报告

学号: 3021244270　　姓名: 黄琬婷　　班级:新工科一班

1. 目标

练习使用关联规则 Apriori 算法。完成以下两个任务，并与作业里手动计算的结果进行对比分析。

2. 数据

课堂收集的真实数据和 DBLP 数据集

任务 1

选择以下所有人、男生、女生数据分别进行关联规则的抽取，并与作业中进行所有人数据抽取的关联规则进行比较分析

ID	性别	常用 app
1	女	微信、小红书、Ins、菜鸟、知到
2	女	微信、小红书、淘宝、bilibili
3	女	夸克、B 站、知乎、qq、微信
4	女	微信、哔哩哔哩、明日方舟、淘宝、微北洋
5	女	微信、QQ、菜鸟、知到、知乎、bilibili、爱奇艺、淘宝
6	男	微信、QQ、抖音、12306、知到、网易云
7	男	Qq、微信、bilibili、知到、刺猬猫、美团
8	男	Qq、微信、B 站、京东、高德地图
9	男	微信、QQ、淘宝、B 站、扇贝单词
10	男	微信、QQ、B 站、原神

执行 Apriori 算法，记录算法设置和结果，要求:

1）统一表格 app 的名称

ID	性别	常用 app
1	女	微信、小红书、Ins、菜鸟、知到
2	女	微信、小红书、淘宝、B 站
3	女	夸克、B 站、知乎、qq、微信
4	女	微信、B 站、明日方舟、淘宝、微北洋
5	女	微信、QQ、菜鸟、知到、知乎、B 站、爱奇艺、淘宝
6	男	微信、QQ、抖音、12306、知到、网易云
7	男	QQ、微信、B 站、知到、刺猬猫、美团
8	男	QQ、微信、B 站、京东、高德地图
9	男	微信、QQ、淘宝、B 站、扇贝单词
10	男	微信、QQ、B 站、原神

