数据集介绍

阿里云天池数据集: User Behavior Data from Taobao for Recommendation

字段	说明
User ID	整数类型，序列化后的用户ID
Item ID	整数类型，序列化后的商品ID
Category ID	整数类型，序列化后的商品所属类目ID
Behavior type	字符串，枚举类型，包括('pv', 'buy', 'cart', 'fav')
Timestamp	行为发生的时间戳

Behavior type	说明
pv	商品详情页pv，等价于点击
buy	商品购买
cart	将商品加入购物车

导入数据

数据库准备

create database taobao;
use taobao;

create table user_behavior (user_id int(9), item_id int(9), category_id int(9), behavior_type varchar(5), timestamp int(14));

kettle配置

连接池100 最大空闲空间100 默认提交取消

参数配置

字段	值
useServerPrepStmts	false

字段	值
useCompression	true
rewriteBatchedStatements	true

线程8个

数据预处理

```
    USE taobao;
DESC user_behavior;
SELECT * FROM user_behavior LIMIT 5;

-- 改变字段名
ALTER TABLE user_behavior CHANGE TIMESTAMP TIMESTAMPS INT(14);
DESC USER_BEHAVIOR;

-- 检查空值
SELECT * FROM user_behavior WHERE user_id IS NULL;
SELECT * FROM user_behavior WHERE item_id IS NULL;
SELECT * FROM user_behavior WHERE category_id IS NULL;
SELECT * FROM user_behavior WHERE behavior_type IS NULL;
SELECT * FROM user_behavior WHERE timestamps IS NULL;

--检查重复值
WITH rankedbehavior AS (
    SELECT
        user_id,
        item_id,
        timestamps,
        ROW_NUMBER() OVER (PARTITION BY user_id, item_id, timestamps ORDER BY
timestamps) AS ROW_NUM
    FROM user_behavior
)
SELECT
    user_id,
    item_id,
    timestamps
FROM rankedbehavior
WHERE ROW_NUM > 1;
```

```
-- 去重
ALTER TABLE user_behavior ADD ID INT FIRST;
SELECT * FROM user_behavior LIMIT 5;
ALTER TABLE user_behavior MODIFY ID INT PRIMARY KEY AUTO_INCREMENT;

WITH rankedbehavior AS (
    SELECT
        id,
        user_id,
        item_id,
        timestamps,
        ROW_NUMBER() OVER (PARTITION BY user_id, item_id, timestamps ORDER BY
ID) AS ROW_NUM
    FROM user_behavior
)
DELETE FROM user_behavior
WHERE id IN (
    SELECT id
    FROM rankedbehavior
    WHERE ROW_NUM > 1
);

-- 新增日期：DATE TIME HOUR
-- 更改BUFFER值
SHOW VARIABLES LIKE '%_BUFFER%';

SET GLOBAL innodb_buffer_pool_size = 10700000000;

-- DATETIME
ALTER TABLE user_behavior ADD datetimes TIMESTAMP(0);
UPDATE user_behavior
SET datetimes = FROM_UNIXTIME(timestamps);
SELECT * FROM user_behavior LIMIT 5;

-- DATE
ALTER TABLE user_behavior ADD dates CHAR(10);
ALTER TABLE user_behavior ADD times CHAR(8);
ALTER TABLE user_behavioR ADD hours CHAR(2);
```

```
-- UPDATE DATES TIMES AND HOURS
UPDATE user_behavior
SET dates = SUBSTRING(datetimes, 1, 10);
UPDATE user_behavior
SET times = SUBSTRING(datetimes, 12, 8);
UPDATE user_behavior
SET hours = SUBSTRING(datetimes, 12, 2);
SELECT * FROM user_behavior LIMIT 5;

-- 去异常时间
SELECT MAX(datetimes),MIN(datetimes) FROM user_behavior;
DELETE FROM user_behavior
WHERE datetimes < '2017-11-25 00:00:00'
OR datetimes > '2017-12-03 23:59:59';

-- 数据概览
DESC user_behavior;
SELECT * FROM user_behavior LIMIT 5;
SELECT COUNT(1) FROM user_behavior;-- 剩余100095496条数据
```

