古代玻璃制品的成分分析与鉴别问题研究

摘要

本文研究的是高钾与铅钡两种古代玻璃制品表面风化与其纹饰、类型、颜色之间的关系；同时依据玻璃文物中各化学成分含量的统计规律，对高钾与铅钡两种玻璃进行亚类划分，并探究每一亚类中各化学成分之间的关联关系以及亚类与亚类间这种关联关系的差异性的问题。

针对问题一，对表单 1 中的三个分类变量进行定类赋值，建立以玻璃的纹饰、颜色、种类为自变量，是否发生风化为因变量的**二元 logistic 回归模型**，并利用霍斯默-莱梅肖检验来验证模型的有效性。此外，还通过灰色关联分析法得到各自变量的重要程度，并与拟合函数相结合，最终得出了玻璃类型与表面是否风化关联度最大，且铅钡玻璃较易风化、纹饰 B 较易风化以及黑色较易风化的结论。接着将玻璃制品按照类别和是否风化分为 4 类，通过建立 **CART 决策树模型**，得到了不同类别的玻璃制品在风化前后的化学成分变化规律，给出了根据化学成分判别玻璃制品类型的标准。最后利用玻璃风化前后化学成分平均变化率对其风化前的化学成分含量进行预测。

针对问题二，利用问题一所给出的预测模型，获取 58 个文物未风化前的化学成分，并基于前文中的 CART 决策树模型，建立了以 14 种化学成分占比量为自变量，两种玻璃类别为因变量的**随机森林模型**，从而得出高钾玻璃与铅钡玻璃制品的分类规律：PbO、BaO、K_2O 的含量为分类时最重要的特征，且 PbO 含量大于 7.535% 时该样品一定为铅钡类玻璃。此后再次使用随机森林模型筛选出了在每一类玻璃中对于抵抗风化最为重要的化学成分，并建立 **K 均值聚类模型**对其进行亚类划分。最终将所有玻璃分为**高钾抗风化、高钾不抗风化、铅钡抗风化、铅钡不抗风化**四个亚类。此后，通过将分类结果与真实的情况对比，并随机去掉几组数值，观察聚类中心的变化程度来对该模型的合理性与敏感性进行了验证。

针对问题三，首先利用问题二中所建立的随机森林模型，对未给定类型的玻璃文物进行大类的划分；随后根据样本中各化学成分含量到问题二中 K 均值聚类模型所得的各亚类图心的距离，建立了分类模型，从而完成对样本所属类型的划分，并得出 A1 为高钾抗风化玻璃；A6，A7 为高钾不抗风化玻璃；A2，A3，A4，A8 为铅钡抗风化玻璃；A5 为铅钡不抗风化玻璃的结论。此外，对于该问中所用到的随机森林模型进行**数据洗牌**，并重新对样本进行分类，通过对比前后两次分类的结果，发现样本的分类并无变化，完成了对模型敏感性的分析。

针对问题四，利用 **Pearson 相关分析**，分别在每一亚类中构造各化学成分含量彼此之间的相关系数，建立相关矩阵模型，并依据矩阵中元素的取值来判断该类玻璃中各化学成分关联关系的正负与强弱。对于比较亚类与亚类间关联关系的差异性比较的问题，我们选择将两个亚类各自的相关矩阵做差，建立**矩阵二范数模型**，来刻画差异性大小。

关键词：二元 logistic 回归模型　CART 决策树模型　随机森林模型　Pearson 相关分析

一、 问题重述

1.1 问题背景

研究古代文化的重要手段就是分析同时期的历史文物。古代中国通过丝绸之路架起了中西方文化沟通的桥梁，现在出土的文物中有不少是从西方传入中国的文物，其中玻璃是早期贸易往来的宝贵物证。古代劳动人民吸收其技术后，制作出具有本地特色的铅钡玻璃。我们希望可以通过分析其化学成分区分出玻璃的种类，以及其抗风化能力，对进一步研究古代玻璃技术的发展、演变，以及对我国其他地区玻璃技术发展的影响提供更多的方法。

1.2 数据分析

本题给出一个附件数据，本文将对这些数据进行初步解读。

表单一：给出不同文物的纹饰、类型、颜色以及表面是否风化的数据。可以初步推断各组数据之间存在某种联系。

表单二：给出不同文物不同采样点采集的 14 种氧化物的成分占比数据，可以看出除二氧化硅为大量成分外，别的化学成分含量都极小，呈稀疏矩阵的样式。

表单三：给出 8 个文物的 14 种氧化物成分占比以及表面是否风化的数据，需要我们给出玻璃类别数据。

1.3 具体问题

依据附件中的相关数据进行分析建模，解决以下问题：

(1) 探究玻璃文物的类型、纹饰和颜色与表面风化的函数关系；根据高钾玻璃和铅钡玻璃两种类型，研究文物样品表面是否风化化学成分含量的统计规律，并依据风化点检测数据，预测其风化前的化学成分含量。

(2) 分析高钾玻璃、铅钡玻璃的分类依据；通过对其化学成分分析，给出划分亚类玻璃的方法及划分结果，并分析分类结果的合理性和敏感性。

(3) 依据化学成分含量，对表单 3 中未知类别玻璃文物进行分类，并对分类结果的敏感性进行分析。

(4) 探究同种玻璃中的化学成分之间的关联关系，再研究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

问题一是一个相关性分析以及统计规律分析问题。为了同时得到纹饰、类型、颜色对玻璃风化的影响关系，考虑到玻璃类型和纹饰都是定类变量，而表面风化只有是与否两个取值，选择建立二元 logistic 回归模型。通过拟合所得的函数即可得出此三种变量对与玻璃风化影响程度的正负与强弱。同时我们利用灰度关联分析对所得的关系进行验证，以证明结果的可靠性。而对于文物样品表面有无风化化学成分含量的统计规律的分析，

我们选择建立决策树模型将样本分为高钾风化、高钾无风化、铅钡风化、铅钡无风化四种类型，并以分类时所依据的各类氧化物的取值范围作为该类中化学成分含量的统计规律。此外，在剔除异常取值后，我们对高钾类玻璃与铅钡类玻璃风化前后各化学成分含量的均值进行了统计，并以两者的均值之差来对风化前的化学成分含量进行预测。

2.2 问题二

问题二是一个分类问题，要求我们分析高钾玻璃、铅钡玻璃的分类规律，并在此基础上对每种玻璃进行亚类划分。为了得到各类氧化物的含量在划分高钾玻璃与铅钡玻璃时的重要程度，我们在问题一决策树模型的基础上建立以 14 种氧化物含量为自变量，以两种玻璃为因变量的随机森林模型。通过该模型的随机抽样、随机取特征值来完成对样本集的分类与对各氧化物重要性的打分。接着我们利用随机森林模型解得的各氧化物的重要性，选出权值较大的化学成分进行 K 均值聚类，以此为依据在两种大类玻璃下划分亚类。此外，我们通过分类的结果与其是否发生风化的真实结果进行比较，从而验证模型的合理性；通过随机取掉一组数据，再次使用聚类模型，对比两次聚类中心的距离大小从而分析模型的敏感性。

2.3 问题三

问题三同样是一个分类问题。由于表单三中给出的玻璃文物的类型未知，且问题二中对于高钾玻璃与铅钡玻璃划分亚类的指标又不尽相同，故我们首先利用前文中所建立的随机森林分类模型对这些样本是属于高钾玻璃还是铅钡玻璃进行预测。 随后根据预测的结果，我们建立分类模型，通过计算其相应氧化物的含量到各亚类图心的距离，对该样本进行正确的亚类划分。最后，我们随机去掉随机森林分类模型训练集中的几组数据，得到一个重新训练的模型，通过对比同一样本前后两次分类的结果，从而验证模型的敏感性。

2.4 问题四

问题四是一个关联性分析问题。依照题目要求，我们首先应考虑在同一类玻璃中，各类氧化物的含量彼此之间的关联关系。在此基础上，再来探究不同类玻璃之间这种关联关系有无差异性。对此我们选择建立相关矩阵分析模型。首先通过对某类玻璃中各类氧化物的含量之间的 Pearson 系数的计算，完成对该类玻璃相关矩阵的构建。随后对于不同类的玻璃，我们将两者的相关矩阵做差，得到一个新的矩阵，通过计算该矩阵的二范数，从而完成对不同类玻璃之间这种关联关系的差异性的计算。

三、 模型假设

1. 假设玻璃化学成分含量测定准确无误；
2. 假设未检查到的化学成分其含量可以认为是零；
3. 假设这组玻璃为同一时期的历史文物；
4. 假设玻璃纹饰、类型、颜色与是否风化的数据具有一定代表性；
5. 假设随机森林模型训练中采样随机、特征选取随机。

四、 符号说明

符号	符号意义
α_i	变量i的拟合参数
x_1	纹饰 A 的哑变量
x_2	纹饰 B 的哑变量
x_3	纹饰 C 的哑变量
x_4	高钾类玻璃的哑变量
x_5	铅钡类玻璃的哑变量
x_6	玻璃颜色的变量
D	决策树模型中样本的总个数
C_i	决策树模型中第i类样本的个数
$H(D)$	样本的经验熵
$H(D\|A)$	氧化物 A 在样本中的条件熵
$g(D,\mathrm{A})$	氧化物 A 在样本中的信息增量
m	随机森林模型中构建的决策树数量
$h_i(x)$	随机森林模型中第i棵决策树的分类结果
I_{ij}	氧化物E_j在第i棵树T_i中的重要性
IMP_j	氧化物E_j最终的重要程度
d	未分类样本到各个类别图心的最短距离
F_B	因素 B 的检验统计量
Y	样本玻璃各类氧化物所构成的向量
ρ_{ij}	氧化物i与j之间的 Pearson 相关系数
R	两类玻璃间的相关系数矩阵

五、 问题一的模型建立与求解

5.1 数据预处理

首先根据题目条件，对表单 2 中每个文物的各类化学物质占比进行求和，筛选出总和不在 85%~105%这个范围的文物，对其数据进行删除。通过检验应该删去 15 和 17 两个数据。

问题一要求我们对文物玻璃的类型、纹饰和颜色与其表面风化程度之间的关系进行分析，对此我们首先将这三个指标分为定类尺度与定序尺度。

对于文物玻璃的类型与纹路，我们将其设为定类尺度，其赋值规则如下：

表 5.1　文物玻璃纹饰、类型赋值表

纹饰赋值	
纹饰	编号
A	1
B	2
C	3

类型赋值	
类型	编号
高钾	1
铅钡	2

由于颜色的深浅不同，风化程度的强弱也不同，故对于文物玻璃的颜色，我们依照深浅将其设为定序尺度，其赋值规则如下：

表 5.2　文物玻璃颜色赋值表

颜色赋值									
颜色	浅绿	浅蓝	绿	蓝绿	蓝	深绿	深蓝	紫	黑
赋值	1	2	3	4	5	6	7	8	9

5.2 二元 logistic 回归模型的建立与求解

5.2.1 模型的建立

题目中将文物玻璃的表面风化程度分为风化与无风化两类，故该因变量为一个二分因变量。对此，我们用 0 表示无风化，用 1 表示风化，并建立二元 logistic 回归模型来探究文物玻璃的类型、纹饰和颜色与其表面风化程度之间的关系。

设文物玻璃表面风化的概率为 p，为使 p 的取值在 $[0,1]$ 之间，我们选取 $ln\left(\frac{p}{1-p}\right)$ 作为因变量，此时玻璃的类型、纹饰和颜色与其表面风化程度之间的拟合函数为：

$$ln\left(\frac{p}{1-p}\right)=\alpha_0+\alpha_1x_1+\alpha_2x_2+\alpha_3x_3+\alpha_4x_4+\alpha_5x_5+\alpha_6x_6 \tag{1}$$

经过变换，p 可以表示为：

$$p=\frac{1}{1+e^{-(\alpha_0+\alpha_1x_1+\alpha_2x_2+\alpha_3x_3+\alpha_4x_4+\alpha_5x_5+\alpha_6x_6)}} \tag{2}$$

此时 p 的取值在 $[0,1]$ 之间。

同时，根据附件中的数据可知，在 58 个样本数据中，有 24 件文物未风化、34 件文物风化。根据极大似然估计原理，我们希望出现这种情况的概率最大，故我们可以得到如下的极大似然函数 L：

$$L=\prod_{i=1}^{n}{p_i}^{Y_i}(1-p_i)^{1-Y_i} \tag{3}$$

对其左右两边同时取对数有：

$$ln\,L=\sum_{i=1}^{n}[Y_i lnp_i+(1-Y_i)\ln\,(1-p_i)] \tag{4}$$

综上，我们建立了二元 logistic 回归模型：

$$\begin{cases} p = \dfrac{1}{1+e^{-(\alpha_0+\alpha_1x_1+\alpha_2x_2+\alpha_3x_3+\alpha_4x_4+\alpha_5x_5+\alpha_6x_6)}} \\ lnL = \sum_{i=1}^{n}[Y_i lnp_i + (1-Y_i)\ln(1-p_i)] \end{cases} \tag{5}$$

通过极大化 lnL ，可以求解到最优的表面风化概率 p 值，随后利用牛顿迭代法，即可求得待估参数 α_0、α_1、α_2、α_3 的取值，从而探究出文物玻璃的类型、纹饰和颜色与其表面风化程度之间的关系。

