

课 程 报 告

题 目 基于 spark 的电商销售数据分析

学生姓名 丁家宁

学生学号 1221004046

专业班级 大数据 222

完成日期 2025.8.1

目录

1. 引言	1
1.1 研究背景	1
1.2 国内外现状	1
1.3 研究内容	2
2. 系统设计	3
2.1 系统架构	3
2.2 技术选型	4
2.3 需求设计	4
2.4. 系统流程	6
3. 系统实现	7
3.1 环境部署	7
3.2 数据清洗	9
3.3 数据存储	14
3.4MAPREDUCE 数据分析	15
3.5HIVE 数据查询	17
4. 数据可视化	21
4.1 前端设计	21
4.2 美妆数据分析结果可视化	22
4.3 词云图模块设计	25
5. 结论	25
参考文献	26

基于 SPRAK 的电商销售数据分析

1. 引言

1.1 研究背景

电子商务在现代服务业中占据着举足轻重的地位，其迅猛发展之势不仅为经济注入了新的活力，更带来了海量的数据资源。电商数据分析，作为大数据应用领域的一颗璀璨明珠，因其能够深入揭示消费者行为、市场趋势以及业务运营的内在规律，历来备受各大电商平台的青睐与重视。在当今信息爆炸的时代，电商平台通过对海量数据的精准分析，不仅能够更好地把握市场脉搏，优化商品推荐、精准营销等策略，还能显著提升用户体验，从而增强平台的竞争力。因此，电商数据分析已经成为电商平台不可或缺的核心竞争力之一，其在电子商务产业中的重要地位日益凸显。

Hadoop 是一个开源的分布式计算框架，具备处理大规模数据集的强大能力。它不仅提供了可靠的数据存储服务，还确保了高效的数据处理性能，尤其适用于处理海量的电商销售数据。基于 Hadoop 的电商销售数据分析系统，能够充分利用 Hadoop 的强大功能，对电商销售数据进行高效的分析和挖掘。

利用 Hadoop，电商平台可以构建一个高效、可扩展的销售数据分析系统。这样的系统不仅能够处理海量的销售数据，还能通过复杂的算法和模型，揭示销售数据的深层次信息。例如，通过用户购买行为分析，系统能够识别出热销产品、预测市场趋势，甚至为每个用户定制个性化的推荐列表。同时，Hadoop 的可靠性保证了数据的安全存储，即便面对硬件故障，也能确保数据的完整性和一致性。此外，Hadoop 生态系统中的其他工具，如 Hive、Pig、Spark 等，也为数据分析提供了强大的支持。这些工具可以帮助数据分析师更加便捷地查询、处理和分析数据，大大提高了数据分析的效率和准确性。

1.2 国内外现状

在全球范围内，电子商务正经历着前所未有的增长和创新。随着互联网技术的不断进步，电子商务已经成为全球经济的重要组成部分，不仅改变了消费者的购物习惯，也重塑了企业的商业模式。在国内，电子商务的快速发展得益于政府的支持和推动，以及消费者对在线购物的逐渐接受和依赖。政府出台了一系列政

策，鼓励电子商务的发展，包括减税、简化行政审批流程等措施。同时，国内企业也在不断加大在技术研发上的投入，以期在激烈的市场竞争中保持领先地位。在国际市场上，电子商务同样展现出强劲的增长势头。许多跨国公司通过电子商务平台拓展了其全球业务，实现了产品的全球销售和品牌的国际化。此外，随着移动支付和跨境物流的发展，电子商务的便利性和效率得到了显著提升。

然而，电子商务的发展也面临着一些挑战。数据安全和隐私保护问题日益突出，消费者对个人信息的保护意识不断增强。同时，随着数据量的爆炸性增长，如何有效管理和分析这些数据，以提供更加精准和个性化的服务，成为电商企业需要解决的问题。为了应对这些挑战，国内外的电商企业和研究机构正在采取一系列措施。一方面，加强数据安全和隐私保护的技术研究，确保用户信息的安全。另一方面，通过人工智能、大数据等先进技术，提高数据处理的智能化水平，为用户提供更加个性化的购物体验。

未来，电子商务将继续作为推动经济发展的重要力量。随着 5G、物联网、区块链等新技术的应用，电子商务将迎来更加广阔的发展空间。企业需要不断创新和适应，以满足消费者日益多样化和个性化的需求，同时也要注意社会责任，保护消费者权益，促进电子商务的健康可持续发展。

1.3 研究内容

本课题旨在设计和实现一个基于 Hadoop 的电商销售数据分析系统。该系统将通过数据采集、清洗、存储和分析等环节，对电商销售数据进行全面的挖掘和分析。通过对数据的深入分析，可以帮助企业发现潜在的商机、提高销售额、优化运营效率等。同时，该系统还可以提供可视化的报表和图表，使决策者能够直观地了解销售情况和趋势。

本系统实现了对双十一美妆产品数据的存储、处理、分析与可视化。首先，从相关的化妆品网站采集所需的数据，然后使用 pandas 进行数据预处理。接下来，我们将预处理过的数据存储在 hive 数据库中。之后，我们使用 mapreduce 程序按照不同类型的统计要求来统计与分析数据。根据图表结合前端代码进行可视化处理，利用可视化工具 Echarts 绘制可交互的图表。我们将数据存入 hive 中，再通过对用户语句的分析，构建查询语句，并在 hive 数据库中进行查找，从而得到结果。根据结果使用 echarts 绘制相对应的图表，并在前端页面展示，

