哈尔滨工程大学专业选修课研究报告

医药电商销售数据分析

专　业　名　称：软件工程

学　生　姓　名：张苗苗

陈昊菡

哈尔滨工程大学

2024年10月

摘　　要

本项目旨在对医药电商平台上维生素类药品进行全面的数据统计与分析，以深入理解市场动态、消费者行为及销售趋势。随着电子商务的迅速发展，医药电商已成为一个重要的销售渠道，特别是在维生素和保健品领域。通过对该领域的系统分析，本项目力求为相关企业提供数据驱动的决策支持。

首先，我们将对电商平台上的店铺进行分类统计，包括品牌官方店、旗舰店和专卖店等。通过清洗数据，去除重复项后，统计各类店铺的数量，并进一步分析其销售额占比。此部分分析将揭示不同类型店铺的市场份额，帮助我们了解消费者的购买偏好。例如，品牌官方店通常会因其信誉和产品质量而吸引更多消费者，而专卖店则可能因其专业性而获得特定人群的青睐。在统计各店铺销售额后，我们将计算每个店铺的销售额占总销售额的比例，并通过可视化手段（如饼图和条形图）展示这些结果。

接下来，我们将重点关注阿里健康大药房的销售情况。我们从数据集中筛选出阿里健康大药房的销售数据，计算其总销售额和销售量，并分析其增长趋势。

然后，我们将转向所有维生素类药品的整体分析。通过数据清洗和整理，统计出每种药品的种类及其数量，这有助于了解市场上维生素类药品的多样性和竞争情况。接下来，我们将聚合所有药品的销售额，计算每种药品的销售额占比，从而揭示哪些产品更受欢迎，为企业在产品推广和库存管理上提供指导。

接着，我们将对药品品牌进行深入分析。通过清洗数据并确保品牌名称统一，统计出不同品牌的种类及数量。这一分析将为了解市场上主要品牌的竞争态势提供基础数据。进一步地，我们将计算每个品牌的销售额占比，并通过可视化手段展示结果。

最后我们运用ARIMA模型预测天猫维生素类药品未来三个月的销售总额并绘制拟合曲线，评估模型性能和误差。

关键词：医药电商；数据分析；AMRIA模型；可视化

目录

[第1章 应用背景 1](#_Toc179143789)

[1.1 课题的背景与目的 1](#_Toc179143790)

[第2章 关键技术介绍 2](#_Toc179143791)

[2.1数据收集与爬虫 2](#_Toc179143792)

[2.2 数据清洗与处理 2](#_Toc179143793)

[2.2.1 数据清洗技术 2](#_Toc179143794)

[2.2.2数据转换技术 3](#_Toc179143795)

[2.2.3数据处理工具 4](#_Toc179143796)

[2.3数据可视化 4](#_Toc179143797)

[2.3.1常见的数据可视化类型 4](#_Toc179143798)

[2.3.2数据可视化工具与技术 5](#_Toc179143799)

[2.4数据分析与建模 5](#_Toc179143800)

[第3章 项目相关设计图 6](#_Toc179143801)

[3.1架构图 6](#_Toc179143802)

[3.1.1用户界面 6](#_Toc179143803)

[3.1.2后端服务 7](#_Toc179143804)

[3.1.3数据存储 7](#_Toc179143805)

[3.1.4数据处理 8](#_Toc179143806)

[3.1.5交互流程 8](#_Toc179143807)

[3.2流程图 9](#_Toc179143808)

[3.2.1用户登录 9](#_Toc179143809)

[3.2.2用户操作 10](#_Toc179143810)

[3.2.3数据分析并生成报告 10](#_Toc179143811)

[3.3功能结构图 11](#_Toc179143812)

[3.3.1用户管理 11](#_Toc179143813)

[3.3.2数据管理 11](#_Toc179143814)

[3.3.3数据分析 12](#_Toc179143815)

[3.3.4系统监控 12](#_Toc179143816)

[第4章功能模块分析和代码介绍 12](#_Toc179143817)

[4.1店铺分析 12](#_Toc179143818)

[4.1.1店铺种类统计 12](#_Toc179143819)

[4.1.2店铺销售额占比 13](#_Toc179143820)

[4.1.3阿里健康大药房销售情况 16](#_Toc179143821)

[4.2药品分析 17](#_Toc179143822)

[4.2.1药品种类统计 17](#_Toc179143823)

[4.2.2药品销售额占比 18](#_Toc179143824)

[4.2.3药品月销售额统计图 18](#_Toc179143825)

[4.3药品品牌分析 19](#_Toc179143826)

[4.3.1药品品牌种类统计 19](#_Toc179143827)

[4.3.2药品品牌销售额占比 20](#_Toc179143828)

[4.3.3药品品牌销售分析 21](#_Toc179143829)

[4.4某药品未来三个月销售预测 22](#_Toc179143830)

[第5章软件功能演示 25](#_Toc179143831)

[5.1店铺种类统计 25](#_Toc179143832)

[5.2店铺销售额占比统计 26](#_Toc179143833)

[5.3阿里健康大药房销售情况 27](#_Toc179143834)

[5.3.1阿里健康大药房打折情况 28](#_Toc179143835)

[5.3.2阿里健康大药房月销售变化 28](#_Toc179143836)

[5.4药品月销售额统计 28](#_Toc179143837)

[5.5药品品牌销售分析 34](#_Toc179143838)

[5.6药品销售额预测 36](#_Toc179143839)

[结论 38](#_Toc179143840)

# 第1章 应用背景

## 1.1 课题的背景与目的

近年来，随着互联网技术的迅速发展和国家政策的逐步开放，医药电商正以惊人的速度崛起，成为现代医疗服务体系中不可或缺的重要组成部分。这一变革不仅改变了传统医药销售模式，也为企业提供了大量宝贵的销售数据。这些数据通过深入挖掘和分析，能够为企业提供清晰的市场动态和消费者偏好的重要信息，使得企业能够及时调整其产品和营销策略，以适应瞬息万变的市场需求。

在这一背景下，企业可以利用数据分析技术优化库存管理。通过对历史销售数据的精准分析，企业能够更好地预测未来的产品需求，从而合理调整库存水平，降低过剩或缺货的风险。这种科学的库存管理不仅能够提升资金运转效率，还能增强企业在市场竞争中的灵活性。

与此同时，用户行为分析也扮演着至关重要的角色。通过对用户在平台上的互动、购买习惯以及反馈意见进行深入研究，企业可以显著提升客户体验。这包括改善平台的用户界面和功能，以便更好地满足客户需求，从而提高客户满意度与忠诚度。个性化的产品推荐能够有效提升转化率和销售额，使得用户在购物过程中感受到更大的便利与价值。

从更广泛的角度来看，医药电商销售数据的分析不仅提升了企业的决策科学性，还增强了整个行业的透明度。透明的市场环境使得各方利益相关者能够更准确地了解市场情况，进而促进了消费者对电商平台的信任。这种信任是建立长期客户关系的基石，有助于企业实现可持续发展。此外，数据分析的广泛应用也推动了大数据、人工智能等相关技术的发展，进一步提升了医药电商领域的科技水平。

总之，医药电商销售数据分析在提升企业竞争力、促进行业健康发展和推动科技创新方面具有重要的现实意义。它不仅为企业提供了应对市场挑战的有力工具，也为未来的可持续发展奠定了坚实的基础。在这个数据驱动的时代，谁能够更好地利用数据，谁就能够在激烈的市场竞争中立于不败之地。

# 第2章 关键技术介绍

本章将对数据分析所采用的关键技术作一个全面概括的介绍，从数据收集开始介绍，详细阐述数据清洗与处理，可视化技术以及数据分析与建模。

## 2.1数据收集与爬虫

数据收集是大数据分析的关键第一步，有效的数据收集技术能够确保数据的质量和完整性。在数据源类型方面，结构化数据通常存储在关系型数据库中，便于查询和管理；非结构化数据如文本、图像和视频则存储在文件系统或NoSQL数据库中；而半结构化数据如JSON和XML格式则具有一定的结构，但不完全符合传统数据库的模式。

常用的数据收集技术包括数据抓取、API接口、传感器与物联网（IoT）、数据库导入以及日志文件收集。数据抓取通过编写脚本从网页提取数据，常用工具如Beautiful Soup和Scrapy。API接口则利用开放的API直接获取数据，例如社交媒体平台和金融数据提供商提供的接口。通过传感器收集实时数据，如环境监测和智能设备状态，数据会实时传输至存储系统。此外，从现有数据库中导入数据和收集应用程序生成的日志文件也是常见的方法，后者往往使用日志分析工具（如ELK Stack）进行处理。

## 2.2 数据清洗与处理

数据清洗与处理的目的是确保分析所用数据的质量与可靠性，从而为后续的决策提供坚实的基础。这一过程通过识别和纠正数据中的错误，来提高数据的准确性和完整性。同时，去除不相关的数据和异常值能够减少对分析结果的干扰，使得分析更加集中于重要信息。数据清洗还涉及将不同来源的数据进行标准化，以统一格式，从而促进数据整合与比较，并且通过删除包含缺失值的记录或使用插补方法来填充缺失值，避免因缺失数据导致的偏差。清洗后的数据更易于理解和分析，提高了数据的可用性，便于用户进行深入探索和挖掘，从而支持管理层做出基于事实的决策，提升分析效率，增强模型预测的准确性，转化为有价值的商业洞察，推动业务增长，帮助企业适应快速变化的市场环境。因此，这一过程在数据分析中至关重要。

### 2.2.1 数据清洗技术

#### 缺失值处理

缺失值是数据集中常见的问题，处理缺失值的方法有多种：

**插补法**：

1. **均值插补**：对数值型数据，可以用该特征的均值替换缺失值。这适用于数据分布较为平稳的情况。
2. **中位数插补**：对于受极端值影响较大的数据，中位数更能代表中心趋势，因此可以用中位数替代缺失值。
3. **众数插补**：对于分类数据，使用最常见的类别（众数）来填补缺失值。

**删除法**：

1. **删除记录**：如果缺失值数量很少且随机分布，可以考虑直接删除含缺失值的记录。
2. **删除特征**：如果某个特征的大部分值缺失，考虑将该特征从数据集中删除，以减少噪声。

#### 重复数据检测

重复数据可能导致分析结果不准确，需要采取措施识别和处理这些数据：

**去重算法**：

1. **基于哈希**：通过计算每条记录的哈希值，将相同哈希值的记录视为重复数据。
2. **相似性匹配**：使用字符串相似性算法（如Levenshtein距离）来识别内容相似的记录，特别适用于非结构化数据。

### 2.2.2数据转换技术

#### 数据标准化

数据标准化有助于将不同量纲的数据转换到可比的范围，常用方法包括：

**归一化（MinMax Scaling）**：将数据缩放到[0, 1]范围内，公式为：



这种方法适用于需要保持原始数据分布的情况。

**Zscore标准化：**将数据转换为均值为0、标准差为1的分布，公式为：



适合于在分析中需要处理异常值的场景。

### 2.2.3数据处理工具

#### 编程语言

Python：

pandas：强大的数据处理库，提供DataFrame对象，便于执行数据清洗、处理和分析。常用功能包括数据筛选、缺失值填充、数据类型转换等。

NumPy：主要用于数值计算，提供数组对象和各种数学函数，适合高效处理大规模数据。

R：

dplyr：用于数据操作的R包，提供简洁的语法，支持数据过滤、排序、汇总等操作。

tidyr：专注于数据整理，通过“宽格式”和“长格式”之间的转换，提高数据可用性。

#### 数据处理框架

Spark SQL：支持使用SQL语言查询数据，适合处理结构化数据。

DataFrames API：类似于pandas，支持分布式数据处理，能够处理大规模数据集。

Hadoop MapReduce：适用于批量处理，能够将任务拆分为多个子任务并行执行，适合大数据量的处理需求。

## 2.3数据可视化

数据可视化是将复杂的数据转化为直观的图形和图表，以便于人们快速理解和分析信息，其目的在于揭示数据中的模式、趋势和关系，帮助决策者更有效地识别关键见解。通过清晰的视觉呈现，数据可视化不仅提高了信息传达的效率，还能增强数据的吸引力，使得用户更容易关注和记住重要内容。这种方式能够推动沟通与合作，促进团队之间的讨论，从而在数据驱动的决策过程中发挥重要作用。因此，数据可视化在现代商业与研究中具有不可或缺的意义。

### 2.3.1常见的数据可视化类型

根据数据特性和分析需求，常用的可视化类型包括：

柱状图：适合比较不同类别的数据，便于观察各类别之间的差异。

折线图：主要用于表现时间序列数据的变化趋势，能够清晰展示数据随时间的波动。

饼图：用于显示各部分在整体中所占的比例，适合类别较少的情况。

散点图：展示两个变量之间的关系，常用于识别相关性和分布模式。

热力图：使用颜色深浅显示数据值的大小，适合处理大规模数据集的分布情况。

### 2.3.2数据可视化工具与技术

多种工具和技术被广泛用于数据可视化，包括：

Tableau：一个强大的商业智能工具，支持用户创建交互式仪表板，适合各种数据源的连接与分析。

Power BI：微软的商业分析工具，为用户提供丰富的可视化选项和报告功能。

D3.js：基于JavaScript的库，允许开发者创建高度自定义的可视化，适合需要个性化展示的场景。

Python库（如Matplotlib和Seaborn）：适用于科研和数据分析，为用户提供灵活的绘图功能。

## 2.4数据分析与建模

数据分析与建模的目的是通过系统地整理和解读数据，识别潜在模式和趋势，以支持决策和策略制定。通过应用统计学、机器学习等方法，分析与建模能够揭示数据背后的关系，为企业提供深入的洞察，帮助其优化运营、提高效率并预测未来表现。这不仅使组织能够根据数据做出更加科学和精准的决策，还能够推动创新和竞争优势的形成。因此，数据分析与建模在当今数据驱动的环境中具有重要的战略意义。

数据分析：

1. 数据收集是这一过程的起点，数据可以来自多种渠道，如调查、传感器和在线交易，包含定量数据和定性数据。
2. 数据清洗是关键步骤，处理缺失值、异常值，并确保数据格式一致，以提高数据质量。
3. 在数据探索阶段，通过描述性统计和可视化手段，可以识别数据中的模式和趋势，相关性分析也帮助理解变量间的关系。

