题目：基于变差系数和相对变率亚类划分模型的玻璃制品成分分析与鉴别

摘 要

古代玻璃极易受埋藏环境的影响而风化，致使玻璃表面形态和内部化学成分受到改变，从而影响到对玻璃类别的判断。本文针对玻璃风化情况，在建立多元线性回归风化修正模型的基础上，构建基于变差系数和相对变率的亚类划分模型，并进一步构建模型灵敏度评估体系，对玻璃制品进行分析与鉴别。

针对问题一， 第一问采用**卡方独立性检验**探究两个分类变量是否存在关联关系，即文物的表面风化和其玻璃类型、纹饰和颜色的关系。 结果表明：玻璃表面风化与其类型、纹饰有关，与颜色无关。第二问，将玻璃类型和有无风化同时作为独立变量，探究化学成分含量的统计规律。采用**无交互双因素方差分析**，得到对变量影响显著的化学成分；将其与对分类与风化的描述性统计结果结合讨论，得到统计规律。第三问建立**多元线性回归预测模型**，用已知的变量值预测因变量取值。最后进行拟合优度检验，判断模型精度。

针对问题二，考虑风化后化学成分改变，建立基于多元线性回归和 XGBoost 的修正分类模型。将修正数据带入分类模型中训练，采用 **LIME 方法**解释分类规律。结果为：过高、过低的 PbO 可以有效对铅钡、高钾玻璃分类。进一步从**变差系数**、**相对变率**两个角度出发，建立基于化学成分的亚类划分模型，得出最终亚类划分结果：亚分类可将高钾、铅钡玻璃分别再分为 4 类、5 类，并通过不断改变化学成分进行灵敏度分析。结果表明：随空值填充值变化，变差系数有较小浮动，灵敏度较差，模型稳定性好。

针对问题三，分析亚分类结果中各成分含量特征，人工初步判断未知玻璃文物类型，进一步基于第二问的修正分类模型和亚类划分模型，进行分类和亚类划分。通过改变**XGBoost 分类模型**的决策树个数、学习率、随机采样比率，对模型灵敏度进行检验。结果表明：XGBoost 分类模型分结果与初分类结果一致，准确率为 1.0；进一步亚分类可将高钾、铅钡玻璃为 3 类、2 类；参数改变准确率变化范围在 0.1 以内，效果良好。

针对问题四，采取协方差计算各化学成分之间的相关系数，发现 SiO_2 与许多化学成分都有显著正相关。再进一步基于 **Wilcoxon 秩和检验**，对不同类别文物的化学成分的相关系数向量进行显著性检验，从而得到高钾、铅钡玻璃化学成分的差异性。结果表明：高钾玻璃中只有 SiO_2 与 P_2O_5、MgO、K_2O、Fe_2O_3、BaO、Al_2O_3 等 6 种化学成分具有显著正相关关系；两种玻璃中 SiO_2、P_2O_5、MgO、K_2O、Fe_2O_3、BaO、Al_2O_3 等 7 种成分之间存在显著差异。

关键词：无交互双因素分析 多元线性回归 XGBoost 变差系数 wilcoxon 秩和检验

一、问题重述

玻璃的生产和发展具有鲜明的时代性和地域性。我国古代玻璃最早从国外传来，后按照本土方式和审美进行制造。古代玻璃的主要原料是石英砂，在炼制过程中需加入助熔剂降低熔化温度，加入稳定剂使反应稳定。加入的助熔剂不同，最后制成的玻璃的化学成分也不同。

然而，古代玻璃极易受埋藏环境的影响而风化。风化导致了玻璃表面形态和内部化学成分受到改变，从而影响到对玻璃类别的正确判断。专家们通过各类检测手段，分析出部分文物样品的化学成分，再与对应的外观特征结合，将其分为高钾玻璃和铅钡玻璃两种类型。不同的化学成分受风化的影响程度也不尽相同。本文根据附件中给出的信息数据和实际情况，建立模型解决以下问题：

(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。

(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。

(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。

(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。

二、问题分析

针对问题一，题目要求首先分析文物的表面风化与其玻璃类型、纹饰和颜色的关系。即对两个分类变量间的关联关系进行分析。题目要求其次结合玻璃类型分析化学成分含量在有无风化情况下的统计规律。即探究化学成分在不同的玻璃类型下对文物表面是否风化有无显著差异。题目要求最后根据风化点检测数据，预测其风化前的化学成分含量。可以建立预测模型，找到因变量与自变量的关系方程后，用一个已知的变量去根据预测风化前的化学成分。

针对问题二，题目要求首先分析玻璃的分类规律。分析分类规律可以以机器学习算法为基础建立分类模型。由于风化会导致玻璃的化学成分比例发生生变化，所以需要先使用第一问建立的化学成分预测模型，还原各成分风化前的含量，以消除风化对判断玻璃类型的影响。题目要求其次对每个类别进行亚类划分。在已有的分类基础上，需要选出信息量大，代表性强的化学成分作为亚类进行划分。题目要求最后对分类结果的合理

性和敏感性进行分析。选择合适的方式进行敏感度分析，敏感性越差，模型稳定性越强。通过分析模型建立原理的正确性以说明亚类划分结果的合理性。

针对问题三，题目是在问题二的分类规律的基础上，鉴别未知文物类型，并对分类结果的敏感性进行分析。思路与问题二一致，相当于对问题二所建模型的现实应用，检验模型的可行性。将数据处理后带入建好的模型即可得出具体类型。为更好的检验模型可行性，应采用另一种方式进行敏感性分析。

针对问题四，题目要求分析不同种类玻璃样品化学成分间的关联关系，并比较不同类别之间化学成分关联关系的差异性。对于某一种类样品化学成分间的关联关系，可以通过对两个变量的相关系数分析来描述两个变量之间的关系。并进一步用相关性的其他验证性质分析两变量间的关联关系。对于不同种类样品化学成分关联关系的差异性，对二者化学成分的相关系数向量进行差异性检验。

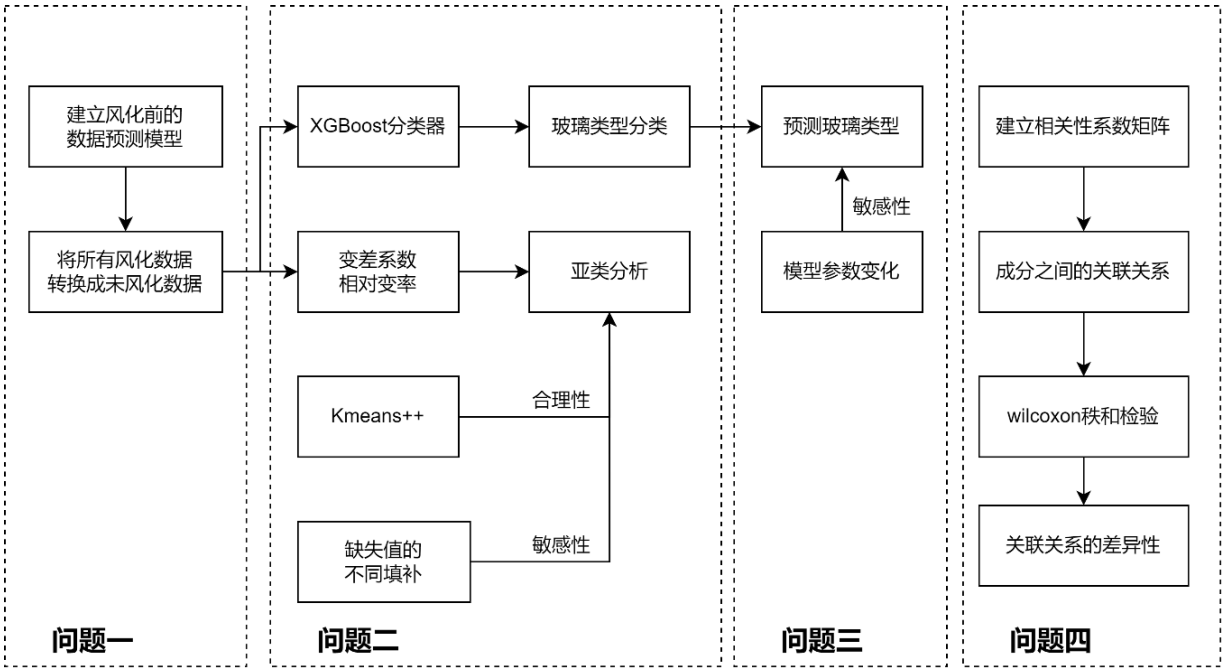

图 1 技术路线图

三、模型假设

为使建立的模型得到简化，本文进行以下假设：

(1) 假设为玻璃文物在风化过程中，其内部元素仅与外部环境发生反应，仅将与外部环境反应产生的磨损当作风化过程中的变化；

(2) 假设风化开始后，认为所有该批样品经历相同的风化过程，忽略各种随机因素造成的干扰。

四、符号说明

符号	符号说明
x_i	第 i 个非随机因素
y_i	第 i 个回归方程
A_i	高钾玻璃回归方程组别 i
B_i	铅钡玻璃回归方程组别 i
S_x	标准差
R	相关性系数矩阵
r_{ij}	第 i 行第 j 个相关系数
n_i	样本 i 的大小
U_i	第 i 个检验样本统计量
R_i	样本 i 的检验等级

五、问题一分析与处理

需要进行全文的数据预处理，剔除无效值。选择卡方独立性检验分析两类变量间的关联性和依存性。用无交互双因素方差对各因素的各个水平对两个指标影响的显著性进行分析，将其结果与对玻璃样品在分类、风化两种指标下的化学成分进行的描述性统计分析综合，作为最终的统计规律。最后建立多元线性方程来预测风化前的化学成分含量，并进行拟合优度检验。

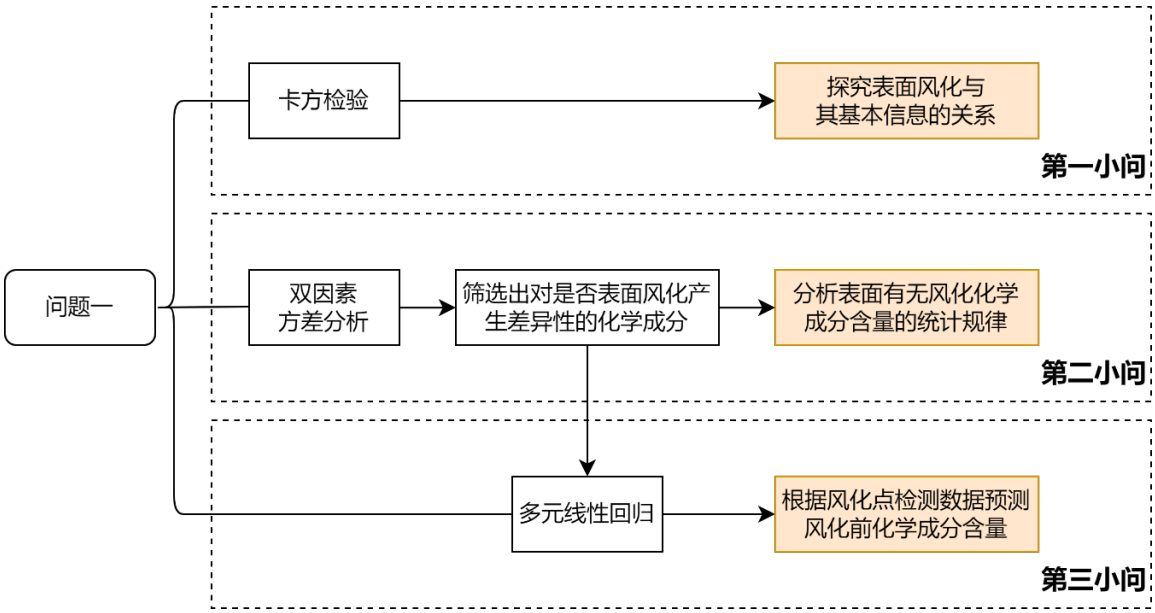

图 2 问题一流程图

5.1 数据预处理

题目中说明附件表单 2 给出的相应的主要成分所占比例数据为成分性，将各成分比例累加和介于 85%~105%间的数据视为有效数据。计算出各成分比例的累加和后，发现：

