**数据分析与处理部分实践作业 2#**

# 线性回归算法实现

## 线性回归介绍

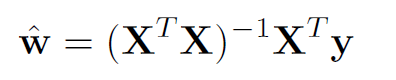
### 1.1.1线性回归模型

给定由d个属性秒速的示例，其中是x在第i属性上的取值，线性模型期望学得一个通过线性组合来进行预测的函数：

简单的说，首先收集数据，然后假设含有未知参数的模型，即含有未知参数的函数，利用这个模型或者函数，可以预测新的数据。

### 1.1.2线性回归模型的解析解

线性回归的求解公式（解析解）：



由推导公式，这里需要把X向量最后一列恒置为1：

|  |  |
| --- | --- |
|  | 1 |
|  | 1 |
|  | 1 |
|  | 1 |
|  | 1 |
| … | … |

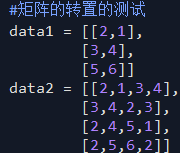
此外，真实的Y值也用向量形式表示

根据解析解可以看出，实现线性回归，就是实现矩阵的转置和求解矩阵的逆，解析解求出以后，线性回归模型就建立起来了，绘制回归直线或者进行预测都可以轻松实现。

## 线性回归实现

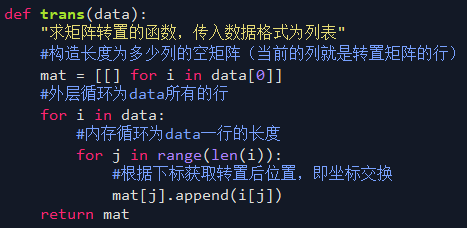
### 1.2.1矩阵的转置实现

1、数据传入：任意阶的矩阵



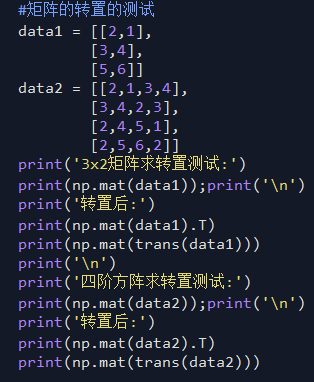
这里定义了两个测试矩阵，一个是3x2的矩阵，一个是4阶方阵做为输入数据。

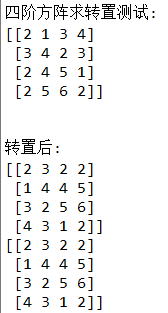
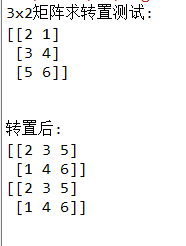
2、构造计算矩阵转置的函数：



3、步骤为：（1）首先建立一个原矩阵长度为多少列的矩阵，因为转置后新矩阵的行数就是原矩阵的列数；（2）循环所有行，内层循环为每一行的长度，新矩阵每个元素的转置结果为坐标交换的原矩阵的值。

