作业二分析过程报告（对于Building\_Permits.csv数据集进行处理）

要求

1. 对数据集进行处理，转换成适合关联规则挖掘的形式

答：要对数据集进行关联规则挖掘首先对数据进行预处理。数据预处理主要包括数据清理、数据集成和数据规约等。

1. 数据规约

数据规约主要是在保证数据集完整性的前提下，去掉数据集中不重要的属性，以提高数据挖掘的效率。

去除依据：去掉可以代替的属性，去除冗余的属性，去除缺失值过半的属性。

保留的属性为：

|  |  |
| --- | --- |
| Column name | 属性 |
| Permit Type | 标称属性 |
| Current Status | 标称属性 |
| Number of Existing Stories | 数值属性 |
| Number of Proposed Stories | 数值属性 |
| Estimated Cost | 数值属性 |
| Revised Cost | 数值属性 |
| Plansets | 数值属性 |
| Existing Construction Type | 标称属性 |
| Proposed Construction Type | 标称属性 |
| Supervisor District | 标称属性 |
| Zipcode | 标称属性 |

（2）数据集成

由于关联规则挖掘的提出主要指针对二值属性的，而本实验的数据集中包括很多标称属性和数值属性。因此要对部分标称属性进行聚合。对于数值属性要分组，分成相对较少的组后，按照二值属性的关联规则挖掘方法进行挖掘。（#表示原始集中的属性值）

（i1）将Permit Type属性值转换成PT+#；

#Permit Type

for i1 in range(0,lenth):

string = 'PT'+str(int(data\_1[i1]))

permit\_type\_list.append(string)

（i2）Current Status属性值保持不变；

#Current Status

current\_status\_list = list(data\_2)

（i3）Number of Existing Stories数值属性，[0,10]标记为NOES\_L，[11,30]标记为NOES\_M，[31,78]标记为NOES\_H；

#Number of Existing Stories

for i3 in range(0,lenth):

if isnan(data\_3[i3]):

tag3 = ""

else:

if 0<=data\_3[i3]<=10:

tag3 = 'NOES\_L'

elif 11<=data\_3[i3]<=30:

tag3 = 'NOES\_M'

elif 31<=data\_3[i3]<=78:

tag3 = 'NOES\_H'

Number\_of\_Existing\_Stories\_list.append(tag3)

（i4）Number of Proposed Stories数值属性，[0,10]标记为NOPS\_L，[11,30]标记为NOPS\_M，[31,78]标记为NOPS\_H；

#Number of Proposed Stories

for i4 in range(0,lenth):

if isnan(data\_4[i4]):

tag4 = ""

else:

if 0<=data\_4[i4]<=10:

tag4 = 'NOPS\_L'

elif 11<=data\_4[i4]<=30:

tag4 = 'NOPS\_M'

elif 31<=data\_4[i4]<=78:

tag4 = 'NOPS\_H'

Number\_of\_Proposed\_Stories\_list.append(tag4)

（i5）Estimated Cost数值属性，[0,100000]标记为EC\_L，(100000,1000000]标记为EC\_M，(1000000, 537958646]标记为EC\_H；

#Estimated\_Cost\_list

for i5 in range(0,lenth):

if isnan(data\_5[i5]):

tag5 = ""

else:

if 0<=data\_5[i5]<=100000:

tag5 = 'EC\_L'

elif 100000<data\_5[i5]<=1000000:

tag5 = 'EC\_M'

elif 1000000<data\_5[i5]<=537958646:

tag5 = 'EC\_H'

Estimated\_Cost\_list.append(tag5)

（i6）Revised Cost数值属性，[0,100000]标记为RC\_L，(100000,1000000]标记为RC\_M，(1000000, 537958646]标记为RC\_H；

#Revised\_Cost\_list

for i6 in range(0,lenth):

if isnan(data\_6[i6]):

tag6 = ""

else:

if 0<=data\_6[i6]<=100000:

tag6 = 'RC\_L'

elif 100000<data\_6[i6]<=1000000:

tag6 = 'RC\_M'

elif 1000000<=data\_6[i6]<=780500000:

tag6 = 'RC\_H'

Revised\_Cost\_list.append(tag6)

