|  |
| --- |
| 大数据课程设计  1亿条淘宝用户行为数据分析  实验报告  **指导教师：付燕宁 王岩**  **年 级：2020**  **班 级：软件8班**  **小组编号：8组**  **组长学号姓名：55200831 王逸飞 30% 100**  **组员学号姓名：55200813 郭子豪 30% 95**  **组员学号姓名：55200815姚子轩 10% 70**  **组员学号姓名：55200816姜思远 10% 70**  **组员学号姓名：55200821姚晟荣 10% 70**  **组员学号姓名：55200822王一凡 10% 70**  **2023年6月14日**  **吉林大学软件学院** |

|  |  |  |
| --- | --- | --- |
| **完成实验内容** | | |
| 本项目使用阿里云公开数据集，基于“清洗 hive + 分析 hive + 可视化 echarts”的技术栈，完成了1 亿条淘宝用户的行为数据分析。 | | |
| **小组成员任务完成情况** | | |
| 姓名 | 具体完成任务 | 工作量  百分比 |
| 王逸飞 | 数据存储管理（hadoop生态的完全分布式+查询）、数据分析挖掘（数据挖掘算法） | 30% |
| 郭子豪 | 基于echarts的图形化展示 | 30% |
| 姚子轩 | 数据采集（数据源+整理读写等） | 10% |
| 姜思远 | 完全分布式 Hadoop安装+hive安装 | 10% |
| 姚晟荣 | PPT制作 | 10% |
| 王一凡 | 报告整理 | 10% |
| **小组成员协作情况** | | |
| 1. **组长: 王逸飞**   在这个数据分析项目中，我具体承担了数据存储管理和数据分析挖掘的任务。  首先，作为数据存储管理的责任人，我的任务是建立一个基于Hadoop生态的完全分布式数据存储和查询系统。我的思路是搭建一个Hadoop集群，并使用HDFS来存储大规模的用户行为数据。我在集群中设置了适当的副本数以确保数据的可靠性和容错性。此外，我使用Hive作为数据仓库，通过定义表结构和分区方式来优化数据查询性能。在数据加载过程中，我使用了Hive的内置函数和语句来进行数据清洗、去重和格式转换。  在数据分析挖掘方面的任务中，我的目标是使用数据挖掘算法来揭示淘宝用户行为的规律和趋势。我的思路是首先进行用户行为统计，包括总访问量、总用户量、每日平均访问量和平均用户量等指标。然后，我计算了不同用户行为类型的转化率，例如点击转化为收藏或加购物车的转化率，以及加购物车或收藏转化为购买的转化率。接下来，我进行了用户行为习惯的分析，包括活跃时段和一周内的活跃分布。  在任务完成过程中，我遇到了一些问题。例如，数据存储管理方面面临集群的配置和调优问题，以及数据加载和清洗过程中的性能瓶颈。在数据分析挖掘方面，会遇到算法选择、模型训练和结果解释等方面的挑战。  为了解决这些问题，我可以参考Hadoop和Hive的官方文档、在线资源和社区讨论，寻求相关的配置和优化建议。对于数据挖掘算法的选择和模型训练，我可以进行综合评估和实验，并根据业务需求和分析目标选择合适的算法和模型。同时，我可以利用可视化工具和图表来呈现分析结果，以便更好地解释和传达分析发现。  最终，我的工作结果是一个稳定、高效的数据存储和查询系统，能够处理大规模的用户行为数据，并提供准确、可靠的查询结果。通过数据挖掘算法的应用，我揭示了淘宝用户行为的规律和趋势，并提供了有关用户流量、转化率和行为习惯的详细分析报告。这些结果为淘宝平台的运营决策和优化提供了重要的参考依据，并为用户行为分析领域的进一步研究提供了有价值的数据资源。   1. **组员: 郭子豪**   在这个数据分析项目中，我具体承担了使用 ECharts 对别人的数据分析结果进行图形化展示的任务。下面是关于我任务完成的具体化的介绍，包括任务承担的内容、思路、遇到的问题以及解决方法，以及最终的工作结果。  首先，我的任务是将他人已完成的数据分析结果转化为可视化图形，并利用 ECharts 进行展示。我的思路是通过与数据分析团队沟通，了解他们的分析报告和结果。然后，我使用 ECharts 提供的图表库和功能，根据数据分析结果的特点和目标受众的需求，选择适当的图表类型来展示数据。我着重考虑图表的清晰度、易读性和美观性，以确保最终的展示效果能够有效传达分析结果。  在任务完成的过程中，我会遇到一些问题。其中包括以下几个方面：  数据理解和解读：在将数据分析结果转化为图形展示之前，我需要充分理解数据的含义和分析的结果。这涉及到与数据分析团队的深入交流和解读，以确保我准确理解他们的分析逻辑和结论。  图表类型选择：不同类型的数据适合不同的图表展示方式。我需要根据数据的性质和分析目的，选择合适的图表类型，如折线图、柱状图、饼图等。这需要我对 ECharts 的图表库有一定的了解，并根据数据的特点做出合理的选择。  数据处理和预处理：在进行图形化展示之前，我需要对数据进行一些处理和预处理。这涉及数据清洗、筛选、聚合等操作，以确保数据的准确性和一致性。我可以使用 ECharts 提供的数据处理功能，如数据过滤、排序、格式转换等，来完成这些操作。  为了解决这些问题，我可以与数据分析团队保持密切的沟通和协作，确保对数据的理解一致。我还可以利用 ECharts 官方文档和在线资源，学习和掌握不同图表类型的使用方法，并参考示例代码和案例进行实践和调试。