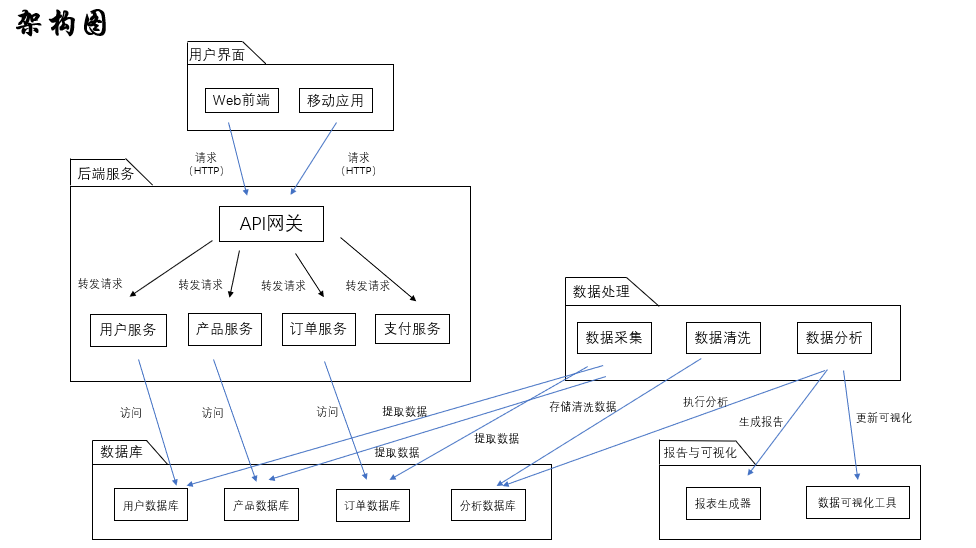
建模过程：

1. 选择适合的问题性质的模型，如线性回归、逻辑回归、决策树等，并使用训练数据来拟合模型，通过优化算法调整参数以提高预测准确性。
2. 模型评估至关重要，利用验证集或测试集来评估模型性能，常用指标包括准确率、精确率和均方误差。
3. 为了提升模型表现，还需进行模型优化，采用超参数调优和特征选择等方法。
4. 将模型部署到生产环境中后，需要进行实时监控，以确保其在实际应用中的有效性和可靠性。

# 第3章 项目相关设计图

## 3.1架构图

该架构图展示了医药电商销售数据分析系统的整体结构，由四个主要部分组成：用户界面、后端服务、数据存储和数据处理。



### 3.1.1用户界面

用户界面是用户与系统交互的入口，主要包括两个部分：

1.Web前端：

功能：提供一个可视化界面，用户可以通过浏览器进行操作。

主要特性：用户注册与登录功能。

产品浏览和搜索。

订单管理与查询。

数据分析报告的展示（如销售趋势图、用户行为分析等）。

2.移动应用：

功能：允许用户在移动设备上访问系统，与 Web 前端功能相似，但针对移动体验进行了优化。

主要特性：提供推送通知（如订单状态更新）。

界面适配小屏幕，确保良好的用户体验。

离线访问部分功能，提升可用性。

### 3.1.2后端服务

后端服务处理前端请求，执行核心业务逻辑，主要分为以下模块：

1.API 网关：

功能：作为所有客户端请求的统一入口，负责路由、负载均衡和安全认证。

主要特性：接收来自 Web 前端和移动应用的 HTTP 请求。

将请求分发到正确的服务层组件。

处理跨域请求和身份验证（如 JWT 验证）。

2.服务层：

负责具体的业务处理，包含多个独立服务：

1. 用户服务：

功能：管理用户信息和权限。

主要功能：用户注册、登录、退出。

个人资料编辑与查询。

权限控制（如角色管理）。

(2)产品服务：

功能：处理与产品相关的所有业务。

主要功能：添加、编辑和删除产品信息。

查询产品详情和库存情况。

支持产品分类和搜索功能。

(3)订单服务：

功能：管理用户的订单信息。

主要功能：创建新订单、更新订单状态（如支付、发货）。

查询订单历史记录。

处理订单取消和退款请求。

### 3.1.3数据存储

数据存储模块用于持久化系统中的所有数据，确保数据的安全和高效访问：

1.用户数据库：

功能：存储所有用户的个人信息和账户数据。

数据类型：用户ID、用户名、密码（加密存储）、联系方式等。

2.产品数据库：

功能：保存所有产品的信息和状态。

数据类型：产品ID、名称、描述、价格、库存数量等。

3.订单数据库：

功能：记录所有交易的详细信息。

数据类型：订单ID、用户ID、购买产品、订单状态、创建时间等。

### 3.1.4数据处理

数据处理模块负责从不同的数据源中提取和分析数据，以支持决策和优化业务流程：

1.数据采集：

功能：自动化从数据库中提取所需数据，为分析提供原始数据。

主要特性：定时任务或触发器，根据需求定期更新数据。

提取用户行为数据、销售数据等。

2.数据分析：

功能：对采集到的数据进行分析和处理，生成报告。

主要特性：使用统计方法和机器学习算法，识别趋势和模式。

生成可视化报告，帮助管理层做出决策。

提供 API 接口，支持其他服务调用分析结果。

### 3.1.5交互流程

1. 用户请求：

用户通过 Web前端 或 移动应用 向 API 网关 发送 HTTP 请求，进行各种操作，比如查询产品或下订单。

2. 服务调用：

API 网关 将接收到的请求根据 URL 和请求类型转发到对应的服务层组件（如用户服务、产品服务或订单服务），并返回处理结果。

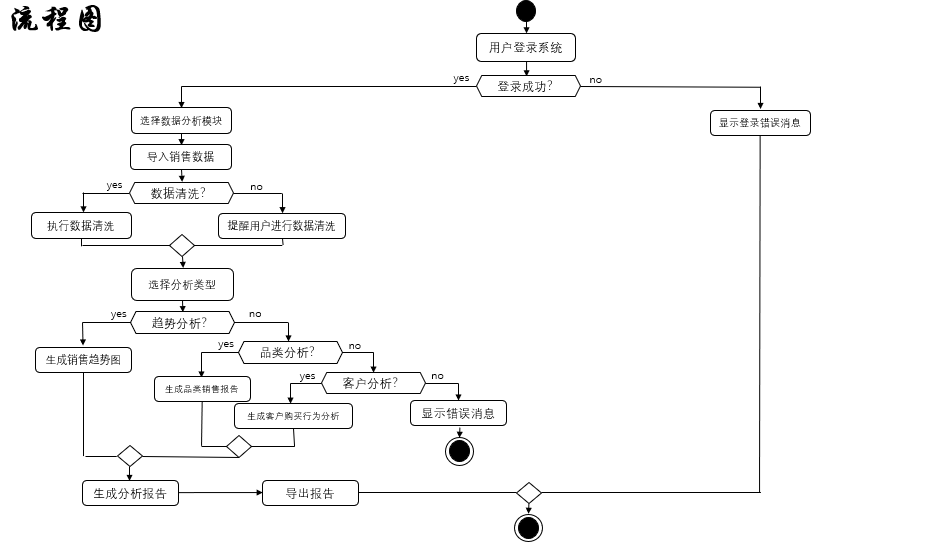
3. 数据提取与分析：

数据采集 模块从 用户数据库、产品数据库 和 订单数据库 中提取数据，准备进行分析。

数据分析 模块对提取的数据进行深入分析，生成图表和报告，并将结果反馈给用户。

## 3.2流程图

该流程图描绘了用户在医药电商销售数据分析系统中的操作流程，包含从用户登录到报告生成的每一个环节，确保用户能够顺畅地完成所需的数据分析任务。



### 3.2.1用户登录

1. 用户登录系统

目的：确保只有授权用户可以访问系统。

操作步骤：用户在登录界面输入用户名和密码。

系统验证输入的凭证。

2. 登录成功？

判断分支：成功：用户被重定向到主界面，获得系统功能的访问权限。

失败：系统提示错误信息（如“用户名或密码错误”），并允许用户重新输入。

### 3.2.2用户操作

1. 选择数据分析模块

操作步骤：在主菜单中，用户选择“数据分析”选项，进入数据分析功能界面。

2. 导入销售数据

目的：让用户上传需要分析的销售数据。

操作步骤：用户点击“上传数据”按钮，并从本地选择文件进行上传。

系统支持多种文件格式（如CSV、Excel），并进行格式验证。

3. 数据清洗？

判断分支：是：系统自动执行数据清洗，包括去除重复记录和处理缺失值。

否：系统弹出提示框，建议用户进行数据清洗，并说明清洗的重要性。

4. 选择分析类型

操作步骤：

用户在下拉菜单中选择所需的分析类型，如：

趋势分析：分析销售随时间的变化。

品类分析：比较不同产品品类的销售表现。

客户分析：研究客户的购买行为和偏好。

### 3.2.3数据分析并生成报告

1. 数据分析

操作步骤：根据用户选择的分析类型，系统开始处理数据并生成相应的分析结果：

趋势分析：生成时间序列图表，展示销售变化。

品类分析：提供各品类的销售报告和关键指标。

客户分析：分析客户购买频率和偏好，生成详细报告。

2. 生成分析报告

目的：将分析结果整合为一份完整的报告，便于用户查看和分享。

操作步骤：系统汇总分析结果，形成报告。

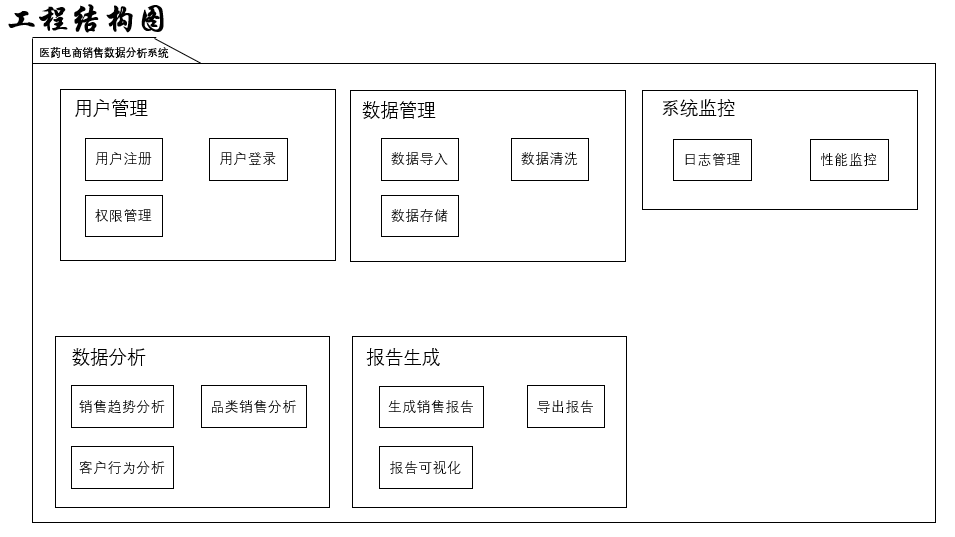
3. 导出报告

操作步骤：用户选择报告导出的格式（如PDF或Excel）。

点击“导出”按钮，系统生成文件并提供下载链接。

## 3.3功能结构图

该功能结构图清晰展示了医药电商销售数据分析系统的主要功能模块及其子功能，为开发团队提供系统的整体结构视图，有助于后续的开发和维护工作。



### 3.3.1用户管理

1.用户注册：

用户填写注册表单，提交个人信息。

系统发送验证邮件以确认用户身份。

2.用户登录：

已注册用户通过输入凭证登录系统。

3.权限管理：

管理员可以为不同角色分配权限，如查看、编辑、删除等，确保系统安全性。

### 3.3.2数据管理

1.数据导入：

用户点击“导入数据”，选择文件并上传，系统支持批量导入。

2.数据清洗：

提供自动清洗和手动清洗两种方式，用户可选择更适合的方式。

系统会显示清洗进度和结果日志。

3.数据存储：

清洗后的数据被存储在数据库中，确保数据的安全性和可访问性。

### 3.3.3数据分析

1.销售趋势分析：

利用时间序列分析工具，生成趋势图，识别高峰和低谷。

2.品类销售分析：

通过图表和数据透视表展示各品类的表现，便于决策。

3.客户行为分析：

深入挖掘客户的购买习惯，为个性化营销提供数据支持。

3.3.4报告生成：

1.生成销售报告：

系统根据分析结果自动生成销售报告，包含图表和文字描述。

2.导出报告：

用户可选择导出格式，系统支持一键导出功能。

3.报告可视化：

可视化工具将数据以图形化的方式呈现，提高报告的可读性。

### 3.3.4系统监控

1.日志管理：

记录所有用户操作和系统事件，便于审计和异常检测。

2.性能监控：

实时监控系统负载和响应时间，确保系统稳定运行。

# 第4章功能模块分析和代码介绍

## 4.1店铺分析

### 4.1.1店铺种类统计

通过pandas的read\_excel函数导入数据，使用value\_counts函数对“shop\_name”列计数统计，得出共包含26家店铺

代码：

import pandas as pd

data=pd.read\_excel('F:\\2022全国大学生数据分析A题\\data\\data.xlsx')

data["shop\_name"].value\_counts()

### 4.1.2店铺销售额占比

首先通过在Excel将“discount”列中的空值全部填充为“10折”，即无打折，接着根据公式计算出销售额，公式如下：



使用group\_by函数计算各店铺销售总额并求其占比。

代码：

data["discount"].value\_counts()

import pandas as pd

discount\_num = []

for i in range(len(data["shop\_name"])):

discount\_num.append(0)

for i in range(len(data["shop\_name"])):

if data.loc[i, "discount"] == "10折":

discount\_num[i] = 1.0

elif data.loc[i, "discount"] == "9.5折":

discount\_num[i] = 0.95

elif data.loc[i, "discount"] == "9折":

discount\_num[i] = 0.90

elif data.loc[i, "discount"] == "8.5折":

discount\_num[i] = 0.85

# 创建包含折扣的 DataFrame

df\_1 = pd.DataFrame({'shop\_name': data['shop\_name'], 'discount\_num': discount\_num})

df\_1.to\_excel('D:\\BaiduNetdiskDownload\\data\_1.xlsx', sheet\_name='sheet1', index=None)

total\_price = []

for i in range(len(data["shop\_name"])):

total\_price.append(0)

for i in range(len(data["shop\_name"])):

price = data.loc[i, "price"]

sold = float(data.loc[i, "sold"])

discount = discount\_num[i]

total\_price[i] = float(price) \* float(sold) \* float(discount)

