基于 XGBoost 与非线性规划的蔬菜类商品定价与补货策略模型

摘要

生鲜食品由于其短暂的保鲜期，大部分品种无法隔日售卖，这决定了生鲜商超需要每日进行补货。本文主要研究了生鲜商超中蔬菜类商品的销售规律及相互关系，并研究了销售总量与定价方式间的潜在关系；在满足约束条件的前提下，制订了使商超收益最大化的补货与定价策略；最后对继续优化补货与定价策略提出了若干种可能的方式。这对于经常面临补货和定价抉择的蔬菜商家来说具有着重要的现实意义。

首先，针对于给定数据进行一定的初步分析和**预处理**。重新编号进行数据转换来方便后续处理，对部分异常值进行处理，对缺失值进行处理。

针对问题一：本问是考察对于一个统计和相关性分析的数据模型。对于销售量的分布规律，本文分别从**空间和时间上**进行分析，空间上从品类和同类单品上统计占比并进行可视化分析，时间上采用不同的尺度进行可视化分析并**得出了多条与现实相符合的结论**，包括销售组合、季节影响、疫情等多种情况。对于销售量的相互关系，不同品类之间采用 **Spearman 相关性分析**，不同单品之间采用**层次聚类**并且利用 Spearman 对聚类结果进行了验证。

针对问题二：本问考察的是一个基于给定数据进行预测然后再进行数学建模的非线性规划问题。本文首先采用了**季节性 ARIMA 时间序列分析**，在周月年三个不同时间尺度上进行批发价的预测。与此同时，**结合关联性分析和 XGBoost 回归分析对销售总量和成本加成定价的关系进行拟合**。基于以上的数据与关系，**本问建立了一个综合考虑成本定价、供需关系等多种因素的目标规划模型**，最后通过 Matlab 求解得到了题目所要求的补货总量和定价策略。

针对问题三：本问考察的是在问题二基础上进一步考虑可销售商品总数和单品最小订购量的非线性规划模型。本文首先对 6 月 24-30 日的可售单品进行排序并筛选，获得 33 种候选单品。与问题二类似，通过 **ARIMA 时间序列分析**对 7 月 1 日各单品的批发价进行预测，并训练 **XGBoost 回归模型**获得利润率与各单品销售量的关系。根据以上数据，本问通过 **0-1 变量**建立了考虑新增约束条件并且能够对候选单品进行筛选的非线性规划模型，并通过**粒子群算法（PSO）**进行了模型求解。**求解结果表明，当选取 33 种候选单品，并进行特定定价与补货决策时，商超的总收益能达到最大值 2948.18 元**。

针对问题四：本问考察的是对蔬菜商品的补货和定价决策的**影响因素**的理解。根据前三问建立的模型以及求解结果，为了使模型更加严谨，本文考虑了更多影响因素，可以从**供给侧和需求侧**两大维度进行分析，提出了供应链可靠性、竞争市场、地理位置等因素对供需关系的影响。

关键字：定价与补货　规划问题　Spearman　XGBoost　时间序列分析　粒子群算法

一、 问题重述

1.1 问题的背景及意义

近年来，随着经济社会发展人民对于生活品质的要求逐渐提高，其中也表现在对于食品品质的要求提高上。蔬菜类生鲜食品由于其营养价值和新鲜程度受到广大消费者的追捧。我国蔬菜类生鲜产品的产量和销量近年来稳步增长，在世界范围内均位于前列。然而，蔬菜类生鲜产品的保鲜期普遍较短，在存储、运输、销售过程中的变质损耗较多，这对生鲜商超的定价与补货策略产生了挑战。生鲜商超需要根据各商品的历史销售量进行未来销售量的预测，并进行相应的补货和定价。因此对于蔬菜类商品的自动定价与补货策略的研究具有重要意义，具体如下。

- 优化供应链管理：研究生鲜商超的补货问题涉及到货物的采购、库存管理和物流配送等方面。通过研究如何更好地补货，商超可以优化供应链管理，降低库存成本，减少食品过期损失，提高资金周转率。
- 适应市场变化：食品市场条件不断变化，包括需求侧和供给侧的季节性变化等。研究补货和定价问题可以帮助商超更灵活地调整策略，以适应市场变化，降低风险。
- 增加竞争力：市场竞争激烈，研究生鲜商超需要不断创新和改进以保持竞争力。通过深入研究补货和定价策略，商超可以找到差异化的竞争优势，吸引更多顾客。
- 减少资源浪费：不合理的补货和定价策略可能导致资源浪费，包括食品浪费和不必要的成本增加。研究如何精确补货和定价可以减少这些浪费，对环境和经济都有积极影响。

1.2 需要解决的问题

在生鲜商超中，一般蔬菜类商品的保鲜期都比较短，且品相随销售时间的增加而变差，大部分品种如当日未售出，隔日就无法再售。因此，商超通常会根据各商品的历史销售和需求情况每天进行补货。

由于商超销售的蔬菜品种众多、产地不尽相同，而蔬菜的进货交易时间通常在凌晨3:00- 4:00，为此商家须在不确切知道具体单品和进货价格的情况下，做出当日各蔬菜品类的补货决策。蔬菜的定价一般采用“成本加成定价”方法，商超对运损和品相变差的商品通常进行打折销售。可靠的市场需求分析，对补货决策和定价决策尤为重要。从需求侧来看，蔬菜类商品的销售量与时间往往存在一定的关联关系；从供给侧来看，蔬菜的供应品种在 4 月至 10 月较为丰富，商超销售空间的限制使得合理的销售组合变得极为重要。

附件 1 给出了某商超经销的 6 个蔬菜品类的商品信息；附件 2 和附件 3 分别给出了该商超 2020 年 7 月 1 日至 2023 年 6 月 30 日各商品的销售流水明细与批发价格的相关数据；附件 4 给出了各商品近期的损耗率数据。请根据附件和实际情况建立数学模型解决以下问题：

1. **问题**一：蔬菜类商品不同品类或不同单品之间可能存在一定的关联关系，请分析蔬菜各品类及单品销售量的分布规律及相互关系。
2. **问题**二：考虑商超以品类为单位做补货计划，请分析各蔬菜品类的销售总量与成本加成定价的关系，并给出各蔬菜品类未来一周 (2023 年 7 月 1-7 日) 的日补货总量和定价策略，使得商超收益最大。
3. **问题**三：因蔬菜类商品的销售空间有限，商超希望进一步制定单品的补货计划，要求可售单品总数控制在 27-33 个，且各单品订购量满足最小陈列量 2.5 千克的要求。根据 2023 年 6 月 24-30 日的可售品种，给出 7 月 1 日的单品补货量和定价策略，在尽量满足市场对各品类蔬菜商品需求的前提下，使得商超收益最大。
4. **问题四**：为了更好地制定蔬菜商品的补货和定价决策，商超还需要采集哪些相关数据，这些数据对解决上述问题有何帮助，请给出你们的意见和理由

二、 问题分析

2.1 问题一分析

针对问题一，需要分析蔬菜各品类和单品销售量的分布规律和相互关系。对于分布规律，本文分别从空间上和时间上进行描述性分析：空间上，统计各品类和单品历史上的总销售量，分析总销量中各品类的占比和各品类中不同单品的占比，以饼状图和小提琴图的形式进行表示，得到销售量占比最多的品类和单品；时间上，分别以天、月、季度为单位，统计了各品类销售量随时间变化的折线图，进而反映销售量的时间规律。对于相互关系，本文从品类和单品两个层次进行分析：品类间，本文选择 Spearman 相关系数进行相关性分析；单品间，本文利用层次聚类反映相似程度，进而反映相关性，并利用 Spearman 相关系数对聚类结果进行验证。

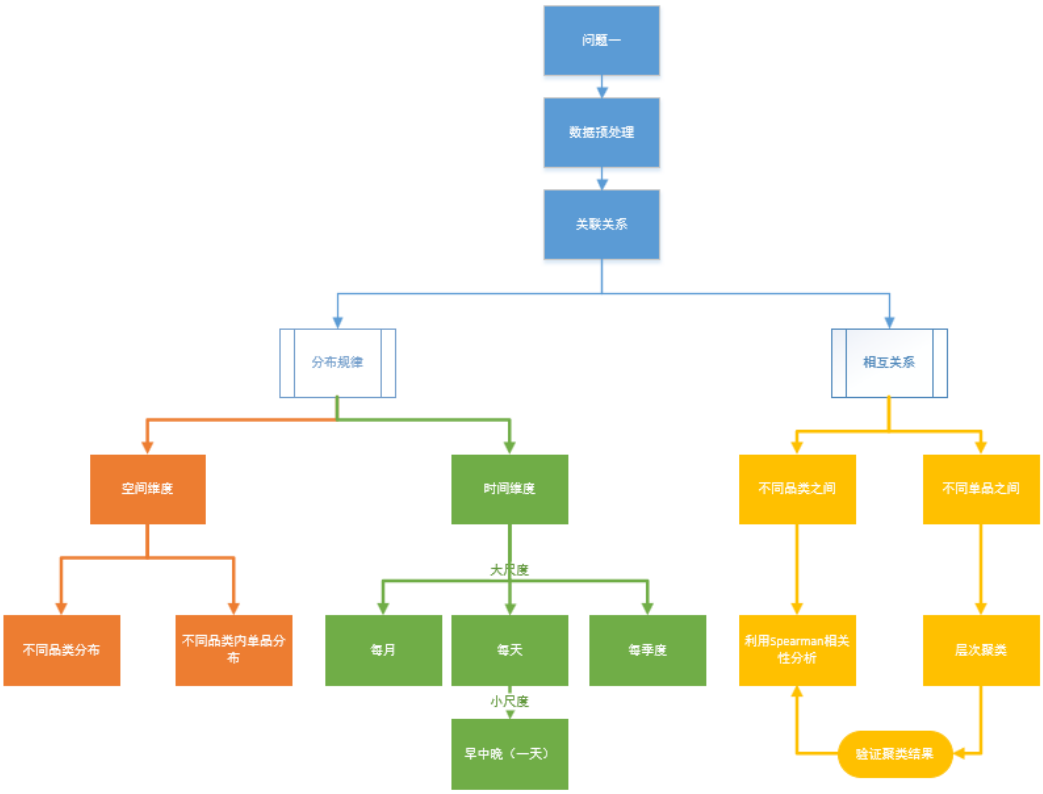


图 1　问题一思路流程图

2.2 问题二分析

针对问题二，为了以品类为单位做补货计划，本文将该问题划分为三个步骤：首先，由于价格变动影响需求变化，利润率和销售量之间有着紧密的关系，因此本文通过 Spearman 关联性分析，得到销售量、利润率、成本价、售价之间的相关系数矩阵，为后续的补货和定价计划提供历史规律的基础；然后，为了精准地把控在未来七天内的各类的批发价，本文提出季节性 ARIMA 算法，通过以长、中、短三个时间尺度的历史批发价即成本数据为训练数据，预测出未来一周的批发价即成本；在具备了以上数据后，由于为了能够更好地拟合现实情况，并结合本题中所隐含的多种限制条件，建立目标规划模型。并且综合考虑利润最大化、成本加成定价模型、供需关系、库存限制、仓储成本、商品退货情况、销售空间、顾客最大购买力等因素，其中成本加成定价部分使用 XGBoost 模型进行拟合销售量与利润率之间的数量关系。最后利用 Matlab 求解出未来一周内各类的补货总量以及定价策略。

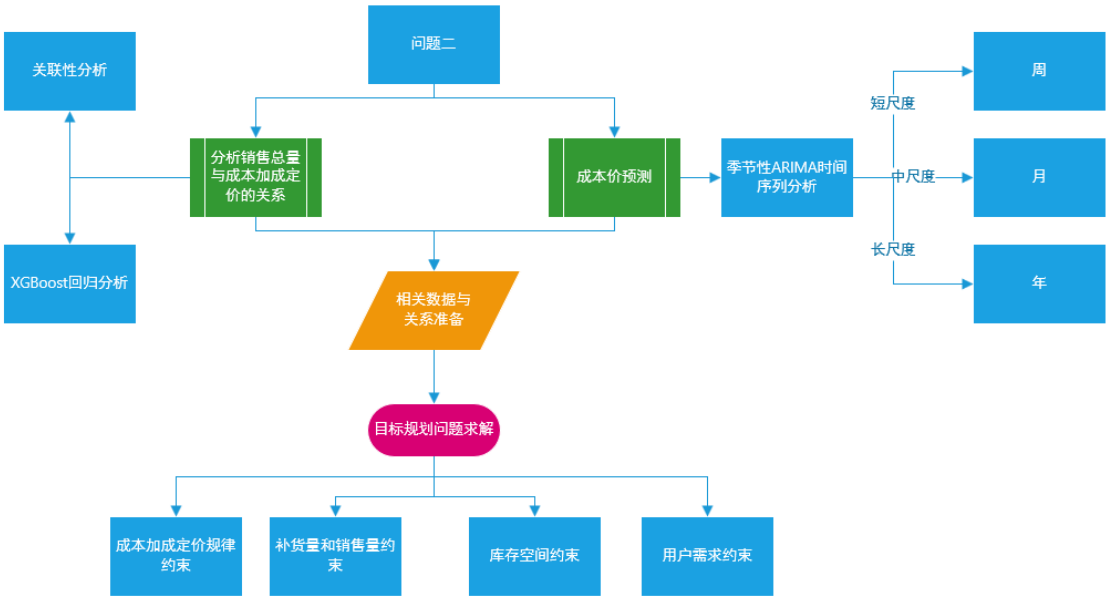


图 2　问题二思路流程图

2.3 问题三分析

针对问题三，在问题二的基础上增加了可售单品总数和各单品最小订购量的约束条件。由于约束条件的增多，各约束条件的优先级和权重难以衡量，所以本文建立了非线性规划模型，引入了 0-1 变量对不同单品进行选择。在模型建立方面，与问题二类似，通过 XGBoost 回归模型建立利润率和销售量间的关系，并通过 ARIMA 算法，对 7 月 1 日的批发价进行预测。在模型求解方面，本文首先将 0-1 变量进行等效替换，接着采用粒子群算法（Particle Swarm Optimization，PSO）对非线性模型进行求解。