在数据处理和预处理方面，我可以运用 ECharts 提供的功能和方法，结合数据分析的需求，灵活处理数据，使其适应图表展示的要求。  最终，我的工作结果是一个基于 ECharts 的图形化展示页面，通过清晰、美观的图表，将别人的数据分析结果直观地呈现给目标受众。通过图形化展示，我使得数据分析结果更易于理解和解读，帮助目标受众更好地把握数据的洞察和趋势。同时，我的工作结果还为数据分析团队提供了一个直观的展示平台，便于他们与他人分享和交流分析成果。   1. **组员: 姚子轩**   数据采集任务简述：  任务名称：数据采集任务  任务描述：从 https://tianchi.aliyun.com/dataset/649 网站上采集数据  任务目标：获取网站上的相关数据用于后续分析和处理  任务步骤：  网站分析：仔细阅读 https://tianchi.aliyun.com/dataset/649 网站页面的内容，了解数据的结构、格式和获取方式。注意查看是否需要注册或登录才能访问数据。  数据获取方式：确定数据获取方式。网站上的数据可能以文件、API接口或其他形式提供。如果是文件下载，注意查看下载链接或按钮的位置，以便后续下载数据。  数据下载：使用合适的工具或编程语言，通过访问下载链接或使用相关API接口，下载所需的数据文件。根据网站的要求，可能需要提供访问令牌、身份验证或其他参数。  数据存储：选择适当的数据存储方式，例如本地文件系统或数据库，将下载的数据保存到指定的存储位置。确保数据的完整性和安全性。  数据清洗和处理（可选）：根据任务需求，对下载的数据进行清洗和处理。这可能包括去除重复数据、修复错误、格式转换等操作，以使数据符合后续分析的要求。  数据验证：验证下载的数据是否完整且准确。可以检查数据文件的大小、记录数、字段结构等来确保数据的完整性。  任务文档化：记录任务相关的详细信息，包括数据来源、获取方式、存储位置、数据清洗步骤等。这有助于日后的复查和共享任务信息。  注意事项：  遵守网站的使用条款和法律法规，确保数据采集过程的合法性。  注意网站的访问频率限制和数据获取的限制，避免对网站造成过大的负载或违反使用规定。  如果有使用第三方工具或库进行数据采集的情况，遵守相应工具或库的许可证和使用规定。   1. **组员: 姜思远**   要进行完全分布式的Hadoop安装，您需要进行以下步骤：  1. 准备环境：  - 购买或准备一组适当配置的物理或虚拟机器，这些机器将构成Hadoop集群。  - 每台机器上都应该安装一个兼容的操作系统，如Linux（例如Ubuntu、CentOS等）。  2. 安装Java：  - Hadoop是用Java编写的，所以您需要在每台机器上安装Java运行时环境（JRE）或Java开发工具包（JDK）。  3. 配置SSH：  - 为了能够在集群的不同节点之间进行通信，您需要在每台机器上设置SSH密钥认证。  - 确保在集群的每台机器上都可以通过SSH互相访问。  4. 下载和解压缩Hadoop：  - 从Apache Hadoop官方网站（http://hadoop.apache.org）下载最新的稳定版本。  - 将下载的文件解压缩到您选择的目录。  5. 配置Hadoop集群：  - 进入Hadoop配置目录，通常是`$HADOOP\_HOME/etc/hadoop`。  - 编辑`hadoop-env.sh`文件，设置JAVA\_HOME变量为您的Java安装路径。  - 编辑`core-site.xml`文件，设置Hadoop的核心配置，如文件系统URI和临时目录等。  - 编辑`hdfs-site.xml`文件，设置Hadoop分布式文件系统（HDFS）的配置，如副本数量、数据块大小等。  - 编辑`mapred-site.xml`文件，设置MapReduce的配置，如任务跟踪器和任务分配器等。  - 编辑`yarn-site.xml`文件，设置YARN资源管理器的配置，如节点管理器和应用程序管理器等。  - 如果您有更多的节点，复制以上文件到每个节点上，并根据需要进行相应的修改。  6. 配置主节点和从节点：  - 在主节点上，编辑`masters`文件，将主节点的主机名或IP地址添加到文件中。  - 在从节点上，编辑`slaves`文件，将从节点的主机名或IP地址添加到文件中，每行一个。  7. 分发Hadoop文件：  - 使用Hadoop提供的脚本将Hadoop文件分发到集群的所有节点上。  - 在Hadoop安装目录下，运行以下命令来分发文件：  ```  $ bin/hadoop namenode -format  $ sbin/start-dfs.sh  $ sbin/start-yarn.sh  ```  8. 验证安装：  - 打开Web浏览器，访问Hadoop的Web界面，通常是主节点的IP地址加上端口号（例如http://<主节点IP>:50070）。  - 您应该能够看到Hadoop的管理界面，其中包含有关HDFS  要安装Hive，您可以按照以下步骤进行操作：  1. 准备环境：  - 确保已经在系统上安装了Java，并设置了JAVA\_HOME环境变量。  - 确保Hadoop已经正确安装和配置。  2. 下载Hive：  - 从Apache Hive官方网站（https://hive.apache.org）下载最新的稳定版本。  - 将下载的文件解压缩到您选择的目录。  3. 配置Hive：  - 进入Hive配置目录，通常是`$HIVE\_HOME/conf`。  - 复制`hive-default.xml.template`文件并重命名为`hive-site.xml`。  - 编辑`hive-site.xml`文件，设置Hive的配置，如Hadoop的相关配置、数据库连接等。  - 如果需要使用MySQL或其他数据库作为Hive的元数据存储，确保您已经安装了相应的数据库，并将数据库连接配置正确填写在`hive-site.xml`文件中。  4. 配置Hive环境变量：  - 打开`~/.bashrc`或`~/.bash\_profile`文件，并添加以下行（根据您的安装目录进行相应修改）：  ```  export HIVE\_HOME=/path/to/hive  export PATH=$PATH:$HIVE\_HOME/bin  ```  - 执行`source ~/.bashrc`或`source ~/.bash\_profile`命令，使环境变量生效。  5. 初始化Hive元数据存储：  - 运行以下命令来初始化Hive的元数据存储（前提是您已经正确配置了`hive-site.xml`中的数据库连接）：  ```  $ schematool -dbType <database\_type> -initSchema  ```  - `<database\_type>`是您使用的数据库类型，如`mysql`、`derby`等。  6. 启动Hive：  - 运行以下命令来启动Hive的命令行界面：  ```  $ hive  ```  - 如果一切顺利，您将看到Hive的命令行提示符。  7. 验证安装：  - 在Hive命令行界面下，可以尝试运行一些Hive查询来验证安装是否成功。  - 例如，您可以运行以下查询来创建一个表并插入一些数据：  ```  hive> CREATE TABLE my\_table (id INT, name STRING);  hive> INSERT INTO my\_table VALUES (1, 'Alice'), (2, 'Bob');  hive> SELECT \* FROM my\_table;  ```  - 如果查询成功执行并返回结果，则说明Hive安装成功。  这样，您就完成了Hive的安装过程。请注意，Hive需要与Hadoop集成使用，因此确保Hadoop已正确配置并运行。   1. **组员: 姚晟荣**   任务具体化：PPT制作  完成任务的思路：  1. 确定PPT的内容，包括需要呈现的数据和图表。  2. 设计PPT的模板和风格，选择合适的颜色、字体和图片。  3. 制作PPT的幻灯片，按照逻辑顺序排列内容，注意视觉效果和文字表述。  4. 检查和修改PPT，确保内容准确、清晰、易懂。  遇见的问题及解决：  1. 制作PPT的幻灯片需要较高的技术水平和耐心。  2. 检查和修改PPT需要较高的认真程度和细致性。  工作结果：   1. 制作出一份高质量的PPT，呈现出清晰、生动、有趣的数据和图表。 2. **组员: 王一凡**   一、承担的任务  汇总小组成员的研究成果：收集并整理小组成员在大数据课程中的研究成果，包括数据分析、模型构建、实验结果等。  组织报告结构：设计一个清晰、合理的报告结构，以便于读者更好地理解我们的工作。  编写报告正文：撰写报告正文，详细介绍我们的研究方法、实验过程和结果分析。  审核和修改报告：在对报告进行审核，确保内容准确无误，并根据小组成员的意见进行修改。  二、任务完成的思路  沟通与协作：与小组成员保持密切沟通，了解他们的研究进展，并提供必要的支持。  分析与整理：对其进行了详细的分析和整理，以便于在报告中清晰地展示。  逻辑与条理：注重报告的逻辑性和条理性，确保报告内容连贯、易于理解。  三、遇到的问题及解决方法  问题：部分小组成员的成果格式不统一，导致整理工作困难。  解决方法：制定了统一的格式要求，并与小组成员沟通，要求他们按照统一格式提交研究成果。  问题：报告审核过程中发现部分内容描述不清晰。  解决方法：与相关小组成员沟通，要求他们提供更详细的解释，并对报告进行修改。  四、工作结果：  统一的研究成果格式：通过与小组成员沟通和协作，实现了研究成果格式的统一，使得报告内容更加规范和整洁。  清晰的报告结构：根据小组成员的研究成果设计了一个清晰、合理的报告结构，使得报告内容条理分明，便于读者理解。  详实的报告正文：撰写了详实的报告正文，全面介绍了我们的研究方法、实验过程和结果分析。  高质量的报告呈现：确保了报告内容的准确性和可读性，提高了报告的整体质量。 | | |
| **数据分析主要过程** | | |
