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（1卷）

任

务

书

参赛队编号：

**背景描述**

大数据时代，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；  从本地仓库中拉取镜像，并启动3个容器  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  同时将/opt目录下的所有安装包移动到3个容器节点中。 |

### 任务一：Hadoop 完全分布式安装配置

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. 将Master节点JDK安装包解压并移动到/usr/java路径(若路径不存在，则需新建)，将命令复制并粘贴至对应报告中;
2. 修改/root/profile文件，设置JDK环境变量，配置完毕后在Master节点分别执行“java”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，使用绝对路径从Master节点复制JDK解压后的安装文件到Slave1、Slave2节点，并配置相关环境变量，将全部复制命令复制并粘贴至对应报告中;
4. 在Master节点将Hadoop解压到/opt目录下，并将解压包分发至Slave1、Slave2节点中,配置好相关环境，初始化Hadoop环境namenode，将初始化命令及初始化结果复制粘贴至对应报告中；
5. 启动Hadoop集群，查看Master节点jps进程，将查看结果复制粘贴至对应报告中。

### 任务二：Spark on Yarn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. 将scala包解压到/usr/路径，配置环境变量使其生效，将完整命令复制粘贴至对应报告中（若已安装，则可跳过）；
2. 配置/root/profile文件，设置Spark环境变量，并使环境变量生效将环境变量配置内容复制粘贴至对应报告中;
3. 完成on yarn相关配置，使用spark on yarn 的模式提交$SPARK\_HOME/examples/jars/spark-examples\_2.11-2.1.1.jar 运行的主类为org.apache.spark.examples.SparkPi，将运行结果粘贴至对应报告中。

### 任务三：Flink on Yarn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. 将Flink包解压到路径/opt目录下，将完整命令复制粘贴至对应报告中;
2. 修改/root/profile文件，设置Flink环境变量，并使环境变量生效将环境变量配置内容复制粘贴至对应报告中;
3. 开启Hadoop集群，在yarn上以per job模式（即Job分离模式，不采用Session模式）运行 $FLINK\_HOME/examples/batch/WordCount.jar，将运行结果最后10行复制粘贴至对应报告中。

示例 ：

flink run -m yarn-cluster -p 2 -yjm 2G -ytm 2G $FLINK\_HOME/examples/batch/WordCount.jar

## 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的元数据启动命令为:  nohup hive --service metastore &  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。 |

### 任务一：数据抽取

编写Scala工程代码，将MySQL的shtd\_store库中表CUSTOMER、NATION、PART、PARTSUPP、REGION、SUPPLIER的数据全量抽取到Hive的ods库中对应表customer，nation，part，partsupp，region，supplier中，将表ORDERS、LINEITEM的数据增量抽取到Hive的ods库中对应表ORDERS，LINEITEM中。

1. 抽取shtd\_store库中CUSTOMER的全量数据进入Hive的ods库中表customer。字段排序、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）。并在hive cli执行show partitions ods.customer命令，将结果截图复制粘贴至对应报告中；
2. 抽取shtd\_store库中NATION的全量数据进入Hive的ods库中表nation。字段排序、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）。并在hive cli执行show partitions ods.nation命令，将结果截图复制粘贴至对应报告中；
3. 抽取shtd\_store库中PART的全量数据进入Hive的ods库中表part。字段排序、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）。并在hive cli执行show partitions ods.part命令，将结果截图复制粘贴至对应报告中；
4. 抽取shtd\_store库中PARTSUPP的全量数据进入Hive的ods库中表partsupp。字段排序、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）。并在hive cli执行show partitions ods.partsupp命令，将结果截图复制粘贴至对应报告中；
5. 抽取shtd\_store库中REGION的全量数据进入Hive的ods库中表region，字段排序、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）。并在hive cli执行show partitions ods.region命令，将结果截图复制粘贴至对应报告中；
6. 抽取shtd\_store库中SUPPLIER的全量数据进入Hive的ods库中表supplier，字段排序、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）。并在hive cli执行show partitions ods.supplier命令，将结果截图复制粘贴至对应报告中；
7. 抽取shtd\_store库中ORDERS的增量数据进入Hive的ods库中表orders，要求只取某年某月某日及之后的数据（包括某年某月某日，如：1992-04-01），根据ORDERS表中ORDERKEY作为增量字段（提示：对比MySQL和Hive中的表的ORDERKEY大小，假设hive中orderkey最大的是5800257），只将新增的数据抽入，字段类型不变，同时添加动态分区，分区字段类型为String，且值为ORDERDATE字段的内容（ORDERDATE的格式为yyyy-MM-dd，分区字段格式为yyyyMMdd），。并在hive cli执行select count(distinct(dealdate)) from ods.orders命令，将结果截图复制粘贴至对应报告中；

