

蔬菜类商品销售数据驱动的定价与补货决策方法

摘要

本文研究蔬菜类商品的自动定价与补货决策问题，旨在通过对不同品类及不同单品的历史销售信息进行量化分析，进而提供给终端经销商有效的补货数量与定价策略，在深入探讨不同品类与单品间相互关系的基础上，实现商超收益的最大化。

问题一的求解更多依靠于良好的数据分类与处理。在模型建立方面，对于问题二，本文建立了 SARIMA 预测模型与规划模型，通过遗传算法完成在利润最大化时未来一周的补货与定价决策的求解；对于问题三，根据题目要求增加时间与补货量的约束信息，建立指标体系运用层次分析法与 TOPSIS 法筛选出优选可售单品，而后同问题二模型；对于问题四，在查阅大量文献的基础上，归纳出四类可用于优化决策模型的相关数据。

针对问题一：在对历史销售数据进行描述性统计分析后，选取利润率与批发价格两个指标，对众多蔬菜单品进行聚类分析，整合为 12 个代表性单品。对于各目标品类或单品分布规律的研究，本文采用时间序列分析，以季度为基本单位，构建其销售总量随季节变动的趋势图。对于其相互关系的研究，本文基于利润率与批发价格指标得到目标品类或单品的综合得分，在此基础上进行 Person 相关系数分析。

针对问题二：第一部分求解各品类销售总量与成本加成定价的关系，由于两者的直接关系不显著，本文选择将其转化为探究销售总量与利润率之间的关系。同问题一，分析各目标品类与单品的季节销售总量、季节利润率，进行回归分析。第二部分本文采用 SARIMA 预测模型、规划模型，利用附件所提供的数据实现对销售总量与批发价格的预测，运用遗传算法进行求解在利润最大化的前提下，给出 2023 年 7 月 1 日至 7 日的日补货量和定价策略，最终求得的最大利润为 8584.3054 元。

针对问题三：问题三在问题二的基础上增加了时间与补货总量的约束，同时将研究对象对焦为给定时间内的可售单品。对此，本文建立以销售总量、加权利润率及季节性销售评级为指标的评价体系，用以筛选出符合条件的目标单品。同时，为避免主观性对量化结果的影响，本文采用层次分析法与熵权法对各指标赋予权重，使用 TOPSIS 法进行量化，同问题二运用遗传算法求解出得到 7 月 1 日的补货与定价决策，最终求得的最大利润为 912.7051437 元。

针对问题四：本题聚焦于优化数据选择，查阅大量文献后，本文归纳总结出四类数据，分别是产品角度的产品变质率与存储时间的关系、供应链角度的进货和配送数据、市场条件角度的市场竞争和外部经济数据以及消费者角度的消费者反馈数据。

关键词：补货定价策略 时间序列预测 聚类分析 遗传算法

一、问题重述

1.1 问题背景

在现实生活中，由于商超所销售的蔬菜品种繁多、产地各异且易变质、保鲜期短，在不确定具体单品和进货价格的情况下进行进货交易时，为避免信息不对称或其他负的外部性所造成的潜在利润损失，经销商往往会根据已有的销售信息及需求状况对其补货数量做出评估，并据此决定相应的定价策略。补货量决定商超是否有机会获得更大的收益，而定价策略直接影响销售量，进而决定商超是否能避免货物滞销和经营亏损。

通常情况下，商超倾向于采用“成本加成定价”的方法对蔬菜进行定价，即按产品单位成本加上一定比例的利润制定产品价格，对于在运输过程中产生损耗的商品则进行打折销售。商超补货量的预测可以从供需两个方面来看，其需求量通常受到时间和季节的影响，其中 4-10 月的供应品种较为丰富；供给通常会参考各蔬菜品类的历史销售与需求情况。那么，如何有效利用历史信息做出定价与补货策略是亟待解决的关键问题。

1.2 问题要求

附件 1 提供了某商超经销的 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。基于上述背景和已知附件信息，我们需要建立数学模型解决以下问题：

问题一 分析蔬菜各品类及单品销售量的分布规律及相互关系。

问题二 以品类为单位做补货计划，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，最终要使得商超收益最大化。

问题三 考虑到蔬菜类商品的销售空间有限，在满足可售单品总数约 27-33 个，且各单品订购量满足最小陈列量 2.5 千克的要求，拟为商超进一步制定单品的补货计划。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，使得商超收益最大化。

问题四 为了更好地制定蔬菜商品的补货和定价决策，探究商超还需要采集哪些相关数据去更好地解决上述问题。

二、问题分析

2.1 对问题一的分析

针对问题一，在对不同蔬菜品类及单品的分布规律与相互关系进行分析时，需考虑到蔬菜类产品与交易时间密切相关的特殊属性。为避免目标单品数量过多而导致的数据

过于分散，同时要求最大化利用附件中所提供的产品价格类指标，故本文选取利润率与批发价格两个指标，对众多蔬菜单品进行聚类分析，整合为12个代表性单品。

结合题目及已有数据，本文在对品类与单品的分布规律进行研究时，采用时间序列分析，以季度为基本单位，构建各目标品类或单品的销售总量随季节变动趋势图及销售价格的变动箱线图。对于其相互关系的研究，本文选取基于利润率与批发价格指标拟合出目标品类或单品的综合得分指数，并在此基础上利用Person相关系数进行分析，得到系数热力图。

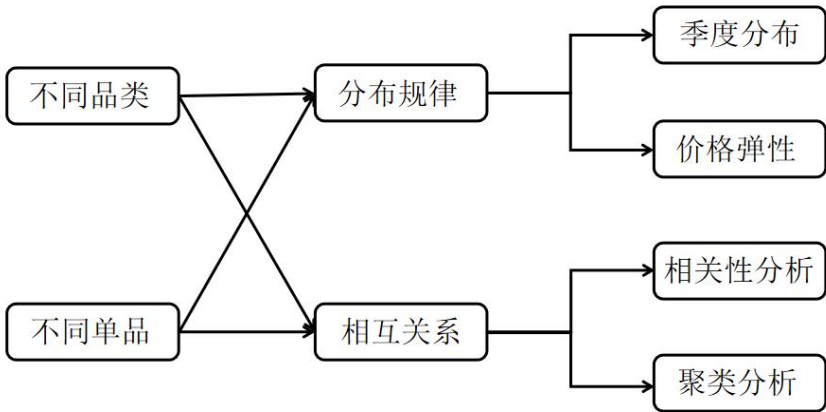
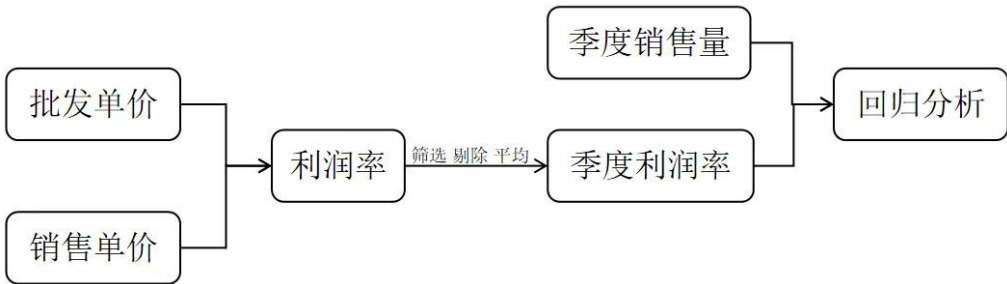


图 1 问题一解决流程图

2.2对问题二的分析

第一部分探讨对于各品类销售总量与成本加成定价的关系。“成本加成定价法”是指在成本的基础上加上一定比例的利润进行定价的方法，而增加的比率称为利润率。因此，销售总量与成本加成定价的关系可转化为其与利润率的关系。同问题一，本题则仍以季度为单位对数据进行预处理，包括各品类季度销售总量和季度利润率。其中，季度利润率的处理较为复杂，不可直接取平均值，存在较大误差，我们综合考虑了利润率出现频数和占比，对其进行筛选，剔除异常值和偏离值，对经过处理后的利润率求均值，得出季度均利润率，最后对两者进行回归分析。

第二部分关于补货和定价决策，本文先采用SARIMA预测模型根据2023年6月份数据对7月1日至7日的销售总量进行预测，再利用规划模型得到利润表达式，运用遗传算法对销售单价和补货量进行求解，在满足约束条件且利润最大的前提下，求解出2023年7月1日至7日的日补货量和定价策略。



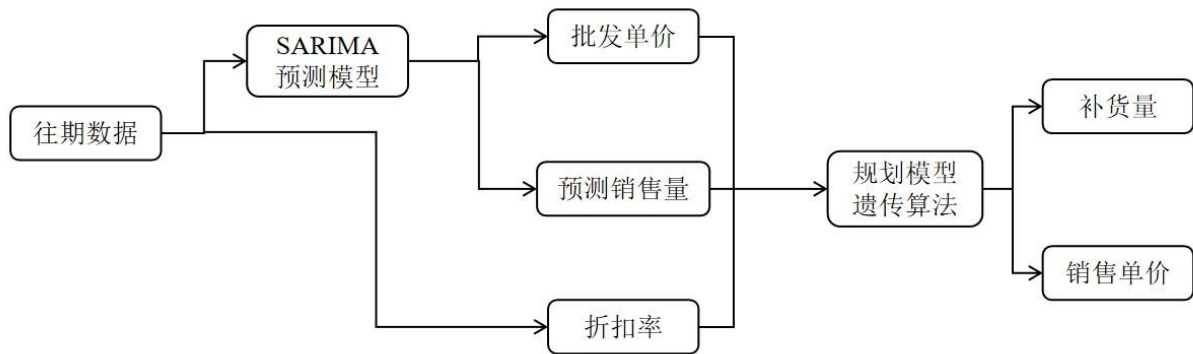


图 2 问题二解决流程图

2.3对问题三的分析

问题三的定价及补货决策与问题二相似，区别在于问题三在二的基础上增加了时间与补货总量的约束，同时将研究对象聚焦为给定时间内的可售单品，问题二所建立的时间序列预测模型及规划模型同样适用。对此，在遵循系统性、可持续性、可操作性原则的基础上，本文建立以销售总量、加权利润率及季节性销售评级为指标的评价体系，用以筛选出符合设定条件的目标单品。

为避免主观性对量化结果的影响，本文采用层次分析法对各指标赋予权重，并使用TOPSIS进行量化，对各单品的可售程度进行综合评估。在此基础上，为满足商超利润最大化的目标，运用遗传算法求解出目标函数得到其补货与定价决策。

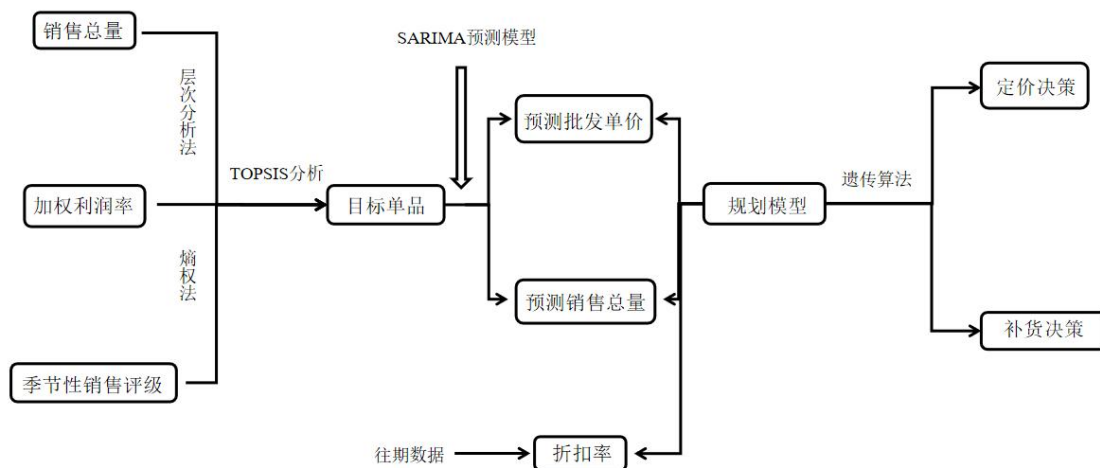


图 3 问题三解决流程图

2.4对问题四的分析

为总结优化补货和定价决策模型所需的数据，本文在搜集并筛选出主题相关度较高的大量文献后，对各权威文献中所采用的数据进行分类总结，得到不同视角的四类数据，分别是产品角度的产品变质率与存储时间的关系、供应链角度的进货和配送数据、市场条件角度的市场竞争和外部经济数据以及消费者角度的消费者反馈数据。

三、模型假设与符号说明

3.1 模型假设

- 3.1.1 终端销售商的补货数量没有限制，且销售过程中不存在缺货现象。
- 3.1.2 蔬菜类产品具有易变质的特征，当日未售出的商品隔日无法售出。
- 3.1.3 折扣价格只适用于已损耗产品，不影响未售出商品。
- 3.1.4 销售数据在短期内相对稳定，不受突发事件的影响。
- 3.1.5 假设消费者偏好不受定价及价格决策的影响。
- 3.1.6 忽略市场内其他参与者的竞争与合作行为，认为其决策不对主体产生影响。

3.2 符号说明

符号	意义	单位
P_{ij}	某日某品类销售单价	元/千克
P_{iq}	某日某单品销售单价	元/千克
P_{mij}	某日某品类加权最大销售单价	元/千克
P_{tij}	某日某品类批发单价	元/千克
P_{tiq}	某日某单品批发单价	元/千克
P_{kij}	某日某品类折扣单价	元/千克
Q_{ij}	某日某品类销售量	千克
Q_{tij}	某日某品类补货量	千克
Q_{iq}	某日某单品销售量	千克
Q_m	某周某单品销售量	千克
Q	某周单品销售总量	千克
K_j	某品类折扣率	百分比
S_j	某品类均损耗率	百分比
r_{ij}	利润率	百分比
W	利润	元
W_1	正常品收益	元