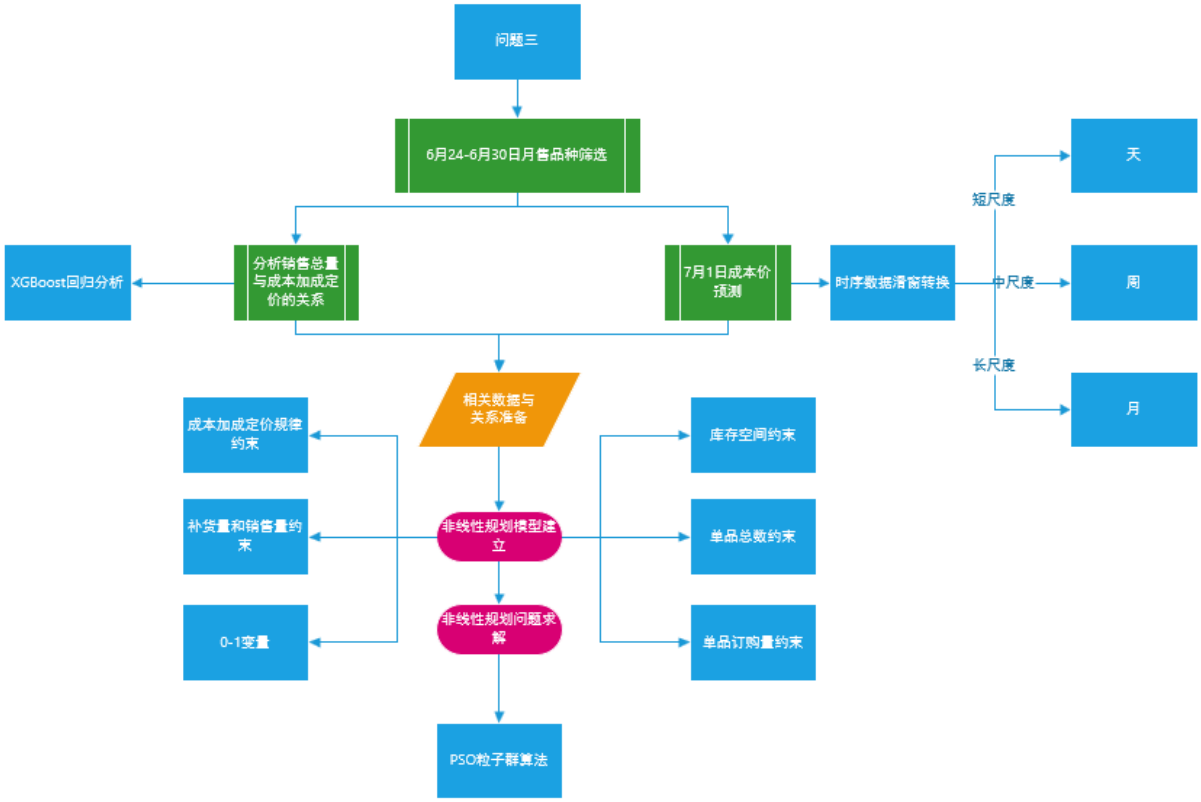


图 3　问题三思路流程图

2.4 问题四分析

针对问题四，本文首先考虑了问题二和问题三所建立的模型的缺点，为了使模型更加严谨、结果更加准确，本文对影响商超补货和定价决策进行了分析概括，整体来说，从供给侧和需求侧两大维度进行归纳：

从供给侧的角度出发，商超可以考虑采集以下数据：

- **供应链数据**：包括蔬菜种植地点、种植季节、种植数量、农药使用情况等。这些数据可以帮助商超了解供应链的稳定性和可靠性，预测蔬菜的供应量和质量，并及时调整补货计划。
- **生产技术数据**：可以了解农业技术的应用情况，例如温室种植、水培种植、有机种植等。这些数据可以帮助商超评估蔬菜的生长环境和种植方式对产量和质量的影响，从而更好地选择可靠的供应商和品种。

从需求侧的角度出发，商超可以考虑采集以下数据：

- 市场调研数据：商超可以进行市场调研，了解竞争对手的定价策略、蔬菜需求的趋势等。这些数据可以帮助商超制定有竞争力的定价策略，提高销售额和市场份额。

以上数据可以帮助商家更好地进行供应链优化、需求预测，优化补货和定价策略，从而提高蔬菜商品的销售效益和竞争力。

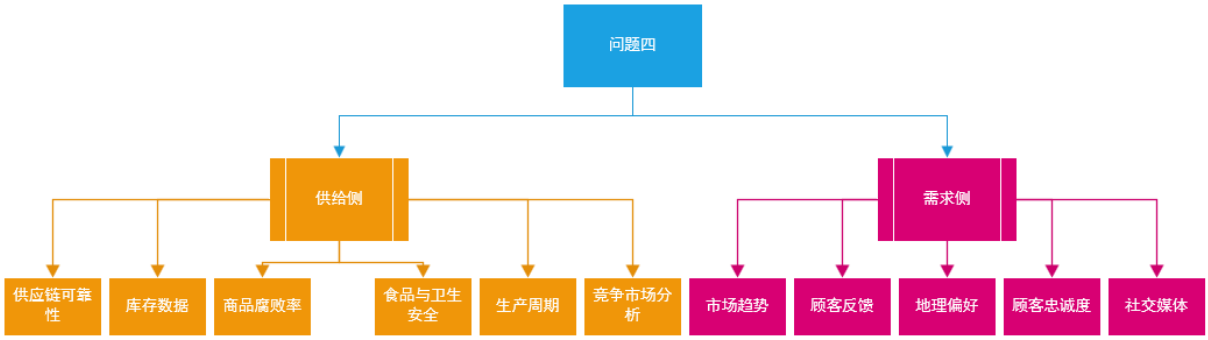


图 4　问题四思路流程图

三、 模型假设

1. 假设损耗率只代表运输过程的损耗，不考虑在一天或者一周内随时间变化的腐烂变质以及品相变差导致的损耗。
2. 假设问题二和问题三中的退货可以使用平均每 kg 的退货率衡量，并且退货是全额退货。
3. 假设问题二和问题三中不同单品或品类销售量和利润率的关系是一致的。
4. 假设问题二中可以使用历年的最高收益来衡量规划的收益目标最低值。
5. 假设问题二中可以使用历年的销售量最大值代表顾客的最大购买能力和消费水平以及菜类商品的最大销售容量。
6. 假设问题二中由于品类的量较大，运输过程的损耗可以忽略不计。
7. 假设问题二中的各品类的批发价格保持稳定，不发生突变。
8. 假设 7.1-7.7 的销售与进货不受其他意外的自然、社会因素的影响。

四、 符号说明

符号	说明
$P_{1,2,3}$	优先因子
$d_1^-, d_2^-, d_2^+, d_3^+$	偏差变量
S_{ji}	第 i 类蔬菜在第 j 天的销售量（kg）
S_i^*	第 i 种蔬菜在 7 月 1 日的销售量（kg）
q_i	第 i 类蔬菜的平均退货率
q_i^*	第 i 种蔬菜的平均退货率
T_{ji}	第 i 类蔬菜在第 j 天的定价（元）
T_i^*	第 i 种蔬菜在 7 月 1 日的定价（元）
X_{ji}	第 i 类蔬菜在第 j 天的补货量（kg）
X_{ji}^*	第 i 种蔬菜在 7 月 1 日的补货量（kg）
b_i	第 i 类蔬菜的平均批发价（元）
b_i^*	第 i 种蔬菜的批发价（元）
H_{ji}	第 i 类蔬菜历史最销售量（kg）
l_i	第 i 种蔬菜的损耗率
n_{ji}	第 i 类蔬菜在第 j 天的利润率

五、 数据预处理

5.1 数据转换

由于附件中单品编码和分类编码较长，不利于后续数据处理与分析，因此对单品及分类进行重新编号，如表1和表2所示。

表 1　单品重新编码（详见附件）

原单品编码	单品名称	新单品编码
102900005115168	牛首生菜	1
102900005115199	四川红香椿	2
102900005115625	本地小毛白菜	3
102900005115748	白菜苔	4
102900005115762	苋菜	5
.....		

表 2　分类重新编码

分类编码	分类名称	新分类编码
1011010101	花叶类	1
1011010201	花菜类	2
1011010402	水生根茎类	3
1011010501	茄类	4
1011010504	辣椒类	5
1011010801	食用菌	6

另外，为方便后续数据分析，将附件二中“销售类型”和“是否打折销售”进行转换。“销售类型”为“销售”的替换为“0”，为“退货”的替换为“1”；“是否打折销售”为“否”的替换为“0”，为“是”的替换为“1”。

5.2 异常值处理

在附件 3 中，本文发现批发价格存在低于 0.1 元的情况，如表3。在全国农产品批发市场价格信息系统进行查询发现，并没有一种蔬菜商品的价格低于 0.1 元。因此，本文认为批发价低于 0.1 元的数据属于异常值，并进行剔除。

表 3　部分批发价低于 0.1 元的数据

日期	单品编码	批发价格 (元/千克)
2022-08-24	102900011021842	0.01
2022-08-25	102900005115823	0.01
2022-08-25	102900011021842	0.01
2022-08-26	102900011021842	0.01
.....		

5.3 缺失值处理

本文发现在附件 2 中，有部分附件 1 的蔬菜单品并未出现，缺失单品如表4所示。缺失原因可能是数据来源的生鲜商超并不售卖这些单品，在此后的数据分析与处理中，对这些单品进行忽略。

表 4　缺失单品

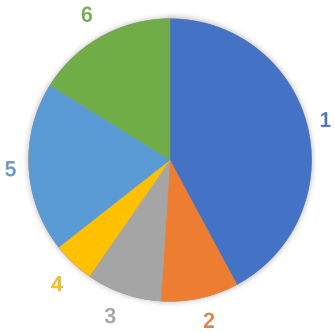
原单品编码	单品名称	新单品编码	原分类编码	分类名称	新分类编码
102900005116776	本地菠菜	21	1011010101	花叶类	1
102900005116042	藕	106	1011010402	水生根茎类	3
102900011023648	芜湖青椒 (2)	151	1011010504	辣椒类	5
102900011032145	芜湖青椒 (份)	164	1011010504	辣椒类	5
102900011011782	虫草花 (盒)(1)	199	1011010801	食用菌	6

六、 问题一的模型建立与求解

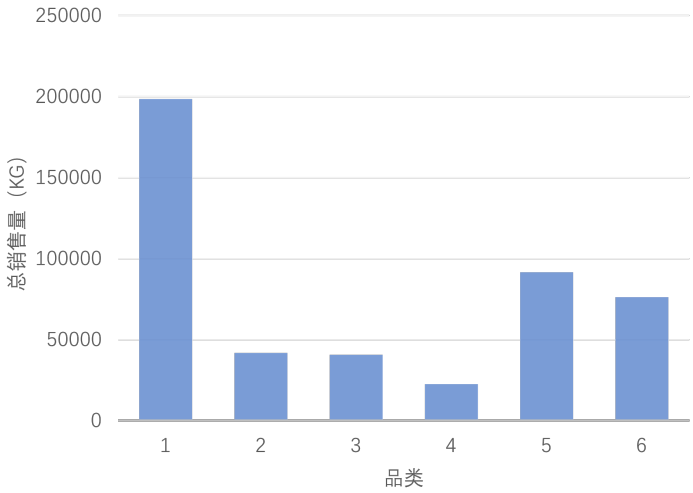
6.1 分布规律

6.1.1 空间上的分布规律

首先考虑不同品类之间的总销售量比例，如图5所示。可以看出在 6 个蔬菜品类中，花叶类（1）总销售量最多，辣椒类（5）次之，茄类（4）最少。



(a) 饼状图

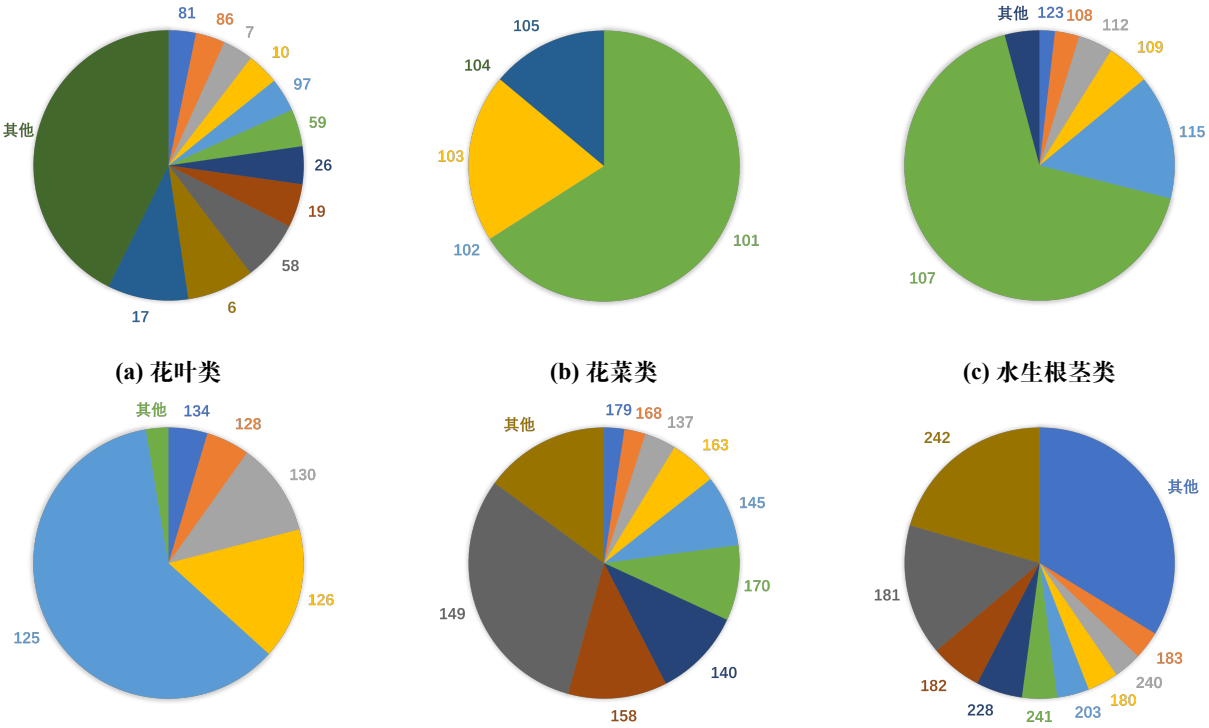


(b) 柱状图

图 5　6 类蔬菜销售量统计

接着考虑同类蔬菜中，不同单品之间的总销售量比例，如图7所示。分析可得：

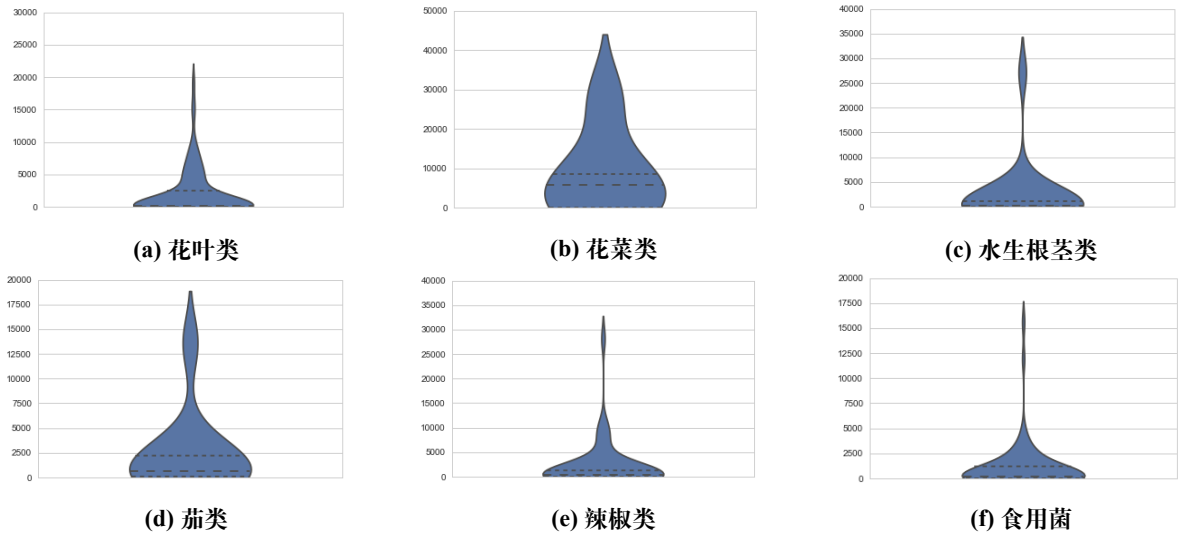
- 花叶类中：历史总销量集中在 0-5000kg，大白菜（17）总销售量最多，红橡叶（79）最少。
- 花菜类中：历史总销量集中在 0-15000kg，西蓝花（101）总销售量最多，紫白菜 (2)（104）最少。
- 水生根茎类中：历史总销量集中在 0-5000kg，净藕 (1)（107）销售量最多，荸荠（118）最少。
- 茄类中：历史总销量集中在 0-5000kg，紫茄子 (2)（125）销售量最多，圆茄子 (1)（133）最少。
- 辣椒类中：历史总销量集中在 0-5000kg，芜湖青椒 (1)（149）销售量最多，水果辣椒 (橙色)（148）最少。
- 食用菌中：历史总销量集中在 0-2500kg，金针菇 (盒)（242）销售量最多，活体银耳（207）金针菇 (份)（218）和虫草花 (盒)(2)（250）最少。



