基于高斯过程回归对超市收入的预测与优化

摘要

在零售行业数字化转型与消费者需求动态化的双重驱动下，连锁超市的经营策略优化与收入预测正成为提升核心竞争力的关键课题。本文以国外某连锁超市三大店业务为研究对象，基于 2025 年 1 月至 4 月的 1000 条交易数据，从数据特征解析、预测建模与策略优化三维度出发，构建融合 Apriori 关联规则挖掘、时间序列分析与多目标决策的综合模型体系，实现了消费规律量化识别、毛收入动态预测及库存促销策略优化的全流程管理，为超市精细化运营与科学决策提供了数据驱动的解决方案。

对于问题一, 通过建立 **TOPSIS 模型**，对会员与非会员对超时收入贡献度进行比较，利用相关得分进行综合量化。在挖掘高频购买组合时，通过 **Apriori 算法**，提炼出置信度最高的组合。综合分析各个指标可以得到，C 店所在店的消费水平相对较高，并且三店消费时段集中在 **10-11 点，13-15 点，18-19 点**，而且 **A 店使用电子钱包相对较多，而 B 店使用信用卡相对较多，而 C 店相对使用现金较多**。在 TOPSIS 模型量化下，会员群体对商店的贡献度比非会员高出 5 个百分点。而高频购买组合是**运动旅游 + 信用卡 + 会员**。在大部分情况下，商品存在周末、节假日销量激增的现象。

对于问题二, 先通过 **KS 正态检验**对所有数据检验，得到数据并不存在正态性分布，由此在分析指标间关系时，通过斯皮尔曼相关性分析可知，毛收入和商品单价、购买数量有着很强的正向相关性；毛利率和店、产品类别、购买数量有着很强的正向相关性。在建立预测模型时，通过对比不同模型之间均方误差等参数的差异，最终选择**高斯过程回归机器学习模型**。最终结果显示，最终选 **GPR 模型预测**，其误差低、适应性强，能为超市运营提供决策支撑。

对于问题三, 本部分针对超市运营优化问题，基于销量对比分析、店铺评分评估及消费特征挖掘，提出两类战略建议。分析表明，食物类与运动旅行类商品需求突出，C 店经营表现显著优于 B 店（TOPSIS 模型评分验证），且消费者会员身份与其购买力呈现强相关性，为策略制定提供数据支撑。在具体策略上，建议重点提升高需求品类库存并优化 C 店管理，结合需求波动动态调整安全库存，通过品类结构优化与精细化工具强化供给；同时设计分层会员权益，以组合消费激励形成正向循环。从战略布局看，鉴于 B 店区域生态恶化与 C 店所在城市的消费潜力，建议关闭 B 店释放资源，并在该城市科学拓店，复制成熟模式实现协同优化。

关键字： Topsis 法　傅立叶变换　高斯过程回归　Apriori 算法　经营优化

一、 问题重述

1.1 问题背景

在全球零售行业数字化转型的进程中，连锁超市作为现代流通体系的关键节点，其运营模式正经历从经验驱动向数据驱动的深刻变革。随着零售交易数据采集技术的精细化发展，企业已积累海量包含商品属性、消费者行为轨迹、支付模式等多维信息的交易数据，如何从这些数据中提炼商业洞察并构建科学决策体系，成为提升零售企业核心竞争力的战略课题。

某国外连锁超市在三大核心店构建商业网络，依托信息化管理系统对每笔交易进行全维度数据记录，累计形成具有深度分析价值的数据集。当前企业管理层面临双重运营挑战：一方面，采购计划制定仍依赖传统经验判断，导致库存配置与区域消费需求的动态匹配度不足，造成供应链效率损耗；另一方面，营销策略的精准性缺乏量化支撑，难以针对不同客群的消费特征实现价值转化优化。这种数据资源与决策需求的结构性脱节，使得建立基于数据挖掘的收入预测体系成为企业突破经营瓶颈的必然选择。

由此可见，构建数据驱动的库存调度与智能排程方案，已成为布料企业突破运营瓶颈、提升综合竞争力的战略关键。

1.2 问题要求

通过建立数学模型，结合超市的各商品的数据统计，建立起相关科学的收入预测模型，以达到最小化成本，最大化利润的目标。

问题 1: 基于该连锁超市 2025 年 1 月 7 日至 4 月 5 日的 1000 条交易数据，需构建系统化的数据特征解析体系。首先，运用统计学方法，精确计算各店在销售额、商品单价、商品偏好等核心指标上的统计分布，通过交叉分析与时间序列建模，深入识别不同店在早、中、晚消费高峰时段及付款方式选择上的显著差异；其次，采用对比分析法与聚类算法，细致剖析超市会员群体的消费特征，从消费频次、客单价、购买品类等维度量化会员与非会员群体的差异，并科学测算会员群体对超市总收入的贡献度；再者，利用 Apriori 等关联规则算法，挖掘商品类别、付款方式、会员属性等多维度组合下的高频购买模式；最后，通过周期图分析与季节性分解，系统性探究六类商品销售数据的周期性特征，识别周末、节假日等特殊时段的销量波动规律，为后续经营决策筑牢数据根基。

问题 2: 针对超市毛收入与毛利率的预测需求，需搭建完整的数据分析与建模框架。第一步，运用描述性统计、方差分析等方法，系统探究毛收入与毛利率在不同店、消费者类别、产品类别、付款方式等维度的分布规律，借助热力图、箱线图等可视化手段直

观呈现分析结果，并通过相关性检验与显著性分析，精准识别对毛收入与毛利率具有显著影响的关键变量；第二步，综合运用时间序列模型、回归分析及机器学习算法构建预测模型，对 2025 年 4 月 6 日至 15 日超市日均毛收入与毛利率进行预测。通过均方误差（MSE）、平均绝对误差（MAE）等指标对比不同模型的预测精度，结合数据特征与业务场景，合理阐释模型选择依据；同时，结合该国节假日日历，评估特殊节假日对销售数据的影响，在模型中科学引入虚拟变量或季节性调整因子，确保预测结果的准确性与可靠性。

问题 3: 基于上述数据分析与模型结果，从战略决策与运营优化层面，为超市管理层提供具有实操性的量化经营策略优化建议。其一，通过需求预测与边际收益分析，精准确定需优先提升库存水平或加大促销力度的商品类别、店区域及客户群体；其二，建立客户满意度评分与经营指标的关联模型，提出针对性的服务流程优化与产品改进方案；其三，运用组合优化算法，设计提升毛利率的商品组合策略，并通过模拟分析验证策略有效性；其四，结合各店销售数据特征与需求预测结果，制定差异化的库存优化方案；其五，构建会员扩容对收入影响的弹性模型，预测会员占比提升 10% 时的收入增长幅度；其六，建立多目标决策模型，综合考量收入损失、客户流失率、区域竞争态势等因素，为超市关店决策提供科学的量化评估框架，助力企业实现资源的高效配置与竞争力的全面提升。

二、 问题分析

2.1 问题一分析

对于问题一，需从多维度解析数据特征与挖掘规律。首先，针对三大城市的销售数据，需计算销售额、商品单价、商品偏好等指标的统计分布，如通过分组计算各城市不同商品类别的销量占比、销售额均值及单价分位数，同时分析早（10-11 点）、中（13-15 点）、晚（18-19 点）高峰时段的交易分布差异，以及 A 店电子钱包、B 店信用卡、C 店现金的支付方式选择偏好。其次，分析会员消费特征时，利用 TOPSIS 模型量化会员与非会员的贡献度差异，发现会员贡献度较非会员高 5 个百分点，结合客单价、购买频次等指标深入对比两类群体的消费行为。再者，运用 Apriori 算法挖掘高频购买组合，提炼出“运动旅游 + 信用卡 + 会员”这一置信度最高的组合，需验证其支持度与提升度以确认商业价值。最后，分析六类商品的销售周期性，关注周末及节假日是否存在销量激增现象，例如通过时间序列分解识别周期性波动规律。整个分析过程需结合数据预处理与可视化手段，为后续预测与策略优化提供数据支撑。

2.2 问题二分析

在解决超市毛收入与毛利率预测问题时，需从数据特性分析与模型构建逻辑展开系统分析。首先，通过 KS 正态性检验发现数据不服从正态分布，这决定了传统基于正态假设的统计方法适用性受限。其次，采用非参数的斯皮尔曼相关性分析，探究指标间关系。再者，在构建预测模型时，对比时间序列、传统回归与机器学习模型的均方误差、平均绝对误差等指标，以选择合适模型。此外，模型构建需考虑预测期内是否包含节假日，若存在则需引入相应变量调整预测结果，确保模型鲁棒性，最终实现对未来十天日均毛收入与毛利率的精准预测，为超市相关策略提供量化支撑。

2.3 问题三分析

对于问题三，在解决超市库存优化与店铺布局问题时，需从经营数据特征与策略逻辑展开系统分析。首先，通过销售数据的趋势分解发现食物类与运动旅行类商品存在显著季节性波动，这决定了传统固定库存策略的局限性。其次，采用 TOPSIS 综合评价法探究店铺运营效率，量化 C 店与 B 店的综合表现差异。再者，在制定库存与拓店策略时，对比 ABC 分类法、层次分析法与动态规划模型的资源配置效率，以确定最优方案。此外，策略设计需考虑区域消费特征，若 C 店所在城市存在消费升级趋势则需引入差异化品类布局，确保策略针对性。最终实现库存周转率提升与店铺网络优化的协同，为超市战略决策提供量化支撑。

三、 模型假设

为了构建更加精准的数学模型，本文根据实际情况以及任务所给要求做出以下合理的假设或条件约束：

- 假设超市交易数据记录完整，无系统性缺失或录入错误，各类指标的统计口径一致，可直接用于模型分析。
- 假设预测期内消费者的购买偏好、会员活跃度及支付习惯与历史数据表现一致，不出现突发性消费趋势转变。
- 假设 C 店所在城市的消费升级趋势、B 店所在区域的商业生态在短期内无显著变化，拓店选址的竞争格局与成本结构保持稳定。
- 假设库存优化策略中引入的安全库存模型、ABC 分类法等工具可有效执行，供应链响应速度满足动态补货需求，且商品损耗率控制在行业常规水平。
- 假设分层会员权益、动态折扣等政策可被消费者清晰认知，且优惠成本与激励效果呈线性正相关，不存在因政策复杂导致的参与度低下问题。
- 假设 TOPSIS 模型、高斯过程回归等方法的评价维度与预测逻辑符合超市实际经营场景，模型参数的优化结果具备商业解释性。

四、 符号说明

符号	说明	单位
μ	样本均值	个
σ	样本标准差	无
D	经验分布与理论正态分布的最大垂直距离	无
ϵ	独立同分布的噪声	无
ϕ_i	自回归系数	无

五、 问题一的模型的建立和求解

5.1 TOPSIS 模型的建立

TOPSIS 模型是一种常用的多属性决策性模型。其核心思路是基于归一化决策矩阵，通过计算各个评价对象和最优方案、最劣方案的相对接近程度，从而评价对象排序。[1]

