赛区评阅编号（由赛区组委会填写）：

# 2023 年高教社杯全国大学生数学建模竞赛

## 承　诺　书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称 “竞赛章程和参赛规则”，可从 http://www.mcm.edu.cn 下载）。

我们完全清楚，在竞赛开始后参赛队员不能以任何方式，包括电话、电子邮件、“贴吧”、QQ 群、微信群等，与队外的任何人（包括指导教师）交流、讨论与赛题有关的问题；无论主动参与讨论还是被动接收讨论信息都是严重违反竞赛纪律的行为。

**我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。**

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从 A/B/C/D/E 中选择一项填写）：C

我们的报名参赛队号（12 位数字全国统一编号）：202317241264

参赛学校（完整的学校全称，不含院系名）：华中科技大学

参赛队员 (打印并签名) ：1. 马韬

2. 李安琪

3. 赵济键

指导教师或指导教师组负责人 (打印并签名)：董锐

（指导教师签名意味着对参赛队的行为和论文的真实性负责）

日期：2023 年 10 月 3 日

**（请勿改动此页内容和格式。此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）**

赛区评阅编号：
（由赛区填写）

全国评阅编号：
（全国组委会填写）

# 2023 年高教社杯全国大学生数学建模竞赛

# 编 号 专 用 页

赛区评阅记录（可供赛区评阅时使用）：

| | | | | | | |
|---|---|---|---|---|---|---|
| 评阅人 | | | | | | |
| 备注 | | | | | | |

送全国评阅统一编号:
（赛区组委会填写）

**（请勿改动此页内容和格式。此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页。）**

# 基于数据分析优化模型的蔬菜补货与定价策略模型

## 摘要

本文主要基于组委会提供的商超蔬菜销售信息数据建立 **XGBoost 回归拟合模型**、**基于遗传算法的非线性规划模型**、用于结果检验的 **WSO_BiLSTM 预测模型**、基于粒子群算法的规划模型来解决商超蔬菜商品定价销售与补货决策的问题。

首先，由于数据量庞大，需要对数据进行**预处理**，包括缺失值、异常值的处理和对数据的整合、降维等，利用 SPSS Pro 进行缺失值的补充，利用 Matlab 进行“**3σ 原理**”检测异常值，并使用数据中位数填充，有的异常值还需我们接介入进行解释；利用 **Excel** 进行数据的整合、降维，详情见支撑材料。数据处理为下面的模型建立提供便利。

问题一中，分析蔬菜各品类及单品销售量的分布规律及相互关系。首先探寻分布规律，利用 **Excel** 处理好的数据集进行蔬菜各类和个单品之间销售量的**柱状图**和随时间段以及随季度的销售量**折线图**。对于相互关系，针对六类蔬菜品类每日销售量建立 **Pearson 相关性分析，**而对于 251 种蔬菜单品总销量进行**聚类分析**，根据聚类分析散点图探寻一定的相互关系。在问题一的处理中，我们发现的蔬菜品类具有相当可观的周期性规律，可以为之后的预测提供依据。

问题二中，分析蔬菜品类销售量与成本加成定价的关系。成本加成定价即制定**利润率策略**，故利用 Excel 处理的数据进行品类**销售量和利润率**分别随时间的曲线，然后先用线性回归拟合，发现**假设错误**，于是采用 **XGBoost 回归拟合模型**，计算均方误差，**假设成立。**又要求给出各品类未来一周的日补货总量和定价策略，使收益最大，则建立**基于遗传算法的非线性规划模型**，进行求解，由于数据量的完备，本文又建立销售量的 **WSO_BiLSTM 预测模型**进行补货量的检验，证明模型的准确性。

问题三中，要求制定 7 月 1 日的单品补货量和定价策略，且要求**可售单品总数**控制在 27-33 个，并**各单品订购量满足最小陈列量在 2.5 千克以上**，而对于 7 月 1 日的预测都建立在 6 月 24-30 日的可售品种之内的数据基础之上。因此，首先我们可以分析本共有 251 种单品，但从 251 种单品种找到 27-33 种的组合方式有相当多的情况，因此我们要缩小范围，首先，将 7 天中的不可售品种剔除，并且将每天销售量均低于 2.5 千克的商品删去（因为可以预测到 7 月 1 日也会低于 2.5 千克，因此提前删除，减少运算次数），然后便可建立**规划模型**，即使已经减少了许多数据带来的复杂运算，可本问规划模型依旧有众多组合，因此需要采用一种尽力减少运算的优化算法，可以考虑到**粒子群算法**进行求解。

问题四中，要求找出为了更好地制定蔬菜商品的补货和定价决策还需要采集哪些相关数据，并给出理由。由于售卖蔬菜是一个受众多因素影响的事件，比如**消费者的喜好购买数据、相竞争的商超情况、市场份额占比、促销情况、商超的地理因素等**，围绕着这几个方面，本文给出一定的意见和理由支撑。

**关键词：**数据处理　Pearson 相关性分析　优化模型　预测模型　XGBoost 回归拟合模型

# 一、问题重述

## 1.1 问题背景

这是一个关于蔬菜商品定价销售与补货决策的问题。由于生鲜商超中的蔬菜类商品保鲜期较短，并且随销售时间增加导致品相变差，甚至隔日可能无法卖出，故商超会根据商品历史销售和需求每天进行补货。

由于商超的蔬菜种类多以及进货交易时间在凌晨，故商家必须在不确切知道具体单品和进货单价价格的情况，进行补货决策。并结合商品的特点和市场需求分析进行定价销售。对商超来说，从需求侧，蔬菜销售量与时间存在一定的关联关系；从供给侧来看，蔬菜供应会在 4-10 月比较丰富，因此商超的定价与补货策略显得尤为重要。

## 1.2 问题提出

结合附件 1：商超销售 6 个蔬菜品类的商品信息；附加 2：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相关数据；附件 3：该商超 2020 年 7 月 1 日至 2023 年 6 月 30 日各商品的批发价格相关数据；建立数学模型解决下列问题：

1. 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. 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. 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. 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 问题总分析

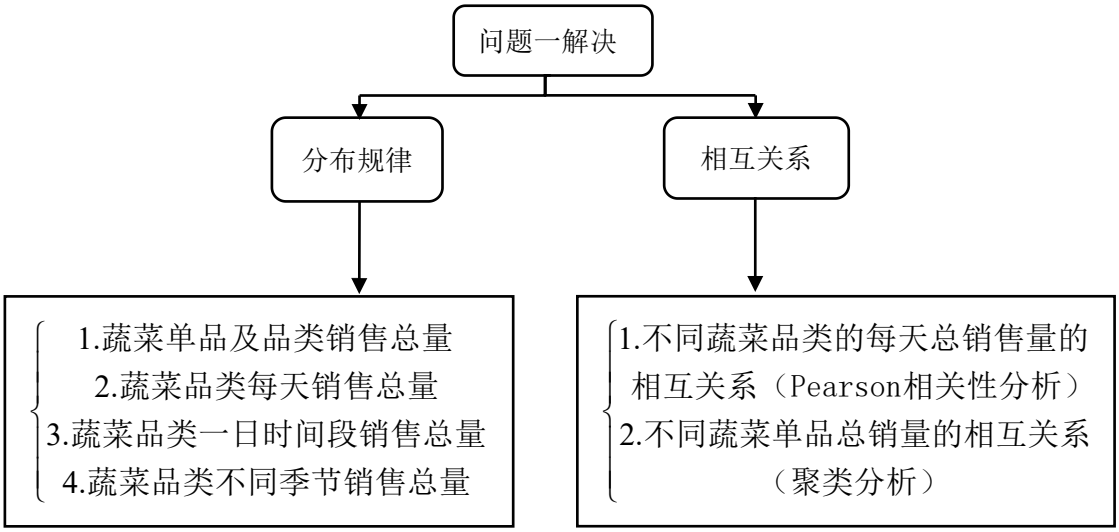
本题共涉及数据处理、预测模型、非线性回归模型、目标优化模型等，首先问题一是问题二、三的基础，因此在对问题一处理时要根据问题二、三进行相关销量的分布规律探寻，结合 Matlab 和 Excel 进行数据的清洗和分布规律的探寻。接着，问题二结合问题一中对蔬菜品类的数据处理和每日不同时间段销售总量的规律并计算出成本加成定价数据，利用非线性回归模型找到各蔬菜品类的销售总量与成本加成定价的关系，利用预测模型预测出未来一周的销售总量，然后建立单目标优化模型，结合各蔬菜品类的销售总量与成本加成定价的关系，制定出未来一周的日补货总量和定价策略。问题三首先根据每种蔬菜的销售利润，结合约束条件选择其中利润最高的 30 种蔬菜单品，然后利用已选蔬菜单品的销售规律，利用预测模型分别预测出 7 月 1 日的单品销售量，并用单目标优化模型得出当日单品补货量和定价策略。

## 2.2 具体问题分析

### 2.2.1 问题一分析

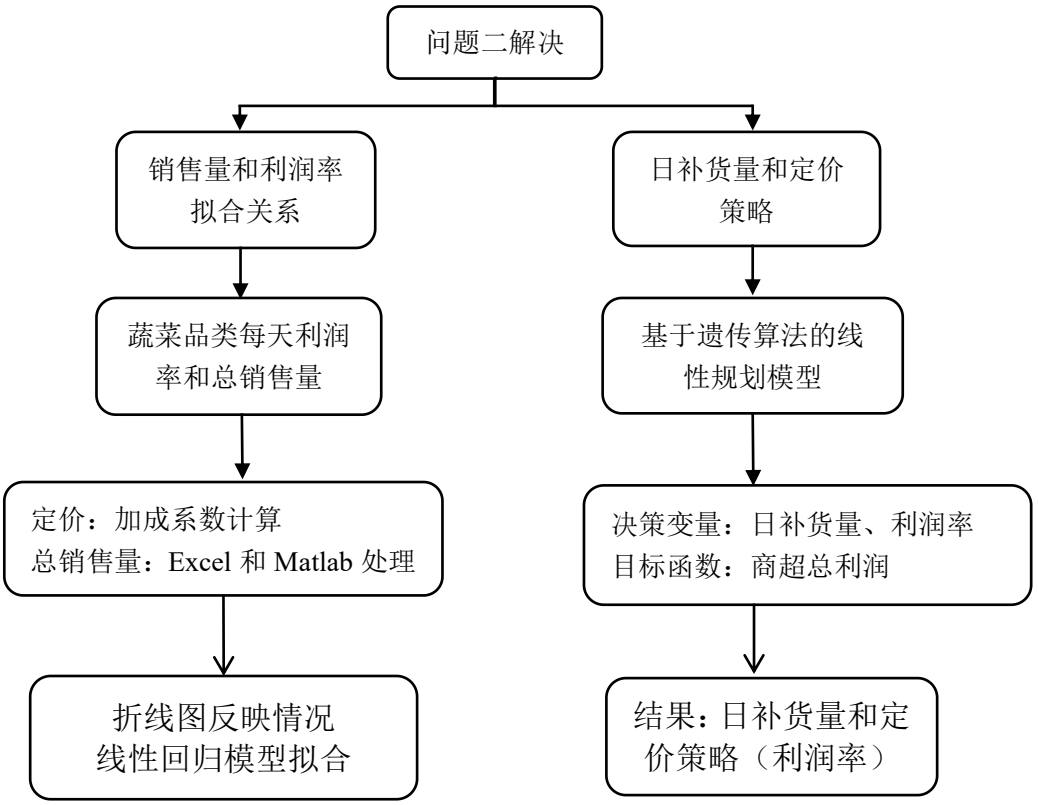
问题一要求分析蔬菜品类销售量的分布规律及相互关系，以及单品销售量的分布规律和相互关系。那么考虑到附件 2 的数据庞大性，本文首先利用 Excel 和 Matlab 计算出各单品与蔬菜各品类的总销售量，并将其可视化，为了更好的分析蔬菜单品及品类的分布规律，利用 Excel 结合 Matlab 整理出不同蔬菜品单品及品类在每天时长的销售量分布、

每个季度销售量分布、每年销售量分布，并进行了可视化处理。而考虑到蔬菜单品和品类销售量的相互关系，首先利用聚类分析法分析不同蔬菜单品总销售量的相互关系。然后利用 Pearson 相关性分析不同蔬菜品类的每天销售量的相互关系。




### 2.2.2 问题二分析

问题二要求分析各蔬菜品类的销售总量与成本加成定价的关系，首先成本加成定价即可以认为销售量和定价的关系，即需要计算蔬菜品类对应的成本，而这个成本既可以使用蔬菜品类对应的多个蔬菜单品的加成系数进行计算，然后利用 Excel 和 Matlab 对数据进行处理找到每个蔬菜品类每天销售量和定价的数据，然后对数据进行检验，发现可以使用线性回归模型进行拟合，确定蔬菜品类的销量的定价的关系，问题二又要求给出各蔬菜品类未来一周的日补货总量和定价策略，并使商超收益最大，于是本文将日补货量和利润率作为决策变量，以商超收益为目标函数，并利用销量与定价的拟合关系预测，建立基于遗传算法的线性规划模型，即可求解。



### 2.2.3 问题三分析

问题三要求制定 7 月 1 日的单品补货量和定价策略，且要求可售单品总数控制在 27-33 个，并各单品订购量满足最小陈列量在 2.5 千克以上，而对于 7 月 1 日的预测都建立在 6 月 24-30 日的可售品种之内的数据基础之上。因此，首先我们可以分析总共有 251 种单品，但从 251 种单品种找到 27-33 种的组合方式有相当多的情况，因此我们要缩小范围，然后将 7 天中的不可售品种剔除，并且将每天销售量均低于 2.5 千克的商品删去（因为可以预测到 7 月 1 日也会低于 2.5 千克，因此提前突出，减少运算次数），然后便可建立规划模型，即使已经减少了许多数据带来的复杂运算，可本问规划模型依旧有众多组合，因此需要采用一种尽力减少运算的优化算法，可以考虑到粒子群算法进行求解。

### 2.2.4 问题四分析

问题四中，要求找出为了更好地制定蔬菜商品的补货和定价决策还需要采集哪些相关数据，并给出理由。由于售卖蔬菜是一个受众多因素影响的事件，比如消费者的喜好购买数据、相竞争的商超情况、市场份额占比、促销情况、商超的地理因素等，围绕着这几个方面，本文给出一定的意见和理由支撑。

## 三、模型假设

1、假设不考虑补货带来的其他无关成本；
2、假设补货都是基于对供货商能力的充分考虑；
3、假设蔬菜单品的损耗率不随时间的不同而变化；
4、假设在每次商超批发前该蔬菜单品都销售完毕；
5、假设市场上的竞争对手行为相对稳定，不会影响商超价格策略；

