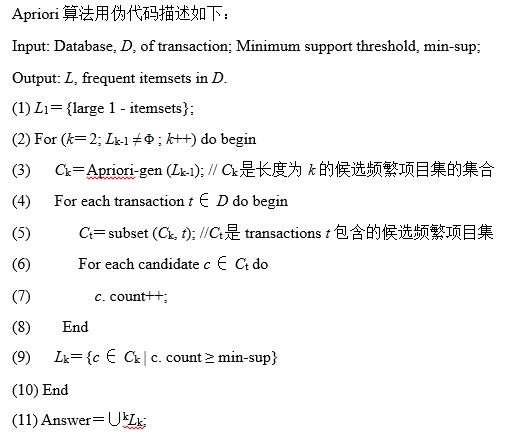
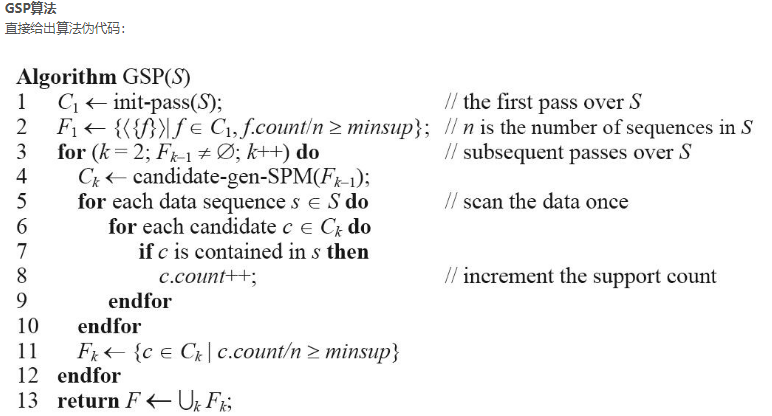
目的：对于现烤烘焙类产品进行门店各天的商品选品推荐，组合推荐

算法选择：序列挖掘算法 （关联规则Aprior）

模型思路及代码API：



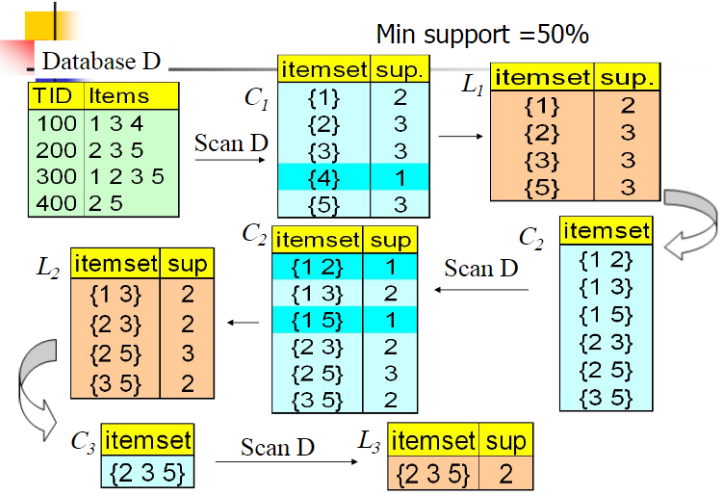
模型思路类似GSP算法：



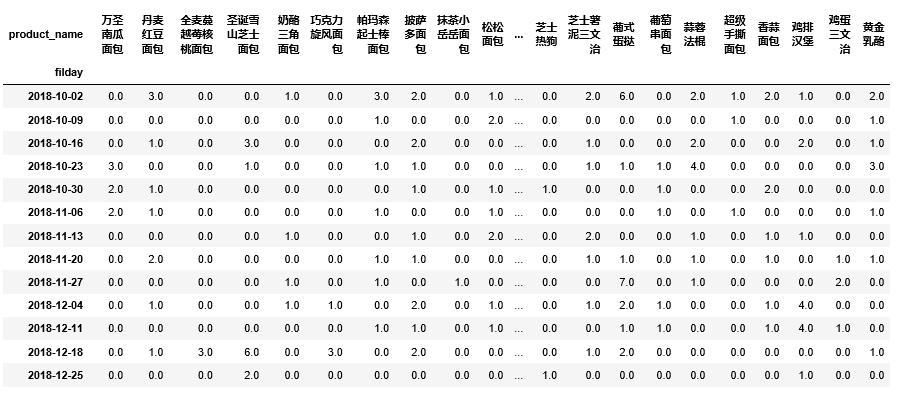
首先按照星期来划分数据集，周一至周日共7条线，或者按照工作日与非工作日来区分数据集属性。若使用一年的数据，那每个星期的数据源可以有365/7 =52条（考虑近期时间越准确的因素）。