2)编写 Apriori 算法，以及相应备注

```
min_sup = 2
min_conf = 0.6 # 最大 K 项集
K = 4

# apriori 算法
def apriori():
#   1.读入数据
    data_set = load_data() # 2.计算每项的支持数
    C1 = create_C1(data_set)
    item_count = count_itemset1(data_set, C1)
    # 3.剪枝，去掉支持数小于最小支持度数的项
    L1 = generate_L1(item_count)
    # 4.连接
    # 5.扫描前一个项集，剪枝# 6.计数，剪枝
    # 7.重复 4-6，直到得到最终的 K 项频繁项集
    Lk_copy = L1.copy()
    L = []
    L.append(Lk_copy)
    for i in range(2, K + 1):
        Ci = create_Ck(Lk_copy, i)
        Li = generate_Lk_by_Ck(Ci, data_set)
        Lk_copy = Li.copy()
        L.append(Lk_copy)
    # 8.输出频繁项集及其支持度数
    print('频繁项集\t 支持度计数')
    support_data = {}
    for item in L:
        for i in item:
```

```
            print(list(i), '\t', item[i])
            support_data[i] = item[i]
    # 9.对每个关联规则计算置信度，保留大于最小置信度的频繁项为 强关联规则
    strong_rules_list = generate_strong_rules(L, support_data,
data_set)
    strong_rules_list.sort(key=lambda result: result[2], reverse=True)
    print("\nStrong association rule\nX\t\t\tY\t\tconf")
    for item in strong_rules_list:
        print(list(item[0]), "\t", list(item[1]), "\t %.2f" %
(item[2]))


# 读入数据
def load_data():
# 事务 ID 购买商品
   data = {'001': '微信 小红书 Ins 菜鸟 知到', '002': '微信 小红书 淘宝 B 站
','003': '夸克 B 站 知乎 qq 微信', '004': '微信 B 站 明日方舟 淘宝 微北洋
','005': '微信 QQ 菜鸟 知到 知乎 B 站 爱奇艺 淘宝'}

   #data = {'006': '微信 QQ 抖音 12306 知到 网易云','007': 'QQ 微信 B 站 知
到 刺猬猫 美团', '008': 'QQ 微信 B 站 京东 高德地图','009': '微信 QQ 淘宝 B
站 扇贝单词', '010': '微信 QQ B 站 原神'}
    data_set = []
    for key in data:
        item = data[key].split(' ')
        data_set.append(item)
    return data_set


# 构建 1-项集
def create_C1(data_set):
    C1 = set()
    for t in data_set:
        for item in t:
            item_set = frozenset([item])
            C1.add(item_set)
    return C1



# 计算给定数据每项及其支持数，第一次
def count_itemset1(data_set, C1):
    item_count = {}
    for data in data_set:
        for item in C1:
```

```
            if item.issubset(data):
                if item in item_count:
                    item_count[item] += 1
                else:
                    item_count[item] = 1
    return item_count

# 生成剪枝后的 L1
def generate_L1(item_count):
    L1 = {}
    for i in item_count:
        if item_count[i] >= min_sup:
            L1[i] = item_count[i]
    return L1

# 判断是否该剪枝
def is_apriori(Ck_item, Lk_copy):
    for item in Ck_item:
        sub_Ck = Ck_item - frozenset([item])
        if sub_Ck not in Lk_copy:
            return False
    return True


# 生成 k 项商品集，连接操作
def create_Ck(Lk_copy, k):
    Ck = set()
    len_Lk_copy = len(Lk_copy)
    list_Lk_copy = list(Lk_copy)
    for i in range(len_Lk_copy):
        for j in range(1, len_Lk_copy):
            l1 = list(list_Lk_copy[i])
            l2 = list(list_Lk_copy[j])
            l1.sort()
            l2.sort()
            if l1[0:k - 2] == l2[0:k - 2]:
                Ck_item = list_Lk_copy[i] | list_Lk_copy[j] # 扫描前一个
项集，剪枝
                if is_apriori(Ck_item, Lk_copy):
                    Ck.add(Ck_item)
    return Ck



#  生成剪枝后的 Lk
```

```
def generate_Lk_by_Ck(Ck, data_set):
    item_count = {}
    for data in data_set:
        for item in Ck:
            if item.issubset(data):
                if item in item_count:
                    item_count[item] += 1
                else:
                     item_count[item] = 1
    Lk2 = {}
    for i in item_count:
        if item_count[i] >= min_sup:
            Lk2[i] = item_count[i]
    return Lk2

# 产生强关联规则
def generate_strong_rules(L, support_data, data_set):
    strong_rule_list = []
    sub_set_list = [] # print(L)
    for i in range(0, len(L)):
        for freq_set in L[i]:
            for sub_set in sub_set_list:
                if sub_set.issubset(freq_set):
                # 计算包含 X 的交易数
                    sub_set_num = 0
                    for item in data_set:
                        if (freq_set - sub_set).issubset(item):
                            sub_set_num += 1
                    conf = support_data[freq_set] / sub_set_num
                    strong_rule = (freq_set - sub_set, sub_set, conf)
                    if conf >= min_conf and strong_rule not in
strong_rule_list:
                        strong_rule_list.append(strong_rule)
            sub_set_list.append(freq_set)
    return strong_rule_list


if __name__ == '__main__':
    apriori()
```

3）结果分析

女生：

```
频繁项集          支持度计数
['小红书']        2
['知到']          2
['微信']          5
['菜鸟']          2
['淘宝']          3
['B站']           4
['知乎']          2
['知到', '菜鸟']          2
['知到', '微信']          2
['小红书', '微信']        2
['菜鸟', '微信']          2
['淘宝', 'B站']           3
['淘宝', '微信']          3
['B站', '微信']           4
['知乎', 'B站']           2
['知乎', '微信']          2
['知到', '菜鸟', '微信']          2
['淘宝', '微信', 'B站']           3
['知乎', 'B站', '微信']           2
```

```
Strong association rule
X                       Y               conf
['菜鸟']          ['知到']        1.00
['知到']          ['菜鸟']        1.00
['知到']          ['微信']        1.00
['小红书']        ['微信']        1.00
['菜鸟']          ['微信']        1.00
['淘宝']          ['B站']         1.00
['淘宝']          ['微信']        1.00
['B站']           ['微信']        1.00
['知乎']          ['B站']         1.00
['知乎']          ['微信']        1.00
['菜鸟', '微信']          ['知到']          1.00
['知到', '菜鸟']          ['微信']          1.00
['知到', '微信']          ['菜鸟']          1.00
['菜鸟']          ['知到', '微信']          1.00
['知到']          ['菜鸟', '微信']          1.00
['淘宝', 'B站']           ['微信']          1.00
['淘宝', '微信']          ['B站']           1.00
['淘宝']          ['B站', '微信']           1.00
['知乎', 'B站']           ['微信']          1.00
['知乎', '微信']          ['B站']           1.00
['知乎']          ['B站', '微信']           1.00
['微信']          ['B站']         0.80
['B站']           ['淘宝']        0.75
['B站', '微信']           ['淘宝']          0.75
['B站']           ['淘宝', '微信']          0.75
['微信']          ['淘宝']        0.60
['微信']          ['淘宝', 'B站']           0.60
PS C:\Users\11298>
```

