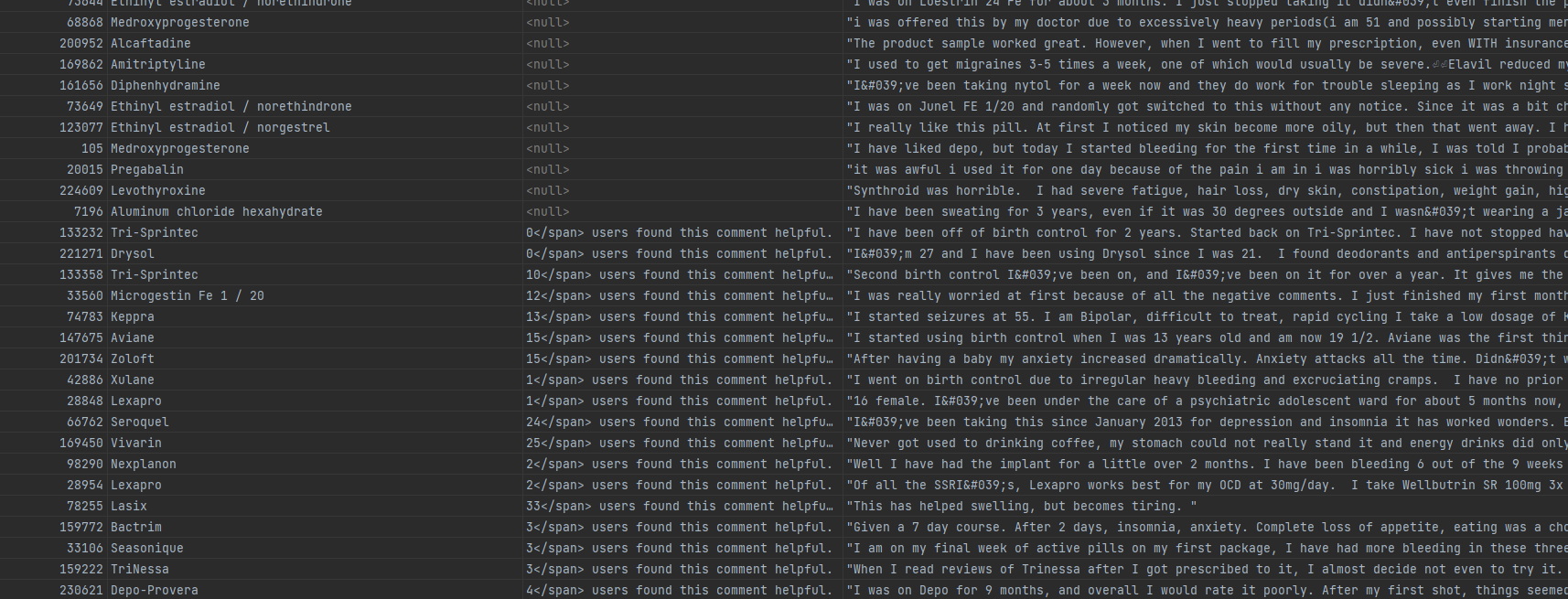
**决策树算法实现实验报告**

201250070 郁博文

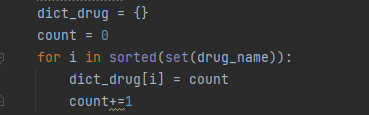
1. 数据预处理

首先将数据里，含有垃圾值的记录删除。

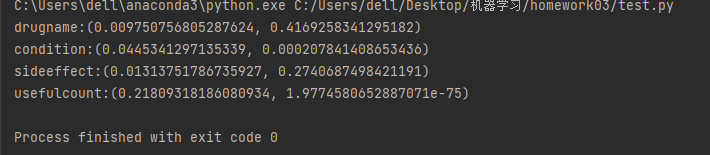
例如如下情况：

删除明显与rating没有关系的列recordID,reviewComment和date

之后，为了使数据更好处理，对drugName, condition, sideEffects进行编码，具体编码策略如下：对于每个不同的值，用数字代替

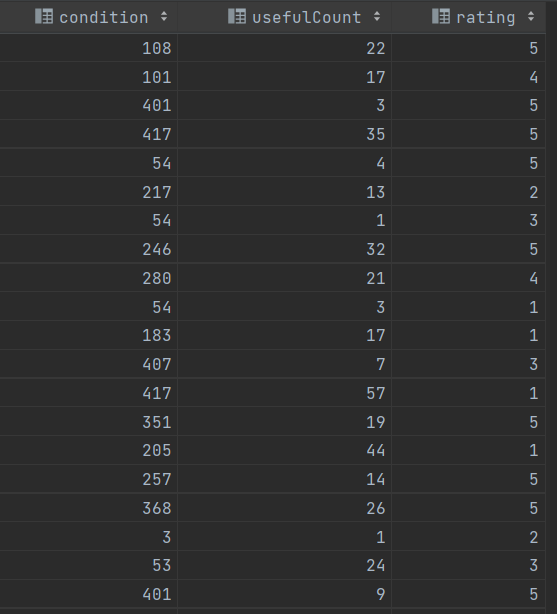


之后为上述三个属性和usefulCount分别计算训练集中它们与rating的皮尔逊相关系数，结果如下：



