机器学习之工资预测

组号：第六组

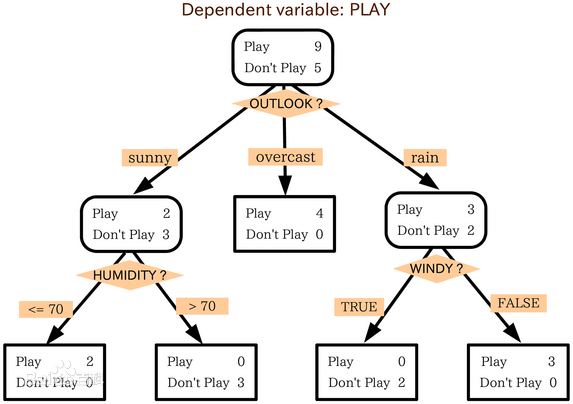
成员：笪宸、李治琦、郑启林、赵昱淞

1. 摘要

本项目深入研究了一些机器学习的相关算法，比如：ID3算法、随机森林（决策树）等，并利用这些算法和公民的一些自身信息，构建了一个随机森林模型，实现了对公民工资进行预测的功能，模型的预测准确率超过80%。本项目为了实现更好的可视化效果，还制作了GUI界面、绘制了随机森林模型的ROC曲线。

1. 算法简介

项目采用算法为随机森林。它是一个由众多决策树组成的分类器，输出的结果由决策树的大多数决定。决策树是一种树形结构，其中每个内部节点表示一个属性上的测试，每个分支代表一个测试输出，每个叶节点代表一种类别。如下图：



随机森林中决策树的生成需要遵循以下原则：

1. 若训练集大小为N，对每棵树而言，随机有放回地从训练集中抽取N个样本作为该树地训练集；
2. 若每个样本的特征维度为M，指定一个常数满足m<<M，随机地从M个特征中选取m个特征子集，每次树进行分裂时，从这m个特征中选取最优的；
3. 每棵树都尽最大程度生长且没有剪枝过程即遵循随机选取数据与随机选取特征两个原则。

采用ID3算法构建决策树。ID3算法是一种贪心算法，用来构造决策树。ID3算法起源于概念学习系统，以信息熵的下降速度为选取测试属性的标准，即在每个节点选取还尚未被用来划分的具有最高信息增益的属性作为划分标准，然后继续这个过程，直到生成的决策树能完美分类训练样例。

1. 材料与方法

我们的算法实现利用的是VMware下的pycharm进行集成开发。在vm安装pycharm首先需要一个linux环境下的pycharm压缩包，解压缩；在解压完成的目录测试pycharm是否可用（输入.pycharm.sh），若不可用则需要提高权限；然后配置pycharm的环境变量（输入vim ~/.bashrc 在其后增加两行export PyCharm=/home/用户名/software/pycharm-2018.1.3

export PATH=$PyCharm/bin:$PATH）；再刷新配置文件（source ~/.bashrc）；再直接输入pycharm.sh运行pycharm。

所用到的算法：ID3算法、随机森林（决策树）

项目大致分为三个部分，第一步用ID3算法计算信息增益，从多个标签中筛选重要特征作为一组数据的特征向量；第二步根据基尼系数，用重要特征构建多个决策树并组成随机森林；第三步把数据集分为训练集和测试集投入到不同的决策树，训练构建的模型并对美国公民的收入做预测并检验准确率。

项目大致分为两个部分，第一部分是从数据集中选取重要的标签，即找出最优的分类特征。本次项目使用的数据一共包含十五类信息，我们挑选十二个便于用数字衡量的。分别是：

1.年龄:数值

2.工作类别:Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov,

State-gov, （Without-pay, Never-worked. ）=？     ==>[1,2,...6,0]

3.教育程度:

分组（pre+1-12）（HS）（somecollege，Assoc-acdm, Assoc-voc)（Bachelors, Masters) (Prof-school，Doctorate）  ==>[0,1,2,3,4]

4.受教育时间:数值.

