2022年全国职业院校技能大赛

高职组

“大数据技术与应用”

赛项赛卷（5卷）

任

务

书

参赛队编号：

**背景描述**

大数据时代背景下，电商经营模式发生很大改变。在传统运营模式中，缺乏数据积累，人们在做出一些决策行为过程中，更多是凭借个人经验和直觉，发展路径比较自我封闭。而大数据时代，为人们提供一种全新的思路，通过大量的数据分析得出的结果将更加现实和准确。商家可以对客户的消费行为信息数据进行收集和整理，比如消费者购买产品的花费、选择产品的渠道、偏好产品的类型、产品回购周期、购买产品的目的、消费者家庭背景、工作和生活环境、个人消费观和价值观等。通过数据追踪，知道顾客从哪儿来，是看了某网站投放的广告还是通过朋友推荐链接，是新访客还是老用户，喜欢浏览什么产品，购物车有无商品，是否清空，还有每一笔交易记录，精准锁定一定年龄、收入、对产品有兴趣的顾客，对顾客进行分组、标签化，通过不同标签组合运用，获得不同目标群体，以此开展精准推送。

因数据驱动的零售新时代已经到来，没有大数据，我们无法为消费者提供这些体验，为完成电商的大数据分析工作，你所在的小组将应用大数据技术，以Scala作为整个项目的基础开发语言，基于大数据平台综合利用Spark、Flink、Vue.js等技术，对数据进行处理、分析及可视化呈现，你们作为该小组的技术人员，请按照下面任务完成本次工作。

## 模块A：大数据平台搭建（容器环境）（15分）

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各模块服务端说明。**  **补充说明：**宿主机可通过Asbru工具或SSH客户端进行SSH访问；  相关软件安装包在宿主机的/opt目录下，请选择对应的安装包进行安装，用不到的可忽略；  所有模块中应用命令必须采用绝对路径；  进入Master节点的方式为  docker exec –it master /bin/bash  进入Slave1节点的方式为  docker exec –it slave1 /bin/bash  进入Slave2节点的方式为  docker exec –it slave2 /bin/bash  MySQL已在容器的Master中安装完毕，用户名/密码为root/123456 |

### 任务一：Hadoop 伪分布式安装配置

本环节需要使用root用户完成相关配置，安装Hadoop需要配置前置环境。命令中要求使用绝对路径，具体要求如下:

1. 从宿主机/opt目录下将文件hadoop-2.7.7.tar.gz、jdk-8u212-linux-x64.tar.gz复制到容器master中的/opt/software路径中（若路径不存在，则需新建），将容器master节点JDK安装包解压到/opt/module路径中(若路径不存在，则需新建)，将JDK解压命令复制并粘贴至对应报告中;
2. 修改容器/etc/profile文件，设置JDK环境变量并使其生效，配置完毕后在Master节点分别执行“java -version”和“javac”命令，将命令行执行结果分别截图并粘贴至对应报告中;
3. 请完成host相关配置，将三个节点分别命名为master、slave1、slave2，并做免密登录，使用ssh命令，从master免密登录slave1中，将ssh免密登录命令与执行结果截图并粘贴至对应报告中;
4. 在Master将Hadoop解压到/opt/module(若路径不存在，则需新建)目录下，完成相关配置，初始化hdfs，将初始化命令文字及初始化结果截图（截取初始化结果结束倒数20行即可）粘贴至对应报告中；
5. 启动Hadoop集群（包括hdfs和yarn），使用jps命令查看master节点Java进程，将查看结果截图粘贴至对应报告中。

### 任务二：Flume安装配置

本环节需要使用root用户完成相关配置，已安装Hadoop及需要配置前置环境，具体要求如下：

1. 从宿主机/opt目录下将文件apache-flume-1.7.0-bin.tar.gz复制到容器master中的/opt/software路径中（若路径不存在，则需新建），将Master节点Flume安装包解压到/opt/module目录下，将解压命令复制并粘贴至对应报告中；
2. 完善相关配置设置，Hive环境变量，并使环境变量生效，执行命令flume-ng version并将命令与结果截图粘贴至对应报告中；
3. 启动Flume传输Hadoop日志（namenode或datanode日志），查看HDFS中/tmp/flume目录下生成的内容，将查看命令及结果（至少5条结果）截图并粘贴至对应报告中。