W_2	损耗品收益	元
C	总成本	元
Q_s	销售总量	千克
W_p	加权利润率	百分比
R	季节性销售评级	分
Z_i^+	每列指标的最大值	—
Z_i^-	每列指标的最小值	—
D_i^+	第 i 个蔬菜单品与理想目标的距离	—
D_i^-	第 i 个蔬菜单品与不理想目标的距离	—
S_i	第 i 个蔬菜单品的综合得分指数	—
X	指标矩阵	—
x_{ij}	第 j 项指标下第 i 个单品的具体值	—
P_{ij}	第 j 项指标下第 i 个单品占该指标的比重	百分比
e_j	第 j 项指标的熵值	—
W_j	指标熵权权重	百分比

四、模型的建立与求解

4.1 问题一模型的建立与求解

4.1.1 数据的统计分析

(1) 数据预处理

由于在现实生活中，蔬菜销售种类繁多、销售单量多，在进行相关分析时，做好数据处理工作无疑是重中之重。数据清洗作为数据预处理的重要环节之一，旨在处理原始数据中的噪声、错误和不一致性，以确保数据的质量和可靠性。数据清洗的一般步骤为检查缺失数据、对异常值进行处理、删除重复记录情况，最后进行归一化处理。

本文主要对数据异常值进行剔除，通过计算指标属性的平均加减3倍标准差的范围，识别出超出范围的异常值。对于这些异常值，我们选择将其视为无效数据，并进行了删除或修正的操作，以确保数据的准确性。此后，我们对清洗后的数据进行了归一化处理，将不同尺度的数据转化为统一尺度的过程，以便更好地支持模型的建立和分析。通过以上步骤，我们得到了一组准确且可靠的数据，为接下来的数据分析和建模提供可靠基础。

(2) 单品的聚类分析

聚类分析法是对具有不同程度相似性的多个变量进行分类统计的一种方法，可以直观且综合的利用多个变量信息进行样本分类，并且可依靠图示进行清晰反映。根据附件一所示，本题共计251个蔬菜单品，而目标单品数量过多易导致数据过于分散，故采取聚类分析可有效避免该问题，本文选取的是具有显著代表性的利润率和批发价格两个指标。具体聚类结果如下表所示。

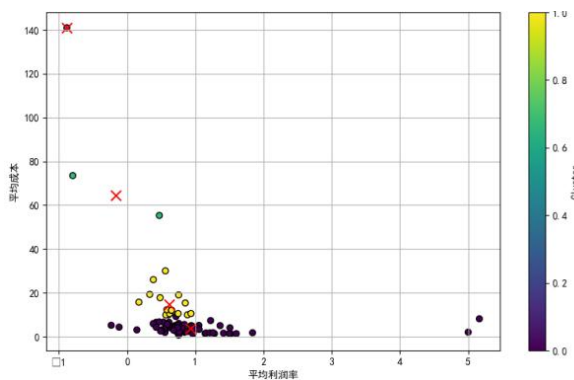


图 4 花叶类聚类散点图

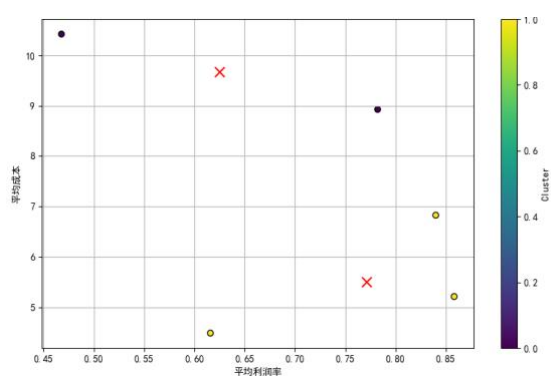


图 5 花菜类聚类散点图

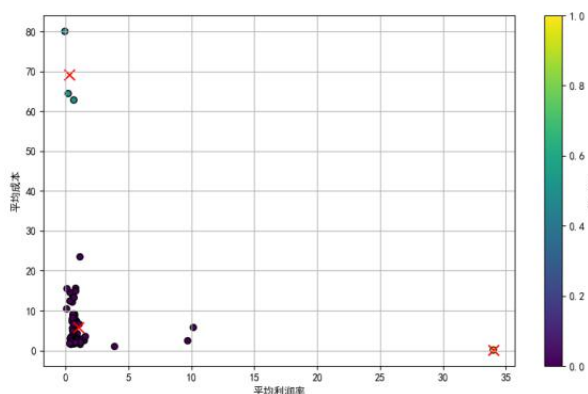


图 6 食用菌聚类散点图

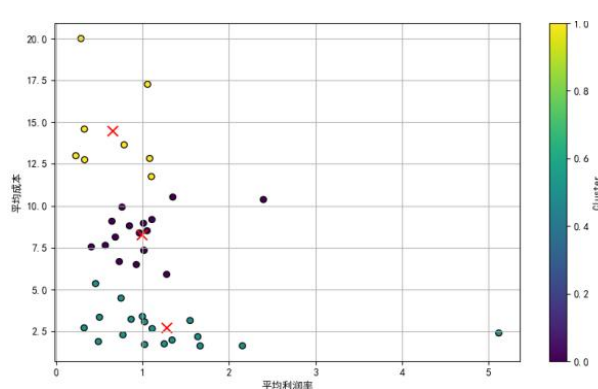


图 7 辣椒类聚类散点图

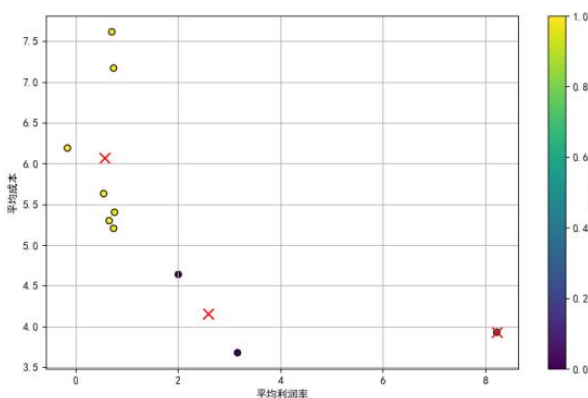


图 8 茄类聚类散点图

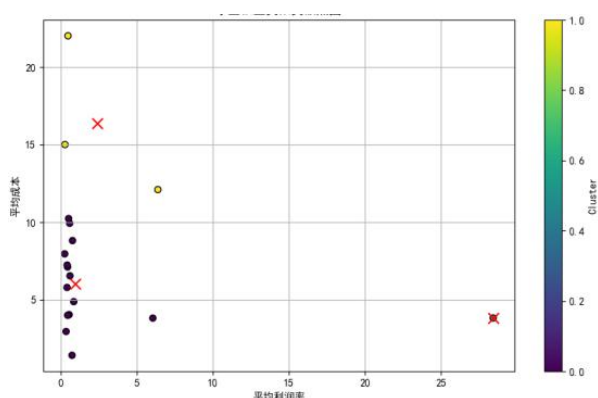


图 9 水生根茎类聚类散点图

表1 蔬菜单品聚类分析表

聚类结果	
①花叶类共四个聚类中心； ②距离聚类中心最近的点依次是：“大白菜秧”、“洪山菜薹珍品手提袋”、“洪山菜薹莲藕拼装礼盒”、“紫苏”。	①花菜类共两个聚类中心； ②距离聚类中心最近的点依次是：“紫白菜(1)”、“枝江青梗散花”。
①辣椒类共三个聚类中心； ②距离聚类中心最近的点依次是：“小皱皮”、“红杭椒(份)”、“水果辣椒”。	①食用菌共三个聚类中心； ②距离聚类中心最近的点依次是：“平菇”、“黑牛肝菌”、“金针菇(份)”。
①茄类共三个聚类中心； ②距离聚类中心最近的点依次是：“紫圆茄”、“青茄子(2)”、“紫茄子(2)”。	①水生根茎类共三个聚类中心； ②距离聚类中心最近的点依次是：“净藕(1)”、“野藕(2)”、“鲜藕带(袋)”。

4.1.2 目标品类或单品的分布规律

(1) 各品类销售量的分布规律

根据附件所提供的2020年7月-2023年6月整三年的销售数据，考虑到蔬菜产品受季节变换的影响较大，故本文将时间划分为12个季度，在此基础上对各品类的销售总量分布情况进行体现。具体结果如下图所示。

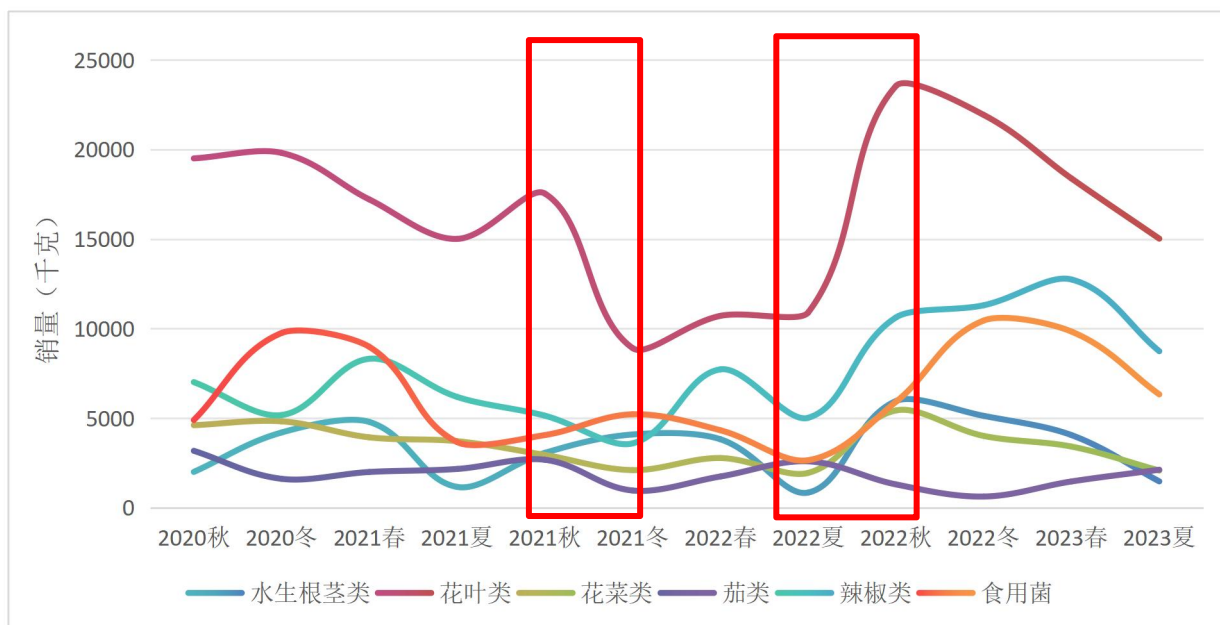


图10 时间序列销售总量折线图

由图示可得，各蔬菜单品的销量于2021年秋冬季出现大幅下降的趋势，并且持续相当长一段时期，直至2022年夏季后出现销售总量的“回春”，甚至超过2020年所示的正常时期。结合现实情境，较大可能是由于受到新冠肺炎疫情的影响。同时，也可观察到，步入2023年后相应品类的销售量又出现了下降趋势，预期趋近于往年的正常水平。