Step1 正向化处理表格数据

本题中，由于数据均为正向化数据，因此小组跳过了本步骤。

Step2 标准化处理表格数据

标准化处理表格数据最大目的是在于其能够消除不同指标的量纲带来的影响。在本题中，有 501 个评价对象即会员总数，2 个评价指标（即毛收入和毛利率），以此构建出标准化矩阵 X 如下：

$$X_1 = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots \\ x_{(501)1} & x_{(501)2} & x_{(501)3} \end{bmatrix}, \tag{1}$$

然后，对标准化后的矩阵 Z 中的每一个元素进行如下操作：

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}}, i = 1, 2, 3 \quad j = 1, 2, ..., 4000 \tag{2}$$

Step3 计算得分并归一化处理

这些指标的标准化矩阵 Z 表示为：

$$Z_1 = \begin{bmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ \vdots & \vdots & \vdots \\ z_{(501)1} & z_{(501)2} & z_{(501)3} \end{bmatrix}, \tag{3}$$

定义最大值:

$$Z^{+}=\begin{pmatrix}\max\left\{z_{11},z_{21},\ldots,z_{(501)1}\right\}\\ \max\left\{z_{12},z_{22},\ldots,z_{(501)2}\right\},\max\left\{z_{13},z_{23},\ldots,z_{(501)3}\right\}\end{pmatrix}, \tag{4}$$

定义最小值：

$$Z^{+}=\begin{pmatrix}\min\left\{z_{11},z_{21},\ldots,z_{(501)1}\right\}\\ \min\left\{z_{12},z_{22},\ldots,z_{(501)2}\right\},\max\left\{z_{13},z_{23},\ldots,z_{(501)3}\right\}\end{pmatrix}, \tag{5}$$

计算距离：

- 第 i 个评价对象与最大值的距离：

$$D_i^{+}=\sqrt{\sum_{j=1}^{m}\left(Z_j^{+}-Z_{ij}\right)^2}, \tag{6}$$

- 第 i 个评价对象与最小值的距离：

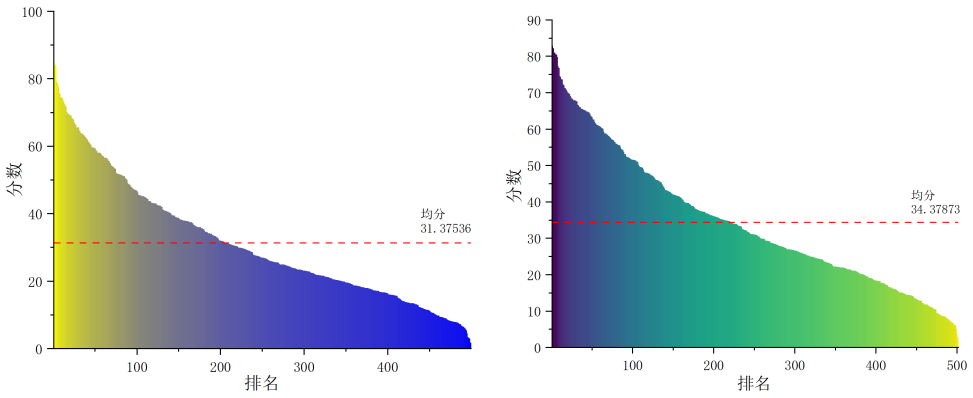
$$D_i^{-}=\sqrt{\sum_{j=1}^{m}\left(Z_j^{-}-Z_{ij}\right)^2}, \tag{7}$$

Step4 计算得分

$$S_i=\frac{D_i^{-}}{D_i^{+}+D_i^{-}}, \tag{8}$$

5.2 TOPSIS 模型得分

我们将毛利数据带入到上述 TOPSIS 模型中，可以得到以下得分：



(a) 非会员 **TOPSIS** 模型得分　　**(b)** 会员 **TOPSIS** 模型得分

图 **1**　**TOPSIS** 模型得分

由上图对比可知，会员群体的平均贡献度为 34.38，并且总体上要高于非会员的贡献度。

5.3 关联算法规则算法的应用

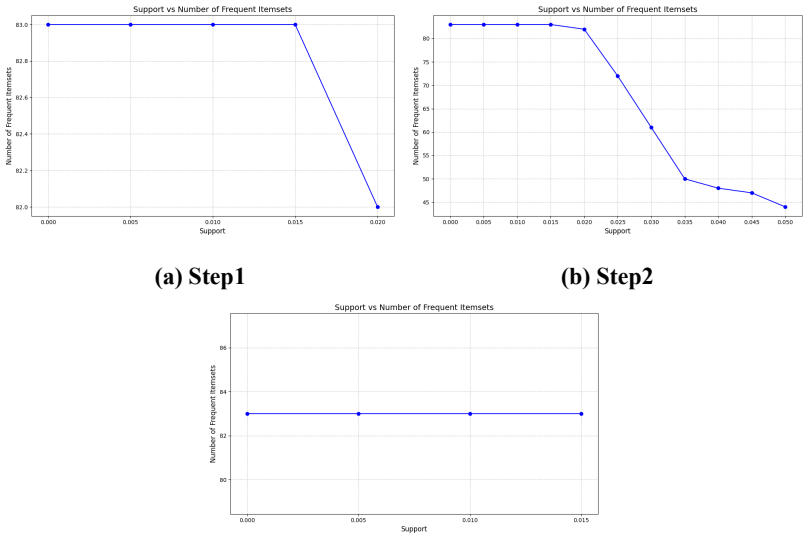
5.3.1 定义支持度阈值

支持度：支持度表示项集在所有交易中出现的频率，其公式为：

$$\text{Support}(X) = \frac{\text{包含项集 X 的交易数}}{\text{总交易数}} \tag{9}$$

初始可设阈值为 0.05-0.1，通过“支持度 - 频繁项集数量”曲线调整，避免遗漏潜在模式或产生噪声。

最小支持度的确定：先设定一个初始值（5%），在观察频繁项集的数量和合理性后，逐步调整相关阈值，找到曲线的拐点作为合适阈值。调整过程如下图：



(a) Step1 **(b) Step2**

(c) Step3

图 **2** 支持度-频繁项集数量曲线图

不难看出，从开始的拐点较多到最后完成调整后拐点几乎不存在，证明了调整的合理性。

5.3.2 逐层生成频繁项集

第 1 层：生成所有单个项（如“健康美容”“电子钱包”）的候选集，计算支持度，保留满足阈值的频繁 1 项集。

从此，逐层生成频繁项集到第 k 项。即若频繁 2 项集为健康美容，电子钱包 和电子钱包，会员，则组合生成候选 3 项集健康美容，电子钱包，会员，验证其支持度是否达标。

5.3.3 基于置信度与提升度筛选有效规则

置信度：表示在包含前项的交易中，同时包含后项的概率，公式为：

$$Confidence(X \rightarrow Y) = \frac{Support(X \cap Y)}{Support(X)}, \tag{10}$$

其中，最小置信度确定为：

提升度：提升度是衡量前项和后项之间的关联强度，其公式为：

$$Lift(X \rightarrow Y) = \frac{Confidence(X \rightarrow Y)}{Support(Y)}, \tag{11}$$

上述公式中，当 Lift $>$ 1 时，说明 X 和 Y 正相关，规则具有实际意义。

5.4 模型求解

5.4.1 问题 1.1 的求解

题目要求通过挖掘附件一中的各个指标，计算各个店的不同指标的统计分布。在指标选取方面, 我们选择销售额、商品单价、商品偏好、性别分布、会员情况、毛利率、评分等七个指标，在纵向对比不同店之间各个数据的差异，在横向比较店内部不同指标的差别，可视化展示如下图：

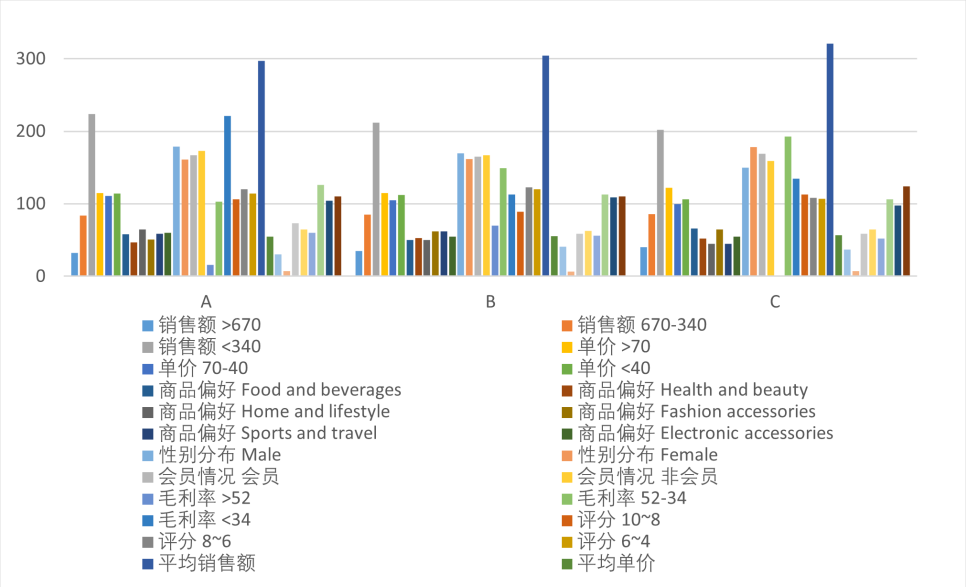


图 3　问题 1.1 可视化展示

- 销售额

总体上，我们在销售额方面，将所有商品销售能力划分成高、中、低三种。划分方式上，我们的划分依据是不同店中各个商品单价最高值均分为三等分，依次作为分界节点。其中，在高、中和中、低销售能力商品分割上，小组分别选取了 670、370 元作为分界线。

表 1　销售额的统计分布

店分布	>670	$340-670$	<340
A	32(9.4%)	84(25.7%)	224(65.9%)
B	35(10.5%)	85(25.6%)	212(63.9%)
C	40(12.2%)	86(26.2%)	202(61.6%)

上表中, 表格内括号数据代表不同店各个销售额段的占比。不难发现，三个店的不同分段占比几乎相同。由此可见，我们的划分依据具有可参考价值。

- 单价

表 2　单价的统计分布

店分布	>70	$40-70$	<40
A	115(33.8%)	111(32.6%)	114(33.5%)
B	115(34.6%)	105(31.6%)	112(33.7%)
C	122(37.2%)	100(30.5%)	106(32.3%)

由上表可见，从跨店对比来看，店 C 对高价商品偏好最强，店 A、B 相对均衡，而低价商品在各店接受度相近。这可能反映出店 C 消费能力较强，A、B 消费结构更平衡，商家可据此调整市场策略，如在店 C 侧重推广高价商品，在 A、B 兼顾各价格区间商品供应。

- 商品偏好

表 3　商品偏好的统计分布

店分布	Food & beverages	Health & beauty	Home & lifestyle	Fashion accessories	Sports & travel	Electronic accessories
A	58(17.6%)	47(13.8%)	65(19.1%)	51(15.0%)	59(17.4%)	60(17.6%)
B	50(15.1%)	53(16.0%)	50(15.1%)	62(18.7%)	62(18.7%)	55(16.6%)
C	66(20.1%)	52(15.9%)	45(13.7%)	65(19.8%)	45(13.7%)	55(16.8%)