获客情况

```
-- 创建临时表
CREATE TABLE temp_behavior LIKE user_behavior;-- 截取数据
INSERT INTO temp_behavior SELECT
*
FROM
  user_behavior
  LIMIT 10000;
SELECT
  *
FROM
  temp_behavior;

-- pv
SELECT
  dates,
  COUNT(*) AS 'pv'
FROM
  temp_behavior
```

```
WHERE
  behavior_type = 'pv'
GROUP BY
  dates;

-- uv
SELECT
  dates,
  COUNT(DISTINCT (user_id)) AS 'uv'
FROM
  temp_behavior
WHERE
  behavior_type = 'pv'
GROUP BY
  dates;

-- pv/uv
SELECT
  dates,
  COUNT(*) 'pv',
  COUNT(DISTINCT user_id) 'uv',
  ROUND(COUNT(*) / COUNT(DISTINCT user_id), 1) 'pv/uv'
FROM
  temp_behavior
WHERE
  behavior_type = 'pv'
GROUP BY
  dates;

-- 处理真实数据
CREATE TABLE PV_UV_PUV(
dates CHAR(10),
pv INT(9),
uv INT(9),
puv DECIMAL(10,1));

INSERT INTO pv_uv_puv
SELECT
  dates,
  COUNT(*) 'PV',
```

```
  COUNT(DISTINCT user_iD) 'UV',
  ROUND(COUNT(*) / COUNT(DISTINCT user_id), 1) 'PV/UV'
FROM
  user_behavior
WHERE
  behavior_type = 'PV'
GROUP BY
  dates;

SELECT * FROM pv_uv_puv;

DELETE FROM pv_uv_puv WHERE dates IS NULL;
```

留存情况

```
SELECT * FROM user_behavior WHERE dates IS NULL;
DELETE FROM user_behavior WHERE dates IS NULL;

SELECT user_id,dates
FROM temp_behavior
GROUP BY user_id,dates;

-- 自关联
WITH a AS (
  SELECT user_id, dates
  FROM temp_behavior
  GROUP BY user_id, dates
)
SELECT a.user_id, a.dates AS date1, b.dates AS date2
FROM a
JOIN a AS b
ON a.user_id = b.user_id ;

-- 筛选
WITH a AS (
  SELECT user_id, dates
  FROM temp_behavior
```

```
  GROUP BY user_id, dates
)
SELECT a.user_id, a.dates AS date1, b.dates AS date2
FROM a
JOIN a AS b
ON a.user_id = b.user_id AND a.dates < b.dates;

-- 留存数
WITH user_dates AS (
  SELECT user_id, dates
  FROM temp_behavior
  GROUP BY user_id, dates
)
SELECT
  a.dates,
  COUNT(CASE WHEN DATEDIFF(b.dates, a.dates) = 0 THEN b.user_id END) AS
retention_0,
  COUNT(CASE WHEN DATEDIFF(b.dates, a.dates) = 1 THEN b.user_id END) AS
retention_1,
  COUNT(CASE WHEN DATEDIFF(b.dates, a.dates) = 3 THEN b.user_id END) AS
retention_3
FROM
  user_dates a
JOIN
  user_dates b
ON
  a.user_id = b.user_id
  AND a.dates <= b.dates
GROUP BY
  a.dates;

-- 留存率
WITH user_dates AS (
  SELECT user_id, dates
  FROM temp_behavior
  GROUP BY user_id, dates
)
SELECT
  a.dates,
  COUNT(CASE WHEN DATEDIFF(b.dates, a.dates) = 1 THEN b.user_id END)/
```

```
COUNT(CASE WHEN DATEDIFF(b.dates, a.dates) = 0 THEN b.user_id END) AS
retention
FROM
  user_dates a
JOIN
  user_dates b
ON
  a.user_id = b.user_id
  AND a.dat
  es <= b.dates
GROUP BY
  a.dates;

-- 保存结果
CREATE TABLE retention_rate(
dates CHAR(10),
retention_1 FLOAT
);

INSERT INTO retention_rate
WITH user_dates AS (
  SELECT user_id, dates
  FROM user_behavior
  GROUP BY user_id, dates
)
SELECT
  a.dates,
  COUNT(CASE WHEN DATEDIFF(b.dates, a.dates) = 1 THEN b.user_id END)/
COUNT(CASE WHEN DATEDIFF(b.dates, a.dates) = 0 THEN b.user_id END) AS
retention
FROM
  user_dates a
JOIN
  user_dates b
ON
  a.user_id = b.user_id
  AND a.dates <= b.dates
GROUP BY
  a.dates;
```

