评分：\_\_\_\_\_\_\_\_\_



**数据组工作总结**

**课程名称**  **生产实习A**

**课程号 0101A007**

**授课老师**  **谭福平 等**

**专 业**  **数学与应用数学**

**信息与计算科学**

**学号姓名**  **18120366 谢晨**

**18121474 商紫嫣**

**18122089 周春芳**

**18122610 杨济航**

**18122879 黄欣越**

**18123299 张欣怡**

**打印日期 2021.7.15**

**数据组工作总结**

**摘要**

本文简要介绍了新致人岗匹配项目数据组的任务以及成员，总结了数据组目前的工作成果，包括数据合并、数据结构调整、长文本关键词提取、字段描述与关系说明、正负样本处理、长文本信息汇总、错行处理等，并对后续数据处理工作进行展望。

**一、数据组介绍**

人岗匹配，就是按照“岗得其人”“人适其岗”的原则，根据不同的人个体间不同的素质将不同的人安排在各自最合适的岗位上，从而做到“人尽其才，物尽其用”。

新致人岗匹配项目的立项目的是训练出能够输出个人简历与岗位需求的匹配度的模型，以便用于简历的筛选。项目数据组的主要任务是对新致软件提供的简历与岗位数据进行数据整体认知、数据质量分析、数据预处理，辅助算法组进行特征构造、特征选择等工作以及为算法组提高符合规范的数据。

**二、工作成果总结**

第一阶段：

**1.数据合并**

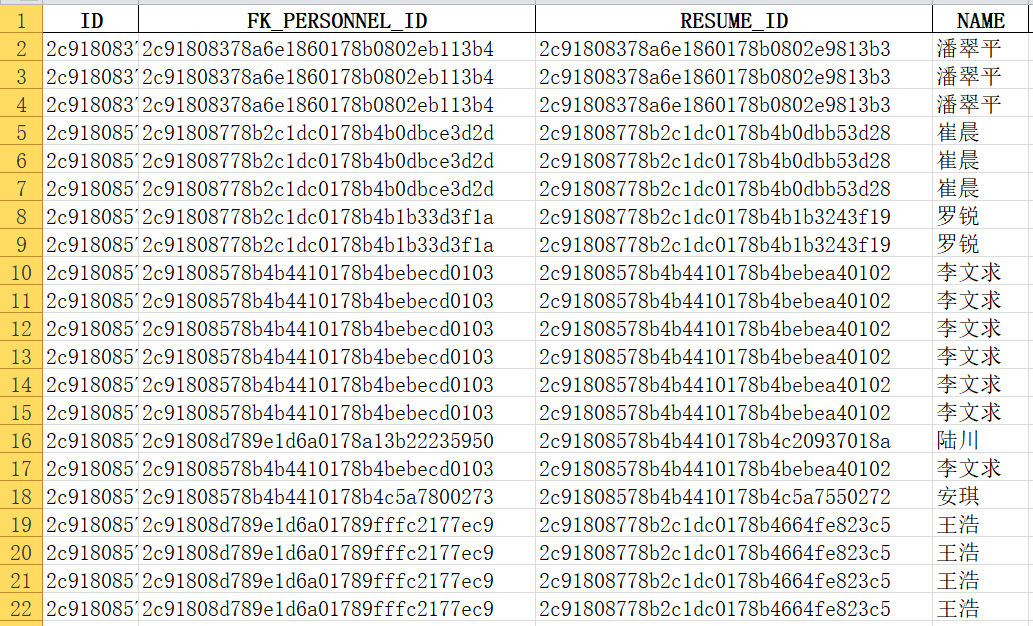
根据新致软件提供的第一批数据（包括：ttyc\_candidate、ttyc\_label、ttyc\_personel\_educational\_experience、ttyc\_personnel\_member、ttyc\_position\_type、ttyc\_position\_label、ttyc\_position、ttyc\_personnel、ttyc\_personel\_label、ttyc\_personnel\_work\_experience、ttyc\_personnel-tfp、ttyc\_personnel\_project\_experienceT）做处理。