各季度销售总量变化趋势图如上所示，易得各蔬菜品类在相同季节售出量基本相同，

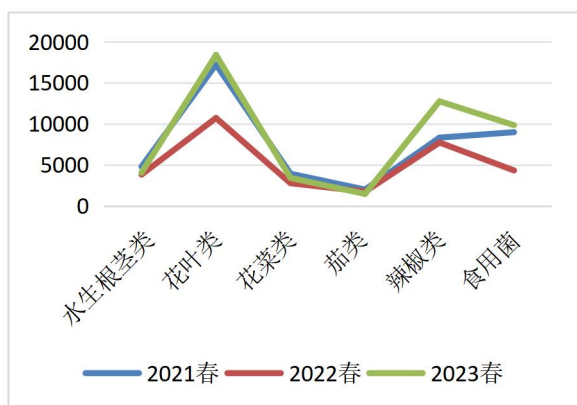


图 11 春季销售总量折线图

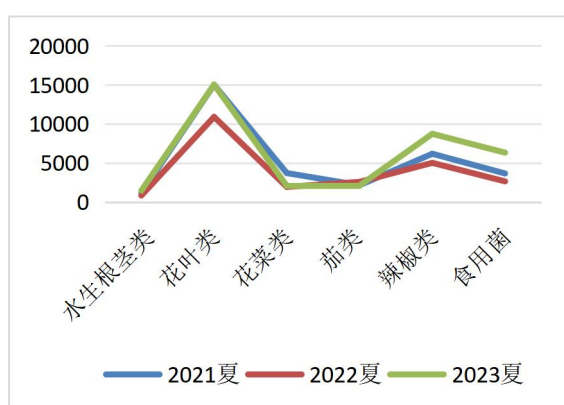


图 12 夏季销售总量折线图

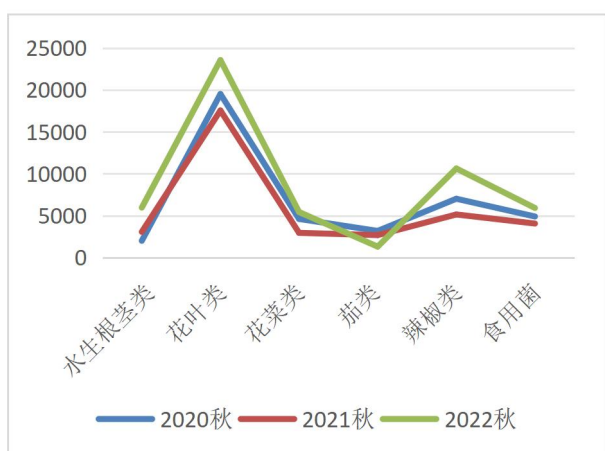


图 13 秋季销售总量折线图

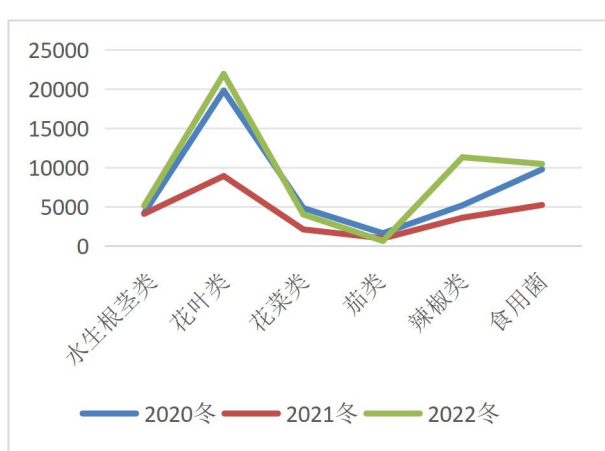


图 14 冬季销售总量折线图

再一次佐证蔬菜类产品受季节影响较大的基本规律。那么，在进行决策制定时，参考近年来相同时节的数据十分有必要。

（2）各代表性单品的分布规律

在将蔬菜单品聚类为代表性单品后，同样采取相同的办法对目标品类或单品的销售量随季节的变化趋势用图示进行呈现。具体分布规律图如下所示。

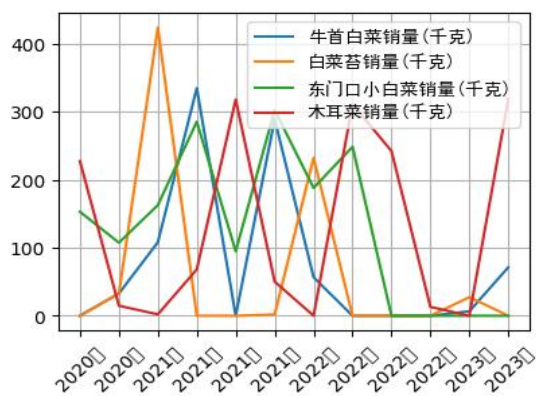


图 15 花叶类单品季度销量图

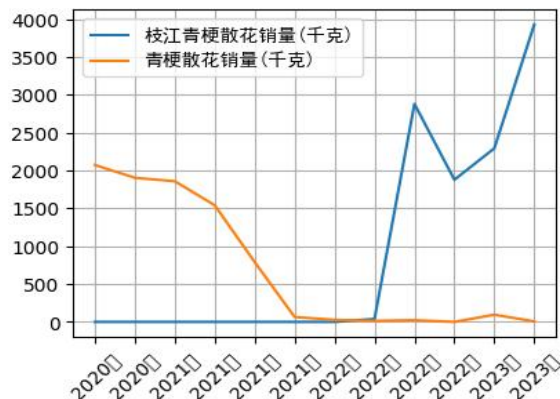


图 16 花菜类单品季度销量图



图 17 食用菌单品季度销量图

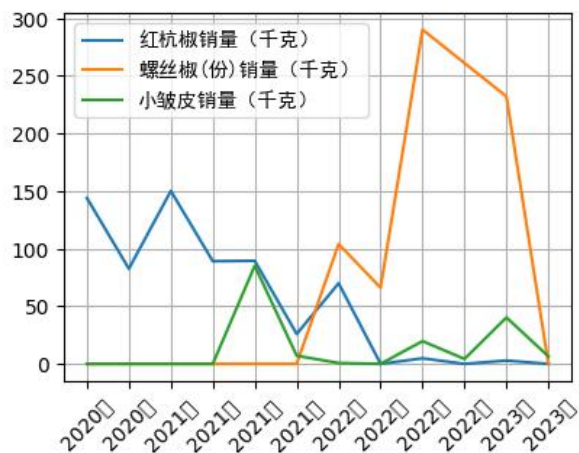


图 18 辣椒类单品季度销量图

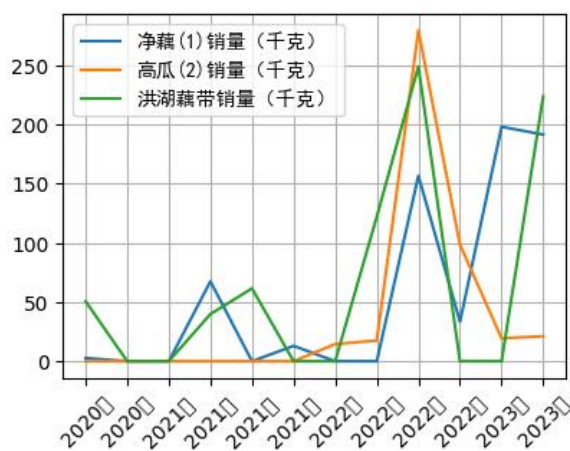


图 19 水生根茎类单品季度销量图

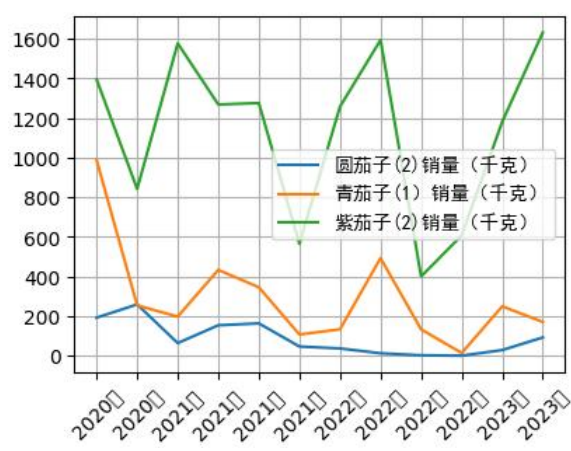


图 20 茄类单品季度销量图

由上图可得，花叶类、茄类单品的销量随季节变换的幅度非常明显、且更加具有规律性；水生根茎类、花菜类的销量几乎不随季节变换而有过多变化，只有在特定年限有突出的销量，考虑与其当年的生产水平相关；食用菌、辣椒类处于中立状态，较为显著。

4.1.3 目标品类或单品的相互关系

(1) 各品类的相互关系

在对各蔬菜品类进行相互关系分析时，本文根据进行单品聚类分析的两个指标——利润率与批发价格拟合出目标品类或单品的综合得分指数，并在此基础上利用Person相关系数进行分析，得到系数热力图。具体结果如下图所示。

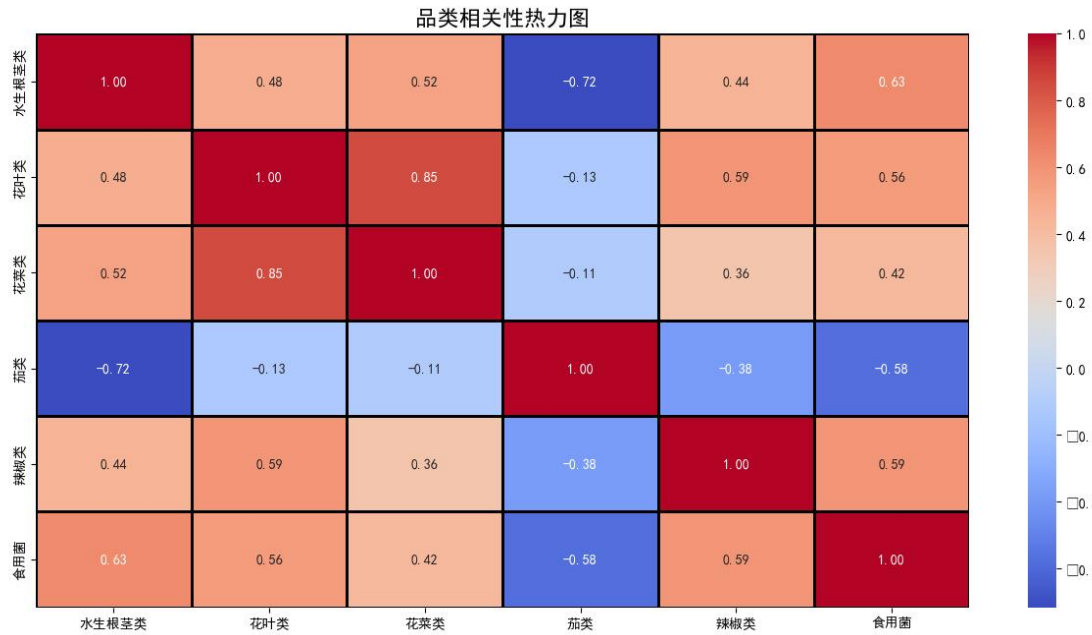


图21 蔬菜品类Person相关系数热力图

由上图分析可得，食用菌与水生根茎类（0.713）、花菜类与花叶类（0.832）具有较强的正相关关系，而茄类与水生根茎类（-0.769）具有较强的负相关关系。一定程度上，反应消费者在对应的两个品类的选择中具有较强的关联关系，故在实际销售中，可以采取捆绑销售、搭配销售等促销方式进行，用以提高收益。

（2）各单品的相互关系

经过对蔬菜单品的聚类分析，从中筛选出了18个具有代表性的单品。所有单品之间由于利润率、批发价格的相似性而被无形划分为不同的类别，并以每个子集内最具代表性的产品替代描述。由4.1.1（2）中聚类分析的图示表明，存在一些特殊的蔬菜单品被单独分到一个聚类簇中，为保证数据的严谨、完整，故本文选择保留该类产品。总的来说，聚类分析提供了一种有效识别相似特征蔬菜单品的方法，使得我们进一步了解它们的区别与联系，有利于我们更好地了解蔬菜特性，为蔬菜的种植、采购和消费等提供更有针对性的指导和决策依据。

4.2问题二模型的建立与求解

4.2.1各品类销售总量与成本加成定价的关系

（1）求解思路

由“成本加成定价法”的概念可得销售总量与成本加成定价的关系可转化为其与利润率的关系。其中，利润率是由销售价格与批发价格的差值除以批发价格构成的。本题以品类为单位进行统计，因此要对销售记录中各品类下的单品销售价格和批发价格进行加权平均计算，以单品销售量占品类总销售量的比值作为权重。由问题一可知，产品销量与季度相关，本题则仍以季度为单位，对数据进行预处理，得到各品类季度销售总量和季度利润率，最后对两者进行回归分析。

$$r_{ij} = (P_{ij} - P_{tij})/P_{tij} \quad (1)$$

其中,i表示天数, j表示品类, r_{ij} 为某日某品类的利润率, P_{ij} 表示某日某品类销售单价, P_{tij} 表示某日某品类批发单价。

(2) 回归分析结果

利用stata软件对数据预处理得到的各品类销售量和利润率进行回归分析, 以销售量为因变量, 利润率为自变量, 得到6个品类各自的回归模型以及其拟合优度、P值、系数等(如下表)。由表中数据可知, 各品类销售量与利润率之间呈现负相关关系, 即两者反向变动, 利润率升高, 成本加成定价升高, 销售量下降。由此推出各品类销售量与成本加成定价负相关。

但是各品类回归模型的拟合优度均在60%及以下, 其中茄类模型的拟合优度只有34%。通常情况下, 一个回归模型的拟合优度应该在80%以上才被认为是比较好且可利用的。因此, 考虑到模型的准确性, 后续模型将不再采用此处两者之间的线性回归关系。

表2 各品类回归模型情况

品类名称	拟合优度	P 值	系数
水生根茎类	0.4745	0.0132	-8158.348
花叶类	0.5747	0.0043	-69853.65
花菜类	0.4695	0.0139	-12314.06
茄类	0.3454	0.0445	-5912.614
辣椒类	0.5271	0.0075	-23900.34
食用菌	0.4898	0.0113	-30753.47

4.2.2未来一周(2023年7月1日至7日)的补货和定价决策

(1) 求解思路

本文先采用SARIMA预测模型根据2023年6月份数据对7月1日至7日的销售总量进行预测, 再利用规划模型以及遗传算法对销售单价和补货量进行求解, 在利润最大的前提下, 增加一系列约束条件, 例如销售价格大于等于批发价格等等, 最终通过模型求解出2023年7月1日至7日的日补货量和定价策略。

(2) 构建规划模型

规划模型主要用于解决资源分配问题和决策问题。它可以帮助我们在给定的约束条件下, 找到最佳的决策方案或资源分配方案。规划模型涉及到定义目标函数、约束条件和决策变量。本题中, 目标函数是总利润, 决策变量是补货量和销售单价, 约束条件则是对决策变量的限制。

商超利润最大化是决策的前提, 利润由正常价售卖的商品收益加上折扣价售卖的商品收益再减去总成本构成。由附件2的销售数据统计出来的销售总量中既包含正常价售卖的也包含折扣价售卖的, 正常价收益部分需要排除折扣价销售的部分, 因此需要指标折扣率, 即销售记录中打折销售的销量占品类总销量的比率, 则正常品收益等于销售单

价乘上销售总量再乘（1—折扣率）。

表3 各品类折扣率

品类	折扣率
花菜类	0.053
花叶类	0.05
辣椒类	0.035
茄类	0.025
食用菌	0.078
水生根茎类	0.098

$$W_1 = \sum_{i=1}^7 \sum_{j=1}^6 [P_{ij} \times Q_{ij} \times (1-K_j)] \quad (2)$$

W_1 表示正常品收益， i 表示天数， j 表示品类， P_{ij} 表示某日某品类销售单价， Q_{ij} 表示某日某品类销售量， K_j 表示某品类折扣率。

每种蔬菜都有各自的损耗率，可能是运输途中的损耗，亦或是存储时间导致的品相变差甚至腐烂变质。商超对于此类产品进行打折销售，因此损耗品收益等于损耗率乘补货量乘折扣单价。

$$W_2 = \sum_{i=1}^7 \sum_{j=1}^6 (S_j \times Q_{tij} \times P_{kij}) \quad (3)$$

W_2 表示正常品收益， i 表示天数， j 表示品类， S_j 表示某品类均损耗率， Q_{tij} 表示某日某品类补货量， P_{kij} 表示某日某品类折扣单价。

