用在下面函数的后面

Df:下面函数得到的频繁项集

metric:可选值['support','confidence','lift','leverage',' conviction']。

> metric = 'support' => 取值范围 [0,1] metric = 'confidence' => 取值范围 [0,1] metric = 'lift' => 取值范围 [0, inf]

min_threshold:最小值域,对应上面的metric

support_only: 默认是 False。仅计算有支持度的项集, 若缺失支持度则用 NaNs 填充。

rules.sort_values(by="lift",ascending=False)最后用这个函数筛选排序

Df: dataframe数据,两列,一列作为index(用df.set_index指定) ,一列是onehot化的商品集(可以用df.get_dummies对一列数据一次性onehot)

Min_support: 最小支持度

use_colnames:true表示返回的集合用物品名称,false为 用编号

max_len: 最大物品组合

itemsets.sort_values(by="support",ascending=False) 最后用这个函数筛选和排序 mlxtend.frequent_patterns.association_ rules

apriori

mlxtend.frequent_patterns.apriori

Apriori算法是一种挖掘关联规则的频繁项集算法

目的是找出事物之间存在的隐藏的关系

两部: 求频繁项集, 根据关联规则筛选排序

支持度support: support(A=>B) = P(A∪B),表示A和B 同时出现的概率

置信度confidence: confidence(A=>B)=support(AUB) / support(A),表示A和B同时出现的概率占A出现概率的比值。

最小提升度lift: lift(A=>B)=support(AUB) / (support(A)*support(B)),该值大于1才有意义,一般取3。反应物品A出现对物品B出现的影响

频繁项集:满足最小支持度阈值的集合,可能是1项,可 能是多项。

关联规则基于transaction,推荐的基础是最近一次的点击/购买,不需要考虑用户一定时期内的偏好

最小支持度差别较大,可能在0.01到0.5之间,需要实验

注意

基本部分

总概

最小置信度可能是0.5-1