题目：基于随机森林和均值漂移算法的玻璃成分分类评价模型

摘要

玻璃作为我国古代丝绸之路的重要贸易物品，研究其类型化学成分和类别，可以帮助我们更好的追溯古代手工制造业的发展情况。本文通过研究玻璃表面风化与玻璃的关系，分析玻璃的化学成分，利用统计学习方法，建立了玻璃分类模型，并且对中未知类别玻璃文物进行了预测。该模型可以用于已知玻璃化学成分对其进行分类。

针对问题一，我们首先对数据进行预处理，然后运用**卡方检验**的方法，我们得到表面风化和**纹理**数据的皮尔逊卡方渐进**显著性 P 值为 0.085，不存在显著性原则差异**；表面风化程度与其**玻璃类型**的连续性修正渐进**显著性 P 值为 0.05，存在显著性原则差异**；表面风化程度与其**颜色**的皮尔逊卡方渐进**显著性 P 值为 0.268，不存在显著性原则差异。**其次，为了研究文物样品表面有无风化化学成分含量的统计规律，应使用**数据统计特征提取**方法，我们采用**平均数**来描述数据统计规律。通过数据可视化对比，对于铅钡类文物，其化学元素含量的统计规律为**二氧化硅与氧化铅在风化过后的含量变化最大**。而对于高钾类文物，其化学元素含量的统计规律为**二氧化硅、氧化钾与氧化钙在风化过后的含量变化最大**。最后，我们通过使用**“风化回溯模型”**，来还原并预测各个风化文物未风化时的各个化学成分含量。采取风化过后的每个文物的每个化学元素**减少对应的平均值**的含量，来计算并预测得出这些风化文物在风化前时各个化学成分的含量。

针对问题二，首先运用风化回溯模型将所有未风化的数据还原为风化的数据，然后为了问题三的预测以及选择合适的合适的化学成分进行进行亚类划分，我们选择了随机森林分类器进行监督学习，运用交叉验证和网格搜索，在测试集上得到了 **100%的正确率**，并且选择出了最重要的四种化学元素：**氧化铅，氧化钾，氧化钡和氧化钙**，对应的平均重要性为 **[41%,19%,19% ,8%]**，并进行了成分选择灵敏度检验。之后使用了 MeanShift 算法分别对铅钡玻璃和高钾玻璃进行亚分类。对于铅钡玻璃，我们分为了**三类**，分别为**高铅类**（氧化铅含量占 30%-70%), **高钡类**（氧化钡含量占 25%-36%）以及**均匀类**（氧化铅含量占 10%-30%，氧化钡占 2%-24%）。对于高钾玻璃，我们分为了**两类**，分别**钾硅类**（氧化钙小于 2%），**钾钙类**（氧化钙占 2%-9%），并找到了相应的论文进行合理性解释。

针对问题三，我们使用了改进的平均值回溯模型，并通过玻璃分类模型将所有未知文物分为了铅钡玻璃和高钾玻璃，其中文物 A2 有可能为铅钡玻璃也有可能为高钾玻璃，所以将 A2 的两种可能都考虑了进去。接着将回溯后数据带入对应的亚分类模型，对于铅钡玻璃，**A3 为高铅类，A4，A8，A2，A5 都为均匀类。**对于高钾玻璃，**A1，A6，A7，A2 均为钾钙类。**并通过加入 α 扰动项进行敏感性分析，得到结论为我们的亚分类模型具有**很好的稳定性。**

针对问题四，我们通过**斯皮尔曼系数**对铅钡类和高钾类玻璃的化学成分两两之间进行了相关性分析，并选取了相关系数较高的两组数据进行分析：在铅钡类得到**相关系数为-0.757** 的氧化铅与二氧化硅和**相关系数为 0.674** 的氧化铝和氧化镁，在高钾类玻璃得到**相关系数为 0.752** 的氧化钙与氧化钾和**相关系数为-0.747** 的二氧化硅和氧化钙。同时用 Friedman 检验了不同类别之间化学成分的差异性。

关键词：卡方检验　玻璃分类　风化回溯模型　随机森林　Mean-Shift 算法 斯皮尔曼相关性分析

一、 问题重述

1.1 问题的背景

考古学家说，99%的人类历史都是由文物和化石来展现的。通过分析文物可以研究其所处的时代，它为什么出现在那里的原因，以及所具有的文化功能和社会价值。而玻璃作为丝绸之路的早期贸易往来的宝贵物证，研究其特征和化学成分，可以推动我国古代历史、文化以及生产力的研究。

但我国古代玻璃因其化学成分的不同，特别容易受环境的影响。在整个的风化过程中，玻璃内部的化学成分会和大气发生作用从而导致其成分产生变化，从而给考古工作增加了难度。因此，如何对风化过后的玻璃进行处理和分类，以及对其化学成分进行分析，便具有了重要意义。

现有一批我国古代玻璃制品的相关数据，考古工作者依据这些文物样品的化学成分和其他检测手段已将其分为高钾玻璃和铅钡玻璃两种类型。

1.2 问题提出

现有一批我国古代玻璃制品的相关数据，考古学家根据其化学成分和其他检测方法，将其分为高钾和铅钡两类。给出附件表单 1（文物的分类信息），附件表单 2（相应的主要成分所占比例）以及附件表单 3（未知类别玻璃文物的化学成分）。

根据题目背景和上述附件，我们需要建立数学模型解决如下问题：

(1) 问题一：分析这些玻璃文物的表面风化与玻璃类型、装饰、颜色之间的关系；再从玻璃类型和玻璃有无风化分析其化学成分含量的统计规律；从风化点的数据去推测风化前的化学成分含量。

(2) 问题二：根据附件表单 1 和 2，推出高钾和铅钡玻璃的分类规则；选择每一类别玻璃合适的化学成分，对其进行细分，给出具体的分类方法和分类结果，评价分类的合理性和敏感性。

(3) 问题三：根据附件表单 3，根据未知玻璃的化学成分，去推算它所属的类型，并对结果进行敏感性分析。

(4) 问题四：对不同类别的玻璃之间的化学成分进行分析，找出其关联关系，比较不同类别玻璃之间化学成分关联关系的差异性。

二、 问题分析

2.1 问题一分析

针对问题一，我们可以大致将其分为以下三个小问：

问题(1):分析玻璃文物纹饰、类型、颜色与表面风化的关系，因为这四个指标都是定类变量,并非连续变量，我们采取卡方检验的方法，分别以玻璃文物纹饰、类型、颜色为变量，风化程度为因变量，分析卡方值及渐进显著性 P 值来判断两者之间是否存在关系。

问题(2) :基于这些玻璃文物的类型,分析文物样品表面有无风化化学成分含量的统计规律;对于类型只有两种,一种是高钾,一种是铅钡。在做完缺失值处理之后,我们可以对比文物样

本表面有无风化的化学成分的一些统计学规律，利用数据统计特征提取方法,不过在做分析之前,我们可以做一些数据总合或者分类汇总柱状图等,观察推断出来两种玻璃文物类型的风化前后化学成分变化差异或者相关情况,进而推断出来统计规律。

问题(3) :根据未风化点检测数据,预测其风化前的化学成分含量。根据问题 2 得到的风化前后各个化学成分的平均值及相关差值,预测已经风化的文物在风化前的化学成分含量。

2.2 问题二分析

问题(1):可以对风化数据处理，采用随机森林算法训练高钾玻璃、铅钡玻璃的分类模型。

问题(2):然后对分类过后的模型，得到化学成分的权重，然后以权重大的化学成分作为亚分类依据，利用 mean-shift 算法对其进行聚类，然后再分析亚类的合理性。

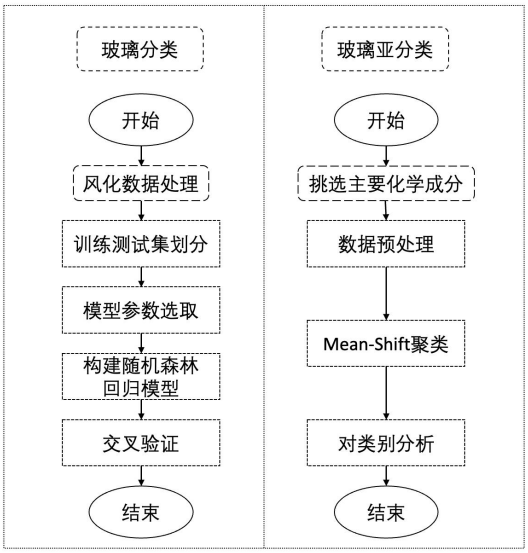


图 1 问题二思路结构图

2.3 问题三分析

问题(1) :首先将所有数据回溯为风化前的数据，对此需要去对文物的类别进行分类，然后将回溯之后的数据代入对应类型的亚分类模型中进行具体的分类。

问题(2) :为了分析分类结果的敏感性，我们需要对数据加上一个微小的扰动，使原始数据发生变化，观察分类结果的变化规律，得出模型对数据的扰动是否敏感。

2.4 问题四分析

可以对铅钡类和高钾类玻璃的化学成分两两之间进行了相关性分析，再用 Friedman 检验不同类别之间化学成分的差异性。

三、 模型假设

1. 假设附件表单给出的颜色都是物体原本的颜色，即最开始未风化时玻璃所具备的颜色。
2. 假设空白数据为不能识别的数据和未检测到的数据，例如：颜色为空白的数据，我们可以