处理要求：根据算法组需求将不同文件中求职者的信息汇总。

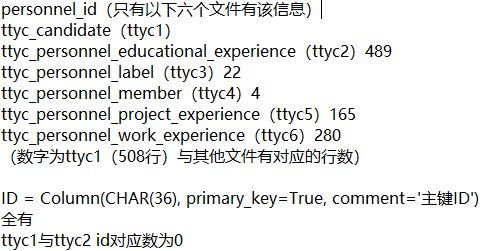
处理方式：因为不同表格中有求职者不同的信息（部分可用），所以我们需要将同一个人的所有信息汇总到一张表格上，以此方便对不同信息的取用。

利用matlab对表格中fk\_id（全局编号）做人的对应和信息汇总尝试。（处理了文件ttyc\_candidate、ttyc\_personnel\_educational\_experience、ttyc\_personnel\_label、ttyc\_personnel\_member、ttyc\_personnel\_project\_experience、ttyc\_personnel\_work\_experience）

结果：合并成功的实例：ttyc\_personnel\_educational\_experience和ttyc\_candidate的合并结果如下，其中ID代表当前表的编号信息，FK\_PERSONNEL\_ID是全局编号，RESUME\_ID是简历中的编号。



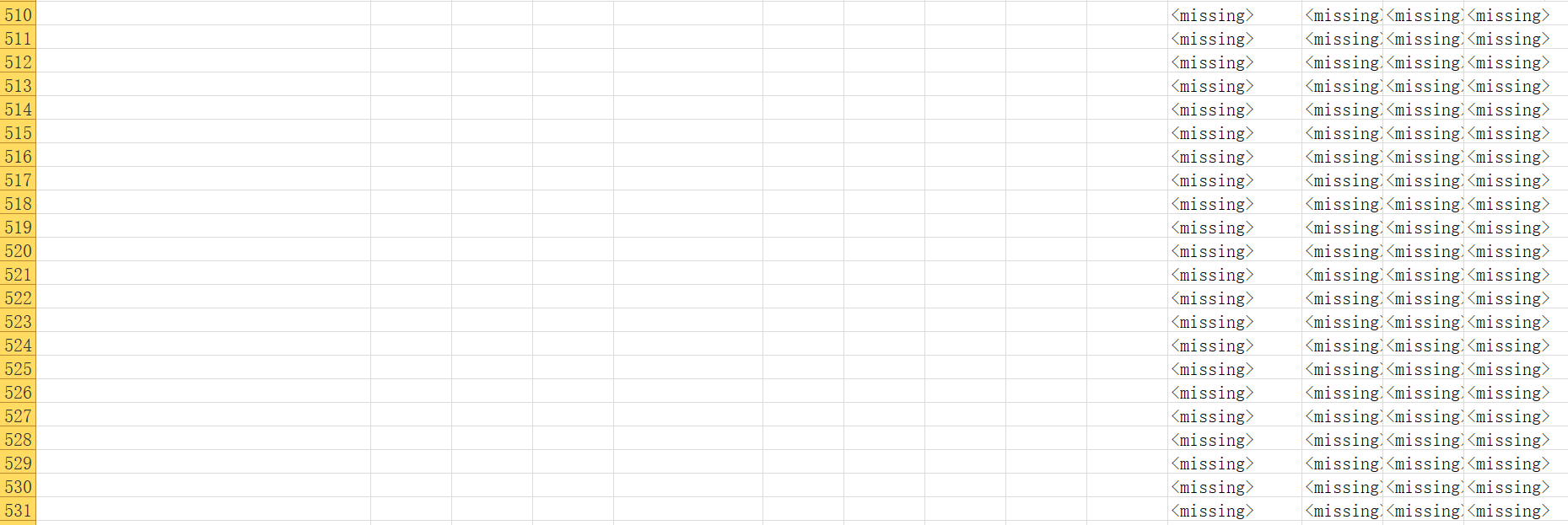
更多结果：



（图中行尾的数字代表ttyc1文件与其他文件所能对应上的行数；对应数为0则印证了ID是在当前表中的编号，在其他表中无效。）

问题与解决：

1. ID的含义：如上所述，我们通过将不同表中代号不同的ID进行对应，弄清了各个ID的具体含义。
2. ID匹配有误导致的信息缺失。针对ttyc\_personnel文件，我们需要求职者的工作经验和个人特长的信息，从文件ttyc\_personnel\_work\_experience中获取，这一步需要将对应ID的人匹配从而获取有效信息，对应率只有百分之五十）。如下图所示，对应不上的ID显示<missing>。在新致提供的第二批数据中，这个问题得到了解决。



