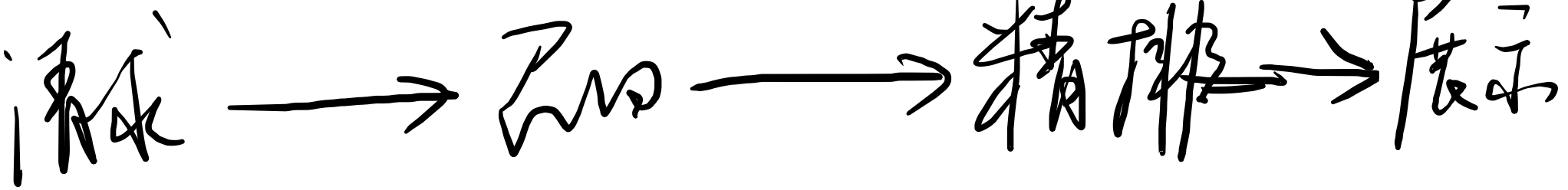
搜索系统的离线架构：

1、内容源的收集与规范化

2、用户的收集与规范化

3、离线训练模型。

总体架构：

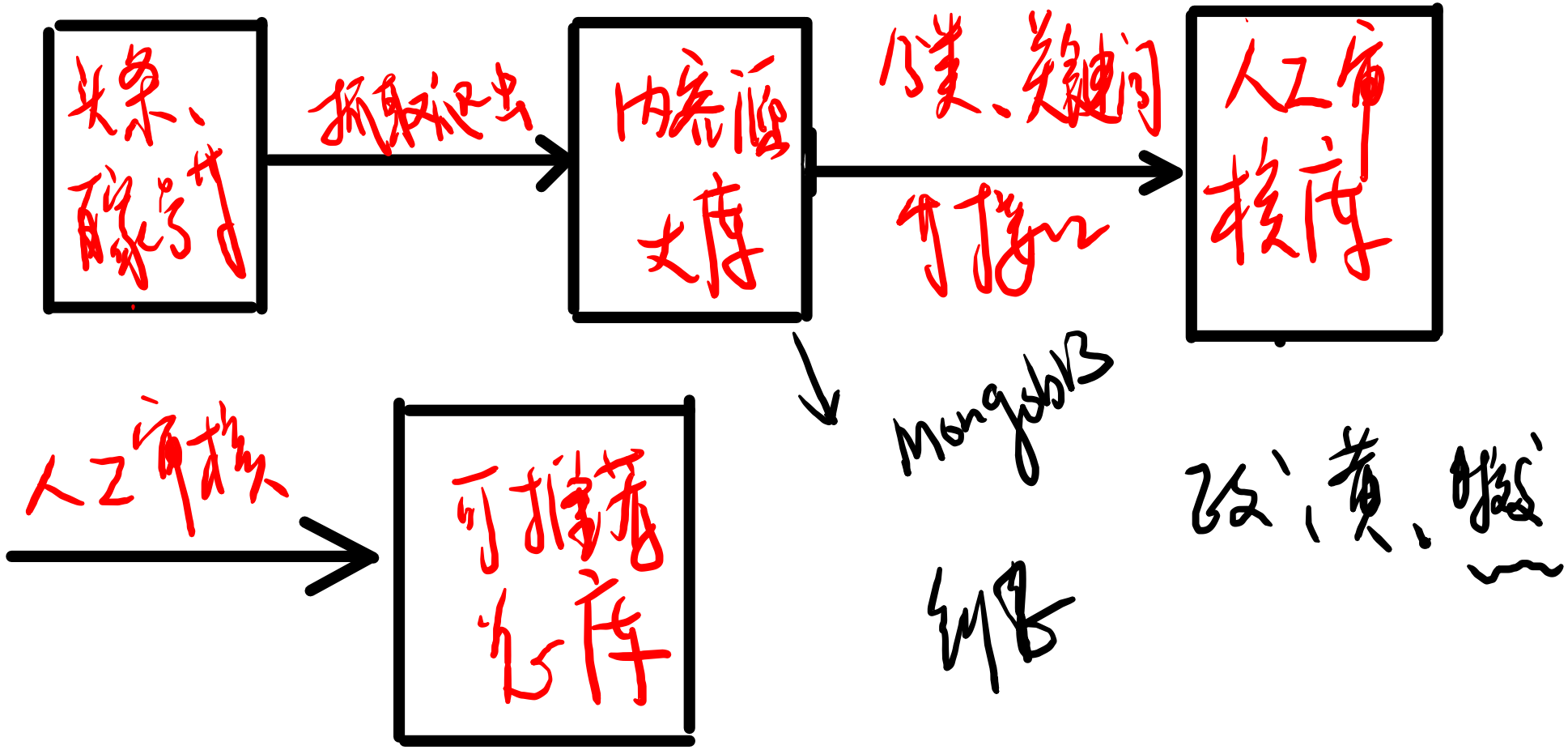


各个部分的目的：

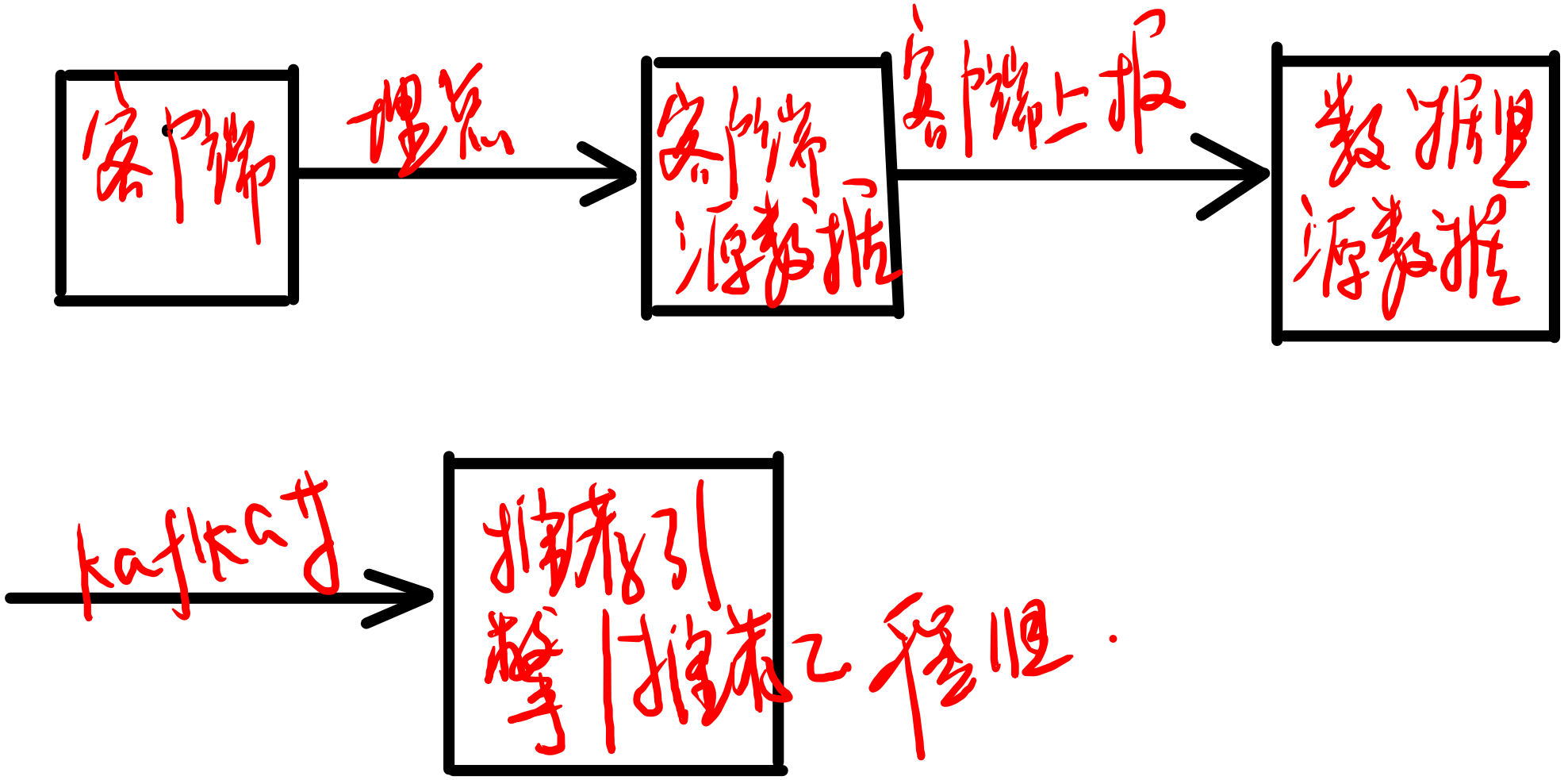