### 任务三：Sqoop安装配置

本环节需要使用root用户完成相关配置，已安装Hadoop及需要配置前置环境，具体要求如下：

1. 从宿主机/opt目录下将文件apache-hive-2.3.4-bin.tar.gz、mysql-connector-java-5.1.47.jar复制到容器master中的/opt/software路径中（若路径不存在，则需新建）, 将容器master节点Hive安装包解压到/opt/module目录下，将命令复制并粘贴至对应报告中；
2. 完善其他Sqoop相关配置，设置Sqoop环境变量，并使环境变量生效，执行命令sqoop version并将命令与其结果截图粘贴至对应报告中；
3. 测试Sqoop连接容器master节点的MySQL数据库是否成功并展示所有的database，并将连接命令与结果截图粘贴至对应报告中。

## 模块B：离线数据处理（25分）

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各模块服务端说明。**  **补充说明：**各主机可通过Asbru工具或SSH客户端进行SSH访问；  Master节点MySQL数据库用户名/密码：root/123456（已配置远程连接）；  Hive的配置文件位于/opt/apache-hive-2.3.4-bin/conf/  Spark任务在Yarn上用Client运行，方便观察日志。  **注：该Spark版本无法进行本地调试，请打包上传集群调试。** |

### 任务一：数据抽取

使用Scala编写spark工程代码，将MySQL的shtd\_store库中表user\_info、sku\_info、base\_province、base\_region、order\_info、order\_detail的数据增量抽取到Hive的ods库中对应表user\_info、sku\_info、base\_province、base\_region、order\_info、order\_detail中。

1. 抽取shtd\_store库中user\_info的增量数据进入Hive的ods库中表user\_info。根据ods.user\_info表中operate\_time或create\_time作为增量字段(即MySQL中每条数据取这两个时间中较大的那个时间作为增量字段去和ods里的这两个字段中较大的时间进行比较)，只将新增的数据抽入，字段名称、类型不变，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.user\_info命令，将结果截图粘贴至对应报告中；
2. 抽取shtd\_store库中sku\_info的增量数据进入Hive的ods库中表sku\_info。根据ods.sku\_info表中create\_time作为增量字段，只将新增的数据抽入，字段名称、类型不变，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.sku\_info命令，将结果截图粘贴至对应报告中；
3. 抽取shtd\_store库中base\_province的增量数据进入Hive的ods库中表base\_province。根据ods.base\_province表中id作为增量字段，只将新增的数据抽入，字段名称、类型不变并添加字段create\_time取当前时间，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.base\_province命令，将结果截图粘贴至对应报告中；
4. 抽取shtd\_store库中base\_region的增量数据进入Hive的ods库中表base\_region。根据ods.base\_region表中id作为增量字段，只将新增的数据抽入，字段名称、类型不变并添加字段create\_time取当前时间，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.base\_region命令，将结果截图粘贴至对应报告中；
5. 抽取shtd\_store库中order\_info的增量数据进入Hive的ods库中表order\_info，根据ods.order\_info表中operate\_time或create\_time作为增量字段(即MySQL中每条数据取这两个时间中较大的那个时间作为增量字段去和ods里的这两个字段中较大的时间进行比较)，只将新增的数据抽入，字段名称、类型不变，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.order\_info命令，将结果截图粘贴至对应报告中；
6. 抽取shtd\_store库中order\_detail的增量数据进入Hive的ods库中表order\_detail，根据ods.order\_detail表中create\_time作为增量字段，只将新增的数据抽入，字段名称、类型不变，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。使用hive cli执行show partitions ods.order\_detail命令，将结果截图粘贴至对应报告中。