（i7）Plansets标称属性，属性值转换为P+#；

#Plansets

for i11 in range(0,lenth):

if isnan(data\_11[i11]):

string11 = ""

else:

string11 = 'P'+str(int(data\_11[i11]))

Plansets\_list.append(string11)

（i8）Existing Construction Type标称属性，将属性值转换为ECT+#

#Existing Construction Type

for i12 in range(0,lenth):

if isnan(data\_12[i12]):

string12 = ""

else:

string12 = 'ECT'+str(int(data\_12[i12]))

Existing\_Construction\_Type\_list.append(string12)

（i9）Proposed Construction Type标称属性，将属性值转换为PCT+#

#Proposed Construction Type

for i13 in range(0,lenth):

if isnan(data\_13[i13]):

string13 = ""

else:

string13 = 'PCT'+str(int(data\_13[i13]))

Proposed\_Construction\_Type\_list.append(string13)

（i10）Supervisor District标称属性，将属性值转换成SD+#

#Supervisor District

for i14 in range(0,lenth):

if isnan(data\_14[i14]):

string14 = ""

else:

string14 = 'SD'+str(int(data\_14[i14]))

Supervisor\_District\_list.append(string14)

（i11）Zipcode标称属性，保持原值

数据集处理代码如下：

for i15 in range(0,lenth):

if isnan(data\_15[i15]):

string15 = ""

else:

string15 = str(int(data\_15[i15]))

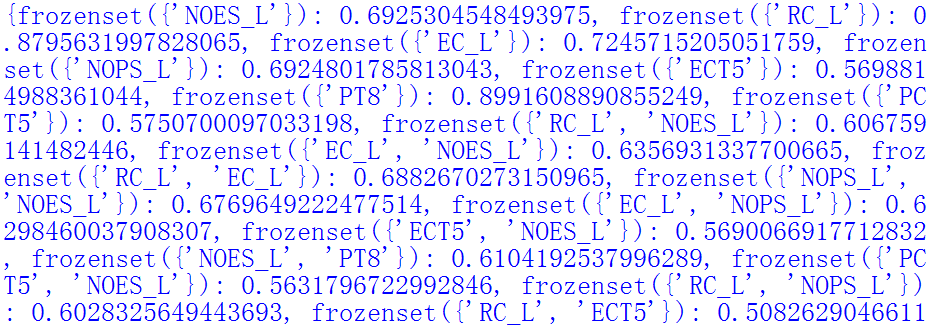
Zipcode\_list.append(string15)

最后将处理过的数据存放在一个新的文件test2.csv中,以便关联规则挖掘使用。

1. 找出频繁项集

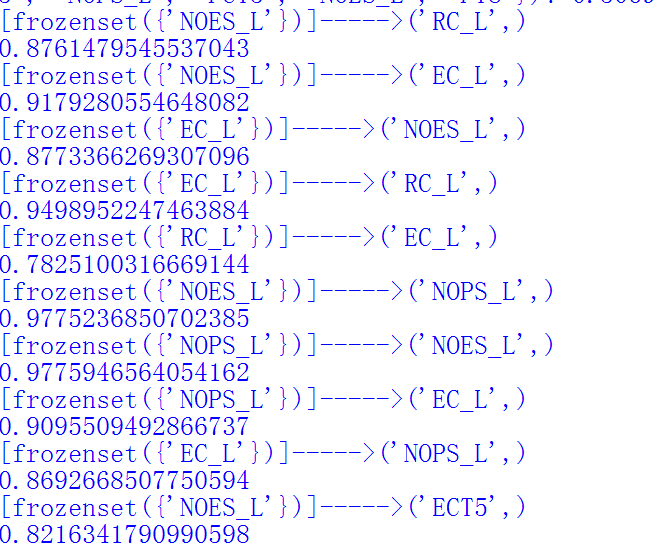
读取在第一步中处理好的文件，然后使用apriori算法找出频繁项集，设置的最小支持度为0.5，则频繁项集结果如下所示：

符合最小支持度的频繁项集为部分结果截图：



1. 导出关联规则，计算其支持度和置信度

部分结果如图所示



1. 对规则进行评价，可以使用Lift，也可以使用教材所提及的其他指标