(一) 召回：从百万甚至千万级物品中快速、全面的选出高级别的候选物品

(二) 精排：对召回结果进行精细化打分

1、内容源的收集与规范化



2. 日志的收集与规范化

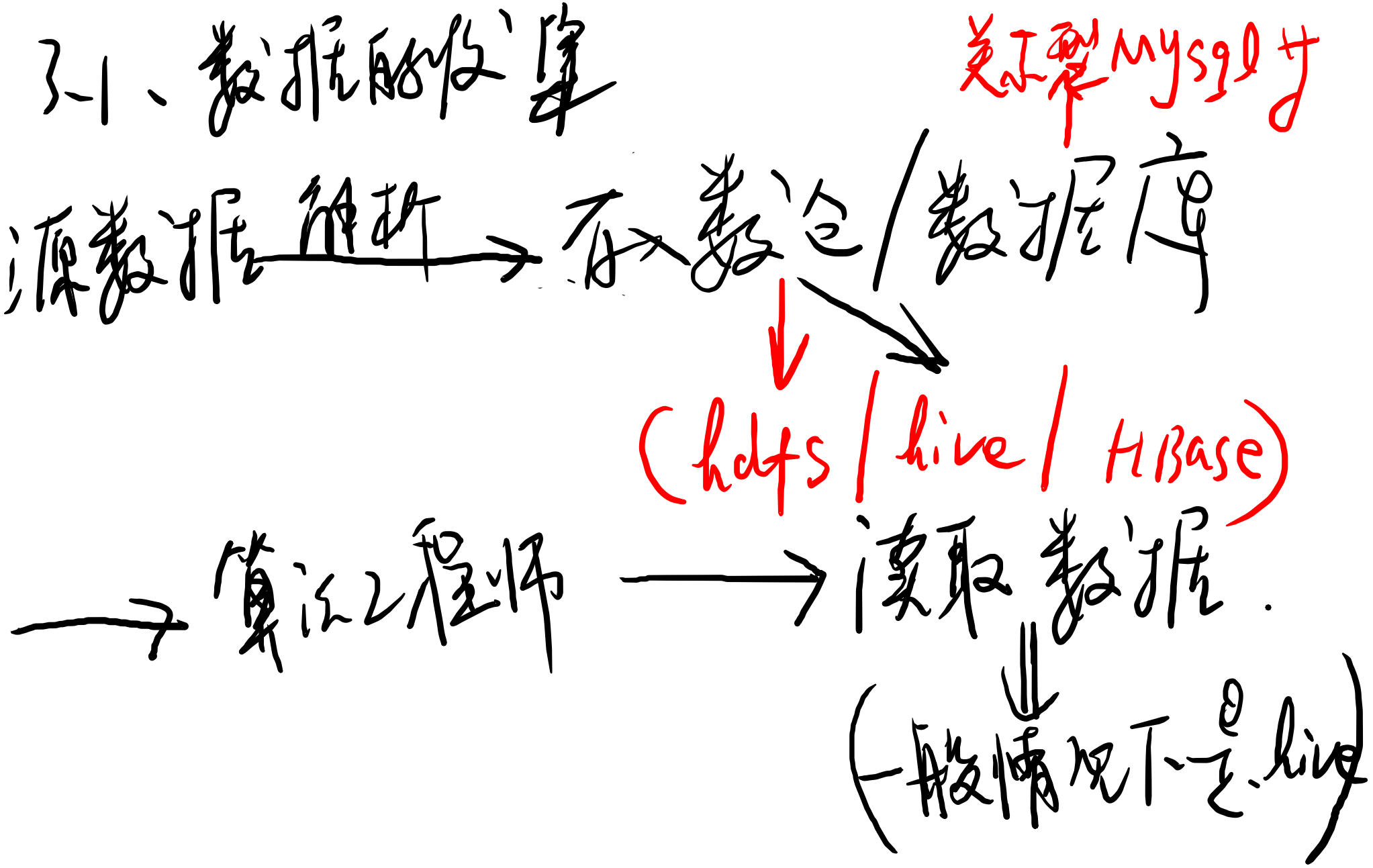


3. 离线训练模型、

3.1、数据的收集

3.2、模型的训练与更新

3-1、数据的收集


关系型MySql
源数据
解析
存入数仓/数据库
