数据分析与实践实验 4

LaTeX by 常征 PB23030850

2025 年 5 月 7 日

目录

1 （15%）读取数据集 data.csv，进行数据预处理。

1.1 选取问卷中的 SC155Q01HA,SC155Q02HA,SC155Q03HA,SC155Q04HA,SC155Q05HA 个离散性特征作为特征集，分别介绍这些特征所代表的含义和各自取值范围

我们从 codebook 可以得知，各自的取值范围都是 1-4 的整数。

SC155Q01HA 表示连接到互联网的电子设备的数量是否充足，SC155Q02HA 表示学校的网络带宽或者速度是否足够，SC155Q03HA 表示用于教学的电子设备的数量是否充足，SC155Q04HA 表示用于电子设备的计算能力是否充足强，SC155Q05HA 表示用于是否有足够多的软件。

```
file_path = 'data.csv'
df = pd.read_csv(file_path, index_col = 0)
target = ['SC155Q01HA', 'SC155Q02HA', 'SC155Q03HA', 'SC155Q04HA', 'SC155Q05HA']
data = df[target]
data_drop = data.dropna()
data_drop.columns = ['1', '2', '3', '4', '5']
print(data_drop)
```

✓ 0.2s

```
         1    2    3    4    5
1      2.0  2.0  2.0  1.0  1.0
2      2.0  2.0  2.0  2.0  2.0
3      2.0  3.0  1.0  1.0  1.0
4      2.0  3.0  2.0  2.0  2.0
5      2.0  4.0  2.0  2.0  2.0
...    ...  ...  ...  ...  ...
21899  2.0  3.0  2.0  2.0  2.0
21900  3.0  3.0  2.0  2.0  2.0
21901  3.0  3.0  3.0  2.0  2.0
21902  4.0  3.0  3.0  3.0  3.0
21903  3.0  3.0  2.0  2.0  2.0

[20681 rows x 5 columns]
```

图 1: 1.1 结果

1.2 （10%）注意到选取的特征可能存在相同取值（如特征 A 和 B 都可能取值 0），不便于后续的关联分析过程。请构建项集索引，并依据索引内容进行特征值替换。项集索引字典形式如下：

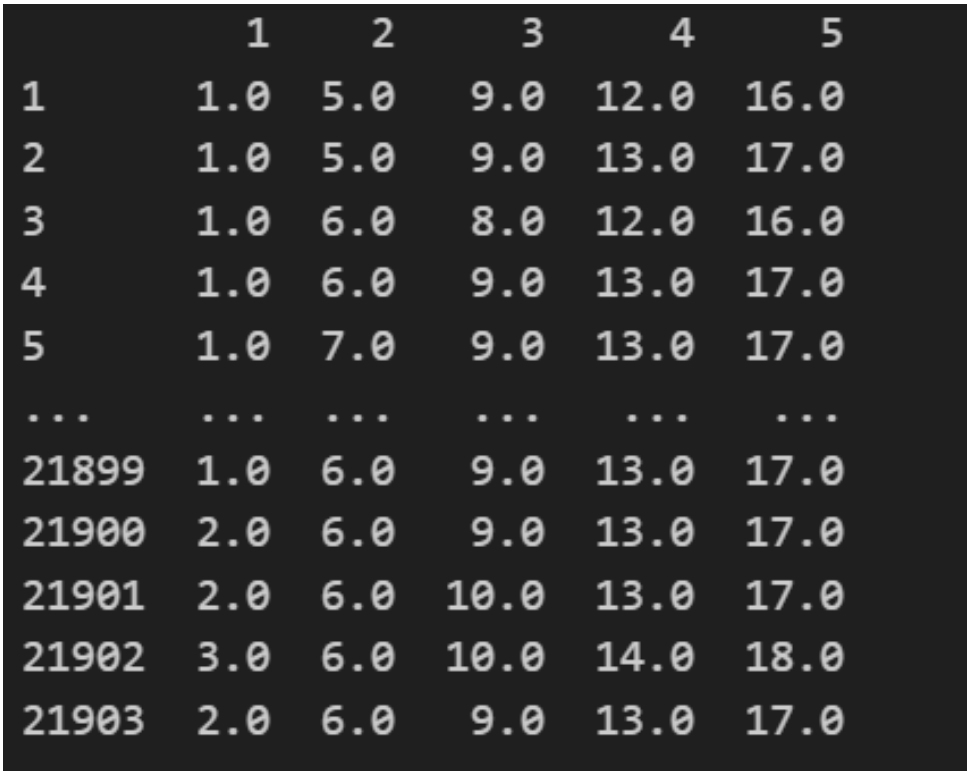
因为有第一问的基础，我们可以知道每一个特征的取值都是 1-4，所以我们可以直接先构建字典，后进行替换。

