中国研究生创新实践系列大赛
“华为杯”第二十一届中国研究生
数学建模竞赛

学　　校		杭州电子科技大学
参赛队号		
队员姓名	1.	张丁鹏
	2.	欧阳利鑫
	3.	杨鸿铭

中国研究生创新实践系列大赛

“华为杯”第二十一届中国研究生

数学建模竞赛

题 目： **在线零售行业中商品影响力分析**

摘　　要：

本文基于在线零售数据集，对客户购买行为、商品销售趋势以及客户关系网络进行了全面分析。首先，通过数据预处理和特征提取，识别了关键商品和客户群体。随后，使用多种预测方法（**简单平均法、加权平均法、指数平滑法、移动平均法、ARIMA 模型和灰色预测**）对商品销量进行预测，并通过客户关系网络模型深入分析客户间的购买关联性。最后，通过构建个性化推荐模型和客户关系演化网络，评估了促销活动的效果，活动的转化率和品牌影响力。

针对问题一，我们首先对数据进行了预处理，删除了退货，折扣等无效的交易记录共 **8829** 行，占全部 406829 行数据的 **2.17%**。接着，我们使用 **FP-growth** 算法对数据进行关联规则挖掘，识别出高频项集和强关联规则。最终，我们分析了这五个商品对其他商品购买行为的影响，如 **85099B 和 20725 这个商品组合的支持度是 0.0229，置信度是 0.265625，提升度是 3.819724593**，这代表人们在购买 85099B 这个商品后很可能会同时购买 20725；然后，我们找出了购买数量最多的前五个商品，其商品代码为 **84077，22197，85099B，84879，21212**，它们分别是 WORLD WAR 2 GLIDERS ASSTD DESIGNS，SMALL POPCORN HOLDER，JUMBO BAG RED RETROSPOT，ASSORTED COLOUR BIRD ORNAMENT 和 PACK OF 72 RETROSPOT CAKE CASES。

针对问题二，首先，我们建立了以商品销售数量和销售频次计算热销程度评分的模型，并通过**熵权法**计算了的权重。接着，我们采用多种时间序列预测方法（**简单平均法、加权平均法、指数平滑法、移动平均法、ARIMA 模型和灰色预测**）对商品的销售数量进行预测，并选择了预测误差最小的 ARIMA 模型作为最优方法，参数为**(1,0,0)**。随后，我们预测了第六个月的畅销商品分别为 **47566，85123A，84755，84077，85099B**，并使用所有数据进一步预测了未来一个月的畅销商品 **84879，85123A，23203，85099B，22197**。此外，我们还以 6 月购买金额最多的客户的消费金额作为预算金额，建立购买第一问中找到的五件商品方案的模型，以购买数量为自变量，设置总预算约束和根据热销程度设置各商品预算下限，购买商品数量最多为目标建立整数线性规划模型，使用 **Gurobi** 求解器求得购买各商品最优方案为 **12992，7784，5297，3826，11186**。

针对问题三，我们构建了**基于客户购买行为的关系网络模型**，并对该网络的拓扑特

征进行了分析。具体来说，我们将客户之间的购买行为建模为无向图，其中节点代表客户，边代表客户共同购买的商品，边的权重表示共同购买商品的数量。通过对网络的拓扑特征进行计算，得到了如下结果：**平均介数为 0.0001**，**平均度分布为 0.5364**，**平均聚类系数为 0.7837**，网络直径为 **3**，网络平均最短路径长度为 **1.4636**，具有**小世界特性**和**无标度特性**，这表明该网络的客户群体比较紧密，有大部分类似的购买记录。

针对问题四，我们通过客户关系网络模型，识别出网络中度中心性最大的前五个客户当作网络中最重要的五个客户，其中的客户 ID 分别为 **8407、22197、85099B、84879 和 21212**。我们还确定了被最多不同客户购买的前五件商品当作网络中最重要的五件商品，其中分别是 **22423，85123A，47566，84879，22720**。随后，通过分析这些商品的购买模式，我们发现第四问的商品市场覆盖面较广，而第一问的商品大部分是由少数客户多次购买的。最后，我们基于协同过滤和关联规则方法，构建了个性化推荐模型，向指定客户推荐了购买概率较高的前五个商品,如推荐客户 17949 购买 **22423，23245，22699，22197，22697** 商品。

针对问题五，我们构建了基于时间序列的客户关系演化网络模型，并分析了 2011 年每个月的网络特征，发现平均度分布从 0.3769 逐渐增加到 0.4215，平均介数从 0.000854 下降至 0.000348，平均聚类系数从 0.6764 上升至 0.6967。基于全年数据，预测了 2012 年 1 月的网络特征，结果显示**平均度分布为 0.3372，平均介数为 0.0007，平均聚类系数为 0.6623**。此外，我们通过引入促销指数对促销活动效果进行了评估，发现促销活动在 6 月和 11 月对客户购买行为产生了显著影响，促销商品的平均促销指数在这两个月份显著增加，且客户转化率也相应提高，反映出促销活动对客户群体的影响显著。活动的转化率通过客户级别和商品级别的转化率进行评估，客户级别的转化率为 $\text{客户级别转化率} = \left(\frac{\text{购买促销商品的客户数}}{\text{参与促销活动的总客户数}}\right) \times 100\%$，而商品级别的转化率则是 $\text{客户级别转化率} = \left(\frac{\text{购买促销商品的客户数}}{\text{参与促销活动的总客户数}}\right) \times 100\%$，品牌影响力则通过分析变化率 $\text{高中心性客户变化率} = \left(\frac{\text{促销后高中心性客户数} - \text{促销前高中心性客户数}}{\text{促销前高中心性客户数}}\right) \times 100\%$ 来评估。

关键词：FP-growth 算法，熵权法，客户关系网络图，ARIMA，客户购买行为

目录

一、问题重述

1.1 问题的背景

在线零售（或称为线上零售）是指通过互联网或其他电子渠道，针对个人或者家庭、企业的需求销售商品或提供服务。在线零售的起源可以追溯到互联网技术的兴起和普及。随着互联网技术的快速发展，特别是万维网（WWW）的出现，电子商务逐渐进入人们的视野。早期的在线零售主要集中在书籍、音乐等数字化商品的销售上，1995 年，亚马逊和 eBay 的成立标志着在线零售新时代的到来。这些平台通过提供便捷的在线购物体验，打破了传统零售的地域和时间限制。后来，基于互联网技术的普及、电商平涌的涌现、支付方式的多样化、物流体系的完善，在线零售业务获得了长足的发展。在线零售的发展经历了初期探索（图书、音乐数字化产品）、快速扩张（服装、家电、日用品）、多元化发展（生鲜、药品、跨境电商）和即时零售（下单即达购物）等重要的阶段。从初始的“18 天不出门仅用互联网生活”挑战到现在的“不购物的一周”体验生活，人们的生活方式也发生巨大的变化。

对于科学家、社会学家和经济学家而言，不仅要享受在线零售带来的生活便利，而且要分析和研究在线零售背后隐藏的一些规律和原理。他们重点关注的问题包含商品分析、库存分析、购买者（或用户）分析、销售活动分析、物流布局和全渠道分析等。近年来，在线零售领域的研究取得了丰硕的成果，如即时零售模式在疫情期间得到快速推广，成为消费者重要的购物方式之一，并在未来几年将继续保持高速增长。大数据和人工智能技术在在线零售领域得到广泛应用，可有效帮助企业精准定位目标客户、优化产品线和营销策略。通过不断优化网站设计、提高页面加载速度、简化购买流程等措施，用户体验得到显著提升，进而提升了转化率和用户满意度。越来越多的企业开始尝试全渠道布局，通过整合线上线下资源，提升购物体验和市场竞争力

1.2 问题的提出

现有一跨国在线零售的数据集，里面包含 2010 年 12 月 1 日至 2011 年 12 月 9 日期间一家英国注册非商店在线零售商的所有交易记录。整个数据包含发票编号、变易代码、商品描述、数量、发票日期、单价、用户 ID 和国家等信息且不包含缺失值。当前，有关这一数据集的研究大部分忽略了商品描述信息后的关于用户或交易时间的分析，这将忽略商品购买中的有价值信息。而对一个在线零售商，分析哪些商品畅销、消费者的特征、商品的品牌影响力等是他更加关心的问题。所以我们的目标是利用数据进行各种分析其商品和客户群体还有交易时间之间的联系，发现其中的商品和消费者特征帮助店铺进行更好的运营和营销活动。

问题一：每个客户购买商品的习惯是客户分析的基础。通过客户的购买记录，请建立合适的数学模型找出客户购买行为的关联模式，找出客户购买最多的前五个商品，并对这五个商品之间的相关性进行分析。并且分析这五个商品对其他商品购买产生的影响。

问题二：利用前五个月的销售商品数据建模合适的模型预测第六个月的畅销商品，

再把这个模型推广到利用全部数据预测后一个月的畅销商品。确定所有商品的购买意愿，然后给出某个客户（比如购买商品金额最多的客户）在一定的预算金额下购买第一问中找到的五件商品的最优模型。

问题三：如果一个商品被两个或多个客户所购买，则认为这些客户之间存在某种联系（如共同的需求或爱好），请建立客户关系的网络模型。分析这个模型的相关拓扑特征，并给出合适的解释说明。

问题四：基于第三问建立的客户关系网络模型，建模获得这个在线零售商最重要的前五个客户，及此网络中的五个商品，比较这五个商品与第一问的五个商品的差异性并给出相关解释。请建立数学模型帮助这个在线零售商针对热销商品的特定客户群体进行个性化推荐。