(a) 花叶类　(b) 花菜类　(c) 水生根茎类

(d) 茄类　(e) 辣椒类　(f) 食用菌

图 6　各分类销售量饼状图



(a) 花叶类　(b) 花菜类　(c) 水生根茎类

(d) 茄类　(e) 辣椒类　(f) 食用菌

图 7　各分类销售量小提琴图

6.1.2 时间上的分布规律

在进行了数量上分布规律的分析后，本文接下来进行对于不同品类和不同单品销售量时间上分布规律的分析。本文将数据进行预处理后，进一步将数据按照特定要求进行处理。分为了以天为单位的六种品类蔬菜的销售量统计和以扫码销售时间的六种品类

蔬菜的销售量统计。在这之后，使用 python 的 matplotlib 库进行绘制并进行可视化的分析。可视化结果呈现如下：

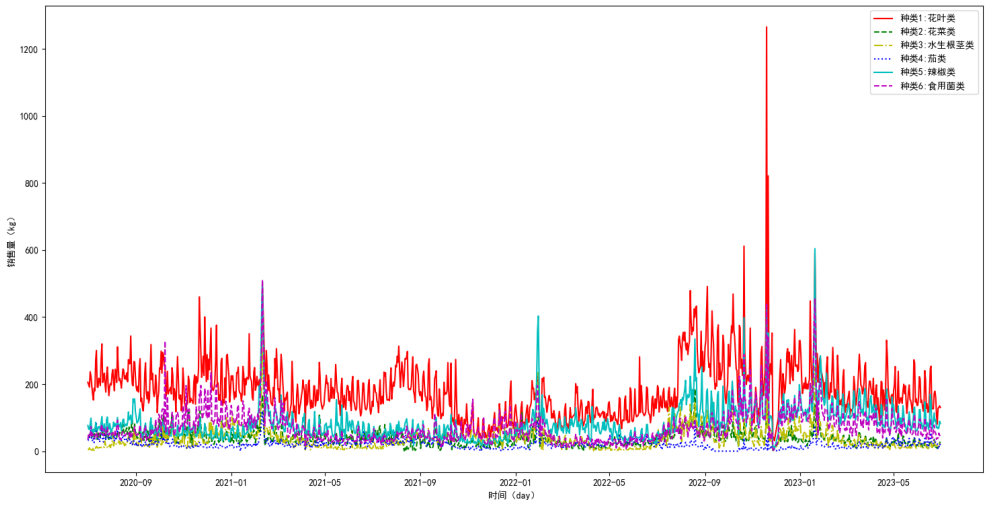


图 8　以天为单位的六种品类蔬菜时间分布规律

图8为以天为单位的六种品类蔬菜的时间分布规律可视化结果。分析可得：

1. 销售量的总体水平：可以看到种类一的花叶类的销售量数据最为突出且一直处于较高水平，其次是辣椒类和食用菌类。花菜类、茄类、水生根茎类销售量较低且处于较平稳的变化状态。数据量的关系和 6.1.1 的分析结果基本相吻合——花叶类销售量最多，茄类最少等结论。
2. 销售量的变化趋势：可以看到使用天作为基本单位后，销售量呈现了随时间极为剧烈的变化波动，这其实也是侧面对于月份、季节等其它影响因素的体现，与现实情况相符合。
3. 新冠疫情对于蔬菜销售量的影响：仔细观察并分析一些特殊节点和阶段的销售量变化后，得出了以下一些特殊时间规律体现了疫情对于蔬菜销售量的潜在影响：
    - 第一次大规模爆发，散点爆发和动态清零（2020 初-2021 初）：从 2020 年上半年的正式爆发到 2021 年 3 月的严格防控。此时疫情初期，人员延迟复工、进口国限制措施等因素叠加，农产品出口延期或订单减少，对果蔬、水产品等影响较大。同时供给端短期出现紧平衡、结构性短缺和部分农产品滞销卖难并存，部分地区农产品市场价格出现明显波动。可以发现，图8在 2020 年到 2021 年初期确实出现了明显的变化，尤其是食用菌和辣椒类产品，出现了明显的波动和下降趋势。
    - 大规模疫苗接种和第二次大规模爆发（2021 初-2022 末）：从 2021 年的大规模疫苗接种在到 2022 年由于变异株导致的二次爆发。端零售渠道结构发生了明显变化，超市消费受影响不大，生鲜电商、社区菜店和社区团购消费激增，农贸市场

消费下降明显。同时农产品需求端发生变化，各地机关团体食堂、餐饮与节庆活动相关的集体性消费需求明显减弱，但家庭日常生活消费需求是刚性的。可以发现，图8在 2021 年到 2022 年确实处于明显的下降趋势是并且一直保持稳定在一个较低的销售量水平。

- 第三次大规模爆发以及正式大流行结束（2022 末-2023 初）：从 2022 末的最后一次疫情爆发到 2023 年 2 月中国正式宣布大流行结束。由于疫情的严重程度逐渐减弱再到正式结束，可能会带动居民对于农贸市场以及蔬菜产业的消费需求的反弹；另外随着居民疫情防控意识增强，消费者更加关注农产品安全质量，价格不再是决定性因素，卫生干净和食品新鲜成为首要决策因素，这样的情况也会对蔬菜需求量带来一定影响。可以发现，图8在 2022 年末和 2023 年初确实存在一个非常明显的销售量反弹，这有可能是疫情放缓所带来的需求反弹导致；另外疫情彻底结束后这种反弹没有出现较长时间的维持，可能也与上述提到的消费者对于蔬菜类商品要求提高有关。

我们通过分析蔬菜销售量，发现了其中新冠疫情各阶段和时间节点对于蔬菜销售量的影响，并且通过分析，这些现实情况于本文的结果相一致，进一步证明了本文数据与分析结果的科学性和准确性。

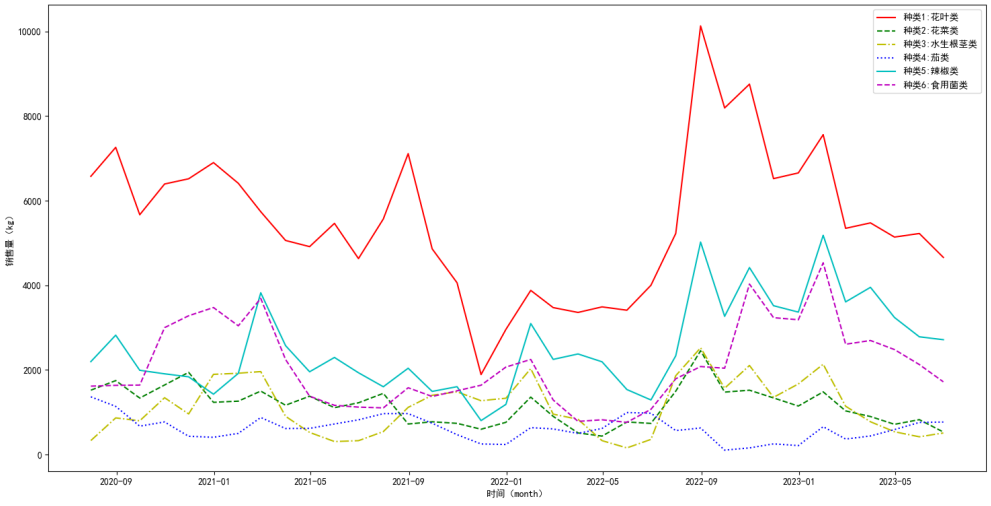


图 9　以月为单位的六种品类蔬菜时间分布规律

图9为以天为单位的六种品类蔬菜的时间分布规律可视化结果。分析可得：完整的年份有 2021 和 2022 两年，这两年数据中都可以发现从 4 月到 10 月的一共六种品类销售量都有着明显的上升趋势并维持在较高水平，之后在 11 月之后呈现一个下降趋势，并在 12 月和来年的 1 月达到最低。**这与题目中所述的“蔬菜的供应品种在 4 月至 10 月较为丰富”相印证**。

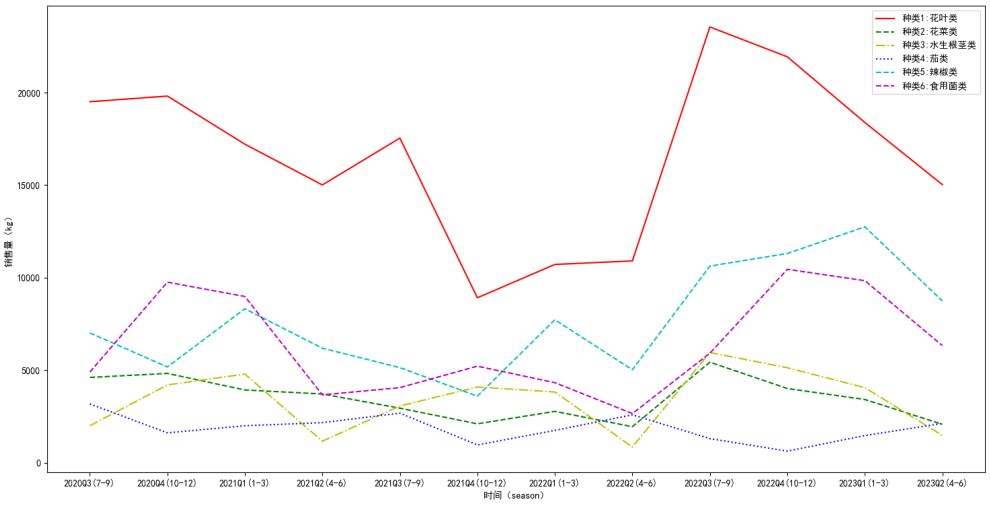


图 10　以季度为单位的六种品类蔬菜时间分布规律

图10为以季度为单位的六种品类蔬菜的时间分布规律可视化结果。分析可得：

- 生产季节性: 蔬菜的生产受季节影响较大，不同的季节，适合种植的蔬菜种类不同。例如，春季和秋季适合种植叶菜类、根茎类蔬菜，夏季适合种植茄果类、瓜类等。可以发现春季（1-3）和秋季（7-9）花叶类和花菜类确实有相同并且较明显的上升趋势，例如 2022Q3 对于花叶和花菜类都达到了销售量的顶峰；同时茄类也可以看到在每年的 4-6 月份有着明显的上升，例如 2023Q2。
- 消费习惯：消费者的购买习惯也会随着季节变化。例如，冬季人们更倾向于购买能增加热量的蔬菜，如土豆、南瓜等；夏季则更喜欢清淡的蔬菜，如黄瓜、西红柿等。可以发现冬季的辣椒类的销售量确实在冬季呈现着较明显的上升趋势，这可能就与上面提到的冬季人们的销售习惯即倾向于购买能够增加热量有关的蔬菜有关。
- 销售组合：由于题目中所述的从供给侧来看，一些季节的影响和商超销售空间的限制，使得商超可能会采用捆绑销售，即销售组合的销售策略。可以发现，图中的花叶类和花菜类在季度为时间单位衡量下，有着明显的共同变化趋势，这可能就与二者相似的种植和丰收习惯有关，进而导致的商超的销售组合，**这也侧面印证了题目中所述的可能出现的销售组合**。

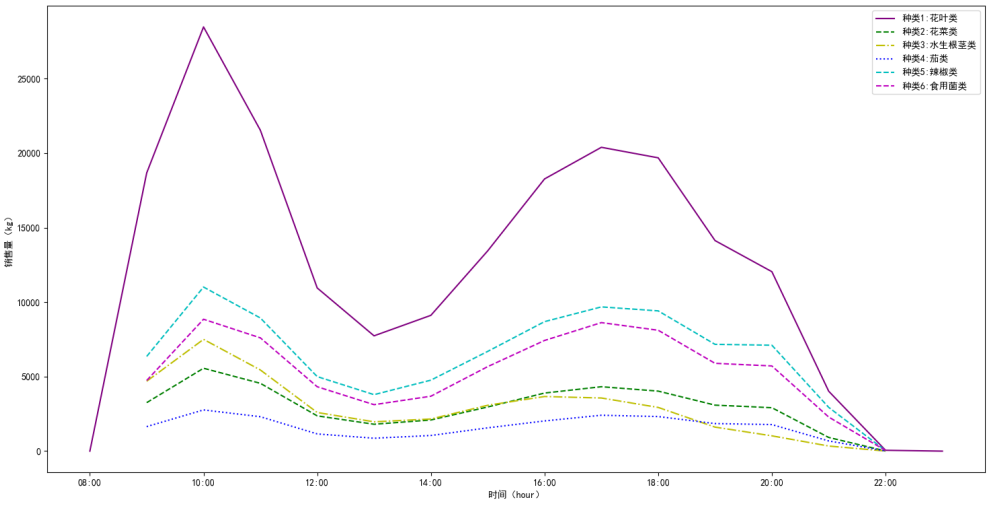


图 11　一天内六个品类的销售量变化

图11为在一天内的六种品类蔬菜的购买时间分布规律。分析可得：所有的品类商品都呈现了一个在清晨（10 点之前）销售量急剧上升并在 10 点达到顶峰的规律，这可能与人们倾向于在早晨买菜为午饭准备食物的规律有关；同时所有品类商品也都呈现了一个在下午 5 点前的上升趋势，并且在夜晚甚至深夜后达到最低，这可能与人们也倾向于在下午买菜为晚上准备食物的规律有关。**因为这也与实际生活情况相符合，侧面印证了本文分析结果的科学合理性**。

