本科生论坛交流分享

我的竞赛经历与体会

胡兴发

数理科学学院

2022 年 9 月 19 日

主要内容

- 检验自己平时的学习成果
- 培养兴趣，开阔眼界，提升高度
- 对考研复试的帮助

反例构造

若正项级数 $\sum_{n=0}^{\infty} u_n$ 收敛，则 $u_n \to 0$，而 $\int_a^{+\infty} f(x)dx$ 收敛一般不意味着 $f(x) \to 0(x \to +\infty)$.

例如 $\int_0^{+\infty} \sin x^2 dx = \int_0^{+\infty} \frac{\sin t}{2\sqrt{t}} dt(x = \sqrt{t})$ 收敛，但 $\sin x^2 \nrightarrow 0$（当 $x \to +\infty$ 时）.

问题简化

设 $\lim\limits_{n\to+\infty} a_n = a$, $\lim\limits_{n\to+\infty} b_n = b$. 证明:

$$\lim_{n\to+\infty} \frac{a_1 b_n + a_2 b_{n-1} + \cdots + a_n b_1}{n} = ab$$

证: 能否“不妨设” $b = 0$?

记 $\beta_n = b_n - b$, 则 $\lim\limits_{n\to+\infty} \beta_n = 0$,

$$\lim_{n\to+\infty} \frac{a_1 b_n + a_2 b_{n-1} + \cdots + a_n b_1}{n}$$

$$= \lim_{n\to+\infty} \frac{a_1 \beta_n + a_2 \beta_{n-1} + \cdots + a_n \beta_1}{v_n} + b \lim_{n\to+\infty} \frac{a_1 + a_2 + \cdots + a_n}{n}$$

主要内容

2022 年高教社杯全国大学生数学建模竞赛题目

（请先阅读“全国大学生数学建模竞赛论文格式规范”）

C 题　古代玻璃制品的成分分析与鉴别

丝绸之路是古代中西方文化交流的通道，其中玻璃是早期贸易往来的宝贵物证。早期的玻璃在西亚和埃及地区常被制作成珠形饰品传入我国，我国古代玻璃吸收其技术后在本土就地取材制作，因此与外来的玻璃制品外观相似，但化学成分却不相同。

玻璃的主要原料是石英砂，主要化学成分是二氧化硅（SiO_2）。由于纯石英砂的熔点较高，为了降低熔化温度，在炼制时需要添加助熔剂。古代常用的助熔剂有草木灰、天然泡碱、硝石和铅矿石等，并添加石灰石作为稳定剂，石灰石煅烧以后转化为氧化钙（CaO）。添加的助熔剂不同，其主要化学成分也不同。例如，铅钡玻璃在烧制过程中加入铅矿石作为助熔剂，其氧化铅（PbO）、氧化钡（BaO）的含量较高，通常被认为是我国自己发明的玻璃品种，楚文化的玻璃就是以铅钡玻璃为主。钾玻璃是以含钾量高的物质如草木灰作为助熔剂烧制而成的，主要流行于我国岭南以及东南亚和印度等区域。

古代玻璃极易受埋藏环境的影响而风化。在风化过程中，内部元素与环境元素进行大量交换，导致其成分比例发生变化，从而影响对其类别的正确判断。如图 1 的文物标记为表面无风化，表面能明显看出文物的颜色、纹饰，但不排除局部有较浅的风化；图 2 的文物标记为表面风化，表面大面积灰黄色区域为风化层，是明显风化区域，紫色部分是一般风化表面。在部分风化的文物中，其表面也有未风化的区域。

图 1　未风化的蜻蜓眼玻璃珠样品

图 2　风化的玻璃棋子样品

现有一批我国古代玻璃制品的相关数据，考古工作者依据这些文物样品的化学成分和其他检测手段已将其分为高钾玻璃和铅钡玻璃两种类型。附件表单 1 给出了这些文物的分类信息，附件表单 2 给出了相应的主要成分所占比例（空白处表示未检测到该成分）。这些数据的特点是成分性，即各成分比例的累加和应为 100%，但因检测手段等原因可能导致其成分比例的累加和非 100%的情况。本题中将成分比例累加和介于 85%~105%之间的数据视为有效数据。

请你们团队依据附件中的相关数据进行分析建模，解决以下问题：

问题 1　对这些玻璃文物的表面风化与其玻璃类型、纹饰和颜色的关系进行分析；结合玻璃的类型，分析文物样品表面有无风化化学成分含量的统计规律，并根据风化点检测数据，预测其风化前的化学成分含量。

问题 2　依据附件数据分析高钾玻璃、铅钡玻璃的分类规律；对于每个类别选择合适的化学成分对其进行亚类划分，给出具体的划分方法及划分结果，并对分类结果的合理性和敏感性进行分析。

问题 3　对附件表单 3 中未知类别玻璃文物的化学成分进行分析，鉴别其所属类型，并对

分类结果的敏感性进行分析。

问题 4 针对不同类别的玻璃文物样品，分析其化学成分之间的关联关系，并比较不同类别之间的化学成分关联关系的差异性。

附件

表单 1 玻璃文物的基本信息

表单 2 已分类玻璃文物的化学成分比例，其中

(1) 文物采样点为该编号文物表面某部位的随机采样，其风化属性与附件表单 1 中相应文物一致。

(2) 部位 1 和部位 2 是文物造型上不同的两个部位，其成分与含量可能存在差异。

(3) 未风化点是风化文物表面未风化区域内的点。

(4) 严重风化点取自风化层。

表单 3 未分类玻璃文物的化学成分比例

基于回归的古代玻璃分类与预测模型

摘　要

古代玻璃在长时间的埋藏过程中，极易受周围环境影响发生风化，变暗、透明度降低或是在其表面形成一层风化产物结壳．风化严重的玻璃表面已经完全被风化物所覆盖，其原貌几乎无法辨认，以至于一部分考古工作难以进行．

为了更准确的判断古代玻璃的种类，本文使用了 Logistic 回归、灰色关联分析、Fisher 判别法的分类算法对古代玻璃进行分类，使得分类过程更简单，分类效率更高．

针对问题一，利用卡方检验能判断附件中所给玻璃文物的类别、颜色、纹饰对风化的显著性影响．发现表面风化与纹饰、颜色不存在显著性差异，而与文物类型存在显著性差异．本文利用 SPSS 软件对高钾、铅钡玻璃的化学成分进行统计，发现风化后高钾玻璃的化学成分含量明显减少，而铅钡玻璃的化学成分含量明显上升．我们分别找“无风化”、“相同纹饰”、“相同类型”的数据取其均值来对风化前玻璃文物的化学成分进行预测．

针对问题二，在问题一对数据进行描述性统计后，针对不同化学成分利用 Python 进行系统聚类，得到了聚类谱系图以及聚类热力图．分析图像可知，得到了四种聚类簇群．

针对问题三，我们基于问题一、问题二的分类基础，利用 Logstic 回归对附件三中未知类别的玻璃文物进行预测，又利用 Fisher 判别法再次预测并对比两次结果，发现预测的正确率为 100%．紧接着我们分别对数据集施加 5%、10%、15% 波动比例的干扰，再利用干扰后的数据预测原数据集中的玻璃种类．

针对问题四，本文建立灰色关联模型定量分析各化学成分与高钾、铅钡玻璃的关联度，我们发现，高钾玻璃与二氧化硅的灰色关联度最大，其值为 76.64，铅钡玻璃与二氧化硅、氧化铅的灰色关联度最大，其值分别为 23.87 和 23.50．最后我们对高钾、铅钡玻璃计算出的灰色关联系数进行方差分析，通过 F 统计量检验得出高钾与铅钡玻璃之间的差异性．

关键词：Logistic 回归；系统聚类法；灰色关联分析；Fisher 判别法

一、问题重述

1.1 问题背景

古代玻璃是丝绸之路上中外经济、技术和文化交流的重要物资[1]．通过丝绸之路传入中国的玻璃器皿，无论在器具形态、生产工艺、化学成分、分布面积等方面，都具有明显的时代特征．近几十年来，综合考古学、材料学和历史学等多学科交叉研究[2]，形成了关于中国古代不同时期玻璃器的器形风格、地域分布、成分体系、着色特征、制作工艺、产地来源等研究结果[3][4]，对不同来源的古代玻璃沿丝绸之路的分布和传播有了较清晰的认识．

古玻璃在长期埋藏过程中，很容易受到周围环境的影响，风化、变暗、透明度降低、光晕颜色，或在其外部形成风化产物结壳[5]．严重风化的玻璃表面已被风化物完全覆盖，其原始外观几乎无法辨认，因此考古发掘报告中描述的一些材料或玉器实际上是玻璃器皿[6]．

1.2 问题提出

根据以上背景，以及给出的三个表单数据，解决以下问题：

(1) 研究表单一中所给玻璃文物的表面风化与玻璃类型、纹饰、颜色的关联性；分析风化化学成分含量的统计规律；根据风化点检测数据预测风化前化学成分含量．

(2) 分析表单一中的数据，得到高钾玻璃和铅钡玻璃的分划准则，为子类别选择合适的化学成分，给出划分方法和结果，并分析分类结果的合理性和敏感性．

(3) 对附件表单三中未知类别的玻璃文物的化学成分进行分析，鉴别其所属类型，并对分类结果的敏感性进行分析．

(4) 针对不同类别的玻璃文物样品，分析其化学成分之间的关联关系，并且比较不同类别之间的化学成分关联关系的差异性．

其中表单一给出了玻璃文物的基本信息，表单二给出了已分类玻璃文物的化学成分比例，表单三给出了未分类玻璃文物的化学成分比例．

二、问题分析

2.1 问题一的分析

问题一首先要求我们根据表单一中所给数据分析玻璃文物的表面风化与玻璃类型、纹饰、颜色的关联性．由于玻璃中高含量的铅与周围环境中的 CO_2、水蒸气等反应生成 $PbCO_3$ 导致玻璃风化，因此，要想准确判断出古代玻璃是否被风化，我们选取玻璃类型、纹饰、颜色作为指标，利用卡方检验分别分析表面风化与每一个指标的差异性，分析显著性 P 是否小于 0.05．其次，题目要求我们分别讨论高钾玻璃和铅钡

玻璃的化学成分的变化规律，我们利用 SPSS 分别对高钾玻璃和铅钡玻璃进行描述性统计，找出其统计规律.