问题五：请根据所有数据建立客户关系的演化网络模型，并分析此模型在未来一个月内的演化特征。基于此模型，评价在线零售商促销活动的效果、转化率和品牌影响。

二、符号说明和基本假设

3.1 基本假设

1、假设原始数据无误。

2、数据中无漏记订单。

3、无除折扣外的促销活动，例如买一送一，以旧换新等。

4、客户 ID 和商品 ID 对应的客户和商品唯一。

3.2 符号说明

符号	说明
x_i	商品i的购买数量
u	商品购买预算
w_i	商品i的热销程度评分
p_i	商品i的价格
ω	权重
Q	销售量
F	销售次数
V	节点集
E	边集
N_k	度为k的节点数量
N	总节点数量
E_i	节点i的邻居的边的数量
C_i	节点i的聚类系数
$l_{i,j}$	节点i和节点j的最小路径长度
C_a	促销活动后的平均中心性值
C_b	促销活动前的平均中心性值
N_n	新增客户数量
N_c	总客户数量

三、问题一

3.1 问题分析

通过对客户购买记录的分析，问题一的主要任务是建立数学模型，识别客户购买行为的关联模式，找出客户购买最多的前五个商品，并分析这些商品之间的相关性。此外，还需要研究这五个商品对其他商品购买行为的影响。具体而言，通过数据预处理，识别出客户购买的高频商品，然后利用 FP-growth 算法分析商品之间的关联性，最终识别出具有强关联的商品组合。

3.2 数据预处理

在本次数据分析中，我们首先对`Online Retail`进行了预处理，以确保后续分析的准确性和有效性。预处理的步骤如下：

1.**删除无效发票数据**：对于 `InvoiceNo` 列中首位为 `C` 的数据，表示这些记录是交易取消的数据，我们找到对应数量以及客户 ID 的订单进行删除。

2.**删除无效的商品交易数据**：对于 `Quantity` 列中小于等于零的数据，我们视为无效交易（例如退货或错误输入），因此也将这些记录删除。

3.**处理客户 ID 缺失数据**：对于 `CustomerID` 列中缺失的数据，我们假设这些记录是散客购买行为。为了统一处理，我们将这些缺失的 `CustomerID` 填充为 `0`，以便后续分析时能够区分散客和注册客户。

4.**计算销售额**：为了进行销售分析，我们引入了一个新的列 `TotalPrice`，其值为 `Quantity` 与 `UnitPrice` 的乘积。该列表示每条交易记录的总销售额。

5.**数据分割**：在计算销售总额时，我们将包括散客的所有数据用于计算；但在进行用户信息挖掘时，我们只使用有有效 `CustomerID` 的数据，即不包含散客的数据。

3.3 购买最多的前五个商品

根据购买数量进行排序，得到客户购买最多的前五个商品如下表所示：

表 3-1 购买最多的前五个商品

商品代码	描述	购买数量
84077	二战滑翔机，款式多样	54127
22197	小号爆米花盒	48946
85099B	大号红色复古点袋	43108
84879	各色鸟形装饰品	35296
21212	72 只装复古蛋糕盒	33501

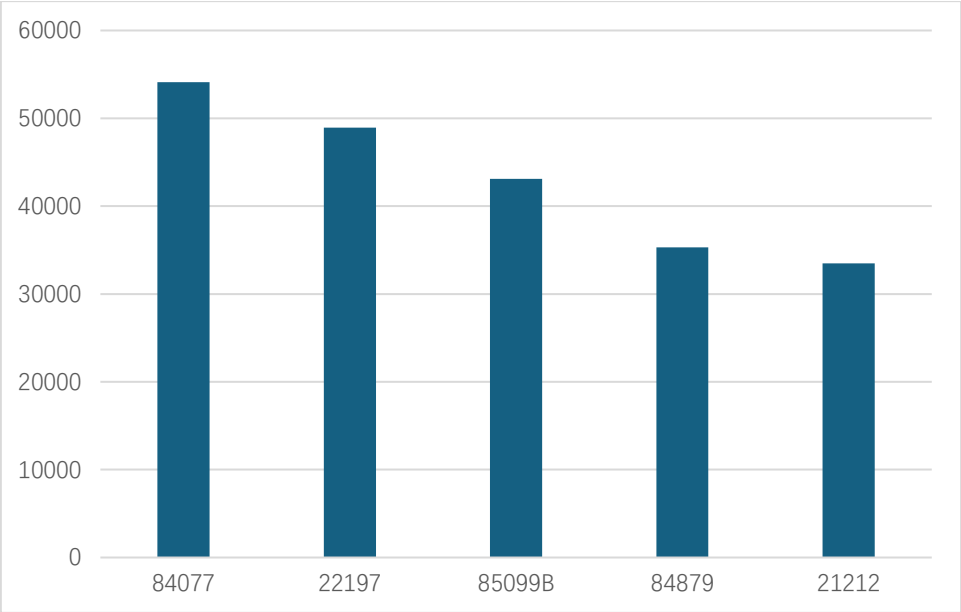

图 3-1 购买最多的五个商品柱状图

3.4 基于 FP-growth 算法的关联度分析

在本题中，我们对购买最多的前五个商品之间的关联性及其对其他商品购买行为的影响进行了分析。分析步骤如下：

1. 关联规则挖掘

我们首先使用 `FP-growth` 算法从交易数据中提取频繁项集，并根据支持度和置信度筛选出有意义的关联规则。

2. 筛选条件

支持度（support）：选择支持度大于 0.025 的规则，这意味着这些规则在交易数据中出现的频率较高。

置信度（confidence）：选择置信度大于 0.4 的规则，表示在前件出现的情况下，后件出现的概率较高。

提升度（Lift）是关联规则挖掘中的一个重要指标，用于衡量商品之间的关联性。它表示在给定前件的情况下，后件出现的概率与后件在所有交易中独立出现的概率的比值。换句话说，提升度指标帮助我们判断前件和后件之间的关联性是否大于随机出现的情况。

该步骤的目的是筛选出在数据集中具有较强关联性的商品组合。

表 3-2 较强关联性的商品组合

antecedents	consequents	support	confidence	lift
22726	22727	0.028593008	0.671736375	14.19761169
22727	22726	0.028593008	0.604332953	14.19761169
82494L	82482	0.025248166	0.577065351	12.21059744
82482	82494L	0.025248166	0.534246575	12.21059744
22386	85099B	0.029456193	0.626865672	7.262238806
22382	20725	0.025194217	0.472672065	6.797090297
20725	22384	0.028215365	0.405740884	8.078209488
22384	20725	0.028215365	0.561761547	8.078209488

20727	22384	0.025032369	0.441064639	8.78149747
22384	20727	0.025032369	0.498388829	8.78149747
20725	20727	0.02789167	0.401086113	7.067045813
20727	20725	0.02789167	0.491444867	7.067045813
20727	22383	0.025086319	0.442015209	7.855411233
22383	20727	0.025086319	0.445829338	7.855411233
20725	22383	0.027999568	0.402637704	7.155601606
22383	20725	0.027999568	0.497603068	7.155601606
22697	22699	0.029186448	0.7829233	18.53418427
22699	22697	0.029186448	0.690932312	18.53418427
23203	85099B	0.025248166	0.433333333	5.020166667

3.5 商品之间的关联分析

1. 购买最多的前五个商品之间的共现分析

为了研究购买最多的前五个商品之间的关联性，我们计算了这些商品在同一订单中出现的次数，并生成了共现矩阵。共现矩阵中的每个元素表示两个商品在同一订单中同时出现的次数。较大的数值意味着这些商品更频繁地同时出现在订单中。为直观地观察商品之间的共现关系，使用热力图展示共现矩阵如下。

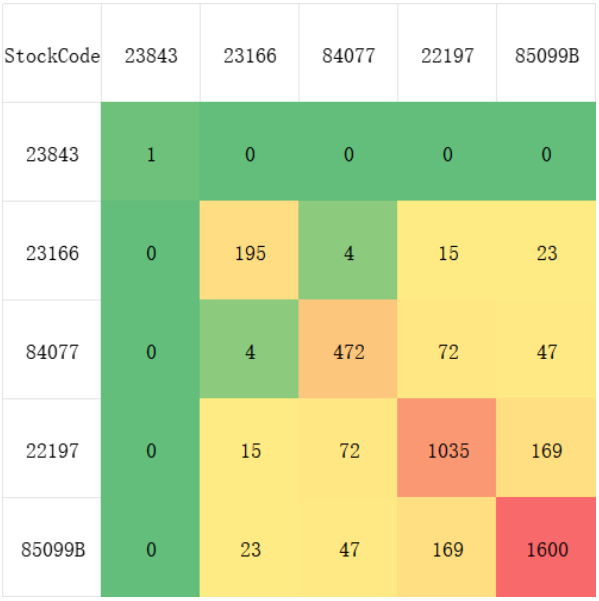

图 3-2 共现矩阵热力图

2. 购买最多的前五个商品对其他商品购买的影响

我们还分析了购买最多的前五个商品对其他商品购买行为的影响。由于篇幅有限，我们选择支持度大于 0.01，置信度大于 0.2 的规则，筛选出与购买最多的前五个商品有强关联性的其他商品进行展示。

表 3-3 五个商品与其他商品的关联性

antecedents	consequents	support	confidence	lift
85099B	20725	0.0229	0.265625	3.819724593
85099B	22386	0.0294	0.3412	7.2622
22386	23203	0.0103	0.3516	6.0353
23203	22386	0.0103	0.4102	8.7307

85099B	21931	0.0234	0.2712	6.4959
22386	21931	0.0118	0.4029	9.6495
21931	22386	0.0118	0.5069	10.7877
22411	21931	0.0100	0.4722	11.3089
21931	22411	0.0100	0.4308	10.1097
85099B	85099C	0.0195	0.2262	6.4618
85099B	21929	0.0178	0.2068	6.3698
85099B	22411	0.0213	0.2475	5.8071
22386	22411	0.0101	0.3443	8.0789
22411	22386	0.0101	0.4747	10.1032
20725	22384	0.0100	0.44	8.7603
22384	20725	0.0100	0.6750	9.7078
85099B	85099F	0.0223	0.2587	7.3336
22386	85099F	0.01251	0.4249	12.0429
85099F	22386	0.01251	0.5603	11.9257
85099B	23199	0.0189	0.2193	5.6634
23203	23209	0.0108	0.4316	7.9528
85099B	23203	0.0108	0.7921	13.5957
85099B	23203	0.0252	0.2925	5.0201
85099B	23202	0.0186	0.2156	5.1772
23203	23202	0.0121	0.4829	11.5947
23202	23203	0.01219	0.6550	11.2429

