

**数据挖掘**

**互评作业二**

**题　　目： 频繁模式与关联规则挖掘**

**学 院： 计算机学院**

**专业名称： 计算机科学与技术**

**学 号： 3120191033**

**姓 名： 彭成**

**任课教师： 汤世平老师**

2020年5月17日星期日

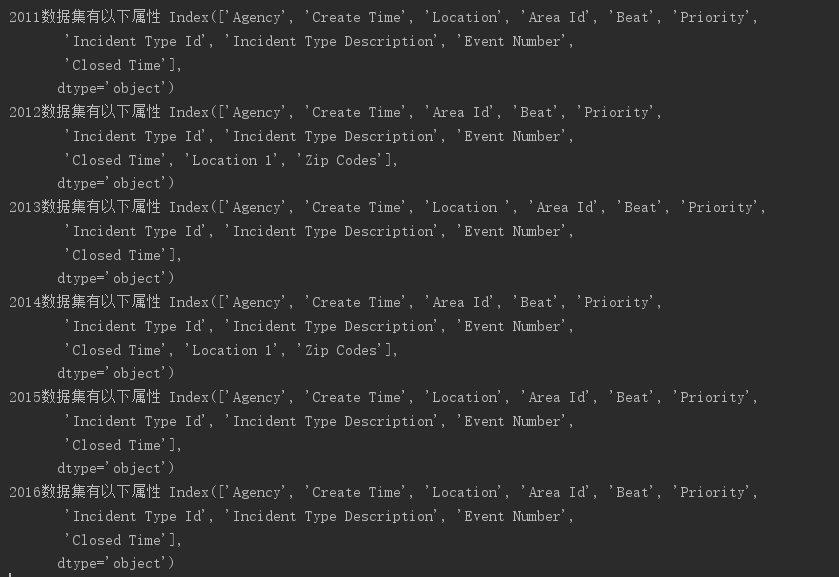
# 数据集说明

## 1.1 数据集选择

数据集名称：oakland-crime-statistics-2011-to-2016

## 详细说明以及处理

在这个数据集中，一共包含6个数据子集，分别为2011-2016年度的奥克兰犯罪情况。其详细属性如下：



通过观察数据集可得知，这六年的数据属性基本一样，值得进行分析与预处理的有如下几个属性：Agency, Location, Area id, Beat, Incident Type id，Incident Type Descripe, Event Number，其中2012年和2014年的属性为Location 1，经过特殊处理变为Location。由于Incident Type id与Incident Type Descripe一一对应，我们只对Incident Type id进行分析。Event Number对应每一行数据，不具备重复性，不对其进行分析。

