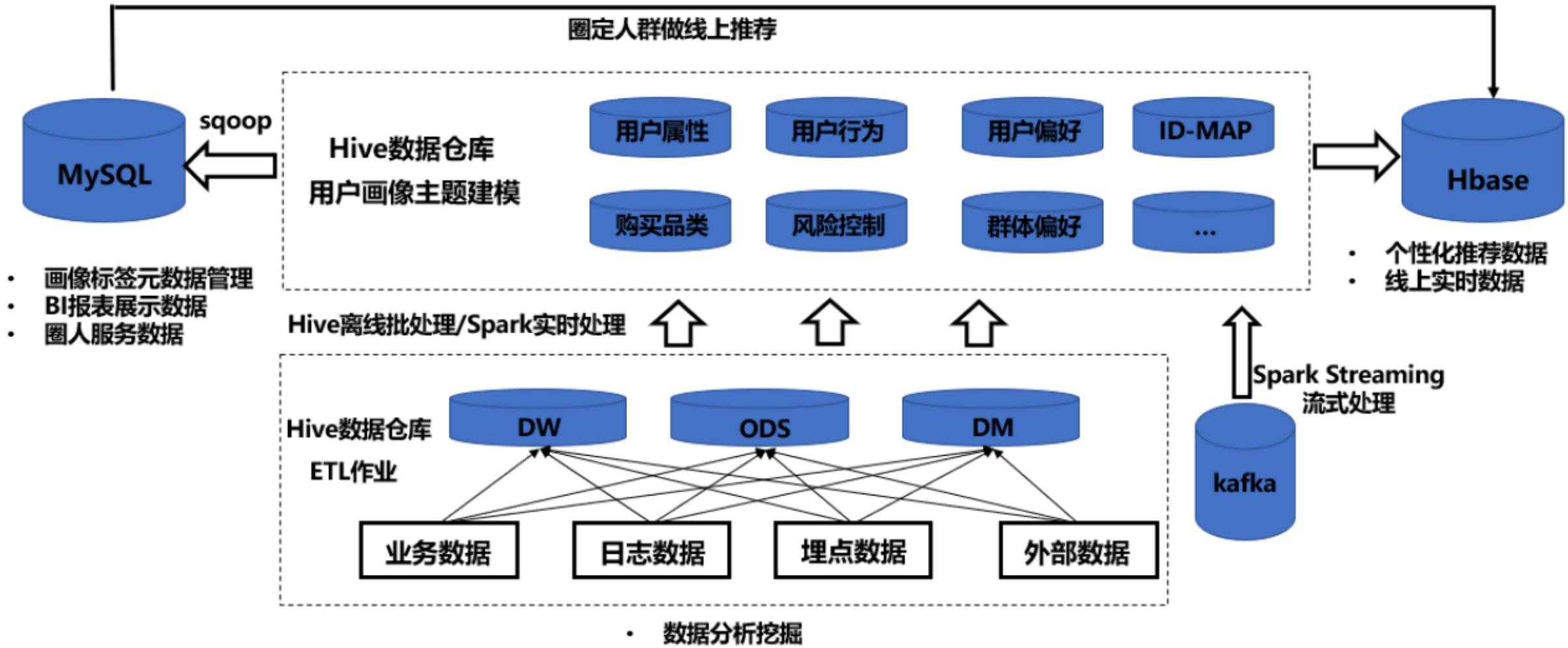
天善智能
TIANSHAN SOFT

用户画像基础内容

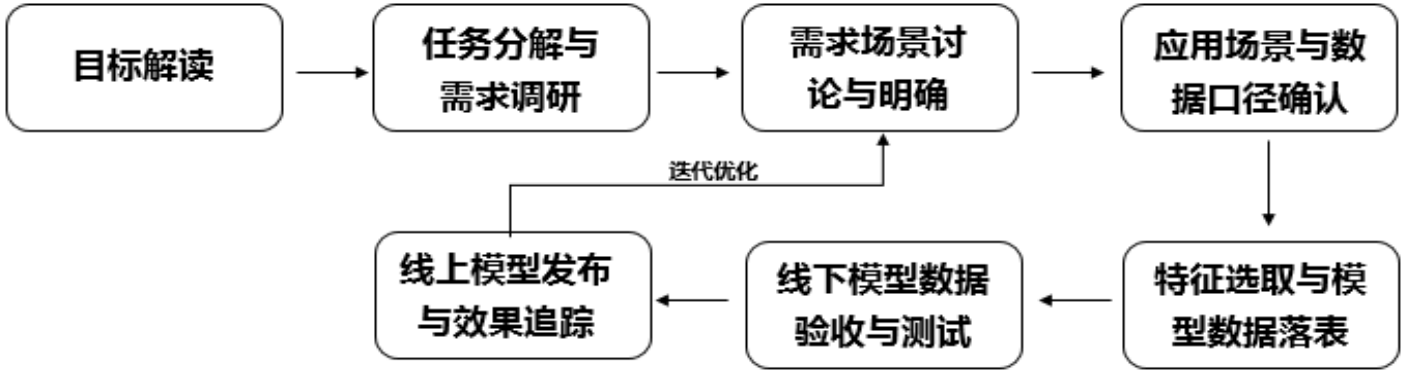
讲师：watermelon

数据架构

从用户画像的数据架构谈需要掌握的大数据模块和开发语言



开发流程



第一阶段：目标解读

在建立用户画像前，首先需要明确用户画像服务于企业的对象，根据业务方需求，未来产品建设目标和用户画像分析之后预期效果；

第二阶段：任务分解与需求调研

经过第一阶段的需求调研和目标解读，我们已经明确了用户画像的服务对象与应用场景，接下来需要针对服务对象的需求侧重点，结合产品现有业务体系和“数据字典”规约实体和标签之间的关联关系，明确分析纬度；