### 任务二：数据清洗

使用Scala编写spark工程代码，将ods库中相应表数据全量抽取到Hive的dwd库中对应表中。表中有涉及到timestamp类型的，均要求按照yyyy-MM-dd HH:mm:ss，不记录毫秒数，若原数据中只有年月日，则在时分秒的位置添加00:00:00，添加之后使其符合yyyy-MM-dd HH:mm:ss。

1. 抽取ods库中user\_info表中昨天的分区（任务一生成的分区）数据，并结合dim\_user\_info最新分区现有的数据，根据id合并数据到dwd库中dim\_user\_info的分区表（合并是指对dwd层数据进行插入或修改，需修改的数据以id为合并字段，根据operate\_time排序取最新的一条），分区字段为etl\_date且值与ods库的相对应表该值相等，同时若operate\_time为空，则用create\_time填充，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”。若该条记录第一次进入数仓dwd层则dwd\_insert\_time、dwd\_modify\_time均存当前操作时间，并进行数据类型转换。若该数据在进入dwd层时发生了合并修改，则dwd\_insert\_time时间不变，dwd\_modify\_time存当前操作时间，其余列存最新的值。使用hive cli执行show partitions dwd.dim\_user\_info命令，将结果截图粘贴至对应报告中；
2. 抽取ods库sku\_info表中昨天的分区（任务一生成的分区）数据，并结合dim\_sku\_info最新分区现有的数据，根据id合并数据到dwd库中dim\_sku\_info的分区表（合并是指对dwd层数据进行插入或修改，需修改的数据以id为合并字段，根据create\_time排序取最新的一条），分区字段为etl\_date且值与ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”。若该条数据第一次进入数仓dwd层则dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。若该数据在进入dwd层时发生了合并修改，则dwd\_insert\_time时间不变，dwd\_modify\_time存当前操作时间，其余列存最新的值。使用hive cli查询表dim\_sku\_info的字段id、sku\_desc、dwd\_insert\_user、dwd\_modify\_time、etl\_date，条件为最新分区的数据，id大于等于15且小于等于20，并且按照id升序排序，将结果截图粘贴至对应报告中；
3. 抽取ods库base\_province表中昨天的分区（任务一生成的分区）数据，并结合dim\_province最新分区现有的数据，根据id合并数据到dwd库中dim\_province的分区表（合并是指对dwd层数据进行插入或修改，需修改的数据以id为合并字段，根据create\_time排序取最新的一条），分区字段为etl\_date且值与ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”。若该条数据第一次进入数仓dwd层则dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。若该数据在进入dwd层时发生了合并修改，则dwd\_insert\_time时间不变，dwd\_modify\_time存当前操作时间，其余列存最新的值。使用hive cli在表dwd.dim\_province最新分区中，查询该分区中数据的条数，将结果截图粘贴至对应报告中；
4. 抽取ods库base\_region表中昨天的分区（任务一生成的分区）数据，并结合dim\_region最新分区现有的数据，根据id合并数据到dwd库中dim\_region的分区表（合并是指对dwd层数据进行插入或修改，需修改的数据以id为合并字段，根据create\_time排序取最新的一条），分区字段为etl\_date且值与ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”。若该条数据第一次进入数仓dwd层则dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。若该数据在进入dwd层时发生了合并修改，则dwd\_insert\_time时间不变，dwd\_modify\_time存当前操作时间，其余列存最新的值。使用hive cli在表dwd.dim\_region最新分区中，查询该分区中数据的条数，将结果截图粘贴至对应报告中；
5. 将ods库中order\_info表昨天的分区（任务一生成的分区）数据抽取到dwd库中fact\_order\_info的动态分区表，分区字段为etl\_date，类型为String，取create\_time值并将格式转换为yyyyMMdd，同时若operate\_time为空，则用create\_time填充，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列，其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。使用hive cli执行show partitions dwd.fact\_order\_info命令，将结果截图粘贴至对应报告中；
6. 将ods库中order\_detail表昨天的分区（任务一中生成的分区）数据抽取到dwd库中fact\_order\_detail的动态分区表，分区字段为etl\_date，类型为String，取create\_time值并将格式转换为yyyyMMdd，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列，其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。使用hive cli执行show partitions dwd.fact\_order\_detail命令，将结果截图粘贴至对应报告中。