发现有部分数据存在缺失值的情况。使用上次预处理的方法，舍去有缺失值的行后，由原来的1046388条数据变为剩下859898条数据，在此基础上进行实验。

# 找出频繁模式

## 2.1 使用算法及简单介绍

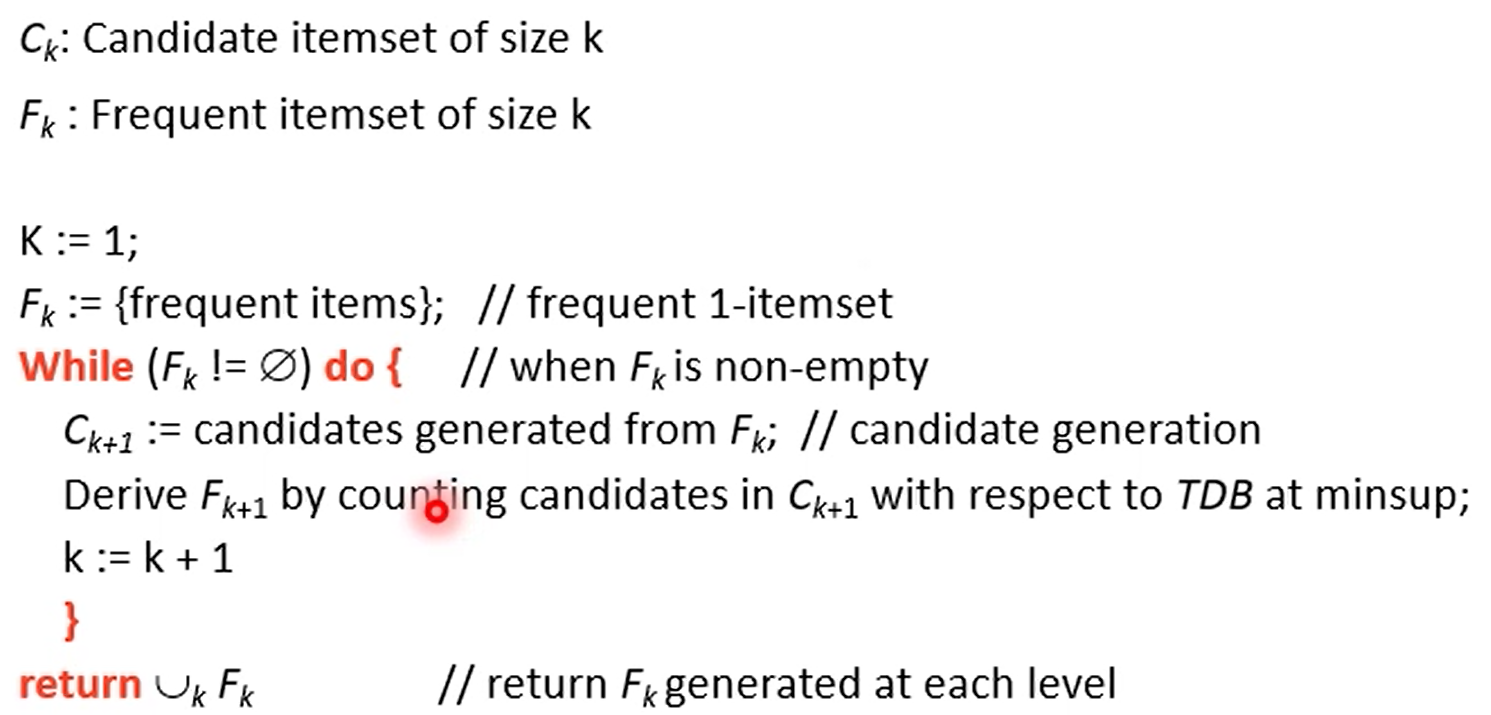
算法名称：Apriori算法

算法介绍：Apriori算法是第一个关联规则挖掘算法，也是最经典的算法。它利用逐层搜索的迭代方法找出数据库中项集的关系，以形成规则，其过程由连接（类矩阵运算）与剪枝（去掉那些没必要的中间结果）组成。（该部分引用自百度百科）。