5.婚姻状况:

(婚姻质量)（Married-civ-spouse，Widowed）（Divorced，Separated，Married-spouse-absent）(other)  ==>[0,1,2]

6.职业:

（脑力劳动:Tech-support,Sales, Exec-managerial, Prof-specialty,Machine-op-inspct,Adm-clerical,Craft-repair)

（体力劳动:Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces, Handlers-cleaners）

（other-service）

==>[0,1,2]

7.家庭关系:

Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.   ==>[0,1,2,3,4,5]

8.种族: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.  ==>[0,1,2,3,4]

9.性别: Female, Male.  ==>[0,1]

10.资本收益: 连续数值.

11.资本亏损: 连续数值.

12.每周工作小时数: 连续数值. （和工资的关系）

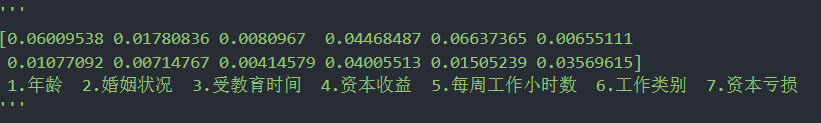
具体情况见excel表格feature\_extracted

1. 重要特征的筛选

衡量一个集合纯合的指标是熵

信息增益在决策树算法中是用来选择特征的指标，信息增益越大，则这个特征的选择性越好。ID3算法算得的信息增益为：

利用ID3算法计算得到的十二个标签的信息增益如下图：

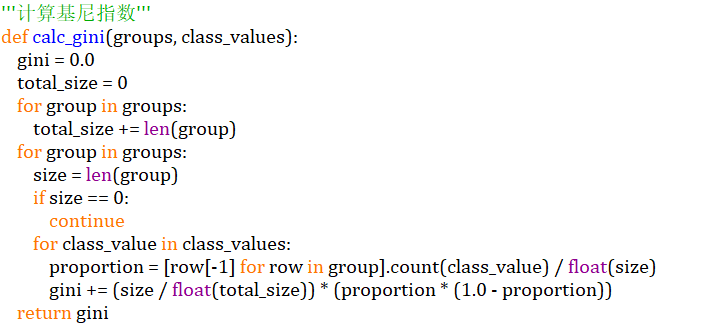


信息增益较大的七个标签为：年龄、婚姻状况、受教育时间、资本收益、每周工作小时数、工作类别和资本损益。由于资本收益的增益较大所以也把资本亏损加入进来考虑（可能是因为样本中资本亏损的个数较少）。