(1) 编号 15 和编号 17 的玻璃文物样本各成分比例累加和小于 85%，属于无效数据，应当剔除；

(2) 编号 19、编号 40、编号 48、编号 58 的颜色缺失，应当剔除；

(3) 严重风化检测点玻璃样本所含的二氧化硅成分比例远小于其他检测点含量，已失去玻璃的基本性质，无法进行玻璃性质相关分析，应当剔除；

(4) 将表面风化但存在未风化点的数据记为未风化。

5.2 卡方独立性检验分析统计规律

卡方独立性检验主要用于多个因素多项分类的计数分析，也就是研究两类变量间的关联性和依存性。本题使用卡方独立性检验判断所分析变量间是否相关。

5.2.1 卡方独立性检验分析的步骤

(1) 建立原假设和备择假设。

原假设 H_0 ：表面风化和对应的自变量间无关联，相互独立。

备择假设 H_1：表面风化和对应的自变量间有关联，相互不独立。

(2) 计算自由度和理论频数。

$$x_{1-\frac{\alpha}{2}}^{2}(df) < x^2 < x_{\frac{\alpha}{2}}^{2}(df) \tag{1}$$

自由度公式为：

$$df = (r-1)(c-1) \tag{2}$$

理论频数公式为：

$$e_{ij} = \frac{F_{Yi} \cdot F_{Xj}}{n} \tag{3}$$

(3) 计算统计量：

$$x^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(f_{ij}-e_{ij})^2}{e_{ij}} \sim x^2(df) \tag{4}$$

(4) 查 χ^2 分布临界值表，确定接受域和拒绝域：

$$x_{1-\frac{\alpha}{2}}^{2}(df) < x^2 < x_{\frac{\alpha}{2}}^{2}(df) \tag{5}$$

(5) 做出统计决策。

利用卡方检验分析表中输出 P 值与显著性水平 α=0.1 进行比较，做出统计决策。

若 $P<\alpha$，则拒绝原假设 H_0，表明表面风化和对应的自变量间有关联，相互不独立；反之，则不拒绝原假设 H_0，表明表面风化和对应的自变量间无关联，相互独立。

5.2.2 卡方独立性检验分析的求解

本题将表面风化作为一类变量，玻璃类型、纹饰和颜色分别作为另一类变量进行独立性卡方检验。所得结果如表 1 所示：

表 1 卡方独立性检验分析表

	自由度	卡方统计量	*P* 值
类型	1	2.758	0.09676.
纹饰	2	5.720	0.05727.
颜色	7	7.011	0.4277
显著程度：0.001: ‘***’；0.01: ‘**’；0.05：‘*’；0.1 ‘.’；1 ‘’			

如表 1 所示，$P_{类型}$ <0.1,$P_{纹饰}$<0.1，$P_{颜色}$>0.1，

结果表明：玻璃文物的表面风化与玻璃的类型和纹饰有关，与颜色无关。

5.3 无交互双因素分析统计规律

本题以玻璃样品的类型为自变量，有无风化状态分别作为因变量进行双因素方差分析，以判断各因素的各个水平对两个指标的影响是否显著[2]。由于风化状态因素水平下的指标好坏及其程度不受未风化因素水平的影响，所以两个因素间无交互作用。

5.3.1 无交互双因素分析的原理和步骤

设类别为因素 A，有 a 个不同取值代表 a 个不同水平，分别用 $A_1,A_2,\ldots,A_a$ 来表示，设表面风化为因素 B，有 b 个不同取值代表 b 个不同水平，分别用 $B_1,B_2,\ldots,B_b$ 来表示。则 A 与 B 的不同组合 $A_iB_j(i=1,2,\ldots,a;j=1,2,\ldots,b)$的数量共有 ab 个，可以得到 ab 个观测值 X_{ij}，得到对于任意以后观测值 X_{ij}，都可以表达成如下的线性组合：

$$X_{ij}=\mu+\alpha_i+\beta_j+\gamma_{ij} \tag{6}$$

其中,μ 是模型的常数项;α_i 表示因素 A 的水平为 i 时的水平误差;β_j 表示因素 B 的水平为 j 时的水平误差。

同时假定 X_{ij} 相互独立,假定每个总体均服从正态分布且每个总体的残差、方差一致。

无交互双因素分析具体步骤如下：

(1) 进行检验假定，建立原假设和备择假设。

通过检验因素的水平均值是否相等，来判断自变量与因变量是否相关。提出如下假设：原假设 H_{0A}：各个总体的水平的均值相等，因素 A 对实验结果的影响比随机误差对实验结果的影响小；

备择假设 H_{1A}：至少存在两个总体的水平不相等，因素 A 对实验结果的影响比随机误差对实验结果的影响大。

因素 *B* 的假设同上。

(2) 构造检验统计量。

为构造检验统计量，首先需要计算出总误差平方和(*SST*)并将其分解为对应的误差因素 *A* 平方和（*SSA*）和误差因素 *B* 平方和（*SSB*）和随机误差平方和（*SSE*）。

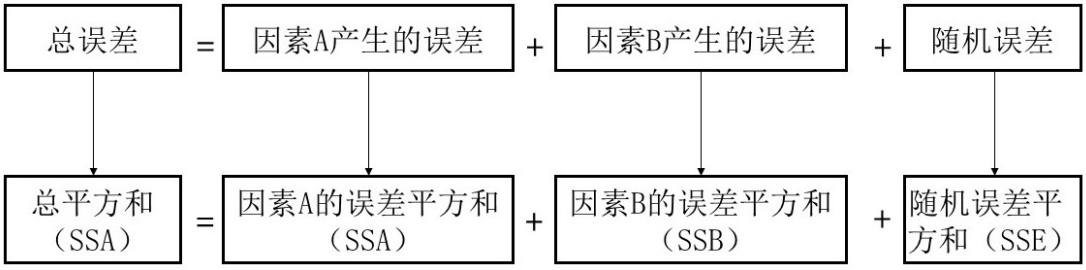

图 3 无交互作用的双因素方差分析误差分解图

三者的恒定关系为：

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}(y_{ijk}-\overline{\overline{y}})^2 = JK\sum_{i=1}^{I}(Y_{i\bullet}-\overline{\overline{y}})^2 + IK\sum_{j=1}^{J}(Y_{\bullet j}-\overline{\overline{y}})^2 + \sum_{i=1}^{I}\sum_{j=1}^{J}(y_{ijk}-\overline{y})^2 \tag{7}$$

即 *SST*=*SSA*+*SSB*+*SSE*。

为消除观测值数目对计算结果的影响，用均方误差进行后续计算。

(3) 计算检验统计量。

SSA 的均方误差记作 $MSA, MSA=\dfrac{SSA}{k-1}, F_A=\dfrac{MSA}{MSE}$

SSB 的均方误差记作 $MSB, MSB=\dfrac{SSB}{k-1}, F_B=\dfrac{MSB}{MSE}$

SSE 的均方误差记作 $MSE, MSE=\dfrac{SSE}{k-1}$。

以 SiO_2 为例，以此计算出其对应的双因素方差分析表，如表 2 所示。

表 2 SiO_2双因素方差分析表

	自由度	平方和	均方	*F* 值
因素 A	1.0	18778.049	60.980	7.000761e-11
因素 B	1.0	539.104	1.750	1.90496e-01
随机误差	64.0	19707.911	NaN	NaN

(4) 给定显著性水平，构造拒绝域。

给定显著性水平 α=0.1，确定拒绝域为$W=\{F\geq F_\alpha(f_{a/b},f_e)\}$。

(5) 做出统计决策。

利用方差分析表中输出 *P* 值与显著性水平 α=0.01 进行比较，做出统计决策。

若 $P<\alpha$，则拒绝原假设 H_{0A}，表明因素 *A* 对因变量的影响是显著的；反之，则不拒绝原假设 H_{0A}，表明因素 *A* 对因变量无显著影响。因素 *B* 同理。

5.3.2 无交互双因素分析的求解

最终以 P 值和 α 值的比较得出结论，将 14 种化学元素进行上述无交互双因素分析，得到的对应 P 值的柱状图如图 4 所示：

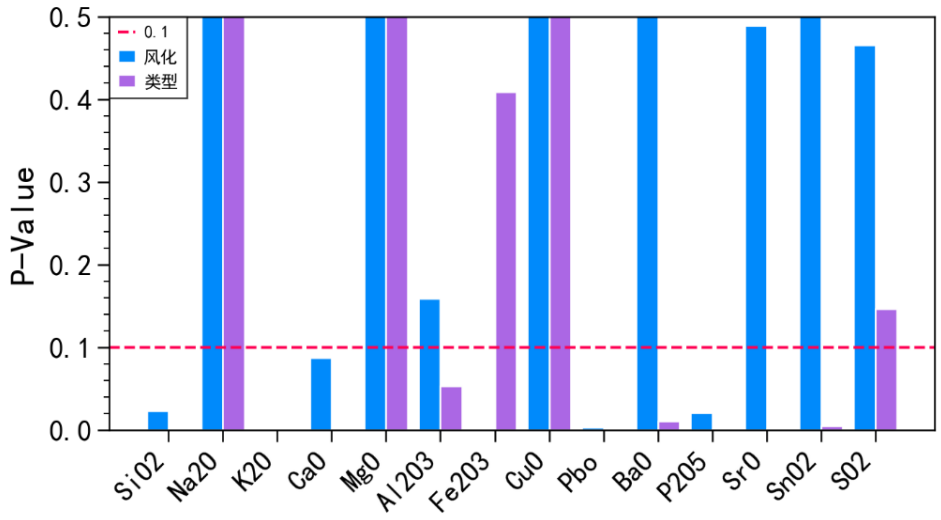

图 4 不同类型下 14 种化学元素对应 P 值柱状图

5.3.3 统计规律的分析

由图 4 可得，玻璃类型作为指标时，SiO_2,K_2O,CaO,Al_2O_3,PbO,BaO,P_2O_5,SrO,SnO_2 的 P 值均小于 0.1，即以上化学元素对玻璃类型影响显著；表面风化作为指标时，SiO_2，K_2O，CaO，Fe_2O_3, PbO,P_2O_5 的 P 值均小于 0.1,即以上化学成分对表面风化影响显著。

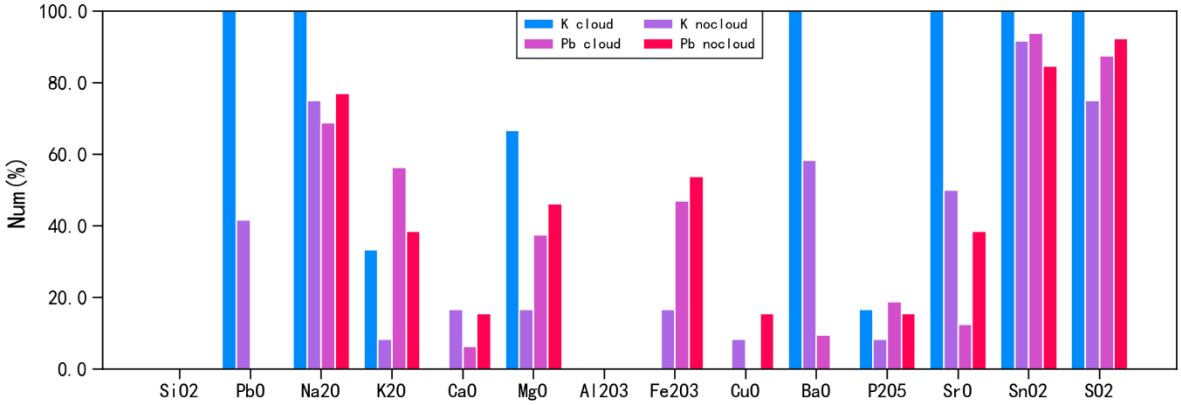

图 5 化学成分缺失值占比柱状图

由图 5 可得：PbO,Na_2O,BaO,SrO,SnO_2,SO_2 在高钾玻璃未风化情况下，缺失值达到 100%，说明，高钾玻璃未风化时，不含这 6 种化学成分。CuO 是两类玻璃风化后必出现的化学成分之一。

