

阿里巴巴口碑商家流量分析系统

1. 问题描述

随着移动定位服务的流行，阿里巴巴和蚂蚁金服逐渐积累了来自用户和商家的海量线上线下交易数据。蚂蚁金服的 O2O 平台“口碑”用这些数据为商家提供了包括交易统计，销售分析和销售建议等定制的后端商业智能服务。举例来说，口碑致力于为每个商家提供销售预测。基于预测结果，商家可以优化运营，降低成本，并改善用户体验。

在这个问题中，我们将提供用户的浏览和支付历史，以及商家相关信息，并希望学员利用所学的 hadoop 与 spark 大数据技术，按照要求完成系大数据系统设计、程序实现和可视化展示部分。

注意：给定的数据为业务情景数据，所有数据均已进行了采样和脱敏处理，字段取值与分布均与真实业务数据不同。

2. 数据说明

我们提供从 2015.07.01 到 2016.10.31（除去 2015.12.12）的商家数据，用户支付行为数据（约 7000w 条）以及用户浏览行为数据（约 500w 条）。提供数据的类型统一为 string 类型，提交预测的类型为整形。文件统一为 utf-8 编码，没有标题行，并以 “,” 分隔的 csv 格式。

数据集下载地址为：链接:<https://pan.baidu.com/s/1geMIy5t> **密码:**p2lm

（1）shop_info：商家特征数据

Field	Sample	Description
shop_id	000001	商家id
city_name	北京	市名
location_id	001	所在位置编号，位置接近的商家具有相同的编号
per_pay	3	人均消费（数值越大消费越高）
score	1	评分（数值越大评分越高）
comment_cnt	2	评论数（数值越大评论数越多）
shop_level	1	门店等级（数值越大门店等级越高）
cate_1_name	美食	一级品类名称
cate_2_name	小吃	二级分类名称
cate_3_name	其他小吃	三级分类名称

(2) user_pay：用户支付行为

Field	Sample	Description
user_id	0000000001	用户id
shop_id	000001	商家id，与shop_info对应
time_stamp	2015-10-10 11:00:00	支付时间

3. user_view：用户浏览行为

Field	Sample	Description
user_id	0000000001	用户id
shop_id	000001	商家id，与shop_info对应
time_stamp	2015-10-10 10:00:00	浏览时间

3. 系统设计与实现要求

最终提交的系统是一个完整的“数据仪表盘”，即由各种可视化图表组成的数据可视化系统，类似于以下系统（这些系统仅是示例，其包含的各种图表跟本项目无关）：



可视化部分建议选择三种方式之一实现：

- (1) 自己采用 echart , D3 等可视化库实现一个可视化系统 (web 系统 , 推荐该方式)
- (2) 使用 airbnb 开源的 superset 定制一个仪表盘
- (3) 使用商业的 tableau (可选择试用版) 定义一个仪表盘

请为该系统起一个名字 , 比如 “ 信贷需求数据仪表盘 ” 、 “ 信贷数据分析系统 ” 等。

请为下面每个任务设计合理的可视化方式 (不一定受限于提示给定的图表 , 可自行修改) , 并将其合理的组织在一个可视化系统中。

请重视可视化系统并考虑如何设计和实现它 , 这是数据价值呈现的最直接方式 。

后面任务部分不仅涉及到只读类似的图表 , 也有交互式的输入界面 (根据用户的输入返回输出结果) , 请考虑将这些可视化和交互式界面有机结合在一个系统中。

4. 数据分析任务

(1) 任务 1

将 shop_info 商家数据导入到 MySQL 中 , user_pay 和 user_view 导入到 HDFS 中。

(2) 任务 2

利用 Sqoop 将 MySQL 中 shop_info 表导入到 HDFS 中

(3) 任务 3

利用 Presto 分析产生以下结果 , 并通过 web 方式可视化 :

- 以城市为单位，统计每个城市总体消费金额（饼状图）
- 以天为单位，统计所有商家交易发生次数和被用户浏览次数（曲线图）
- 统计最受欢迎的前 10 类商品（按照二级分类统计），并输出他们的人均消费（选择合适图表对其可视化，类似排行榜）


（4）任务 4

利用 Spark RDD 或 Spark DataFrame 分析产生以下结果：

- 平均日交易额最大的前 10 个商家，并输出他们各自的交易额，并选择合适的图表对结果进行可视化；
- 输出北京、上海、广州和深圳四个城市最受欢迎的 5 家奶茶商店和中式快餐编号（这两个分别输出出来）
 - 最受欢迎是指以下得分最高： $0.7 \times (\text{平均评分}/5) + 0.3 \times (\text{平均消费金额}/\text{最高消费金额})$ ，注：最高消费金额和平均消费金额是从所有消费记录统计出来的；
 - 并选择合适的图表对结果进行可视化（类似排行榜）。
- 留存分析（不了解留存分析的，可参考：
<https://www.sensorsdata.cn/manual/retention.html>），对于平均日交易额最大的前 3 个商家，对他们进行漏斗分析，以浏览行为作为分析目标，输出 2016.10.01~2016.10.31 共 31 天的留存率，输出为类似以下矩阵（注意表中数值不一定准确，仅用作示例说明），请选择合适的图表进行可视化：

日期	第 0 天	第 1 天	第 2 天	第 3 天	第 4 天	第 5 天	第 6 天	...
2016-10-01	100%	82%	60%	30%	28%	25%	20%	...
2016-10-02	100%	85%	68%	55%	49%	40%	38%	...
...	