## 四、符号说明

| 符号 | 说明 | 单位 |
|---|---|---|
| $X_i$,$Y_i$ | 六种蔬菜品类的销售总量，一一对应为$X_i$,$Y_i$ | Kg |
| K | 聚类数值 | 无量纲 |
| F、P | 聚类分析中的显著性水平 | 无量纲 |
| $x_i$ | 第 i 个蔬菜单品的定价 | 元/Kg |
| $z_i$ | 第 i 种蔬菜单品的批发价格 | 元 |
| $r_i$ | 蔬菜单品的利润率 | 无量纲 |
| $p_i$ | 第 i 种蔬菜单品的销售量 | Kg |
| $R_j$ | 第 j 个蔬菜品类的利润率 | 无量纲 |
| $a_{ij}$ | 第 i 类蔬菜品类在第 j 天的日补货量 | Kg |
| $w_{ij}$ | 第 i 类蔬菜品类在第 j 天的日利率 | 无量纲 |
| $s_i$ | 第 i 类蔬菜品类的亏损率 | 无量纲 |
| $c_{ij}$ | 第 i 类蔬菜品类第 j 天的成本 | 元/Kg |
| $q_i$ | 筛选变量 | 无量纲 |

## 五、模型的建立与求解

### 5.1 数据预处理

首先，本题给出多个附件包含大量数据，此后所有的建模准备皆是在提供大量数据的前提下进行的，故本文将对数据进行一定的清洗与预处理。对附件二近 88 万数据进行缺失值和异常值的检查并进行处理，具体操作有：

1. 针对缺失值及处理方法：

通过 Excel 软件对附件二中数据进行缺失值统计，如果存在缺失值则将该数据剔除。

2. 针对异常值及处理方法：

通过 Excel 软件对附件二中数据进行分析，近 88 万的账单明细数据进行检查，可以发现一些销量呈现负数，考虑到现实问题中的退货事件，故我们可以认为销量成负数为退货事件。

经过 Matlab 的处理，经过数据筛选可以看出有不少蔬菜单品在某段时间每成本极低，因此为了数据普遍性，应该进行数据异常值的处理。这里我们采用“3σ 标准差原则”：

1. 计算数据均值和标准差
2. 确定异常值阈值，即与均值相差超过 3 倍标准差的数据点
3. 处理异常值：对于被标记为异常值的数据点，进行中位数替代处理。

其他附件可以观察到不存在数据错误，通过本次处理，我们可以认为该数据都可以进行数据分析。

### 5.1 问题一模型

问题一指出蔬菜单品及品类之间存在一定的相互关系，要求找到蔬菜单品及品类销售量的分布规律和相互关系，关于问题的解决思路，在问题一分析中已经指出。

#### 5.1.1 附件数据处理

由于附件二中庞大的数据，以及附件之间的紧密联系，需要对附件中的数据进行合并、绑定、排序、摘取从而得到多个可以直接使用 Matlab、Python 或 SPSS 进行绘图、相关性分析等一系列操作，而针对数据的处理，我们使用 Excel 数据透视表，所得到的一系列处理过的数据表格放在支撑材料中，附录给出说明。

#### 5.1.2 模型解答

##### 5.1.2.1 分布规律

1）蔬菜单品及品类销量总量

由于蔬菜单品经过处理共有 246 中售卖单品，故利用柱方图展示，下面展示整体柱方图：

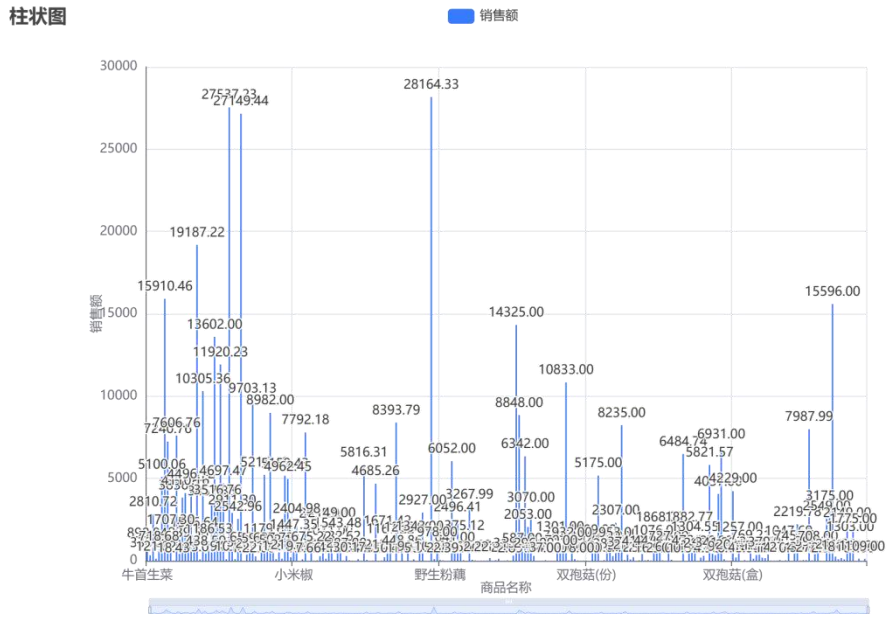


图 1 各蔬菜单品总销售量柱状图——整体展示

由此图可以得出，云南生菜、大白菜、西兰花、净藕（1）、芜湖青椒、云南生菜（份）、小米椒、金针菇（盒）等蔬菜总售卖量均超过一万五千次。可见各蔬菜单品总销售量的一定分布规律。

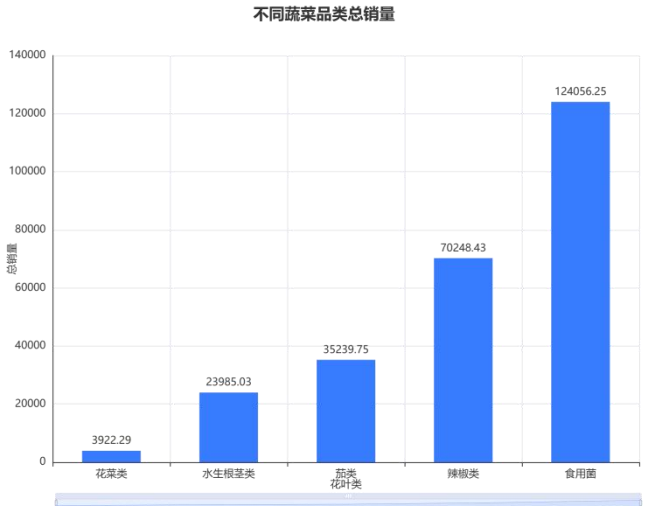



图 2 不同蔬菜品类总销量柱状图

由此图可以看出，该商超对食用菌的售卖总量远高于其他蔬菜品类，其中花菜类销售量最少，可能原因是收到当地的一些习俗或者环境因素影响，当地人可能对食用菌的要求更高，对花菜类了要求更低，或者食用菌在当地生产较为良好，而花菜类在当地不适宜生长等多个因素，由此图可以得知各蔬菜品类的一些分布规律。

2）蔬菜品类每天销售总量

由于蔬菜品类只有 6 种，因此为了显示随时间序列蔬菜销售量的分布规律，并且可以对之后的问题解决奠定良好基础，因此下面本问通过 Matlab 对数据进行处理并画出折线图展示不同种蔬菜品类销售总量随时间的变化：

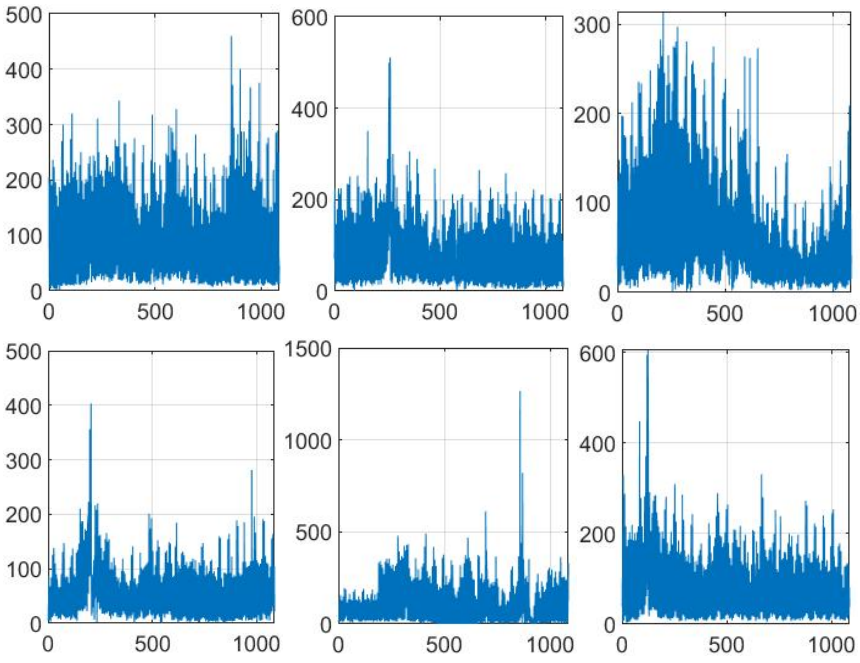


图 3 6 类蔬菜品类随时间变化曲线

该图是六种蔬菜品类随时间的变化的曲线，该图可以进行不同蔬菜品类销售量随时间序列的对比，比如可以看出不同蔬菜品类销售量的密集程度。

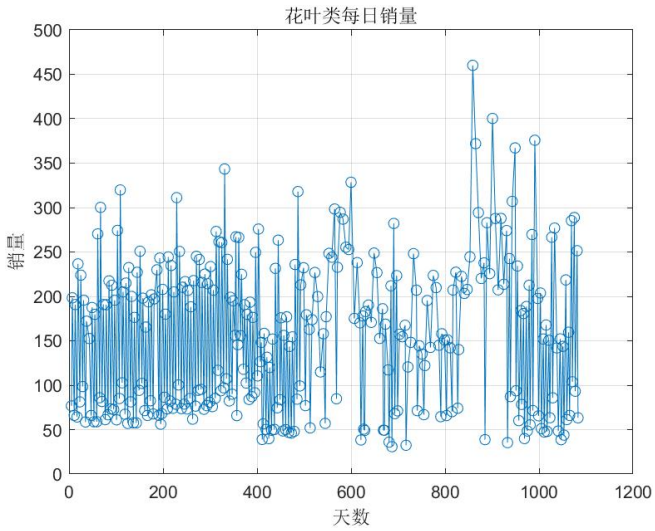



图 4 花叶类销量随天数变化

由于六类蔬菜品类较多，不便全部放在正文中，因此放在附录中展示。该图是花叶类销量随天数的变化折线图，可以看出其分布规律。

3）蔬菜品类在一天时间段的销售量变化

首先考虑到商超卖蔬菜在每天的不同时间段的销量会给发生变化，又由于蔬菜单品种类过多不具有普遍性，因此考虑蔬菜品类在一天时间段的销量变化。

本问使用 Excel 的数据透视表进行处理，即利用数据透视表可以得出如下形式：

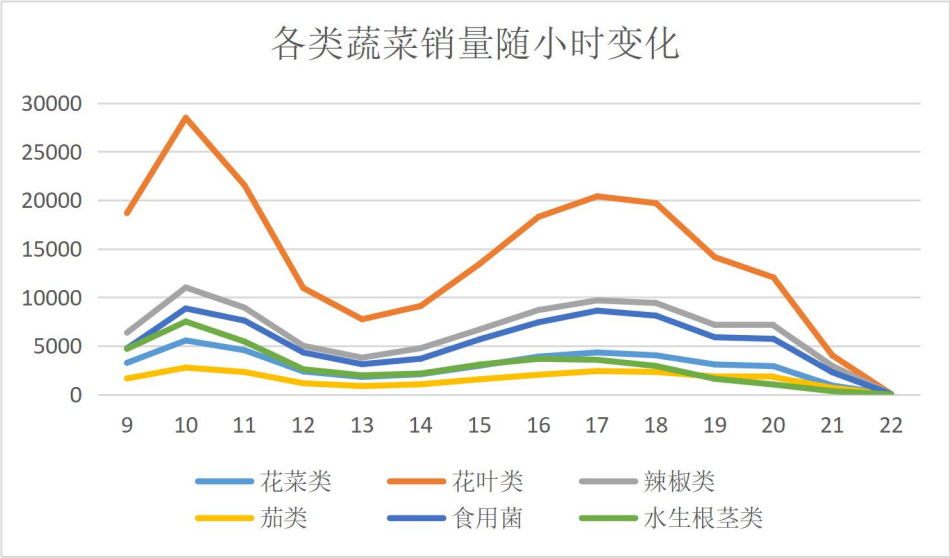



图 5 各类蔬菜销量随小时变化折线图

4）蔬菜品类在不同季度的销售量变化

每类蔬菜由于各种因素在不同季节会出现不同的售卖情况，故考虑到普遍性选择蔬菜品类在不同季度的销售量变化。

本问使用 Excel 的数据透视表进行处理，可得出如下形式：

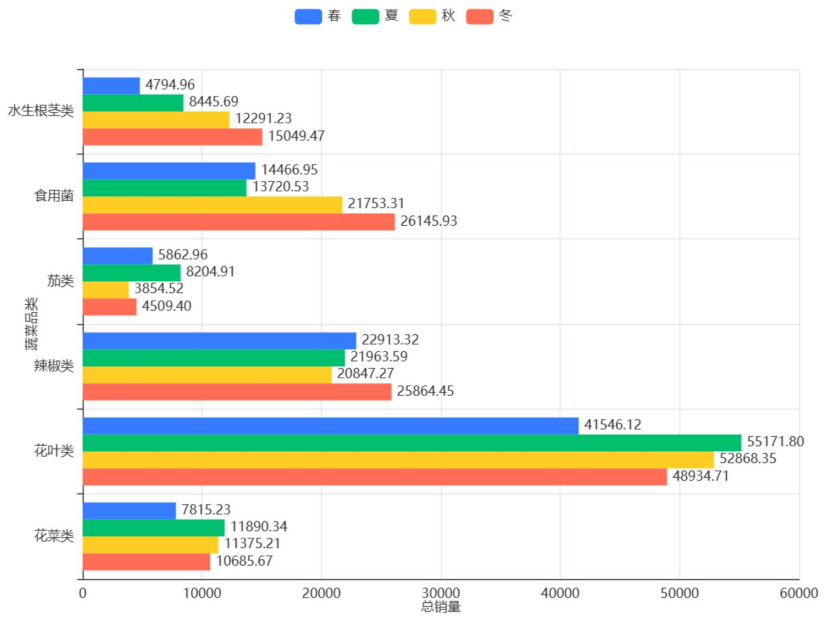



图 6 各蔬菜品类在不同季节销量柱状图

5.1.2.2 相互关系

本题询问的是蔬菜各品类及单品销售量的相互关系，及探寻一定的相关性，而联系现实可以考虑到蔬菜品类之间可能会有共同买的情况，因此可以探求不同蔬菜品类每天销售总量的相互关系。而针对蔬菜单品销售量来说，首先可以像不同蔬菜品类一样，探寻每天销售总量的相互关系，其次考虑到蔬菜单品种类较多，可以采用聚类分析法。