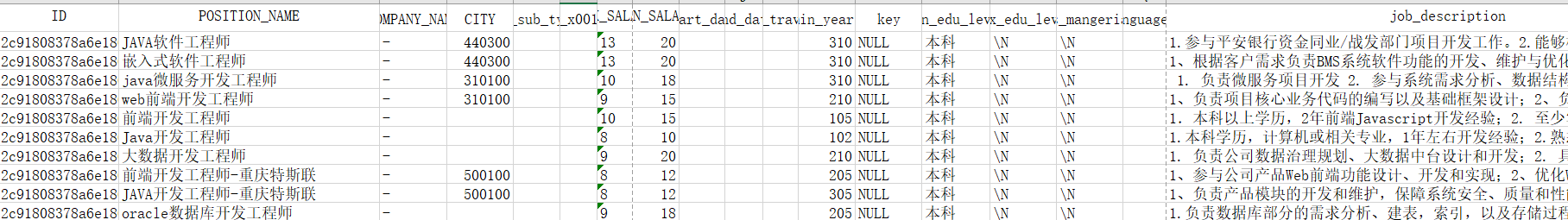
**2.数据结构调整**

处理要求：我们与高珩同学所负责的算法组进行交流后，根据提出的阿里匹配算法的数据要求，对新致提供的数据进行了结构调整以适应算法所需代码。

处理方式：通过excel处理得到结果table1.xlsx、table2.xlsx。按table1要求对ttyc\_personnel.xlsx进行处理，结果：personnel-table1.xlsx。处理的规范、字段说明以及结果在文件第一阶段的处理结果中。

具体过程：为了得到tabel1，通过excel对ttyc\_personnel.xlsx进行处理，根据所提供的数据参照表，对各个标签进行了删选和整理，保留了主要标签的内容，下图为部分标签。该数据表保留了地域、求职岗位、期望薪资和项目经历等标签。其中一些缺失的项，比如地域，可以通过vlook-up函数匹配从而进行填充。



为了得到tabel2，通过Excel工具，根据所要的标签列进行整合，并按照算法所需要求更改了相关格式。

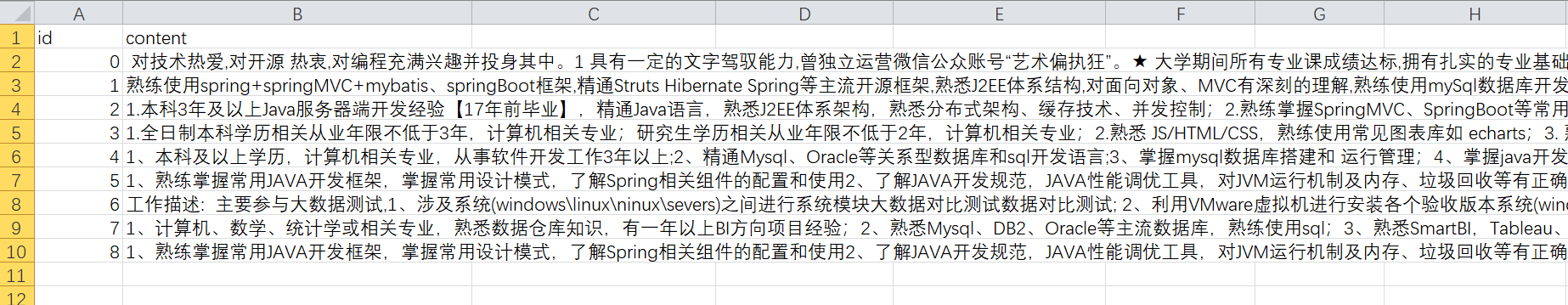
问题：在过程中遇到了数据缺漏较多。如：我们将求职者生活的城市转用编码代替，部分信息缺失，部分信息为非城市名，无法转为编码.