四、问题二

4.1 问题分析

在问题二中，我们的任务是利用前五个月的销售数据来预测第六个月的畅销商品，并将该模型推广应用到预测后一个月的畅销商品。为了实现这一目标，我们首先需要建立一个评估商品畅销度的模型，该模型应当能够综合考虑商品的销售数量和销售频率。接着，我们需要使用熵权法为这些指标赋予合理的权重。随后，通过对比多种时间序列预测方法（**简单平均法、加权平均法、指数平滑法、移动平均法、ARIMA 模型和灰色预测**）确定最佳的预测方法，并利用该方法对第六个月及未来一个月的畅销商品进行预测。此外，我们还需要针对购买金额最多的客户，在一定预算下给出了该客户购买最优商品组合的模型。

4.2 热销程度模型的定义

热销程度模型用于评估商品的畅销度。模型公式为：

$$\text{Bestseller\_Score} = w_1 \times Q + w_2 \times F \tag{2-1}$$

其中，Bestseller_Socre 表示热销程度，Q表示商品的销售量，F表示商品的销售次数。权重w_1和w_2的值通过熵权法进行计算。

4.3 熵权法计算权重

熵权法是一种客观赋权方法，通过衡量数据的离散程度来确定各指标的权重。具体步骤如下：

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{2-2}$$

1. 数据标准化：为了消除量纲的影响，对原始数据进行 Min-Max 标准化，将所有数据值转换到 [0,1] 区间。公式如下：

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{2-3}$$

在我们的模型中，我们对商品的销售量Q和销售频率F分别进行标准化，得到标准化后的值Q`和$F$`。

2. 计算指标的比例矩阵P_{ij}：

对于每个标准化后的值，计算其在对应指标列中的比例，公式为：

$$P_{ij} = \frac{x'_{ij}}{\sum_{i=1}^{n} x'_{ij}} \tag{2-4}$$

其中，x'_{ij} 表示第 i 个样本在第 j 个指标下的标准化值，n 为样本总数。

计算信息熵 E_j：信息熵反映了指标在整个数据集中的不确定性，公式为：

$$E_j = -\frac{1}{\ln(n)}\sum_{i=1}^{n} P_{ij}\ln(P_{ij}+\epsilon) \tag{2-5}$$

其中n是样本数量，ϵ是一个很小的数，用于避免$ln(0)$的情况。

4. 计算权重 w_j：

根据信息熵计算权重，公式为：

$$w_j = \frac{1-E_j}{\sum_{j=1}^{m}(1-E_j)} \tag{2-6}$$

其中，m是指标的数量。在我们的模型中，两个指标分别为销售量和销售频率，计算得出对应的权重ω_1和ω_2分别为 **0.2648** 和 **0.7352**。

4.4 多种方法预测效果对比

在确定了权重之后，下一步是确定最佳预测方法，我们使用前五个月的数据来预测其中十件商品第六个月的销售量，我们尝试了多种时间序列模型，包括**简单平均法、加权平均法、指数平滑法、移动平均法、ARIMA 模型和灰色预测**模型。得到的预测值如下表：

表 4-1 每种方法预测的销售量

商品	简单平均	加权平均	指数平滑	移动平均	ARIMA	灰色预测	真实值
22616	2956.0	2160.0	1910.0	2393.0	2056.0	11870.0	3096
85123A	3356.0	3266.0	3550.0	3163.0	3356.0	22050.0	2988
21977	2257.0	2758.0	3071.0	3034.0	3324.0	16221.0	1778
17003	1978.0	1876.0	2437.0	2034.0	1948.0	11942.0	1459
84879	2484.0	2474.0	2631.0	2635.0	2248.0	14786.0	1989
22197	2955.0	3580.0	4336.0	3524.0	2955.0	17461.0	2397
85099B	3248.0	3324.0	3240.0	3535.0	3320.0	19314.0	1920
15036	2748.0	2735.0	2976.0	2980.0	2357.0	15730.0	1516
21212	3738.0	3042.0	3193.0	3249.0	2769.0	18658.0	3529
84077	4908.0	5823.0	6352.0	6352.0	4800.0	32393.0	1618

进一步得到预测误差：

表 4-2 每种预测方法的平均误差

方法	平均预测误差
简单平均	1005.3
加权平均	1139.7
指数平滑	1346.3
移动平均	1249.5
ARIMA	**999.3**
灰色预测	15813.5

最终，通过对比各模型的预测误差，我们选择了 ARIMA 模型用于后续的预测。

4.5 使用 ARIMA 模型预测畅销商品

ARIMA 模型是一个广泛用于时间序列预测的统计模型。它包括三个参数：`p`（自回归项数），`d`（差分次数）和`q`（移动平均项数）。在本次预测中，我们使用自动化的方法选择最佳参数，并对每个商品的销售量和销售频率进行预测。

具体步骤如下：

创建时间序列：对于每个商品，我们将前五个月的销售数据构造成时间序列。

参数选择与模型拟合：我们使用自动化的方法为每个时间序列选择最佳的 ARIMA 模型参数。该方法通过多种组合测试`p`、`d`、`q`的值，选择预测误差最小的组合，并对其进行拟合。

进行预测：在模型拟合后，我们使用该模型对第六个月的销售量和销售频率进行预测。

处理预测结果：预测完成后，我们将预测值处理为最接近的整数销售量和频率。

在得到所有商品的销售量和销售频率预测值后，我们通过标准化处理，使得不同商品的销售量和频率具有可比性。标准化后的值用来计算每个商品的热销评分。

使用**前五个月的数据预测第六个月畅销商品**，结果如下表所示：

表 4-3 第六个月畅销商品预测值

商品代码	数量	购买次数	标准化销售量	标准化购买次数	畅销评分
47566	2824	217	0.6400	1.0000	0.7837
85123A	3398	167	0.7418	0.7764	0.7556
84755	3976	108	0.8440	0.5151	0.7127
84077	4857	37	0.0481	0.1995	0.6804
85099B	2892	99	0.6681	0.4744	0.6098

使用**全部数据预测后一个月畅销商品**，结果如下表所示：

表 4-4 下一年一月的畅销商品预测

商品代码	数量	购买次数	标准化销售量	标准化购买次数	畅销评分
84879	4069	106	0.9757	0.5598	0.8097
85123A	2767	222	0.7058	0.9579	0.8064
23203	2440	235	0.6379	1.0000	0.7824
85099B	3552	123	0.8686	0.6189	0.7689
22197	3783	80	0.9165	0.4704	0.7384

4.6 一定预算下的最优模型

1、问题假设与数据处理

通过以上计算，我们求出了第 1 题提到的客户购买最多的 5 件商品在 6 月份的热销程度评分和商品单价如表所示，此外，我们假设以 6 月份消费金额最多的用户消费金额为预算，以每个商品热销程度作为购买该商品的预算比例依据，且 80%的预算按热销程度分配，剩余 20%自由分配，以购买的商品总数最多为目标建立优化模型。

通过数据处理，我们找到 6 月消费最多的是 CustomerID 为 18102 的客户，消费金额为 41959.44

表 4-5 第 1 题购买最多的五个商品 6 月价格和热销程度表

商品代码	价格	热销程度
84077	0.29	0.28
22197	0.84	0.4859
85099B	2.03	0.7991
84879	1.69	0.4806
21212	0.54	0.4489

2、模型建立

(1)决策变量

每个商品购买数量x_i

$$x_i \in N^+; i = 1,2..5 \tag{2-7}$$

其中N^+表示正整数

(2)目标函数

最大化商品购买总数

$$max \sum_{i=1}^{5} x_i \tag{2-8}$$

(3)约束条件

i. 总花费不超过预算

$$\sum_{i=1}^{5} p_i \times x_i \leq u \tag{2-9}$$

ii. 每个商品的花费不低于其预算

$$p_i \times x_i \geq 0.8 \times u \times \frac{w_i}{\sum_{i=1}^{5} w_i}; i = 1,2,\dots 5 \tag{2-10}$$

(4)整体模型

$$max \sum_{i=1}^{5} x_i$$

$$s.t. \begin{cases} \sum_{i=1}^{5} p_i \times x_i \leq u \\ p_i \times x_i \geq 0.8 \times u \times \frac{w_i}{\sum_{i=1}^{5} w_i}; i = 1,2,\dots 5 \\ x_i \in N^+; i = 1,2..5 \end{cases}$$

使用 Gurobi 求解器解得最优方案如下表所示：

表 4-6 最优购买方案表

商品	数量
二战滑翔机，款式多样	12992
小号爆米花盒	7784
大号红色复古点袋	5297
各色鸟形装饰品	3826
72 只装复古蛋糕盒	11186

五、问题三

5.1 问题分析

问题三的主要任务是通过客户的购买记录构建一个客户关系网络模型，并对其进行拓扑特征分析。具体而言，客户关系被建模为一个无向图，其中每个节点代表一个客户，如果两个客户购买了相同的商品，则在他们之间连接一条边。通过分析该网络的度分布、聚类系数、平均路径长度和连通分量等拓扑特征，可以揭示客户之间的联系强度、群体结构以及客户购买行为的相互关联性。

5.2 客户关系网络模型的建立与拓扑特征分析

5.2.1 客户关系网络模型的构建

在本问题中，客户关系被建模为一个无向图 $G=(V,E)$，其中：

节点集 V：代表所有的客户，每个节点对应一个客户。

边集 E：如果两个客户购买了同一个商品，则在这两个客户之间连接一条边。

权重 W ：客户连接的边的权重，两个客户购买同一件商品的次数作为权重值。

具体来说，若客户 i 和客户 j 都购买了商品 k，则在图中添加一条连接节点 i 和节点 j 的无向边。

该客户关系网络使用邻接矩阵 A 表示。矩阵中的元素 A_{ij} 定义如下：

$$A_{ij}=\begin{cases}1 & \text{如果客户 } i \text{ 和客户 } j \text{ 购买了同一商品}\\ 0 & \text{否则}\end{cases} \tag{5-1}$$