直观地展示双十一美妆产品销售数据中所隐藏的信息。

2. 系统设计

2.1 系统架构

系统通过数据接入层从多源收集原始数据,经过数据清洗层的预处理后存储于 Hive 数据仓库中。利用 MapReduce 进行复杂的数据分析和处理,然后通过 Hive SQL 提供数据查询服务。最后,使用 ECharts 和 Bootstrap 技术在可视化层展示分析结果,为用户提供直观的交互界面。整个流程确保了数据处理的高效性和分析结果的易用性。以下是系统的具体架构:

1. 数据接入层

功能: 负责从不同数据源收集原始数据。

技术实现: 文件上传、数据库导入等。

组件: 文件上传接口、数据库相关语句等。

2. 数据清洗层

功能: 对原始数据进行预处理,包括去除重复记录、处理缺失值、数据标准化等。

技术实现: 使用 Python 的 pandas 库进行数据操作,使用 jieba 库进行中文分词和文本处理。

组件: 数据清洗脚本、ETL 工具等。

3. 数据存储层 (Hive)

功能: 存储清洗后的数据,为上层分析提供持久化的数据支持。

技术实现: 使用 Hive 作为数据仓库,存储结构化数据。

组件: Hive 表、HDFS。

4. 处理层 (MapReduce)

功能: 执行复杂的数据分析和统计任务。

技术实现: 编写 MapReduce 程序,利用 Hadoop 集群进行并行计算。

组件：MapReduce 作业、Hadoop 集群。

5. 查询层 (Hive SQL)

功能：提供数据检索和查询操作，支持用户自定义的查询需求。

技术实现：使用 Hive SQL 执行查询，获取数据分析结果。

组件：Hive 客户端、Hive Server。

6. 可视化层 (echarts)

功能：将数据分析结果以图表形式展示，提供直观的数据交互界面。

技术实现：使用 ECharts 作为前端图表库，结合 Bootstrap 进行前端页面设计。

组件：前端 Web 应用、ECharts 图表组件。

2.2 技术选型

数据清洗：Python (jieba 库, 正则表达式)

数据存储：HDFS, Hive

数据处理：MapReduce

数据查询：Hive SQL (HQL)

数据可视化：ECharts, Bootstrap

2.3 需求设计

2.3.1 数据清洗需求分析

数据集存在重复项，数据不规范（如：销量和评价部分值为空，id 重复，多个价格）等问题。数据集在使用之前需要进行清洗，将重复的数据删除，不规范的数据删除或填补为合理的数据。此外，从数据集中的商品名称一列，使用 python 中的 jieba 库并结合正则化处理，提取出商品的详细信息，如商品类别、商品单位等信息。

2.3.2 数据存储需求分析

本项目将数据集上传到虚拟机上并存储到 HIVE 表中。

2.3.3 MapReduce 需求分析

MapReduce 数据分析模块，自行设计分析任务并编写 MR 程序处理这些统计分析任务。本项目主要有以下 MR 统计分析任务：

1. 统计不同化妆品品牌的销售额；
2. 统计不同化妆品品牌的销量；
3. 化妆品品牌的评论数量排行；
4. 男士化妆品销量排行；
5. 统计不同类别化妆品的销量；
6. 统计不同类化妆品的销售额；
7. 统计男士女士化妆品销售量比例；

2.3.4 Hive 查询需求分析

Hive 数据查询模块，自行设计查询条件并编写 HQL 语句完成查询任务。在虚拟机上编写 hql 语句并保存为 hql 文件，使用外部命令执行 hql 文件，将查询结果打印在控制台或存储到 hive 表中或存储到指定的 txt 文件中。本项目设计的 Hive 查询任务主要有：

1. 销量排名前十的化妆品牌
2. 销售额前十的化妆品牌
3. 男士化妆品销量最多的品牌
4. 销售额从高到低的化妆品大类
5. 销量前 100 的景点分省分布情况
6. 价格排名前十的商品
7. 评论数最多的前二十个商品
8. 男士化妆品与女士化妆品的占比
9. 各品牌的口红种类
10. 化妆品的品牌平均销售额
11. 护肤品类别中销售额前十的商品

- 12. 套装类别中销量前十的商品
- 13. 预售商品销量前十的品牌
- 14. 护肤品大类中销量最高的子类
- 15. 护肤品大类中销售额最高的子类

2.3.5 数据可视化需求分析

项目的可视化部分需要包含各省会城市化妆品订单完成用户数省份分析，销售额大类占比和各品牌均价分析及部分其他可视化图表。数据需要先编写 MR 或者 HQL 对原始数据集进行统计分析得出各项任务的分析结果，再根据所分析结果在 echarts 进行图表展示，利用 bootstrap 框架以及 echarts 等工具完成可视化。完成模块如下：