| 1. **使用Hive进行数据处理** 2. 数据导入   将数据加载到 hive, 然后通过 hive 对数据进行数据处理。  -- 建表  drop table if exists user\_behavior;  create table user\_behavior (  `user\_id` string comment '用户ID',  `item\_id` string comment '商品ID',  `category\_id` string comment '商品类目ID',  `behavior\_type` string comment '行为类型，枚举类型，包括(pv, buy, cart, fav)',  `timestamp` int comment '行为时间戳',  `datetime` string comment '行为时间')  row format delimited  fields terminated by ','  lines terminated by '\n';  -- 加载数据  LOAD DATA LOCAL INPATH '/home/getway/UserBehavior.csv'  OVERWRITE INTO TABLE user\_behavior ;   1. 数据清洗   数据处理主要包括：删除重复值，时间戳格式化，删除异常值。  --数据清洗，去掉完全重复的数据  insert overwrite table user\_behavior  select user\_id, item\_id, category\_id, behavior\_type, timestamp, datetime  from user\_behavior  group by user\_id, item\_id, category\_id, behavior\_type, timestamp, datetime;  --数据清洗，时间戳格式化成 datetime  insert overwrite table user\_behavior  select user\_id, item\_id, category\_id, behavior\_type, timestamp, from\_unixtime(timestamp, 'yyyy-MM-dd HH:mm:ss')  from user\_behavior;  --查看时间是否有异常值  select date(datetime) as day from user\_behavior group by date(datetime) order by day;  --数据清洗，去掉时间异常的数据  insert overwrite table user\_behavior  select user\_id, item\_id, category\_id, behavior\_type, timestamp, datetime  from user\_behavior  where cast(datetime as date) between '2017-11-25' and '2017-12-03';  --查看 behavior\_type 是否有异常值  select behavior\_type from user\_behavior group by behavior\_type;   1. **使用Hive进行数据分析** 2. 用户流量及购物情况   \* 小结：2017-11-25 到 2017-12-03 这段时间，PV 总数为 89,660,671 ，UV 总数为 987,991。从日均访问量趋势来看，进入 12 月份之后有一个比较明显的增长，猜测可能是因为临近双 12 ，电商活动引流产生，另外，2017-12-02 和 2017-12-03 刚好是周末，也可能是周末的用户活跃度本来就比平常高。总体的复购率为 66.01%，说明用户的忠诚度比较高。   1. 用户行为转换率   \* 小结：2017-11-25 到 2017-12-03 这段时间，点击数为 89,660,671 ，收藏数为 2,888,258，加购物车数为5,530,446，购买数为 2,015,807。总体的转化率为 2.25%，这个值可能是比较低的，从加到购物车数来看，有可能部分用户是准备等到电商节日活动才进行购买。所以合理推断：一般电商节前一段时间的转化率会比平常低。   1. 用户行为习惯   \* 小结：晚上21点-22点之间是用户一天中最活跃的时候，凌晨 4 点，则是活跃度最低的时候。一周中，工作日活跃度都差不多，到了周末活跃度有明显提高。   1. 基于 RFM 模型找出有价值的用户   RFM 模型是衡量客户价值和客户创利能力的重要工具和手段，其中由3个要素构成了数据分析最好的指标，分别是：  \* R-Recency（最近一次购买时间）  \* F-Frequency（消费频率）  \* M-Money（消费金额）  \* 小结：可以根据用户的价值得分，进行个性化的营销推荐。   1. 商品维度的分析   \* 小结：缺失商品维表，所以没有太多分析价值。假如有商品维表，可以再展开，以商品纬度进行分析，比如不同行业、不同产品的转化率，还有竞品分析等等。   1. **使用Echarts进行数据可视化** | | |
| **实验中出现的问题及解决方案** | | |
| 1. 数据质量问题：   问题描述：数据质量是进行数据分析和图形化展示的基础，但数据存在缺失、错误或不一致等问题，影响结果的准确性和可靠性。  解决方案：  数据清洗：对原始数据进行清洗，去除重复、缺失和异常值。  数据转换：对数据进行规范化和标准化处理，确保数据一致性。  数据验证：验证数据的准确性和完整性，比较与其他数据源的一致性。   1. 数据安全和隐私问题：   问题描述：在进行数据分析和展示时，必须保护数据的安全性和隐私，尤其涉及敏感信息时需要遵守相关法规和隐私政策。  解决方案：  数据脱敏：对敏感数据进行脱敏处理，保护用户隐私。  访问权限控制：采用权限管理机制，限制只有授权人员可以访问和操作数据。  加密传输：使用加密技术保证数据在传输过程中的安全性。   1. 技术兼容性问题：   问题描述：使用ECharts进行图形化展示时，会遇到与数据源、开发环境或其他技术组件的兼容性问题。  解决方案：  技术评估和测试：在项目开始之前，进行技术评估和测试，确保各组件之间的兼容性。  更新和调整组件：及时更新和调整相关的技术组件，以解决兼容性问题。   1. 沟通和合作问题：   问题描述：在项目部署过程中，需要与数据分析团队、开发团队和其他相关人员进行有效的沟通和合作，以确保任务的顺利完成。  解决方案：  沟通机制：建立定期开会、交流进展的沟通机制，确保各方的需求和期望得到充分理解。  协作工具：使用协作工具和项目管理平台，共享信息、任务和文档，提高团队合作效率。 | | |