5.2.2 模型的求解与结论分析

首先分析是否可用二元 logistic 回归，显然样本总量是大于自变量个数的 10 倍，符合模型使用规则。由于玻璃的颜色的变量呈现深浅的连续变化规律，故我们将颜色做为连续性变量，将玻璃的纹饰和类别设置为哑变量，将三者代入所建立的模型中，得到模型参数。

方程中的变量

		B	标准误差	瓦尔德	自由度	显著性	Exp(B)
步骤 1[a]	纹饰			.059	2	.971	
	纹饰(1)	-.173	.715	.059	1	.808	.841
	纹饰(2)	42.353	20077.547	.000	1	.998	2.474E+18
	类型(1)	-22.041	11569.879	.000	1	.998	.000
	颜色	-.085	.127	.453	1	.501	.918
	常量	1.232	.695	3.145	1	.076	3.430

a. 在步骤 1 输入的变量：纹饰, 类型, 颜色。

图 5.1 二元 logistic 回归模型参数值

于是我们得到是否风化与颜色、纹饰、类型 3 个变量之间的函数表达式如下：

$$p = \frac{1}{1+e^{-(1.232-0.173x_2+4\ .353x_3-22.041x_5-0.085x_6)}}$$

根据各变量的显著性可以看出，纹饰 B 与铅钡类玻璃对文物表面是否风化的影响较为显著。此外，在模型求解的过程中我们使用了极大似然法，这会导致单个变量的显著性大于 0.05，为了能够更加准确的探究自变量与因变量之间的关系，我们将使用灰色关联分析法对四者之间的关联度进行分析，首先数据进行均值化处理，接着求解母序列和特征序列之间的灰色关联系数值进而求解灰色关联度值。其结果如下：

表 5.3 三个自变量与因变量的关联度表

关联度结果		
评价项	关联度	排名
类型	0.696	1
纹饰	0.674	2
颜色	0.556	3

关联度值介于 0 到 1 之间，该值越大代表其与是否发生风化之间的相关性越强，也即意味着其评价越高。从上表可以看出：玻璃的类型和纹饰对是否发生风化的影响要比颜色影响更大，且玻璃类型的关联度最高。

5.2.3 模型的检验

我们将采用 Hosmer-Lemeshow 检验方法对模型进行检验。得到的 P 值如果大于 0.05 则接受原假设，即模型对真实值拟合是有效的，如果 P 值小于 0.05 则接受备择假设，即模型拟合是无效的。利用 spss 软件得到的结果如下：

霍斯默-莱梅肖检验

步骤	卡方	自由度	显著性
1	3.649	7	.819

图 5.2 模型检验图

P 值为 0.819 显然是远大于 0.05，证明了模型的有效性。为了进一步评价模型，我们对模型预测准确度进行分析，通过对比真实值和预测值，得到模型的准确率。

分类表[a]

			预测		
			表面风化		
	实测		0	1	正确百分比
步骤 1	表面风化	0	16	8	66.7
		1	1	29	96.7
	总体百分比				83.3

a. 分界值为 .500

图 5.3 预测结果分析图

由图可知模型在对风化的拟合程度较好，准确率达到 96.7%，而其在无风化的预测显得不足。但因为总体准确百分比为 83.3%，我们任然认为模型是可靠的。

5.3 CART 决策树模型的建立与求解

5.3.1 CART 决策树模型的建立

问题一的后半部分要求分析文物表面有无风化化学成分的统计规律，并根据检测数据对其风化前的化学成分含量进行预测。对此我们建立决策树模型，尝试依据玻璃中各类氧化物的含量将其分为高钾无风化、高钾风化、铅钡无风化和铅钡风化四种类型，并以真实结果进行比较。最后借助分类时所依据的各类氧化物化学含量的数值来探究这四类玻璃化学成分的统计规律。

构建决策树模型，首先需确定各类氧化物在玻璃分类中的重要程度。以氧化物 SiO_2 为例，设样本的个数为D，我们将其分为 4 类，其中每一类的样本数记作C_i，此时根据经验熵的定义，所有样本的经验熵$H(D)$可表示为：

$$H(D) = -\sum_{i=1}^{4} \frac{|C_i|}{|D|} log_2 \frac{|C_i|}{|D|} \tag{6}$$

而 SiO_2 的条件熵$H(D|A)$可表示为：

$$H(D|A)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i) \tag{7}$$

其中n为 SiO_2 不同的取值。但由于 SiO_2 为连续性变量，不能有限地划分为n类，故我们利用二分法对其进行离散化。将 SiO_2 的含量从小到大排序，记为$\{s_1,s_2,\dots,s_n\}$由于其为连续属性，我们将S_a作为划分点，其中：

$$S_a=\left\{\frac{a_i+a_{i+1}}{2}|1\le i\le n-1\right\} \tag{8}$$

此时，划分点不同，SiO_2 的条件熵也不同，为了使划分的结果为最优，我们有：

$$g(D,\mathrm{SiO2})=\max\left[H(D)-\sum_{\lambda\in\{-,+\}}\frac{|D_s^{\lambda}|}{|D|}H\left(D_s^{\lambda}\right)\right] \tag{9}$$

此时$g(D,\mathrm{SiO2})$被称作 SiO_2 的信息增量，而我们选择使$g(D,\mathrm{SiO2})$取得最大值的S_a作为最佳的划分点。同理，我们可以计算出其余各类氧化物的信息增量$g(D,\mathrm{x})$，并且选择决策树的首个结点A为：

$$A=\max\{g(D,\mathrm{x}_1),g(D,\mathrm{x}_2),\dots,g(D,\mathrm{x}_{14})\} \tag{10}$$

以此类推，完成对决策树模型的构建，并以最优的划分点S_a来表示每一类氧化物在不同类别中的统计规律。

5.3.2 模型的求解

首先我们先对数据进行预处理，将数据分为高钾风化类，高钾无风化类，铅钡风化类和铅钡无风化类。并分别对其赋值：

表 5.4 分类赋值表

分类赋值				
类别	高钾风化	高钾无风化	铅钡风化	铅钡无风化
赋值	1	2	3	4

我们将 14 种化学成分占比作为定量数据，分类的赋值作为定类数据导入模型中，通过计算机拟合得到 CART 决策树，其参数表格及可视化图形如下：

表 5.5 分类赋值表

参数名	参数值
训练用时	0.007s
数据切分	0.7
数据洗牌	是
节点分裂评价准则	gini
特征划分点选择标准	best
划分时考虑的最大特征比例	None
内部节点分裂的最小样本数	2
叶子节点的最小样本数	1
叶子节点的最大数量	50
树的最大深度	10

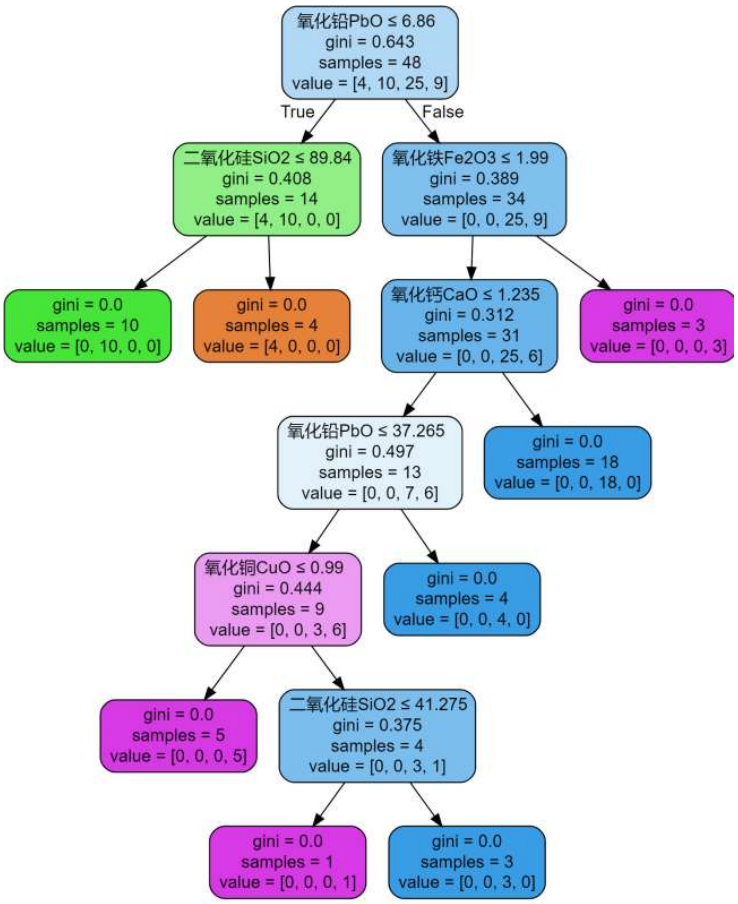

图 5.4 决策树结构

5.3.3 模型的结果

根据树形结构可得到不同类别玻璃有无风化化学成分含量的统计规律：

表 5.6 不同类别化学成分统计规律表

统计规律	类别
$PbO \leqslant 6.86$，$SiO_2 > 89.84$	1
$PbO \leqslant 6.86$，$SiO_2 \leqslant 89.84$	2
$Pb > 6.86$，$Fe_2O_3 > 1.99$	3
$Pb > 6.86$，$Fe_2O_3 \leqslant 1.99, CaO > 1.235$	3
$Pb > 37.265$，$Fe_2O_3 \leqslant 1.99, CaO \leqslant 1.235$	3
$6.86 < Pb \leqslant 37.265$，$Fe_2O_3 \leqslant 1.99, CaO \leqslant 1.235$，$CuO \leqslant 0.99$	3
$6.86 < Pb \leqslant 37.265$，$Fe_2O_3 \leqslant 1.99, CaO \leqslant 1.235$，$CuO > 0.99, SiO_2 > 41.275$	3
$6.86 < Pb \leqslant 37.265$，$Fe_2O_3 \leqslant 1.99, CaO \leqslant 1.235$，$CuO > 0.99, SiO_2 \leqslant 41.275$	4

根据上表我们得到由化学成分判别玻璃制品类型的判断依据。

5.3.4 模型的检验

我们将通过计算模型的准确率、召回率、精确率以及 F1 值对模型进行评价。先给出如下定义，不妨设：

TZ：准确预测且为正类数。
TF：准确预测且为负类数。
FZ：将负类数错误预测为正类数。
FF：将正类数错误预测为负类数。
可得到模型检验的公式为：

$$精确率 = \frac{TZ}{(TZ + FZ)} \tag{11}$$

$$召回率： = \frac{TZ}{(TZ + FF)} \tag{12}$$

$$准确率： \frac{(TZ + TF)}{(TZ + FZ + TF + FF)} \tag{13}$$

$$F1 = \frac{2 \times 精确率 \times 召回率}{(精确率 + 召回率)} \tag{14}$$

通过求解得到模型评价结果：

表 5.7 模型评价参数表

	准确率	召回率	精确率	F1
训练集	1	1	1	1
测试集	0.952	0.952	0.96	0.953

准确率，召回率、精确率指标都是越大模型越好，改模型在这三个指标中都达到了 0.95 以上，证明模型是可靠的。F1 值是精确率和召回率的调和平均，其值为 0.953 说明了模型在对精确率和召回率的兼顾上做的也很好。

5.4.1 平均变化量预测模型

传统的预测问题可以通过机器学习的方法求解，但由于现在我们所能利用的数据十分有限，不能保证机器学习的可靠性，故我们采用最为简单的利用平均变化量，利用逆向思维对问题进行求解。

首先我们对四类玻璃中的没一类化学物质求平均值，不妨设$x_1, x_2, \ldots, x_n$为每类化学成分的样本观测值。

其平均值为：

$$AVR = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{15}$$

将两类玻璃风化前后的每一化学成分的平均值做差可得到平均变化量：

$$\Delta = AVR_{风化} - AVR_{未风化} \tag{16}$$

将风化玻璃的各化学成分减去平均变化量，即可得到我们预测的玻璃在风化前的化学成分含量。

5.4.2 模型求解与结果

利用 Excel 中的内置函数可以对四类玻璃各化学成分的平均值进行求解：

表 5.8 四类玻璃各化学成分的平均值

分类	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
1	93.96333	0	0.543333	0.87	0.196667	1.93	0.265	1.561667	0	0	0.28	0	0	0
2	67.98417	0.695	9.330833	5.3325	1.079167	6.62	1.931667	2.4525	0.411667	0.598333	1.4025	0.041667	0.196667	0.101667
3	24.91269	0.216154	0.133462	2.695385	0.65	2.97	0.584615	2.275769	43.31385	11.80731	5.27730769	0.418462	0.068462	1.366154
4	54.65957	1.682609	0.218696	1.320435	0.640435	4.456087	0.736522	1.431739	22.08478	9.001739	1.04913043	0.268261	0.046522	0.15913

对其进行做差，可以得到平均变化量：

表 5.9 玻璃风化前后各化学成分的平均变化量

分类	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
'1-2	25.97917	-0.695	-8.7875	-4.4625	-0.8825	-4.69	-1.66667	-0.89083	-0.41167	-0.59833	-1.1225	-0.04167	-0.19667	-0.10167
'3-4	-29.7469	-1.46645	-0.08523	1.37495	0.009565	-1.48609	-0.15191	0.84403	21.22906	2.805569	4.228177	0.150201	0.02194	1.207023

