古代玻璃制品的成分分析与鉴别

摘要

古代玻璃制品长年受埋葬环境的影响，玻璃表面容易出现风化，导致化学成分比例发生变化。本文根据题目给出的一批古代玻璃制品的相关数据，分析玻璃文物“理化特征”与“表面风化”、“玻璃类型”之间的规律，以及这些“理化特征”本身之间的关系。

对于问题一，本文首先使用统计图表和**卡方检验**，对玻璃的物理特征与表面风化的关系分别进行定性分析和定量分析，发现玻璃类型与表面风化的相关性最为显著，而其他物理特征和表面风化的相关性较差；其次将样本分为风化组与无风化组，对同组内的各化学成分含量取平均值，对比分析高钾玻璃和铅钡玻璃在风化前后的化学成分含量变化规律；最后以同种玻璃风化前后的**平均变化率**作为预测依据，预测出风化检测点在风化前的化学成分含量，并对预测结果进行合理性检验，将**详细预测结果表格置于文末附录**。

对于问题二，首先，在分类规律方面，本文模拟考古工作者的玻璃类型判断逻辑，对物理特征使用统计方法分析，对化学性质使用**决策树和随机森林模型**分析，得出以 SiO_2，PbO，BaO 为主，K_2O 为辅的分类规律。其次，在亚类划分方面，本文设定**“大标准差大均值准则”**和**“定类成分回避准则”**作为化学成分指标选择的依据，建立**基于轮廓系数确定最优聚类簇数的 K-means 模型**，将高钾类和铅钡类都划分出 2 个亚类。在合理性分析中，分析出这 4 个亚类分别为低钙铝的高钾亚类、高钙铝的高钾亚类、高铝低磷的铅钡亚类和高磷低铝的铅钡亚类；在敏感性分析中，分析出高钾玻璃亚类对 SiO_2 敏感性高，对 CaO 有一定敏感性，而铅钡玻璃亚类对 SiO_2 敏感性高，对 P_2O_5 有一定敏感性。

对于问题三，本文首先综合使用 **spearman 相关系数**和**主成分分析法**，先后得到两组特征指标；其次，给这两组指标赋予合理权重，建立**基于 KNN 算法的预测分类模型**，预测出表单 3 中含有 PbO 的文物都是铅钡玻璃，而其他的都是高钾玻璃，并对预测结果进行合理性检验；最后，在敏感性分析中可以得出，分类结果对 SiO_2 敏感性较高，对 PbO 和 Al_2O_3 具有一定敏感性。

对于问题四，本文首先使用 **spearman 相关系数**作为衡量各化学成分之间关联关系的指标，求出相关系数矩阵并绘制热力图，据此分析各化学成分相互之间的相关性；最后再综合使用 **Wilcoxon 符号秩检验**，从宏观上进一步探究不同玻璃类别的各化学成分之间的化学成分的差异性。

本文综合使用了多种统计分析手段与多种机器学习模型，对结果合理性进行定性分析与定量分析，分析全面透彻。同时，本文在分析分类规律时使用随机森林模型以模拟考古工作者的判断逻辑，使模型与现实紧密结合；在亚类划分时中提出了两个准则，具有新颖性与创新性。

关键字： 随机森林　K-means 聚类算法　KNN 分类算法　spearman 相关系数

一、 问题重述

1.1 问题背景

在古代，我国与西方主要通过丝绸之路进行经济文化的交流。通过这些贸易往来，早期的玻璃得以从西方传入我国，成为我国古代玻璃工艺发展的起源。随着本土的玻璃制作工艺发展，我国古代玻璃的材料渐渐本土化，化学成分与西方玻璃形成差异。经过考古研究发现，中国出土的古代玻璃成分既有铅钡玻璃，又有钠钙玻璃，同时还出土了大量钾玻璃。[1] 其中，战国时期楚国地区以铅钡玻璃为主，而岭南地区则以钾玻璃为主。

由于古代玻璃埋在土中，容易和大气发生作用发生侵蚀，也称“风化”[2]，严重破坏了玻璃的外表，导致考古工作者在研究文物样品时困难重重。因此，如何根据文物表面的特征和检测出的化学成分，分析出各个特征之间的规律，从而进行对该文物的分类，就变成一个非常重要的问题。

1.2 问题要求

1. 对玻璃文物的类型、纹饰、颜色和表面风化与否的关系进行分析；结合玻璃类型，分析是否风化的玻璃的化学成分含量的统计规律，并根据风化点的数据预测出风化前的化学含量。
2. 依据附件数据分析两种类型玻璃的分类规律；在每个类别中，选择恰当的化学成分对其进行亚类划分，给出划分方案和结果，并分析结果的合理性和敏感性。
3. 分析附件表单 3 中未知类别的玻璃文物的化学成分，鉴别其所属类别，并分析分类结果的敏感性。
4. 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

2.1 问题一的分析

问题一第一小问要求我们分析玻璃文物的表面风化与其玻璃类型、纹饰和颜色的关系。首先，我们进行数据预处理，剔除化学含量比例累加总和不在有效数据规定区间的无效数据，结合表单 1 和表单 2 的数据，使用随机森林对缺失颜色进行补全；最后，我们分别将玻璃类型、纹饰和颜色作为特征指标，绘制出这些特征与玻璃表面风化关系的统计图表，分析定性关系，并综合使用卡方检验，分析定量关系。

第二小问要求我们结合玻璃类型，分析风化与否与化学成分含量的统计规律，并在此基础上根据风化点的检测数据进行其风化前化学含量的预测。我们按照采样点是否属于风化点将数据分为风化组和无风化组，计算风化前后各化学成分含量的平均值，找出统计规律，并据此建立模型，对那些风化后的样本进行风化前化学成分含量的预测。

2.2 问题二的分析

问题二第一小问要求我们找出高钾、铅钡两类玻璃文物的分类规律，考虑到分类信息是考古工作者根据化学成分和其他检测手段提供的，因此我们根据附件中化学成分与类型的数据，拟合出决策树与随机森林模型，从而模拟考古工作者的决策过程，再结合颜色、纹饰等辅助信息，得出最终的分类规律。

第二小问要求我们对高钾、铅钡两类玻璃文物分别进行亚类划分，并分析划分结果的合理性和敏感性，这首先需要选择合适的化学成分作为分类依据，基于样本数据的实际情况我们提出“大标准差大均值”和“定类成分回避”两个准则来进行化学成分选择，再使用 K-means 模型进行聚类，通过轮廓系数选择最合适的聚类簇数，从而得出亚类划分结果，再结合亚类的风化情况与化学成分分析亚类划分的合理性，并通过观察亚类化学成分波动对划分结果的影响，探究亚类划分的敏感性。

2.3 问题三的分析

问题三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，鉴别它们的所属类型，实质上就是要我们采用合理的方式，提取出能正确区分玻璃类型的化学成分作为区分指标，对未知类型的文物进行预测，并分析预测效果的敏感性。

我们首先将 14 个化学成分作为指标，计算各化学成分与玻璃类型的 spearman 相关系数，提取出相关性较高的化学成分，并对剩下的化学成分使用主成分分析法进行降维处理，得出贡献率较高的若干指标。接着，综合这些指标，赋予合理的权重，使用 KNN 算法建立预测模型，再使用模型对表单 3 中的数据进行预测，得到预测结果。最后，我们对预测结果进行合理性和敏感性的分析。

2.4 问题四的分析

问题四要求我们对不同类别的玻璃分析其化学成分之间的关系（简称为“横向比较”），并比较不同类别玻璃之间化学成分关联关系的差异性（简称为“纵向比较”）。实质上是要求我们选择合适的指标，来刻画“横向比较”的相关性和“纵向比较”的差异性，并做出相应的分析。

与问题三类似，我们使用 spearman 相关系数作为衡量各化学成分之间关联关系的指标，得到各化学成分之间的相关系数矩阵，绘制出热力图，分析“横向比较”的相关

性；再使用 Wilcoxon 符号秩检验，从宏观上进一步探究“纵向比较”的差异性。

三、 模型假设

假设 1 **每个玻璃文物样品的化学成分信息准确，足以完全反映出其所属类别。**

假设 2 **风化样品的采样点若是未风化点，则将该样品划入无风化组进行分析。**

四、 符号说明

表 **1** 符号说明

符号	定义	单位
$predict$	风化检测点风化前化学成分含量的预测值	%
$sample$	风化检测点实际化学成分含量	%
δ	同类玻璃类型化学成分含量的平均变化率	%
$component_i$	第 i 种化学成分的预测含量	%
$component_i^{norm}$	缩放后的化学成分预测含量	%
σ	样本标准差	%
$mean$	样本均值	%
η	某化学成分含量总和除以被检测到的次数	%

注：表中未出现的符号在文中均有详细说明

五、 数据预处理

5.1 剔除无效数据

题目明确指出：“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”。因此，我们筛选出成分比例累加和低于 85% 的两个数据（如表 2），并将这两项数据的全部信息剔除。

表 **2**　无效数据

文物采样点	纹饰	类型	颜色	表面风化	化学成分百分比合计
15	C	高钾	浅蓝	无风化	79.74
17	C	高钾	浅蓝	无风化	71.84

5.2 填补缺失的颜色信息

附件表单 1 中，文物编号为 19，40，48，58 的四个玻璃文物的颜色信息缺失。由于本题所给样本总量为 58，在剔除了上部分的两个无效数据之后，样本量就仅剩 56 个。考虑到本题需要通过这些样本数据来分析规律，对样本数量的依赖较大，若样本数据过少，将不利于后续问题的模型建立与分析。因此，我们并不打算将这四个文物进行剔除，而是希望进行数据填补，将它们的颜色信息预测出来。

由于玻璃颜色与其化学成分之间存在直接的关系，所以，若只关注附件表单 1 的数据，使用众数进行填补，则这个填补的数据只具备统计学上的意义，而失去科学性、现实性。因此，我们结合表单 1 和表单 2 两个表单的数据信息，先将与待填补样本物理特征相同所有样本筛选出，根据化学成分含量来预测玻璃文物的颜色。

为此，我们使用随机森林，根据风化、类型、纹饰情况相同样本的化学成分含量与颜色的关系，预测 19，40，48，58 号文物缺失的颜色信息。填补结果如表 3 所示。

表 **3**　颜色信息填补结果

文物编号	纹饰	类型	颜色	表面风化
19	A	铅钡	黑	风化
40	C	铅钡	深绿	风化
48	A	铅钡	浅蓝	风化
58	C	铅钡	浅蓝	风化

六、 问题一的求解

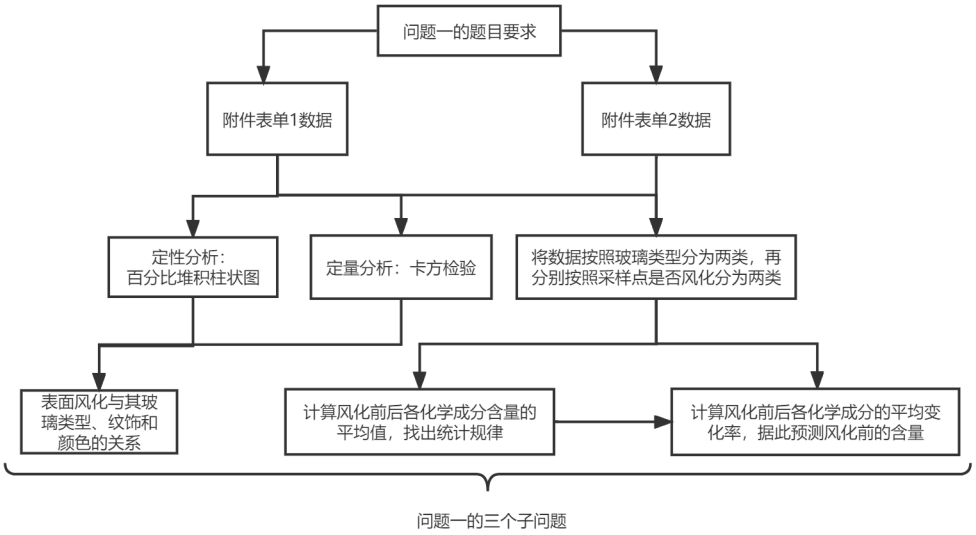


图 **1**　问题一求解思路框图

6.1 分析表面风化与类型、纹饰、颜色的关系

问题一要求我们对玻璃文物表面风化与其玻璃类型、纹饰和颜色的关系进行分析，即此时只需要我们关注附件表单 1 提供的信息，通过研究表面风化与其他三个特征之间分别的关系，对数据进行初步、简要地刻画。因此，我们分别统计出表面风化与三个特征之间的数据，使用 Excel 做出柱状图，进行初步的规律分析。

6.1.1 分析表面风化与类型关系

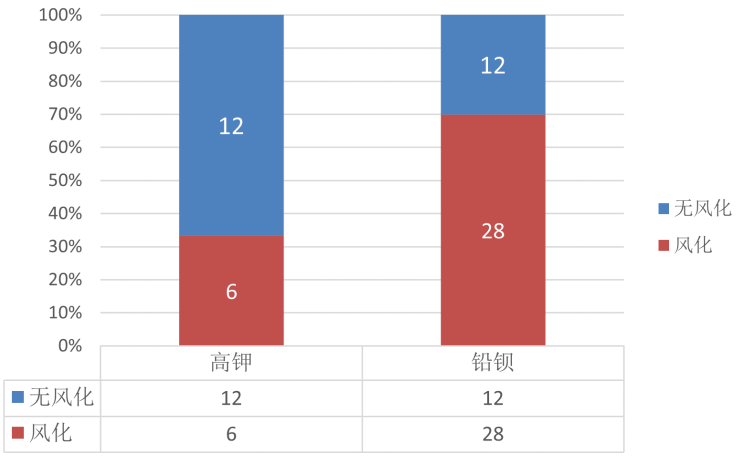
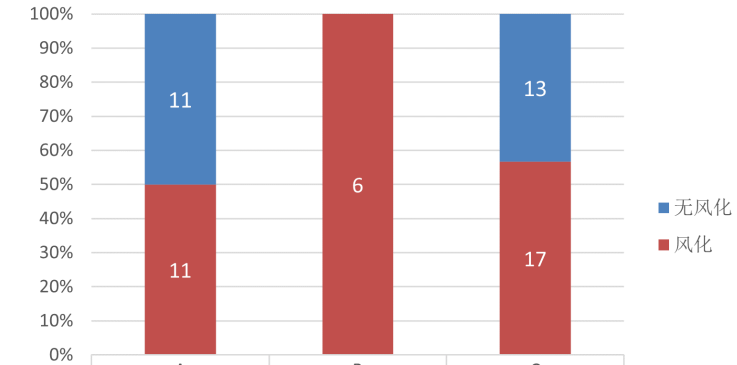


图 **2**　表面风化与玻璃类型

由图 2可以看出，对高钾玻璃而言，66.6% 的文物表面无风化，而对铅钡玻璃而言，30% 的文物表面无风化。这说明，玻璃类型对表面风化的影响较大，其中，类型为高钾玻璃的无风化比例更高，说明高钾玻璃在长期埋葬于地下的过程中，稳定性较好，较不容易产生风化的现象。[3]

6.1.2 分析表面风化与纹饰关系



	A	B	C
无风化	11	0	13
风化	11	6	17

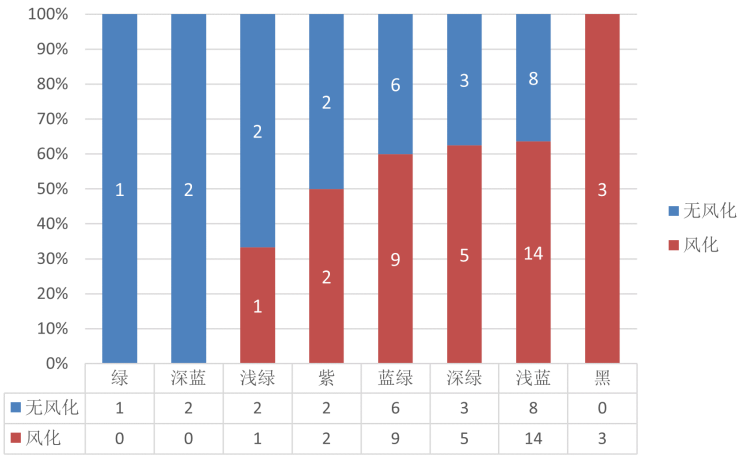
图 **3**　表面风化与玻璃纹饰

由图 3可以看出，B 类纹饰的玻璃文物表面有风化的比例为 100%，而 A 类、C 类纹饰表面有分化的比例居中，分别为 50% 和 43.3%。因此可以分析得出，B 类纹饰有很可能对玻璃的风化与否造成直接的显著影响，而 A 类、C 类纹饰的影响效果并不太显著。但考虑到这三种纹饰类型之间的样本存在不平衡的特点（数量比例为 22:6:30），因此，分析得出的 B 类纹饰的显著影响效果也有可能是因样本数量过少而导致的。

6.1.3 分析表面风化与颜色关系

由图 4可知，绿色和深蓝的玻璃文物全无风化，而黑色的却全部是风化；浅蓝、蓝绿和深绿这三种颜色的玻璃表面风化的数量占比都超过了 50%，而浅绿和紫色数量较少，但也出现了风化的现象。

可以发现，颜色方面并未存在显著规律性。经过查阅文献 [4] 可知，古玻璃制造工艺中还广泛使用 MnO、Cr_2O_3、CoO 等化学物质作为着色剂，分别能使玻璃显示为紫-紫红色、黄绿色、蓝色等，这些化学成分的信息可能因为采样部位的随机性、检测手段的误差而并未纳入数据中，因此我们无法得知准确的化学成分含量信息，也就无法准确地描述出颜色与表面风化的关系。



	绿	深蓝	浅绿	紫	蓝绿	深绿	浅蓝	黑
无风化	1	2	2	2	6	3	8	0
风化	0	0	1	2	9	5	14	3

图 **4**　表面风化与玻璃颜色

6.1.4 卡方检验

卡方检验是以 χ2 分布为基础的非参数假设检验方法。该检验的基本思想是：首先假设 H_0 成立，计算出偏离程度 χ2，由此确定当前统计量的概率 P。如果 P 值很小，说明观察值与理论值偏离程度太大，应当拒绝无效假设，表示比较资料之间有显著差异；否则就不能拒绝无效假设。

在本题中，卡方检验的零假设 H_0 是：所选特征与表面风化没有关系。通过计算，我们得到检验的结果（表 4）。

表 **4**　卡方检验结果

类型	纹饰	颜色
0.02938188*	0.74329816	0.55800793

我们可以分析得到以下信息：

1. “类型”的 P 值小于 0.05，因此，在 95% 置信区间内“类型”与“表面风化”的相关性是显著的，这与 6.1.1 展现的图表结果一致。
2. “纹饰”的 P 值明显大于 0.05，表明其与“表面风化”的相关性较差。结合 6.1.2 的图表可知，A、C 两类纹饰的风化占比居中，B 类纹饰的玻璃文物虽然全部风化，但样本数过少，并不具备说服力。
3. “颜色”的 P 值也大于 0.05，表明其与“表面风化”的相关性较差。结合 6.1.3 的图表可知，不同颜色的样本数量之间也存在着不平衡的特点，分类效果最好的“黑色”仅有三个样本数量，而样本数量最多的“浅蓝”风化率也只有 63.6%。

6.2 分析表面风化与化学成分含量的统计规律

该小问要求我们对两种玻璃文物类型分别分析。我们将一种类型的玻璃文物，按其采样点是否风化，分成风化组和无风化组两组。其中，根据假设 2，我们将那些在风化样本的未风化点采样的样本，划分到无风化组。在风化组与未风化组中，我们分别对化学成分的含量取平均值，进行对比分析。

