Annoy ([Approximate Nearest Neighbors](http://en.wikipedia.org/wiki/Nearest_neighbor_search#Approximate_nearest_neighbor) )

协同过滤算法。这个算法是目前最受欢迎的算法之一，广泛被用在推荐系统领域。

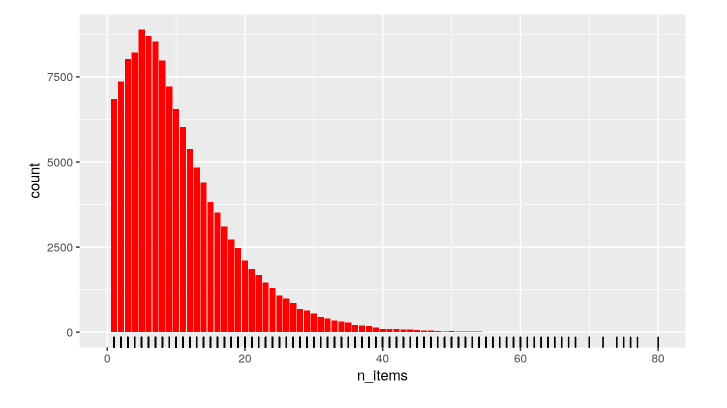


Prior推测结果，train label ,test 无label

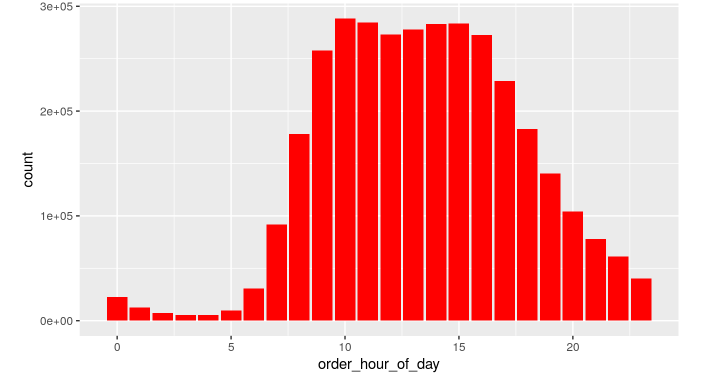
预测：

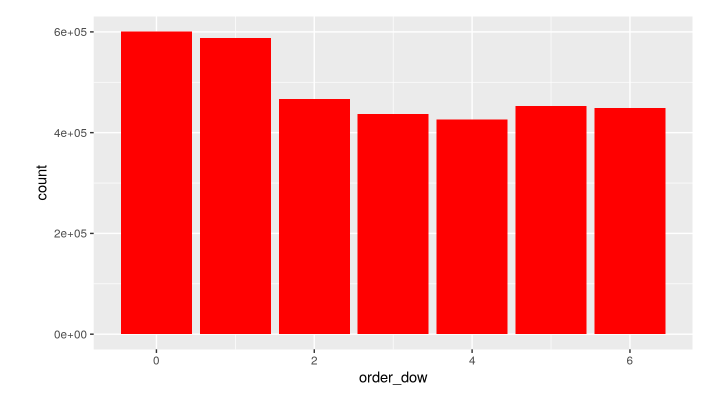
输入order id,预测product ids

1.一个用户一次买多少商品

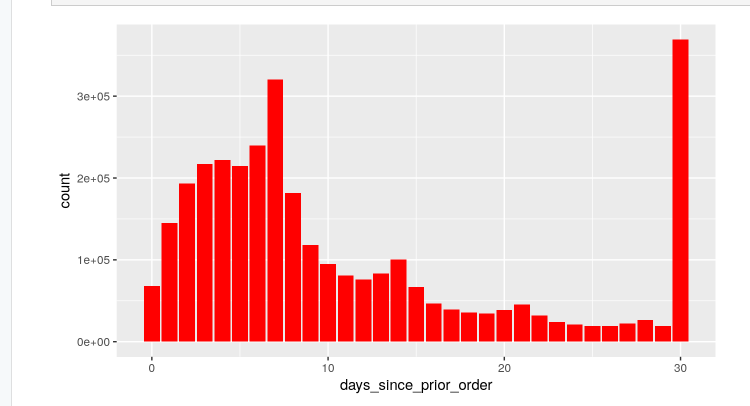


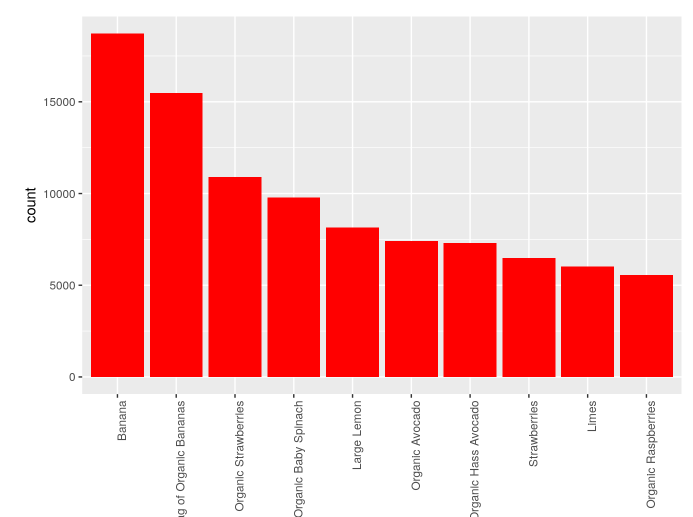
2.购物时间点，小时和周



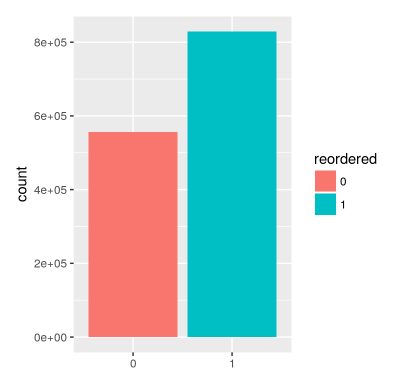


3.reorder的时间距上次

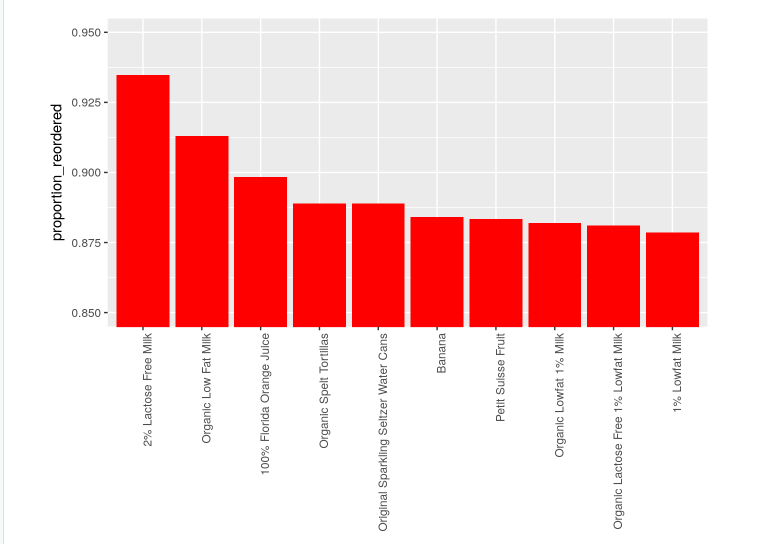