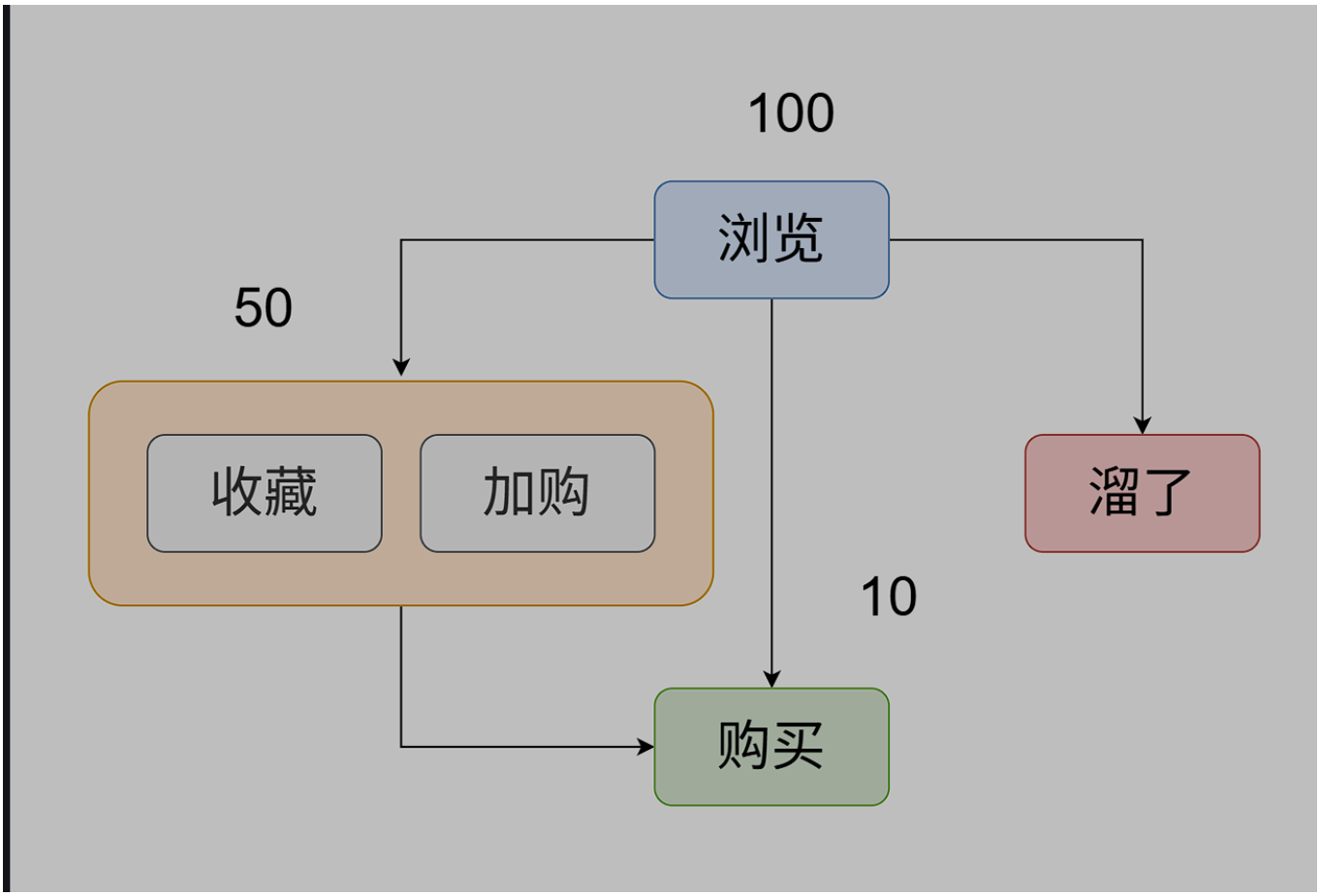
```
SELECT * FROM retention_rate

-- 跳失率 --88/89660669
-- 跳失用户 -- 88个
WITH a AS (
  SELECT user_id FROM user_behavior
  GROUP BY user_id
  HAVING COUNT(behavior_type) =1
)

SELECT * FROM a;
 -- 总访问量
SELECT SUM(pv) FROM pv_uv_puv; -- 89660669
```