到hive的ods库中orders中找出最大的key值

在查询mysql数据的时候，直接加上这个条件即可，select \* from orders where orderkey > {最大的key值} and orderdate >= ‘1998-01-01’

1. 抽取shtd\_store库中LINEITEM的增量数据进入Hive的ods库中表lineitem，根据LINEITEM表中orderkey作为增量字段，只将新增的数据抽入，字段类型不变，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）。并在hive cli执行show partitions ods.lineitem命令，将结果截图复制粘贴至对应报告中。

### 任务二：数据清洗

编写Scala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. 将ods库中customer表数据抽取到dwd库中dim\_customer的分区表，分区字段为etldate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”，dwd\_insert\_time、dwd\_modify\_time均填写操作时间，并进行数据类型转换。在hive cli中按照cust\_key顺序排序，查询dim\_customer前1条数据，将结果内容复制粘贴至对应报告中；
2. 将ods库中part表数据抽取到dwd库中dim\_part的分区表，分区字段为etldate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”，dwd\_insert\_time、dwd\_modify\_time均填写操作时间，并进行数据类型转换。在hive cli中按照part\_key顺序排序，查询dim\_part前1条数据，将结果内容复制粘贴至对应报告中；
3. 将ods库中nation表数据抽取到dwd库中dim\_nation的分区表，分区字段为etldate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”，dwd\_insert\_time、dwd\_modify\_time均填写操作时间，并进行数据类型转换。在hive cli中按照nation\_key顺序排序，查询dim\_nation前1条数据，将结果内容复制粘贴至对应报告中；
4. 将ods库中region表数据抽取到dwd库中dim\_region的分区表，分区字段为etldate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”，dwd\_insert\_time、dwd\_modify\_time均填写操作时间，并进行数据类型转换。在hive cli中按照region\_key顺序排序，查询dim\_region表前1条数据，将结果内容复制粘贴至对应报告中；
5. 将ods库中orders表数据抽取到dwd库中fact\_orders的分区表，分区字段为etldate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”，dwd\_insert\_time、dwd\_modify\_time均填写操作时间，并进行数据类型转换。在执行hive cli执行select count(distinct(dealdate)) from dwd.fact\_orders命令，将结果内容复制粘贴至对应报告中；
6. 待任务5完成以后，需删除ods.orders中的分区，仅保留最近的三个分区。并在hive cli执行show partitions ods.orders命令，将结果截图粘贴至对应报告中；

Alter table xxx drop partition(xxxdate < ‘2010-01-01’)

1. 将ods库中lineitem表数据抽取到dwd库中fact\_lineitem的分区表，分区字段为etldate且值与ods库的相对应表该值相等，抽取的条件为根据orderkey和partkey进行去重，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均填写操作时间，并进行数据类型转换。在hive cli执行show partitions dwd.fact\_lineitem命令，将结果截图粘贴至对应报告中。

### 任务三：指标计算

1. 编写Scala工程代码，根据dwd层表统计每个地区、每个国家、每个月下单的数量和下单的总金额，存入MySQL数据库shtd\_store的nationeverymonth表（表结构如下）中，然后在Linux的MySQL命令行中根据订单总数、消费总额、国家表主键三列均逆序排序的方式，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 |
| nationkey | int | 国家表主键 |  |
| nationname | text | 国家名称 |  |
| regionkey | int | 地区表主键 |  |
| regionname | text | 地区名称 |  |
| totalconsumption | double | 消费总额 | 当月消费订单总额 |
| totalorder | int | 订单总数 | 当月订单总额 |
| year | int | 年 | 订单产生的年 |
| month | int | 月 | 订单产生的月 |

1. 请根据dwd层表计算出某年每个国家的平均消费额和所有国家平均消费额相比较结果（“高/低/相同”）,存入MySQL数据库shtd\_store的nationavgcmp表（表结构如下）中，然后在Linux的MySQL命令行中根据订单总数、消费总额、国家表主键三列均逆序排序的方式，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;

Select nation.nationkey,nation.name,avg(totalprice) over(partition by nation.nationkey,substr(orderdate,1,4)) as nation\_avg,avg(totalprice) over() as all\_avg from orders inner join customer on orders.custkey = customer.custkey inner join nation on customer.nationkey = nation.nationkey

出来的结果去重

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| nationkey | int | 国家表主键 |  |
| nationname | text | 国家名称 |  |
| nationavgconsumption | double | 该国家内客单价 | 该国家已购买产品的人均消费额 |
| allnationavgconsumption | double | 所有国家内客单价 | 所有国家已购买的产品的人均消费额 |
| comparison | string | 比较结果 | 国家内人均和所有国家人均相比  结果有：高/低/相同 |

