实验3：基于WEKA的关联规则算法性能比较

**1 实验目的**

1. 掌握WEKA平台挖掘关联规则的方法。

**2 实验平台与工具**

1. Windows、Linux操作系统
2. 数据分析与挖掘系统WEKA

**3 实验内容**

1. 基于WEKA平台比较Apriori和FP-Growth两种关联规则算法性能
   1. 随数据规模变化后两种算法性能变化比较与分析；
   2. 相同数据规模，不同支持数（或者支持度）变化后两种算法性能变化比较与分析；
   3. 采用曲线图展示不同算法的计算效率，对于曲线走势进行描述，并且分析实验结果。

**4. 规则与要求**

（1）独立完成，严禁相互抄袭（如有发现抄袭和被抄袭均判为0分），以及从网络上直接摘抄别人的观点和总结（该行为将影响报告成绩）。

（2）实验报告符合学术写作的排版要求，请参考群文件中的“报告模板.docx”和“参考文献格式.docx”的排版格式。

（3）实验报告内容详实，采用图文混合的方式叙述安装和配置过程。

Tip：Win+Shift+S 在Windows中可以快速截屏。

（4）报告文件见附件，提交报告时请以附件形式插入到超星作业中。

实验报告

报告标题：基于WEKA的关联规则算法性能比较

学号：21190630

姓名：黄艺杰

日期：2023年4月25日

# 一、实验环境

1. 操作系统：Window10

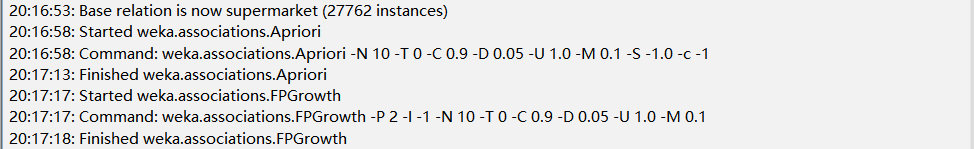
2. 软件（含版本号）：weka-3-8-6-azul-zulu-windows

# 二、实验内容及其完成情况

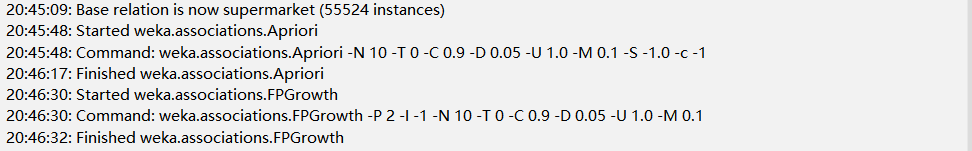
基于WEKA平台比较Apriori和FP-Growth两种关联规则算法性能

1. 随数据规模变化后两种算法性能变化比较与分析

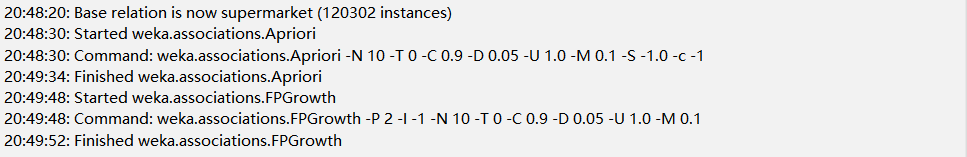
当数据集为27762条数据时，Apriori算法所消耗的时间是15秒，FP-Growth算法消耗时间是1秒



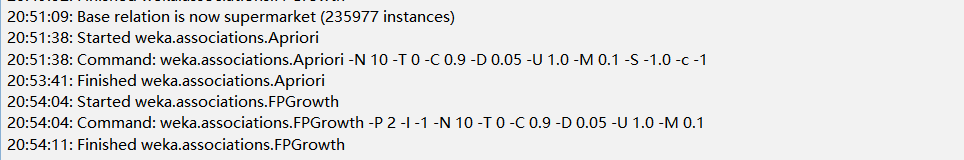
当数据集为55524条数据时，Apriori算法所消耗的时间是29秒，FP-Growth算法消耗时间是2秒



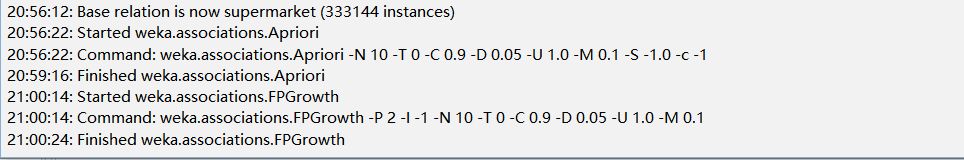
当数据集为120302条数据时，Apriori算法所消耗的时间是64秒，FP-Growth算法消耗时间是4秒