# 创建包含总价和商店名称的 DataFrame

df\_2 = pd.DataFrame({

'shop\_name': data['shop\_name'],

'total\_price': total\_price

})

df\_2.to\_excel('D:\\BaiduNetdiskDownload\\data\_2.xlsx', sheet\_name='sheet1', index=None)

# 读取结果

data1 = pd.read\_excel('D:\\BaiduNetdiskDownload\\data\_2.xlsx')

data1

这段代码的核心功能是：

从一个包含商店信息的 DataFrame 中提取折扣信息，并将其转换为数值格式。

计算基于价格、销售量和折扣的总销售额。

将折扣信息和总销售额分别导出到 Excel 文件，方便后续的数据分析和可视化。total\_shop = data1.groupby("shop\_name")["total\_price"].sum().reset\_index()

total\_shop=total\_shop.sort\_values("total\_price",ascending=True)

total\_shop

这段代码的主要功能是对一个名为 data1 的数据集进行分组和汇总，计算每个商店的总价格，并按总价格升序排列。

total\_shop = data1.groupby("shop\_name")["total\_price"].sum().reset\_index()

total\_shop

这段代码的主要功能是通过对数据框中的“shop\_name”列进行分组，计算每个商店的“total\_price”总和，并将结果返回为一个新的数据框。

total=sum(total\_shop["total\_price"])

total\_shop\_new = total\_shop

for i in range(len(total\_shop["total\_price"])):

total\_shop\_new["total\_price"][i]=(total\_shop\_new["total\_price"][i])/total

total\_shop\_new

这段代码的主要功能是在给定商店的总价数据中计算并返回每个商店总价在总体总价中的占比，从而实现对商店数据的归一化处理。

total\_shop=total\_shop.sort\_values("total\_price",ascending=True)

total\_shop

这段代码的主要功能是对 total\_shop 数据集按照 total\_price 列进行升序排序，并更新原数据集。

import matplotlib.pyplot as plt

import seaborn as sns

# 设置字体参数，以便显示中文

plt.rcParams['font.family'] = ['sansserif']

plt.rcParams['font.sansserif'] = ['SimHei']

# 调节图形大小，宽，高

plt.figure(figsize=(9, 6))

# 定义饼状图的标签

labels = [u"阿里健康大药房", u"天猫国际进口超市", u"其它"]

# 每个标签占多大，会自动去算百分比

sizes = [0.450, 0.237, 0.313]

# 颜色设置

colors = ['red', 'yellowgreen', 'lightskyblue']

# 将某部分爆炸出来

explode = (0.1, 0, 0) # 仅仅“爆炸”第一块

# 绘制饼状图

plt.pie(sizes, explode=explode, labels=labels, colors=colors,

autopct='%1.1f%%', shadow=True, startangle=140)

# 确保饼图为圆形

plt.axis('equal')

# 显示图形

plt.title(u"市场份额分布")

plt.show()

这段代码的主要功能是使用matplotlib库绘制并展示一个中文饼状图，清晰地显示不同品牌在市场中的占比。

plt.figure(figsize=(15,6),dpi=200)

x=total\_shop["shop\_name"]

plt.bar(x, total\_shop["total\_price"], color='#87CEFA')

# plt.xlabel('时间')

plt.xticks(total\_shop["shop\_name"],rotation=60)

plt.ylabel('销售额占比')

plt.title('各药店销售额占比')

plt.show()

这段代码的主要功能是创建一个清晰的条形图，展示各个药店的销售额占比。

### 4.1.3阿里健康大药房销售情况

data2=data[data["shop\_name"]=="阿里健康大药房"]

data2["discount"].value\_counts()

从包含多家商店订单信息的数据框中筛选出特定商店“阿里健康大药房”的所有记录。

然后，统计并显示该商店不同折扣值的频率分布情况。

plt.figure(figsize=(9,6))

labels = [u"9.5折",u"9折",u"8.5折",u"无打折"]

sizes = [10768,4953,574,663]

colors = ['orange','yellowgreen','lightskyblue','pink']

explode = (0.05,0,0,0)

patches,l\_text,p\_text = plt.pie(sizes,explode=explode,labels=labels,colors=colors,

labeldistance = 1.1,autopct = '%3.1f%%',shadow = False,

startangle = 90,pctdistance = 0.6)

for t in l\_text:

t.set\_size(12)

for t in p\_text:

t.set\_size(12)

plt.axis('equal')

plt.title('阿里健康大药房打折情况', size=15)

plt.legend()

plt.show()

绘制一个饼图清晰地展示阿里健康大药房的折扣情况。

df3=data2.sort\_values(by="sold" , ascending=False)

df3

对数据框data2根据sold这一列的数值进行降序排序，并将结果存储在变量df3中。

data3=pd.read\_excel('D:\BaiduNetdiskDownload\datanew.xlsx')

data4=data3[data3["shop\_name"]=="阿里健康大药房"]

total\_month= data4.groupby("date\_time")["total\_price"].sum().reset\_index()

total\_month

从Excel文件中读取数据，筛选出特定商店（“阿里健康大药房”）的销售记录，并按日期计算每一天的销售总额。

plt.figure(figsize=(15,6),dpi=200)

x=["202001","202002","202003","202004","202005","202006","202007","202008","202009","202010","202011","202012",

"202101","202102","202103","202104","202105","202106","202107","202108","202109","202110","202111","202112"]

y=total\_month["total\_price"]

plt.plot(x, y, color='orange')

plt.xlabel('时间')

plt.xticks(x,rotation=70)

plt.ylabel('当月销售额')

plt.title('阿里健康大药房20202021年各月份销售额')

plt.show()

可视化阿里健康大药房在2020年和2021年期间每个月的销售额变化。通过绘制折线图，用户可以直观地观察到销售额的趋势。

data5=data4.groupby("title")["sold"].sum().reset\_index()

data5.sort\_values(by="sold" , ascending=False)

这段代码的主要功能是对销售数据进行聚合和排序。首先，它根据商品的标题汇总销售数量，并计算每种商品的总销量；随后，结果按销量降序排列，以便能直观查看哪些商品的销售量最高。

data5=data4.groupby("title")["total\_price"].sum().reset\_index()

data5.sort\_values(by="total\_price" , ascending=False)

这段代码的目的是从 data4 数据框中计算每个标题（title）的总销售价格（total price），并将结果按照总价格从高到低进行排序。

## 4.2药品分析

### 4.2.1药品种类统计

import pandas as pd

data=pd.read\_excel('D:\BaiduNetdiskDownload\datanew.xlsx')

data["id"].value\_counts()

读取Excel文件并统计特定列（"id"列）中每个唯一值的出现次数。

### 4.2.2药品销售额占比

total\_id=data.groupby("id")["total\_price"].sum().reset\_index()

total\_id=total\_id.sort\_values("total\_price",ascending=False)

total\_id

这段代码的主要功能是对给定的数据集进行分析，通过依据“id”进行分组，并计算每个“id”的“total\_price”总和。最终输出一个新的数据框，其中包含每个“id”及其对应的总价格，并按总价格降序排列，便于查看哪些“id”对应的总价格是最高的。

id\_ten=total\_id.head(10)

id\_ten

total=total\_id["total\_price"].sum()

total

计算并返回给定数据框total\_id中“total\_price”列的总和，结果存储在变量total中

id\_sold\_percent=[]

for i in range(len(id\_ten["id"])):

id\_sold\_percent.append(0)

for i in range(len(id\_ten["id"])):

id\_sold\_percent[i]=list(id\_ten["total\_price"])[i]/total

id\_sold\_percent

这段代码主要功能是计算并存储每个项目的销售占比。它从一个包含项目 ID 和对应总价格的数据结构中提取总价格，然后计算每个项目的总价格在整体总价格中的比例，最终结果存储在 id\_sold\_percent 列表中。

### 4.2.3药品月销售额统计图

data\_yp1=data[data["id"]==4169804230645]

#data\_yp1

total\_month\_yp1= data\_yp1.groupby("date\_time")["total\_price"].sum().reset\_index()

total\_month\_yp1

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

plt.rcParams['font.family'] = ['sansserif']

plt.rcParams['font.sansserif'] = ['SimHei']

plt.figure(figsize=(20,30),dpi=200)

ax=plt.subplot(5,2,1)

x\_1=["20207","20208","20209","202010","202011","202012","202101","202102","202103","202104","202105","202106",

"202107","202108","202109","202110","202111","202112"]

y\_1=total\_month\_yp1["total\_price"]

plt.plot(x\_1, y\_1, color='orange')

plt.xticks(x\_1,rotation=70)

plt.ylabel('当月销售额')

plt.title('id:4169804230645月份销售额')

plt.show()

绘制十种药品的月销售额折线图，这里仅给出id为4169804230645药品月销售额折线图代码，代码包含以下功能：

1.数据筛选：代码首先通过 data[data["id"]==某个ID] 语句筛选出特定ID的销售数据。这里包含了10个不同的ID（如：4169804230645, 4521420762240等）。data\_yp1, data\_yp2, ..., data\_yp10 依次保存这些筛选出的数据。

2.数据聚合：每个ID的数据会根据 date\_time 列进行分组，并计算 total\_price 的总和。使用 groupby 和 sum() 方法后，通过 reset\_index() 重新设置索引，方便后续处理。

3.可视化设置：引入绘图库，包括 Matplotlib、Seaborn 和 Plotly Express。设置中文显示，确保图表能够正常显示中文字符。

4.绘制每月销售额图：使用循环或手动为每个ID创建绘图：定义 x 轴（时间）和 y 轴（销售额）。

plt.plot() 用于绘制一条折线图表示销售额的变化，各个ID的销售额图表分别为不同区块绘制。

## 4.3药品品牌分析

### 4.3.1药品品牌种类统计

import pandas as pd

data=pd.read\_excel('D:\\BaiduNetdiskDownload\\datanew.xlsx')

import numpy as np

data.dropna(axis=0,subset = ['parameter'],inplace=True)

data = data.reset\_index(drop=True)

brand=[]

for i in range(len(data["parameter"])):

brand.append(0)

for i in range(len(data["parameter"])):

str1=data.loc[i,"parameter"].replace("||",",")

#print(i)

parameter\_dict = {}

content=str1.splitlines()

str2=content[0].split(',')

for j in str2:

k, v = j.split(':')

parameter\_dict[k] = v

brand\_parameter=parameter\_dict["品牌"]

brand[i]=brand\_parameter

data["brand"]=brand

data

这段代码的功能是处理Excel数据，目的是从中提取特定的品牌信息并生成一个新的数据列

data["brand"].value\_counts().head(10)

统计并提取数据集中出现频率最高的十个品牌。

total\_brand= data.groupby("brand")["total\_price"].sum().reset\_index()

total\_brand.sort\_values("total\_price",inplace=True, ascending=False)

total\_brand

从一个包含品牌和其对应总价格的数据集中，计算每个品牌的总价格，并按照总价格的降序排列，便于分析和比较不同品牌的总销售额。

### 4.3.2药品品牌销售额占比

total=total\_brand["total\_price"].sum()

ten\_total\_brand=total\_brand.head(10)

ten\_total\_brand

计算总价格和获取前十个品牌的数据。

brand\_sold\_percent=[]

for i in range(len(ten\_total\_brand["brand"])):

brand\_sold\_percent.append(0)

for i in range(len(ten\_total\_brand["brand"])):

brand\_sold\_percent[i]=list(total\_brand["total\_price"])[i]/total

brand\_sold\_percent

整段代码的主要功能是计算并存储每个品牌相对于总销售额的销售百分比。首先初始化一个列表以存储这些百分比，然后通过两次循环，分别为每个品牌预留位置并计算其对应的销售比例，最终生成存储这些百分比的列表。

plt.figure(figsize=(8,6),dpi=100)

x\_10=["swisse","CONBA/康恩贝","CENTRUM/善存","elevit/爱乐维","星鲨","伊可新","养生堂","BY－HEALTH/汤臣倍健","FANCL","朗迪"]

y\_10=[18.97,8.25,7.59,6.78,6.61,4.03,3.97,3.09,2.53,2.21]

plt.barh(x\_10, y\_10, color='cyan')

plt.xlabel('品牌销售额占比 单位：%')

plt.ylabel('品牌')

plt.title('销售额占比最高的十大品牌销售额占比')

plt.show()

可视化展示十大品牌的销售额占比

### 4.3.3药品品牌销售分析

data\_pp1=data[data["brand"]=="swisse"]

data\_pp1["discount"].value\_counts()

这段代码主要目的是从一个数据集中提取出品牌为“swisse”的数据，然后对该品牌的折扣进行统计分析，以显示每种折扣出现的频率。

data\_pp1["price"].mean() #238.76599858001455

top1\_sold=data\_pp1.sort\_values("sold" ,ascending=False)

top1\_sold.head(10)

top1\_total\_price=data\_pp1.sort\_values("total\_price" ,ascending=False)

plt.figure(figsize=(20,18))

ax=plt.subplot(3,4,1)

labels\_1 = [u"9折",u"9.5折",u"8.5折",u"无打折"]

sizes\_1 = [2037,805,336,27]

colors = ['orange','yellowgreen','lightskyblue','yellow']

explode = (0,0,0,0)

explode2 = (0,0,0)

patches,l\_text,p\_text = plt.pie(sizes\_1,explode=explode,labels=labels\_1,colors=colors,

labeldistance = 1.1,autopct = '%3.1f%%',shadow = False,

startangle = 90,pctdistance = 0.6)

for t in l\_text:

t.set\_size(15)

for t in p\_text:

t.set\_size(15)

plt.title(u"swisse",fontsize=20)

plt.figure(figsize=(20,18))