(hdfs/hive/HBase)
算法工程师
读取数据
(一般情况下是hive)

3-1 数据的收集.

算法 etc hive —spark、mapReduce、HQL→ 上传至 hdfs 或本地服务器

移动数据 ——————————→

pytorch / tf / keras 等从 hdfs 或本地服务器读取并训练.

3.2、模型的训练与更新

业界主流的几种方式：

1）模型增量更新（天/小时级别）

2）模型每隔一段时间全量手动train

3.2、模型的训练与更新

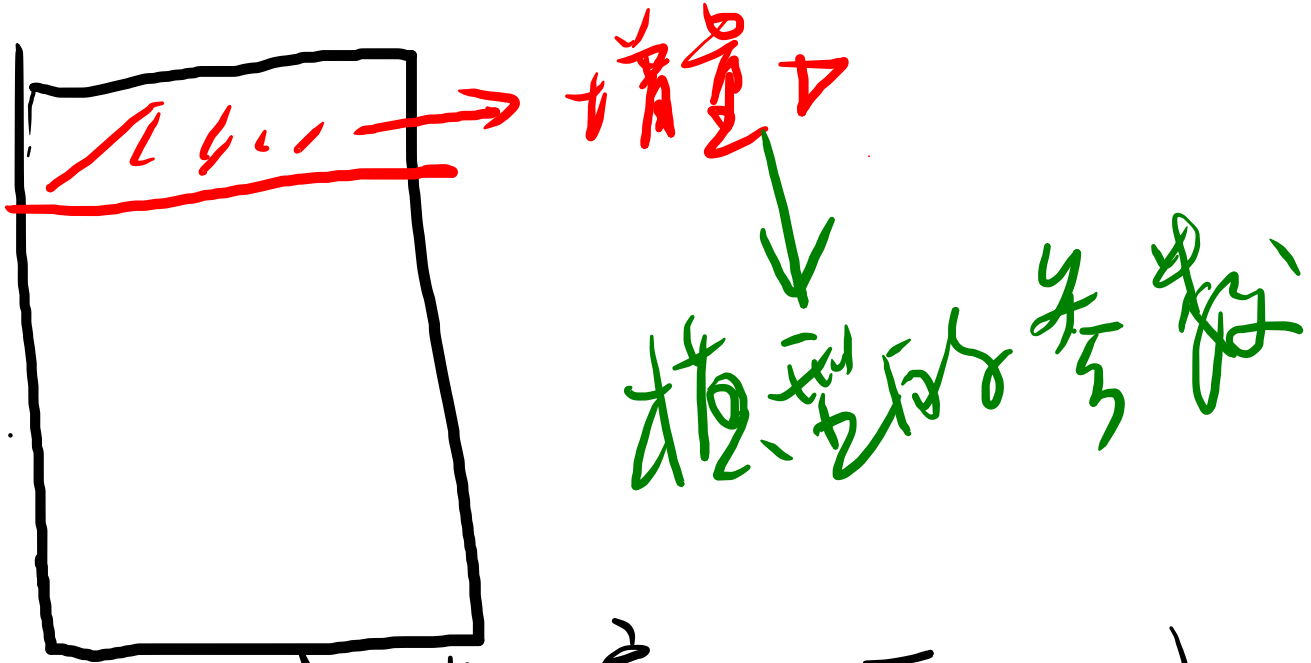
模型的全量与增量：比较常见。

每十天、半个月、七天等间隔时间，全量训练一次。

全量：重新训练

模型的增量更新

1、首先明确什么是增量更新



2、更新频率：天、小时

模型的增量更新.

2、如何增量更新?