### 任务三：指标计算

使用Scala编写spark工程代码，并计算相关指标。

**注：在指标计算中，不考虑订单信息表中order\_status字段的值，将所有订单视为有效订单。计算订单金额或订单总金额时只使用final\_total\_amount字段。需注意dwd所有的维表取最新的分区。**

1. 根据dwd层表统计每个省每月下单的数量和下单的总金额，并按照year，month，region\_id进行分组,按照total\_amount降序排序，形成sequence值，将计算结果存入Hive的dws数据库的province\_consumption\_day\_aggr表中（表结构如下），然后使用hive cli根据订单总数、订单总金额、省份表主键均为降序排序，查询出前5条，在查询时对于订单总金额字段将其转为bigint类型（避免用科学计数法展示），将SQL语句与执行结果截图粘贴至对应报告中;

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| province\_id | int | 省份表主键 |  |
| province\_name | string | 省份名称 |  |
| region\_id | int | 地区表主键 |  |
| region\_name | string | 地区名称 |  |
| total\_amount | double | 订单总金额 | 当月订单总金额 |
| total\_count | int | 订单总数 | 当月订单总数 |
| sequence | int | 次序 |  |
| year | int | 年 | 订单产生的年,为动态分区字段 |
| month | int | 月 | 订单产生的月,为动态分区字段 |

1. 根据dwd层的数据，请计算连续两天下单的用户与已下单用户的占比，将结果存入MySQL数据库shtd\_result的userrepurchasedrate表中(表结构如下)，然后在Linux的MySQL命令行中查询结果数据，将SQL语句与执行结果截图粘贴至对应报告中；

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| purchaseduser | int | 下单人数 | 已下单人数 |
| repurchaseduser | int | 连续下单人数 | 连续两天下单的人数 |
| repurchaserate | text | 百占比 | 连续两天下单人数/已下单人数百分比（保留1位小数，四舍五入，不足的补0）例如21.1%，或者32.0% |

1. 根据dwd层的数据，请计算每个省份累计订单量（订单信息表一条算一个记录），然后根据每个省份订单量从高到低排列，将结果打印到控制台（使用spark中的show算子，同时需要显示列名），将执行结果截图粘贴至对应报告中。

例如：可以考虑首先生成类似的临时表A：

|  |  |
| --- | --- |
| province\_name | Amount（订单量） |
| A省 | 10122 |
| B省 | 301 |
| C省 | 2333333 |

然后生成结果类似如下：其中C省销量最高，排在第一列，A省次之，以此类推。

|  |  |  |
| --- | --- | --- |
| C省 | A省 | B省 |
| 2333333 | 10122 | 301 |

**提示：可用str\_to\_map函数减轻工作量**

## 模块C：数据挖掘（10分）

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各模块服务端说明。**  **补充说明：**各主机可通过Asbru工具或SSH客户端进行SSH访问；  Master节点MySQL数据库用户名/密码：root/123456（已配置远程连接）；  Hive的配置文件位于/opt/apache-hive-2.3.4-bin/conf/  Spark任务在Yarn上用Client运行，方便观察日志。  该模块均使用Scala编写，利用Spark相关库完成。 |

### 任务一：特征工程

剔除订单信息表与订单详细信息表中用户id与商品id不存在于现有的维表中的记录，同时建议多利用缓存并充分考虑并行度来优化代码，达到更快的计算效果。

1. 据Hive的dwd库中相关表或MySQL数据库shtd\_store中订单相关表（order\_detail、order\_info、sku\_info），对用户购买过的商品进行去重，将其转换为以下表：第一列为用户id mapping，第二列为用户购买过的商品id mapping，按照user\_id与sku\_id进行升序排序，输出前5行，将结果截图粘贴到对应报告中。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| user\_id | int | 用户id的mapping对应键 |  |
| sku\_id | int | 商品id的mapping对应键 |  |