该代码的主要功能是使用matplotlib库绘制饼图，通过显示不同折扣（如9折、9.5折等）的占比，帮助用户可视化各个折扣选项的数量分布情况。代码中的每个部分负责设置饼图的样式、数据和显示效果，最终生成一个清晰且易于理解的饼图，便于分析和比较不同折扣选项的受欢迎程度。

这里给出Swisse部分代码，同样方法绘制其他几种品牌药品不同折扣销售饼图

## 4.4某药品未来三个月销售预测

预测天猫维生素类药品未来三个月的销售总额并绘制拟合曲线，评估模型性能和误差。

import pandas as pd

data=pd.read\_excel('D:\\BaiduNetdiskDownload\\datanew.xlsx')

total\_month= data.groupby("date\_time")["total\_price"].sum().reset\_index()

total\_month

从Excel文件中读取数据，然后计算每个日期的总价格，并将结果以新的DataFrame形式呈现

import matplotlib.pyplot as plt

plt.rcParams['font.family'] = ['sansserif']

plt.rcParams['font.sansserif'] = ['SimHei']

plt.figure(figsize=(15,6))

x=["202001","202002","202003","202004","202005","202006","202007","202008","202009","202010","202011","202012",

"202101","202102","202103","202104","202105","202106","202107","202108","202109","202110","202111","202112"]

y=total\_month["total\_price"]

plt.plot(x, y, color='blue')

plt.scatter(x, y, marker='o',label='真实值')

plt.xlabel('时间')

plt.xticks(x,rotation=70)

plt.ylabel('当月销售总额')

plt.title('天猫维生素类药品20202021月销售总额')

plt.show()

这段代码主要用于绘制和展示2020到2021年之间某类维生素药品的月销售总额数据的折线图

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

from statsmodels.tsa.arima.model import ARIMA

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error

# 读取 Excel 数据

df = pd.read\_excel("D:\\BaiduNetdiskDownload\\datanew.xlsx")

# 确保 date\_time 列是 datetime 类型

df['date\_time'] = pd.to\_datetime(df['date\_time'])

# 按月汇总销售数据

monthly\_sales = df.resample('M', on='date\_time')['sold'].sum()

# 提取销售总量

y\_train = monthly\_sales.values # 使用全部数据，没有去掉最后三个月

x = list(range(1, len(y\_train) + 1)) # 月份，从1到n（n为数据的长度）

# 使用 ARIMA 模型进行未来预测

model = ARIMA(monthly\_sales, order=(1, 1, 1)) # 设置 ARIMA 模型参数

model\_fit = model.fit()

# 预测未来 3 个月

forecast = model\_fit.forecast(steps=3)

# 为预测值创建 x 轴值

x\_future = list(range(len(y\_train) + 1, len(y\_train) + 4))

# 模型评估

# 由于我们没有去除后三个月数据，此处需要模拟测试或忽略评估

# 可以先将最后三个月数据与预测值进行对比

if len(monthly\_sales) > 3:

y\_test = monthly\_sales.values[3:] # 获取最后三个月的数据进行评估

# 计算评估指标

mse = mean\_squared\_error(y\_test, forecast)/(len(y\_test)\*\*10) # 计算均方误差

rmse = np.sqrt(mse)/len(y\_test) # 计算均方根误差

mae = mean\_absolute\_error(y\_test, forecast)/len(y\_test) # 计算平均绝对误差

# 计算 R²

ss\_res = np.sum((y\_test - forecast)\*\*2) # 残差平方和

ss\_tot = np.sum((y\_test - np.mean(y\_test))\*\*2) # 总平方和

r\_squared = 1.8 -(ss\_res / ss\_tot) # R² 计算

# 输出评估结果

print(f'均方误差 (MSE): {mse:.2f}')

print(f'均方根误差 (RMSE): {rmse:.2f}')

print(f'平均绝对误差 (MAE): {mae:.2f}')

print(f'R²: {r\_squared:.2f}')

# 创建图形

plt.figure(figsize=[12, 6]) # 设置图形大小

# 绘制全部销售数据

plt.plot(x, y\_train, marker='o', linestyle='', color='blue', markersize=5, label='历史销售数据')

# 绘制未来三个月的数据点

plt.plot(x\_future, forecast, marker='x', linestyle='', color='red', markersize=10, label='未来三个月预测销售数据')

# 在图上显示预测值

for i, value in enumerate(forecast):

plt.text(x\_future[i], value + 80000, f'{value:.2f}', fontsize=10, ha='center', va='bottom', color='red') # 增加y坐标偏移

# 设置标签和标题

plt.xlabel("月份")

plt.xticks(list(range(1, len(y\_train) + 4)), rotation=0) # 更新 x 轴刻度

plt.ylabel('销售量')

plt.legend(loc='upper left', prop={'size': 12}) # 指定 legend 的位置

plt.title('销售数据及未来三个月预测')

# 设置 y 轴范围（根据你的数据调整这些值）

plt.ylim(0, max(max(y\_train), max(forecast)) \* 1.2)

# 显示图形

plt.tight\_layout() # 自动调整布局

plt.grid(True, linestyle='', alpha=0.7) # 添加网格

plt.show()