1）不同蔬菜品类每天销售总量的相互关系

首先可以使用 Excel 表格和 Matlab 去对附件二进行处理，即将每天的单品对应的蔬菜品类得出，首先得到每天的不同品类的销售额，然后将相同品类的销售额相加，即可得到每类在每天的销售总量，从而创造出六种蔬菜品类对应的 1085 天的销售量的向量，然后利用 Pearson 相关性分析法进行相关性分析。

Pearson 相关性分析模型建立

1. 建立前准备

作出假设：

（1）观察值是独立的；

（2）六个变量之间两两独立；

（3）变量之间存在线性关系；

2. 计算平均值：

对六个变量每两个变量分别计算平均值： $\bar{X}$、$\bar{Y}$

3. 计算差值

对于每一对数据点，分别计算两个变量的差值：观察值减去平均值。

4. 计算 Pearson 相关系数：

$$r=\frac{\sum(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum(X_i-\bar{X})^2(Y_i-\bar{Y})^2}}$$

这里， $X_i$ 和 $Y_i$ 分别是两个变量的观察值， $\bar{X}$、$\bar{Y}$ 分别是两个变量的平均值。r 的取值范围在-1 到 1 之间，表示两个变量之间的线性关系强度和方向如果 r 接近 1，表

示存在强正相关性；如果 r 接近-1，表示存在强负相关性；如果 r 接近 0，则表示没有线性关系。

5.检验相关性的显著性

可以进行假设检验来确定 Pearson 相关系数是否显著。计算相关系数的 p 值。如果 p 值小于设定的显著性水平（通常为 0.05），则可以拒绝原假设，认为相关性是显著的。

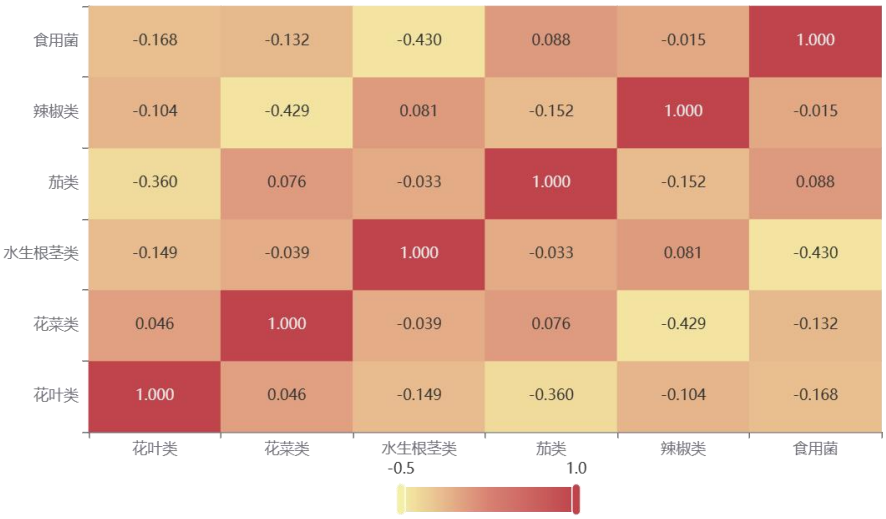



图 7 不同蔬菜品类销售向量热力图

由此图可以看出，花菜类和花叶类、花菜和水声根茎类、水声根茎类和茄类、食用菌和辣椒类都具有显著相关性。

2）不同蔬菜单品总销售量的相关性

由于蔬菜单品种类较多，因此我们可以采用聚类分析法进行初步探寻相互关系。

聚类分析模型

1. 数据预处理

对数据进行必要的预处理，包括缺失值处理、标准化或归一化等。确保数据在进行聚类之前处于一致的尺度和格式。由于之前的数据都已经进行了该项处理，故可以直接进行下一步。

2.选择聚类数目（K 值）：

K 均值聚类，确定要分为多个簇。通常可以通过可视化方法（肘部法则、轮廓系数等）来确定 K 值。

3.初始化聚类中心：

对于 K 均值聚类，开始时需要随机选择 K 个数据点作为初始聚类中心。

4.迭代聚类过程：

重复以下步骤，直到满足停止条件（例如，聚类中心不再发生显著变化）：a.分配每个数据点到最近的聚类中心。b.更新每个聚类中心为其成员数据点的平均值。

利用 Matlab 可以得出：

输出结果 1：字段差异性分析：

| | 聚类类别（平均值±标准差） | | F | P |
|---|---|---|---|---|
| | 类别 1(n=227) | 类别 2(n=19) | | |
| 销售额 | 920.25±1557.6 | 13793.637±6943.277 | 500.652 | 0.000*** |
| 商品名称 | 127.181±69.988 | 79.526±72.16 | 8.091 | 0.005*** |

注：***、**、*分别代表 1%、5%、10%的显著性水平

由此表可以得出两类不具有显著相关性，故可以进行分簇表示。

输出结果 2：聚类汇总：

| 聚类类别 | 频数 | 百分比% |
|---|---|---|
| 聚类类别_1 | 227 | 92.276 |
| 聚类类别_2 | 19 | 7.724 |
| 合计 | 246 | 100.0 |

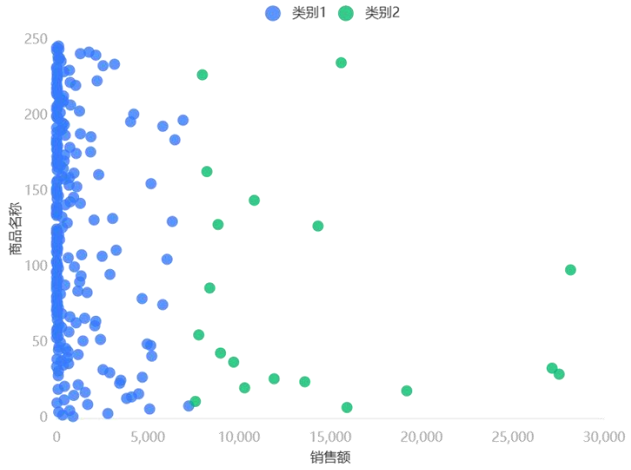



图 8 聚类分析图

由此图可以看出，大部分蔬菜单品总销售量都在 7500kg 以下，而少量的蔬菜单品大于 7500kg 以上，可以分析出当地人对这些单品需求量很大，反映出一些蔬菜商品的相关性。

### 5.2 问题二模型

问题二中涉及到“成本加成定价”策略，经过查阅得知：成本加成定价是一种确定产品价格的方法，它基于成本之上以一定利润率来确定最终的销售价格。即是一种城建的销售策略。而本题探寻蔬菜品类销售量和成本加成定价的关系，即需要探寻蔬菜品类销售量和利润率的关系。

问题二要求求出蔬菜品类销售量和成本加成定价（利润率）的关系，在这里使用 Excel 和 Matlab 处理后的数据，将画出的时间序列下的销售量和定价折线图进行初步判断，然后决定使用线性回归模型进行拟合，然后对拟合的函数关系进行拟合系数检验，检验通过则说明模型可以使用，则拟合成功，可以为后续定价决策提供依据。问题二又要求求出未来 7 天的日补货量和定价决策，本文决定利用求出的销售量和定价的关系，然后将利润率和日补货常量作为决策变量，构建以最大化商超收益为目标的线性规划模型。为了使日补货量和定价决策更为准确，本文决定使用处理之后附件二的庞大数据作为训练集利用 WSO_BiLSTM 预测模型进行销量的预测，如果日补货量大于预测销售量，则说明经过检验，原解合理。

#### 5.2.1 拟合数据处理

要求以蔬菜品类为准，探寻蔬菜品类的每天销售量和成本加成定价的关系，这就要求我们将附件三、附件四中的数据整合至附件二中，至此，处理后的附件二已经具备每次售卖每个单品的分类、名称、销售日期、销售时间、销售单价、销售量、销售单价、批发价格、损耗率等；故目前任何基于近 88 万数据的操作皆可在此处理后的附件二中进行处理，由于数据庞大，故已放在支撑材料中。

**每种蔬菜品类的每天的销售总量**

已经通过 Excel 和 Matlab 处理完成，即首先通过将每日中蔬菜单品的编号通过附件

一中的分类进行检索，确定分类后，计入该类的销售总量中，进行如此操作近 88 万次即可得出该数据。

**每种蔬菜品类每天的利润率**

处理之后的附件二是随时间序列的蔬菜单品的销售，故需要先算出蔬菜单品每天的利润率：

$$r_i = (\frac{x_i}{z_i} - 1) \times 100\%$$

其中 $x_i$ 表示第 i 种蔬菜单品的定价，$z_i$ 表示第 i 种蔬菜单品的批发价格。

将每个蔬菜单品每天的利润率算出后，接下来就要合并为蔬菜品类，我们可以利用每个单品的销售量进行赋加权系数：

$$R_j = \frac{p_i \cdot r_i}{\sum p_i} \times 100\%$$

其中，$R_j$ 代表第 j 个蔬菜品类的利润率，$p_i$ 表示第 i 种蔬菜单品的销售量，$r_i$ 表示第 i 种蔬菜单品的利润率，$\sum p_i$ 表示第 j 类蔬菜品类对应各种蔬菜单品的总销量。

以此种计算方法并借助 Excel 和 Matlab 从而得到对应数据表格，放在了支撑材料中。

**折线图对比**

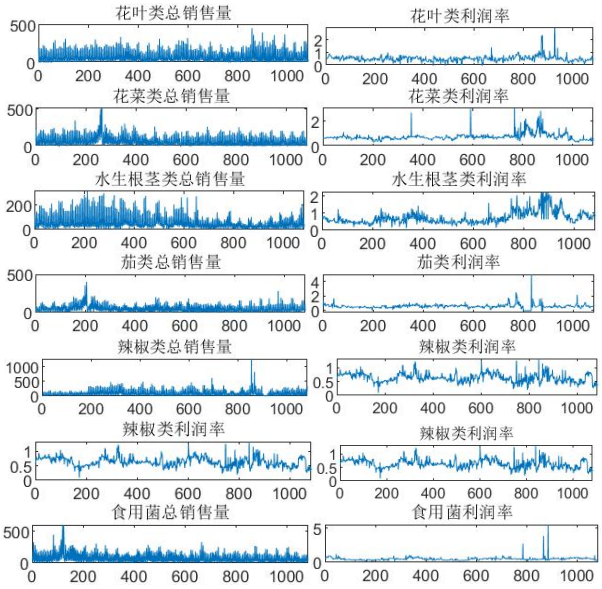



图 9 各类蔬菜品类销量和利润率随时间变化曲线

起初，在使用 Matlab 进行图像绘制时，发现有利润率很多值远远高于普遍值，于是经过 Excel 表格的筛选，发现了一定数量的异常值，使用“3σ 标准差原则”，将异常值筛选出，然后再使用该组数据的中位数进行替换。

**回归分析模型建立**

**线性回归拟合模型——失败**

首先经过线性回归拟合模型，经过 Matlab 和 Excel 处理之后，可以得到六类蔬菜品类的拟合图像，为了防止内容冗余，展示其中两种的线性拟合图像：

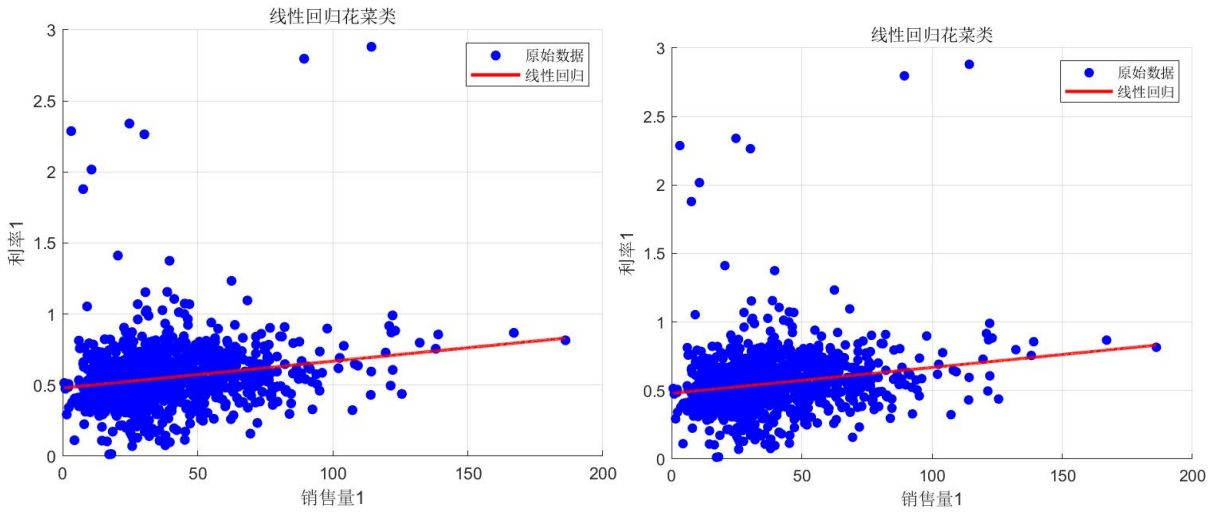



图 10 两类蔬菜品类的线性回归拟合模型

两者所求出的 r 值分别为 0.03 和 0.05，因此远远不符合拟合需要达到的程度，故舍去，因此该模型无法进行线性拟合。

**XGBOOST 回归拟合模型**

由于数据量庞大，因此可以使用神经网络思想的 XGBOOST 回归拟合模型，XGBOOS 回归拟合模型相当于一个封装的函数，即利用大量的数据将此函数变成一个利润率与销售量的关系函数，输入利润率即可得到预测的销售量。

它基于梯度提升机（Gradient Boosting Machine）的思想，通过集成多个决策树模型来提高预测性能。

1.从数据源中获取回归任务所需的特征和目标变量

2.将数据集分为训练集和测试集，训练集即为蔬菜各类 1085 天的利润率，测试集即为蔬菜各类 1085 天的销售量。

3.定义 XGBoost 回归模型的参数，即迭代次数、最大树深度、学习率、回归任务的目标函数。

4.将数据转换成 XGBoost 的数据格式

5.训练 XGBoost 回归模型（利用 Matlab 中的 XGBoost 库）

6.使用训练好的模型进行预测

7.计算均方误差 (MSE)

8.绘制预测值与实际值之间的折线拟合图。

利用以上逻辑去使用 Matlab 进行代码的编写，可以得到均方误差和拟合散点图：

| | |
|---|---|
| 花菜类 | 均方误差：0.02 |
| 花叶类 | 均方误差：0.01 |
| 水生根茎类 | 均方误差：0.004 |
| 茄类 | 均方误差：0.01 |
| 辣椒类 | 均方误差：0.35 |
| 食用菌 | 均方误差：0.01 |