4、调用函数，得到转置矩阵的结果，并调用numpy库函数计算转置进行结果对比：

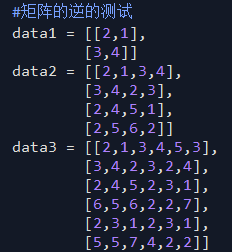




结果一致，实现矩阵的转置。

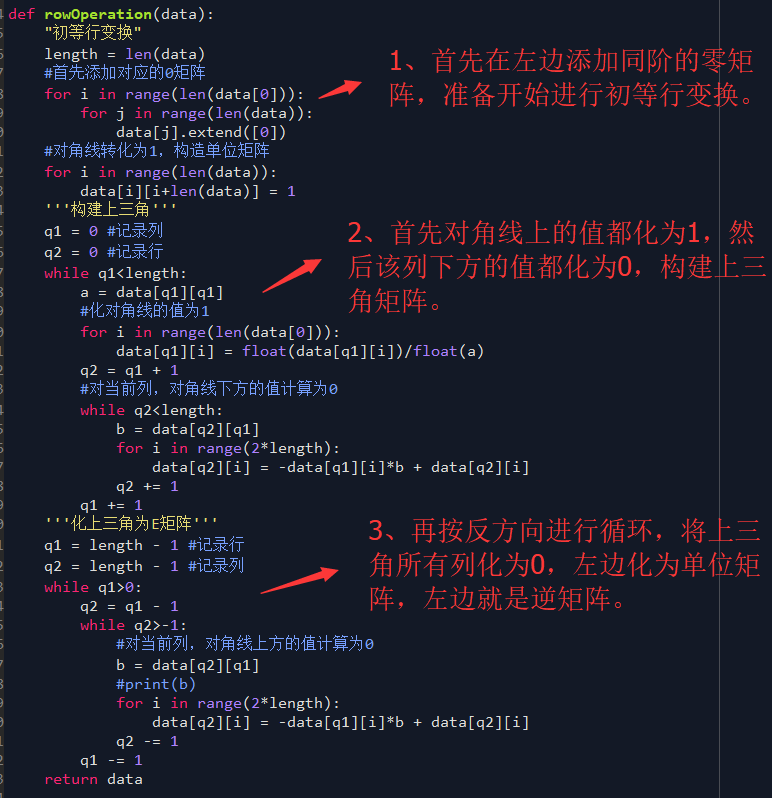
### 1.2.2矩阵的逆实现

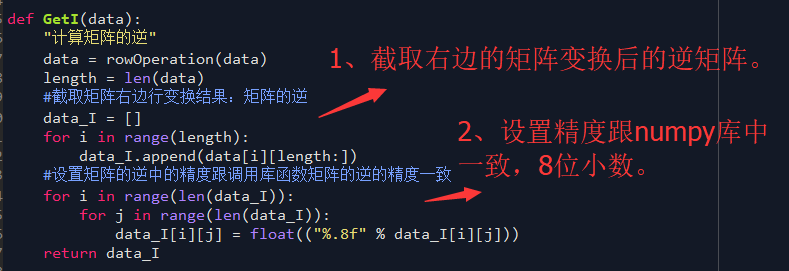
1、数据传入：



分别传入了二阶、四阶、六阶矩阵作为测试，更高阶的也可以计算。

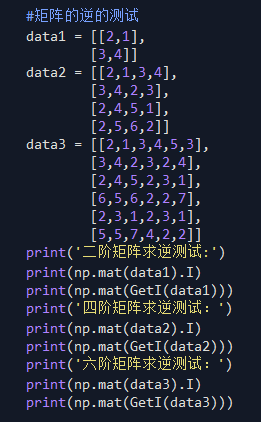
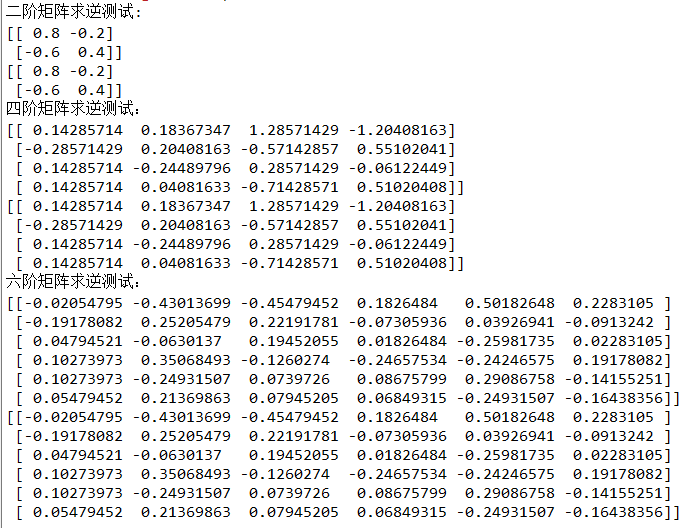
2、构造计算矩阵的逆的函数：