认为玻璃风化较严重，不能判断其颜色。

3. 假设玻璃化学成分比例累加和介于 85%~105%之间的数据为有效数据。
4. 假设玻璃风化过后，只是改变了化学成分的含量，而不会增加或减少新的化学成分种类。

四、 符号说明

符号说明	说明
A	实际观测值矩阵
E	期望值矩阵
x^2	卡方值
m	矩阵行数
n	矩阵列数
D	卡方自由度
g	预测值
k	决策树个数
ρ	残差的加权关联系数
S_k	数据集的点到 x 的距离小于球半径 h 的数据点
AV_change	风化前后各化学元素改变的平均值
AV_Max/Min	风化前后各化学元素改变的边界值

五、 模型的建立与求解

5.1 问题一的模型建立与求解

由于附件表单 2 中给出的是玻璃化学成分不一定是有效的，为了满足模型假设，我们首先对数据进行了筛选，剔除掉编号为 15 号和 17 号的玻璃数据。由于题目已经说明某些数据存在合理缺失项，因此我们便不再对缺失值进行检测和处理。对于颜色为空白的数据，认为其是风化较严重而不能判断颜色。

5.1.1 数据关联性分析

根据表单 1 中的数据可以得到:纹饰、类型、颜色、表面风化四个变量均为定类变量。因此，我们对定类变量之间的相关性和差异性进行分析采用卡方检验进行检验。

我们应用该检验的基本思想是：首先假设表面风化与其纹饰、玻璃类型和颜色不存在显著性差异（H0 成立）。根据附件表单 1 的数据，构建实际观测值矩阵 A 和期望值矩阵 E，通过卡

方值计算公式求得观察值与理论值之间的偏离程度。根据显著性水平及自由度，查验卡方检验临界值表，得到临界值并与卡方值进行比较。

其皮尔逊 x^2 值的计算公式为：

$$E=\frac{\text{行合计}\times\text{列合计}}{\text{总计}}$$

$$x^2=\sum\frac{(A-E)^2}{E}=\sum_{i=1}^{m}\frac{(A_{ij}-E_{ij})^2}{E_{ij}},j=\overline{1,n}$$

卡方自由度计算公式为：

$$D=(n-1)\times(m-1)$$

上述两公式中 A 为实际观测值矩阵、E 为期望值矩阵、x^2 为卡方值、m 表示矩阵行数、n 表示矩阵列数、D 为卡方自由度

在 H_0 假设成立的情况下获得表面风化与其玻璃类型、纹饰和颜色存在关系的显著性概率 P 值。如果 P 值小于 0.05，说明观察值与理论值偏离程度太大，应当拒绝 H_0 假设，表示表面风化与其玻璃类型、纹饰和颜色存在显著性差异；否则就接受 H0 假设，说明表面风化与其玻璃类型、纹饰和颜色不存在显著性差异。

情况 1. 研究表面风化与其纹饰的关系

H_0 假设：表面风化与其纹饰不存在显著性差异

实际观测值矩阵 *A:*

按纹饰交叉表		纹饰			总计
		A	B	C	
表面风化	风化	11	6	17	34
	无风化	11	0	11	22
总计		22	6	28	56

期望值矩阵 *E*:

按纹饰交叉表		纹饰			总计
		A	B	C	
表面风化	风化	13.36	3.64	17	34
	无风化	8.64	2.36	11	22
总计		22	6	28	56

所有理论数>5，自由度为 2，总样本量大于 40，我们将使用皮尔逊卡方进行检验：

卡方检验值与 P 值:

卡方检验			
	x^2 值	自由度	渐进显著性 P 值
皮尔逊卡方	4.941	2	0.085

结果分析：

通过计算卡方值进行卡方检验分析显示：基于表面风化程度与纹饰，皮尔逊卡方渐进显著性 P 值为 0.085；大于 0.05，水平上不呈现显著性，接受原假设的原则，因此我们可得对于表面风化和纹理数据不存在显著性原则差异，也就是说表面是否风化与其纹饰不存在明显关系。

情况 2. 研究表面风化与其玻璃类型的关系

H_0 假设：表面风化与其玻璃类型不存在显著性差

实际观测值矩阵 *A:*

按类型交叉表		类型		总计
		高钾	铅钡	
表面风化	风化	6	28	34
	无风化	10	12	22
总计		16	40	56

期望值矩阵 *E:*

按类型交叉表		类型		总计
		高钾	铅钡	
表面风化	风化	9.71	24.29	34
	无风化	6.29	15.71	22
总计		16	40	56

所有理论数为 4；在一到五之间，自由度为 1，总样本量大于 40，我们将使用连续性修正的卡方值进行检验：

$$x^2 = \sum \frac{(|A-E|-0.5)^2}{E} = \sum_{i=1}^{m} \frac{(|A_{ij}-E_{ij}|-0.5)^2}{E_{ij}}, j = \overline{1,n}$$

上述两公式中 *A* 为实际观测值矩阵、*E* 为期望值矩阵、x^2 为卡方值、*m* 表示矩阵行数、*n* 表示矩阵列数、*D* 为卡方自由度

连续性修正卡方检验值与 P 值:

卡方检验			
	x^2 值	自由度	渐进显著性 P 值
连续性修正	5.06	1	0.05
似然比	5.00	1	0.025

结果分析：

通过计算卡方值进行卡方检验分析显示：基于表面风化程度与其玻璃类型，连续性修正渐进显著性 P 值为 0.05；小于等于 0.05，水平上呈现显著性，拒绝原假设的原则，因此我们可得对于表面风化和玻璃类型数据存在显著性原则差异，也就是说表面是否风化与其玻璃类型存在明显关系。

情况 3. 表面风化与其颜色的关系

H_0 假设：表面风化与其颜色不存在显著性差

实际观测值矩阵 *A:*

按颜色交叉表		颜色									总计
		无法检测颜色	黑	蓝绿	绿	浅蓝	浅绿	深蓝	深绿	紫	
表面风化	风化	4	2	9	0	12	1	0	4	2	34
	无风化	0	0	6	1	6	2	2	3	2	22
总计		4	2	15	1	18	3	2	7	4	56

期望值矩阵 *E:*

按颜色交叉表		颜色									总计
		无法检测颜色	黑	蓝绿	绿	浅蓝	浅绿	深蓝	深绿	紫	
表面风化	风化	2.43	1.21	9.11	0.61	10.93	1.82	1.21	4.25	2.43	34
	无风化	1.57	0.79	5.89	0.39	7.07	1.18	0.79	2.75	1.57	22
总计		4	2	15	1	18	3	2	7	4	56

所有理论数>5，自由度为 8，总样本量大于 40，我们将使用皮尔逊卡方进行检验：
卡方检验值与 P 值:

卡方检验			
	x^2 值	自由	渐进显著性 P 值

		度	
皮尔逊卡方	9.96	8	0.268
似然比	13.01	8	0.111

结果分析：

通过计算卡方值进行卡方检验分析显示：基于表面风化程度与其颜色，皮尔逊卡方渐进显著性 P 值为 0.268；大于 0.05，水平上不呈现显著性，接受原假设的原则，因此我们可得对于表面风化和颜色数据不存在显著性原则差异，也就是说表面是否风化与其颜色不存在明显关系。

综上，我们得出结论玻璃表面风化程度与其颜色和纹饰并没有显著关系，而与其类型有显著关系。

5.1.2 化学成分统计规律

根据问题 1 的第一小问已知，文物的类型与风化程度有密切关联，接下来我们会结合玻璃的类型与其风化程度进行分类统计其化学成分。我们将该问题以玻璃类型分类与文物样品表面有无风化分为了四个部分：第一组为风化的铅钡文物样品、第二组为风化的高钾类文物样品、第三组为无风化的铅钡文物样品、第四组为无风化的高钾类文物样品：

组别	类型	风化程度	纹饰	颜色	总含量范围
第一组	高钾	风化	B	蓝绿	99.57% - 100%
第二组		无风化	A、C	蓝绿、浅蓝、深绿、深蓝	96.06% - 100%
第三组	铅钡	风化	A、C	浅蓝、紫、蓝绿、深绿、浅绿、黑、无法检验颜色	90.17% - 99.89%
第四组		无风化	A、C	浅蓝、紫、深蓝、深绿、浅绿、绿	88.41% - 99.98%

