算法 1：Apriori 关联规则

输入：事务集

输出：强关联规则

k←1；

所有的候选 k 项集←{项目|扫描事务集中每个事务，找到不重复的项目}；

// 支持度大于最小支持度的 k 项集就是频繁 k 项集

计算所有候选 k 项集中候选 k 项集的支持度，得到频繁 k 项集；

while 有频繁 k 项集 **do**

    // （1）组合后元素个数为 k+1（2）重复的组合只保留一个

    由所有频繁 k 项集两两组合得到候选 k+1 项集；

    利用定理对候选 k+1 项集剪枝，删掉所有不可能是频繁的项集

    计算候选 k+1 项集中所有 k+1 项集的支持度，得到所有频繁 k+1 项集；

    if 没有频繁 k+1 项集为空 **then**

        break；

    end if

    所有频繁 k 项集←所有频繁 k+1 项集；

    k←k+1;

end while

依次获取频繁 k 项集的所有真子集，生成关联规则

计算所有关联规则的置信度，返回所有强关联规则

制表位把竖线做出来：https://zhuanlan.zhihu.com/p/660791698

算法 1：Kmeans 聚类分析

输入：各个样本的坐标 x^i，簇的数量 k

输出：聚好的各个类

flag $\leftarrow$ True // 检测聚类中心是否发生变化

随机选择 k 个点的坐标作为初始聚类中心

while flag **do**

  计算每个样本到各个聚类中心的距离，将其分配到最近聚类中心所在簇；

  忽略聚类中心，重新计算各个簇的聚类中心；

  if 聚类中心没有发生改变 **then**

    flag $\leftarrow$ False

  end if

end while

return 聚好的各个类