时间序列分析

```
SELECT 'action'
-- 统计日期一小时的行为
SELECT dates, hours,
COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS 'pv',
COUNT(IF(behavior_type='cart',behavior_type,NULL)) AS 'cart',
COUNT(IF(behavior_type='fav',behavior_type,NULL)) AS 'fav',
COUNT(IF(behavior_type='buy',behavior_type,NULL)) AS 'buy'
FROM temp_behavior
GROUP BY dates,hours
ORDER BY dates,hours

-- 存储结果
CREATE TABLE date_hour_behavior(
  dates char(10),
  hores char(2),
  pv int,
  cart int,
  fav int,
  buy int
);

INSERT INTO date_hour_behavior
SELECT dates, hours,
COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS 'pv',
```

```
COUNT(IF(behavior_type='cart',behavior_type,NULL)) AS 'cart',
COUNT(IF(behavior_type='fav',behavior_type,NULL)) AS 'fav',
COUNT(IF(behavior_type='buy',behavior_type,NULL)) AS 'buy'
FROM user_behavior
GROUP BY dates,hours
ORDER BY dates,hours

SELECT * FROM date_hour_behavior;
```

用户转化率分析




```
-- 统计各类行为用户数
SELECT
  behavior_type,
  COUNT(DISTINCT user_id) AS user_num
FROM
  temp_behavior
GROUP BY
  behavior_type
ORDER BY
```

```
  behavior_type DESC;

-- 保存
CREATE TABLE behavior_user_num(
  behavior_type varchar(5),
  user_num int
);

INSERT INTO behavior_user_num
SELECT
  behavior_type,
  COUNT(DISTINCT user_id) AS user_num
FROM
  user_behavior
GROUP BY
  behavior_type
ORDER BY
  behavior_type DESC;

SELECT * FROM behavior_user_num;

SELECT 672404/984105;-- 转化率为0.6833，也就是有68%的用户浏览后下单了

-- 统计各类行为数量
SELECT
  behavior_type,
  COUNT(*) AS behavior_count_num
FROM
  temp_behavior
GROUP BY
  behavior_type
ORDER BY
  behavior_type DESC;

  -- 保存
  CREATE TABLE behavior_num(
  behavior_type varchar(5),
  behavior_count_num int
);
```

```
INSERT INTO behavior_num
SELECT
  behavior_type,
  COUNT(*) AS behavior_count_num
FROM
  user_behavior
GROUP BY
  behavior_type
ORDER BY
  behavior_type DESC;

SELECT * FROM behavior_num;

SELECT 2015807/89660669; -- 浏览行为的购买率

SELECT (2888257+5530445)/89660669 -- 浏览行为的收藏与加入购物车率
```

