Spark SQL 实验：电商用户行为分析

**一、实验目标**

1. 掌握 Spark SQL 核心语法（`SELECT`/`GROUP BY`/`WINDOW`/条件聚合等），理解结构化数据处理逻辑。

2. 对比 Spark RDD 与 Spark SQL 的差异，体会 SQL 对“结构化分析场景”的简洁性优势。

3. 基于电商用户行为数据，用 Spark SQL 完成基础统计与进阶分析，输出可落地的业务结论。

**二、环境准备与数据加载**

1. 环境准备（复用原有基础，补充 SQL 配置）

- 已安装 Spark、Python 环境。

- 确保 `pyspark` 库已安装（与 Spark 版本一致，如 `pip install pyspark==3.0.1`）。

2. Spark SQL 数据加载与视图创建（新增核心步骤）

通过 Spark SQL 加载 CSV 数据，并创建 **临时视图**（用于后续 SQL 查询），核心是定义严格 Schema 提升效率：

**三、核心任务（Spark SQL 实现）**

所有任务均通过 `spark.sql("SQL语句")` 执行，结果保存至 `output` 文件夹，支持 csv格式。

**基础分析（必做）**

**任务 1.1：**统计 4 种行为总次数 + 计算购买转化率

**任务目标**：明确各行为（点击/收藏/加购/购买）的分布，量化“点击→购买”的核心转化效率。

**伪代码**：

1. 从临时视图user\_behavior\_view中，按action字段分组，用COUNT(\*)统计每种行为的总次数，按总次数降序排序，创建临时视图action\_count\_view；

2. 从action\_count\_view中，通过子查询分别提取action = 'buy'的总次数和action = 'click'的总次数，按 “（购买次数 / 点击次数）×100” 计算购买转化率，保留 2 位小数；

3. 将action\_count\_view的结果保存至指定路径，终端输出转化率结果。

**任务 1.2：**分析商品类别热度（按点击量降序）

**任务目标**：识别最受欢迎的商品类别，为“品类运营”提供依据。

**伪代码**：

1. 从临时视图user\_behavior\_view中，筛选action = 'click'的行为数据；
2. 按category字段分组，用COUNT(\*)统计每个品类的点击量，按点击量降序排序，创建临时视图category\_click\_view；
3. 将category\_click\_view的结果保存至指定路径，可输出前 N 条数据查看热门品类。

**任务 1.3：**分析用户活跃时段（按小时统计行为次数）

**任务目标**：识别用户一天中的活跃高峰，指导“限时促销”“客服排班”。

**伪代码**：

1. 从临时视图user\_behavior\_view中，用时间函数将timestamp（Unix 时间戳）转换为 “小时” 格式（如FROM\_UNIXTIME(timestamp, 'HH')）；
2. 按转换后的 “小时” 字段分组，用COUNT(\*)统计每个小时的行为总次数，按小时升序排序，创建临时视图hourly\_behavior\_view；
3. 将hourly\_behavior\_view的结果保存至指定路径，筛选行为次数较高的小时段作为活跃高峰。

**进阶分析（选做，能力提升）**

**任务 2.1：**高价值用户识别（Top3 购买用户）

**任务目标**：找出购买次数最多的 3 个用户，为“高价值用户运营”（如专属权益）提供名单。

**伪代码**：

1. 从临时视图user\_behavior\_view中，筛选action = 'buy'的行为数据；
2. 按user\_id字段分组，用COUNT(\*)统计每个用户的购买次数，按购买次数降序排序；
3. 取排序后的前 3 条数据，创建临时视图top3\_buy\_user\_view，将结果保存至指定路径。

**任务 2.2：**转化漏斗分析（点击→收藏→加购→购买）

**任务目标**：量化漏斗各环节的转化损耗，定位“流失严重的环节”（如收藏→加购转化率低，需优化商品详情页）。

**伪代码**：

1. 行为路径标记：从用户行为视图中，按用户ID和商品ID分组，对每个用户-商品的行为按时间戳排序，用窗口函数标记行为顺序（如第1次行为、第2次行为等），创建临时视图`user\_item\_behavior\_path`，包含字段：user\_id、item\_id、action、timestamp、action\_order。

2. 漏斗有效性过滤：

从`user\_item\_behavior\_path`中，筛选行为类型为click、collect、cart、buy的数据，

仅保留“click（action\_order=1）→collect（action\_order=2）→cart（action\_order=3）→buy（action\_order=4）”的连续行为，