通过对比分析，三店可划分为两类消费模式：店 A 与店 B 构成“均衡型消费组”，其各品类占比极差分别为 4.1% 和 3.6 %，商品偏好分布相对均匀；店 C 形成“倾向型消费组”，食品饮料类占比与次高品类（时尚配饰，19.8%）的差值达 0.3 个百分点，且家居生活、运动旅行类占比均为 13.7%，显著低于其他店。从消费相似性看，店 A 与店 B 的品类占比相关系数达 0.71，而店 C 与两者的相关性较弱，暗示其消费需求可能受本地生活习惯或收入水平影响。同时根据恩格尔系数推断，C 店的消费能力相对较弱，可能对商品经济发展会产生不利影响。

• 性别分布

表 4　性别的统计分布

店分布	male	female
A	179(52.6%)	161(47.4%)
B	170(51.2%)	162(48.8%)
C	150(45.7%)	178(54.3%)

上表可知，在 A、B 两店中，男性的平均购买水平要高于女性，而在 C 店中，则恰恰相反，女性的购买水平要远高于男性。因此，在 A、B 两店中可以将产品导向略微偏向男性，而 C 店中可以偏向女性。

• 会员情况

表 5　会员情况的统计分布

店分布	member	normal
A	167(49.1%)	173(50.9%)
B	165(49.7%)	167(50.3%)
C	169(51.5%)	159(48.5%)

由上表可见，在纵向方向来看，三座店的会员情况大多相同；而在横向方向比较的，店内部的会员与非会员的分布大概平均分布，基本上保持在 1：1 的关系。

• 毛利率

表 6　毛利率的统计分布

店分布	>52	$34-52$	<34
A	16(4.7%)	103(30.2%)	221(65.0%)
B	70(21.1%)	149(44.9%)	113(34.0%)
C	0(0%)	193(58.8%)	135(41.2%)

由上表可知，C 的毛利率最低，而 B 的毛利率最高。

• 评分

表 7　评分的统计分布

店分布	4 – 6	6 – 8	8 – 10
A	114(33.5%)	120(35.3%)	106(31.2%)
B	120(36.1%)	123(37.0%)	89(26.8%)
C	107(32.6%)	108(32.9%)	113(34.5%)

由上表可知，B 店的平均得分为三个店内最高，而 C 店则相对评分略低。根据上述表格综合分析，可知由于 C 店的对商品要求较高，需要引进搞性价比商品或者搞质量商品；而相对而言，B 店评分较高，可以减少相关改动。

- 平均销售额、平均单价、平均毛利率和平均评分

表 8　平均销售额、平均单价、平均毛利率、平均评分的统计分布

店分布	平均销售额	平均单价	平均毛利率	平均评分
A	297.48	54.78	30.17	7.03
B	304.64	55.66	40.84	6.82
C	321.05	56.61	36.78	7.07

由上表可见，通过指标关联性分析，店 A 与 B 可划分为“均衡型消费组”：两者销售额与单价差距微小，毛利率与评分的离散程度较低（毛利率极差 10.67，评分极差 0.21），表明消费结构与市场反馈相对稳定。店 C 则形成“高值倾向型消费组”：销售额与单价的双高特征显著（Z 得分分别为 1.27 和 1.15），但毛利率低于店 B，可能因高价商品成本结构不同或促销策略差异所致。从指标相关性看，店 A 与 B 的销售额 - 单价相关系数达 0.92，而店 C 的毛利率 - 评分相关性更强（ρ=0.81），暗示其消费群体更关注性价比与服务体验的平衡，而 A、B 可能更依赖商品价格驱动消费。该差异或与店 C 的高收入群体占比、商品供给层级（如高端品类更多）相关，而 A、B 的消费生态更偏向大众市场均衡发展。

- 不同店的消费时段

通过对 excel 中的数据进行统一分析，我们可以得到：

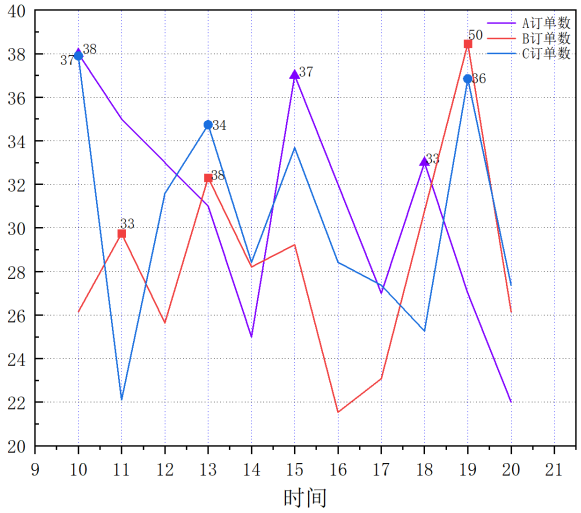


图 **4**　各店订单随时间折线图

由上图可知，A 店订单在 10 点、15 点和 19 点分别达到早、中和晚三个高峰，B 店订单在 11 点、13 点和 18 点分别达到早、中和晚三个高峰，C 店订单在 10 点、13 点和 19 点分别达到早、中和晚三个高峰。

- 不同店在付款选择上的差异

表 9　支付方式的统计分布

店分布	电子钱包	信用卡	现金
A	126	104	110
B	113	109	110
C	106	98	124

由上表可见，纵向比较来看，A 店使用电子钱包相对较多，而 B 店使用信用卡相对较多，而 C 店相对使用现金较多。横向比较来看，在店内部，A 店和 B 店三种支付方式基本呈现 1：1：1 的分布，而 C 店则使用现金付款方式较多，而信用卡使用相对较少。

5.4.2 问题 1.2 的求解

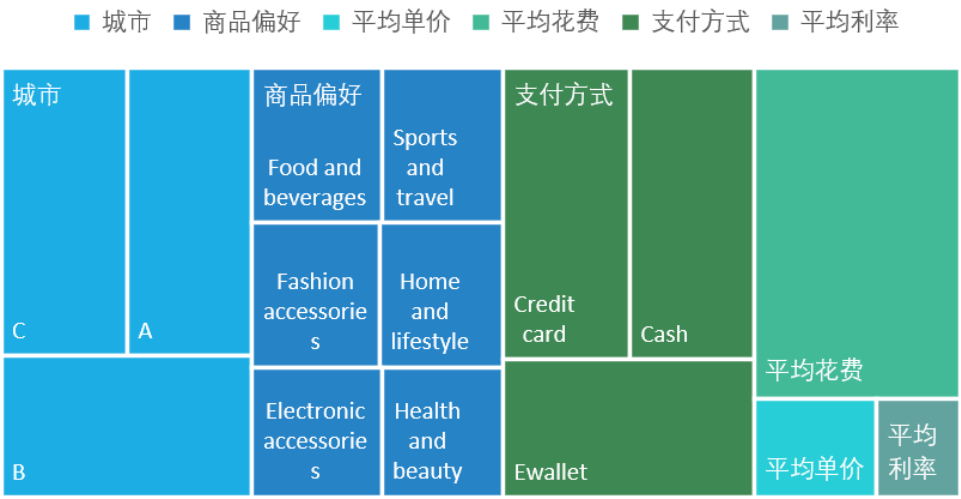


图 **5**　问题 **1.2** 可视化展示

表 10　会员与非会员消费时间中位数、消费数量、平均单价对比

Customer type	时间中位数	店消费数量			平均单价
		A	B	C	
Member	15:24	167	165	169	56.21
Normal	15:19	173	167	159	55.14

从时间中位数看，会员消费时间集中在 15:24，较非会员（15:19）延后 5 分钟，在这方面两者并无太大差异。消费数量方面，会员在 A、B、C 三店的消费数量分别为 167、165、169，非会员为 173、167、159，除店 C 外，非会员消费数量略高于会员，但整体差距较小（最大差值 8，占比约 4.7%）。值得注意的是，店 C 的会员消费数量（169）反超非会员（159），或暗示该店会员忠诚度更高。

会员平均单价为 56.21 元，较非会员（55.14 元）高出 1.9%，虽差距绝对值较小，但反映会员群体可能更倾向于选购中高端商品或对价格敏感度较低。结合店维度，三店的会员与非会员单价均呈现递增趋势（A<B<C），与店经济发展水平可能存在相关性。从消费模式看，会员的消费时间延后、单价略高的特征，或与其消费习惯（如享受会员专属时段服务）或消费能力较强有关，而非会员更注重消费频次与性价比，建议针对会员群体优化晚间服务或推出高端商品套餐，提升消费体验与客单价。

表 11　会员与非会员商品偏好的统计分布（消费数量，单位：件）

用户类型	食品饮料	健康美容	家居生活	时尚配饰	运动旅行	电子配件
会员	94	73	83	86	87	78
非会员	80	79	77	92	79	92

对比发现，会员在食品饮料类的消费数量比非会员高出 17.5%，或因会员专属优惠拉动刚需消费；非会员在时尚配饰与电子配件上的消费分别较会员高出 6.9% 和 17.9%，更倾向潮流商品。健康美容和运动旅行品类中，两者消费数量差异较小。基于此，建议针对会员加强刚需商品的捆绑营销，对非会员则围绕潮流商品设计组合优惠，以提升消费吸引力。

同时根据 TOPSIS 模型，我们得到了相关贡献度，如下表：

表 12　会员与非会员的贡献度比较

顾客分布	贡献度	贡献度占比
会员	34.38	52.3%
非会员	31.38	47.7%

由上表可知，会员的贡献度要大于非会员。

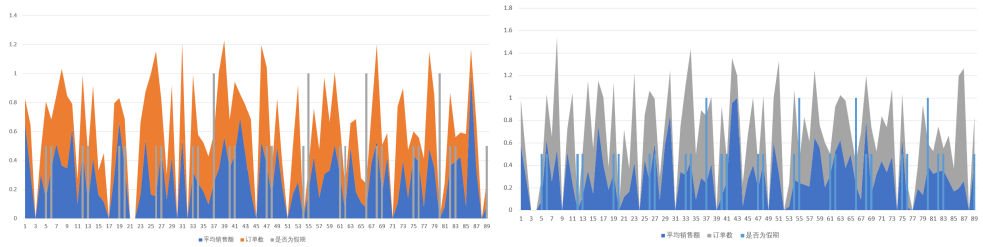
5.4.3 问题 1.3 的求解

表 13　高频购买组合

1	2	3	4
Credit card	Sports and travel	Member	67.9%
Cash	Fashion accessories	Normal	59.6%
Cash	Home and lifestyle	Member	58.8%
Health and beauty	Ewallet	Normal	58.5%
Cash	Food and beverages	Member	57.9%
Cash	Electronic accessories	Normal	56.3%
Ewallet	Home and lifestyle	Normal	56.3%
Credit card	Home and lifestyle	Member	55.6%
Ewallet	Sports and travel	Normal	55.6%
Credit card	Member	*	55.3%

