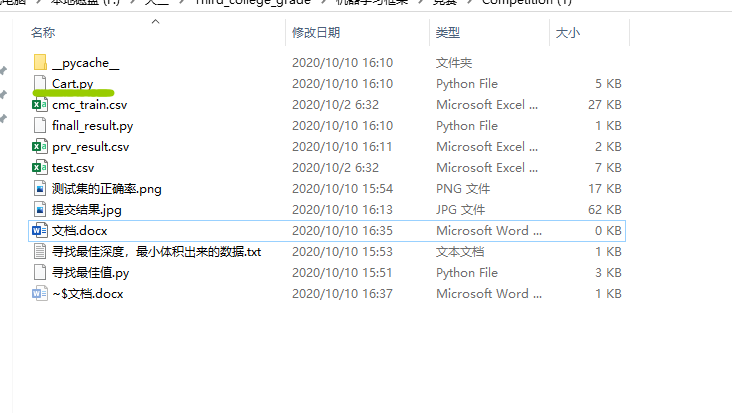
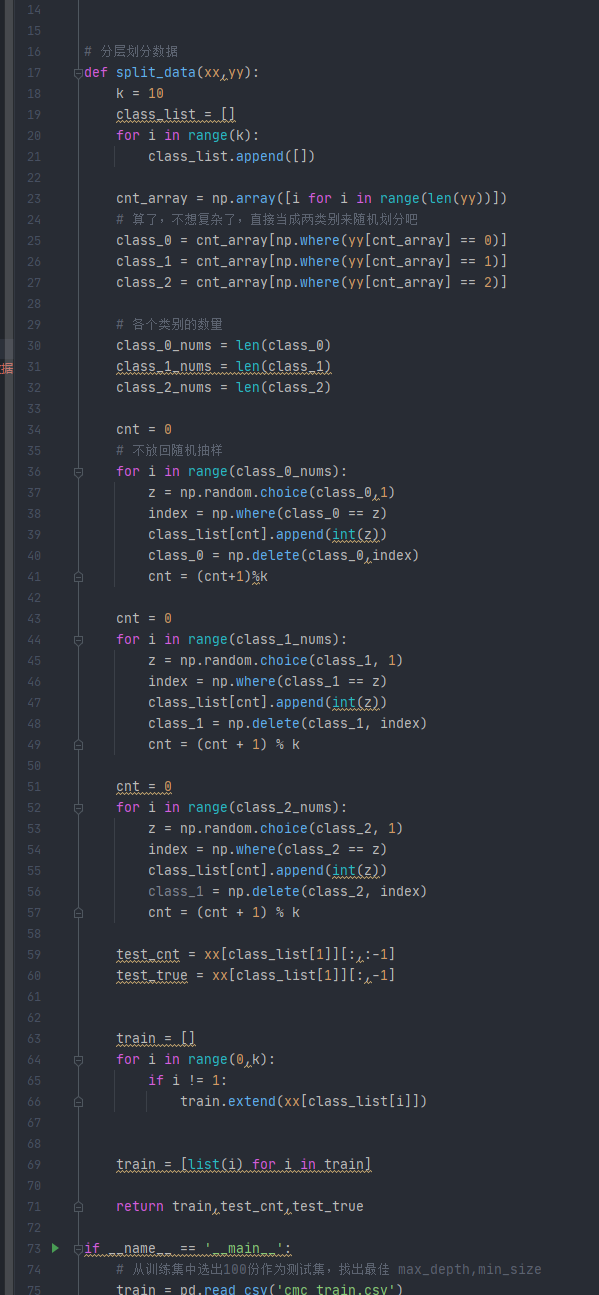
这次比赛,我采用的是Cart 决策树模型 ：

构建决策树模型（代码在Cart.py文件中）



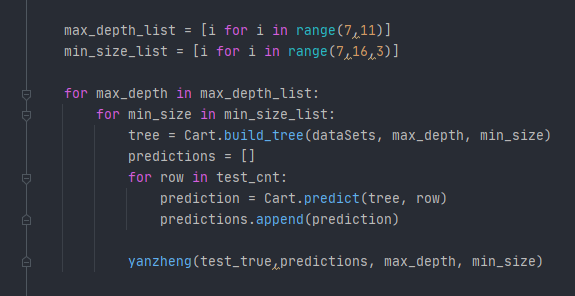
对训练数据进行分层随机划分 取90% 为训练集 10%为测试集 （寻找最佳值.py）

划分函数 :