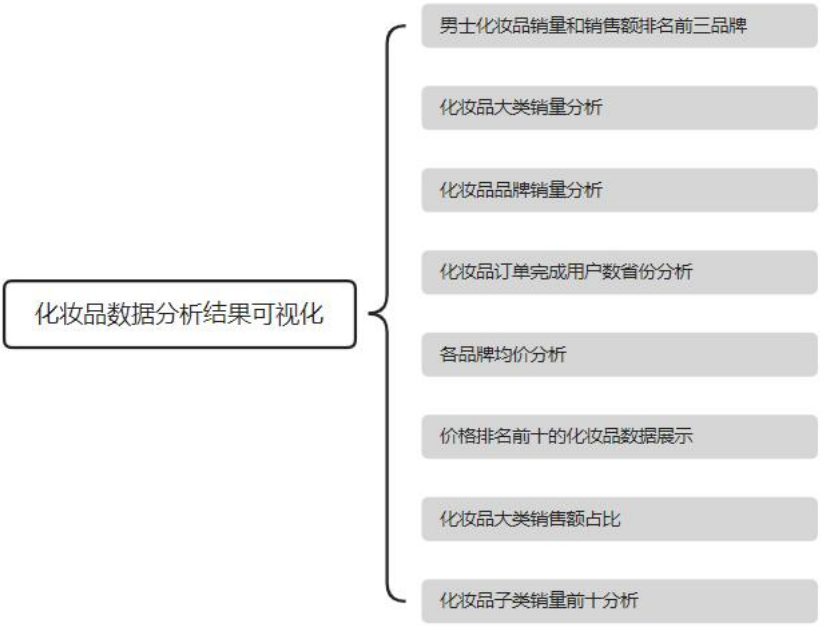


图 3.1 数据可视化模块

2.4. 系统流程

整个系统主要是分为五个部分：数据预处理、数据存储、mapreduce 统计分析、hive 数据查询、可视化前端展示。整个项目流程的图示如下：

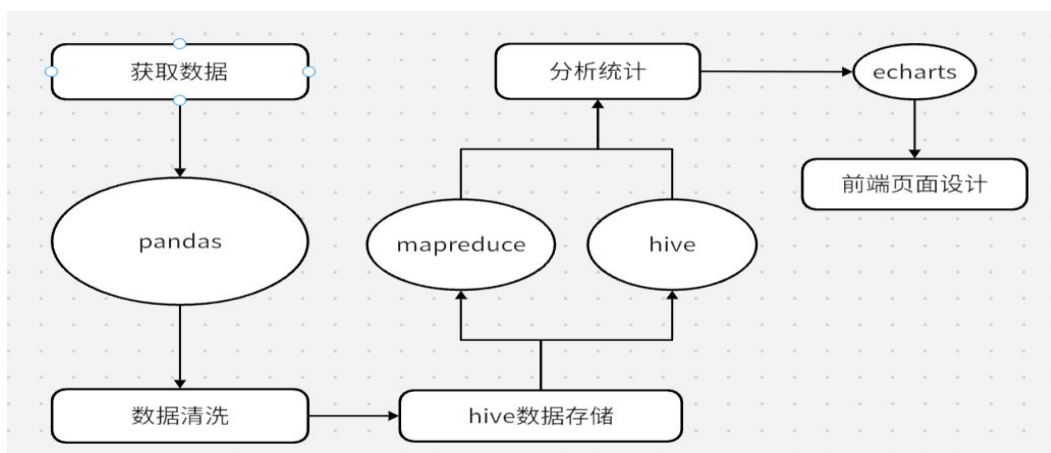


图 3.2 系统流程图

3. 系统实现

3.1 环境部署

3.1.1 硬件部署

服务器：选用至少两个物理 CPU 核心的服务器，确保足够的处理能力以应对大规模数据的处理和分析。服务器支持虚拟化技术，以便在需要时能够灵活地扩展集群规模。

内存：根据数据量的大小和处理需求，每台服务器应配置足够的内存。Hadoop 和 Hive 在处理大规模数据时会对内存有较高的要求。

存储：配置高性能的分布式存储系统 HDFS（Hadoop Distributed File System），用于存储原始数据和处理结果。根据数据量的大小和增长趋势，规划足够的存储空间。

网络：服务器之间应使用高速网络连接，以确保数据在集群中的高效传输。

配置合适的网络设备和安全策略，确保网络的稳定性和安全性。

3.1.2 软件部署

操作系统：选择稳定且兼容 Hadoop 和 Hive 的操作系统 Linux。在每台服务器上安装相同的操作系统版本，以确保集群的一致性。

Hadoop：在所有服务器上安装 Hadoop 集群，并配置好相关的配置文件，如 `core-site.xml`、`hdfs-site.xml`、`yarn-site.xml` 等。配置 NameNode、

DataNode、ResourceManager、NodeManager 等角色，确保集群能够正常运行。

Hive: 在 Hadoop 集群上安装 Hive, 并配置好 Hive 的元数据存储和 HDFS 数据存储路径。

PyCharm: 在计算机上安装 PyCharm IDE, 用于编写和运行 Python 脚本进行数据清洗和预处理。配置 Python 环境, 安装必要的 Python 库 (如 pandas、numpy 等) 以支持数据处理和分析。

ECharts: 在 Web 应用程序的服务器上安装 ECharts 库, 用于生成和展示可视化图表。前端页面中引入 ECharts 的 JavaScript 文件, 并编写相应的代码来渲染图表。

3.1.3 网络部署

网络架构: 选用合理的网络架构, 确保服务器之间的网络连接畅通无阻。

配置防火墙和安全策略, 保护服务器免受潜在的网络攻击。

数据传输: 确保网络环境稳定可靠, 支持高效的数据传输。

使用加密技术保护数据传输过程中的数据安全。

Web 访问: 配置 Web 服务器的访问权限和安全性设置, 确保只有授权用户能够访问 Web 应用程序。

使用负载均衡技术提高 Web 应用程序的可用性和性能。

3.1.4 其他注意事项

备份与恢复: 定期备份 Hadoop 和 Hive 的数据以及配置文件, 以防止数据丢失或损坏。制定数据恢复计划, 确保在发生故障时能够迅速恢复系统。

监控与日志: 使用监控工具 (如 Hadoop 的 YARN ResourceManager UI、Hive 的 Web UI 等) 实时监控 Hadoop 和 Hive 集群的运行状态。

配置日志收集和分析系统, 以便及时发现和解决问题。

性能调优: 根据系统的实际运行情况, 对 Hadoop 和 Hive 进行性能调优, 提高系统的处理能力和响应速度。

3.2 数据清洗

本项目数据清洗于 pycharm 中通过 python 中 Numpy、Pandas、Skearn 库完成。

(1) 数据导入

通过 pandas 库的 read_csv 方法导入数据。

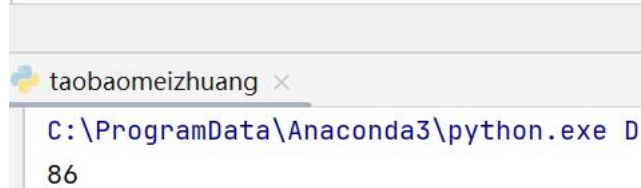
```
# 读取文件查看数据集信息
data = pd.read_csv('F:\双十一美妆数据.csv', encoding='gbk')
```

图 4.1 数据导入

(2) 去除重复值

由下图可知，此数据集中有 86 行重复数据值。

```
# 查看重复数据
print(len(data[data.duplicated()]))
```



The screenshot shows a Jupyter Notebook terminal window with the title 'taobaomeizhuang'. The command prompt is 'C:\ProgramData\Anaconda3\python.exe D'. The output of the print statement is '86'.