2.2 问题二的分析

问题二首先要求我们根据附件数据分析高钾玻璃、铅钡玻璃的分类规律．我们先利用 SPSS 软件对数据进行筛选，根据化学成分含量分出高钾类和铅钡类的玻璃．其次，题目要求我们针对高钾玻璃和铅钡玻璃分别选择化学成分对其进行亚分类．本文使用系统聚类算法将多个化学变量进行层次聚类，提取出影响分类结果的化学成分，最后使用 Python 对数据进行扰动，将扰动后的数据代入模型之中，并将扰动前后的结果进行对比，分析其灵敏度.

2.3 问题三的分析

问题三首先要求我们对附件表单 3 中未知类别玻璃文物的化学成分进行分析，鉴别其所属类型．本文我们分别使用 logistic 回归模型和 Fisher 判别法对玻璃文物进行分类．logistic 回归利用 logistic 函数将因变量的取值范围控制在 0 和 1 之间，表示取值为 1 的概率．我们利用 SPSS 软件生成一组虚拟变量并对未知类型进行预测，最后通过 SPSS 对预测成功率进行分析．紧接着我们利用 Fisher 线性判别法，该方法通过给定训练集样例，将样例投影到一维直线上，使得同类型的投影点尽可能接近和密集，异类的投影点尽可能远离．最后，我们利用 SPSS 对分类的准确率进行分析，并对以上两种方法的结果进行讨论.

2.4 问题四的分析

问题四首先要求我们针对不同类别的玻璃文物样品，分析其化学成分之间的关系．本文使用灰色关联分析法，首先将铅钡玻璃和高钾玻璃进行分类，将其种类作为母序列，设定不同的化学成分作为子序列．将原始序列标准化后，我们分别求解出各个比较序列的灰色关联度，从而得出不同类别之间化学成分的关联关系和差异性.

三、基本假设

(1) 假设样本数据之间的差距在可控范围之内；
(2) 假设进行统计时，每个抽取样本之间是相互独立的；
(3) 假设题目所给出的数据真实可靠.

四、符号说明

序号	符号	符号说明
1	P	显著性水平
2	$odds$	优势比
3	$\hat{y}_i$	逻辑回归预测值
4	ω	Fisher 判别法超平面法向量
5	Υ	灰色关联度
6	ρ	分辨系数 (通常取 0.5)
7	a, b	两极最小差，两极最大差
8	X_0	母序列
9	X_i	子序列 ($i \neq 0$)

五、模型一的建立与求解

5.1 数据预处理

根据题目背景中给出的条件，累积的成分比例和在 85% 至 105% 之间的数据被视为有效数据．首先我们使用 Python 对数据进行清洗，发现 15 号和 17 号的总成分累加和小于 85%，因此在下面的计算过程中去除 15 号和 17 号这两组数据．

5.2 问题的分析

问题一首先要求我们针对玻璃类型、纹饰、颜色这三个指标分别求其与表面风化的相关性．分析表单 1 中的数据可以得知，文物编号、纹饰、类型、颜色、表面风化四个变量都是固定类型的变量，因此建立卡方检验模型来对这些定量变量之间的相关性进行分析．其次，题目要求结合玻璃类型，分析文物样品表面有无风化的化学成分含量的统计规律，我们筛选出高钾玻璃和铅钡玻璃两个大类，最后分析其余变量的变化规律．

5.3 模型的建立

5.3.1 卡方检验模型

皮尔逊卡方检验是最著名的卡方检验之一，主要比较分类变量和分类变量之间的差异．它可以用来比较两种情况下的变量：适应度测试和独立性测试．

(1) 独立性检验步骤

计算卡方检验的统计值 χ^2：将对应的观察值和理论值（期望值）的差值平方，再除以理论值，最后相加：

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(O_{i,j}-E_{i,j})^2}{E_{i,j}} \tag{1}$$

(2) 计算 χ^2 统计值的自由度

检验统计量为 $x^2 = \sum\frac{(O-E)^2}{E}$，其中指 O 观察频数，E 指期望频数．若用 x_2 作为卡方分布中检验统计量，则表示为 $x^2 \sim \chi^2_\alpha(\nu)$，其中 ν 为自由度，α 为显著性水平，在拟合优度检验中，ν 等于组数减去限制数，针对两个变量的独立性检验中，对 h 行 k 列的列联表，有 $\nu = (h-1)\times(k-1)$．

(3) 卡方拟合优度检验

假设有一总体 X，服从卡方分布我们设 $H_0 : P(X = a_i) = p_i(i = 1,\ldots,k)$，其中，$a_i$，$p_i(i = 1,\ldots,k)$ 都为已知，且 $a_1,\ldots,a_k$ 两两不同，$p_i > 0(i = 1,\ldots,k)$，现在从总体中抽样次，得样本 $X_1,\ldots,X_n$，利用它们去检验 H_0 是否成立的过程称为拟合优度检验．

表 1 经验值统计表

类别	α_1	α_2	$\ldots$	α_i	$\ldots$	α_k
理论值	np_1	np_2	$\ldots$	np_i	$\ldots$	np_k
经验值	ν_1	ν_2	$\ldots$	ν_i	$\ldots$	ν_k

我们称 $Z = \sum(\text{理论值}-\text{经验值})^2/\text{理论值} = \sum_{i=1}^{k}(np_i-\nu_i)^2/(np_i)$ 为拟合优度 χ^2 统计量．

(4) 连续性修正

$$\chi^2_{Yates} = \sum_{i=1}^{R}\sum_{j=1}^{C}\frac{(|A_{ij}-T_{ij}|-0.5)^2}{T_{ij}} \sim \chi^2((R-1)(C-1)) \tag{2}$$

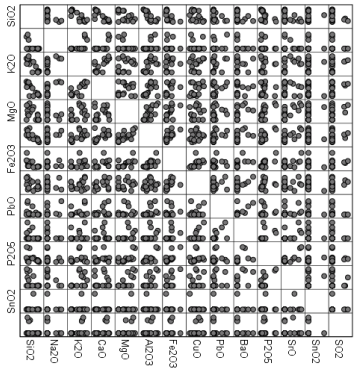
为了降低利用卡方分布统计方法的误差，有必要进行连续性修正．

5.3.2 基于卡方检验的相关性分析

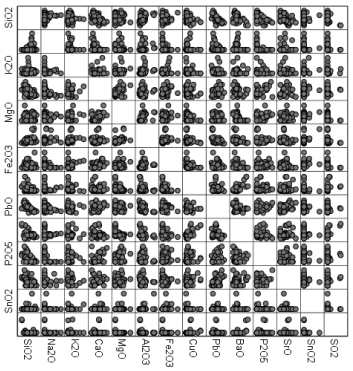
首先将表单 1 中的数据导入 Excel，并进行简单的描述性统计以及筛选处理．结果如附录表 13所示．

表 13为铅钡玻璃和高钾玻璃的描述性统计结果，分别对每一个指标进行统计，得到每个指标对应数据的范围、最大最小值、平均值、标准差、方差、偏度、峰度等统计量．为了检验每一指标对应的数据之间是否存在着线性关系，我们利用 SPSS 软件画出各指标数据之间的散点图．

如图 1所示，高钾、铅钡两玻璃之间各化学成分两两之间并没有明显的线性关系．为进一步确定对相关性进行分析，我们还需要进行卡方检验．我们利用 SPSS 软件进



(a) 高钾玻璃化学成分矩阵散点图



(b) 铅钡玻璃化学成分矩阵散点图

图 1 利用 SPSS 软件画出各指标数据之间的散点图

行卡方检验，卡方检验可以比较定类变量之间的差异性．我们认为卡方值越大则差异越大．检验结果如附录表 12所示．

由表 12可知，基于表面风化和纹饰，显著性值为 0.056[1]，水平上不呈现显著性，接受原假设，因此对于表面风化和纹饰数据不存在显著性差异．

基于表面风化和类型，显著性值为 0.020[2]，水平上不呈现显著性，接受原假设，因此对于表面风化和纹饰数据存在显著性差异．

基于表面风化和颜色，显著性值为 0.507[3]，水平上不呈现显著性，接受原假设，因此对于表面风化和颜色数据不存在显著性差异．

5.3.3 化学成分统计规律

分析表单 1 中数据可知，玻璃类型为高钾类与铅钡类[7]，因此我们针对这两类玻璃，对其余变量进行分析．统计结果如表 2所示．

表 2 化学成分分类表

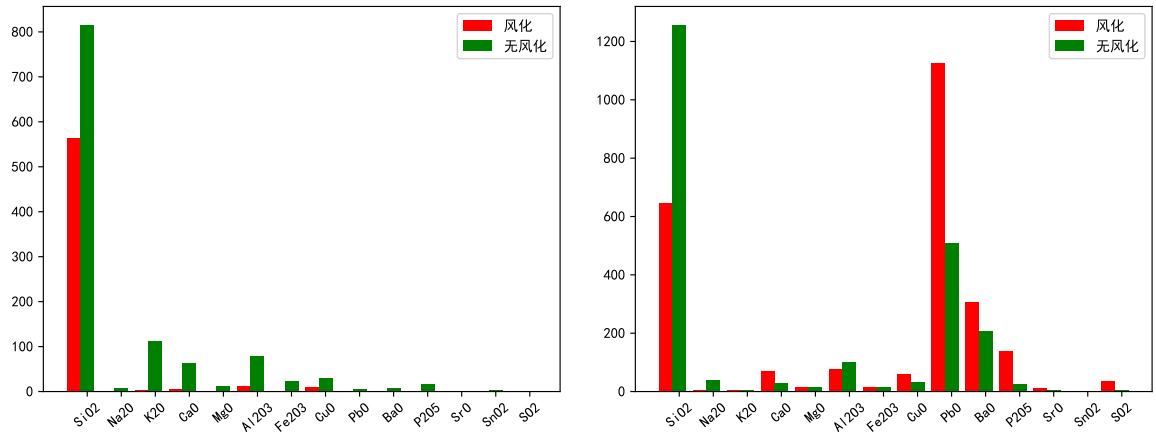
玻璃类型	是否风化	纹饰	颜色	总含量范围
高钾	风化	B	蓝绿	98.1% 100%
	无风化	A、C	蓝绿、绿	88.41% 100%
铅钡	风化	A、C	蓝绿、浅蓝、浅绿、深绿、紫、黑	90.17% 99.89%
	无风化	A、C	绿、浅绿、深绿深蓝、浅蓝、紫	88.41% 99.98%