为了研究文物样品表面有无风化化学成分含量的统计规律，应使用数据统计特征提取方法，其中包括众数、中位数、平均数等。结合附件中给出的数据，我们需要分析对于风化与无风化的不同文物的各种化学成含量的数据规律，考虑到样本的数量以及数据的特征，最后我们决定采用可以显示某种化学成分各个风化或不风化文物一组数据特征的平均数来描述数据统计规律。

通过给定数据计算出四组各个文物化学成分的平均值如下表：

风化程度	类型			
	铅钡	平均值	高钾	平均值
风化	二氧化硅(SiO2)	33.61	二氧化硅(SiO2)	93.96
	氧化钠(Na2O)	0.95	氧化钠(Na2O)	0.00
	氧化钾(K2O)	0.14	氧化钾(K2O)	0.54
	氧化钙(CaO)	2.35	氧化钙(CaO)	0.87

	氧化镁(MgO)	0.70	氧化镁(MgO)	0.20
	氧化铝(Al2O3)	3.84	氧化铝(Al2O3)	1.93
	氧化铁(Fe2O3)	0.56	氧化铁(Fe2O3)	0.27
	氧化铜(CuO)	2.00	氧化铜(CuO)	1.56
	氧化铅(PbO)	36.87	氧化铅(PbO)	0.00
	氧化钡(BaO)	10.49	氧化钡(BaO)	0.00
	五氧化二磷(P2O5)	4.16	五氧化二磷(P2O5)	0.28
	氧化锶(SrO)	0.37	氧化锶(SrO)	0.00
	氧化锡(SnO2)	0.06	氧化锡(SnO2)	0.00
	二氧化硫(SO2)	0.99	二氧化硫(SO2)	0.00
未风化	二氧化硅(SiO2)	53.44	二氧化硅(SiO2)	67.98
	氧化钠(Na2O)	0.77	氧化钠(Na2O)	0.70
	氧化钾(K2O)	0.26	氧化钾(K2O)	9.33
	氧化钙(CaO)	1.23	氧化钙(CaO)	5.33
	氧化镁(MgO)	0.49	氧化镁(MgO)	1.08
	氧化铝(Al2O3)	3.19	氧化铝(Al2O3)	6.62
	氧化铁(Fe2O3)	0.93	氧化铁(Fe2O3)	1.93
	氧化铜(CuO)	1.56	氧化铜(CuO)	2.45
	氧化铅(PbO)	23.59	氧化铅(PbO)	0.41
	氧化钡(BaO)	10.50	氧化钡(BaO)	0.60
	五氧化二磷(P2O5)	0.90	五氧化二磷(P2O5)	1.40
	氧化锶(SrO)	0.30	氧化锶(SrO)	0.04
	氧化锡(SnO2)	0.06	氧化锡(SnO2)	0.20
	二氧化硫(SO2)	0.28	二氧化硫(SO2)	0.10

分析上表我们可以通过风化与未风化的差值来得出，风化前后哪些物化学物质的变化程度最大，也就是说哪种化学元素对于风化的影响最大。下图中，我们讲前后各个化学元素值放入图像中去直观的分析：

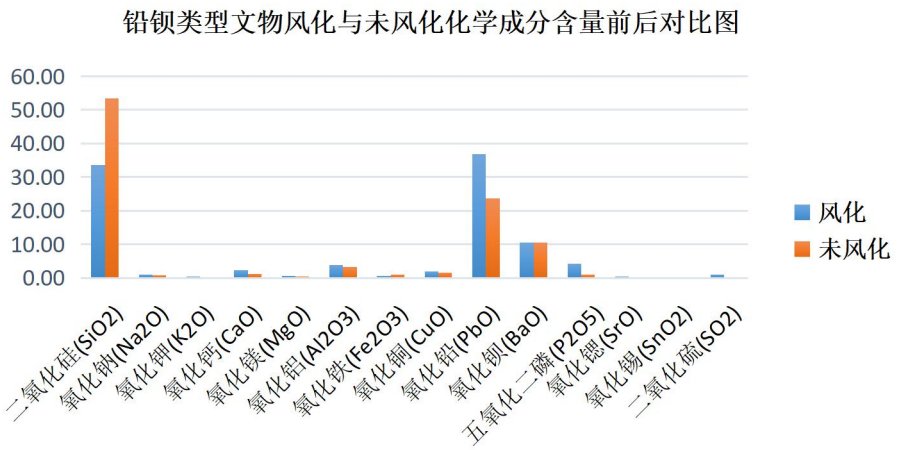


图 2

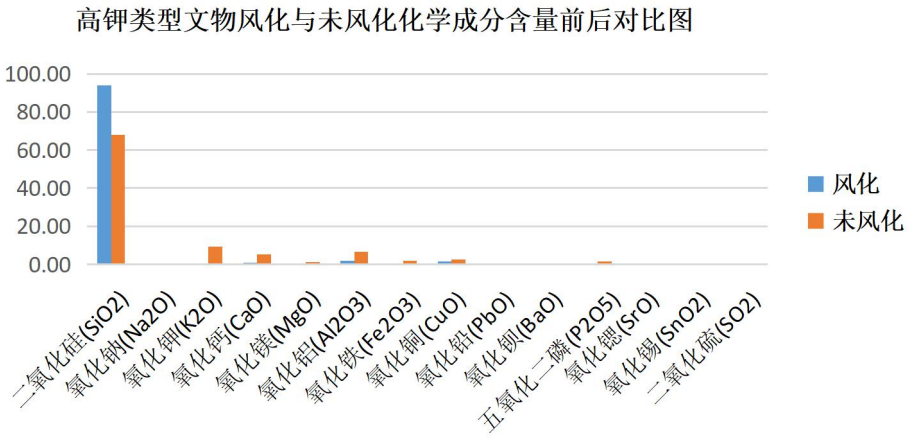


图 3

观察上图比较我们可以明显看出：铅钡类文物风化与未风化化学元素含量差异最明显的是二氧化硅与氧化铅，通过图像可知风化后会使铅钡类文物的二氧化硅的含量大幅度减少、氧化铅的含量大幅度增加。值得注意的是，与文献[9]的结论“氧化钡比氧化铅的风化前后差值更大”相比，我们得到的铅钡类文物中氧化钡的风化前后变化差值并不是很大，我们分析这个差异的原因可能是论文研究的各个文物的出土年份并不相同。而高钾类文物风化与未风化化学元素含量差异最明显的则是二氧化硅与氧化钙，风化会导致二氧化硅的含量大幅度增加、氧化钾与氧化钙的含量大幅度减少。

总的来说，对于铅钡类文物，其化学元素含量的统计规律为二氧化硅与氧化铅在风化过后的含量变化最大。而对于高钾类文物，其化学元素含量的统计规律为二氧化硅、氧化钾与氧化钙在风化过后的含量变化最大。

5.1.3 风化回溯模型

在上一小问中，我们通过计算每种化学成分含量的平均值来分析不同文物类型的风化前后化学成分的变化影响关系。我们思考，也可以通过将风化过后的每个文物的每个化学元素减少对应的平均值的含量，来计算得出这些风化文物在风化前时各个化学成分的含量。

铅钡与高钾风化与未风化差值，如下表：

铅钡	风化与未风化差值	高钾	风化与未风化差值
二氧化硅(SiO2)	-19.83	二氧化硅(SiO2)	25.98
氧化钠(Na2O)	0.18	氧化钠(Na2O)	-0.70
氧化钾(K2O)	-0.12	氧化钾(K2O)	-8.79
氧化钙(CaO)	1.11	氧化钙(CaO)	-4.46
氧化镁(MgO)	0.21	氧化镁(MgO)	-0.88
氧化铝(Al2O3)	0.64	氧化铝(Al2O3)	-4.69

氧化铁(Fe203)	-0.38	氧化铁(Fe203)	-1.67
氧化铜(Cu0)	0.44	氧化铜(Cu0)	-0.89
氧化铅(Pb0)	13.28	氧化铅(Pb0)	-0.41
氧化钡(Ba0)	-0.01	氧化钡(Ba0)	-0.60
五氧化二磷(P205)	3.25	五氧化二磷(P205)	-1.12
氧化锶(Sr0)	0.07	氧化锶(Sr0)	-0.04
氧化锡(Sn02)	-0.01	氧化锡(Sn02)	-0.20
二氧化硫(S02)	0.71	二氧化硫(S02)	-0.10

为预测风化前各个种类文物的化学含量，我们采用回溯算法，通过上图中铅钡类与高钾类的文物风化前后的差值，将风化过后的每个文物的每个化学元素减少对应的平均值的含量，由此我们便可得出其风化前的预测结果。随后，我们为避免结果出现过小或过大的值，将各个类型化学成分未风化数据的最大值与最小值当做此预测值的前后边界值，见下表：