5.4 基于多元线性回归的化学成分预测模型

本题采用以多元线性回归为核心建立预测模型。主要思路是：先根据因变量与多个自变量的实际观测值建立因变量对多个自变量的多元线性回归方程，对该线性方程进行

分析检验，判断其准确性。然后利用该线性关系表达式，用一个变量值去预测另一个因变量的取值[3]。

5.4.1 化学成分预测模型的建立

设可预测的随机变量为 y，它受到 p 个非随机因素 $x_1,x_2,\ldots,x_{p-1},x_p$ 和不可预测的随机因素 ε 的影响。多元线性回归模型为：

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\ldots+\beta_{p-1}x_{p-1}+\beta_px_p+\varepsilon \tag{8}$$

其中，$\beta_0,\beta_1,\ldots,\beta_{p-1},\beta_p$ 为回归系数。

对 y 和 $x_1,x_2,\ldots,x_{p-1},x_p$ 分别进行 n 次独立观测，取得 n 组数据，则有

$$\begin{cases} y_1=\beta_0+\beta_1x_{11}+\beta_2x_{12}+\ldots+\beta_{p-1}x_{1p-1}+\beta_px_{1p}+\varepsilon_1 \\ y_2=\beta_0+\beta_1x_{21}+\beta_2x_{22}+\ldots+\beta_{p-1}x_{2p-1}+\beta_px_{2p}+\varepsilon_2 \\ y_3=\beta_0+\beta_1x_{31}+\beta_2x_{32}+\ldots+\beta_{p-1}x_{3p-1}+\beta_px_{3p}+\varepsilon_3 \end{cases} \tag{9}$$

其中，$\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n$ 相互独立，且服从 $N(0,\sigma^2)$ 分布。

使用最小二乘法估计模型参数 β，表达式为：

$$\hat{\beta}=(X^TX)^{-1}X^TY \tag{10}$$

误差方差 σ^2 的估计为：

$$\sigma^2=\frac{1}{n-p}(\hat{\varepsilon}^T,\hat{\varepsilon}) \tag{11}$$

5.4.2 化学成分预测模型的求解

本题以是否风化作为分类指标，以是否分类和有效化学成分（即上文得到的对表面风化影响显著 6 个化学成分 SiO_2，K_2O，CaO，Fe_2O_3, PbO,P_2O_5）按顺序设为自变量 x，任意状态下的 SiO_2，K_2O，CaO，Fe_2O_3, PbO, P_2O_5 按顺序设为因变量 y，进行多元回归分析。得到的两种类型玻璃的回归方程如表 3、表 4 所示。

表 3 高钾玻璃预测模型中的回归方程

组别	因变量	回归方程
A_1	SiO_2	$y_1=84.35x_{11}+1.77x_{13}+6.61x_{14}-14.51x_{15}+10.88x_{16}+24.07x_{17}$
A_2	K_2O	$y_2=-5.27x_{21}-1.51x_{22}+0.04x_{24}-0.03x_{25}+1.22x_{26}+0.05x_{27}$
A_3	CaO	$y_3=-1.09x_{31}+0.26x_{32}+0.01x_{33}-0.002x_{35}-0.02x_{36}+0.004x_{37}$
A_4	Fe_2O_3	$y_4=-0.5x_{41}+0.3x_{42}+0.0002x_{43}-0.10x_{44}+0.30x_{46}+0.004x_{47}$
A_5	PbO	$y_5=0.74x_{51}+1.06x_{52}+0.006x_{53}+0.27x_{54}-0.09x_{55}+0.007x_{57}$
A_6	P_2O_5	$y_6=-0.76x_{61}+0.01x_{62}-0.31x_{63}+0.78x_{64}-0.04x_{65}+0.08x_{66}$

表 4 铅钡玻璃预测模型中的回归方程

组别	因变量	回归方程
B_1	SiO_2	$y_1 = -15.97x_{11} + 0.78x_{13} + 22.74x_{14} - 5.91x_{15} + 3.56x_{16} - 1.83x_{17}$
B_2	K_2O	$y_2 = 1.30x_{21} + 0.0025x_{22} + 1.53x_{24} + 0.05x_{25} - 0.004x_{26} + 1.06x_{27}$
B_3	CaO	$y_3 = 0.18x_{31} + 0.009x_{32} + 0.19x_{33} - 0.007x_{35} - 0.02x_{36} + 0.67x_{37}$
B_4	Fe_2O_3	$y_4 = 3.56x_{41} + 0.10x_{42} - 0.04x_{43} - 1.26x_{44} + 2.41x_{46} - 0.63x_{47}$
B_5	PbO	$y_5 = -0.25x_{51} - 0.01x_{52} + 0.05x_{53} - 0.06x_{54} + 0.27x_{55} + 0.004x_{57}$
B_6	P_2O_5	$y_6 = -0.26x_{61} - 0.01x_{62} + 0.05x_{63} - 0.06x_{64} + 0.27x_{65} + 0.003x_{66}$

以上求得的多元线性方程为任意风化状态下化学成分间的线性关系。将分类指标固定为未风化后，将风化点的有效化学成分带入回归方程中，即可预测风化前的化学成分含量。

5.4.3 化学成分预测模型的拟合优度检验

为评价得到的预测值的准确度，本题对回归直线对观测值的拟合程度，即拟合优度进行检验。在多元线性回归方程中，用判定系数 R^2 衡量拟合优度[6],其数学模型为：

$$R^2 = 1 - \frac{SSE}{SST} \tag{12}$$

其中，$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$ 为总离差平方和，$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ 为残差平方和。

R^2 的值越接近 1，表明回归方程对实际观测值的拟合度效果越好，相反 R^2 越接近 0，拟合效果越差。

本题上述多元线性回归方程的拟合优度如表 5 所示。

表 5 多元线性回归方程的拟合优度

组别	A_1	A_2	A_3	A_4	A_5	A_6	B_1	B_2	B_3	B_4	B_5	B_6
R^2	0.938	0.793	0.722	0.69	0.829	0.932	0.69	0.7	0.695	0.879	0.79	0.79

由表可得，上述多元线性回归方程的 R^2 皆不小于 0.69，最大可达到 0.938，可见该基于多元线性回归的化学成分预测模型效果好，得到的预测值准确率高。

六、问题二分析与处理

先利用建立好的化学成分预测模型，还原各成分风化前的含量，以此消除玻璃风化对判断其类型的影响。再建立基于 XGBoost 的玻璃样品分类模型进行分类预测。根据每个化学成分的变差系数确定该化学成分代表性大小，进行亚类划分。再根据相对变率挑选在该化学成分下存在显著差异的玻璃化学成分。最后选择局部敏感性分析来评价结果敏感性。

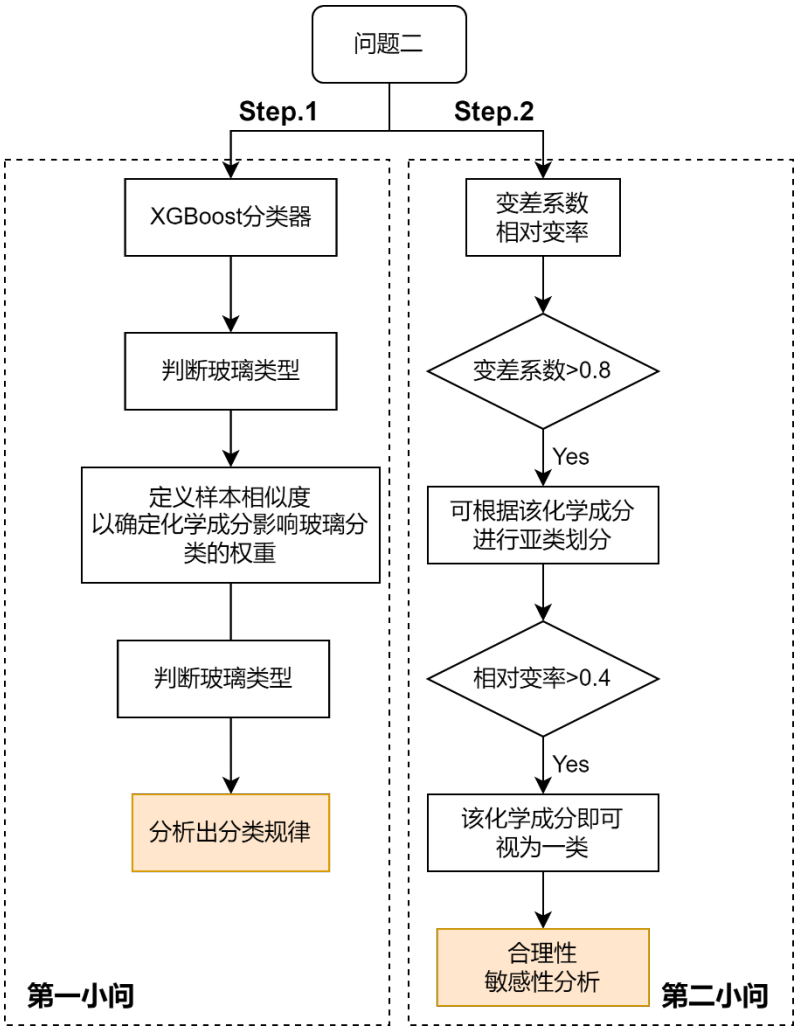

图 6 问题二流程图

6.1 基于 XGBoost 的玻璃样品分类模型

6.1.1 分类数据准备

由于古代玻璃在风化过程中，样品内部与外部环境间不断发生化学反应，进行元素交换，导致玻璃文物的化学成分比例发生显著变化，影响对其类型的正确判断。因此，本题首先利用问题一中建立的基于多元线性回归的化学成分预测模型，对表单 2 中的风化点数据进行预测，从而还原出所有玻璃文物检测点风化前各化学成分含量，以此消除风化对判断玻璃类型产生的影响。

6.1.2 玻璃样品分类模型的建立

XGBoost 是一种源于梯度提升算法（GDBT），以分类回归树为基础的机器学习算法。其在预测准确率、泛化性能、不易过拟合和可扩展性方面明显优于随机森林、决策树、朴素贝叶斯等算法且自由度优势更高。其次，XGBoost 算法既适用于化学成分含量这一连续变量，又适用于纹饰、颜色、表面风化等分类变量[5]，适合于玻璃分类的应用场景。

XGBoost 的基本原理如下：首先构建单棵树预测训练集，在此基础上添加叶子节点即其见点数平方和的正则项，以此进行多线程并行计算，并可有效防止过拟合。接着通过迭代增加新树来拟合上次预测的残差，将多个分类性能较低的弱学习器集成为一个准确率较高的强学习器，提高预测模型准确性。其目标函数为：

$$L=\sum_{i=1}^{n} l(y_i,\hat{y}_i)+\sum_{i=1}^{t}\Omega(f_i) \tag{13}$$

其中，i 为样本索引号，n 为导入第 t 棵树的数据总量，t 表示建立的所有 CART 树。公式右半的第一部分为经验损失项，表示训练数据预测值与真实值之间的损失；第二部分为正则项，表示全部 t 棵树的复杂度之和。

6.1.3 模型的求解与分析

在建模过程中，模型存在重要参数，如树的最大深度、样本采样值以及学习速率等。本文采用网格搜索法对参数进行寻优，通过遍历不同的参数组合，规避局部最优，结合指数网格划分，达到模型最优化的效果。其基本原理[4]是：首先选取对模型影响最大的参数进行调优，通过给定取值区间，按照顺序进行搜索，最优值找到后再对下一个影响较大的参数进行调优，以此类推，直到所有参数调优完毕。确定模型最优参数配置后，选取与数学统计模型中相同的训练样本构建 XGBoost 模型。