当数据集为235977条数据时，Apriori算法所消耗的时间是123秒，FP-Growth算法消耗时间是7秒

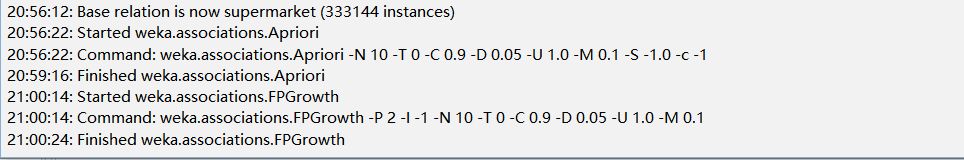


当数据集为333144条数据时，Apriori算法所消耗的时间是174秒，FP-Growth算法消耗时间是10秒

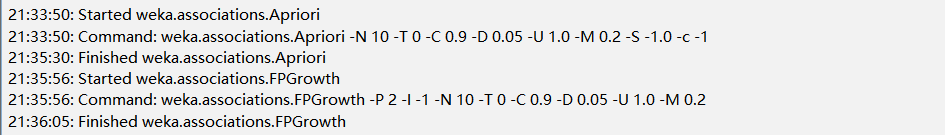


1. 相同数据规模，不同支持数（或者支持度）变化后两种算法性能变化比较与分析

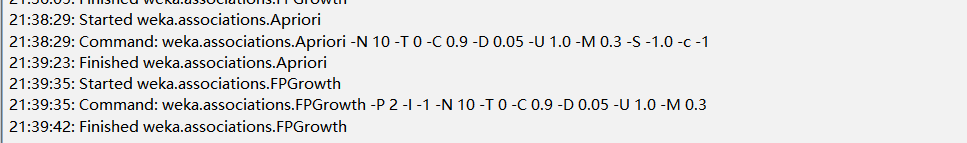
选用数据集为333144条数据，将最小支持度设为0.1-1.0时，Apriori算法所消耗的时间是174秒，FP-Growth算法消耗时间是10秒



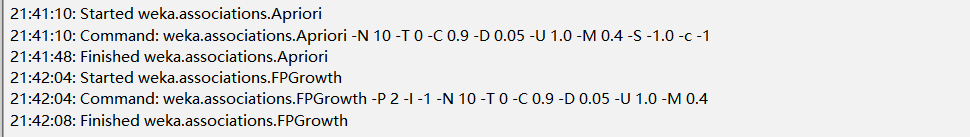
当最小支持度为0.2-1.0时，Apriori算法所消耗的时间是100秒，FP-Growth算法消耗时间是9秒



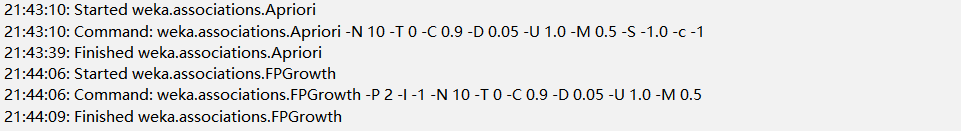
当最小支持度为0.3-1.0时，Apriori算法所消耗的时间是54秒，FP-Growth算法消耗时间是7秒



当最小支持度为0.4-1.0时，Apriori算法所消耗的时间是38秒，FP-Growth算法消耗时间是4秒



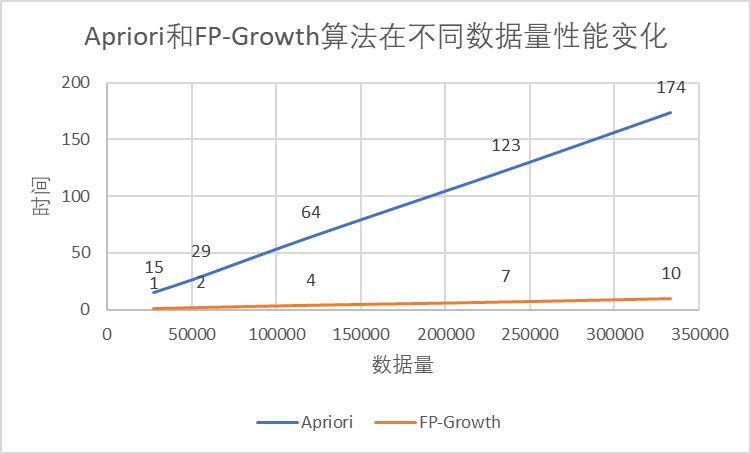
当最小支持度为0.5-1.0时，Apriori算法所消耗的时间是29秒，FP-Growth算法消耗时间是3秒



