****

课 程：2019软件工程综合实训

项 目： 数据挖掘实训项目

院 系：数据科学与计算机学院

专 业： 软件工程

学生姓名： 谭江华

学 号： 16340203

授课教师： 郑子彬

2019 年 7 月 13 日

|  |
| --- |
| **实训报告要求**  每位同学根据自己具体的实战比赛过程撰写实训报告。   1. 请用提供的该标准模板，正文长度大于或是等于5页（不含本模板的第一页、第二页、目录页以及参考文献页，提交文件时请将本页删掉） 2. 在参考文献页最后附上自己的比赛代码上传github的链接 3. 文字内容严禁抄袭，会用自动比对系统进行检查，如果的确需要使用网上已有的内容，请做好引用，标明出处。 4. 做到图文并茂，图片尽量自己画，如果实在需要用网上的图片，请做好引用，标明图片出处，否则视为抄袭。 5. 公式请在word里面直接插入公式，别用图片。 6. 正文字号为小四号字体，行距为单倍行距。   提交deadline：2019/07/13 24:00  提交方式：报告连同展示ppt（共两样）放在同一文件夹发送至邮箱  mamm7@mail2.sysu.edu.cn  文件命名：姓名+学号+ppt/报告/截图  文件夹命名：姓名+学号 |

**目 录**

目录

**[比赛介绍](#_Toc23781_WPSOffice_Level1)** **[4](#_Toc23781_WPSOffice_Level1)**

**[赛事数据分析](#_Toc16252_WPSOffice_Level1)** **[4](#_Toc16252_WPSOffice_Level1)**

**[进行的工作](#_Toc13882_WPSOffice_Level1)** **[6](#_Toc13882_WPSOffice_Level1)**

[配置开发环境：](#_Toc16252_WPSOffice_Level2) [6](#_Toc16252_WPSOffice_Level2)

[安装模块：](#_Toc13882_WPSOffice_Level2) [6](#_Toc13882_WPSOffice_Level2)

[数据预处理：](#_Toc27343_WPSOffice_Level2) [6](#_Toc27343_WPSOffice_Level2)

[特征工程：](#_Toc15056_WPSOffice_Level2) [7](#_Toc15056_WPSOffice_Level2)

[模型训练：](#_Toc6400_WPSOffice_Level2) [9](#_Toc6400_WPSOffice_Level2)

**[提交成绩](#_Toc27343_WPSOffice_Level1)** **[10](#_Toc27343_WPSOffice_Level1)**

**[代码展示](#_Toc15056_WPSOffice_Level1)** **[10](#_Toc15056_WPSOffice_Level1)**

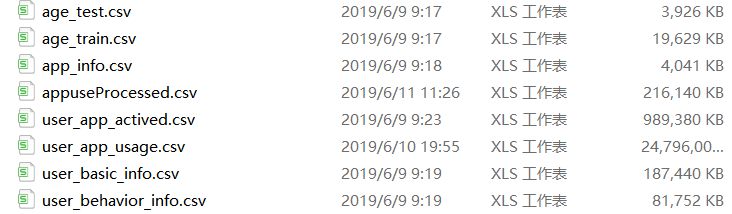
**比赛介绍**

本学期参加的实训为数据挖掘的实训，通过参加DIGIX极客算法精英大赛，根据手机用户的人口属性（如性别、年龄、常驻地等）数据对手机用户群进行画像分析，猜测用户的年龄段，方便厂商了解产品的人群定位和优化产品设计等。

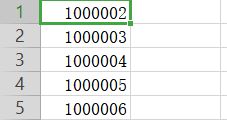
年龄是用户人口属性的重要维度，本次比赛任务为根据用户的手机使用行为习惯来预估用户所处的年龄段。每个用户（以唯一ID标识）对应唯一的年龄段。年龄段有6种划分，分别代表不同年龄段，分别为：小于等于18岁， 19-23岁， 24-34岁， 35-44岁， 45-54岁，大于等于55岁。我们需要做的工作是根据华为提供数据构建预测模型进行年龄段预估，在测试数据集上给出预估结果。

**赛事数据分析**

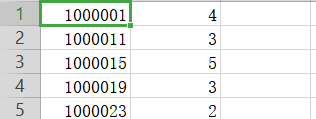
根据我报名之后从官网下载的数据来看，一共有如下几个文件：



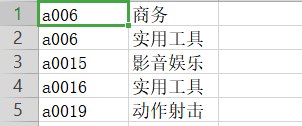
其中age\_test文件中只有一列数据，那就是用户标识uId，并且是经过匿名化处理的用户唯一标识，取值从1000001开始，依次递增。