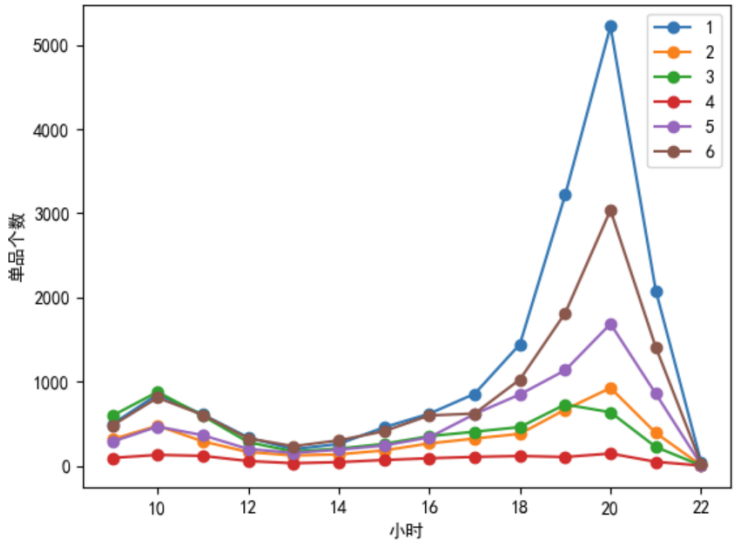


图 12　一天内六个品类打折的规律

图12为在一天内的六种品类蔬菜的打折规律。分析可得，在 20 点左右打折的商品数急剧增加，考虑到蔬菜类产品容易腐败而对品质产生影响，且大部分品种如当日未售

出，隔日就无法再售，因此在晚间时分商家会采取打折的方式尽可能吸引更多的顾客从而甩卖掉品质下降的蔬菜。**这同样与实际生活情况相符合，侧面印证了本文分析结果的科学合理性**。

6.2 相关性分析

6.2.1 相关系数分析

对于品类之间的相互关系，本文选择利用相关系数进行分析。Spearman 相关系数是一种非参数性的相关系数，它用于衡量两个变量之间的单调关系，而不仅仅是线性关系。它通过将变量的观测值转换为相对排名来计算相关系数。与 Pearson 相关系数相比，Spearman 相关系数不需要对数据做出线性假设，因此适用于本题的数据。Spearman 相关系数的计算公式如下：

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n\left(n^2-1\right)}$$

对 6 个品类随日期变化的销售量求 Spearman 相关系数，如图13所示（P 值均通过显著性检验）。分析可得：品类 1-2、1-5、1-6、3-6、5-6 之间存在较明显的相关性。另外可以发现，品类 4 与其他各品类间均没有明显的相关性。

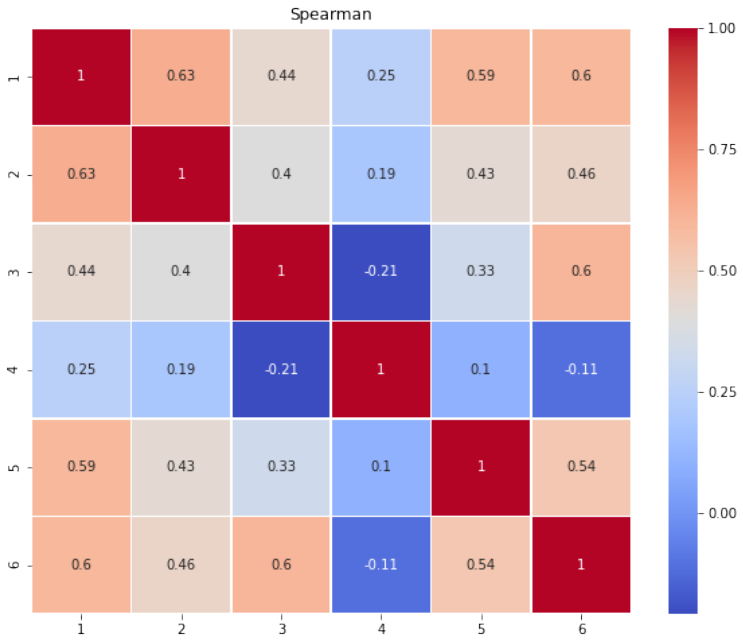


图 13　Spearman 相关系数矩阵

6.2.2 聚类分析

对于单品之间的相互关系，本文选择利用层次聚类进行分析。层次聚类是一种用于将数据集中的对象或数据点按照相似性进行逐层划分的聚类方法。在层次聚类中，数据点首先被视为单独的聚类，然后逐渐合并为更大的聚类，直到所有数据点都合并为一个大的聚类，形成一个层次结构的聚类树状图，称为聚类树。

由于题目中蔬菜单品数量较多，通过相关系数两两进行分析过于繁杂且并不能清晰表明品类之间的关系，所以本文采用单品销售量随日期的变化进行层次聚类，以此来表示品类之间的关系。本文认为，通过层次聚类被划分到同一类下的单品，在销售量随日期变化上具有相似性，进而认为该类下的单品具有关联性。

在层次聚类前需要对销售量数据进行标准化。这是因为分析相关性时考虑的是单品间的变化趋势是否具有相似性，但是由于不同单品间的销售量差异较大。若直接对销售量进行聚类会导致两种变化趋势相似，而销售量差距较大的两种单品不能被聚为一类。所以标准化的目的是去除销售量的影响，突出变化趋势的相似性。

本文采用的标准化方法为 z-score，将销售量数据转化为均值为 0，标准差为 1 的标准化数据，计算公式如下：

$$z = \frac{x - \mu}{\sigma}$$

μ 为平均值，σ 为标准差

经过标准化后对数据进行层次聚类，如图14所示。本文选取了被聚为一类的几组数据作为测试数据进行相关性分析，如表5、6和7所示（P 值均通过显著性检测），以此证明使用层次聚类反映相关性的合理性。测试结果表明，利用层次聚类的方法聚为一类的数据间具有相关性。

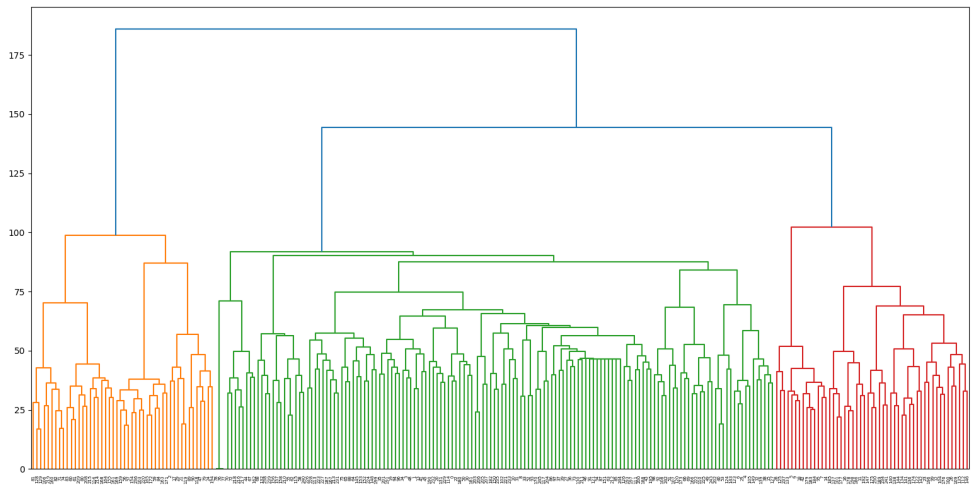


图 14　层次聚类结果

表 5　测试数据 1 的 Spearman 相关系数

	灯笼椒 (1)（143）	灯笼椒红灯笼椒 (1)（147）
灯笼椒 (1)（143）	1.000000	0.666166
灯笼椒红灯笼椒 (1)（147）	0.666166	1.000000

表 6　测试数据 2 的 Spearman 相关系数

	姬菇 (1)（184）	海鲜菇 (1)（188）
姬菇 (1)（184）	1.000000	0.834011
海鲜菇 (1)（188）	0.834011	1.000000

表 7　测试数据 3 的 Spearman 相关系数

	青尖椒（136）	小米椒（144）
青尖椒（136）	1.000000	0.698401
小米椒（144）	0.698401	1.000000

七、 问题二的模型建立与求解

7.1 对销售总量与成本加成定价的关联分析

为了探究各类别销售总量与成本加成定价的关系，本文首先通过 Spearman 相关性分析方法定性分析销售量、批发价格、销售价、利润率之间的相关程度，得到相关系数矩阵，从而为后续定量探寻销售量与价格相关量的关系建立基础。计算结果如下所示：

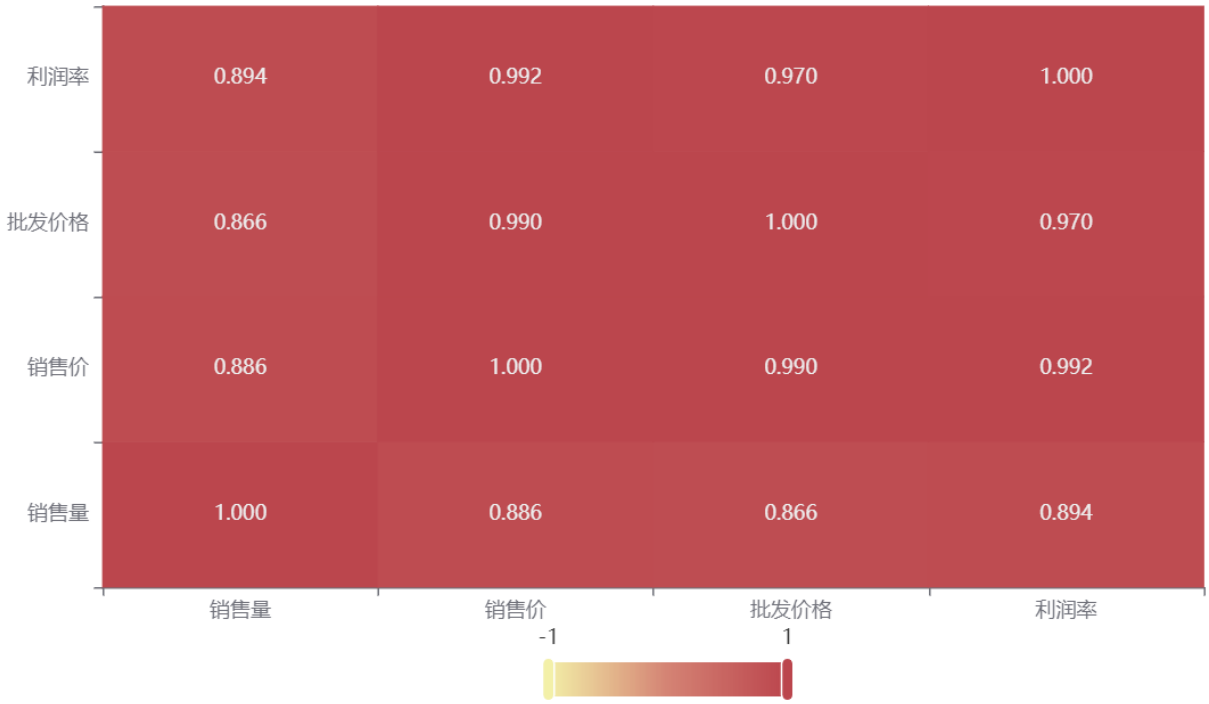


图 15　销售量与成本定价的相关系数矩阵

经过计算可知：

- 销售量与批发价格之间的相关系数为 0.866，显著性为 0.0021。表明两者之间存在显著的关联。由于批发成本侧面反映了商品的价值，销售量可以体现出用户的购买需求，印证了现实中供给与需求的紧密关系。
- 销售量与销售价之间的相关系数为 0.886，显著性为 0.0018。表明两者之间存在显著的关联。这表明销售价的制定对销售量有一定的影响，可能存在价格敏感性，即降价可能会促进销售量的增长。
- 销售量与利润率之间的相关系数为 0.894，显著性为 0.0004。表明两者之间存在显著的关联。这意味着销售量的增加或减少可能会对利润率产生影响，销售量的增加通常伴随着利润的增加。

成本加成定价是指按产品单位成本加上一定比例的利润制定产品价格的方法，计算公式为：价格 = 单位成本 (1+ 成本利润率)。本文在研究销售量与成本加成定价的关系时，选取成本利润率作为成本加成定价的指标。成本利润率在本题中的计算公式为：

成本利润率 = 价格/单位成本-1= 销售价/批发价-1。

由上述分析可知，各类别销售总量与利润率的相关性程度最高，因此我们选择采用多种机器学习回归方法法研究每日销售量与成本利润率之间的关系，从而反映出销售量与成本加成定价的关系，为后续的补货和定价决策提供支撑。

- **线性回归**：线性回归假设预测变量与目标变量之间存在线性关系，并基于最小二乘法来拟合最优的线性模型。它的预测结果是目标变量的线性组合，可以用于解释特

征的重要性。然而，线性回归可能不适用于非线性关系和复杂的数据集。

- **决策树回归**：决策树回归模型通过树形结构来进行决策，根据特征的取值将数据分割为不同的区域。然后利用每个区域中的平均值或中位数来对新数据进行预测。但其容易过拟合，可以通过剪枝、限制树的深度等来减小过拟合的风险。
- **随机森林回归**：随机森林回归是一种集成学习模型，通过组合多个决策树来进行预测。它具有很高的预测准确性和鲁棒性，能够处理大量特征和高维数据，能够有效地减少过拟合，并且对于大规模数据和高维数据的处理效果较好。
- **XGBoost 回归**：XGBoost 回归使用一种加权的梯度提升方法，通过逐步改进模型的预测结果以提高准确性。它提供了一些正则化技术来防止过拟合，并具有特征重要性评估的功能，能够自动处理缺失值和可处理非线性关系。

结果如表8所示，可以看出 XGBoost 回归方法的 MAPE 值较小，R^2 值最接近 1，说明 XGBoost 回归效果较好。因此在后续问题的求解中，可以采用 XGBoost 回归的方法根据批发价格和成本利润率预测当日销量。图16展示了 XGBoost 回归在测试集上的预测效果。

表 8　各机器学习回归方法的模型评估

	MAPE	R^2
XGBoost 回归	12.573	0.887
决策树回归	25.147	0.412
随机森林回归	9.811	0.308
线性回归	29.742	0.001

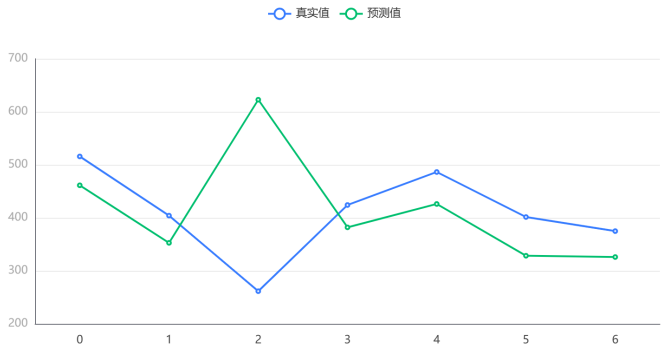


图 16　XGBoost 回归在测试集上的结果

7.2 基于 ARIMA 算法的批发价预测模型

7.2.1 时间序列分析介绍

时间序列分析算法旨在研究时间序列数据中的潜在模式、趋势、周期性和预测性等特征。

ARIMA 模型是一种广泛应用于时间序列分析的统计模型。它结合了自回归（AR）、移动平均（MA）和差分整合（I）这三个部分。AR 模型通过过去观测值的线性组合来建模时间序列的未来值，MA 模型通过过去观测值的线性组合来建模序列中的随机误差项，而差分整合部分则处理序列的非稳定性，通过差分运算将非平稳序列转化为平稳序列。