图 4.2 去除重复值

通过 pandas 库的 drop_duplicates 方法去除重复值，保证数据完整性。

```
# 某时间某个特定id的购买的商品记录应该只有一条，去除这些重复值
data= data.drop_duplicates(subset=['id', 'update_time'])
```

图 4.3 去除重复值

(3) 去除缺失值

① 数据集总体缺失值情况说明

由下图可知，数据集总体缺失值情况较少，购买时间、id 和商品标题、店名以及价格等列均没有缺失值，商品价格以及评论数量的缺失值仅占总体数据集的 8.5%左右，故选择将进行删除处理。

```
# 查看每一列的缺失比例
print((data.isnull().sum()) / data.shape[0])
```

```

update_time    0.000000
id             0.000000
title          0.000000
price          0.000000
sale_count     0.085296
comment_count  0.085296
店名           0.000000
dtype: float64

```

图 4.4 缺失值查看

② 去除数据集价格和评论缺失值

用 `dropna` 删除缺失的数据，并再次查看缺失值情况，发现数据集目前没有缺失值。

```

# 删除价格丢失的数据
data.dropna(subset=['sale_count'], inplace=True)
print((data.isnull().sum()) / data.shape[0])

update_time    0.0
id             0.0
title          0.0
price          0.0
sale_count     0.0
comment_count  0.0
店名           0.0
dtype: float64

```

图 4.5 去除缺失值

(4) 重置索引

由于对数据集的去重和删除缺失值，导致数据集的索引出错，对数据集重置索引。

```

# 重置索引
data.index=range(len(data.index))

```

图 4.6 重置索引

(5) 数据集分类

① 数据集分类总体说明

查看数据的商品名发现其中包含有化妆品类别的相关信息，从商品的名称进行判断时，设置数据集中的一行划到某个分类后，就不归入另外一个分类，所以判定时候要先判定是否是套装，再判断是否是其他分类，最

后未分类的才是其他类。

```
# 查看title列的信息
print(data.iloc[:,2])

0      CHANDO/自然堂 雪域精粹纯粹滋润霜50g 补水保湿 滋润水润面霜
1      CHANDO/自然堂凝时鲜颜肌活乳液120ML 淡化细纹补水滋润专柜正品
2      CHANDO/自然堂活泉保湿修护精华水（滋润型135mL 补水控油爽肤水
3      CHANDO/自然堂 男士劲爽控油洁面膏 100g 深层清洁 男士洗面奶
4      CHANDO/自然堂雪域精粹纯粹滋润霜（清爽型）50g补水保湿滋润霜
...
```

图 4.7 数据 title 列信息

②设置分类字典

根据数据集中的 title 列分析，得到大致的化妆品类别，并设置分类字典，添加了各类化妆品以及套装的分类。

```
# 分类字典
category_map={}
for line in category_name.strip('\n').split('\n'):
    line_split=line.split()
    for cat in line_split[2:]:
        category_map[cat]=line_split[0:2]

# 添加其他和套装的分类
category_map['其他']=['其他','其他']
```

图 4.8 设置分类字典

③提取分类并添加分类字段

创建了一个空列表 cats，用于存储分类结果，循环遍历数据集中的标题列，并检查该标题是否包含 category_map 字典中的任何键。如果找到了匹配的键，就将对应的值添加到 cats 列表中，并将 is_find 标志设置为 True。如果在遍历完 category_map 后仍未找到匹配的键，则将 category_map['其他']添加到 cats 列表中。最后，将 cats 列表中的主分类和子分类分别解压缩到 main_cat 和 sub_cat 变量中，并将它们作为新的列添加到 data 中。

```

# 提取分类
cats=[]
# 以Id去重后进行数据分类，然后按id填入trade完成分类
for title in data.title:
    is_find=False
    for key,value in category_map.items():
        if key in title:
            cats.append(value)
            is_find=True
            break
    if not is_find:
        cats.append(category_map['其他'])

# 添加分类字段
main_cat,sub_cat = zip(*cats)
data['main_cat']=main_cat
data['sub_cat']=sub_cat

```

图 4.9 提取分类字段

④查看当前分类情况

当前数据集中‘其他’类别的化妆品占比 17.5%，比例仍然过大，需要继续处理‘其他’类的化妆品。

```

# 查看类别中其他数据的占比
print(data[data.main_cat == '其他'].shape[0] / data.shape[0])

```

图 4.10 查看数据占比

⑤二次分类

对‘其他’类别进行二次处理，‘其他’类中包含有 suit_keys 关键字的统一被认为‘套装’类别。

```

# 处理‘其他’类别化妆品
data_other = data[data.main_cat == '其他']
suit_keys = ['合一','件套','套装'] # 套装标志性词
for ix in data_other.index:
    title = data_other.loc[ix,'title']
    for word in suit_keys:
        if word in title:
            data.loc[ix,'main_cat']='套装' # 直接修改数据集数据
            data.loc[ix,'sub_cat']='套装'
            break