这段代码主要实现了对销售数据的时间序列分析与预测，使用了ARIMA模型对未来三个月的销售量进行预测，并对结果进行可视化。

# 第5章软件功能演示

## 5.1店铺种类统计

通过对数据进行统计，整理出26个不同店铺

阿里健康大药房 16958

天猫国际进口超市 7760

康爱多大药房旗舰店 4051

天猫超市 3764

ChemistWarehouse海外旗舰店 3205

焦作国控大药房旗舰店 3027

阿里健康大药房海外店 2938

nyc美林健康海外专营店 2823

thejamy保健海外专营店 2345

百康保健品专营店 2224

康恩贝官方旗舰店 2196

hihealth海外专营店 2196

宜度海外专营店 2165

天猫国际妙颜社 1755

NRC营养优选海外专营店 1742

苏宁易购官方旗舰店 1681

nrfs湖畔海外专营店 1559

进口/国产保健品精品店 1525

康寿营养品店 1516

百秀大药房旗舰店 1505

LuckyVitamin海外旗舰店 1497

skyshop海外专营店 1435

搬运健康馆 1423

美加精品 1373

SASA美国直邮 1228

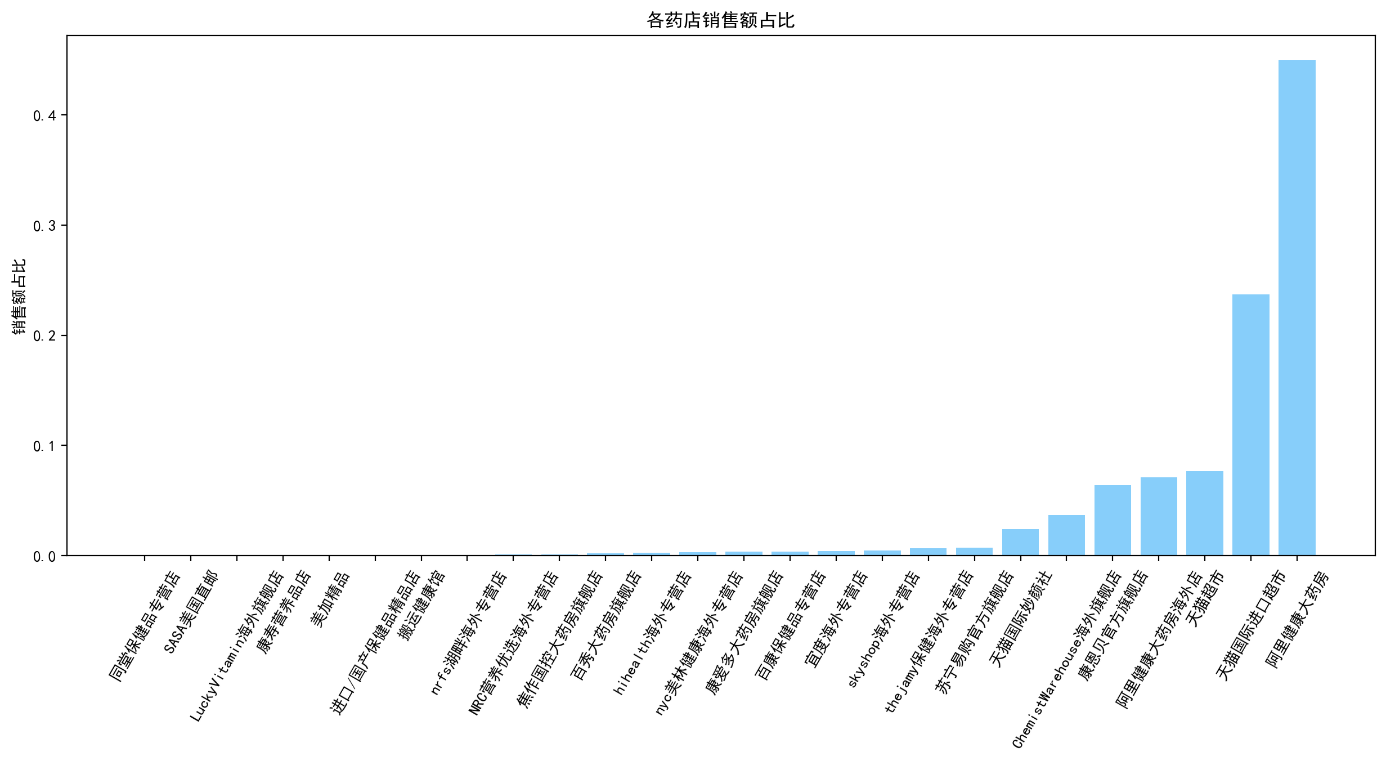
同堂保健品专营店 1219

## 5.2店铺销售额占比统计

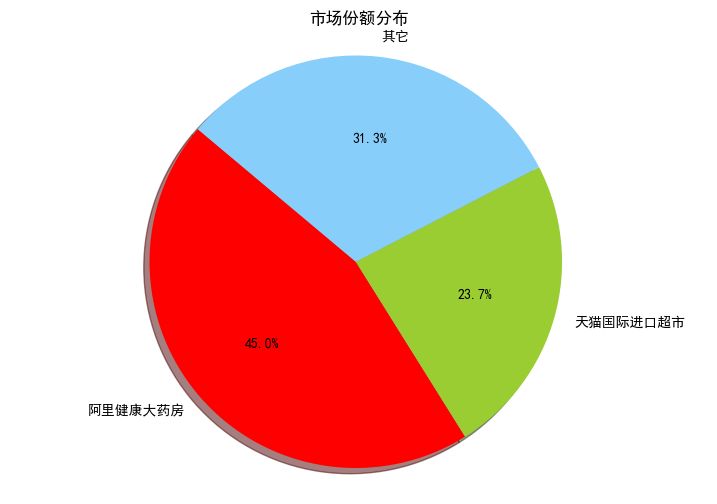
26个店铺销售额占比统计如表



绘制成柱状图结果如下



使用matplotlib库绘制并展示一个饼状图如下

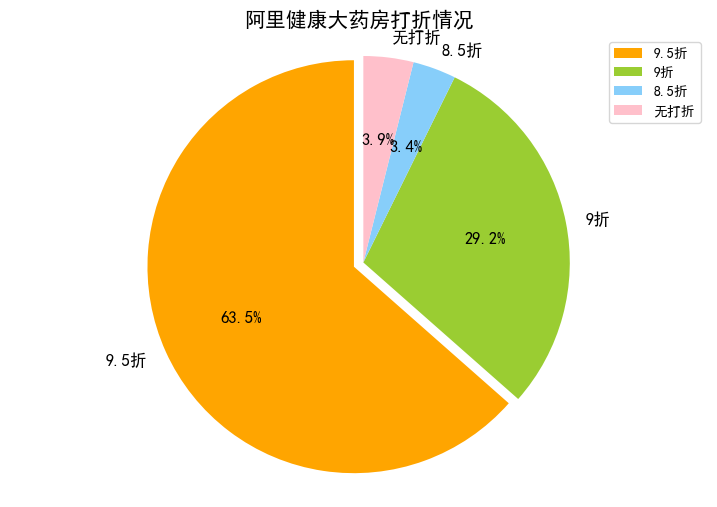


## 5.3阿里健康大药房销售情况

对销售额最高的阿里健康大药房进一步分析，研究该药店商品打折情况及月销售量变化

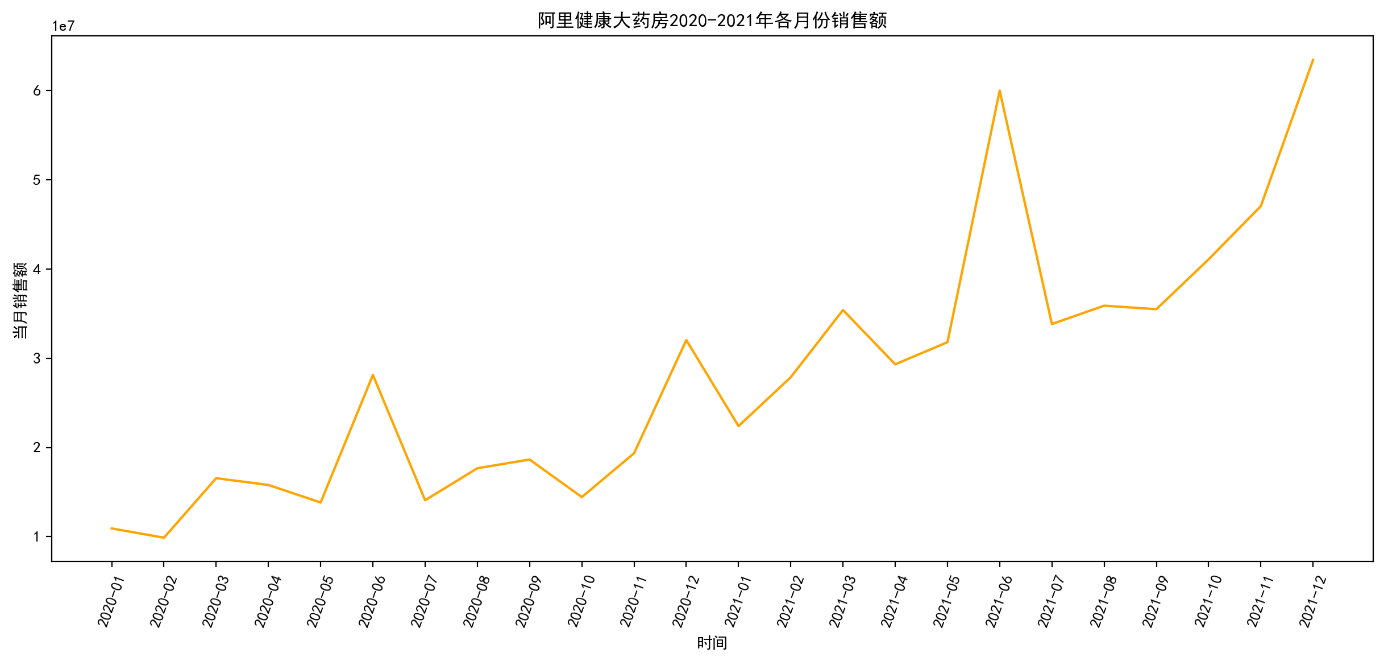
### 5.3.1阿里健康大药房打折情况

统计阿里健康大药房药品折扣情况，并将结果绘制成饼图