综上所述，按照玻璃类型选择化学物质平均变化量，对其做差即可得到预测的风化前的化学成分含量，在此展示部分数据，全部结果放入支撑集：

表 5.10 预测玻璃风化前各化学成分

文物采样点	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	66.02687	0	1.135234	0.96505	1.170435	7.216087	2.011906	0	26.20094	0	0	0.039799	0	0
7	66.65083	0	0	5.5325	0	6.67	1.836667	4.130833	0	0	1.7325	0	0	0
8	49.88687	0	0	0.10505	0	2.826087	0	9.56597	7.450936	28.42443	0	0.219799	0	1.372977
08严重风化点	34.35687	0	0	1.81505	0	2.596087	0	2.29597	11.22094	27.81443	3.331823	0.379799	0	13.82298
9	69.04083	0	9.3775	5.0825	0	6.01	1.986667	2.440833	0	0	1.4725	0	0	0
10	70.79083	0	9.7075	4.6725	0	5.5	1.926667	1.730833	0	0	0	0	0	0
11	63.33687	0	0.295234	2.13505	0.700435	4.176087	0	4.08597	4.160936	11.80443	5.151823	0.219799	0	0
12	68.31083	0	9.7975	5.1825	0	6.15	1.956667	2.540833	0	0	1.2725	0	0	0
19	59.38687	0	0	1.55505	0.580435	5.056087	1.481906	2.66597	21.59094	2.544431	4.601823	0.039799	0	0

六、问题二的模型建立与求解

6.1 数据预测量

为了解决分类问题，将风化和无风化的玻璃直接进行分类是不合理的，为此我们将统一使用玻璃风化前各化学成分的含量数据。风化玻璃的数据将利用上一问中其无风化的化学成分含量预测数据替代。

6.2 随机森林分类模型的建立与求解

6.2.1 随机森林分类模型的建立

问题二的前半部分要求我们对两种玻璃的分类规律进行探究。对此我们建立随机森林分类模型。通过将表单二中的数据作为训练集与测试集，在建立分类模型的同时，对玻璃中各类氧化物成分比例对分类的重要性进行计算，从而探究出高钾玻璃与铅钡玻璃的划分主要是依据哪几类氧化物的成分比例。

对于该随机森林分类模型而言，随机分为数据随机与特征随机。其中数据随机是指对于表单二中的数据，可以有放回地多次抽取；特征随机是指对于玻璃中的n类氧化物，存在常数$k < n$，使得随机森林中的每棵树的构建，都能从n类氧化物中随机选取k类。

其具体流程如下：

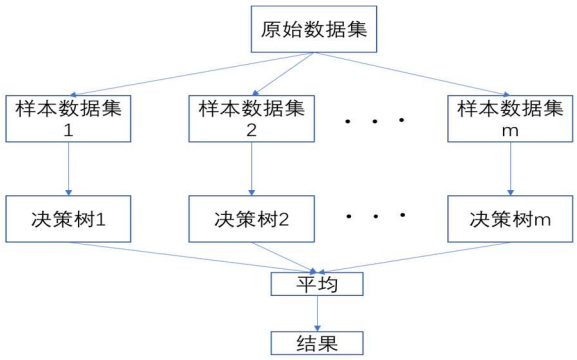

图 6.1　随机森林分类模型构建流程图

通过该过程，我们分别构建了m棵决策树，分别记为T_1、T_2、……、T_m。通过机器学习，对于这m棵树我们可以得到一个模型分类序列，记为:

$$\{h_1(x), h_2(x), \dots, h_m(x)\} \tag{17}$$

在此基础上，采用多数投票法作为随机森林分类模型的最终分类结果：

$$H(x) = argmax \sum_{i=1}^{m} I\left(h_m(x) = a\right) \tag{18}$$

而对于各类氧化物的重要程度，我们设训练的样本数为N，则对于第i棵决策树T_i，将其随机抽样中未曾抽到过的数据记为D_i，并利用T_i将其分类，其中正确的个数记为$R_i^{(l)}$。同时我们对D_i中第j类氧化物E_j的数据进行扰动，并再次利用T_i将其分类，此时正确的个数记为$R_{ij}^{(l)}$。综上，氧化物E_j在第i棵树T_i中的重要性I_{ij}可以表示为：

$$I_{ij} = \frac{1}{N}\sum_{l=1}^{N}\left(R_i^{(l)} - R_{ij}^{(l)}\right) \tag{19}$$

我们将每棵树中氧化物E_j的重要性程度I_{ij}进行综合，并进行归一化处理，即可得到在随机森林分类模型中各类氧化物的重要程度IMP_j:

$$IMP_j = \frac{1}{m}\sum_{i=1}^{m} I_{ij} \tag{20}$$

通过观察不同氧化物的重要程度IMP_j的取值，即可分析出文物玻璃高钾玻璃与铅钡玻璃的划分主要是依据哪几类氧化物的成分比例。

6.2.2 模型的求解分析

首先数据进行分类，并对其进行赋值：

表 6.1　分类赋值表

分类赋值	
高钾	铅钡
1	2

将 14 种氧化物数据以及表面是否风化作为自变量，以分类值做为因变量建立随机森林模型，通过编程求解得到模型参数如下：

表 6.2 随机森林模型参数表

参数名	参数值
数据切分	0.7
数据洗牌	是
交叉验证	否
节点分裂评价准则	Gini
决策树数量	100
有放回采样	TRUE
袋外数据测试	FALSE
划分时考虑的最大特征比例	Auto
内部节点分裂的最小样本数	2
叶子节点的最小样本数	1
叶子节点中样本的最小权重	0
树的最大深度	10
叶子节点的最大数量	50
节点划分不纯度的阀值	0

通过建立随机森林模型我们将得到 100 棵决策树，决策树中的结点和结点权重即为两类玻璃的分类规律。现在只展示部分决策树的图形结果，具体结果放入附录。

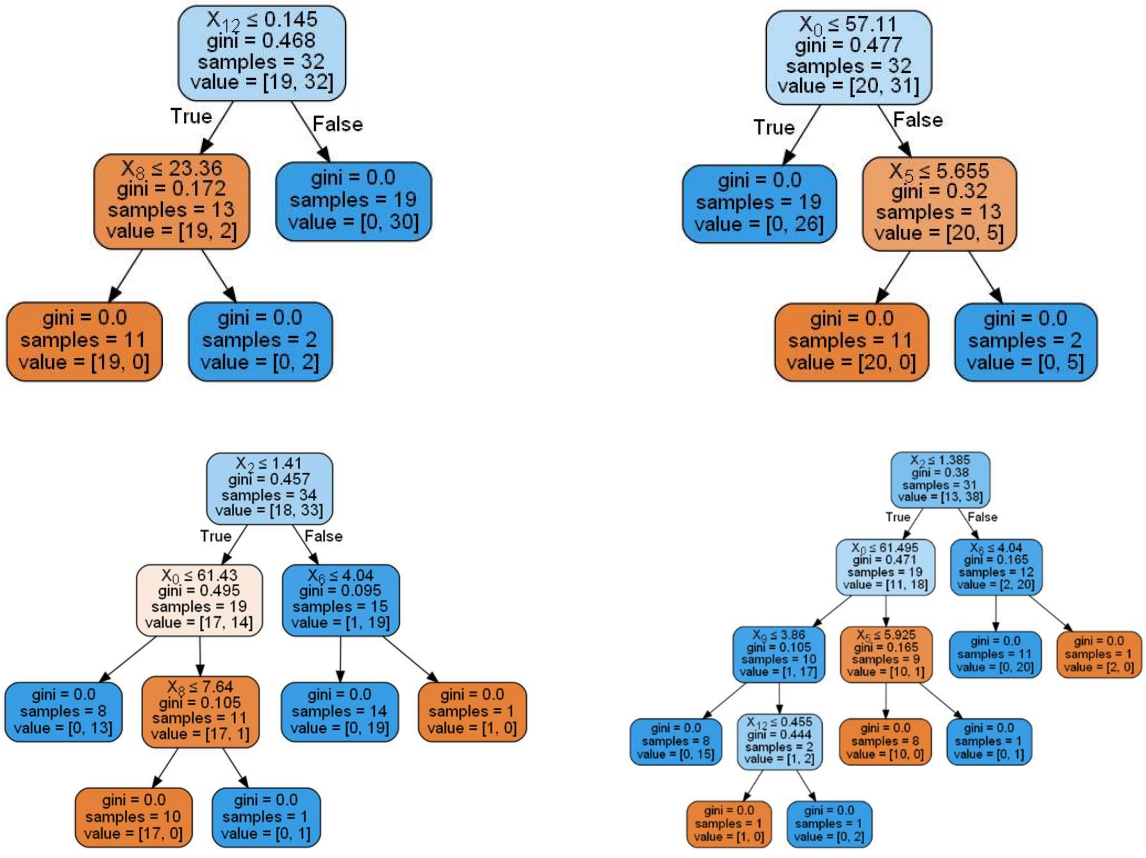

图 6.2 随机森林中的决策树部分展示

6.2.3 结果与分析

通过模型输出，我们可以得到各自变量对结果的影响权重值，即自变量的重要程度。

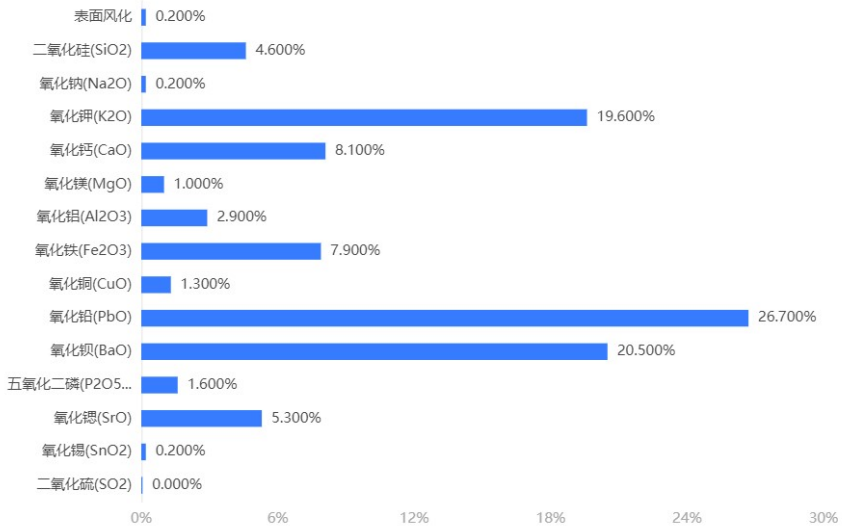

图 6.3 特征重要性

我们发现氧化铅和氧化钡在模型中所占比例很大，说明两者为区分两类玻璃的主要依据，这与题目中所给出的铅钡玻璃氧化铅、氧化钡的含量较高这一信息吻合，从而也证明模型的合理性。

现在给出其中最主要的一颗树：

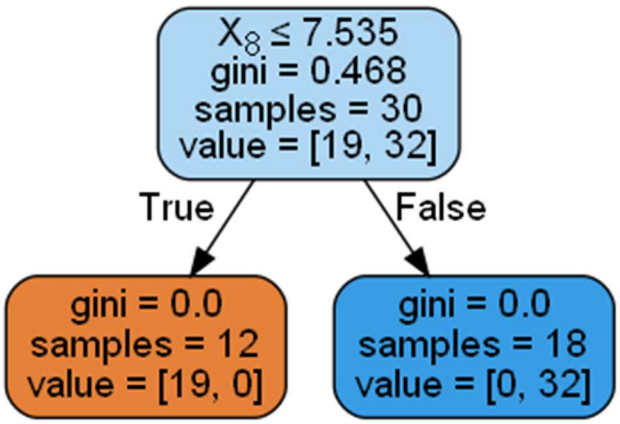

图 6.4 氧化铅决策树

综上所述，高钾玻璃与铅钡玻璃的划分最主要的依据是氧化铅的含量，若玻璃中的氧化铅含量大于 7.535，则为铅钡类玻璃，反之则为高钾玻璃。

6.2.4 模型的检验

我们将沿用模型一中对决策树的评价方法对该随机森林进行评价，并进一步给出预测数据集的混淆矩阵热力图，其模型评估结果如下：

表 6.3 模型评价参数表

	准确率	召回率	精确率	F1
训练集	1	1	1	1
测试集	1	1	1	1

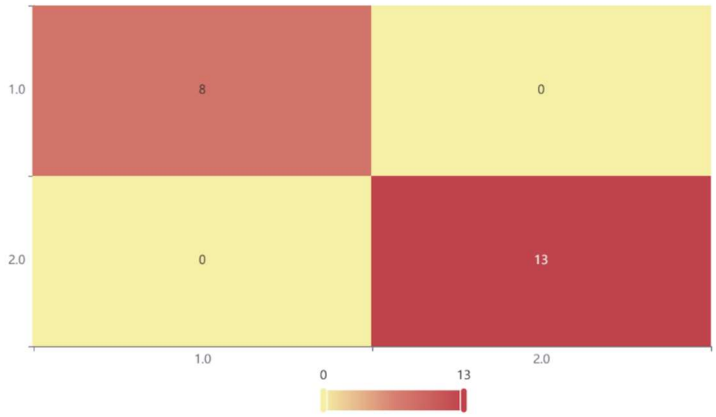

图 6.4 测试集混淆矩阵热力图

混淆矩阵中，矩阵的对角线表示预测准确的数据个数，预测准确个数越多，其颜色越深，模型的准确度也就更高。

从上图可知模型对数据的预测达到了百分之百的准确性，其中一方面是因为我们建立的模型准确可靠；另一方面也是因为样本数据太少，导致的完全拟合效果。

我们分析模型输出的各特征主要程度，其中氧化铅、氧化钡的占比最大，其次为氧化钾。这是因为铅钡玻璃在烧制过程中加入铅矿石作为助熔剂，而钾玻璃是以含钾量高的物质如草木灰作为助熔剂烧制而成的。结果与实际情况十分吻合，有效证明了模型的合理性。