由表 2可知，对于高钾类玻璃，B 类纹饰的都为风化后的玻璃，且其总含量较高，这很可能与其所处环境有关，无风化的玻璃其化学成分总含量少于风化玻璃．而对于

[1]1% 的显著性水平

[2]5% 的显著性水平

[3]10% 的显著性水平



(a) 高钾玻璃风化前后化学成分统计图　　(b) 铅钡玻璃风化前后化学成分统计图

图 2　高钾玻璃和铅钡玻璃风化前后的化学成分统计直方图

铅钡玻璃，其颜色种类明显多于高钾玻璃，风化玻璃的化学成分总含量稍稍高于无风化玻璃.

为了进一步分析高钾玻璃和铅钡这两类玻璃的化学成分含量的统计规律，我们分别针对高钾玻璃和铅钡玻璃风化前后的化学成分作出统计直方图，其结果如图 2所示.

分析图 2可知，对于高钾玻璃风化后其化学成分含量如二氧化硅、氯化镁等皆明显下降，由此可以推断在风化过程中，由于环境因素的改变使得高钾玻璃中化学成分含量降低. 而对于铅钡玻璃，风化后其化学物质除二氧化硅外含量明显上升，由此我们可以推断，铅钡玻璃在埋藏过程中，随着风化越来越严重，其与周围环境的物理化学反应使其化学成分含量明显上升，这一分析结论对考古工作有着参考价值[8].

5.3.4 对风化前的化学成分进行预测

古代玻璃常年埋藏在地下，对于其风化前后的化学成分含量的变化上文已经进行了分析. 由于不同类型玻璃之间的化学成分含量不同，在考古工作中部分化学成分可能没被检测到[9]，反映在数据集中即为“0”值. 本文分别提取“无风化”、“相同纹饰”、“相同类型”的数据然后取均值. 本文在预测过程中，对不满足“相同纹饰”这一条件的数据，只采用“无风化”和“相同类型”这两个条件来取值. 由于预测结果较大，不便展示在正文中，可在支撑材料-t1_forecast.xlsx 中查看.

六、问题二建模与分析

6.1 问题的分析

问题二首先要求我们分析高钾、铅钡玻璃的分类规律. 基于问题一的分析，针对高钾玻璃和铅钡玻璃不同的化学成分数值进行统计，对每一个化学成分含量求均值，

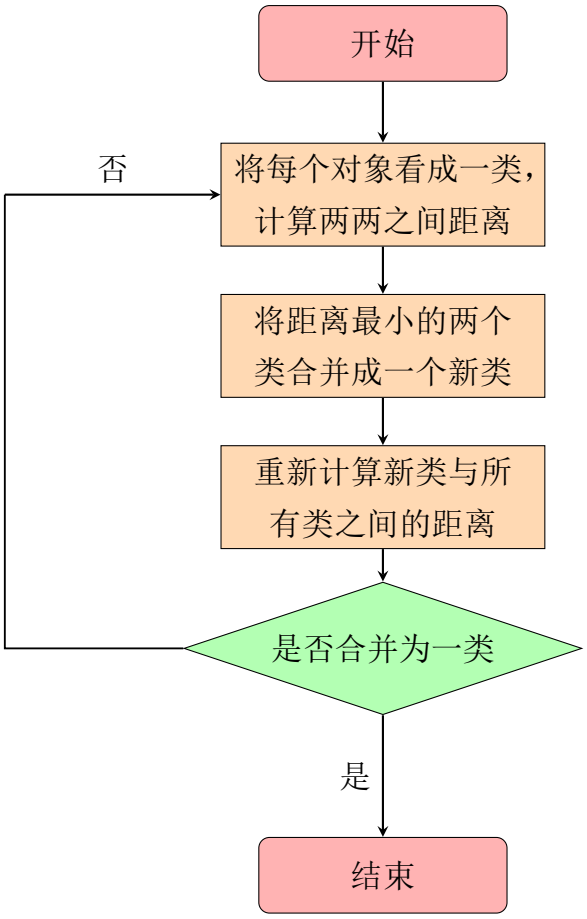


图 3　系统聚类算法流程图

再将两者相除得到权重向量．紧接着利用系统聚类算法对高钾、铅钡两种玻璃成分进行聚类．聚类模型是建立在无标记的数据上的一种无监督算法，我们使用的系统聚类模型聚类算法是基于距离的层次聚类算法，在最小的误差函数基础上对数据进行聚类，即认为两个对象距离越近则他们的相似度就越大．接着利用 Pandas 和 Matplotlib 绘制不同种类玻璃的化学成分分群的概率密度统计图．题目还要求选取化学成分对玻璃进行亚类划分，我们根据得到的聚类结果进行分析．最后对数据集以 5%、10%、15% 的比例进行扰动，再将扰动后的数据代入我们的模型之中进行进行检验，并与之前聚类结果进行对比，对分类模型的敏感性与合理性进行分析．聚类结果见图 4（取 k=4）.

6.2 高钾、铅钡玻璃的分类规律

基于第一问数据的统计性描述，我们得到了高钾玻璃和铅钡玻璃的化学成分含量分布直方图．考虑到玻璃的主要成分为二氧化硅，以及高钾玻璃、铅钡玻璃的化学成分有部分相似，我们利用 SPSS 软件对高钾玻璃、铅钡玻璃的每一化学成分进行求均值，我们将结果存储到表 3中.

下面，我们使用 Matlab 软件将高钾玻璃各成分的平均值除以铅钡玻璃各成分平均

表 3 各化学成分均值表

	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2
高钾玻璃	76.64	0.46	6.40	3.85	0.79	5.06	1.38	2.16	0.27	0.40	1.03	0.03	0.13	0.07
铅钡玻璃	38.88	0.90	0.17	2.05	0.65	3.67	0.66	1.88	33.35	10.49	3.29	0.35	0.06	0.80

值，并对结果进行归一化，以获得权重矩阵．结果如表 4所示．

表 4 权重矩阵

化学成分	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2
权重	0.04	0.01	0.74	0.04	0.02	0.03	0.04	0.02	0.00	0.00	0.01	0.00	0.04	0.00

由表 4可知，氧化钾的权重最大，所以认为氧化钾的含量可以作为区分高钾玻璃和铅钡玻璃的一个标准[8]．

6.3 系统聚类模型

样品与样品之间的常用距离见下表：

表 5 系统聚类常用距离

各种类型的距离	详细说明
绝对值距离	$d(\vec{x}_i,\vec{x}_j)=\sum_{k=1}^{p}\lvert x_{ik}-x_{jk}\rvert$
欧式距离	$d(\vec{x}_i,\vec{x}_j)=\sqrt{\sum_{k=1}^{p}(x_{ik}-x_{jk})^2}$
Minkowski 距离	$d(\vec{x}_i,\vec{x}_j)=\left[\sum_{k=1}^{p}(x_{ik}-x_{jk})^q\right]^{\frac{1}{q}}$
Chebyshev 距离	$d(\vec{x}_i,\vec{x}_j)=\ _{1\leqslant k<p}\lvert x_{ik}-x_{jk}\rvert$
马氏距离	$d(\vec{x}_i,\vec{x}_j)=(\vec{x}_i-\vec{x}_j)'\sum^{-1}(\vec{x}_i-\vec{x}_j)$

1 其中 $\vec{x}_i=(x_{i1},\ldots,x_{ip})',\vec{x}_j=(x_{j1},\cdots,x_{jp})',\sum$ 为样本的协方差矩阵．

6.4 高钾、铅钡玻璃的分类规律

系统聚类的合并算法是通过计算两类数据点间的距离，对最为接近的两类数据点进行组合，并反复迭代这一过程，直到将所有数据点合成一类，并生成聚类谱系图．系统聚类的算法流程如图 3所示．

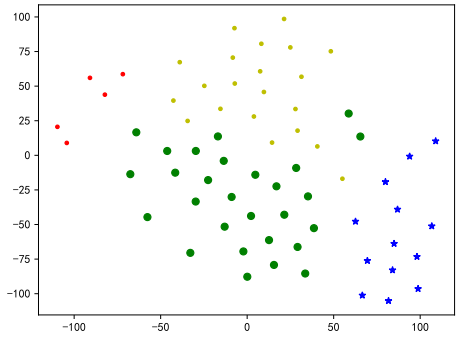


图 4 kmeans 聚类图

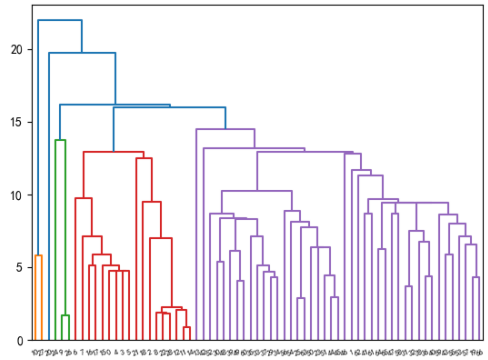


图 5 各化学成分聚类树状图

根据上述计算过程，我们利用 Python 进行求解，得到系统聚类中不同玻璃类型树状图如图 5所示.