对于现烤类产品，时间点的预测把商品组合把控度较严格，所以按照全天的售卖时间线来做各点可用商品组合选取。

基于Aprior算法上最改进及更符合实际应用需求，GSP算法是Aprior算法的一种，相较于Aprior算法可以有效减少需扫描的候选序列，减少无用模式的产生。原始Aprior算法是全遍历形式，耗时很高吃内存。Aprior算法简单的运行遍历图如下：



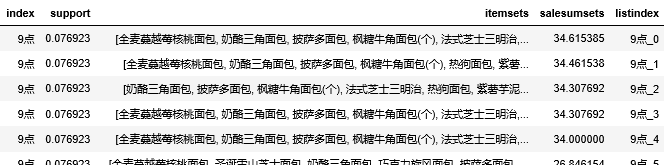
算法输入表分两部分，引入了销量和及销售频次两个指标来进行选品筛选。如下销售表，若销售数量>0则定义为有销售 1，否则为0。



商品组合的SKU数在算法中可理解为SKU序列个数为k时对应的support值情况，该序列产品集最易覆盖销售需求面。Support频次概率指标数是基于商品集最易覆盖销售面的考量；而salesum销售和指标；门店实际业务设立k值范围15-20sku数，仅打印算法该部分商品组合结果做后续时间线接替。

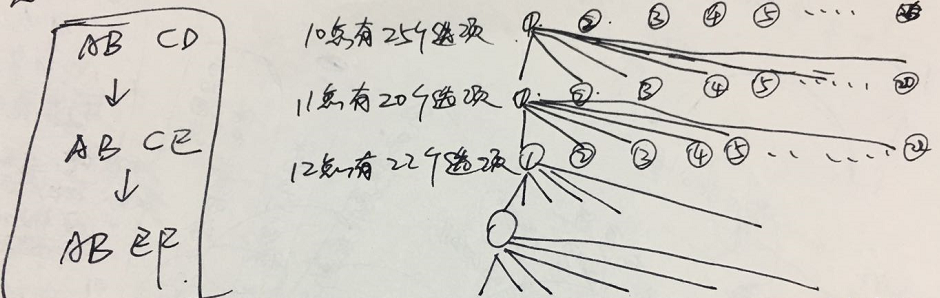
API 函数下面。

第一部分： 结果构建每个时间点下最易售卖及销量和较高的产品组合集：



第二部分：确定时间序列下的商品组合最优接替选项（采用全匹配）：

确定接替选项选取指标值：p1（组合销量和） p2（组合销售频次） p3（组合间接替性）



假设时间点A的组合选项为A1：[a,b,c,d,e] A2:[a,b,d,f] A3:[c,d,e,f]

时间点B的组合选项为B1：[a,b,e,f] B2：[a,c,d,e,f,g] B3:[b,c,d]

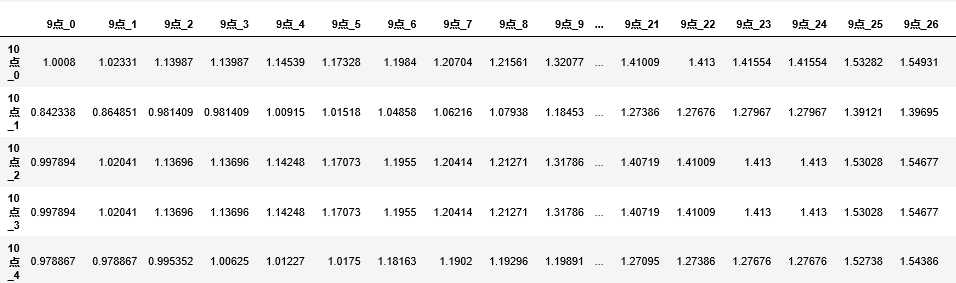
时间点C……

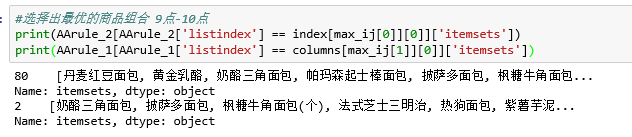
即指标值表现为：

|  |  |  |  |
| --- | --- | --- | --- |
|  | B1 | B2 | B3 |
| A1 |  |  | … |
| A2 |  | … | … |
| A3 | … | … | … |

（p3为商品组合交集个数\*2/组合并集总数 代表商品组合间的接替性）

 各自进行标准化后加和作为综合指标值。





按综合指标值来确定最优时间线上的商品组合接替排序。还可以引入现烤产品的烤制时间，烤制批数等因素考虑。

思路总结：规则算法对于业务细节的抠点和关键点需要紧密斜街和设立相应规则指标来符合现实业务情况，业务跟进及理解并用于模型构建要求度高；规则的按规设立表现在算法运行层面上会导致更改复杂及运行时间复杂度高，很多规则需要全条件匹配思考。目前并没有对运行时间上进行优化，对于整理算法的主要是基于业务转化规则的构思，而对于传统机器理论并没有较多涉及。