6.3 K 均值聚类模型的建立与求解

6.3.1 K 均值聚类模型的建立

问题二的后半部分要求我们在高钾玻璃与铅钡玻璃的基础上依据化学成分进行亚分类。对此，我们首先根据问题一中预测的结果对已风化玻璃的各类氧化物含量进行修正，并将所有样品依照高钾玻璃与铅钡玻璃分为两类。

这里以高钾玻璃为例。对于每件样品中 14 种氧化物的成分含量，依据前文随机森林模型中所计算得到的重要程度IMP_j，我们选取其中的a类作为样本i中的一组a维的向量h_i：

$$h_i = (x_1, x_2, \dots, x_a)^T$$

假设我们希望将数据集分为k类，则可以在x_i中，随机选取 k 个作为初始分类的图心μ_1、μ_2、…、μ_k随后遍历h_i，计算每个样本到各个图心的距离，这里的距离采用欧氏距离的平方：

$$d(h_i, \mu_j) = \sum_{k=1}^{14} \left(h_{ki} - \mu_{k_j}\right)^2 \tag{21}$$

计算完成后，选取距离最小的图心，将该样本归为这一类，重复此过程，对所有样本进行分类。分类完成后，再次找到这k个类别的中心，作为新的图心，并再次计算每个样本到各个图心的距离，重复迭代，直到收敛为止。

这里我们令$k = 3$，利用 K 均值聚类模型，即可完成对高钾玻璃的亚分类；重复此过程，亦可完成对铅钡玻璃的亚分类。

6.3.2 模型的求解与分析

为了选取合适的化学成分，我们可以继续利用随机森林模型，以 14 种化学成分为自变量，以是否发生风化为因变量建立随机森林模型。选择模型输出中自变量特征较大的化学成分为划分依据。利用 K 均值聚类，对每组数据进行分类。

首先对高钾类玻璃进行分析，通过模型得到的化学成分主要性如下：

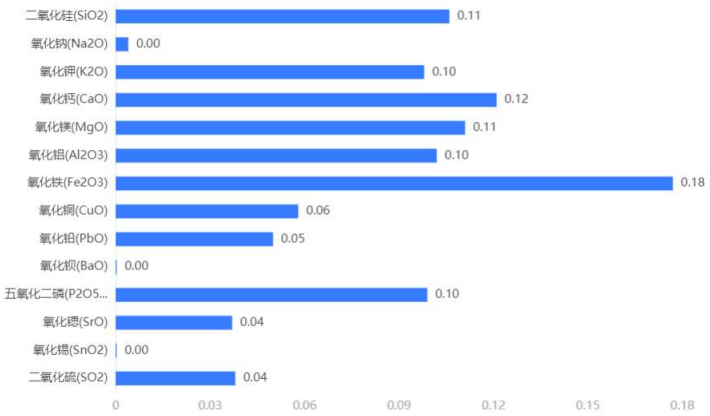

图 6.5 高钾类自变量主要程度

我们将选取氧化铁、氧化钙、氧化镁和氧化硅，进行聚类，经过分析分为两类，其聚类中心比分三类的较明显，故选择将高钾玻璃分为两类，其聚类中心如下：

表 6.4 高钾玻璃聚类中心

最终聚类中心

	聚类	
	1	2
氧化铁Fe2O3	2.08	.20
二氧化硅SiO2	63.20	91.08
氧化钙CaO	5.21	.90
氧化镁MgO	1.10	.31

接着运用同样的方法，求解铅钡类玻璃的化学成分主要度：

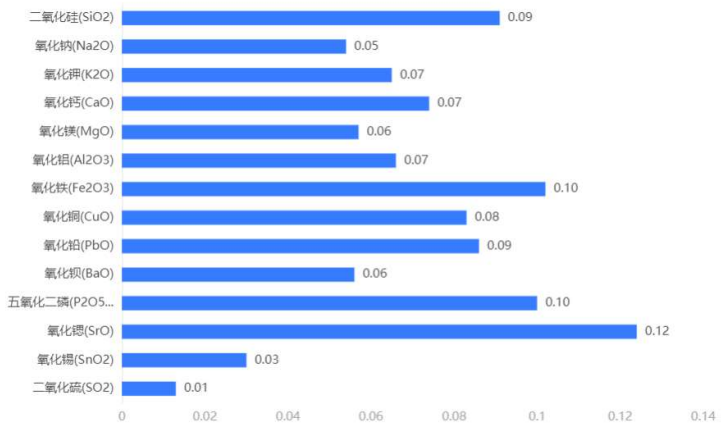

图 6.6 铅钡类自变量主要程度

我们将选取氧化锶、五氧化二磷和氧化铁进行聚类，经过分析分为两类，其聚类中心比分三类的较明显，故选择将铅钡玻璃分为两类，其聚类中心如下：

表 6.5 铅钡玻璃聚类中心

最终聚类中心

	聚类	
	1	2
氧化铁(Fe203)	0.583532144	1.043334191
氧化锶(SrO)	0.282699740	0.262937998
五氧化二磷(P205)	0.364924749	5.481402109

6.3.3 结果与合理性分析

我们将给出高钾玻璃的两种亚类结果，并将其与是否风化进行对比，探究之间是否存在一定的关系：

表 6.6 高钾玻璃亚类表（部分数据）

文物采样点	是否风化	聚类成员编号	到聚类中心的距离
01	无风化	1	6.24379
03 部位 1	无风化	2	4.19800
03 部位 2	无风化	1	1.63307
04	无风化	1	3.32550
05	无风化	1	2.82184
06 部位 1	无风化	1	6.91256
06 部位 2	无风化	1	5.25436
07	风化	2	1.58883
09	风化	2	3.96242

综合图表我们可知，所分的亚类与是否风化存在极强的关系，即第一类玻璃不易风化，第二类玻璃易风化。经检验，只有一项数据不满足此规律，足以证明该分类方案是合理的。

即我们认为氧化铁含量在 2.08，二氧化硅含量在 63.20，氧化钙含量在 5.21，氧化镁含量在 1.1 左右的高钾类玻璃为高钾抗风化玻璃；氧化铁含量在 0.2，二氧化硅含量在 91.08，氧化钙含量在 0.9，氧化镁含量在 0.31 左右的高钾类玻璃为高钾不抗风化玻璃；

同理给出铅钡的两种亚类划分结果：

表 6.7 铅钡玻璃亚类表（部分数据）

文物编号	是否风化	聚类成员编号	到聚类中心的距离
2	风化	1	1.49413
8	风化	1	0.69111
8	风化	2	2.39226
11	风化	2	1.095
19	风化	2	1.00787

20	无风化	2	0.59921
23	无风化	1	0.68987
24	无风化	1	0.88578
25	无风化	1	0.98565

经过分析我们认为 1 类成员为抗风化型，2 类成员为不抗风化型。经检验，该结论的准确性为 73.4%，模型仍然具有一定的合理性。

即我们认为氧化铁含量在 0.58，氧化锶含量在 0.28，五氧化二磷含量在 0.365 左右的高钾类玻璃为铅钡抗风化玻璃；氧化铁含量在 1.04，氧化锶含量在 0.26，五氧化二磷含量在 5.481 左右的高钾类玻璃为铅钡不抗风化玻璃。

6.3.4 模型敏感性检验

我们将分别在两个聚类模型中随机抽取出一条带成员信息的数据，将剩余的数据再次进行聚类分析，我们将得到一组新的聚类中心，通过比较聚类中心发生偏移的大小以及被保留的数据在新的聚类中心下，成员信息是否发生改变来判断模型的敏感性。若聚类中心变动较大且成员信息发生改变即认为模型敏感性强，反之则认为模型的稳定性强。该过程可重复多次已保证结果的可信度。

首先对高钾类玻璃聚类模型进行检验，得到去掉某组数据后的聚类模型：

最终聚类中心

	聚类	
	1	2
氧化铁Fe2O3	2.08	.20
二氧化硅SiO2	63.20	90.82
氧化钙CaO	5.21	.87
氧化镁MgO	1.10	.36

最终聚类中心

	聚类	
	1	2
氧化铁Fe2O3	2.08	.18
二氧化硅SiO2	63.20	90.55
氧化钙CaO	5.21	.93
氧化镁MgO	1.10	.36

图 6.7 去掉某组数据后的聚类中心(高钾类)

通过计算，三个模型的聚类中心差距很小，仅为小数点后两位的差距，因此保留的数据代入检验模型中，分类也不会发生变化，针对高钾玻璃建立的亚类划分模型是稳定的。

同理我们对铅钡类玻璃进行检验：

最终聚类中心

	聚类	
	1	2
氧化铁(Fe2O3)	.577061634	1.043334191
氧化锶(SrO)	.283634018	.262937998
五氧化二磷(P2O5)	.363636885	5.481402109

最终聚类中心

	聚类	
	1	2
氧化铁Fe2O3	.6002044913	1.043334191
氧化锶SrO	.2822111801	.2629379984
五氧化二磷P2O5	.3753511705	5.481402109

图 6.8 去掉某组数据后的聚类中心（铅钡类）

通过计算，三个模型的聚类中心差距同样很小，中心聚类差值最大为 0.023，因此保留的数据代入检验模型中，分类也不会发生变化，针对铅钡玻璃建立的亚类划分模型也是稳定的。

七、问题三的模型建立与求解

7.1 数据预处理

根据假设，我们将表单 3 中的缺失值用 0 替换，并计算每种玻璃的化学成分总量，通过检验得知其均满足题目所给条件，均可视为有效数据。

7.2 分类模型的建立

问题三要求我们分析未知类别玻璃文物的化学成分，并对其进行合理地分类。对此我们首先依据前文中所建立的随机森林分类模型将未知类别玻璃文物分为高钾玻璃与铅钡玻璃两大类，其具体流程如下：

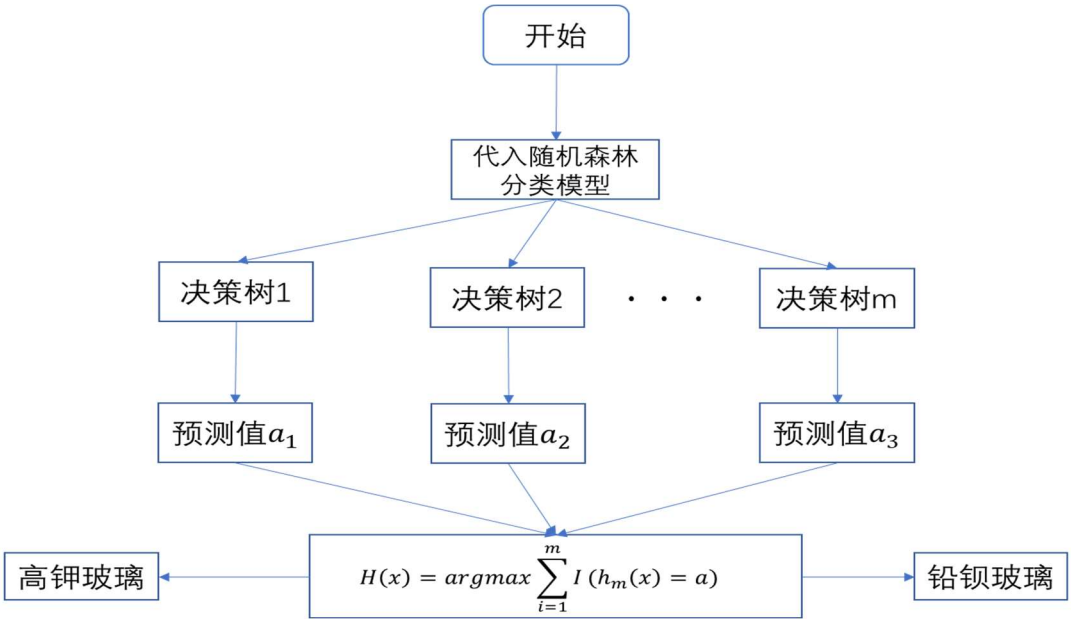

图 7.1　随机森林分类流程图

在完成对文物玻璃大类的划分后，我们建立了分类模型。通过计算依据该样品各类氧化物的含量到此类玻璃中各亚类图心的距离，完成对该样品的具体分类。

假设该样品经由随机森林分类模型分类后归为高钾玻璃类。由问题二可知，我们依据 A、B、C 的含量将高钾玻璃分为三个亚类。此时每个亚类的图心$\mu_{(i)} = (\mu_{i1}, \mu_{i2}, \mu_{i3})$。利用欧式距离的平方，我们可以计算得出：

$$d(y, \mu_{(i)}) = \sum_{k=1}^{3} (y - \mu_{ik})^2 \tag{22}$$

$$\mathrm{d} = \min\{d(y, \mu_{(1)}), d(y, \mu_{(2)}), d(y, \mu_{(3)})\} \tag{23}$$