实际进行替换的时候，我使用了 pd 库 replace 方法。我最开始是逐个元素进行替换，但是跑的时候会报错，这可能是因为上面 dropna 的时候只删除了对应的列，但是没有改列号，所以访问的时候会有 keyerror。

但是用 replace 也有问题，如果采取以下写法的话会出现重复替换的情况，比如 2 先被替换为 1，再被替换为 0。

```
i = 0
for col in data_drop.columns:
    for value in data_drop.values.unique():
        data_replace[col]= data_replace[col].replace(value, value+i-1)
i+=4
```

最后由于我们实际知道每一列的取值都是 1-4，就直接 for value in range(1,5) 了。



	1	2	3	4	5
1	1.0	5.0	9.0	12.0	16.0
2	1.0	5.0	9.0	13.0	17.0
3	1.0	6.0	8.0	12.0	16.0
4	1.0	6.0	9.0	13.0	17.0
5	1.0	7.0	9.0	13.0	17.0
...	...	...	...	...	...
21899	1.0	6.0	9.0	13.0	17.0
21900	2.0	6.0	9.0	13.0	17.0
21901	2.0	6.0	10.0	13.0	17.0
21902	3.0	6.0	10.0	14.0	18.0
21903	2.0	6.0	9.0	13.0	17.0

图 2: 1.2 结果

2 基于预处理后的数据集，编写算法代码进行频繁项集挖掘。

2.1 （30%）请参考以下 Apriori 产生频繁项集的算法流程，自行编写相应代码，分别以最小支持度阈值为 0.25 和 0.5，挖掘频繁项集。

算法听老师讲起来和自己手算的时候都不算困难，但是写成代码的时候就比较麻烦了。

挖掘频繁 1 项集是简单的，下面我以 2 项集为例说明我的代码。

最大的困难在于，如何根据上一次计算出的频繁 k-1 项集的结果，找到所有的候选 k 项集，不过好消息是我上一问起名字的时候偷了懒，给列起名字为'1','2' 之类的单字符，所以我们要判断是不是两个 k-1 项集可以组合成一个 k 项集，只用对字符串进行判断，看是不是除了最后一个以外都相同即可。

前面我们说的是项的组合，比如列 1、2 和列 1、3 均有某个取值是频繁 2 项集，那么列 1、2、3 的某个取值可能是频繁 3 项集。具体是哪个取值我们利用和上面类似的思路进行计算，比如列 1、2 分别取 1，5 的时候是频繁的，那么我们在 13 的频繁项集例尝试查找是不是有包含列 1、2 前 (2-1) 的取值，即寻找有没有列 1 取 1 的频繁项集。

输出的结果是一个嵌套字典，最外层的 key 是数字，比如 1,2, 表示频繁 k 项集，对应的 value 又是一个嵌套字典。

第二层的嵌套字典的 key 表示第几列，例如'1' 表示第一列，'12' 表示第一列和第二列的组合；对应的 value 还是一个嵌套字典。

第三层的嵌套字典是一个普通的字典，其 key 是一个元组，表示实际的频繁 k 项集的取值，比如 (1.0,) 表示 (1.0,) 频繁出现，对应的 value 就是频率。

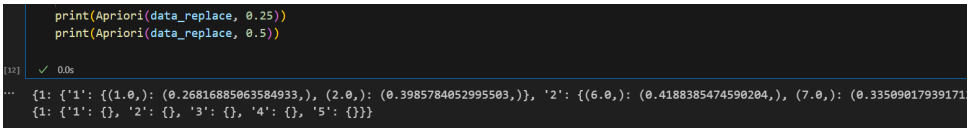


图 3: 2.1 结果

2.2 （15%）当最小支持度为 0.5 时，频繁项集数量较少。请将各特征原始取值为 1 和 2 的单元格统一修改其值为 0，取值为 3 和 4 的单元格统一修改其值为 1。重复 T1-Q2 的项集索引构建过程，并以最小支持持度阈值为 0.5，挖掘频繁项集。