提示：

Mapping操作：例如用户id：1、4、7、8、9，则做完mapping操作转为字典类型，键0对应用户id1，键1对应用户id4，以此类推

结果格式如下：

-------user\_id\_mapping与sku\_id\_mapping数据前5条如下：-------

0:0

0:89

1:1

1:2

1:3

0.0 1 0 0 0 0 0

1.0 1 1 1 0 0 0

1. 根据第1小题的结果，对其进行聚合，其中对sku\_id进行one-hot转换，将其转换为以下格式矩阵：第一列为用户id，其余列名为商品id，按照用户id进行升序排序，展示矩阵第一行前5列数据，将结果截图粘贴至对应报告中。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| user\_id | double | 客户key |  |
| sku\_id0 | double | 用户是否购买过商品1 | 若用户购买过该商品，则值为1，否则为0 |
| sku\_id1 | double | 用户是否购买过商品2 | 若用户购买过该商品，则值为1，否则为0 |
| sku\_id2 | double | 用户是否购买过商品3 | 若用户购买过该商品，则值为1，否则为0 |
| ..... |  |  |  |

结果格式如下：

---------------第一行前5列结果展示为---------------

0.0,1.0,0.0,0.0,0.0

### 任务二：推荐系统

1. 根据任务一的结果，对其进行SVD分解，对数据进行降维保留前5个奇异值信息，根据该用户已购买的商品分别与未购买的商品计算余弦相似度再进行累加求均值，将累加值最大的5件商品id进行输出作为推荐使用。将输出结果截图粘贴至对应报告中。

结果格式如下：

------------------------推荐Top5结果如下------------------------

相似度top1(商品id：1，平均相似度：0.98)

相似度top2(商品id：71，平均相似度：0.78)

相似度top3(商品id：22，平均相似度：0.76)

相似度top4(商品id：351，平均相似度：0.73)

相似度top5(商品id：14，平均相似度：0.52)

## 模块D：数据采集与实时计算（20分）

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各模块服务端说明。**  **补充说明：**各主机可通过Asbru工具或SSH客户端进行SSH访问；  请先检查ZooKeeper、Kafka、Redis端口是否已启动，若未启动则各启动命令如下：  ZK启动（netstat -ntlp查看2181端口是否打开）  /usr/zk/zookeeper-3.4.6/bin/zkServer.sh start  Redis启动（netstat -ntlp查看6379端口是否打开）  /usr/redis/bin/redis-server /usr/redis/bin/redis.conf  Kafka启动（netstat -ntlp查看9092端口是否打开）  /opt/kafka/kafka\_2.11-2.0.0/bin/kafka-server-start.sh -daemon（空格连接下一行）/opt/kafka/kafka\_2.11-2.0.0/config/server.properties  Flink任务在Yarn上用per job模式（即Job分离模式，不采用Session模式），方便Yarn回收资源。 |

### 任务一：实时数据采集

1. 在Master节点使用Flume采集实时数据生成器10050端口的socket数据，将数据存入到Kafka的Topic中（Topic名称为order，分区数为4），使用Kafka自带的消费者消费order（Topic）中的数据，将前2条数据的结果截图粘贴至对应报告中；
2. 采用多路复用模式，Flume接收数据注入kafka 的同时，将数据备份到HDFS目录/user/test/flumebackup下，将查看备份目录下的第一个文件的前2条数据的命令与结果截图粘贴至对应报告中。