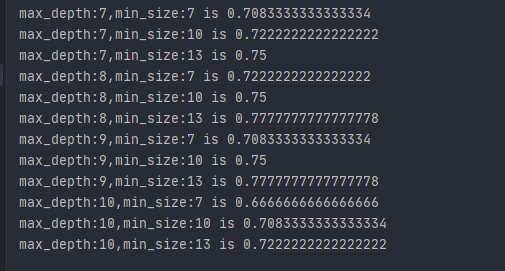


通过循环找出最佳

max\_depth, min\_size



验证结果：



因为该数据集有9个特征，所以最大深度不能超过9

有因为根据上述结果 分别测试

Max\_depth min\_size

7 13

8 10

8 11

8 12

9 10

9 11

9 13

根据每个特征的信息增益

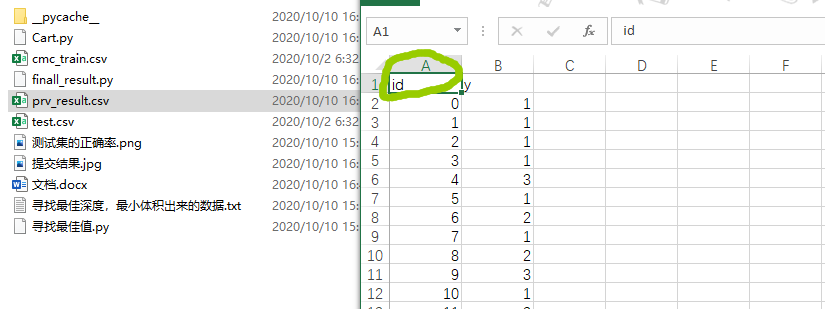
[0.09827542219198349,  
 0.060622044887663984,   
 0.04072441733137455,   
 0.11356575441032546,   
 0.009094221095042654,   
 0.003067720381586092,   
 0.028752402267523847,   
 0.04386051129049262,   
 0.019985660632640956]

来判断 ，应该有8个特征是比较重要的，所有，最大深度设置为8

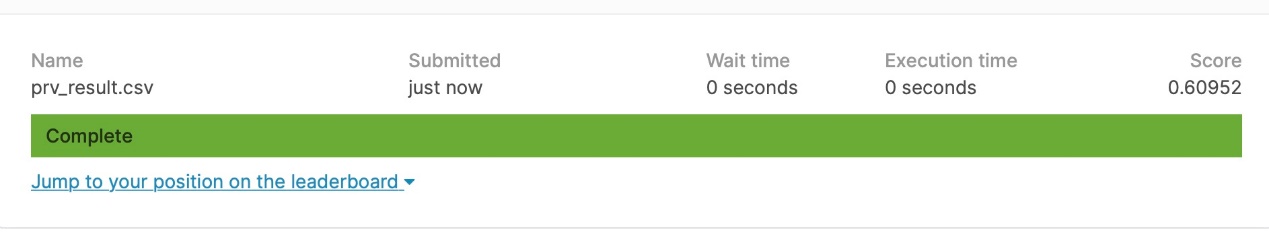
根据所有的训练数据进行构建模型，来预测数据：



最后将生成的csv文件的第一列加上 “id”



最后提交csv文件

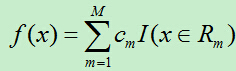


Cart 模型流程：

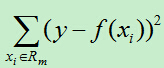
**回归树：使用平方误差最小准则**

训练集为：D={(x1,y1), (x2,y2), …, (xn,yn)}。

输出Y为连续变量，将输入划分为M个区域，分别为R1,R2,…,RM,每个区域的输出值分别为：c1,c2,…,cm则回归树模型可表示为：



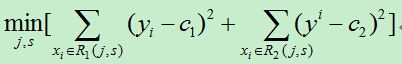
则平方误差为：



假如使用特征j的取值s来将输入空间划分为两个区域，分别为：



我们需要最小化损失函数，即：



其中c1,c2分别为R1,R2区间内的输出平均值。（此处与统计学习课本上的公式有所不同，在课本中里面的c1,c2都需要取最小值，但是，在确定的区间中，当c1,c2取区间输出值的平均值时其平方会达到最小，为简单起见，故而在此直接使用区间的输出均值。）

为了使平方误差最小，我们需要依次对每个特征的每个取值进行遍历，计算出当前每一个可能的切分点的误差，最后选择切分误差最小的点将输入空间切分为两个部分，然后递归上述步骤，直到切分结束。此方法切分的树称为最小二乘回归树。

**最小二乘回归树生成算法：**

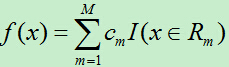
1）依次遍历每个特征j，以及该特征的每个取值s，计算每个切分点（j,s）的损失函数，选择损失函数最小的切分点。



2）使用上步得到的切分点将当前的输入空间划分为两个部分

3）然后将被划分后的两个部分再次计算切分点，依次类推，直到不能继续划分。

4）最后将输入空间划分为M个区域R1,R2,…,RM,生成的决策树为：



其中cm为所在区域的输出值的平均。