```

图 4.11 二次分类

(6) 数据集分词处理

①正则化处理 title

使用正则表达式替换 title 中的英文、数字、【】包括起来的活动信息以及 + - / ' 等特殊字符。

```
# 正则化处理
data_other = data[data.main_cat == '其他'][['title', 'main_cat', 'sub_c
re_bracket = re.compile('【[^】+】') # 匹配括号内容
re_en_num = re.compile('[\da-zA-Z]+') # 英文和数字
re_spec = re.compile("[/+-_*']+") # 特殊
sub_str = '' # 替换字符串，这里是空字符串
data_other['title2'] = data_other.title # 复制title数据
```

图 4.12 数据集分词处理

②分词处理

从数据中提取唯一的店名，并将其转换为列表形式，然后使用 jieba.add_word() 方法将这些店名添加到 jieba 分词库中，以提高分词的精度。并对其他标题进行分词，遍历数据中的其他标题（data_other.title2），使用 jieba.lcut() 方法对每个标题进行分词，并将分词结果存储在列表 title_cuts 中。经过处理，可以发现‘其他’类别的占比下降到 12.7%。

```
# 分词
title_cuts = []
for title in data_other.title2:
    title_cuts.append(jieba.lcut(title))

print(data[data.main_cat == '其他'].shape[0] / data.shape[0])

0.1276925522613465
```

图 4.13 分词处理

(7) 根据 title 提取规格

根据 title 标题中商品含量以及价格，计算各个商品的规格单价，在数据集中增加新的列 type，由于匹配单位过程过于复杂，并考虑到多种情况，参考了网上部分资料，此处只放置了部分核心代码图。

```

data_spec = data.drop_duplicates(subset='id')[['id', 'title', 'main
col_list=['spec_data', 'spec_unit', 'spec_mount', 'discount', 'for_ma
for col in col_list: # 为trade添加新列
    data[col] = None
for col in col_list[:3]:
    data_spec[col] = None
data_spec['type'] = None

# 修正单位
no_unit = ['色', '号', '折', '周', '重', '秒', 'pa', '补', '享', '月', '牧', '清', '部', '预', '正', '天']
fix_unit = {'m': ('m l', 'mL'), 'p': ('p', '片')}
suit_unit = ['套', '种', '款']
for ix in data_spec.index:
    unit = data_spec.loc[ix, 'spec_unit']
    if unit in no_unit: # 没有单位的处理
        data_spec.loc[ix, 'type'] = 0
        data_spec.loc[ix, 'spec_unit'] = ''
        data_spec.loc[ix, 'spec_data'] = 1
        data_spec.loc[ix, 'spec_mount'] = 1
        continue

```

图 4.14 提取规格

3.3 数据存储

本项目数据存储于 Hive 中，通过 Hive 完成相应数据分析后，将分析结果存储于 MySQL 中，MySQL 表设计根据任务需求变化。

Hive 表设计：

```

hive> desc formatted meizhuang;
OK
# col_name          data_type          comment
update_time        string
id                  string
title               string
price               float
sale_count          int
comment_count       int
store               string
main_cat            string
sub_cat             string
spec_data           float
spec_unit           string
spec_mount          int
discount            string
for_man             int

```

图 4.15hive 表设计

3. 4MapReduce 数据分析

本项目主要使用 eclipse 编写 MR 程序，根据需求分析完成 MR 数据分析任务。本项目设计并完成了以下 MR 统计分析任务：

1. 统计不同化妆品品牌的销售额；

对于第一个任务，在 Mapper 中对每次读入的一行字符串进行分割，将 store 作为 key，price 作为 value 传入 Reducer，在 Reducer 中遍历 values 将所有的 price 加起来得到不同品牌的销售额，最后输出到 context 中。