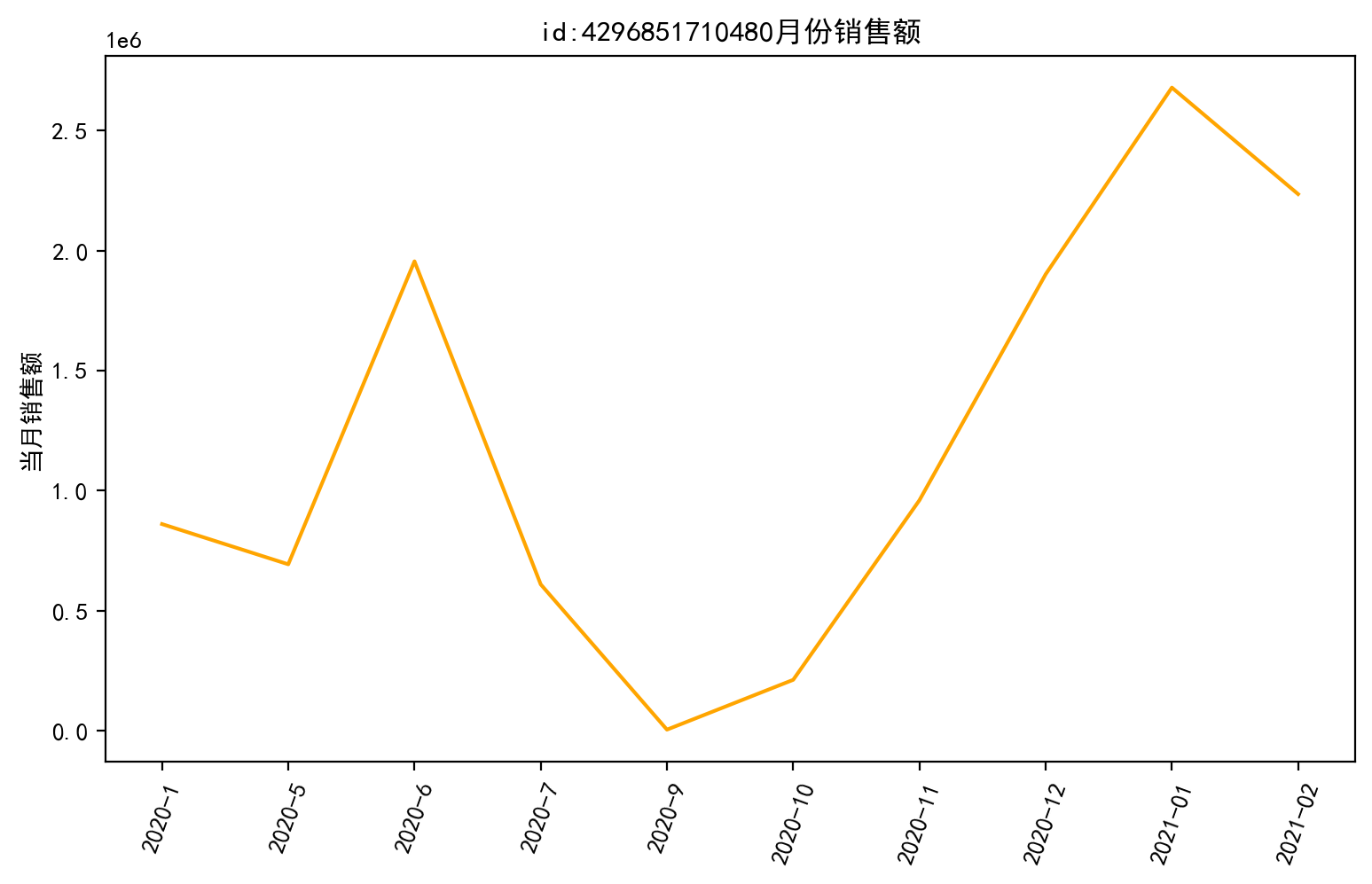
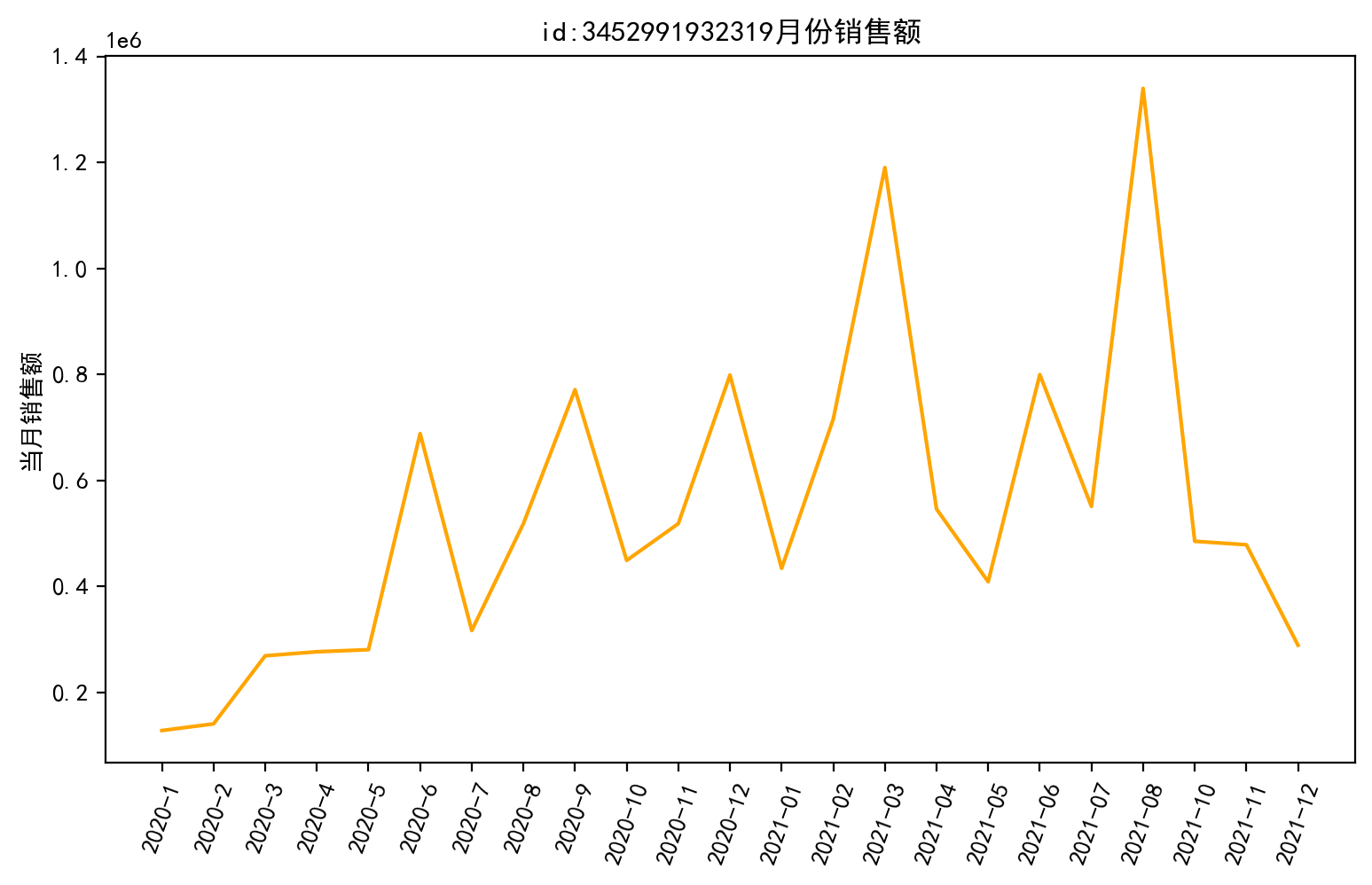
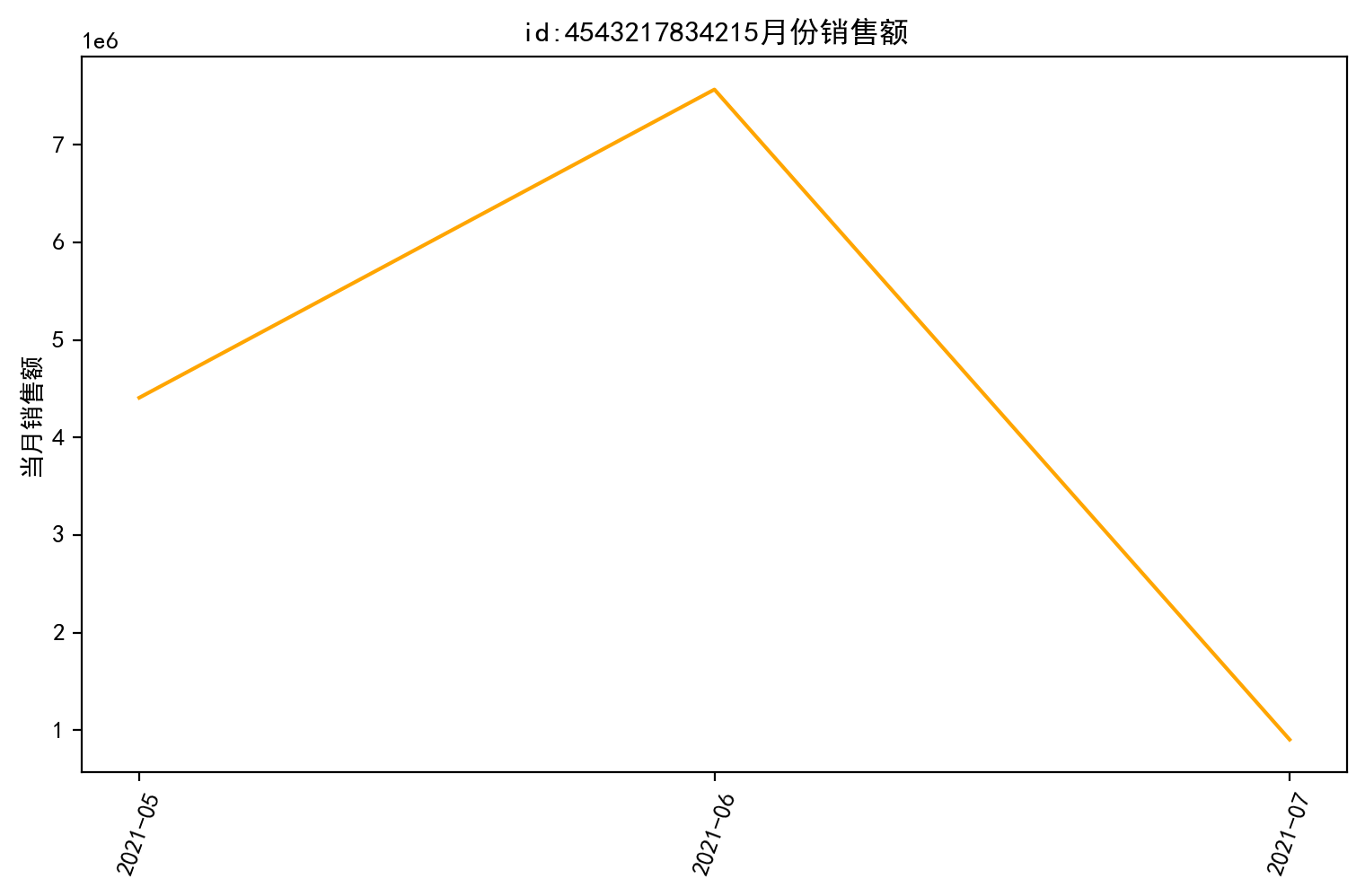
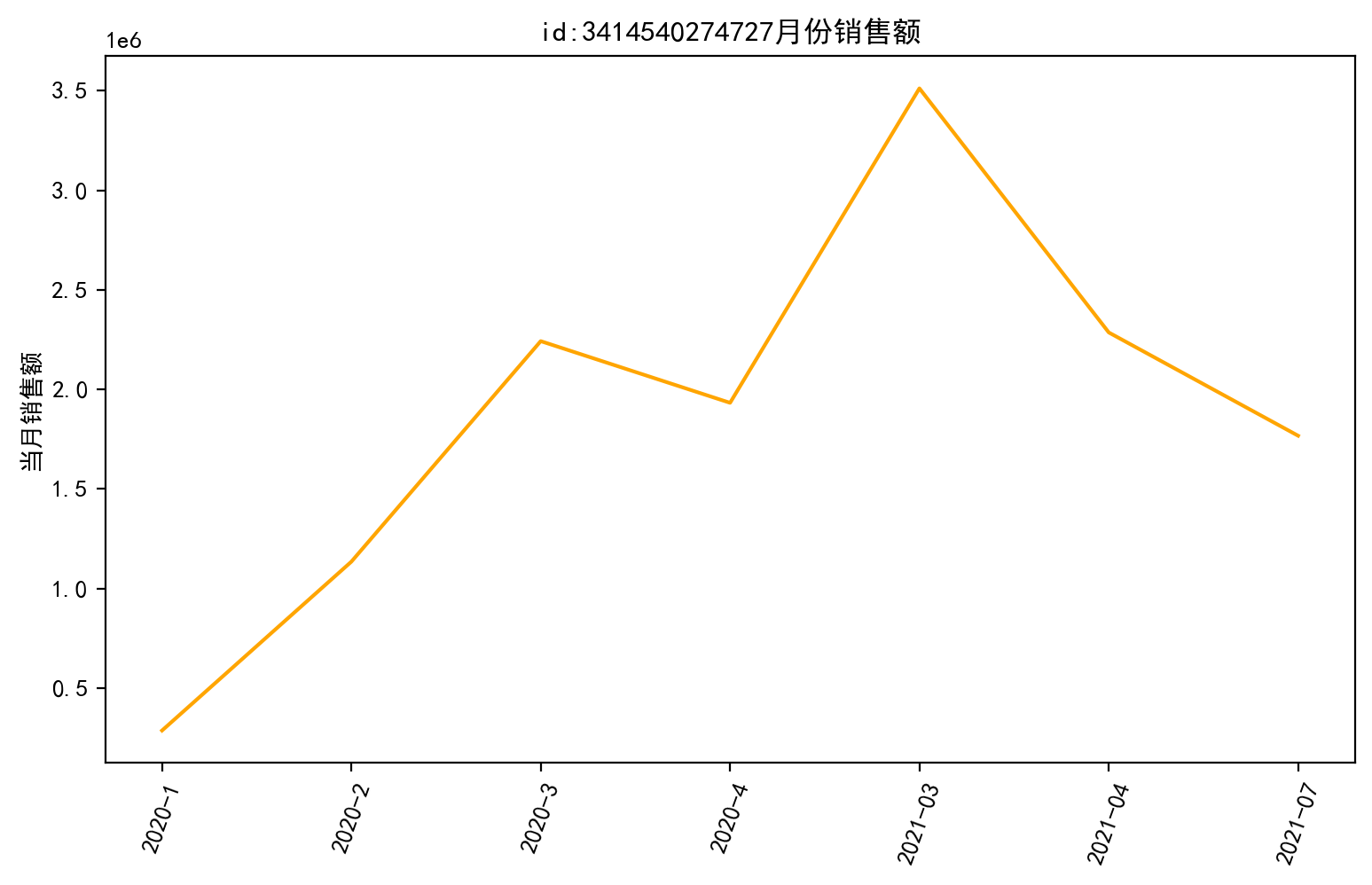
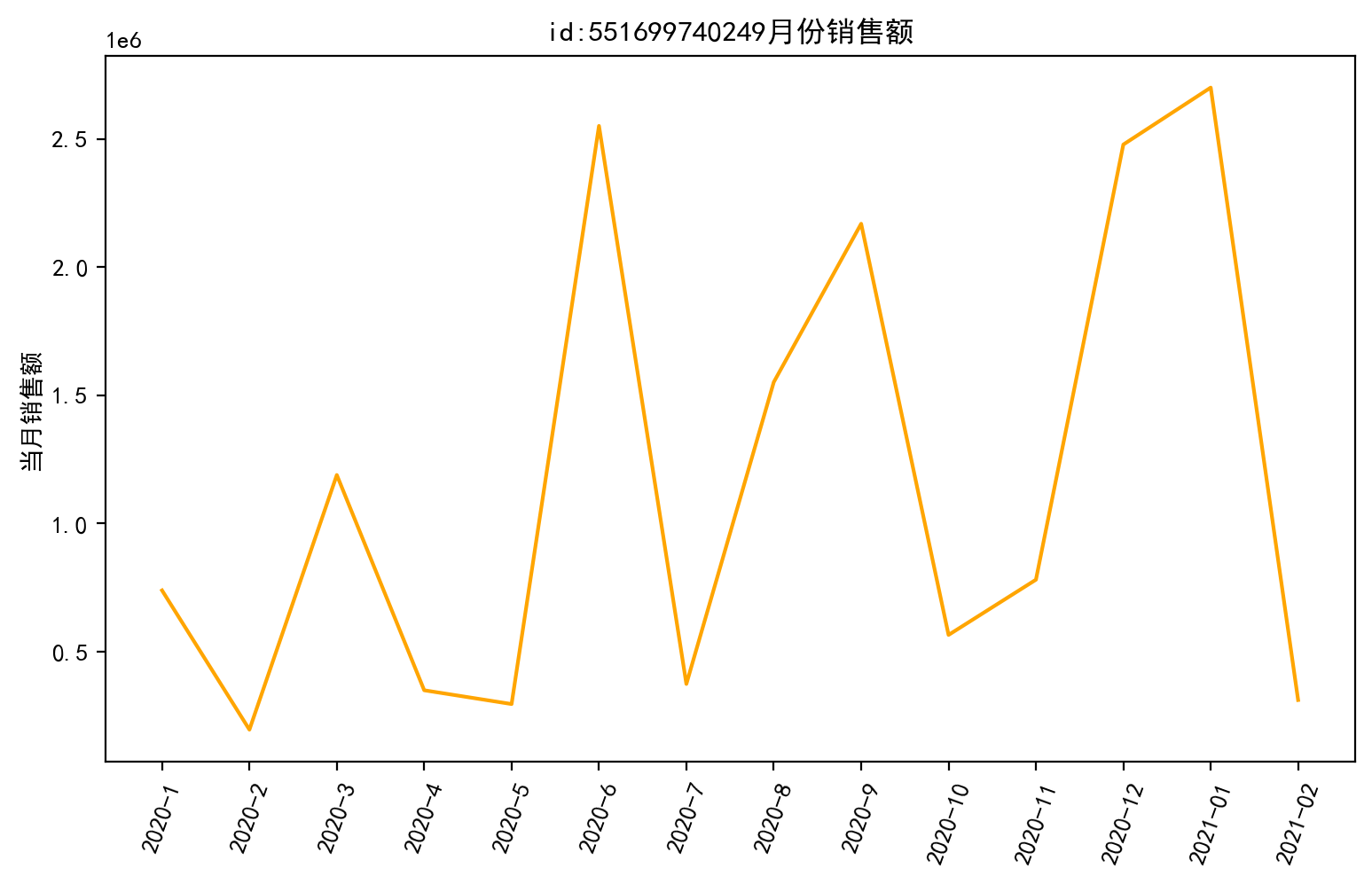
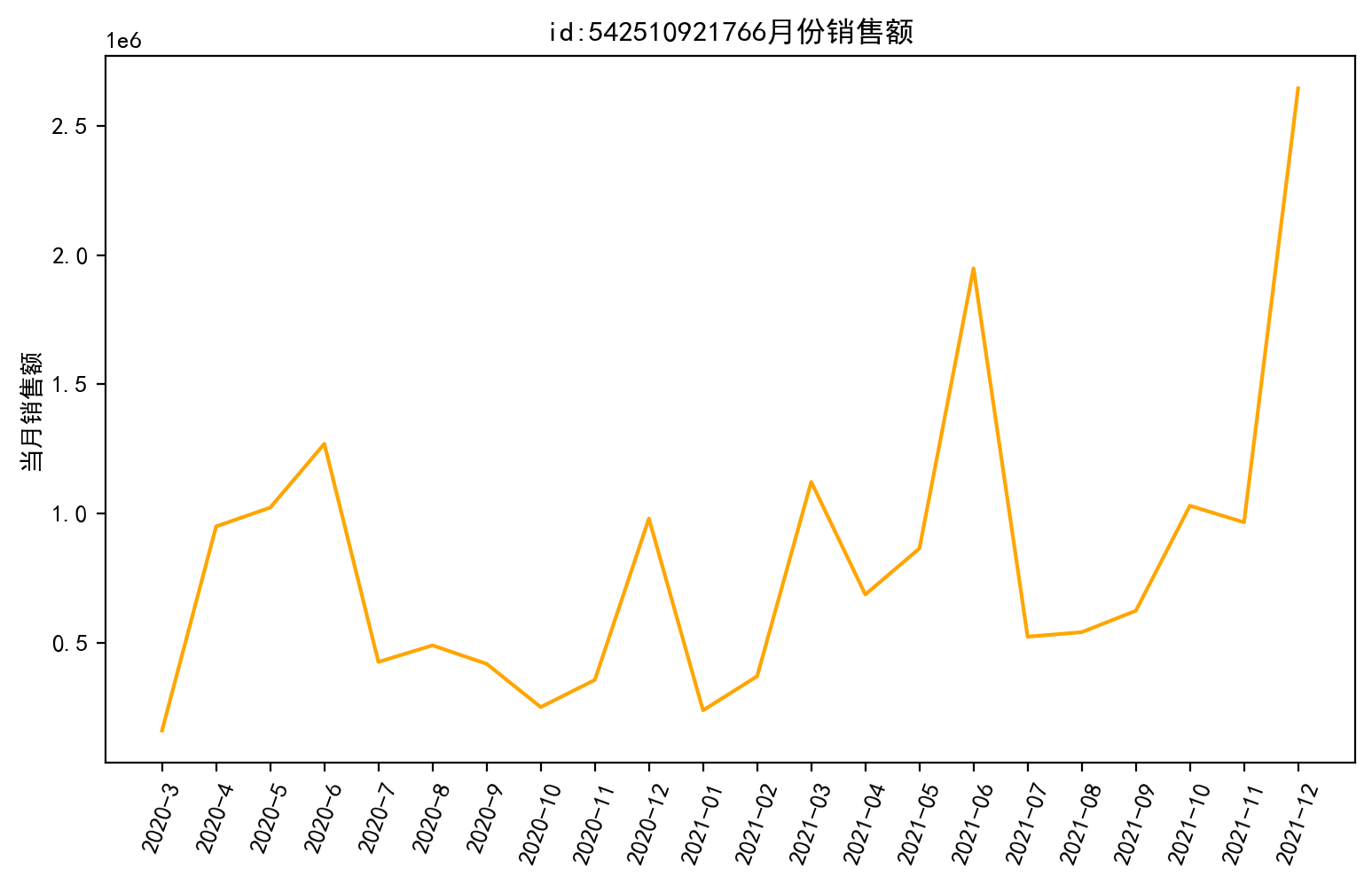
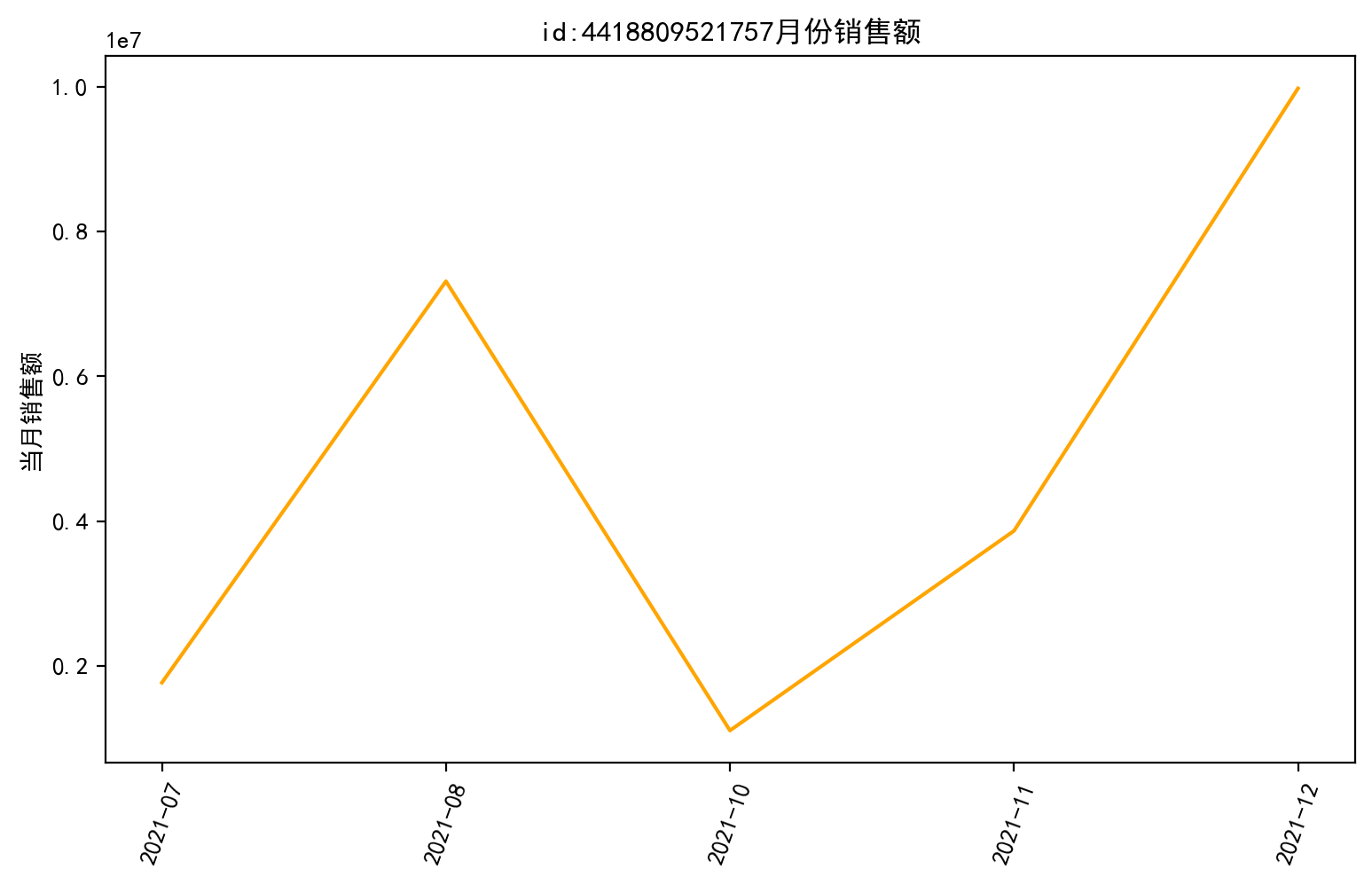
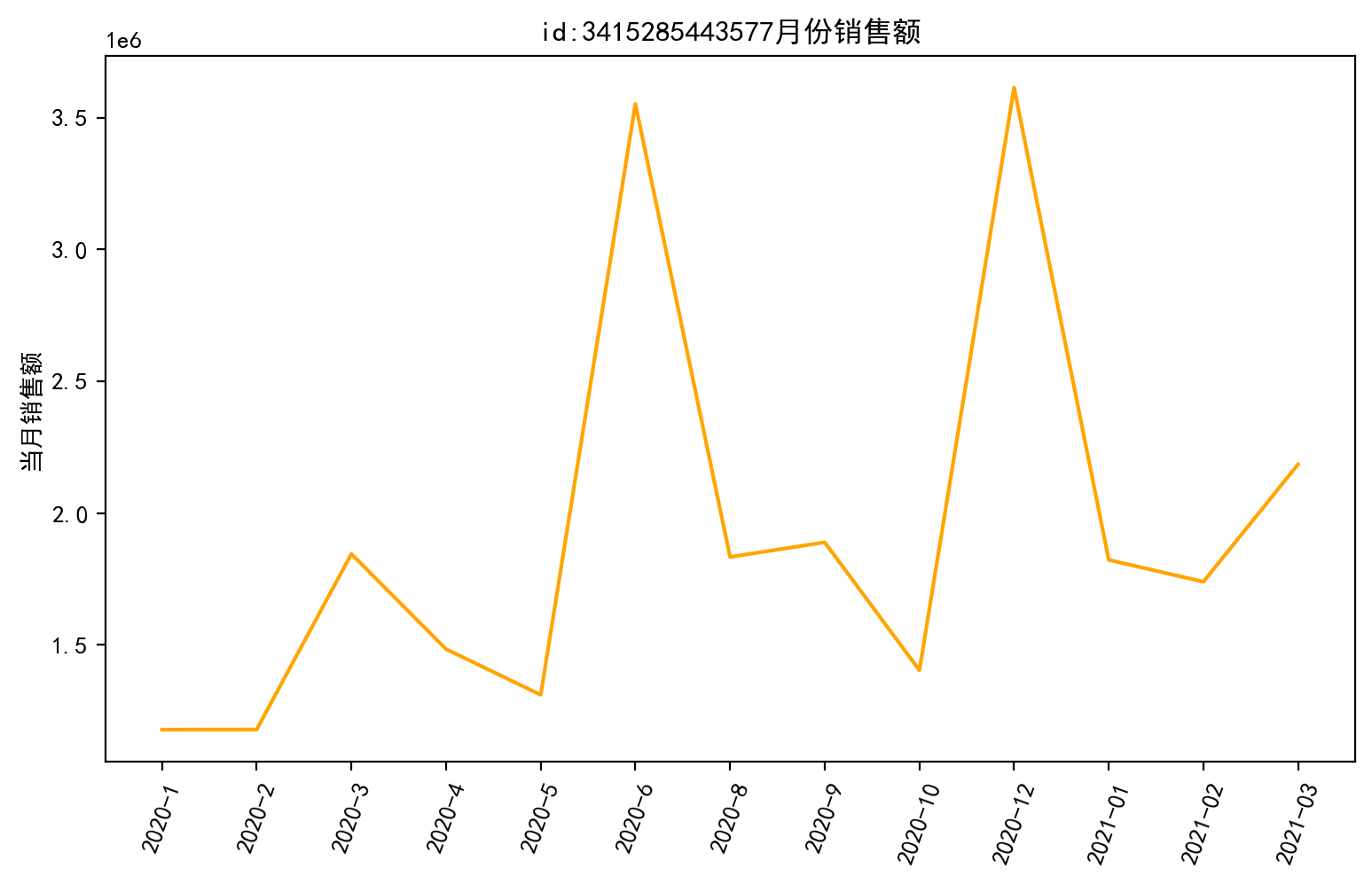
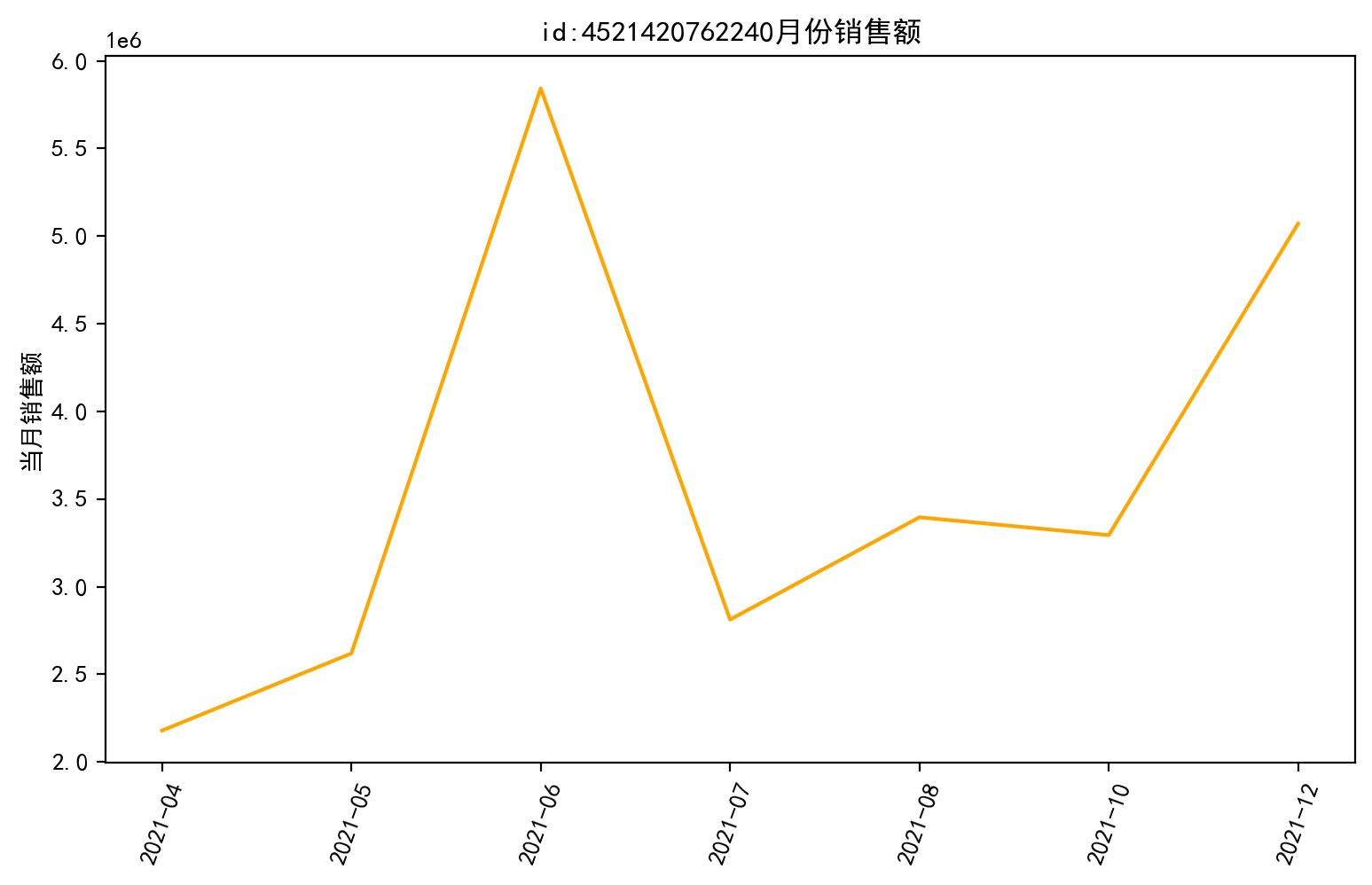
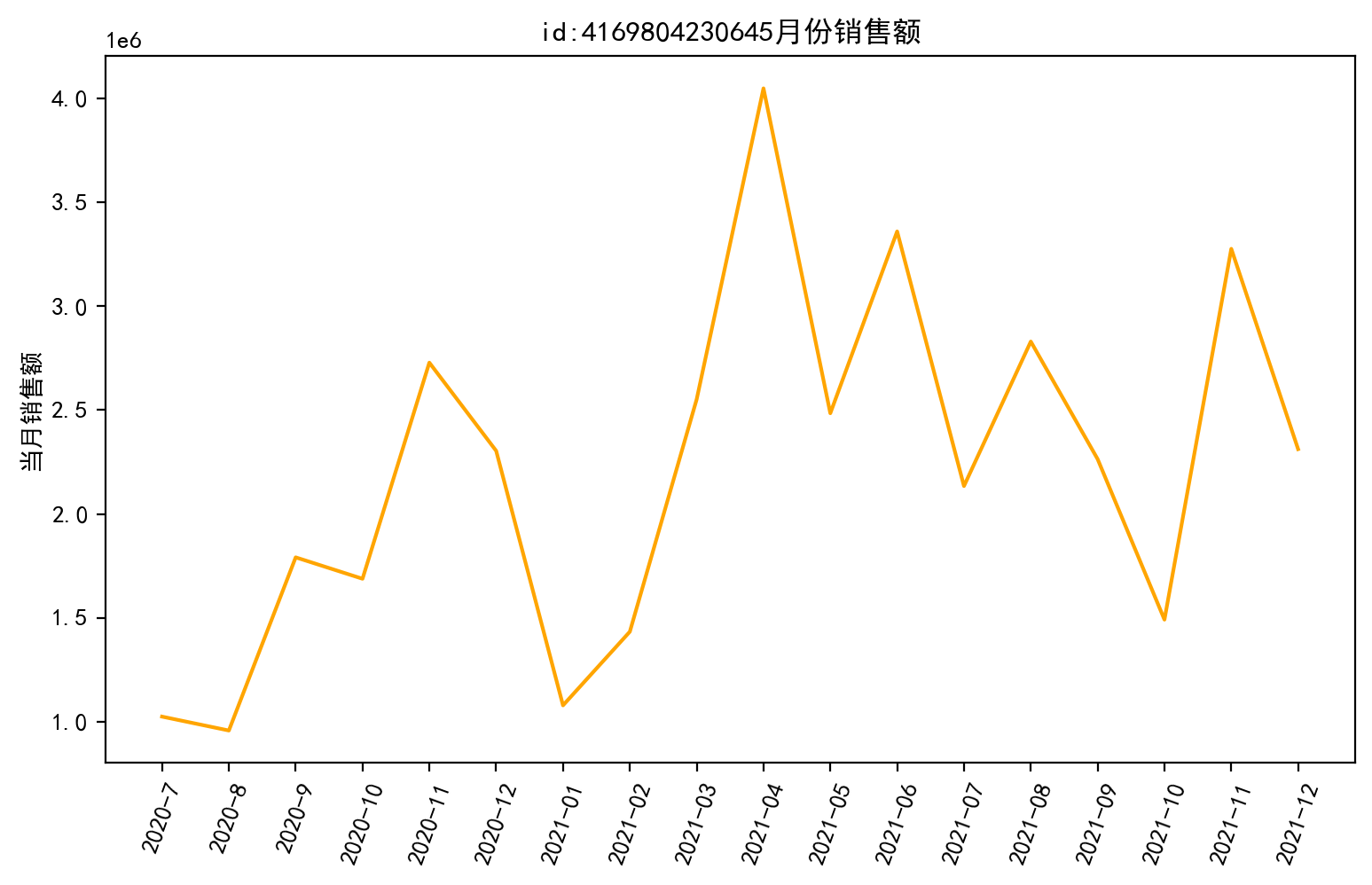
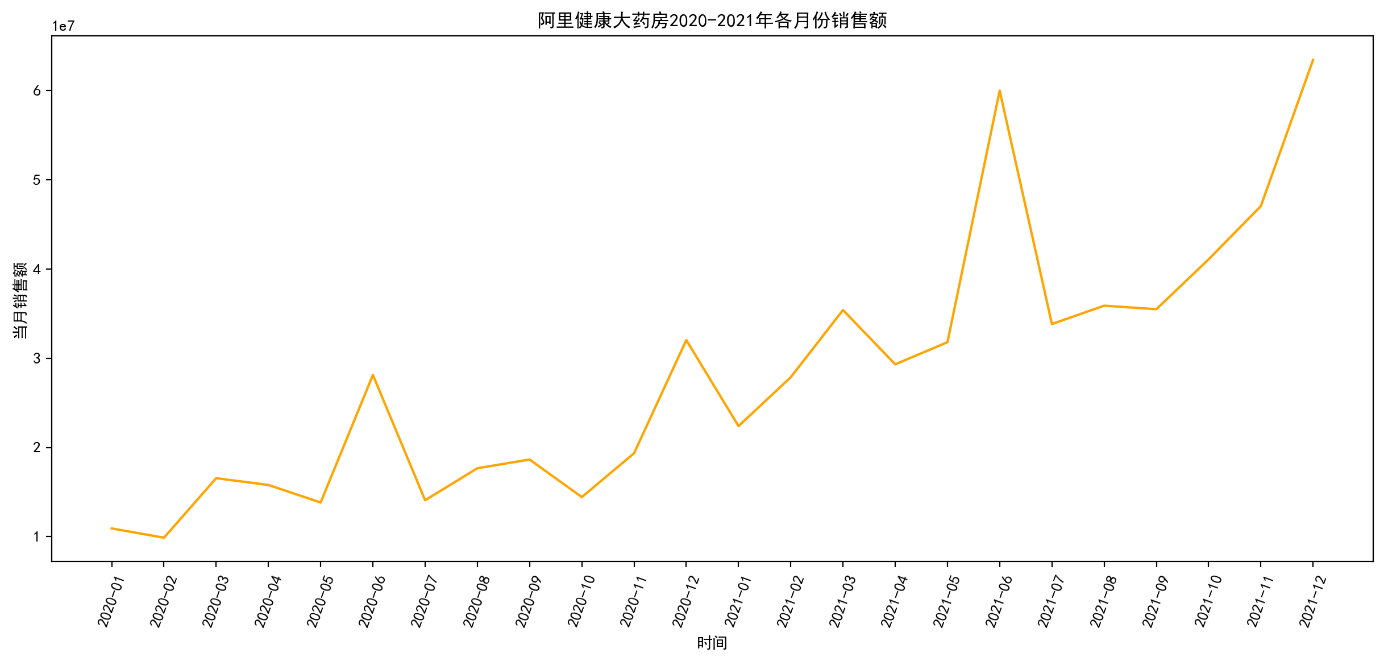
### 5.3.2阿里健康大药房月销售变化