未风化文物化学成分含量		二氧化硅 SiO_2	氧化钠 Na_2O	氧化钾 K_2O	氧化钙 CaO	氧化镁 MgO	氧化铝 Al_2O_3	氧化铁 Fe_2O_3	氧化铜 CuO	氧化铅 PbO	氧化钡 BaO	五氧化二磷 P_2O_5	氧化锶 SrO	氧化锡 SnO_2	二氧化硫 SO_2
高钾类	最大值	87.05	3.38	14.52	8.7	1.98	11.15	6.04	5.09	1.62	2.86	4.5	0.12	2.36	0.47
	最小值	59.01	0	5.19	0	0	3.05	0	0	0	0	0	0	0	0
铅钡类	最大值	75.51	4.66	1.41	4.49	1.67	5.45	4.59	8.46	39.22	26.23	5.75	0.91	0.44	3.66
	最小值	31.94	0	0	0	0	1.44	0	0	9.3	3.42	0	0	0	0

表中我们可知不同类别的化学成分含量的最大值与最小值，只要超出或低于此区间的值，我们就会将超过的值算作区间内的最大值，低于的值算作区间内的最小值。

随后我们可以将风化值反推回未分化时的预测值，详情表格数据可见附录“论文支撑材料数据--还原为未风化总表”查看。

5.2 问题二的模型建立与求解：

为了还原玻璃化学成分的本身和增大分类模型的数据量，我们利用问题一中第三小问中的风化回溯模型，将所有已风化的数据还原为为风化的数据，然后就可以用原始的化学成分来推断高钾玻璃、铅钡玻璃的分类规律，从而得到更好的分类与聚类模型。

5.2.1 玻璃分类树模型

将所有数据还原为未风化之后，我们拥有所有铅钡玻璃和高钾玻璃所对应的化学成分，所以，可以将玻璃分类看作是有监督的机器学习。构建玻璃分类树模型，通过玻璃的化学成分，推出铅钡玻璃和高钾玻璃的分类规律。

为了避免单个决策树训练的模型出现过拟合的情况，我们选择采用随机森林算法[5]。随机森林回归算法的基本原理为：首先，在原始数据集中有放回地随机抽取数据组成训练样本集；然后，根据抽取的训练样本集分别构建决策树，得到各决策树的回归结果。最后，对各决策树的回归结果计算均值得到最终结果。原理示意图如图 3 所示：

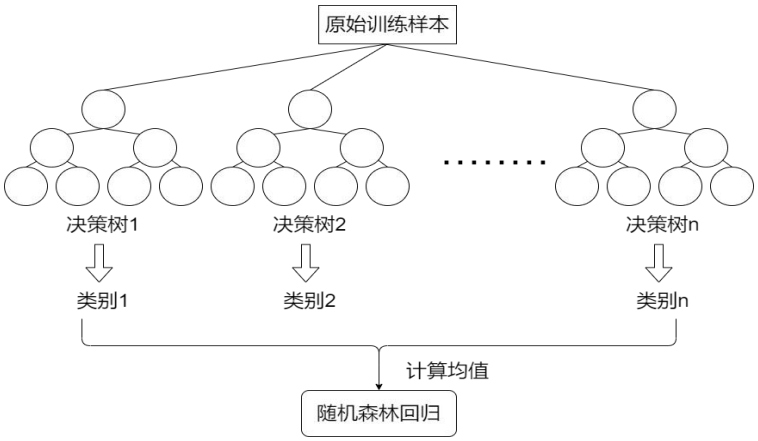


图 4 随机森林算法流程图

随机森林回归算法的数学推导为:对原始数据集中自变量(输入数据)X 和因变量(需预测输出数据) Y,假设(X,Y)的分布独立,随机在(X.Y)中抽取训练样本集 K,预测结果设为 *g(X)*,则其均方泛化误差为:

$$E_{X,Y}\left[Y-\mathrm{g}(X)\right]^2$$

假定有 k 颗决策树,对 k 颗决策树计算其预测值{g(K, X_k)}的均值得到随机森林回归的预测结果。当$k \to \infty$时,有下式:

$$E_{X,Y}\left[Y-g_k(X,K_k)\right]^2 \to E_{X,Y}\left[Y-E_k(X,K_k)\right]^2$$

上式中后式表示泛化误差,记为 PE^{**},当 k 趋近于无穷大时,每颗决策树的泛化误差记为 PE^{*},PE^{**}满足:

$$PE^{*}=E_K Ex,y\left[Y-g\,(X,K)\right]^2$$

其中 K 满足:

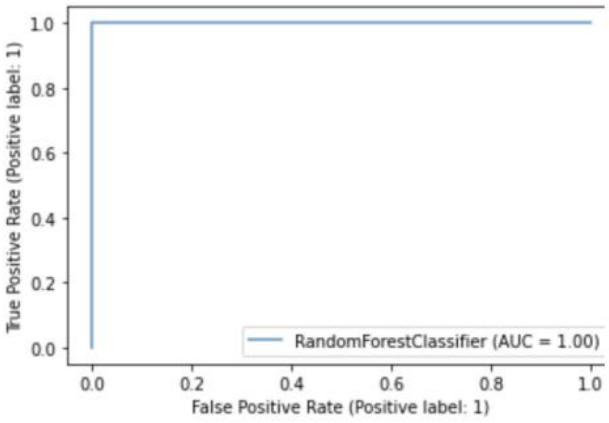
$$PE^{**}<\rho PE^{*}$$

其中ρ为残差的加权关联系数，最终随机森林的回归函数为:

$$Y = E_k g\left(X, K\right)$$

具体实验步骤为：

（1） 数据预处理：通过风化回溯模型将风化数据转化为未风化数据。
（2） 特征提取以及训练集划分：将 14 个化学成分当做特征值 X，玻璃种类为目标变量 Y，由于数据为百分数，不再进行归一化。将风化还原过后的数据集按 7：3 的比例拆训练集和测试集。
（3） 分类模型建立：将每个类别的化学成分作为输入值，使用 Python 中 sklearn 库 RandomForestRegressor 函数，得到高钾玻璃、铅钡玻璃的分类模型。
（4） 模型参数选择：应用网格搜索和交叉验证，首先将 n_estimators 设置为范围为 10~50，max_features 设置范围为{ ‘sqrt’,‘log2’}，通过嵌套循环搜索每一种参数组合。
（5） 模型分析：分析结果，当 n_estimators 为 10、max_features 为‘sqrt’时，得到最好的交叉验证结果，并且在测试集中预测正确率为 100%。ROC 曲线和混交矩阵如下图所示：



	precision	recall	f1-score	support
0	1.00	1.00	1.00	6
1	1.00	1.00	1.00	15
accuracy			1.00	21
macro avg	1.00	1.00	1.00	21
weighted avg	1.00	1.00	1.00	21

图 5 ROC 曲线以及混淆矩阵

因此，我们通过随机森林模型构建玻璃分类树模型，并且得到了表现很好的分类器，接下来我们将使用该模型中的结果筛选出不同种类中的主要化学成分。

5.2.2 化学成分权重分析

首先对未风化的铅钡玻璃和高钾玻璃的平均值进行观察，如下图所示：

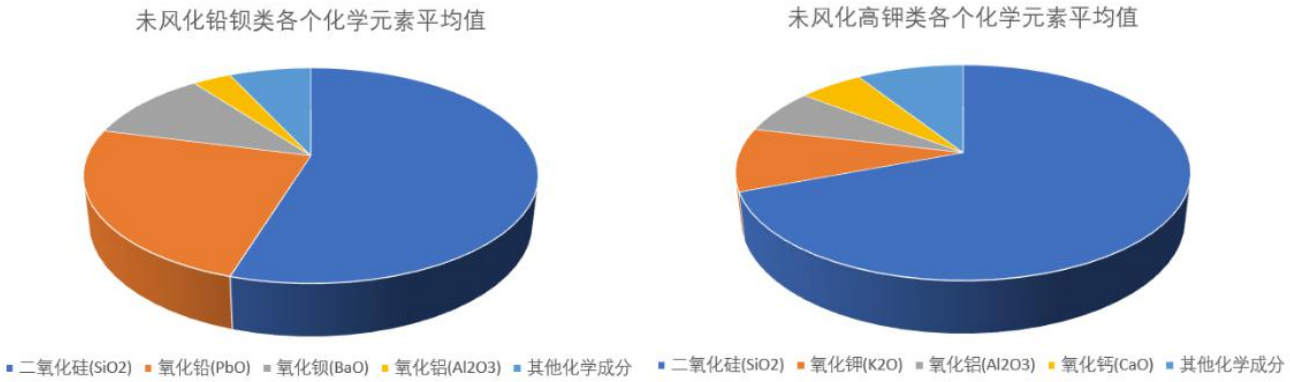


图 6

可以观察到在铅钡玻璃中，二氧化硅，氧化铅，氧化钡，氧化铝展主要部分。在高钾玻璃中，二氧化硅，氧化钾，氧化铝，氧化钙占主要部分。

接着我们通过改变玻璃分类树模型中 max_features 的值，通过调用模型中 feature_importances_参数得到各个化学成分在模型中的重要性，排序后的结果如下表:

Max_features	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**平均值**
氧化铅(PbO)	0.42	0.62	0.60	0.46	0.24	0.36	0.16	0.41	0.41
氧化钾(K2O)	0.30	0.27	0.10	0.29	0.16	0.29	0.14	0.01	0.19
氧化钡(BaO)	0.09	0.10	0.22	0.09	0.31	0.16	0.32	0.19	0.19
氧化钙(CaO)	0.07	0.00	0.00	0.07	0.17	0.08	0.13	0.12	0.08
氧化铝(Al2O3)	0.06	0.00	0.05	0.00	0.03	0.06	0.06	0.12	0.05
二氧化硅(SiO2)	0.00	0.01	0.00	0.01	0.06	0.01	0.03	0.08	0.02
氧化铁(Fe2O3)	0.02	0.00	0.00	0.00	0.03	0.00	0.10	0.02	0.02
氧化锶(SrO)	0.03	0.00	0.02	0.05	0.01	0.01	0.01	0.04	0.02

这里所呈现的重要性与之前分析的结果吻合，结果如下图所示：

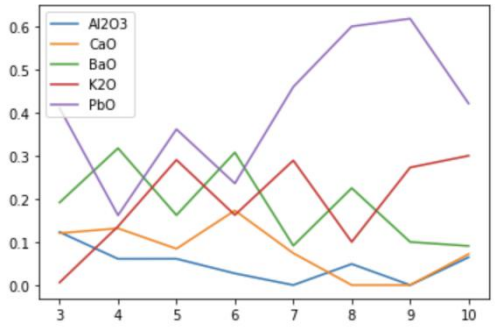


图 7 成分选择灵敏度分析

由图可以得知除了 PbO 以外，其他四个化学元素波动都不会超过 0.1（也就是对应 10%的占比），并且根据不同的参数变化，各个化学元素的变化虽然是无规律的，但是他们的重要性始终保持在前四名。除此之外，可以明显的观察到 Al2O3 的指标低于其他四种化学元素。在文献[6]中指出：Al2O3 是高钾玻璃中一种常见的杂质，并且可能来自多种原材料之中。在文献[10]中也提到：Al2O3 的含量与 K2O 的含量没有线性关系，这表明尽管一些富含钾的铝硅酸盐矿物可能被用作玻璃制造的原料，但它们不是 K2O 的主要来源。所以我们有理由认为 Al2O3 并不是进行分类的主要化学元素。

通过观察铅钡玻璃和高钾玻璃化学成分的关系，发现氧化铅和氧化钡主要存在于铅钡玻璃中，而氧化钾和氧化钙主要存在于高钾玻璃中。所以，我们选择使用氧化铅和氧化钡对铅钡玻璃进行亚类划分，使用氧化钾和氧化钙对高钾玻璃进行亚类划分。并且通过分析得到这四种化学物质在模型参数变化时，体现出了较好的稳定性。因此，我们认为以这四种化学元素所构建的玻璃亚分类模型会体现出较好的稳定性和分类结果。

5.2.3 玻璃亚分类模型

在选择出主要化学元素之后，需要选择出合适的聚类算法。因为在这个问题中我们提前未知需要分成几类，并且样本数量较少，所以我们选择使用 Mean-Shift 均值漂移算法[4]进行聚类分析。

Mean-Shift 算法介绍：

给定 d 维空间的 n 个数据点集 X，那么对于看空间中的任意点 x 的 Mean-Shift 向量基本形式可以表示为：

$$M_h = \frac{1}{K}\sum_{x_i \in S_k}(x_i - x)$$

这个向量就是漂移向量，其中 S_k 表示的是数据集的点到 x 的距离小于球半径 h 的数据点。也就是：

$$S_h(x) = \{y : (y - x_i)_T (y - x_i) < h^2\}$$

而漂移的过程，就是通过计算得漂移向量，然后把球圆心 x 的位置更新一下，更新公式为：

$$x = x + M_h$$

使得圆心的位置一直处于力的平衡位置。

具体的 mean shift 聚类过程如下：

1、在未标注的数据点中随机选择一个点作为中心。

2、 找出到中心距离在 bandwidth 内的所有点，记为集合 M，认为这些点属于簇 c。同时，将这些插入属于该类的概率加 1，该参数将用于最后一步的分类。

3、 计算从中心到集合 M 的每个元素的向量，并将这些向量相加得到位移向量 shift。

4、 中心 = 中心 + Shift。即中心向偏移方向移动，移动距离为||shift||。

5、 重复步骤 2、3、4，直到 shift 的大小变小（即收敛），记住此时的中心。请注意，在迭代过程中找到的所有点都必须分配给簇 c。

6、如果当前簇 c 的中心与其他现有集群 c2 的中心之间的距离小于阈值，则我们将 c2 和 c 合并。否则，把 c 作为新的分类，并添加 1 个类。

7、 重复 1、2、3、4、5，直到所有点都标记为可访问。

8、 分类：根据每个类的访问频率，将访问频率最高的类作为当前点集所属的类。

对于铅钡玻璃，得到结果如下图：

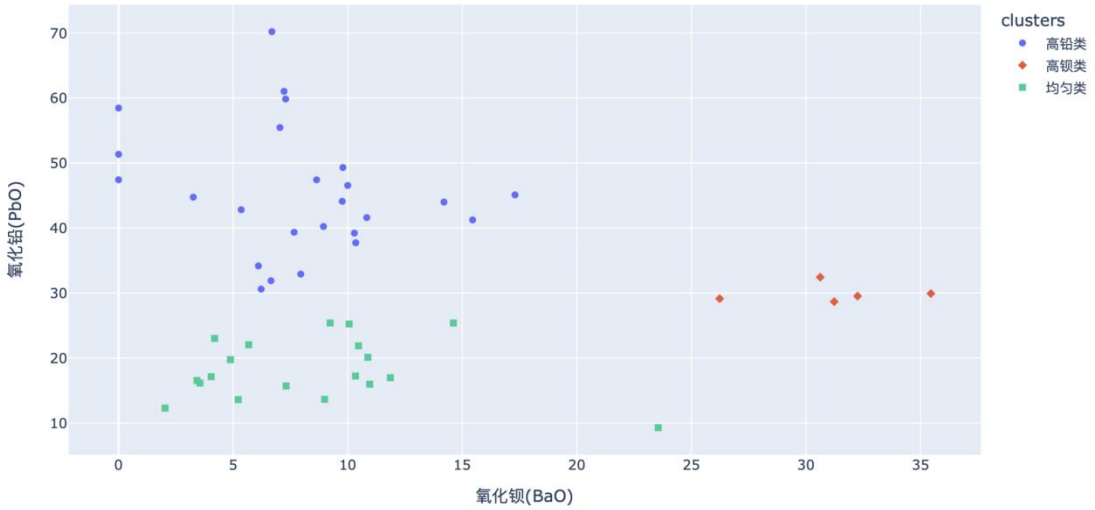


图 8 铅钡玻璃化学成分聚类图

质心为：{[36.9404 ， 8.9192],
[21.265 ， 7.48363636],
[29.185 ， 28.39833333]}

通过模型计算，得到铅钡玻璃可以被分为 3 种亚类。我们根据氧化铅以及氧化钡的含量，我们将其命名为**高铅类**（氧化铅含量占 30%-70%），**高钡类**（氧化钡含量占 25%-36%）以及**均匀类**（氧化铅含量占 10%-30%，氧化钡占 2%-24%）我们所得到的分类结果与文章[9] 中得到的结果相似，因此认为我们的亚类划分是合理的。同时，该文章指出，高铅类与均匀类的玻璃制品主要来自战国时期，而高钡类玻璃数量在汉代之后逐渐增多。可以推测汉代以后的铅钡玻璃配方比战国时期的更成熟。