商超以批发价购入蔬菜，则可计算成本为批发价乘以补货量。

$$C = \sum_{i=1}^7 \sum_{j=1}^6 (P_{tij} \times Q_{tij}) \quad (4)$$

C 表示总成本， i 表示天数， j 表示品类， P_{tij} 表示某日某品类批发单价， Q_{tij} 表示某日某品类补货量。

综合上述三个式子可以得到总利润（ W ），即正常品收益加损耗品收益减总成本。

$$W = \sum_{i=1}^7 \sum_{j=1}^6 [P_{ij} \times Q_{ij} \times (1-K_j) + S_j \times Q_{tij} \times P_{kij} - P_{tij} \times Q_{tij}] \quad (5)$$

(3) 模型的简化

观察数据可知，以折扣价出售的情况并不常见，在销售记录中占比极少，因此我们考虑对模型进行简化，去掉利润中的损耗品收益部分。为证明此次模型简化的合理性，我们以季度为单位，分别统计12个季度的折扣收益、正常收益以及折扣收益占总收益的比率。统计结果表明，折扣收益占比大多不足0.1，有7个季度不足0.05，有3个季度在0.01左右，可见折扣收益对于总收益来说影响极小。

表4 折扣收益分析

季度	折扣收益	正常收益	总收益	折扣收益占比
2020 秋	11174. 7478	351826. 711	363001. 4588	0. 0308
2020 冬	3936. 4809	302591. 3332	306527. 8141	0. 0128
2021 春	4886. 5864	401758. 1738	406644. 7602	0. 0120
2021 夏	2053. 4892	207547. 1892	209600. 6784	0. 0098
2021 秋	29950. 618	253098. 2882	283048. 9062	0. 1058
2021 冬	11556. 8916	178511. 4175	190068. 3091	0. 0608
2022 春	7280. 5265	263792. 7186	271073. 2451	0. 0269
2022 夏	6902. 4128	167626. 8252	174529. 2380	0. 0395
2022 秋	15923. 7863	296894. 2065	312817. 9928	0. 0509
2022 冬	14693. 9895	253657. 9377	268351. 9272	0. 0548
2023 春	16584. 2132	323649. 6096	340233. 8228	0. 0487
2023 夏	15175. 3885	207692. 9356	222868. 3241	0. 0681

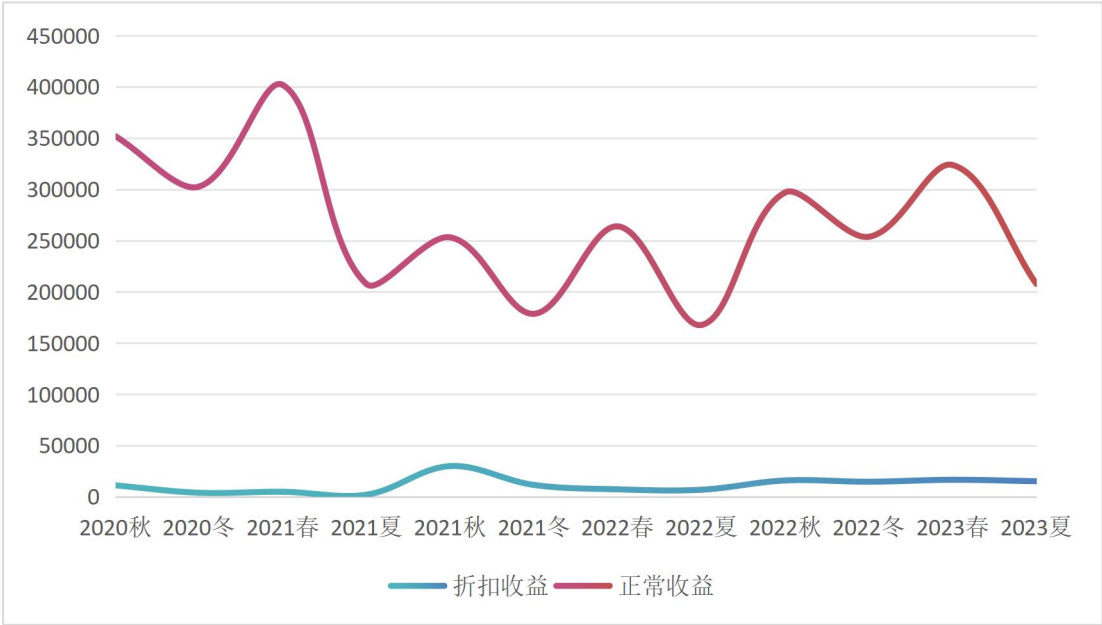


图22 折扣收益与正常收益比较

因此，模型的简化可以进行，折扣收益可以忽略，由此得到以下利润总公式。

$$W = \sum_{i=1}^7 \sum_{j=1}^6 [P_{ij} \times Q_{ij} \times (1-K_j) - P_{tij} \times Q_{tij}] \quad (6)$$

i 表示天数， j 表示品类， P_{ij} 表示某日某品类销售单价， Q_{ij} 表示某日某品类销售量， K_j 表示某品类折扣率， P_{tij} 表示某日某品类批发单价， Q_{tij} 表示某日某品类补货量。

上式的约束条件为：

- i. 销售量 \leq 补货量 \leq 1.1倍的销售量
- ii. $0 < \text{销售单价} \leq \text{加权最大销售单价}$
- iii. 销售单价 $>$ 批发单价

$$\begin{cases} Q_{ij} \leq Q_{tij} \leq 1.1Q_{ij} \\ 0 < P_{ij} \leq P_{mij} \\ P_{ij} > P_{tij} \end{cases} \quad (7)$$

其中，加权最大销售单价是指根据日销量占月销量的比例作为权重，对日销售单价的最大值进行加权平均，得到2023年6月份各品类的加权最大销售单价，这样的处理方式比直接求均值的可靠性更强，误差更小。

（4）SARIMA模型预测

SARIMA也称为季节性自回归滑动平均模型（Seasonal Autoregressive Integrated Moving Average），是一种时间序列预测模型，用于分析和预测具有季节性特征的数据。它基于ARIMA模型的扩展，通过引入季节差分来处理季节性变动，能够捕捉时间序列数据的自相关性和趋势，用于预测未来的数值。SARIMA模型包括三个主要组件：自回归部分（AR），差分部分（I），移动平均部分（MA）。三者结合起来以捕捉数据中的趋势、季节性和随机性特征。通过SARIMA模型利用2023年6月份的数据对未来一周的销售量和批发价进行预测，具体结果见下表。

表5 批发价预测值

品类	7月1日	7月2日	7月3日	7月4日	7月5日	7月6日	7月7日
花菜类	7.9397	8.0706	8.4115	8.2765	7.9610	7.1989	7.9618
花叶类	2.9559	3.1481	2.9724	3.0877	3.1455	3.1548	3.2882
辣椒类	5.0208	5.2098	4.8194	4.6982	4.9755	5.1219	4.5474
茄类	5.0598	4.7674	4.8212	5.5146	5.0319	5.0285	4.5146
食用菌类	8.3225	7.2275	4.9354	8.6418	5.9966	6.8846	7.3013
水生根茎类	14.2343	13.3069	11.8130	12.4961	13.6454	13.9509	13.7352

表6 销售量预测值

种类	7月1日	7月2日	7月3日	7月4日	7月5日	7月6日	7月7日
花菜类	21.6831	18.1048	15.1611	17.2305	21.4458	22.9970	22.3163

花叶类	122.8199	102.1370	111.5377	126.0457	131.7295	127.3434	123.6064
辣椒类	68.5599	68.5159	76.1433	73.0172	75.3751	79.6324	76.3762
茄类	22.9080	24.7705	14.1267	17.1531	13.7446	19.5269	18.9188
食用菌类	34.7126	41.4617	41.0224	45.5163	46.6916	44.0852	37.5914
水生根茎类	16.8577	14.1535	17.1252	17.2647	18.7487	18.0789	17.3866

(5) 遗传算法求解

当补货量和销售单价被一定条件限制时,可以将上述非线性规划问题看作一个组合优化问题,因此可以采用启发式算法提高求解效率。遗传算法作为一种基于自然原理的启发式算法,它的本质是通过群体搜索技术,根据适者生存的原则逐代进化,最终得到最优解或准最优解。该特性符合我们要求解的最优补货和定价策略模型,故采用遗传算法求解该模型。

遗传算法的一般流程分为求初始种群、确定目标函数、交叉操作、变异操作和选择等若干步骤,据此编写求解程序而后得到补货总量和销售单价的最优组合(如下表所示),此时未来一周(2023年7月1日至7日)的总利润最高,为8584.3054元。

表7 各品类补货和定价决策

品类	日期	销售单价(元/千克)	补货总量(千克)
花菜类	2023年7月1日	6.394515724	21.70137536
花菜类	2023年7月2日	6.420707387	18.2518801
花菜类	2023年7月3日	5.9743392	15.56931498
花菜类	2023年7月4日	6.325063871	17.51974501
花菜类	2023年7月5日	6.130440302	21.98875202
花菜类	2023年7月6日	5.562367434	23.35900112
花菜类	2023年7月7日	5.762596323	23.22611093
花叶类	2023年7月1日	13.48796133	123.6968318
花叶类	2023年7月2日	13.94001483	104.8682025
花叶类	2023年7月3日	13.8662425	115.0426826
花叶类	2023年7月4日	13.92300898	127.004146
花叶类	2023年7月5日	13.8723575	133.1802103
花叶类	2023年7月6日	12.16656954	130.9595125
花叶类	2023年7月7日	13.91087096	125.0186867
辣椒类	2023年7月1日	6.035443052	68.86305211
辣椒类	2023年7月2日	5.655821512	69.40421079
辣椒类	2023年7月3日	5.995850983	78.31296187
辣椒类	2023年7月4日	5.944816671	74.42039193
辣椒类	2023年7月5日	5.492132956	76.61503493

辣椒类	2023 年 7 月 6 日	6.051924281	79.82208132
辣椒类	2023 年 7 月 7 日	5.958884174	78.35892526
茄类	2023 年 7 月 1 日	6.290404655	23.66221676
茄类	2023 年 7 月 2 日	6.755343519	25.68022698
茄类	2023 年 7 月 3 日	5.81190599	15.16494884
茄类	2023 年 7 月 4 日	5.995135586	17.44825625
茄类	2023 年 7 月 5 日	6.538794357	13.77839639
茄类	2023 年 7 月 6 日	6.6397136	20.70180655
茄类	2023 年 7 月 7 日	6.822250321	19.34105175
食用菌	2023 年 7 月 1 日	8.277329561	34.85106064
食用菌	2023 年 7 月 2 日	8.021231711	42.64857989
食用菌	2023 年 7 月 3 日	8.292464299	43.22231646
食用菌	2023 年 7 月 4 日	8.231126342	45.5677768
食用菌	2023 年 7 月 5 日	7.189946528	49.05923859
食用菌	2023 年 7 月 6 日	8.307519098	45.41892561
食用菌	2023 年 7 月 7 日	8.035191098	38.95431982
水生根茎类	2023 年 7 月 1 日	16.69838265	16.97300531
水生根茎类	2023 年 7 月 2 日	16.72618162	14.42302425
水生根茎类	2023 年 7 月 3 日	15.90382211	17.22213921
水生根茎类	2023 年 7 月 4 日	15.71957929	17.59894669
水生根茎类	2023 年 7 月 5 日	15.87524823	19.25961788
水生根茎类	2023 年 7 月 6 日	16.75970151	18.24817675
水生根茎类	2023 年 7 月 7 日	14.69714907	17.74901996

4.3问题三模型的建立与求解

4.3.1指标设定及体系构建

(1) 销售总量 Q_s

对2023年6月24-6月30的每个可售单品的销售量进行加总，直接反映其当周的销售情况。具体公式表达为：

$$Q_s = \sum_{i=7}^n Q_{iq} \quad (8)$$

其中， Q_{iq} 为某日某单品销售量， n 取一周的七天， i 为天数， q 为指定单品。

(2) 加权利润率 W_p

本文在对周利润率进行测算时，考虑到不同可售单品在当周的销售总量会对其利润有较大影响，故选择使用当周内各单品的销售总量占比情况对其进行赋权，将每个单品

的周利润率赋予权重后得到其加权利润率。具体公式表达为：

$$W_p = \frac{\sum Q_m (P_{iq} - P_{tiq})}{\sum Q P_{iq}} \quad (9)$$

其中， Q_m 为本周某单品销售量， Q 为本周单品销售总量， m 表示不同单品， P_{iq} 为某日某单品的销售单价， P_{tiq} 为某日某单品的批发单价。

(3) 季节性销售评级 R

由于蔬菜销售与季节有很强的相关关系，在运用6月24日-30日之间的数据进行筛选时，还需考虑到季节在其中的影响情况。故本文选择使用2023年6月份各单品的销售总量占比情况对其进行划分并赋值，该月份销售总量占比越高则所赋分值越高，相应筛选时选中的概率增大；反之，该月份销售总量占比越低则所赋分值越低，被选中的概率减小。具体赋分（10分制）情况如下：

$$\begin{array}{lll} >6\%: 9分 & \leq 6\%: 7分 & \leq 4\%: 5分 \\ \leq 2\%: 3分 & \leq 1\%: 1分 & \leq 0.3\%: 0分 \end{array}$$

该指标是对本月份此前售卖行为的整体评价，可以较大程度上反映其销售趋势与程度，对未来的决策有较强的预测作用。因而本文选择该指标作为评判标准之一，并采取赋分制对不同销售量占比的单品进行等级划分，且原则上将占比低于0.3%的销售单品直接排除，不予考虑。

4.3.2 可售蔬菜单品筛选模型构建及求解

(1) 层次分析法与熵权法计算综合权重

本文利用层次分析法和熵值法对指标体系进行综合评价。层次分析法是一种应用网络系统理论和多目标综合评价方法，是在对复杂的决策问题的本质、影响因素及其内在关系等进行深入分析的基础上，利用较少的定量信息使决策的思维过程数学化，从而为多目标、多准则或无结构特性的复杂决策问题提供简便决策的方法。熵值法是一种依据各指标值所包含的信息量的多少确定指标权重的客观赋权法，某个指标的熵越小，说明该指标值的变异程度越大，提供的信息量也就越多，在综合评价中起的作用越大，则该指标被赋予的权重也应越大。