由此均方误差表可知，该 XGBoost 回归拟合模型都到达了可以肯定原假设的水平，因此 XGBoost 回归模型可行，可以作为一个经过神经网络训练好的封装函数，可以用于以下数据建模的核心函数。

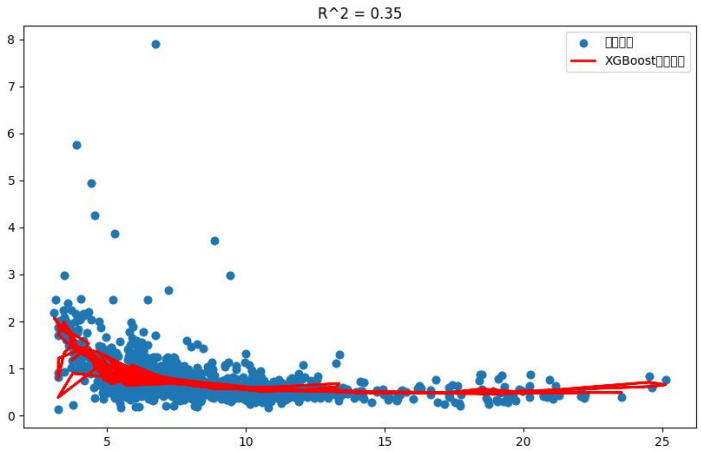



图 11 辣椒类拟合模型

**优化模型求解**

对问题二要求的未来七天六种蔬菜品类的日补货量和定价策略进行求解。定价策略就是制定合适的利润率。因此，此优化的模型的决策变量为日补货量 $a_{ij}$（i 代表蔬菜品类，j 代表天数），利率 $w_{ij}$ 两种决策变量。

由于六种蔬菜品类售卖得到的利润没有必然联系，因此可以分别使用优化模型进行计算，优化模型如下：

$$\text{目标函数：}\ Max: f(w_{ij})\cdot c_{ij}(1+w_{ij})-c_{ij}\cdot a_{ij}$$

$$\text{约束条件}\begin{cases}0<c_{ij}(1+w_{ij})<34\\ a_{ij}(1-s_i)>f(w_{ij})\\ a_{ij}>0\end{cases}$$

其中，$f(w_{ij})$ 是上述 XGBoost 回归模型训练出的预测销量函数，$s_i$ 为第 i 类蔬菜品类的亏损率，$c_{ij}$ 为第 i 类蔬菜品类第 j 天的成本。

利用 Matlab 进行基于遗传算法的非线性优化模型，利用 Matlab 中的遗传算法库，加入目标函数和约束条件，跑出 42 次模型求解，即可得出最终结果如下：

**利润率：**

| | 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 | 53.16% | 24.22% | 50.06% | 77.52% | 42.25% | 30.87% | 59.87% |
| 花叶类 | 50.14% | 21.55% | 45.21% | 33.54% | 42.25% | 35.54% | 44.87% |
| 水生根茎类 | 82.54% | 45.51% | 82.74% | 26.52% | 55.95% | 37.87% | 40.24% |
| 茄类 | 99.06% | 50.18% | 60.23% | 124.56% | 94.12% | 110.00% | 74.56% |
| 辣椒类 | 72.25% | 109.63% | 110.58% | 40.09% | 112.02% | 120.23% | 82.26% |
| 食用菌 | 34.05% | 104.52% | 117.56% | 89.63% | 93.98% | 95.53% | 55.98% |

**日补货量：**

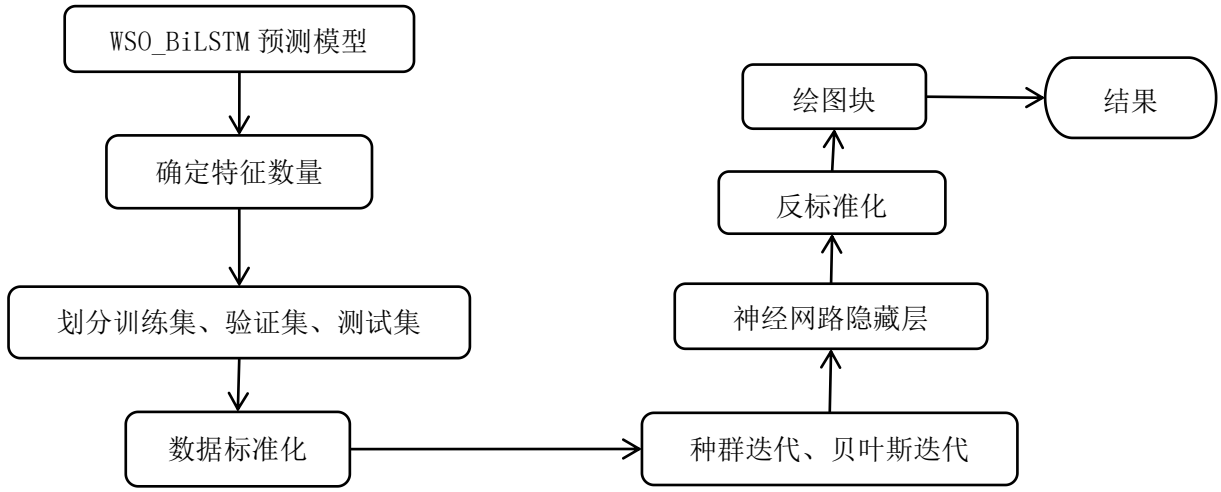
| | 7月1日 | 7月2日 | 7月3日 | 7月4日 | 7月5日 | 7月6日 | 7月7日 |
|---|---|---|---|---|---|---|---|
| 花菜类 | 55.25 | 21.52 | 42.25 | 39.66 | 54.23 | 31.54 | 30.01 |
| 花叶类 | 200.15 | 256.45 | 268.23 | 150.63 | 250.54 | 260.54 | 200.58 |
| 水生根茎类 | 21.85 | 11.68 | 15.65 | 9.52 | 20.54 | 10.54 | 16.54 |
| 茄类 | 10.25 | 7.89 | 5.45 | 11.52 | 3.54 | 6.59 | 6.02 |
| 辣椒类 | 78.54 | 110.25 | 112.52 | 40.25 | 114.58 | 110.78 | 65.58 |
| 食用菌 | 66.96 | 65.20 | 35.75 | 50.26 | 45.87 | 59.71 | 59.89 |

### 5.2.3 预测模型

本问以处理之后的附件二的庞大数据为训练集，构建 WSO_BiLSTM 预测模型去预测未来七天的销售量，考虑到要对数据进行充分的利用，本问决定以此模型来验证规划模型的合理性。

**WSO_BiLSTM 预测模型的建立**

（1）模型构建




按照此模型的建立过程，进行 Matlab 代码编程，以经过处理的每一类蔬菜品类的每天销售总量为基础数据，经过 490*7 次迭代然后可以预测出未来七天销售量(kg)的结果如下：

| | 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 |
|---|---|---|---|---|---|---|---|
| 花叶类 | 83.89956 | 48.55992 | 38.77241 | 37.9893 | 108.2921 | 49.71332 | 83.89956 |
| 花菜类 | 53.90938 | 18.51673 | 38.74285 | 37.96541 | 108.3105 | 29.65858 | 53.90938 |
| 水生根茎类 | 13.91919 | 8.47354 | 8.7133 | 3.94152 | 8.3289 | 4.60385 | 8.91919 |
| 茄类 | 3.92901 | 4.43034 | 8.68374 | 3.91764 | 10.3474 | 9.54911 | 3.92901 |
| 辣椒类 | 39.38873 | 51.8.715 | 38.65419 | 57.89375 | 100.3658 | 49.49437 | 93.93883 |
| 食用菌 | 73.94864 | 48.34396 | 38.62464 | 37.86987 | 10.8842 | 49.43963 | 56.94864 |

通过该表和上述规划模型得出的日补货量对比可知，日补货量量有 80%以上都高于预测销售量，因此，可以保证上述模型可以在一定程度上保证市场需求，因此可以说明数学模型的准确性。

### 5.3 问题三模型

问题三首先进行的是基础数据的筛选，然后建立以利润率和日补货量为决策变量，以总利润率为目标函数的优化模型，但是由于此模型的多维参数，导致传统的一些优化

模型，例如模拟退火，遗传算法等处理起来需要的算力极其庞大，因此本文决定采用粒子群算法进行规划，而在规划的过程中，有一些常量需要基于 6 月 24-30 日的数据进行预测，因此还涉及简单的时间预测模型。

### 5.3.1 数据筛选

原数据 6 月 24 日-30 日具有 251 中单品，为了简化之后的规划模型的运算，故需要进行提前的数据筛选，首先将不可售品种剔除，借助 Excel 数据透视表进行处理，处理后的表格放在第三问支撑材料中（见命名），然后由于 7 月 1 日是由这七天的数据预测的，因此可以认为倘若这七天每天的销售量均低于 2.5 千克的单品就可以剔除，依旧是 Excel 数据透视表进行处理，表格放在支撑材料中。

### 预测模型

第三问同样可以采用 WSO_BiLSTM 预测模型，以 6 月 24-30 日的经过筛选后的产品每天的成本价数据为训练集，从而预测出 7 月 1 日的筛选出的 n 个单品的当日成本和当日损耗率，为之后的规划模型打下基础。所采用的逻辑已经在第二问中指出，可以预测出以下结果：

| 蔬菜单品 | 预测成本 |
|---|---|
| 菜心 | 3.23 |
| 大白菜 | 1.50 |
| 高瓜(1) | 7.01 |
| ... | ... |
| 长线茄 | 6.57 |
| 竹叶菜 | 1.95 |
| 紫茄子(2) | 4.30 |

### 基于粒子群算法的规划模型

由于针对筛选后的 n 种蔬菜单品，7 月 1 日可以选定多种不同组合，故引入一种筛选变量 $q_i$ (取 0 或 1)，这样取值为 1 代表该蔬菜单品当天销售，取值为 0 代表该蔬菜单品当天未出售，就可以将所有选择进行对比，找到最优解。

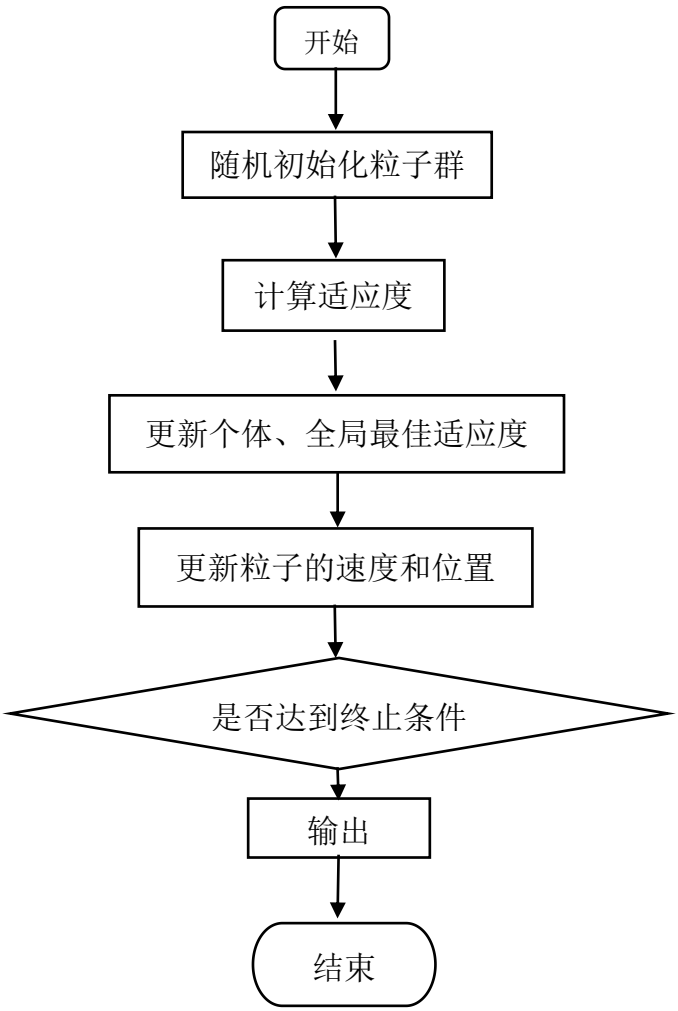
决策变量：利润率 $w_i$ 和补货量 $x_i$

$$\text{目标函数：}\ Max:\sum_{i=1}^{n} q_i(1+w_i)c_i\cdot f(w_i)-x_i\cdot c_i$$

$$\text{约束条件}\begin{cases} x_i \geq 2.5 \\ 27 \leq \sum_{i=1}^{n} q_i \leq 33 \\ q_i = 0\text{或}1 \\ x_i(1-s_i) > f(w_i) \end{cases}$$

其中，n 表示经过筛选的蔬菜单品的数量，$f(w_i)$ 表示第 i 个蔬菜单品的销售量，$q_i$ 取 0 或 1，$c_i$ 表示第 i 个蔬菜单品的成本（为上述预测模型所预测数据），$s_i$ 为第 i 个蔬菜单品的损耗率。

**粒子群优化过程：**




结果如下：

| 单品 | 日补货量（KG） | 定价策略（%） | 单品 | 日补货量 | 定价策略（%） |
|---|---|---|---|---|---|
| 本地小毛白菜 | 3.7 | 110.85 | 红椒(2) | 2.78 | 106.4 |
| 云南生菜 | 9.98 | 132.54 | 蟹味菇与白玉菇双拼(盒) | 2.01 | 86.21 |
| 茼蒿 | 9.56 | 134.89 | 洪湖莲藕(粉藕) | 8.12 | 80.54 |
| 牛首油菜 | 15.88 | 94.4 | 小青菜(1) | 5.41 | 130.54 |
| 紫茄子(2) | 3.00 | 50.02 | 云南生菜(份) | 12.54 | 100.89 |
| 西峡香菇(1) | 18.23 | 65.21 | 云南油麦菜(份) | 10.23 | 95.21 |
| 西兰花 | 25.88 | 130.54 | 菠菜(份) | 14.00 | 120.54 |
| 净藕(1) | 2.20 | 60.78 | 鱼腥草(份) | 1.06 | 70.54 |
| 枝江红菜苔 | 2.55 | 130.54 | 小米椒(份) | 20.48 | 193.54 |
| 白玉菇(袋) | 29.87 | 60.45 | 冰草(盒) | 58.21 | 130.54 |
| 芜湖青椒(1) | 3.54 | 140.56 | 金针菇(盒) | 25.12 | 78.21 |
| 杏鲍菇(2) | 15.00 | 75.98 | 黄白菜(2) | 7.94 | 100.01 |
| 奶白菜(份) | 11.21 | 54.21 | 圆茄子(2) | 4.03 | 40.52 |
| 双孢菇(盒) | 4.56 | 60.66 | 青红杭椒组合装（份） | 6.00 | 65.42 |