6.2.1 铅钡玻璃风化前后化学成分含量分析

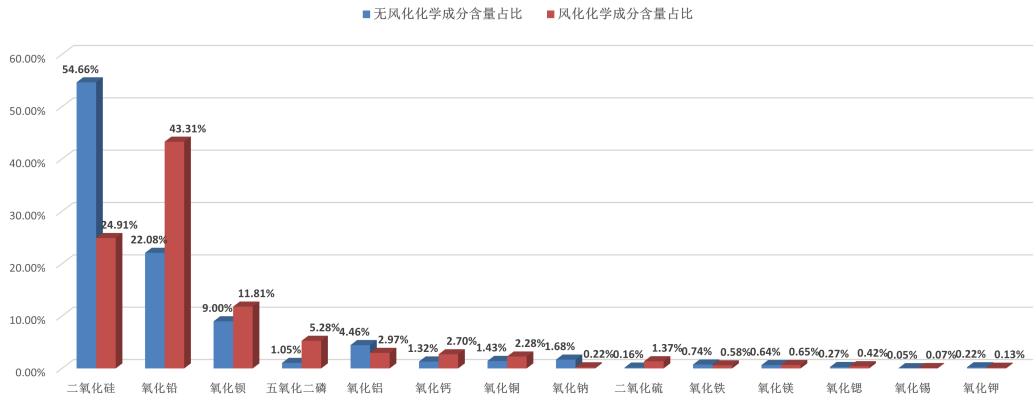


图 **5**　铅钡玻璃风化前后化学成分含量平均变化

表 **5**　铅钡玻璃风化前后化学成分含量平均变化率

	二氧化硅	氧化铅	氧化钡	五氧化二磷	氧化铝	氧化钙	氧化铜	氧化钠	二氧化硫	氧化铁	氧化镁	氧化锶	氧化锡	氧化钾
风化前后变化率	-54.42%	96.13%	18.08%	403.02%	-33.35%	104.13%	58.95%	-87.15%	758.51%	-20.62%	1.49%	55.99%	47.16%	-38.97%

铅钡玻璃主要的化学成分是氧化铅、氧化钡和二氧化硅。结合图 5和表 5可以看出，铅钡玻璃在风化后，二氧化硅的含量减少了 54.42%，而氧化铅和氧化钡的含量出现了上升。同时发现，五氧化二磷的含量在风化后有 400% 的上升率，较为突出；其他成分在风化前后变化不明显或含量均较低，不做具体分析。因此可以得出结论：铅钡玻璃在风化之后二氧化硅的含量将下降，而氧化铅、氧化钡、五氧化二磷的含量将上升。

6.2.2 高钾玻璃风化前后化学成分含量分析

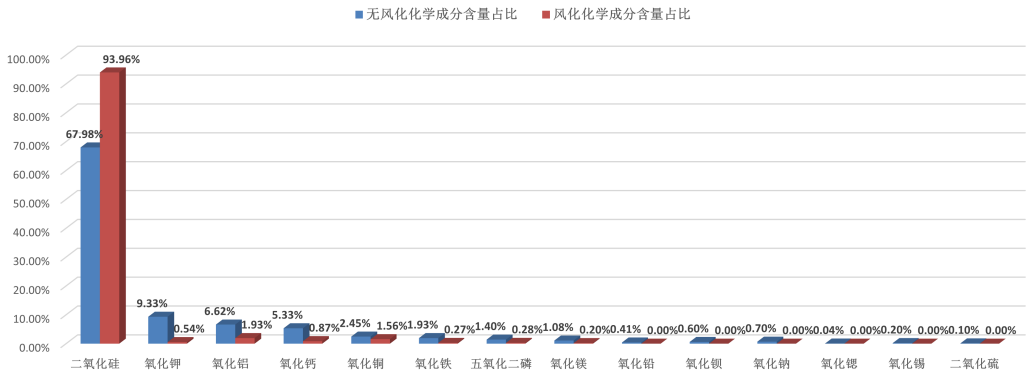


图 6　高钾玻璃风化前后化学成分含量平均变化

表 6　高钾玻璃风化前后化学成分含量平均变化率

	二氧化硅	氧化钾	氧化铝	氧化钙	氧化铜	氧化铁	五氧化二磷	氧化镁	氧化铅	氧化钡	氧化钠	氧化锶	氧化锡	二氧化硫
风化前后变化率	38.21%	-94.18%	-70.85%	-83.68%	-36.32%	-86.28%	-80.04%	-81.78%	-100.00%	-100.00%	-100.00%	-100.00%	-100.00%	-100.00%

高钾玻璃的主要化学成分是二氧化硅和氧化钾。结合图 6和表 6可以看出，高钾玻璃在风化后，二氧化硅的含量明显增加，变化率 38.21%，而氧化钾的含量下降了 94.18%，下降程度显著；同时发现，风化后氧化铝和氧化钙含量也有明显的减少，变化率在-80%-70% 的水平；其他成分在风化前后含量均较低，不做具体分析。因此可以得出结论：高钾玻璃在风化之后二样化硅的含量将上升，氧化钾的含量将显著下降，氧化铝和氧化钙的含量将有所下降。

6.3 预测风化检测点风化前的化学成分含量

我们已经在上一个子问题中计算出了同类型玻璃风化前后的各化学成分含量的平均变化率。由于这个平均变化率综合了全部同类样本信息，具有一定的普适性，因此，我们可以通过平均变化率来预测风化检测点风化前的化学成分含量。

考虑到风化前后，有些化学成分会出现从有到无的情况，所以我们规定：

- 若风化检测点某化学成分的含量为 0，则用未风化时该化学成分含量的平均值当作预测值。
- 若某项化学成分含量的平均变化率为-100%，即风化使得该成分消失，则同样用未风化时该化学成分含量的平均值当作预测值。

在排除上述两种特殊情况后，预测的数学公式为：

$$predict = \frac{sample}{1+\delta}$$

其中，δ 为平均变化率，$predict$ 为预测值，$sample$ 为实际样本数据。

预测后，我们再对每个样本的各化学成分预测值进行缩放，以保证相对比例不变的情况下，化学成分含量总和为 100%。缩放的数学公式如下：

$$component_i^{norm} = \frac{component_i}{\sum_{i=1}^{14} component_i}$$

其中，$component_i$ 为第 i 种化学成分的预测含量，$component_i^{norm}$ 为缩放后的预测含量。

表 7　两种玻璃风化点在风化前化学成分含量的部分预测结果

铅钡玻璃文物编号	二氧化硅	氧化钠	氧化钾	氧化钙	氧化镁	氧化铝	氧化铁	氧化铜	氧化铅	氧化钡	五氧化二磷	氧化锶	氧化锡	二氧化硫
2	60.93	1.29	1.32	0.88	0.89	6.58	1.79	0.13	18.51	6.89	0.54	0.09	0.04	0.12
8	16.28	2.71	0.35	2.52	1.03	2.68	1.19	3.18	26.63	37.58	2.42	0.55	0.07	2.82
11	65.56	1.50	0.31	1.53	0.62	3.59	0.66	2.76	11.52	9.91	1.66	0.21	0.04	0.14
19	61.24	1.58	0.21	1.35	0.55	5.04	1.58	2.08	20.56	3.84	1.65	0.11	0.04	0.15
高钾玻璃文物编号	二氧化硅	氧化钠	氧化钾	氧化钙	氧化镁	氧化铝	氧化铁	氧化铜	氧化铅	氧化钡	五氧化二磷	氧化锶	氧化锡	二氧化硫
7	65.57	0.68	9.13	6.42	1.06	6.64	1.21	4.98	0.40	0.59	2.99	0.04	0.19	0.10
9	70.98	0.72	10.46	3.92	1.11	4.67	2.41	2.51	0.43	0.62	1.81	0.04	0.20	0.10
10	71.72	0.71	16.18	1.32	1.11	2.85	1.94	1.35	0.42	0.61	1.44	0.04	0.20	0.10

注：详细预测结果数据请见附录

为了验证预测结果的可信度，我们将预测出含量取平均值，与无风化组的均值进行比较，观察其差异性。

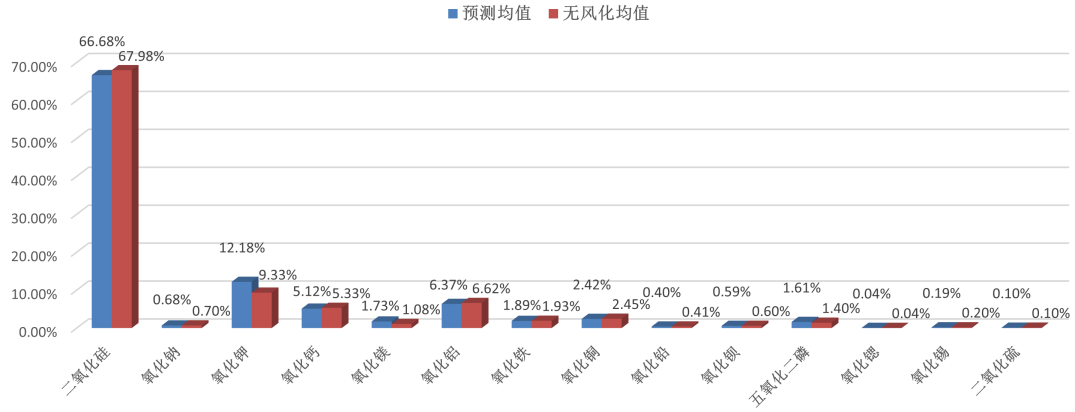


图 7　高钾类样品风化预测点预测结果检验

对于高钾类样品，氧化钾的预测均值较高，这主要是因为风化会导致氧化钾的含量显著下降，所以在反过来做线性映射式预测时，会使得预测值偏高。当然这也和待预测样本本身氧化钾含量较高有关。其他成分的预测含量都十分接近，所以预测结果总体上合理。

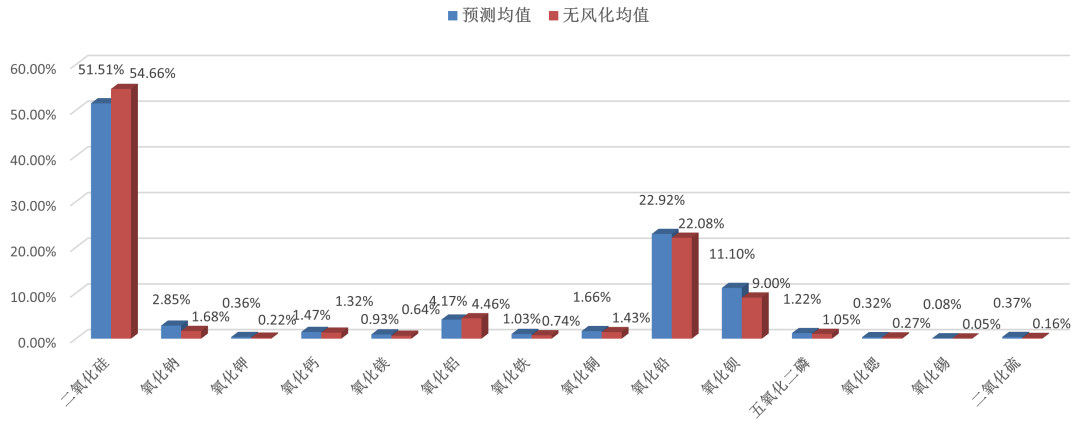


图 **8**　铅钡类样品风化预测点预测结果检验

对于铅钡类样品，二氧化硅的预测均值偏低，氧化钡的预测均值偏高，这一方面是因为待预测样品本身的性质，另一方面是因为二氧化硅的含量占比较大，一定程度上会放大误差。而氧化钡在风化后增加不明显，所以反过来通过线性映射关系预测时，其含量不会明显降低。考虑到二氧化硅与氧化钡的预测值偏差并不大，且其他成分的预测均值都较为接近，所以预测结果总体上合理。

七、 问题二模型的建立与求解

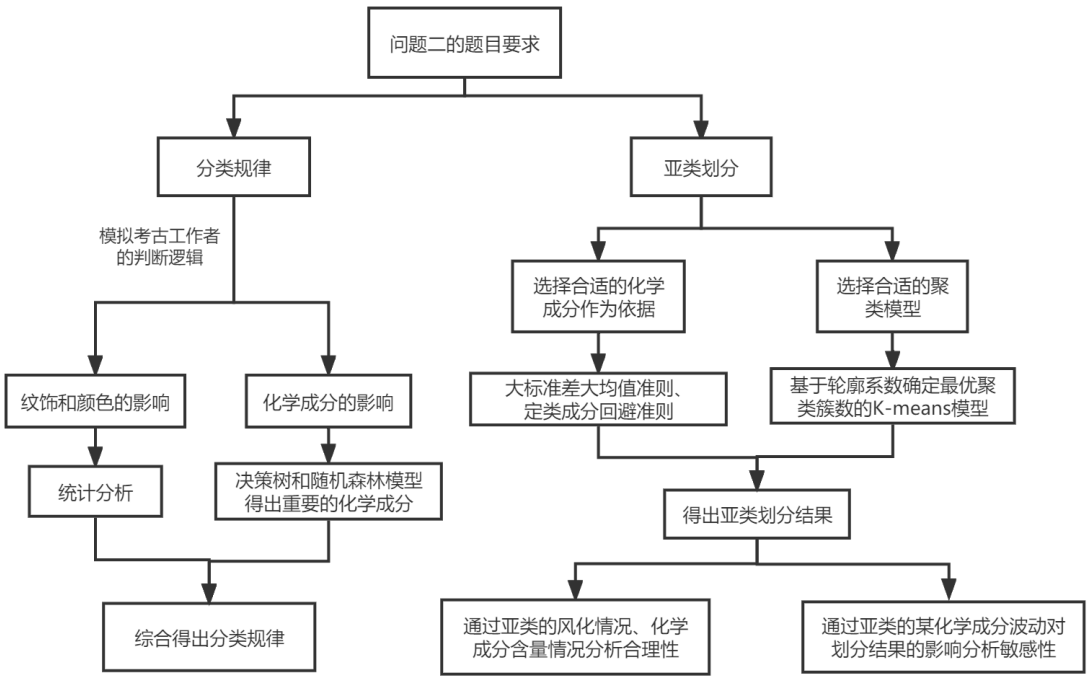


图 **9**　问题二分析思路框图

7.1 分析高钾、铅钡玻璃的分类规律

题目指出：“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”，这提示我们，在分析分类规律时，可以模拟考古工作者的判断逻辑，从而更自然地抓住分类规律。

从考古工作者的角度来说，判断的主要依据是玻璃文物的化学成分含量。由于颜色等物理特征被化学成分所决定，且区分具有一定的主观性，因此，这些物理特征可以作为玻璃类型判断的辅助。所以在下文中，我们先通过统计分析来大致确定纹饰、颜色对判断类型有哪些直观上的帮助，再对化学成分进行更深入的分析。

7.1.1 纹饰、颜色对分类的影响

（1）纹饰对判断类型的影响

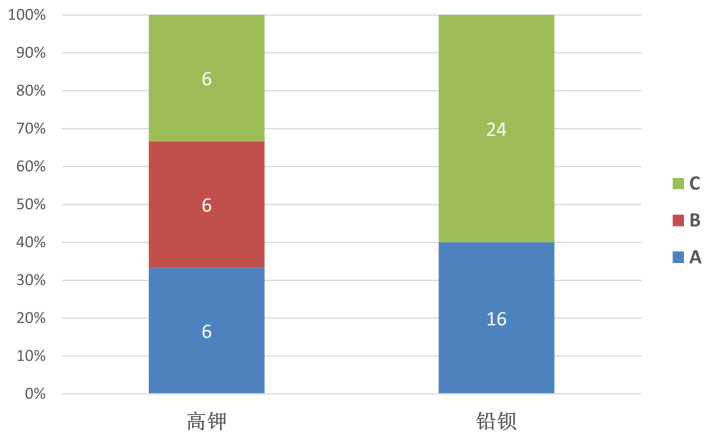


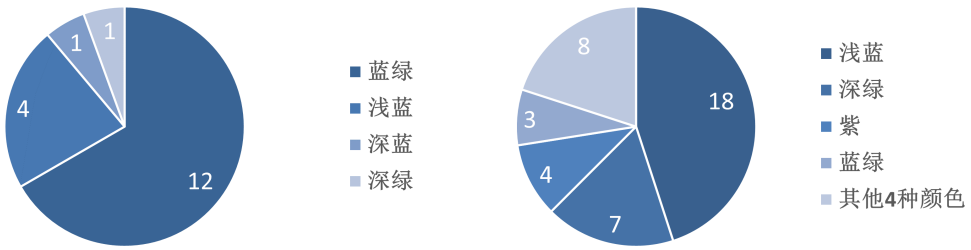
图 **10**　不同类型玻璃的纹饰情况

由图 10可以看出，B 类纹饰只在高钾类中出现，而 A 和 C 两类纹饰在两类中均有出现，且没有明显的比例差异。总的来说，如果发现玻璃文物的纹饰是 B 类，则该文物属于高钾玻璃类的概率大于铅钡玻璃。

（2）颜色对判断类型的影响

由图 11可以看出，蓝绿色玻璃样本大多集中在高钾玻璃中，且占比较高；浅蓝与深绿色玻璃样本大多集中在铅钡玻璃中，且占比较高；其他颜色的占比均较少。总的来说，蓝绿色玻璃样本属于高钾类的概率更大，浅蓝与深绿色玻璃样本属于铅钡类的概率更大。

图 **11**　不同类型玻璃的颜色情况



(a) 高钾玻璃的颜色情况　　**(b)** 铅钡玻璃的颜色情况

7.1.2 决策树模拟判断模型

上文提到，可以通过模拟考古工作者判断逻辑的方式来分析玻璃样本的分类规律。由此我们联想到了决策树模型，决策树通过对样本的某些特征依次进行“提问”，逐步确定样本所属类型。而考古工作者在依据化学成分对玻璃文物进行分类时，一般是先看某些较为重要的化学成分含量，得出大致的情况，再根据其他次重要的化学成分，最终确定其分类，这与决策树的原理十分一致。

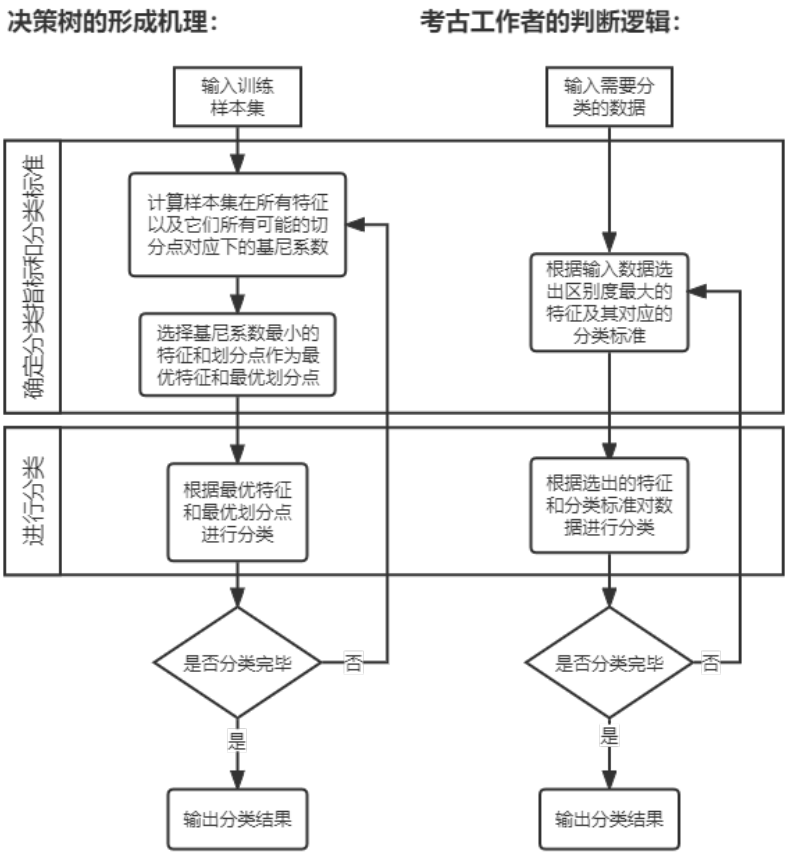
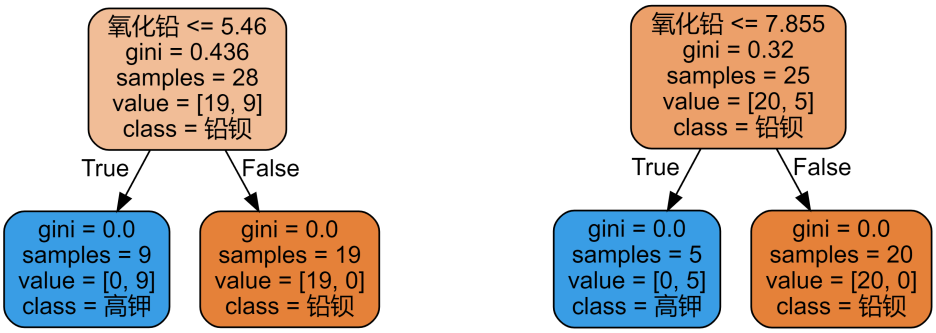