对于高钾玻璃，得到结果如下图：

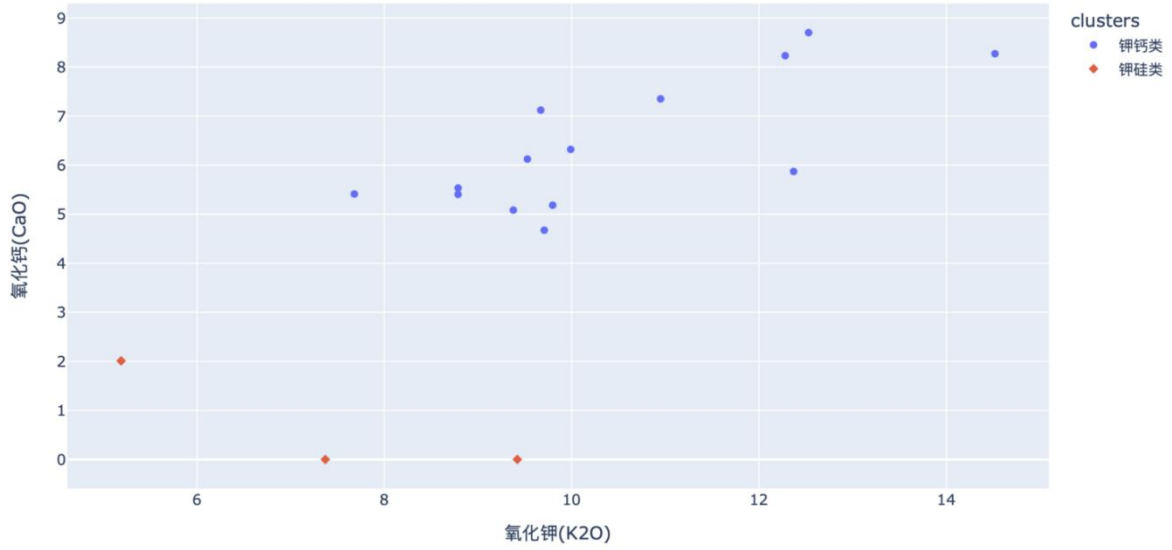


图 9 高钾玻璃化学成分聚类图

质心为 {[9.695, 5.82409091],
[8.395, 0.0]}

观察图片可知，通过 Mean-Shift 算法大致以氧化钙的浓度为标准分成了两个类。其中一个类别中氧化钙的浓度非常小（小于 2%），另一个氧化钙的浓度在 2%和 9%之间。然后我们将二氧化硅-氧化钙-氧化钾含量三维图展现出来，如图 10 所示。

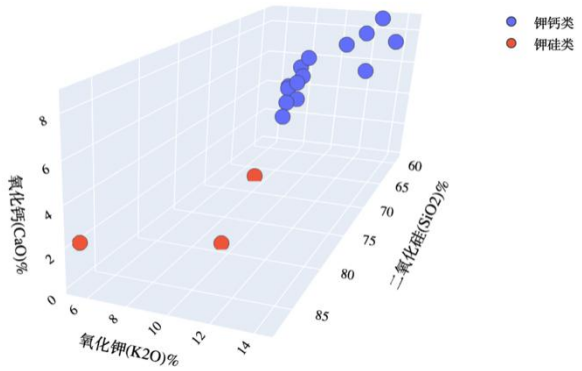


图 10 二氧化硅-氧化钙-氧化钾含量三维图

可以观察到二氧化硅的占比在各个点上都有所不同，并且都大于 60%。据此，我们将含氧化钙浓度小（小于 2%）的亚类称为**钾硅类**，将氧化钙浓度高（2%-9%）的亚类称为**钾钙类**。

在文献[11] 同样指出了在汉代，以钾和硅为主的玻璃在中国南方已经普遍被使用。在文章[12]中提到最早的以氧化硅，氧化钾和氧化钙为主的玻璃制品被发现于战国勾践的剑的配饰上，同时文章[13]也指出以这三种物质为主的玻璃制品在湖北省被大量发现。因此，可以认为我们的模型在高钾玻璃的亚分类上**也有较好的合理性。**

5.3 问题三的模型建立与求解

首先需要通过风化回溯模型将已风化的数据还原为未风化的数据，然而由于他们的种类是未知的，所以构建了改进的平均值预测算法对已风化的数据进行判断其是属于铅钡还是高钾玻璃。然后将数据代入对应的亚分类模型中，并且通过加入 α 扰动项，对数据加入微小扰动，观察其亚分类结果是否有明显波动。

5.3.1 玻璃类型预测模型

表三中的数据分为未风化和已风化，需要还原为未风化，但是未知其是什么种类。通过对数据的观察，由第一问的平均值变化规律可知，如果是铅钡玻璃，风化之后他的氧化铅含量评卷会增加总含量的 13,28%，我们已知，A6 与 A7 风化之后没有检测到氧化铅，所以我们初步判断，A6 和 A7 文物为高钾类玻璃文物。同时，我们根据风化之后铅钡玻的二氧化硅含量会减少 19.83%，满足此条件的只有 A2 和 A5 文物，所以我们初步认定，A2 与 A5 为铅钡类玻璃文物。为了验证我们的猜想，我们设置了改进的平均值预测算法。

步骤 1.

所以我们将问题一风化回溯模型中的变化量取了平均值：

$$\text{改变的平均值}\quad AV\_change=\frac{Pb\_change+K\_change}{2}$$

$$\text{边界的平均值}\quad AV\_Max/Min=\frac{Pb\_Max/Min+K\_Max/Min}{2}$$

步骤 2.

确定一个干扰项$\boldsymbol{\alpha}$，此干扰项$\boldsymbol{\alpha}\in(-1,1)$,步长为 0.1 共取 20 个值，代入回溯检验模型，公式如下：

$$\text{风化数据}+(1+\alpha)*AV_{\text{change}}=\text{预测的未风化数据}$$

步骤 3.

代入问题二中所得到的分类回归树模型中进行分类，并统计为铅钡类玻璃文物和高钾类玻璃文物的次数。可能性结果如下表：

文物编号 / 分类结果	A2	A5	A6	A7
铅钡玻璃	80%	45%	0%	0%
高钾玻璃	20%	55%	100%	100%

结合初步分析及实验结果，文物编号为 A6 和 A7 的一定为高钾玻璃文物，A2 一定为铅钡类玻璃文物。我们又由上述表观察发现，A5 号文物有可能是高钾玻璃文物也铅钡类玻璃文物，所以在接下来的亚分类中，我们会将 A5 号文物既作为高钾玻璃也作为铅钡玻璃进行下一步分类。

除此之外，由于未风化的文物不需要进行回溯操作，我们只需要将未风化的数据代入回归分类树模型进行分类和检验即可，得到的结果如下：A1 为高钾类玻璃文物,A3、A3、A8 为铅钡类玻璃文物。

步骤 4.

将表三中的所有风化文物化学成分数通过回溯还原为未风化时各个文物的化学成分含量值，并进行统一分类之后，再将其代入玻璃亚分类模型进行对应的亚类化分。

得到的结果如下表：

铅钡玻璃：

化学成分 / 文物编号	氧化铅(PbO)	氧化钡(BaO)	具体亚类划分
A3	39.58	4.69	高铅类
A4	24.28	8.31	均匀类
A8	21.24	11.34	均匀类
A2	21.02	3.42	均匀类
A5	9.3	3.42	均匀类

高钾玻璃：

化学成分 / 文物编号	氧化钾(K2O)	氧化钙(CaO)	具体亚类划分
A1	0	6.08	钾钙类
A6	10.14	5.1	钾钙类
A7	9.77	5.58	钾钙类
A2	9.16	6.1	钾钙类

观察上表，文物分类基本与其化学元素含量相匹配，具有合理性。

5.3.2 敏感性分析

对我们在问题二中所选择出来的主要化学元素所对应的特征向量进行数据上的微调，通过观察分类结果的变化，进行分析分类结果的稳定性。我们将确定一个干扰项$\boldsymbol{\alpha}$，此干扰项$\boldsymbol{\alpha}\in$(-0.5,1=0.5),步长为 0.1，共取 10 个值，代入通过回溯检验模型改进的微调模型得到增加了干扰项之后的数据，公式如下：

$$\text{数据} + \alpha \times \text{对应平均值} = \text{扰动后数据}$$

将得到的新数据带入对应的亚分类模型中，得到对应的分类结果。

对于铅钡玻璃，结果如图：

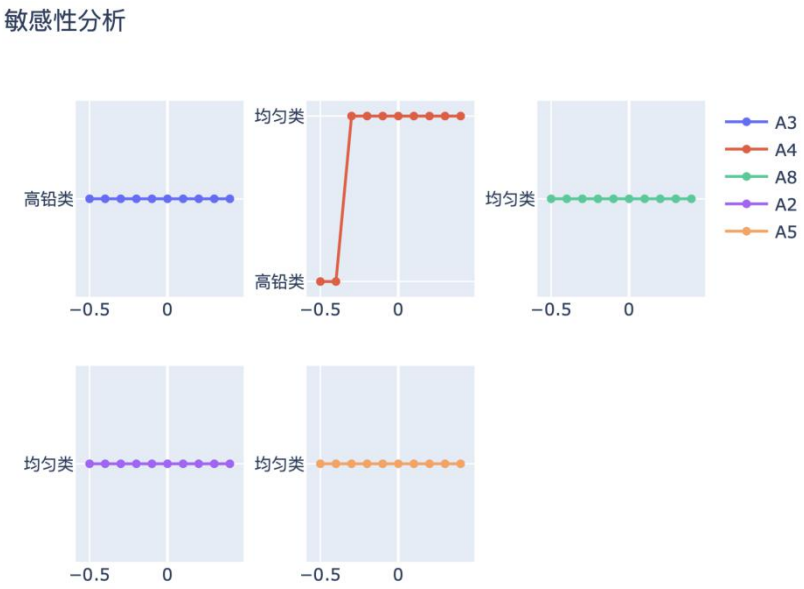


图 11 铅钡玻璃亚分类敏感性分析

可以观察到，除了 A4 以外，其他文物的数值在经过微小扰动后，所分到的类别仍然不变，并且 A4 只有在 α 取-0.5 和-0.4 时被分到了与原始结果不同的类别。说明我们的铅钡玻璃亚分类模型对数据微扰动并不敏感，具有较好的稳定性。