男生：

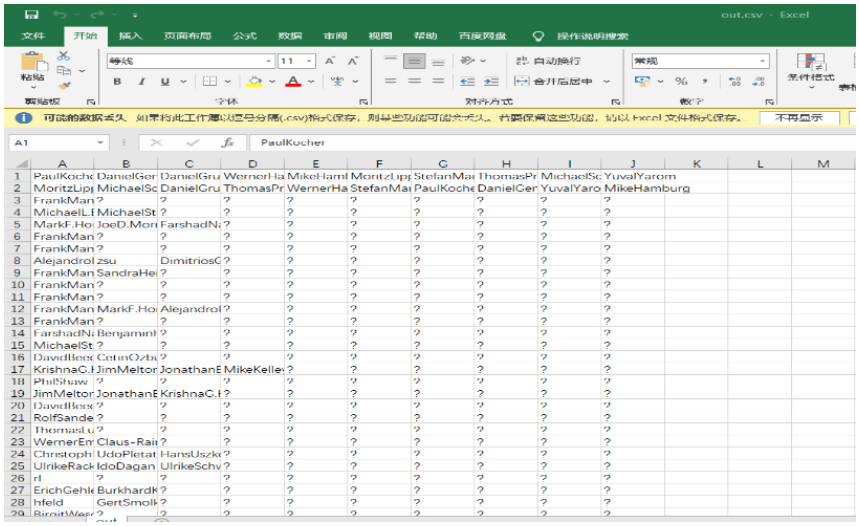
```
PS C:\Users\11298> & C:/Users/11298/AppData/Local/Programs/Python/Python310/python.exe c:/Users/11298/Desktop/Apriori.py
频繁项集        支持度计数
['QQ']   5
['微信']        5
['知到']        2
['B站']         4
['QQ', '知到']   2
['微信', '知到']        2
['微信', 'QQ']   5
['QQ', 'B站']    4
['微信', 'B站']         4
['微信', 'QQ', '知到']   2
['微信', 'QQ', 'B站']    4

Strong association rule
X                       Y               conf
['知到']        ['QQ']          1.00
['知到']        ['微信']        1.00
['微信']        ['QQ']          1.00
['QQ']   ['微信']       1.00
['B站']         ['QQ']          1.00
['B站']         ['微信']        1.00
['微信', '知到']        ['QQ']          1.00
['QQ', '知到']   ['微信']        1.00
['知到']        ['微信', 'QQ']          1.00
['微信', 'B站']         ['QQ']          1.00
['QQ', 'B站']    ['微信']        1.00
['B站']         ['微信', 'QQ']          1.00
['QQ']   ['B站']         0.80
['微信']        ['B站']         0.80
['微信', 'QQ']   ['B站']         0.80
['微信']        ['QQ', 'B站']   0.80
['QQ']   ['微信', 'B站']         0.80
```

总体而言，女生频繁项集更多，支持度计数更多，关联性更强，这表明女生会更趋近于用同一种类 app 且 app 使用种类比男生多。微信、QQ、B 站是男女生使用度均很高的 app，而小红书、淘宝、菜鸟，是女生使用较多、男生少使用的 app。

任务 2

使用 DBLP 数据集，提出一种方法，挖掘密切相关的（即经常一起合写文章）合著者关系。实验使用的数据条数，作者数量，根据自己的内存环境选取并加以说明即可。

1） 下载 dblpjson-csv.zip 文件，解压后先后运行 dblpxml-json.py， dblpjson-csv.py 文件，得到 out.csv 文件，存储了合著者信息。