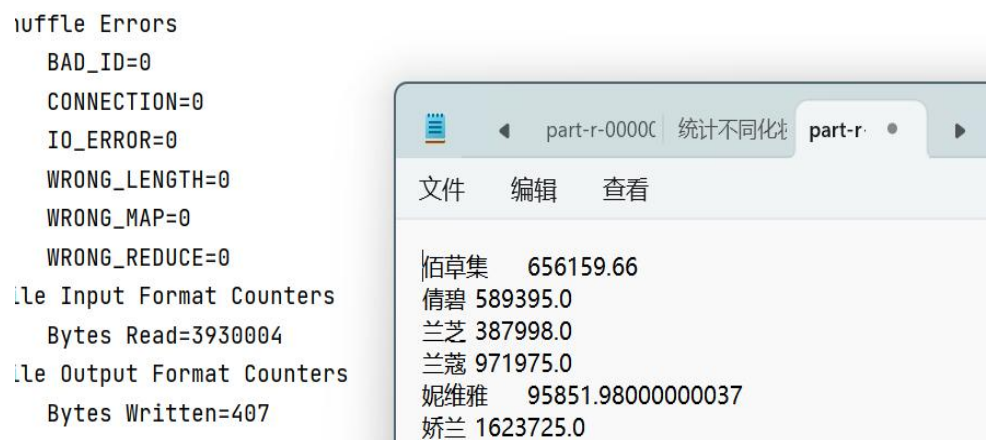


图 4.16 统计不同品牌销售额

2. 统计不同化妆品品牌的销量；

对于第二个任务统计不同化妆品品牌的销量，实际上与第一个任务同属一类，大部分代码相同，将 key 换成 sale_count 即可。

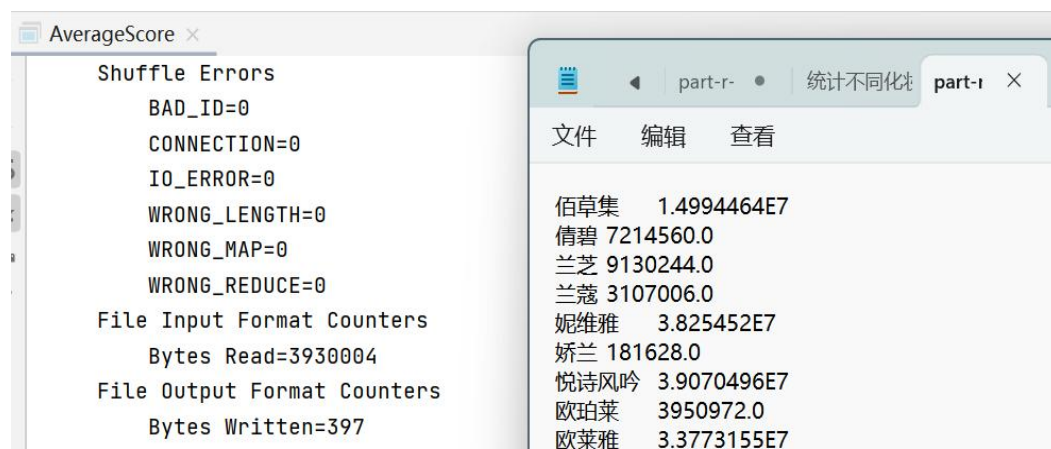


图 4.17 不同品牌销量

3. 化妆品品牌的评论数量排行；

对于第三个任务统计化妆品品牌的评论数量排行任务，我重写了 `steps()` 方法中，定义了一个包含一个 `MRStep` 的列表。在 `mapper()` 方法中，我们将输入行拆分为键和值，并将它们作为整数和字符串输出。在 `reducer()` 方法中，再对 Mapper 输出的值进行排序，并将排序后的值与键一起输出。最后在主函数中运行 `SortedOutput` 类的 `run()` 方法，输出排序后的化妆品品牌和对应的评论数量。

```
ext.write(stor  
ass AverageScd
```

佰草集	1158994.0
妮维雅	3704355.0
兰蔻	446919.0
倩碧	739967.0
兰芝	873534.0
娇兰	51092.0

图 4.18 不同品牌评论数排行

4. 统计不同类别的化妆品销量；

对于第四个任务统计不同类别的化妆品销量，做法与第一个任务相同。区别在于第六个任务将 `main_cat`（主类别）作为键，将 `sale_count` 作为 `value` 进行输出。

```
Averagescore x  
WRONG_MAP=0  
WRONG_REDUCE=0  
File Input Format Counters  
Bytes Read=3930004  
File Output Format Counters  
Bytes Written=94
```

其他	2.0187049E7
化妆品	5.1774376E7
套装	6.442199E7
护肤品	1.73513207E8

图 4.19 不同类别化妆品销量

5. 统计不同类别的化妆品销售额；

对于第五个任务，统计不同类别的化妆品销售额，其大致做法与第六个任务一致，主要区别在于 `values` 键的不同。

```
WRONG_REDUCE=0  
File Input Format Counters  
Bytes Read=3930004  
File Output Format Counters  
Bytes Written=88
```

其他	1997215.0
化妆品	5083545.0
套装	5685958.0
护肤品	1.5458546E7

图 4.20 不同类别化妆品销售额

6. 统计男士女士化妆品销售量比例；

Map 阶段，将数据按照性别（for_man）进行分组，同时统计每个性别下的商品销售量。在 Reduce 阶段，对每个性别下的商品销售量进行汇总，计算男性和女性的总销售量。最后，根据男性和女性的总销售量计算出男女化妆品销售量的比例。

WRONG_MAP=0	0	0.8183954242
WRONG_REDUCE=0	1	0.1816045758
File Input Format Counters		
Bytes Read=3930004		
File Output Format Counters		
Bytes Written=41		

图 4.21 男女化妆品销售量比例

3.5Hive 数据查询

本项目主要在 Linux 虚拟机上使用 Hive 进行建表、查询等操作，根据需求分析完成 Hive 数据查询任务。本项目设计的 Hive 查询任务主要有：

1. 销量排名前十的化妆品牌

对 store 进行分组，然后对每个组的销量进行统计。最后，我们需要按照销量降序排列，并选择销量前十的 store。

代码：

```
SELECT store, SUM(sale_count) as total_sales
FROM meizhuang
GROUP BY store
ORDER BY total_sales DESC
LIMIT 10;
```

2. 统计不同化妆品品牌的销量

计算每个商店的销售总额，并按照销售总额降序排列，这个做法与第一个一样，按照列名 store 分组即可。

代码：

```
SELECT store, SUM(spec_mount) as total_s
FROM meizhuang
GROUP BY store
ORDER BY total_s DESC;
```

3. 销售额前十的化妆品牌

统计销售额前 10 的景点，使用 order by sales DESC 对店铺名 store 进行排序，并使用 limit 10 输出前 10 条数据即可。

代码：

```
SELECT store, SUM(spec_mount) as total_sales
FROM meizhuang
GROUP BY store
ORDER BY total_sales DESC
LIMIT 10;
```

4. 各省份销量总和

对于第四个任务各省份化妆品订单销量总和，按省份分组，并使用聚合函数 sum 对销量 sales 进行求和。

代码：

```
select province,sum(spec_mount) Sales
from meizhuang
group by province;
```

5. 统计男士化妆品销量最多的品牌

对于第五个任务销量前 100 的景点分省分布情况。使用到嵌套查询，从表中筛选出“for_man”列值为 1 的记录，并按照“store”列的值进行分组，计算每个分组中“spec_mount”列的总和，然后按照总和降序排序，最后返回总和最高的一条记录。

代码：

```
hive> select store,sum(spec_mount)as totalman
> from meizhuang
> where for_man=1
> group by store
> order by totalman desc
> limit 1;
```