行为路径分析

```
SELECT '难度增加'
DROP VIEW user_behavior_view
DROP VIEW user_behavior_standard
DROP VIEW user_behavior_path
DROP VIEW path_count

-- 临时表
CREATE VIEW user_behavior_view AS
SELECT user_id,item_id,
COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS 'pv',
COUNT(IF(behavior_type='fav',behavior_type,NULL)) AS 'fav',
COUNT(IF(behavior_type='cart',behavior_type,NULL)) AS 'cart',
COUNT(IF(behavior_type='buy',behavior_type,NULL)) AS 'buy'
FROM temp_behavior
GROUP BY user_id,item_id;

-- 用户行为标准化
CREATE VIEW user_behavior_standard AS
SELECT user_id,item_id,
(CASE WHEN pv>0 THEN 1 ELSE 0 END) AS 浏览了,
(CASE WHEN fav>0 THEN 1 ELSE 0 END) AS 收藏了,
```

```
(CASE WHEN cart>0 THEN 1 ELSE 0 END) AS 加购了,
(CASE WHEN buy>0 THEN 1 ELSE 0 END) AS 购买了
FROM user_behavior_view

-- 路径类型
CREATE VIEW user_behavior_path AS
SELECT *,
CONCAT(浏览了,收藏了,加购了,购买了) AS 购买路径类型
FROM user_behavior_standard AS a
WHERE a.购买了>0

-- 统计各类购买行为数量
CREATE VIEW path_count AS
SELECT 购买路径类型,
COUNT(*) AS 数量
FROM user_behavior_path
GROUP BY 购买路径类型
ORDER BY 数量 DESC;

-- 口语话表达
CREATE TABLE renhua(
 path_type char(4),
 description varchar(50)
);

INSERT INTO renhua
VALUES('0001','直接购买了'),
('1001','浏览后购买了'),
('0011','加购后购买了'),
('1011','浏览加购后购买了'),
('0101','收藏后购买了'),
('1101','浏览收藏后购买了'),
('0111','收藏加购后购买了'),
('1111','浏览收藏加购后购买了')

SELECT * FROM renhua;

SELECT *
FROM path_count t
JOIN renhua r
```

```
ON t.`购买路径类型`= r.path_type
ORDER BY t.`数量` DESC

-- 保存
CREATE TABLE path_result(
  path_type char(4),
  description varchar(50),
  num int
);
-- 主表
CREATE VIEW user_behavior_view AS
SELECT user_id,item_id,
COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS 'pv',
COUNT(IF(behavior_type='fav',behavior_type,NULL)) AS 'fav',
COUNT(IF(behavior_type='cart',behavior_type,NULL)) AS 'cart',
COUNT(IF(behavior_type='buy',behavior_type,NULL)) AS 'buy'
FROM user_behavior
GROUP BY user_id,item_id;

-- 用户行为标准化
CREATE VIEW user_behavior_standard AS
SELECT user_id,item_id,
(CASE WHEN pv>0 THEN 1 ELSE 0 END) AS 浏览了,
(CASE WHEN fav>0 THEN 1 ELSE 0 END) AS 收藏了,
(CASE WHEN cart>0 THEN 1 ELSE 0 END) AS 加购了,
(CASE WHEN buy>0 THEN 1 ELSE 0 END) AS 购买了
FROM user_behavior_view

-- 路径类型
CREATE VIEW user_behavior_path AS
SELECT *,
CONCAT(浏览了,收藏了,加购了,购买了) AS 购买路径类型
FROM user_behavior_standard AS a
WHERE a.购买了>0

-- 统计各类购买行为数量
CREATE VIEW path_count AS
SELECT 购买路径类型,
COUNT(*) AS 数量
FROM user_behavior_path
```

```
GROUP BY 购买路径类型
ORDER BY 数量 DESC;

INSERT INTO path_result
SELECT path_type,description,数量
FROM path_count t
JOIN renhua r
ON t.`购买路径类型`= r.path_type
ORDER BY t.`数量` DESC

SELECT * FROM path_result;

SELECT SUM(buy)
FROM user_behavior_view
WHERE buy>0 AND fav=0 AND cart=0;

-- 1528016 直接购买的人数
-- 总购买人数2015807
-- 2015807-1528016=收藏加购再购买的人数
SELECT 2015807-1528016;-- 487791人
-- 更新收藏与加入购物车的购买率 89660669为总pv人数
SELECT 487791/(2888257+5530445) ;-- 0.0579
```