2） 运行 weka，Explorer 中点击 open file，文件类型改为.csv，选择 out.csv，而后保存为.arff 形式，获得 out.arff

```
F: > out.arff
1   @relation out
2
3   @attribute PaulKocher {MoritzLipp,FrankManola,MichaelL.Brodie,MarkF.Hornick,AlejandroP.Buchmann,FarshadNayeri,MichaelStonebraker,DavidBeech,KrishnaG.Kulkarni,PhilShaw,
4   @attribute DanielGenkin {MichaelSchwarz0001,MichaelStonebraker,JoeD.Morrison,zsu,SandraHeiler,MarkF.Hornick,BenjaminHurwitz,CetinOzbutun,JimMelton,JonathanBauer,Claus-
5   @attribute DanielGruss {DanielGruss,FarshadNayeri,DimitriosGeorgakopoulos,AlejandroP.Buchmann,JonathanBauer,KrishnaG.Kulkarni,HansUszkoreit,UlrikeSchwall,MichaelLey,Er
6   @attribute WernerHaas0004 {ThomasPrescher0002,MikeKelley,AlbertMaier,mper,AngelikaStorrer,IrvingL.Traiger,Chin-LiangChang,PetraSteffens,Claus-RainerRollinger,BruceG.Li
7   @attribute MikeHamburg {WernerHaas0004,ErichGehlen,NickRoussopoulos,RudiStuder,PeterH.Schmitt,rgH.Siekmann,BerndPage,AnnaSlobodov,KlausW.Wagner,AlinDeutsch,JohnLions,r
8   @attribute MoritzLipp {StefanMangard,BirgitWesche,RudiStuder,UlrikeWeiland,ChristianStangier,MaryF.Fernandez,RitaLey,BreannaWorthington-Eyre,DionisisD.Kehagias,goras,N
9   @attribute StefanMangard {PaulKocher,rl,D.Eberle,DanielaFlorescu,AndreasRock,AndreaLamont,MiranMosmondor,nez,ThiloErnst,AndreasHein,HideakiNii,JasonE.Shoemaker,FotisLi
10  @attribute ThomasPrescher0002 {DanielGenkin,nthner,AlonY.Levy,MichelePla-Mobarak,ReinerWichert,a,rnKiselev,ToyotaMorimoto,FrancisJ.DoyleIII,AthinaGrammatikopoulou,Cath
11  @attribute MichaelSchwarz0001 {YuvalYarom,ChristopherHabel,DanSuciu,PeterWolf,CarmenPastor,GerdKock,HisayoS.Momose,KosmasDimitropoulos,UrsulaMartin,PerryL.Miller,s,Jul
12  @attribute YuvalYarom {MikeHamburg,RobinCover,DanielS.Coming,MirceaNicolescu,ShuvraS.Bhattacharyya,BabitaMalik,on,DavidB.Resnik,HuaHe,MarkR.Geier,TrevorDavis0002,Xiong
13
14  @data
15  MoritzLipp,MichaelSchwarz0001,DanielGruss,ThomasPrescher0002,WernerHaas0004,StefanMangard,PaulKocher,DanielGenkin,YuvalYarom,MikeHamburg
16  FrankManola,?,?,?,?,?,?,?,?,?
17  MichaelL.Brodie,MichaelStonebraker,?,?,?,?,?,?,?,?
18  MarkF.Hornick,JoeD.Morrison,FarshadNayeri,?,?,?,?,?,?,?
19  FrankManola,?,?,?,?,?,?,?,?,?
20  FrankManola,?,?,?,?,?,?,?,?,?
21  AlejandroP.Buchmann,zsu,DimitriosGeorgakopoulos,?,?,?,?,?,?,?
22  FrankManola,SandraHeiler,?,?,?,?,?,?,?,?
23  FrankManola,?,?,?,?,?,?,?,?,?
24  FrankManola,?,?,?,?,?,?,?,?,?
25  FrankManola,MarkF.Hornick,AlejandroP.Buchmann,?,?,?,?,?,?,?
```