### 任务二：使用Flink处理Kafka中的数据

编写Scala代码，使用Flink消费Kafka中Topic为order的数据并进行相应的数据统计计算（订单信息对应表结构order\_info,订单详细信息对应表结构order\_detail（来源类型和来源编号这两个字段不考虑，所以在实时数据中不会出现），同时计算中使用order\_info或order\_detail表中create\_time或operate\_time取两者中值较大者作为EventTime，若operate\_time为空值或无此属性，则使用create\_time填充，允许数据延迟5S，订单状态order\_status分别为1001:创建订单、1002:支付订单、1003:取消订单、1004:完成订单、1005:申请退回、1006:退回完成。另外对于数据结果展示时，不要采用例如：1.9786518E7的科学计数法）。

1. 使用Flink消费Kafka中的数据，统计商城实时订单数量（需要考虑订单表的状态，若有取消订单、申请退回、退回完成则不计入订单数量，其他状态则累加），将key设置成totalcount存入Redis中。使用redis cli以get key方式获取totalcount值，将结果截图粘贴至对应报告中，需两次截图，第一次截图和第二次截图间隔1分钟以上，第一次截图放前面，第二次截图放后面；
2. 在任务1进行的同时，使用侧边流，使用Flink消费Kafka中的订单详细信息数据，实时统计商城中消费额前3的商品（不考虑订单状态，不考虑打折），将key设置成top3itemconsumption存入Redis中（value使用String数据格式，value为前3的商品信息并外层用[]包裹，其中按排序依次存放商品id:销售额，并用逗号分割）。使用redis cli以get key方式获取top3itemconsumption值，将结果截图粘贴至对应报告中，需两次截图，第一次截图和第二次截图间隔1分钟以上，第一次截图放前面，第二次截图放后面；

示例如下：

top3itemconsumption：[1:10020.2,42:4540.0,12:540]

1. 采用双流JOIN的方式（本系统稳定，无需担心数据迟到与丢失的问题,建议使用滚动窗口），结合订单信息和订单详细信息（需要考虑订单状态，若有取消订单、申请退回、退回完成则不进行统计），拼接成如下表所示格式，其中包含订单id、订单总金额、商品数，将数据存入MySQL数据库shtd\_result的orderpostiveaggr表中（表结构如下），然后在Linux的MySQL命令行中根据id降序排序，查询出前5条，将SQL语句与执行结果截图粘贴至对应报告中。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| id | int | 订单id |  |
| orderprice | double | 订单总金额 | 统计订单信息中 final\_total\_amount字段 |
| orderdetailcount | int | 商品数量总计 | 累加订单详细信息中 sku\_num字段 |

## 模块E：数据可视化（15分）

**环境说明：**

|  |
| --- |
| **数据接口地址及接口描述详见各模块服务端说明。** |

### 任务一：用柱状图展示消费额最高的地区

编写Vue工程代码，根据接口，用柱状图展示2020年消费额最高的5个地区，同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至对应报告中。

### 任务二：用饼状图展示各地区消费能力

编写Vue工程代码，根据接口，用饼状图展示2020年各地区的消费总额占比，同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至对应报告中。

### 任务三：用折线图展示每年上架商品数量的变化

编写Vue工程代码，根据接口，用折线图展示每年上架商品数量的变化情况，同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至对应报告中。

### 任务四：用条形图展示平均消费额最高的地区

编写Vue工程代码，根据接口，用条形图展示2020年平均消费额最高的5个地区，同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至对应报告中。

### 任务五：用散点图展示省份平均消费额

编写Vue工程代码，根据接口，用基础散点图展示2020年最高10个省份平均消费额，同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至对应报告中。

## 模块F：综合分析（10分）

### 任务一：对于UDF/UDAF/UDTF理解

在模块B中可能会用到UDF/UDAF/UDTF，它们的区别是什么，请问它们分别解决了什么问题？将内容编写至对应报告中。

### 任务二：对于数据挖掘模块中的用户推荐有什么好的建议

在模块C中使用基于用户的推荐系统思路对用户的相似性进行计算，从而为每个用户推荐商品，你认为可以从哪些方面再进行优化？这种推荐策略对业务的发展会起到什么样的作用？将内容编写至对应报告中。

### 任务三：简要描述任务过程中的问题并进行总结

将内容编写至对应报告中。