本题寻优后修改的参数为：tree_depth（每棵树的最大深度）、subsample（样本采样值）、learning_rate（每一次提升的学习效率）。最终选择了迭代次数为 5，树的最大深度为 6 ，样本采样值为 0.2，提升学习效率为 0.3 的模型参数。

本题还引入 LIME（Local Interpretable Model-Agnostic Explanations）解释 XGBoost 分类模型，进一步解释高钾玻璃和铅钡玻璃的划分依据。其步骤如下：

***Step* 1**：预测样本附近随机采样：对于连续性特征，在预测样本点附近使用标准正态分布产生指定个数；对于类别性特征，根据训练集分布采样，当新生成样本的类别特征与预测样本相同时，该类别特征取值为 1，否则取值为 0。

***Step* 2**：对新生成的样本打标签：将随机采样得到的新样本放入已训练好的复杂模型中训练，得到相应的预测结果。

***Step* 3**：计算权重：新样本与预测点欧式距离越短，则效果越好，应赋予更高的权重。

***Step* 4**：筛选用于解释的特征，拟合新的线性简单模型：

设期望用于解释的特征为 p 个，筛选变量后构建简单的线性回归模型。

分别使用 LIME 对高钾玻璃、铅钡的化学成分作全局解释，得到其特征变量对预测结果为高钾玻璃的权重分布图，见图 7、图 8，其中值即为权重系数，大于 0 代表该特征支持模型判断该样本为高钾玻璃，小于 0 则支持判为铅钡玻璃。

以 PbO 为例，图中 PbO <1.52 时正权重系数较大，即支持样本判为高钾玻璃；23.02<PbO<40.75 时负权重系数较大，即不支持判为高钾玻璃，其余特征权重系数相对较小，作用有限。以判断 PbO 的含量为例，表明使用该模型进行玻璃样本分类是一种有效的方法。

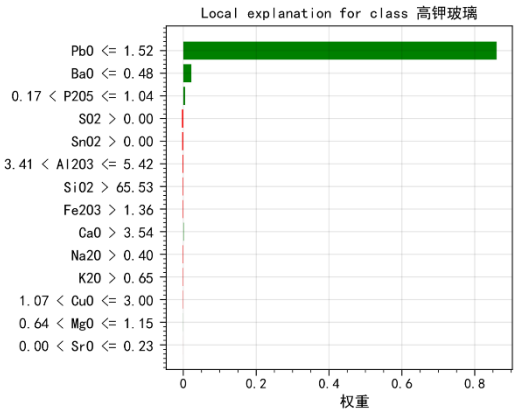

图 8 高钾玻璃权重分布图

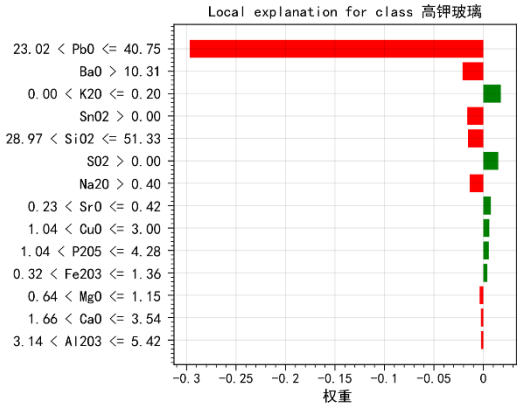

图 7 铅钡玻璃权重分布图

6.1.4 模型的精度评价

本题的玻璃样品分类模型主要在基分类器 CART 决策树和排序特征重要性两个方面来体现其对变量的解释程度，在对测试集的预测准确性来体现模型的准确度。选择 ROC 曲线进行分类结果准确率的评价。

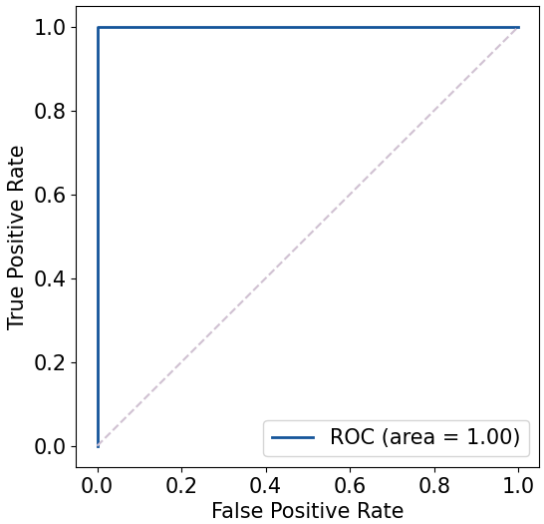

图 9 玻璃样品分类结果的 ROC 曲线

由图 9 可知，图中 FPR=0, TPR=1，AUC=1。因此可以得出结论：该模型是完美分类器，划分结果在多分类变量(纹饰、颜色、表面风化) 和连续值变量(化学成分含量) 的预测中，其准确率皆达到 100%。

6.2 基于变差系数和相对变率的玻璃亚类划分模型

6.2.1 基于变差系数和相对变率的玻璃亚类划分模型

首先通过求取不同化学成分的变差系数，消除平均数不同对各化学成分变异程度比较的影响，衡量各化学成分含量的变异程度[7]。变差系数是描述数据离散程度的一个归一化量度，其定义为标准差与平均值之比，具体公式如下：

$$V_p = \frac{S_x}{\bar{x}} \times 100\% = \frac{1}{\bar{x}} \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x}^2)} \times 100\% \tag{14}$$

其中，S_x 为标准差，$\bar{x}$ 为平均值。

变差系数越大，说明该化学成分在不同样本中变化程度越大，说明该化学成分包含的信息更多，更具代表性。各化学成分的变差系数如表 6 所示：

表 6 各化学成分的变差系数表

化学成分	SiO_2	Na_2O	K_2O	CaO	MgO	Al_2O_3	Fe_2O_3	CuO	PbO	BaO	P_2O_5
变差系数	0.16	0.23	0.63	0.74	0.76	0.61	1.02	0.59	0.89	1.40	1.13

本题选出变差系数>0.8 的化学成分，作为主化学成分进行亚类划分。

接着采用相对变率说明化学成分含量变化的大小。相对变率表示某两个变量间的相对变化速率。相对变率数值上等于绝对变率与平均值之比，公式为：

$$V_r = \frac{\sum_{i=1}^{n} \left| x_i - \bar{x} \right|}{n\bar{x}} \times 100\% \tag{15}$$

相对变率越大，说明在选定的主化学成分下，其余化学成分的变动越大，说明该化学成分在主化学成分中含的信息更多，更具代表性。本题选出相对变率>0.4 的次化学成分视为主化学成分下的一类。

最终的亚类划分结果[1]如图 10 所示。

由图可得，高钾玻璃的亚类划分结果为：K-P_2O_5 类,K-Fe_2O_3 类,K-PbO 类,其他类；铅钡玻璃的亚类划分结果为：Pb-P_2O_5 类,Pb-Fe_2O_3 类,K-CuO 类,Pb-K_2O 类,其他类。

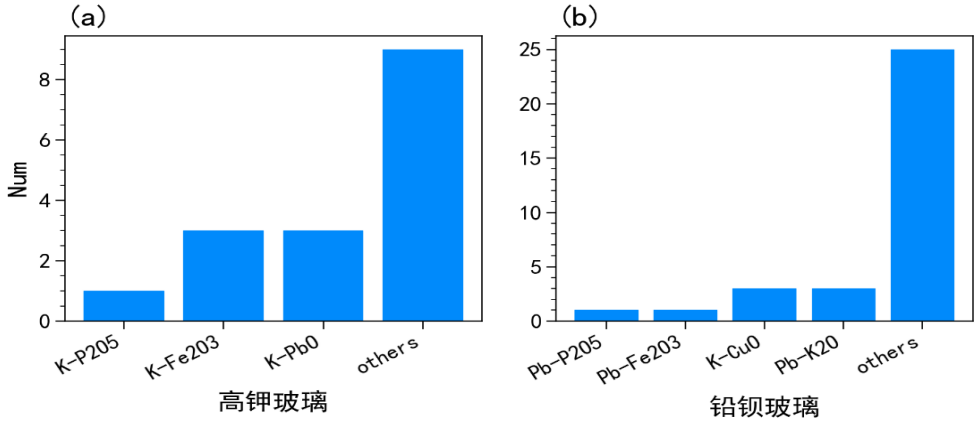

图 10 两类玻璃亚类划分结果柱状图

各玻璃样品的亚类划分结果如表 7 所示：

表 7 各玻璃样品的亚类划分结果

亚类划分类别	样本编号
K-P_2O_5 类	6
K-Fe_2O_3 类	5，13，16
K-PbO 类	3，14，21
其他类（高钾）	1，4，7，9，10，12，18，22，17
Pb-P_2O_5 类	49
Pb-Fe_2O_3 类	31
K-CuO 类	8，24，26
Pb-K_2O 类	2，20，30
其他类（铅钡）	余下所有样本

6.2.2 划分结果的敏感性、合理性分析

本题采用局部敏感性分析来评价划分结果敏感性。以不同方式填充各化学成分含量的空值，并和原值进行对比，通过比较模型的变化程度判断划分结果的敏感性。

如图 11 所示，改变系统参数后，模型的变化程度不明显，总体趋势和原系统相近，说明该基于变差系数的玻璃亚类划分结果敏感性较差，即模型稳定性强。

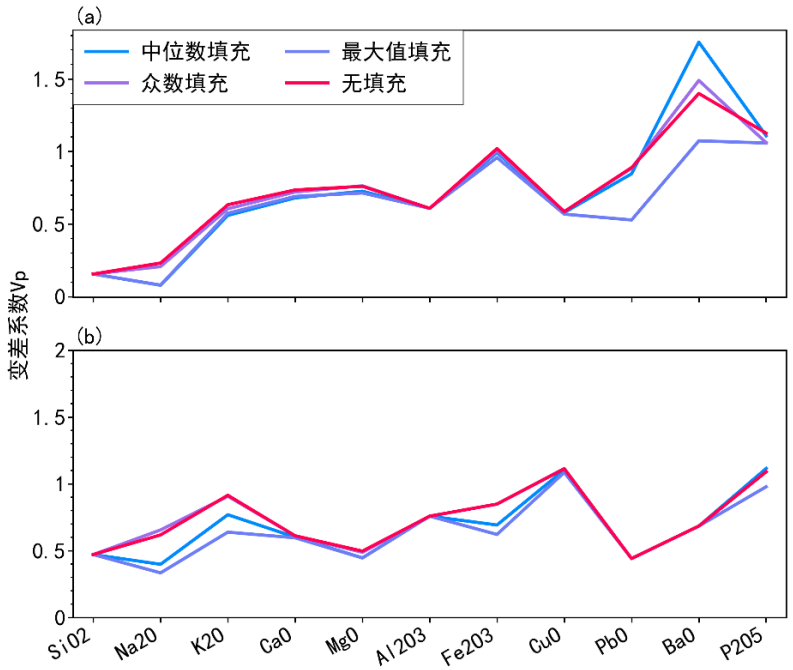

图 11 变差系数敏感性分析图

本题通过对变差系数和相对比率的统计分析，建立玻璃亚类划分模型。运用相关统计学规律，先根据变差系数选出合适的主化学成分，后根据相对比率选出主化学成分确定下的代表性强的次化学成分，符合题目要求。由此可见，本题所建基于变差系数和相对变率的玻璃亚类划分模型合理有效。

七、问题三分析与处理

此题是对问题二中建立的两个分类模型进行的实际应用。将预测风化前的未知类别玻璃文物的化学成分分别代入基于 XGBoost 的玻璃样品分类模型和基于变差系数的玻璃亚类划分模型，即可鉴别未知类别文物类型。再通过 OAT 法进行敏感性分析，以检验模型可行性。