从而完成对分类模型的建立：

$$\begin{cases} \mathrm{d} = d\left(y, \mu_{(1)}\right)，\ 该样本为A类 \\ \mathrm{d} = d\left(y, \mu_{(2)}\right)，\ 该样本为B类 \\ \mathrm{d} = d\left(y, \mu_{(3)}\right)，\ 该样本为C类 \end{cases} \tag{24}$$

同理，若该样品经由随机森林分类模型分类后归为铅钡玻璃类，同样有:

$$\mathrm{d}' = \min\left\{d\left(y, \lambda_{(1)}\right), d\left(y, \lambda_{(2)}\right), d\left(y, \lambda_{(3)}\right)\right\} \tag{25}$$

$$\begin{cases} \mathrm{d}' = d\left(y, \lambda_{(1)}\right)，\ 该样本为D类 \\ \mathrm{d}' = d\left(y, \lambda_{(2)}\right)，\ 该样本为E类 \\ \mathrm{d}' = d\left(y, \lambda_{(3)}\right)，\ 该样本为F类 \end{cases} \tag{26}$$

7.2 模型的求解与结果

将表单 3 中的数据代入到随机森林中，我们可将这批文物分为高钾和铅钡两个大类结果如下：

表 7.1 大类划分表

文物编号	表面风化	类别
A1	无风化	高钾
A2	风化	铅钡
A3	无风化	铅钡
A4	无风化	铅钡
A5	风化	铅钡
A6	风化	高钾
A7	风化	高钾
A8	无风化	铅钡

接着我们可以通过问题一所给出的预测模型，预测已风化文物在风化前的化学成分含量，我们将得到的 8 个无风化文物的化学成分，分别和第二问所建立的聚类中心求距离。其结果如下：

表 7.2 K 均值聚类表

文物编号	表面风化	距离 1	距离 2
A1	无风化	5.43267941	8.5524543
A2	风化	9.775477	11.24575
A3	无风化	6.907658	8.164433
A4	无风化	10.05621	11.10471
A5	风化	5.562306	0.428586
A6	风化	9.24294335	6.88697752
A7	风化	9.2335	6.87787
A8	无风化	1.315128	5.409705

通过比较得知 A1 为高钾抗风化玻璃；A6，A7 为高钾不抗风化玻璃；A2，A3，A4，A8 为铅钡抗风化玻璃；A5 为铅钡不抗风化玻璃。

7.3 模型的敏感性分析

我们沿用上一问的思路，对随机森林在训练模型的过程中进行数据洗牌，如果训练处理的模型和之前的模型的自变量主要程度最大的几个化学成分与原模型给出的一致，则说明模型稳定；反之说明模型敏感。

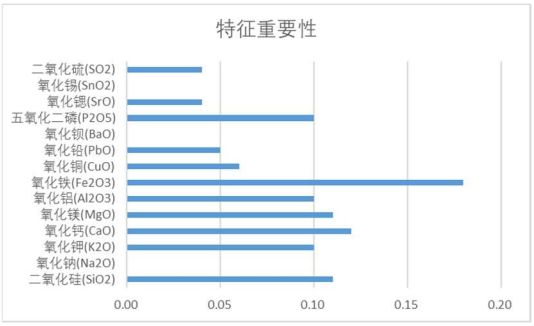

图 7.2 高钾随机森林特征权重

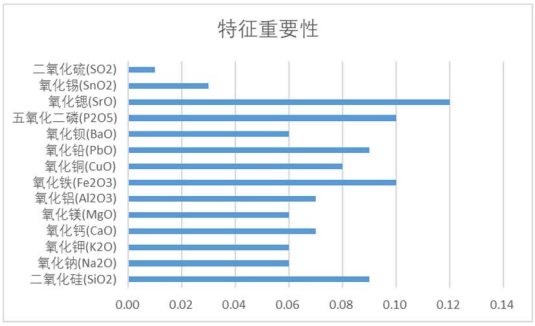

图 7.3 铅钡随机森林特征权重

我们对数据重新洗牌分析，得到各类型玻璃的特征权值（见图 7.2，7.3），比较图 7.2 与图 6.5 以及图 7.3 与图 6.6，发现最重要的化学成分没有改变。，这说明模型具有稳定性。

八、问题四的模型建立与求解

8.1 相关系数矩阵模型的建立与求解

8.1.1 相关系数矩阵模型的建立

问题四首先要求我们从内部对每种类型玻璃的化学成分之间的关联进行分类，随后再从类与类之间比较这种关联的差异性。对此，我们首先建立相关矩阵模型来探究每类玻璃化学成分之间的关联。

对于每一类玻璃，它的化学成分主要为 14 种氧化物，其构成一个 14 维的向量Y：

$$Y = (y_1, y_2, \dots, y_{14})$$

利用 Pearson 相关系数的计算公式，我们可以计算出这 14 种氧化物两两之间的相关系数ρ_{ij}，从而将向量Y转化为一个相关矩阵R:

$$\rho_{ij} = \frac{cov(y_i, y_j)}{\sigma(y_i)\sigma(y_j)} = \frac{E(y_i y_j) - E(y_i)E(y_j)}{\sqrt{E(y_i) - E^2(y_i)}\sqrt{E(y_j) - E^2(y_j)}} \tag{27}$$

$$R = \begin{Bmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \rho_{n1} & \rho_{n2} & \cdots & \rho_{nn} \end{Bmatrix}$$

由于R关于对角线对称，且对角线上的元素都为 1，故我们可将 R 简化为R′:

$$R' = \begin{Bmatrix} 1 & & & \\ \rho_{21} & 1 & & \\ \cdots & \cdots & \cdots & \\ \rho_{n1} & \rho_{n2} & \cdots & 1 \end{Bmatrix}$$

通过观察矩阵中元素ρ_{ij}的取值，即可判断出氧化物i与氧化物j之间的关联关系。由此，该模型可以写作：

$$\begin{cases} 0.7 \le \rho_{ij} \le 1，\text{氧化物}i\text{与氧化物}j\text{强正相关} \\ 0.4 \le \rho_{ij} < 0.7，\text{氧化物}i\text{与氧化物}j\text{中度正相关} \\ 0.2 \le \rho_{ij} < 0.4，\text{氧化物}i\text{与氧化物}j\text{弱正相关} \\ -0.2 < \rho_{ij} < 0.2，\text{氧化物}i\text{与氧化物}j\text{几乎不相关} \\ -0.4 < \rho_{ij} \le -0.2，\text{氧化物}i\text{与氧化物}j\text{弱负相关} \\ -0.7 < \rho_{ij} \le -0.4，\text{氧化物}i\text{与氧化物}j\text{中度负相关} \\ -1 \le \rho_{ij} \le -0.7，\text{氧化物}i\text{与氧化物}j\text{强负相关} \end{cases}$$

8.1.2 矩阵二范数模型的建立

而对于不同类之间关联关系的差异性，则可以通过比较它们各自的相关矩阵来实现。设R_i、R_j为各类氧化物在第i类玻璃与第j类玻璃中的相关矩阵，此时我们将两个矩阵做差，并记为R_{ij}，即：

$$R_{ij} = R_i - R_j \tag{28}$$

此后，通过度量R_{ij}的范数，来衡量这两个矩阵的差异性大小。此处我们选用 2 范数，即：

$$\left\| R_{ij} \right\|_2 = \sqrt{\lambda_{\max}\left(R_{ij}^T * R_{ij}\right)} \tag{29}$$

通过计算，比较各类玻璃之间$\left\| R_{ij} \right\|_2$的大小，即可比较出不同类玻璃之间关联关系的差异性。

8.2 模型求解

8.2.1 同类别玻璃化学成分关联关系求解

首先我们将按照第二问所分好的四类，分别导入 Python，用 pandas 中的内置函数对相关系数矩阵进行求解，结果如下：

表 8.1 铅钡抗风化相关系数矩阵

	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
二氧化硅	1	0.07801436	-0.0276634	-0.3617	0.217818	0.386018	0.044084	-0.41371	-0.47454	-0.55583	-0.181983779	-0.61626	0.217651	-0.320471
氧化钠	0.07801436	1	-0.1032784	-0.19118	0.164866	0.086944	-0.22302	-0.0471	-0.17021	-0.07922	-0.255343835	-0.03905	-0.02152	-0.155487
氧化钾	-0.0276634	-0.1032784	1	0.368007	0.400267	0.319428	0.2579	-0.21744	0.039373	-0.13262	-0.119856162	-0.05839	0.274565	0.1485032
氧化钙	-0.3616955	-0.1911761	0.36800747	1	0.341077	0.334818	0.510999	-0.14784	0.23462	-0.17992	0.296057509	0.280705	0.340249	0.0544999
氧化镁	0.21781751	0.16486588	0.40026707	0.341077	1	0.526624	0.186866	-0.22245	-0.14616	-0.37626	-0.054238544	0.063965	0.345015	-0.217468
氧化铝	0.38601803	0.08694381	0.31942798	0.334818	0.526624	1	0.158898	-0.23066	-0.40343	-0.31204	-0.148575964	-0.126	0.516238	-0.111127
氧化铁	0.04408353	-0.223023	0.25790035	0.510999	0.186866	0.158898	1	-0.25071	0.127523	-0.32609	0.306313483	-0.04535	0.287866	-0.139618
氧化铜	-0.4137075	-0.0470972	-0.2174423	-0.14784	-0.22245	-0.23066	-0.25071	1	-0.26422	0.761017	-0.111864643	0.256693	-0.16423	0.177053
氧化铅	-0.4745444	-0.1702088	0.03937296	0.23462	-0.14616	-0.40343	0.127523	-0.26422	1	-0.3256	-0.104238995	0.353412	-0.20686	-0.284047
氧化钡	-0.5558283	-0.07922	-0.1326226	-0.17992	-0.37626	-0.31204	-0.32609	0.761017	-0.3256	1	0.112538733	0.251085	-0.12063	0.5573613
五氧化二磷	-0.1819838	-0.2553438	-0.1198562	0.296058	-0.05424	-0.14858	0.306313	-0.11186	-0.10424	0.112539	1	0.16494	-0.00018	0.447634
氧化锶	-0.6162625	-0.0390505	-0.0583906	0.280705	0.063965	-0.126	-0.04535	0.256693	0.353412	0.251085	0.164940428	1	-0.08156	0.1324688
氧化锡	0.21765067	-0.0215202	0.2745649	0.340249	0.345015	0.516238	0.287866	-0.16423	-0.20686	-0.12063	-0.000179416	-0.08156	1	-0.065357
二氧化硫	-0.3204713	-0.1554874	0.1485032	0.0545	-0.21747	-0.11113	-0.13962	0.177053	-0.28405	0.557361	0.447633963	0.132469	-0.06536	1

表 8.2 铅钡不抗风化相关系数矩阵

	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
二氧化硅	1	0	-0.1088057	0.252783	0.318089	0.434277	0.272682	-0.08783	-0.09615	-0.63702	-0.00423842	-0.18525	0.111234	-0.578498
氧化钠	0	0	0	0	0	0	0	0	0	0	0	0	0	0
氧化钾	-0.1088057	0	1	-0.29966	0.063299	0.101773	0.205394	0.501343	-0.40548	0.390546	-0.291212826	-0.3112	-0.20902	-0.20902
氧化钙	0.25278314	0	-0.2996605	1	0.470195	-0.07785	0.279187	-0.55061	0.12993	-0.40963	-0.067967651	-0.23995	-0.03145	-0.114071
氧化镁	0.31808862	0	0.06329852	0.470195	1	0.326622	0.442056	-0.69852	0.346814	-0.5659	-0.106718948	0.18034	0.088546	-0.414389
氧化铝	0.43427741	0	0.1017727	-0.07785	0.326622	1	0.693134	-0.26546	0.010085	-0.41665	0.156702629	0.027137	0.428474	-0.611304
氧化铁	0.2726824	0	0.2053941	0.279187	0.442056	0.693134	1	-0.23045	-0.20672	-0.21899	-0.062714846	-0.26312	0.100292	-0.350461
氧化铜	-0.0878321	0	0.50134298	-0.55061	-0.69852	-0.26546	-0.23045	1	-0.67967	0.649588	-0.141859758	-0.37052	-0.1737	0.1562326
氧化铅	-0.0961499	0	-0.4054811	0.12993	0.346814	0.010085	-0.20672	-0.67967	1	-0.66162	0.400856308	0.423382	-0.04121	-0.292672
氧化钡	-0.6370228	0	0.39054559	-0.40963	-0.5659	-0.41665	-0.21899	0.649588	-0.66162	1	-0.392071024	-0.17142	-0.04782	0.6981849
五氧化二磷	-0.0042384	0	-0.2912128	-0.06797	-0.10672	0.156703	-0.06271	-0.14186	0.400856	-0.39207	1	0.580685	-0.24191	-0.323064
氧化锶	-0.1852468	0	-0.3111954	-0.23995	0.18034	0.027137	-0.26312	-0.37052	0.423382	-0.17142	0.58068456	1	-0.0279	0.1409329
氧化锡	0.11123416	0	-0.2090198	-0.03145	0.088546	0.428474	0.100292	-0.1737	-0.04121	-0.04782	-0.241906197	-0.0279	1	-0.083333
二氧化硫	-0.5784976	0	-0.2090198	-0.11407	-0.41439	-0.6113	-0.35046	0.156233	-0.29267	0.698185	-0.32306364	0.140933	-0.08333	1