5.2.2 拓扑特征分析

在建立客户关系网络模型后，网络的拓扑特征分析包括以下几个方面：

(a) 度分布

度分布 $P(k)$ 用于描述网络中节点的度的分布情况。节点的度 d_i 表示与节点 i 相连的边的数量。度分布 $P(k)$ 的定义为：

$$P(k)=\frac{N_k}{N} \tag{5-2}$$

其中，N_k 表示度为 k 的节点数量，N 是总节点数。

(b) 聚类系数

聚类系数 C_i 衡量一个节点的邻居之间相互连接的程度。对于客户关系网络，节点 i 的聚类系数 C_i 计算公式为：

$$C_i=\frac{2E_i}{d_i(d_i-1)} \tag{5-3}$$

其中，E_i 表示节点 i 的邻居之间实际存在的边的数量，d_i 是节点 i 的度。整个网络的平均聚类系数 C 为所有节点聚类系数的平均值：

$$C = \frac{1}{N}\sum_{i=1}^{N} C_i \tag{5-4}$$

(c) 网络的平均路径长度（Average Path Length）

平均路径长度 L 描述网络中任意两个节点之间的最短路径长度的平均值，表示客户之间的关联距离。其计算公式为：

$$L = \frac{1}{\frac{N(N-1)}{2}} \sum_{i \neq j} l_{ij} \tag{5-5}$$

其中，l_{ij} 是节点 i 和节点 j 之间的最短路径长度。

(d) 连通分量（Connected Components）

连通分量是指在图中，任意两个节点之间都存在路径相连的最大子图。通过分析连通分量的数量和大小，可以识别出客户群体之间的分隔情况，揭示不同客户群体的购物行为是否独立或相互关联。

5.3 结果可视化分析

为了可视化效果图 5-1 仅保留了购买商品数大于等于 4000 的节点，展示了客户关系网络中的所有客户节点以及他们之间的关系。每个节点代表一个客户，节点之间的边表示两个客户购买了相同的商品，且边的权重为他们共同购买商品的次数。网络图突出了客户群体之间的关系，尤其是那些购买相同商品次数较多的客户之间的紧密联系。这些群体形成了若干个子网络，表明在整个客户群体中，存在一些客户群体彼此之间有较强的购买行为关联。

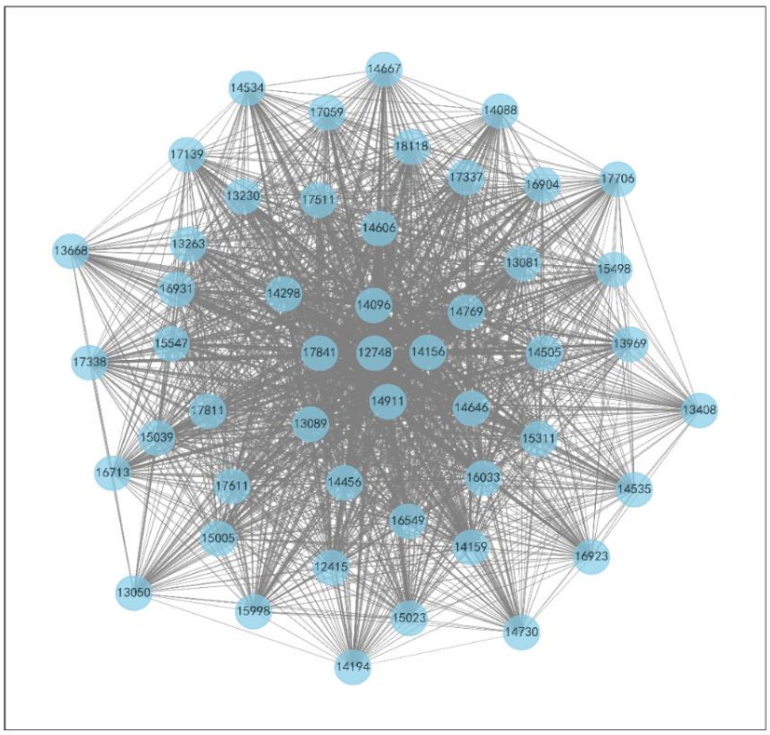

图 5-1 客户关系网络图

其中根据拓扑特征分析中的指标公式计算得到了具体数值如表所示，其中网络平均度分布和平均聚类系数较高代表该客户网络具有较强的紧密性。

表 5-1 拓扑特征指标值

特征指标	数值
平均介数	0.0001
平均度分布	0.5364
平均聚类系数	0.7837
网络直径	3
网络平均最短路径长度	1.4636

六、问题四

6.1 问题分析

问题四的主要任务是基于已建立的**客户关系网络模型**，识别在线零售商中最重要的前五个客户，并分析这些客户的购买行为，同时比较这些客户购买的五件商品与问题一中客户购买最多的五件商品之间的差异性。最后，通过建立**基于协同过滤和关联规则的个性化推荐模型**，为在线零售商提供针对热销商品的特定客户群体的个性化推荐方案。

6.2 基于客户关系网络模型的最重要客户与商品分析

6.2.1 网络中最重要的前五个客户

在第三问中，我们通过分析客户关系网络模型，确定了该网络中度中心性最大的前五个客户。这些客户的 ID 分别为：**8407，22197，85099B，84879，21212**。度中心性衡量的是一个节点连接到其他节点的数量，度中心性越高，说明该客户与其他客户之间的联系越紧密，这些客户在整个网络中具有较高的影响力。因此，这些客户被视为在线零售商的最重要客户。

6.2.2 网络中最重要的前五件商品

通过分析客户关系网络模型，我们还识别出被最多不同客户购买的五件商品。这些商品分别是：

表 6-1 网络最重要前五件商品

商品代码	描述	购买客户数
22423	摄政风格三层蛋糕架	881
85123A	白色悬挂心形蜡烛灯座/奶油色悬挂心形蜡烛灯座	856
47566	派对彩旗	708
84879	各色鸟形装饰品	678
22720	三件套食品储存罐设计	640

这些商品尽管可能没有被某些客户大量购买，但它们在不同客户中具有广泛的吸引力，因此在整体客户群体中具有高覆盖率。

6.2.3 重要商品对比分析

第一问得到的购买最多的前五个商品见表 4-1，一四问中的**相同项是 84879，各色鸟形装饰品**，不同项有 22720，22423，47566，85123A。

其中一四问中唯一的相同商品是 **84879，各色鸟形装饰品。**该商品不仅在大量客户中被购买（体现了它在整体客户群体中的广泛吸引力），还受到了特定客户群体的多次重复购买。这表明该商品具有跨越不同客户群体的普遍吸引力，能够满足多样化的需求，无论是偶尔购买的普通客户还是对特定商品有深度兴趣的客户都对其产生了购买行为。

通过比较第一问和第四问中的商品，可以发现以下几点重要的差异：

1. 购买模式的差异：

第四问的商品：这些商品是被最多不同客户购买的，表明它们在广泛的客户群体中具有普遍的吸引力，即使每个客户可能只购买了一次或少量这些商品。它们的市场覆盖面较广，满足了不同客户的基本需求或装饰需求。

第一问的商品：这些商品大部分是由少数客户多次购买的，反映了特定客户的个人偏好或重复购买行为。这类商品通常具有较强的娱乐性、收藏价值或个性化特征，吸引了那些对特定商品有较大需求的客户。

2. 购买商品的客户群体的差异：

第四问中购买重要商品的客户：在第四问中购买重要商品的客户群体，他们的购买行为显示出广泛的联系性，但他们的购买商品与整体客户群体的购买习惯有所不同。这些客户在整个网络中的高连接性意味着他们可能会影响其他客户的购买决策，零售商应关注这些客户的需求并为其提供定制化服务。

第一问中购买重要商品的客户：这些客户购买了大量的特定商品，这些商品在整体客户群体中的覆盖面较小，但它们表现出较高的个人偏好和兴趣。零售商可以通过分析这些客户的购买模式，提供个性化的推荐和服务。

6.2.4 相关解释与策略建议

通过对这些商品和客户的差异性分析，可以提出以下策略建议：

1. 个性化营销策略：

针对第四问的商品：这些商品适合广泛的市场推广和普遍的营销活动。由于它们被大量不同客户购买，可以通过增加市场曝光和推广力度来进一步提高销售额。

针对第一问的商品：这些商品更适合精准营销和个性化推荐。零售商可以利用客户数据，向那些具有类似购买习惯的客户推荐这些商品，以增加客户的购买频率和忠诚度。

2. 客户关系管理：

最重要的前五个客户：零售商应特别关注这些度中心性高的客户，分析他们的购买行为，了解他们的需求和偏好。通过提供个性化服务和推荐，提升这些客户的满意度和购买黏性。

3. 交叉销售策略：

结合不同商品的优势：零售商可以将第四问中的商品与第一问中的商品进行组合销售。例如，在客户购买广受欢迎的商品时，向其推荐具有个性化特征的商品，从而实现交叉销售，提升整体销售业绩。

通过这些策略，零售商可以更好地满足不同客户群体的需求，提升整体的市场竞争力和销售业绩，同时保持最重要客户的高满意度和忠诚度。

6.3 基于协同过滤和关联规则的个性化推荐模型

6.3.1 数据准备

首先，我们需要收集并整理有关客户购买历史的数据，包括客户 ID、商品 ID、购买次数、购买日期等。

客户矩阵 C：记录每个客户的购买行为，矩阵的元素 C_{ij} 表示客户 i 购买商品 j 的次数。

商品矩阵 P：记录每个商品的属性（如类别、价格等），矩阵的元素 P_{jk} 表示商品 j 的第 k 个属性。

6.3.2 热销商品的识别

使用前面提到的热销程度模型来识别热销商品。热销程度模型公式为：

$$\text{Bestseller\_Score} = w_1 \times \text{Quantity\_norm} + w_2 \times \text{Frequency\_norm} \tag{6-1}$$