于是把上述七个重要特征作为数据的特征向量。

1. 决策树的构建以及训练和测试的结果

决策树的构建：



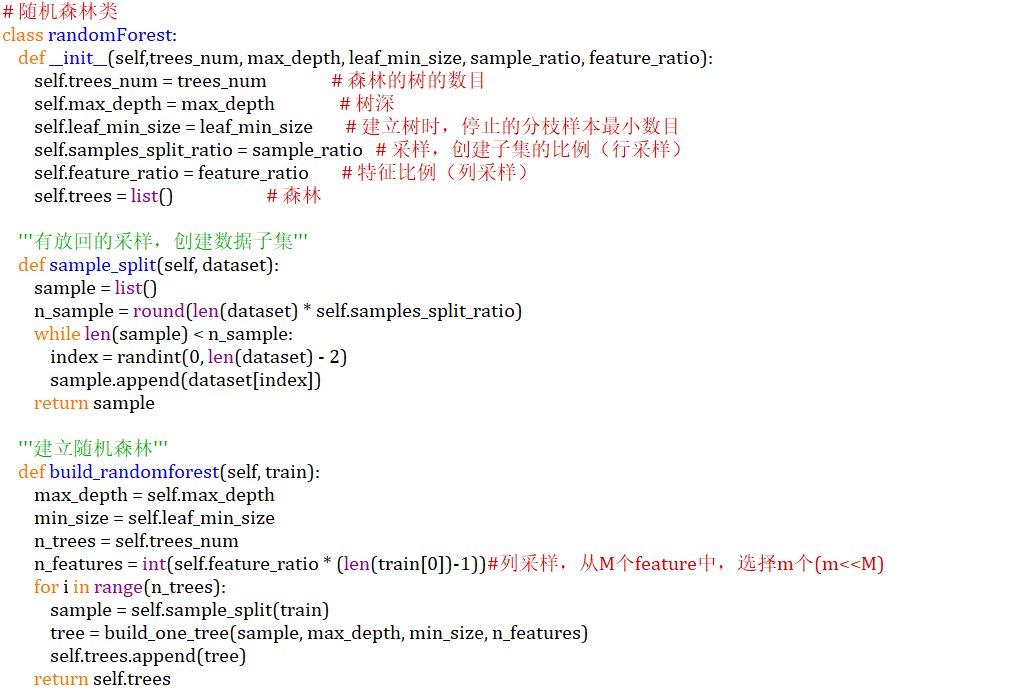
计算每个重要特征的基尼系数：

基尼系数越小表示数据集越稳定。



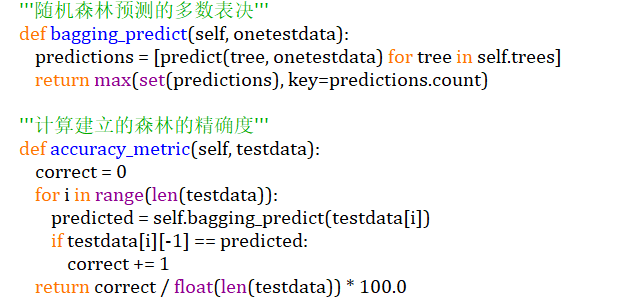
计算得到每个重要特征的基尼系数后。我们把越重要，对结果影响越大的特征放在决策树的上部。先从最有把握的特征对结果进行预测可以提高模型预测的准确率。

创建随机森林类：



构建随机森林是有三个可调整的参数。

1.决策树的个数 trees\_nums 2.每颗决策树用到的重要特征的个数 feature\_ratio 3.树深（递归次数） max\_depth。（注：真正的列采样数为n\_features，但是它通过feature\_ratio间接调整的。）



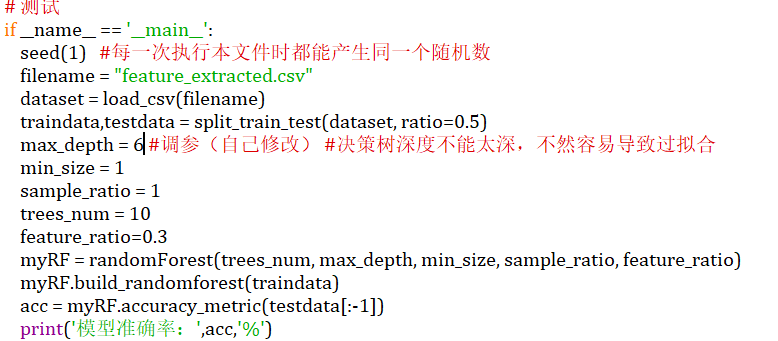
if testdata[i][-1] == predict。默认每个特征向量的最后一个元素是对应的类别。

数据集的传入：



百分之五十的数据作为训练集，百分之五十作为测试集。

预测结果、测试模型：



三个可变参数的选取

首先每颗决策树用到的重要特征的个数要远小于特征的总数，一组数据有七个特征值。因此选取的n\_features值均为2，即feature\_ratio的值为，若n\_features值为3，就满足不了n\_features远小于特征总数，若n\_features为1，那就成了简单的统计问题，构建决策树的意义就不大了。