表 8.3 高钾抗风化相关系数矩阵

	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
二氧化硅	1	-0.3653151	-0.3440878	-0.83351	0.265595	-0.74034	-0.6351	-0.54434	-0.31456	-0.29155	-0.166418097	-0.04718	0.86937	0.0508962
氧化钠	-0.3653151	1	0.6332073	0.524661	-0.79124	0.341613	-0.22184	-0.09928	0.406706	-0.28828	-0.321211757	-0.53624	-0.21721	-0.440587
氧化钾	-0.3440878	0.6332073	1	0.509088	-0.70099	-0.07397	-0.4377	0.401854	0.499373	0.17695	-0.749731402	-0.39356	-0.27818	-0.163812
氧化钙	-0.8335053	0.52466063	0.50908777	1	-0.47721	0.531782	0.264894	0.486274	0.2142	-0.07332	-0.236304659	-0.48936	-0.91082	0.2497064
氧化镁	0.2655951	-0.7912371	-0.7009856	-0.47721	1	0.11088	0.224554	-0.35364	-0.22183	0.098249	0.366017601	0.590444	0.245893	0.3533759
氧化铝	-0.7403417	0.34161328	-0.0739675	0.531782	0.11088	1	0.584563	-0.13038	0.371966	0.037784	0.387643778	0.178077	-0.57533	-0.148406
氧化铁	-0.6351025	-0.2218436	-0.4377037	0.264894	0.224554	0.584563	1	0.368724	-0.1862	0.262231	0.830289394	0.484762	-0.49802	-0.034371
氧化铜	-0.5443351	-0.099283	0.40185391	0.486274	-0.35364	-0.13038	0.368724	1	-0.0563	0.468685	-0.051054894	0.02062	-0.59038	0.1830748
氧化铅	-0.3145559	0.406706	0.49937332	0.2142	-0.22183	0.371966	-0.1862	-0.0563	1	0.561006	-0.279287918	0.145647	-0.24838	-0.503792
氧化钡	-0.2915468	-0.2882841	0.1769502	-0.07332	0.098249	0.037784	0.262231	0.468685	0.561006	1	0.121927672	0.655986	-0.1896	-0.384571
五氧化二磷	-0.1664181	-0.3212118	-0.7497314	-0.2363	0.366018	0.387644	0.830289	-0.05105	-0.27929	0.121928	1	0.613591	-0.00038	-0.242549
氧化锶	-0.0471847	-0.5362397	-0.3935615	-0.48936	0.590444	0.178077	0.484762	0.02062	0.145647	0.655986	0.613591053	1	0.211604	-0.337401
氧化锡	0.86936968	-0.2172146	-0.2781822	-0.91082	0.245893	-0.57533	-0.49802	-0.59038	-0.24838	-0.1896	-0.000381324	0.211604	1	-0.289764
二氧化硫	0.05089617	-0.440587	-0.1638115	0.249706	0.353376	-0.14841	-0.03437	0.183075	-0.50379	-0.38457	-0.242549161	-0.3374	-0.28976	1

表 8.3 高钾不抗风化相关系数矩阵

	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
二氧化硅	1	-0.2380059	-0.2455559	-0.40475	-0.22467	-0.41038	-0.45007	-0.26195	0.522746	0.230461	-0.174389524	-0.19435	0	0
氧化钠	-0.2380059	1	0.5739192	0.661112	-0.06392	-0.0212	-0.53335	-0.3735	-0.05195	-0.16353	-0.373642946	0.246273	0	0
氧化钾	-0.2455559	0.5739192	1	0.262746	-0.18473	0.020394	-0.30626	-0.66051	-0.40202	-0.34868	-0.234381088	0.270906	0	0
氧化钙	-0.4047453	0.66111216	0.26274618	1	-0.23146	-0.39001	-0.24957	-0.10691	-0.08247	-0.35544	-0.70077586	-0.44436	0	0
氧化镁	-0.2246686	-0.0639213	-0.1847252	-0.23146	1	0.832659	0.459264	0.069246	0.291977	0.575295	0.609593288	0.554224	0	0
氧化铝	-0.4103844	-0.0211976	0.02039358	-0.39001	0.832659	1	0.51554	0.216837	0.05325	0.513622	0.855173568	0.845692	0	0
氧化铁	-0.4500718	-0.5333524	-0.3062563	-0.24957	0.459264	0.51554	1	0.650937	0.182433	0.517067	0.535633045	0.158669	0	0
氧化铜	-0.2619548	-0.3734972	-0.6605061	-0.10691	0.069246	0.216837	0.650937	1	0.278666	0.434393	0.423225267	-0.0006	0	0
氧化铅	0.52274639	-0.0519479	-0.4020157	-0.08247	0.291977	0.05325	0.182433	0.278666	1	0.855032	0.051653306	0.030037	0	0
氧化钡	0.23046061	-0.1635266	-0.3486848	-0.35544	0.575295	0.513622	0.517067	0.434393	0.855032	1	0.501610225	0.439483	0	0
五氧化二磷	-0.1743895	-0.3736429	-0.2343811	-0.70078	0.609593	0.855174	0.535633	0.423225	0.051653	0.50161	1	0.732658	0	0
氧化锶	-0.1943453	0.24627255	0.27090605	-0.44436	0.554224	0.845692	0.158669	-0.0006	0.030037	0.439483	0.732658122	1	0	0
氧化锡	0	0	0	0	0	0	0	0	0	0	0	0	0	0
二氧化硫	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.2.2 不同类别玻璃化学成分关联关系求解

接着我们将利用上诉的四个相关系数矩阵两类做差，通过求解做差后的矩阵二范数用于刻画两两之间的关系：

表 8.5 矩阵二范数值

类别	二范数值
高钾不抗风化-高钾抗风化	5.860381
高钾不抗风化-铅钡不抗风化	8.653242
高钾不抗风化-铅钡抗风化	12.33243
高钾抗风化-铅钡不抗风化	7.146556
高钾抗风化-铅钡抗风化	9.554843
铅钡不抗风化-铅钡抗风化	4.262572

8.3 结论与分析

根据分析相关系数我们可以得知，铅钡抗风化玻璃在氧化铜和氧化钡之间存在强正

相关关系；铅钡不抗风化玻璃中的氧化铝和氧化铁存在强正相关，氧化铜分别与氧化镁和氧化铝存在强负相关；对于高钾抗风化玻璃，二氧化硅与氧化铝、氧化钙和氧化锡存在强负相关，氧化钾分别与氧化镁和五氧化二磷存在较强正相关，氧化铁和五氧化二磷存在强正相关；对于高钾不抗风化氧化钙与五氧化二磷存在强负相关，氧化铝分别与氧化镁，五氧化二磷，氧化锶存在强正相关，氧化钡分别与氧化铅和五氧化二磷存在强正相关关系。

通过比对作差矩阵的二范数，我们发现高钾不抗风化和铅钡抗风化之间化学成分的差异性最大，在不同大类情况下，高钾抗风化与铅钡不抗风化之间的化学成分差异最小。

九、 模型评价与改进

9.1 模型评价与改进

（1）模型的优点：

- 本文建立的随机森林分类模型能够较好解决高维数据分类问题，且可以直接给出各因素的重要要程度，对我们所处理不平衡的数据集，它可以有效平衡误差。
- 本文所建立的分类模型可以和其抗风化性结合，通过数学方法给出了具有显示意义的结果。

（2） 模型的缺点：

- 本文在亚类划分时未考虑到文物的生产年限以及生产技术，该因素对玻璃抗风化能力存在一定影响。

十、 参考文献

[1] SPSS 分析技术:二元 logistic 回归-CDA 数据分析师官网, https://www.cda.cn/view/122775.html,2022.9.16.

[2] 姚登举, 杨静, 詹晓娟. 基于随机森林的特征选择算法[J]. 吉林大学学报(工学版), 2014, 44(01):137-141. DOI:10.13229/j.cnki.jdxbgxb201401024.

[3] 陈姝聿, 侯志力. 浅析古代丝绸之路和中国古代玻璃[J]. 中国民族博览, 2019(05):87-88.

十一、 附录

11.1 支撑材料

问题一	信息赋值化学分类过滤均值.csv 表单 2 数据预处理.csv 信息赋值化学分类过滤均值变化量.csv 预测化学成分风化前数据.csv 按类别与是否风化分 4 类数据.xlsx
问题二	随机森林模型.pkl tree4 高钾玻璃分类数据.xlsx 铅钡玻璃分类数据.xlsx
问题三	表单 3 预测分类数据.xlsx
问题四	范数计算 2.csv

11.2 Python 代码

xlsx2csv.py

```
# xlsx 转 csv
import os
import pandas as pd

base_dir = '相关系数矩阵 result/'

for dirpath, dirnames, filenames in os.walk(base_dir):
    for filename in filenames:
        if filename.endswith('.xlsx'):
            df = pd.read_excel(base_dir + filename)
            df.to_csv(base_dir + filename.replace('.xlsx', '.csv'), index=Fa
lse)
```

信息赋值化学分类过滤均值.py

```
import os

ftype2 = '信息赋值化学分类过滤.csv'
favg = '信息赋值化学分类过滤均值.csv'
fdiff = '信息赋值化学分类过滤均值变化量.csv'
```

```
if os.path.exists(favg):
    os.remove(favg)
if os.path.exists(fdiff):
    os.remove(fdiff)

fin = open(ftype2, 'r', encoding='utf-8_sig')
fout = open(favg, 'w', encoding='utf-8_sig')
fout.write('分类,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)\n')

avg_dict = {}
avg_count = {}

for line in fin:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物'):
        continue
    column = line.split(',')
    for i in range(len(column)):
        if column[i] == '':
            column[i] = '0'
    no = column[-1]  # 类别
    if no not in avg_dict:
        avg_dict[no] = [float(_) for _ in column[6:-1]]
        avg_count[no] = 1
    else:
        for i in range(len(avg_dict[no])):
            avg_dict[no][i] += float(column[i+6])
        avg_count[no] += 1

# 计算均值
for key in avg_dict:
    for i in range(len(avg_dict[key])):
        avg_dict[key][i] /= avg_count[key]


# 写入文件
# for key in avg_dict:
#     fout.write(f'{key},{",".join([str(_) for _ in avg_dict[key]])}\n')
```

```
# 有序处理
keys = sorted(list(avg_dict.keys()))
for key in keys:
    fout.write(f'{key},{",".join([str(_) for _ in avg_dict[key]])}\n')


fin.close()
fout.close()

# 计算变化量
fout = open(fdiff, 'w', encoding='utf-8_sig')
fout.write('分类,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)\n')

s = ''
for v1, v2 in zip(avg_dict['1'], avg_dict['2']):  # 1 高钾风化 2 高钾未风化
    s += f'{v1-v2},'
s = s[:-1]
fout.write(f'\'1-2,{s}\n')

s = ''
for v1, v2 in zip(avg_dict['3'], avg_dict['4']):  # 3 铅钡风化 4 铅钡未风化
    s += f'{v1-v2},'
s = s[:-1]
fout.write(f'\'3-4,{s}\n')

fout.close()
```

信息赋值化学分类过滤推算.py

```
import os

ftype2 = '信息赋值化学分类过滤.csv'
fdiff = '信息赋值化学分类过滤均值变化量.csv'
typebefore = '信息赋值化学分类过滤风化前.csv'
typeafter = '信息赋值化学分类过滤风化后.csv'

temp = [typebefore, typeafter]
for i in temp:
    if os.path.exists(i):
        os.remove(i)
```

```
diff_dict = {}

fin = open(fdiff, 'r', encoding='utf-8_sig')
for line in fin:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('分类'):
        continue
    line = line.split(',')
    diff_dict[line[0]] = line[1:]

fin.close()


fin = open(ftype2, 'r', encoding='utf-8_sig')
fout = open(typebefore, 'w', encoding='utf-8_sig')

fout.write('文物编号,纹饰,类型,颜色,表面风化,文物采样点,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),分类\n')

for line in fin:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物编号'):
        continue
    data = line.split(',')
    mytype = data[-1]
    if mytype == '2' or mytype == '4':
        fout.write(line + '\n')
        continue
    elif mytype == '1':
        for i in range(len(diff_dict["'1-2"])):
            if data[i+6] == '':
                continue
            data[i+6] = str(float(data[i+6]) - float(diff_dict["'1-2"][i]))
    elif mytype == '3':
        for i in range(len(diff_dict["'3-4"])):
```

```
            if data[i+6] == '':
                continue
            data[i+6] = str(float(data[i+6]) - float(diff_dict["'3-4"][i]))
    else:
        print('error')
    fout.write(','.join(data) + '\n')

fin.close()
fout.close()

fin = open(ftype2, 'r', encoding='utf-8_sig')
fout = open(typeafter, 'w', encoding='utf-8_sig')

fout.write('文物编号,纹饰,类型,颜色,表面风化,文物采样点,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),分类\n')

for line in fin:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物编号'):
        continue
    data = line.split(',')
    mytype = data[-1]
    if mytype == '1' or mytype == '3':
        fout.write(line + '\n')
        continue
    elif mytype == '2':
        for i in range(len(diff_dict["'1-2"])):
            if data[i+6] == '':
                continue
            data[i+6] = str(float(data[i+6]) + float(diff_dict["'1-2"][i]))
    elif mytype == '4':
        for i in range(len(diff_dict["'3-4"])):
            if data[i+6] == '':
                continue
            data[i+6] = str(float(data[i+6]) + float(diff_dict["'3-4"][i]))
    else:
        print('error')
    fout.write(','.join(data) + '\n')
```

```
fin.close()
fout.close()
```