图 **12**　决策树和人的判断机理

由问题一的求解可知，玻璃样品风化前后的化学成分含量会出现较大变化，并且考古工作者在判断玻璃样品类型时，一般会先得知其是否风化，再去具体分析其化学成分含量。因此，我们根据是否风化将样品分成两组，且根据假设 2，在风化样品无风化点的采样数据会被归入无风化组。这与问题一的处理一致。

我们使用 scikit-learn 库的决策树模型，对两组样本分别使用决策树进行拟合，保证拟合出来的决策树模型在测试组中的准确率高于 95%，并使用 graphviz 库，对决策树的拟合结果进行可视化，如图 13所示：

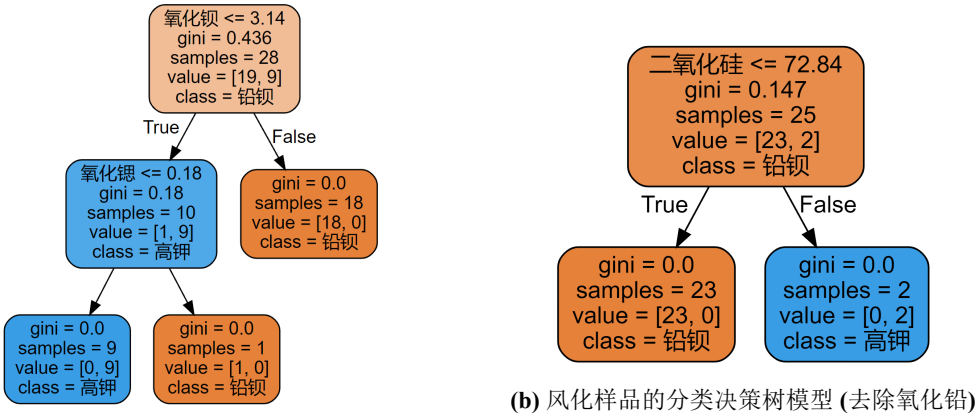
图 **13**　玻璃样品的分类决策树模型



(a) 无风化样品的分类决策树模型　　**(b)** 风化样品的分类决策树模型

不难看出，无论是风化样品还是无风化样品，氧化铅都是判断其类别的主要依据。在本题的样本中，使用氧化铅作为判断依据可以达到 100% 的准确率。考虑到本题样本数据的特殊性，我们尝试删除氧化铅这个特征，探究在这种情况下决策树模型的表现：

图 **14**　玻璃样品的分类决策树模型 **(**去除氧化铅**)**



(a) 无风化样品的分类决策树模型 **(**去除氧化铅**)**　　**(b)** 风化样品的分类决策树模型 **(**去除氧化铅**)**

分析图 14可知，当不提供氧化铅信息时，对于无风化样品，分类的主要依据变成了氧化钡；对于风化样品，分类的主要依据变成了二氧化硅。

通过上面的分析可知，无风化样品与风化样品的判断逻辑确实有所不同，并且通过样品的少数重要化学成分即可对样品的类型作出准确度较高的判断。但考虑到一棵决策

树的效果可能出现较大偏差，且 scikit-learn 对决策树的实现具有一定的随机性，因此，我们进一步使用随机森林模型。

随机森林模型集成了多棵决策树，在判断类型时，采用“表决投票、服从多数”的方式，这样大大减小了决策树随机性带来的影响，也符合多个考古工作者共同商定的情况；与此同时，通过多棵决策树对化学成分的选择情况，也可以得出哪些化学成分对于判断类型起着主要作用。

7.1.3 随机森林模拟判断模型

我们使用 scikit-learn 库的 RandomForestRegressor 模型进行拟合，得到的两个随机森林模型在测试集上的准确率均超过了 95%。同时，我们可以得到各化学成分在判断类型中的重要性，如图 15和图 16所示。需要指出的是，图中未列出重要性为 0 的化学成分。

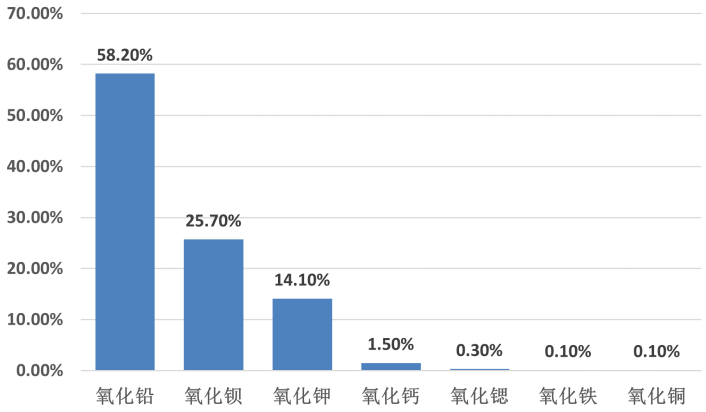


图 **15**　无风化样品化学成分重要性

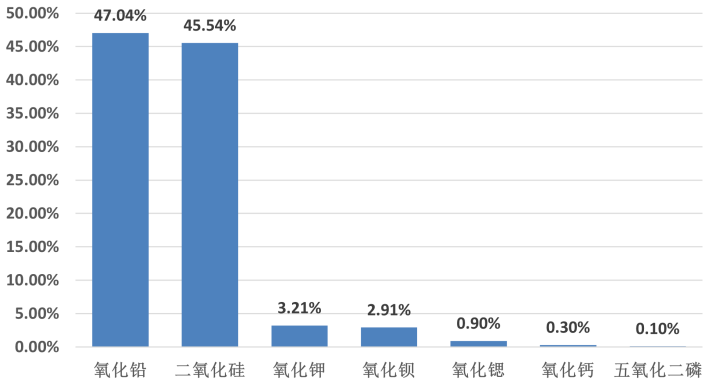


图 **16**　风化样品化学成分重要性

可以看出，无论样品是否风化，氧化铅的含量都是判断的主要依据。对于无风化样品，还可以通过氧化钡和氧化钾的含量辅助判断；对于风化样品，二氧化硅的含量与氧

化铅的重要程度相近，通过这两个化学成分即可做出准确率较高的判断。

7.1.4 分类规律总结

综合以上分析，我们可以得出如下几条分类规律：

- 对于无风化样品，若其氧化铅和氧化钡含量都明显高于 5% 的水平，则认为其为铅钡类；否则，认为其为高钾类。
- 对于有风化样品，若其氧化铅含量明显高于 7% 的水平，或其二氧化硅的含量明显低于 70% 的水平，则可以确定其为铅钡类；否则，认为其是高钾类。
- 在无法通过以上两条规律明确判断的情况下，可以通过氧化钾的含量具体分辨。
- B 类纹饰的风化样品属于高钾类的概率更大、蓝绿色样品属于高钾类的概率更大、浅蓝色样品属于铅钡类的概率更大，可以据此检验分类判断结果。

7.2 高钾、铅钡玻璃的亚类划分

7.2.1 亚类划分的化学成分选择

亚类划分的主要目的是，根据类内某些化学成分含量的差异性，来将其进一步划分。这需要我们找到合适的统计量来刻画数据的差异性。

我们联想到了变异系数 c_v 这个统计量，其计算公式为

$$c_v = \frac{\sigma}{mean}$$

其中，σ 表示样本的标准差，$mean$ 表示样本的平均值。但由于本题中有些并不重要的化学成分含量的平均值过低（接近于 0），这会导致其变异系数过大，影响分类结果的精准性。

为了弥补 c_v 精度不佳的缺陷，我们综合使用标准差 σ 和“排零均值”η 两个统计量来描述数据的差异性。选择理由为：

1. 标准差 σ 可以表现数据的离散程度，若标准差大，则说明含量变化明显，可以成为我们选择化学成分的主要依据。
2. “排零均值”η 是某化学成分含量总和除以被检测到的次数。若“排零均值”小，则说明该化学成分对玻璃文物理化性质的影响小，可以成为我们排除化学成分的主要依据。

在不考虑风化的影响下，高钾玻璃的 K_2O 含量都较高，而铅钡玻璃的 PbO 和 BaO 含量都较高。因此，K_2O 相对于高钾类，PbO 和 BaO 相对于铅钡类，并不适用于亚类划分。并且，由问题一可知，SiO_2 的含量变化足以反映同类玻璃是否发生风化。

综合以上考虑，我们可以提出两个选择准则：

1. 大标准差大均值准则: 分别计算出高钾、铅钡类各化学成分标准差 σ 和“排零均值” η，并只选取 σ 和 η 都大于 2% 的化学成分。
2. 定类成分回避准则：在进行高钾玻璃的亚类划分时，不考虑 K_2O；在进行铅钡亚类划分时，不考虑 PbO 和 BaO。

(1) 高钾类亚类划分化学成分的选择

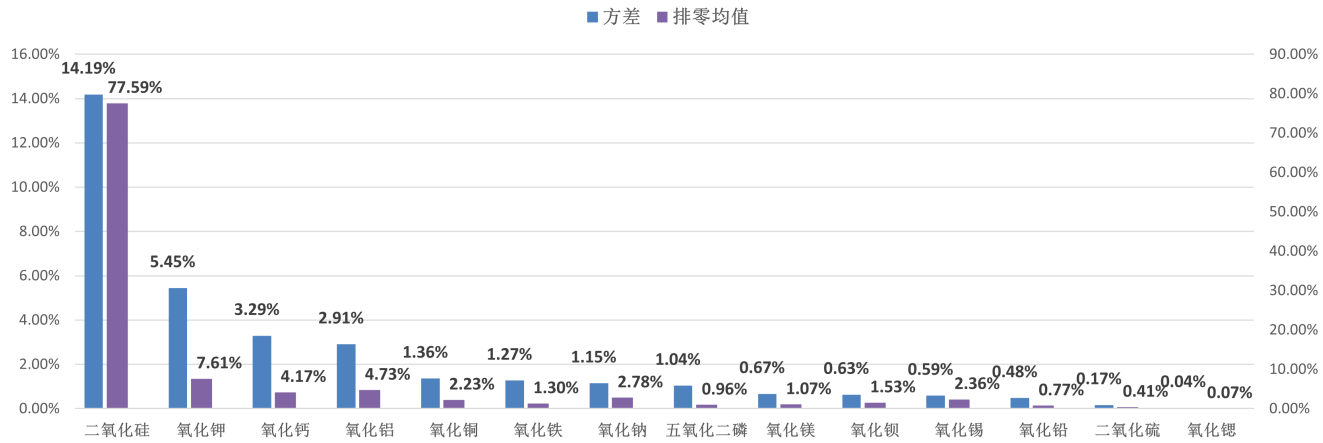


图 **17**　高钾类样品和化学成分的标准差和“排零均值”

由图 17可知，除去氧化钾，标准差 σ 和“排零均值” η 都大于 2% 的化学成分有二氧化硅、氧化钙、氧化铝，故我们选取这三个化学成分作为亚类划分的依据。

(2) 铅钡类亚类划分化学成分的选择

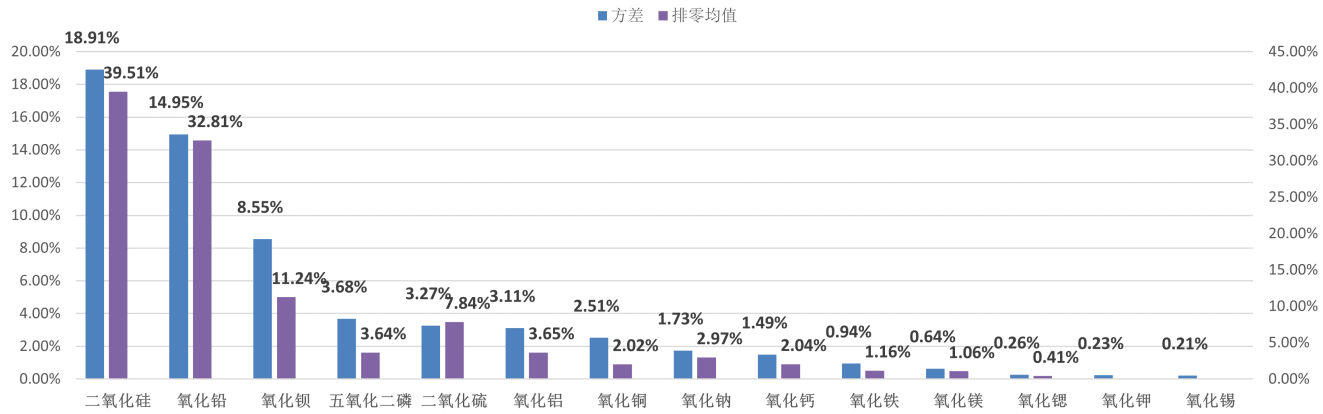


图 **18**　铅钡类样品和化学成分的标准差和“排零均值”

由图 18可知，除去氧化铅和氧化钡，标准差 σ 和“排零均值” η 都大于 2% 的化学成分有二氧化硅、五氧化二磷、二氧化硫、氧化铝、氧化铜，故我们选取这五个化学成分作为亚类划分的依据。

7.2.2 亚类划分数目的选择

在亚类划分时，由于样本数量少，因此亚类数目也应该较少，从而保证每个类中有较多的样本，以使得亚类划分具有代表性和普适性。在该问中，我们选择使用 K-means 聚类模型，并使用轮廓系数进行聚类效果的评价。K-means 聚类算法具体步骤如 algorithm 1所示。

Algorithm 1 K-means 聚类算法

Input: 样本集 $D = \{x_1, \cdots, x_n\}$;

Output: 聚类 $\{C_1, \cdots, C_K\}$;

1: 随机选择 K 个聚类均值 $\boldsymbol{m}_j, j = 1, \cdots, K$;
2: **while** 直到 K 个均值仍发生变化 **do**
3: $C_j = \varnothing, j = 1, \cdots, K$
4: **for** i=1 to n **do**
5: $k = \underset{1 \leqslant j \leqslant K}{\arg\min} \|\boldsymbol{x}_i - \boldsymbol{m}_j\|, C_k = C_k \cup \{\boldsymbol{x}_i\}$
6: **end for**
7: 更新 K 个聚类的均值: $\boldsymbol{m}_j = \frac{1}{n_j} \sum_{x \in c_j} x, j = 1, \cdots, K$
8: **end while**

聚类模型的一个重要评价指标是轮廓系数，它的定义如下

$$s = \begin{cases} 1 - a/b, & \text{if } a < b \\ 0, & \text{if } a = b \\ b/a - 1, & \text{if } a > b \end{cases}$$

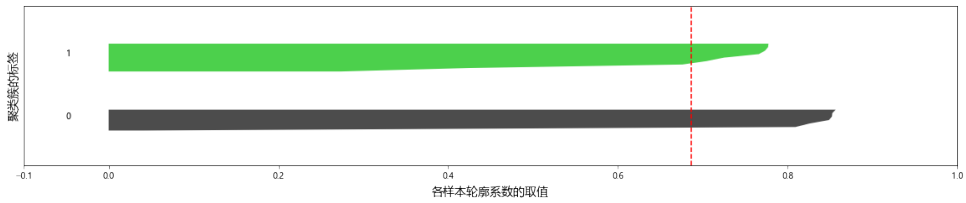
其中，a 表示样本与其自身所在的簇中的其他样本的相似度，等于样本与同一簇中所有其他点之间的平均距离；b 表示样本与其他簇中的样本的相似度，等于样本与下一个最近的簇中的所有点之间的平均距离。

由定义可知，轮廓系数的范围在-1 和 1 之间，反映了一个样本适合于该聚类簇的程度，越接近 1 代表与同一聚类簇的其他样本相似度越高，若小于 0 则表示该样本与其他聚类簇样本的相似度高于本聚类簇（这说明聚类效果很差）。我们可以画出在不同聚类簇数下所有样本的轮廓系数取值，并在图中标明轮廓系数的平均值，通过这样的直观呈现来选择合适的聚类簇数。

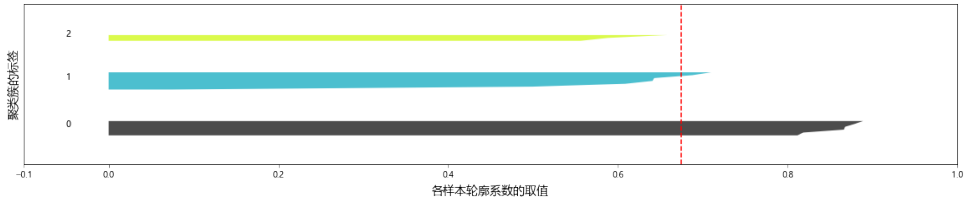
(1) 高钾类亚类数目的选择

分别设置聚类簇数为 2、3、4，按照上文所述作出图像如图 19所示。

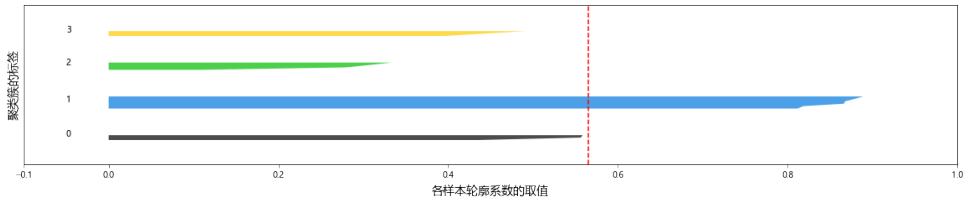
图 **19**　高钾样品不同聚类簇数下的轮廓系数情况



(a) 聚类簇数为时 **2** 的轮廓系数



(b) 聚类簇数为时 **3** 的轮廓系数



(c) 聚类簇数为时 **4** 的轮廓系数

图中红色的虚线代表平均轮廓系数。可以看出，当聚类簇数为 2 时，平均轮廓系数最高（红色虚线最靠右），且每个聚类簇中都有至少半数样本的轮廓系数大于平均值；当聚类簇数为 3 时，有两个聚类簇中大部分样本的轮廓系数都小于平均值；当聚类簇数为 4 时，轮廓系数明显减小，并且出现了两个聚类簇中所有样本的轮廓系数都小于平均值的情况。

由于聚类簇数为 2 的轮廓系数较优，且我们原先也希望选择分类数目较少的聚类结果。因此，我们确定聚类簇数为 2，即高钾类样品的亚类划分数目为 2。

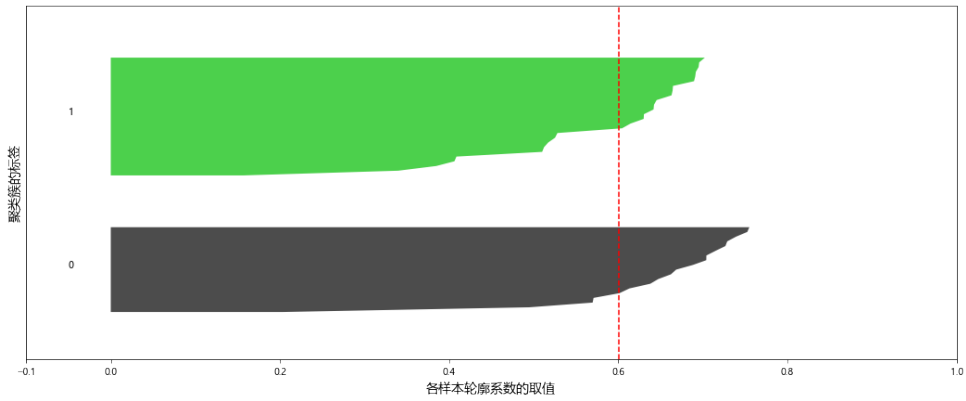
(2) 铅钡类亚类数目的选择

分别设置聚类簇数为 2、3、4，按照上文所述作出图像如图 20所示。

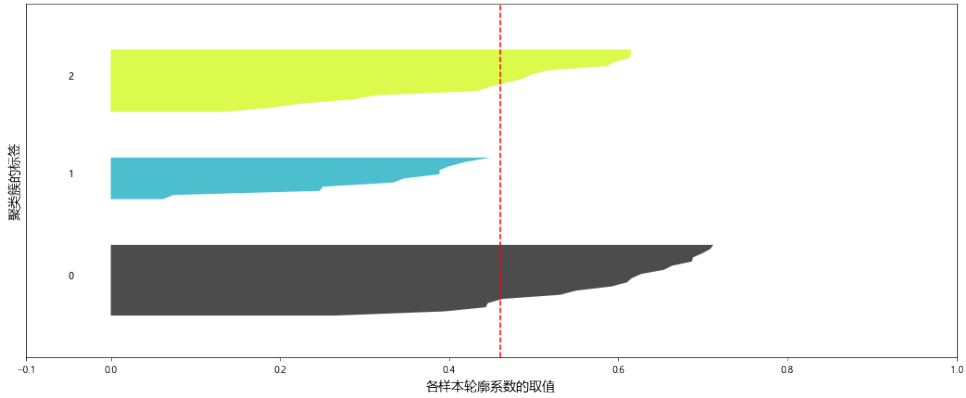
图中红色的虚线代表平均轮廓系数。可以看出，当聚类簇数为 2 时，平均轮廓系数最高（红色虚线最靠右），且每个聚类簇中都有至少半数样本的轮廓系数大于平均值；当聚类簇数为 3 和 4 时，轮廓系数明显减小，并且出现了某个聚类簇中所有样本的轮廓系数都小于平均值的情况。