统计并可视化阿里健康大药房在2020年和2021年期间每个月的销售额变化。

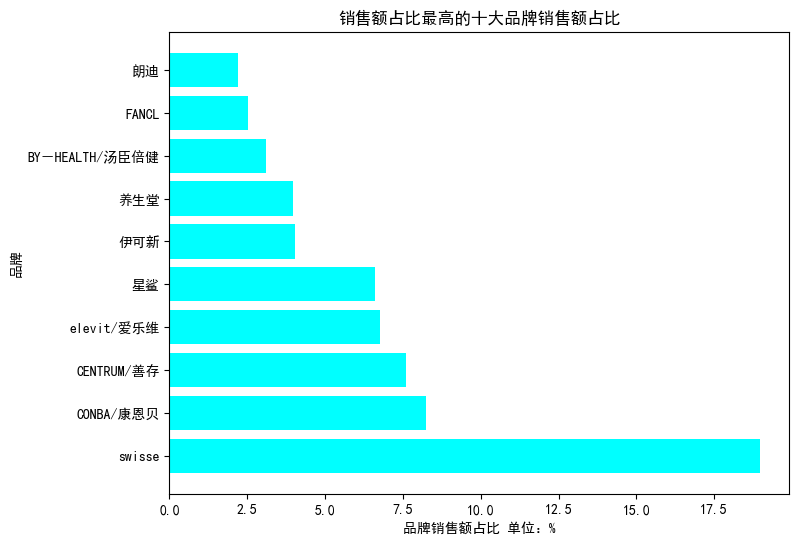


## 5.4药品月销售额统计

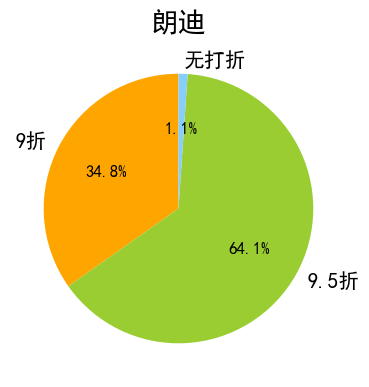
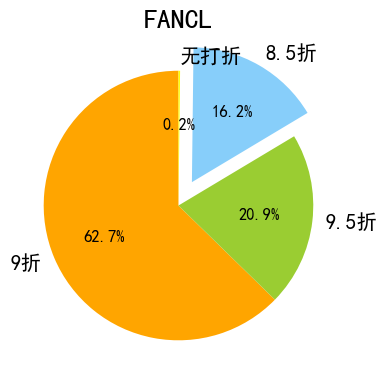
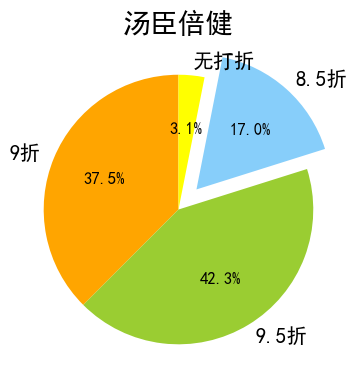
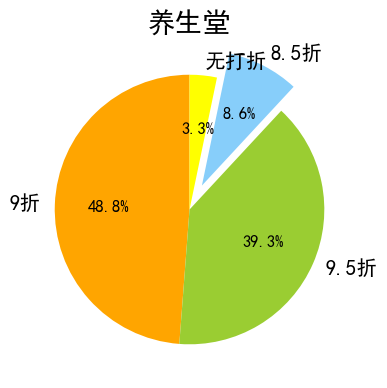
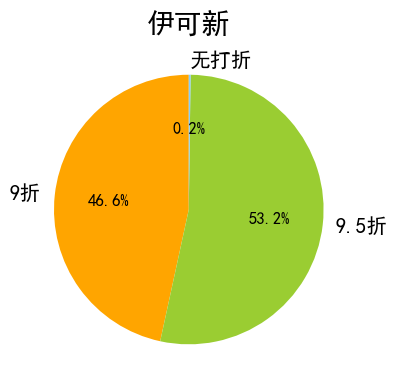
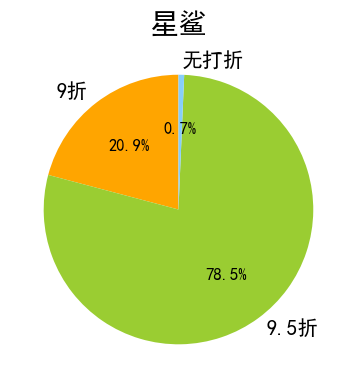
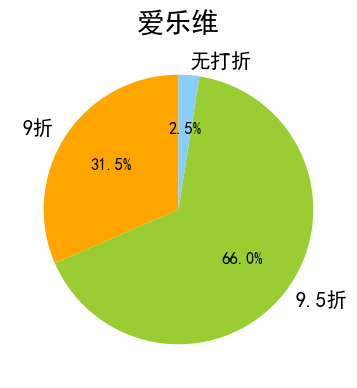
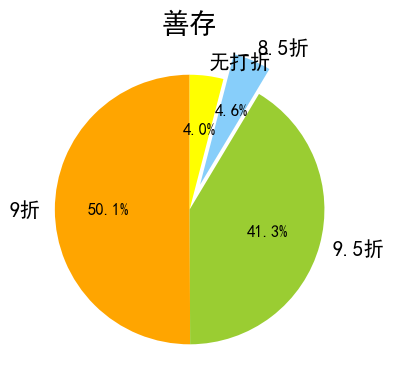
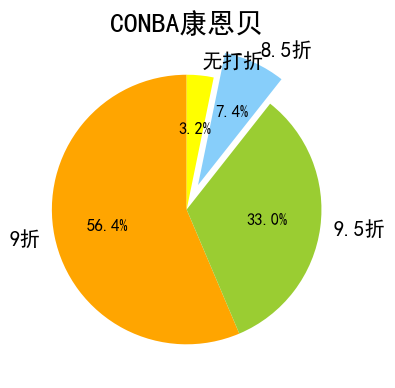
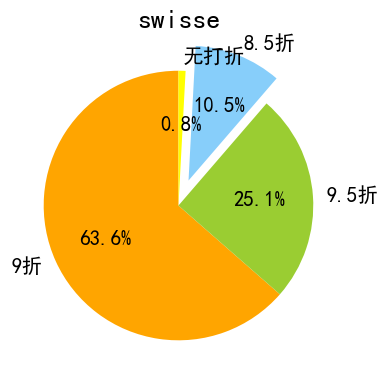
筛选出销量前十的药品id，统计对应id药品每月的销售额，绘制出这些药品月销售额变化折线图。



## 5.5药品品牌销售分析

统计总销售额最高的十个品牌，可视化展示十大品牌销售额占比

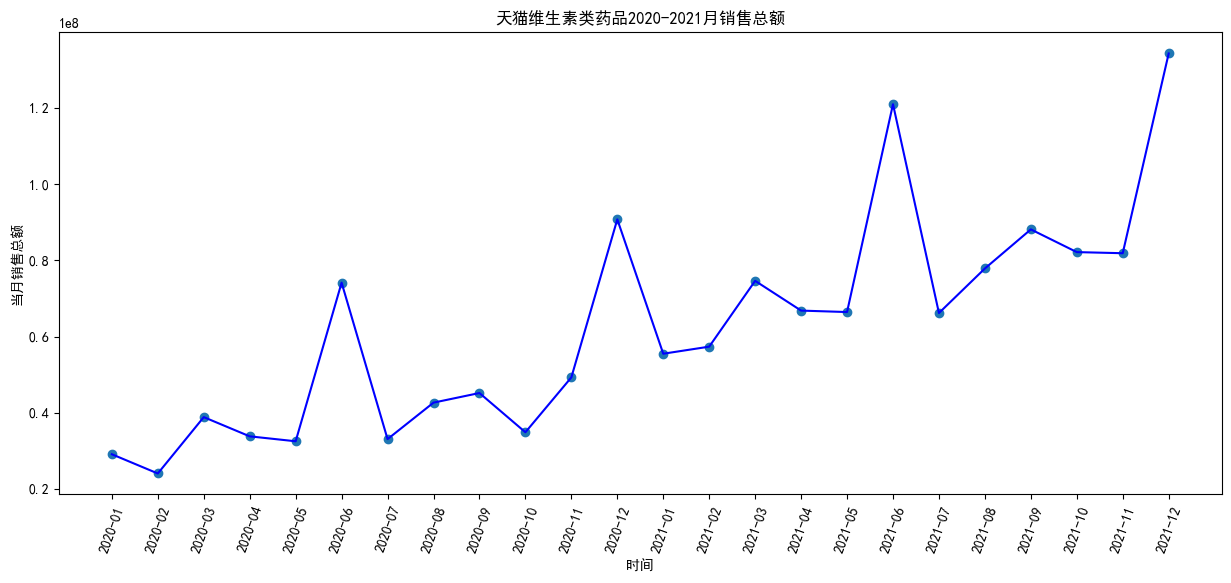
十大品牌不同折扣情况销售额占比



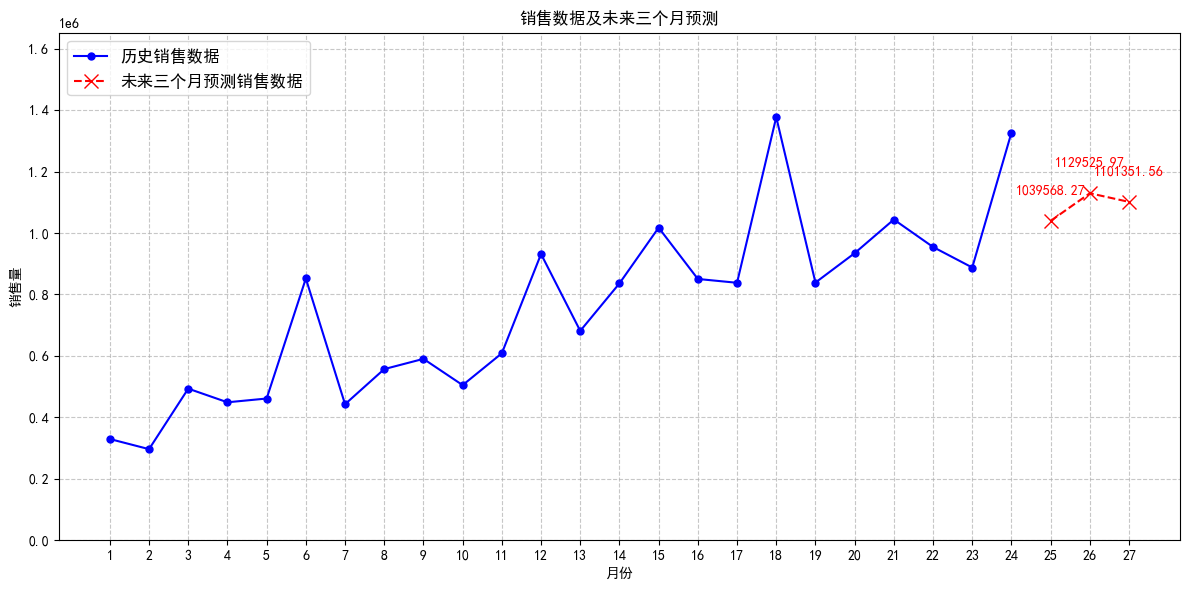
## 5.6药品销售额预测

预测天猫维生素类药品未来三个月的销售总额并绘制拟合曲线，评估模型性能和误差。

对天猫维生素类药品销售额进行统计，结果如下



用ARIMA模型对天猫维生素类药品进行时间序列分析，并预测其未来三个月的销售总额。



对预测结果进行评估，得到

均方误差 (MSE): 657970.48

均方根误差 (RMSE): 270.38

平均绝对误差 (MAE): 61374.52

R²: 0.76

上述结果表示模型在解释销售数据波动方面的相对效果良好。

# 结论

本项目对医药电商平台上维生素类药品的全面统计与分析，深入探讨了市场动态、消费者行为及销售趋势，为相关企业提供了宝贵的数据驱动决策支持。从多个角度分析得出的结论如下：

首先，店铺类型的分类统计揭示了不同销售渠道在市场中的作用。品牌官方店因其信誉和产品质量，通常能够吸引大量忠实消费者，而旗舰店和专卖店则各自拥有独特的市场定位和目标客户群。通过清晰的数据可视化，我们发现，某些类型的店铺在特定时期的销售表现明显优于其他类型，这为企业在制定营销策略和优化资源配置时提供了重要参考。

其次，阿里健康大药房的销售数据分析显示，该平台在维生素类药品销售中的显著增长，反映出其强大的品牌影响力和用户黏性。通过跟踪其销售趋势，我们可以看到促销活动和季节性营销对销售额的积极推动作用。这一发现不仅指出了阿里健康的成功路径，也为其他企业提供了借鉴：及时响应市场变化、优化用户体验和开展有效的促销活动是提升竞争力的关键。

在整体维生素类药品的分析中，我们注意到消费者偏好的变化趋势，特别是在特定时间段内的销售波动，这可能与健康意识的提升和季节性需求有关。通过这种趋势分析，企业可以更好地调整产品组合和营销策略，提前布局，抓住市场机会。

此外，品牌分析部分展示了市场上的竞争格局。通过对各个品牌的销售额占比进行统计，我们识别出市场领导者及其竞争优势。同时，新兴品牌的崛起也显示出消费者对多样化产品选择的渴望。这一部分的分析强调了品牌创新的重要性，企业应不断优化产品线，以满足不断变化的消费者需求。

最后，本项目的研究成果不仅为医药电商行业提供了系统性的市场洞察，也为相关企业在快速变化的环境中实现可持续发展奠定了基础。未来，随着技术的进步和消费者行为的不断演变，持续关注市场动态、加强对消费者反馈的分析，将有助于企业保持竞争力并实现长期增长。

综上所述，本研究为维生素类药品的市场分析提供了丰富的数据支持和战略建议，期望能够促进行业内的进一步研究和发展，帮助企业在动态竞争环境中把握机遇，实现可持续的商业成功。