1. 编写Scala工程代码，根据dwd层表统计连续两个月下单并且下单金额保持增长的用户，订单发生时间限制为大于等于某年，存入MySQL数据库shtd\_store的usercontinueorder表(表结构如下)中。然后在Linux的MySQL命令行中根据订单总数、消费总额、客户主键三列均逆序排序的方式，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 |
| custkey | int | 客户主键 |  |
| custname | text | 客户名称 |  |
| month | text | 月 | 记录当前月和下月，用下划线‘\_’相连。  例如： 202201\_202202  表示2022年1月到2月用户连续下单。 |
| totalconsumption | double | 消费总额 | 连续两月的订单总额 |
| totalorder | int | 订单总数 | 连续两月的订单总数 |

## 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的元数据启动命令为:  nohup hive --service metastore &  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

1. 根据dwd库中fact\_orders表，将其转换为以下类型矩阵：其中A表示用户A，B表示用户B，矩阵中的【0,1】值为1表示A用户与B用户之间购买了1个相同的零件，0表示A用户与B用户之间没有购买过相同的零件。将矩阵保存为txt文件格式并存储在HDFS上，使用命令查看文件前2行，将执行结果截图粘贴至对应报告中；



1. 对dwd库中dim\_part获取partkey 、mfgr、brand、size、retailprice五个字段并进行数据预处理，再进行归一化并保存至dwd.fact\_part\_machine\_data中，对制造商与品牌字段进行one-hot编码处理（将制造商与品牌的值转换成列名添加至表尾部，若该零部件属于该品牌则置为1，否则置为0）,并按照partkey，size进行顺序排序，然后在Hive cli中执行命令desc dwd.fact\_part\_machine\_data 中查询出结果，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 |
| partkey | double | 零件key |  |
| mfgr | double | 制造商 |  |
| brand | double | 品牌 |  |
| size | double | 型号 |  |
| retailprice | double | 价格 |  |
| Manufacturer#1 | int | 制造商1 | 若属于该制造商，则内容为1否则为0 |
| Manufacturer#2 | int | 制造商2 | 若属于该制造商，则内容为1否则为0 |
| ..... |  |  |  |
| Brand#1 | int | 品牌1 | 若属于该品牌，则内容为1否则为0 |
| Brand#2 | int | 品牌2 | 若属于该品牌，则内容为1否则为0 |
| …… |  |  |  |

### 任务二：推荐系统

1. 根据任务一的结果，获取与该用户相似度（矩阵内的值最高）最高的前10个用户，并结合hive中dwd层的fact\_orders表、fact\_lineitem表、fact\_part\_machine\_data表，获取到这10位用户已购买过的零部件，并剔除该用户已购买的零部件，并通过计算用户已购买产品与该数据集中余弦相似度累加，输出前5零部件key作为推荐使用。将输出结果保存至MySQL的part\_machine表中。然后在Linux的MySQL命令行中查询出前5条数据，将SQL语句与执行结果截图粘贴至对应报告中。

## 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端口看是否已启动，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26001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），将Flume的配置截图粘贴至对应报告中；
2. Flume接收数据注入kafka 的同时，将数据备份到HDFS目录/user/test/flumebackup下，将备份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工程代码，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。

1. 使用Flink消费Kafka中的数据，统计个人实时订单总额，将key设置成totalprice存入Redis中（再使用hash数据格式，key存放为用户id，value存放为该用户消费总额），使用redis cli以get key方式获取totalprice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一分钟以上，第一次截图放前面，第二次放后面。
2. 在任务1进行的同时需监控若发现ORDERSTATUS字段为F,将数据存入MySQL表alarmdata中（可考虑侧边流的实现），然后在Linux的MySQL命令行中根据ORDERKEY逆序排序，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;
3. 使用Flink消费kafka中的数据，统计每分钟下单的数量，将key设置成totalorder存入redis中。使用redis cli以get key方式获取totalorder值，将结果粘贴至对应报告中，需两次截图，第一次截图（应该在job启动2分钟数据稳定后再截图）和第二次截图时间间隔应达一分钟以上，第一次截图放前面，第二次放后面。（注：流数据中，时间未精确到时分秒，建议StreamTimeCharacteristic设置成ProcessingTime（默认）或IngestionTime。）

## 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国家

编写Vue工程代码，根据接口，用柱状图展示某年某月消费额最高的5个国家，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，用饼状图展示某年第一季度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：用折线图展示总消费额变化

编写Vue工程代码，根据接口，用折线图展示某年上半年商城总消费额的变化情况，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国家

编写Vue工程代码，根据接口，用条形图展示某年平均消费额最高的5个国家，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：用折柱混合图展示地区平均消费额和国家平均消费额

编写Vue工程代码，根据接口，用折柱混合图展示某年地区平均消费额和国家平均消费额的对比情况，柱状图展示平均消费额最高的5个国家，折线图展示每个国家所在的地区的平均消费额变化，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：如何解决Job运行效率低的问题

在模块B中出现某些Job运行时间较长，你认为可能是哪些情况造成？有什么相关的方法吗？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。