① 层次分析法赋权

据此本题建立层次结构模型拟解决商超最优决策问题。将商超定价与补货决策问题划分为三个层次：最上层为目标层，即用来直接指示各可售蔬菜的最终总得分；中间层为准则层，包括销售总量、加权利润率、季节性销售评级的三个指标；最下层为方案层，即满足利润最大化时的商超最优定价与补货决策。

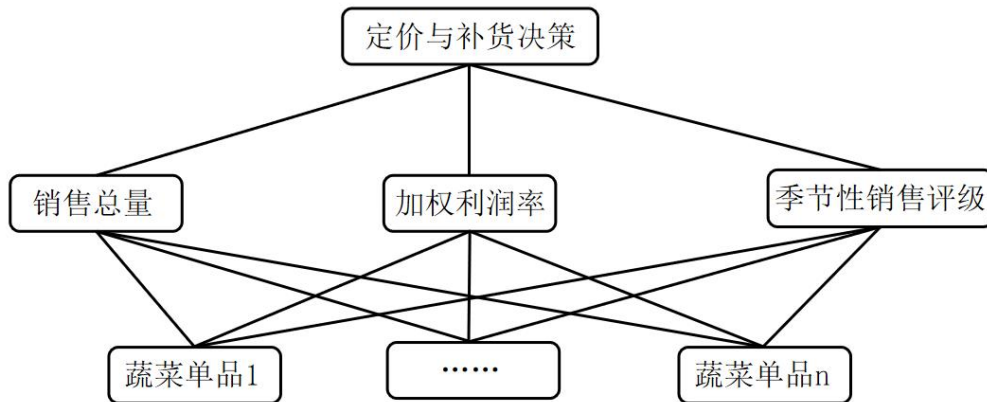


图23 层次结构模型

此时，指标的 AHP 权重分别为：

表8 指标权重表

项	权重（%）
销售总量（千克）	54%
加权利润率（百分比）	29.7%
季节性销售评级（分）	16.3%

②熵值法赋权

赋权的基本步骤为：

a. 指标矩阵的构建

$$X = (x_{ij}) = \begin{pmatrix} X_{11} & \cdots & X_{1m} \\ \vdots & \vdots & \vdots \\ X_{n1} & \cdots & X_{nm} \end{pmatrix}_{n \times m}, \begin{cases} i = 1, 2, \dots, n \\ j = 1, 2, \dots, m \end{cases} \quad (10)$$

b. 指标的无量纲及非负化处理

对于正向指标，做以下处理：

$$x_{ij} = \frac{x_{ij} - \min(x_{ij})}{\max(x_{ij}) - \min(x_{ij})} \quad (11)$$

对于负向指标，做以下处理：

$$x_{ij} = \frac{\max(x_{ij}) - x_{ij}}{\max(x_{ij}) - \min(x_{ij})} \quad (12)$$

c. 计算 j 项指标下第 i 个方案占该指标的比重

$$P_{ij} = \frac{X_{ij}}{\sum_{i=1}^n X_{ij}} \quad (j = 1, 2, \dots, m) \quad (13)$$

d. 计算第 j 项指标的熵值

$$e_j = -\frac{1}{\ln(n)} \sum_{i=1}^n \frac{X_{ij}}{\sum_{i=1}^n X_{ij}} \ln \frac{X_{ij}}{\sum_{i=1}^n X_{ij}} \quad (14)$$

e. 确定指标权重

$$W_j = 1 - \frac{e_j}{\sum_{j=1}^m (1 - e_j)} \quad (15)$$

此时，指标的熵权权重分别为：

表9 指标权重表

项	权重 (%)
销售总量（千克）	32%
加权利润率（百分比）	34.9%
季节性销售评级（分）	33.1%

③综合权重计算

综合权重=0.5×AHP 权重+0.5×熵权权重

最终，得到指标的综合权重分别为：

表10 指标权重表

项	权重 (%)
销售总量（千克）	43%
加权利润率（百分比）	32.3%
季节性销售评级（分）	24.7%

(2) TOPSLS方法量化可售蔬菜单品

TOPSIS方法是基于数据对样本进行排序的一种方法，其基本思想是根据样本数据构造一个理想化的目标，如在本例中就是构造一个各方面指标都达到最优的蔬菜单品，然后测量实际单品和这个理想化单品的接近程度，越接近就代表其能带来的收益越低。

找出每列也就是每个指标的最大值，记为 z_i^+ ($i=1, 2, \dots, m$)，组成向量

$$Z^+ = \{z_1^+, z_2^+, \dots, z_m^+\} \quad (16)$$

该向量代表理想的蔬菜单品。同样的，找出每列也就是每个指标的最小值，记为 z_i^- ($i=1, 2, \dots, m$)，组成向量

$$Z^- = \{z_1^-, z_2^-, \dots, z_m^-\} \quad (17)$$

该向量代表最不理想的蔬菜单品，即每个正向化后的指标都达到了最小。

定义第 i 个样本与理想目标的距离为 D_i^+ ，计算公式为

$$D_i^+ = \sqrt{\sum_{j=1}^m (z_j^+ - z_{ij})^2} \quad (18)$$

定义第*i*个蔬菜单品与不理想目标的距离为 D_i^- ，计算公式为

$$D_i^- = \sqrt{\sum_{j=1}^m (z_j^- - z_{ij})^2} \quad (19)$$

定义第*i*个蔬菜单品的得分为 S_i ，计算公式为

$$S_i = \frac{D_i^-}{D_i^+ + D_i^-} \quad (20)$$

显然， S_i 位于[0,1]之间。则当 S_i 越接近于1，说明蔬菜单品*i*距离理想化目标越近，该商超的最终收益就越大；当 S_i 越接近于0，说明蔬菜单品*i*距离理想化目标越远，该商超的最终收益就越小。

最终得到30家蔬菜单品的综合得分指数如图所示。

表11 蔬菜单品综合得分指数表

单品名称	综合得分指数	单品名称	综合得分指数	单品名称	综合得分指数
小米椒(份)	0.801614093	姜蒜小米椒组合装(小份)	0.390382919	木耳菜	0.237765236
云南生菜(份)	0.696974721	螺丝椒(份)	0.387475219	长线茄	0.229004814
云南油麦菜(份)	0.656278746	奶白菜	0.381169549	红薯尖	0.227207387
芜湖青椒(1)	0.599382376	苋菜	0.377407593	西峡花菇(1)	0.205867871
金针菇(盒)	0.590046463	双孢菇(盒)	0.372934393	净藕(1)	0.197202783
竹叶菜	0.525597023	娃娃菜	0.368828019	洪湖藕带	0.168585588
西兰花	0.514546538	海鲜菇(包)	0.35488355	青红杭椒组合装(份)	0.146910777
小皱皮(份)	0.514290836	螺丝椒	0.346094164	云南生菜	0.117879966
紫茄子(2)	0.49899594	上海青	0.262605703	圆茄子(2)	0.110966535
菠菜(份)	0.43837966	小青菜(1)	0.254632017	七彩椒(2)	0.107768244

(3) 模型合理性检验

由于缺乏已知的直接与可售蔬菜单品综合得分指数相关的数据，我们很难对该评价模型的正确性进行精确的检验。在此情况下，我们使用单品的季节性销量评级与所得的综合得分情况进行对照，结果如下图所示。

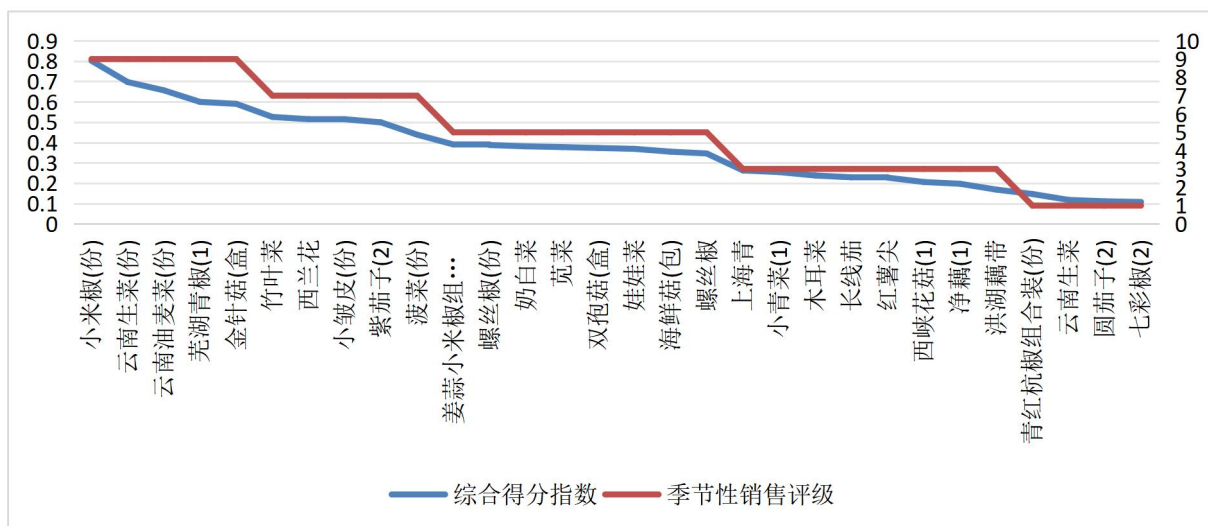


图24 季节性销售与总评序列

从总体趋势上来看，上图两个变量的变化趋势基本相同、拟合程度较高，说明本文计算出的综合得分指数较为准确，模型结果基本正确，可以依据此数据对可售单品进行筛选。但筛选原则不仅受季节性销量评级的影响，也与突发事件等及其他符合现实生活情况的本文未提及的因素有很大的关系。

4.3.3商超最优定价与补货决策模型构建与求解

(1) 求解思路

问题二已经建立起银行最优定价与补货决策模型，经过与其季节性销售评级的检验发现模型实现评价的效果较好。由于问题三仍然是对其最优定价与补货策略的确定，故结合题意，可在问题二的基础上实现，后续过程与之建立过程相重复，故不再赘述。

(2) SARIMA模型预测与遗传算法求解

SARIMA模型预测结果见表12，而后遗传算法求解得到的补货与定价决策结果见下表所示，其中最优组合计算得出2023年7月1日的总利润最高为912.7051437元。

表12 可售蔬菜单品预测批发价与销售价

单品名称	预测销售量	预测批发价格	非折扣率
小米椒(份)	19.58162629	2.135366185	0.913071534
云南生菜(份)	6.133799073	3.511691607	0.856434953
云南油麦菜(份)	16.57920286	2.855284339	0.769265486
芜湖青椒(1)	8.405243707	3.664308401	0.981428265
金针菇(盒)	10.19228662	1.455304293	0.924122891
竹叶菜	13.13000358	2.122374182	0.986655134
西兰花	11.47963381	7.595753841	0.958882947
小皱皮(份)	6.57032633	2.059595432	0.821745848
紫茄子(2)	18.81630379	3.43	0.963540339
菠菜(份)	3.817512586	4.051368878	0.778407302
姜蒜小米椒组合装(小份)	12.38945179	2.331131898	0.866702355
螺丝椒(份)	3.879986914	4.625490209	0.86086851
奶白菜	13.19921273	2.576356399	0.993410708

苋菜	5.448453487	2.31806972	0.990193223
双孢菇(盒)	5.886522148	3.405740597	0.800567108
娃娃菜	2.622291676	4.394365004	0.929953302
海鲜菇(包)	3.086545529	1.954780841	0.775384615
螺丝椒	10.95238293	9.348704965	0.987961901
上海青	2.101441835	4.107023232	0.986421533
小青菜(1)	1.961707481	2.672039732	0.998609355
木耳菜	14.19712522	3.136325739	0.945966908
长线茄	10.3856023	7.018672885	0.999830652
红薯尖	4.520670269	2.730565843	0.963475204
西峡花菇(1)	9.85998585	15.6	0.99535373
净藕(1)	4.782851687	10.31418731	0.894667176
洪湖藕带	3.321837933	18	0.962264151
青红杭椒组合装(份)	2.5989603	3.476469502	0.90548054
云南生菜	6.740293149	5.713278064	0.95775566
圆茄子(2)	2.204857485	3.088133201	0.999435984
七彩椒(2)	1.226695695	12.19568184	0.854442344

表13 可售蔬菜单品补货与定价策略

单品	销售单价	补货总量
小米椒(份)	4.264043733	19.89016524
云南生菜(份)	7.003797731	6.450301773
云南油麦菜(份)	5.695877776	17.08822076
芜湖青椒(1)	7.095495679	8.420805307
金针菇(盒)	2.880819253	10.22645012
竹叶菜	4.242883578	13.45845529
西兰花	15.07638807	11.52271265
小皱皮(份)	3.984766515	6.587402851
紫茄子(2)	6.832618364	18.83472647
菠菜(份)	8.066794302	3.83854152
姜蒜小米椒组合装(小份)	4.646498325	12.4227334
螺丝椒(份)	9.165408227	3.90684329
奶白菜	4.987393907	13.24338684
苋菜	4.621363369	5.489443702
双孢菇(盒)	6.368284625	5.913881268
娃娃菜	8.711718029	2.66910708
海鲜菇(包)	3.873882234	3.160786673

螺丝椒	18. 10207436	10. 96180335
上海青	7. 925687423	2. 181358744
小青菜(1)	3. 998772361	1. 998324507
木耳菜	6. 228337074	14. 27530394
长线茄	14. 01590661	10. 64617525
红薯尖	5. 261785688	4. 584626388
西峡花菇(1)	31. 1578747	10. 01808658
净藕(1)	20. 58678658	5. 028291721
洪湖藕带	35. 92031854	3. 342903073
青红杭椒组合装(份)	6. 880835669	2. 623546461
云南生菜	11. 30994343	6. 757749389
圆茄子(2)	5. 98371106	2. 217002986
七彩椒(2)	23. 37960515	1. 23380253

4.4问题四的解析

本题提供了商超经销蔬菜品类信息、各商品的销售流水明细、批发价格和损耗率数据。为了更好地制定蔬菜商品的补货和定价决策，商超还需采集更丰富的相关数据，下文将列举有助于进一步完善补货和定价策略的数据信息及其原因。