这个基本上就是复制粘贴上面的代码，修改一下具体的数据就好了。

输出的结果解释同上。

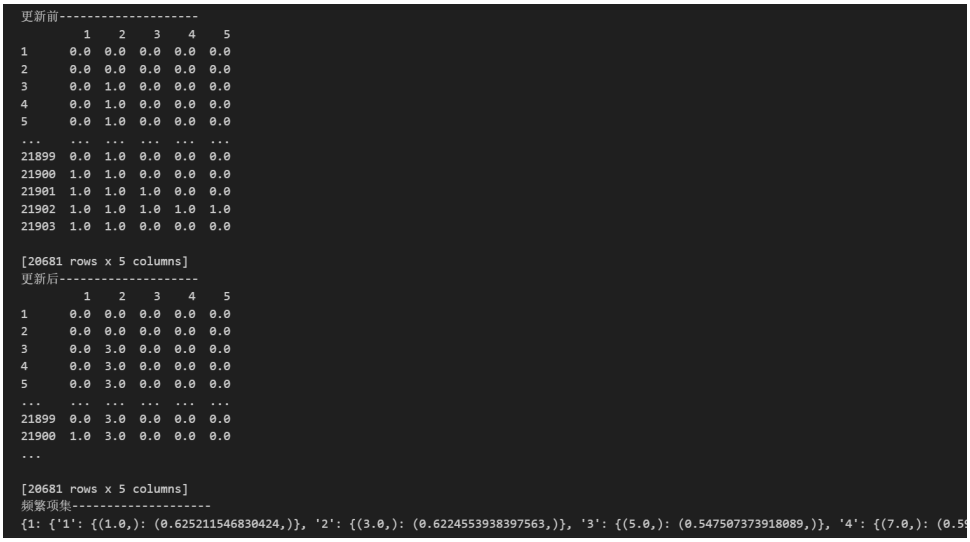


图 4: 2.2 结果

2.3 （15%）分析 Q1 和 Q2 的结果，你有什么发现？请根据各特征定义，分析产生这种情况的原因。

Q1 的 0.5 频繁项集是空的，而对数据进行二值化处理后，Q2 的 0.5 频繁项集相对而言很多。

分析：各个特征的大小实际上就说明的是电子设备等硬件条件如何，原始数据中，各特征的取值层级较多（例如 1 ~ 4 级），导致每一个具体数值在样本中出现的次数都较为分散。，而 Q2 把它简单二分为 0/1，可以理解为好/不好的样本数量分别被集中统计，提高了出现频率，使得频繁项集更容易被识别。

3 基于 T2-Q2 得到的频繁项集挖掘结果，编写算法代码进行关联规则提取。

3.1 请以最小置信度阈值为 0.8，提取形如 X->1 的关联规则，并输出它们的置信度和提升度。

完整的求出所有的关联规则还是很麻烦的，因为相当于要求我们用代码写出一个集合的全部子集，不过这次实验只要求我们求出形如 {X}->{1} 的规则，那就是求出一个集合只含一个元素的子集，更不用说我们使用字符串来表示的，那就是**遍历一个字符串的每一个元素**，就比较简单了。

不过要注意的一点是我的某个字典的索引用的是元组，因此不可以直接 d1[target][val[i]]，需要把 val[i] 写成只含 1 个元素的元组。

```
关联规则: 1.0 -> 3.0,置信度: 0.8226604795050271, 提升度: 1.3216376428682872
关联规则: 3.0 -> 1.0,置信度: 0.8263031150469975, 提升度: 1.3216376428682872
关联规则: 1.0 -> 5.0,置信度: 0.8125290023201858, 提升度: 1.4840512494024343
关联规则: 5.0 -> 1.0,置信度: 0.9278459772145193, 提升度: 1.4840512494024343
关联规则: 1.0 -> 7.0,置信度: 0.8010054137664347, 提升度: 1.3423217698811796
关联规则: 7.0 -> 1.0,置信度: 0.8392350700915647, 提升度: 1.3423217698811796
关联规则: 1.0 -> 9.0,置信度: 0.8194895591647333, 提升度: 1.2957082242420377
关联规则: 9.0 -> 1.0,置信度: 0.8100917431192662, 提升度: 1.2957082242420377
关联规则: 7.0 -> 9.0,置信度: 0.8674337573940524, 提升度: 1.3715135731396328
关联规则: 9.0 -> 7.0,置信度: 0.8184250764525994, 提升度: 1.3715135731396328
```

图 5: 3.1 结果

3.2 （10%）参考项集索引的对应关系，对以上频繁项集和关联规则结果进行简要分析和总结。

频繁项集： 1、3、5、7、9 是频繁项集，而这些都是表示电子资源不错的。那么我们可以得出，大部分受调查的学校普遍电子设施等都比较充足。

关联规则：我发现大部分频繁项都是相互关联的，即如果 1->2 的置信度达到 80%，那么 2->1 的置信度也会达到 80%。

按照我们之前的项集索引，1，3，5，7，9 都对应的是 3、4, 即设备情况比较好的。那么我们可以大致得出结论，**如果学校的某一项电子资源比较好，那么很大概率它各个方面的电子资源都不错**。