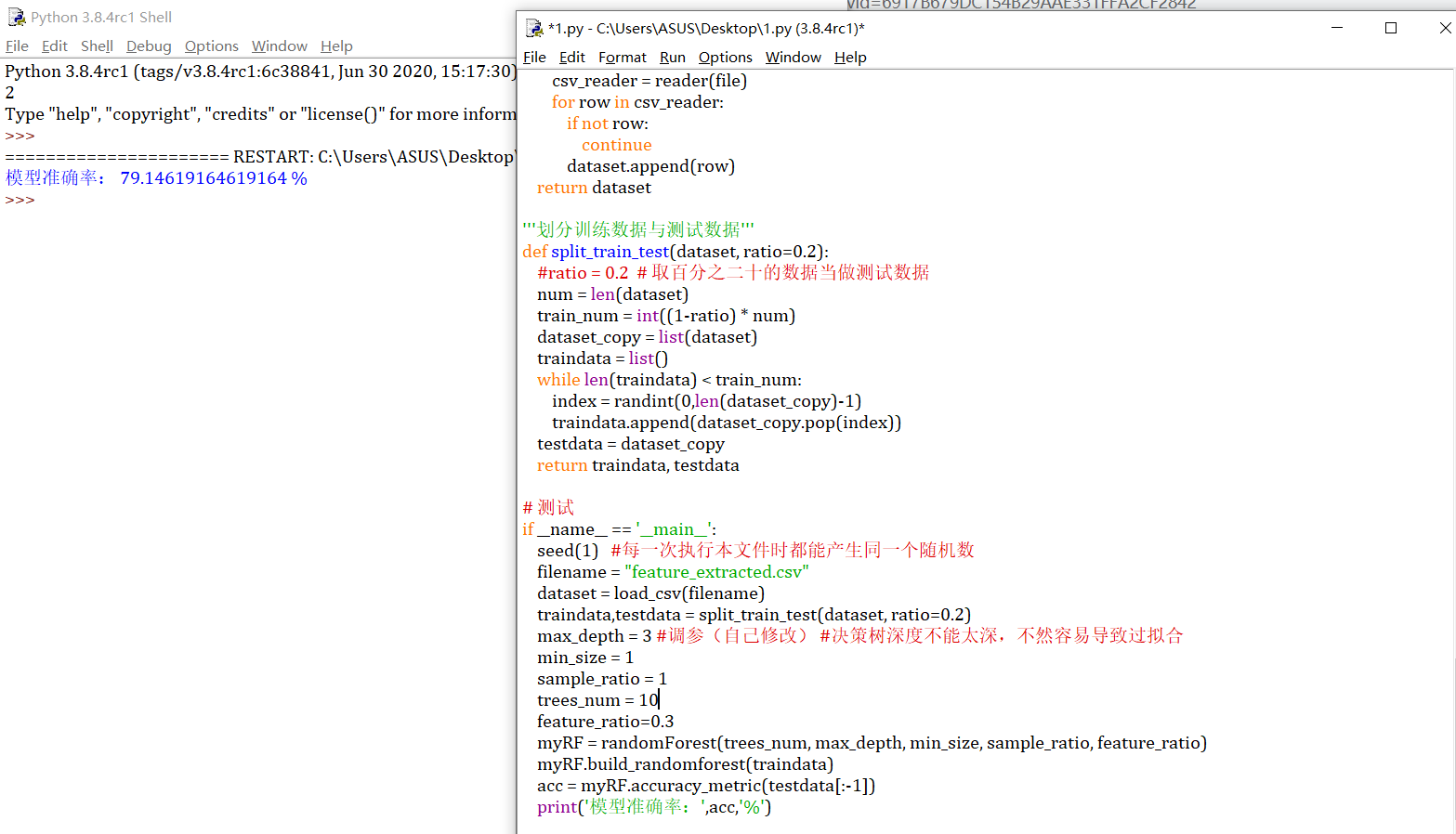
从七个特征值中选取两个作为分支建树，而且每个分支的先后顺序都已经确定了（根据尼基系数确定的），因此不重复的决策树一共有21个。在测试过程中我们分别设置决策树的个数trees\_nums为10、20、30进行了测试。

树深（递归次数），相关文献说树深不能选的太大否则会出现过拟合的状态，一般特征数远大于n\_features时，树深可以选择10以上。由于本项目受测试数据的限制，重要特征的总数有限。因此大部分测试中树深选择为3。