发现四者中只有condition和usefulCount的p值小于0.05，故舍弃其他两个

最终数据集如下：



1. 决策树算法

1、信息熵

一个节点信息熵，是每个属性出现的概率。

每种特征中每个属性的信息熵

每个特征的信息熵

信息增益，a是属性

算法过程

1. 将所有的特征看成一个一个的节点。创建根节点。

2. 遍历所有特征。遍历到其中某一个特征时，遍历当前特征的所有分割方式，找到最好的分割点，将数据划分为不同的子节点，计算划分后子节点的信息熵。

3. 在遍历的所有特征中，比较寻找最优的特征以及最优特征的最优划分方式。选择信息增益最高的特征，根据特征则对当前数据集进行分割操作，产生子树。

4. 对新的子节点继续执行2 - 3步，直到下面的停止条件退出循环。

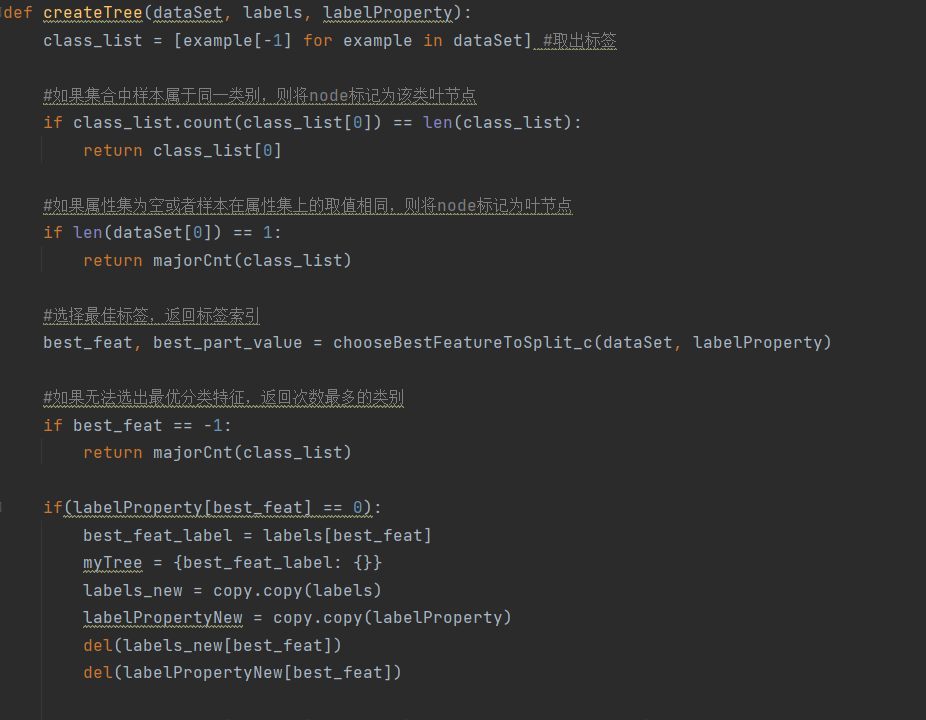
停止条件:

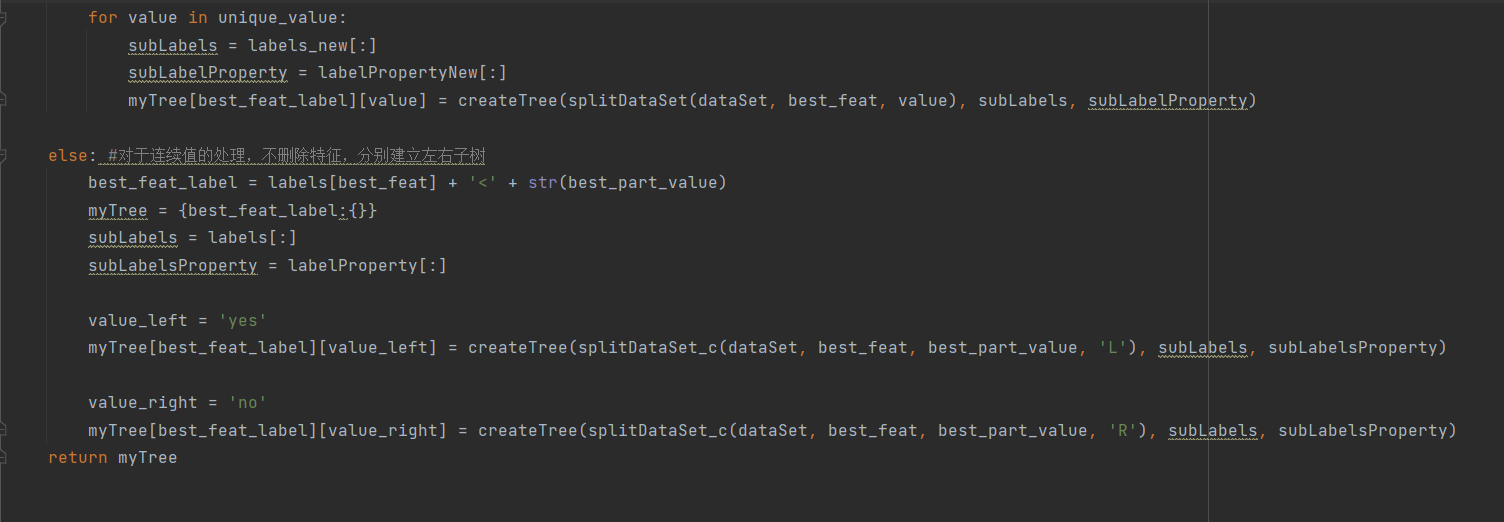
1. 当子节点中只有一种类型或为空的时候停止构建(会导致过拟合)

2. 当前节点种样本数小于某个值，同时迭代次数达到指定值，停止构建，此时使用该节点中出现最多的类别样本数据作为对应值(比较常用)