详细的实现步骤如代码截图所示。

3、调用函数，获取求逆结果，并与numpy库中的求逆函数结果进行对比：

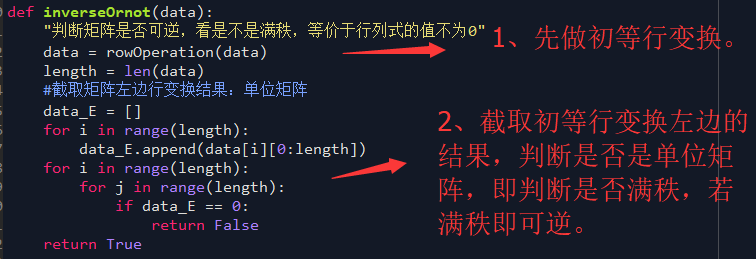
 

可以发现结果完全一致，说明求逆函数实现成功。

### 1.2.3判断矩阵是否可逆

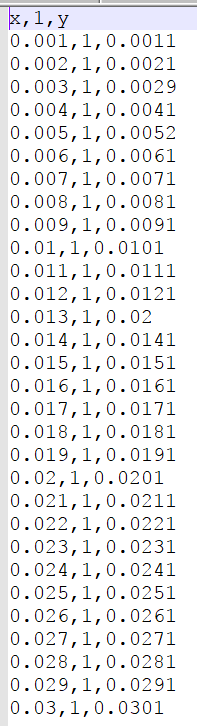
1、一个矩阵是否可逆，等价于它的秩不为0，也等价于矩阵的秩与阶数相同，即一定是满秩矩阵。

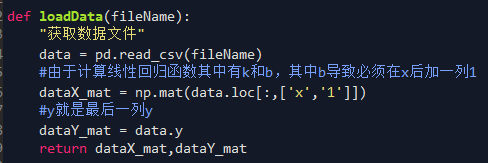
2、构建判断是否可以函数：



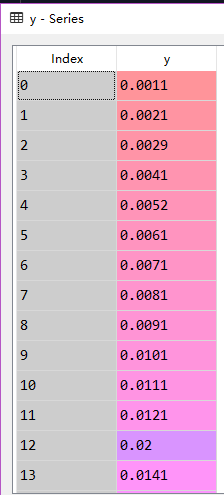
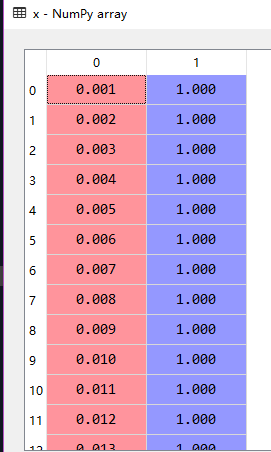
### 1.2.3求解解析解和绘制线性回归图形