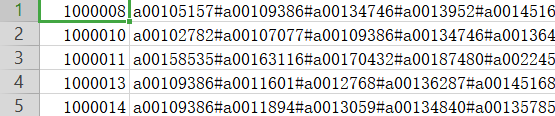
age\_train文件中，有两列数据，一列也是用户标识uId，另一列是该用户对应的年龄范围，年龄的范围取值有1,2,3,4,5,6这六个数字，其中，1代表小于等于18岁，2代表19-23岁，3代表24-34岁，4代表35-44岁，5代表45-54岁，6代表大于等于55岁。



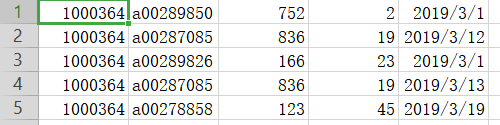
app\_info文件则有两列数据，一列为应用标识appId，另一列为应用类型category。此外如果一个app属于多种类型，那么它会有多条记录。



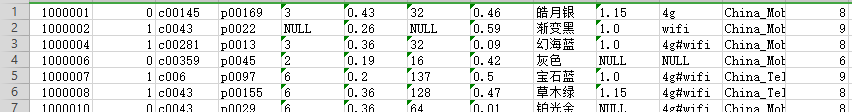
user\_app\_actived是用户激活app的列表文件，是一个强特征表，在我的特征工程中这一个文件会被重点用到，他也有两列数据，一个为用户标识uId，另一个为用户激活的应用标识appId，每个app都用一个#隔开，一个用户可以激活若干个app。



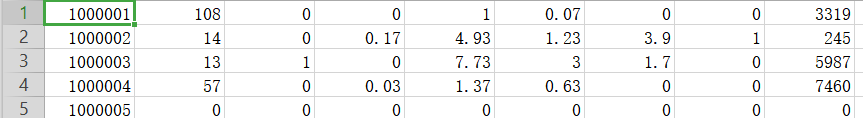
user\_app\_usage文件存放了30天内按天统计每个用户对具体某个app的累计打开次数和使用时长，格式为：“uId，appId，duration，times，use\_date”，由于它太大，我没有使用到它。



user\_basic\_info为用户基本特征数据集，每行代表一个用户的基本信息，包含用户人口属性，设备基本属性，各字段之间由逗号分隔，格式为：“uId，gender，city，prodName，ramCapacity，ramLeftRation，romCapacity，romLeftRation，color，fontSize，ct，carrier，os”。如果特征取值未知，均以NULL表示。



user\_ behavior\_info为用户行为类汇总特征数据集，每行代表一个用户的行为类信息，包含对设备的使用行为汇总数据。如果特征取值未知，均以0表示。



**进行的工作**

**配置开发环境**：

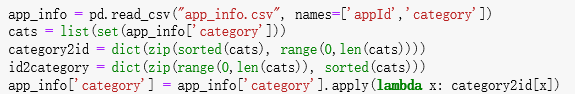
win10+python2.7（使用jupyter notebook运行ipynb文件，这个交互式文件可以在文档中执行代码，并且以网页形式展示运行结果和保存运行记录）

**安装模块：**

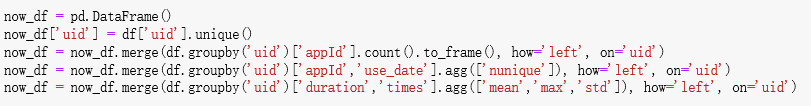
由于本赛题数据量较大，要处理的东西多，所以要安装numpy（高性能科学计算和数据分析的基础包，可以方便的进行数组处理），并在numpy的基础上，安装pandas，使得以numpy为中心的应用变得更加简单，pandas是一个公认的数据处理利器，本次比赛中使用到的例如DataFrame这种数据结构，使得处理数据更加方便。

**数据预处理：**

首先在读取完数据之后，对数据量较大的user\_app\_usage.csv和app\_info.csv进行合并，这里由于app\_info文件中的应用类型是中文字符（具体应用类型），因此在合并之前，要将具体类型转换为可训练的字符类型，这里是将其变为数字。

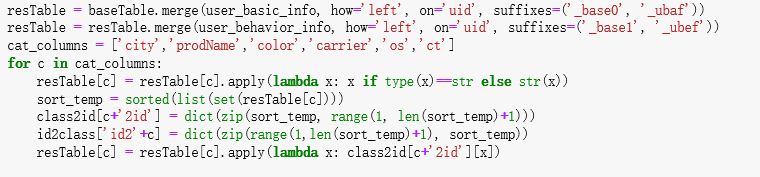


之后顺便统计出每个用户使用app的时长duration的中值，最大值，标准差，以及打开次数times的中值，最大值，标准差等。



之后重新命名这些列，并且根据uId将两个表合并起来，将他们存为一个新的csv文件，取名为appuseProcessed.csv。

然后还要将训练文件和测试文件分别和user\_basic\_info.csv 和 user\_behavior\_info.csv合并起来，并且将这两个文件中的字符值编码成可以训练的数值类型。