## 5.4 问题四模型

## 5.5 问题四的求解

为了更好地制定蔬菜商品的补货和定价决策，商家可采集一下相关数据来解决该问题：

**（一）消费者的购买情况：**

1.购买历史数据：统计顾客的购买历史数据，包括购买的蔬菜品类、购买频率、购买渠道等。通过分析购买的情况，商超可以识别畅销产品和不受欢迎产品，以调整库存和补货计划。

2.购物车订单分析：分析每次订单顾客购物车中的组合，了解哪些蔬菜品类通常一起购买。这有助于商超制定促销活动，推动相关品类的销售。

3.购买频率：确定顾客购买蔬菜的频率，例如每周、每月或季度。这将有助于商超规划库存和补货计划，以满足不同时间段的需求。

**（二）商品的供应：**

1. 供应商数据：收集与不同供应商的合作信息，包括供应商的交货时间、产品质量和合同条款，以此了解每个供应商的可靠性和表现，以决定是否需要更换供应商。

2. 库存水平：跟踪不同蔬菜品类的库存水平，以确保库存处于适当的水平。分析库存数据有助于防止过多或过少的库存，减少损失和滞销商品。

3. 产品质量数据：收集关于蔬菜品质的数据，包括新鲜度、损耗率和质量评级。这有助于商超选择高质量的产品供应商，并确保产品的质量符合顾客期望。

4. 供应链效率：分析供应链的效率和流程，以识别需要的改进点。这可以帮助商超降低供应链成本并提高运营效率。

**（三）相竞争的商超：**

1. 销售数据：收集竞争对手的销售数据，包括销售额、销售量、销售渠道等。这将帮助商超了解竞争对手的市场份额和销售趋势。

2. 定价策略：了解竞争对手的定价策略，包括价格水平、折扣率、促销策略等。这有助于商超确定是否需要调整自己的价格以保持竞争力。

3.市场定位：了解竞争对手的市场定位和目标客户群体。这有助于商超确定自己的营销策略。

**（四）市场数据：**

1. 价格趋势：分析蔬菜品类的价格趋势，了解价格是否有季节性变化，以及市场上价格高低的波动。这将有助于商超确定定价策略，以在竞争激烈的市场中保持竞争力。

2. 市场份额：了解商超在蔬菜市场中的市场份额，以及与竞争对手的市场份额比较。这有助于商超评估自己在市场中的优势与不足，并识别潜在的增长机会。

**（五）促销数据：**

1. 促销效果分析：对于不同促销活动，包括折扣率、促销时段、促销渠道等。通过分析促销活动的数据，商超可以确定哪些促销策略最有效，以及如何优化促销活动。

2. 促销时段分析：分析不同时间段的促销效果，了解哪个季节、哪个时间段的促销更具吸引力。这有助于商超在关键时段提供有吸引力的促销活动。

3. 促销渠道分析：评估不同促销渠道的效果，包括线上促销、线下促销和移动应用促销等。这可以帮助商超确定哪些渠道最适合他们的目标市场。

4. 促销产品分析：了解哪些产品在促销期间销售增长最显著。这将有助于商超确定促销产品的选择和库存管理。

5. 促销定价策略：分析不同促销活动的定价策略，包括折扣率、买赠活动等。根据反馈情况有助于商超确定如何制定具有吸引力的价格策略。

**（六）天气因素：**

1. 温度分析：了解温度对不同蔬菜品类的销售影响。考虑温度的改变对蔬菜损耗和品质的影响，来制定更好的补货与定价决策、。

2. 降水量分析：分析降水量对销售的影响。过多或不足的降水可能会影响蔬菜的供应和品质，从而影响销售。

3. 天气预测：使用天气预测数据来预测未来的天气状况，以便调整库存和补货计划。如果预测到极端天气，商超可以提前做好备货准备。

4. 天气警报和灾害预警：监测天气警报和灾害预警，以及其对供应链和店铺运营的潜在影响。及时采取措施以减少潜在的天气风险。

**（七）商超的地理因素：**

1. 店铺位置分析：评估不同店铺的位置，包括城市、社区或农村地区。了解店铺位置的特点，如人口密度、周边竞争对手、消费水平等。

2. 地区销售差异：比较不同地区店铺的销售数据，了解地区之间的销售差异。这可以帮助商超优化产品组合和促销策略，以满足不同地区的需求。

3. 交通和物流：通过深入分析地理因素，分析地理因素对交通和物流的影响，以优化货物配送和库存管理，商超可以更好地适应不同地理区域的需求和市场条件，制定更精确的销售策略和运营计划，以提高业务表现。地理因素分析有助于商超更好地满足不同地区的需求，并在竞争激烈的市场中取得成功。

**（八）季节因素：**

1. 季节性销售趋势：分析不同蔬菜品类在不同季节的销售趋势。了解哪些产品在特定季节更受欢迎，以确定季节性供应和促销策略。

2. 季节性产品选择：根据季节性需求选择适当的蔬菜品种。季节性产品选择可以确保产品供应和销售的连续性。

3. 季节性供应链管理：与供应商协调季节性供应链管理，以确保在需求高峰期能够及时供应足够的产品。

**（九）损耗率与质量控制数据：**

1. 损耗率趋势分析：分析不同蔬菜品类的损耗率趋势，了解哪些产品容易损失，并在哪个季节或时段损耗率较高。这有助于商超调整库存管理和补货计划。

2. 损耗原因分析：确定导致损耗的主要原因，例如运输损耗、储存条件不当或过期产品等。这将有助于商超采取纠正措施减少损耗。

3. 供应商质量评估：评估不同供应商提供的产品的质量，包括损耗率、品相和新鲜度等。这有助于商超选择可靠的供应商。

# 六、模型的优缺点、改进

## 6.1 模型的优点

1. 模型可视化丰富：对于该模型使用 Matlab 绘图极其丰富，在对数据的预处理何分析上提供了很大的帮助，使该模型更加具有可视性，在会绘图方面常采用对比分析，使数据变化更加具有启发性，有助于深入了解数据的特征和关联性。

2. 数据的处理完备：本模型对 Excel 表格数据中的处理十分完备，将所有附件中的数据皆可一一整合，并大多进行的加权平均，使得该模型之后的数据处理变得极大便利。

并且，在对数据预处理上利用了 3σ 原则检测异常值，确保模型输入数据质量更高，从而提高了模型的可靠性。

3.预测模型充当检测模型：对于规划模型得到的结果，为了保证模型的可靠性，再次建立时间序列的预测模型，将规划所得数据进行了预测，对比规划与预测模型的数据，进行检验，证明了模型建立的可靠性。

4.优化模型：本模型针对不同的数据特点，比如参数的维度、数据的大小、约束条件的多少进行优化模型的选取，比如模拟退火优化模型、粒子群优化模型等，同样增加了模型的可靠性。

5.拟合模型的多次检验：在本文模型中，首先我们先猜测的线性拟合模型，结果得到的均方误差达到了很大的水平，因此舍弃，采用非线性拟合，考虑到数据的庞大性，故我们利用神经网络进行训练，于是达到了一个拟合效果非常不错的模型，故使模型预测更加准确。

### 6.2 模型的缺点

1.启发式算法导致每次所求的优化模型结果每次出现的局部最优解不一样，故需要进行多次运算才能确定一个较为优化的解，故使得模型运算量较为庞大。

2.模型的推广存在困难：模型建立之后，实施到实际商场运行中会面临较大的困难，需要考虑多种其他因素，例如数据的处理和集成、决策的更新等

3.模型的非精确泛化性：该模型建立时最多只能提供日决策，无法提供更多更详细的数据，导致模型趋于泛化，不太灵活。

4.算法优化没有达到理想，使用的大多是一种典型算法，而非组合模型。

# 七、参考文献

[1]亚辉 W，超杰 M，Chong W， 等.基于改进粒子群优化算法的桥式起重机特性分析与优化设计[J].低频噪声振动与有源控制学报，2023,42（1）.
[2]王坤章，蒋书波，张豪等.基于 XGBoost 的回归-分类-回归寿命预测模型[J].中国测试，2023,49（08）：104-109.
[3]赵敏,高建华.结合遗传规划和遗传算法的代码异味检测方法[J].小型微型计算机系统,2020,41(11):2434-2441.
[4]伊丽莎白 F，南希 C，玛丽塞拉 L 等.与秘鲁牙科学生孕妇药理学管理知识相关的因素：逻辑回归分析。BMC 医学教育，2023,23（1）.
H Z， et al. 基于深度学习的时间序列预测模型评估用于预测海域叶绿素 A 和溶解氧[J].海洋科学与工程学报，2023,11（7）.
[1]Andreu L J F，Morales L A J，Guillen H Z， et al. 基于深度学习的时间序列预测模型评估用于预测海域叶绿素 A 和溶解氧[J].海洋科学与工程学报，2023,11（7）.

# 附录一

**支撑材料目录：**

**问题一：**

数据文件：

各品类的总销售量统计. xlsx
各品类各时间段销售量. xlsx
各单品各时间段销售量统计. xlsx
各品类各时间段打折数据统计.xlsx
每日每类均定价计算. xlsx

代码，图形文件：

6 种类随每天的销量变化. pdf
six_sorts_correlation_analysis.m
six_sorts_days.m
The_varieties_of_days_of_six_sorts_of_vegetables.m
myplot.png
不同蔬菜品类销售量占比. png
不同蔬菜品类总销量. png
折线图. png
柱状图. png

**问题二：**

数据文件：

各品类均利率. xlsx
每日各品类销售量用于处理回归.xlsx
每日销售定价用于处理回归. xlsx
七月各品类成本与损耗率. xlsx
通过回归得出的销售量与定价函数关系. xlsx

代码与图形文件：

XGBoost 模型（文件夹）
第二问线性回归（文件夹）
关于销售量与定价的线性回归分析（文件夹）
规划模型（文件夹）
预测模型（文件夹）
折线图（文件夹）

**问题三：**

数据文件：

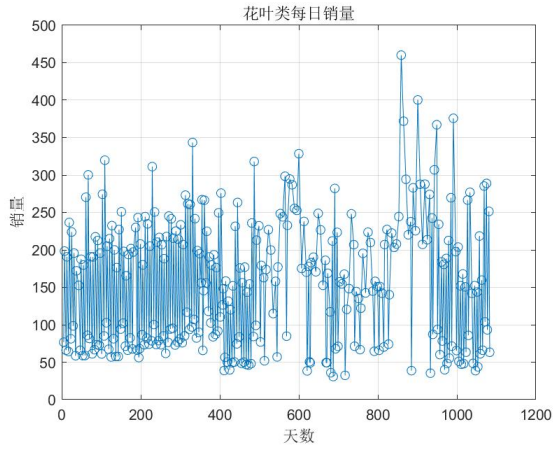
各品类七天均损耗率. xlsx
基础数据统合. xlsx
七月各品类均成本. xlsx
筛出七天内的 39 种满足条件的.xlsx