信息赋值化学分类过滤风化前 0.py

```
import os

fnin = '信息赋值化学分类过滤风化前.csv'
fnout = '信息赋值化学分类过滤风化前 0.csv'

if os.path.exists(fnout):
    os.remove(fnout)

fin = open(fnin, 'r', encoding='utf-8_sig')
fout = open(fnout, 'w', encoding='utf-8_sig')
fout.write('文物编号,纹饰,类型,颜色,表面风化,文物采样点,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),分类\n')

for line in fin:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物编号'):
        continue
    data = line.split(',')
    for i in range(6, len(data)):
        if data[i] == '' or float(data[i]) < 0:
            data[i] = '0'
    fout.write(','.join(data) + '\n')

fin.close()
fout.close()
```

分类 2 均值及变化量计算.py

```
import os

ftype2 = '基本信息赋值和化学成分比例分类 2.csv'
favg = '分类 2 均值.csv'
fdiff = '分类 2 均值变化量.csv'
```

```
if os.path.exists(favg):
    os.remove(favg)
if os.path.exists(fdiff):
    os.remove(fdiff)

fin = open(ftype2, 'r', encoding='utf-8_sig')
fout = open(favg, 'w', encoding='utf-8_sig')
fout.write('分类,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)\n')

avg_dict = {}
avg_count = {}

for line in fin:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物'):
        continue
    column = line.split(',')
    for i in range(len(column)):
        if column[i] == '':
            column[i] = '0'
    no = column[-1]  # 类别
    if no not in avg_dict:
        avg_dict[no] = [float(_) for _ in column[6:-1]]
        avg_count[no] = 1
    else:
        for i in range(len(avg_dict[no])):
            avg_dict[no][i] += float(column[i+6])
        avg_count[no] += 1

# 计算均值
for key in avg_dict:
    for i in range(len(avg_dict[key])):
        avg_dict[key][i] /= avg_count[key]


# 写入文件
# for key in avg_dict:
#     fout.write(f'{key},{",".join([str(_) for _ in avg_dict[key]])}\n')
```

```
# 有序处理
keys = sorted(list(avg_dict.keys()))
for key in keys:
    fout.write(f'{key},{",".join([str(_) for _ in avg_dict[key]])}\n')


fin.close()
fout.close()

# 计算变化量
fout = open(fdiff, 'w', encoding='utf-8_sig')
fout.write('分类,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)\n')

s = ''
for v1, v2 in zip(avg_dict['1'], avg_dict['2']):  # 1 高钾风化 2 高钾未风化
    s += f'{v1-v2},'
s = s[:-1]
fout.write(f'\'1-2,{s}\n')

s = ''
for v1, v2 in zip(avg_dict['3'], avg_dict['4']):  # 3 铅钡风化 4 铅钡未风化
    s += f'{v1-v2},'
s = s[:-1]
fout.write(f'\'3-4,{s}\n')

fout.close()
```

基本信息和成分比例合并.py

```
import os

# fn_baseinfo = '玻璃文物的基本信息.csv'
fn_baseinfo = '玻璃文物的基本信息赋值.csv'
fn_chem = '已分类玻璃文物的化学成分比例.csv'

# fn_baseinfo_chem = '玻璃文物的基本信息和化学成分比例.csv'
fn_baseinfo_chem = '玻璃文物的基本信息赋值和化学成分比例.csv'
if os.path.exists(fn_baseinfo_chem):
    os.remove(fn_baseinfo_chem)
```

```
fin_baseinfo = open(fn_baseinfo, 'r', encoding='utf-8_sig')
fin_chem = open(fn_chem, 'r', encoding='utf-8_sig')
fout = open(fn_baseinfo_chem, 'w', encoding='utf-8_sig')
fout.write('文物编号,纹饰,类型,颜色,表面风化,文物采样点,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)\n')

# 读取基本信息
baseinfo = {}
for line in fin_baseinfo:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物编号'):
        continue
    column = line.split(',')
    baseinfo[column[0]] = line
fin_baseinfo.close()

# 读取化学成分比例
for line in fin_chem:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物采样点'):
        continue
    column = line.split(',')
    no = column[0][:2]
    print(f'{baseinfo[no]},{line}')
    fout.write(f'{baseinfo[no]},{line}\n')

fin_chem.close()
fout.close()
```

基本信息赋值和化学成分比例分类 2.py

```
import os

finfochem = '玻璃文物的基本信息赋值和化学成分比例.csv'
ftype2 = '基本信息赋值和化学成分比例分类 2.csv'

if os.path.exists(ftype2):
```

```
    os.remove(ftype2)

fin = open(finfochem, 'r', encoding='utf-8_sig')
fout = open(ftype2, 'w', encoding='utf-8_sig')
fout.write('文物编号,纹饰,类型,颜色,表面风化,文物采样点,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),分类\n')

for line in fin:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物编号'):
        continue
    data = line.split(',')
    mytype = (int(data[2]) - 1) * 2
    # 如果data[5]含有"未风化"字符串
    if data[4] == '0' or '未风化' in data[5]:
        fout.write(line+f',{mytype+2}\n')
    elif data[4] == '1':
        fout.write(line+f',{mytype+1}\n')
    else:
        print('error')
```

基本信息风化预测空间.py

```
fin = open('玻璃文物的基本信息.csv', 'r', encoding='utf-8_sig')

纹饰 = set()
纹饰.add('')
类型 = set()
类型.add('')
颜色 = set()
颜色.add('')
表面风化 = set()
表面风化.add('')

lines = fin.readlines()
datas = []
# 集合去重
for line in lines:
    if line.strip() == '':
```

```
        continue
    if line.startswith('文物编号'):
        continue
    文物编号_, 纹饰_, 类型_, 颜色_, 表面风化_ = line.strip().split(',')
    纹饰.add(纹饰_)
    类型.add(类型_)
    颜色.add(颜色_)
    表面风化.add(表面风化_)
    datas.append([文物编号_, 纹饰_, 类型_, 颜色_, 表面风化_])
fin.close()


hypothesis_space1 = []
hypothesis_space2 = []

for 纹饰_ in 纹饰:
    for 类型_ in 类型:
        for 颜色_ in 颜色:
            hypothesis_space1.append([纹饰_, 类型_, 颜色_])
            hypothesis_space2.append([纹饰_, 类型_, 颜色_])


# 风化版本空间
for it in datas:
    if it[4] == '风化':
        for h in hypothesis_space1:
            if (h[0] != '' and it[1] != h[0]) or (h[1] != '' and it[2] != h
[1]) or (h[2] != '' and it[3] != h[2]):
                hypothesis_space1.remove(h)
    else:
        for h in hypothesis_space1:
            if it[1] == h[0] and it[2] == h[1] and it[3] == h[2]:
                hypothesis_space1.remove(h)

print('风化版本空间为: ')
for h in hypothesis_space1:
    print(h)

# 无风化版本空间
for it in datas:
    if it[4] == '无风化':
        for h in hypothesis_space2:
```

```
            if (h[0] != '' and it[1] != h[0]) or (h[1] != '' and it[2] != h
[1]) or (h[2] != '' and it[3] != h[2]):
                hypothesis_space2.remove(h)
    else:
        for h in hypothesis_space2:
            if it[1] == h[0] and it[2] == h[1] and it[3] == h[2]:
                hypothesis_space2.remove(h)

print('无风化版本空间为: ')
for h in hypothesis_space2:
    print(h)
```

差值范数计算 2.py

```
from pprint import pprint

import numpy as np
import pandas as pd

base_dir = '相关系数矩阵 result/'

data_files = ['高钾不抗风化',
              '高钾抗风化',
              '铅钡不抗风化',
              '铅钡抗风化', ]
suffix = '_corr.csv'

header = ...

fout = '范数计算 2.csv'
foout = open(fout, 'w', encoding='utf-8_sig')


def myfill(d1):
    global header
    header = d1.pop(0)

    for d in d1:
        d.pop(0)
    for i in range(len(d1)):
        for j in range(len(d1[i])):
            if d1[i][j] == '':
                d1[i][j] = 0.0
```

```
def mysub(d1, d2):
    for i in range(len(d1)):
        for j in range(len(d1[i])):
            f1 = 0.0
            try:
                f1 = float(d1[i][j])
            except:
                f1 = 0.0
            f2 = 0.0
            try:
                f2 = float(d2[i][j])
            except:
                f2 = 0.0
            d1[i][j] = f1 - f2


def myinsert(d1):
    global header
    d1.insert(0, header)
    for i in range(len(d1)-1):
        d1[i+1].insert(0, header[i+1])


for i in range(len(data_files)):
    n1 = data_files[i]
    fn1 = base_dir + n1 + suffix
    f1 = open(fn1, 'r', encoding='utf-8')
    d1 = []
    for line in f1:
        d1.append(line.split(','))
    f1.close()
    myfill(d1)
    for j in range(i+1, len(data_files)):
        n2 = data_files[j]
        fn2 = base_dir + n2 + suffix
        f2 = open(fn2, 'r', encoding='utf-8')
        d2 = []
        for line in f2:
            d2.append(line.split(','))
        f2.close()
```

```
        myfill(d2)
        mysub(d1, d2)
        # myinsert(d1)
        # pprint(d1)
        df = pd.DataFrame(d1)
        # 范数
        norm = np.linalg.norm(df)
        foout.write(f'{n1}-{n2},{norm}\n')

foout.close()
```

按分类 2 均值变化量推算风化前后.py

```
import os

ftype2 = '基本信息赋值和化学成分比例分类 2.csv'
fdiff = '分类 2 均值变化量.csv'
typebefore = '按分类 2 均值变化量恢复风化前.csv'
typeafter = '按分类 2 均值变化量演变风化后.csv'

temp = [typebefore, typeafter]
for i in temp:
    if os.path.exists(i):
        os.remove(i)

diff_dict = {}

fin = open(fdiff, 'r', encoding='utf-8_sig')
for line in fin:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('分类'):
        continue
    line = line.split(',')
    diff_dict[line[0]] = line[1:]

fin.close()


fin = open(ftype2, 'r', encoding='utf-8_sig')
fout = open(typebefore, 'w', encoding='utf-8_sig')
```

```
fout.write('文物编号,纹饰,类型,颜色,表面风化,文物采样点,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),分类\n')

for line in fin:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物编号'):
        continue
    data = line.split(',')
    mytype = data[-1]
    if mytype == '2' or mytype == '4':
        fout.write(line + '\n')
        continue
    elif mytype == '1':
        for i in range(len(diff_dict["'1-2"])):
            if data[i+6] == '':
                continue
            data[i+6] = str(float(data[i+6]) - float(diff_dict["'1-2"][i]))
    elif mytype == '3':
        for i in range(len(diff_dict["'3-4"])):
            if data[i+6] == '':
                continue
            data[i+6] = str(float(data[i+6]) - float(diff_dict["'3-4"][i]))
    else:
        print('error')
    fout.write(','.join(data) + '\n')

fin.close()
fout.close()

fin = open(ftype2, 'r', encoding='utf-8_sig')
fout = open(typeafter, 'w', encoding='utf-8_sig')

fout.write('文物编号,纹饰,类型,颜色,表面风化,文物采样点,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),分类\n')

for line in fin:
```

```
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物编号'):
        continue
    data = line.split(',')
    mytype = data[-1]
    if mytype == '1' or mytype == '3':
        fout.write(line + '\n')
        continue
    elif mytype == '2':
        for i in range(len(diff_dict["'1-2"])):
            if data[i+6] == '':
                continue
            data[i+6] = str(float(data[i+6]) + float(diff_dict["'1-2"][i]))
    elif mytype == '4':
        for i in range(len(diff_dict["'3-4"])):
            if data[i+6] == '':
                continue
            data[i+6] = str(float(data[i+6]) + float(diff_dict["'3-4"][i]))
    else:
        print('error')
    fout.write(','.join(data) + '\n')

fin.close()
fout.close()
```