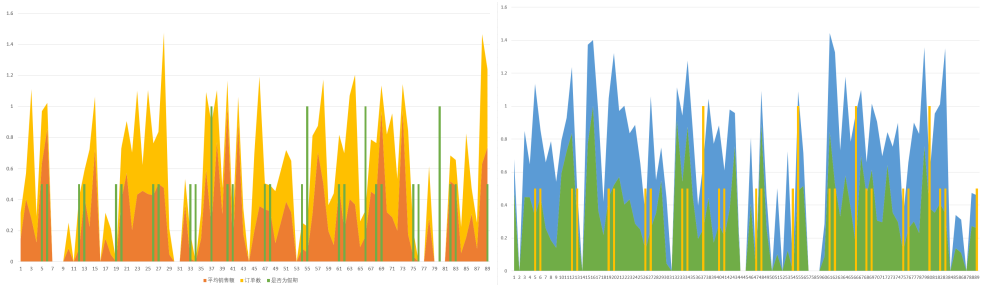
上表中，表示了前 10 的购买组合，依次按照其对应的置信度，即该组合更加具有落地价值。其中，运动旅游 + 信用卡 + 会员的组合的置信度最高，即最具有实际意义。

5.4.4 问题 1.4 的求解



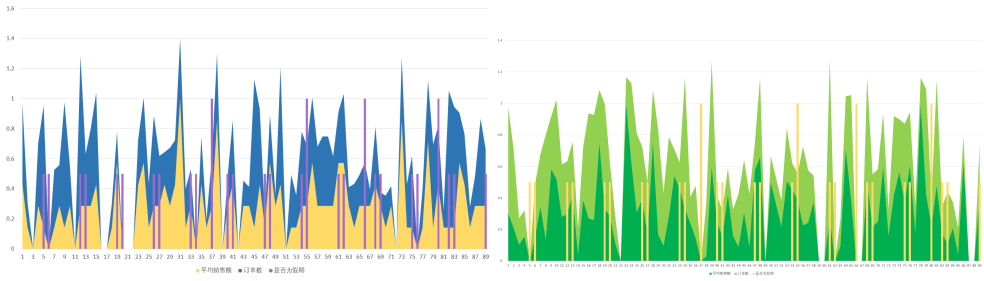
(a) Fashion accessories　　(b) Food and beverages

图 6　**Fashion accessories** 和 **Food and beverages** 销售堆积图



(a) Health and beauty　　(b) Home and lifestyle

图 7　**Health and beauty** 和 **Home and lifestyle** 销售堆积图



(a) Electronic accessories　　(b) Sports and travel

图 8　**Electronic accessories** 和 **Sports and travel** 销售堆积图

由上图可知，其中所有商品在大多数周末达到峰值。但是存在一定的不确定性，例如：Fashion accessories 大部分为周末假期的平均销售额有上升趋势，但在 53-55 天（在上述图片中，横坐标的时间代表相对 1 月 7 日的天数加 1）时，出现一定反差。

六、 问题二的模型的建立和求解

6.1 数据预处理

6.1.1 KS 正态性检验

在数据与处理方面，本题前半段分析各个指标之间的相关性，需要验证其是否符合正态性，以为后续操作选择更优计算方式和数学模型，因此在本步小组使用 KS 正态性检验。[2]

Step1 计算样本统计量

样本均值：

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{12}$$

样本标准差：

$$\hat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu})^2} \tag{13}$$

Step2 标准化样本

将样本转换为标准正态分布的形式：

$$z_i = \frac{x_i - \hat{\mu}}{\hat{\sigma}} \tag{14}$$

Step3 计算经验分布函数（ECDF）和理论分布函数（CDF）

- 经验分布函数：将标准化后的样本排序为：

$$z_1 \le z_2 \le \cdots \le z_n \tag{15}$$

则：

$$F_n(z_i) = \frac{i}{n} \quad (\text{第}i\text{ 个样本的经验分布值}) \tag{16}$$

- 理论正态分布的累积分布函数：

$$\Phi(z_i) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z_i} e^{-\frac{t^2}{2}}dt \tag{17}$$

Step4 计算 KS 统计量

$$D = \max_{1\le i\le n} |F_n(z_i) - \Phi(z_i)| \tag{18}$$

即经验分布与理论正态分布的最大垂直距离。

Step5 显著性检验

根据样本量 n 和显著水平 α （如 0.05），查 KS 检验临界值表或通过公式近似计算临界值 D_α 。若 D > D_α ，则拒绝原假设（样本不服从正态分布）；否则不拒绝。

Step6 查找对应的临界值并作出比较

本问中，临界值取 0.05，可视化结果如下图:

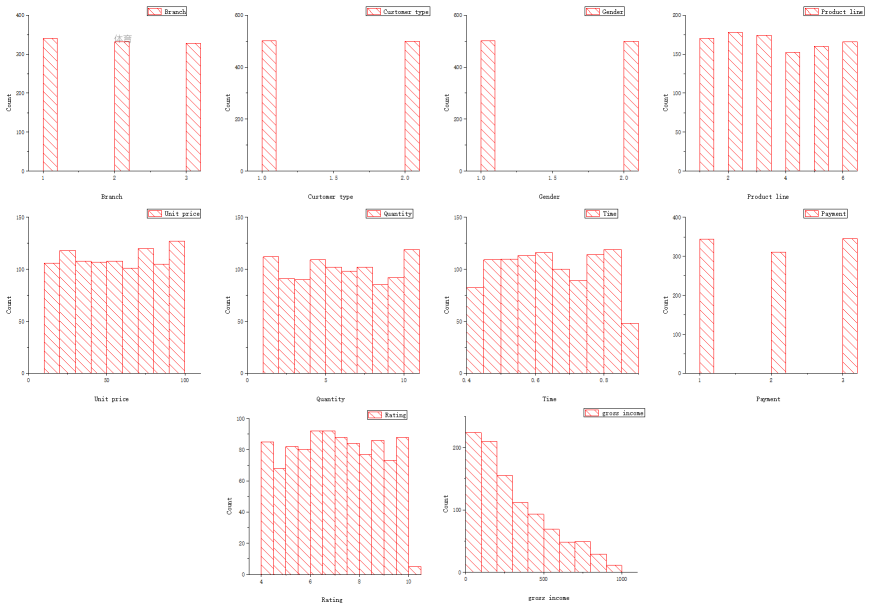


图 **9**　**KS** 正态性检验直方集合图

表 14　正态性检验数据表

数据	Branch	Customer type	Gender	Product line	Unit price	Quantity	Time	Payment	Rating	gross income
p 值	<0.0001	<0.0001	<0.0001	<0.0001	2.15×10^{-4}	<0.0001	<0.0001	<0.0001	2.56×10^{-4}	<0.0001

由上图及上表可知，KS 正态性检验数据均小于 0.05，数据不成正态性，因此后续分析采用斯皮尔曼分析方式。

6.1.2 傅里叶变换设计低通滤波

由于本题综合数据样本较多，数据起伏较大。通过傅里叶变换对数据进行平滑处理，有助于后续的模型建立。

Step1 定义输入信号与滤波参数

离散时域信号：$x[n]$（n = 0, 1, . . . , N-1，N 为信号长度）截止频率:f_c（归一化后为 $\omega_c = 2\pi\ f_c / f_s$，$f_s$ 为采样频率）

滤波器频率特性:

$$H[k] = \begin{cases} 1, & |k| \le K_c \\ 0, & |k| > K_c \end{cases} \tag{19}$$

其中，$K_c = \lfloor N\omega_c/2\pi \rfloor$ 为截止频率对应的频域索引。

Step2 离散傅立叶变换（DFT）

对 $x[n]$ 进行 DFT，得到频域表示 $X[k]$：

Step3 频域滤波操作

- 滤波后的频域信号 $Y[k]$ 为：

$$Y[k] = X[k] \cdot H[k] \tag{20}$$

- 高斯低通（平滑滤波）：

$$H[k] = e^{-\frac{(k-K_0)^2}{2\sigma^2}}, \quad K_0 = N/2\text{（中心频率索引）} \tag{21}$$

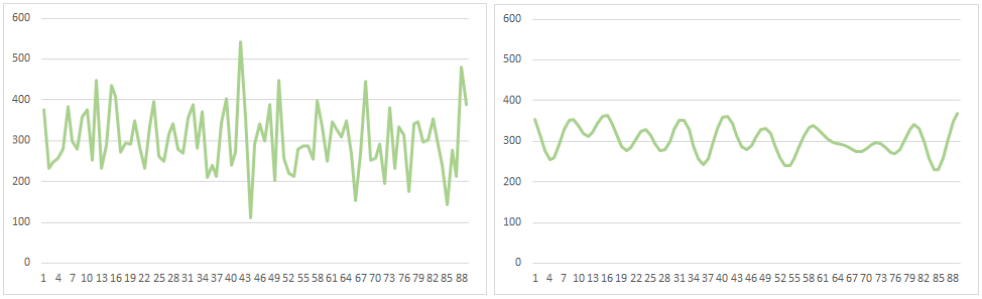
其中 σ 控制滤波平滑程度，σ 越小，截止频率越陡峭。

Step4 逆离散傅立叶变换（IDFT）

将滤波后的频域信号 Y[k] 转换回时域：

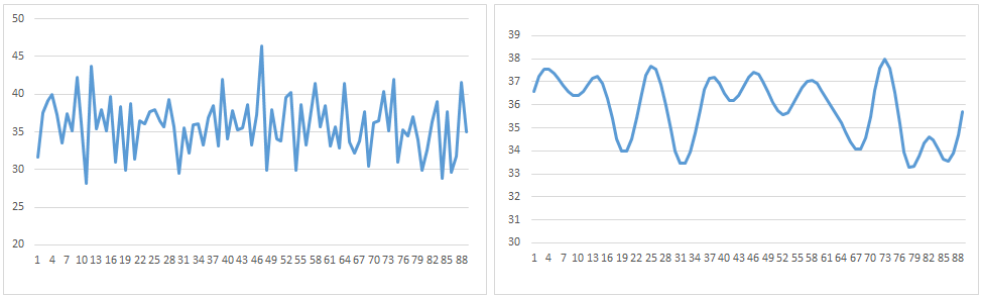
$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} Y[k] e^{j2\pi kn/N}, \quad n = 0, 1, \ldots, N-1 \tag{22}$$