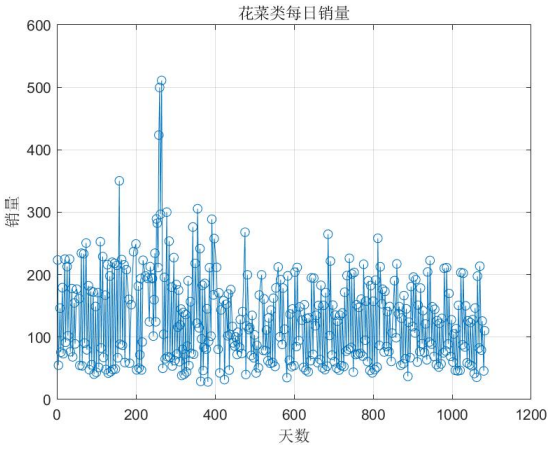
代码图形文件：

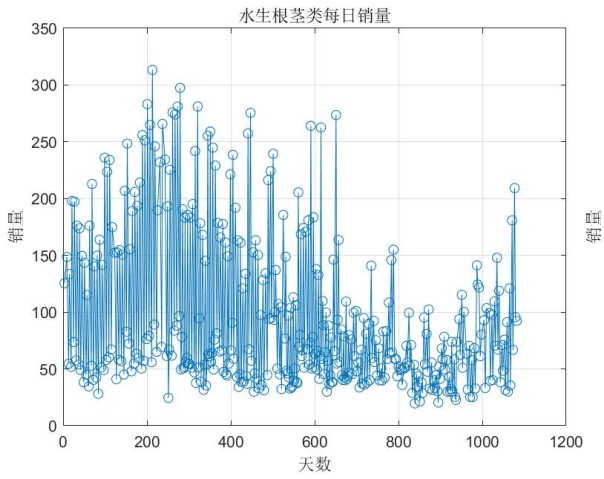
Predict_cost.m
第三问预测数据.xlsx

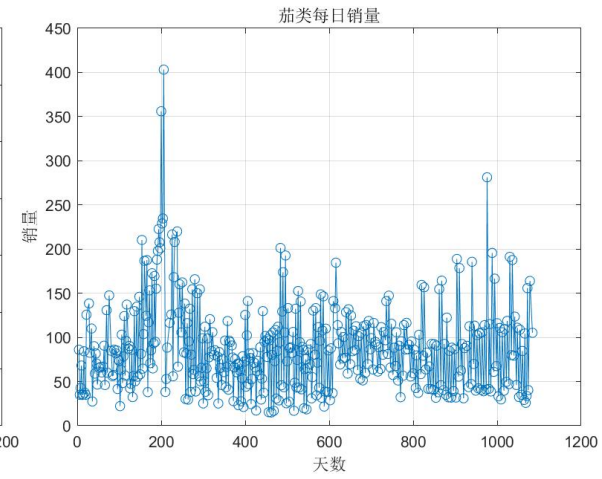
# 附录二

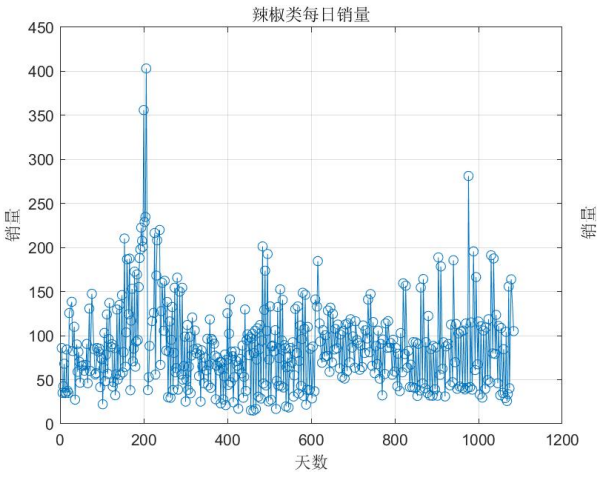
**图表：**

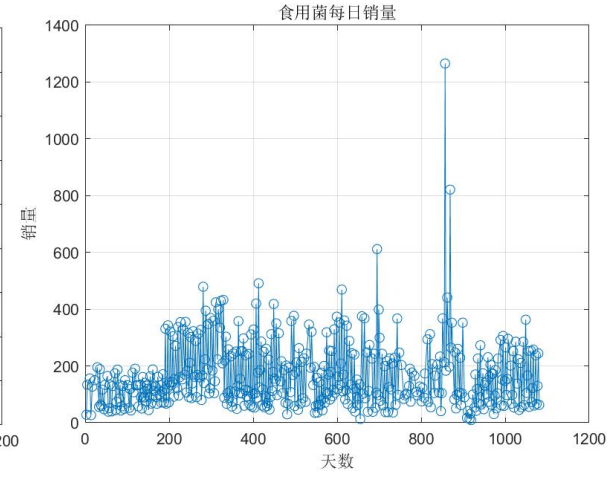

花叶类每日销量
销量
天数

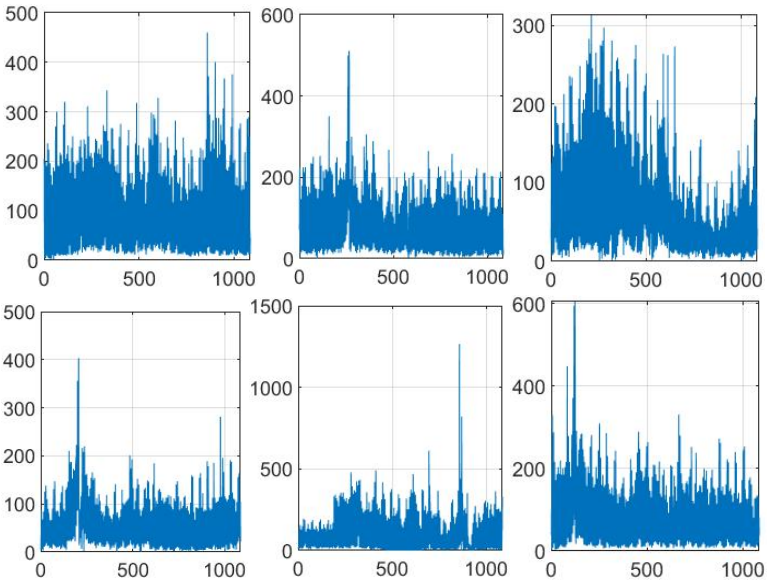

花菜类每日销量
销量
天数

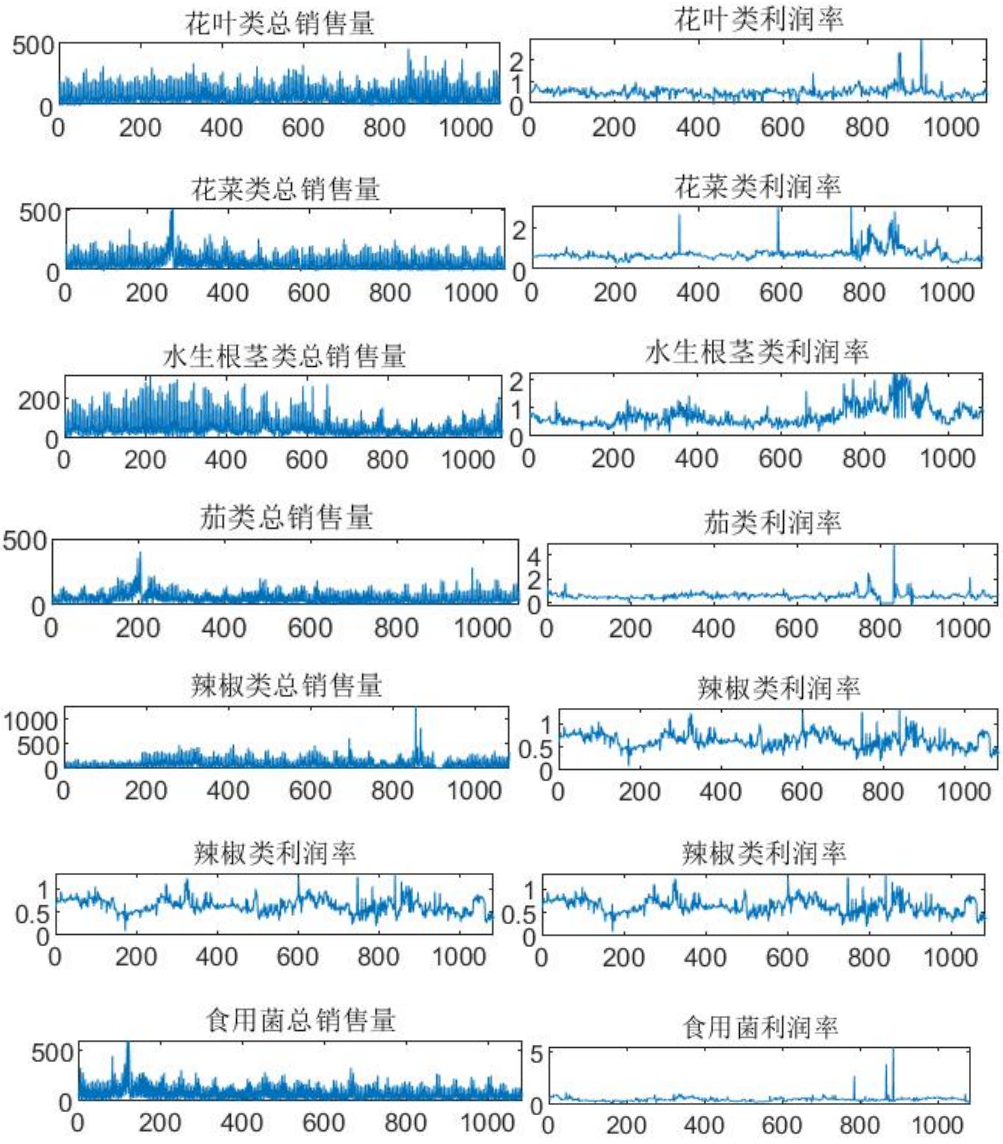

水生根茎类每日销量
销量
天数


茄类每日销量
销量
天数

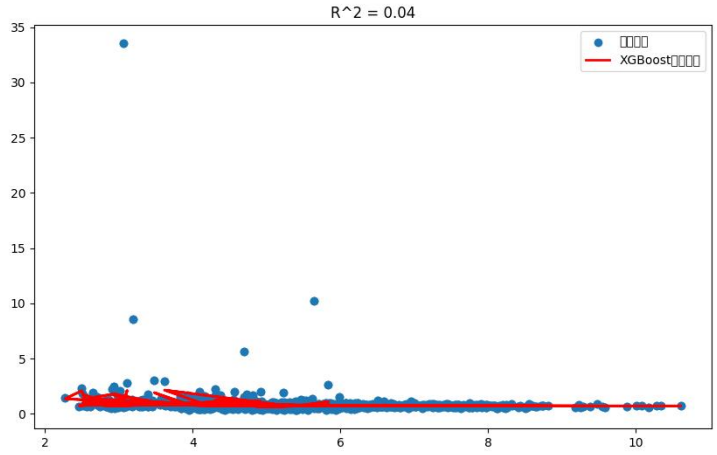

辣椒类每日销量
销量
天数


食用菌每日销量
销量
天数

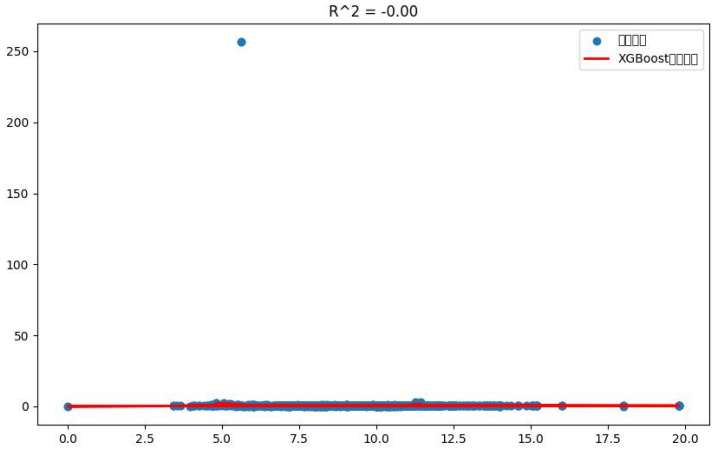



花叶类总销售量
花叶类利润率
花菜类总销售量
花菜类利润率
水生根茎类总销售量
水生根茎类利润率
茄类总销售量
茄类利润率
辣椒类总销售量
辣椒类利润率
辣椒类利润率
辣椒类利润率
食用菌总销售量
食用菌利润率

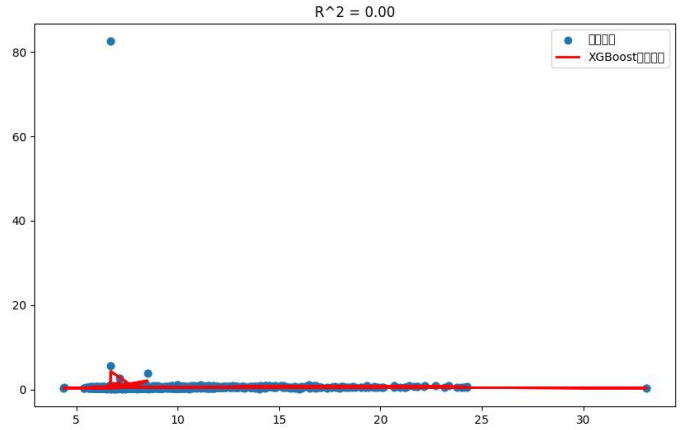
## 花叶类



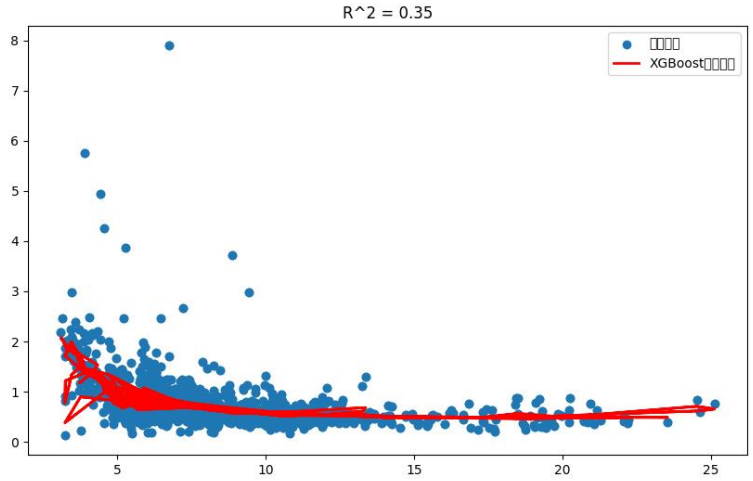

## 花菜类



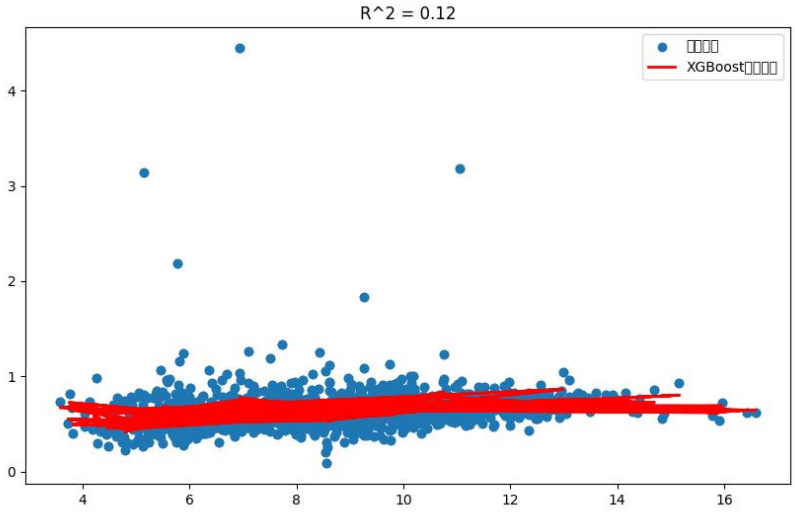

## 水生根茎类



# 辣椒类


R^2 = 0.35
□□□□
XGBoost□□□□
8
7
6
5
4
3
2
1
0
5
10
15
20
25

# 食用菌


R^2 = 0.12
□□□□
XGBoost□□□□
4
3
2
1
0
4
6
8
10
12
14
16

# 附录三

## 代码：

```
clc,clear;
x=[1:1085]
%读取数据
average_price=xlsread('各品类均利率.xlsx')
%%%%%y 是 total_sale,z 是 average_price
y1=[205.402000000000    46.6400000000000    4.85000000000000    35.3740000000000    76.7150000000000    35.3650000000000    19
    8.362000000000    43.9430000000000    4.60000000000000    ..........]'
z6=average_price(:,6)

% 查找异常值（使用 3 倍标准差法）
mean_val = mean(z1);
std_dev = std(z1);
threshold = 3 * std_dev;

outliers = abs(z1 - mean_val) > threshold;

% 处理异常值（删除或替换为中位数）
z1(outliers) = median(z1);

% 数据已处理后的结果
disp(z1);


mean_val = mean(z2);
std_dev = std(z2);
threshold = 3 * std_dev;

outliers = abs(z2 - mean_val) > threshold;

% 处理异常值（删除或替换为中位数）
z2(outliers) = median(z2);

% 数据已处理后的结果
disp(z2);


%%%
mean_val = mean(z3);
std_dev = std(z3);
threshold = 3 * std_dev;