求职者状态不明（求职者状态信息在文件中有当前状态，当前状态有tobeconfirmed，tobearranged等信息，我们不清楚面试流程与状态的对应关系，即不知道哪些样本属于通过初审的，我们需要获知状态与面试流程的对应关系，从而判断如何区分正负样本。

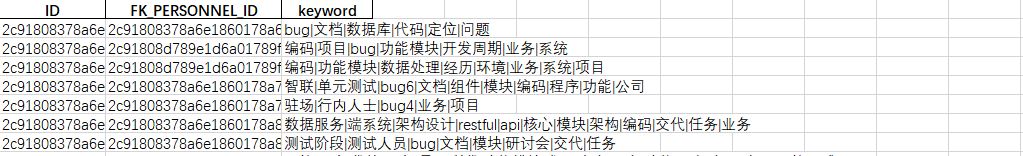
**3．长文本与提取关键词**

处理要求：算法组要求我们提取简历中长文本信息的关键词，以备后续使用。

处理方式：为此我们首先对数据的格式进行了预处理，然后编写代码，通过Jieba分词对长文本进行分词，提取出现频率最高的10个词语作为该文本的关键词。

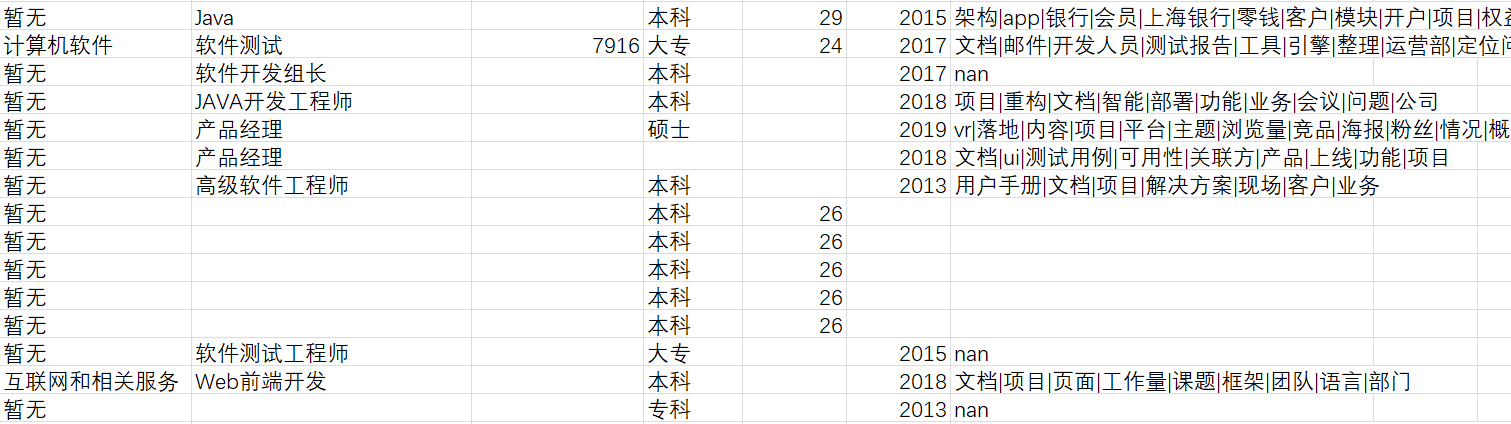
结果：预处理后的格式如下图所示：

切割分词后的结果：



第一列是ID，第二列是全局ID，第三列是提取出的关键词

问题与解决：

1. 处理过程中关键词提取没有问题，但是也遇到了信息缺漏的问题，即上面陈述的求职者experience信息匹配不上导致的信息缺失较多。
2. 针对ttyc\_personnel1文件，我们需要求职者的工作经验和个人特长的信息，从文件ttyc\_personnel\_work\_experience中获取，这一步需要将对应ID的人匹配从而获取有效信息，对应率只有百分之五十）。在结果文件（personnel-table1(1)）中，我们可以看到关键词的缺失。如下图所示，最后一列中的文本即为关键词，nan代表原始数据便缺失，空白行即为对应错误。