6. 销量从高到低的化妆品大类

对于第六个任务销售额前 20 的景点，将 price 和 sales 相乘并取别名 turnover，使用 order by 按照 turnover 进行降序排序，limit 20 取前 20。

代码：

```
SELECT update_time, id, title, price, sale_count, comment_count, store,
main_cat, sub_cat, spec_data, spec_unit, spec_mount, discount, for_man
FROM meizhuang
GROUP BY main_cat
ORDER BY SUM(sale_count) DESC;
```

7. 价格排名前十的商品

根据 titel 分组，price 排序，给出价格排名前十的商品。

代码:

```
SELECT title, price
FROM meizhuang
ORDER BY price DESC
LIMIT 10;
```

8. 评论数最多的前二十个商品

根据 comment_count 分组, price 排序, 给出价格排名前十的商品。

代码:

```
SELECT title, comment_count
FROM meizhuang
ORDER BY comment_count DESC
LIMIT 20;
```

9. 男士化妆品与女士化妆品的占比

表格中 for_man 一列值为 1 则代表男士化妆品, 0 代表女士化妆品。如果 for_man=1, 则将 comment 字段的值累加到 male_comments 中, 否则, 累加 0, 最后将累计的 male_comments 除以总评论的数量, 使用小的常数($1e-9$)防止除数为 0 的情况。女士化妆品也是运用同样的方法。

代码:

```
SELECT
    SUM(CASE WHEN for_man = 1 THEN comment_count ELSE 0 END)
AS male_comments,
    SUM(CASE WHEN for_man = 0 THEN comment_count ELSE 0 END)
AS female_comments,
    SUM(CASE WHEN for_man = 1 THEN comment_count ELSE 0 END) /
(SUM(comment_count) + 1e-9) AS male_ratio,
    SUM(CASE WHEN for_man = 0 THEN comment_count ELSE 0 END) /
(SUM(comment_count) + 1e-9) AS female_ratio
FROM meizhuang;
```

10. 化妆品品牌的平均销售额

对于第十个任务统计不同星级景区的平均评分, 做法类似第八个任务, 条件改为 title 中含有 '口红'。

代码:

```
SELECT store, COUNT(DISTINCT title) AS lipstick_types
FROM meizhuang
WHERE title LIKE '%口红%'
GROUP BY store;
```

11. 化妆品的品牌平均销售额

第一个子查询从名为 "your_table_name" 的表中选择 store、title 和

总销售额 (price * sale_count)。然后按照 store 和 title 进行分组，以计算每个商店中每个产品的总销售额。第二个子查询 (brand_counts) 从名为 "your_table_name" 的表中选择 store 和不同 title 的数量。然后按照 store 进行分组，以计算每个商店中不同产品的数量。最后的主查询从 brand_sales 子查询中选择 store 和平均销售额 (AVG(bs.total_sales))。然后使用 JOIN 操作将 brand_sales 和 brand_counts 子查询连接在一起，基于 store 列进行匹配。最后，按照 store 进行分组，以计算每个商店的平均销售额

代码：

```
WITH brand_sales AS (  
    SELECT store, title, SUM(price * sale_count) AS total_sales  
    FROM meizhuang  
    GROUP BY store, title  
)  
brand_counts AS (  
    SELECT store, COUNT(DISTINCT title) AS num_products  
    FROM meizhuang  
    GROUP BY store  
)  
SELECT bs.store, AVG(bs.total_sales) AS avg_sales  
FROM brand_sales bs  
JOIN brand_counts bc ON bs.store = bc.store  
GROUP BY bs.store;
```

12. 统计护肤品类别中销售额前十的商品

对于第十二个任务，筛选出主分类为“护肤品”的商品，并按照商品名称分组，计算每个商品的总销售额（价格乘以销售数量），然后按照总销售额降序排列，最后返回前 10 个结果。

代码：

```
SELECT title, SUM(price * sale_count) AS total_sales  
FROM meizhuang  
WHERE main_cat = '护肤品'  
GROUP BY title  
ORDER BY total_sales DESC  
LIMIT 10;
```

13. 套装类别中销量前十的商品

对于第十三个任务套装类别中销量前十的商品，用 WHERE 筛选出主分类为“套装”的商品，并按照商品名称进行分组，计算每个商品的销售数量总和（命名为 total_sales），然后按照销售数量降序排列，最后返回前 10 个结果。

代码：

```
SELECT title, SUM(sale_count) AS total_sales  
FROM meizhuang  
WHERE main_cat = '套装'
```

```
GROUP BY title
ORDER BY total_sales DESC
LIMIT 10;
```

14. 预售商品销量前十的品牌

对于第十四个任务统计预售商品销量前十的品牌，首先查询 spec_unit 列中是否含有‘预售’，再用 group by 按照 store 分组，统计销量并输出排名前十的品牌。

代码：

```
SELECT store AS brand, SUM(sale_count) AS total_sales
FROM meizhuang
WHERE spec_unit LIKE '%预售%'
GROUP BY store
ORDER BY total_sales DESC
LIMIT 10;
```

15. 输出护肤品大类中销量最高的子类

对于第十五个任务输出护肤品大类中销量最高的子类，使用 where 条件判断 main_cat='护肤品'，按子类 sub_cat 分组并使用 count 函数计数，统计输出销量最高的子类。

代码：

```
SELECT sub_cat, main_cat, SUM(sale_count) as total_sales
FROM meizhuang
WHERE main_cat = '护肤品'
GROUP BY sub_cat
ORDER BY total_sales DESC
LIMIT 1;
```