由于聚类簇数为 2 的轮廓系数较优，且我们原先也希望选择聚类数目较少的聚类结果。因此，我们确定聚类簇数为 2，即铅钡类样品的亚类划分数目为 2。

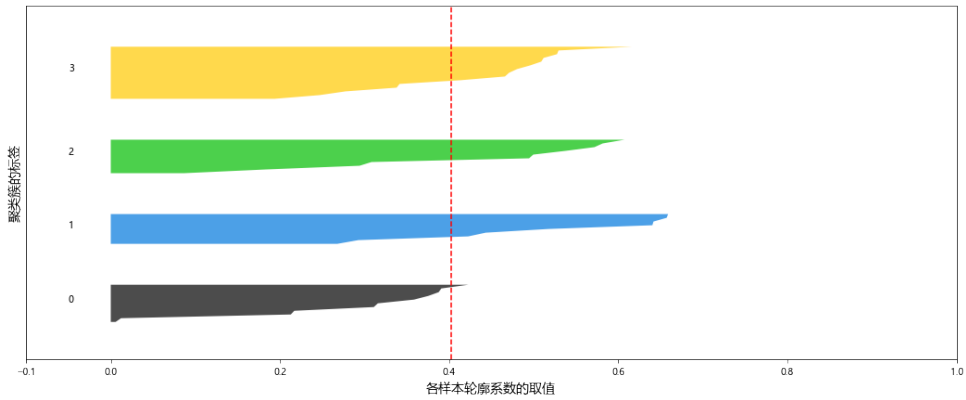
图 **20**　铅钡样品不同聚类簇数下的轮廓系数情况



(a) 聚类簇数为时 **2** 的轮廓系



(b) 聚类簇数为时 **3** 的轮廓系数



(c) 聚类簇数为时 **4** 的轮廓系数

7.2.3 亚类划分结果合理性的分析

在确定好亚类的化学成分与要划分的亚类数目后，即可通过 scikit-learn 库的 KMeans 模型进行正式聚类，部分亚类划分结果如下表所示。详细结果见附录。

表 **8**　高钾玻璃亚类划分的部分结果

文物编号	文物采样点	亚类标号	二氧化硅 (SiO2)	氧化钙 (CaO)	氧化铝 (Al2O3)	氧化钾 (K2O)	氧化铁 (Fe2O3)	氧化镁 (MgO)	氧化铜 (CuO)	五氧化二磷 (P2O5)	纹饰	颜色	表面风化
9	09	0	95.02	0.62	1.32	0.59	0.32	0	1.55	0.35	B	蓝绿	风化
10	10	0	96.77	0.21	0.81	0.92	0.26	0	0.84	0	B	蓝绿	风化
5	05	1	61.58	7.35	7.5	10.95	2.62	1.77	3.27	0.94	A	蓝绿	无风化
6	06 部位 2	1	59.81	5.41	10.05	7.68	6.04	1.73	2.18	4.5	A	蓝绿	无风化

表 **9**　铅钡玻璃亚类划分的部分结果

文物编号	文物采样点	亚类标号	二氧化硅 (SiO2)	五氧化二磷 (P2O5)	氧化铝 (Al2O3)	氧化铅 (PbO)	氧化钡 (BaO)	氧化铜 (CuO)	氧化钙 (CaO)	二氧化硫 (SO2)	纹饰	颜色	表面风化
45	45	0	61.28	0	5	15.99	10.96	0.53	0.84	0	A	浅蓝	无风化
46	46	0	55.21	0.2	4.79	25.25	10.06	0.77	0	0	A	浅蓝	无风化
8	08 严重风化点	1	4.61	7.56	1.11	32.45	30.62	3.14	3.19	15.03	C	紫	风化
50	50	1	17.98	6.34	1.87	44	14.2	1.13	3.19	0	A	黑	风化

为了更充分地说明亚类划分结果的合理性，我们从宏观与微观两个角度展开分析。宏观上主要看不同亚类之间化学成分、风化程度的平均差别，确定该亚类的主要特征；微观上主要挑选一些较为特殊的样本，说明其被划分为该亚类的合理性。

(1) 宏观分析：

对于高钾类样品，其两个亚类的风化情况和各化学成分的平均含量如图 21和图 22所示。需要指出的是，图 21依据的是采样点的风化情况而非样本整体的风化情况，图 22将作为亚类划分依据的化学指标排在了最左侧，并省略了在两个亚类中平均含量都小于 1% 的化学成分。

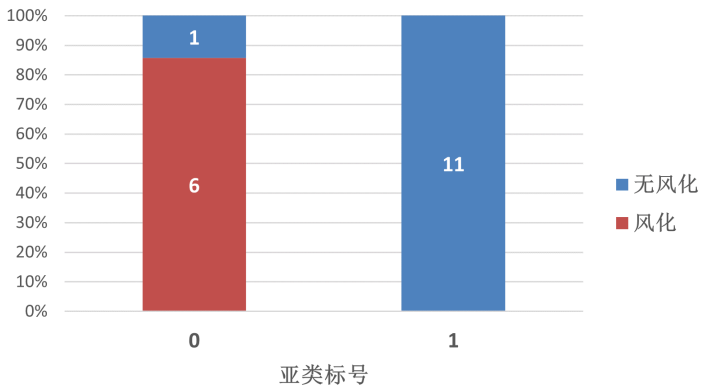


图 **21**　高钾类样品亚类的风化情况

可以看出，亚类 0 是风化程度较高的高钾亚类玻璃，亚类 1 是风化程度较低的高钾亚类玻璃，且亚类 1 的氧化钙、氧化铝含量明显高于亚类 0。考虑到二氧化硅作为通用的玻璃原料，可不参与亚类命名，故认为亚类 0 是低钙铝的高钾亚类玻璃，亚类 1 是高

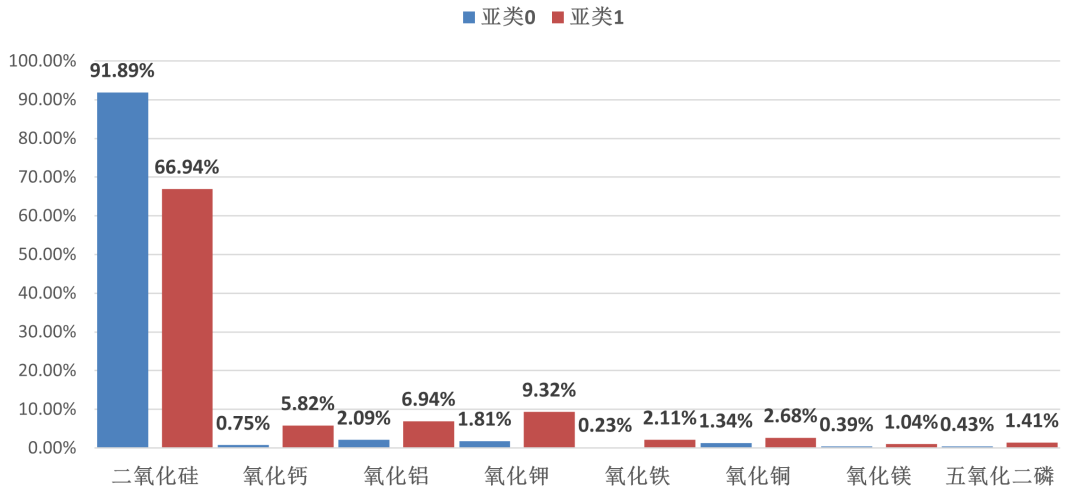


图 **22**　高钾类样品亚类的化学成分含量

钙铝的高钾亚类玻璃，风化会使得高钙铝的高钾亚类玻璃逐渐转化为低钙铝的高钾亚类玻璃。

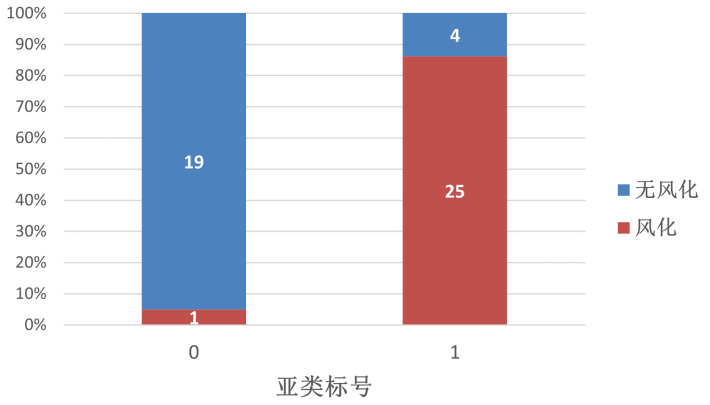


图 **23**　铅钡类样品亚类的风化情况

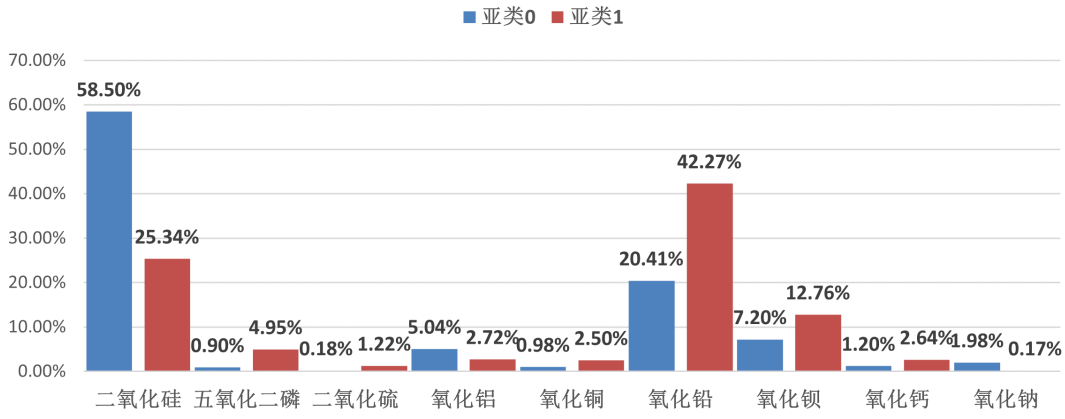


图 **24**　铅钡类样品亚类的化学成分含量

对于铅钡类样品,其两个亚类的风化情况和各化学成分的平均含量如图23和图24所示。

可以看出，亚类 0 是风化程度较低的铅钡亚类玻璃，亚类 1 是风化程度较高的铅钡亚类玻璃，且亚类 0 的氧化铝含量明显高于亚类 1，亚类 1 的五氧化二磷含量明显高于亚类 0。故可以认为亚类 0 是高铝低磷的铅钡亚类玻璃，亚类 1 是高磷低铝的铅钡亚类玻璃，风化会使得高铝低磷的铅钡亚类玻璃逐渐转化为高磷低铝的铅钡亚类玻璃。

(2) 微观分析：

我们发现在低钙铝的高钾亚类玻璃中，编号为 18 的文物采样点是无风化的，而该亚类中其他样品均为风化样品，这个样品的部分信息如表 10 所示：

表 **10**　低钙铝的高钾亚类的中特殊样品

文物采样点	二氧化硅	氧化钙	氧化铝	表面风化
18	79.46	0	3.05	无风化

可以看出，虽然其未风化，但其氧化钙和氧化铝的含量相比于高钙铝的高钾亚类来说很低，并且其二氧化硅含量相对来说较高，可以认为该样本已经呈现出了风化的趋势，所以这样的亚类划分是合理的。

我们发现在高铝低磷的铅钡亚类玻璃中，编号为 48 的文物采样点是风化的，而该亚类中其他采样点均无风化，这个样品的部分信息如表 11 所示：

表 **11**　高铝低磷的铅钡亚类的中特殊样品

文物采样点	二氧化硅	五氧化二磷	氧化铝	表面风化
48	53.33	1.1	13.65	风化

可以看出，虽然其发生了表面风化，且二氧化硅含量相对该亚类的平均值来说较低，但其氧化铝明显偏高，五氧化二磷含量相对偏低，符合高铝低磷的铅钡亚类的特征，可以认为该样本的采样点尚未呈现出风化的趋势，所以这样的亚类划分是合理的。

7.2.4 亚类划分敏感性的探究

我们探究敏感性的思路为: 对于被分到同一亚类的所有样品，让它们的某项化学成分含量上下波动，再用同样的方法聚类，观察当波动率达到什么量级时，聚类结果会发生变化。敏感性分析的结果如表 12 所示。

表 **12**　两种玻璃亚类划分对部分化学成分的敏感性

高钾玻璃亚类			铅钡玻璃亚类		
化学成分	波动率	聚类结果是否保持不变	化学成分	波动率	聚类结果是否保持不变
二氧化硅	-1%	是	二氧化硅	-5%	是
二氧化硅	-2%	否	二氧化硅	-10%	否
二氧化硅	-5%	否	二氧化硅	-15%	否
氧化铝	50%	是	五氧化二磷	500%	是
氧化铝	500%	是	五氧化二磷	1000%	是
氧化铝	1000%	是	五氧化二磷	1600%	否
氧化钙	50%	是	五氧化二磷	1700%	否
氧化钙	300%	是	氧化铝	-10%	是
氧化钙	500%	否	氧化铝	-50%	是
氧化钙	700%	否	氧化铝	-90%	是

(1) 高钾玻璃亚类的敏感性分析

由前文的分析可知，高钾玻璃亚类划分的主要依据是二氧化硅、氧化铝、氧化钙，因此，我们选择让低钙铝的高钾亚类样品的这三个化学成分分别进行不同程度的波动，并观察聚类结果是否保持不变。

需要说明的是，波动率有正有负，是因为对于低钙铝的高钾亚类样品来说，二氧化硅降低、氧化铝升高、氧化钙升高才有可能使其聚类结果发生变化；氧化铝和氧化钙波动范围较大，是因为其实际含量本身就很小，需要较大的波动率才能让其含量有明显变化，从而影响聚类结果。

可以看出，亚类划分结果对于二氧化硅的敏感性很高，对于氧化钙有一定敏感性，对于氧化铝不敏感。

(2) 铅钡玻璃亚类的敏感性分析

由前文的分析可知，铅钡玻璃亚类划分的主要依据是二氧化硅、五氧化二磷、氧化铝，故我们可以选择让高铝低磷的铅钡亚类样品的这三个化学成分分别进行不同程度的波动，并观察聚类结果是否保持不变。

可以看出，亚类划分结果对于二氧化硅的敏感性很高，对于五氧化二磷有一定敏感性，对于氧化铝不敏感。

总的来说，二氧化硅作为含量较高的化学成分，其小幅度波动会导致含量发生较大变化，从而影响亚类划分结果，体现出较高的敏感性；氧化钙和五氧化二磷作为含量较低的化学成分，需要较大幅度波动才会让含量有明显变化，从而能影响亚类划分结果，

体现出一定的敏感性；氧化铝作为含量较低的化学成分，尽管有较大幅度的波动，依然难以影响亚类划分的结果，说明其在亚类划分中充当着辅助判断的作用，要结合其他主要依据才能发挥其作用。

八、 问题三模型建立与求解

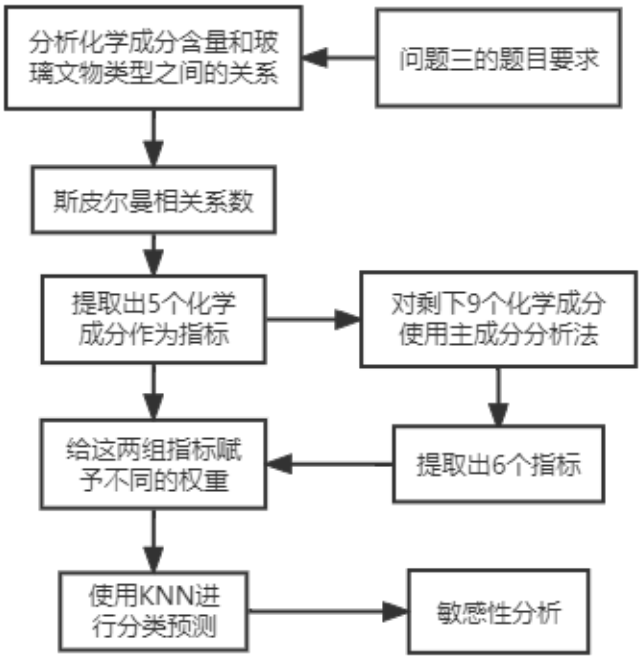


图 **25**　问题三分析思路框图

8.1 指标选取

在指标选取时，不将“表面风化”这个特征作为指标放入该问中进行分析。原因有三：

1. “风化”本身就是玻璃表面的复杂化学反应，因此，“表面风化”这个特征能通过样品化学成分的构成和含量反映出来。
2. 本题样本量较少，若再将所有样本分为风化组和无风化组并进行行建模求解，每个组的样本数会更少，将出现预测效果过差的情况。
3. 预测之后我们需要对预测结果的合理性做出分析。我们可以将“表面风化”特征作为一个判断的重要标志，从而可以有效反映出模型的合理性。

综上，我们只选择所有化学成分作为指标进行分析。

8.1.1 化学成分含量与玻璃文物类型的关系

在数据处理阶段，我们首先利用 Jarque-Bera 统计量对各化学成分数据进行正态分布检验（详细结果见附录），发现多个化学成分的数据并不符合正态分布。为了克服数据分布的不正态性，建立出适合该数据的模型，我们使用 spearman 相关系数进行分析。

（1）spearman 相关系数

我们将一列数按照从小到大排序后，这个数所在的位置成为这个数的“等级”。若此时有 X 和 Y 两组数据，则其 spearman 相关系数为

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

其中，d_i 为 X 和 Y 之间的等级差。

（2）计算各化学成分与玻璃类型的 spearman 相关系数

我们计算出了各化学成分与玻璃类型的 spearman 相关系数（详见附录中的**??**），提取出了与玻璃类型相关性较强的 5 个化学成分：氯化钾、二氧化硅、氧化锶、氧化钡和氧化铅。

8.1.2 利用主成分分析法进行降维

考虑到附件所提供的 14 个化学成分，都影响了考古工作者对玻璃文物的类型的认定。为此，我们决定使用主成分分析法来对这 9 个化学成分进行降维处理，从而提取出新的反映玻璃文物理化性质的抽象指标。

（1）主成分分析法

主成分分析是一种把多指标转化为几个综合指标的多元统计分析的方法，主要目的是用较少的变量去解释原来资料中的大部分信息。通常选出的变量要比原始指标的变量少，能解释大部分资料中变异的几个新指标变量，即所谓的主成分，并以此解释资料的综合性指标 [5]。具体的步骤如图 26所示。