四．具体结果

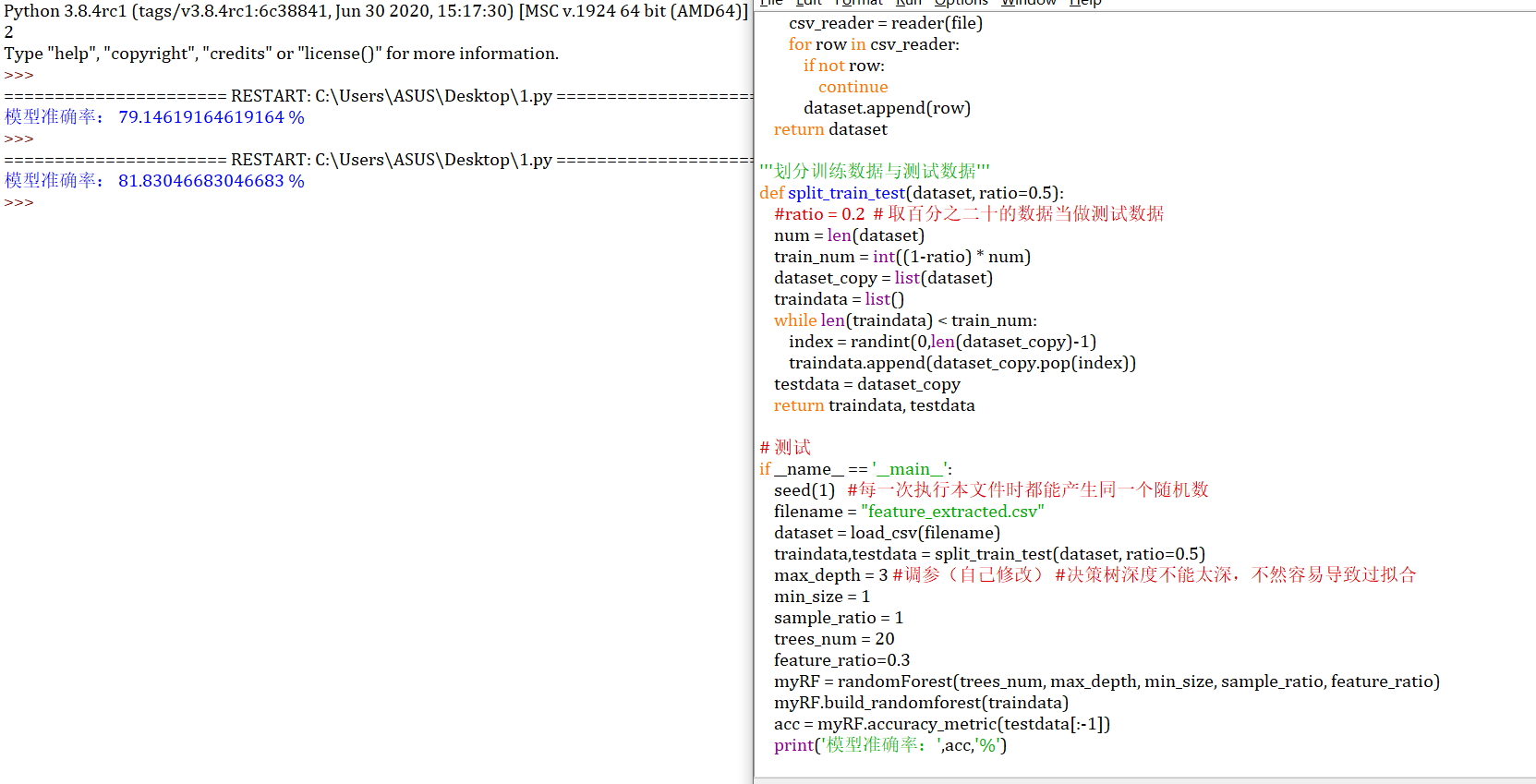
1. feature\_ratio=0.3 n\_features=2 trees\_nums=10 max\_depth=3

模型准确率为79.14619%



1. feature\_ratio=0.3 n\_features=2 trees\_nums=20 max\_depth=3

模型准确率为81.83047%(前三组中最高)