4. 数据可视化

4.1 前端设计

前端可视化模块是以一个网页的形式，通过 bootstrap 框架以及 echarts 可视化图表实现来展示所有的数据和实现所有的任务，主要包括八个部分：化妆品订单完成用户数省份城市分析，男士化妆品销量和销售额排名前三品牌，化妆品销量大类占比、化妆品品牌销售分析、各品牌均价分析、价格前十的化妆品数据展示、销售额大类占比和化妆品销量前十的子类分析。前端部分是主要展示了双十一美妆数据统计分析后的部分信息，将其通过各式 echarts 图表的形式展示出来；地图界面主要通过调用 echarts 进行设计，可以在地图上鼠标定位到具体省份城市查看相对应的美妆数据，根据标记点的大小判断省份美妆产品数据的大致情况；男士化妆

品销量和销售额排名前三品牌展示了品牌销量最高不代表销售额最高，也得出在男性群体中，妮维雅、欧莱雅、相宜本草这三个品牌最受欢迎；其他的 echarts 图表也都清晰的展示了相关的美妆数据分析。



.图 5.1 前端设计

4.2 美妆数据分析结果可视化

化妆品数据分析结果可视化主页通过可视化大屏的方式展示，使用各式 echarts 图表进行布局，展示了各化妆品销量、各品牌销售额以及均价等信息。



图 5.2 男士化妆品销量



图 5.3 化妆品品牌销量



图 5.4 化妆品销量大类占比



图 5.5 化妆品订单省份分析

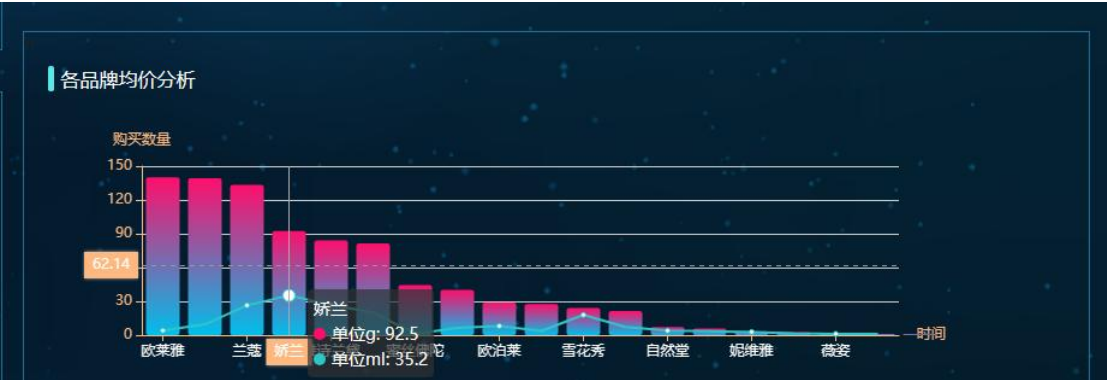


图 5.6 各品牌均价分析



图 5.7 销售额大类占比



图 5.8 销量前十子类分析

4.3 词云图模块设计

词云图模块设计主要通过 numpy 和 wordcloud 以及 echarts 完成。



图 5.9 销量前十子类分析

5. 结论

在完成这个基于 Hadoop 的双十一美妆数据分析系统项目的过程中，我遇到了很多困难和挑战。首先，由于数据量庞大，数据的清洗和处理成为了一个耗时且复杂的过程。我首先利用 Pandas 库对原始数据进行清洗和预处理，包括去除重复值、缺失值处理等。然后，我使用 Hive 和 MapReduce 来对大规模数据进行分布式处理和分析。在这个过程中，我学习了 Hive 的基本操作和语法，以及更熟练的使用 MapReduce 进行并行计算。最后，用

Echarts 库来实现数据的可视化展示,通过调整布局和样式,使得图表更加清晰和易于理解。对于 Hadoop 生态系统中的各种组件和技术,在老师的课堂上我并没有完全理解他的架构以及各种原理,导致我花费了很多的时间去学习和理解。最后,将数据可视化展示时,也遇到了一些布局和样式上的问题。为了解决这些困难,我在网上查找了很多相关资料,向同学寻求帮助。在这个项目中,我还学到了很多大数据相关的知识点。首先,我更加深入了解了 Hadoop 生态系统中的各个组件,包括 HDFS、MapReduce、Hive 等,利用它们进行大规模数据处理和分析。其次,我学习了 Pandas 库的基本使用方法,包括数据读取、清洗、转换和分析等。最后,我掌握了 Echarts 库的使用技巧,能够将数据以图表的形式直观地展示出来。通过这个项目的实践,我不仅巩固了大数据相关的理论知识,还提升了自己的实际操作能力。我学会了如何处理大规模的数据集,如何利用 Hadoop 生态系统中的工具进行数据分析和处理,以及如何将数据可视化展示。

总的来说,这次实训经历对我来说是一次非常有挑战性的经历。通过克服各种困难和挑战,我不仅学到了很多东西,还提升了自己的实践能力和解决问题的能力。我相信这些经验和知识将会对我的未来学习和工作产生积极的影响。

参考文献

[1] 左圆圆,王媛媛,蒋珊珊,徐榕荟.数据可视化分析综述[J].科技与创新,2019(11):82-83.

[2] 饶家玮.大数据时代下数据分析理念的辨析[J].商讯,2020(12):181.

[3] 李国杰,程学旗.大数据研究:未来科技及经济社会发展的重大战略领域——大数据的研究现状与科学思考[J].中国科学院刊,2012,27(06):647-657.

[4] 尹廷钧,李灵慧,周蕊.大数据挖掘中的数据分类算法综述[J].数字技术与应用,2021,39(01):102-104.

[5] 孟小峰,慈祥.大数据管理:概念、技术与挑战[J].计算机研究

与发展, 2013, 50(01):146-169.

[6] 孙大为, 张广艳, 郑纬民. 大数据流式计算: 关键技术及系统实例
[J]. 软件学报, 2014, 25(04):839-862.