由图 5可知，我们根据化学成分的不同将两种玻璃聚成了 4 类，在图 5利用 4 种不同的颜色进行了说明.

为了进一步体现出该模型的聚类效果，针对不同的化学成分，利用 Python 绘制了聚类效果热力图，如附录图 14所示.

6.5 亚类分类模型的灵敏度分析

表 6 亚类分类模型的灵敏度分析

	风化	无风化	无噪声		噪声 10%		噪声 20%		噪声 50%	
高钾玻璃	26	23	28	21	28	20	32	19	31	18
铅钡玻璃	6	12	5	13	5	14	5	11	7	11

如表 6所示，四个亚类对应高钾和铅钡玻璃的风化和无风化．添加噪声后再次聚类，可见本模型的灵敏度较高.

七、问题三建模与分析

7.1 问题的分析

问题三要求我们分析表单 3 中的数据，分析附件中未知玻璃文物的化学成分，并确定其类型．在问题一、二数据处理的基础上，我们对玻璃类型进行分类，分为了高钾玻璃和铅钡玻璃两类．针对这两类玻璃，我们将高钾玻璃赋值为 0，铅钡玻璃赋值为 1，这样处理后我们很容易联想到使用 logistic 回归分析法进行分析，本文使用 SPSS

软件对数据进行处理；同时，我们对数据集建立 Fisher 判别法模型进行了分析．Fisher 判别法尝试将样本投影到一维直线上，使得相似样本的投影点尽可能密集、靠近和疏远．紧接着题目要求我们对分类结果的敏感性进行分析，我们利用 Python 对数据集进行扰动，再将扰动后的数据代入 logistic 模型和 Fisher 模型[10]，对比扰动前后的结果，然后对分类结果的敏感性进行分析．

7.2 Logistic 模型的建立

7.2.1 Logistic 函数

Logistic 回归模型中的因变量仅有 0 和 1(代表是或否、发生或者不发生) 两种情况．假设在 p 个自变量 $x_1, x_2, \cdot, x_p$ 作用下，记 y 取 1 的概率为 $p = P(y = 1|X)$，取 0 的概率为 $1 - p$，取 1 和 0 两个概率之比为 $\frac{p}{1-p}$，称为事件的优势比 (odds)，对 odds 取自然对数得到 Logistic 变换 $Logit(p) = \ln(\frac{p}{1-p})$．

令 $Logit(p) = \ln(\frac{p}{1-p}) = z$, 则 $p = \frac{1}{1+e^{-z}}$ 即为 Logistic 函数，如图 6所示：

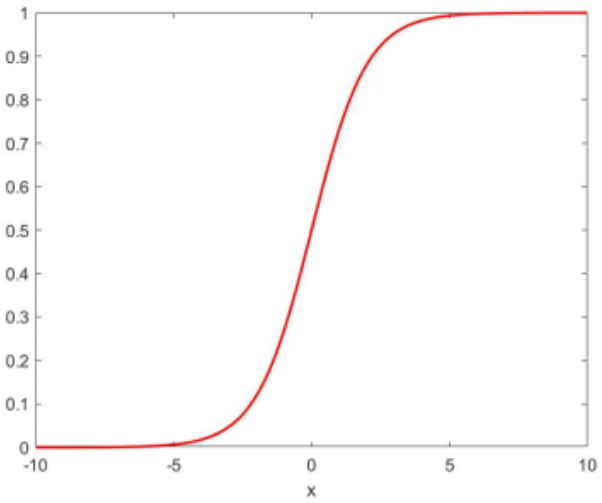


图 6 Logistic 函数

当 p 在 (0,1) 之间变化时，odds 的取值范围是 $(0, +\infty)$，则 $\ln(\frac{p}{1-p})$ 的取值范围是 $(-\infty, +\infty)$．

7.2.2 Logistic 回归模型

Logistic 回归模型是建立 $\ln(\frac{p}{1-p})$ 与自变量的线性回归模型．

Logistic 回归模型为：

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \varepsilon \tag{3}$$

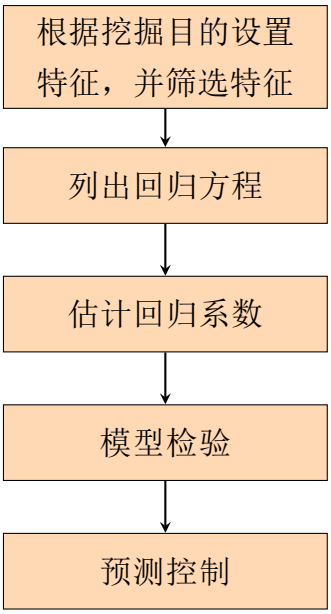


图 7 Logistic 回归建模流程图

分类表[a,b]

			预测		
			类型=铅钡		
	实测		.00	1.00	正确百分比
步骤 0	类型=铅钡	.00	0	18	.0
		1.00	0	49	100.0
	总体百分比				73.1

a. 常量包括在模型中。

b. 分界值为 .500

图 8 预测分类表

因为 $\ln(\frac{p}{1-p})$ 的取值范围是是 $(-\infty,+\infty)$，这样，自变量 $x_1,x_2,\ldots,x_p$ 可在任意范围内取值．记 $g(x)=\beta_0+\beta_1x_1+\ldots+\beta_px_p$，得到：

$$\frac{p}{1-p}=e^{\beta_0+\beta_1x_1+\ldots+\beta_px_p+\varepsilon} \tag{4}$$

β_0：在 $x_1,x_2,\ldots,x_p$ 全部取 0，发生概率之比的自然对数；

β_1：某自变量 x_i 变化时，即 $x_i=1$ 与 $x_i=0$ 相比，$y=1$ 优势比的对数值．

本文如果 $\hat{v}_l\geqslant 0.5$，我们认为其预测的是铅钡玻璃，反之我们预测的是高钾玻璃．

7.2.3 Logistic 回归建模

Logistic 回归模型的建模步骤如图所示：

我们将高钾玻璃赋值为 0，铅钡玻璃赋值为 1，首先利用 SPSS 软件生成一个新的虚拟变量序列，再调用 SPSS 逻辑回归方法，对表单 3 中的数据进行预测，预测成功率如下表所示：

由上表可知我们预测的成功率为 100%，实验过程中我们分别尝试增加迭代次数，从开始的 20 次逐渐增加到 50 次，发现迭代 26 次时已得到最优预测结果．由次可见，这个模型的预测效果较为理想，使得我们的分类有较强的说服力．

方程中的变量见下表：

根据问题三预测结果，表 3 中未知类型的玻璃种类分别为：高钾、铅钡、铅钡、铅钡、铅钡、高钾、高钾、铅钡．

表 7 方程变量表

B	标准误差	瓦尔德	自由度	显著性	Exp(B)
1.001	.276	13.202	1	.000	2.722

		预测组成员信息		
	类型=铅钡	.00	1.00	总计
原始 计数	.00	18	0	18
	1.00	0	49	49
	未分组个案	1	0	1
%	.00	100.0	.0	100.0
	1.00	.0	100.0	100.0
	未分组个案	100.0	.0	100.0

a. 正确地对 100.0% 个原始已分组个案进行了分类。

图 9 分类结果

二氧化硅(SiO2)	.043
氧化钠(Na2O)	.320
氧化钾(K2O)	-.188
氧化钙(CaO)	.023
氧化镁(MgO)	.204
氧化铝(Al2O3)	.209
氧化铁(Fe2O3)	.098
氧化铜(CuO)	-.182
氧化铅(PbO)	.118
氧化钡(BaO)	.216
五氧化二磷(P2O5)	.095
氧化锶(SrO)	-.608
氧化锡(SnO2)	.015
二氧化硫(SO2)	-.102
(常量)	-7.377

未标准化系数

图 10 典型判别函数系数表

7.3 Fisher 线性判别法

Fisher 判别法是通过给定训练集，将样例投影到一维的直线上，使得同类型数据集中，异类数据尽量分散．首先我们将所有的点分隔在超平面 $\omega^T x = 0$ 两侧，即每个点都投影到 ω 这个法向量上，保证类内小，类内大．

$$X = (x_1, x_2, \ldots, x_N)^T = \begin{pmatrix} x_1^T \\ x_2^T \\ \cdots \\ x_N^T \end{pmatrix}_{N\times P}, Y = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}_{n\times 1} \tag{5}$$

$$\{(x_i, y_i)\}_{i=1}^N, x_i \in R^P, y_i \in \{+1, -1\}, X_{cl} = \{x_i \mid y_i = +1\},$$

$$X_{c2} = \{x_i \mid y_i = -1\}, |X_{c1}| = N_1, |X_{c2}| = N_2, \text{且} N_1 + N_2 = N \tag{6}$$

基于上述数据集，本文利用 SPSS 软件中判别式方法对玻璃种类进行分类，我们求解得到的线性系数向量见图 10.

由图 10可得出 Fisher 线性判别法中的线性系数向量，问题初步获解.

下面我们对分类结果的准确性进行分析，其结果见图 9．通过图 9可知，我们利用 Fisher 判别法得到的预测结果准确率为 100%．这也增强了上文 Logistic 回归模型预测的说服力，从侧面对分类结果进行了验证.