标记每条行为是否属于有效漏斗（is\_valid\_funnel=1表示有效），

创建临时视图`valid\_funnel\_behavior`。

3. 统计各环节独立用户数（仅含有效漏斗行为）：

从`valid\_funnel\_behavior`中，用条件聚合分别统计：

点击环节独立用户数：click\_user = 统计action='click'且is\_valid\_funnel=1的distinct user\_id数量；

收藏环节独立用户数：collect\_user = 统计action='collect'且is\_valid\_funnel=1的distinct user\_id数量；

加购环节独立用户数：cart\_user = 统计action='cart'且is\_valid\_funnel=1的distinct user\_id数量；

购买环节独立用户数：buy\_user = 统计action='buy'且is\_valid\_funnel=1的distinct user\_id数量；

创建临时视图`funnel\_user\_count\_view`。

4. 计算各环节转化率：

从`funnel\_user\_count\_view`中，计算各环节转化率：

点击→收藏转化率：click\_to\_collect = (collect\_user / click\_user) \* 100，保留2位小数；

收藏→加购转化率：collect\_to\_cart = (cart\_user / collect\_user) \* 100，保留2位小数；

加购→购买转化率：cart\_to\_buy = (buy\_user / cart\_user) \* 100，保留2位小数；

整体转化率：overall\_conversion = (buy\_user / click\_user) \* 100，保留2位小数；

创建临时视图`funnel\_conversion\_view`。

5. 保存与输出结果：

将`funnel\_conversion\_view`的结果保存为CSV文件到指定路径（如output/sql\_advanced\_funnel）；

终端输出各环节转化率结果。

**任务 2.3：**商品复购率分析

**任务目标**：统计每个商品的复购率（购买≥2次的用户数/总购买用户数），识别“复购率高的忠诚商品”。

**伪代码**：

1. 从临时视图user\_behavior\_view中，筛选action = 'buy'的行为数据，按item\_id和user\_id分组，用COUNT(\*)统计每个用户对每个商品的购买次数，创建临时视图item\_user\_buy\_count\_view；
2. 从item\_user\_buy\_count\_view中，按item\_id分组：

统计该商品总购买用户数：COUNT(DISTINCT user\_id)；

统计复购用户数（购买次数≥2）：COUNT(DISTINCT CASE WHEN buy\_times >= 2 THEN user\_id END)；

按 “（复购用户数 / 总购买用户数）×100” 计算复购率，保留 2 位小数，按复购率降序排序，创建临时视图item\_repurchase\_view；

1. 将item\_repurchase\_view的结果保存至指定路径。

**任务 2.4：**用户次日留存率

**任务目标**：计算“当天活跃用户在次日仍活跃”的比例，衡量产品对用户的“留存能力”。

**伪代码**：

1. 从临时视图user\_behavior\_view中，提取user\_id和 “timestamp转换后的日期”（如DATE(FROM\_UNIXTIME(timestamp))），去重后得到用户每日活跃记录，创建临时视图user\_daily\_active\_view；
2. 从user\_daily\_active\_view中，用LAG()窗口函数（按user\_id分区、active\_date排序）获取每个用户 “前一天的活跃日期”，创建临时视图user\_prev\_active\_view；
3. 从user\_prev\_active\_view中，按active\_date分组：

统计当天活跃用户数：COUNT(DISTINCT user\_id)；

统计次日留存用户数（前一天活跃且当天仍活跃，即DATEDIFF(active\_date, prev\_active\_date) = 1）；

按 “（留存用户数 / 当天活跃用户数）×100” 计算次日留存率，保留 2 位小数，创建临时视图user\_retention\_view；

1. 将user\_retention\_view的结果保存至指定路径。

**任务2.5：** 分析统计主站最受欢迎的视频

任务目标统计受欢迎的视频TOP 20

### 数据源：[网](https://www.imooc.com/)站访问日志，使用access\_20161111.log作为初始数据，数据为日期,时间,网址，流量,ip地址 ,其中url形如”http://www.imooc.com/video/4500” 其中video 为视频，后面代表的是视频id。

拓展:

以access\_20161111.log为基础创建数据表，比对原生sql查询语句完成任务的时间以及使用sparksql花费时间的差距。

学习项目地址:https://github.com/ptyadana/SQL-Data-Analysis-and-Visualization-Projects