1、传入数据集，csv格式：

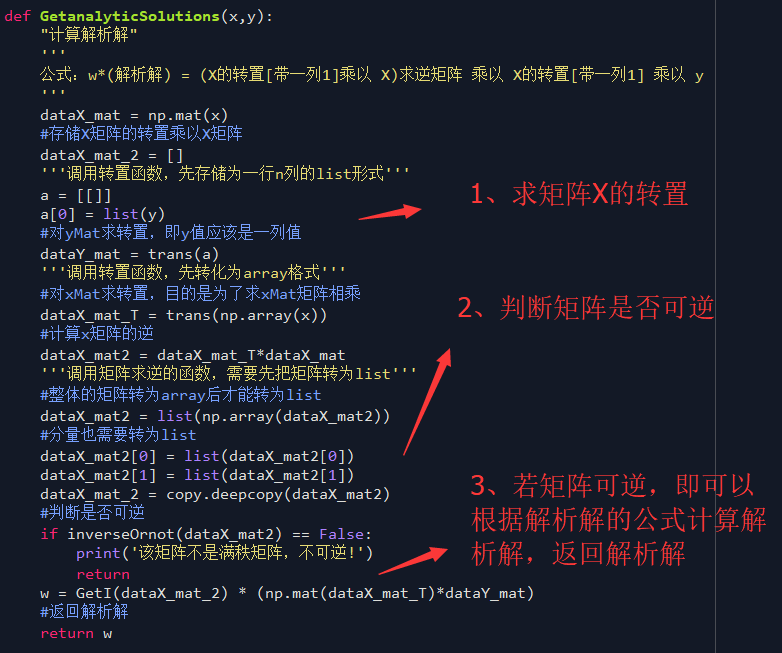




选取前两列为X向量，最后一列为Y向量，读取后的结果：

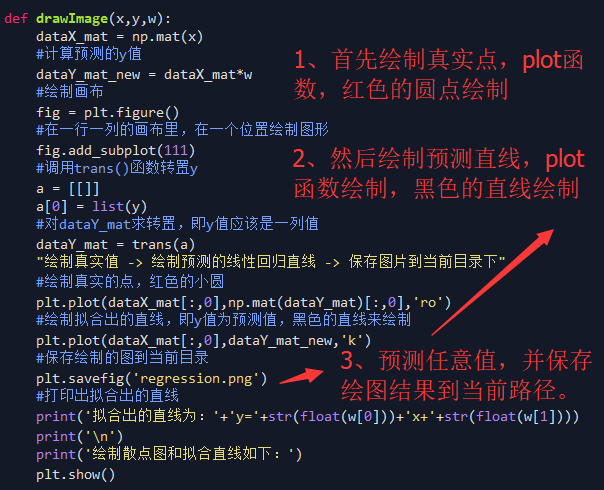


2、计算解析解：

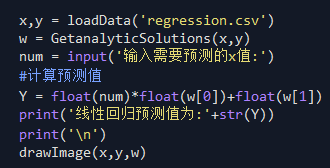


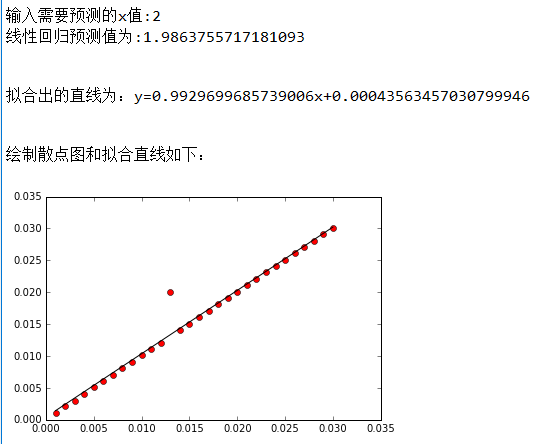
按照解析解的公式，首先计算X矩阵的转置，然后计算X的转置乘以X矩阵的逆，最后根据公式乘法做对应值的乘积即得到解析解w，其中w[0]为k，w[1]为b，线性回归模型基本求解完成。

3、绘制线性回归直线：



4、调用函数，获取结果：





红色点为真实的离散点，黑色直线为预测出的回归直线。

# J48算法实现与结果评估程序实现

## J48算法实现

### 2.1.1 J48分类介绍

J48是C4.5决策树学习算法，决策树是一类常用的机器学习方法，基于树的结构来进行决策，是面临决策问题是自然的一种处理机制。决策树的学习目的是为了产生一棵泛化能力强，即处理未见示例能力强的决策树，其生成是一个递归的过程。

决策树最重要的划分选择，J48是利用增益率来进行选择最优属性划分的：

**“信息熵”**是度量样本集合纯度最常用的指标，假定样本集合D中第k类样本所占比例为(k=1,2,…,|y|)，则D的信息熵定义为：

的值越小，则D的纯度越高。根据信息熵公式计算出某个属性a的样本的信息熵，再考虑不同分支结点所包含样本数不同，给其赋予权重，样本数越多的分支结点的影响越大，于是计算出“信息增益”：