对于高钾玻璃，结果如图：

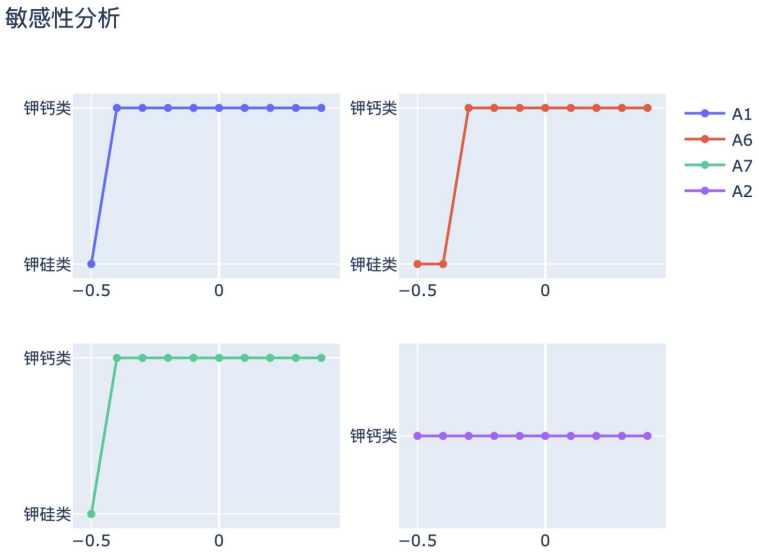


图 12 高钾玻璃亚分类敏感性分析

得到的结果与铅钡玻璃亚分类模型类似，只有在巨大扰动时才会出现分类结果的变化。因此可以得出结论：问题三所得出的分类模型有较好的稳定性，对数据的微小扰动并不敏感，具有较好的泛化能力。

5.4 问题四的模型建立与求解

为了求出不同类别之间化学成分的关联关系，我们分别对铅钡玻璃、高钾玻璃的化学成分进行两两相关性分析。

5.4.1 化学成分关联性分析

考虑到数据并非严格正态分布，所以我们利用斯皮尔曼相关系数去检验化学成分之间的相关性。在铅钡玻璃中，将样本数量为 49 的化学成分转为为等级数据，再用斯皮尔曼相关系数计算公式

$$r = \frac{\sum_i (x_i - x_{mean})(y_i - y_{mean})}{\sqrt{\sum_i (x_i - x_{mean})^2 \sum_i (y_i - y_{mean})^2}}$$

得到相关系数热力表：

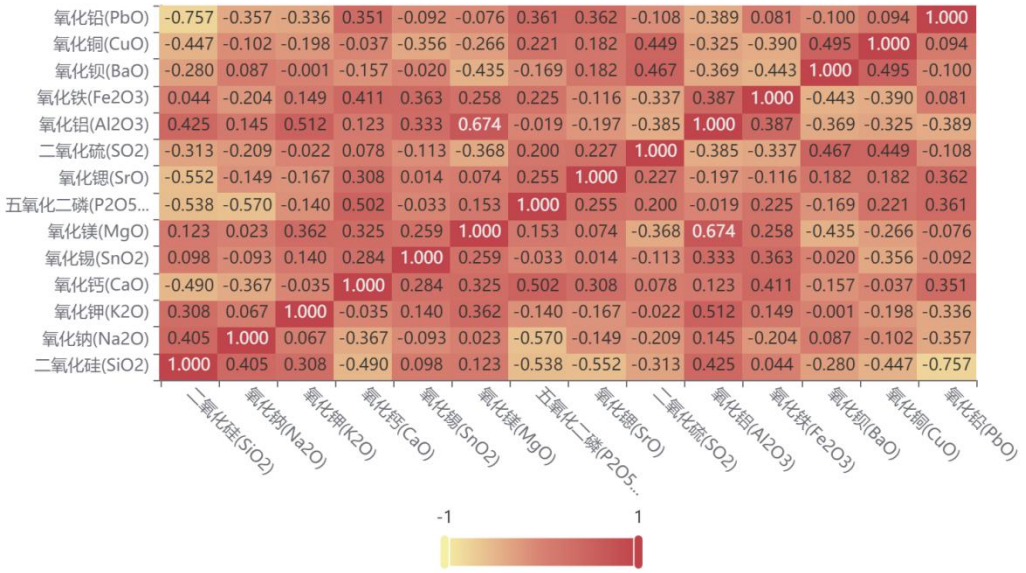


图 13 相关系数热力表

选取相关系数为-0.757 的氧化铅与二氧化硅和相关系数为 0.674 的氧化铝和氧化镁，做出其散点图如下：

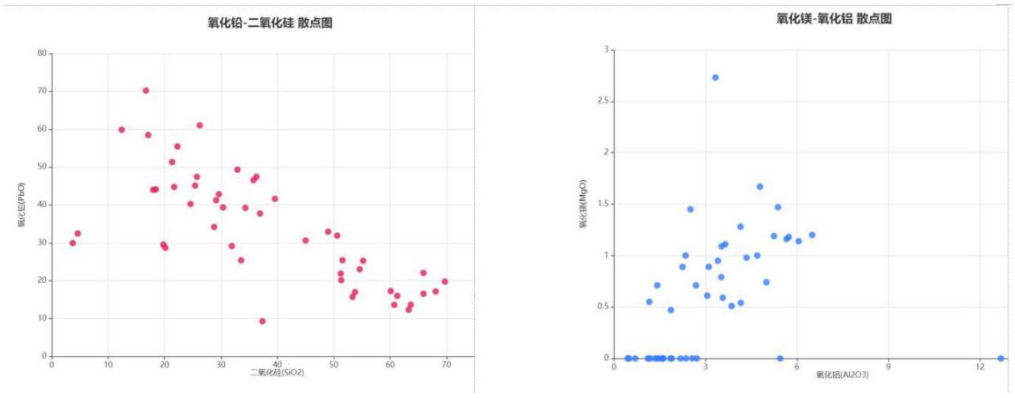


图 14

分析可知，氧化铅与二氧化硅有很强的亲和力，在较低的温度下与二氧化硅化合，生成流动性很好的硅酸铅。所以随着二氧化硅的增加，氧化铅会与之发生反应，从而含量随之减少。为提高玻璃密度，常常会添加适量氧化镁，所以氧化铝的增加也会增加氧化镁，所以成正相关。