4.4.1产品变质率与存储时间的关系

企业通常采购并存储产品以满足未来需求，然而在存储过程中很多产品因为变质腐烂或挥发等因素产生数量损耗，蔬菜类产品便属于易变质腐烂的产品。在中国或其它发展中国家，果蔬类产品变质比例常常达到 15%以上^[1]，产品变质问题往往导致企业库存成本大幅上升^[2]。因此，企业需要将产品变质作为补货量的重要考虑因素。通常情况下，产品需求均依赖于产品价格，价格与补货决策若未能充分协调，易变质产品容易积压变质，从而造成较大的成本损失^[3]。

通过收集和分析历史数据，商超可以观察不同品类蔬菜在不同储存条件下的变质速度，了解各品类蔬菜在特定温度和湿度条件下的变质速度，并建立相应的模型进行分析，可以帮助商超确定蔬菜的最佳存储时长，并根据库存量和销售预期来安排补货计划，同时对于变质率较快的产品进行折扣定价，以避免库存积压和损失。

4.4.2进货和配送数据

易变质特性使库存决策进一步复杂化，因此供应链协调或者跨部门协作变得十分必要。当产品供给有限或者产品需求依赖于供给时，供应商、零售商和运输部门需要进行协调。研究表明，供应链整合不仅有利于供应链成员利润的提升，更有利于产品在流通过程中的数量损耗^[4]。

记录进货量、进货价、进货质量、供应商等数据，帮助商超评估不同供应商的表现和蔬菜的进货成本。通过分析不同批次的进货数据，可以发现价格波动、质量问题等情

况，以便进行供应链优化和成本控制。此外，了解蔬菜的运输和配送时间、效率和成本，利用模型比较分析出更优的运输策略，优化供应链的物流管理，减少库存损耗和滞销的风险，降低蔬菜变质腐烂风险，提高商品的新鲜度和品质，便于优化补货和定价策略。

4.4.3 市场竞争和外部经济数据

当今企业常常面临双重竞争压力，即来自竞争对手的直接竞争和消费者策略行为引发的跨期竞争。激烈的市场竞争和复杂的市场环境，使得企业的定价决策日趋复杂。竞争者的进入改变了市场竞争的格局，深刻影响在位企业定价决策及消费者购买决策^[5]。了解市场上其他竞争对手的蔬菜种类、价格、营销活动等信息，有助于商超制定差异化策略、优化定价和促销活动，并跟踪促销活动的销售结果，包括促销时段、促销策略、折扣力度等信息，从而评估促销对于蔬菜销售量的影响和效果，从而进一步完善定价策略，以提高市场份额和竞争力。同时，要考虑蔬菜市场的宏观经济因素，如通货膨胀率、人口变化、城市化程度等，有助于预测市场供需变化、调整定价策略，以及确定市场增长机会和风险。

4.4.4 消费者反馈数据

在易逝品的动态定价研究中，顾客行为是导致定价问题复杂的重要原因^[6]。在信息技术快速发展的电子商务环境下，消费者可以获得更多的产品信息从而加大了零售商定价决策的难度^[7]。国内学者刘晓峰等研究了确定性和不确定性需求情形下厂商的定价和库存决策，结果表明，厂商可以根据市场上高价值和低价值消费者的构成，通过适当的库存数量和价格设定，增大消费者买不到产品的风险，从而减少消费者的等待行为^[8]。更重要的是，若在定价决策中忽视消费者策略行为的影响，企业常常会蒙受重大的收益损失。收集消费者对不同蔬菜品种的偏好、购买动机、意见和评价等数据，关注社交媒体平台上与蔬菜相关的话题、评论和趋势，帮助商超了解消费者需求和趋势，获取市场反馈和消费者洞察，以便及时调整补货和定价策略，吸引更多消费者，改进服务质量。

五、模型灵敏度和稳定性分析

5.1 模型灵敏度分析

5.1.1 遗传算法求解规划模型的灵敏度分析

对于遗传算法求解规划模型的灵敏度检验，我们采用改变利润函数的约束条件 i 来实现，即改变约束条件 i 销售量 \leq 补货量 ≤ 1.1 倍的销售量中 1.1 倍数关系，每次加 0.1 ，循环 100 次，观察最大利润的变化情况。

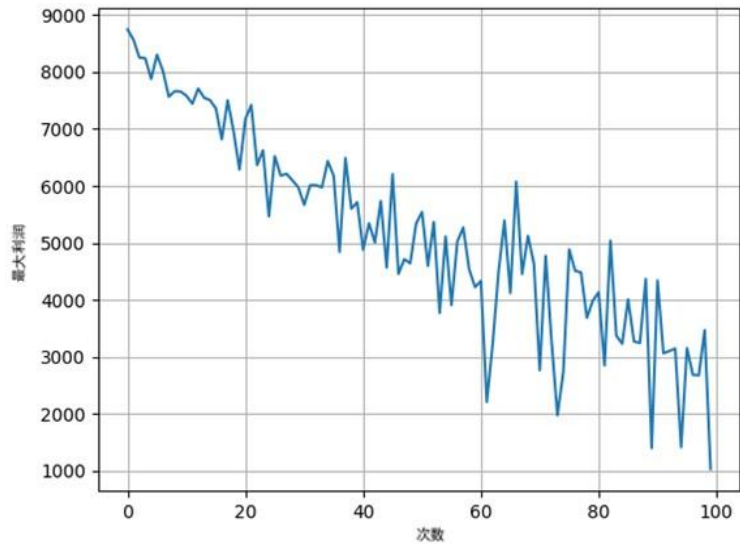


图25 遗传算法求解规划模型的灵敏度分析

观察图像可知，最大利润随着循环次数的增加有较大变动，灵敏度较高。随着倍数关系的增加，规划模型求解的最大利润有明显下降趋势，说明利用遗传算法求解规划模型的灵敏度较高。

5.1.2 综合评价模型的灵敏度分析

对于综合评价规划模型的灵敏度检验，采用改变权重组合来实现。本文模型中，综合权重= $0.5 \times \text{AHP 权重} + 0.5 \times \text{熵权权重}$ ，为检测其灵敏度，改变式子中的 0.5，观察综合得分指数的变化情况。由下图可知，随着权重组合的变化，综合得分指数有较为明显的变化，则综合评价模型灵敏度较高。

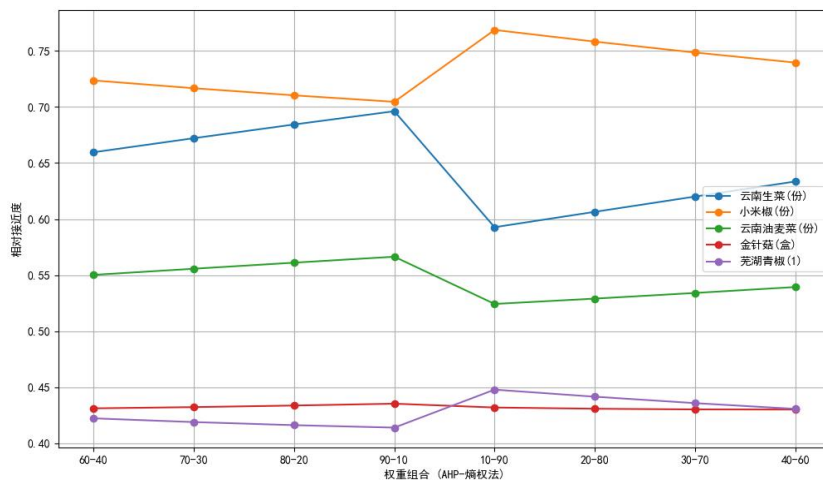


图26 综合评价模型的灵敏度分析

5.2 模型稳定性分析

为检测遗传算法求解规划模型的稳定性，采用增加模型运行次数来检验。将此模型运行 100 次，观察其最大利润是否有明显变化。由图可知，运行 100 次模型得到的最大利润结果较为稳定，曲线趋势平缓，则此模型稳定性较高，结果较为可靠。

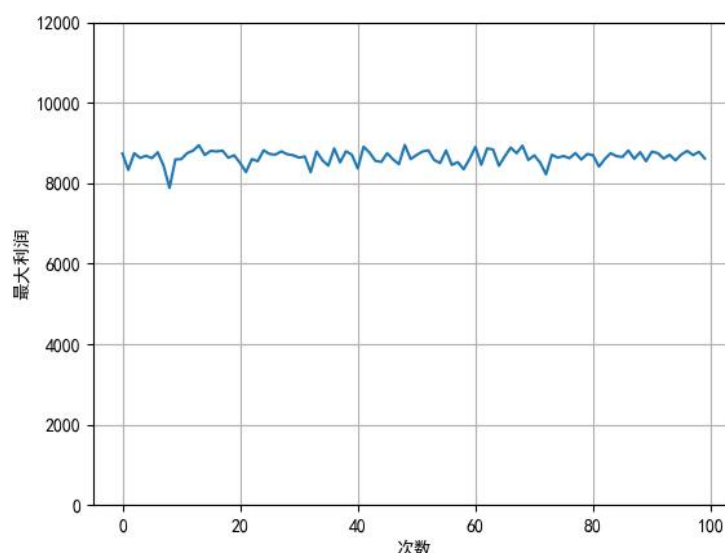


图27 遗传算法求解规划模型的稳定性分析

六、模型评价与推广

6.1 模型的优点

6.1.1 将多方面影响因素综合归纳为销售总量、加权利润率和季节性销售评级这三个方面，给出了商超补货及定价合理的策略，对实际具有较强的指导作用。

6.1.2 考虑蔬菜的随时间变化的特殊属性，合理利用 SARIMA 模型进行了科学预测。

6.1.3 采用遗传算法替代大规模非线性规划，兼顾了效率和准确性。

6.1.4 对必要指标进行加权平均的处理，使得数据运用更加完备、计算更加精确。

6.2 模型的缺点

6.2.1 模型考虑的因素不够全，在对蔬菜品类的相关性分析中计算综合指数时，只考虑了利润率与批发价格的影响；筛选可售单品时只考虑了影响较为显著的三个指标。

6.2.2 在筛选可售单品时，季度性销售评级的赋值具有一定的主观性，存在改进空间。

6.2.3 补货决策与定价决策之间可能存在关联关系，并未考虑在内。

6.2.4 销售总量与成本加成定价之间的相互关系所拟合的模型的预测准确度不高。

6.3 未来工作与推广

6.3.1 商超的定价与补货决策，事实上是销售商与消费者之间进行价格博弈的过程，可以从博弈论的角度出发，对模型进行拓展。

6.3.2 对于计算蔬菜品类的相关性分析中综合指数时、筛选可售单品时的指标选择及数据量化更进一步优化。

6.3.3 可以收集更多的真实统计数据，对规划模型进行优化、使得决策精计算更加精确。

参考文献

- [1] Ferguson, M.E., Ketzenberg, M.E. (2005).Information Sharing to Improve Retail Product Freshness of Perishables[M],Third Edition. Georgia Institute of Technology.
- [2] 邱祝强.基于冷藏链的生鲜农产品物流网络优化及其安全风险评价研究[D].中南大学,2008.
- [3] 张金隆,吴翔,徐浩轩.易变质新产品定价与补货联合决策模型[J].系统工程学报,2018,33(01):79-89.
- [4] 黄虹富. 易逝品供应链库存、定价决策与协调机制研究[D].东南大学,2020.
- [5] 张新鑫,申成霖,侯文华.考虑竞争者进入威胁的易逝品动态定价机制[J].管理科学学报,2016,19(10):34-47.
- [6] 胡玉生.基于顾客购买行为的易逝品动态定价[D].北京理工大学,2018.
- [7] 陈欣.考虑顾客满意度的易逝品跨期动态定价研究[D].沈阳工业大学,2021.
- [8] 刘晓峰,黄沛.基于策略型消费者的最优动态定价与库存决策[J].管理科学学报,2009,12(05):18-26.

附录

由于本文所做的数据处理部分较多，除正文中所示数据外，其余均可见支撑材料，代码部分如下附录所示。

1. 数据预处理--数据合并

```
import pandas as pd

data1 = pd.read_excel(r'附件 1.xlsx')
data2 = pd.read_excel(r'附件 2.xlsx')
print(data2.head())

merged_data = pd.merge(data1, data2, on='单品编码', how='inner')
merged_data.to_excel('合并数据结果 2.xlsx', index=False)
```

2. 问题一

(1) 单品季度作图

```
from matplotlib.font_manager import FontProperties

import pandas as pd

import matplotlib.pyplot as plt

font = FontProperties(fname=r"path/to/your/chinese/font.ttf")

data = pd.read_excel(r'E:\python 作业\新建文件夹 (2)\单品季度销量.xlsx')
groups = [4, 2, 3, 3, 3, 3]

titles = [

    "花叶类单品季度销售总额",
    "花菜类单品季度销售总额",
    "食用菌单品季度销售总额",
    "辣椒类单品季度销售总额",
    "水生根茎类单品季度销售总额",
    "茄类单品季度销售总额"

]

x_labels = ["2020 秋", "2020 冬", "2021 春", "2021 夏", "2021 秋", "2021 冬", "2022 春",
"2022 夏", "2022 秋", "2022 冬", "2023 春", "2023 夏"]
```

```

fig, axs = plt.subplots(2, 3, figsize=(15, 10))
plt.subplots_adjust(hspace=0.3)
col_start = 0
for i in range(2):
    for j in range(3):
        plt.rc('font', family='SimHei')
        group_size = groups[i*3 + j]
        group_data = data.iloc[:, col_start:col_start+group_size]
        group_data.plot(ax=axs[i, j], xticks=range(len(x_labels)), grid=True)
        axs[i, j].set_title(titles[i*3 + j])
        axs[i, j].set_xticklabels(x_labels, rotation=45)
        col_start += group_size
plt.show()