信息增益准则对可取数目较多的属性有所偏好，为减少这种偏好带来的不利影响，C4.5使用“增益率”来选择最优划分属性，而不直接使用信息增益。

**“增益率”**的定义为：

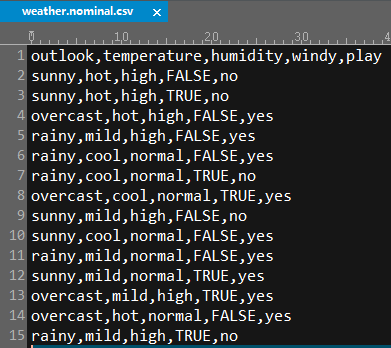
,其中

因此，实现J48算法首先需要计算信息熵，然后通过计算属性的增益率来划分最优属性，也需要写划分数据集函数来根据最优属性来进行数据集的划分。

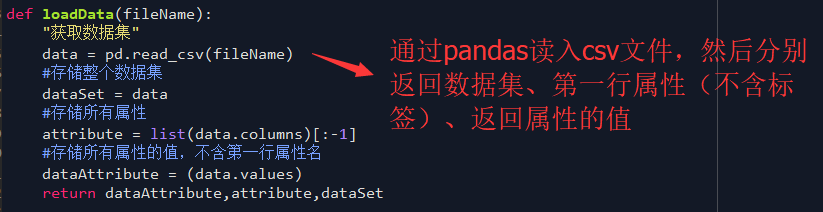
### 2.1.2 J48算法实现过程

#### 2.1.2.1 读入数据

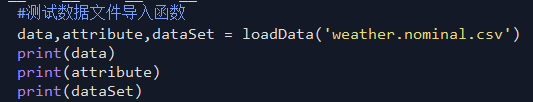
1、选择weka安装目录下data文件夹中的weather.nominal.arff数据集，转为weather.nominal.csv格式待使用：