| **已完成部分的源程序关键代码** | | |
| 1. **使用Hive进行数据处理** 2. 数据导入   将数据加载到 hive, 然后通过 hive 对数据进行数据处理。  -- 建表  drop table if exists user\_behavior;  create table user\_behavior (  `user\_id` string comment '用户ID',  `item\_id` string comment '商品ID',  `category\_id` string comment '商品类目ID',  `behavior\_type` string comment '行为类型，枚举类型，包括(pv, buy, cart, fav)',  `timestamp` int comment '行为时间戳',  `datetime` string comment '行为时间')  row format delimited  fields terminated by ','  lines terminated by '\n';  -- 加载数据  LOAD DATA LOCAL INPATH '/home/getway/UserBehavior.csv'  OVERWRITE INTO TABLE user\_behavior ;   1. 数据清洗   数据处理主要包括：删除重复值，时间戳格式化，删除异常值。  --数据清洗，去掉完全重复的数据  insert overwrite table user\_behavior  select user\_id, item\_id, category\_id, behavior\_type, timestamp, datetime  from user\_behavior  group by user\_id, item\_id, category\_id, behavior\_type, timestamp, datetime;  --数据清洗，时间戳格式化成 datetime  insert overwrite table user\_behavior  select user\_id, item\_id, category\_id, behavior\_type, timestamp, from\_unixtime(timestamp, 'yyyy-MM-dd HH:mm:ss')  from user\_behavior;  --查看时间是否有异常值  select date(datetime) as day from user\_behavior group by date(datetime) order by day;  --数据清洗，去掉时间异常的数据  insert overwrite table user\_behavior  select user\_id, item\_id, category\_id, behavior\_type, timestamp, datetime  from user\_behavior  where cast(datetime as date) between '2017-11-25' and '2017-12-03';  --查看 behavior\_type 是否有异常值  select behavior\_type from user\_behavior group by behavior\_type;  **二、 使用Hive进行数据分析**   1. 用户流量及购物情况   --总访问量PV，总用户量UV  select sum(case when behavior\_type = 'pv' then 1 else 0 end) as pv,  count(distinct user\_id) as uv  from user\_behavior;  --日均访问量，日均用户量  select cast(datetime as date) as day,  sum(case when behavior\_type = 'pv' then 1 else 0 end) as pv,  count(distinct user\_id) as uv  from user\_behavior  group by cast(datetime as date)  order by day;  --每个用户的购物情况，加工到 user\_behavior\_count  create table user\_behavior\_count as  select user\_id,  sum(case when behavior\_type = 