数据挖掘作业二

关联规则挖掘

姓名：CHANTHAMATH DETHSOMSAY 学号：3820181029

一、 问题描述

本次作业中，将对 Wine Reviews 数据集进关联规则挖掘。

1. 对数据集进行处理，转换成适合关联规则挖掘的形式
2. 找出频繁项集
3. 导出关联规则，计算其支持度和置信度
4. 对规则进行评价

二、 关联规则挖掘过程

1. 对数据进行处理

获取 Wine Review 数据集的所有属性信息。分析数据集 Unamed 属性是索引，description 是属性描述，因此没有必要可以删除

```
# 删除Unnamed: 0和description
winedata = winedata.copy().drop(columns=['Unnamed: 0','description'])
winedata.info()
np.array(winedata.head())
```

在数据集中可以发现 points 和 price 是连续型数据，因此需要对这两个连续特征作离散化处理

```
# 处理points
bin = [0,75,80,85,90,95,100]
winedata['points'] = pd.cut(winedata['points'],bin)
winedata['points'] = winedata['points'].astype('str')
# 处理price
bin = [0,20,30,40,50,60,2100]
winedata['price'] = pd.cut(winedata['price'],bin)
winedata['price'] = winedata['price'].astype('str')
```

2. 找出频繁项集

采用 Aprior 算法构建频繁项集。Aprior 算法首先会生成所有单个属性名、属性值的项集列表，然后扫描全部数据集来查看哪些项集满足最小支持度要求，其中不满足最小支持度的集合会被去掉，对剩下的集合进行组合

```
def aprioriGen( Lk, k ):
    retList = []
    lenLk = len( Lk )
    for i in range( lenLk ):
        for j in range( i + 1, lenLk ):
            L1 = list( Lk[ i ] )[ : k - 2 ];
            L2 = list( Lk[ j ] )[ : k - 2 ];
            L1.sort();L2.sort()
            if L1==L2:
                retList.append( Lk[ i ] | Lk[ j ] )
    return retList

def apriori( dataSet, minSupport = 0.5 ):

    C1 = createC1( dataSet )    # 构建初始候选项集C1
    D =list( map( set, dataSet )) # 将dataSet集合化
    L1, suppData = scanD( D, C1, minSupport )     # 构建初始的频繁项集，即所有项集只有一个元素
    L = [ L1 ]
    k = 2
    while ( len( L[ k - 2 ] ) > 0 ):
        Ck = aprioriGen( L[ k - 2 ], k )
        Lk, supK = scanD( D, Ck, minSupport )
        suppData.update( supK )
        L.append( Lk )
        k += 1
    return L, suppData
```

3. 结果

频繁项集输出结果：

```
Out[22]: [[frozenset({'(80, 85]'}),
          frozenset({'Chardonnay'}),
          frozenset({'Jim Gordon'}),
          frozenset({'@gordone_cellars'}),
          frozenset({'Russian River Valley'}),
          frozenset({'Syrah'}),
          frozenset({'(90, 95]'}),
          frozenset({'Red Blend'}),
          frozenset({'(50, 60]'}),
          frozenset({'(30, 40]'}),
          frozenset({'Washington'}),
          frozenset({'Sean P. Sullivan'}),
          frozenset({'Columbia Valley (WA)'}),
          frozenset({'Columbia Valley'}),
          frozenset({'@wawinereport'}),
          frozenset({'(40, 50]'}),
          frozenset({'Sonoma'}),
          frozenset({'Matt Kettmann'}),
          frozenset({'Central Coast'}),
```

关联规则输出结果：

```
Out[24]: [[frozenset({'California'}),
           frozenset({'@gordone_cellars', 'California'}),
           1.0],
          [frozenset({'California'}), frozenset({'California', 'Jim Gordon'}), 1.0],
          [frozenset({'Jim Gordon'}),
           frozenset({'@gordone_cellars', 'Jim Gordon'}),
           1.0],
          [frozenset({'@gordone_cellars'}),
           frozenset({'@gordone_cellars', 'Jim Gordon'}),
           1.0],
          [frozenset({'@vboone'}),
           frozenset({'@vboone', 'Russian River Valley'}),
           0.9761721752498077],
          [frozenset({'California'}),
           frozenset({'California', 'Russian River Valley'}),
           1.0],
          [frozenset({'Virginie Boone'}),
           frozenset({'Russian River Valley', 'Virginie Boone'}),
           0.9761721752498077],
```