```

```
outliers = abs(z3 - mean_val) > threshold;

% 处理异常值（删除或替换为中位数）
z3(outliers) = median(z3);

% 数据已处理后的结果
disp(z3);



%%%%%%%%%%
mean_val = mean(z4);
std_dev = std(z4);
threshold = 3 * std_dev;

outliers = abs(z4 - mean_val) > threshold;

% 处理异常值（删除或替换为中位数）
z4(outliers) = median(z4);

% 数据已处理后的结果
disp(z4);

% 查找异常值（使用 3 倍标准差法）
mean_val = mean(z5);
std_dev = std(z5);
threshold = 3 * std_dev;

outliers = abs(z5 - mean_val) > threshold;

% 处理异常值（删除或替换为中位数）
z5(outliers) = median(z5);

% 数据已处理后的结果
disp(z5);

% 查找异常值（使用 3 倍标准差法）
mean_val = mean(z6);
std_dev = std(z6);
threshold = 3 * std_dev;

outliers = abs(z6 - mean_val) > threshold;

% 处理异常值（删除或替换为中位数）
```

```
z6(outliers) = median(z6);

% 数据已处理后的结果
disp(z6);


subplot(6,2,1)
plot(x, y1);
title('花叶类总销售量');

subplot(6,2,2)
plot(x, z1);
title('花叶类利润率');

subplot(6,2,3)
plot(x, y2);
title('花菜类总销售量');

subplot(6,2,4)
plot(x, z2);
title('花菜类利润率');

subplot(6,2,5)
plot(x, y3);
title('水生根茎类总销售量');

subplot(6,2,6)
plot(x, z3);
title('水生根茎类利润率');

subplot(6,2,7)
plot(x, y4);
title('茄类总销售量');

subplot(6,2,8)
plot(x, z4);
title('茄类利润率');
```

```

subplot(6,2,9)
plot(x, y5);
title('辣椒类总销售量');


subplot(6,2,10)
plot(x, z5);
title('辣椒类利润率');


subplot(6,2,11)
plot(x, y6);
title('食用菌总销售量');


subplot(6,2,12)
plot(x, z6);
title('食用菌利润率');
```

```
clc,clear;
x=[1:1085]
y1=[205.402000000000    46.6400000000000    4.85000000000000    35.3740000000000    76.7150000000000    35.3650000000000.......
.]'
[peaks, peakLocations] = findpeaks(y1, x);
figure
plot(peakLocations, peaks, '-o')
title('花叶类每日销量')
xlabel('天数')
ylabel('销量')
grid on;
saveas(gcf, 'myplot.png');

[peaks, peakLocations] = findpeaks(y2, x);
figure
plot(peakLocations, peaks, '-o')
title('花菜类每日销量')
xlabel('天数')
ylabel('销量')
grid on;

[peaks, peakLocations] = findpeaks(y3, x)
```

```
figure
plot(peakLocations, peaks, '-o')
title('水生根茎类每日销量')
xlabel('天数')
ylabel('销量')
grid on;

[peaks, peakLocations] = findpeaks(y4, x)
figure
plot(peakLocations, peaks, '-o')
title('茄类每日销量')
xlabel('天数')
ylabel('销量')
grid on;

[peaks, peakLocations] = findpeaks(y4, x)
figure
plot(peakLocations, peaks, '-o')
title('辣椒类每日销量')
xlabel('天数')
ylabel('销量')
grid on;

[peaks, peakLocations] = findpeaks(y5, x)
figure
plot(peakLocations, peaks, '-o')
title('食用菌每日销量')
xlabel('天数')
ylabel('销量')
grid on;
```

```
clc,clear;
x=[1:1085]
y1=[205.402000000000    46.6400000000000    4.85000000000000    35.3740000000000    76.7150000000000    35.3650000000000.....
..]'
figure
subplot(2,3,1)
plot(x, y1);
grid on;

subplot(2,3,2)
plot(x, y2);
```

```
grid on;

subplot(2,3,3)
plot(x, y3);
grid on;

subplot(2,3,4)
plot(x, y4);
grid on;

subplot(2,3,5)
plot(x, y5);
grid on;

subplot(2,3,6)
plot(x, y6);
grid on;
```

```
import numpy as np
import xgboost as xgb
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score

import  pandas as pd
excel_file1 = "第 2 步，每日销售定价用于处理回归.xlsx"
excel_file2 = "第 4 步，各品类均利率.xlsx"
df1 = pd.read_excel(excel_file1)
df2 = pd.read_excel(excel_file2)
X=df1['食用菌']
y=df2['食用菌']

# 合并数据
X_all = np.concatenate([X, X])
y_all = np.concatenate([y, y])

# 异常值处理：删除超过 3 倍标准差的异常值
mean_y = np.mean(y_all)
std_y = np.std(y_all)
threshold = 3 * std_y
X_clean = X_all[abs(y_all - mean_y) < threshold]
y_clean = y_all[abs(y_all - mean_y) < threshold]

```

```
# 定义 XGBoost 回归模型参数
params = {
    'objective': 'reg:squarederror',  # 回归任务的目标函数
    'n_estimators': 100,              # 迭代次数
    'max_depth': 3,                   # 最大树深度
    'learning_rate': 0.1              # 学习率
}

# 创建并拟合 XGBoost 模型
model = xgb.XGBRegressor(**params)
model.fit(X_clean.reshape(-1, 1), y_clean)

# 预测并计算 R 值
y_pred = model.predict(X_all.reshape(-1, 1))
r_squared = r2_score(y_all, y_pred)

# 绘制原始数据和拟合曲线
plt.figure(figsize=(10, 6))
plt.scatter(X_all, y_all, label='原始数据')
plt.plot(X_all, y_pred, color='red', linewidth=2, label='XGBoost 拟合曲线')
plt.title(f'R^2 = {r_squared:.2f}')
plt.legend()
plt.show()
```

```
% 定义文件路径和数据集名称
file_paths = {'各品类均利率.xlsx', '各品类销售量.xlsx'};
dataset_names = {'花菜类', '花叶类', '辣椒类', '茄类', '食用菌类', '水生根茎类'};
data1=readtable("各品类均利率.xlsx");
data1.Properties.VariableNames(1) = "date";
data1.Properties.VariableNames(2) = "veg1";
data1.Properties.VariableNames(3) = "veg2";
data1.Properties.VariableNames(4) = "veg3";
data1.Properties.VariableNames(5) = "veg4";
data1.Properties.VariableNames(6) = "veg5";
data1.Properties.VariableNames(7) = "veg6";
datasale=readtable("各品类销售量.xlsx");
datasale.Properties.VariableNames(1) = "date";
datasale.Properties.VariableNames(2) = "veg1";
datasale.Properties.VariableNames(3) = "veg2";
datasale.Properties.VariableNames(4) = "veg3";
datasale.Properties.VariableNames(5) = "veg4";
datasale.Properties.VariableNames(6) = "veg5";
datasale.Properties.VariableNames(7) = "veg6";
```

```
20.
21.     % 循环处理每组数据
22.     for i = 1:6
23.         % 导入数据
24.
25.
26.         % 提取销售量和利率
27.         sale_data = datasale.(['veg' num2str(i)]);
28.         rate_data = data1.(['veg' num2str(i)]);
29.
30.         % 找出非零的数据点的索引
31.         nonzero_indices = (rate_data ~= 0);
32.
33.         % 使用逻辑索引筛选非零的数据点
34.         x_nonzero = sale_data(nonzero_indices);
35.         y_nonzero = rate_data(nonzero_indices);
36.
37.         % 异常值处理（三倍标准差法）
38.         mean_y = mean(y_nonzero);
39.         std_y = std(y_nonzero);
40.         threshold = 3*std_y;
41.         outliers = abs(y_nonzero - mean_y) > threshold;
42.
43.         mean_x = mean(x_nonzero);
44.         std_x = std(x_nonzero);
45.         threshold = 3*std_x;
46.         outliers = abs(x_nonzero - mean_x) > threshold;
47.         % 将异常值更改为中位数
48.         y_nonzero(outliers) = median(y_nonzero);
49.         x_nonzero(outliers) = median(x_nonzero);
50.         % 执行一元线性回归
51.         coefficients = polyfit(x_nonzero, y_nonzero, 1);
52.
53.         % 提取斜率和截距
54.         slope = coefficients(1);
55.         intercept = coefficients(2);
56.
57.         % 绘制原始数据点和回归线
58.         figure;
59.         scatter(x_nonzero, y_nonzero, 'b', 'filled');
60.         hold on;
61.         x_range = min(x_nonzero):0.01:max(x_nonzero);
62.         y_fit = slope * x_range + intercept;
63.         plot(x_range, y_fit, 'r', 'LineWidth', 2);
```

```
        legend('原始数据', '线性回归');
        xlabel(['销售量' num2str(i)]);
        ylabel(['利率' num2str(i)]);
        title(['线性回归' dataset_names{i}]);
        grid on;
        % 在执行一元线性回归后，获取回归系数
        % 在执行一元线性回归后，获取回归系数


        % 计算残差
        y_fit = slope * x_nonzero + intercept;
        residuals = y_nonzero - y_fit;

        % 计算总平方和
        total_sum_of_squares = sum((y_nonzero - mean(y_nonzero)).^2);

        % 计算残差平方和
        residual_sum_of_squares = sum(residuals.^2);

        % 计算自变量数量
        num_predictors = 1;  % 如果有多个自变量，根据实际情况更改

        % 计算调整的 R-squared 值
        n = length(y_nonzero);  % 样本数量
        adjusted_r_squared = 1 - ((n - 1) / (n - num_predictors - 1)) * (residual_sum_of_squares / total_sum_of_squares);

        % 输出调整的 R-squared 值
        fprintf('调整的 R-squared 值：%f\n', adjusted_r_squared);
        fprintf('a=%f,b=%f\n',slope,intercept)


            % 在这里可以根据需要进一步汇总或保存回归结果
        end
```

```
% 定义文件路径和数据集名称
file_paths = {"C:\Users\边锋 BF\Desktop\关于销售量与定价的回归分析\销售定价.xlsx" ,"C:\Users\边锋 BF\Desktop\关于销售量与定价的回归分析\销售量.xlsx", };
dataset_names = {'花菜类', '花叶类', '辣椒类', '茄类', '食用菌类', '水生根茎类'};
data1=readtable("销售量.xlsx" );
data1.Properties.VariableNames(1) = "date";
data1.Properties.VariableNames(2) = "veg1";
data1.Properties.VariableNames(3) = "veg2";
```

```
data1.Properties.VariableNames(4) = "veg3";
data1.Properties.VariableNames(5) = "veg4";
data1.Properties.VariableNames(6) = "veg5";
data1.Properties.VariableNames(7) = "veg6";
datasale=readtable( "销售定价.xlsx"  );
datasale.Properties.VariableNames(1) = "date";
datasale.Properties.VariableNames(2) = "veg1";
datasale.Properties.VariableNames(3) = "veg2";
datasale.Properties.VariableNames(4) = "veg3";
datasale.Properties.VariableNames(5) = "veg4";
datasale.Properties.VariableNames(6) = "veg5";
datasale.Properties.VariableNames(7) = "veg6";

% 循环处理每组数据
for i = 1:6
    % 导入数据



    % 提取销售量和利率
    sale_data = datasale.(['veg' num2str(i)]);
    rate_data = data1.(['veg' num2str(i)]);

    % 找出非零的数据点的索引
    nonzero_indices = (rate_data ~= 0);

    % 使用逻辑索引筛选非零的数据点
    x_nonzero = sale_data(nonzero_indices);
    y_nonzero = rate_data(nonzero_indices);

    % 异常值处理（三倍标准差法）
    mean_y = mean(y_nonzero);
    std_y = std(y_nonzero);
    threshold = 3* std_y;
    outliers = abs(y_nonzero - mean_y) > threshold;

    % 将异常值更改为中位数
    y_nonzero(outliers) = median(y_nonzero);

    % 执行一元线性回归
    coefficients = polyfit(x_nonzero, y_nonzero, 1);

    % 提取斜率和截距
    slope = coefficients(1);
    intercept = coefficients(2);
```

```
52.
53.         % 绘制原始数据点和回归线
54.         figure;
55.         scatter(x_nonzero, y_nonzero, 'b', 'filled');
56.         hold on;
57.         x_range = min(x_nonzero):0.01:max(x_nonzero);
58.         y_fit = slope * x_range + intercept;
59.         plot(x_range, y_fit, 'r', 'LineWidth', 2);
60.         legend('原始数据', '线性回归');
61.         xlabel(['销售量' num2str(i)]);
62.         ylabel(['利率' num2str(i)]);
63.         title(['线性回归' dataset_names{i}]);
64.         grid on;
65.         % 在执行一元线性回归后，获取回归系数
66.         % 在执行一元线性回归后，获取回归系数
67.
68.
69.         % 计算残差
70.         y_fit = slope * x_nonzero + intercept;
71.         residuals = y_nonzero - y_fit;
72.
73.         % 计算总平方和
74.         total_sum_of_squares = sum((y_nonzero - mean(y_nonzero)).^2);
75.
76.         % 计算残差平方和
77.         residual_sum_of_squares = sum(residuals.^2);
78.
79.         % 计算自变量数量
80.         num_predictors = 1;  % 如果有多个自变量，根据实际情况更改
81.
82.         % 计算调整的 R-squared 值
83.         n = length(y_nonzero);  % 样本数量
84.         adjusted_r_squared = 1 - ((n - 1) / (n - num_predictors - 1)) * (residual_sum_of_squares / total_sum_of_squares);
85.
86.         % 输出调整的 R-squared 值
87.         fprintf('调整的 R-squared 值：%f\n', adjusted_r_squared);
88.         fprintf('斜率：%f 截距：%f\n',slope,intercept);
89.
90.
91.
92.             % 在这里可以根据需要进一步汇总或保存回归结果
93.         end
```

```
% 定义不等式约束条件
nonlcon = @(x) constraints(x);

% 定义变量的下界和上界
lb = [0; 0];  % 变量下界
ub = [Inf; Inf];  % 变量上界

% 使用遗传算法求解带不等式约束的优化问题
[x, fval] = ga(fitnessFunction, 2, [], [], [], [], lb, ub, nonlcon, options);

disp('最优解为：');
disp(x);
disp('目标表达式的最大值等于：');
disp(-fval);

% 定义目标函数（包括惩罚项）
function [obj] = obj_function(x)
    % 目标函数
    obj = 6.639983587*(-2.636564*x(1)+63.0362159)*(1+x(1))-x(2)*6.639983587;
end

% 定义不等式约束条件
function [c, ceq] = constraints(x)
    % 不等式约束条件
    c1 = (-2.636564*x(1)+63.0362159 )*(1-0.94376296)- x(2); % x1^2 - x2 >= 0
    c3=x(2);
     c2 =(1+x(1))*6.639983587-34; % -x1 - x2^2 + 2 <= 0
     c4=(1+x(1))*6.639983587;

    % 等式约束为空
    ceq = [];

    % 将不等式约束的值放入向量 c 中
    c = [c1; -c2;c3;c4]; % 注意不等式约束的方向取反
end
```

```
clc;clear;close all;
load('C2_predict.mat')

data_str='代码数据.xlsx' ;  %读取数据的路径


data1=readtable(data_str,'VariableNamingRule','preserve'); %读取数据
```

```
data2=data1(:,2:end);
data=table2array(data1(:,2:end));
data_biao=data2.Properties.VariableNames;  %数据特征的名称

 A_data1=data;
 data_biao1=data_biao;
 select_feature_num=G_out_data.select_feature_num1;   %特征选择的个数

data_select=A_data1;
feature_need_last=1:size(A_data1,2)-1;



%% 数据划分
  x_feature_label=data_select(:,1:end-1);     %x 特征
  y_feature_label=data_select(:,end);           %y 标签
 index_label1=1:(size(x_feature_label,1));
 index_label=G_out_data.spilt_label_data;  % 数据索引
 if isempty(index_label)
     index_label=index_label1;
 end
spilt_ri=G_out_data.spilt_rio;  %划分比例 训练集:验证集:测试集
train_num=round(spilt_ri(1)/(sum(spilt_ri))*size(x_feature_label,1));          %训练集个数
vaild_num=round((spilt_ri(1)+spilt_ri(2))/(sum(spilt_ri))*size(x_feature_label,1)); %验证集个数
 %训练集，验证集，测试集
 train_x_feature_label=x_feature_label(index_label(1:train_num),:);
 train_y_feature_label=y_feature_label(index_label(1:train_num),:);
 vaild_x_feature_label=x_feature_label(index_label(train_num+1:vaild_num),:);
vaild_y_feature_label=y_feature_label(index_label(train_num+1:vaild_num),:);
 test_x_feature_label=x_feature_label(index_label(vaild_num+1:end),:);
 test_y_feature_label=y_feature_label(index_label(vaild_num+1:end),:);
%Zscore 标准化
%训练集
 x_mu = mean(train_x_feature_label);  x_sig = std(train_x_feature_label);
 train_x_feature_label_norm = (train_x_feature_label - x_mu) ./ x_sig;    % 训练数据标准化
 y_mu = mean(train_y_feature_label);  y_sig = std(train_y_feature_label);
train_y_feature_label_norm = (train_y_feature_label - y_mu) ./ y_sig;    % 训练数据标准化
%验证集
 vaild_x_feature_label_norm = (vaild_x_feature_label - x_mu) ./ x_sig;    %验证数据标准化
 vaild_y_feature_label_norm=(vaild_y_feature_label - y_mu) ./ y_sig;  %验证数据标准化
%测试集
test_x_feature_label_norm = (test_x_feature_label - x_mu) ./ x_sig;    % 测试数据标准化
 test_y_feature_label_norm = (test_y_feature_label - y_mu) ./ y_sig;    % 训练数据标准化