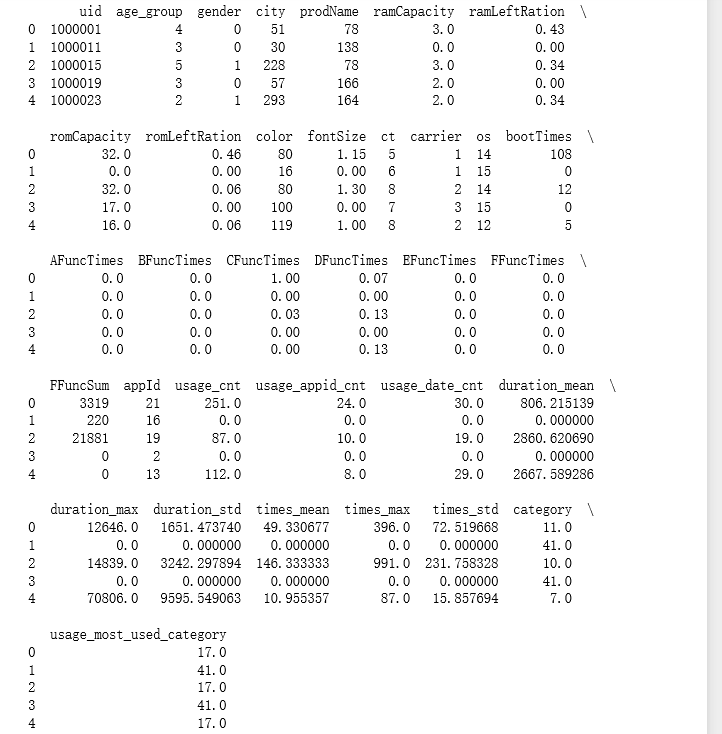
将上面合并得到的表，再和user\_app\_actived这张强特征表以及app\_info.csv对用户激活的app类型进行统计整合。（即利用user\_app\_actived进行一次特征工程）。

**特征工程：**

主要实现在mergeAppData这个函数中。

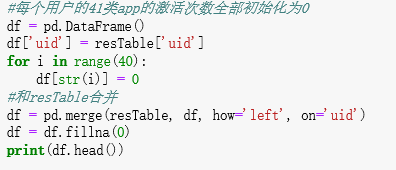


首先基于之前数据预处理过的表，将它和user\_app\_actived表根据uid进行一次合并，并且统计出每个uid对应的激活app的个数，再和前面得到的appuseProcessed表进行合并，这样就简单地结合了user\_app\_actived这个特征表，可以输出结合后的表看看。

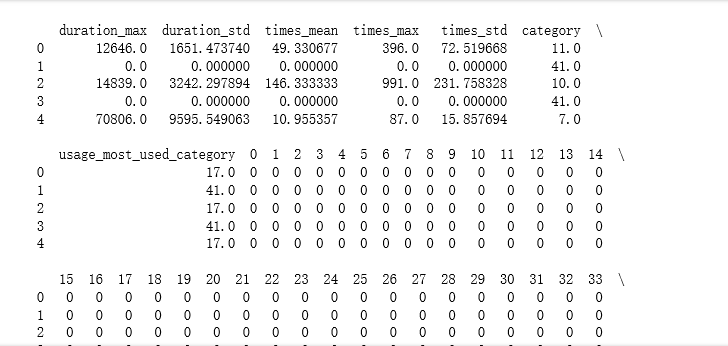


但这还不够，如果可以对每个用户激活的app类型个数都进行一下统计，或许预测效果会更好，于是我对use\_app\_actived再次挖掘。

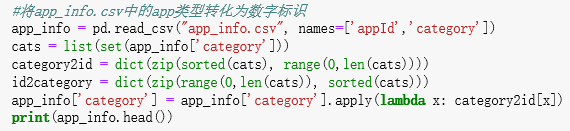
首先根据前面可以知道一共有41中app类型，于是我这里直接创建多一个表，对每个uId都添加41个列数据，分别代表该用户对每个app类型的激活次数，由于还没统计，所以我讲他们都初始化为0，等待统计后再填写进结果。