在本实验中，使用Apriori算法来构建频繁项集。在本实验中，我们约定支持度的阈值为10%，置信度的阈值为50%。

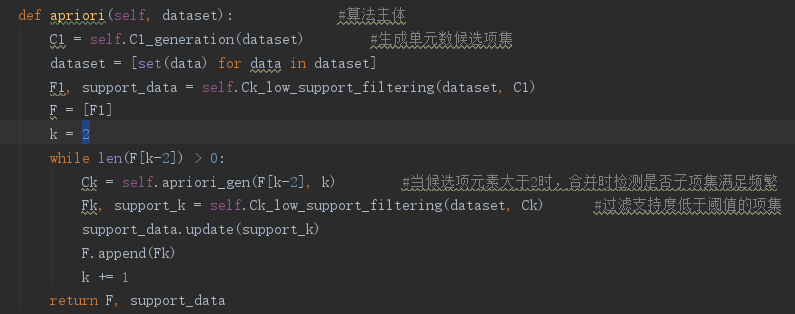


算法的主要流程如课件中所示（如下图）：

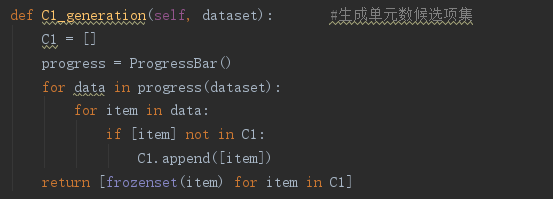


相应代码为：

Apriori主函数：



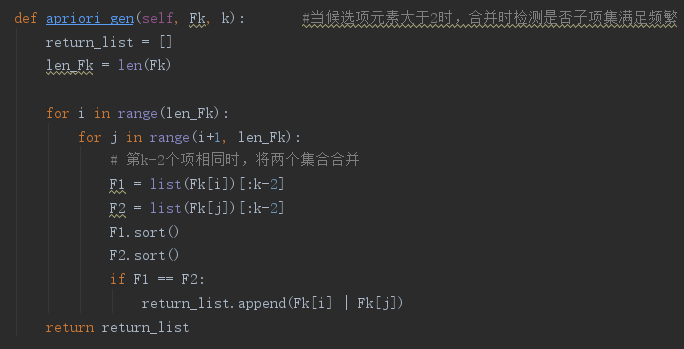
单元素候选集生成函数：



过滤低支持度函数：



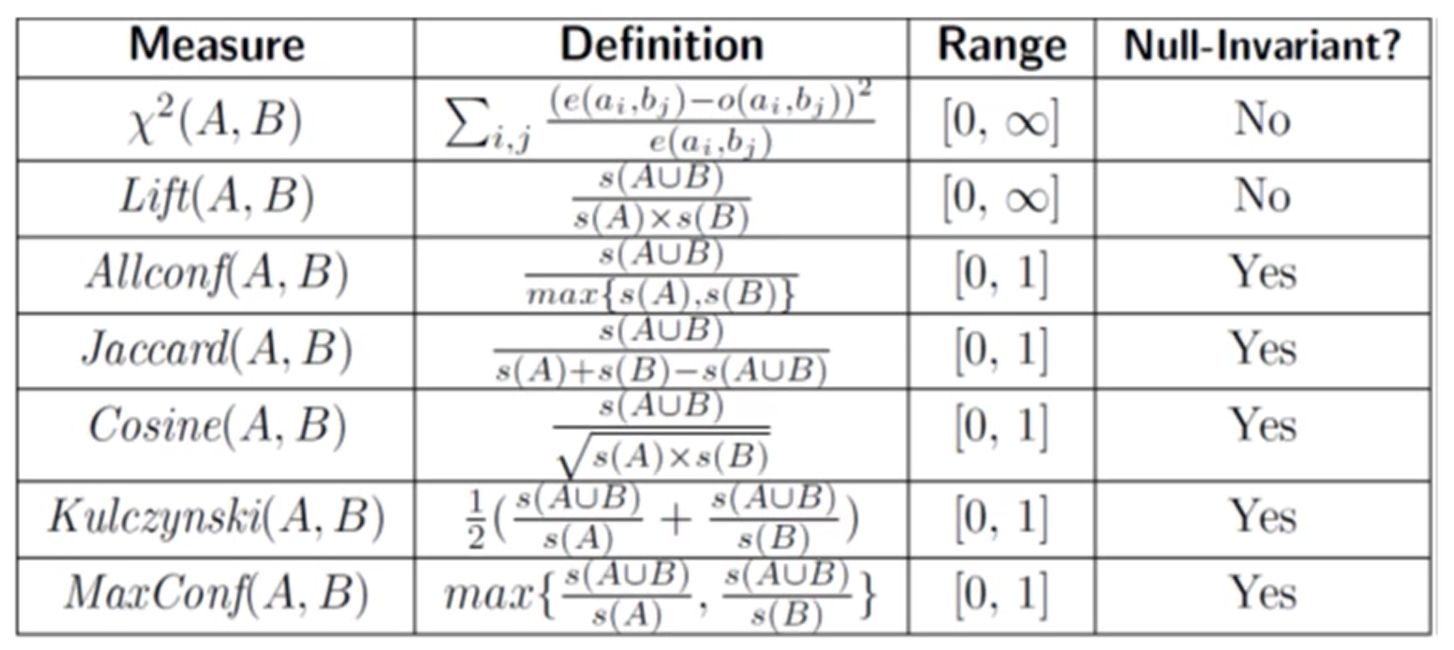
合并筛选函数：



最终产生的频繁项集结果保存在“频繁项集.json”文件中，效果如下图所示：

# 导出关联规则（包含评价）

基于2中使用Apriori算法得出的频繁项集，我们计算关联规则以及使用评价指标来评价它们。本实验使用的是课件中的Lift和Jaccard两种指标进行评价。



其中计算的公式为：

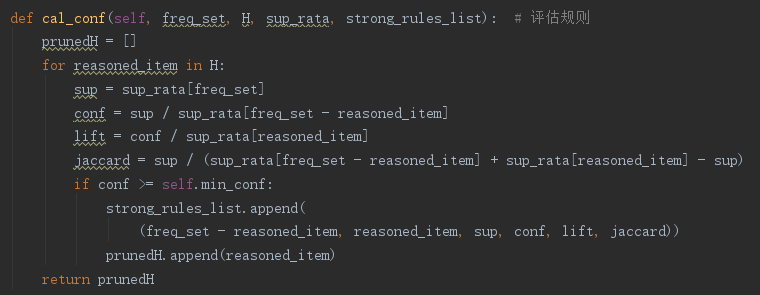
支持度：

置信度：

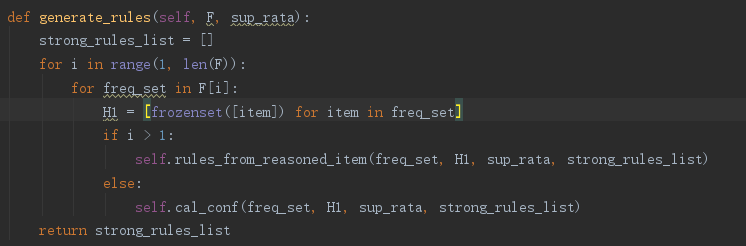
Lift:

Jaccard:

以上计算对应的代码为：

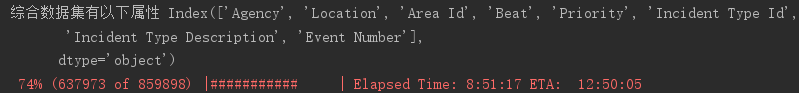


计算并保留达到置信度阈值的函数为：



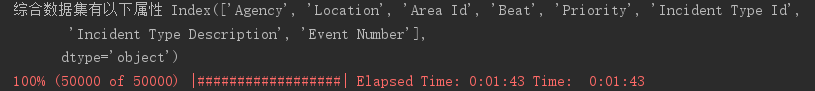
# 分析挖掘结果

由于电脑问题，跑完全部数据花费的时间实在是太长了。由下图可见，原本预计的时间是5小时能跑完，但跑了一晚上之后发现还需要13小时。于是我放弃跑完全部数据，取前50000个数据进行实验。

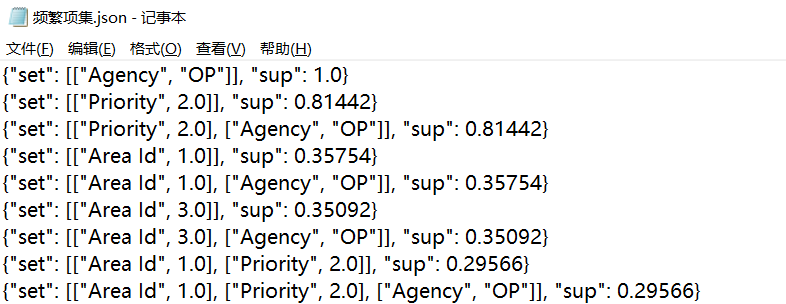




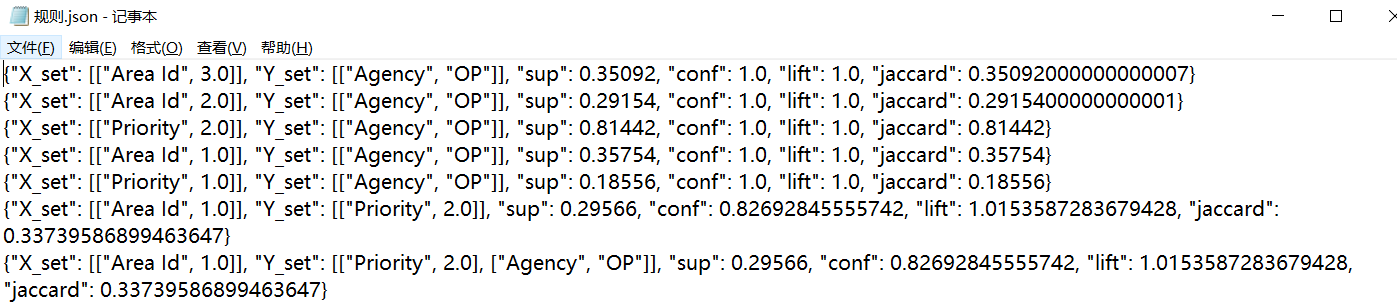
更换后的时间如下所示，只需几分钟：



我们将得到的频繁项集放到了./results/频繁项集.json文件中，按照支持度由大到小排列，形式如下图所示：



将得到的关联规则以及评价结果放到了./results/规则.json文件中，按照置信度由大到小排列，形式如下图所示：



由于所有的Agency属性的值都是OP，所以对其分析没有实际意义，我们跳过包含Agency属性的频繁项集与规则进行分析。

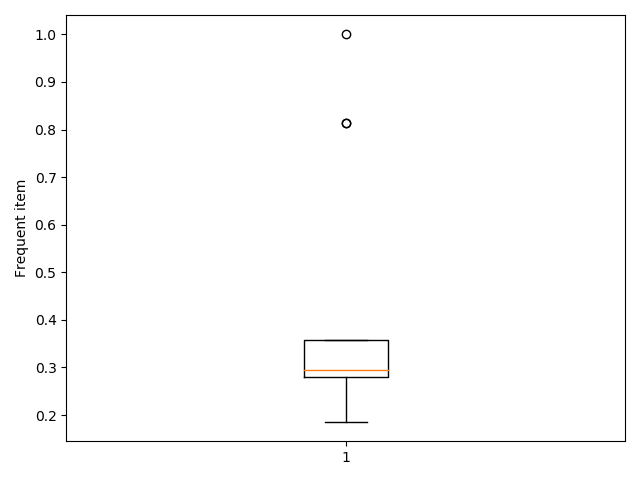
我们可以由频繁项集.json得知，Area Id为1.0时支持度最高，也就是说在该地区的犯罪事实出现最多。而且Area Id和Priority的关联度较高。

我们可以由规则.json得知，["Area Id", 1.0]与["Priority", 2.0]的置信度较高，这说明犯罪的严重性与所在地有着较强联系。

# 可视化

分别使用盒图与散点图对频繁项集与规则进行可视化。

对频繁项集使用盒图可视化可得：



对规则的指出度与置信度使用散点图可视化可得：