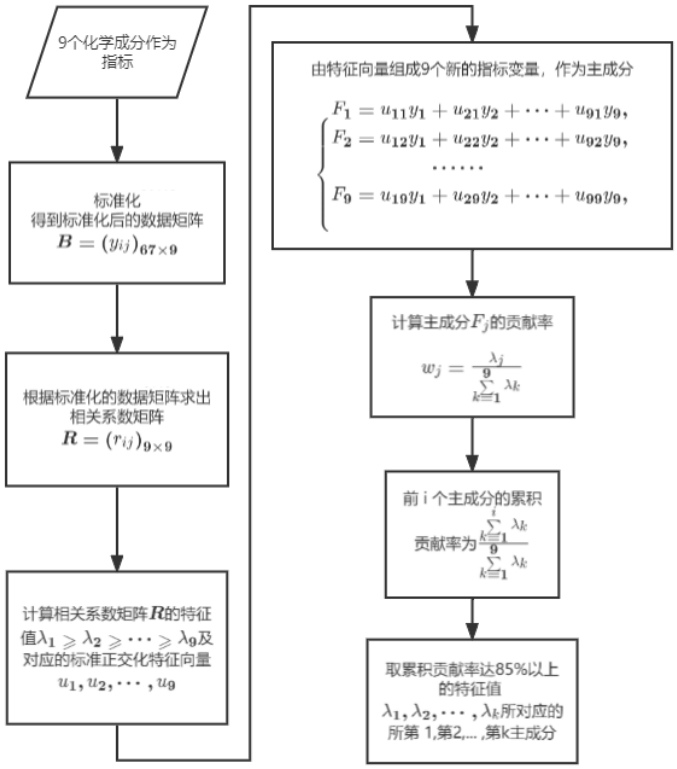


图 **26**　主成分分析法步骤

（2）特征数与累计贡献率的关系图

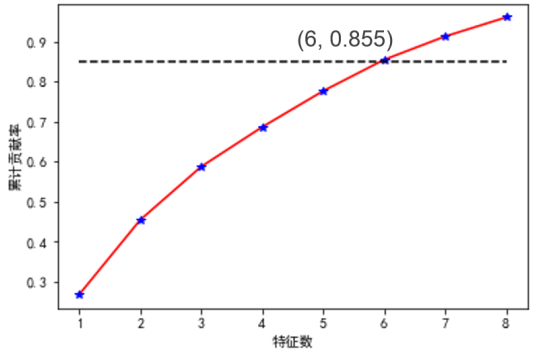


图 **27**　特征数与累计贡献率的关系

根据图 27，我们发现当特征数取 6 时，累计贡献率达 85.5%，大于 85%。所以，我们将特征数取为 6，得到的 6 个主成分作为新的指标，用于下文分析。

8.2 基于 KNN 算法的预测分类模型

在上文中，我们先使用 spearman 相关系数提取出 5 个指标，后对剩下的化学成分使用主成分分析法提取出 6 个指标。由于主成分分析法的使用是在 spearman 相关系数

法之后，降维的原始指标并未被 spearman 相关系数法选中，因此，在将这两组指标合为一组之前，需要对前 5 个指标赋予更大的权重，以体现其在 spearman 相关系数法中更优秀的表现。因此，我们将前 5 个指标赋予权重 0.7，将后 6 个指标赋予权重 0.3。

KNN 算法（全称为 K 近邻算法）是模式识别领域中一种用于分类和回归的非参数统计方法。KNN 对新数据点的预测结果，是通过在整个训练集上搜索与该数据点最相似的 K 个实例，并总结这 K 个实例的输出变量而得出的。

我们需要预测表单 3 给出的文物属于高钾玻璃还是铅钡玻璃，因此，K 近邻算法的类别数是 2。结合上文给出的权重，我们将这 11 个指标的样本集进行赋权，得到赋权后的样本集 D。具体算法的步骤如 algorithm 2。

Algorithm 2 KNN 算法

Input: 需要识别的样本 x; 赋权后的样本集 $D = \{x_1, \cdots, x_n\}$; 参数 K;

Output: $j = \arg\max_{1,2} k_i$;

1: 计算 x 与 D 中每个样本的距离;

2: 寻找与 x 距离最近的前 K 个样本, 统计其中属于各个类别的样本数 $k_i, i = 1, 2$;

在 KNN 算法中，参数 K 的选择对识别结果有很大的影响。若 K 值选择过小，算法的性能将接近于最近邻分类算法，无法发挥 KNN 算法的优势；若 K 值选择过大，距离较远的样本也会对分类结果产生作用，这样也会引起分类误差。综合考量本题样本的特征，我们最终选择将 K 设为合适的 5。

8.3 玻璃文物类型的预测结果以及合理性分析

表 **13**　未知类别玻璃文物所属类型鉴别结果

文物编号	表面风化	二氧化硅	氧化钠	氧化钾	氧化钙	氧化镁	氧化铝	氧化铁	氧化铜	氧化铅	氧化钡	五氧化二磷	氧化锶	氧化锡	二氧化硫	鉴别结果
A1	无风化	78.45			6.08	1.86	7.23	2.15	2.11			1.06	0.03		0.51	高钾玻璃
A2	风化	37.75			7.63		2.33			34.3		14.27				铅钡玻璃
A3	无风化	31.95		1.36	7.19	0.81	2.93	7.06	0.21	39.58	4.69	2.68	0.52			铅钡玻璃
A4	无风化	35.47		0.79	2.89	1.05	7.07	6.45	0.96	24.28	8.31	8.45	0.28			铅钡玻璃
A5	风化	64.29	1.2	0.37	1.64	2.34	12.75	0.81	0.94	12.23	2.16	0.19	0.21	0.49		铅钡玻璃
A6	风化	93.17		1.35	0.64	0.21	1.52	0.27	1.73			0.21				高钾玻璃
A7	风化	90.83		0.98	1.12		5.06	0.24	1.17			0.13			0.11	高钾玻璃
A8	无风化	51.12	0.00	0.23	0.89	0.00	2.12	0.00	9.01	21.24	11.34	1.46	0.31	0.00	2.26	铅钡玻璃

首先，编号为 A2，A3，A4，A5，A8 的文物都被预测为铅钡玻璃，它们氧化铅含量的最小值为 12.23%，由上文的表 6可以看出，对高钾玻璃而言，无论是风化组还是无风化组，氧化铅的含量都远远小于 12.23%，因此，A2，A3，A4，A5，A8 被预测为铅钡玻璃的结果是合理的。

其次，编号为 A6，A7 的文物被预测为高钾玻璃。它们的共同特点都是表面风化，且二氧化硅分别高达 93.17% 和 90.83%。结合表 6可以看出，高钾玻璃在风化后的二氧化硅含量较高，平均值为 93.96%，这恰好符合了 A6，A7 的特点。因此，A6 和 A7 被预测为高钾玻璃的结果是合理的。

最后，编号为 A1 的文物被预测为高钾玻璃。A1 文物的特点是表面无风化。结合表 6和表 5，对比分析二氧化硅、氧化钙、氧化镁、氧化铝、氧化铁、氧化锶这些化学成分，发现 A1 的各个化学成分含量更接近高钾玻璃无风化组的表现。因此，A1 被预测为高钾玻璃的结果是合理的。

8.4 敏感性分析

8.4.1 数据的缩放更新

在进行敏感性分析时，每改变一次某化学成分含量，就需要对数据进行缩放更新，使得文物的其各化学成分比例的累加和是 100%，降低数据偏离程度，更有现实意义。具体步骤如 algorithm 3：

Algorithm 3 改变化学成分含量后的数据更新

Input: 附件表单 3 中的原始数据；要增加含量的化学成分 k；增加的幅度 i；

Output: 缩放后的 11 个指标；

1: 将化学成分 k 的含量乘以系数 $(1+i)$；

2: 对化学成分含量数据进行缩放：每个数据除以所在样本的化学成分含量之和；

3: 根据 8.1 中的方法选取 11 个指标；

8.4.2 含量较高的化学成分含量变化对分类结果的影响

（1）二氧化硅、氧化铅和氧化铝含量变化对分类结果的影响

由于在高钾玻璃和铅钡玻璃中，二氧化硅、氧化铅和氧化铝的含量高于其他大部分化学成分的含量，因此，我们首先对其进行敏感性分析，以原含量的 10% 为增量或减少量，不断提高化学成分的相对含量，观察并对比相应的分类结果。

表 **14** 化学成分含量变化对预测结果的影响

化学成分	增幅	变化情况	化学成分含量
二氧化硅	30%	A5(铅钡 → 高钾)	66.69%
氧化铅	-20%	A5(铅钡 → 高钾)	10.07%
氧化铝	-30%	A5(铅钡 → 高钾)	9.32%

由表 15可知，当二氧化硅的含量增加了 10%、氧化铅含量减少 20%、氧化铝含量减少 30% 之后，文物 A5 的预测结果均由原来的铅钡变为高钾，并且此时三种化学成分的含量均在可接受范围内，说明分类结果对二氧化硅、氧化铅和氧化铝的含量变化具有敏感性。

（2）氧化钡、氧化钾含量变化对分类结果的影响

与上述方法类似，我们依次将氧化钡和氧化钾化学含量以 10% 的幅度逐渐减小，观察分类结果的变化。结果显示，在对附件表单 3 的 8 个样本的预测中，即使二者的含量减少到 0，分类结果也没有明显变化，这说明在本模型中分类结果对氧化钡、氧化钾的含量变化不敏感。

8.4.3 含量较高的化学成分含量变化对分类结果的影响

（1）二氧化硅、氧化铅和氧化铝含量变化对分类结果的影响

由于在高钾玻璃和铅钡玻璃中，二氧化硅、氧化铅和氧化铝的含量高于其他大部分化学成分的含量，因此，我们首先对其进行敏感性分析，以原含量的 10% 为增量或减少量，不断提高化学成分的相对含量，观察并对比相应的分类结果。

表 **15**　化学成分含量变化对预测结果的影响

化学成分	增幅	变化情况	化学成分含量
二氧化硅	30%	A5(铅钡 → 高钾)	66.69%
氧化铅	-20%	A5(铅钡 → 高钾)	10.07%
氧化铝	-30%	A5(铅钡 → 高钾)	9.32%

由表 15可知，当二氧化硅的含量增加了 10%、氧化铅含量减少 20%、氧化铝含量减少 30% 之后，文物 A5 的预测结果均由原来的铅钡变为高钾，并且此时三种化学成分的含量均在可接受范围内，说明分类结果对二氧化硅、氧化铅和氧化铝的含量变化具有敏感性。

（2）氧化钡、氧化钾含量变化对分类结果的影响

与上述方法类似，我们依次将氧化钡和氧化钾化学含量以 10% 的幅度逐渐减小，观察分类结果的变化。结果显示，在对附件表单 3 的 8 个样本的预测中，即使二者的含量减少到 0，分类结果也没有明显变化，这说明在本模型中分类结果对氧化钡、氧化钾的含量变化不敏感。

8.4.4 其它化学成分含量变化对分类结果的影响

由于其它化学成分含量相对较小，且不同样本之间差异较大，只有对化学成分改变足够大的量才能对分类结果产生影响，下面给出部分化学成分含量增加导致分类结果发生变化时的相关信息。

表 **16**　单变量变化下预测结果首次变化相关信息

化学成分	增幅	变化情况	化学成分含量	实际最大值
氧化钙	8000%	A7(高钾 → 铅钡)	42.11%	8.7%
五氧化二磷	4000%	A1(高钾 → 铅钡)	30.63%	14.13%

由表 16可知，氧化钙的增幅达到 8000%，缩放后的含量达到 42.11% 后，预测结果才发生变化，但此时已经远超题目附件表单 2 中氧化钙含量最大值, 不具有实际性。同理，五氧化二磷和其他含量较低的化学成分，不具有敏感性。

综合上述分析可以看出，对该基于 KNN 算法的预测分类模型的分类效果而言，二氧化硅、氧化铅和氧化铝的含量变化具有敏感性，而其他化学成分不具备敏感性。

九、 问题四模型建立与求解

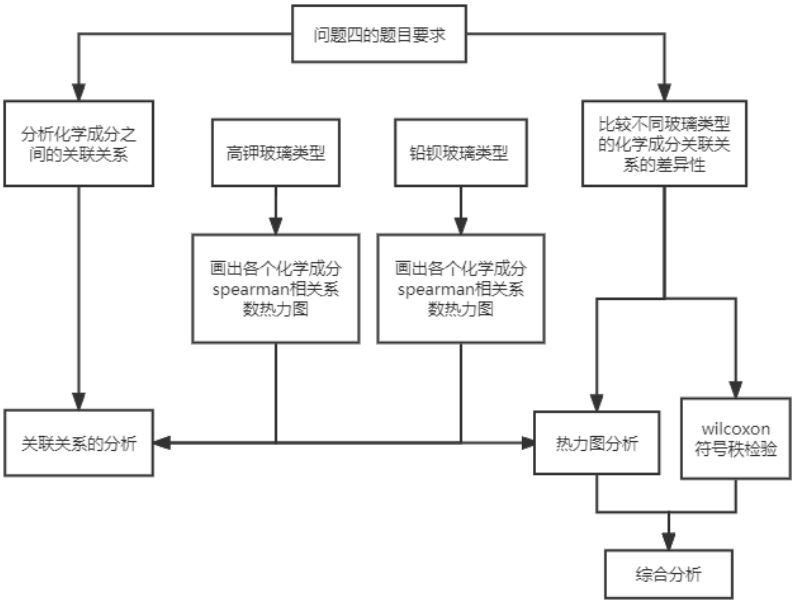


图 **28**　问题四分析思路框图

9.1 不同化学成分之间的关联关系

由于多个化学成分的数据正态性不明显（详细正态检验结果见附录）。所以我们采用 spearman 相关系数来表示关联关系，并绘制热力图进行直观展示。

9.1.1 高钾玻璃不同化学成分之间的关联关系

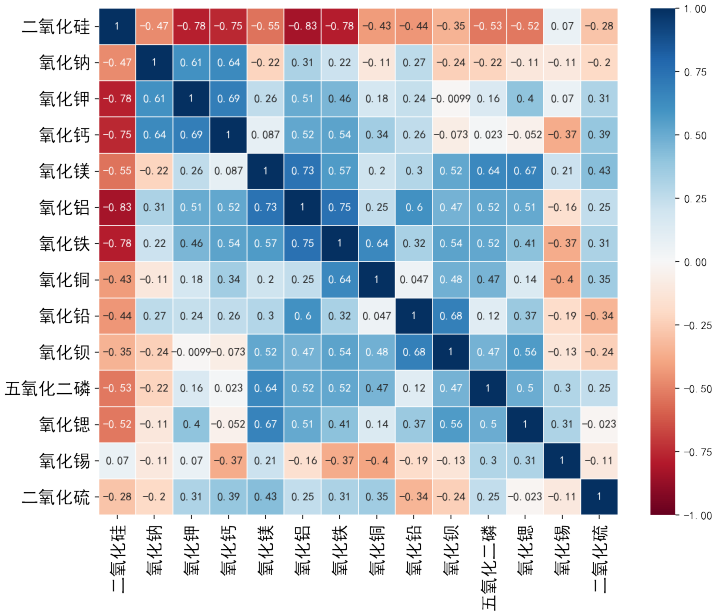


图 **29**　高钾玻璃各化学成分 **spearman** 相关系数热力图

(1) SiO_2 与其它大部分化学成分的关系

SiO_2 与其它大部分化学成分都呈负相关。从表 6（高钾玻璃风化前后化学成分含量平均变化率）可以看出，高钾玻璃风化前后 SiO_2 含量出现了上升而其它大部分化学成分含量均出现了下降。这也从侧面验证了问题 1 分析结果的正确性。

SiO_2 与其它化学成分的相关系数存在差异性。一方面，SiO_2 与 K_2O、CaO、Al_2O_3、Fe_2O_3 的相关性较高，从表 6（高钾玻璃风化前后化学成分含量平均变化率）中可以看出，这是因为风化使高钾玻璃的 SiO_2 含量上升，同时这几种化学成分的含量出现了大幅下降；另一方面，SiO_2 与 Na_2O、PbO、BaO、SrO、SnO、SO_2 的相关性较低，这是因为这几种化学成分的的样本数据大部分为 0，相较其它成分而言无法体现出与 SiO_2 的相关性。

(2) 铁铝氧化物与其他金属氧化物的关系

Al_2O_3 和 Fe_2O_3 与多数含量较高的金属氧化物的相关性都较高。由表 6可知，当 SiO_2 含量上升之后，K_2O，CaO，Al_2O_3，Fe_2O_3 等化学成分的下降是同步的。

9.1.2 铅钡玻璃不同化学成分之间的关联关系

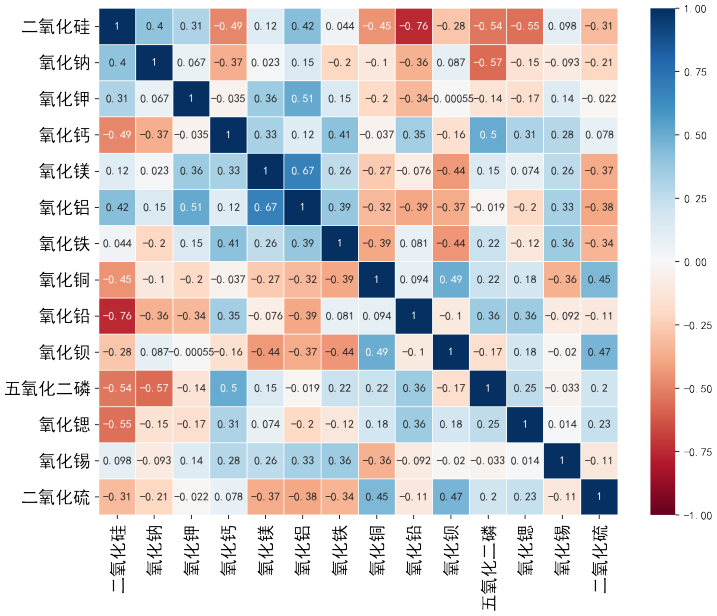


图 **30**　铅钡玻璃各化学成分 **spearman** 相关系数热力图

(1) SiO_2 与其它大部分化学成分的关系

SiO_2 与其它化学成分之间既有正相关，又有负相关，但负相关指标的相关性更大。其中，SiO_2 和 PbO 高度负相关。由图 5（铅钡玻璃风化前后化学成分含量平均变化）可知，在铅钡玻璃中 PbO 与 SiO_2 的含量都较高，合计占比达到 70% 左右；同时，由表 5（铅钡玻璃风化前后化学成分含量平均变化率可知），风化后，SiO_2 的含量下降了 54.42%，同时 PbO 的含量上升了 96.13%。因此，PbO 与 SiO_2 呈现高度负相关关系。

(2) MgO 与 Al_2O_3 的关系

同样由表 5可知，风化后 MgO 的含量下降了 81.78%，同时 Al_2O_3 的含量下降了 70.85%，符合正相关关系。

9.2 不同玻璃类型之间的化学成分关联关系的差异性

9.2.1 根据热力图进行分析

(1) 从整体分析

高钾玻璃的各化学成分之间相关性比铅钡玻璃高，可以推测高钾玻璃各化学成分之间可能存在更多的化学反应。

(2) SiO_2 与其它化学成分的关系

高钾玻璃中，SiO_2 与其它成分主要呈负相关且相关性较高；铅钡玻璃中，SiO_2 与其它化学成分之间正负相关均有，且相关程度较低。由表 6和表 5可知，高钾玻璃中 SiO_2 含量变化后，其它成分含量朝相反方向变化，且幅度较小；而铅钡玻璃中 SiO_2 含量变化后，其它化学成分含量有增有减，且幅度较大。

(3) Al_2O_3 与 P_2O_5 的关系

高钾玻璃中 P_2O_5 与 Al_2O_3 呈正相关且相关性较大，铅钡玻璃中 P_2O_5 与 Al_2O_3 呈负相关且相关性较小。这说明在高钾玻璃中，P_2O_5 与 Al_2O_3 的含量变化是同步的，而在铅钡玻璃中，P_2O_5 与 Al_2O_3 的含量变化是相反的。这验证了我们在问题二的铅钡玻璃亚类划分中得出的结果：风化会使得高铝低磷的铅钡亚类玻璃逐渐转化为高磷低铝的铅钡亚类玻璃。