```

(2) 聚类分析

```

import pandas as pd
from sklearn.cluster import KMeans
import pylab as plt
import numpy as np

file_path = r'E:\python 作业\新建文件夹 (2)\1.xlsx'
data = pd.read_excel(file_path)
features = data[['平均利润率', '平均成本']]
kmeans = KMeans(n_clusters=3)
kmeans.fit(features)
data['Cluster'] = kmeans.labels_
centroids = kmeans.cluster_centers_
print(centroids)

for i, center in enumerate(centroids):
    distances = np.linalg.norm(features - center, axis=1)
    closest_indices = distances.argsort()[:3]

```

```

closest_names = data.loc[closest_indices, '单品名称'].tolist()
print(f'距离聚类中心 {i+1} 最近的前三个点是: {closest_names}')
plt.figure(figsize=(10, 6))
plt.rc('font', family='SimHei')
plt.scatter(data['平均利润率'], data['平均成本'], c=data['Cluster'], cmap='viridis', marker='o',
            edgecolor='k')
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', marker='x', s=100)
plt.xlabel('平均利润率')
plt.ylabel('平均成本')
plt.title('花菜类聚类散点图')
plt.grid(True)
plt.colorbar(label='Cluster')
plt.show()

```

(3) 数据提取及品类作图

```

import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties
file_paths = [r'E:\python 作业\新建文件夹 (2)\2020 秋.xlsx', r'E:\python 作业\新建文件夹
(2)\2020 冬.xlsx', r'E:\python 作业\新建文件夹 (2)\2021 春.xlsx',
               r'E:\python 作业\新建文件夹 (2)\2021 夏.xlsx', r'E:\python 作业\新建文件
夹 (2)\2021 秋.xlsx', r'E:\python 作业\新建文件夹 (2)\2021 冬.xlsx',
               r'E:\python 作业\新建文件夹 (2)\2022 春.xlsx', r'E:\python 作业\新建文件
夹 (2)\2022 夏.xlsx', r'E:\python 作业\新建文件夹 (2)\2022 秋.xlsx',
               r'E:\python 作业\新建文件夹 (2)\2022 冬.xlsx', r'E:\python 作业\新建文件
夹 (2)\2023 春.xlsx', r'E:\python 作业\新建文件夹 (2)\2023 夏.xlsx']
results = []
for i, file_path in enumerate(file_paths):
    df = pd.read_excel(file_path)
    sum_sales = df.groupby('分类名称')['销量(千克)'].sum().reset_index()

```

```

sum_sales['时间点'] = i + 1
results.append(sum_sales)

all_data = pd.concat(results)
output_path = r'E:\python 作业\新建文件夹 (2)\combined_results.xlsx'
all_data.to_excel(output_path, index=False)
pivot_result = all_data.pivot(index='时间点', columns='分类名称', values='销量(千克)')
plt.figure(figsize=(12, 10))
pivot_result.plot(linestyle='-', linewidth=2)
font = FontProperties(fname=r"C:\Windows\Fonts\simhei.ttf")
time_labels = ['2020 秋', '2020 冬', '2021 春', '2021 夏', '2021 秋', '2021 冬', '2022 春', '2022
夏', '2022 秋', '2022 冬', '2023 春', '2023 夏']
plt.xticks(ticks=range(1, 13), labels=time_labels, rotation=45, fontproperties=font)
plt.title('时间序列销量折线图', fontproperties=font, fontsize=16)
plt.xlabel('时间点', fontproperties=font, fontsize=14)
plt.ylabel('销量(千克)', fontproperties=font, fontsize=14)
plt.legend(title='分类名称', bbox_to_anchor=(1.05, 1), loc='upper left', prop=font)
plt.grid(True)
plt.tight_layout()
plt.show()

```

(4) 相关性分析

```

import pandas as pd
from matplotlib.font_manager import FontProperties
import seaborn as sns
import matplotlib.pyplot as plt
from pandas import read_excel
file_path = r'E:\python 作业\季度销量.xlsx'
data = read_excel(file_path)
correlation_matrix = data.iloc[:, 1:].corr()

```



```

plt.figure(figsize=(10, 8))
plt.rcParams['axes.unicode_minus'] = False
plt.rc('font', family='SimHei')
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=1,
            linecolor='black')
plt.title('品类相关性热力图', fontsize=16)
plt.show()

```

3. 问题二

(1) SARIMA 模型预测销售总量

```

import pandas as pd

from statsmodels.tsa.arima.model import ARIMA

import matplotlib.pyplot as plt

from matplotlib.font_manager import FontProperties

font = FontProperties(fname=r"C:\Windows\Fonts\simhei.ttf", size=12)

file_path = r'E:\python 作业\新建文件夹 (2)\销售额预测.xlsx'

data = pd.read_excel(file_path)

categories_data = {}

unique_categories = data['品类'].unique()

for category in unique_categories:

    categories_data[category] = data[data['品类'] == category].reset_index(drop=True)

arima_predictions = {}

for category, df in categories_data.items():

    df_arima = df[['销售时间', '销售总量']].set_index('销售时间')

    model = ARIMA(df_arima, order=(5,1,0))

    model_fit = model.fit()

    forecast = model_fit.get_forecast(steps=7)

```

```

conf_int = model_fit.get_forecast(steps=7).conf_int()
last_date = df_arima.index[-1]
future_dates = list(range(1, 8))
arima_predictions[category] = {
    'dates': future_dates,
    'forecast': forecast.predicted_mean.values,
    'conf_int': conf_int.values
}

fig, axes = plt.subplots(2, 3, figsize=(15, 10))
axes = axes.ravel()
for i, (category, prediction) in enumerate(arima_predictions.items()):
    axes[i].plot(prediction['dates'], prediction['forecast'], color='blue', label='预测销售额')
    axes[i].fill_between(prediction['dates'], prediction['conf_int'][:, 0],
prediction['conf_int'][:, 1], color='gray', alpha=0.3, label='信心区间')
    axes[i].set_title(f'{category} 销售预测', fontproperties=font)
    axes[i].set_xlabel('日期', fontproperties=font)
    axes[i].set_ylabel('销售数量', fontproperties=font)
    axes[i].legend(prop=font)
    axes[i].grid(True)
plt.tight_layout()
plt.subplots_adjust(hspace=0.3) # 调整上下图之间的间距
plt.show()

```

(2) SARIMA 模型预测批发价格

```

import pandas as pd

from statsmodels.tsa.statespace.sarimax import SARIMAX

data_path = '/mnt/data/问题 2 数据预测.xlsx'
data = pd.read_excel(data_path)

```

```

data['分类名称'] = data['分类名称'].replace('食用菌', '食用菌类')
categories = ['花菜类', '花叶类', '辣椒类', '茄类', '食用菌类', '水生根茎类']
data_dict = {category: data[data['分类名称'] == category].reset_index(drop=True) for
category in categories}

def get_simple_sarima_model(ts):
    try:
        model = SARIMAX(ts, order=(1, 1, 1), seasonal_order=(1,1,1,12)).fit(dis=0)
    except:
        model = None
    return model

sarima_models = {}
sarima_predictions = {}
for category in categories:
    ts = data_dict[category].set_index('销售日期')['批发价格(元/千克)']
    model = get_simple_sarima_model(ts)
    if model:
        sarima_models[category] = model
        sarima_predictions[category] = model.get_forecast(steps=7).predicted_mean
    else:
        sarima_predictions[category] = None
predictions_df = pd.DataFrame()
for category in categories:
    predictions_df[category] = sarima_predictions[category]
output_path = '/mnt/data/vegetable_price_predictions.xlsx'
predictions_df.to_excel(output_path)

```

(3) 遗传算法

```
from sko.GA import GA
```

```
import pandas as pd
```

```
sales_volume_data = pd.read_excel(r'E:\python 作业\新建文件夹 (2)\问题 2 规划模型数据_销售总量.xlsx')
```

```
wholesale_price_data = pd.read_excel(r'E:\python 作业\新建文件夹 (2)\问题 2 规划模型数据_批发价格.xlsx')
```

```
discount_rate_data = pd.read_excel(r'E:\python 作业\新建文件夹 (2)\规划模型数据.xlsx')
```

```
sales_volume_dict = sales_volume_data.set_index('种类').T.to_dict('dict')
```

```
wholesale_price_dict = wholesale_price_data.set_index('种类').T.to_dict('dict')
```

```
discount_rate_data.set_index('种类', inplace=True)
```

```
discount_rate_dict = discount_rate_data.to_dict('index')
```

```
categories = sales_volume_data['种类'].tolist()
```

```
days = sales_volume_data.columns[1:].tolist()
```

```
def fitness(x):
```

```
    total_profit = 0
```

```
    k = 0
```

```
    for i, category in enumerate(categories):
```

```
        for j, day in enumerate(days):
```

```
            selling_price = x[k]
```

```
            replenishment_volume = x[k + 1]
```

```
            sales_volume = sales_volume_dict[category][day]
```

```
            wholesale_price = wholesale_price_dict[category][day]
```

```
            discount_rate = discount_rate_dict[category]['折扣率']
```

```

        daily_profit = (selling_price * sales_volume * (1 - discount_rate)) - (
                        replenishment_volume * wholesale_price)
        total_profit += daily_profit
        k += 2

    return -total_profit # We return negative value because we want to maximize the total
profit, and the library minimizes the fitness function

```

```

lb = []
ub = []

for category in categories:
    for day in days:
        lb.extend([discount_rate_dict[category]['销售单价 MIN'],
sales_volume_dict[category][day]])
        ub.extend([discount_rate_dict[category]['销售单价 MAX'], 1.1 *
sales_volume_dict[category][day]])

ga = GA(func=fitness, n_dim=2 * len(categories) * len(days), size_pop=50, max_iter=200,
lb=lb, ub=ub)
best_x, best_y = ga.run()

# print('Best Solution:', best_x)
print('Maximum Profit:', -best_y)
results = []

k = 0

for i, category in enumerate(categories):
    for j, day in enumerate(days):
        selling_price = best_x[k]
        replenishment_volume = best_x[k+1]

```

```
results.append({'种类': category, '日期': day, '销售单价': selling_price, '补货总量':  
replenishment_volume})
```

```
k += 2
```

```
results_df = pd.DataFrame(results)  
# results_df.to_excel('问题 2 求解的数据.xlsx', index=False)
```

4. 问题三

(1) 批发价预测

```
import pandas as pd  
from datetime import datetime  
from statsmodels.tsa.arima.model import ARIMA  
  
file_path = r'E:\python 作业\问题三预测批发价.xlsx'  
data = pd.read_excel(file_path)  
  
start_date = datetime(2023, 6, 24)  
end_date = datetime(2023, 6, 30)  
filtered_data = data[(data['销售日期'] >= start_date) & (data['销售日期'] <= end_date)]  
  
product_names_order = [  
    "小米椒(份)", "云南生菜(份)", "云南油麦菜(份)", "芜湖青椒(1)", "金针菇(盒)", "竹  
叶菜", "西兰花",  
    "小皱皮(份)", "紫茄子(2)", "菠菜(份)", "姜蒜小米椒组合装(小份)", "螺丝椒(份)", "  
奶白菜", "苋菜",  
    "双孢菇(盒)", "娃娃菜", "海鲜菇(包)", "螺丝椒", "上海青", "小青菜(1)", "木耳菜", "  
长线茄",  
    "红薯尖", "西峡花菇(1)", "净藕(1)", "洪湖藕带", "青红杭椒组合装(份)", "云南生菜",  
"圆茄子(2)",  
    "七彩椒(2)"
```

```
]
```

```
grouped_data_time_series = filtered_data.groupby(['单品名称', '销售日期'])['批发价格(元/千克)'].mean().reset_index()
```

```
predicted_prices_dict = {}
```

```
date_list = pd.date_range(start="2023-06-24", end="2023-06-30")
```

```
for product_name in product_names_order:
```

```
    product_data = grouped_data_time_series[grouped_data_time_series['单品名称'] == product_name]
```

```
    if not product_data.empty:
```

```
        product_data.set_index('销售日期', inplace=True)
```

```
        product_data = product_data.reindex(date_list)
```

```
        product_data.fillna(product_data['批发价格(元/千克)'].mean(), inplace=True)
```

```
        model = ARIMA(product_data['批发价格(元/千克)'], order=(1, 1, 1))
```

```
        model_fit = model.fit()
```

```
        forecast = model_fit.forecast(steps=1)
```

```
        predicted_prices_dict[product_name] = forecast[0]
```

```
    else:
```

```
        predicted_prices_dict[product_name] = None
```

```
predicted_prices_df = pd.DataFrame(list(predicted_prices_dict.items()), columns=['单品名称', '预测批发价格(元/千克)'])
```

```
output = '预测批发价格_2023 年 7 月 1 日.xlsx'
```

```
predicted_prices_df.to_excel(output, index=False)
```

(2) 遗传算法

```
import pandas as pd
```

```

from sko.GA import GA

data_df = pd.read_excel(r'E:\python 作业\问题 3 的规划求解 1.xlsx')

def objective_function(vars):
    x = vars[0:30]
    y = vars[30:60]

    sales_revenue = sum(x[i] * data_df['预测销售量'][i] * data_df['非折扣率'][i] for i in
range(30))
    replenishment_cost = sum(y[i] * data_df['预测批发价格'][i] for i in range(30))

    return -(sales_revenue - replenishment_cost)

def constraint1(vars):
    x = vars[0:30]
    return data_df['预测批发价格'] - x

def constraint2(vars):
    x = vars[0:30]
    return x - 2 * data_df['预测批发价格']

def constraint3(vars):
    y = vars[30:60]
    return data_df['预测销售量'] - y

```



```

def constraint4(vars):
    y = vars[30:60]
    return y - 1.1 * data_df['预测销售量']

ga = GA(func=objective_function, n_dim=60, size_pop=50, max_iter=200,
        lb=[*data_df['预测批发价格'].tolist(), *data_df['预测销售量'].tolist()],
        ub=[*(2 * data_df['预测批发价格']).tolist(), *(1.1 * data_df['预测销售量']
        ')].tolist()),
        constraint_eq=[constraint1, constraint2, constraint3, constraint4])

best_vars, best_score = ga.run()
results_df = data_df[['索引']].copy()
results_df['销售单价'] = best_vars[0:30]
results_df['补货总量'] = best_vars[30:60]
results_df.to_excel('优化结果.xlsx', index=False)

print("Best Variables: ", best_vars)
print("Best Score: ", -best_score)