3） 打开 out.arff，在“Associate”（关联规则分析）中，点击 choose，选择 Apriori，点击旁边框中的内容，修改参数

N:规则数

N:规则数

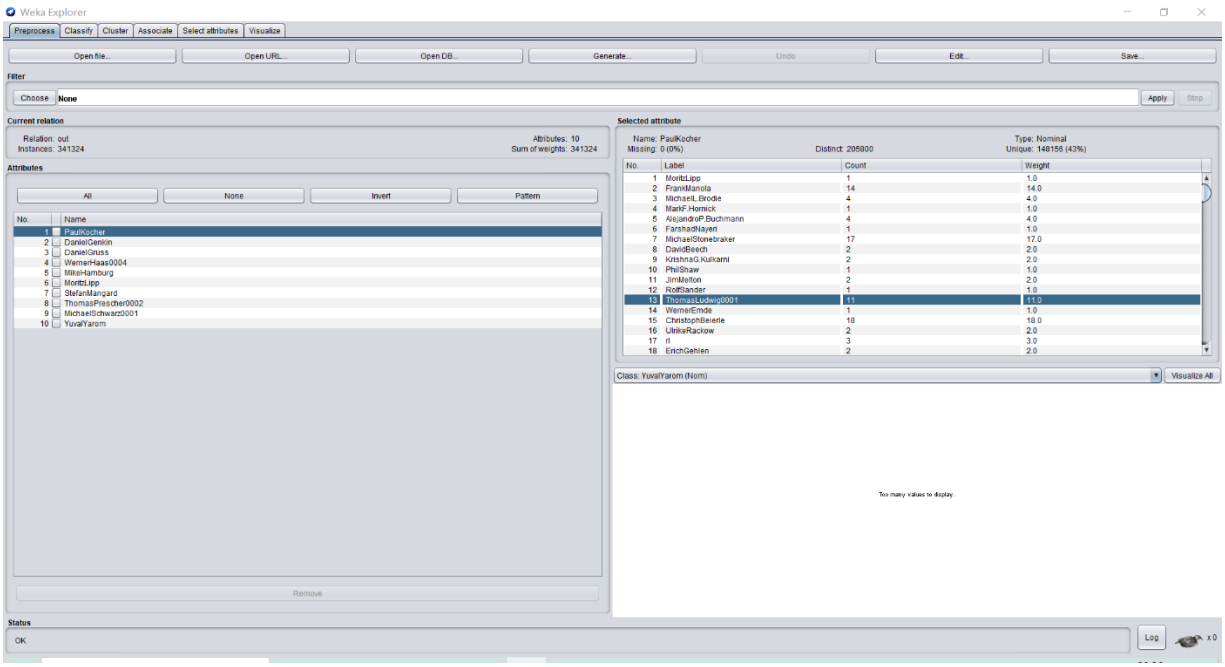
T:度量单位的选择

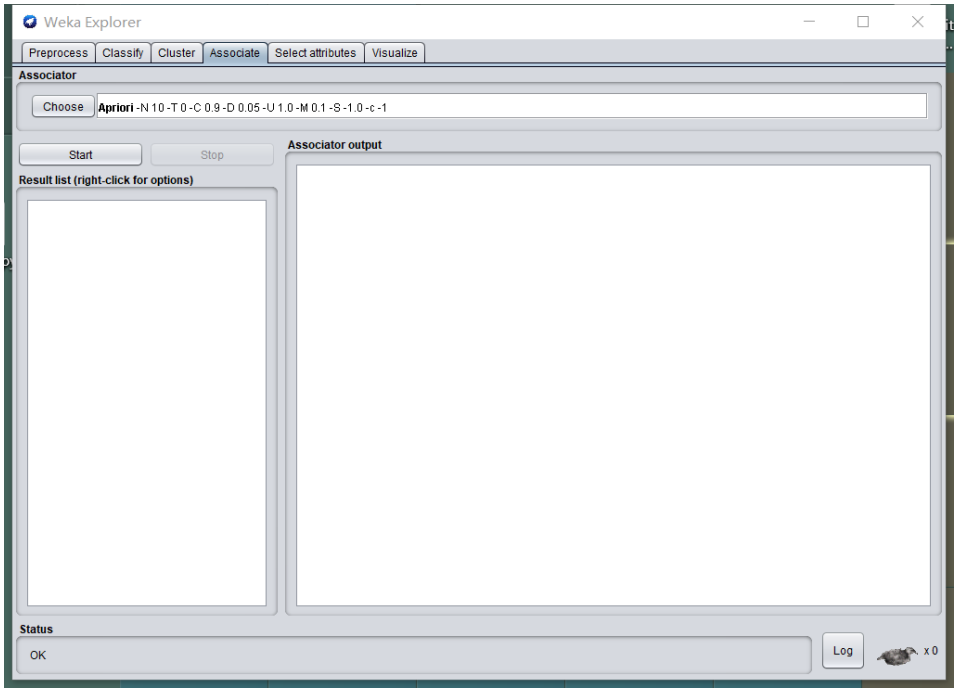
D:递减迭代值

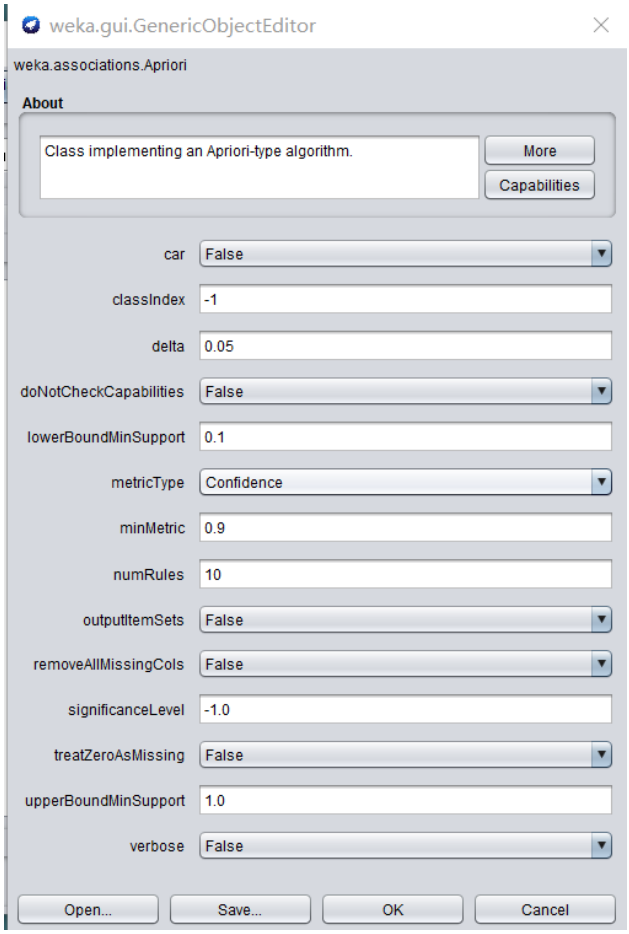
U:最小支持度上界 M:最小支持度下届

s:重要程度

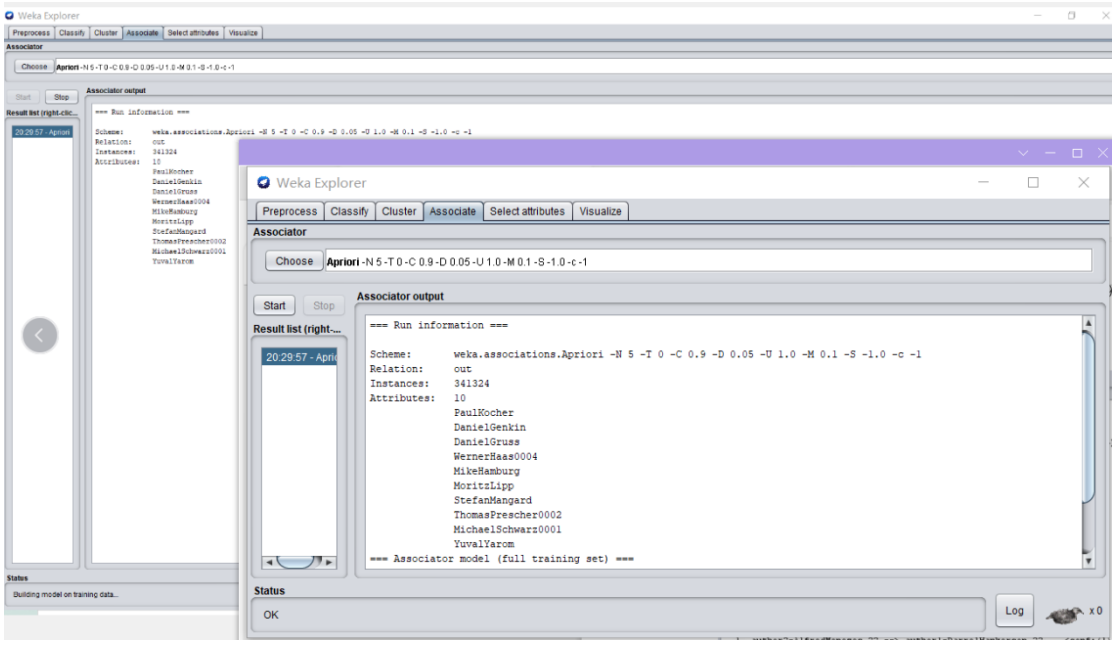
c:类索引为 c 输出项集设为真







4） Start 开始，得出结论



1. author2=AlfredMenezes 22 ==> author1=DarrelHankerson 22 <conf:(1)> lift:(1363.59) lev:(0) [21] conv:(21.98)
2. author1=DarrelHankerson 22 ==> author2=AlfredMenezes 22 <conf:(1)> lift:(1363.59) lev:(0) [21] conv:(21.98)
3. author1=PhilippeBonnet 17 ==> author2=DennisE.Shasha 17 <conf:(1)> lift:(1666.61) lev:(0) [16] conv:(16.99)
4. author1=XinJin0001 16 ==> author2=JiaweiHan0001 16 <conf:(1)> lift:(1249.96) lev:(0) [15] conv:(15.99)