1. feature\_ratio=0.3 n\_features=2 trees\_nums=30 max\_depth=3

模型准确率为80.55282%



1. feature\_ratio=0.3 n\_features=2 trees\_nums=25 max\_depth=3

模型准确率为80.51597%

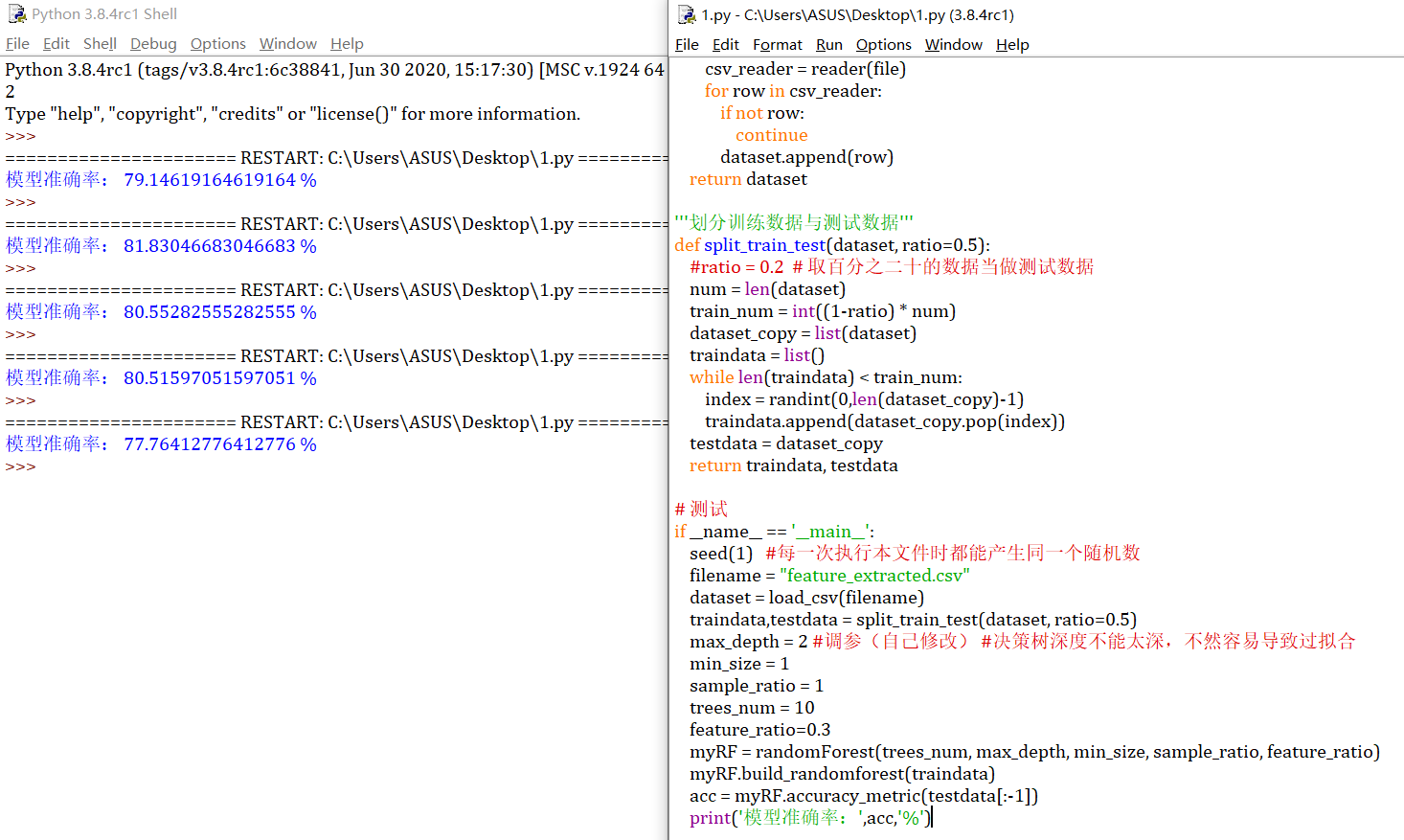
手机屏幕截图

描述已自动生成

前三组预测精度比较发现，选取的决策树个数越接近于最大不重复的决策树个数，模型的预测精度越高。（从七个特征值中选取两个可以得到不重复的决策树为21个，正好第二组选定决策树个数为20的时候，模型预测精度在这四组中最高。）