玻璃文物的基本信息赋值.py

```
import os

import tqdm

fn_baseinfocsv = '玻璃文物的基本信息.csv'
fn_baseinfoassigncsv = '玻璃文物的基本信息赋值.csv'
fn_baseinfomapmd = '玻璃文物的基本信息赋值.md'
fn_baseinfomapdocx = '玻璃文物的基本信息赋值.docx'

temp = [fn_baseinfoassigncsv, fn_baseinfomapmd, fn_baseinfomapdocx]
for file in temp:
    if os.path.exists(file):
        os.remove(file)
```

```
fin = open(fn_baseinfocsv, 'r', encoding='utf-8_sig')
fout = open(fn_baseinfoassigncsv, 'w', encoding='utf-8_sig')

纹饰映射 = {'A': 1, 'B': 2, 'C': 3}
类型映射 = {'高钾': 1, '铅钡': 2}
颜色映射 = {'浅绿': 1, '浅蓝': 2, '绿': 3, '蓝绿': 4, '蓝': 5,
        '深绿': 6, '深蓝': 7, '紫': 8, '黑': 9, '': 0}
表面风化映射 = {'无风化': 0, '风化': 1}

for line in tqdm.tqdm(fin):
    文物编号, 纹饰, 类型, 颜色, 表面风化 = line.strip().split(',')
    # print(f'{文物编号},{纹饰},{类型},{颜色},{表面风化}')
    if 文物编号 == '文物编号':
        fout.write(line)
        continue
    纹饰 = 纹饰映射[纹饰]
    类型 = 类型映射[类型]
    颜色 = 颜色映射[颜色]
    表面风化 = 表面风化映射[表面风化]
    fout.write(f'{文物编号},{纹饰},{类型},{颜色},{表面风化}\n')
fin.close()
fout.close()

fmap = open(fn_baseinfomapmd, 'w', encoding='utf-8')
fmap.write('\n### 纹饰赋值\n')
fmap.write('| 纹饰 | 编号 |\n')
fmap.write('| --- | --- |\n')
for k, v in 纹饰映射.items():
    fmap.write(f'| {k} | {v} |\n')

fmap.write('\n### 类型赋值\n')
fmap.write('| 类型 | 编号 |\n')
fmap.write('| --- | --- |\n')
for k, v in 类型映射.items():
    fmap.write(f'| {k} | {v} |\n')

fmap.write('\n### 颜色赋值\n')
fmap.write('| 颜色 | 编号 |\n')
fmap.write('| --- | --- |\n')
for k, v in 颜色映射.items():
    fmap.write(f'| {k} | {v} |\n')
```

```
fmap.write('\n### 表面风化赋值\n')
fmap.write('| 表面风化 | 编号 |\n')
fmap.write('| --- | --- |\n')
for k, v in 表面风化映射.items():
    fmap.write(f'| {k} | {v} |\n')

fmap.close()
os.system(f'pandoc {fn_baseinfomapmd} -o {fn_baseinfomapdocx}')
```

相关矩阵及范数计算.py

```
import os

import numpy as np
import pandas as pd

base_dir = '求相关系数矩阵/'
target_dir = '相关系数矩阵 result/'

data_files = ['高钾不抗风化.xlsx',
              '高钾抗风化.xlsx',
              '铅钡不抗风化.xlsx',
              '铅钡抗风化.xlsx', ]

if not os.path.exists(target_dir):
    os.mkdir(target_dir)


# fnorm = '二范数.csv'
# fout = open(target_dir + fnorm, 'w', encoding='utf-8_sig')

for data_file in data_files:
    df = pd.read_excel(base_dir+data_file)
    # norm = np.linalg.norm(df)
    corr = df.corr()
    corr.to_excel(target_dir + data_file.replace('.xlsx', '_corr.xlsx'))
    # fout.write(data_file[:-5] + ',' + str(norm) + '\n')

# fout.close()
```

聚类分析后的表单均值计算.py

```
from audioop import avg


fin = open('聚类分析后的表单.csv', 'r', encoding='utf-8_sig')
fout = open('聚类分析后的表单均值.csv', 'w', encoding='utf-8_sig')
fout.write('分类,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)\n')

avg_dict = {}
avg_count = {}

for line in fin:
    line = line.strip()
    if line == '':
        continue
    if line.startswith('文物'):
        continue
    column = line.split(',')
    no = column[-1]
    if no not in avg_dict:
        avg_dict[no] = [float(_) for _ in column[1:-1]]
        avg_count[no] = 1
    else:
        # print(len(range(0, len(column)-2)))
        for i in range(0, len(column)-2):
            avg_dict[no][i] = avg_dict[no][i] + float(column[i+1])
        avg_count[no] = avg_count[no] + 1

for key in avg_dict:
    for i in range(0, len(avg_dict[key])):
        avg_dict[key][i] = avg_dict[key][i] / avg_count[key]

for key in avg_dict:
    fout.write(f'{key},{",".join([str(_) for _ in avg_dict[key]])}\n')

fin.close()
fout.close()
```

范数计算废弃.py

```
import os
import pandas as pd
```

```
import numpy as np

base_dir = '相关系数矩阵 result/'

data_files = ['高钾不抗风化',
              '高钾抗风化',
              '铅钡不抗风化',
              '铅钡抗风化', ]
suffix = '_corr.xlsx'

for i in range(len(data_files)):
    n1 = data_files[i]
    f1 = base_dir + n1 + suffix
    d1 = pd.read_excel(f1, skiprows=1)

    df_T = pd.DataFrame(d1.values.T)
    print(d1)
    d1.drop([0, 1, 2, 3], inplace=True)
    d1.fillna(0.0, inplace=True)
    # d1.drop([0], axis=0, inplace=True)
    for j in range(i+1, len(data_files)):
        n2 = data_files[j]
        f2 = base_dir + n2 + suffix
        d2 = pd.read_excel(f2)
        d2.fillna(0.0, inplace=True)
        # d2.drop([0], axis=0, inplace=True)
        # d2.drop([0], axis=1, inplace=True)
        # 范数
        d3 = d1.subtract(d2)
        np.linalg.norm(d3)
        break
    break
```

随机森林-信息赋值化学分类过滤-14-分类.py

```
import os

import joblib
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction import DictVectorizer
from sklearn.model_selection import GridSearchCV, train_test_split
```

```
boliwenwu = pd.read_csv("信息赋值化学分类过滤.csv")

feature = boliwenwu[["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(Ba
O)", "五氧化二磷(P2O5)", "氧化锶(SrO)", "氧化锡(SnO2)", "二氧化硫(SO2)"]]
target = boliwenwu["分类"]

# 填充缺失值
feature = feature.fillna(0.0)

# 抽取字典特征
dv = DictVectorizer()
feature = dv.fit_transform(feature.to_dict(orient="records"))
feature = feature.toarray()

# 划分训练集和测试集
x_train, x_test, y_train, y_test = train_test_split(
    feature, target, test_size=0.25)

# 使用随机森林模型
rfc = RandomForestClassifier()

# 训练
rfc.fit(x_train, y_train)

print("准确率: ", rfc.score(x_test, y_test))

joblib.dump(rfc, 'modlerf3.pkl')
```

随机森林-信息赋值化学分类过滤风化前 0-14-类型.py

```
import os

import joblib
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction import DictVectorizer
from sklearn.model_selection import GridSearchCV, train_test_split

boliwenwu = pd.read_csv("信息赋值化学分类过滤风化前 0.csv")

feature = boliwenwu[["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(Ba
O)", "五氧化二磷(P2O5)", "氧化锶(SrO)", "氧化锡(SnO2)", "二氧化硫(SO2)"]]
target = boliwenwu["类型"]

# 填充缺失值
feature = feature.fillna(0.0)

# 抽取字典特征
dv = DictVectorizer()
feature = dv.fit_transform(feature.to_dict(orient="records"))
feature = feature.toarray()

# 划分训练集和测试集
x_train, x_test, y_train, y_test = train_test_split(
    feature, target, test_size=0.25)

# 使用随机森林模型
rfc = RandomForestClassifier()

# 训练
rfc.fit(x_train, y_train)

print("准确率：", rfc.score(x_test, y_test))

joblib.dump(rfc, 'modlerf4.pkl')
```

随机森林-玻璃文物的基本信息赋值和化学成分比例-14-类型.py

```
import os

import joblib
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction import DictVectorizer
from sklearn.model_selection import GridSearchCV, train_test_split

boliwenwu = pd.read_csv("玻璃文物的基本信息赋值和化学成分比例.csv")

feature = boliwenwu[["表面风化", "二氧化硅(SiO2)", "氧化钠(Na2O)", "氧化钾(K2
O)", "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(Ba
O)", "五氧化二磷(P2O5)", "氧化锶(SrO)", "氧化锡(SnO2)", "二氧化硫(SO2)"]]
```

```
target = boliwenwu["类型"]

# 填充缺失值
feature = feature.fillna(0.0)

# 抽取字典特征
dv = DictVectorizer()
feature = dv.fit_transform(feature.to_dict(orient="records"))
feature = feature.toarray()

# 划分训练集和测试集
x_train, x_test, y_train, y_test = train_test_split(
    feature, target, test_size=0.25)

# 使用随机森林模型
rfc = RandomForestClassifier()

# 训练
rfc.fit(x_train, y_train)

print("准确率：", rfc.score(x_test, y_test))

joblib.dump(rfc, 'modlerf2.pkl')
```

随机森林-玻璃文物的基本信息赋值和化学成分比例-超参数搜索.py

```
import os

import joblib
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction import DictVectorizer
from sklearn.model_selection import GridSearchCV, train_test_split

boliwenwu = pd.read_csv("玻璃文物的基本信息赋值和化学成分比例.csv")

feature = boliwenwu[["表面风化", "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)"]]
target = boliwenwu["类型"]

# 填充缺失值
```

```
feature = feature.fillna(0.0)

# 抽取字典特征
dv = DictVectorizer()
feature = dv.fit_transform(feature.to_dict(orient="records"))
feature = feature.toarray()

# 划分训练集和测试集
x_train, x_test, y_train, y_test = train_test_split(
    feature, target, test_size=0.25)

# 使用随机森林模型
rf = RandomForestClassifier()
# 添加超参数搜索
param = {"n_estimators": [10, 20, 30, 40], "max_depth": [25, 35, 45]}
model = GridSearchCV(rf, param_grid=param, cv=5)

# 训练
model.fit(x_train, y_train)

# 交叉验证网格搜索的结果
print("准确率：", model.best_score_)
print("最好的模型：", model.best_estimator_)

joblib.dump(model, 'modlerf1.pkl')
```

随机森林模型可视化.py

```
import os

import joblib
import tqdm
from sklearn import tree

num = 4

# 读取模型
model = joblib.load(f'modlerf{num}.pkl')

# 目录
directory = f'tree{num}'
```

```
if not os.path.exists(directory):
    os.mkdir(directory)


for idx, estimator in tqdm.tqdm(enumerate(model.estimators_)):
    # 导出dot文件
    tree.export_graphviz(
        estimator, feature_names=["二氧化硅(SiO2)", "氧化钠(Na2O)", "氧化钾(K
2O)", "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)"],
        out_file=f'{directory}/tree{idx}.dot', filled=True, rounded=True, s
pecial_characters=True)
    f = open(f'{directory}/tree{idx}.dot', 'r', encoding='utf-8')
    dot = f.read()
    dot = dot.replace('helvetica', 'DengXian')
    f.close()
    f = open(f'{directory}/tree{idx}.dot', 'w', encoding='utf-8')
    f.write(dot)
    f.close()
    # 转换为png
    os.system(
        f'.\\Graphviz\\bin\\dot.exe -Tpng {directory}/tree{idx}.dot -o {dire
ctory}/tree{idx}.png')
```

随机森林模型预测.py

```
import joblib
import pandas as pd

modelname = 'modlerf4.pkl'
filename = '未分类玻璃文物的化学成分比例赋值.csv'

# 读取模型
model = joblib.load(modelname)
# 预测
boliwenwu = pd.read_csv(filename)
# Nan数据填充0.0
boliwenwu = boliwenwu.fillna(0.0)
# 构造特征值
feature = boliwenwu[["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(Ba
O)", "五氧化二磷(P2O5)", "氧化锶(SrO)", "氧化锡(SnO2)", "二氧化硫(SO2)"]]
y_predict = model.predict(feature)
print(y_predict)
```

11.3 SPSS 代码

```
PRESERVE.
SET DECIMAL DOT.

GET DATA  /TYPE=TXT
  /FILE="C:\Users\27274\Desktop\新建文件夹\玻璃文物的基本信息赋值.csv"
  /DELIMITERS=","
  /QUALIFIER='"'
  /ARRANGEMENT=DELIMITED
  /FIRSTCASE=2
  /DATATYPEMIN PERCENTAGE=95.0
  /VARIABLES=
  文物编号 AUTO
  纹饰 AUTO
  类型 AUTO
  颜色 AUTO
  表面风化 AUTO
  /MAP.
RESTORE.
CACHE.
EXECUTE.

Data written to the working file.
5 variables and 58 cases written.
Variable: 文物编号       Type: Number  Format : F2
Variable: 纹饰             Type: Number  Format : F1
Variable: 类型             Type: Number  Format : F1
Variable: 颜色             Type: Number  Format : F1
Variable: 表面风化       Type: Number  Format : F1

Substitute the following to build syntax for these data.
  /VARIABLES=
   文物编号 F2
   纹饰 F1
   类型 F1
   颜色 F1
```

表面风化 F1

```
DATASET NAME 数据集1 WINDOW=FRONT.
LOGISTIC REGRESSION VARIABLES 表面风化
  /METHOD=ENTER 纹饰 类型 颜色
  /CONTRAST (纹饰)=Indicator
  /CONTRAST (类型)=Indicator
  /PRINT=GOODFIT
  /CRITERIA=PIN(0.05) POUT(0.10) ITERATE(20) CUT(0.5).
```