4.哪种商品火爆

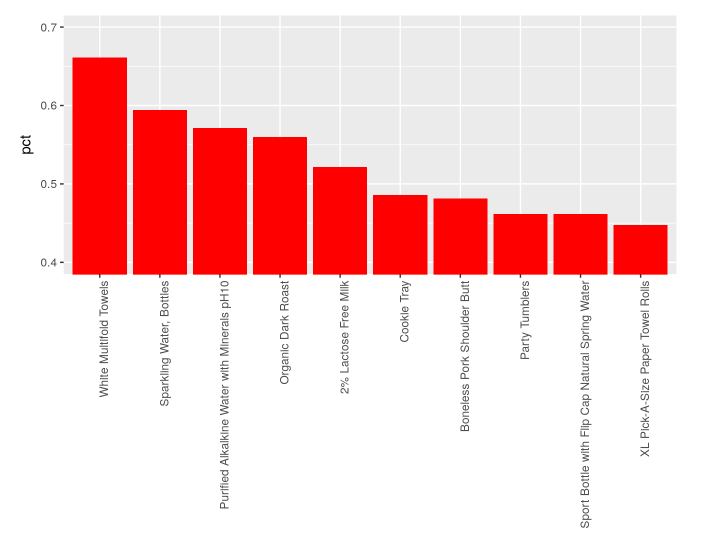
5.同一个商品的重复购买order



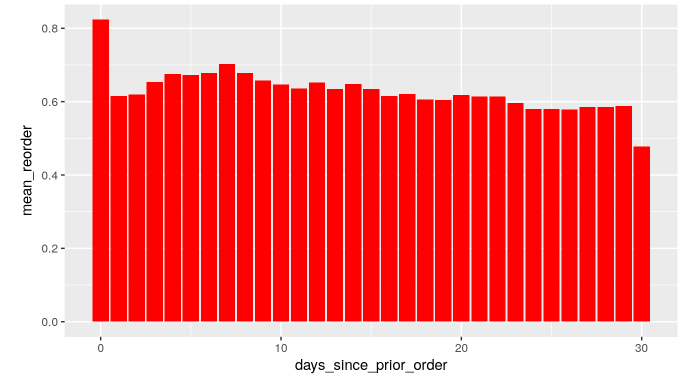
6.哪种商品最容易reorder



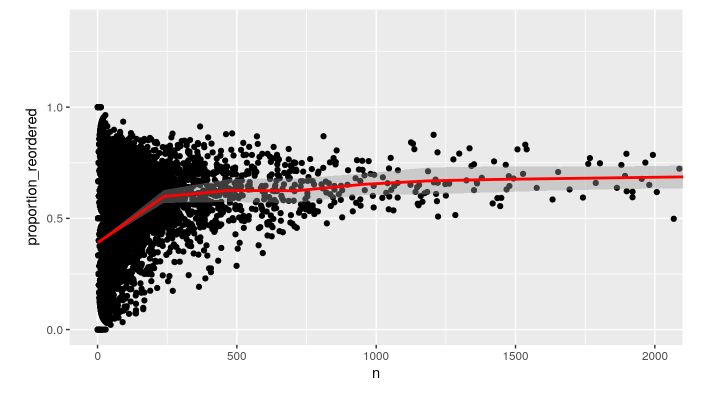
7哪种商品首先放入购物篮



8 距上一次购物时间 与 再次订购的关系



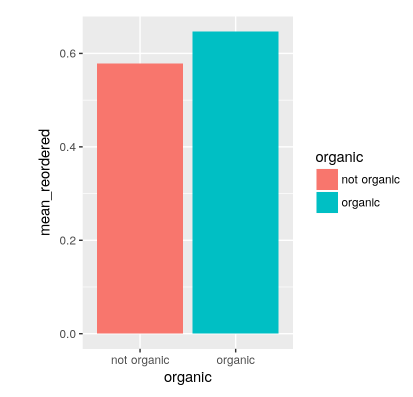
9商品购买次数 与 再次被订购 关系



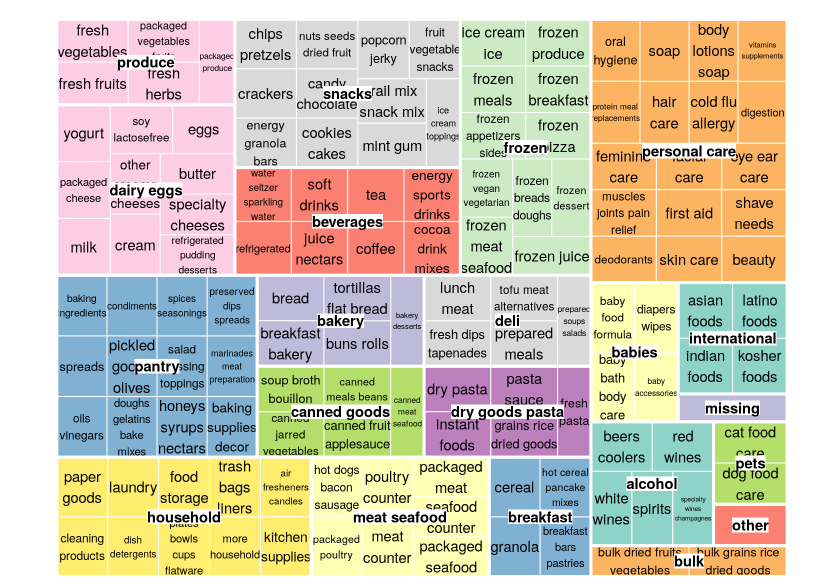
10.有机食品与无机食品比例



11有机与无机重新被购买



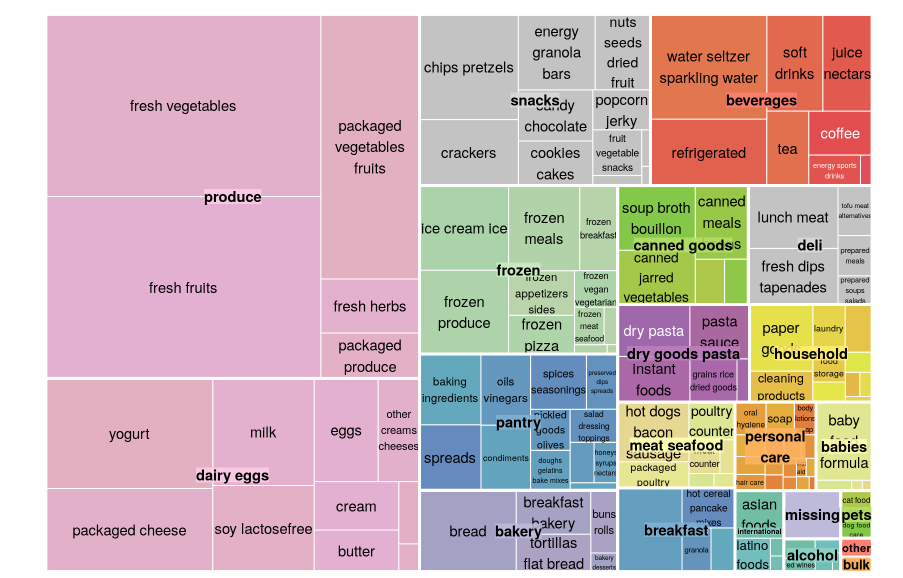
12整个商品分布 21个区 134个通道



13每个区每个通道商品种类数



14那个区哪个通道卖的最的商品



15每次都重购的用户

预测是否之前的商品被重购

1/每次都重购的去除

2.针对用户，商品，