(4) Al_2O_3 和 CaO 的关系

两类玻璃中 Al_2O_3 和 CaO 的含量都呈正相关，其中高钾玻璃相关性明显更强，验证了我们在问题二的高钾玻璃亚类划分中得出的结果：风化会使得高钙铝的高钾亚类玻璃逐渐转化为低钙铝的高钾亚类玻璃。

9.2.2 Wilcoxon 符号秩检验

为从宏观上进一步探究不同玻璃类别之间的化学成分关联关系的差异性，我们采用 Wilcoxon 符号秩检验的方法，依次对两类玻璃相关系数矩阵中同一化学成分与其它化学成分的关系进行配对检验，来观察两类玻璃中同一化学成分与其它化学成分的关系是否具有差异性。Wilcoxon 符号秩检验具体步骤如下:

Step 1: 求出成对观测数据的差 di ，并将 di 的绝对值按大小顺序编上等级。

Step 2: 等级编号完成以后恢复正负号，分别求出正等级之和 $T+$ 和负等级之和 $T-$，选择 $T+$ 和 $T-$ 中较小的一个作为威尔科克森检验统计量 T 。

Step 3: 作出判断。
根据显著性水平 α 查附表，得到临界值 $T\alpha$ ，若 $T < T\alpha$ ，则拒绝原假设 H_0 。

Step 4: 得出结果，绘制表格（如表 17 ）。

表 **17**　斯皮尔曼相关系数的 **wilcoxon** 符号秩检验

化学成分	统计量	P 值	是否有差异 (90%)
二氧化硅	20	0.080613	是
氧化钠	29	0.263494	无
氧化钾	23	0.124175	无
氧化钙	39	0.674987	无
氧化镁	24	0.142213	无
氧化铝	20	0.080613	是
氧化铁	21	0.093492	是
氧化铜	25	0.162199	无
氧化铅	20	0.080613	是
氧化钡	26	0.184235	无
五氧化二磷	13	0.025329	是
氧化锶	18	0.059172	是
氧化锡	20	0.080613	是
二氧化硫	30	0.294507	无

由表 17 可知，在 90% 的置信水平下，两类玻璃中 SiO_2、Al_2O_3 等 7 种化学成分分别与其它化学成分的关系也具备差异性：

(1) SiO_2 与其他化学成分相关性的差异性

在受风化后，高钾玻璃的 SiO_2 含量上升，铅钡玻璃的 SiO_2 含量下降。

(2) P_2O_5 和 PbO 与其他化学成分相关性的差异性

在铅钡玻璃风化前后，P_2O_5 和 PbO 的含量会上升；而在高钾玻璃中 P_2O_5 和 PbO 含量较低，风化前后变化不明显。

十、 模型和方法的评价

10.1 风化检测点风化前化学成分含量的均值线性映射预测法

（1）优点

- 使用均值和线性关系进行预测，简洁有效，预测结果具有普适性。

- 对于某些化学成分从有到无的情况做了特殊处理，使预测方法更为全面。

（2）缺点

- 单个指标的线性映射预测方式较为简陋，未能充分考虑各化学成分含量之间的关系。
- 在线性映射时，各化学成分的权重相同，但可以根据化学成分的重要性，赋予不同的权重，再进行预测。

10.2 模拟分类判断逻辑的决策树与随机森林模型

（1）优点

- 决策树的判断方式较接近于人的判断方式，更能模拟考古工作者对于样品的分类规律，十分贴合题意。
- 通过随机森林可以筛选出对于分类判断较为重要的化学成分，从而可以排除其他干扰，更好地说明分类规律。

（2）缺点

- 无论是决策树还是随机森林，对特征的选择都具有随机性，这很可能会导致结果的不稳定性。
- 分类规律主要强调了化学成分的影响，并没有很好地将颜色、纹饰等离散类数据与化学成分结合起来判断分类。

10.3 基于轮廓系数确定最优聚类簇数的 K-means 模型

（1）优点

- 在选择化学成分时依据“大标准差大均值”和“定类成分回避”准则，充分体现同类样品中的个体差异，从而使得划分出的亚类有可区分性。
- 利用轮廓系数评估 K-means 聚类模型，并画出了不同聚类簇数下轮廓系数的直观展示图，使亚类划分数目的选择更有说服力

（2）缺点

- 聚类时依据的化学成分含量相差较大，但它们在聚类中的权重相同，这可能让含量较少但对于亚类划分较为重要的化学成分发挥不出原有的作用。
- 在分析亚类划分敏感性时只让单一化学成分发生波动，不能判断亚类划分对于多个化学成分联合波动的敏感性。

10.4 基于 KNN 算法的预测分类模型

（1）优点

- 使用 spearman 相关系数提取的指标与主成分分析得到的指标分别具有不同的权重，符合指标提取的实际情况，客观严谨。
- KNN 算法思想简单，适用于小样本，精度高，无需估计参数，对异常值不敏感。这些特点适用于本题样本数据。

（2）缺点

- 使用主成分分析法得出的指标不具备解释性。

10.5 化学成分间相关性及其在两类玻璃中差异性的方法分析

（1）优点

- 考虑了样本的分布规律，结合 J-B 检验确立了 spearman 相关系数这一较为合理的评价指标。
- 利用 Wilconxon 符号秩检验得出不同类别玻璃的化学成分关联关系的差异性，既符合样本分布特征的要求，又充分利用了样本信息。

（2）缺点

- 由于数据量少，存在偶然性，某些相关性指标解释性较差，两类玻璃中同一化学成分与其它化学成分关系的差异难以通过现实数据进行解释。

十一、 参考文献

[1] 张福康, 程朱海, 张志刚. 中国古琉璃的研究 [J]. 硅酸盐学报,1983,(01):67-76.

[2] 王承遇, 陶瑛. 硅酸盐玻璃的风化 [J]. 硅酸盐学报,2003,(01):78-85.

[3] 王承遇, 陶瑛, 陈敏, 黄明. 钠钙铝镁硅酸盐玻璃和碱铅硅酸盐玻璃的风化 [J]. 硅酸盐通报,1989,(06):1-9.DOI:10.16552/j.cnki.issn1001-1625.1989.06.001

[4] 段浩, 干福熹, 赵虹霞. 实验室模拟过渡金属离子掺杂的中国古代玻璃的着色现象 [J]. 硅酸盐学报,2009,37(12):1982-1989.

[5] 司守奎, 孙玺菁.Python 数学实验与建模. 北京：科学出版社,2020.

附录 A　支撑材料文件列表

文件夹	文件名	主要内容/用途
源代码	data_preprocess.py	随机森林预测缺失的颜色
	q1.py	有无风化成分含量统计处理、风化前含量预测
	q2_1.py	决策树随机森林探索分类规律
	q2_2.py	亚类划分
	q3.py	指标筛选、KNN 算法预测、敏感度分析
	q4.py	计算相关系数、Wilcoxon 符号秩检验
数据	附件.xlsx	题目附件
	文物样品信息汇总.xlsx	删除无效数据，合并题目附件的表单 1 和表单 2
	文物样品信息汇总-填补缺失值.xlsx	填补颜色缺失，填补空白单元格
	表 1-填补缺失值.xlsx	填补附件表单 1 中缺失的颜色
	高钾风化点预测结果.xlsx	高钾类样品风化点检测点的风化前预测结果
	铅钡风化点预测结果.xlsx	铅钡类样品风化点检测点的风化前预测结果
	高钾亚类划分结果.xlsx	高钾类样品亚类划分的结果
	铅钡亚类划分结果.xlsx	铅钡类样品亚类划分的结果
	附件表单 3 预测结果.xlsx	附件表单 3 玻璃样品的预测结果

表 **19**　高钾玻璃风化点在风化前化学成分含量的预测结果

文物编号	二氧化硅	氧化钠	氧化钾	氧化钙	氧化镁	氧化铝	氧化铁	氧化铜	氧化铅	氧化钡	五氧化二磷	氧化锶	氧化锡	二氧化硫
7	65.57218	0.679991	9.129332	6.416733	1.055862	6.644839	1.212422	4.978337	0.402777	0.585412	2.989463	0.040767	0.19242	0.099471
9	70.98263	0.717584	10.4615	3.923657	1.114234	4.674792	2.408374	2.513275	0.425044	0.617776	1.810092	0.043021	0.203057	0.10497
10	71.7212	0.711938	16.1845	1.318525	1.105468	2.846054	1.941409	1.351318	0.4217	0.612916	1.436681	0.042682	0.20146	0.104144
12	65.87079	0.671062	16.74763	4.261103	1.041997	4.835389	2.04109	2.501972	0.397488	0.577725	0.725461	0.040232	0.189893	0.098165
22	59.80279	0.622042	11.3742	9.106571	3.143207	10.74494	2.28344	0.77307	0.368452	0.535523	0.941454	0.037293	0.176022	0.090994
27	66.11007	0.684904	9.195286	5.677855	2.920091	8.484363	1.436684	2.383341	0.405686	0.589641	1.777019	0.041061	0.19381	0.10019

附录 B　题目完整答案表

问题一

表 **18**　铅钡玻璃风化点在风化前化学成分含量的预测结果

文物编号	二氧化硅	氧化钠	氧化钾	氧化钙	氧化镁	氧化铝	氧化铁	氧化铜	氧化铅	氧化钡	五氧化二磷	氧化锶	氧化锡	二氧化硫
2	60.93146	1.287988	1.31705	0.877487	0.889964	6.580829	1.79373	0.12521	18.51179	6.890571	0.543268	0.093236	0.035611	0.12181
8	45.79947	1.743968	0.226671	0.751473	0.66379	2.083807	0.76338	6.788001	15.15657	24.67761	0.739719	0.245844	0.048218	0.311479
8	16.28066	2.708374	0.352019	2.515431	1.030862	2.680684	1.185526	3.179734	26.63217	37.57563	2.419159	0.546894	0.074883	2.817981
11	65.56193	1.496854	0.306126	1.529676	0.622323	3.590424	0.655212	2.759172	11.51663	9.90882	1.658884	0.211008	0.041386	0.141563
19	61.24403	1.584613	0.205959	1.351773	0.547462	5.044353	1.578	2.079615	20.56142	3.841222	1.653171	0.114708	0.043812	0.149863
26	12.92014	2.663557	1.037583	2.334219	1.013804	2.802581	1.165909	3.585231	24.14943	42.78295	1.900785	0.629176	0.073644	2.940988
26	45.26083	1.753938	0.227967	0.735343	0.667584	1.094779	0.767745	6.931734	15.69499	25.62928	0.648623	0.300708	0.048494	0.23798
34	66.88876	1.433674	0.349053	0.32558	0.545685	2.070997	0.504522	0.809431	20.22336	6.495952	0.057592	0.120169	0.039639	0.135588
36	62.81379	12.50307	0.16598	0.131142	0.46336	1.736843	0.291681	0.309519	15.34997	5.973755	0.010068	0.102039	0.033659	0.115132
38	59.20433	8.802691	0.179208	0.272974	0.524797	3.159705	0.299385	0.376335	20.60239	6.116098	0.078194	0.215378	0.038122	0.130398
39	57.49643	1.679768	0.218326	0.542857	0.639353	0.748916	0.735278	0.552694	31.06532	5.495141	0.230219	0.390389	0.046443	0.158862
40	43.4601	1.994582	0.259244	1.085942	0.759178	0.800347	0.283751	1.697199	42.43597	6.046032	0.417118	0.516749	0.055147	0.188635
41	46.38194	1.926881	0.825674	2.782592	3.08032	5.721542	2.582498	0.136887	25.76164	8.521127	1.698351	0.345041	0.053276	0.182232
43	34.90267	2.156873	0.280338	3.290553	1.124069	4.327341	1.227355	4.314504	39.11758	7.124337	1.344841	0.525924	0.059634	0.203983
43	53.04185	1.874548	0.243643	3.492928	1.042794	5.699871	1.950938	1.058342	25.41984	2.768896	2.841562	0.33567	0.051829	0.177283
48	70.96584	3.77696	0.318029	0.837872	0.920268	12.42116	0.787018	0.868353	4.858193	3.380066	0.13263	0.097202	0.539899	0.096513
49	59.86804	1.594744	0.207276	2.12652	1.372735	7.650456	3.271702	0.41739	16.51757	4.407712	2.091452	0.279491	0.044092	0.150821
50	47.84119	2.040563	0.26522	1.895194	0.561599	3.402558	0.504192	0.862146	27.20732	13.12897	1.528527	0.513112	0.056419	0.192983
51	54.68829	1.704199	0.221502	1.7763	1.187533	7.977994	1.518446	0.872958	20.78077	6.903197	1.630945	0.253223	0.323478	0.161172
51	49.38878	1.774059	0.230582	2.649711	1.50631	3.970597	0.557891	0.497487	27.59988	9.490986	1.834045	0.282841	0.04905	0.167779
52	54.98832	9.246905	0.21294	1.082776	0.527644	1.694617	0.282137	0.428795	23.54205	6.413656	1.105273	0.274644	0.045297	0.154942
54	50.57393	1.740801	0.5425	1.616787	1.304781	6.44186	0.761994	0.540231	29.25582	5.552827	0.872065	0.583647	0.048131	0.164634
54	41.04839	1.839856	0.239134	1.443835	1.195873	5.988123	0.805353	0.921809	32.59311	9.84299	3.071566	0.785092	0.050869	0.174002
56	60.97455	1.60416	0.208499	0.565127	0.610576	2.646266	0.702183	0.473835	20.05187	11.22971	0.48141	0.255754	0.044353	0.151711
57	55.00201	1.65936	0.215674	0.632885	0.631586	3.225605	0.726345	0.719699	22.67777	13.00706	1.034635	0.264554	0.045879	0.156932
58	61.8588	1.56102	0.516878	1.58616	0.722128	4.899651	1.005169	1.82686	18.61386	5.417883	1.658067	0.142738	0.04316	0.147631

问题二

表 **20**　铅钡亚类划分结果

文物编号	文物采样点	纹饰	颜色	表面风化	亚类标号
2	02	A	浅蓝	风化	1
8	08	C	紫	风化	1
8	08 严重风化点	C	紫	风化	1
11	11	C	浅蓝	风化	1
19	19	A	黑	风化	1
20	20	A	浅蓝	无风化	1
23	23 未风化点	A	蓝绿	风化	0
24	24	C	紫	无风化	1
25	25 未风化点	C	浅蓝	风化	0
26	26 严重风化点	C	紫	风化	1
26	26	C	紫	风化	1
28	28 未风化点	A	浅蓝	风化	0
29	29 未风化点	A	浅蓝	风化	0
30	30 部位 1	A	深蓝	无风化	1
30	30 部位 2	A	深蓝	无风化	1
31	31	C	紫	无风化	0
32	32	C	浅绿	无风化	0
33	33	C	深绿	无风化	0
34	34	C	深绿	风化	1
35	35	C	浅绿	无风化	0
36	36	C	深绿	风化	1
37	37	C	深绿	无风化	0
38	38	C	深绿	风化	1
39	39	C	深绿	风化	1
40	40	C	深绿	风化	1
41	41	C	浅绿	风化	1
42	42 未风化点 2	A	浅蓝	风化	0
42	42 未风化点 1	A	浅蓝	风化	0
43	43 部位 1	C	浅蓝	风化	1
43	43 部位 2	C	浅蓝	风化	1
44	44 未风化点	A	浅蓝	风化	0

接下页

表 20 – 接上页

文物编号	文物采样点	纹饰	颜色	表面风化	亚类标号
45	45	A	浅蓝	无风化	0
46	46	A	浅蓝	无风化	0
47	47	A	浅蓝	无风化	0
48	48	A	浅蓝	风化	0
49	49 未风化点	A	黑	风化	0
49	49	A	黑	风化	1
50	50	A	黑	风化	1
50	50 未风化点	A	黑	风化	0
51	51 部位 1	C	浅蓝	风化	1
51	51 部位 2	C	浅蓝	风化	1
52	52	C	浅蓝	风化	1
53	53 未风化点	A	浅蓝	风化	0
54	54	C	浅蓝	风化	1
54	54 严重风化点	C	浅蓝	风化	1
55	55	C	绿	无风化	0
56	56	C	蓝绿	风化	1
57	57	C	蓝绿	风化	1
58	58	C	浅蓝	风化	1

注：亚类 0 是高铝低磷的铅钡亚类玻璃，亚类 1 是高磷低铝的铅钡亚类玻璃

表 **21**　高钾亚类划分结果

文物编号	文物采样点	纹饰	颜色	表面风化	亚类标号
1	01	C	蓝绿	无风化	1
3	03 部位 2	A	蓝绿	无风化	1
3	03 部位 1	A	蓝绿	无风化	1
4	04	A	蓝绿	无风化	1
5	05	A	蓝绿	无风化	1
6	06 部位 2	A	蓝绿	无风化	1
6	06 部位 1	A	蓝绿	无风化	1

接下页

表 21 – 接上页

文物编号	文物采样点	纹饰	颜色	表面风化	亚类标号
7	07	B	蓝绿	风化	0
9	09	B	蓝绿	风化	0
10	10	B	蓝绿	风化	0
12	12	B	蓝绿	风化	0
13	13	C	浅蓝	无风化	1
14	14	C	深绿	无风化	1
16	16	C	浅蓝	无风化	1
18	18	A	深蓝	无风化	0
21	21	A	蓝绿	无风化	1
22	22	B	蓝绿	风化	0
27	27	B	蓝绿	风化	0

注：亚类 0 是低钙铝的高钾亚类玻璃，亚类 1 是高钙铝的高钾亚类玻璃

问题三

表 **22**　所有玻璃样本数据的正态分布检验

	统计量	P 值	是否通过正态分布检验
二氧化硅	2.35874868	0.307471051	是
氧化钠	150.5145726	0	否
氧化钾	67.99002366	1.77636E-15	否
氧化钙	11.536452	0.003125297	否
氧化镁	4.140841209	0.126132719	是
氧化铝	45.50154701	1.31663E-10	否
氧化铁	117.6121116	0	否
氧化铜	119.225825	0	否
氧化铅	3.361493546	0.186234849	是
氧化钡	46.08295013	9.84497E-11	否
五氧化二磷	27.34871228	1.1516E-06	否
氧化锶	13.01205265	0.001494406	否

接下页

表 22 – 接上页

	统计量	P 值	是否通过正态分布检验
氧化锡	3208.090699	0	否
二氧化硫	2145.024438	0	否

注：在 95% 的置信水平下

表 **23**　各化学成分与玻璃类型的 **spearman** 相关系数

	spearman 相关系数	P 值	是否显著
氧化铅	0.769526944	2.77808E-14	是
氧化钡	0.714478322	1.12556E-11	是
二氧化硅	-0.670339743	5.47293E-10	是
氧化锶	0.588083666	1.6693E-07	是
氧化钾	-0.583212628	2.23069E-07	是
氧化铁	-0.288781393	0.017786704	否
氧化铝	-0.241148193	0.049315527	否
氧化铜	-0.22727802	0.064367271	否
氧化钙	-0.193342028	0.116969442	否
五氧化二磷	0.156964545	0.204615417	否
氧化钠	0.117289257	0.344528893	否
氧化镁	-0.094775676	0.445527388	否
二氧化硫	-0.064922237	0.601696794	否
氧化锡	0.063271581	0.610987757	否

注：在 99% 的置信水平下

附录 C　程序代码

数据预处理

data_preprocess.py:

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder

df1 = pd.read_excel('附件.xlsx', sheet_name=0)
df2_valid = pd.read_excel('文物样品信息汇总.xlsx')
print('缺失值比例情况: \n', df1.isnull().sum() / df1.shape[0])
filter_df1 = df2_valid[(df2_valid.纹饰 == 'A') & (df2_valid.类型 == '铅钡') &
    (df2_valid.表面风化 == '风化')].drop(
['文物编号', '文物采样点'], axis=1)
filter_df2 = df2_valid[(df2_valid.纹饰 == 'C') & (df2_valid.类型 == '铅钡') &
    (df2_valid.表面风化 == '风化')].drop(
['文物编号', '文物采样点'], axis=1)
X_train1 = filter_df1.drop(['纹饰', '类型', '颜色', '表面风化'],
    axis=1).fillna(0)[~filter_df1.颜色.isna().values]
Y_train1 = filter_df1.颜色.dropna()
X_pre1 = filter_df1.drop(['纹饰', '类型', '颜色', '表面风化'],
    axis=1).fillna(0)[filter_df1.颜色.isna().values]
X_train2 = filter_df2.drop(['纹饰', '类型', '颜色', '表面风化'],
    axis=1).fillna(0)[~filter_df2.颜色.isna().values]
Y_train2 = filter_df2.颜色.dropna()
X_pre2 = filter_df2.drop(['纹饰', '类型', '颜色', '表面风化'],
    axis=1).fillna(0)[filter_df2.颜色.isna().values]
le1 = LabelEncoder().fit(Y_train1)
rfr1 = RandomForestRegressor(random_state=2022, n_estimators=100)
rfr1 = rfr1.fit(X_train1, le1.transform(Y_train1))
res1 = le1.inverse_transform(rfr1.predict(X_pre1).round().astype('int'))
score1 = rfr1.score(X_train1, le1.transform(Y_train1))
print('铅钡A纹饰颜色预测结果及准确率: ')
print(res1)
print(score1)
le2 = LabelEncoder().fit(Y_train2)
rfr2 = RandomForestRegressor(random_state=2022, n_estimators=100)
rfr2 = rfr2.fit(X_train2, le2.transform(Y_train2))
res2 = le2.inverse_transform(rfr2.predict(X_pre2).round().astype('int'))
score2 = rfr2.score(X_train2, le2.transform(Y_train2))
print('铅钡C纹饰颜色预测结果及准确率: ')
print(res2)
print(score2)
df2_valid.iloc[19, 18] = res1[0]
df2_valid.iloc[42, 18] = res2[0]
```

```
df2_valid.iloc[52, 18] = res1[1]
df2_valid.iloc[66, 18] = res2[1]
df2_valid.fillna(0, inplace=True)
df2_valid.to_excel('文物样品信息汇总-填补缺失值.xlsx', index=False)
```

问题一

q1.py:

```
import pandas as pd
import numpy as np
df1_valid = pd.read_excel('表1-填补缺失值.xlsx')
df2_valid = pd.read_excel('文物样品信息汇总-填补缺失值.xlsx')
from sklearn.feature_selection import chi2
from sklearn.preprocessing import LabelEncoder
rel_y = df1_valid.表面风化
rel_cls = LabelEncoder().fit(df1_valid.类型).transform(df1_valid.类型)
rel_wen = LabelEncoder().fit(df1_valid.纹饰).transform(df1_valid.纹饰)
rel_col = LabelEncoder().fit(df1_valid.颜色).transform(df1_valid.颜色)
rel_X = np.array([rel_cls,rel_wen,rel_col]).T
chi2s, chi2_p_values = chi2(rel_X, rel_y)
print('类型、纹饰、颜色的卡方检验p值: ')
print(chi2_p_values)
# 高钾类风化前后化学成分统计和预测
df2_sta = df2_valid.drop(['文物编号', '纹饰', '颜色'],axis=1)
chemical_component_K = df2_sta[(df2_sta.类型 == '高钾')].drop(['类型'],axis=1)
cck_whe_mean = chemical_component_K[chemical_component_K.表面风化 ==
    '风化'].drop(['文物采样点','表面风化'],axis=1).mean()
cck_no_whe_mean = chemical_component_K[chemical_component_K.表面风化 ==
    '无风化'].drop(['文物采样点','表面风化'],axis=1).mean()
cck_whe_ratio = (cck_whe_mean - cck_no_whe_mean)/cck_no_whe_mean
pd.DataFrame([cck_no_whe_mean,cck_whe_mean,cck_whe_ratio],
index=['无风化化学成分含量占比','风化化学成分含量占比','风化后化学成分含量变化率'])
.to_excel('高钾玻璃风化化学成分含量的变化.xlsx')
cck_whe = chemical_component_K[chemical_component_K.表面风化 ==
    '风化'].drop(['文物采样点','表面风化'],axis=1)
cck_ratio_idx = 0
cck_whe = chemical_component_K[chemical_component_K.表面风化 ==
    '风化'].drop(['文物采样点','表面风化'],axis=1)
cck_whe_pred = np.zeros(cck_whe.shape)
for cck in cck_whe_mean:
cck_ratio = cck_whe_ratio[cck_ratio_idx]
cck_whe_each = cck_whe.iloc[:,cck_ratio_idx]
if cck_ratio == -1:
# 风化后化学成分直接消失了的，按均值预测
cck_whe_pred[:,cck_ratio_idx] = cck_no_whe_mean[cck_ratio_idx]
```

```
        else:
        # 风化后化学成分直接消失了的，按均值预测，没有消失的按比例预测
        tmp = cck_whe.iloc[:,cck_ratio_idx]
        cck_whe_pred[:,cck_ratio_idx] =
            np.where(tmp==0,cck_no_whe_mean[cck_ratio_idx],tmp/(1+cck_ratio))
        cck_ratio_idx += 1
        cck_pre_res = cck_whe_pred / cck_whe_pred.sum(axis=1).reshape(-1,1)
        # 生成预测结果
        k_number = df2_valid.loc[cck_whe.index]['文物编号']
        k_pre_df = pd.DataFrame(cck_pre_res,index=k_number,columns=cck_whe.columns)
        k_pre_df.to_excel('高钾风化点预测结果.xlsx')
        # 检验预测结果
        print('高钾类预测风化前的均值')
        print(k_pre_df.mean())
        print('高钾类无风化样本的均值')
        print(cck_no_whe_mean)
        # 高钾类风化前后化学成分统计和预测
        chemical_component_PB = df2_sta[(df2_sta.类型 == '铅钡')].drop(['类型'], axis=1)
        ccpb_whe = chemical_component_PB[(chemical_component_PB.表面风化 == '风化')]
        ccpb_whe = ccpb_whe.drop([23,25,29,30,44,45,48,53,56,60])
        ccpb_whe_mean = ccpb_whe.drop(['文物采样点', '表面风化'], axis=1).mean()
        ccpb_no_whe = chemical_component_PB[(chemical_component_PB.表面风化 == '无风化')]
        ccpb_no_whe = pd.concat([ccpb_no_whe,
        chemical_component_PB
        .loc[[23,25,29,30,44,45,48,53,56,60]]]).sort_index()
        ccpb_no_whe_mean = ccpb_no_whe.drop(['文物采样点', '表面风化'], axis=1).mean()
        ccpb_whe_ratio = (ccpb_whe_mean - ccpb_no_whe_mean) / ccpb_no_whe_mean
        pd.DataFrame([ccpb_no_whe_mean,ccpb_whe_mean,ccpb_whe_ratio],
        index=['无风化化学成分含量占比','风化化学成分含量占比','风化后化学成分含量变化率'])
        .to_excel('铅钡玻璃风化化学成分含量的变化.xlsx')
        ccpb_whe_to_pre = ccpb_whe.drop(['文物采样点', '表面风化'], axis=1)
        ccpb_ratio_idx = 0
        ccpb_whe_pred = np.zeros(ccpb_whe_to_pre.shape)
        for ccpb in ccpb_whe_mean:
        ccpb_ratio = ccpb_whe_ratio[ccpb_ratio_idx]
        tmp = ccpb_whe_to_pre.iloc[:, ccpb_ratio_idx]
        # 风化后化学成分直接消失了的，按均值预测，没有消失的按比例预测
        ccpb_whe_pred[:, ccpb_ratio_idx] = np.where(tmp == 0,
            ccpb_no_whe_mean[ccpb_ratio_idx], tmp / (1 + ccpb_ratio))
        ccpb_ratio_idx += 1
        ccpb_whe_res = ccpb_whe_pred / ccpb_whe_pred.sum(axis=1).reshape(-1,1)
        # 生成预测结果
        pb_number = df2_valid.loc[ccpb_whe.index]['文物编号']
        pb_pre_df = pd.DataFrame(ccpb_whe_res,index=pb_number,columns=cck_whe.columns)
        pb_pre_df.to_excel('铅钡风化点预测结果.xlsx')
        # 检验预测结果
        print('铅钡类预测风化前的均值')
```

```
print(pb_pre_df.mean())
print('铅钡类无风化样本的均值')
print(ccpb_no_whe_mean)
```

问题二

q2_1.py:

```
import pandas as pd
from sklearn import tree
from sklearn.model_selection import train_test_split
import graphviz

df2_valid = pd.read_excel('文物样品信息汇总-填补缺失值.xlsx')
# 将风化样本的无风化采样点归到无风化组
df_whe = df2_valid[df2_valid.表面风化 == '风化'].drop([23, 25, 29, 30, 44, 45, 48,
    53, 56, 60])
df_no_whe = pd.concat(
[df2_valid[df2_valid.表面风化 == '无风化'], df2_valid.loc[[23, 25, 29, 30, 44, 45,
    48, 53, 56, 60]]]).sort_index()
Y_whe = df_whe.类型
Y_no_whe = df_no_whe.类型
X_whe = df_whe.drop(['文物编号', '文物采样点', '类型', '表面风化', '纹饰', '颜色'],
    axis=1)
X_no_whe = df_no_whe.drop(['文物编号', '文物采样点', '类型', '表面风化', '纹饰',
    '颜色'], axis=1)
Xtrain_whe, Xtest_whe, Ytrain_whe, Ytest_whe = train_test_split(X_whe, Y_whe,
    test_size=0.2, random_state=400)
Xtrain_no_whe, Xtest_no_whe, Ytrain_no_whe, Ytest_no_whe =
    train_test_split(X_no_whe, Y_no_whe, test_size=0.2,
random_state=2022)
print('决策权拟合，有风化：')
clf_whe = tree.DecisionTreeClassifier(random_state=400).fit(Xtrain_whe, Ytrain_whe)
print('测试集准确率为：')
print(clf_whe.score(Xtest_whe, Ytest_whe))
feature_name = ['二氧化硅', '氧化钠', '氧化钾', '氧化钙', '氧化镁', '氧化铝',
    '氧化铁', '氧化铜', '氧化铅', '氧化钡', '五氧化二磷', '氧化锶', '氧化锡',
    '二氧化硫']
dot_data = tree.export_graphviz(clf_whe
, feature_names=feature_name
, class_names=["铅钡", "高钾"]
, filled=True
, rounded=True
)
graph = graphviz.Source(dot_data) # 可以通过ipython进行展示
print('决策树的可视化可以通过ipython展示')
```

```
print('*' * 30)
print('决策权拟合，无风化：')
clf_no_whe = tree.DecisionTreeClassifier(random_state=1000).fit(Xtrain_no_whe,
    Ytrain_no_whe)
print('测试集准确率为：')
print(clf_no_whe.score(Xtest_no_whe, Ytest_no_whe))
feature_name = ['二氧化硅', '氧化钠', '氧化钾', '氧化钙', '氧化镁', '氧化铝',
    '氧化铁', '氧化铜', '氧化铅', '氧化钡', '五氧化二磷', '氧化锶', '氧化锡',
    '二氧化硫']
dot_data = tree.export_graphviz(clf_no_whe
, feature_names=feature_name
, class_names=["铅钡", "高钾"]
, filled=True
, rounded=True
)
graph = graphviz.Source(dot_data) # 可以通过ipython进行展示
print('决策树的可视化可以通过ipython展示')
print('*' * 30)
print('决策权拟合，有风化，去除铅：')
X_whe_dropPb = X_whe.drop('氧化铅(PbO)', axis=1)
Xtrain_whe_dropPb, Xtest_whe_dropPb, Ytrain_whe_dropPb, Ytest_whe_dropPb =
    train_test_split(X_whe_dropPb, Y_whe,
test_size=0.2,
random_state=2022)
clf_whe_dropPb =
    tree.DecisionTreeClassifier(random_state=2022).fit(Xtrain_whe_dropPb,
    Ytrain_whe_dropPb)
print('测试集准确率为：')
print(clf_whe_dropPb.score(Xtest_whe_dropPb, Ytest_whe_dropPb))
feature_name = ['二氧化硅', '氧化钠', '氧化钾', '氧化钙', '氧化镁', '氧化铝',
    '氧化铁', '氧化铜', '氧化钡', '五氧化二磷', '氧化锶', '氧化锡', '二氧化硫']
dot_data = tree.export_graphviz(clf_whe_dropPb
, feature_names=feature_name
, class_names=["铅钡", "高钾"]
, filled=True
, rounded=True
)
graph = graphviz.Source(dot_data) # 可以通过ipython进行展示
print('决策树的可视化可以通过ipython展示')
print('*' * 30)
print('决策权拟合，无风化，去除铅：')
X_no_whe_dropPb = X_no_whe.drop('氧化铅(PbO)',axis=1)
Xtrain_no_whe_dropPb, Xtest_no_whe_dropPb, Ytrain_no_whe_dropPb, Ytest_no_whe_dropPb
    = train_test_split(X_no_whe_dropPb, Y_no_whe, test_size=0.2,random_state=800)
clf_no_whe_dropPb =
    tree.DecisionTreeClassifier(random_state=2022).fit(Xtrain_no_whe_dropPb,
    Ytrain_no_whe_dropPb)
```

```
print('测试集准确率为：')
print(clf_no_whe_dropPb.score(Xtest_no_whe_dropPb, Ytest_no_whe_dropPb))
feature_name = ['二氧化硅', '氧化钠', '氧化钾', '氧化钙', '氧化镁', '氧化铝',
    '氧化铁', '氧化铜', '氧化钡', '五氧化二磷', '氧化锶', '氧化锡', '二氧化硫']
dot_data = tree.export_graphviz(clf_no_whe_dropPb
, feature_names=feature_name
, class_names=["铅钡", "高钾"]
, filled=True
, rounded=True
)
graph = graphviz.Source(dot_data) # 可以通过ipython进行展示
print('决策树的可视化可以通过ipython展示')
print('*' * 30)
print('随机森林拟合：')
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder
Xtrain_whe, Xtest_whe, Ytrain_whe, Ytest_whe = train_test_split(X_whe, Y_whe,
    test_size=0.2,random_state=400)
Xtrain_no_whe, Xtest_no_whe, Ytrain_no_whe, Ytest_no_whe =
    train_test_split(X_no_whe, Y_no_whe, test_size=0.2,random_state=300)
le1 = LabelEncoder().fit(Ytrain_whe)
rfr1 = RandomForestRegressor(random_state=2022, n_estimators=1000)
rfr1 = rfr1.fit(Xtrain_whe, le1.transform(Ytrain_whe))
score1 = rfr1.score(Xtest_whe, le1.transform(Ytest_whe))
print('有风化，测试集准确率：')
print(score1)
le12 = LabelEncoder().fit(Ytrain_no_whe)
rfr2 = RandomForestRegressor(random_state=2022, n_estimators=1000)
rfr2 = rfr2.fit(Xtrain_no_whe, le1.transform(Ytrain_no_whe))
score2 = rfr2.score(Xtest_no_whe, le1.transform(Ytest_no_whe))
print('无风化，测试集准确率：')
print(score2)
print('有风化，化学成分在分类中的重要性：')
print(pd.Series(rfr1.feature_importances_,index=X_whe.columns).sort_values(ascending=False))
print('无风化，化学成分在分类中的重要性：')
print(pd.Series(rfr2.feature_importances_,index=X_no_whe.columns)
.sort_values(ascending=False))

\begin{flushleft}
  q2\_2.py:
\end{flushleft}
\begin{lstlisting}[language=python]
  import pandas as pd
  import numpy as np
  from sklearn.cluster import KMeans
  from sklearn.metrics import silhouette_samples, silhouette_score
  import matplotlib.pyplot as plt
```

```
import matplotlib.cm as cm
import warnings

warnings.filterwarnings("ignore",category=Warning)
plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']

df2_valid = pd.read_excel('文物样品信息汇总-填补缺失值.xlsx')
print('*'*50)
print('高钾亚类划分')
df2_k_cc = df2_valid[df2_valid.类型=='高钾']
.drop(['文物编号','文物采样点','纹饰','类型','颜色','表面风化'],
axis=1)
df2_k_cc.iloc[1] = df2_k_cc.iloc[1:3].mean()
df2_k_cc.iloc[5] = df2_k_cc.iloc[5:7].mean()
df2_k_cc = df2_k_cc.drop([3,7])
print('各化学成分标准差：')
print(df2_k_cc.std().sort_values(ascending=False))
k_no_zero_count = (df2_k_cc.values != 0).sum(axis=0)
print('各化学成分排零均值：')
print((df2_k_cc.sum()/k_no_zero_count).sort_values(ascending=False))
df2_k_cc_cluster = df2_k_cc[['二氧化硅(SiO2)','氧化铝(Al2O3)','氧化钙(CaO)']]
print('最终选择的用于亚类划分的化学成分：')
print(df2_k_cc_cluster)
print('根据轮廓系数图像选择合适的n_clusters：')
X_k = df2_k_cc_cluster
for n_clusters in [2,3,4]:
n_clusters = n_clusters
fig, ax1 = plt.subplots(1)
fig.set_size_inches(20, 3.5)
ax1.set_xlim([-0.1, 1])
ax1.set_ylim([0, X_k.shape[0] + (n_clusters + 1) * 10])
clusterer = KMeans(n_clusters=n_clusters, random_state=2022).fit(X_k)
cluster_labels = clusterer.labels_
silhouette_avg = silhouette_score(X_k, cluster_labels)
print("For n_clusters =", n_clusters,
"The average silhouette_score is :", silhouette_avg)
sample_silhouette_values = silhouette_samples(X_k, cluster_labels)
y_lower = 10
for i in range(n_clusters):
ith_cluster_silhouette_values = sample_silhouette_values[cluster_labels == i]
ith_cluster_silhouette_values.sort()
size_cluster_i = ith_cluster_silhouette_values.shape[0]
y_upper = y_lower + size_cluster_i
color = cm.nipy_spectral(float(i)/n_clusters)
ax1.fill_betweenx(np.arange(y_lower, y_upper)
,ith_cluster_silhouette_values
,facecolor=color
```

```
,alpha=0.7
)
ax1.text(-0.05
, y_lower + 0.5 * size_cluster_i
, str(i))
y_lower = y_upper + 10

# ax1.set_title("The silhouette plot for the various clusters.")
ax1.set_xlabel("各样本轮廓系数的取值",fontsize=14)
ax1.set_ylabel("聚类簇的标签",fontsize=14)
ax1.axvline(x=silhouette_avg, color="red", linestyle="--")
ax1.set_yticks([])
ax1.set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1])
plt.show()
clusterer = KMeans(n_clusters=2, random_state=2022).fit(df2_k_cc_cluster)
pri_k_cluster_res = clusterer.labels_
k_cluster_res = clusterer.labels_
# 输出结果
k_cluster_res = list(k_cluster_res)
k_cluster_res.insert(1,k_cluster_res[1])
k_cluster_res.insert(5,k_cluster_res[5])
k_cluster_info_all = df2_valid[df2_valid.类型=='高钾']
k_cluster_info_all['亚类标号'] = k_cluster_res
k_cluster_info_all.to_excel('高钾亚类划分结果.xlsx',index=False)
# 分析结果
sub_cl0_mean = k_cluster_info_all[k_cluster_info_all.亚类标号 == 0].drop(
['文物编号', '亚类标号', '文物采样点', '颜色', '类型', '表面风化', '纹饰'],
    axis=1).mean()
sub_cl1_mean = k_cluster_info_all[k_cluster_info_all.亚类标号 == 1].drop(
['文物编号', '亚类标号', '文物采样点', '颜色', '类型', '表面风化', '纹饰'],
    axis=1).mean()
print('亚类化学成分均值对比：')
print(pd.DataFrame([sub_cl0_mean, sub_cl1_mean], columns=sub_cl0_mean.index))
print('灵敏度分析：')
ccs = ['二氧化硅(SiO2)','氧化铝(Al2O3)','氧化钙(CaO)']
pre_labels = pri_k_cluster_res
pre_data = np.array(df2_k_cc_cluster)
wav_data = pre_data[pre_labels==0]
record = np.zeros((20,5),dtype=object)
rd_idx = 0
for i in range(len(ccs)):
for wav in [-0.01,-0.02,-0.05,-0.1,0.5,1,3,5,7,8,9,10]:
if (i == 0) & (wav > 0):
continue
if (i != 0) & (wav < 0):
continue
new_data = np.array(pre_data)
```