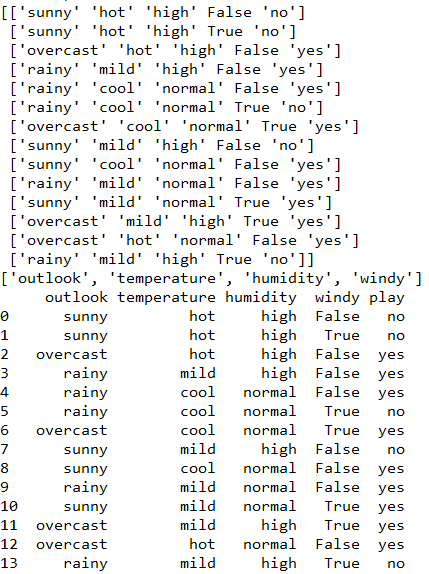


2、 函数读入.csv格式的文件：



3、调用函数返回结果：

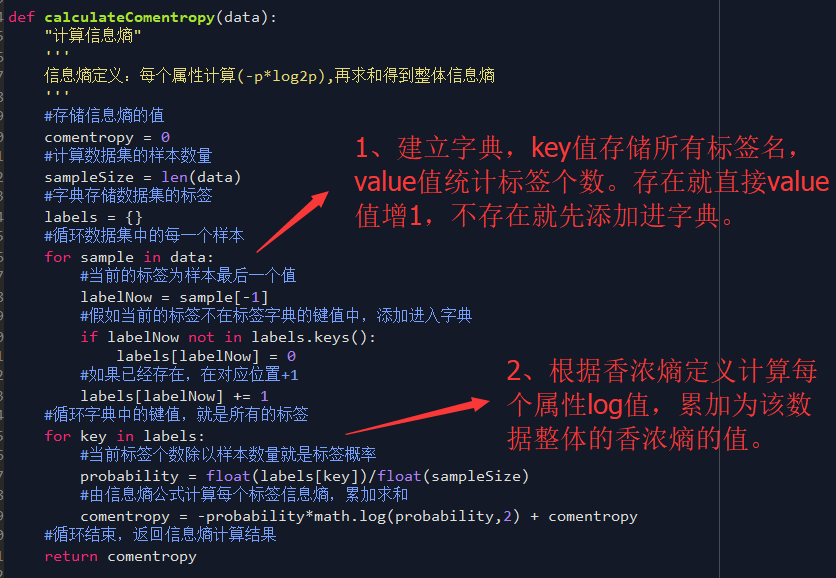




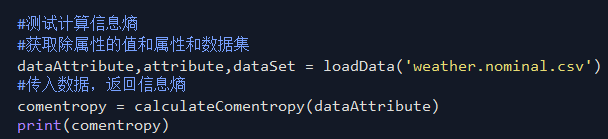
#### 2.1.2.2 计算香浓熵

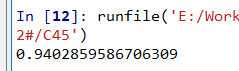
1、数据集传入weather数据集。

2、构造计算香浓熵函数：



3、调用函数，返回香浓熵计算结果：

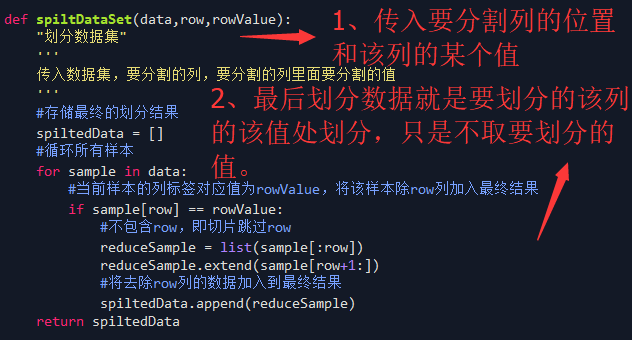




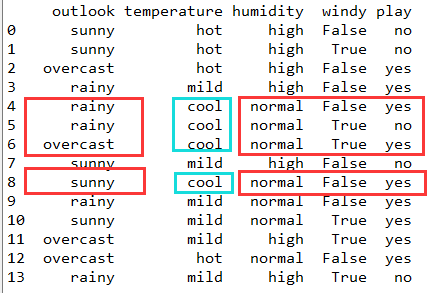
#### 2.1.2.3 划分数据集

1、在后面属性信息增益率计算需要先写出划分数据集的函数，传入weather数据集。

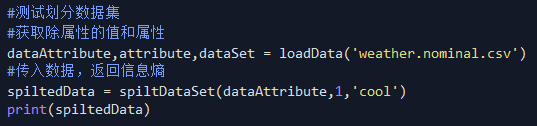
2、构造函数：



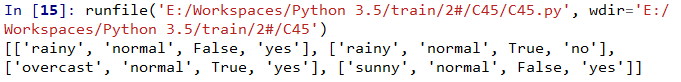
3、调用函数，例如传入数据集，按数据集的**第1列的cool值**划分，**预期划分结果就是红色框定的部分**：