7.4 灵敏性分析

我们分别对表单 2 中的数据按 5%、10%、15% 的比例进行扰动，并将扰动后的数据代入我们的 Logisic 模型中进行预测，假设题目所给的数据是可靠的，那么我们可以以此来评估模型的灵敏度．重复 3000 次预测并取均值，预测准确率分别为：96.02，84.93，72.67．由此可见本模型的灵敏度较高．

八、问题四模型的建立与求解

8.1 问题的分析

问题四首先要求我们针对不同类别的玻璃文物样品，分析其化学成分之间的关系，我们分别设高钾类玻璃和铅钡类玻璃为母序列，在问题二的基础上，我们将主成分作为子序列，建立灰色关联模型．通过计算母序列与子序列的灰色关联系数，并对其进行归一化得到各个指标的权重向量．问题四还要求我们比较不同类别之间的化学成分关联关系的差异性，我们将不同类别的玻璃中所计算出来的灰色关联度进行方差分析，利用 SPSS 软件进行显著性检验，以此来观察高钾玻璃和铅钡玻璃之间的差异性．

8.2 灰色关联模型的建立

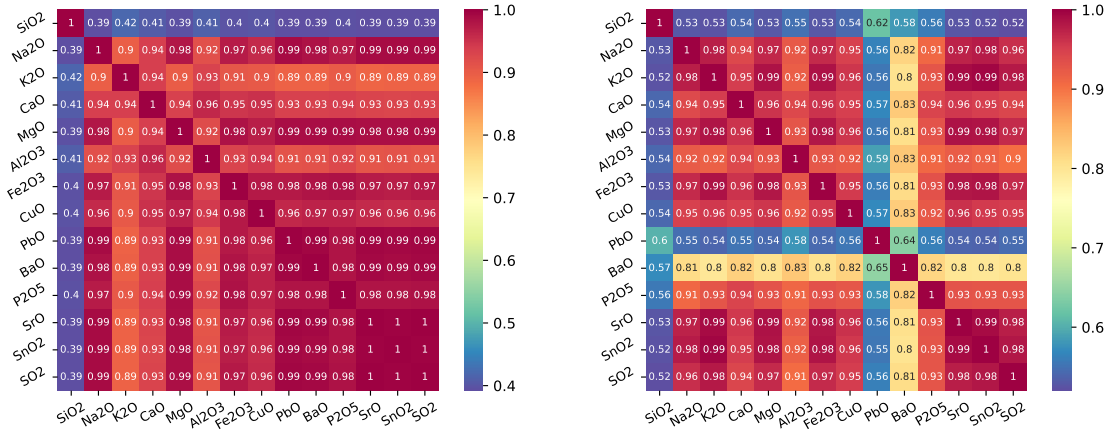
在一般抽象系统中往往包含多种因素，人们往往想知道在众多因素中的主要因素和次要因素；影响系统发展的因素，对系统的发展影响较小的因素；对系统的发展起到推动作用的因素，需要对哪一部分阻碍因素进行抑制．若想要得到答案，对系统进行分析是必要的．在传统的数理统计中有回归分析等方法进行量化与定性分析，但是它们要求有大量的数据，并且要求样本服从某种概率分布，有的还要求各个因素的数据之间彼此无关，上述统计方法有很大的局限性．本文构建的灰色关联模型判断序列曲线是否密切相关是根据序列曲线的几何相似性来确定的．

8.2.1 进行系统性分析

母序列：能反映系统行为特征的数据序列，类似于因变量 Y，此处记作 X_0，本文中 X_0 就是玻璃的种类．

子序列：是影响系统行为的因素组成的数据序列，类似于自变量 X，此处记作 $(X_1, X_2, \ldots, X_m)$，本文中 (X_1, X_2, X_3) 分别对应第一、第二和第三种主成分．

预处理：对母序列和子序列中的每个指标进行预处理．先求出每个指标的均值，再用该指标中的每个元素都除以其均值．



(a) 高钾玻璃化学成分热力图　　(b) 铅钡玻璃化学成分热力图

图 11　高钾玻璃和铅钡玻璃化学成分热力图

计算子序列中各个指标与母序列的关联系数：

$$\begin{cases} X_0 = (X_0(1), X_0(2), \ldots, X_0(n))^T \\ X_1 = (X_1(1), X_1(2), \ldots, X_1(n))^T \\ \ldots \\ X_m = (X_m(1), X_m(2), \ldots, X_m(n))^T \end{cases} \tag{7}$$

两极最小差

$$a = \min_i \min_k \mid X_0(k) - X_i(k) \mid \quad (i = 1, 2, \cdots, m)(k = 1, 2, \cdots, n) \tag{8}$$

两极最大差

$$b = \max_i \max_k \mid X_0(k) - X_i(k) \mid \quad (i = 1, 2, ..., m)(k = 1, 2, ..., n) \tag{9}$$

定义

$$\gamma(X_0(k), X_i(k)) = \frac{a + \rho b}{\mid X_0(k) - X_i(k) \mid + \rho b}, \quad \rho：\text{分辨系数, 一般取 } 0.5 \tag{10}$$

即

$$\gamma(X_0, X_i) = \frac{1}{n} \sum_{k=1}^{n} \gamma(X_0(k), X_i(k)) \tag{11}$$

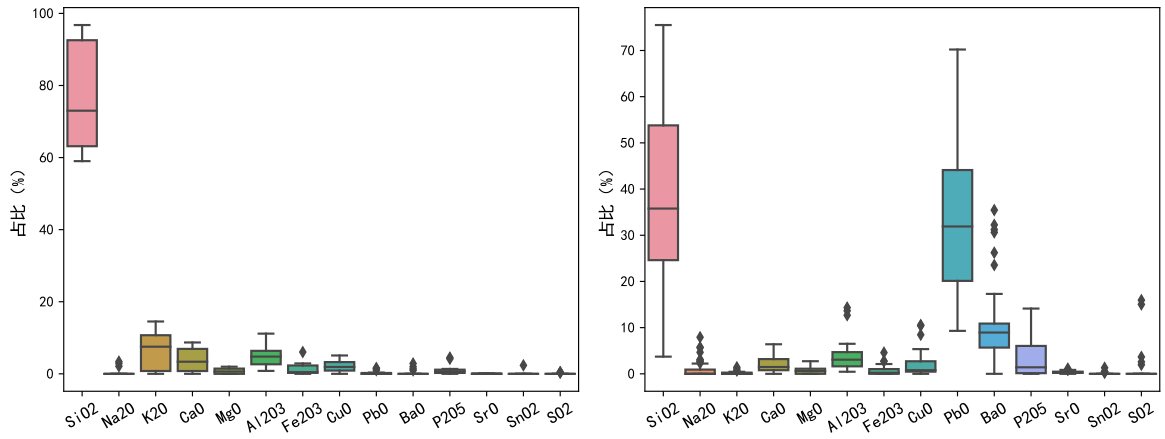
为 X_0 和 X_i 的灰色关联度.

8.2.2 对两种玻璃的灰色关联分析

基于问题一和问题二的数据分析，我们利用 Matlab 软件，对两类玻璃的化学成分数据进行标准化处理．紧接着我们对标准化处理后的矩阵计算母序列分别与子序列的绝对值差．最后，我们计算出母序列与子序列的灰色关联度．以上结果详见支撑材料．利用 Python 对两类玻璃的化学成分作出灰色关联热力图，见附录11(a)和11(b)．

表 8 灰色关联度

	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2
高钾玻璃	76.64	0.46	6.40	3.85	0.79	5.06	1.38	2.16	0.27	0.40	1.03	0.03	0.13	0.07
铅钡玻璃	23.87	0.68	0.08	1.33	0.46	2.41	0.29	0.74	23.50	5.26	2.21	0.23	0.04	0.07



(a) 高钾玻璃方差分析箱线图　　(b) 铅钡玻璃方差分析箱线图

图 12 高钾玻璃和铅钡玻璃方差分析箱线图

利用 Matlab 求出高钾类玻璃的各化学成分的灰色关联度，如表 8所示.

由表 8可知，二氧化硅的灰色关联度最大，说明高钾类玻璃与二氧化硅的关联性最强；二氧化硅和氧化铅与铅钡类玻璃的关联性较强，其他成分次之.

8.2.3 对两类玻璃灰色关联系数进行方差分析

为了确定各个变量对结果影响力的大小，我们利用 Python 对两种玻璃的化学成分含量进行 F 统计量检验. 针对不同化学成分含量画出箱线图，其结果如表 9和表 10所示.

表 9 高钾玻璃方差分析结果

	df	sum_sq	mean_sq	F	PR(>F)
C(Treat)	13	94843.76	7295.67	382.28	2.77E-151
Residual	238	4542.19	19.08		

分析表 9，可以发现各化学成分之间的差异更为显著，数值明显小于 0.05. 为了进一步比较，如下表所示为我们将方差分析后的结果. 可见，不同水平的因子（Treat）对因变量有显著影响. 紧接着，我们进一步进行多重分析，其结果见支撑材料.

表 10 铅钡玻璃方差分析结果

	df	sum_sq	mean_sq	F	PR(>F)
C(Treat)	13	102571.79	7890.14	160.49	5.34E-196
Residual	672	33036.49	49.16		

九、模型的推广与评价

9.1 问题一模型的评价

卡方检验模型的优点在于能够对定类变量 (如本文中的文物类别、纹饰、颜色) 之间进行差异性的分析，其作用在于能根据样本数据推断总体的分布与期望分布是否有显著性差异.

卡方检验模型的缺点在于卡方验证容易受到样本量的影响，样本量不同得到的结果也可能不同. 本文使用 SPSS 软件对卡方值进行了修正，很大程度上增强了检验结果的准确性.

9.2 问题二模型的评价

系统聚类模型是实际应用过程中应用最广泛的模型之一. 它可以对不同类型的样本和变量进行分类. 此外，它还有非常丰富的类间距离计算方法，如绝对值距离、欧几里得距离、闵可夫斯基距离、切比雪夫距离和马氏距离. 由于聚类的每一步都需要计算上述类间距离，当数据量大时，计算速度相对较慢，这也是该模型的不足之处. 当使用系统聚类模型应用于实际问题时，其结果可能不令人满意，主要原因是我们做的是一个数学上的处理，我们要对此找一个合理的解释.