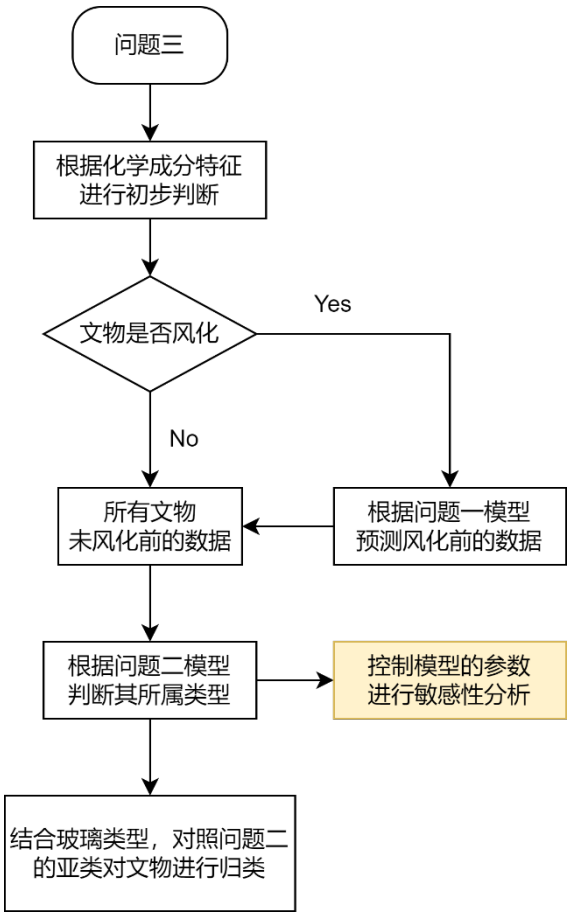

图 12 问题三流程图

7.1 基于玻璃样品分类模型的类型鉴别

7.1.1 未知类别文物的类别鉴别步骤

(1).数据预处理：由于古代玻璃在风化过程中，样品内部与外部环境间不断发生化学反应，进行元素交换，导致玻璃文物的化学成分比例发生显著变化，影响对其类型的正确判断。因此，本题首先利用问题一中建立的基于多元线性回归的化学成分预测模型，对表单 2 中的风化点数据进行预测，从而还原出所有玻璃文物检测点风化前各化学成分含量，以此消除风化对判断玻璃类型产生的影响。

(2).基于变差系数和相对变率的玻璃亚类划分模型：利用该模型，选择相同的 XGBoost 分类器，将未知类别玻璃文物检测点风化前各化学成分含量输入模型中，得到高钾/铅钡玻璃的分类结果。该模型拟合优度同样为 100%

(3).基于变差系数的玻璃亚类划分模型：将各化学成分代入其中，计算对应的变差系数确定出主化学成分后，再计算对应的相对变率，得出更具有代表性的次化学成分，构建亚类划分类别。

7.1.2 未知类别文物的类别鉴别结果

通过玻璃样品分类模型计算后得到的分类结果为：铅钡玻璃：A2,A3,A4,A5,A8;高钾玻璃：A1,A6,A7。通过玻璃亚类划分模型计算后，得到的总分类结果如表 8 所示：

表 8 未知类型玻璃样品的亚类划分结果

亚类划分类别	样本编号
K-Fe_2O_3 类	A1
其他类（高钾）	A6,A7
Pb-P_2O_5 类	A2
Pb-Fe_2O_3 类	A3,A4
其他类（铅钡）	A5,A8

7.2 分类结果敏感性分析

本题采用 OAT 法分析来评价划分结果敏感性。OAT 通常包括：移动一个输入变量，保持其他变量的基线值(nominal value)，然后将变量返回到其标称值，然后以相同的方式对每个其他输入进行重复。敏感性通过观察输出的变化来判断，若输出结果的变化程度不明显，总体趋势和原系统相近，说明分类结果敏感性较差，即模型稳定性强。

本题选取 *n_estimators*,*learning_rate*,*subsamples* 三个参数作为每次移动的输入变量。每次只更改一个变量值，各更改 6 次，所得三种修改参数敏感性分析图如图 14 所示：

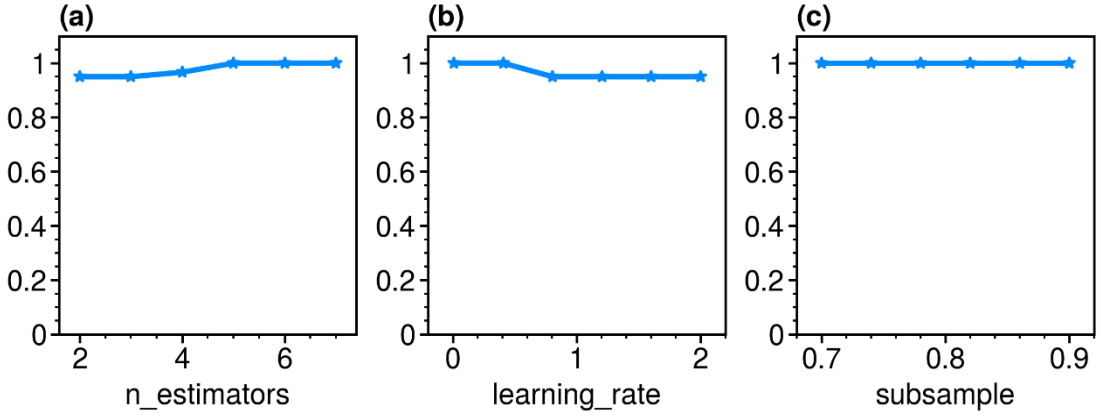

图 13 三种参数修改后敏感性分析图

由图可知：*n_estimators* 和 *learning_rate* 在修改前后稍有变化，但变化程度低；*subsample*s 修改前后一直稳定。因此，结果表明：模型总体趋势和原系统相近，分类结果敏感性较差，即模型稳定性强。

八、问题四分析与处理

本题首先对两种玻璃的化学成分进行相关性分析，计算出相关性系数矩阵后，根据所得相关性和显著性综合分析，判断出各化学成分间的关联关系。以通过相关系数矩阵得到的不同类别的各化学成分的相关系数变量为基础，采用 Wilcoxon 秩和检验判断相同化学成分下，两组相关系数向量是否存在差异。

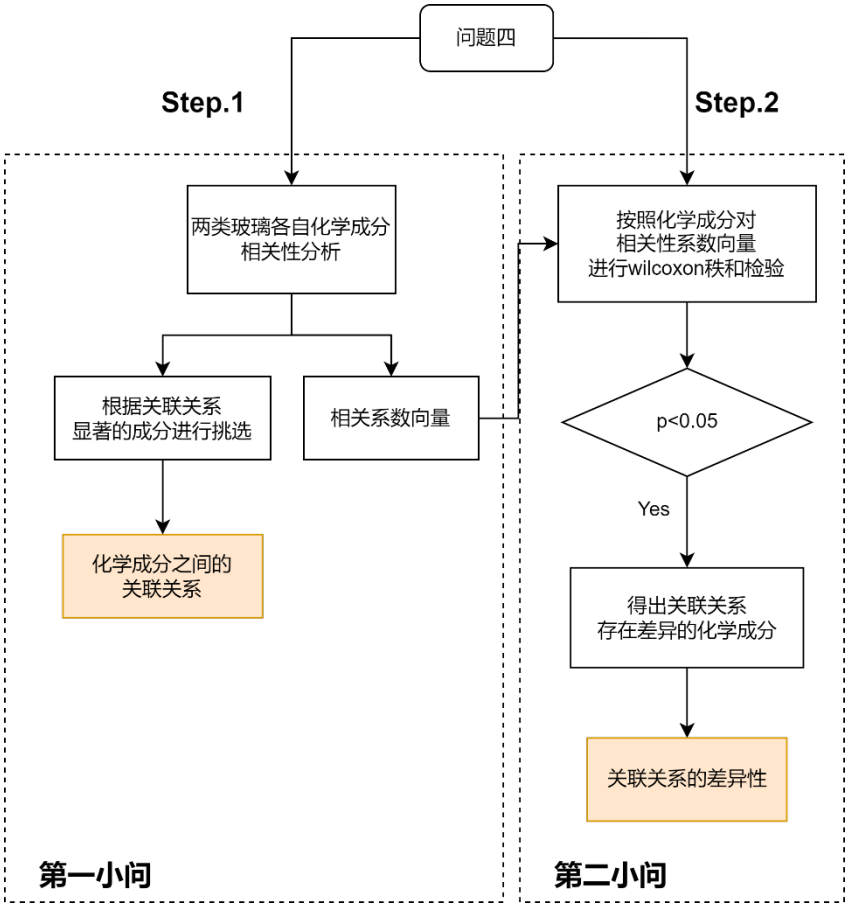

图 14 问题四流程图

8.1 各化学成分相关性分析

本题首先利用问题一中建立的基于多元线性回归的化学成分预测模型，对表单 2 中的风化点数据进行预测，从而还原出所有玻璃文物检测点风化前各化学成分含量，以此消除风化对判断玻璃类型产生的影响。对得到的检测点风化前各化学成分含量进行分析，发现高钾玻璃和铅钡玻璃的氧化锡（SnO_2）、二氧化硫（SO_2）两种化学成分存在相当一部分缺失值。故将这两种化学成分删除，得到高钾和铅钡两种玻璃的 12 中化学成分含量。

8.1.1 计算相关性系数

相关系数用于衡量两个变量的总体误差，相当于消除量纲的表示变量间相关性的一个矩阵。当相关性系数介于-1~0 之间时，表明变量之间存在负相关关系；当相关性系数

介于 0~1 之间时，表明变量之间存在正相关关系；当相关性系数为 0 时，二者之间不存在相关性。相关性系数的数值规律为：相关性系数越接近 1，表明变量之间的相关性越强，当相关系数越接近 0，表明变量之间的相关性越弱。

相关系数公式如下：

$$\mathrm{cov}(X,Y)=\frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{n-1} \tag{16}$$

依此构建相关性系数矩阵 R。设 $(x_1,x_2,...,x_n)$ 是一个 n 维随机变量，任意 x_i 与 x_j 的相关系数 $a_{ij}(i,j=1,2,...,n)$ 存在，则以 a_{ij} 为元素的 n 阶矩阵称为该维随机向量的相关矩阵，记作 R。将处理好的数据代入，本题共有 12 种变量，所得本题的相关性系数矩阵 R 为：

$$R=\begin{bmatrix} a_{11} & \cdots & a_{112} \\ \vdots & \ddots & \vdots \\ a_{121} & \cdots & a_{1212} \end{bmatrix} \tag{17}$$