其中，w_1 和 w_2 分别是销量和销售频次的权重。

6.3.3 协同过滤模型

为了计算每个客户对每个热销商品的兴趣度，我们采用协同过滤的方法。协同过滤模型通过以下公式计算客户 i 对商品 j 的兴趣度 S_{ij}：

$$S_{ij} = \sum_{k \in N(i)} \text{sim}\,(i,k) \cdot C_{kj} \tag{6-2}$$

其中，$N(i)$ 是与客户 i 最相似的 k 个客户集合，$\text{sim}(i,k)$ 是客户 i 与客户 k 之间的相似度，C_{kj} 是客户 k 对商品 j 的购买记录。

相似度 $\text{sim}(i,k)$ 可以通过余弦相似度或皮尔逊相关系数计算：

$$\text{sim}\,(i,k) = \frac{\sum_{j \in P(i) \cap P(k)} C_{ij} \cdot C_{kj}}{\sqrt{\sum_{j \in P(i)} C_{ij}^2} \cdot \sqrt{\sum_{j \in P(k)} C_{kj}^2}} \tag{6-3}$$

其中，$P(i)$ 和 $P(k)$ 分别表示客户 i 和客户 k 已经购买的商品集合。

6.3.4 关联规则推荐

除了协同过滤之外，我们还可以通过关联规则对客户进行商品推荐。通过 FP-Growth 或 Apriori 算法挖掘客户的购买记录，找到频繁项集和关联规则。

每条关联规则形如：

$$\{X\} \Rightarrow \{Y\} \tag{6-4}$$

其中，X 和 Y 是商品集合。根据客户已经购买的商品 X，我们可以推荐给客户 Y 中的商品，尤其是那些尚未被该客户购买的商品。

6.3.5 最终推荐模型

将协同过滤和关联规则的推荐结果结合，计算最终的推荐评分 R_{ij}：

$$R_{ij} = \alpha \cdot S_{ij} + \beta \cdot A_{ij} \tag{6-5}$$

其中，S_{ij} 是协同过滤计算的兴趣度，A_{ij} 是通过关联规则计算的推荐得分，α 和 β 是权重系数，用于调整两种推荐方法的影响以提高推荐的精准度。

6.3.6 个性化推荐结果

最终我们的个性化推荐模型为客户推荐了以下产品，其中推荐商品的顺序依据个性化推荐模型的评分进行排序：

表 6-2 客户个性化推荐商品单

客户 ID	推荐商品 1	推荐商品 2	推荐商品 3	推荐商品 4	推荐商品 5
17949	22423	23245	22699	22197	22697
12931	84879	22139	23203	23084	22197
16333	22086	21915	84077	21891	22620
14646	22423	85123A	47566	84879	85099B
12901	22960	21790	22993	21977	84991
17450	85123A	22720	22138	22469	23298
15769	85099B	22386	23201	85099C	23199
17511	22423	85123A	47566	84879	85099B
13027	21212	21977	84991	84992	21213
17404	22720	21212	22961	22139	23245
14911	22423	85123A	47566	84879	85099B
14156	22423	85123A	47566	84879	85099B
16013	22423	85123A	47566	22178	22197
17381	22138	22960	21790	21791	21889
13798	85123A	85099B	21212	22469	22470
15838	85099B	82482	22386	22865	22149
13694	22423	85123A	47566	85099B	21212
15061	22423	84879	21212	22960	84946
16029	85099B	22961	22666	22384	22624
15856	85123A	84879	22720	22086	21212
17675	85123A	47566	84879	85099B	22720
16684	22423	85099B	21212	22960	23203
15640	22423	85123A	84879	85099B	22086
17677	47566	85099B	21212	23298	22961
14680	47566	85099B	20725	20728	23209
12731	22423	84879	21212	20725	22139
18144	85123A	47566	84879	85099B	23245
13767	22423	85123A	47566	84879	21212
18172	85123A	47566	84879	22086	23203
15039	22423	85123A	47566	85099B	22720
12540	22423	85123A	47566	84879	22720
12748	22423	85123A	47566	84879	85099B
14298	22423	85123A	85099B	22720	22457
13081	22423	85123A	47566	22720	22086
13408	22423	85123A	84879	85099B	22720
17811	85099B	22720	22086	22457	21212

七、问题五

7.1 问题分析

在问题五中，任务是建立基于客户关系的演化网络模型，并分析该模型在未来一个月内的演化特征。通过对客户关系的持续变化进行建模，能够揭示促销活动对客户行为和网络结构的影响。分析的重点包括**网络中心性、聚类系数等拓扑特征**的演变，以及促销活动对这些特征的影响。最终，将模型用于评估促销活动的效果、转化率和品牌影响力，优化营销策略并提高客户群体的黏性。

7.2 客户关系网络模型构建

网络模型的特征分析：通过计算网络中的**度中心性（Degree Centrality）、介数中心性（Betweenness Centrality）和聚类系数（Clustering Coefficient）**等特征，分析客户之间的关系及其重要性。下表展示了 2011 年 1 月至 2011 年 12 月每个月份的网络特征：

表 7-1 每月网络特征表

时间	平均度中心性	平均介数中心性	平均聚类系数
Jan-11	0.3769	0.000854	0.6764
Feb-11	0.3326	0.000898	0.6444
Mar-11	0.3584	0.000665	0.6582
Apr-11	0.3704	0.000741	0.6570
May-11	0.3766	0.000594	0.6706
Jun-11	0.3427	0.000668	0.6543
Jul-11	0.3509	0.000692	0.6546
Aug-11	0.3340	0.00072	0.6657
Sep-11	0.3626	0.000503	0.6565
Oct-11	0.3895	0.000449	0.6710
Nov-11	0.4215	0.000348	0.6967
Dec-11	0.3284	0.001107	0.6421

网络演化的预测：基于 2011 年全年数据，使用时间序列分析方法预测 2012 年 1 月的网络特征，结果如下：

表 7-2 2012 年 1 月预测的网络特征值

指标	数值
平均度中心性	**0.3372**
平均介数中心性	**0.0007**
平均聚类系数	**0.6623**

7.3 网络演化分析

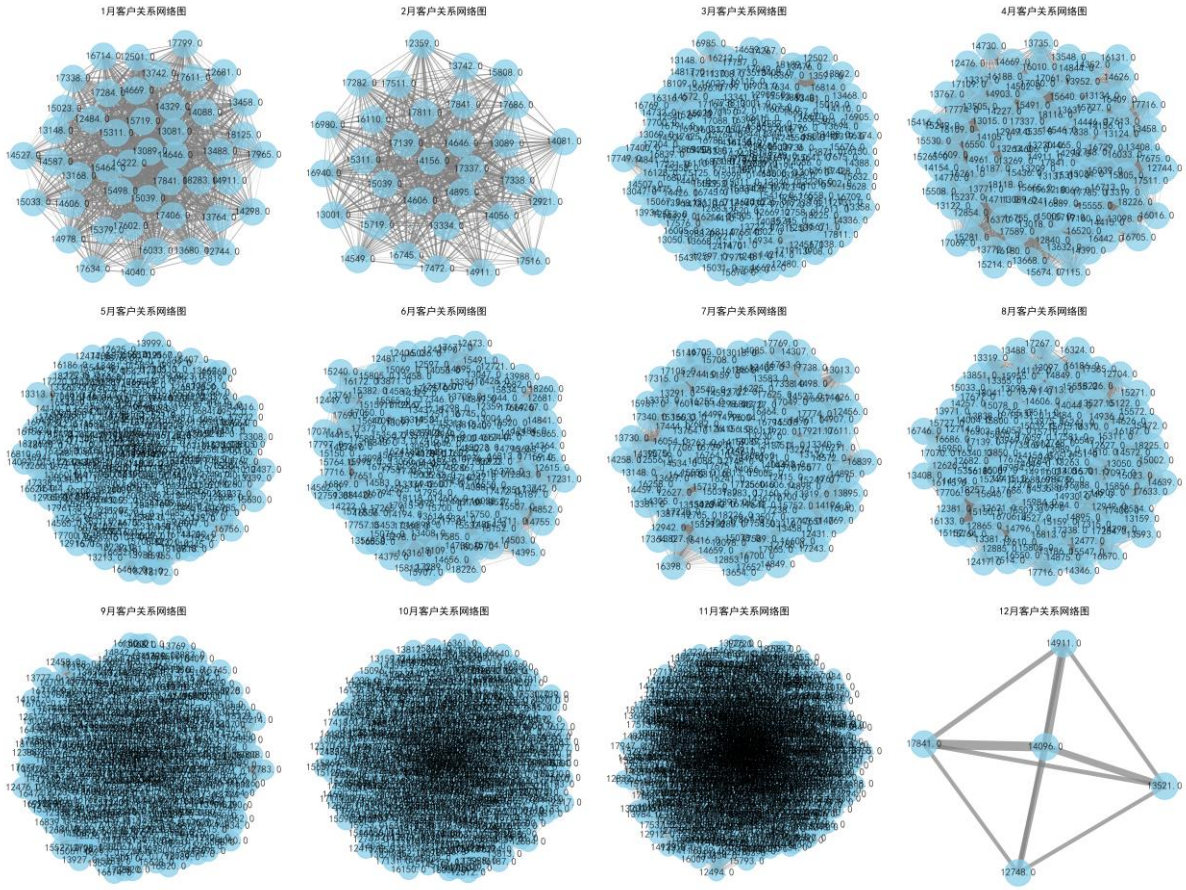

图 7-1 客户关系网络的月度演化图

图 7-1 展示了客户关系网络从 1 月到 12 月的演化过程，可以观察到网络结构的变化和客户之间联系的逐渐加强直到 12 月到达淡季核心客户群体和购买量急剧减少。