第一阶段补充：

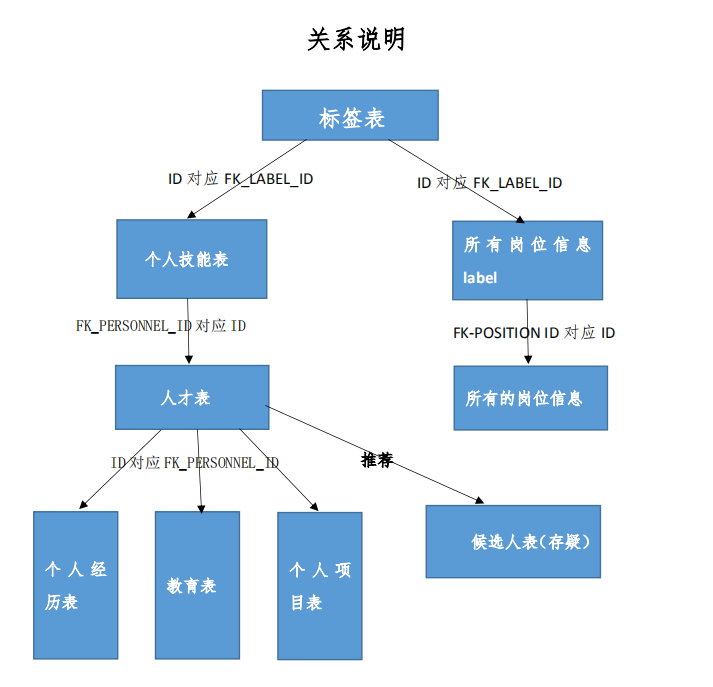
除上述问题外，数据中的“求职者状态”一列含义不明，因状态不明，很难对数据给予分类和标签，进而设计算法。另外，算法组对该算法实现存在困难，因此我们处理后的数据最终未被采用。

第二阶段（处理4.30-6.7的新数据）

**1.****字段描述与关系说明**

处理要求：通过分析新致软件提供的数据中的实体关系，进行岗位推荐数据的字段描述和关系说明的整理，同时结合对表格间主外键关系的分析，搞清楚了每一个字段的信息以及字段间的关系与对应，为后续工作打好基础。

具体过程：仔细研究了4.30-6.7内所有数据表中的各个标签的含义和关系，分析了它们的作用。还根据表内的标签理解了各个表之间的关系。叙述并绘制了关系说明，作了数据库文档，绘制了关系说明图，方便其他项目小组成员阅读和理解数据。



**2.正负样本处理**

处理要求：依据算法组要求，制定候选人表中样本是否通过初筛的判断条件，并以此条件筛选出有标签的候选人正负样本集。

处理方式：根据面试流程图，我们结合数据中的两列信息（candidate state：面试者处在的面试状态 和inappropriate before state：被拒绝的求职者被拒绝时所处的阶段）来判断求职者具体有没有通过初筛。候选人有如下几种情况：

1/第一列为inappropriate 第二列为tobeconfirmed

求职者已被拒，被拒绝时的阶段为待确认阶段，即初筛阶段。

标签：未通过初筛（721条），即负样本。

2/第一列为tobeconfirmed 第二列为null

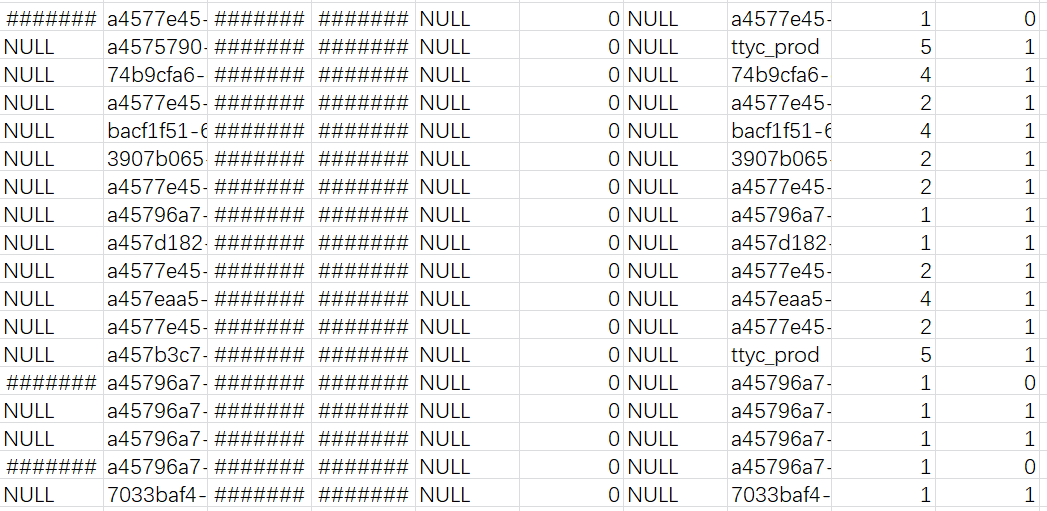
简历还在待初筛阶段，没有结果，是无效信息。

标签：无，可以考虑作为作为测试集（6248条）。

3/其他

标签：通过初筛（3853条），即正样本。

结果：文件候选人+标签.xlsx，例子如下图所示



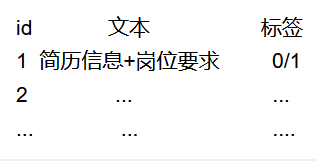
最后一列为标签：0代表负样本，即未通过初筛的简历；1代表正样本，即通过了初筛的简历。