```

```
%% 参数设置
num_pop=G_out_data.num_pop1;   %种群数量
num_iter=G_out_data.num_iter1;   %种群迭代数
method_mti=G_out_data.method_mti1;   %优化方法
BO_iter=G_out_data.BO_iter;   %贝叶斯迭代次数
min_batchsize=G_out_data.min_batchsize;   %batchsize
max_epoch=G_out_data.max_epoch1;   %maxepoch
hidden_size=G_out_data.hidden_size1;   %hidden_size


%% 算法处理块


 hidden_size=G_out_data.hidden_size1;    %神经网络隐藏层
disp('BiLSTM 回归')
t1=clock;
 max_epoch=G_out_data.max_epoch1;    %神经网络隐藏层
for i = 1: size(train_x_feature_label,1)      %修改输入变成元胞形式
    p_train1{i, 1} = (train_x_feature_label_norm(i,:))';
end
for i = 1 : size(test_x_feature_label,1)
     p_test1{i, 1}  = (test_x_feature_label_norm(i,:))';

 end

 for i = 1 : size(vaild_x_feature_label,1)
     p_vaild1{i, 1}  = (vaild_x_feature_label_norm(i,:))';
 end


 [Mdl,fitness] = optimize_fitrBiLSTM(p_train1,train_y_feature_label_norm,p_vaild1,vaild_y_feature_label_norm,num_pop,num_iter,
method_mti,max_epoch,min_batchsize);

y_train_predict_norm = predict(Mdl, p_train1,'MiniBatchSize',min_batchsize);

 y_vaild_predict_norm = predict(Mdl, p_vaild1,'MiniBatchSize',min_batchsize);
 y_test_predict_norm =  predict(Mdl, p_test1,'MiniBatchSize',min_batchsize);
 t2=clock;
 Time=t2(3)*3600*24+t2(4)*3600+t2(5)*60+t2(6)-(t1(3)*3600*24+t1(4)*3600+t1(5)*60+t1(6));


```

```
graph= layerGraph(Mdl.Layers); figure; plot(graph)
analyzeNetwork(Mdl)


 y_train_predict=y_train_predict_norm*y_sig+y_mu;  %反标准化操作
 y_vaild_predict=y_vaild_predict_norm*y_sig+y_mu;
 y_test_predict=y_test_predict_norm*y_sig+y_mu;
 train_y=train_y_feature_label; disp('**********************************************************************************************************************')
train_MAE=sum(abs(y_train_predict-train_y))/length(train_y) ; disp(['训练集平均绝对误差 MAE： ',num2str(train_MAE)])
train_MAPE=sum(abs((y_train_predict-train_y)./train_y))/length(train_y);  disp(['训练集平均相对误差 MAPE：',num2str(train_MAPE)])
train_MSE=(sum(((y_train_predict-train_y)).^2)/length(train_y)); disp(['训练集均方根误差 MSE： ',num2str(train_MSE)])
 train_RMSE=sqrt(sum(((y_train_predict-train_y)).^2)/length(train_y)); disp(['训练集均方根误差 RMSE： ',num2str(train_RMSE)])
train_R2= 1 - (norm(train_y - y_train_predict)^2 / norm(train_y - mean(train_y))^2);    disp(['训练集均方根误差 R2：',num2str(train_R2)])
vaild_y=vaild_y_feature_label;disp('**********************************************************************************************************************')
vaild_MAE=sum(abs(y_vaild_predict-vaild_y))/length(vaild_y) ; disp(['验证集平均绝对误差 MAE： ',num2str(vaild_MAE)])
vaild_MAPE=sum(abs((y_vaild_predict-vaild_y)./vaild_y))/length(vaild_y);  disp(['验证集平均相对误差 MAPE：',num2str(vaild_MAPE)])
vaild_MSE=(sum(((y_vaild_predict-vaild_y)).^2)/length(vaild_y)); disp(['验证集均方根误差 MSE： ',num2str(vaild_MSE)])
 vaild_RMSE=sqrt(sum(((y_vaild_predict-vaild_y)).^2)/length(vaild_y)); disp(['验证集均方根误差 RMSE： ',num2str(vaild_RMSE)])
vaild_R2= 1 - (norm(vaild_y - y_vaild_predict)^2 / norm(vaild_y - mean(vaild_y))^2);      disp(['验证集均方根误差 R2:  ',num2str(vaild_R2)])
 test_y=test_y_feature_label;disp('**********************************************************************************************************************');
test_MAE=sum(abs(y_test_predict-test_y))/length(test_y) ; disp(['测试集平均绝对误差 MAE： ',num2str(test_MAE)])
test_MAPE=sum(abs((y_test_predict-test_y)./test_y))/length(test_y); disp(['测试集平均相对误差 MAPE： ',num2str(test_MAPE)])
 test_MSE=(sum(((y_test_predict-test_y)).^2)/length(test_y)); disp(['测试集均方根误差 MSE： ',num2str(test_MSE)])
 test_RMSE=sqrt(sum(((y_test_predict-test_y)).^2)/length(test_y)); disp(['测试集均方根误差 RMSE： ',num2str(test_RMSE)])
  test_R2= 1 - (norm(test_y - y_test_predict)^2 / norm(test_y - mean(test_y))^2);    disp(['测试集均方根误差 R2：',num2str(test_R2)])
disp(['算法运行时间 Time: ',num2str(Time)])
%% 绘图块
color_list=G_out_data.color_list;   %颜色数据库
rand_list1=G_out_data.rand_list1;   %颜色数据库
Line_Width=G_out_data.Line_Width;   %线粗细
makesize=G_out_data.makesize;   %标记大小
yang_str2=G_out_data.yang_str2;   %符号库
yang_str3=G_out_data.yang_str3;   %符号库
kuang_width=G_out_data.kuang_width;   %画图展示数据
show_num=G_out_data.show_num;   %测试集画图展示数据
show_num1=G_out_data.show_num1;   %验证集画图展示数据
```

```
show_num2=G_out_data.show_num2;   %训练集画图展示数据

FontSize=G_out_data.FontSize;   %字体设置
xlabel1=G_out_data.xlabel1;   %
ylabel1=G_out_data.ylabel1;   %
title1=G_out_data.title1;   %
legend1=G_out_data.legend1;   %图例
box1=G_out_data.box1;   %框
le_kuang=G_out_data.le_kuang;   %图例框
grid1=G_out_data.grid1;   %网格
figure
XX=1:length(train_y_feature_label);
 index_show=1:show_num2;
plot(gca,XX(index_show),train_y_feature_label(index_show),yang_str2{1,3},'Color',color_list(rand_list1(1),:),'LineWidth',Line_Width(1))
hold (gca,'on')
 plot(gca, XX(index_show),y_train_predict(index_show),yang_str3{1,1},'Color',color_list(rand_list1(2),:),'LineWidth',Line_Width(2),'MarkerSize',makesize)
hold (gca,'on')
 set(gca,'FontSize',FontSize,'LineWidth',kuang_width)
 xlabel(gca,xlabel1)
 ylabel(gca,ylabel1)
 title(gca,'训练集结果')
 legend(gca,legend1)
  box(gca,box1)
 legend(gca,le_kuang) %图例框消失
 grid(gca,grid1)


figure
XX=1:length(vaild_y_feature_label);
 index_show=1:show_num1;
plot(gca,XX(index_show),vaild_y_feature_label(index_show),yang_str2{1,3},'Color',color_list(rand_list1(1),:),'LineWidth',Line_Width(1))
hold (gca,'on')
 plot(gca, XX(index_show),y_vaild_predict(index_show),yang_str3{1,1},'Color',color_list(rand_list1(2),:),'LineWidth',Line_Width(2),'MarkerSize',makesize)
hold (gca,'on')
 set(gca,'FontSize',FontSize,'LineWidth',kuang_width)
 xlabel(gca,xlabel1)
 ylabel(gca,ylabel1)
 title(gca,'验证集结果')
 legend(gca,legend1)
  box(gca,box1)
```

```
 legend(gca,le_kuang) %图例框消失
 grid(gca,grid1)


figure
XX=1:length(test_y_feature_label);
 index_show=1:show_num;
plot(gca,XX(index_show),test_y_feature_label(index_show),yang_str2{1,3},'Color',color_list(rand_list1(1),:),'LineWidth',Line_Width(1))
hold (gca,'on')
 plot(gca, XX(index_show),y_test_predict(index_show),yang_str3{1,1},'Color',color_list(rand_list1(2),:),'LineWidth',Line_Width(2),'MarkerSize',makesize)
hold (gca,'on')
 set(gca,'FontSize',FontSize,'LineWidth',kuang_width)
 xlabel(gca,xlabel1)
 ylabel(gca,ylabel1)
 title(gca,'测试集结果')
 legend(gca,legend1)
  box(gca,box1)
 legend(gca,le_kuang) %图例框消失
 grid(gca,grid1)
```

```
% 初始化种群
for i = 1:33
    c=xlsread('第三问预测数据','c2:c34');
    f=[19.74514286
17.23542857
10.80185714
8.857142857
7.142857143
38.29585714
10.145
19.05328571
7
6.337285714
29.14285714
19.57585714
25.01857143
11.28571429
9.865714286
21.10271429
16.57571429
14.508
```

```
15.89185714
19.85714286
5.252571429
14.28571429
55.24685714
63.26014286
4.667142857
19.25428571
26.02128571
25.14285714
9.641714286
12.28571429
3.780571429
50.73128571
]



f = @(x1, x2) ((1+x1)*c(i)*f-x2*c(i)); % 新的目标函数，最大化 x1^2 + x2^2

figure(1);
[X1, X2] = meshgrid(linspace(2.5, 30, 100), linspace(0, 2, 100));
Z = (1+x1)*c(i)*f-x2*c(i);
contour(X1, X2, Z, 50);

N = 50;                    % 初始种群个数
d = 2;                     % 空间维数
ger = 100;                 % 最大迭代次数
limit = [-5, 5; -5, 5];    % 设置位置参数限制
vlimit = [-1, 1; -1, 1];   % 设置速度限制
w = 0.8;                   % 惯性权重
c1 = 0.5;                  % 自我学习因子
c2 = 0.5;                  % 群体学习因子

% 随机初始化种群的位置
x = zeros(N, d);
for i = 1:d
    x(:, i) = limit(i, 1) + (limit(i, 2) - limit(i, 1)) * rand(N, 1);
end

% 随机初始化种群的速度
v = rand(N, d);
for i = 1:d
    v(:, i) = vlimit(i, 1) + (vlimit(i, 2) - vlimit(i, 1)) * rand(N, 1);
```

```
end

xm = x;                % 每个个体的历史最佳位置
ym = zeros(1, d);      % 种群的历史最佳位置
fxm = zeros(N, 1);     % 每个个体的历史最佳适应度
fym = -inf;            % 种群历史最佳适应度

% 绘制初始状态图
hold on;
plot(xm(:, 1), xm(:, 2), 'ro');
title('初始状态图');
xlabel('x1');
ylabel('x2');
grid on;

figure(2);
% 群体更新
iter = 1;
record = zeros(ger, 1);  % 记录器
while iter <= ger
    fx = -(x(:, 1).^2 + x(:, 2).^2);  % 计算个体当前适应度，负号是因为我们要最大化目标函数

    % 更新个体历史最佳适应度和位置
    for i = 1:N
        if fxm(i) < fx(i)
            fxm(i) = fx(i);
            xm(i, :) = x(i, :);
        end
    end

    % 更新群体历史最佳适应度和位置
    if fym < max(fxm)
        [fym, nmax] = max(fxm);
        ym = xm(nmax, :);
    end

    % 速度更新
    v = w * v + c1 * rand * (xm - x) + c2 * rand * (repmat(ym, N, 1) - x);

    % 边界速度处理
    for i = 1:d
        v(v(:, i) > vlimit(i, 2), i) = vlimit(i, 2);
        v(v(:, i) < vlimit(i, 1), i) = vlimit(i, 1);
    end
```

```
110.
111.     % 位置更新
112.     x = x + v;
113.
114.     % 边界位置处理
115.     for i = 1:d
116.         x(x(:, i) > limit(i, 2), i) = limit(i, 2);
117.         x(x(:, i) < limit(i, 1), i) = limit(i, 1);
118.     end
119.
120.     % 记录最大值
121.     record(iter) = -fym;  % 注意取负号还原最大值
122.
123.     % 绘制状态位置变化图
124.     clf;
125.     contour(X1, X2, Z, 50);
126.     hold on;
127.     plot(xm(:, 1), xm(:, 2), 'ro');
128.     title('状态位置变化');
129.     xlabel('x1');
130.     ylabel('x2');
131.     grid on;
132.     pause(0.1);
133.
134.     iter = iter + 1;
135. end
136.
137. % 绘制收敛过程图
138. figure(3);
139. plot(record);
140. title('收敛过程');
141.
142. % 绘制最终状态位置图
143. figure(4);
144. contour(X1, X2, Z, 50);
145. hold on;
146. plot(xm(:, 1), xm(:, 2), 'ro');
147. title('最终状态位置');
148. xlabel('x1');
149. ylabel('x2');
150. grid on;
151.
152. disp(['最大值：', num2str(-fym)]);  % 注意取负号还原最大值
153. disp(['变量取值：x1 = ', num2str(ym(1)), ', x2 = ', num2str(ym(2))]);
```

```
end
```