其中，对角线上的值，如 $a_{11},\ldots,a_{1212}$ 皆为 1。

8.1.2 相关性的显著性检验

本题得到的两类玻璃的 12 种化学成分间的相关性与显著性均在图 15,图 16 中体现。

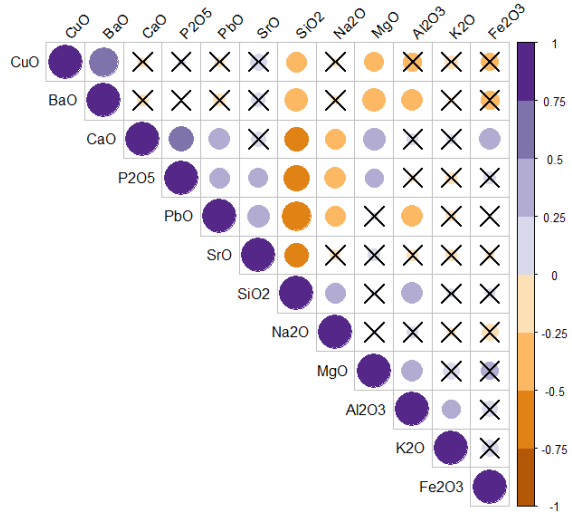

图 16 高钾玻璃化学成分相关性和显著性分析图

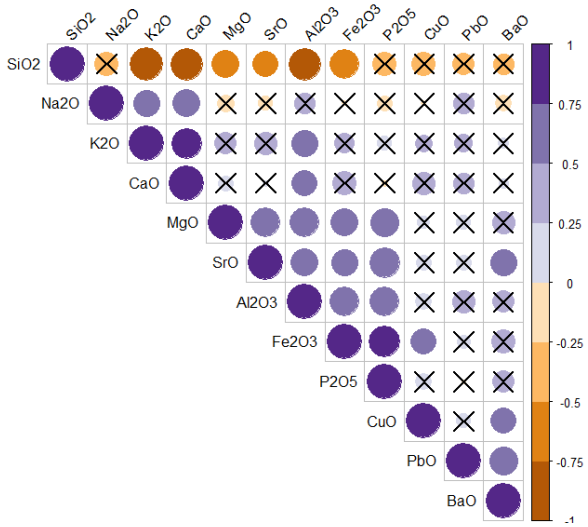

图 15 铅钡玻璃化学成分相关性和显著性分析图

其中，X 标表示两个化学成分间关系不显著。圆形图案颜色越深，表示变量间相关性越高；图案面积越大，表示变量间相关性的显著性越高。

8.1.3 各化学成分间的关联关系

结合上图各化学成分的相关性和显著性分析，得出两种类别下各化学成分间的关联关系如表 9、10 所示：

表 9 高钾玻璃各化学成分间的关联关系

	正相关显著	负相关显著
SiO_2	无	K_2O, CaO, MgO, SrO, Al_2O_3 ,Fe_2O_3
Na_2O	K_2O, CaO	无
K_2O	Na_2O, CaO, Al_2O_3	SiO_2
CaO	Na_2O, K_2O, Al_2O_3	SiO2
MgO	SrO, Al_2O_3 ,Fe_2O_3 ,P_2O_5	SiO_2
SrO	MgO ,Al_2O_3 ,Fe_2O_3, P_2O_5 ,BaO	SiO_2
Al_2O_3	K_2O, CaO, MgO ,SrO ,Al_2O_3 , Fe_2O_3 P_2O_5	SiO_2
Fe_2O_3	MgO，SrO，Al_2O_3 P_2O_5 ,CuO	SiO_2
P_2O_5	MgO, SrO ,Al_2O_3, Fe_2O_3	无
CuO	Fe_2O_3 ,BaO, Al_2O_3, MgO	无
PbO	BaO,	无
BaO	CuO, PbO SrO	无

通过分析发现，12 个化学成分除了 SiO_2 与 K_2O,CaO,MgO,SrO,Al_2O_3,Fe_2O_3 存在显著正相关关系，其余都与其相应的变量显著负相关。

表 10 铅钡玻璃各化学成分间的关联关系

	正相关显著	负相关显著
CuO	BaO	SiO_2,MgO
BaO	CuO	SiO_2,MgO,Al_2O_3
CaO	P_2O_5,PbO,MgO,Fe_2O_3	SiO_2,Na_2O
P_2O_5	PbO,SrO,MgO,CaO,	SiO_2,Na_2O
PbO	SrO,P_2O_5,	SiO_2,Na_2O,Al_2O_3
SrO	P_2O_5,PbO,	SiO_2
SiO_2	Na_2O,Al_2O_3	CuO,BaO,CaO,P_2O_5,PbO,SrO
Na_2O	SiO_2	CaO,P_2O_5,PbO,
MgO	CaO,P_2O_5	Al_2O_3,CuO,BaO
Al_2O_3	SiO_2	K_2O,BaO,PbO,MgO
K_2O	无	Al_2O_3
Fe_2O_3	CaO	无

通过分析发现，12 个化学成分间对应的正负相关显著性相对均匀。

8.2 Wilcoxon 秩和检验分析关联关系差异性

Wilcoxon 秩和检验基于样本数据的秩和，其将两样本看为一个整体，从小到大升序排列并排等级，即同一编秩。若原假设中两个独立样本来自相同的总体，则秩将均匀发布在两个样本内；若备选假设两个独立样本来自不相同的总体，则其中一个样本将会有更多的秩值，这样即可得到一个较小的秩，而另一个样本会有更多更大的秩值，因此就会得到一个较大的秩和。

作出假设：原假设 H_0: 量纲独立样本没有直接差异；备择假设 H_1：量纲独立样本有直接差异。

检验步骤如下：

Step 1：将检验的两组独立样本进行混合，根据大小升序排列并排等级，即为秩；

Step 2：分别求出两个样本的等级和 R_1、R_2；

Step 3：计算 U 检验统计量 U_1、U_2，公式如下:

$$\begin{cases} U_1 = R_1 - \dfrac{n_1(n_1+1)}{2} \\ U_2 = R_2 - \dfrac{n_2(n_2+1)}{2} \end{cases} \tag{18}$$

其中：n_1、n_2 分别为两个样本的大小，R_1、R_2 分别为两个样本的等级和。

使用 U_1、U_2 中的最小值用于与显著性检验 U_α 相比较，若 $U_{min}<U_\alpha$，则拒绝 H_0，接受 H_1，表明两样本之间存在差异。

8.2.2 Wilcoxon 秩和检验求解

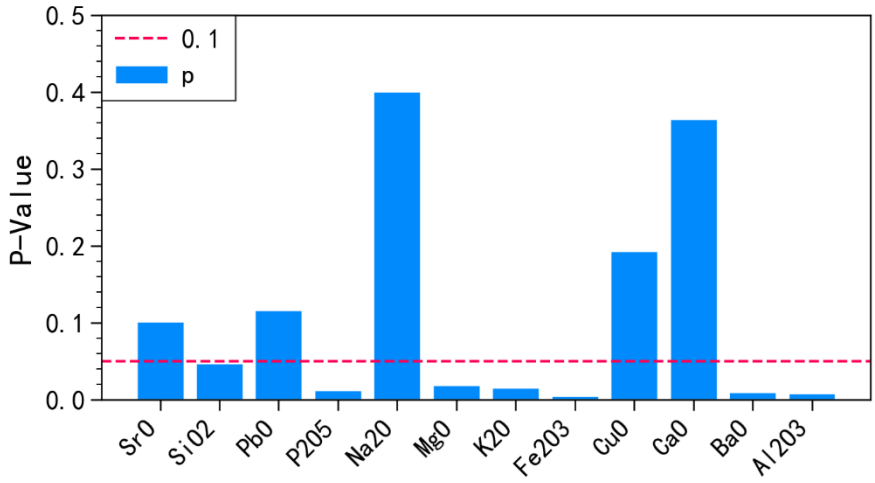

图 17 Wilcoxon 秩和检验下化学成分的 P 值

由结果可知，SiO_2、P_2O_5、MgO、K_2O、Fe_2O_3、BaO、Al_2O_3 的 P 值均小于 0.05，原假设不成立，即这些化学成分之间是有差异性的。

九、模型评价与改进

9.1 模型的优点

(1)结合专业知识，引入变差系数和相对变率进行玻璃样本亚类划分。以统计规律为基本原理进行分类，使结果更合理；

(2) 利用建立的化学成分预测模型，预测出风化点数据在未风化时各化学成分含量。将此数据带入分类模型中进行分类，消除了风化对判断玻璃类型产生的影响，提高模型分类准确性。

9.2 模型的缺点与改进

(1)本文建立的基于 XGBoost 的玻璃样品分类模型中，采用网格搜索法进行参数寻优。该方法适用于本文问题，但此方法下，若参数空间较大，模型会为更好的表现进行大量遍历，牺牲很多训练时间，降低效率。在这种情况下，可以采用随机搜索的方法进行参数寻优。随机搜索在提高效率的同时，可以得到更优解。

(2)本文第三问在鉴别未知类别玻璃样本的类别过程中，采用 OAT 法对鉴别结果进行敏感性分析。该方法一次只能改变一个输入参数，无法检测到多个输入耦合的情况。可以采用 Variance-based 法，将输出方差分解为可归属于输入变量和变量组合的部分，可同时考虑到多个参数组合改变的情况。

参考文献

[1] 李青会,黄教珍,李飞,干福熹.中国出土的一批战国古玻璃样品化学成分的检测[J].文物保护与考古科学,2006(02)

[2] 戴金辉,韩存.双因素方差分析方法的比较[J].统计与决策,2018,34(04)

[3] 王惠文,孟洁.多元线性回归的预测建模方法[J].北京航空航天大学学报,2007(04)

[4] 纪昌明,周婷,向腾飞,黄海涛.基于网格搜索和交叉验证的支持向量机在梯级水电系统隐随机调度中的应用[J].电力自动化设备,2014,34(03)

[5] 庞吉玉,张安兵,王贺封,侯毅凯,马杰.基于无人机多光谱影像和 XGBoost 模型的城市河流水质参数反演[J/OL].中国农村水利水电:1-17[2022-09-18].

[6] 温忠麟,侯杰泰,马什赫伯特.结构方程模型检验:拟合指数与卡方准则[J].心理学报,2004(02)

[7] 施能.气象科研与预报中的多元分析方法[M],第二版，气象出版社，2002.2，2-3

附录

附录一 支撑材料文件列表

1. 图片
2. 数据
3. 代码

附录二 文中涉及的程序

问题一：

```
# -*- coding: utf-8 -*-
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import proplot as pplt
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import joblib
import shap
import lime
import lime.lime_tabular
from sklearn.model_selection import GridSearchCV, cross_validate
from sklearn.model_selection import KFold, cross_val_score
from sklearn import    metrics
mpl.rcParams.update(
    {
    'font.family': 'sans-serif',                    #设置字体样式
    'font.size':16,
    'font.sans-serif': ['SimHei'],
    'axes.labelsize': 16,
    'xtick.labelsize':16,
    'ytick.labelsize':16
    }
    )
```

```
plt.rcParams['axes.unicode_minus'] = False     #用来正常显示负号

fig, ax = pplt.subplots([[1,2],[3,3]], refheight= 2,refwidth= 3, dpi= 300,)

# 数据读取
data = pd.read_excel(r'../数据/附件.xlsx', sheet_name= '表单 2').iloc[:, 3:]
# data = data.fillna(0)
# 4 类划分
data_K = data[data['类型'] == '高钾']
data_Pb = data[data['类型'] == '铅钡']
data_K_cloud = data_K[data_K['表面风化'] == '风化'].iloc[:, 2:-1]
data_Pb_cloud = data_Pb[data_Pb['表面风化'] == '风化'].iloc[:, 2:-1]
data_K_nocloud = data_K[data_K['表面风化'] == '无风化'].iloc[:, 2:-1]
data_Pb_nocloud = data_Pb[data_Pb['表面风化'] == '无风化'].iloc[:, 2:-1]
# 各成分缺失值数量统计
mean_dis1 = pd.DataFrame(data= {'K cloud':data_K_cloud.isnull().sum()/len(data_K_cloud)*100,
                                'K
nocloud':data_K_nocloud.isnull().sum()/len(data_K_nocloud)*100,
                                'Pb
cloud':data_Pb_cloud.isnull().sum()/len(data_Pb_cloud)*100,
                                'Pb
nocloud':data_Pb_nocloud.isnull().sum()/len(data_Pb_nocloud)*100},
                               )
# 绘图
fig, ax = pplt.subplots(refheight= 2, refwidth= 6,dpi= 300)
ax.bar(mean_dis1, cycle= ['#008afb','#ab67e4','#d24fc9','#ff0053'], edgecolor='white')
ax.legend(loc='best',ncols=2,    prop={'size': 6})
ax.format(ylim=(0, 100), ytickminor=False, ylocator=20, yformatter='{x:.1f}',ylabel= 'Num(%)',
          ylabelsize= 10, yticklabelsize= 8,xticklabelsize= 8, grid= False)
# plt.savefig('../图片/Q1_分类&风化 4 类_化学成分缺失值占比.png',dpi=300)

# 箱线图
for i in np.arange(14):
    data_chem_temp = pd.concat([data_K_cloud.iloc[:, i],data_K_nocloud.iloc[:, i],\
                                data_Pb_cloud.iloc[:, i],data_Pb_nocloud.iloc[:, i]], axis=1)
```

```
data_chem_temp.columns = ['K cloud','K nocloud', 'Pb cloud', 'Pb nocloud']
# 绘图
fig, ax = pplt.subplots(
                        refheight= 3,
                        refwidth= 4,   #对单个图而言
                        )
ax.box(data_chem_temp, cycle= 'Accent')
ax.format(
        ylabelsize= 16,
        yticklabelsize= 14,
        xticklabelsize= 14,
        grid= False
       )
# plt.savefig(r'../图片/Q1_2_成分含量统计 box_{}.svg'.format(data_K_cloud.iloc[:, i].name), dpi=300)
```