5. author1=JonasMellin 15 ==> author2=MikaelBerndtsson 15 <conf:(1)> lift:(1428.52) lev:(0) [14] conv:(14.99)
6. author2=HenryLin 14 ==> author1=XiaoboZhou 14 <conf:(1)> lift:(2142.79) lev:(0) [13] conv:(13.99)
7. author1=XiaoboZhou 14 ==> author2=HenryLin 14 <conf:(1)> lift:(2142.79) lev:(0) [13] conv:(13.99)
8. author2=JonasMellin 13 ==> author1=MikaelBerndtsson 13 <conf:(1)> lift:(2307.62) lev:(0) [12] conv:(12.99)
9. author1=MikaelBerndtsson 13 ==> author2=JonasMellin 13 <conf:(1)> lift:(2307.62) lev:(0) [12] conv:(12.99)
10. author2=YiZhang0001 12 ==> author1=EthanZhang 12 <conf:(1)> lift:(2499.92) lev:(0) [11] conv:(12)
11. author1=EthanZhang 12 ==> author2=YiZhang0001 12 <conf:(1)> lift:(2499.92) lev:(0) [11] conv:(12)
12. author1=AlexanderKaplan 11 ==> author2=RainerTichatschke 11 <conf:(1)> lift:(2307.62) lev:(0) [10] conv:(11)