调用函数验证结果：



返回划分结果：



和我预期的结果一致，说明划分数据集函数构建正确

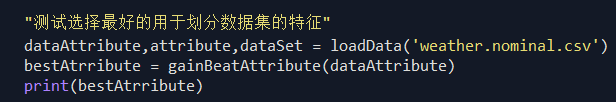
#### 2.1.2.4 计算信息增益率

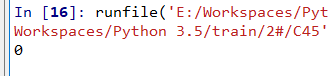
1、传入weather数据集

2、构建计算信息增益率的函数：

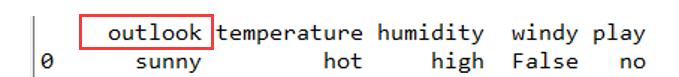


3、调用函数，返回最优划分属性列的列标记：





返回为0，也就是说当前数据集最优划分属性的列标记为0，即应该是outlook这个属性为最优划分属性：



#### 2.1.2.5 建立树模型

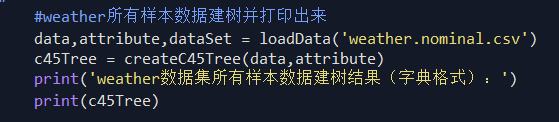
1、传入weather数据集

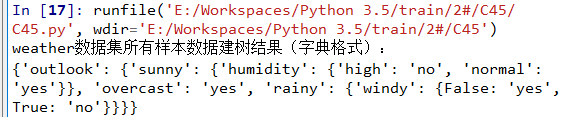
2、构建建树函数：



相当于对当前最优属性列中，按每个值进行遍历，建树的字典分别等于当前属性值递归调用建树的子树。

3、调用函数，并将建立的树模型按字典的形式打印出来：





## 结果评估程序实现

### 算法性能评估介绍

在周志华的机器学习算法2.2评估方法中有介绍，**本文使用了留出法**，实现K折代码很类似，只是不需要随机采样这部分代码过程。

#### 2.2.1.1 留出法

需要保证数据正负分布的一致性，随机分层抽样测试集testdata、训练集traindata，二者为互斥集合。通过取2/3到4/5为训练集，剩余为测试集进行评估。

单次留出法不稳定，需要多次随机划分和重复评估取均值为评估结果，如本文随机分层采样了100次来评估模型

#### 2.2.1.2 K折交叉验证

分层划分为K份，每次用1个作为测试集，剩余K-1个作为训练集，返回K个评估结果的均值。

### 留出法性能评估实现

#### 2.2.2.1 整体思路

1、取得数据的分类标签名，按分类名分别截取正类样本、负类样本；

2、对正（负）样本的数据集分别增加一列rand，里面产生0到length的随机数，并保证产生的随机数每个值都不同，如（0,5）🡪3,2,1,0,4

3、对此时的数据样本按rand列排序，如0,1,2,3,4，重新建立数据集的索引；

4、假如分层抽样比例为1/5，那么测试集每次就选取[0:（1/5）\*length]，以5为例，就是第一个样本为测试集，其余[(1/5）\*length: ]为训练集，这样就保证了每次选择的训练集和测试集是随机的。对于正负样本集分别进行上述操作，连接起来构建好训练集和测试集。

5、对于构建的测试集，截取它真实的标签，等待评估模型训练结果。

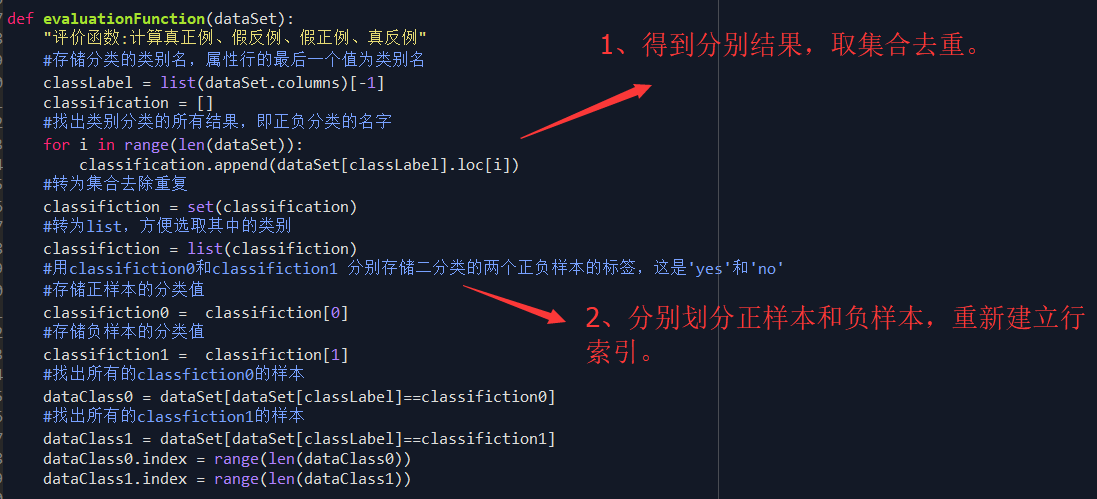
6、将训练集放进决策树模型中去建立模型，对于测试集中每个样本依次按分类模型去跑出预测标签。