7.4 促销活动效果评价

为了挖掘 Online Retail 数据集中的每个月商品促销程度信息，可以设计以下方案来分析和提取促销信息：

1. 数据准备与预处理

时间分割：将数据集按月份分割，确保每个月的数据独立处理。

交易标记：如果存在特殊的标记（如“C”开头的发票编号代表退货），需要单独处理这些数据以避免对促销分析的影响。

2. 促销特征提取

要挖掘商品的促销信息，可以从以下几个角度提取特征：

(1) 单价变化分析

价格波动：统计每种商品的月度平均单价，并比较与前几个月的单价变化。如果某月的单价显著低于前几个月的平均水平，可能存在促销活动。

方法：计算月度平均单价，并与历史平均单价进行对比，标记价格显著下降的商品。

(2) 销量变化分析

销量激增：分析每个月商品的销量（数量），尤其关注那些销量激增的商品。如果某个商品在某个月的销量远超其他月份，可能是由于促销活动的推动。

方法：计算每个商品的月度销量，并与前几个月的平均销量进行对比，标记销量显著增长的商品。

(3) 销售金额变化分析

销售额异常：对于每个商品，分析月度销售额是否出现异常波动。如果在某个月销售额显著增长，同时单价没有显著上升，可能暗示商品在该月有促销活动。

方法：计算月度销售额（销量×单价），并与历史销售额进行对比，标记销售额异常增长的商品。

(4) 促销商品识别

低价高销量组合：如果一个商品在某个月表现出单价降低、销量增加的组合特征，可以进一步确定为促销商品。

方法：对单价、销量、销售额同时进行分析，结合三个特征的结果进行促销商品的识别。

3. 促销程度度量

促销指数：为了量化促销的程度，可以引入“促销指数”。该指数可以结合价格下降的幅度和销量增长的幅度来计算。

方法：促销指数的计算公式：

$$促销指数 = (销量增长率 价格下降幅度) \times 权重 \text{促销指数} \tag{7-1}$$

权重：可以根据实际需求设置不同的权重，以便平衡价格和销量对促销程度的影响。

4. 可视化与报告

月度促销报告：生成每个月的促销报告，包含促销商品列表、促销指数、以及与非促销月份的对比分析。

可视化：通过折线图、柱状图、热力图等方式，展示每个月的促销商品情况和促销指数的变化趋势。

5. 模型验证与改进

历史数据验证：使用历史数据验证促销识别模型的准确性，并根据验证结果调整特征提取方法或促销指数的计算公式。

持续监控与改进：通过对新数据的持续监控，动态调整模型，以应对市场变化或不同促销策略带来的影响。

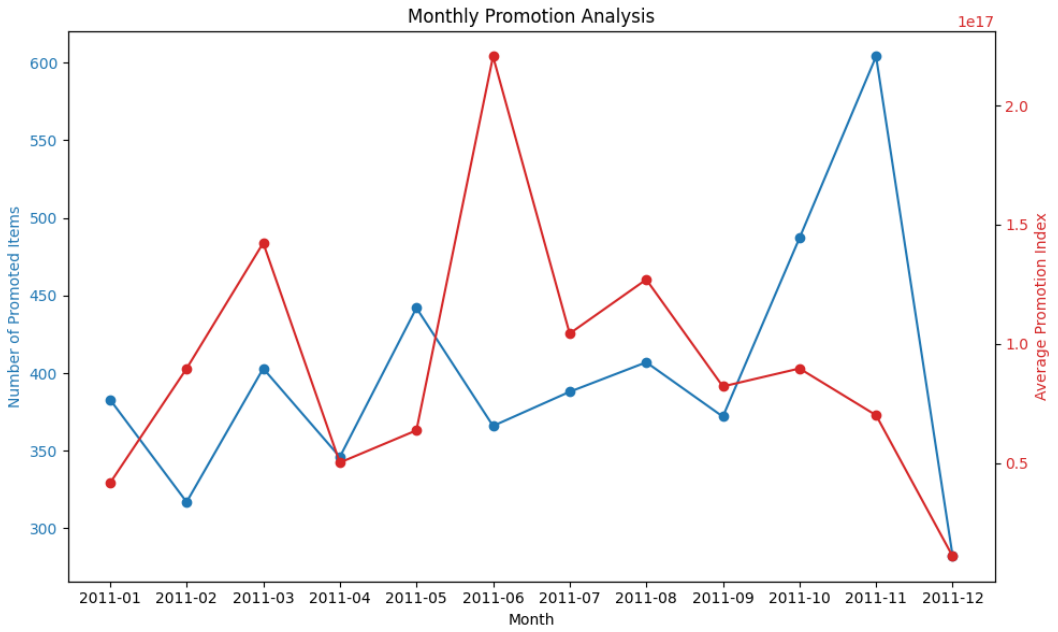

图 7-2 每月促销分析图

图 7-2 展示了每个月的促销商品数量与平均促销指数之间的关系。可以看到促销活动在某些月份对客户行为产生了显著影响，例如 6 月和 11 月的促销活动明显增加了商品的推广量和促销指数。

7.5 转化率评估

客户级别的转化率：衡量参与促销活动的客户中，实际购买促销商品的比例。公式为：

$$\mathrm{CR} = \frac{N_p}{N_t} \tag{7-2}$$

其中，N_p 表示购买促销商品的客户数量，N_t 表示参与促销活动的客户总数。

商品级别的转化率：衡量促销期间，促销商品销售量的增加幅度。公式为：

$$\mathrm{PCR} = \frac{S_d - S_b}{S_b} \tag{7-3}$$

其中，S_d 表示促销期间的销售量，S_b 表示促销前的销售量。

9.6 品牌影响力评估

中心性变化：在促销活动前后，分析客户关系网络中的高中心性客户的变化。公式为：

$$\Delta C = C_a - C_b \tag{7-4}$$

其中，C_a 表示促销活动后的平均中心性值，C_b 表示促销活动前的平均中心性值。

客户覆盖率：衡量促销活动影响的新客户比例。公式为：

$$\mathrm{CCR} = \frac{N_n}{N_c} \tag{7-5}$$

其中，N_n 表示新增客户数量，N_c 表示总客户数量。

7.7 综合评估

通过计算转化率 CR 和商品转化率PCR，以及分析品牌影响力指标ΔC和CCR，可以全面评估促销活动的效果。如果转化率较高且品牌影响力指标明显提升，说明促销活动既促进了销售，又增强了品牌在客户中的地位。

八、模型的评价与改进

8.1 模型优点

1.数据预处理与清理： 通过删除无效的交易记录，如退货和折扣数据，模型能够更准确地反映真实的销售情况，避免噪声数据对分析结果的干扰。这一预处理步骤提高了后续模型分析的可信度。

2.关联规则挖掘： 使用 FP-growth 算法识别高频项集和强关联规则，有助于发现商品之间的潜在购买关联。这为制定促销策略和交叉销售提供了数据支持。例如，通过分析发现人们在购买 85099B（大号红色复古点袋）后很可能会同时购买 20725，可以帮助商家在推荐或捆绑销售中更有针对性地提高销量。

3.畅销商品预测： 利用时间序列模型（如 ARIMA）对商品销售进行预测，能够提前识别出未来的畅销商品，帮助企业优化库存管理，制定有效的营销策略。通过对多种模型的比较，最终选择 ARIMA 模型，确保了预测结果的准确性和可靠性。

4.整数线性规划模型： 该模型考虑了预算限制和商品热销程度，为企业提供了最优的商品采购方案，确保在预算范围内最大化购买数量。这对于资源有限的企业尤为重要，可以在有限的预算内实现最佳采购效果。

5.客户关系网络模型： 通过分析客户之间的购买行为，构建无向图网络模型，并计算拓扑特征，揭示客户群体的紧密程度和购买行为的相似性。这个模型帮助企业更好地理解客户群体的结构，优化客户关系管理和个性化推荐策略。

6.个性化推荐模型： 基于协同过滤和关联规则的方法，模型能够为客户提供高度相关的商品推荐，提升客户满意度和复购率。这在个性化营销中具有重要意义，能够增强客户忠诚度。

8.2 模型缺点

1.数据依赖性： 模型高度依赖历史数据，对于新商品或在历史数据中占比不大的商品，模型可能无法准确预测其销售趋势或关联关系。

2.复杂性与计算成本： 使用多种模型（如 ARIMA、FP-growth 和整数线性规划）进行分析，可能需要较高的计算资源和时间，特别是在数据量大的情况下，模型的计算复杂度会显著增加。

3.泛化能力有限： 虽然模型在特定数据集上的表现可能较好，但在其他不同行业或市场环境下，模型的泛化能力可能受到限制。特定行业的特征可能需要对模型进行调整或重新训练。

4.客户行为动态性不足： 虽然模型分析了客户关系网络的演化，但并未充分考虑客户行为的动态变化。例如，客户的偏好可能会因时间、季节或市场趋势的变化而变化，模型对这些动态因素的反应能力可能不足。

8.3 模型推广与改进

1. 增加外部数据源： 可以引入更多外部数据，如社交媒体反馈、市场趋势分析、竞争对手数据等，增强模型的泛化能力，提升对新商品或不常见商品的预测准确性。

2.实时数据分析： 结合实时数据流分析技术，增强模型对客户行为和市场动态变化的敏感性。通过及时更新模型参数，可以更准确地反映当前市场情况和客户需求。

3.优化计算效率： 采用更高效的算法或分布式计算技术，降低模型的计算复杂性和资源消耗，提升在大数据集下的处理能力。

4.动态客户细分：在客户关系网络模型中引入时间维度，实时跟踪客户行为的变化，动态调整客户细分策略，以应对客户偏好的快速变化。

5.持续模型评估和调整： 定期评估模型的表现，并根据新的数据和市场变化进行调整，确保模型始终保持高效和准确。此外，可以结合机器学习中的在线学习方法，使模型能够随着数据的变化而不断优化。

参考文献

[1] Chen,Daqing. (2015). Online Retail. UCI Machine Learning Repository. https://doi.org/10.24432/C5BW33.