季节性分解方法是一种常用的时间序列分析方法，旨在将时间序列分解为趋势、季节性和随机成分。该方法常用的有加法模型和乘法模型两种。加法模型认为时间序列是由趋势成分、季节性成分和随机成分的加和构成。乘法模型则认为时间序列是趋势成分、季节性成分和随机成分的乘积构成。这些分解方法通常利用移动平均、加权移动平均或回归分析等技术来识别和估计趋势和季节性成分，并使用残差来表示随机成分。

由于蔬菜类的销售数据与季节因素联系紧密，且体现了较强的周期性，因此本文联合 ARIMA 模型与季节性分解方法，提出了基于季节规律的 ARIMA 算法，企图挖掘过去两年的各类别批发价数据，并寻找内在规律，借以预测未来一周内的各品类蔬菜的批发价，为商家的补货和定价决策提供有力支持。

7.2.2 季节性 ARIMA 算法的批发价预测模型

当构建时间序列预测模型时，选取历史的不同时间尺度的数据作为训练数据必要的。这是因为时间序列数据通常包含多个时间尺度上的趋势和模式，既要考虑历史规律对批发价造成的影响，又要兼顾近期销售情况对未来一周各类别批发价的影响。本文在此定义三种类型的时间尺度：长尺度（年）、中尺度（月）、短尺度（周）。通过分别选取不同长度的时间段的历史批发价数据作为训练数据，对未来一周的各类别批发价进行预测，最后对这三种尺度下的预测数据进行加权平均得到最终结果。

在融合了季节性分解方法与传统 ARIMA 算法，并以 2020.7-2023.6 的历史批发数据作为训练数据输入到模型中，对六个类别分别进行操作，得到六个类别在 2023.7.1-2023.7.7 的预测批发价，下面以单品数量最多的类别一为例，展示通过时间序列分析模型得到的结果以及模型检验的过程。图19为预测所得的结果：

图 17　未来一周批发价预测结果：长尺度

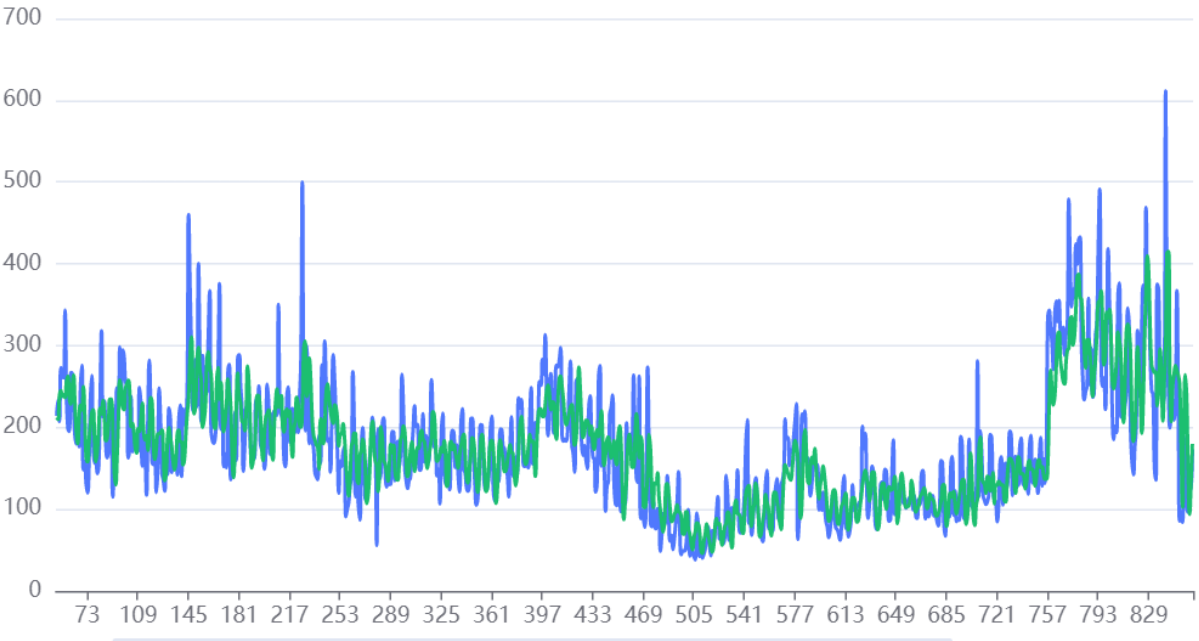


图 18　未来一周批发价预测结果：中尺度

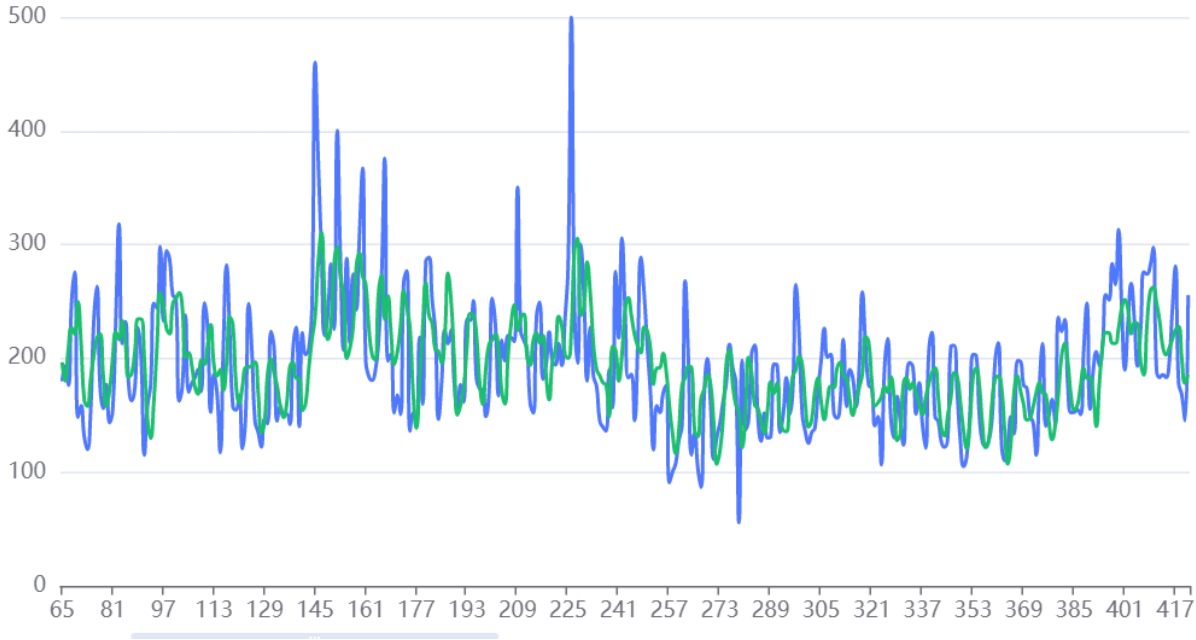


图 19　未来一周批发价预测结果：短尺度

综合长、中、短三种尺度的预测模型的结果，类别 1 未来一周每天具体的批发价预测值如下：

表 9　7.1-7.7 预测批发价

日期	预测批发价（元/千克）
7.1	6.362159524
7.2	7.100312638
7.3	6.839272125
7.4	7.318493621
7.5	7.174246732
7.6	7.596121656
7.7	6.493471193

得到预测数据后，需要对数据进行检验以验证模型的正确性，ARIMA 模型要求序列满足平稳性，查看差分前后数据对比图20，可见上下波动幅度不大，同时对时间序列进行自相关分析，根据截尾情况估算其 p、q 值，统计 ADF 检验的结果，如图25所示：

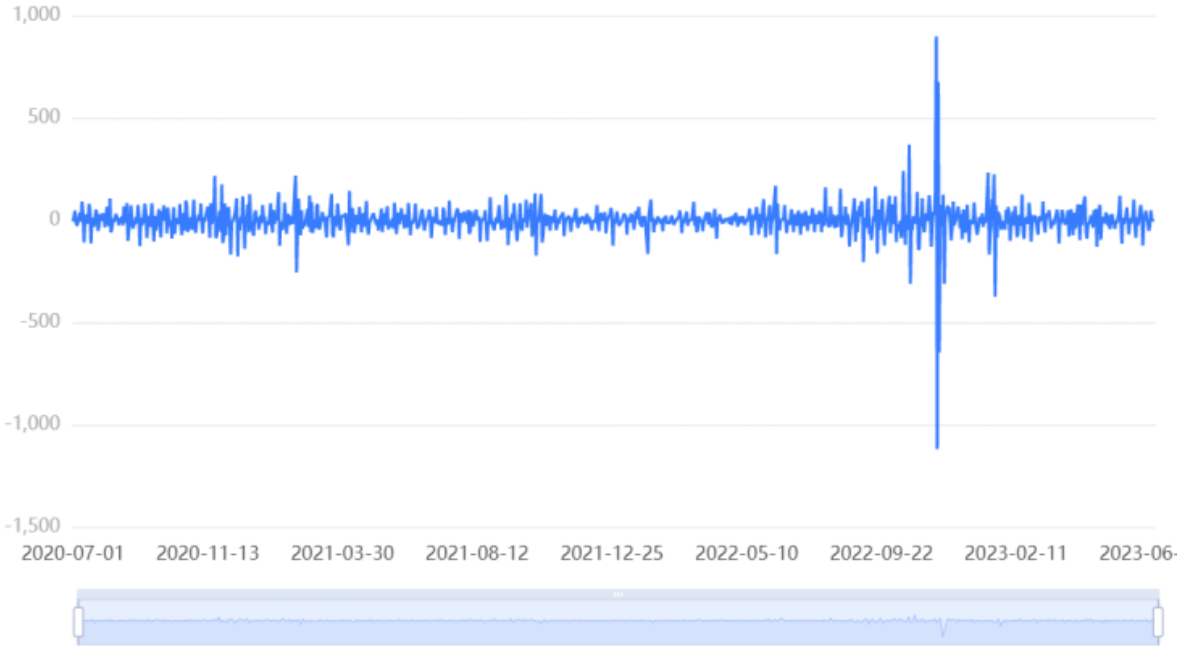


图 20　模型检验：差分图

表 10　模型检验：ADF 检验

ADF 检验表

变量	差分阶数	t	P	AIC	临界值		
					1%	5%	10%
批发价	0	-2.819	0.056*	11690.798	-3.437	-2.864	-2.568
	1	-11.77	0.000***	11685.767	-3.437	-2.864	-2.568
	2	-15.098	0.000***	11770.28	-3.437	-2.864	-2.568

注：***、**、* 分别代表 1%、5%、10% 的显著性水平

分析结果如下：

- 若呈现显著性 (P<0.05)，则说明拒绝原假设，该序列为一个平稳的时间序列，反之则说明该序列为一个不平稳的时间序列。临界值 1%、5%、10% 不同程度拒绝原假设的统计值和 ADF Test result 的比较，ADF Test result 同时小于 1%、5%、10% 即说明非常好地拒绝该假设。
- 差分阶数：本质上就是下一个数值，减去上一个数值，主要是消除一些波动使数据

趋于平稳，非平稳序列可通过差分变换转化为平稳序列。

- AIC 值：衡量统计模型拟合优良性的一种标准，数值越小越好。

可见，在差分为 2 阶时，显著性 P 值为 0.000***，水平上呈现显著性，拒绝原假设，该序列为平稳的时间序列。因此，该预测模型是合理的。

7.3 目标规划模型的建立

在得到了上述预测的数据之后，本文正式进行对第二问的数学模型建立。显而易见，本问是在考察一个多目标规划问题的数学模型建立。但是本文考虑到由于上述预测的数据即使综合了多种尺度下的预测结果，但是如果直接当作未来 7 天内真实的批发价即成本仍然缺乏科学性，例如一些其它因素可能会对批发价即成本造成影响，比如一些品类蔬菜可能面临的灾害以及随时可能复发的新一波疫情浪潮。同时也为了能够更好地拟合现实中多变的情况和多个目标的优先级，本文决定舍弃传统的将大多数约束条件视为硬性约束条件的多目标规划，而转而采用运筹学中的目标规划算法对问题进行求解。

7.3.1 目标规划模型简介

$$\min\{P_l(\sum_{\mathrm{k}=1}^{K}(W_l k^{-} d_k^{-} + W_l k^{+} d_k^{+})), l = 1, 2, \cdots, L\} \tag{1}$$

$$s.t. \begin{cases} c_{kj}x_j + d_k^{-} - d_k^{+} = g_k(k = 1, 2, \cdots, K) \\ \sum_{j=1}^{n} a_{ij}x_j \leq (=, \geq) b_i(i = 1, 2, \cdots, m) \\ x_j \geq 0(j = 1, 2, ..., n) \\ d_k^{-}, d_k^{+} \geq 0(k = 1, 2, ..., K) \end{cases} \tag{2}$$

上述式 1 中的 P_l 表示优先因子，式 1 代表目标函数。

上述式 2 中第一行式子中的 g_κ 代为第 k 个目标约束的预期目标值，W_{lk}^{-} 和 W_{lk}^{+} 为 P_l 优先因子对应各目标的权系数。式 2 第一行式子代表目标约束，式 2 中第二行式子代表绝对约束，式 2 中第四行式子代表偏差变量。

目标规划问题的建模步骤：

- 根据要研究的问题所提出的各目标与条件，列出绝对约束；
- 可根据决策者的需要，将某些或全部绝对约束转化为目标约束。这时只需要给绝对约束加上负偏差变量和减去正偏差变量即可；
- 给各目标赋予相应的优先因子 $\mathrm{P_k}(\mathrm{k} = 1, 2, \ldots, \mathrm{K})$；
- 对同一优先等级中的各偏差变量，若需要可按其重要程度的不同赋予相应的权系数；

- 根据决策者的要求，构造一个由优先因子和权系数相对应的偏差变量组成的要求实现极小化的目标函数。

因此与传统的线性规划问题相比，目标规划模型只需寻找目标函数的满意解即可无需找到最优解；并不一定需要满足所有约束；并且目标与约束条件都具有轻重缓急、主次之分。因此本文最后决定采用该模型构建第二问的数学模型。

7.3.2 目标规划模型的建立

基于上述对目标规划模型的理解，本文将相关约束要求总结成如下几点，并依次分析：

1. 商超收益最大：
   - 优先级：考虑到商超收益应该是商超进行补货计划和定价策略的第一目标，本问题将其优先因子设为优先级最高。
   - 计算方式：采用销售所获取的总收益（其中真是销售的量需要减去退货的量）减去总成本（采用补货计划与成本的乘积）。
   - 目标：选取超过历年来的最大收益作为目标。
   - 偏差变量权系数分配：由于要求超过目标值，所以只有负偏差变量，且权重为 1。
2. 满足需求与仓储成本：由于这二者都可以通过一天的销售完成后的剩余量进行衡量，因此将其同时归为一类。
   - 优先级：首先考虑到如果无法满足需求。不仅会带来收益上的损失，也会一定程度上对消费者的信任造成影响，进一步对供应商的信誉度造成负面影响。再同时考虑到如果购货量过多，远超过需求量，会对多余的蔬菜产生一定的浪费，同时也会造成存储压力的增大和成本的升高，带来不必要的损失。因此本问题将其优先因子设为优先级第二高。
   - 计算方式：采用补货量减去销售总量，再加上退货量。
   - 目标：由于综合考虑以上两个因素，如果最终能够保证一天的销售完成后的剩余量正好为 0 将是最好，因此将目标设为 0。
   - 偏差变量权系数分配：由于本问中考虑到如果无法满足需求带来的负面影响肯定要大于仓储成本带来的负面影响，因此负偏差系数的重要性要大于正偏差系数，将二者比重设为 7：3。
3. 销售空间有限以及顾客最大购买力：
   - 优先级：考虑到本问中并未提供销售空间的大小数据，本问决定使用历年的最大销售量作为衡量销售空间的标准，同时也可以一定程度上反映顾客最大的购买力。如果补货量超过该数值，会造成没有必要的成本浪费，仓储成本以及面临销售空间不够用的可能。但考虑到该情况发生几率较低，因此本问题将其优先因子