问题与解决：

我们在第一阶段中反映了求职者状态不明的问题，而在收到求职的流程图后，我们研究并判断了各个简历的最终状态，由此来将正负样本分开，解决了这一问题。通过分析筛选后的数据发现候选人的正负样本不平衡，需要在后续处理中注意。

**3.长文本信息汇总**

处理要求：算法组要求将数据处理成如下格式，以便后续使用。



处理方式：根据算法组要求汇总长文本信息（简历信息+岗位要求）。其中，简历信息包括包括应聘者的PERSONAL\_ADVANTAGE（个人优势描述），个人项目经历中的EXPERIENCE\_DESCRIBE（经历介绍）和RESPONSIBILITIES（责任描述）；岗位要求则POSITON\_DISCRIPTION(职业描述)和POSITION\_REQUIREMENT(职业要求)。应聘者普遍项目经历丰富，同一个人有多段项目经历，对信息进行字符串合并作为其总的项目经历的长文本信息，如下图所示：

我们通过matlab编写代码进行实现，以此提高数据处理的效率。详细过程和代码在附录中（代码是通用的，后续数据在处理成相应格式后，都可以经由此代码进行汇总的实现）

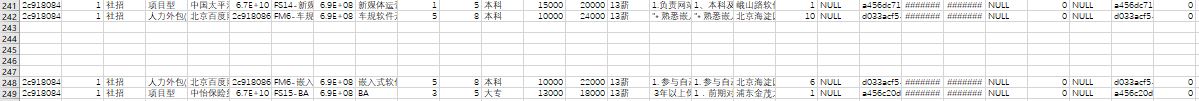
结果：0.csv与1.csv.文件，如下图所示：



**4.错行处理**

处理要求：我们发现数据中有某些行数据有错位：是由于从数据库导出的文本字段中有引号，直接读取csv数据会错位。

处理方式：因为错位数据较少，我们本来计划删除这些数据，发现原因后，我们处理了csv字段中的引号的转义问题，从而解决了错位。根据错行的信息规律，写了两段代码，分别针对个人项目经历和所有的岗位信息两个数据文件，保留了大部分的有效信息。代码原理：我们找到一列有规律的数据，如果该行信息不符合，采用循环的方式将之后的单元格信息向前移动。

结果：处理后的结果如下：

（空白行是原来错行的信息，已被恢复到前一行）

**三、总结与展望**

任何模型的训练都离不开数据，我们在数据处理中遇到了许多困难，也想了许多应对策略。这几个月的工作任务让我们了解到了很多数据探索和数据处理的知识与规范。数据工作对于整个项目来说非常重要，因此，在今后的工作中也要继续增强专业知识，对待问题要考虑周全、灵活应变。在小组方面，要分工合理明确、团结一致，发挥团队协作的巨大力量。

人岗匹配项目的数据还有存在问题，这都需要数据组做进一步处理。在后续工作中，我们计划进一步完善数据质量分析以及数据清洗工作，提高数据质量。

**参考文献：**

[1] 龚纯,王正林编.MATLAB语言常用算法程序集[M].北京: 电子工业出版社,2008.

**[**2] Wes McKinney.Python for Data Analysis[M]. O'Reilly Media,2012.

**致谢**

不知不觉，夏季学期生产实习已经接近尾声。在此期间，我们数据组完成了许多数据处理任务，学会了如何进行数据工作，收获了很多实践经验，而这与众多老师与同学的帮助是分不开的。