注：**第 0 天留存率**表示当天活跃的用户比例（一定是 100%，比如有 1000 人浏览），**第 1 天留存率**表示第 0 天活跃的用户在第一天也活跃的比例（比如前面 1000 人中第 1 天也活跃的用户有 820 人，则留存率为 82%），**第 2 天留存率**表示第 0 天活跃的用户在第 2 天也活跃的比例（比如前面 1000 人中第 2 天也活跃的用户有 600 人，则留存率为 60%），以此类推....

- 给定一个商店（可动态指定），输出该商店每天、每周和每月的被浏览数量，并选择合适的图表对结果进行可视化；
- 找到被浏览次数最多的 50 个商家，并输出他们的城市以及人均消费，并选择合适的图表对结果进行可视化。

（5）任务 5：流式分析


利用 spark streaming 分析所有商家实时交易数据：

- 将 shop_info 存放在 mysql 中（任务 1 已做）

- 编写 spark 程序，将 user_pay 数据依次写入 kafka 中的 user_pay 主题中，每条数据写入间隔为 10 毫秒，其中 user_id 为 key，shop_id+“,”+time_stamp 为 value
- 编写 spark streaming 程序，依次读取 kafka 中 user_pay 主题数据，并统计：
 - 每个商家实时交易次数，并存入 redis，其中 key 为“jiaoyi+<shop_id>”，value 为累计的次数，选择合适的图表对结果实时可视化（为了方便可视化，可选择 mysql 而不是 redis，如果使用 mysql，请自行设计表结构）；
 - 每个城市发生的交易次数，并存储 redis，其中 key 为“交易+<城市名称>”，value 为累计的次数，选择合适的图表对结果实时可视化（为了方便可视化，可选择 mysql 而不是 redis，请自行设计表结构）。

（6）任务 6：历史账单查询

实现毫秒级历史支付账单查询：给定任意用户 ID 以及时间区间（比如 2015.07.01~2015.07.10），输出该时间内该用户所有支付信息（包括商家 ID、商家所在城市、支付金额等），要求给出的方案支持动态模式修改（数据列不断增加和修改），海量数据存储（至少 10TB 规模）以及毫秒级查询和数据修改。

- 需要实现毫秒级查询信息（不能超过 1 秒）（提示，可尝试使用 HBase）
- 请编写交互式图形界面更加友好地完成查询 
- （可选）设计 REST API，给定用户 ID 及时间区间，返回对应所有标签（json 格式），并能支撑 1k RPS（每秒 1000 个并发访问请求）

(7) 任务 7：用户和商家画像系统

- 用户画像系统：给定用户的 ID，毫秒级返回该用户对应的所有标签以及标签对应的值，标签包括（一般标签数目会超过万级别，而且实时更新，本项目只给出几个）：
 - last_7_day_review：过去 7 天浏览店铺数目
 - last_1_month_review：过去 1 个月浏览店铺数目
 - last_3_month_review：过去 3 个月浏览店铺数目
 - last_7_day_pay：过去 7 天支付笔数
 - last_1_month_pay：过去 1 个月支付笔数
 - last_3_month_pay：过去 3 个月支付笔数
 - user_city：用户所在城市（把访问过最多商家所在城市作为用户城市）
- 商家画像系统：给定商家的 ID，毫秒级返回该商家对应的所有标签以及标签对应的值，标签包括：
 - last_7_day_reviewed：过去 7 天被浏览店的数目
 - last_1_month_reviewed：过去 1 个月被浏览的数目
 - last_3_month_reviewed：过去 3 个月被浏览的数目
 - last_7_day_payed：过去 7 天被支付的笔数
 - last_1_month_payed：过去 1 个月被支付的笔数
 - last_3_month_payed：过去 3 个月被支付的笔数
 - score：评分

这些标签更新间隔是 1 天，请给出存储和标签更新方案（提示：可采用 hbase 存储标签，使用 spark 更新标签），要求给出的方案支持动态模式修改（标签不断增加和修改），海量数据存储（至少 10TB 规模）以及毫秒级查询和数据修改。

（可选）设计 REST API，给定用户或商家 ID，返回对应所有标签（json 格式），并能支撑 1k RPS（每秒 1000 请求）

5. 性能优化

（1）请在任务 3 和任务 4 中各挑选一个性能较低（运行时间较长）的子任务，并给出几种性能优化的方法（可从存储方式、算子优化、参数调优等方面考虑），包括：

- 性能超过 10%的方法
- 性能超过 50%的方法
- 性能超过 1 倍甚至几倍的方法

并给出采用具体的优化方法后性能提升的比例。

（2）请给出历史账单查询子系统的优化方法，并给出优化后的性能提升比例。