重购数 比例(购买该商品的订单数/总的订单数)， 该商品时间重要性（星期，小时），总的时间重要性，相隔时间的平均值，物品重要性（该物品个数占该用户购买总数的比例），上一次距当前时间，商品购买顺序 1+2/（3+5）

影响这次订单里该商品是否会重购的因素以订单来看

0 之前没有购买，这次购买 //0 之前购买，这次也购买 1 之前购买，这次不购买

df = pd.read\_csv('test.csv', chunksize=10000)

for chunk in df:

# TODO: process the chunk as a normal DataFrame

pandas 的 DataFrame.to\_csv 也考虑这个问题了。我们只需要添加一个参数 mode='a'

from abc import ABCMeta, abstractmethod

class MyABC:

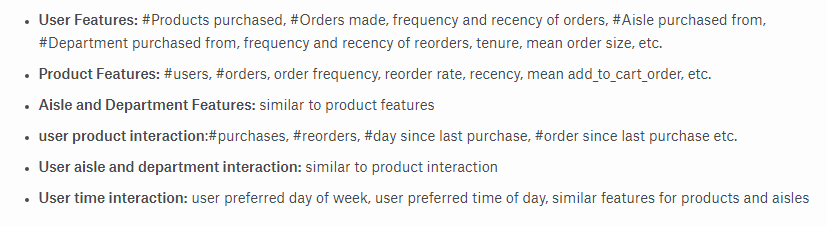
\_\_metaclass\_\_ = ABCMeta

@abstractmethod

def foo():

pass

bar = MyABC() # fails at instantiation, because the class has at least one method which is abstract (foo)



物品重要性：Groupby userid,productid 用户买的各个物品数

Group by user，用户买的所有物品

重购数比例：groupby userid,product id 购买该商品的订单数

Group by useid ,orderid 购买总的订单数

//该商品时间重要性：groupy userid,productid,oderdow

//总的时间重要性：groupy userid, oderdow

总的相隔时间的平均值：group by userid, oderid mean days since prior day

该物品相隔时间的平均值：group by userid productid mean days since prior day

商品购买顺序：group by userid, productid,addtocartnumber .mean 平均购买顺序

item\_importance,user\_id,product\_id,order\_importance,order\_mean\_since\_prior,item\_mean\_since\_prior,add\_to\_cart\_order,label是否之前购买的商品会重购