9.3 问题三模型的评价

Logistic 回归模型的优点在于其非常适合二分类的问题，且简单易于推广使用. 例如器材分类的预测、对模型外观的分类等等问题我们都可以使用 Logistic 回归. 问题三中建立的 Fisher 判别模型的预测结果表明，Logistic 回归对未知类别的玻璃文物种类进行分类是十分有效的. Logistic 和 Fisher 模型的另外一个优点在于它们都适宜于处理多分类问题.

作为一种线性理模型，Logistic 它很难解决非线性的数据. 通常对于使用 Logistic 预测结果较差时，可在回归模型中加入平方项和交互项等，但是这种方法在提高预测能力的同时可能会出现过拟合的现象，如下图所示：

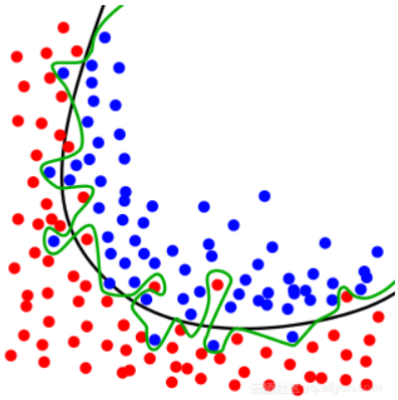


图 13 过拟合现象

9.4 问题四模型的评价

灰色关联分析模型的优点在于其弥补了采用传统数理统计方法作为系统分析所导致的缺憾．它也适用于样本的数量和样本的规律性，并且由于其计算量小，非常方便．定量结果和定性分析结果之间不会有差异．

灰色关联分析模型的缺点是，随着理论应用领域的不断扩大，其理论体系并不完善，应用范围有限．

参考文献

[1] 干福熹. 中国古代玻璃的起源和发展 [J]. 自然杂志, 2006, 28(4): 187–193.

[2] 王承遇, 陶瑛. 硅酸盐玻璃的风化 [J]. 硅酸盐学报, 2003, 31(1): 78–85.

[3] 任展展. 魏晋南北朝隋唐玻璃器研究 [J], 2020.

[4] SELIGMAN C, RITCHIE P, BECK H. Early Chinese glass from pre-Han to Tang times[J]. Nature, 1936, 138(3495): 721–721.

[5] 王婕, 李沫, 马清林, et al. 一件战国时期八棱柱状铅钡玻璃器的风化研究 [J]. 玻璃与搪瓷, 2014, 42(2): 6–13.

[6] 黄启善. 广西古代玻璃制品的发现及其研究 [J]. 考古, 1988(3): 264–276.

[7] SCOTT D A. Metallography and microstructure in ancient and historic metals[M]. [S.l.]: Getty publications, 1992.

[8] 李晓岑. 关于中国铅钡玻璃的发源地问题 [J]. 自然科学史研究, 1996, 15(2): 144–150.

[9] BRILL R H. The record of time in weathered glass[J]. Archaeology, 1961, 14(1): 18–22.

[10] 汪家清, 张鑫, 曹彤, et al. 基于 Fisher 线性判别分析对乳腺微钙化性质的预测研究 [J]. 中国医学装备, 2022.

附　录

附录 1：支撑材料说明

表 11　支撑材料清单

序号	内容
0	论文中使用的图片
1	MATLAB 求解代码
2	Python 求解代码
3	SPSS 处理得到的数据

附录 2：卡方检验表

表 12　卡方检验表

指标	名称	表面风化		总计	x^2	连续性修正	P
		无风化	风化				
纹饰	C	13	15	28	5.747	5.747	0.056
	A	11	9	20			
	B	0	6	6			
合计		24	30	54			
类型	高钾	12	6	18	5.400	4.134	0.020
	铅钡	12	24	36			
合计		24	30	54			
颜色	蓝绿	6	9	17	6.287	6.287	0.507
	浅蓝	8	12	20			
	紫	2	2	4			
	深绿	3	4	7			
	深蓝	2	0	2			
	浅绿	2	1	3			
	黑	0	2	2			
	绿	1	0	1			
合计		24	30	54			

附录 3：高钾玻璃的描述性统计

表 13 高钾玻璃的描述性统计

	N 统计	范围统计	最小值统计	最大值统计	合计统计	均值统计	标准偏差统计	方差统计	偏度		峰度	
									统计	标准错误	统计	标准错误
氧化硅 (SiO2)	49	71.79	3.72	75.51	1904.90	38.8755	18.64646	347.691	.147	.340	-1.019	.668
氧化钠 (Na20)	49	7.92	.00	7.92	44.32	.9045	1.81276	3.286	2.299	.340	5.069	.668
氧化钾 (K2O)	49	1.41	.00	1.41	8.50	.1735	.27550	.076	2.744	.340	9.280	.668
氧化钙 (Ca0)	49	6.40	.00	6.40	100.45	2.0500	1.63461	2.672	.745	.340	-.291	.668
氧化镁 (MgO)	49	2.73	.00	2.73	31.63	.6455	.63005	.397	.763	.340	.700	.668
氧化铝 (AI2O3)	49	13.89	.45	14.34	179.71	3.6676	3.00900	9.054	2.211	.340	5.531	.668
氧化铁 (Fe203)	49	4.59	.00	4.59	32.14	.6559	.94845	.900	2.067	.340	5.316	.668
氧化铜 (Cu0)	49	10.57	.00	10.57	92.10	1.8796	2.47043	6.103	2.286	.340	5.256	.668
氧化铅 (PbO)	49	60.91	9.30	70.21	1634.11	33.3492	14.94731	223.422	390	.340	-.622	.668
氧化钡 (BaO)	49	35.45	.00	35.45	514.03	10.4904	8.33136	69.412	1.612	.340	2.253	.668
五氧化二磷 (P205)	49	14.13	00	14.13	161.34	3.2927	3.90923	15.282	1.097	.340	.210	.668
氧化锶 (SrO)	49	1.12	.00	1.12	17.05	.3480	.26350	.069	.768	.340	.612	.668
氧化锡 (SnO2)	49	1.31	.00	1.31	2.85	.0582	.21300	.045	4.773	.340	25.716	.668
氧化硫 (SO2)	49	15.95	.00	15.95	39.18	.7996	3.13870	9.851	4.473	.340	19.619	.668
总和	49	11.57	88.41	99.98	4762.31	97.1900	2.65918	7.071	-1.514	.340	2.087	.668
有效个案数 (成列)	49											

附录 4：铅钡玻璃的描述性统计

表 14 铅钡玻璃的描述性统计

	N 统计	范围统计	最小值统计	最大值统计	合计统计	均值统计	标准偏差统计	方差统计	偏度		峰度		
									统计	标准错误	统计	标准错误	
二氧化硅 (SiO2)	18	37.76	59.01	96.77	1379.59	76.6439	3.40984	14.46673	209.286	.205	.536	-1.816	1.038
氧化钠 (Na2O)	18	3.38	.00	3.38	8.34	.4633	.25661	1.08871	1.185	2.127	.536	3.143	1.038
氧化钾 (K2O)	18	14.52	.00	14.52	115.23	6.4017	1.25105	5.30775	28.172	-.045	.536	-1.715	1.038
氧化钙 (CaO)	18	8.70	.00	8.70	69.21	3.8450	.77973	3.30813	10.944	.189	.536	-1.782	1.038
氧化镁 (MgO)	18	1.98	.00	1.98	14.13	.7850	.16777	.71180	.507	.287	.536	-1.398	1.038
氧化铝 (Al2O3)	18	10.34	.81	11.15	91.02	5.0567	.72518	3.07666	9.466	.477	.536	-.627	1.038
氧化铁 (Fe2O3)	18	6.04	.00	6.04	24.77	1.3761	.36912	1.56603	2.452	1.630	.536	3.373	1.038
氧化铜 (CuO)	18	5.09	.00	5.09	38.80	2.1556	.35172	1.49224	2.227	.514	.536	-.639	1.038
氧化铅 (PbO)	18	1.62	.00	1.62	4.94	.2744	.12118	.51414	.264	1.972	.536	2.723	1.038
氧化钡 (BaO)	18	2.86	.00	2.86	7.18	.3989	.19837	.84163	.708	2.112	.536	3.766	1.038
五氧化二磷 (P2O5)	18	4.50	.00	4.50	18.51	1.0283	.30184	1.28060	1.640	2.154	.536	4.125	1.038
氧化锶 (SrO)	18	.12	.00	.12	.50	.0278	.01034	.04387	.002	1.231	.536	-.066	1.038
氧化锡 (SnO2)	18	2.36	.00	2.36	2.36	.1311	.13111	.55626	.309	4.243	.536	18.000	1.038
二氧化硫 (SO2)	18	.47	.00	.47	1.22	.0678	.03704	.15716	.025	2.025	.536	2.510	1.038
有效个案数（成列）	18												

附录 5：聚类中心

表 15 聚类中心

化学成分	初始聚类中心		最终聚类中心	
	1	2	1	2
二氧化硅 (SiO2)	59.01	96.77	63.62	89.66
氧化钠 (Na2O)	2.86	.00	.93	.00
氧化钾 (K2O)	12.53	.92	10.82	1.99
氧化钙 (CaO)	8.70	.21	6.36	1.33
氧化镁 (MgO)	.00	.00	1.13	.44
氧化铝 (Al2O3)	6.16	.81	7.35	2.76
氧化铁 (Fe2O3)	2.88	.26	2.31	.44
氧化铜 (CuO)	4.73	.84	2.82	1.49
氧化铅 (PbO)	.00	.00	.41	.14
氧化钡 (BaO)	.00	.00	.58	.22
五氧化二磷 (P2O5)	1.27	.00	1.52	.53
氧化锶 (SrO)	.00	.00	.05	.01
氧化锡 (SnO2)	.00	.00	.00	.26
二氧化硫 (SO2)	.00	.00	.14	.00