设为优先级第三高。

- 计算方式：采用补货量减去历年的最大销售量。
- 目标：选取小于 0，即补货量小于历年最大销售量为目标。
- 偏差变量权系数分配：由于要求低于目标值，所以只有正偏差变量，且权重为 1。

4. 成本加成定价：

- 有关利润率和销售量之间的关系：本文采用 7.1 中使用的 XGBoost 模型进行拟合关系。并在该问中具体求出利润率和销售量的相关关系。
- 成本加成定价：定价 = 成本（1+ 利润率），将该约束条件加入到目标规划模型中，也是唯一一个硬性约束条件。

除此之外，本问还考虑将退货量以历年的平均退货率作为数据进行计算，即为每 kg 销售出去的商品有多少被退货回来。基于上述分析，本文建立如下的目标规划模型：

$$\min \quad z = P_1 d_1^- + P_2(3d_2^- + 7d_2^+) + P_3 d_3^+ \tag{3}$$

$$\text{s.t.} \quad \sum_{i=1}^{6}(S_{ji}(1-q_i)) \cdot T_{ji} - (Xb^T)_{ji}^T + d_{1i}^- - d_{1i}^+ = 10147.93 \tag{4}$$

$$\sum_{i=1}^{6}(X_{ji}) - \sum_{i=1}^{6}(S_{ji}(1-q_{ji})) + d_{2i}^- - d_{2i}^+ = 0 \tag{5}$$

$$\sum_{i=1}^{6}(X_{ji} - H_j) + d_{3i}^- - d_{3i}^+ = 0 \tag{6}$$

$$T_{ji} = b_{ji}(1+n_{ji}) \tag{7}$$

$$n_{ji} \xrightarrow{XGBoost} S_{ji} \tag{8}$$

$$i = (1, 2, \cdots, 6), \quad j = (1, 2, \cdots, 7)$$

模型解释：

- 式 3 是目标规划模型的优化函数，即对三个约束条件进行优先级排序后，对所有偏移变量最小化优化，采用序贯法求解。
- 式 4 是总利润的限制条件，表示未来一周六大品类总销售额减去总成本要尽可能接近历史最大利润，保证收益最大化。
- 式 5 是库存的限制条件，表示补货量减去销售总量，再加上退货量，所得结果要尽可能接近于 0，既要满足用户需求，又不应过多而产生存储成本。
- 式 6 是销售空间的限制条件，表示补货量需要小于用户历史上最大购买量，以避免蔬菜资源过剩而产生不必要的成本。
- 式 7 是成本加成定价的表达式，表示销售价 = 成本 *（1+ 利润率）
- 式 8 是通过 XGBoost 回归模型得到的利润率与销售量的关系，借助该关系可以将销

售量用利润率来表示，从而有利于规划模型的求解。

7.4 求解结果

利用 Matlab 对上述目标规划模型进行求解，可以分别得出 7 月 1 日-7 月 7 日各品类的日补货总量以及定价策略，如表11和12。

表 11　补货策略（单位：kg）

日期	品类					
	1	2	3	4	5	6
7.1	248.79	62.38	39.17	50.98	101.45	66.67
7.2	275.81	69.56	101.67	47.76	97.58	93.35
7.3	214.69	47.29	48.98	35.45	143.23	72.03
7.4	199.08	61.25	108.72	51.26	176.25	88.293
7.5	224.12	58.57	81.25	32.09	119.08	129.08
7.6	301.45	49.15	106.61	53.78	200.11	69.09
7.7	267.34	45.69	78.90	34.02	132.21	108.42

表 12　定价策略（单位：元）

日期	品类					
	1	2	3	4	5	6
7.1	19.28	12.18	18.11	8.15	21.84	12.44
7.2	10.09	12.96	13.76	8.06	18.78	13.97
7.3	8.37	11.09	15.26	8.45	8.98	16.45
7.4	9.67	11.28	14.18	7.99	14.89	13.48
7.5	12.22	10.89	13.98	9.34	6.17	18.99
7.6	15.65	13.37	12.76	7.29	12.01	11.24
7.7	8.91	13.09	11.59	7.76	7.58	12.18

7.5 灵敏度分析

本文中除了对重要性优先级进行了人为比较之外，更具有主观性的模型超参数在于（5）式中对于正和负两个偏移变量的权重系数分配。由于本问建模过程中考虑到如果（5）式不能满足等于 0 的条件，小于 0 则代表着没有满足购买的需求，大于 0 则代表着可能带来的仓储成本。因此小于 0 比大于 0 的后果更加严重，因此本文的模型更应该使得大于 0 作为目标，即负偏移变量比重大于正偏移变量。最后决定采用 $\frac{d_2^-}{d_2^+}=\frac{7}{3}$ 的系数比重建立模型。

通过反复调整参数配比即 $\frac{d_2^-}{d_2^+}$ 的比重，并且由于负偏移变量的重要性肯定要大于正偏移变量的重要性，因此让 $\frac{d_2^-}{d_2^+}$ 在 1 往上变化。并记录每次的 $\frac{d_2^-}{d_2^+}$ 与利润，得到如下图所示结果：

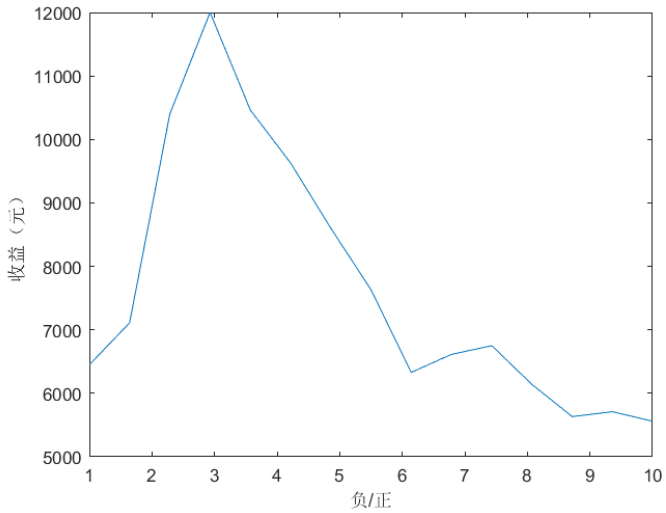


图 21　灵敏度分析结果

由上图所示，在本文模型中采用的 $\frac{d_2^-}{d_2^+}=\frac{7}{3}$ 的值能够达到较高水平，但仍然不是峰值，实际上峰值是在 3.00 左右即在 $\frac{d_2^-}{d_2^+}=\frac{7}{3}$ 与 $\frac{d_2^-}{d_2^+}=\frac{7.5}{2.5}$ 之间能够取到利润的最大值。同时在 $\frac{d_2^-}{d_2^+}$ 的取值在 4 之后，实际上已经达到了 $\frac{8}{2}\leq\frac{d_2^-}{d_2^+}\leq\frac{9}{1}$，此时的权重配比已经过于极端，因此导致了利润的急剧下降，也符合预期。可以发现，本文中建立模型所选取的参数也与最优值相差不远。**模型超参数选取较为成功**。

八、 问题三的模型建立与求解

8.1 销售单品的选择

鉴于题目要求 “可售单品总数控制在 27-33 个”，需要提前划定可售单品的范围。为尽可能满足市场对各商品的需求，本文选取了 6 月 24 日-30 日预测的总销售量位于前 33 位的单品作为可售单品，如表13所示。

表 13　可售单品

单品名	单品编码
芜湖青椒（1）	102900011016701
生菜	102900011030059
竹叶菜	102900005115786
.....	.....

8.2 销售单品的批发价预测

在本文中，参照问题二中预测各品类成批发价的季节性 ARIMA 算法，以月、周、日分别作为长、中、短尺度的时间范围，用历史的单品销量数据预测出选择出的 33 个单品在 7 月 1 日的批发价，三种尺度的预测结果图如下：