RFM模型

```
--最近购买时间
SELECT
  user_id,
  MAX(dates) AS '最近购买时间'
FROM
  temp_behavior
WHERE
  behavior_type = 'buy'
GROUP BY
  user_id
ORDER BY
  2 DESC;

-- 购买次数
SELECT
```

```
  user_id,
  COUNT(user_id) AS '购买次数'
FROM
  temp_behavior
WHERE
  behavior_type = 'buy'
GROUP BY
  user_id
ORDER BY
  2 DESC

-- 统一最近购买时间与购买次数
SELECT
  user_id,
  COUNT(user_id) AS '购买次数',
  MAX(dates) AS '最近购买时间'
FROM
  user_behavior
WHERE
  behavior_type = 'buy'
GROUP BY
  user_id
ORDER BY
  2 DESC ,3 DESC

-- 存储上面的结果
DROP TABLE IF EXISTS rfm_model;
CREATE TABLE rfm_model(
user_id int,
frequency int,
recent char(10)
)

INSERT INTO rfm_model
SELECT
  user_id,
  COUNT(user_id) AS '购买次数',
  max(dates) AS'最近购买时间'
FROM
  user_behavior
```

```
WHERE
  behavior_type = 'buy'
GROUP BY
  user_id
ORDER BY
  2 DESC ,3 DESC
  SELECT * FROM rfm_model;

-- 根据购买次数对用户进行分层
ALTER TABLE rfm_model ADD COLUMN fscore INT;

UPDATE rfm_model
SET fscore = CASE
    WHEN frequency BETWEEN 100 AND 262 THEN 5
  WHEN frequency BETWEEN 50 AND 100 THEN 4
  WHEN frequency BETWEEN 20 AND 50 THEN 4
  WHEN frequency BETWEEN 5 AND 20 THEN 2
    ELSE 1
END;

 -- 根据最近购买时间对用户进行分层
 ALTER TABLE rfm_model ADD COLUMN rscore INT;
 UPDATE rfm_model
 SET rscore = CASE
 WHEN recent = '2017-12-03' THEN 5
 WHEN recent IN ('2017-12-01','2017-12-02')THEN 4
 WHEN recent IN ('2017-11-29','2017-11-30') THEN 3
 WHEN recent IN ('2017-11-27','2017-11-28') THEN 2
 ELSE 1
 END

 SELECT * FROM rfm_model;

 -- 分层
 SET @f_avg = NULL;
 SET @r_rag = NULL;
 SELECT AVG(fscore) INTO @f_avg FROM rfm_model;
 SELECT AVG(rscore) INTO @r_avg FROM rfm_model;

 SELECT
```

```
    *,
    (CASE
 WHEN fscore >@f_avg AND rscore>@r_avg THEN '价值用户'
 WHEN fscore >@f_avg AND rscore<@r_avg THEN '保持用户'
 WHEN fscore <@f_avg AND rscore>@r_avg THEN '发展用户'
 WHEN fscore <@f_avg AND rscore<@r_avg THEN '挽留用户'
 END ) class
 FROM rfm_model

 -- 插入
 ALTER TABLE rfm_model ADD COLUMN class varchar(10);
 UPDATE rfm_model
 SET class = CASE
 WHEN fscore >@f_avg AND rscore>@r_avg THEN '价值用户'
 WHEN fscore >@f_avg AND rscore<@r_avg THEN '保持用户'
 WHEN fscore <@f_avg AND rscore>@r_avg THEN '发展用户'
 WHEN fscore <@f_avg AND rscore<@r_avg THEN '挽留用户'
 END

 -- 统计各分区用户数
SELECT
  class,
  COUNT(user_id)
FROM
  rfm_model
GROUP BY
  class;
```