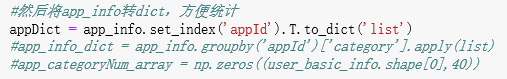
创建完之后顺便和上面简单结合的表再合并一下，并输出表，查看下效果。



这里由于前面的app类型已经被转化成了数字标识，为了方便统计，这里我将app\_info中的app类型也转换为数字标识，步骤和上面一样。

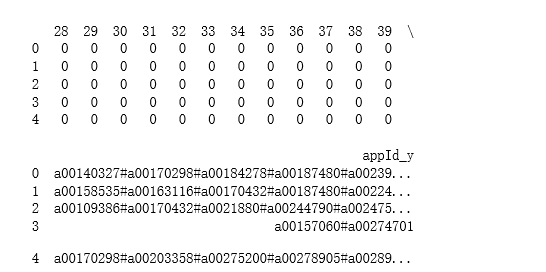


然后为了进一步方便统计，将转成数字标识后的app\_info再转dict。

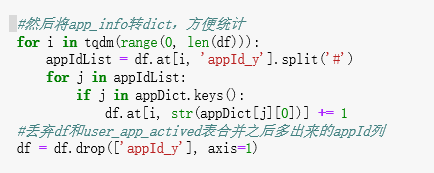


然后为了方便迭代中统计该uId的各类app激活数目，我直接将上面结合过后的df表（含有41列app类型的表）和user\_app\_actived再根据uId合并一次，虽然这两个表都有一个列叫做appId，合并之后会各自命名为appId\_x和appId\_y利于区分。





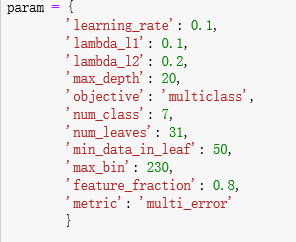
这时候就可以正式进入迭代去统计每个uId对应的激活app类型的数目了。为了方便显示进度，使用tqdm（需要安装这个module）来显示进度。对每个uId中的appId\_y根据#这个符号进行拆分，拆分后，遍历每一个元素，根据前面得到的app\_info转dict后的表，就可以方便地统计41类app的激活数目了。



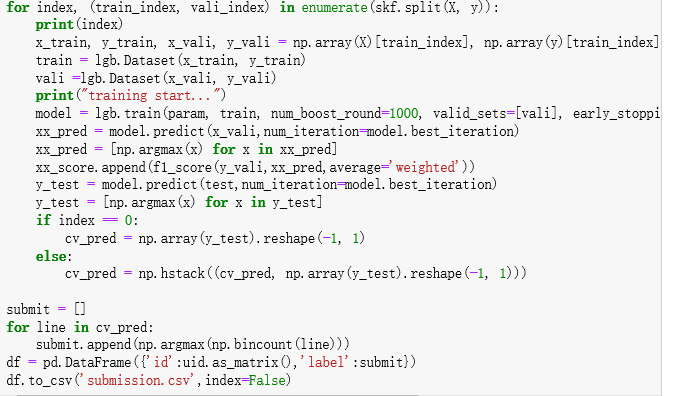
在统计完成后，appId\_y已经没有价值了，顺手将它扔掉。

**模型训练：**

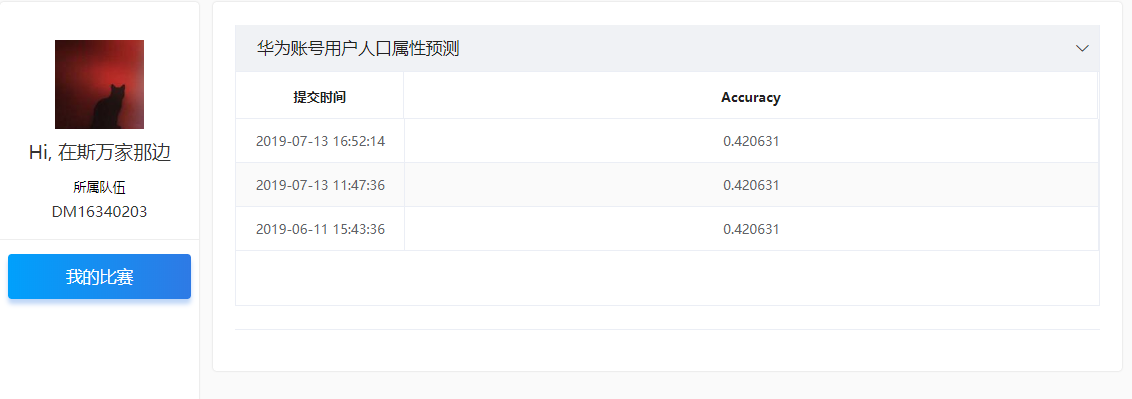
这里利用lightgbm对前面得到的数据进行训练。LightGBM是一个梯度Boosting框架，使用基于决策树的学习算法。在使用它的时候涉及到一系列的参数。



训练过程大致是迭代1k次，找到其中错误率最小的模型，作为输出，然后根据这个算出测试集的结果，存为submission.csv。



**提交成绩**



**代码展示**

<https://github.com/MockingT/DM>