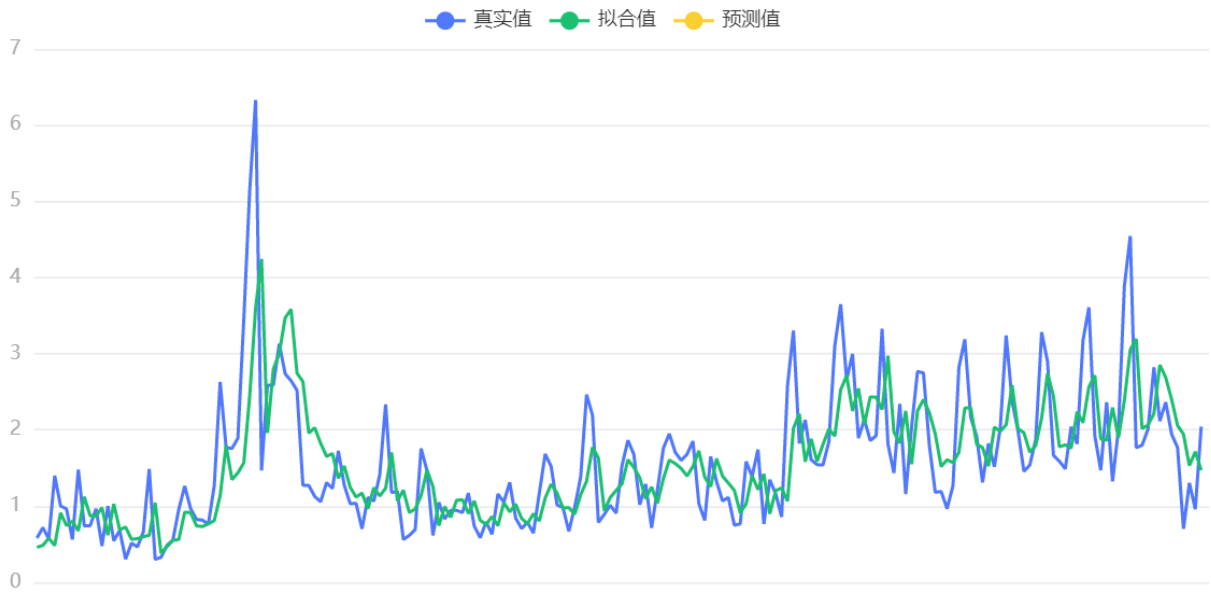


图 22　7.1 批发价预测结果：长尺度

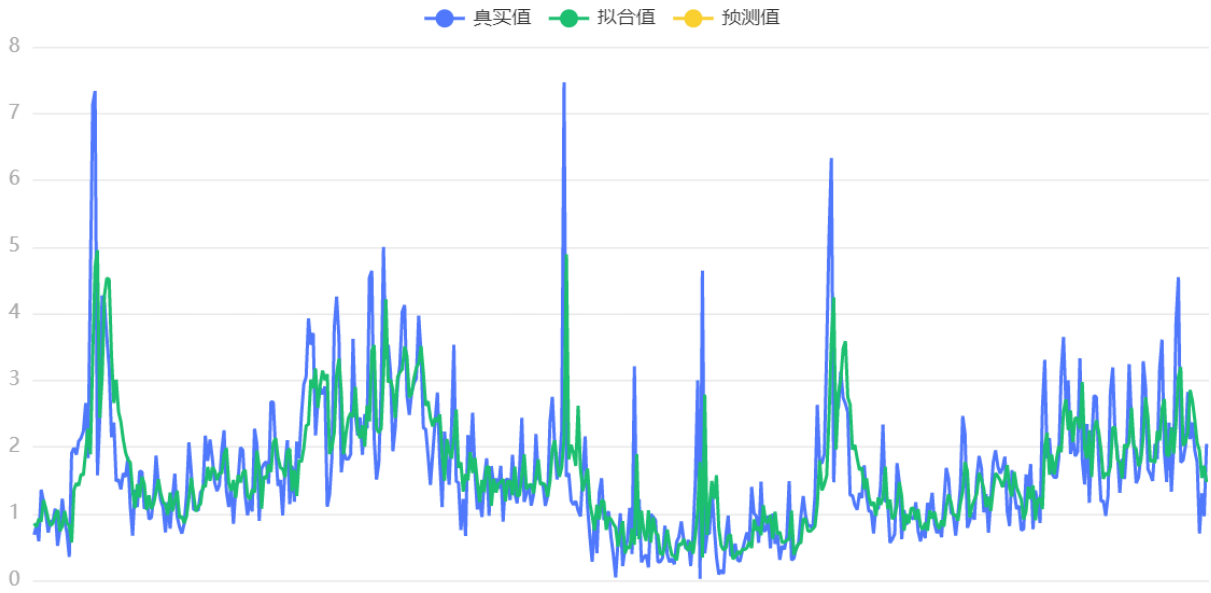


图 23　7.1 批发价预测结果：中尺度

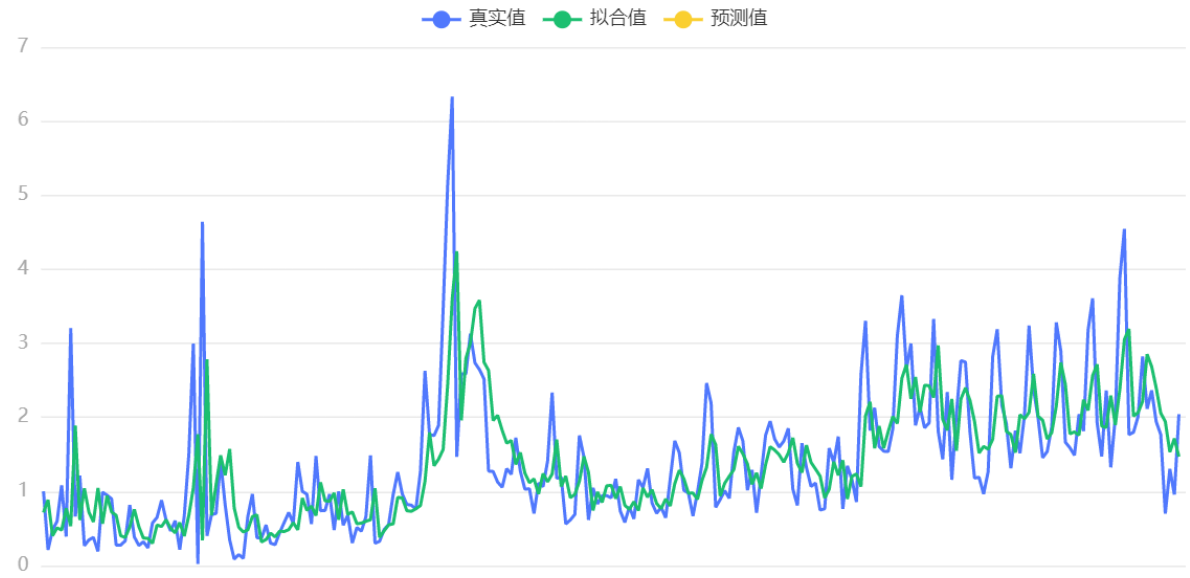


图 24　7.1 批发价预测结果：短尺度

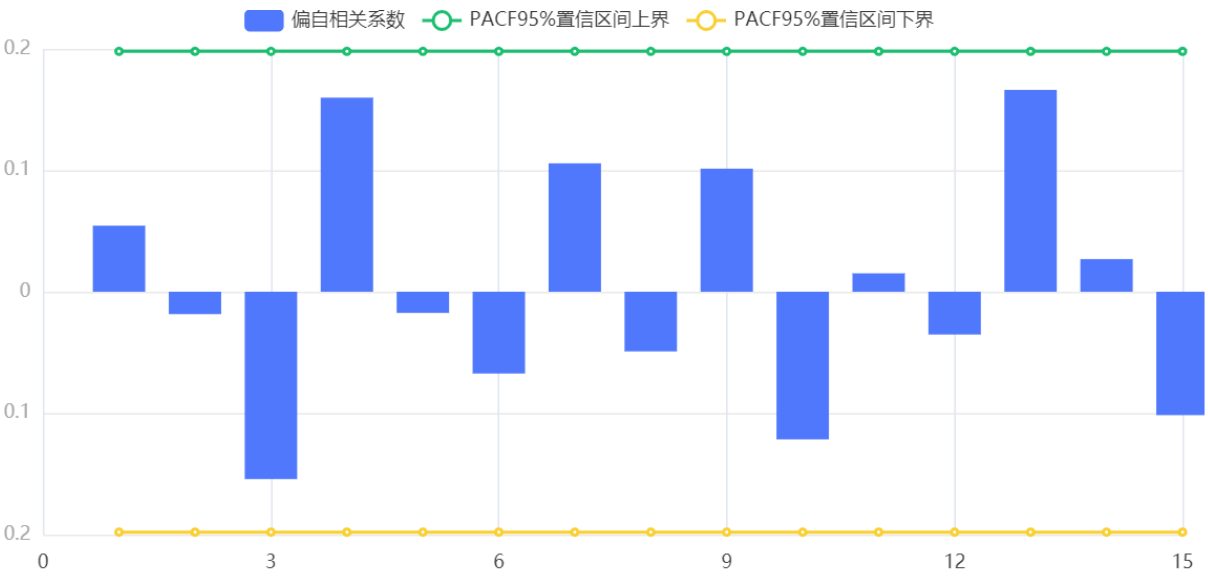


图 25　模型检验：差分图

经过 ADF 检验，33 种单品的批发价预测均有统计学意义，说明模型正确。

8.3 非线性规划目标的建立

本文同样也是一个多目标规划模型，不过不同于第二问，本问中出现了多个需要进行约束的条件，而本问也无法人为对这些约束条件的重要性进行确定，因此决定舍弃第二问的目标规划模型，转而使用传统的非线性规划模型求解。

本文利用预测得到的每小时单品销售，对 7 月 1 日补货进行规划，并对 9:00-21:00 每小时的定价策略进行实时调整，并将打折销售纳入考虑范围。以下为本文所建立的模

型：

$$\max \quad \sum_{i=1}^{23} a_i S_i^* T_i^* (1 - q_i^*) - \sum_{i=1}^{23} a_i X_i^* b_i^* (1 + l_i) \tag{9}$$

$$\text{s.t.} \quad a_i = \begin{cases} 0 \\ 1 \end{cases} \qquad i = 1 \cdots 33 \tag{10}$$

$$a_i X_i^* \geq 2.5 a_i \tag{11}$$

$$27 \leq \sum_{i=1}^{33} a_i \leq 33 \tag{12}$$

$$\sum_{i=1}^{3} X_i^* \leq 10147.93 \tag{13}$$

$$T_i \geq b_i \tag{14}$$

$$T_i = b_i (1 + n_i) \tag{15}$$

$$n_i \xrightarrow{XGBoost} S_i \tag{16}$$

$$S_i^*, X_i^*, T_i^* \geq 0 \tag{17}$$

模型解释：

- 式10决定了第 i 种商品是否进行销售，销售值为 1，不销售值为 0。由于题目将单品控制在 27-33 个，并要求各单品订购量满足最小陈列量 2.5 千克，因此式11、12对 a_i 进行了限制。
- 式9为目标函数，即最大化利润值。第 1 项为总销售额，第 2 项为成本，包括进货成本和损耗导致的成本。
- 式13表示考虑到商超的销售空间，因此总补货量小于等于历史单日补货量最大值。
- 式14表示定价需要大于等于进货价。
- 式15为成本加成定价公式，本题中成本为批发价。
- 式16表示通过训练完成的 XGBoost 回归模型，求的利润率 n_i 所对应的预测销售量 S_i。

8.4 非线性规划目标的求解

8.4.1 粒子群算法简介

粒子群算法（PSO）是一种元启发式优化算法，灵感来源于鸟群或鱼群等群体的群体行为。它通过模拟群体中个体之间的协作和信息交流来搜索问题的最优解。粒子群算法能够搜索整个解空间，寻找全局最优解；算法的思想直观且易于理解，实现相对简单；

粒子之间的搜索是并行进行的，适合在多核或分布式系统上使用；适用于连续优化问题和离散优化问题，可以灵活地应用于不同类型的问题。本文利用粒子群算法来优化补货决策和定价策略，以最大化商超的收益。

8.4.2 基于粒子群算法的求解过程

本文的求解过程如下：

1. 定义问题：明确收益最大化的目标，以及可售单品数量和最小陈列量的约束条件。
2. 初始化粒子群：每个粒子代表一个不同的补货方案和定价策略。
3. 计算适应度：根据历史销售数据和损耗率，评估每个粒子的适应度，即预测的收益值。
4. 更新粒子位置和速度：根据历史最优解和全局最优解的信息，更新每个粒子的补货量和定价策略。
5. 处理约束条件：对超出可售单品数量和陈列量要求的粒子进行约束处理，调整补货量和定价策略。
6. 更新最优解：根据最新适应度值，更新个体和全局最优解，记录收益最大的补货方案和定价策略。
7. 终止条件：达到预设的迭代次数，或满足商超需求的终止条件。
8. 输出结果：输出最优补货方案和定价策略，即使商超收益最大化。

8.4.3 模型求解结果

利用 Python 中的 pso 函数对上述模型进行求解，图26显示了某一单品在求解过程中的迭代过程。求解结果中，共选取了 33 种单品，如表14所示。

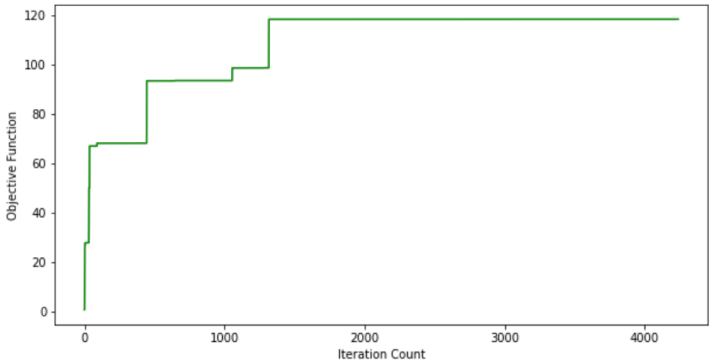


图 26　某单品 PSO 迭代折线图

表 14　定价策略和进货量（详见附件）

单品编码	定价（元/kg）	补货量（kg）	商超收益（元）
102900011016701	6.32	41.01	110.51
102900011030059	9.95	64.99	373.33
102900005115786	4.97	30.75	74.19
	.....		
	总收益（元）		2948.18

九、 问题四的模型建立与求解

为帮助商超更全面地了解市场动态，更精确地制定补货和定价决策，降低风险，提高销售和盈利能力。本文从供给侧和需求侧两方面进行分析。

9.1 供给侧

1. 库存数据：库存数据是有效补货的关键。商超需要定期监控蔬菜商品的库存水平，以确保库存不会积压或不足。了解库存周转率有助于确定最佳的补货时间和数量，从而降低滞销和损失。另外，还需要具体量化库存成本，方便统计总运营成本。
2. 供应链可靠性：了解供应链的质量和可靠性，包括供货产品品质、供应商的信誉和交货准时性，有助于商超确保蔬菜商品的稳定供应。
3. 商品腐败率：生鲜产品最大的吸引力之一来自于其新鲜程度，消费者是否进行消费很大程度上取决于商品的新鲜程度。因此提前了解不同商品的腐败率，可以提前制定相应的进货计划和定价策略，在商品临近腐败时进行打折处理，以避免商品腐败造成的损失。
4. 生产周期：考虑蔬菜的生长周期和季节性因素，以更好地规划供应和库存管理。这有助于避免供应不足或供应过剩。
5. 食品与卫生安全：随着居民疫情防控意识增强，消费者更加关注农产品安全质量，价格不再是决定性因素，卫生干净和食品新鲜成为首要决策因素。疫情促使消费者对农产品安全和品质的需求放大，推动消费升级。倒逼农业经营主体更好发力供给端，加快农产品标准化、品牌化和质量全程可追溯体系建设，发挥农产品供应链协同作用，以快速响应消费者便利消费、安全消费和品质消费需求。
6. 竞争市场分析：深入研究竞争市场，包括其他生鲜商超的产品组合、价格策略和促

销活动，有助于商超制定更具竞争力的补货和定价策略，以吸引更多客户并增加市场份额。

9.2 需求侧

1. 市场趋势：了解市场趋势和竞争对手的动态是制定战略决策的关键。商超需要分析市场趋势，以适应市场变化。
2. 顾客反馈：顾客反馈是改进蔬菜商品质量和选择的关键信息源。商超应积极收集顾客的投诉、建议和评价，以了解哪些方面需要改进，从而提高顾客满意度，增加忠诚度，进一步增加销售量。
3. 地理偏好：分析不同地理区域的销售数据可以揭示地区性的需求差异。商超可以根据这些数据优化库存和定价策略，以更好地满足不同地区的需求，提高销售和市场份额。
4. 顾客忠诚度：了解顾客的购买频率和偏好可以帮助商超识别忠诚的顾客群体。商超可以通过特别优惠、促销活动或定制化的服务来留住这些忠诚客户，提高客户保留率。
5. 社交媒体：监控社交媒体平台和在线评论可以帮助商超了解公众对其产品的看法。这有助于识别潜在的问题，改进产品质量，回应顾客关切，并提高口碑。

十、 模型评价

10.1 模型的优点

1. 第一问从时间和空间（种类）两个维度分析，由整体到局部，分层合理清晰。
2. 第二问采用 XGBoost、随机森林、决策树等多种机器学习回归模型探寻二元关系并进行对比，能够合理而准确地选择最合适的模型和参数，从而找出拟合效果最好的模型。
3. 第二问采用时间序列分析时考虑了长、中、短三个尺度并进行加权得到最终结果，兼顾历史影响与近期规律，具有说服力。
4. 第二问合理地运用目标规划模型，深度挖掘成本定价、用户购买、库存影响、损耗与打折等因素，提出多个限制条件，真实准确地反映实际情况。
5. 第二问最后对人为设定的超参数进行了灵敏度分析，更加科学合理。
6. 第三问采用粒子群算法对非线性规划模型进行求解，能够适用于本文的连续优化问题。

10.2 模型的缺点

1. 时间序列模型的参考数据太少，规律性不强，所得的预测值具有误差。

2. 目标规划模型将多个任务的重要程度进行人为主观赋权，有陷入局部最优解的可能。
3. 因缺少相关数据，未考虑运输成本、地理位置、政策、天气等因素对模型的影响，使模型不能完全模拟实际情况。

10.3 模型的改进和推广

1. 可将该模型应用到其他企业的选购定价决策以及政府对于公共设施的合理配置方面，能够考虑多方因素，实现多目标规划与决策，有利于企业优化资源配置并节省成本。
2. 时间序列预测模型可以应用到如经济、能源、环境等方面，有助于提前进行资源配置，帮助相关人员进行决策。
3. XGBoost 回归模型在使用之前可以进行预训练，从而提升预测精度。
4. 问题三采用合理的方式如询问专家和深入调研确定多个目标的优先级后，继续结合目标规划问题建立模型。

参考文献

[1] 杨璐, 陈彦如, 杨洁. 考虑缺货和产品缺陷的联合补货模型 [J]. 管理工程学报,2018,32(04):195-203.DOI:10.13587/j.cnki.jieem.2018.04.024.

[2] 司守奎.《数学建模算法与应用》(第三版). 国防工业出版社

[3] 蔡明、孙杰.《三种机器学习算法在回归应用中的对比分析》. 湖北省气象信息与技术保障中心. 智能计算机与应用. 2022,12(08)

[4] 王思睿, 王林, 彭璐等. 基于 Lipschitz 连续性的异质品联合补货-配送协同优化研究 [J/OL]. 中国管理科学:1-12[2023-09-10].DOI:10.16381/j.cnki.issn1003-207x.2021.1101.

[5] 田俊峰, 孙西秀, 杨梅. 考虑需求与价格——质量相关的补货与定价联合决策广义模型 [J]. 物流工程与管理,2014,36(05):188-192.

[6] 赵玲, 刘志学. 考虑顾客退货和固定成本的联合补货与定价策略 [J]. 运筹与管理,2022,31(06):105-110+124.

[7] 康怀飞. 考虑消费者策略行为和预期后悔的零售商定价与库存决策 [D]. 西南交通大学,2022.DOI:10.27414/d.cnki.gxnju.2022.000889.

[8] 康莎. 不同需求特征下生鲜农产品双渠道销售定价与库存补货联合决策研究 [D]. 重庆交通大学,2022.DOI:10.27671/d.cnki.gcjtc.2022.000130.

[9] 彭志强. 面向策略性消费者的动态定价及应对机制研究 [D]. 重庆大学,2010.

[10] 卢静. 生鲜农产品的库存控制及动态定价研究 [D]. 天津大学,2019.DOI:10.27356/d.cnki.gtjdu.2019.000182.

[11] 陈佳. 预售模式下生鲜农产品的定价、订货与保鲜联合决策 [D]. 西南交通大学,2022.DOI:10.27414/d.cnki.gxnju.2022.002554.

[12] 宋孟书. 亏损规避零售商在二次补货合同下的供应链渠道协调 [D]. 北京邮电大学,2020.DOI:10.26969/d.cnki.gbydu.2020.002641.