'pv' then 1 else 0 end) as pv, --点击数  sum(case when behavior\_type = 'fav' then 1 else 0 end) as fav, --收藏数  sum(case when behavior\_type = 'cart' then 1 else 0 end) as cart, --加购物车数  sum(case when behavior\_type = 'buy' then 1 else 0 end) as buy --购买数  from user\_behavior  group by user\_id;  --复购率：产生两次或两次以上购买的用户占购买用户的比例  select sum(case when buy > 1 then 1 else 0 end) / sum(case when buy > 0 then 1 else 0 end)  from user\_behavior\_count;  \* 小结：2017-11-25 到 2017-12-03 这段时间，PV 总数为 89,660,671 ，UV 总数为 987,991。从日均访问量趋势来看，进入 12 月份之后有一个比较明显的增长，猜测可能是因为临近双 12 ，电商活动引流产生，另外，2017-12-02 和 2017-12-03 刚好是周末，也可能是周末的用户活跃度本来就比平常高。总体的复购率为 66.01%，说明用户的忠诚度比较高。   1. 用户行为转换率   --点击/(加购物车+收藏)/购买 , 各环节转化率  select a.pv,  a.fav,  a.cart,  a.fav + a.cart as `fav+cart`,  a.buy,  round((a.fav + a.cart) / a.pv, 4) as pv2favcart,  round(a.buy / (a.fav + a.cart), 4) as favcart2buy,  round(a.buy / a.pv, 4) as pv2buy  from(  select sum(pv) as pv, --点击数  sum(fav) as fav, --收藏数  sum(cart) as cart, --加购物车数  sum(buy) as buy --购买数  from user\_behavior\_count  ) as a;  \* 小结：2017-11-25 到 2017-12-03 这段时间，点击数为 89,660,671 ，收藏数为 2,888,258，加购物车数为5,530,446，购买数为 2,015,807。总体的转化率为 2.25%，这个值可能是比较低的，从加到购物车数来看，有可能部分用户是准备等到电商节日活动才进行购买。所以合理推断：一般电商节前一段时间的转化率会比平常低。   1. 用户行为习惯   -- 一天的活跃时段分布  select hour(datetime) as hour,  sum(case when behavior\_type = 'pv' then 1 else 0 end) as pv, --点击数  sum(case when behavior\_type = 'fav' then 1 else 0 end) as fav, --收藏数  sum(case when behavior\_type = 'cart' then 1 else 0 end) as cart, --加购物车数  sum(case when behavior\_type = 'buy' then 1 else 0 end) as buy --购买数  from user\_behavior  group by hour(datetime)  order by hour;  --一周用户的活跃分布  select pmod(datediff(datetime, '1920-01-01') - 3, 7) as weekday,  sum(case when behavior\_type = 'pv' then 1 else 0 end) as pv, --点击数  sum(case when behavior\_type = 'fav' then 1 else 0 end) as fav, --收藏数  sum(case when behavior\_type = 'cart' then 1 else 0 end) as cart, --加购物车数  sum(case when behavior\_type = 'buy' then 1 else 0 end) as buy --购买数  from user\_behavior  where date(datetime) between '2017-11-27' and '2017-12-03'  group by pmod(datediff(datetime, '1920-01-01') - 3, 7)  order by weekday;  \* 小结：晚上21点-22点之间是用户一天中最活跃的时候，凌晨 4 点，则是活跃度最低的时候。一周中，工作日活跃度都差不多，到了周末活跃度有明显提高。   1. 