商品按热度分类

```
-- 品类浏览量
SELECT
  category_id,
  COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS '品类浏览量'
FROM
  temp_behavior
GROUP BY
  category_id
ORDER BY
  2 DESC
```

```
LIMIT 10;

-- 商品浏览量
SELECT
  item_id,
  COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS '商品浏览量'
FROM
  temp_behavior
GROUP BY
  item_id
ORDER BY
  2 DESC
LIMIT 10;

-- 各类别最热门的商品
WITH a AS (
  SELECT
    category_id,
    item_id,
    COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS pv_count,
    RANK() OVER (PARTITION BY category_id ORDER BY
COUNT(IF(behavior_type='pv',behavior_type,NULL)) DESC) AS r
  FROM
    temp_behavior
  GROUP BY
    category_id, item_id
)
SELECT
  category_id,
  item_id,
  pv_count AS '品类商品浏览量'
FROM
  a
WHERE
  a.r = 1
ORDER BY
  '品类商品浏览量' DESC
LIMIT 10;

-- 保存结果
```

```
CREATE TABLE popular_categories(
  category_id INT,
  pv INT
);
CREATE TABLE popular_items(
  item_id INT,
  pv INT
);
CREATE TABLE popular_cateitems(
  category_id INT,
  item_id INT,
  pv INT
);

-- 插入
INSERT INTO popular_categories
SELECT
  category_id,
  COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS '品类浏览量'
FROM
  user_behavior
GROUP BY
  category_id
ORDER BY
  2 DESC
LIMIT 10;

INSERT INTO popular_items
SELECT
  item_id,
  COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS '商品浏览量'
FROM
  user_behavior
GROUP BY
  item_id
ORDER BY
  2 DESC
LIMIT 10;

INSERT INTO popular_cateitems
```

```
WITH a AS (
  SELECT
    category_id,
    item_id,
    COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS pv_count,
    RANK() OVER (PARTITION BY category_id ORDER BY
COUNT(IF(behavior_type='pv',behavior_type,NULL)) DESC) AS r
  FROM
    user_behavior
  GROUP BY
    category_id, item_id
)
SELECT
  category_id,
  item_id,
  pv_count AS '品类商品浏览量'
FROM
  a
WHERE
  a.r = 1
ORDER BY
  '品类商品浏览量' DESC
LIMIT 10;

SELECT * FROM popular_categories;
SELECT * FROM popular_items;
SELECT * FROM popular_cateitems;
```

商品转化率分析

```
-- 特定商品转化率分析
SELECT item_id,
COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS 'pv',
COUNT(IF(behavior_type='fav',behavior_type,NULL)) AS 'fav',
COUNT(IF(behavior_type='cart',behavior_type,NULL)) AS 'cart',
COUNT(IF(behavior_type='buy',behavior_type,NULL)) AS 'buy',
COUNT(DISTINCT IF(behavior_type='buy',user_id,NULL))/COUNT(DISTINCT user_id)
AS 商品转化率
FROM temp_behavior
GROUP BY item_id
```

```
ORDER BY 商品转化率 DESC

-- 保存
CREATE TABLE item_detail(
  item_id INT,
  pv INT,
  fav INT,
  cart INT,
  buy INT,
  user_buy_rate FLOAT
);

INSERT INTO item_detail
SELECT item_id,
COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS 'pv',
COUNT(IF(behavior_type='fav',behavior_type,NULL)) AS 'fav',
COUNT(IF(behavior_type='cart',behavior_type,NULL)) AS 'cart',
COUNT(IF(behavior_type='buy',behavior_type,NULL)) AS 'buy',
COUNT(DISTINCT IF(behavior_type='buy',user_id,NULL))/COUNT(DISTINCT user_id)
AS 商品转化率
FROM user_behavior
GROUP BY item_id
ORDER BY 商品转化率 DESC

SELECT * FROM item_detail;

-- 品类转换率
SELECT category_id,
COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS 'pv',
COUNT(IF(behavior_type='fav',behavior_type,NULL)) AS 'fav',
COUNT(IF(behavior_type='cart',behavior_type,NULL)) AS 'cart',
COUNT(IF(behavior_type='buy',behavior_type,NULL)) AS 'buy',
COUNT(DISTINCT IF(behavior_type='buy',user_id,NULL))/COUNT(DISTINCT user_id)
AS 品类转化率
FROM temp_behavior
GROUP BY category_id
ORDER BY 品类转化率 DESC

CREATE TABLE category_detail(
  category_id INT,
```

```
  pv INT,
  fav INT,
  cart INT,
  buy INT,
  user_buy_rate FLOAT
);

INSERT INTO category_detail
SELECT category_id,
COUNT(IF(behavior_type='pv',behavior_type,NULL)) AS 'pv',
COUNT(IF(behavior_type='fav',behavior_type,NULL)) AS 'fav',
COUNT(IF(behavior_type='cart',behavior_type,NULL)) AS 'cart',
COUNT(IF(behavior_type='buy',behavior_type,NULL)) AS 'buy',
COUNT(DISTINCT IF(behavior_type='buy',user_id,NULL))/COUNT(DISTINCT user_id)
AS 品类转化率
FROM user_behavior
GROUP BY category_id
ORDER BY 品类转化率 DESC

SELECT * FROM category_detail;
```