附录 A　问题三求解结果

单品编码	定价（元/kg）	补货量（kg）	商超收益（元）
102900011016701	6.32	41.01	110.51
102900011030059	9.95	64.99	373.33
102900005115786	4.97	30.75	74.19
102900005116714	13.63	37.15	200.96
102900005115250	21.25	45.15	224.77
102900011031100	9.89	27.11	189.41
102900011030097	6.76	54.22	198.84
102900005116257	7.01	16.77	51.57
102900011000328	11.90	24.78	74.87
102900005115762	4.13	9.41	19.11
106949711300259	2.86	98.31	136.41
102900005115946	9.00	5.43	30.36
102900011008164	5.60	8.00	20.15
102900005116899	14.03	22.03	94.05
102900011023464	6.38	7.24	23.93
102900005119975	4.74	41.24	88.75
102900011032022	6.43	30.36	130.28
102900011032251	13.35	17.12	154.25
102900051000944	29.70	4.46	39.28
102900005118831	6.63	15.52	30.32
102900011022764	17.00	6.24	58.70
102900011034330	5.02	25.45	64.25
102900005115823	7.77	12.13	38.60
102900011035078	44.19	2.79	78.19
106971533450003	3.00	14.32	14.16
102900005118824	14.66	3.26	7.32
102900011034026	23.87	7.03	92.14
102900011032848	3.91	18.42	26.83
102900011030110	12.63	9.63	73.56
102900005115779	8.20	50.35	138.10
102900005116509	5.24	13.41	22.72
102900011032343	26.34	3.53	48.47
102900011001691	15.01	3.47	19.83
总收益（元）			2948.18

附录 B　各类别的销售量统计程序

```
# 读取原始数据
df = pd.read_excel("C:/Users/28678/Desktop/时间序列.xlsx")
# 获取唯一的类别列表
categories = df['类别'].unique()
# 对每个类别进行循环处理
for category in categories:
    # 根据类别筛选数据
    filtered_data = df[df['类别'] == category]
    # 使用groupby和agg函数进行统计
    result_df = filtered_data.groupby(['类别', '销售时间']).agg({'销售量': 'sum'}).reset_index()
    # 保存结果到新的excel表格
    file_name = f'{category}_统计结果.xlsx' # 根据类别生成文件名
    result_df.to_excel(file_name, index=False)
```

附录 C　绘制小提琴图程序

```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import os

# 获取当前文件夹的父文件夹路径
current_directory = os.getcwd()
parent_directory = os.path.dirname(current_directory)

xlsx_file_path = os.path.join(parent_directory, '1.xlsx')

df = pd.read_excel(xlsx_file_path, header=None)
data = df.to_numpy()

print(data[0])

# 绘制小提琴图
plt.figure(figsize=(6, 4))
ax = sns.violinplot(data=data, inner="quart", width=0.4)

plt.xlabel('1')
plt.ylim([0, 30000])
plt.savefig('1.1.png')
plt.show()


# 获取当前文件夹的父文件夹路径
```

```
current_directory = os.getcwd()
parent_directory = os.path.dirname(current_directory)

xlsx_file_path = os.path.join(parent_directory, '2.xlsx')

df = pd.read_excel(xlsx_file_path, header=None)
data = df.to_numpy()


# 绘制小提琴图
plt.figure(figsize=(6, 4))
ax = sns.violinplot(data=data, inner="quart", width=0.4)


plt.ylim([0, 50000]);
plt.savefig('2.1.png')
plt.show()
```

附录 D　Spearman 相关系数求解程序

```
import pandas as pd
import numpy as np
import os
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats

# 获取当前文件夹的父文件夹路径
current_directory = os.getcwd()
parent_directory = os.path.dirname(current_directory)

xlsx_file_path = os.path.join(parent_directory, 'everyday_pinlei.xlsx')

df = pd.read_excel(xlsx_file_path, header=None)
data = df.to_numpy()
data = data[0:,1:]

data = data.astype('float')
mean = np.mean(data, axis=0)
std = np.std(data, axis=0)

# 使用均值和标准差进行归一化
normalized_data = (data - mean) / std

print(normalized_data[0])
```

```
# 计算Spearman秩相关系数
spearman_corr = df.corr(method='spearman')

print("\nSpearman相关系数：")
print(spearman_corr)

plt.figure(figsize=(10, 8)) # 设置图形大小
sns.heatmap(spearman_corr, annot=True, cmap='coolwarm', linewidths=.5)

plt.title('Spearman') # 设置标题
plt.savefig('Spearman.png')
plt.show()
```

附录 E　时间序列分析程序

```
data = pd.DataFrame({
    "A": numpy.random.random(size=20)
})

result = statistical_model_analysis.arima_analysis(data=data, p=0, d=0, q=0, forecast_num=10)
print(result)
```

附录 F　筛选特定时段的数据程序

```
#4、筛选特定时间的销售数据（批发价、售价、销售量）

# 读取销售信息表格和成本价表格
sales_data = pd.read_excel("C:/Users/28678/Desktop/时间序列.xlsx")
cost_data = pd.read_excel("C:/Users/28678/Desktop/成本价.xlsx")

# 合并销售信息表格和成本价表格
merged_data = sales_data.merge(cost_data, on=['销售时间','编号', '类别'], how='inner')
merged_data['销售总价']=merged_data['销售价']*merged_data['销售量']
# 统计各单品总销售量
total_quantity = merged_data.groupby(['类别', '编号'])['销售量'].sum().reset_index()
# 统计各单品总销售价
total_sales_price = merged_data.groupby(['类别', '编号'])['销售总价'].sum().reset_index()
# 统计各单品总成本价
total_cost_price = merged_data.groupby(['类别', '编号'])['成本价'].sum().reset_index()
# 合并各单品统计结果
merged_results = total_quantity.merge(total_sales_price, on=['类别', '编号'], how='inner')
```

```
merged_results = merged_results.merge(total_cost_price, on=['类别', '编号'], how='inner')
# # 计算各单品盈利率
# merged_results['盈利率'] = (merged_results['销售价'] - merged_results['成本价']) /
    merged_results['销售价'] * 100
# merged_results['利润'] = (merged_results['销售价'] - merged_results['成本价'])
# start_date = datetime(merged_results['销售时间'].dt.year.min(), 6, 1)
# end_date = datetime(merged_results['销售时间'].dt.year.max(), 7, 1)
# mask = (merged_results['销售时间'] >= start_date) & (merged_results['销售时间'] <= end_date)
# df_filtered_sales = merged_results[mask]
# df_filtered_sales.to_excel('steph.xlsx', index=False)
# # 将结果按类别分别导出为六个表格
categories = merged_results['类别'].unique()
for category in categories:
    category_data = merged_results[merged_results['类别'] == category]
    category_data.to_excel(f'{category}new.xlsx', index=False)
```

附录 G　统计不同年份的单品和类别的利润等信息程序

```
import pandas as pd
# 读取第一个xlsx文件（销售数据）
sales_data = pd.read_excel("C:/Users/28678/Desktop/打折销售.xlsx")
# 读取第二个xlsx文件（成本数据）
cost_data = pd.read_excel("C:/Users/28678/Desktop/成本价.xlsx")
# 提取所需列的数据
sales_data = sales_data[['销售时间', '单品编号','销售量', '销售单价', '类别']]
cost_data = cost_data[['单品编号', '成本价']]
# 创建新列，提取销售时间的月份和日期部分
sales_data['月份'] = pd.to_datetime(sales_data['销售时间']).dt.month
sales_data['日期'] = pd.to_datetime(sales_data['销售时间']).dt.day
sales_data['年份'] = pd.to_datetime(sales_data['销售时间']).dt.year
# 过滤出7月1日到7月7日的销售数据
sales_data = sales_data[(sales_data['月份'] == 7)and (sales_data['日期'].between(1, 7)and
    ((sales_data['年份']==2020)or(sales_data['年份']==2021)))]
# 合并销售数据和成本数据
merged_data = pd.merge(sales_data, cost_data, on='单品编号')
# 计算利润（销售单价 - 成本价）
merged_data['利润'] = merged_data['销售量'] * (merged_data['销售单价'] - merged_data['成本价'])
# 按年份和大类进行利润求和
profit_sum = merged_data.groupby(['月份', '类别'])['利润'].sum().unstack()
# 创建新的Excel文件并将利润数据写入
profit_sum.to_excel('profit_summary.xlsx')
print("利润数据已成功导入到 profit_summary.xlsx 文件中。")
```

附录 H　寻找日变化规律并筛选 top33 程序

```
import pandas as pd
from openpyxl import Workbook
# 读取xlsx文件内容
df = pd.read_excel("C:/Users/28678/Desktop/逐小时销售量.xlsx")
# 按小时进行分组并计算每个单品的总销售量
grouped_df = df.groupby(['编号', '小时']).sum().reset_index()
# 计算一天内每个单品的总销售量
daily_sales = grouped_df.groupby('编号')['销售量'].sum().reset_index()
# 按销售量降序排序
daily_sales_sorted = daily_sales.sort_values(by='销售量', ascending=False)
# 获取前33个单品的销售量和排名
top_33 = daily_sales_sorted.head(33)
# 创建新的excel表
output_filename = 'new33.xlsx'
workbook = Workbook()
worksheet = workbook.active
# 添加表头
worksheet.append(['排名', '编号', '销售量'])
# 添加数据
for index, row in top_33.iterrows():
    ranking = index + 1
    item_number = row['编号']
    sales = row['销售量']
    worksheet.append([ranking, item_number, sales])
# 保存excel表
workbook.save(output_filename)
print(f"已将统计结果保存到 {output_filename} 文件中！")
# 读取top33.xlsx文件和损耗率文件（假设为loss.xlsx）
df_top33 = pd.read_excel("C:/Users/28678/Desktop/new33.xlsx")
df_loss = pd.read_excel("C:/Users/28678/Desktop/损耗1.xlsx")
# 合并两个数据框，根据单品编号进行匹配
df_merged = pd.merge(df_top33, df_loss, on='编号', how='left')
# 将合并后的数据保存回top33.xlsx文件
df_merged.to_excel("C:/Users/28678/Desktop/new33.xlsx", index=False)
```

附录 I　绘制不同品类时间上分布规律折线图

```
# -*- coding = utf-8 -*-
import matplotlib.pyplot as plt
import pandas as pd
plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号
plt.rcParams['font.sans-serif'] = ['SimHei'] # 显示汉字
```

```
path='data/National_Competition/everyday.txt'
df = pd.read_csv(path, header=None, names=['time', '1', '2', '3', '4', '5', '6'])
df['time'] = pd.to_datetime(df['time'])
df = df.set_index('time')
# 绘制每天的折线图
x= df.index
y1= df['1']
y2= df['2']
y3= df['3']
y4= df['4']
y5= df['5']
y6= df['6']
fig2 = plt.figure(figsize=(18, 9))
plt.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=None, hspace=0.7)
# 左侧坐标
ax1 = fig2.add_subplot(111)
ax1.plot(x, y1, linestyle='-', color='r', label='1')
ax1.plot(x, y2, linestyle='--', color='g', label='2')
ax1.plot(x, y3, linestyle='-.', color='y', label='3')
ax1.plot(x, y4, linestyle=':', color='b', label='4')
ax1.plot(x, y5, linestyle='solid', color='c', label='5')
ax1.plot(x, y6, linestyle='dashed', color='m', label='6')
ax1.set_ylabel('销售量 (kg) ')
ax1.set_xlabel('时间 (day) ')
plt.legend(["种类1:花叶类", "种类2:花菜类", "种类3:水生根茎类",
    "种类4:茄类","种类5:辣椒类","种类6:食用菌类"])
plt.show()
# 以每月为单位对数据进行分组和采样
monthly_data = df.resample('M').sum()
# 将结果保存到一个csv文件中
monthly_data.to_csv('data/National_Competition/output.csv')
# 绘制每个月的折线图
# -*- coding = utf-8 -*-
import matplotlib.pyplot as plt
import pandas as pd
plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号
plt.rcParams['font.sans-serif'] = ['SimHei'] # 显示汉字
path='data/National_Competition/everymonth.txt'
df = pd.read_csv(path, header=None, names=['time', '1', '2', '3', '4', '5', '6'])
df['time'] = pd.to_datetime(df['time'])
df = df.set_index('time')
# 绘制每天的折线图
x= df.index
y1= df['1']
y2= df['2']
y3= df['3']
y4= df['4']
```

```
y5= df['5']
y6= df['6']
fig2 = plt.figure(figsize=(18, 9))
plt.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=None, hspace=0.7)
# 左侧坐标
ax1 = fig2.add_subplot(111)
ax1.plot(x, y1, linestyle='-', color='r', label='1')
ax1.plot(x, y2, linestyle='--', color='g', label='2')
ax1.plot(x, y3, linestyle='-.', color='y', label='3')
ax1.plot(x, y4, linestyle=':', color='b', label='4')
ax1.plot(x, y5, linestyle='solid', color='c', label='5')
ax1.plot(x, y6, linestyle='dashed', color='m', label='6')
ax1.set_ylabel('销售量 (kg) ')
ax1.set_xlabel('时间 (month) ')
plt.legend(["种类1:花叶类", "种类2:花菜类", "种类3:水生根茎类",
    "种类4:茄类","种类5:辣椒类","种类6:食用菌类"])
plt.show()
```

附录 J　目标规模模型求解程序

```
clc
clear
X = optimvar('X', 7, 6, 'LowerBound', 0);
P = optimvar('P', 7, 6, 'LowerBound', 0);
S = optimvar('S', 7, 6, 'LowerBound', 0);
dp = optimvar('dp', 3, 7, 'LowerBound', 0);
dm = optimvar('dm', 3, 7, 'LowerBound', 0);
p = optimproblem('ObjectiveSense', 'min');
b = xlsread('data\各类成本价.xlsx');
Q = xlsread('data\退货率.xlsx');
H = xlsread('data\历史最大销售量.xlsx');
data = xlsread('data\销售量和利润率.xlsx')
M = zeros(1, 7);
M(:) = 10147.93
% 提取销售量和利润率用来训练
S0 = data(:, 1);
n = data(:, 2);
% 使用xgboost进行训练和预测
% 创建DMatrix
dtrain = xgboost.DMatrix(S0, 'label', n);
% 训练模型
bst = xgboost.train(param, dtrain);
preds = bst.predict(dtrain);
p.Constraints.con4 = preds .* (1 + n) == P;
```

```
for j = 1:7
    con1 = 0;
    for i = 1:6
        con1 = con1 + (S(j, i) * (1 + Q(j, i))) * P(j, i);
    end
    con1 = con1 - (X(j, :) * b') + dm(1, j) - dp(1, j) == M(j);
    p.Constraints.(['con1_' num2str(j)]) = con1;
end

for j = 1:7
    con2 = 0;
    for i = 1:6
        con2 = con2 + X(j,i) - S(j,i) * (1 + Q(j,i)) ;
    end
    con2 = con2 + dm(2,j) - dp(2,j) == 0;
    p.Constraints.(['con2_' num2str(j)]) = con2;
end
for j = 1:7
    con3 = 0;
    for i = 1:6
        con3 = con3 + X(j,i) - H(j,i) ;
    end
    con3 = con3 + dm(3,j) - dp(3,j) == 0;
    p.Constraints.(['con3_' num2str(j)]) = con3;
end
goal = 100000*zeros(3,7);
mobj = [dm(1, :); 3 * dm(2, :) + 7 * dp(2, :); dp(3, :)];
for i = 1:3
    p.Constraints.cons2 = [mobj(i,:) <= goal(i,:)];
    p.Objective = mobj(i);
    [sx, fval] = solve(p);
    fprintf('第%d级目标计算结果如下: \n', i)
    fval
    sx.X % 输出 X
    sx.P % 输出 P
    sx.dp % 输出 dp
    sx.dm % 输出 dm
end
```