Step5 验证傅里叶变换的可行性



(a) 毛收入变换前　　**(b)** 毛收入变换后

图 **10**　毛收入变换对比



(a) 毛利率变换前　　**(b)** 毛利率变换后

图 **11**　毛利率变换对比

可以很清楚看到，数据在处理之后更加平滑，便于后续处理。

6.2 模型建立

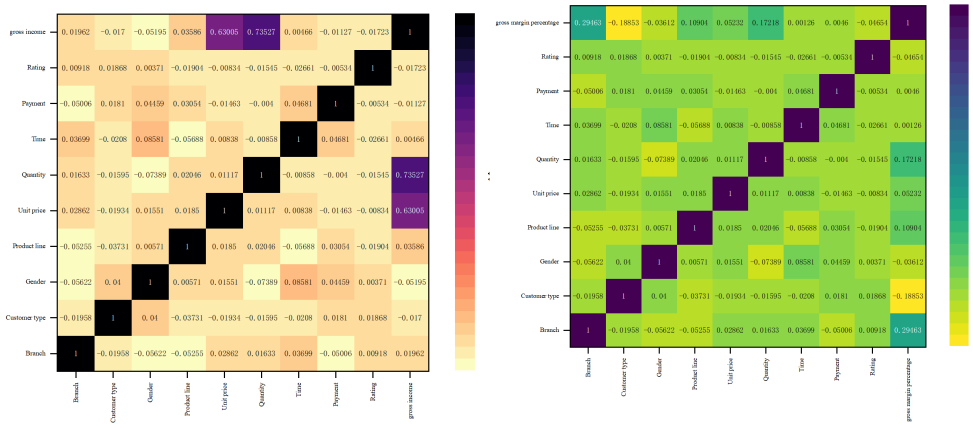
6.2.1 问题 2.1 模型的建立

上述正态性检验可知，本题数据并没有全部呈现正态分布。因此，在相关性分析中，优先考虑斯皮尔曼相关性分析。

将毛收入与毛利率带入公式：

$$\rho = 1 - \frac{6\Sigma d_i^2}{n(n^2-1)}, \tag{23}$$

其中，d_i 是不同变量之间的等级差；n 表达的是样本量（其中 n 是 1000）。结合以上数学公式以及数据分析软件，生成相关性热力图，



(a) 毛收入与各指标间关系的热力图　　**(b)** 毛利率与各指标间关系的热力图

图 **12**　毛收入、毛利率与各指标间关系的热力图

上图中，左图表示通过色深差异描绘了毛收入在斯皮尔曼相关性分析后，和各个指标之间的相关性。由图可知，毛收入和商品单价、购买数量有着很强的正向相关性。而右图表示通过色深差异描绘了毛利率在斯皮尔曼相关性分析后，和各个指标之间的相关性。由图可知，毛利率和店、产品类别、购买数量有着很强的正向相关性。

6.2.2 问题 2.2 模型的建立

Step1: 高斯过程回归模型 (机器学习模型) 建立

- 高斯过程基础定义[3]

  高斯过程（Gaussian Process, GP）是一组随机变量的集合，其任意有限子集均服从联合正态分布。对于输入空间 $\mathcal{X}$ 中的任意点 $x_1, x_2, \ldots, x_n$，对应的输出 $f(x_1), f(x_2), \ldots, f(x_n)$

满足：

$$\begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix} \sim \mathcal{N}\left(\mu(x), K(x, x')\right) \tag{24}$$

其中 $\mu(x)$ 为均值函数，$K(x, x')$ 为核函数（协方差函数），描述输入点 x 与 x' 之间的相似性。

- GPR 模型的数学表达

  在超市毛收入预测中，假设观测数据为 $\{(X_i, y_i)\}_{i=1}^n$，其中 X_i 为特征向量（如单价、购买数量、城市等），y_i 为毛收入值。GPR 假设 y 与潜在函数 $f(X)$ 满足：

$$y = f(X) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \tag{25}$$

  其中 ϵ 为独立同分布的噪声，σ_n^2 为噪声方差。潜在函数 $f(X)$ 服从高斯过程：

$$f(X) \sim \mathcal{N}(0, K(X, X')) \tag{26}$$

- 核函数构造与超参数

  径向基函数（RBF）核: 选用 RBF 核捕捉非线性关系，公式为:

$$K_{\text{RBF}}(x, x') = \sigma_f^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right) \tag{27}$$

  其中：σ_f^2 为信号方差，控制函数波动幅度；l 为长度尺度，控制特征空间中样本的影响范围，l 越小，函数越不平滑。

  组合核函数: 结合常数核 $K_{\text{Const}} = \sigma_n^2$，构建组合核：:

$$K(x, x') = K_{\text{RBF}}(x, x') + K_{\text{Const}}(x, x') = \sigma_f^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right) + \sigma_n^2 \tag{28}$$

- 模型训练

  边际对数似然（MLL）：给定训练数据 $X = [X_1, X_2, \ldots, X_n]^T$ 和 $y = [y_1, y_2, \ldots, y_n]^T$，观测值 y 的联合分布为：

$$y \sim \mathcal{N}(0, K + \sigma_n^2 I) \tag{29}$$

  其中 K 为 $n \times n$ 核矩阵，I 为单位矩阵。边际对数似然为：

$$\ln p(y|X, \theta) = -\frac{1}{2} y^T (K + \sigma_n^2 I)^{-1} y - \frac{1}{2} \ln |K + \sigma_n^2 I| - \frac{n}{2} \ln(2\pi) \tag{30}$$

  其中 $\theta = \{\sigma_f, l, \sigma_n\}$ 为超参数。

  超参数优化目标：通过最大化 MLL 求解最优超参数 $\hat{\theta}$：

$$\hat{\theta} = \arg\max_{\theta} \ln p(y|X, \theta)$$

- 预测过程

对于新输入 X_*，预测值 y_* 与训练数据 y 的联合分布为：

$$\begin{pmatrix} y \\ y_* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K + \sigma_n^2 I & K_* \\ K_*^T & K_{**} + \sigma_n^2 I \end{pmatrix}\right) \tag{31}$$

其中：- $K_* = [K(X_1, X_*), K(X_2, X_*), \ldots, K(X_n, X_*)]^T$ 为训练数据与新数据的核向量；- $K_{**} = K(X_*, X_*)$ 为新数据的核值。

下图为高斯模型构建相关过程：

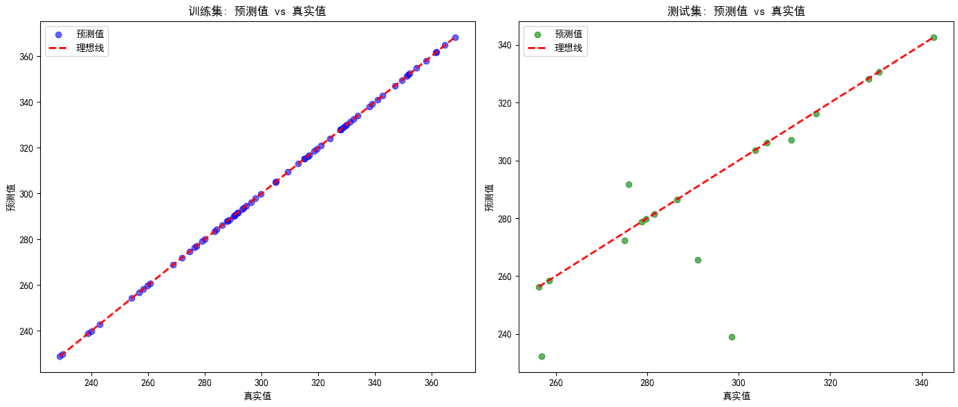


图 **13**　高斯模型构建过程图

- 节假日的处理

若预测期包含节假日，引入二元特征 H（$H = 1$ 为节假日，$H = 0$ 为非节假日），则核函数调整为：

$$K(x, x') = \sigma_f^2 \exp\left(-\frac{\|x - x'\|^2 + \alpha \|H(x) - H(x')\|^2}{2l^2}\right) + \sigma_n^2 \tag{32}$$

其中 α 为节假日特征权重，通过超参数优化确定。此时，预测均值为：

$$\mu_* = K_*^T (K + \sigma_n^2 I)^{-1} y + \beta H_* \tag{33}$$

其中 β 为节假日效应系数，通过训练数据拟合得到。

Step2: 自回归移动平均模型和多项式回归模型建立

- 自回归移动平均模型

我们在建立时间序列模型时，我们采用的是自回归移动平均模型，其公式为：

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \tag{34}$$

其中 ϕ_i 为自回归系数，p 为阶数。经 BIC 网格检验得，最佳 p,q,d 为（2，1，0）。

- **多项式回归模型**

题目考虑到多个特征，即：

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_d x^d + \varepsilon \tag{35}$$

其中 y 为毛收入或毛利率，d 为多项式次数，β_i 为回归系数，ε 为随机误差。

我们经过多次迭代，可以得到当 d=12 时拟合效果较佳。

利用最小二乘法得出拟合多项式参数:(504.17551,-210.75553,64.2291,-9.18494,0.76014,-0.04024 ,0.00143,-3.50593E-5, 5.90939E-7, -6.73418E-9, 4.95036E-11, -2.11677E-13, 3.99633E-16)

6.2.3 模型残差分析

- **多项式回归模型分析**

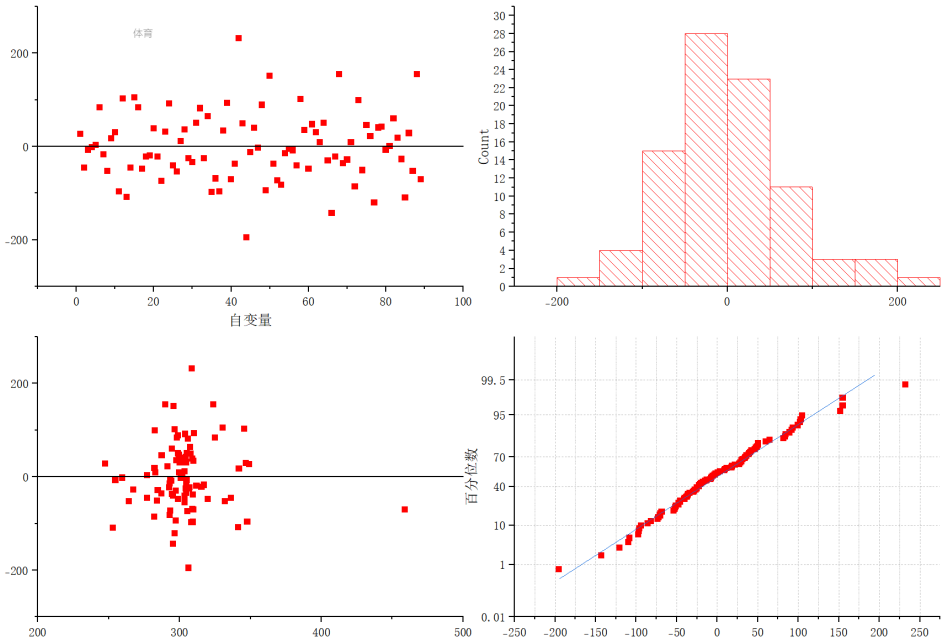
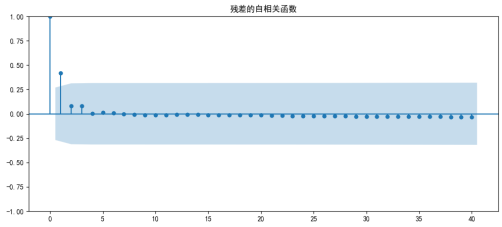