第三阶段：需求场景讨论与明确

在本阶段，数据运营人员需要根据前面与需求方的沟通结果，输出**《产品用户画像规划文档》**，在该文档中明确画像应用场景、最终开发出的标签内容与应用方式，并就该份文档与需求方反复沟通确认无误。

第四阶段：应用场景与数据口径确认

经过第三个阶段明确了需求场景与最终实现的标签纬度、标签类型后，数据运营人员需要结合业务与数据仓库中已有的相关表，明确与各业务场景相关的数据口径。在该阶段中，数据运营方需要输出**《产品用户画像实施文档》**，该文档需要明确应用场景、标签开发的模型、涉及到的数据库与表，应用实施流程；

第五阶段：特征选取与模型数据落表

本阶段中数据分析挖掘人员需要根据前面明确的需求场景进行业务建模，写好HQL逻辑，将相应的模型逻辑写入临时表中，抽取数据校验是否符合业务场景需求。

第六阶段：线下模型数据验收与测试

数据仓库团队的人员将相关数据落表后，设置定时调度任务，进行定期增量更新数据。数据运营人员需要验收数仓加工的HQL逻辑是否符合需求，根据业务需求抽取查看表中数据范围是否在合理范围内，如果发现问题及时反馈给数据仓库人员调整代码逻辑和行为权重的数值。

第七阶段：线上模型发布与效果追踪

经过第六阶段，数据通过验收之后，就可以将数据接口给到搜索、或技术团队部署上线了。上线后通过对用户点击转化行为的持续追踪，调整优化模型及相关权重配置。

需要掌握的相关模块

数据开发

Spark

数据存储和查询

Hbase

Hive

MySQL

流式计算

Kafka

Spark Streaming

作业调度(ETL)

crontab

Airflow

需要掌握的开发语言

scala

Hive SQL

Python

shell

需要掌握的其他非技术内容

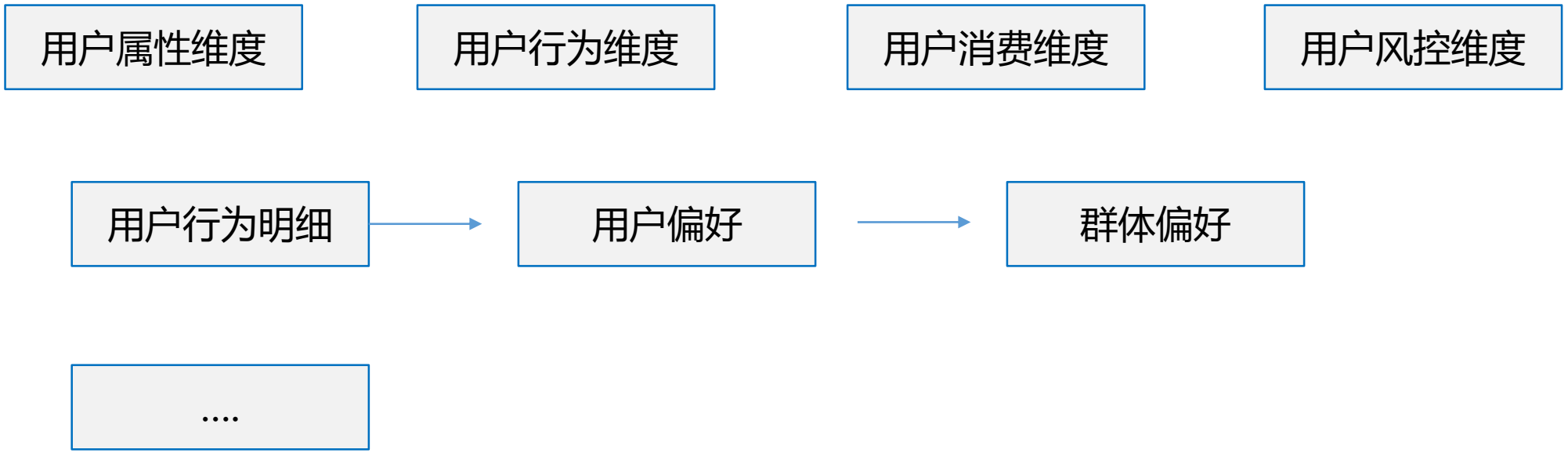
1、数据分析
如何做数据调研、对于需求方提出的标签如何结合数据情况给出相应的解决方案；
…

2、用户画像工程化
用户标签体系中需要开发哪些标签；
这些标签的调度流是如何构成的；
如何对每天的调度作业进行监控；
哪些数据库可用于存储标签，为何用这些数据库进行存储；
…

3、业务知识
需要开发的标签服务于业务上的哪些应用
….