```
            new_wave_data = np.array(wav_data)
            new_wave_data[:,i] = new_wave_data[:,i] * (1 + wav)
            new_data[pre_labels==0] = new_wave_data
            clusterer = KMeans(n_clusters=2, random_state=2022).fit(new_data)
            cl_res = clusterer.labels_
            if cl_res[0]==0:
            cl_res = np.where(cl_res==0,1,0)
            record[rd_idx,:] = [ccs[i],wav,pre_labels,cl_res,np.all(pre_labels==cl_res)]
            rd_idx += 1
            print(pd.DataFrame(record,columns=['化学成分', '波动率', '波动前的聚类结果',
                '波动后的聚类结果', '聚类结果是否保持不变']))
            print('*'*50)
            print('铅钡亚类划分')
            df2_pb_cc = df2_valid[df2_valid.类型 == '铅钡'].drop(['文物编号', '文物采样点',
                '纹饰', '类型', '颜色', '表面风化'], axis=1)
            df2_pb_cc.iloc[13] = df2_pb_cc.iloc[13:15].mean()
            df2_pb_cc.iloc[26] = df2_pb_cc.iloc[26:28].mean()
            df2_pb_cc.iloc[28] = df2_pb_cc.iloc[28:30].mean()
            df2_pb_cc.iloc[39] = df2_pb_cc.iloc[39:41].mean()
            df2_pb_cc = df2_pb_cc.drop([32,45,47,58])
            print('各化学成分标准差：')
            print(df2_pb_cc.std().sort_values(ascending=False))
            print('各化学成分排零均值：')
            pb_no_zero_count = (df2_pb_cc.values != 0).sum(axis=0)
            print((df2_pb_cc.sum()/pb_no_zero_count).sort_values(ascending=False))
            print('最终选择的用于亚类划分的化学成分：')
            df2_pb_cc_cluster = df2_pb_cc[
            [
            '二氧化硅(SiO2)',
            '五氧化二磷(P2O5)',
            '二氧化硫(SO2)',
            '氧化铝(Al2O3)',
            '氧化铜(CuO)']
            ]
            print(df2_pb_cc_cluster)
            print('根据轮廓系数图像选择合适的n_clusters：')
            plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
            X_pb = df2_pb_cc_cluster
            for n_clusters in [2, 3, 4]:
            n_clusters = n_clusters
            fig, ax1 = plt.subplots(1)
            fig.set_size_inches(20,4.5)
            ax1.set_xlim([-0.1, 1])
            ax1.set_ylim([0, X_pb.shape[0] + (n_clusters + 1) * 10])
            clusterer = KMeans(n_clusters=n_clusters, random_state=2022).fit(X_pb)
            cluster_labels = clusterer.labels_
            silhouette_avg = silhouette_score(X_pb, cluster_labels)
```

```
        print("For n_clusters =", n_clusters,
        "The average silhouette_score is :", silhouette_avg)
        sample_silhouette_values = silhouette_samples(X_pb, cluster_labels)
        y_lower = 10
        for i in range(n_clusters):
        ith_cluster_silhouette_values = sample_silhouette_values[cluster_labels == i]
        ith_cluster_silhouette_values.sort()
        size_cluster_i = ith_cluster_silhouette_values.shape[0]
        y_upper = y_lower + size_cluster_i
        color = cm.nipy_spectral(float(i) / n_clusters)
        ax1.fill_betweenx(np.arange(y_lower, y_upper)
        , ith_cluster_silhouette_values
        , facecolor=color
        , alpha=0.7
        )
        ax1.text(-0.05
        , y_lower + 0.5 * size_cluster_i
        , str(i))
        y_lower = y_upper + 10

        # ax1.set_title("The silhouette plot for the various clusters.")
        ax1.set_xlabel("各样本轮廓系数的取值",fontsize=14)
        ax1.set_ylabel("聚类簇的标签",fontsize=14)
        ax1.axvline(x=silhouette_avg, color="red", linestyle="--")
        ax1.set_yticks([])
        ax1.set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1])
        plt.show()
        clusterer = KMeans(n_clusters=2, random_state=2022).fit(X_pb)
        pri_pb_clu_res = clusterer.labels_
        pb_cluster_res = clusterer.labels_
        pb_cluster_res = list(pb_cluster_res)
        pb_cluster_res.insert(13,pb_cluster_res[13])
        pb_cluster_res.insert(26,pb_cluster_res[26])
        pb_cluster_res.insert(28,pb_cluster_res[28])
        pb_cluster_res.insert(39,pb_cluster_res[39])
        # 输出结果
        pb_cluster_info_all = df2_valid[df2_valid.类型=='铅钡']
        pb_cluster_info_all['亚类标号'] = pb_cluster_res
        pb_cluster_info_all.to_excel('铅钡亚类划分结果.xlsx',index=False)
        # 分析结果
        sub_cl0_mean = pb_cluster_info_all[pb_cluster_info_all.亚类标号==0]
        .drop(['文物编号','亚类标号','文物采样点','颜色','类型','表面风化','纹饰'],axis=1).mean()
        sub_cl1_mean = pb_cluster_info_all[pb_cluster_info_all.亚类标号==1]
        .drop(['文物编号','亚类标号','文物采样点','颜色','类型','表面风化','纹饰'],axis=1).mean()
        print('亚类化学成分均值对比：')
        print(pd.DataFrame([sub_cl0_mean,sub_cl1_mean],columns=sub_cl0_mean.index))
        print('灵敏度分析：')
```

```
ccs = ['二氧化硅(SiO2)','五氧化二磷(P2O5)','氧化铝(Al2O3)']
pre_labels = pri_pb_clu_res
pre_data = np.array(df2_pb_cc_cluster)
wav_data = pre_data[pre_labels==0]
record = np.zeros((100,5),dtype=object)
rd_idx = 0
wav_range = [
[-0.01,-0.02,-0.05,-0.1,-0.15,-0.2,-0.3],
[1,3,5,7,8,9,10,16,20],
[-0.1,-0.3,-0.5,-0.7,-0.8,-0.9],
]
for i in range(3):
for wav in wav_range[i]:
new_data = np.array(pre_data)
new_wave_data = np.array(wav_data)
new_wave_data[:,i] = new_wave_data[:,i] * (1 + wav)
new_data[pre_labels==0] = new_wave_data
clusterer = KMeans(n_clusters=2, random_state=2022).fit(new_data)
cl_res = clusterer.labels_
if cl_res[0]==0:
cl_res = np.where(cl_res==0,1,0)
record[rd_idx,:] = [ccs[i],wav,pre_labels,cl_res,np.all(pre_labels==cl_res)]
rd_idx += 1
print(pd.DataFrame(record,columns=['化学成分', '波动率', '波动前的聚类结果',
    '波动后的聚类结果', '聚类结果是否保持不变']))
print('*'*50)
```

问题三

q3.py:

```
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from scipy.stats import zscore
import pylab as plt
from sklearn.model_selection import train_test_split
import scipy.stats as stats
from sklearn.neighbors import KNeighborsClassifier
import warnings
warnings.filterwarnings("ignore")

plt.rcParams["font.sans-serif"] = ["SimHei"] #解决中文字符乱码的问题
plt.rcParams["axes.unicode_minus"] = False #正常显示负号

# 读取数据
```

```
df2_valid = pd.read_excel('文物样品信息汇总-填补缺失值.xlsx')
# 填补缺失值
df2_valid_fill0 = df2_valid.fillna(0)
# 转换虚拟变量
df2_valid_fill0_dummies =
    pd.get_dummies(df2_valid_fill0[['纹饰','类型','颜色','表面风化']])
df2_valid_fill0 = df2_valid_fill0.drop(['纹饰','类型','颜色','表面风化'],axis=1)
df2_valid_fill0_dummies =
    pd.concat([df2_valid_fill0,df2_valid_fill0_dummies],axis=1)
df2_valid_fill0_dummies
df2_valid_fill0_dummies.to_excel('表1和表2数据（含虚拟变量）.xlsx')

# 保留与化学成分信息和类别信息
df = pd.read_excel('表1和表2数据（含虚拟变量）.xlsx')
feature = df[['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)']]
label = df['类型_铅钡']
all_f = df[['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)',
'类型_铅钡']]
all_f.to_excel('指标汇总（所有化学成分）.xlsx')

# 正态分布检验（JB检验）
jb_p = []
jb_values = []
for i in feature.columns:
res = stats.jarque_bera(feature[i])
jb_p.append(res.pvalue)
jb_values.append(res.statistic)
res_N = pd.DataFrame({'统计量':jb_values,
   'P值':jb_p,
   '是否通过正态分布检验\n(95%置信水平)':
   ['是' if x>0.05 else '否' for x in jb_p]})
res_N.index = feature.columns
res_N.to_excel('所有玻璃样本的正态分布检验.xlsx')
print(jb_values,jb_p)

# 计算斯皮尔曼相关系数
res_r = []
res_p = []
for x in feature.columns:
```

```
        r, p = stats.spearmanr(feature[x], label)
        res_r.append(r)
        res_p.append(p)
        res_sp = pd.DataFrame({'相关系数(绝对值)':abs(np.array(res_r)),
           'P值':res_p})
        res_sp.index = feature.columns
        res_sp = res_sp.sort_values(by='相关系数(绝对值)',ascending=False)
        res_sp.to_excel('各化学成分与玻璃类型的spearman相关系数(绝对值).xlsx')

        # 需要降维的指标 ['氧化钠(Na2O)', '氧化钙(CaO)', '氧化镁(MgO)', '氧化铝(Al2O3)',
            '氧化铁(Fe2O3)','氧化铜(CuO)',
         '五氧化二磷(P2O5)', '氧化锡(SnO2)', '二氧化硫(SO2)']
        # 保留的指标 ['二氧化硅(SiO2)', '氧化钾(K2O)', '氧化铅(PbO)', '氧化钡(BaO)',
            '氧化锶(SrO)']

        df = pd.read_excel('指标汇总（所有化学成分）.xlsx')
        X = df.drop(['类型_铅钡'],axis=1)
        y = df['类型_铅钡']
        X_reduce = X[['氧化钠(Na2O)', '氧化钙(CaO)', '氧化镁(MgO)', '氧化铝(Al2O3)',
        '氧化铁(Fe2O3)','氧化铜(CuO)', '五氧化二磷(P2O5)', '氧化锡(SnO2)', '二氧化硫(SO2)']]

        # 降维保留个数的确认
        contribute = []
        X_ = zscore(X_reduce, axis=0) # X需要标准化
        for i in range(1,9):
        pca = PCA(n_components=i)
        result_pca = pca.fit_transform(X_) # 这里的数据实际上已经被标准化了
        t = np.array(pca.explained_variance_ratio_).cumsum()[-1]
        contribute.append(t)
        plt.plot(range(1,9),contribute,'-r',range(1,9),contribute,'b*',range(1,9),
        [0.85,0.85,0.85,0.85,0.85,0.85,0.85,0.85],'--k')
        plt.xlabel('特征数')
        plt.ylabel('累计贡献率')
        plt.savefig('特征数与累计贡献率的关系.png')
        plt.show()

        # 确认保留6个维度，求系数矩阵
        pca = PCA(n_components=6)
        result_pca = pca.fit_transform(X_)
        np.savetxt('特征向量(系数矩阵).txt',np.array(pca.components_.T),)

        # 指标选取（保存相关性大的指标，其余进行降维）
        '''
        函数名：deal_sheet3
        函数作用：对附件表三进行指标提取
        输入：要改变含量的列col，增加幅度i，权中系数w1，w2
        输出：11个指标
```

```
'''
def deal_sheet3(col,i,w1,w2,data):
df_in = None
if data is not None:
df_in = data

else:
df_in = pd.read_excel('附件.xlsx', sheet_name=2)
df_in = df_in.drop(['文物编号', '表面风化'], axis=1)
df_in = df_in.fillna(0)
# print(df3)
df_in[col] = (1+i)*df_in[col]
temp = df_in.sum(axis=1)[:,np.newaxis]
temp = np.tile(temp,(1,df_in.shape[1]))
df_in = (df_in/temp)*100
df_in.to_excel('预测数据'+col+'增加'+str(i)+'.xlsx')
df3_X = df_in[['氧化钠(Na2O)', '氧化钙(CaO)', '氧化镁(MgO)', '氧化铝(Al2O3)',
    '氧化铁(Fe2O3)','氧化铜(CuO)', '五氧化二磷(P2O5)', '氧化锡(SnO2)',
    '二氧化硫(SO2)']]
df3_X_ = zscore(df3_X, axis=0) # X需要标准化
result_pca = df3_X_ @ np.loadtxt('特征向量(系数矩阵).txt')
df3_all_feature = pd.concat([w1*df_in[['二氧化硅(SiO2)', '氧化钾(K2O)',
    '氧化铅(PbO)', '氧化钡(BaO)',
    '氧化锶(SrO)']],pd.DataFrame(w2*result_pca)],axis=1)
return df3_all_feature
'''
函数名：deal_X
函数作用：对数据进行指标提取
输入：原始数据X，权中系数w1，w2
输出：11个指标
'''
def deal_X(X,w1,w2):
temp = X.sum(axis=1)[:,np.newaxis]
temp = np.tile(temp,(1,X.shape[1]))
X = (X/temp)*100
X_reduce = X[['氧化钠(Na2O)', '氧化钙(CaO)', '氧化镁(MgO)', '氧化铝(Al2O3)',
    '氧化铁(Fe2O3)','氧化铜(CuO)', '五氧化二磷(P2O5)', '氧化锡(SnO2)',
    '二氧化硫(SO2)']]
X_reduce_ = zscore(X_reduce, axis=0) # X需要标准化
result_pca = X_reduce_ @ np.loadtxt('特征向量(系数矩阵).txt')
all_feature = pd.concat([w1*X[['二氧化硅(SiO2)', '氧化钾(K2O)', '氧化铅(PbO)',
    '氧化钡(BaO)', '氧化锶(SrO)']],pd.DataFrame(w2*result_pca)],axis=1)
return all_feature

# 读取数据并提取指标
w1 = 0.7
w2 = 0.3
```

```
X = deal_X(X,w1,w2)
X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.3,random_state=8)#8,12,2,6,7 9
X_predict = deal_sheet3('二氧化硅(SiO2)',0,w1,w2,None)

# 预测
clf_sk = KNeighborsClassifier(n_neighbors=5)
clf_sk.fit(X_train, y_train)
print('knn得分:',clf_sk.score(X_test, y_test))
res_pre = clf_sk.predict(X_predict)
print('预测结果',res_pre)
# 将预测结果保存
df3 = pd.read_excel('附件.xlsx', sheet_name=2)
df3['预测结果'] = ['高钾玻璃' if x==0 else '铅钡玻璃' for x in res_pre]
df3.to_excel('附件表单3预测结果.xlsx',index=False)
# 敏感性分析
'''
函数名: Sensitive_Analysis
函数作用: 改变某一化学成分含量, 输出结果变化
输入: 模型model, 要改变的化学成分col, 每次增加的幅度delta, 增加次数i
输出: 准确度得分数组, 和预测结果数组
'''
def Sensitive_Analysis(model,col,delta,i):
model.fit(X_train,y_train)
X_test_ = X_test.copy()
score = []
predict_res = []
for k in range(i):
X_predict_ = deal_sheet3(col,k*delta,w1,w2,None)
if not (col == '氧化铝(Al2O3)'):
X_test_[col] = (k * delta + 1) * X_test[col]
score_temp = model.score(X_test_,y_test)
predict_temp = model.predict(X_predict_)
score.append(score_temp)
predict_res.append(predict_temp)
print('\n'+col+'增加'+str(k*delta)+'倍:')
print('得分',score_temp)
print('预测结果',predict_temp)
return score,predict_res

##%%
# knn算法敏感度分析
prefict_res =
    Sensitive_Analysis(KNeighborsClassifier(n_neighbors=5),'二氧化硅(SiO2)',0.1,8)
prefict_res =
    Sensitive_Analysis(KNeighborsClassifier(n_neighbors=5),'氧化铅(PbO)',-0.1,8)
prefict_res =
```

```
    Sensitive_Analysis(KNeighborsClassifier(n_neighbors=5),'氧化钡(BaO)',-0.1,11)
prefict_res =
    Sensitive_Analysis(KNeighborsClassifier(n_neighbors=5),'氧化钾(K2O)',-0.1,11)
prefict_res =
    Sensitive_Analysis(KNeighborsClassifier(n_neighbors=5),'氧化铝(Al2O3)',-0.1,8)
```

问题四

q4.py:

```
import numpy as np
import pandas as pd
import pylab as plt
import seaborn as sns
import scipy.stats as stats
import warnings
warnings.filterwarnings("ignore")
plt.rcParams["font.sans-serif"] = ["SimHei"] #解决中文字符乱码的问题
plt.rcParams["axes.unicode_minus"] = False #正常显示负号

# 读取数据
df = pd.read_excel('表1和表2数据（含虚拟变量）.xlsx')
feature =
    df[['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)','类型_铅钡']]
feature.to_excel("化学成分含量及类型.xlsx")
feature_group = feature.groupby(by='类型_铅钡')
feature_group = [x[1].drop('类型_铅钡',axis=1) for x in feature_group]
feature_group[0].to_excel('高钾玻璃数据的特征.xlsx',index=False)
feature_group[1].to_excel("铅钡玻璃数据的特征.xlsx",index=False)

# 正态分布检验（JB检验）
def JB_Check(data,mess):
jb_p = []
jb_values = []
plt.figure(14,figsize=(10,100))
i = 1
for item in data.columns:
# JB检验
res = stats.jarque_bera(data[item])
jb_p.append(res.pvalue)
jb_values.append(res.statistic)
# 画直方图
plt.subplot(14,1,i)
plt.hist(data[item], bins=20, density=True)
```

```
plt.title(item)
# plt.savefig('./图片/'+mess+item+'直方图.png')
i += 1
plt.show()
print(mess+'玻璃是否通过正态分布检验', data.columns[np.array(jb_p) > 0.05])
res_N = pd.DataFrame({'统计量': jb_values,
   'P值': jb_p,
   '是否通过正态分布检验\n(95%置信水平)':
   ['是' if x > 0.05 else '否' for x in jb_p]})
res_N.index = data.columns
res_N.to_excel(mess+'玻璃样本的正态分布检验.xlsx')
return jb_values,jb_p

# 正态分布检验
df1 = pd.read_excel('铅钡玻璃数据的特征.xlsx')
JB_Check(df1,'铅钡')
df2 = pd.read_excel('高钾玻璃数据的特征.xlsx')
JB_Check(df2,'高钾')

# 绘制热力图
def graph(data,mess):
fig = plt.figure(figsize = (10,8))
ax = fig.add_subplot(111)
ax = sns.heatmap(data,vmax = 1,vmin = -1,annot = True,annot_kws =
    {"size":10,"weight":"bold"},linewidths = 0.05,cmap="RdBu")
plt.xticks(fontsize = 15)
plt.yticks(fontsize = 15)
plt.savefig(mess+'玻璃各化学成分spearman相关系数热力图.png',dpi=200,
    bbox_inches='tight')
plt.show()
# 斯皮尔曼相关系数
col = ['二氧化硅','氧化钠','氧化钾','氧化钙',
'氧化镁','氧化铝','氧化铁','氧化铜','氧化铅',
'氧化钡','五氧化二磷','氧化锶','氧化锡',
'二氧化硫']
corr1 = df1.corr('spearman')
corr2 = df2.corr('spearman')
corr1.columns = col
corr1.index = col
corr2.columns = col
corr2.index = col
graph(corr1,'铅钡')
graph(corr2,'高钾')

# 秩检验
res_statistic = []
res_p = []
```

```
for i in col:
res = stats.wilcoxon(corr1[i],corr2[i],correction=True)
res_statistic.append(res[0])
res_p.append(res[1])
wilcoxon_df = pd.DataFrame({'统计量':res_statistic,
  'P值':res_p,
  '是否有差异(95%)':['有差异' if x < 0.05 else '无差异' for x in res_p],
  '是否有差异(90%)':['有差异' if x < 0.10 else '无差异' for x in res_p]})
wilcoxon_df.index = col
wilcoxon_df.to_excel('wilcoxon符号秩检验(spearman).xlsx')
```