7、取出预测真实标签和预测标签，分别按真正例、假反例、假正例、真反例的计算公式来算出真正例、假反例、假正例、真反例，并返回。

8、调用计算真正例、假反例、假正例、真反例的函数100次，最后模型评估值为真正例、假反例、假正例、真反例的平均值，然后计算F1值和ACC。

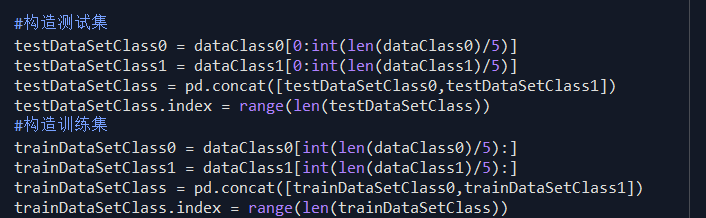
#### 2.2.2.2 实现过程

1、划分正负样本：

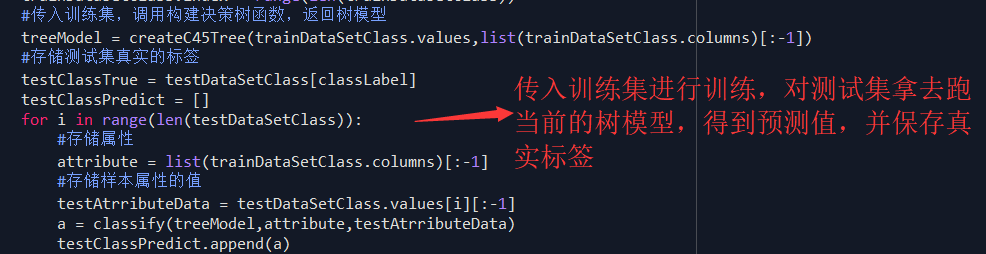


2、随机采样正负样本：





3、测试集跑当前的树模型：



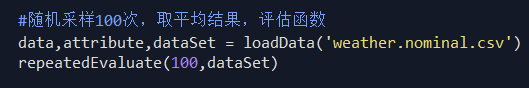
4、根据真实标签和预测分类计算混淆矩阵，并返回：



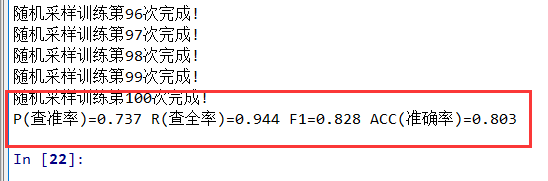
5、多次调用评估函数，取均值：



6、调用函数，输出评估结果：



经过100次随机抽样训练模型，得到最终的评估结果：



# 心得体会

本次作业实现了线性回归算法和J48算法及评估程序，首先对两个算法有了更深刻的理解。在实现线性回归的时候，解析解的公式很简单，但是函数不能直接调库，实现矩阵的逆花费了很多时间和精力，采用的是矩阵行变换的思路，效率很高；在写J48的时候，在评估函数使用了留出法，因为要构造随机函数，分层抽样正负样本，比k折交叉验证实现要困难，后期调bug搞了很久。