感谢谭福平老师对于数据组的耐心指导，在我们迷茫时给出的意见。没有老师的辛勤栽培，孜孜教诲，就没有数据组工作的顺利进行。

同时还要感谢数据组的每一位组员，大家在面对任务时积极主动，细心认真，让数据组工作圆满完成。

**分工比例：**

18120366 谢晨 18%

18121474 商紫嫣 18%

18122089 周春芳 18%

18122610 杨济航 10%

18122879 黄欣越 18%

18123299 张欣怡 18%

**附录**

代码：

1 cuohang1.m

%个人项目经历文件错行问题解决思路：找到一列有规律的，如果该行信息不符合，采用循环的方式将之后的单元格信息向前移动

clc;

clear;

load('matlab1.mat')

a=zeros(511628,1);

for i=2:511628

if(project(i,16)~='0'&&project(i,16)~='1')

a(i,1)=i;

b=strcat(project(i,5),project(i,6));

b=strcat(b,project(i+1,1));

b=strcat(b,project(i+1,2));

b=strcat(b,project(i+1,3));

project(i,5)=b;

for j=6:16

project(i,j)=project(i+2,j-4);

end

end

end

%%清除错误信息

for i=2:511628

if(project(i,16)~='0'&&project(i,16)~='1')

for j=1:16

project(i,j)='';

end

end

end

writematrix(project,"个人项目经历.csv");

2 cuohang2.m

%所有的岗位信息文件错行问题解决思路：找到一列有规律的，如果该行信息不符合，采用循环的方式将之后的单元格信息向前移动

%%

clc;

clear;

load('matlab2.mat')

a=zeros(712,2);

c=0;

num1=0;%%统计修改前

for i=2:712

if(position(i,28)~='0'&&position(i,28)~='1'&&position(i,28)~='2'&&position(i,28)~='3'&&position(i,28)~='4'&&position(i,28)~='5'&&position(i,28)~='6')

a(i,1)=i;

else

num1=num1+1;

end

if(position(i,2)=="1"||position(i,2)=="0")

a(i,2)=i;

end

end

%%

j=0;

for i=2:712

if(a(i-1,1)==0&&a(i,1)~=0&&a(i+1,1)~=0&&a(i+1)==0)

b=position(i,16);

j=1;

end

if(a(i-1,1)~=0&&a(i,1)~=0&&a(i+1,1)~=0&&a(i-1,2)~=0)

b=position(i-1,16);

j=1;

end

if(a(i-1,1)~=0&&a(i,1)~=0&&a(i+1,1)~=0)

b=strcat(b,position(i,1));

j=j+1;

end

if((a(i-1,1)~=0&&a(i,1)~=0&&a(i+1,1)~=0&&a(i,2)~=0)||(a(i-1,1)~=0&&a(i,1)~=0&&a(i+1,1)==0))

for k=18:28

c=position(i,k-16);

position(i-j,k)=c;

position(i-j,16)=b;

position(i-j,17)=b;

end

j=0;

end

end

%%删除无效信息

for i=2:712

if(position(i,28)~='0'&&position(i,28)~='1'&&position(i,28)~='2'&&position(i,28)~='3'&&position(i,28)~='4'&&position(i,28)~='5'&&position(i,28)~='6')

for j=1:28

position(i,j)="";

end

end

end

%%

num2=0;

for i=2:712

if (position(i,28)~='0'&&position(i,28)~='1'&&position(i,28)~='2'&&position(i,28)~='3'&&position(i,28)~='4'&&position(i,28)~='5'&&position(i,28)~='6')

else

num2=num2+1;

end

end

%%

writematrix(position,"position.csv");

3 关键词.py

import jieba

import pandas as pd

import jieba.analyse

'''

用jieba分词并提取长文本关键词的程序

'''

stopwords=[stop.strip() for stop in open('stopwords.txt',encoding='UTF-8').readlines()]#读取停用词

f=pd.read\_excel("sentence.xlsx", header=0)#读取文本

corpus=[]

for text in list(f['content']):

x=text.replace('\t','').replace('\n','').lower()#替换换行符和制表符

segment\_words = jieba.cut(x.strip(), cut\_all=False)#分词

result = ''

for word in segment\_words:

if word not in stopwords and word != ' ':#进一步处理分词格式

result += word

result += " "

result=jieba.analyse.extract\_tags(sentence=result,topK=15,allowPOS=('ns','nt','nz','n','eng'))#提取关键词

key='|'.join(result)

corpus.append(key)

data=[[list(f['id'])[i],corpus[i]] for i in range(len(corpus))]

name=['id','keyword']

test=pd.DataFrame(columns=name,data=data)

print(test)

test.to\_excel('keyword.xlsx',index=False)#导出到xlsx