问题二：

```
# -*- coding: utf-8 -*-
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import proplot as pplt
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import joblib
import shap
import lime
import lime.lime_tabular
from sklearn.model_selection import GridSearchCV, cross_validate
from sklearn.model_selection import KFold, cross_val_score
from sklearn import    metrics
mpl.rcParams.update(
    {
    'font.family': 'sans-serif',                    #设置字体样式
```

```
        'font.size':16,
        'font.sans-serif': ['SimHei'],
        'axes.labelsize': 16,
        'xtick.labelsize':16,
        'ytick.labelsize':16
        }
        )
    plt.rcParams['axes.unicode_minus'] = False     #用来正常显示负号

    # 数据读取
    data = pd.read_excel(r'../数据/附件.xlsx', sheet_name= '表单 1').iloc[:, 1:]
    # 统计分析表 1 特征
    data_K = data[data['类型'] == '高钾'].drop(columns= '类型')
    data_Pb = data[data['类型'] == '铅钡'].drop(columns= '类型')
    #高钾玻璃
    K_dec = data_K.groupby(by= '纹饰').size()
    K_color = data_K.groupby(by= '颜色').size()
    K_cloud = data_K.groupby(by= '表面风化').size()
    #铅钡玻璃
    Pb_dec = data_Pb.groupby(by= '纹饰').size()
    Pb_color = data_Pb.groupby(by= '颜色').size()
    Pb_cloud = data_Pb.groupby(by= '表面风化').size()
    #合并
    dec = pd.DataFrame(data= {'K':K_dec/len(data_K), 'Pb':Pb_dec/len(data_Pb)}).fillna(0).reset_index()
#重置索引
    color = pd.DataFrame(data= {'K':K_color/len(data_K),
'Pb':Pb_color/len(data_Pb)}).fillna(0).reset_index()
    cloud = pd.DataFrame(data= {'K':K_cloud/len(data_K),
'Pb':Pb_cloud/len(data_Pb)}).fillna(0).reset_index()
    dec.index = dec.iloc[:, 0].values
    color.index = color.iloc[:, 0].values
    cloud.index = cloud.iloc[:, 0].values
    dec.drop(columns='纹饰', inplace= True)
    color.drop(columns='颜色', inplace= True)
    cloud.drop(columns='表面风化', inplace= True)
```

```
    # 绘图
    fig, ax = pplt.subplots([[1,2],[3,3]],refheight= 2,refwidth= 3, dpi= 300)
    ax[0].bar(dec*100, cycle= ['#008afb','#ab67e4'], edgecolor='white')
    ax[1].bar(cloud*100,cycle= ['#008afb','#ab67e4'], edgecolor='white')
    ax[2].bar(color*100,cycle= ['#008afb','#ab67e4'], edgecolor='white')
    # 总体设置
    ax.legend(loc= 'best')
    ax.format(ylim=(0, 100), ytickminor=False, ylocator=20, yformatter='{x:.1f}',
                ylabel= 'Num(%)',ylabelsize= 16,yticklabelsize= 12,xticklabelsize= 12,
                abc= '(a)', abcsize= 14,grid= False, abcweight ='light')
    # plt.savefig('../图片/Q2_1_类别与其余文本特征统计结果_表 1 各类占比.png',dpi= 300)

    # 数据读取
    data = pd.read_excel(r'../数据/附件.xlsx', sheet_name= '表单 2').iloc[:, 3:-1].fillna(0)
    data.drop(columns= '表面风化', inplace= True)
    data_target = data.iloc[:, 0]
    data_train = data.iloc[:, 1:]
    # data_train = data.iloc[:, [1,4,5,6,8,11]]
    x_train, x_test, y_train, y_test = train_test_split(data_train, data_target, test_size= 0.2, random_state=
100)
    # 分类
    clf = XGBClassifier()
    clf.fit(x_train, y_train)
    y_pre = clf.predict(x_test)
    y_pre_train = clf.predict(x_train)
    print('训练集准确率：', accuracy_score(y_train, y_pre_train))
    print('测试集准确率：', accuracy_score(y_test, y_pre))
    # joblib.dump(filename=    'XGBoost.model', value= clf) #保存模型

    # lime
    explainer = lime.lime_tabular.LimeTabularExplainer(np.array(data_train),
                                                       feature_names = data_train.columns,
                                                       class_names=['铅钡玻璃','高钾玻璃'],
                                                   #       verbose= True,
```

```
                                                    mode= 'classification')
exp = explainer.explain_instance(data_row= data[data['类型'] == '高钾'].iloc[:, 1:].mean(skipna= True),
\
                                                predict_fn= clf.predict_proba, num_features=14)
fig = exp.as_pyplot_figure()
# plt.savefig(r'../图片/Q2_1_两类玻璃分类规律_表 2_LIME 权重.svg',dpi=300)
# exp.show_in_notebook(show_table=True)

# 规定变差系数 Vp>0.8 即可根据该变量划分亚类, 相对变率 Vr>4.0 即可视为一类
# 读取数据-无风化
data = pd.read_excel(r'../数据/表 2_update.xlsx', index_col= '文物采样点')
data = data[data['类型']=='高钾'].iloc[:, 4:]
# 变差系数与相对变率计算
Vp = data.std(skipna= True)/data.mean(skipna= True) #14 个化学成分的变差系数
chem_vars = Vp[Vp>=0.8].index                    #划分亚类的根据变量
Va = np.abs(data-data.mean(skipna= True))/(len(data)-data.isna().sum()) #14 个化学成分的绝对变率
Vr = Va/data.mean(skipna= True) * 100                    #各个数据的相对变率(%)
print('lenth of data:',len(data))

# 循环遍历分类的变量--查找显著不同的数据的位置
pos = pd.DataFrame()   #用于合并的空 df
for i in chem_vars:
    Vr_temp = Vr[i]
    pos_temp = pd.DataFrame(data= {i:Vr_temp[Vr_temp>= 4 ].index.values})
    pos = pd.concat([pos, pos_temp], axis= 1)
    pos_i = pos.loc[:, i].dropna()
    data_pos = data.loc[pos_i, i]
    print('化学成分{}可分类数据:\n'.format(i), data_pos.sort_values())
    print('--------------------------分隔符-------------------------\n')
    # data_pos.sort_values().to_excel(r'../数据/Q2_亚分类_高钾_{}.xlsx'.format(i))

# 高钾变差系数图
#   = plt.stem(values)
plt.figure(figsize=(10,6), dpi= 300)
(markers, stemlines, baseline) = plt.stem(Vp.index, Vp.values, basefmt= '#767bf5', linefmt='#008afb')
```

```
    plt.setp(markers, marker='D', markersize=10, markerfacecolor="#008afb", markeredgewidth=2,
markeredgecolor= 'orange')
    plt.ylabel('变差系数 Vp', fontsize= 18)
    plt.xticks(fontsize= 18)
    plt.yticks(fontsize= 18)
    # plt.savefig(r'../图片/Q2_2_高钾变差系数.png',dpi= 300)

    # 亚分类绘图
    type_num_K = pd.DataFrame(data= {'K-P2O5':1,'K-Fe2O3':3,\
                                     'K-PbO':3,'others':9}, index= ['num'])
    type_num_Pb = pd.DataFrame(data= {'Pb-P2O5':1,'Pb-Fe2O3':1,\
                                     'K-CuO':3,'Pb-K2O':3,'others':25}, index= ['num'])
    # 绘图
    fig, ax = pplt.subplots(refheight= 2, refwidth= 2.5, ncols= 2,dpi= 300,share= False)
    ax[0].bar(type_num_K.T, c='#008afb')
    ax[1].bar(type_num_Pb.T, c='#008afb')
    ax[0].format(xlabel= '高钾玻璃', xlabelsize= 12, ylabel= 'Num',ylabelsize= 12,)
    ax[1].format(xlabel= '铅钡玻璃', xlabelsize= 12)
    ax.format(yticklabelsize= 10,xticklabelsize= 10,grid= False,xrotation= 30,
              abc= '(a)', abcsize= 14)
    # plt.savefig('../图片/Q2_2_亚分类.png', dpi= 300)

    # 绘图
    fig, ax = pplt.subplots(refheight= 3, refwidth= 8,nrows= 2)
    path = ['高钾', '铅钡']
    for j in np.arange(0,2):
        for i in np.arange(0, 4):   #3 种填充方式
            data = pd.read_excel(r'../数据/表 2_update.xlsx', index_col= '文物采样点')
            data = data[data['类型']== path[j]]
            if i == 1:
                data.fillna(data.mode(), inplace= True)
                fillna_mode = data.std(skipna= True)/data.mean(skipna= True) #变率系数
            elif i == 2:
                data.fillna(data.mean(skipna= True), inplace= True)
                fillna_mean = data.std(skipna= True)/data.mean(skipna= True)
```

```
            elif i==3:
                data.fillna(data.median(skipna= True), inplace= True)
                fillna_median = data.std(skipna= True)/data.mean(skipna= True)
            else:
                nofill = data.std(skipna= True)/data.mean(skipna= True)

        # 折线图
        ax[j].plot(fillna_median, lw= 2.6,label= '中位数填充', c= '#008afb')
        ax[j].plot(fillna_mean, lw= 2.6,label='最大值填充', c= '#6c7df6')
        ax[j].plot(fillna_mode, lw= 2.6,label='众数填充', c= '#9d6eea')
        ax[j].plot(nofill, lw= 2.6,label= '无填充', c= '#ff0053')

ax[0].legend(loc='best',ncols= 2, prop={'size': 18})
ax[1].format(ylim= (0,2))
ax.format(ylabel= '变差系数 Vp', xlabelsize= 18,ylabelsize= 18,yticks = 0.5,
              xticklabelsize= 18,yticklabelsize= 18, xlim = (-0.3, 10.3),alpha= 0.8,
              linewidth = 1.2,xrotation= 30,grid= False,abc= '(a)',abcsize= 18,)
# plt.savefig(r'../图片/Q2_2_变差系数灵敏度分析_高钾铅钡未风化.png', dpi= 300)
```