基于 RFM 模型找出有价值的用户   RFM 模型是衡量客户价值和客户创利能力的重要工具和手段，其中由3个要素构成了数据分析最好的指标，分别是：  \* R-Recency（最近一次购买时间）  \* F-Frequency（消费频率）  \* M-Money（消费金额）  --R-Recency（最近一次购买时间）, R值越高，一般说明用户比较活跃  select user\_id,  datediff('2017-12-04', max(datetime)) as R,  dense\_rank() over(order by datediff('2017-12-04', max(datetime))) as R\_rank  from user\_behavior  where behavior\_type = 'buy'  group by user\_id  limit 10;  --F-Frequency（消费频率）, F值越高，说明用户越忠诚  select user\_id,  count(1) as F,  dense\_rank() over(order by count(1) desc) as F\_rank  from user\_behavior  where behavior\_type = 'buy'  group by user\_id  limit 10;  --M-Money（消费金额），数据集无金额，所以就不分析这一项  对有购买行为的用户按照排名进行分组，共划分为5组，  前 - 1/5 的用户打5分  前 1/5 - 2/5 的用户打4分  前 2/5 - 3/5 的用户打3分  前 3/5 - 4/5 的用户打2分  前 4/5 - 的用户打1分  按照这个规则分别对用户时间间隔排名打分和购买频率排名打分，最后把两个分数合并在一起作为该名用户的最终评分  with cte as(  select user\_id,  datediff('2017-12-04', max(datetime)) as R,  dense\_rank() over(order by datediff('2017-12-04', max(datetime))) as R\_rank,  count(1) as F,  dense\_rank() over(order by count(1) desc) as F\_rank  from user\_behavior  where behavior\_type = 'buy'  group by user\_id)  select user\_id, R, R\_rank, R\_score, F, F\_rank, F\_score, R\_score + F\_score AS score  from(  select \*,  case ntile(5) over(order by R\_rank) when 1 then 5  when 2 then 4  when 3 then 3  when 4 then 2  when 5 then 1  end as R\_score,  case ntile(5) over(order by F\_rank) when 1 then 5  when 2 then 4  when 3 then 3  when 4 then 2  when 5 then 1  end as F\_score  from cte  ) as a  order by score desc  limit 20;  \* 小结：可以根据用户的价值得分，进行个性化的营销推荐。   1. 商品维度的分析   --销量最高的商品  select item\_id ,  sum(case when behavior\_type = 'pv' then 1 else 0 end) as pv, --点击数  sum(case when behavior\_type = 'fav' then 1 else 0 end) as fav, --收藏数  sum(case when behavior\_type = 'cart' then 1 else 0 end) as cart, --加购物车数  sum(case when behavior\_type = 'buy' then 1 else 0 end) as buy --购买数  from user\_behavior  group by item\_id  order by buy desc  limit 10;  --销量最高的商品大类  select category\_id ,  sum(case when behavior\_type = 'pv' then 1 else 0 end) as pv, --点击数  sum(case when behavior\_type = 'fav' then 1 else 0 end) as fav, --收藏数  sum(case when behavior\_type = 'cart' then 1 else 0 end) as cart, --加购物车数  sum(case when behavior\_type = 'buy' then 1 else 0 end) as buy --购买数  from user\_behavior  group by category\_id  order by buy desc  limit 10;  \* 小结：缺失商品维表，所以没有太多分析价值。假如有商品维表，可以再展开，以商品纬度进行分析，比如不同行业、不同产品的转化率，还有竞品分析等等。 | | |