# aii分析报告

1. 情景分析

本次作业要求分别以trade.csv和trade\_new.csv作为交易数据库，从中按照要求整理出所有的用户购买记录作为数据集，并且分别对购买记录中的商品编号pluno、商品类型编号dptno、品牌编号bndno做频繁集挖掘，将用户个人购买记录整合，可以充分利用到用户的个人喜好。

1. 代码说明

fpGrowth.py文件是将FP-Growth算法集成的一个模块文件，将所有的过程都封装到了函数fpGrouth()中，只需传入数据字典、最小阀值两个参数即可，返回是当前阀值下的频繁集。

aii\_pluno.py、aii\_dptno.py、aii\_bndno.py分别是对三种item进行频繁集挖掘的代码，主要完成数据的整理和调用FP算法。

考虑到这是对购物记录做频繁项集的挖掘，我只将含有两个以上item的项集作为输出，只含有一个item的项集并没有什么意义。

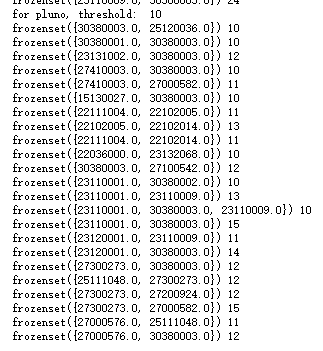
1. 运行结果

由于程序的输出比较长，无法一次截取，所以分别取得到的结果的一部分进行分析：

* 1. Pluno商品

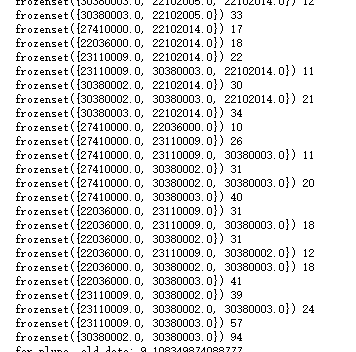
作业要求分别取阀值为2，4，6，8，10，这是由于将用户的数据整合后，数据条目变少了许多，而且如果两件商品的阀值超过10，说明10个以上的人同时买了这两件商品，即可以认为这两件商品是有关联的。所以：

对于旧数据trade，有：



由于结果非常多，这里只截取了一小部分结果，这很有可能是因为每个用户所有购买记录中，所包含的商品数目足够多，而且可以看到每个频繁项集中，所包含的商品只有两三种。

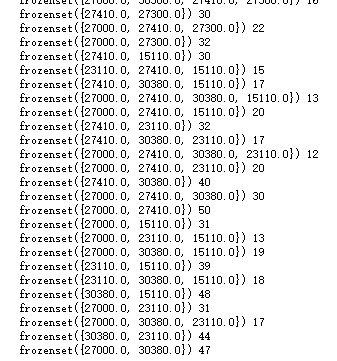
对于新数据trade\_new，有：



可以看出新数据要比旧数据的结果要好，出现了次数超过90的频繁集，可以把这些商品认定为具有较大的潜在关联性，可以用这几条数据做相似性推荐。

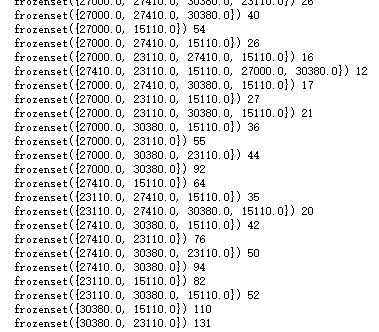
* 1. Dptno商品类型

旧数据：



这是取自阀值为10的时候部分结果，由于商品类型的范围要比具体的商品更广，用它来做频繁集挖掘效果确实比pluno的效果好很多，可以看到有的出现了40次以上。

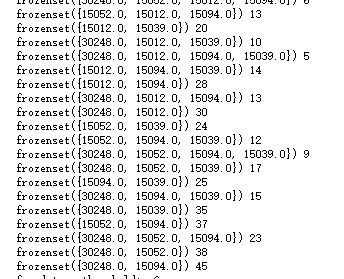
对于新数据：



新数据的效果确实比旧数据要好，可能是由于该超市在原来的基础上对商品的位置摆放，关联度分析上做了改进之后所得到的结果。数据中出现了同时购买次数131次的，说明这两个商品类别的商品具有较强的相似度，它的效果和pluno相比更好一点。

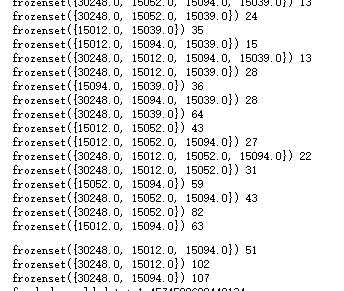
* 1. Bndno品牌：

旧数据：



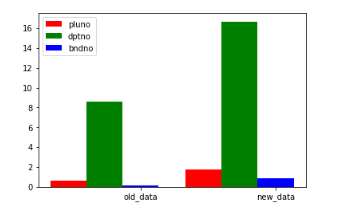
用商品的品牌做挖掘得到的结果和直接用dptno做挖掘的效果差不多，可见对于旧数据而言，dptno和bndno效果一样比较好，而pluno的效果非常差。

新数据：



对于新数据而言，用bndno做挖掘和pluno的结果相似，但稍微比dptno的效果差一点，可见对于新数据而言，无论采用哪种方式，效果都很显著。

* 1. 性能对比



由于print会花掉大量时间，所以统计的时候已经去掉这部分的时间。

由图中可以看出，对于旧数据：dptno和bndno的效果虽然都挺不错，但是dptno所用掉的时间远远高于其余两个，所以使用bndno做挖掘会更好；对于新数据，pluno、dptno和bndno的效果都非常不错，但是明显bndno更快。