附录 6：聚类热力图

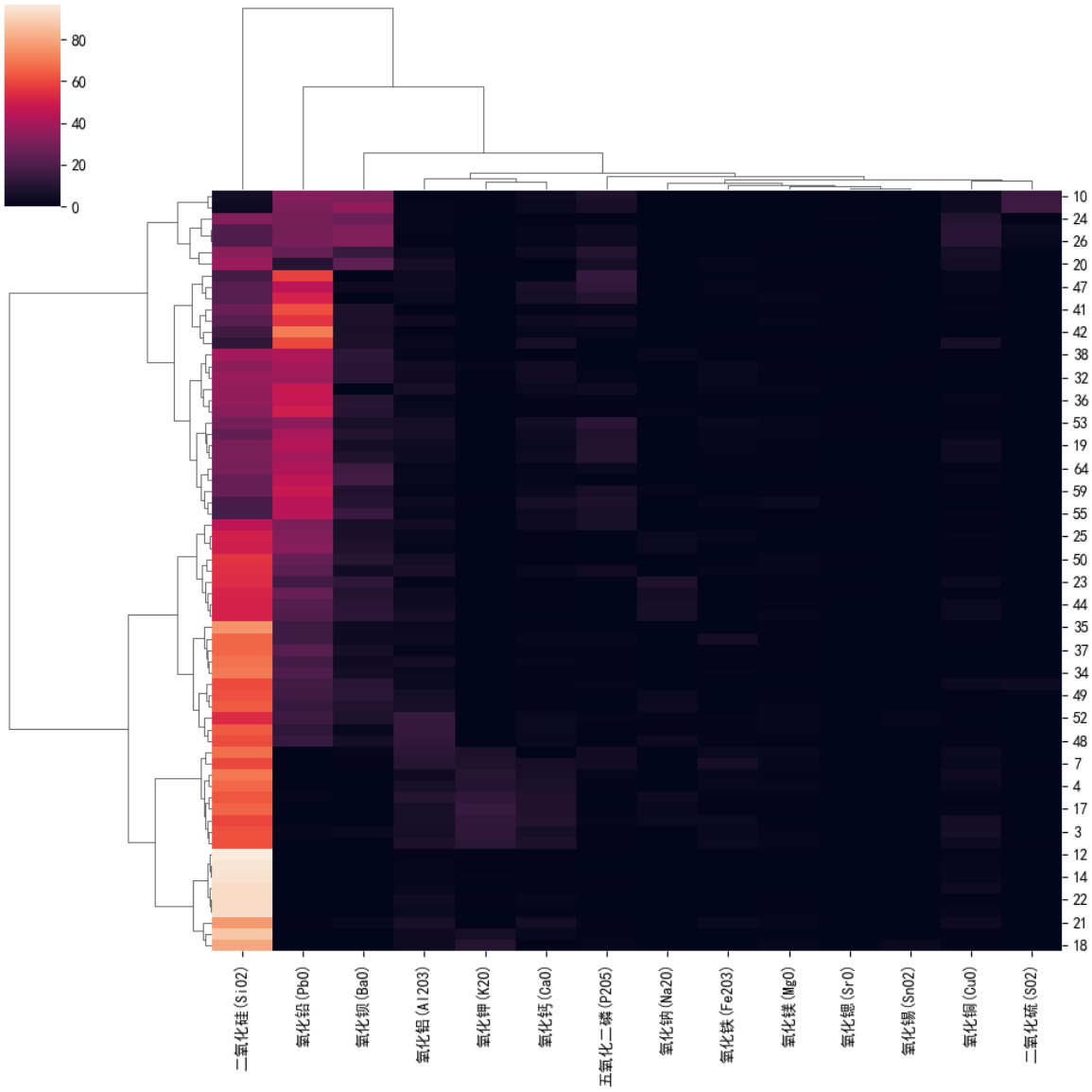


图 14 聚类热力图

附录 7：数据处理代码 dataProcessing.py

```python
import re
import pandas as pd
import itertools
cc = ['二氧化硅(SiO2)', '氧化钠(Na2O)', '氧化钾(K2O)', '氧化钙(CaO)', '氧化镁(MgO)', '氧化铝(Al2O3)', '氧化铁(Fe2O3)', '氧化铜(CuO)', '氧化铅(PbO)', '氧化钡(BaO)', '五氧化二磷(P2O5)', '氧化锶(SrO)', '氧化锡(SnO2)', '二氧化硫(SO2)']
cc_re = list(itertools.chain.from_iterable([re.findall(r"[(](.*?)[)]", _) for _ in cc]))
def get_yes_data():
    baseData = pd.read_excel("data/附件.xlsx", sheet_name=0)
    yesData = pd.read_excel("data/附件.xlsx", sheet_name=1).fillna(0)
    yesData['总和'] = yesData.iloc[:, 1:].apply(lambda x: x.sum(), axis=1)
    yesData['有效性'] = yesData['总和'].map(lambda x: 1 if 85 <= x <= 105 else 0)
    yesData = yesData[yesData['有效性'] == 1]
    yesData.insert(loc=1, column='文物编号', value=[int(yesData.iloc[i, 0][:2]) for i in range(yesData.shape[0])])
    yesData.insert(loc=3, column='采样点风化', value=[yesData.iloc[i, 0][2:] for i in range(yesData.shape[0])])
    yesData = pd.merge(yesData, baseData, on="文物编号")
    yesData.rename(columns={'类型': '类型(KIND)'}, inplace=True)
    # yesData[' 类型 (KIND)'] = yesData[' 类型 (KIND)'].replace(' 高钾': 0, ' 铅钡': 1)
    for index, row in yesData.iterrows():
        if '未' in str(row['采样点风化']):
            yesData['表面风化'][index] = '无风化'
    yesData.rename(columns={'表面风化': '真实风化'}, inplace=True)
    cols = ['文物采样点', '纹饰', '颜色', '真实风化', '类型(KIND)', '二氧化硅(SiO2)', '氧化钠(Na2O)', '氧化钾(K2O)', '氧化钙(CaO)', '氧化镁(MgO)', '氧化铝(Al2O3)', '氧化铁(Fe2O3)', '氧化铜(CuO)', '氧化铅(PbO)', '氧化钡(BaO)', '五氧化二磷(P2O5)', '氧化锶(SrO)', '氧化锡(SnO2)', '二氧化硫(SO2)']
    yesData = yesData[cols]
    yesData.to_excel('data/yesData.xlsx')
    yesData[yesData['类型(KIND)'] == '高钾'].to_excel('data/yesData_GaoJia.xlsx')
    yesData[yesData['类型(KIND)'] == '铅钡'].to_excel('data/yesData_QianBei.xlsx')
    return yesData
def get_no_data():
    noData = pd.read_excel("data/附件.xlsx", sheet_name=2).fillna(0)
    noData['总和'] = noData.iloc[:, 2:].apply(lambda x: x.sum(), axis=1)
    noData['有效性'] = noData['总和'].map(lambda x: 1 if 85 <= x <= 105 else 0)
    noData = noData[noData['有效性'] == 1]
    cols = ['文物编号', '二氧化硅(SiO2)', '氧化钠(Na2O)', '氧化钾(K2O)', '氧化钙(CaO)', '氧化镁(MgO)', '氧化铝(Al2O3)', '氧化铁(Fe2O3)', '氧化铜(CuO)', '氧化铅(PbO)', '氧化钡(BaO)', '五氧化二磷(P2O5)', '氧化锶(SrO)', '氧化锡(SnO2)', '二氧化硫(SO2)']
    noData = noData[cols]
```

```
    noData.to_excel('data/noData.xlsx')
    return noData
```

附录 8：main 函数代码 main.py

```
#!/usr/bin/env python
# coding: utf-8
import pandas as pd
from matplotlib import pyplot as plt
from dataProcessing import get_yes_data
from t1 import t1_draw, t1_forest
from t2 import t2_draw
from t2_1 import t2_draw2
from t3 import t3
from t4 import t4_draw
from t4_2 import t4_draw_2
import warnings
warnings.filterwarnings("ignore")
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('max_colwidth', 100)
pd.set_option('display.width', 1000)
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
if __name__ == "__main__":
    data = get_yes_data()
    t1_draw(data)
    t1_forest(data)
    t2_draw(data)
    t2_draw2(data)
    t3(data)
    t4_draw(data)
    t4_draw_2(data)
```

附录 9：第一题求解代码 t1.py

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages
from dataProcessing import cc_re, cc
def t1_draw(yesData):
    gaoJiaFengHua = yesData[(yesData['类型(KIND)'] == '高钾') & (yesData['真实风化'] ==
        '风化')][cc].sum()
```