1. feature\_ratio=0.3 n\_features=2 trees\_nums=10 max\_depth=2

模型准确率为77.76412%（改变树深）



1. feature\_ratio=0.3 n\_features=2 trees\_nums=10 max\_depth=4

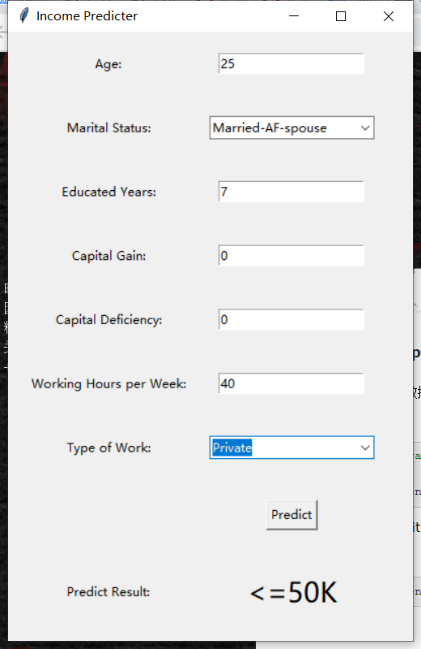
模型准确率为81.79361%（改变树深）

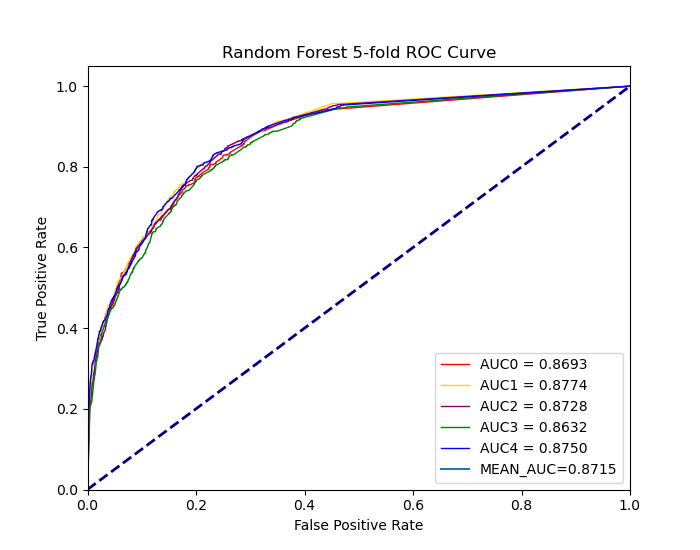
手机截图图社交软件的信息

描述已自动生成

对于单个数据的预测我们做了一个GUI界面。

根据输入的数据（测试者的信息）对该人的工资高低进行预测。



为了更好地评估本预测模型的性能，在一定程度上减少过拟合，并且从数据中提取尽可能多的信息，我们采用5折交叉验证方法对其进行评价。整个数据集被随机分成五个大小大致相同的子集。每个子集依次用作测试集，其他四个子集用作构建模型的训练集。如图所示，最终得到的AUC平均值为0.8715。

五．讨论：

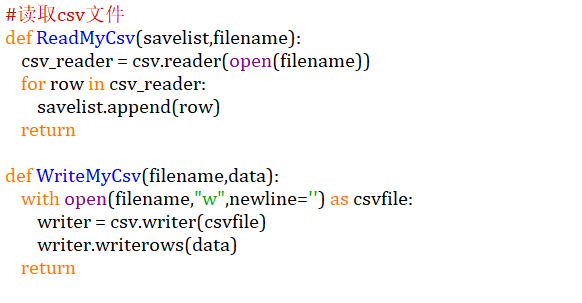
本次项目利用了筛选重要特征、构建决策树、创建随机森林类等方法，构建了一个可以根据公民的年龄、教育程度、婚姻状况等等信息对公民的收入状况的预测的模型。在设置不同参数的测试中该模型的预测精度稳定在80%左右，（最高预测精度有82%）。预测结果比较理想。

