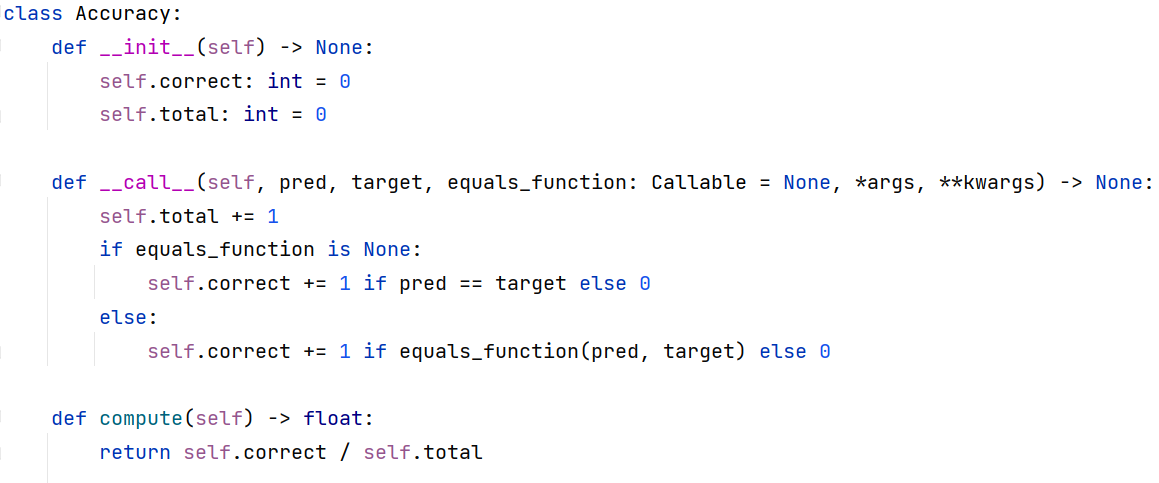
*图：将最优划分属性赋给当前节点的决策标准（standard）*

最后，**为最优划分属性的所有可能取值属性生成分支子节点**，若在最优划分属性上取某值的数据集为空，则将该子节点标记为叶节点，将其分类结果标记为数据集中样本最多的类别，否则对该子节点进行递归调用tree\_generate函数：



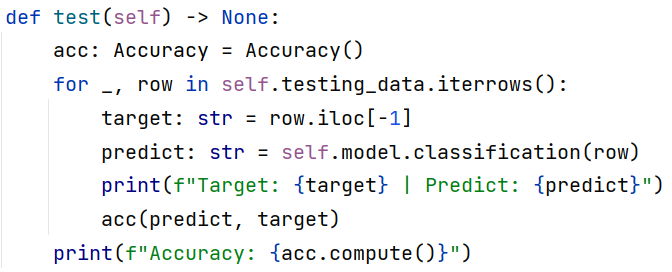
#### 测试

在测试阶段，使用**精度**（Accuracy）作为**度量标准**。实验封装了**Accuracy**类以完成精度的计算，同时Accuracy类**支持传入自定义的判定预测正确的回调方法**，提高程序扩展性：



*图：Accuracy类的代码*

在测试函数test中，创建一个Accuracy实例，对每个数据进行分类，并在最后计算总测试精度：

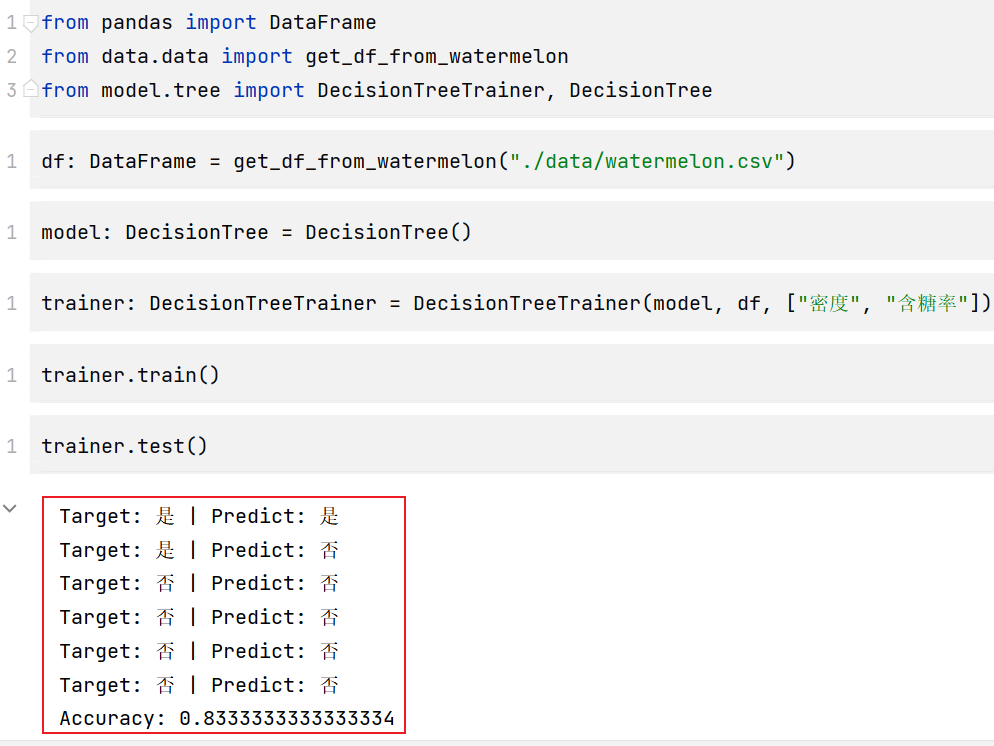


*图：test方法的代码*

## 实验结果

在main.ipynb中，使用Notebook进行代码脚本的运行，并得到实验结果。

下图为一次实验结果的示例图：



*图：实验结果示例图*