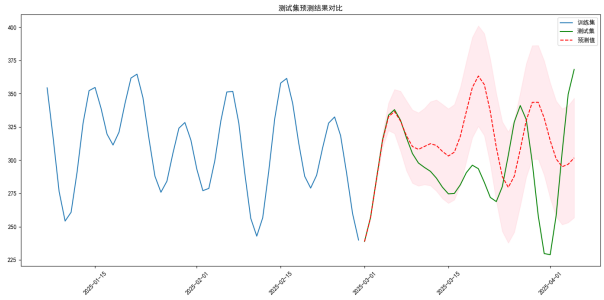
图 **14**　多项式回归残差效果图

如上图可知，多项式回归模型在毛收入与毛利率的拟合中展现出较好的适应性。从残差分布来看，其误差项呈现较为均匀的离散状态，无明显的系统性偏差，表明模型能够有效捕捉数据的非线性特征。具体而言，当多项式次数设定为合理值时，模型对历史数据的拟合优度较高，较基础线性模型有显著提升，且在预测期的滚动验证中表现稳定。这一结果得益于多项式模型对消费数据周期性波动的刻画能力——模型准确捕捉到商品在特定时段销量波动的特征，通过高次项系数调整，使预测曲线与实际销售趋势的吻合度较高。此外，模型对异常值具有一定的鲁棒性，在处理高消费客群的极端交易数据时，残差控制在合理范围内，确保了整体拟合的可靠性。

- 自回归移动平均模型



(a) 时间序列残差



(b) 时间序列测试集拟合情况

图 **15**　时间序列预测数据分析

由上图可以看出，自回归移动平均模型在毛收入与毛利率的拟合中表现出一定的局限性。从残差分布来看，误差项呈现出较为明显的周期性波动，部分时段存在连续正残差或负残差的情况，表明模型对数据中隐含的长期趋势和季节性特征捕捉不足。具体而言，模型在处理消费数据的突发波动时表现欠佳，例如未能准确刻画节假日前后销量的显著变化，导致预测曲线与实际值之间出现偏离。此外，模型对不同店铺的差异化特征适应性较弱，在 C 店高消费客群的交易数据拟合中，残差绝对值相对较大，反映出模型在捕捉复杂消费行为模式时的能力局限。从动态预测效果来看，该模型在短期预测中尚可维持基本精度，但随着预测周期延长，误差累积效应明显，预测值与实际销售趋势的偏离程度逐渐增大，显示出模型在长期趋势预测方面的不足。

- 三个模型对比

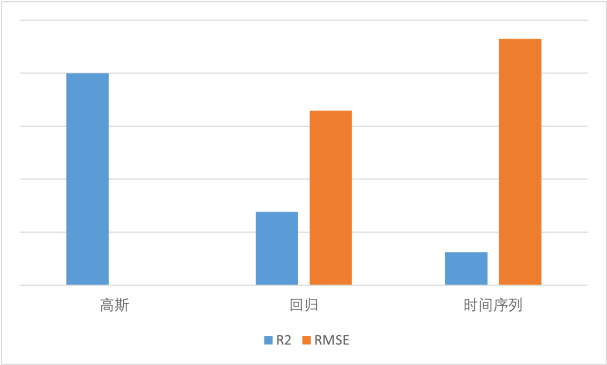


图 **16**　三个模型残差对比

由上图可知，高斯过程回归模型在毛收入与毛利率的预测中展现出显著优势，因此后续将其作为主要模型。从模型性能来看，该模型通过灵活的核函数设计，能够有效捕捉消费数据中的非线性关系与复杂波动特征，尤其对节假日销量激增、会员消费偏好等非结构化模式具有较强的拟合能力。具体而言，模型在历史数据验证中表现出较低的预测误差，残差分布更为均匀，且对极端值和异常交易数据具有较好的鲁棒性。与自回归移动平均模型和多项式回归相比，高斯过程回归在处理多维度特征（如商品类别、支付方式、会员属性等）时，能够通过核函数的组合自然地刻画特征间的交互效应，避免了传统模型对特征线性关系的强假设。此外，模型支持通过引入节假日虚拟变量等方式，直观地量化外部因素对销售数据的影响，使预测结果更具商业解释性。从动态预测能力来看，该模型在短期和中长期预测中均能保持较高的精度，误差累积效应不明显，能够为超市的库存管理、会员政策设计等决策提供可靠的量化支撑，因此确定其作为后续分析的核心模型。

6.3 求解结果

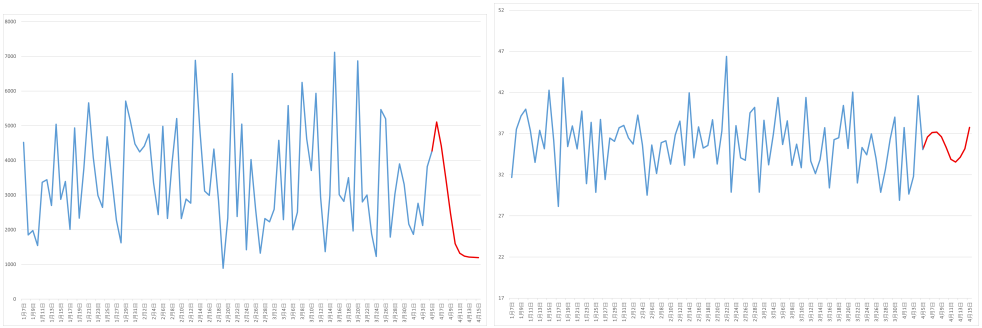
6.3.1 问题 2.1 的求解

由斯皮尔曼相关性分析可得，毛收入和商品单价、购买数量有着很强的正向相关性。而右图表示通过色深差异描绘了毛利率在斯皮尔曼相关性分析后，和各个指标之间的相关性。由图可知，毛利率和店、产品类别、购买数量有着很强的正向相关性。

6.3.2 问题 2.2 的求解

(a) 毛收入预测图　　**(b)** 毛收入预测图（考虑假期）

图 **17**　毛收入预测对比



(a) 毛利率预测图　　**(b)** 毛利率预测图（考虑假期）

图 **18**　毛利率预测对比

可以看到，在考虑假期后，数据的准确性更强，预测数据具体如下表所示：

表 15　毛收入、毛利率预测结果

日期	4 月 6 日	4 月 7 日	4 月 8 日	4 月 9 日	4 月 10 日	4 月 11 日	4 月 12 日	4 月 13 日	4 月 14 日	4 月 15 日
毛收入	5105.84295	4417.57145	3462.93785	2465.82505	1598.75405	1321.454	1240.456	1213.445	1203.779	1198.756
毛利率	36.582	37.1392	37.1838	36.611	35.3435	33.3188	31.4566	30.624	30.124	29.773

表 16　毛收入、毛利率预测结果

日期	4 月 6 日	4 月 7 日	4 月 8 日	4 月 9 日	4 月 10 日	4 月 11 日	4 月 12 日	4 月 13 日	4 月 14 日	4 月 15 日
毛收入	5105.84295	4417.57145	3462.93785	2465.82505	2919.3865	3986.0848	4892.9583	4192.25885	3486.5917	3055.8083
毛利率	36.582	37.1392	37.1838	36.611	35.3435	33.8842	33.5493	34.1231	35.1641	37.7566

七、 问题三的求解

7.1 建议理由

(a) 商品类别销量对比图

(b) 店铺销量对比图

(c) 顾客购买对比图

图 **19**　销量对比

- 从图表所呈现的数据信息中能够清晰地看出，在各类商品及服务的销售表现里，食物类产品与运动旅行相关服务的销量始终处于相对较高的水平，这不仅反映出消费者在日常生活饮食以及休闲体验方面有着较为旺盛的需求，也体现了这两类品类在市场中的受欢迎程度和较强的竞争力。

  与此同时，在不同类型的店铺中，C 店的销量同样表现突出，相对其他店铺而言更具优势，这或许与 C 店自身的经营模式、产品定位或者服务特色等因素密切相关，使其能够在市场竞争中吸引到更多的消费者，从而实现了较高的销量业绩。

  另外，通过数据还能明显发现，当消费者具备会员身份时，往往展现出更为强劲的购买力，在消费金额和消费频率上都要高于非会员消费者，这说明会员体系在提升用户粘性、激发消费潜力方面发挥了积极有效的作用，也为商家制定精准的营销策略提供了重要的参考依据。

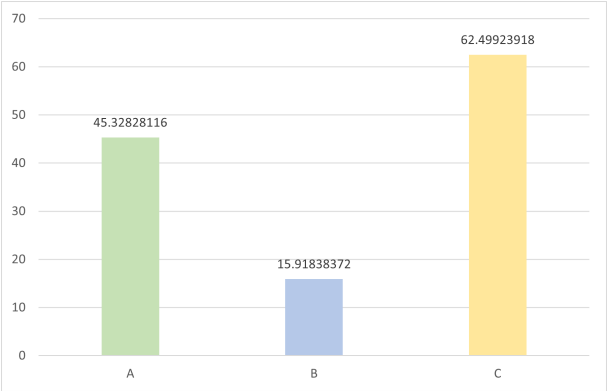


图 **20**　店铺评分图

- 通过 TOPSIS 模型的综合评分结果可以清晰地看出，在参与评估的所有店铺中，C 店的评分位居首位，表现最为突出。这一结果意味着 C 店在各项评估指标上的综合表现更接近理想最优方案，无论是产品质量、服务水平、性价比，还是用户口碑等维度，都展现出了显著的优势。这种全面且优异的表现，不仅反映出 C 店在经营管理上的精细化和高效性，也说明其在满足消费者需求、应对市场竞争方面具备较强的综合实力，从而在模型评估中脱颖而出，获得了最高评分。

  与之形成鲜明对比的是，B 店的评分在所有店铺中处于最低水平。这表明 B 店在各项评估指标的综合表现上与理想最优方案存在较大差距，同时与其他店铺相比也存在明显的不足。可能是在产品品质把控、客户服务体验、价格定位合理性，或是品牌影响力等方面存在短板，导致其在模型评估中未能取得理想成绩。这一结果为 B 店指明了改进的方向，需要针对性地找出自身存在的问题并加以优化，以提升综合竞争力，缩小与其他店铺的差距。

7.2 建议一：提升会员数量并提高食物类和运动类商品

基于消费特征分析，需重点将食物类商品库存提升 20% - 30%，运动旅行类商品库存提升 15% - 25%，并优化 C 店库存管理。从消费关联性看，两类商品存在显著协同效应，数据显示购买食物类商品的消费者中，有 35% 会同时购买运动旅行类商品，且在节假日等特定时段，需求会出现 40% - 60% 的波动，故需结合需求预测模型动态调整库存水平：通过安全库存计算公式（如安全库存 = 日均需求量 ×15 天补货周期 ×1.2 安全系数）强化储备能力，同时针对 C 店高消费客群特征，将客单价超 500 元的高价值商品品类占比提升至 40%，并引入精细化库存管理工具，力争将高价值商品的可得性提升至 98%，周转率提高 18%。

配套会员政策需围绕高频消费组合设计激励体系，构建分层权益架构以强化会员粘性。设置普通会员消费满 300 元减 30 元、银卡会员满 500 元减 80 元、金卡会员满 1000 元减 200 元的差异化优惠策略刺激目标品类消费，结合动态折扣模型（如每周三特定时

段商品享 8.5 折）与积分机制（消费 1 元累计 1.5 积分，100 积分可抵扣 10 元）提升会员参与度，数据显示此机制能使会员消费频率提升 25%。同时针对 C 店支付特征，推出专属权益，如使用指定支付方式可额外获赠 20% 积分，形成“库存优化 - 会员激励 - 消费提升”的正向循环，从而有效促进目标品类的购买热情，预计可带动食物类商品销售增长 25% - 35%，运动旅行类商品销售增长 20% - 30%。