```
    gaoJiaWuFeng = yesData[(yesData['类型(KIND)'] == '高钾') & (yesData['真实风化'] ==
        '无风化')][cc].sum()
    qianBeiFengHua = yesData[(yesData['类型(KIND)'] == '铅钡') & (yesData['真实风化']
        == '风化')][cc].sum()
    qianBeiWuFeng = yesData[(yesData['类型(KIND)'] == '铅钡') & (yesData['真实风化'] ==
        '无风化')][cc].sum()
    x = np.arange(len(cc))
    y1, y2 = np.array(gaoJiaFengHua), np.array(gaoJiaWuFeng)
    fig1 = plt.figure()
    ax = plt.subplot(1, 1, 1)
    width = 0.4
    ax.bar(x, y1, width, color='r')
    ax.bar(x + width, y2, width, color='g')
    ax.set_xticks(x + width)
    ax.set_xticklabels(cc_re, rotation=40)
    plt.legend(labels=["风化", "无风化"], loc="upper right")
    plt.show()
    pp = PdfPages("figures/高钾玻璃风化前后化学成分统计图.pdf")
    pp.savefig(fig1, bbox_inches='tight')
    pp.close()
    y1, y2 = np.array(qianBeiFengHua), np.array(qianBeiWuFeng)
    fig2 = plt.figure()
    ax = plt.subplot(1, 1, 1)
    width = 0.4
    ax.bar(x, y1, width, color='r')
    ax.bar(x + width, y2, width, color='g')
    ax.set_xticks(x + width)
    ax.set_xticklabels(cc_re, rotation=40)
    plt.legend(labels=["风化", "无风化"], loc="upper right")
    plt.show()
    pp = PdfPages("figures/铅钡玻璃风化前后化学成分统计图.pdf")
    pp.savefig(fig2, bbox_inches='tight')
    pp.close()
    print('题目一图片保存完成!')
def t1_forest(yesData):
    for index, row in yesData.iterrows():
        if row['真实风化'] == '风化':
            rwww = pd.concat([row[:5], yesData[(yesData['真实风化'] == '无风化') & (
        yesData['纹饰'] == row['纹饰']) & (yesData['类型(KIND)'] == row['类型(KIND)'
        ])][cc].mean(axis=0)])
            if pd.isnull(rwww['二氧化硅(SiO2)']):
                rwww = pd.concat([row[:5], yesData[(yesData['真实风化'] == '无风化') & (
        yesData['类型(KIND)'] == row['类型(KIND)'])][cc].mean(axis=0)])
            rwww = rwww.to_frame()
            rwww = pd.DataFrame(rwww.values.T, columns=rwww.index)
            yesData = pd.concat([yesData, rwww], ignore_index=True)
    yesData.to_excel('题目一预测.xlsx', index=None)
    print('题目一预测.xlsx保存完成!')
```

附录 10：第二题求解代码其一）t2.py

```python
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.manifold import TSNE
from sklearn.cluster import KMeans
from matplotlib.backends.backend_pdf import PdfPages
from dataProcessing import cc
from t3 import gauss_noisy
def draw(data, errorValue):
    k = 4
    iteration = 114514
    data_zs = 1.0 * (data - data.mean()) / data.std()
    model = KMeans(n_clusters=k, max_iter= iteration )
    model.fit(data_zs)
    r1 = pd.Series(model.labels_).value_counts()
    r2 = pd.DataFrame(model.cluster_centers_)
    r = pd.concat([r2, r1], axis=1)
    r.columns = list(data.columns) + [u'类别数目']
    r = pd.concat([data, pd.Series(model.labels_, index=data.index)], axis=1)
    r.columns = list(data.columns) + [u'聚类类别']
    r.to_excel(f't2_{errorValue}%.xlsx')
    tsne = TSNE()
    tsne.fit_transform(data_zs)
    tsne = pd.DataFrame(tsne.embedding_, index=data_zs.index)
    plt.rcParams['font.sans-serif']=['SimHei']
    plt.rcParams['axes.unicode_minus']=False
    fig = plt.figure()
    d = tsne[r[u'聚类类别']==0]
    plt.plot(d[0], d[1], 'r.')
    d = tsne[r[u'聚类类别']==1]
    plt.plot(d[0], d[1], 'go')
    d = tsne[r[u'聚类类别']==2]
    plt.plot(d[0], d[1], 'b*')
    d = tsne[r[u'聚类类别']==3]
    plt.plot(d[0], d[1], 'y.')
    plt.show()
    pp = PdfPages(f"figures/kmeans聚类图_{errorValue}%.pdf")
    pp.savefig(fig, bbox_inches='tight')
    pp.close()
def t2_draw(yesData):
    data = yesData[cc]
    errorlist = [0.1, 0.2, 0.5]
    for errorValue in errorlist:
```

```
        gauss_noisy(data, errorValue)
        draw(data, errorValue)
```

附录 11：第二题求解代码其二 t2_1.py

```
import seaborn as sns
from scipy.cluster import hierarchy
from scipy import cluster
import matplotlib.pyplot as plt
from sklearn import decomposition as skldec
from dataProcessing import cc
def t2_draw2(yesData):
    data = yesData[cc]
    Z = hierarchy.linkage(data)
    hierarchy.dendrogram(Z, labels=data.index)
    label = cluster.hierarchy.cut_tree(Z, height=0.8)
    label = label.reshape(label.size, )
    pca = skldec.PCA(n_components=0.95) # 选择方差 95pca.fit(data)
    result = pca.transform(data)
    plt.xticks(rotation=20)
    plt.show()
    cg = sns.clustermap(data, method='ward', metric='euclidean')
    plt.setp(cg.ax_heatmap.yaxis.get_majorticklabels(), rotation=0)
    plt.show()
```

附录 12：第三题求解代码 t3.py

```
import itertools
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from dataProcessing import cc, get_no_data, get_yes_data
def gauss_noisy(x, error2):
    for i in range(np.shape(x)[0]):
        for j in range(np.shape(x)[1]):
            x.iloc[i, j] = (x.iloc[i, j] * (1 + (np.random.random() - 0.5) * error2))
def t3(yesData):
    cc2 = [*cc, '类型(KIND)']
    model_data = get_yes_data()[cc2]
    predict_data = get_no_data()[cc2[:-1]]
    x, y = model_data.drop(['类型(KIND)'], axis=1), model_data['类型(KIND)']
```

```
    x_train , x_test , y_train , y_test = train_test_split (x, y, test_size =0.3, random_state
        =10)
    model = LogisticRegression ()
    model. fit ( x_train , y_train )
    print( accuracy_score ( y_test , model. predict ( x_test )))
     predict_result = pd.DataFrame(model.predict( predict_data ))
     predict_result . to_excel ('predict_result.xlsx')
    print( predict_result )
    print('题目三预测结果保存完成！')
     errorlist = [0.05, 0.1, 0.15]
    for errorValue in errorlist :
        su = 0
        for _, _ in itertools .product(range(3), range(100)):
            gauss_noisy( predict_data , errorValue )
            su += accuracy_score( predict_result . values , model. predict ( predict_data ))
        print(su / 100 / 3 * 100)
    # errorValue = 0.05 97.3625 94.925 95.76249999999999
    # errorValue = 0.10 81.4625 90.3625 82.975
    # errorValue = 0.15 63.337500000000006 75.7125 78.95
```

附录 13：第四题求解代码其一 t4.py

```
import numpy as np
import seaborn as sns
import matplotlib . pyplot as plt
from matplotlib .backends.backend_pdf import PdfPages
from dataProcessing import get_yes_data , cc, cc_re
def mmin(x0, x):
    a = np.abs(x - x0)
    b = np.min(a, axis=1)
    return b.min()
def mmax(x0, x):
    a = np.abs(x - x0)
    b = np.max(a, axis=1)
    return b.max()
def kesi(x0, x, amin, bmax, k, ro=0.5):
    c = np.abs(x - x0)
    kesi_k = (amin + ro * bmax) / (c + ro * bmax)
    return kesi_k.mean(axis=1).reshape(-1)
def RA_m(x1, x):
    amin = mmin(x1[0], x)
    bmax = mmax(x1[0], x)
    res = kesi(x1[0], x, amin, bmax, 1, ro=0.5)
    for row in range(1, x1.shape[0]):
        x0 = x1[row]
        amin = mmin(x0, x)
```

```
            bmax = mmax(x0, x)
            res1 = kesi(x0, x, amin, bmax, 1, ro=0.5)
            res = np.vstack((res, res1))
    return res
def draw(data, kind):
    x = data[cc].stack().unstack(0).values
    G = RA_m(x, x)
    plt.figure()
    fig, ax = plt.subplots(1, 1)
    sns.heatmap(G, cmap='Spectral_r', annot_kws={"fontsize": 7}, annot=True, square=
        True)
    cbar = ax.collections[0].colorbar
    cbar.ax.tick_params(labelsize=10)
    plt.xticks(np.arange(x.shape[0]) + 0.5, cc_re, fontsize=9, rotation=30)
    plt.yticks(np.arange(x.shape[0]) + 0.5, cc_re, fontsize=9, rotation=30)
    plt.show()
    pp = PdfPages(f"figures/{kind}玻璃化学成分热力图.pdf")
    pp.savefig(fig, bbox_inches='tight')
    pp.close()
def t4_draw(yesData):
    yesData = get_yes_data()
    yesData_GaoJia = yesData[yesData['类型(KIND)'] == '高钾']
    yesData_QianBei = yesData[yesData['类型(KIND)'] == '铅钡']
    draw(yesData_GaoJia, '高钾')
    draw(yesData_QianBei, '铅钡')
    print('题目四图片保存完成!')
```

附录 14：第四题求解代码其二 t4_2.py

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from matplotlib.backends.backend_pdf import PdfPages
from dataProcessing import cc, cc_re, get_yes_data
def draw(data, kind):
    wr = pd.ExcelWriter(f'{kind}方差分析.xlsx')
    data_melt = data.melt()
    data_melt.columns = ['Treat', 'value']
    fig = plt.figure()
    sns.boxplot(x='Treat', y='value', data=data_melt)
    plt.xticks(np.arange(0, 14, 1), cc_re, rotation=30, fontsize=12)
    plt.ylabel("占比（%）", fontsize=12)
```

```
    plt.xlabel("")
    plt.show()
    pp = PdfPages(f"figures/{kind}玻璃方差分析箱线图.pdf")
    pp.savefig(fig, bbox_inches='tight')
    pp.close()
    model = ols('value ~C(Treat)', data=data_melt).fit()
    anova_table = anova_lm(model, type=2)
    pd.DataFrame(anova_table).to_excel(wr, sheet_name='方差分析')
    Results = pairwise_tukeyhsd(data_melt['value'], data_melt['Treat'])
    df = pd.DataFrame(data=Results._results_table.data[1:], columns=Results._results_table.
        data[0])
    df.to_excel(wr, sheet_name='多重比较', index=False)
    wr.save()
def t4_draw_2(yesData):
    yesData = get_yes_data()
    yesData_GaoJia = yesData[yesData['类型(KIND)'] == '高钾']
    yesData_QianBei = yesData[yesData['类型(KIND)'] == '铅钡']
    draw(yesData_GaoJia[cc], '高钾')
    draw(yesData_QianBei[cc], '铅钡')
```