13. author2=VictorKlee 10 ==> author1=PeterGritzmann 10 <conf:(1)> lift:(2499.92) lev:(0) [9] conv:(10)
14. author2=MarkS.Drew 10 ==> author1=RajeevRamanath 10 <conf:(1)> lift:(2999.9) lev:(0) [9] conv:(10)
15. author1=RajeevRamanath 10 ==> author2=MarkS.Drew 10 <conf:(1)> lift:(2999.9) lev:(0) [9] conv:(10)
16. author1=NikosHardavellas 10 ==> author2=IppokratisPandis 10 <conf:(1)> lift:(2307.62) lev:(0) [9] conv:(10)
17. author2=JamesB.D.Joshi 10 ==> author1=YueZhang0002 10 <conf:(1)> lift:(2999.9) lev:(0) [9] conv:(10)
18. author1=YueZhang0002 10 ==> author2=JamesB.D.Joshi 10 <conf:(1)> lift:(2999.9) lev:(0) [9] conv:(10)
19. author2=EricC.Jensen 10 ==> author1=StevenM.Beitzel 10 <conf:(1)> lift:(2999.9) lev:(0) [9] conv:(10)
20. author1=StevenM.Beitzel 10 ==> author2=EricC.Jensen 10 <conf:(1)> lift:(2999.9) lev:(0) [9] conv:(10)