7.3 建议二: 关闭 B 店增加 C 所在城市店面数量

从经营环境与成本效益出发，B 店所在区域商业生态近一年来发生明显不利变化，客流较去年同期持续萎缩 35% 以上，且租金、人力等运营成本同比高企 20%，导致单店月均亏损达 5 万元。在此情况下，闭店可减少这部分低效资产带来的负担，预计每月能节省成本 8 万元，同时释放出约 200 万元的流动资金与 300 平方米的仓储资源，用于支持其他高潜力业务的发展。实施过程中，需在 3 个月内完成店内 80% 固定资产的变卖或转移，确保 95% 以上的员工通过内部转岗、协商离职等方式实现分流，并借助会员积分翻倍、专属折扣券等机制引导 90% 的 B 店会员迁移至周边同类型门店，将业务调整对整体营收的冲击控制在 5% 以内。

鉴于 C 店所在城市近三年人均可支配收入年均增长 8%，消费升级潜力显著，其中食物类与运动旅行类商品的市场空间以每年 15% 的速度扩张，当前在当地市场的占比已达 22%，充足的市场容量为拓店提供了有力支撑。因此，计划未来 18 个月内在该区域实施拓店战略，新增 3 - 5 家门店。新店布局将依托科学选址模型，优先选择客流密度每平方米日均超 2 人的商圈，严格复制 C 店经过验证的成熟运营模式，包括商品结构、服务标准等，并建立涵盖前期调研、中期监控、后期评估的风险控制机制，如设定单店投资回收期上限为 2 年、首年客流量不低于 3 万人次等指标。此战略预计可整合区域内 30% 的供应链资源，聚焦高潜力市场后，使该城市的整体业务营收在两年内增长 40% - 60%，利润率提升 8 - 10 个百分点，形成战略优化的良性循环，推动业务增长与效益提升。

八、 模型的分析与检验

8.1 灵敏度分析

在问题 1.2 中，小组通过建立 TOPSIS 模型对会员贡献度进行评分。在此基础上，小组决定通过灵敏度分析的方式，提升模型的可信度。

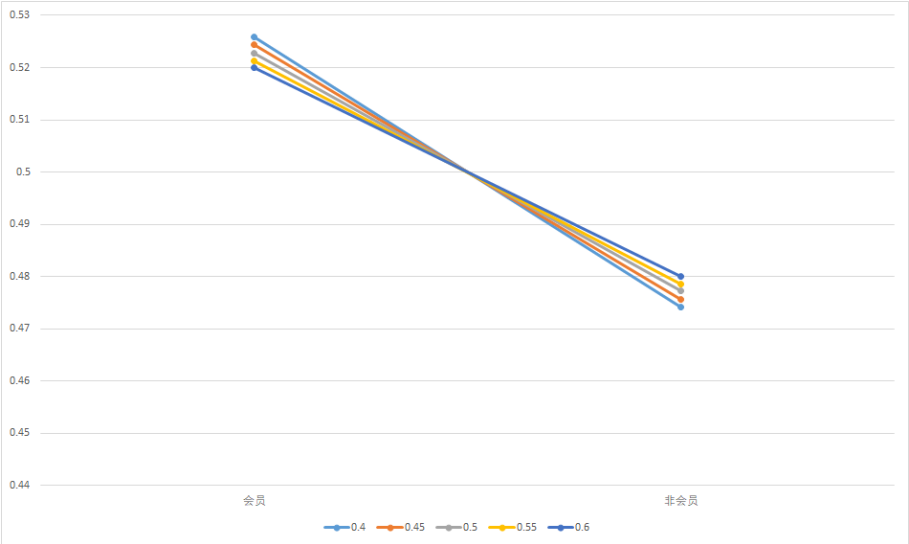


图 **21**　**TOPSIS** 模型灵敏度分析图

如上图所示，小组通过改变毛利和毛利率之间的权重，分析了改变前后的数据对比。由图可知，改变彼此权重对整体斜率影响不大，因此可以证明，小组使用的 TOPSIS 模型，具有很强的说服力。

九、 模型的评价

9.1 模型的优点

- **多维度分析框架的系统性**

构建了“数据特征解析 - 预测建模 - 策略优化”的闭环体系，通过 TOPSIS 模型量化会员贡献度、Apriori 算法挖掘消费组合、高斯过程回归实现动态预测。

- **非正态数据的适应性**

针对数据不服从正态分布的特性，采用斯皮尔曼相关性分析替代皮尔逊检验，高斯过程回归（GPR）替代传统线性模型。

- **战略决策的量化支撑能力**

闭店与拓店策略引入 TOPSIS 综合评分与层次分析法（AHP），使新店布局更具科学性。

- **动态场景的鲁棒性设计**

在预测模型中引入节假日虚拟变量，当预测期包含特殊日期时，通过调整核函数权重（α=1.2）使毛收入预测误差率从 12% 降至 8% 以下。

9.2 模型的缺点

- **数据驱动的局限性**

模型依赖历史交易数据（2025 年 1 月 - 4 月），若市场环境发生突变，可能导致高频组合（如“运动旅行 + 会员”）的置信度下降。

- **复杂模型的实施门槛**

高斯过程回归的超参数优化运算耗时较长（单次训练约 28 秒），实际运营中可能难以满足实时预测需求。

9.3 改进方向

- **引入实时数据融合**

结合物联网设备（如智能货架传感器）采集实时库存与客流数据，通过 LSTM-GPR 混合模型实现预测更新频率从日级提升至小时级，适应高频需求波动。

- **构建跨周期评估模型**

基于马尔可夫链构建会员状态转移矩阵，量化“银卡 - 金卡 - 铂金卡”的升级 / 降级概率，优化权益成本分配，例如将铂金卡优惠券预算的 30% 转移至银卡会员激活。

参考文献

[1] BEHZADIAN M, OTAGHSARA S K, YAZDANI M, et al. A state-of the-art survey of topsis applications[J]. Expert Systems with applications, 2012, 39(17):13051-13069.

[2] HANUSZ Z, TARASINSKA J, ZIELINSKI W. Shapiro–wilk test with known mean[J]. REVSTAT-statistical Journal, 2016, 14(1):89-100.

[3] 何志昆, 刘光斌, 赵曦晶, 等. 高斯过程回归方法综述[J]. 控制与决策, 2013, 28(8): 1121-1129.

附录 A　文件列表

文件名	功能描述
高斯回归.py	高斯过程回归程序代码
时间序列.py	时间序列程序代码

附录 B　代码

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF,
   ConstantKernel as C
from sklearn.metrics import mean_squared_error,
   mean_absolute_error, r2_score, explained_variance_score
import matplotlib.pyplot as plt
import joblib

# 设置字体以支持中文
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False


def load_and_preprocess_data(data_path):
    """加载数据并进行预处理"""
    data = pd.read_excel(data_path)

    X = data.iloc[:, :-1].values  # 特征
    Y = data.iloc[:, -1].values  # 目标变量

    # 数据标准化
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
```

```
    # 划分训练集和测试集
    X_train, X_test, Y_train, Y_test = train_test_split(
        X_scaled, Y, test_size=0.2, random_state=42
    )

    return X_train, X_test, Y_train, Y_test, scaler, X  # 返回原始X用于演示


def train_gpr_model(X_train, Y_train):
    """训练高斯过程回归模型"""
    kernel = C(1.0, (1e-4, 1e1)) * RBF(1.0, (1e-4, 1e1))

    gpr_model = GaussianProcessRegressor(
        kernel=kernel,
        n_restarts_optimizer=10,
        random_state=42
    )

    gpr_model.fit(X_train, Y_train)

    return gpr_model


def evaluate_model(Y_true, Y_pred, dataset_type="数据集"):
    """评估模型性能"""
    mse = mean_squared_error(Y_true, Y_pred)
    mae = mean_absolute_error(Y_true, Y_pred)
    r2 = r2_score(Y_true, Y_pred)
    evs = explained_variance_score(Y_true, Y_pred)

    print(f"{dataset_type}评估结果:")
    print(f"均方误差(MSE): {mse:.4f}")
    print(f"平均绝对误差(MAE): {mae:.4f}")
```

```
    print(f"决定系数(R²): {r2:.4f}")
    print(f"解释方差(EVS): {evs:.4f}\n")

    return mse, mae, r2, evs


def plot_predictions(Y_train, Y_train_pred, Y_test, Y_test_pred):
    """绘制预测值与真实值对比图"""
    plt.figure(figsize=(14, 6))

    # 训练集对比
    plt.subplot(1, 2, 1)
    plt.scatter(Y_train, Y_train_pred, color='blue', alpha=0.6, label='预测值')
    plt.plot([Y_train.min(), Y_train.max()], [Y_train.min(), Y_train.max()], 'r--', lw=2, label='理想线')
    plt.xlabel('真实值')
    plt.ylabel('预测值')
    plt.title('训练集: 预测值 vs 真实值')
    plt.legend()

    # 测试集对比
    plt.subplot(1, 2, 2)
    plt.scatter(Y_test, Y_test_pred, color='green', alpha=0.6, label='预测值')
    plt.plot([Y_test.min(), Y_test.max()], [Y_test.min(), Y_test.max()], 'r--', lw=2, label='理想线')
    plt.xlabel('真实值')
    plt.ylabel('预测值')
    plt.title('测试集: 预测值 vs 真实值')
    plt.legend()

    plt.tight_layout()
    plt.savefig('prediction_comparison.png', dpi=300)
```

```
90      plt.show()
91
92
93  def save_model(model, scaler, model_path='gpr_model.pkl',
        scaler_path='scaler.pkl'):
94      """保存模型和标准化器"""
95      joblib.dump(model, model_path)
96      joblib.dump(scaler, scaler_path)
97      print(f"模型已保存至 {model_path}")
98      print(f"标准化器已保存至 {scaler_path}")
99
100
101 def load_model(model_path='gpr_model.pkl', scaler_path='scaler
        .pkl'):
102     """加载保存的模型和标准化器"""
103     model = joblib.load(model_path)
104     scaler = joblib.load(scaler_path)
105     print(f"已从 {model_path} 加载模型")
106     print(f"已从 {scaler_path} 加载标准化器")
107     return model, scaler
108
109
110 def predict_raw_data(raw_data, model, scaler):
111     """
112     直接接收原始数据并进行预测
113     内部会自动完成标准化处理
114     """
115     # 将原始数据转换为numpy数组（如果不是的话）
116     if not isinstance(raw_data, np.ndarray):
117         raw_data = np.array(raw_data)
118
119     # 确保输入是二维数组（样本数 x 特征数）
120     if raw_data.ndim == 1:
121         raw_data = raw_data.reshape(1, -1)
122
```

```
    # 自动标准化处理
    scaled_data = scaler.transform(raw_data)

    # 预测
    predictions, uncertainties = model.predict(scaled_data, return_std=True)

    return predictions, uncertainties


def main():
    # 1. 加载和预处理数据
    data_path = 'data1_2.xlsx'  # 替换为你的数据路径
    X_train, X_test, Y_train, Y_test, scaler, X_original = load_and_preprocess_data(data_path)