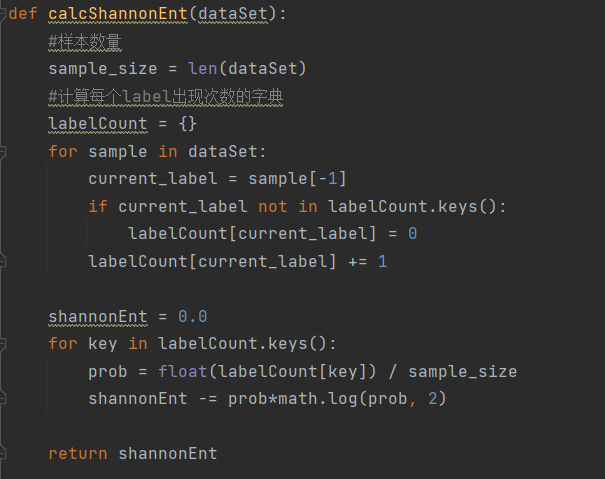
核心代码：

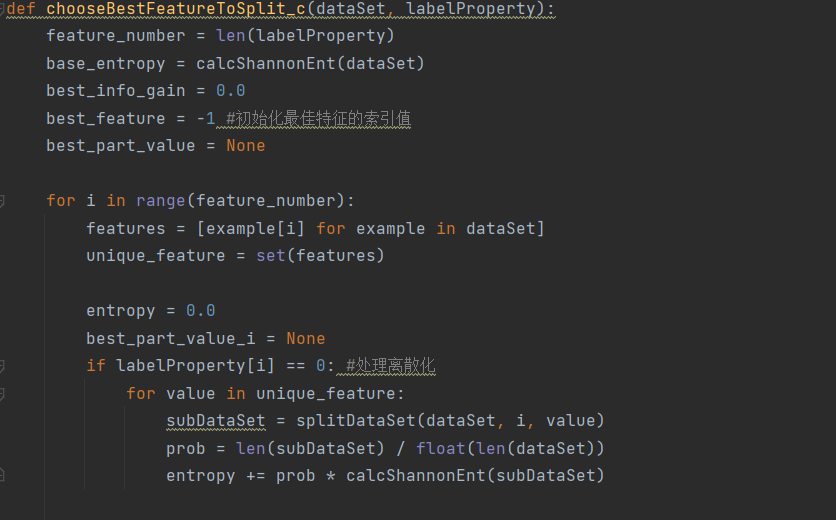
1、建树

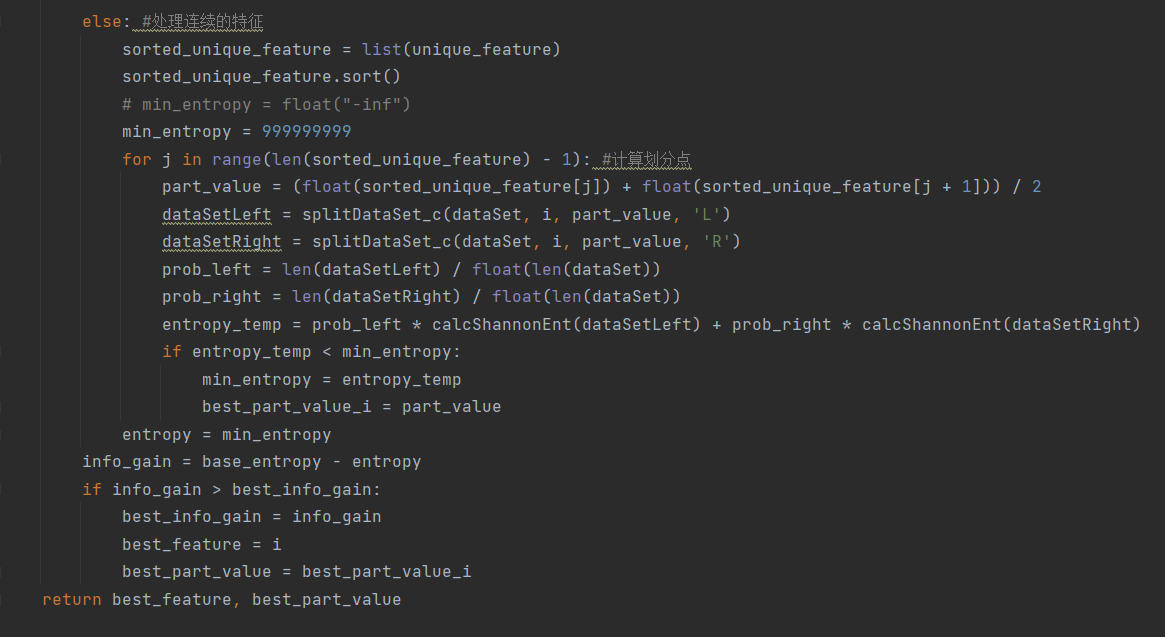




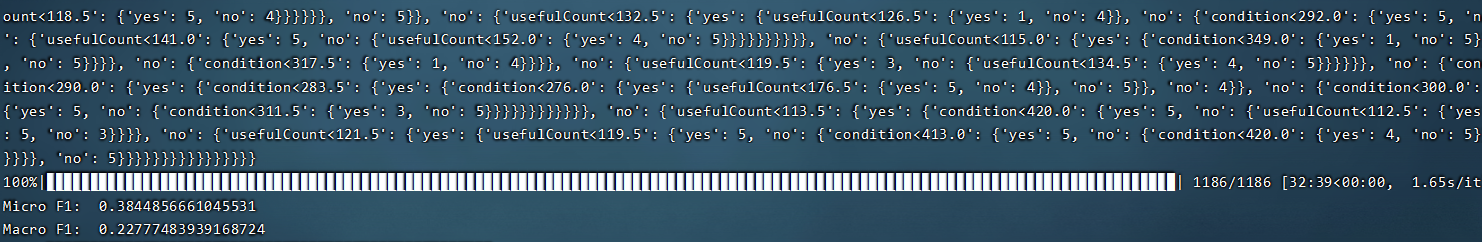
2、计算信息熵并寻找最佳信息增益







1. 实验结果



Micro F1: 0.3844856661045531

Macro F1: 0.22777483939168724

测试集预测结果部分展示：