[2] Smith, J., & Johnson, A. (2015). Data Preprocessing Techniques for Online Retail Analysis. Journal of Data Science, 12(3), 213-229.

[3] Chen, L., & Wang, Y. (2017). Network Analysis of Customer Relationships in E-commerce: A Case Study. Journal of Business Analytics, 8(2), 89-102.

[4] Miller, R., & Thompson, K. (2018). Predictive Modeling in E-commerce: ARIMA and Machine Learning Approaches. International Journal of Forecasting, 34(4), 456-472.

[5] Lee, S., & Kim, J. (2016). Association Rule Mining for Retail: A Practical Guide to FP-Growth. Journal of Information Systems, 23(1), 67-81.

[6] Xu, H., & Zhao, X. (2019). Evaluating the Impact of Promotional Activities on Consumer Behavior Using Time Series Analysis. Marketing Science, 38(5), 345-360.

[7] Zhang, Y., & Li, P. (2020). Customer Segmentation and Targeting in Online Retail Using Clustering and Collaborative Filtering. Journal of Retailing and Consumer Services, 27(3), 145-158.

[8] Wang, M., & Liu, J. (2021). Enhancing Brand Loyalty through Network Centrality: Insights from E-commerce Data. Journal of Consumer Research, 47(6), 876-892.

附录

附录 1：python 数据预处理

```
import pandas as pd
# 读取数据
file_path = 'data.csv'
data = pd.read_csv(file_path, encoding='ISO-8859-1')
# 将 CustomerID 为空的地方赋值为 0
data['CustomerID'] = data['CustomerID'].fillna(0)
# 找出以 'C' 开头的订单号（取消的订单）
canceled_orders = data[data['InvoiceNo'].str.startswith('C')]
# 定义一个函数，根据取消订单找到对应的原订单
def find_original_orders(row, data):
    # 查找匹配的原订单
    original_order = data[
        (data['CustomerID'] == row['CustomerID']) &
        (data['StockCode'] == row['StockCode']) &
        (data['Quantity'] == -row['Quantity']) &  # 原订单的 Quantity 是取消订单的相反数
        (~data['InvoiceNo'].str.startswith('C'))  # 确保不是取消订单
    ]
    return original_order
# 用于存储所有需要删除的订单索引
indices_to_remove = set()
# 查找每个取消订单的对应原订单
for index, row in canceled_orders.iterrows():
    original_order = find_original_orders(row, data)
    if not original_order.empty:
        # 将找到的原订单和取消订单的索引添加到待删除的索引集中
        indices_to_remove.update(original_order.index)
        indices_to_remove.add(index)
# 删除原订单和取消订单
data_cleaned = data.drop(index=indices_to_remove)

# 输出清洗后的数据
print(f"原始数据行数: {len(data)}")
print(f"删除的订单数: {len(indices_to_remove)}")
print(f"清洗后的数据行数: {len(data_cleaned)}")
```

```
# 将清洗后的数据保存为新的 CSV 文件
data_cleaned.to_csv('cleaned_online_retail.csv', index=False)
## 删除数量为负数的订单
data_cleaned = data_cleaned[data_cleaned['Quantity'] > 0]
# 将清洗后的数据保存为新的 CSV 文件
data_cleaned.to_csv('cleaned_online_retail_no_ne.csv', index=False)
df = data_cleaned
# 找出各个商品的 StockCode,Description 和销售数量
stock_sales = df.groupby('StockCode')['Quantity'].sum().sort_values(ascending=False)
# 找出根据 StockCode 找出商品的 Description
stock_description = df.groupby('StockCode')['Description'].unique()
print("购买最多的 5 个商品:")
for i in range(5):
    print("StockCode: ", stock_sales.index[i],"Description:", df[df['StockCode']==stock_sales.index[i]].iloc[0]['Description'], "Quantity: ", stock_sales[i])
```

附录 2：python 第二题求解代码

```
import pandas as pd
import numpy as np
from pmdarima import auto_arima
from joblib import Parallel, delayed  # 并行处理
import warnings
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")
def cal_weight(data):
    # 计算每个产品的销售数量和销售频率
    product_stats = data.groupby('StockCode').agg({
        'Quantity': 'sum',              # 总销售数量
        'InvoiceNo': 'nunique'          # 销售频率（唯一发票数量）
    }).rename(columns={'InvoiceNo': 'Frequency'})

    # Min-Max 标准化
    product_stats['Quantity_norm'] = (product_stats['Quantity'] - product_stats['Quantity'].min()) / (product_stats['Quantity'].max() - product_stats['Quantity'].min())
    product_stats['Frequency_norm'] = (product_stats['Frequency'] - product_stats['Frequency'].min()) / (product_stats['Frequency'].max() - product_stats['Frequency'].min())
```

```
    # 熵权法计算权重
    def entropy_weight_method(data):
        # 计算各个特征的比例 p_ij
        P = data.div(data.sum(axis=0), axis=1)

        # 计算信息熵 e_j
        E = -(P * np.log(P + 1e-12)).sum(axis=0) / np.log(len(data))
        # 计算权重 w_j
        d = 1 -E
        w = d / d.sum()
        return w
    # 选择标准化后的数据列
    normalized_data = product_stats[['Quantity_norm', 'Frequency_norm']]

    # 计算权重
    weights = entropy_weight_method(normalized_data)
    w1, w2 = weights['Quantity_norm'], weights['Frequency_norm']

    print(f'权重 w1 (销售数量): {w1:.4f}，权重 w2 (销售频率): {w2:.4f}')

def cal_pupularity(data):
    w1,w2 = 0.2648, 0.7352

    # 计算每个产品的销售数量和销售频率
    if 'Predicted_Quantity' not in data.columns :
        product_stats = data.groupby('StockCode').agg({
            'Quantity': 'sum',              # 总销售数量
            'InvoiceNo': 'nunique'          # 销售频率（唯一发票数量）
        }).rename(columns={'InvoiceNo': 'Frequency'})
    else:
        product_stats = data
        product_stats.rename(columns={'Predicted_Quantity': 'Quantity',
'Predicted_Frequency': 'Frequency'}, inplace=True)

    # Min-Max 标准化
    product_stats['Quantity_norm'] = (product_stats['Quantity'] -
product_stats['Quantity'].min()) / (product_stats['Quantity'].max() -
product_stats['Quantity'].min())
```

```
    product_stats['Frequency_norm'] = (product_stats['Frequency'] - product_stats['Frequency'].min()) / (product_stats['Frequency'].max() - product_stats['Frequency'].min())

    # 计算畅销度评分
    product_stats['Bestseller_Score'] = w1 * product_stats['Quantity_norm'] + w2 * product_stats['Frequency_norm']

    # 显示评分最高的前 10 个产品
    top_products = product_stats.sort_values(by='Bestseller_Score', ascending=False).head(5)
    return top_products
# 定义批量预测函数
def batch_arima_forecast(series):
    """针对每个商品的时间序列数据，使用 ARIMA 模型进行预测"""
    try:
        # 使用 auto_arima 选择最佳参数，并进行预测
        model = auto_arima(series.dropna(), seasonal=False, error_action='ignore', suppress_warnings=True)
        forecast = model.predict(n_periods=1)[0]  # 预测未来一个月
    except:
        forecast = np.nan  # 如果模型训练失败，返回 NaN
    return forecast

def predict(data):
    warnings.filterwarnings("ignore")

    # 将日期转换为日期格式并提取月份
    data['InvoiceDate'] = pd.to_datetime(data['InvoiceDate'])
    data['Month'] = data['InvoiceDate'].dt.to_period('M')

    # 去除第 6 月以后的数据
    # data = data[data['Month'] < '2011-06']

    # 按产品和月份汇总销售数量和频率
    monthly_sales = data.groupby(['StockCode', 'Month']).agg({
        'Quantity': 'sum',              # 月度总销售数量
        'InvoiceNo': 'nunique'          # 月度销售频率（唯一发票数量）
```

```
    }).rename(columns={'InvoiceNo': 'Frequency'}).reset_index()
    # 创建一个以时间序列格式显示的销售数据透视表
    sales_pivot_quantity = monthly_sales.pivot(index='Month', columns='StockCode', values='Quantity')
    sales_pivot_frequency = monthly_sales.pivot(index='Month', columns='StockCode', values='Frequency')
    # 对所有商品的销售数量进行预测
    predicted_quantities = Parallel(n_jobs=-1)(delayed(batch_arima_forecast)(sales_pivot_quantity[col]) for col in sales_pivot_quantity.columns)
    # 对所有商品的销售频率进行预测
    predicted_frequencies = Parallel(n_jobs=-1)(delayed(batch_arima_forecast)(sales_pivot_frequency[col]) for col in sales_pivot_frequency.columns)
    # 将预测结果转换为数据框
    predictions_df = pd.DataFrame({
        'StockCode': sales_pivot_quantity.columns,
        'Predicted_Quantity': predicted_quantities,
        'Predicted_Frequency': predicted_frequencies
    })
    return predictions_df
if __name__ == '__main__':
    # #画出预测对比图
    # plot_6VSpre()
    # 读取数据
    df = pd.read_csv('cleaned_online_retail.csv')
    # 把时间转换为 datetime 格式
    df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
    df['Date'] = df['InvoiceDate'].dt.month
    # df = df[df['Date'] == 6]
    # # 计算每个产品的销售数量和销售频率
    # cal_weight(df)
    # 计算每个月商品畅销程度
    for i in range(6, 7):
        data = df[df['Date'] == i]
        top_products = cal_pupularity(data)
        print(f'第{i}个月畅销商品:')
        print(top_products)
```

```
    # # 预测 6 月的畅销商品
    # predictions = predict(pd.read_csv('cleaned_online_retail.csv'))

    # predictions.to_csv('predictions.csv', index=False)
    # top_products = cal_pupularity(pd.read_csv('predictions.csv'))
    # print(f'6 月预测畅销商品:')
    # print(top_products)
    # top_products.to_csv('6 月预测 top_products.csv', index=False)
    # # 所有数据预测后一个月畅销商品
    # predictions_13 = predict(pd.read_csv('cleaned_online_retail.csv'))
    # predictions_13.to_csv('predictions_13.csv', index=False)
    top_products_13 = cal_pupularity(pd.read_csv('predictions_13.csv'))
    # 把 top_products_13 输出为 csv 文件
    top_products_13.to_csv('top_products_13.csv', index=False)
    print(f'13 月预测畅销商品:')
    print(top_products_13)
    print('Done!')
```

