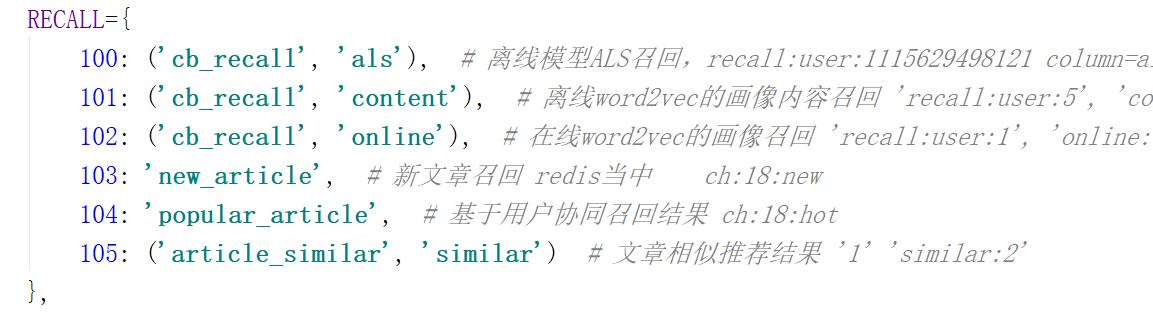
#### 5.4 获取多路召回结果，过滤历史记录逻辑

* 推荐：请求新数据



参照default.py中的recall配置来理解



* 1、循环算法组合参数，遍历不同召回结果合并，18
* 2、过滤，该请求频道(18)的历史推荐记录过滤），推荐频道0频道
  + 0：APP 推荐(循环所有的频道召回结果，是放在0频道)，0 频道也有历史记录，所以0号频道也要过滤一下
* 3、对结果，排序(不做)
* 4、返回结果：
  + 如果有数据，小于需要推荐文章的数量N之后，放入历史推荐记录中history\_recommend，返回结果给用户
  + 如果有，350篇，取出N个，进行返回推荐，放入历史记录history\_recommend
    - (350- N)个文章，放入wait\_recommend

user\_reco\_list是放入reco\_cent的

HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

YARN\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

# 5.6 推荐缓存服务

### 5.6.1 待推荐结果的redis缓存

* 目的：对待推荐结果进行二级缓存，多级缓存减少数据库读取压力
* 每次获取新数据， 所有召回、排序(影响速度)---比如每个用户都去获取13号频道的数据，不是去wait\_recommend里边拿，而是去redis拿
  + 数据放入缓存，以后都从缓存当中去拿
* 步骤：
  + hbase—>redis
  + 配置好redis读取以及hbase工具
  + 首先从获取redis结果，进行判断(缓存拿)
  + 8号数据库中 ，'reco:{}:{}:art' key

**'reco:4:13:art' 1 100**

**'reco:4:13:art' 2 312**

* + 如果redis有，读取需要推荐的文章数量放回，并删除这些文章，并且放入推荐历史推荐结果中
  + 如果redis当中不存在，则从wait\_recommend中读取
    - 如果wait\_recommend中也没有，直接返回空，去进行一次召回读取、排序然后推荐
    - wait*recommend有，从wait*recommend取出所有结果，定**一个数量(如100篇)的文章**存入redis
    - wait\_recommend不够一定数量，全部取出放入redis当中，直接推荐出去

**Recall+new+hot ---->sort--->wait\_recommend**

### 5.6.2 在推荐中心加入缓存逻辑

# 5.7 排序模型在线预测

### 5.7.1排序模型服务

### 5.7.2 排序模型在线预测

* 目的：对召回的结果文章列表，进行该用户对这些文章的点击的概率预测，排序，选出概率前N个结果返回
* 步骤： 1、读取用户特征中心特征
  + 如果该用户特征不存在(离线没有更新到此用户)
  + 不排序
* 2、读取文章特征中心特征、合并用户文章特征构造预测样本
* 3、预测并进行排序是筛选
  + 召回结果过滤之后有1000篇，排序，1000个排序
  + 筛选点击率高的文章
  + 1000个排序，取出前面点击率高的N篇文章(100，200？)

### 5.7.3 添加实时排序的模型预测

<https://zhuanlan.zhihu.com/p/61724627>

<https://blog.csdn.net/ningyanggege/article/details/81133785>

https://liam.page/2019/08/31/a-not-so-simple-introduction-to-FTRL/

读取cb\_recall里边的als，comment，on\_line，然后是热门文章和新文章，过滤history\_recomend，假如总计拿到500篇文章，通过lr\_sort\_service

得到500文章的点击可能性排序，得到200篇，返回

为什么文章画像会可能不存在？

是因为新文章不进给了离线去运算，还给了在线，针对新文章，离线计算文章是4小时一次，新文章的画像还没有算出来，这会造成文章画像不存在