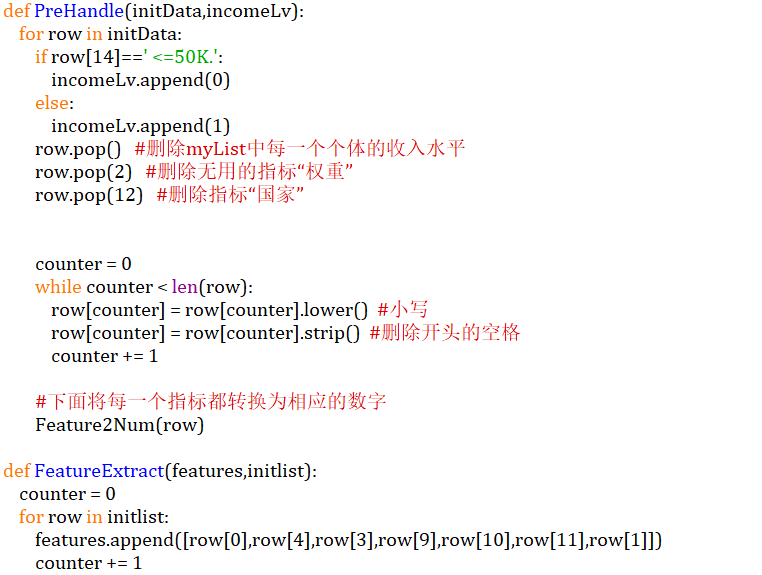
本次项目最关键的两点就是把相关数据从文字信息合理的转为数字信息和利用ID3算法筛选出重要特征。前者把复杂多样的文字信息归类并用数字表示极大地简化了统计的工作量，是整个项目的基础；后者筛选出的与收入有密切联系的七个重要特征，对于模型预测的精确性有重要影响。

本次项目也有一些不足：

1. 统计数据的列数不足（公民的相关信息类型过少），导致每个数据的特征向量的维度不够大（只有七个）。很难满足重要特征的总数远远大于每个决策树用到的重要特征的个数n\_features，很难保证每个决策树之间的独立性。
2. 树深（递归次数）选取的偏小。多次递归可以使得预测更加精确，但是由于担心出现过拟合的状态，于是之前的项目一直将树深选为3，但是后来经过多次测试，在其它参数一定的情况下（feature\_ratio=0.3 n\_features=2 trees\_nums=10），树深从2到6，预测精度是不断变高的（提高精度有3%）。说明该树深选为6时并未出现过拟合的情况，而且模型的最大预测精度可能远高于82%
3. 训练集和测试集的比例选的不太好，本次项目中训练集和测试集的数量比例为1：1。本次项目使用到的数据为一万六千个，用八千多个数据来测试有些多余，可以调成7：3或8：2。
4. 本次模型虽然精度较高，但是预测的时间较长，所以我们又使用了工具包sklearn对整个模型进行了完善，得到了最终版--buildRF.py。在新的模型中我们将读取数据、重要标签的筛选、决策树的构建和随机森林的构建合并在了一起。极大的提升了运行速度，缩减了代码量。

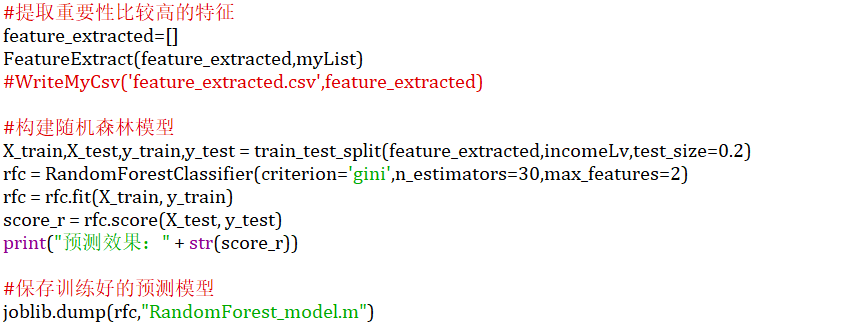
提取数据和数据预处理（部分）的代码如下图：





标签选取和建立模型的代码，如下图：

其中n\_estimators为决策树的个数、max\_features为建树时查找最佳分裂所需考虑的特征数，



n\_estimators=20、max\_features=2、训练集：测试集=4：1

模型准确率为83.42032% 高于原有模型的81.83047%



新构建的模型比原有模型运算速度更快，预测精度也有略微的提高。

六．参考文献

[1]. https://blog.csdn.net/weixin\_39449570/article/details/85061228

[2]. https://baike.baidu.com/item/ID3%E7%AE%97%E6%B3%95/5522381

[3]. https://blog.csdn.net/yangyin007/article/details/82385967

[4]. https://www.cnblogs.com/mdevelopment/p/9450909.html

[5]. https://blog.csdn.net/wenqiwenqi123/article/details/79232536