1. 采用曲线图展示不同算法的计算效率，对于曲线走势进行描述，并且分析实验结果。

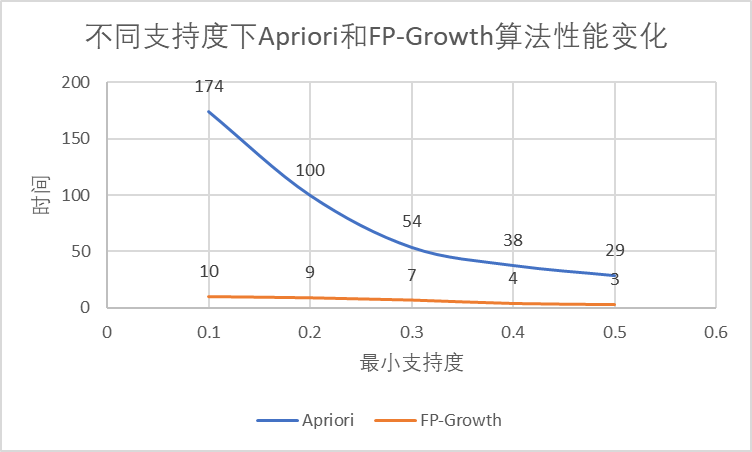
随着数据集数据的量的增加，Apriori算法的运行时间和FP-Growth算法的运行时间都逐渐变长，可以看出数据量和消耗时间是成正相关，几乎可以认为是成正比例的。同时，对于这个数据集而言，FP-Growth算法在时间消耗上要比Apriori要少得多，而且对于不同的数据量，它们增加的速度（斜率）也是Apriori更快。这表明Apriori算法在处理大规模数据时可能会面临一定的性能问题，而FP-Growth算法更适合处理大规模数据。

FP-Growth算法只需要对数据集进行两次扫描，而Apriori算法需要对每个潜在的频繁项集进行扫描，因此FP-Growth算法的速度更快。



在数据量333144条数据时，随着最小支持度的增加，Apriori算法的运行时间和FP-Growth算法的运行时间都逐渐变短，Apriori算法的曲线较陡峭，FP-Growth算法的曲线较平缓，支持度的变化对于Apriori算法的影响更大（斜率），支持度增大可以帮助剪枝从而降低生成的频繁项集数量，从而减少了挖掘算法所需的计算量和内存需求，因此可以提高挖掘算法的运行速度。

在支持度较低的情况下，FP-Growth算法需要构建一棵FP树，而这个过程需要消耗大量的时间和空间。但是，当支持度较高时，由于FP-Growth算法只需要对数据库进行两次遍历，因此它的性能优于Apriori算法 。



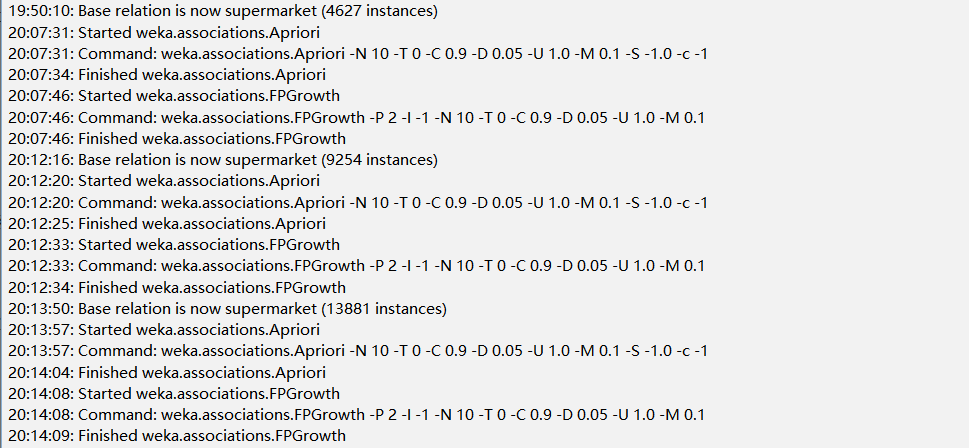
# 三、实验总结

（可以总结实验中出现的问题以及解决的思路，也可以列出没有解决的问题）

问题1：

1. 问题描述

从最开始的时候，我使用的数据量都小于10万条，发现在FP-Growth算法下消耗时间几乎都在1秒以内，得不到增长趋势



1. 问题分析（可能的原因、难点、挑战）

10万条数据虽然对于Apriori算法有足够的区分度，但对于FP-Growth算法还是太少，无法得到其随着数据量增大的变化趋势

1. 解决方案

通过增加数据量得到更明显的时间消耗的变化