问题三：

```
# -*- coding: utf-8 -*-
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split, GridSearchCV,StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, VotingClassifier
from sklearn.metrics import r2_score, accuracy_score, mean_squared_error, auc, roc_curve
from sklearn.svm import SVC
from sklearn.feature_selection import VarianceThreshold,SelectFromModel
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
import lightgbm as lgbm
from matplotlib import colors
```

```
import joblib
mpl.rcParams.update(
    {
    'font.family': 'sans-serif',                    #设置字体样式
    # 'font.size':16,
    'font.sans-serif': ['SimHei'],
    # 'axes.labelsize': 16,
    # 'xtick.labelsize':16,
    # 'ytick.labelsize':16
    }
    )
plt.rcParams['axes.unicode_minus'] = False     #用来正常显示负号

## 数据读取
data= pd.read_excel(r'../数据/表 2_update.xlsx')
data['类型'].loc[data['类型'] == '高钾']= 1
data['类型'].loc[data['类型'] == '铅钡']= 0
Y = data['类型'].astype('int64')
X = data.iloc[:, 5:]
x_train, x_test, y_train, y_test = train_test_split(X ,Y ,test_size= 0.2, random_state= 100)

def draw_h2d(ax,Y_pre,Y, title):    # h-2d
    train_r2 = abs(r2_score(Y, Y_pre) )
    RMSE = (mean_squared_error(Y,Y_pre)**0.5)
    c = ax.hist2d(Y, Y_pre, bins=5, cmap='YlGnBu')
    ax.plot(Y,Y,c='#a61b29',label='y=x')
    ax.set_title(title+'R=%.3f     RMSE=%.3f'     %     (train_r2**0.5,RMSE),loc='left',fontsize
=15,fontweight='heavy')
    ax.set_ylim(0, np.max(Y))
    ax.set_xlim(0, np.max(Y))
    ax.set_xlabel('True Value ')
    ax.set_ylabel('Predicted Value')
    ##实际倾向率
    variance = np.var(Y, ddof=1)              # 计算方差，doff 为贝塞尔（无偏估计）校正系
数
```

```
    covariance = np.cov(Y, Y_pre)[0][1]  # 计算协方差
    w = covariance / variance
    b = np.mean(Y_pre) - w * np.mean(Y)
    y_line = w * Y+ b
    if b<0:
        ax.plot(Y, y_line, c='#15559a',label='y={}x{}'.format(round(w,4),round(b,4)))
    else:
        ax.plot(Y, y_line, c='#15559a',label='y={}x+{}'.format(round(w,4),round(b,4)))
    ax.legend(fontsize = 15)
    return c

def draw_h2d_plus(Classifier, y_pre, y_pre_train):
    mpl.rcParams.update(mpl.rcParamsDefault)  #还原默认绘图风格
    mpl.rc('font', size=15, weight='normal')  #设置全局字体大小
    fig = plt.figure(figsize=(10,12))
    fig_ax1 = fig.add_axes([0.1, 0.6, 0.4, 0.3])
    fig_ax2 = fig.add_axes([0.6, 0.6, 0.4, 0.3])
    c = draw_h2d(fig_ax1, y_pre_train, y_train, Classifier + '_train:')  #随模型修改------
    c2= draw_h2d(fig_ax2, y_pre, y_test, Classifier + '_test:')
    cax = plt.axes([0.13,0.53 , 0.8, 0.02])
    cbar=plt.colorbar(c[3],cax=cax, orientation='horizontal')
    cbar.set_label("density")
    plt.show()

def calculate_auc(y_test, pred): #绘制 roc 曲线
    fpr, tpr, thersholds = roc_curve(y_test, pred)
    roc_auc = auc(fpr, tpr)
    plt.figure(figsize= (6, 6))
    plt.plot(fpr, tpr, color= '#15559a', label='ROC (area = {0:.2f})'.format(roc_auc), lw=2)
    plt.xlim([-0.05, 1.05])
    plt.ylim([-0.05, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.legend(loc="lower right")
    plt.plot([0, 1], [0, 1], color= '#d1c2d3', linestyle= '--')
```

```
    plt.show()

# 训练模型
clf = XGBClassifier()
clf.fit(x_train, y_train)
y_pre = clf.predict(x_test)
y_pre_train = clf.predict(x_train)
print('训练集准确率：', accuracy_score(y_train, y_pre_train))
print('测试集准确率：', accuracy_score(y_test, y_pre))

# 储存模型
joblib.dump(filename= 'XGBoost.model', value= clf) #保存模型
# 绘图-h2d
draw_h2d_plus(Classifier= 'XGBoost', y_pre= y_pre, y_pre_train= y_pre_train)
calculate_auc(y_test= y_test, pred= y_pre)

# sheet 检验与验证
# 数据读取
data= pd.read_excel(r'../数据/Q3_表 3_update.xlsx')
data['类型'].loc[data['类型'] == '高钾']= 1
data['类型'].loc[data['类型'] == '铅钡']= 0
y =   data['类型'].astype('int64')
x = data.iloc[:, 2:-1]
# 分类与检验
clf = joblib.load(filename= 'XGBoost.model')    #读取模型
y_pre = clf.predict(x)
print('准确率：', accuracy_score(y, y_pre))
# 绘图
calculate_auc(y_test= y, pred= y_pre)

# 调参
cv_params = {
        # 'n_estimators':range(2,8,1),
        # 'learning_rate':np.linspace(0.01,2,6),
        'subsample':np.linspace(0.7,0.9,6),
```

```
        }
    other_params = {'learning_rate': 0.1, 'n_estimators': 500, 'max_depth': 5, 'min_child_weight': 1, 'seed': 0,
                    'subsample': 0.8, 'colsample_bytree': 0.8, 'gamma': 0, 'reg_alpha': 0, 'reg_lambda': 1}
    model = XGBClassifier(**other_params)
    gsearch2 = GridSearchCV(estimator=model, param_grid=cv_params, scoring='accuracy', cv=5, verbose=1, n_jobs=4)
    gsearch2.fit(X,Y)
    testscore3 = gsearch2.cv_results_['mean_test_score']
    for i in testscore3:
        print('当前 socre:',i)

    # 灵敏性绘图
    import proplot as pplt
    fig, ax = pplt.subplots(
                                refheight= 2,
                                refwidth= 2,   #对单个图而言
                                ncols=3,
                                share= False,
                                )
    ax[0].plot(np.arange(2,8), testscore1, lw= 2.6, marker='*',c='#008afb')
    ax[1].plot(np.linspace(0.01, 2, 6), testscore2, lw= 2.6, marker='*',c='#008afb')
    ax[2].plot(np.linspace(0.7, 0.9, 6), testscore3, lw= 2.6, marker='*',c='#008afb')
    ax[0].format( xlim = (1.6, 7.4), ylim = (0, 1.1), xlabel= 'n_estimators')
    ax[1].format( xlim = (-0.2, 2.2), ylim = (0, 1.1),xlabel= 'learning_rate')
    ax[2].format( xlim = (0.68, 0.92), ylim = (0, 1.1),xlabel= 'subsample')
    ax.format(xlabelsize= 14, ylabelsize= 14,yticks = 0.2, xticklabelsize= 14, yticklabelsize= 14,
              alpha= 0.8,abc= '(a)',abcsize= 14,linewidth = 1.2, grid= False)
    plt.savefig(r'../图片/Q3_XGBoost_敏感性分析.png', dpi= 300)

    # 规定变差系数 Vp>0.8 即可根据该变量划分亚类，相对变率 Vr>4.0 即可视为一类
    # 读取数据-无风化
    data = pd.read_excel(r'../数据/Q3_表 3_update.xlsx', index_col= '文物编号').drop(columns= ['表面风化','类型'])
```

```
# 变差系数与相对变率计算
Vp = data.std(skipna= True)/data.mean(skipna= True) #14 个化学成分的变差系数
chem_vars = Vp[Vp>=0.8].index                    #划分亚类的根据变量
Va = np.abs(data-data.mean(skipna= True))/(len(data)-data.isna().sum()) #14 个化学成分的绝对变率
Vr = Va/data.mean(skipna= True) * 100                    #各个数据的相对变率(%)
print('lenth of data:',len(data))

# 循环遍历分类的变量--查找显著不同的数据的位置
pos = pd.DataFrame()  #用于合并的空 df
for i in chem_vars:
    Vr_temp = Vr[i]
    pos_temp = pd.DataFrame(data= {i:Vr_temp[Vr_temp>= 4 ].index.values})
    pos = pd.concat([pos, pos_temp], axis= 1)
    pos_i = pos.loc[:, i].dropna()
    data_pos = data.loc[pos_i, i]
    print('化学成分{}可分类数据:\n'.format(i), data_pos.sort_values())
    print('----------------------------分隔符--------------------------\n')
    # data_pos.sort_values().to_excel(r'../数据/Q3_亚分类_{}.xlsx'.format(i))

# 亚分类绘图
import proplot as pplt
type_num_K = pd.DataFrame(data= {'K-Fe2O3':1,'others':2}, index= ['num'])
type_num_Pb = pd.DataFrame(data= {'Pb-Fe2O3':2,\
                                  'Pb-P2O5':1,'others':2}, index= ['num'])
# 绘图
fig, ax = pplt.subplots(refheight= 2, refwidth= 2.5,ncols= 2,dpi= 300,)
ax[0].bar(type_num_K.T, c='#008afb')
ax[1].bar(type_num_Pb.T, c='#008afb')
ax[0].format( ylabel= 'Num',ylabelsize= 12,)
ax[0].format(title='高钾玻璃', titlesize= 12)
ax[1].format(title='铅钡玻璃', titlesize= 12)
ax.format(yticklabelsize= 10, xticklabelsize= 10,grid= False,
          ylocator= 1,abc= '(a)', abcsize= 12,abcweight= 'light')
plt.savefig('../图片/Q3_亚分类.png', dpi= 300)
```

问题四：

相关性系数及其显著性：

```
QB1111[is.na(QB1111)] <- 0
mydata <- GJ1111
mydata

pic01<-cor(mydata)
corrplot(pic01)
corrplot(pic01,col = COLOR01)
corrplot(pic01, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45,col=brewer.pal(n=8, name="PuOr"))

cor.mtest <- function(mat, ...) {
  mat <- as.matrix(mat)
  n <- ncol(mat)
  p.mat<- matrix(NA, n, n)
  diag(p.mat) <- 0
  for (i in 1:(n - 1)) {
    for (j in (i + 1):n) {
      tmp <- cor.test(mat[, i], mat[, j], ...)
      p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
    }
  }
  colnames(p.mat) <- rownames(p.mat) <- colnames(mat)
  p.mat
}

mydatap<- cor.mtest(mydata)

corrplot(pic01,type = "upper", order = "hclust",tl.col = "black", tl.srt = 45,col=brewer.pal(n=8,
name="PuOr"),
         p.mat=mydatap, sig.level = 0.05)
corrplot(pic01, type="upper", order="hclust", p.mat = mydatap, sig.level = 0.05)

corrplot(pic01,method="color", type="upper", order="hclust",
```

```
            p.mat = mydatap, sig.level = 0.01)
```

Wilcoxon 秩和检验：

```
wilcox_test <- wilcox.test(value~group, SrO, paired = FALSE, alternative = 'two.sided')
wilcox_test
wilcox_test$p.value
i=12
p_value[i] <- wilcox_test$p.value
W[i] <-wilcox_test$statistic

library(ggpubr)

ggpaired(SiO2, cond1 = "铅钡", cond2 = "高钾",
            fill = "condition", palette = "jco")

colnames(SiO2) <- c("铅钡","高钾")
```

全文折线图：

```
# -*- coding: utf-8 -*-
#coding: unicode_escape
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import proplot as pplt
import seaborn as sn
mpl.rcParams.update(
    {
    # 'font.family': 'sans-serif',                #设置字体样式
    # 'font.size':16,
    # 'font.sans-serif': ['SimHei'],
    # 'axes.labelsize': 16,
    # 'xtick.labelsize':16,
    # 'ytick.labelsize':16
    }
    )
plt.rcParams['axes.unicode_minus'] = False     #用来正常显示负号
```

```
# 绘图
mpl.rcParams.update(
    {'font.family': 'sans-serif',                #设置字体样式
    'font.size':16,
    'font.sans-serif': ['SimHei'],               #使中文正常显示
    'axes.labelsize': 10,})
plt.rcParams['axes.unicode_minus'] = False     #用来正常显示负号

data = pd.read_excel(r'../数据/铅钡.xlsx')
corr = data.corr()
plt.figure(figsize=(16,16))
sn.heatmap(corr, annot=True, vmin=-1, square=True, cbar_kws={'shrink': 0.85})
#    annot=True,square=True,vmin=-1,vmax=1, cbar=True, cbar_kws={'shrink': 0.5}  可适当修改sn.heatmap()中的参数

# 读取数据
data = pd.read_excel(r'../数据/Q4_P_Value.xlsx', index_col= '化学成分')
# 绘图
fig, ax = pplt.subplots(refheight= 2,refwidth= 4, dpi= 300,)
ax.bar(data, cycle= ['#008afb','#ab67e4'], edgecolor='white')
ax.axhline(0.05, c='#ff0053', lw= 1, ls= '--',label= '0.1')
ax.legend(loc= 'ul',ncols=1,prop={'size': 10})
ax.format(yformatter='{x:.1f}',ylabel= 'P-Value',xlabel= '化学成分',xlabelsize= 12,
          ylabelsize= 12,ylim= (0, 0.5),yticklabelsize= 10,xticklabelsize= 10,grid= False,
          xrotation= 45)
plt.savefig('../图片/Q4_P_Value.png',dpi= 300)
```