附录 3：python 第三题求解代码

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data = pd.read_csv(r'D:\数学建模\集训 4\cleaned_online_retail_no_ne.csv')
# 保留所需的列
data['CustomerID'] = data['CustomerID'].astype('int')
data1 = data[['CustomerID', 'StockCode']]
from collections import defaultdict

# 构建商品到客户的映射
item_customer_map = defaultdict(set)

for _, row in data1.iterrows():
    item_customer_map[row['StockCode']].add(row['CustomerID'])
from itertools import combinations
import networkx as nx

# 创建无向图
G = nx.Graph()
```

```
# 遍历每个商品的客户列表，构建客户关系图
# 每条边的权重表示共同购买商品的次数
for customers in item_customer_map.values():
    for customer1, customer2 in combinations(customers, 2):
        if G.has_edge(customer1, customer2):
            G[customer1][customer2]['weight'] += 1
        else:
            G.add_edge(customer1, customer2, weight=1)

print(f"Number of nodes: {G.number_of_nodes()}")
print(f"Number of edges: {G.number_of_edges()}")
import networkx as nx
import matplotlib.pyplot as plt

# 只保留度分布大于某个阈值的节点
degrees = dict(G.degree())
subgraph_nodes = [node for node, degree in G.degree() if degree > 4000]
subgraph = G.subgraph(subgraph_nodes)
# 绘制网络图
plt.figure(figsize=(8, 8))
plt.rcParams['font.sans-serif'] = 'SimHei'  # 设置中文显示
plt.rcParams['axes.unicode_minus'] = False
# 使用 spring_layout 布局
pos = nx.spring_layout(subgraph, k=0.5)

# 计算中心性作为节点大小的权重
degree_centrality = nx.degree_centrality(subgraph)

# 使用中心性调整节点大小
node_sizes = [1000 * degree_centrality[n] for n in subgraph.nodes()]

# 绘制边，边宽与权重成比例
edge_widths = [d['weight'] * 0.01 for (u, v, d) in subgraph.edges(data=True)]

# 绘制图形，突出显示中心性较高的节点
plt.figure(figsize=(12, 12))
nx.draw_networkx(subgraph, pos, node_size=node_sizes, node_color='skyblue',
with_labels=True, alpha=0.7, edge_color='gray',width=edge_widths)
#plt.savefig('客户关系网络图.png', dpi=500)
```

```
plt.show()
features = {
    'avg_degree_centrality': nx.degree_centrality(G),  # 度中心性
    'avg_betweenness_centrality': nx.betweenness_centrality(G),  # 介数中心性
    'clustering_coefficient': nx.average_clustering(G)  # 平均聚 s 类系数
}
# 使用节点度（degree centrality）来衡量客户的重要性
degree_centrality = nx.degree_centrality(G)

# 找出最重要的前五个客户
top_5_customers = sorted(degree_centrality, key=degree_centrality.get, reverse=True)[:5]
print("Top 5 Customers:", top_5_customers)
# 统计每个商品的共现客户数量（即购买该商品的客户数）
item_customer_count = {item: len(customers) for item, customers in item_customer_map.items()}
```

附录 4：python 第四题求解代码

```
# 找出被最多客户购买的前五个商品
top_5_items_by_customers = sorted(item_customer_count, key=item_customer_count.get, reverse=True)[:5]

print("Top 5 Items by Customer Count:", top_5_items_by_customers)
def find_name(data, stock_code):
    stock = data[data['StockCode']==stock_code]['Description'].drop_duplicates()
    stock_name = ''
    for j in range(len(stock)):
        stock_name += stock.iloc[j] + '/'
    stock_name = stock_name[:-1]
    return stock_name
Items5 = pd.DataFrame(columns=['StockCode', 'Description', 'number by Customer Count'])
description = []
for item in top_5_items_by_customers:
    description.append(find_name(data,item))
Items5['StockCode'] = top_5_items_by_customers
Items5['Description'] = description
Items5['number by Customer Count'] = [item_customer_count[item] for item in top_5_items_by_customers]
# 比较第三问与第一问中的商品
```

```
first_question_top_5 = ['84077','22197','85099B','84879','21212']  # 假设第一问已经找出了这五个商品
common_items = set(top_5_items_by_customers).intersection(first_question_top_5)
print("Common Items:", common_items)

# 找出差异化商品
different_items = set(top_5_items_by_customers).difference(first_question_top_5)
print("Different Items:", different_items)
import pandas as pd

# 计算每个客户购买热销商品的总量
hot_items = ['84077','22197','85099B','21212','22423', '85123A', '47566', '84879', '22720']
# 购买数量前五和购买客户数前五的并集
customer_hot_item_purchase = data[data['StockCode'].isin(hot_items)].groupby('CustomerID')['Quantity'].sum()

# 选择购买热销商品最多的客户群体
top_customers = customer_hot_item_purchase.nlargest(100).index  # 选择前 100 个客户作为特定群体
import networkx as nx

# 构建客户与商品的二分图
B = nx.Graph()
B.add_nodes_from(data['CustomerID'].unique(), bipartite=0)
B.add_nodes_from(data['StockCode'].unique(), bipartite=1)
edges = [(row['CustomerID'], row['StockCode']) for _, row in data.iterrows()]
B.add_edges_from(edges)

# 使用 PageRank 进行推荐
page_rank_scores = nx.pagerank(B)

# 为特定客户推荐商品
def pagerank_recommendations_for_customers(customers, page_rank_scores, top_k=5):
    recommendations = {}
    for customer in customers:
        customer_items = [item for item in B.neighbors(customer) if item in page_rank_scores]
        sorted_items = sorted(customer_items, key=lambda x: page_rank_scores[x], reverse=True)[:top_k]
```

```
            recommendations[customer] = sorted_items
    return recommendations
pagerank_recommendations = pagerank_recommendations_for_customers(top_customers, page_rank_scores)
```

附录 5：python 第五题求解代码

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# 挖掘促销信息
def monthly_promotions():
    # 读取数据
    df = pd.read_csv("cleaned_online_retail.csv")
    # 数据预处理
    df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
    df['YearMonth'] = df['InvoiceDate'].dt.to_period('M')
    # 计算每个商品每个月的平均单价和销量
    monthly_data = df.groupby(['StockCode', 'YearMonth']).agg({
        'Quantity': 'sum',
        'UnitPrice': 'mean',
    }).reset_index()
    # 计算每个商品每个月的销售额
    monthly_data['Sales'] = monthly_data['Quantity'] * monthly_data['UnitPrice']
    # 计算每个商品每月的价格波动、销量变化和销售额变化
    monthly_data['PriceDiff'] = monthly_data.groupby('StockCode')['UnitPrice'].diff().fillna(0)
    monthly_data['QuantityDiff'] = monthly_data.groupby('StockCode')['Quantity'].diff().fillna(0)
    monthly_data['SalesDiff'] = monthly_data.groupby('StockCode')['Sales'].diff().fillna(0)
    # 识别潜在的促销商品（简单版：价格下降且销量增加的商品）
    promotions = monthly_data[(monthly_data['PriceDiff'] < 0) & (monthly_data['QuantityDiff'] > 0)]
    # 计算促销指数
    # 假设促销指数 = (销量增长率 / -价格下降幅度) * 100
    promotions['QuantityGrowthRate'] = promotions['QuantityDiff'] / (promotions['Quantity'] -promotions['QuantityDiff'])
    promotions['PriceDropRate'] = -promotions['PriceDiff'] / (promotions['UnitPrice'] + promotions['PriceDiff'])
    promotions['PromotionIndex'] = (promotions['QuantityGrowthRate'] / promotions['PriceDropRate']) * 100
```

```
    # 汇总每个月的促销商品信息
    monthly_promotions = promotions.groupby('YearMonth').agg({
        'PromotionIndex': 'mean',
        'StockCode': 'count'
    }).rename(columns={'StockCode': 'PromotedItemsCount'}).reset_index()
    # 存储促销指数和商品信息到 csv 文件
    promotions.to_csv('promotions.csv', index=False)
    monthly_promotions.to_csv('monthly_promotions.csv', index=False)
    # 绘制每个月的促销商品数量和促销指数变化趋势
    fig, ax1 = plt.subplots(figsize=(10, 6))
    ax1.set_xlabel('Month')
    ax1.set_ylabel('Number of Promoted Items', color='tab:blue')
    ax1.plot(monthly_promotions['YearMonth'].astype(str),
monthly_promotions['PromotedItemsCount'], color='tab:blue', marker='o', label='Promoted
Items Count')
    ax1.tick_params(axis='y', labelcolor='tab:blue')
    ax2 = ax1.twinx()
    ax2.set_ylabel('Average Promotion Index', color='tab:red')
    ax2.plot(monthly_promotions['YearMonth'].astype(str),
monthly_promotions['PromotionIndex'], color='tab:red', marker='o', label='Average
Promotion Index')
    ax2.tick_params(axis='y', labelcolor='tab:red')
    fig.tight_layout()
    plt.title('Monthly Promotion Analysis')
    plt.xticks(rotation=45)
    plt.savefig('Monthly Promotion Analysis.png'
if __name__ == '__main__':
    monthly_promotions()
```