```

(3) 折扣率结果

```

import pandas as pd

file_path = 'E:\python 作业\折扣率的计算.xlsx'
products_list = [
    "小米椒(份)", "云南生菜(份)", "云南油麦菜(份)", "芜湖青椒(1)", "金针菇(盒)", "竹
    叶菜",
    "西兰花", "小皱皮(份)", "紫茄子(2)", "菠菜(份)", "姜蒜小米椒组合装(小份)", "螺丝
    椒(份)",

```

```

    "奶白菜", "茼菜", "双孢菇(盒)", "娃娃菜", "海鲜菇(包)", "螺丝椒", "上海青", "小青
    菜(1)",
    "木耳菜", "长线茄", "红薯尖", "西峡花菇(1)", "净藕(1)", "洪湖藕带", "青红杭椒组合
    装(份)",
    "云南生菜", "圆茄子(2)", "七彩椒(2)"
]

```

```

data_df = pd.read_excel(file_path)
result_dict = {'商品': [], '否的数量': [], '总计数': [], '折扣率': []}
for product in products_list:
    product_data = data_df[data_df['单品名称'] == product]
    no_discount_count = product_data['是否打折销售'].value_counts().get('否', 0)
    total_count = len(product_data)
    discount_rate = no_discount_count / total_count if total_count != 0 else 0
    result_dict['商品'].append(product)
    result_dict['否的数量'].append(no_discount_count)
    result_dict['总计数'].append(total_count)
    result_dict['折扣率'].append(discount_rate)
result_df = pd.DataFrame(result_dict)
result_df.to_excel('折扣率结果.xlsx', index=False)

```

5. 模型灵敏度与稳定性分析

(1) 遗传算法灵敏度分析

```

from sko.GA import GA
import pandas as pd
import matplotlib.pyplot as plt

sales_volume_data = pd.read_excel(r'E:\python 作业\新建文件夹 (2)\问题 2 规划模型数据_
销售总量.xlsx')
wholesale_price_data = pd.read_excel(r'E:\python 作业\新建文件夹 (2)\问题 2 规划模型数

```

```

据_批发价格.xlsx')
discount_rate_data = pd.read_excel(r'E:\python 作业\新建文件夹 (2)\规划模型数据.xlsx')
sales_volume_dict = sales_volume_data.set_index('种类').T.to_dict('dict')
wholesale_price_dict = wholesale_price_data.set_index('种类').T.to_dict('dict')
discount_rate_data.set_index('种类', inplace=True)
discount_rate_dict = discount_rate_data.to_dict('index')
categories = sales_volume_data['种类'].tolist()
days = sales_volume_data.columns[1:].tolist()

```

```
def fitness(x):
```

```
    total_profit = 0
```

```
    k = 0
```

```
    for i, category in enumerate(categories):
```

```
        for j, day in enumerate(days):
```

```
            selling_price = x[k]
```

```
            replenishment_volume = x[k + 1]
```

```
            sales_volume = sales_volume_dict[category][day]
```

```
            wholesale_price = wholesale_price_dict[category][day]
```

```
            discount_rate = discount_rate_dict[category]['折扣率']
```

```
            daily_profit = (selling_price * sales_volume * (1 - discount_rate)) - (
                replenishment_volume * wholesale_price)

```

```
            total_profit += daily_profit
```

```
            k += 2
```

```
    return -total_profit
```

```
best_solutions = []
```

```
max_profits = []
```

```

for i in range(100):
    coeff = 1.1 + 0.1 * i
    lb = []
    ub = []
    for category in categories:
        for day in days:
            lb.extend([discount_rate_dict[category]['销售单价 MIN'],
sales_volume_dict[category][day]])
            ub.extend([discount_rate_dict[category]['销售单价 MAX'], coeff *
sales_volume_dict[category][day]])

    ga = GA(func=fitness, n_dim=2 * len(categories) * len(days), size_pop=50,
max_iter=200, lb=lb, ub=ub)
    best_x, best_y = ga.run()

    best_solutions.append(best_x)
    max_profits.append(-best_y)

plt.figure()
plt.plot(max_profits)
plt.rc('font', family='SimHei')
plt.xlabel('次数')
plt.ylabel('最大利润')
plt.title('模型的灵敏度分析')
plt.grid(True)
plt.show()

```

(2) TOPSIS 灵敏度分析

```
import numpy as np
```

```

import pandas as pd
from openpyxl import load_workbook
import matplotlib.pyplot as plt

# Step 1: 加载数据
file_path = r'E:\python 作业\新建文件夹 (2)\1-销售总量+季节性销售评级+加权利润率.xlsx'
wb = load_workbook(file_path)
sheet_names = wb.sheetnames
data_df = pd.read_excel(file_path, sheet_name=sheet_names[0])

judgment_matrix = np.array([
    [1, 2, 3],
    [1 / 2, 1, 2],
    [1 / 3, 1 / 2, 1]
])

eigenvalues, eigenvectors = np.linalg.eig(judgment_matrix)
max_eigenvalue_index = np.argmax(eigenvalues)
ahp_weights = eigenvectors[:, max_eigenvalue_index]
ahp_weights_normalized = ahp_weights / np.sum(ahp_weights)

def calculate_entropy(data):
    p = data / np.sum(data)
    entropy = -np.sum(p * np.log(p))
    return entropy

entropies = data_df.iloc[:, 1:].apply(calculate_entropy, axis=0)

total_entropy = np.sum(entropies)

```

```
entropy_weights = (1 - entropies) / (3 - total_entropy)
```

```
def topsis_analysis(combined_weights, data_df):
    decision_matrix = data_df.iloc[:, 1:4].values
    normalized_decision_matrix = decision_matrix /
np.sqrt(np.sum(np.square(decision_matrix), axis=0))
    weighted_normalized_decision_matrix = normalized_decision_matrix *
combined_weights
    positive_ideal_solution = np.max(weighted_normalized_decision_matrix, axis=0)
    negative_ideal_solution = np.min(weighted_normalized_decision_matrix, axis=0)
    distance_to_positive_ideal = np.linalg.norm(weighted_normalized_decision_matrix -
positive_ideal_solution, axis=1)
    distance_to_negative_ideal = np.linalg.norm(weighted_normalized_decision_matrix -
negative_ideal_solution, axis=1)
    relative_closeness = distance_to_negative_ideal / (distance_to_positive_ideal +
distance_to_negative_ideal)

    data_df['相对接近度'] = relative_closeness
    sorted_data_df = data_df.sort_values(by='相对接近度', ascending=False)
    return sorted_data_df
```

```
all_weight_combinations = [(0.6, 0.4), (0.7, 0.3), (0.8, 0.2), (0.9, 0.1), (0.1, 0.9), (0.2, 0.8),
(0.3, 0.7), (0.4, 0.6)]
original_data_df = data_df.copy()
```

```
sensitivity_analysis_data = {product: [] for product in data_df['单品名称'].values[:5]}
for combo in all_weight_combinations:
    combo_weights = combo[0] * ahp_weights_normalized.real + combo[1] *
entropy_weights.values
```

```

sorted_data = topsis_analysis(combo_weights, original_data_df.copy())
for product in sensitivity_analysis_data.keys():
    sensitivity_analysis_data[product].append(
        sorted_data[sorted_data['单品名称'] == product]['相对接近度'].values[0])

plt.figure(figsize=(12, 7))
plt.rc('font', family='SimHei')
for product, closeness_values in sensitivity_analysis_data.items():
    plt.plot([f'{int(combo[0] * 100)}-{int(combo[1] * 100)}' for combo in
all_weight_combinations], closeness_values,
            marker='o', label=product)
plt.legend()
plt.title('灵敏度分析')
plt.xlabel('权重组合 (AHP-熵权法)')
plt.ylabel('相对接近度')
plt.grid(True)
plt.show()

```

(3) 遗传算法稳定性分析

```

from sko.GA import GA
import pandas as pd
import matplotlib.pyplot as plt

sales_volume_data = pd.read_excel(r'E:\python 作业\新建文件夹 (2)\问题 2 规划模型数据_
销售总量.xlsx')
wholesale_price_data = pd.read_excel(r'E:\python 作业\新建文件夹 (2)\问题 2 规划模型数
据_批发价格.xlsx')
discount_rate_data = pd.read_excel(r'E:\python 作业\新建文件夹 (2)\规划模型数据.xlsx')
sales_volume_dict = sales_volume_data.set_index('种类').T.to_dict('dict')
wholesale_price_dict = wholesale_price_data.set_index('种类').T.to_dict('dict')

```

```

discount_rate_data.set_index('种类', inplace=True)
discount_rate_dict = discount_rate_data.to_dict('index')
categories = sales_volume_data['种类'].tolist()
days = sales_volume_data.columns[1:].tolist()

```

```

def fitness(x):
    total_profit = 0
    k = 0
    for i, category in enumerate(categories):
        for j, day in enumerate(days):
            selling_price = x[k]
            replenishment_volume = x[k + 1]
            sales_volume = sales_volume_dict[category][day]
            wholesale_price = wholesale_price_dict[category][day]
            discount_rate = discount_rate_dict[category]['折扣率']

            daily_profit = (selling_price * sales_volume * (1 - discount_rate)) - (
                replenishment_volume * wholesale_price)
            total_profit += daily_profit
            k += 2
    return -total_profit

```

```

lb = []
ub = []
for category in categories:
    for day in days:
        lb.extend([discount_rate_dict[category]['销售单价 MIN'],
            sales_volume_dict[category][day]])

```



```

ub.extend([discount_rate_dict[category]['销售单价 MAX'], 1.1 *
sales_volume_dict[category][day]])

best_solutions = []
max_profits = []
for i in range(100):
    ga = GA(func=fitness, n_dim=2 * len(categories) * len(days), size_pop=50,
max_iter=200, lb=lb, ub=ub)
    best_x, best_y = ga.run()

    best_solutions.append(best_x)
    max_profits.append(-best_y)

plt.figure()
plt.rc('font', family='SimHei')
plt.plot(max_profits)
plt.xlabel('次数')
plt.ylabel('最大利润')
plt.title('模型稳定性分析')
plt.ylim(0, 12000)
plt.grid(True)
plt.show()

```

6.问题二 销售总量与成本加成定价的关系——stata 代码

. reg 水生根茎类销量 水生根茎类利润率

Source	SS	df	MS	Number of obs	=	12
Model	14477517.9	1	14477517.9	F(1, 10)	=	9.03
Residual	16033320.2	10	1603332.02	Prob > F	=	0.0132
				R-squared	=	0.4745
				Adj R-squared	=	0.4220
				Root MSE	=	1266.2
Total	30510838.1	11	2773712.55			

	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
水生根茎类销量					
水生根茎类利润率	-8158.348	2714.982	-3.00	0.013	-14207.7 -2108.992
_cons	7791.942	1512.473	5.15	0.000	4421.941 11161.94

. reg 花叶类销量 花叶类利润率

Source	SS	df	MS	Number of obs	=	12
Model	132847765	1	132847765	F(1, 10)	=	13.51
Residual	98320376.8	10	9832037.68	Prob > F	=	0.0043
				R-squared	=	0.5747
				Adj R-squared	=	0.5321
				Root MSE	=	3135.6
Total	231168142	11	21015285.6			

	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
花叶类销量					
花叶类利润率	-69853.65	19003.51	-3.68	0.004	-112196.1 -27511.2
_cons	76530.87	16344.52	4.68	0.001	40113.02 112948.7

reg 水生根茎类销量 水生根茎类利润率

reg 花叶类销量 花叶类利润率

. reg 花菜类销量 花菜类利润率

Source	SS	df	MS	Number of obs	=	12
Model	6781977.63	1	6781977.63	F(1, 10)	=	8.85
Residual	7663308.75	10	766330.875	Prob > F	=	0.0139
				R-squared	=	0.4695
				Adj R-squared	=	0.4164
Total	14445286.4	11	1313207.85	Root MSE	=	875.4

花菜类销量	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
花菜类利润率	-12314.06	4139.34	-2.97	0.014	-21537.08 -3091.032
_cons	14441.4	3693.123	3.91	0.003	6212.608 22670.19

reg 花菜类销量 花菜类利润率

. reg 茄类销量 茄类利润率

Source	SS	df	MS	Number of obs	=	12
Model	2068362.56	1	2068362.56	F(1, 10)	=	5.28
Residual	3919778.25	10	391977.825	Prob > F	=	0.0445
				R-squared	=	0.3454
				Adj R-squared	=	0.2800
Total	5988140.81	11	544376.437	Root MSE	=	626.08

茄类销量	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
茄类利润率	-5912.614	2573.931	-2.30	0.044	-11647.69 -177.5378
_cons	6644.271	2086.518	3.18	0.010	1995.219 11293.32

reg 茄类销量 茄类利润率

. reg 辣椒类销量 辣椒类利润率

Source	SS	df	MS	Number of obs	=	12
Model	46467796.4	1	46467796.4	F(1, 10)	=	11.14
Residual	41696165.6	10	4169616.56	Prob > F	=	0.0075
				R-squared	=	0.5271
				Adj R-squared	=	0.4798
Total	88163962.1	11	8014905.64	Root MSE	=	2042

辣椒类销量	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
辣椒类利润率	-23900.34	7159.393	-3.34	0.008	-39852.46 -7948.223
_cons	25292.99	5323.007	4.75	0.001	13432.59 37153.39

reg 辣椒类销量 辣椒类利润率

. reg 食用菌销量 食用菌利润率

Source	SS	df	MS	Number of obs	=	12
Model	39841407.4	1	39841407.4	F(1, 10)	=	9.60
Residual	41497481.7	10	4149748.17	Prob > F	=	0.0113
				R-squared	=	0.4898
				Adj R-squared	=	0.4388
Total	81338889.1	11	7394444.46	Root MSE	=	2037.1

食用菌销量	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
食用菌利润率	-30753.47	9925.162	-3.10	0.011	-52868.1 -8638.827
_cons	24997	6049.699	4.13	0.002	11517.43 38476.57

reg 食用菌销量 食用菌利润率