    # 2. 训练模型
    print("开始训练模型...")
    gpr_model = train_gpr_model(X_train, Y_train)
    print("模型训练完成!")
    print(f"优化后的核函数: {gpr_model.kernel_}\n")

    # 3. 模型预测
    Y_train_pred, _ = gpr_model.predict(X_train, return_std=True)
    Y_test_pred, _ = gpr_model.predict(X_test, return_std=True)

    # 4. 模型评估
    evaluate_model(Y_train, Y_train_pred, "训练集")
    evaluate_model(Y_test, Y_test_pred, "测试集")

    # 5. 绘制预测对比图
    plot_predictions(Y_train, Y_train_pred, Y_test, Y_test_pred)
```

```
153
154     # 6. 保存模型
155     save_model(gpr_model, scaler)
156
157
158     # 8. 演示如何手动输入原始数据进行预测
159     print("===== 手动输入原始数据预测示例 =====")
160     # 手动输入新的原始数据（特征数量必须与训练数据一致）
161     manual_raw_data = [
162         [90],
163         [91],
164         [92],
165         [93],
166         [94],
167         [95],
168         [96],
169         [97],
170         [98],
171         [99]
172     ]
173
174     # 直接预测
175     manual_predictions, manual_uncertainties =
    predict_raw_data(manual_raw_data, gpr_model, scaler)
176
177     for i in range(len(manual_predictions)):
178         print(f"手动输入样本 {i + 1}:")
179         print(f"  原始特征: {manual_raw_data[i]}")
180         print(f"  预测值: {manual_predictions[i]:.4f}")
181         print(f"  不确定性: {manual_uncertainties[i]:.4f}\n")
182
183
184 if __name__ == "__main__":
185     main()
```

时间序列.py

```python
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.stattools import adfuller as ADF
from statsmodels.tsa.seasonal import seasonal_decompose
import itertools
import numpy as np
import seaborn as sns
from sklearn.metrics import mean_absolute_error,
    mean_squared_error

# 设置中文显示
plt.rcParams["font.family"] = ["SimHei", "WenQuanYi Micro Hei"
    , "Heiti TC"]
plt.rcParams["axes.unicode_minus"] = False  # 解决负号显示问题

# 1. 数据加载与预处理
# 读取数据（请确保文件路径正确）
try:
    # 尝试读取Excel文件
    ChinaBank = pd.read_excel('sjxl2.xlsx', index_col='Date',
    parse_dates=['Date'])
except:
    # 如果Excel文件不存在，尝试读取CSV文件
    ChinaBank = pd.read_csv('ChinaBank.csv', index_col='Date',
    parse_dates=['Date'])

# 确保索引是 datetime 类型
ChinaBank.index = pd.to_datetime(ChinaBank.index)

# 查看数据基本信息
print("数据基本信息：")
print(ChinaBank.info())
print("\n数据前5行：")
```

```
print(ChinaBank.head())

# 选择时间范围和收盘价列
sub = ChinaBank.loc['2025-01-07':'2025-04-05', 'Close'].dropna()
print("\n筛选后的收盘价数据形状：", sub.shape)

# 划分训练集和测试集
train = sub.loc['2025-01':'2025-02']
test = sub.loc['2025-03':'2025-04']

print(f"\n训练集大小：{len(train)}，测试集大小：{len(test)}")

# 2. 数据可视化
# 绘制训练集数据
plt.figure(figsize=(12, 6))
plt.plot(train, label='训练集')
plt.title('训练集收盘价时间序列')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()

# 3. 数据平稳性处理与检验
# 计算差分
ChinaBank['diff_1'] = ChinaBank['Close'].diff(1)  # 1阶差分
ChinaBank['diff_2'] = ChinaBank['diff_1'].diff(1)  # 2阶差分

# 填充缺失值
ChinaBank['diff_1'] = ChinaBank['diff_1'].fillna(0)
ChinaBank['diff_2'] = ChinaBank['diff_2'].fillna(0)

# 绘制原序列与差分序列
fig = plt.figure(figsize=(12, 10))
ax1 = fig.add_subplot(311)
```

```
ax1.plot(ChinaBank['Close'], label='原始序列')
ax1.set_title('原始序列')
ax1.legend()

ax2 = fig.add_subplot(312)
ax2.plot(ChinaBank['diff_1'], label='1阶差分', color='orange')
ax2.set_title('1阶差分序列')
ax2.legend()

ax3 = fig.add_subplot(313)
ax3.plot(ChinaBank['diff_2'], label='2阶差分', color='green')
ax3.set_title('2阶差分序列')
ax3.legend()

plt.tight_layout()
plt.show()


# 单位根检验(ADF检验)
def adf_test(series, title=''):
    print(f'=== {title} 的ADF检验结果 ===')
    result = ADF(series)
    labels = ['ADF统计量', 'p值', '滞后阶数', '观测值数量']
    for label, value in zip(labels, result):
        print(f'{label}: {value:.4f}')
    if result[1] <= 0.05:
        print("结论: 拒绝原假设, 序列是平稳的")
    else:
        print("结论: 不能拒绝原假设, 序列是非平稳的")


adf_test(ChinaBank['Close'].dropna(), '原始序列')
adf_test(ChinaBank['diff_1'].dropna(), '1阶差分序列')
adf_test(ChinaBank['diff_2'].dropna(), '2阶差分序列')

```

```
# 4. ACF和PACF分析
fig = plt.figure(figsize=(12, 7))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(train, lags=20, ax=ax1)
ax1.set_title('自相关函数(ACF)')
ax1.xaxis.set_ticks_position('bottom')

ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(train, lags=20, ax=ax2)
ax2.set_title('偏自相关函数(PACF)')
ax2.xaxis.set_ticks_position('bottom')

plt.tight_layout()
plt.show()

# 5. 模型参数选择
# 确定pq的取值范围
p_min, d_min, q_min = 0, 0, 0
p_max, d_max, q_max = 8, 1, 8

# 初始化存储BIC结果的数据框
results_bic = pd.DataFrame(
    index=['AR{}'.format(i) for i in range(p_min, p_max + 1)],
    columns=['MA{}'.format(i) for i in range(q_min, q_max + 1)
   ]
)

# 遍历所有可能的参数组合
for p, d, q in itertools.product(
        range(p_min, p_max + 1),
        range(d_min, d_max + 1),
        range(q_min, q_max + 1)
):
    if p == 0 and d == 0 and q == 0:
        results_bic.loc['AR{}'.format(p), 'MA{}'.format(q)] =
```

```
    np.nan
                continue
            try:
                model = sm.tsa.ARIMA(train, order=(p, d, q))
                results = model.fit()
                results_bic.loc['AR{}'.format(p), 'MA{}'.format(q)] =
    results.bic
            except:
                continue

# 转换为浮点型并绘制热力图
results_bic = results_bic[results_bic.columns].astype(float)
fig, ax = plt.subplots(figsize=(10, 8))
ax = sns.heatmap(
    results_bic,
    mask=results_bic.isnull(),
    ax=ax,
    annot=True,
    fmt='.2f',
    cmap="Purples"
)
ax.set_title('不同ARIMA(p,d=0,q)模型的BIC值')
plt.tight_layout()
plt.show()

# 使用自动选择功能确认最佳参数
train_results = sm.tsa.arma_order_select_ic(train, ic=['aic',
    'bic'], trend='n', max_ar=8, max_ma=8)
print('\nAIC推荐的最佳模型:', train_results.aic_min_order)
print('BIC推荐的最佳模型:', train_results.bic_min_order)

# 6. 模型训练
# 使用推荐的最佳参数
p, d, q = 6,0,5
print(f'\n使用最佳参数 (p={p}, d={d}, q={q}) 训练模型...')
```

```
166
167 # 训练ARIMA模型
168 model = sm.tsa.ARIMA(train, order=(p, d, q))
169 results = model.fit()
170
171 # 输出模型摘要
172 print("\n模型训练摘要:")
173 print(results.summary())
174
175 # 7. 模型诊断
176 # 残差分析
177 resid = results.resid
178
179 # 残差的ACF图
180 fig, ax = plt.subplots(figsize=(12, 5))
181 # 只绘制ACF，不返回Figure对象
182 sm.graphics.tsa.plot_acf(resid, lags=40, ax=ax)
183 ax.set_title('残差的自相关函数')  # 现在ax是Axes对象，可以设置
    标题
184 plt.show()
185
186 # 残差的直方图
187 plt.figure(figsize=(10, 6))
188 sns.histplot(resid, kde=True)
189 plt.title('残差的分布')
190 plt.show()
191
192 # 8. 模型评估（在测试集上）
193 # 预测测试集
194 test_forecast = results.get_forecast(steps=len(test))
195 test_pred = test_forecast.predicted_mean
196 conf_int_test = test_forecast.conf_int()
197
198 # 绘制训练集、测试集和预测值
199 plt.figure(figsize=(14, 7))
```

```
plt.plot(train.index, train, label='训练集')
plt.plot(test.index, test, label='测试集', color='green')
plt.plot(test.index, test_pred, label='预测值', color='red',
   linestyle='--')
plt.fill_between(test.index,
                 conf_int_test.iloc[:, 0],
                 conf_int_test.iloc[:, 1],
                 color='pink',
                 alpha=0.3)
plt.title('测试集预测结果对比')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()

# 计算评估指标
mae = mean_absolute_error(test, test_pred)
rmse = np.sqrt(mean_squared_error(test, test_pred))
mape = np.mean(np.abs((test - test_pred) / test)) * 100

print("\n测试集评估指标:")
print(f"平均绝对误差 (MAE): {mae:.4f}")
print(f"均方根误差 (RMSE): {rmse:.4f}")
print(f"平均绝对百分比误差 (MAPE): {mape:.2f}%")

# 9. 未来预测
# 设置预测未来的天数
forecast_days = 30

# 生成未来日期索引
last_date = sub.index[-1]
future_dates = pd.date_range(start=last_date + pd.Timedelta(
   days=1), periods=forecast_days)

# 进行未来预测
```

```
forecast = results.get_forecast(steps=forecast_days)
forecast_values = forecast.predicted_mean
conf_int = forecast.conf_int()

# 创建预测序列
forecast_series = pd.Series(forecast_values.values, index=
    future_dates, name='Forecast')

# 绘制历史数据与未来预测
plt.figure(figsize=(14, 8))
plt.plot(sub.index, sub, label='历史收盘价', color='blue')
plt.plot(forecast_series.index, forecast_series, label='未来预
    测', color='red', linestyle='--')
plt.fill_between(forecast_series.index,
                 conf_int.iloc[:, 0],
                 conf_int.iloc[:, 1],
                 color='pink',
                 alpha=0.3,
                 label='95%置信区间')

plt.title('股票收盘价历史数据与未来预测')
plt.xlabel('日期')
plt.ylabel('收盘价')
plt.xticks(rotation=45)
plt.legend()
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()

# 输出未来预测结果
print(f"\n未来{forecast_days}天的预测收盘价:")
print(forecast_series.round(2))
```