4、画像产品形态及如何服务与业务
画像产品化后包括哪些模块；
如何评价标签在业务上的应用效果；
…

画像体系需要开发的表



表结构设计-日全量

日全量数据表中，每天对应的日期分区中插入截止到当天为止的全量数据，用户使用查询时，只需查询最近一天即可获得最新全量数据。下面以一个具体的日全量表结构例子来做说明。

```
CREATE TABLE dw.userprofile_tag_userid (
tagid STRING COMMENT 'tagid',
userid STRING COMMENT 'userid',
tagweight STRING COMMENT 'tagweight',
reserve STRING COMMENT '预留' )
PARTITIONED BY (data_date STRING COMMENT '数据日期' ,tagtype STRING COMMENT '标签主题分类')
```

这里tagid表示标签名称，userid表示用户id，tagweight表示标签权重，reserve表示一个预留字段。分区方式为（日期+标签主题）分区，设置两个分区字段更便于开发和查询数据。该表结构下的标签权重仅考虑统计类型标签的权重，如：历史购买金额标签对应的权重为金额数量，用户近30日访问天数为对应的天数，该权重值的计算未考虑较为复杂的用户行为次数、行为类型、行为距今时间等复杂情况。**通过全连接的方式与前面每天的数据做join关联**

例如，对于主题类型为“会员”的标签，插入‘20180701’日的全量数据，可通过语句：insert overwrite table dw.userprofile_tag_userid partition(data_date=’20180701’，tagtype=’member’)来实现。查询截止到20180701日的被打上会员标签的用户量，可通过语句：select count(*) from dw.userprofile_tag_userid where data_date=’20180701’来实现。

表结构设计-日增量

日增量数据表中，每天的日期分区中插入当天业务运行产生的数据，用户使用查询时需要限制查询的日期范围，可以找出在特定时间范围内被打上特定标签的用户。下面以一个具体的日增量表结构例子来说明。

```
CREATE TABLE  dw.userprofile_useract_tag (
tagid STRING COMMENT '标签id',
userid STRING COMMENT '用户id',
act_cnt int COMMENT '行为次数',
tag_type_id int COMMENT '标签类型编码',
act_type_id int COMMENT '行为类型编码')
comment '用户画像-用户行为标签表’  PARTITIONED BY (data_date STRING COMMENT '数据日期')
```

这里tagid表示标签名称，userid表示用户id，act_cnt表示用户当日行为次数，如用户当日浏览某三级品类商品3次，则打上次数为3，tag_type_id为标签类型,如母婴、3C、数码等不同类型，act_type_id表示行为类型，如浏览、搜索、收藏、下单等行为。分区方式为按日期分区，插入当日数据。

例如，某用户在‘20180701’日浏览某3C电子商品4次（act_cnt），即给该用户（userid）打上商品对应的三级品类标签（tagid），标签类型（tag_type_id）为3C电子商品，行为类型（act_type_id）为浏览。这里可以通过对标签类型和行为类型两个字段以配置维度表的方式，对数据进行管理。例如对于行为类型（act_type_id）字段，可以设定1为购买行为、2为浏览行为、3为收藏行为等，在行为标签表中以数值定义用户行为类型，在维度表中维护每个数值对应的具体含义。

表结构设计-日增量

日增量的表结构设计参考维度

dpd_event_d	用户行为日志
cookieid	
goodsid	
siteid	
data_date	日分区
visit_cnt	访问次数
cart_cnt	加购次数
fav_cnt	收藏次数
share_cnt	分享次数
checkout_cnt	checkout点击次数
pay_type_cnt	支付类型选择次数
signed_cnt	签到点击次数
reg_cnt	注册次数
trytopay_cnt	支付点击次数
visit_duration	访问时长
paid_cnt	购买次数
order_cnt	下单次数
goods_price	商品单价
goods_impression_cnt	商品曝光次数
goods_click_cnt	商品点击次数
gd_reviews_click_cnt	详情页评论点击次数
goodsdetail_reviews_more_click	详情页评论点击more次数
gd_details_click_cnt	详情页detail点击次数
gd_select_click_cnt	详情页select点击次数
gd_share_click_cnt	详情页分享点击次数
gd_shopgoods_click_cnt	详情页被推荐点击次数
cart_click_cnt	加购点击次数
cart_submit_click_cnt	加购submit点击次数
fav_recommend_cnt	收藏下面推荐商品次数
fav_main_cnt	收藏主商品次数
sizeguide_click_cnt	尺码参考点击次数
gd_shop_tips_click_cnt	详情页购买tips点击次数
gd_brand_click_cnt	详情页品牌点击次数
gd_brand_recommend_cnt	详情页品牌推荐点击次数
gd_coupon_click_cnt	详情页优惠券入口点击次数
gd_coupon_cnt	详情页优惠券领取成功次数