↓

后面讲模型的时候详细讲解

异步召回：

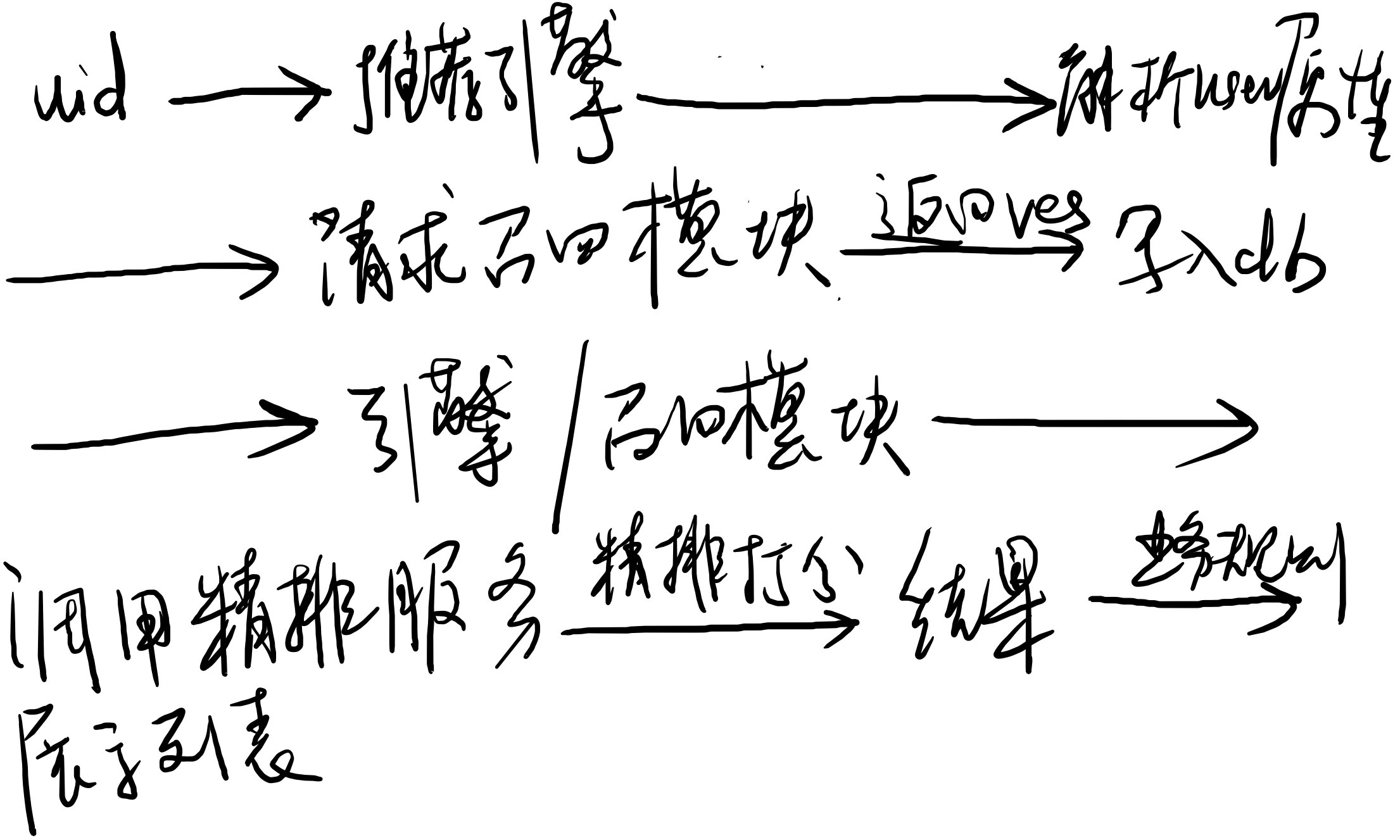
例子、召回模型A 离线一直计算基于模型A的召回结果并放入缓存db（如redis等）

推荐系统架构的时间线都会：

展示：

一个user请求至展示推荐列表的

收后都发生了什么？


uid
推荐引擎
请求召回模块
返回vec
录入db
引擎/召回模块
调用精排服务
精排打分
结果
写规则
展示列表