同理可得到高钾玻璃中化学成分两两之间的关系如下：

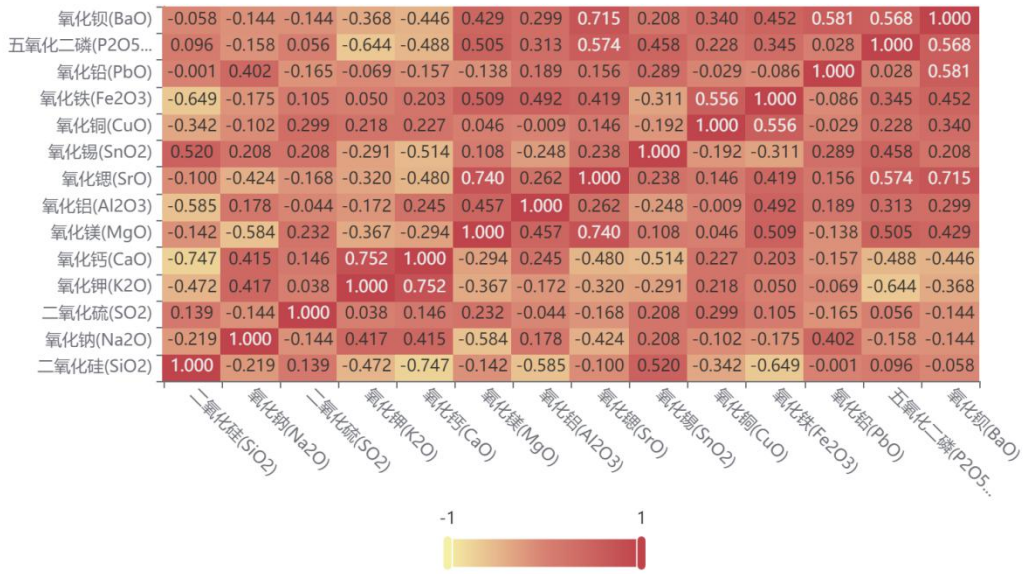


图 15 化学成分两两之间关系热力图

选取相关系数为 0.752 的氧化钙与氧化钾和相关系数为-0.747 的二氧化硅和氧化钙，做出其散点图如下：

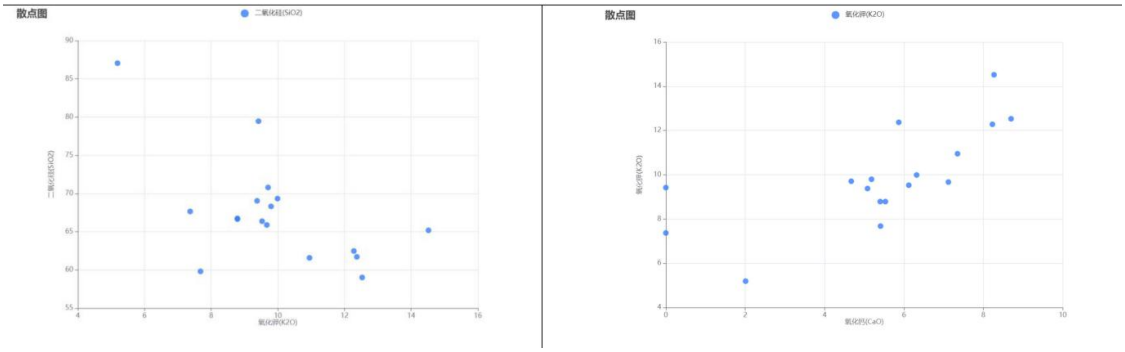


图 16 氧化钙与氧化钾和相关系数散点图

分析可知：在玻璃生产时，一般为适量加入氧化钾，能降低玻璃液的高温黏度，促进玻璃液的熔化和澄清。与普通玻璃相比，含钾玻璃的软化温度更高，韧性更好，强度更高，因此又被称之为硬玻璃。

在玻璃原料中往往也会加入氧化钠和氧化钾，这两种化学元素加入主要是由于它们的熔点低，能在较低的温度下与二氧化硅，反应形成硅酸盐，从而降低玻璃的粘度，加快玻璃的溶制速度，是良好的助溶剂。所以氧化钙和氧化钾成正相关

5.4.2 差异性分析

为了分析两种类型玻璃的化学成分之间的差异性，同时又考虑到变量并不是严格正态分布（正态分布检验见附录），我们选择弗里德曼检验分析多组样本数一致的变量之间有无明显差异。通过分析每组配对样本的 P 值，来判断化学成分之间差异性是否呈现出显著性。

Friedman 统计量定义为：

$$Q=\frac{12}{bk(k+1)}\sum_{i=1}^{k}\left(R_i-\frac{b(k+1)}{2}\right)^2=\frac{12}{bk(k+1)}\sum_{i=1}^{k}R_i^2-3b(k+1)$$

对于有限的 k 和 b，有零假设下的分布表可查，查表是要做变换：

$$W=\frac{Q}{b(k-1)}$$

当查不到时，对于固定的 k，当 b 趋近于无穷时，在零假设下有 Friedman 统计量 $Q\rightarrow x^2(k-1)$

对于铅钡玻璃我们可以得到：

弗里德曼差异性检验P值	二氧化硫(SO2)	氧化锡(SnO2)	氧化钠(Na20)	氧化铝(Al203)	二氧化硅(Si02)	氧化锶(Sr0)	氧化钙(Ca0)	氧化镁(Mg0)	氧化钾(K20)	氧化铅(Pb0)	氧化铁(Fe203)	五氧化二磷(P205)	氧化钡(Ba0)	氧化铜(Cu0)
二氧化硫(S02)														
氧化锡(SnO2)	0.9													
氧化钠(Na20)	0.9	0.667												
氧化铝(Al203)	0.001	0.001	0.001											
二氧化硅(Si02)	0.001	0.001	0.001	0.005										
氧化锶(Sr0)	0.559	0.23	0.9	0.001	0.001									
氧化钙(Ca0)	0.001	0.001	0.001	0.9	0.001	0.012								
氧化镁(Mg0)	0.249	0.067	0.9	0.001	0.001	0.9	0.058							
氧化钾(K20)	0.9	0.9	0.9	0.001	0.001	0.9	0.001	0.844						
氧化铅(Pb0)	0.001	0.001	0.001	0.003	0.9	0.001	0.001	0.001	0.001					
氧化铁(Fe203)	0.52	0.2	0.9	0.001	0.001	0.9	0.014	0.9	0.9	0.001				
五氧化二磷(P205)	0.001	0.001	0.007	0.69	0.001	0.07	0.9	0.236	0.001	0.001	0.083			
氧化钡(Ba0)	0.001	0.001	0.001	0.9	0.52	0.001	0.083	0.001	0.001	0.421	0.001	0.014		
氧化铜(Cu0)	0.001	0.001	0.056	0.291	0.001	0.313	0.9	0.628	0.005	0.001	0.352	0.9	0.001	

图 17 铅钡玻璃差异性 P 值

对于高钾玻璃我们可以得到：

弗里德曼差异性检验P值	二氧化硅(Si02)	氧化铝(Al203)	氧化钠(Na20)	氧化钾(K20)	五氧化二磷(P205)	氧化钙(Ca0)	氧化锡(Sn02)	氧化锶(Sr0)	氧化铅(Pb0)	氧化铜(Cu0)	氧化镁(Mg0)	氧化铁(Fe203)	氧化钡(Ba0)
二氧化硅(Si02)													
氧化铝(Al203)	0.9												
氧化钠(Na20)	0.001	0.001											
氧化钾(K20)	0.9	0.9	0.001										
五氧化二磷(P205)	0.001	0.059	0.9	0.003									
氧化钙(Ca0)	0.248	0.9	0.014	0.786	0.591								
氧化锡(Sn02)	0.001	0.001	0.9	0.001	0.193	0.001							
氧化锶(Sr0)	0.001	0.001	0.9	0.001	0.204	0.001	0.9						
氧化铅(Pb0)	0.001	0.001	0.9	0.001	0.8	0.004	0.9	0.9					
氧化铜(Cu0)	0.015	0.675	0.237	0.183	0.9	0.9	0.004	0.004	0.105				
氧化镁(Mg0)	0.001	0.032	0.9	0.002	0.9	0.463	0.298	0.311	0.9	0.9			
氧化铁(Fe203)	0.002	0.339	0.563	0.045	0.9	0.9	0.025	0.027	0.339	0.9	0.9		
氧化钡(Ba0)	0.001	0.001	0.9	0.001	0.73	0.002	0.9	0.9	0.9	0.077	0.856	1.258	

图 18 高钾玻璃差异性 P 值

六、 模型的评价、改进与推广

6.1 模型的优点：

选择卡方检验可以很好的得到相关性
随机森林模型可以自动处理特征，选择出最好的特征，并且可以得到各个特征的重要行。
玻璃分类数模型表现出了很好的精准度，并且具有很好的泛化能力。
亚分类模型表现出了很好的稳定性，对扰动不敏感。

6.2 模型的缺点：

可以构建更复杂的风化回溯模型，得到更精准的风化前数据。

6.3 模型的推广：

最好加入时间数据，可以加入时间轴数据，构建出更为合理的，与时间相关的分析与鉴定函数，从而更准确的进行分类和回溯。

七、 参考文献

［1］ 吴宗道 周福征 史美光 几个古玻璃的显微形貌、成分及其风化的初步研究 1986

［2］ 干福熹 中国古代玻璃的起源——中国最早的古代玻璃研究 2007

［3］ 王承遇 硅酸盐玻璃的风化 2003

［4］ “Mean shift: A robust approach toward feature space analysis” D. Comaniciu and P. Meer, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002)

［5］ Breiman, “Random Forests”, Machine Learning, 45(1), 5-32, 2001.

［6］ Song Liu a , Qinghui Li a & Fuxi Gan a, Chemical Analyses of Potash–Lime Silicate Glass Artifacts from the Warring States Period in China 2015

［7］ J. Q. Dong, Q. H. Li* and S. Liu The native development of ancient Chinese glassmaking: a case study on some early lead– barium–silicate glasses using a portable 2015

［8］ James W. Lankton and Laure Dussubieux，Early Glass in Asian Maritime Trade: A Review and an Interpretation of Compositional Analyses, 2006

［9］ The native development of ancient Chinese glassmaking: a case study on some early lead – barium – silicate glasses using a portable XRF spectrometer

［10］Xia, X. W.; Liu, S.; Wang, Q.; Liu, Y. X.; Li, Q. H.; Gu, D. H. The non-destructive analysis of glass artifacts unearthed from burials of Yue State at Hongshan. Relics from South 2013, 3, 143–149.

［11］Liu, S.; Li, Q. H.; Fu, Q.; Gan, F. X.; Xiong, Z. M. Application of a portable XRF spectrometer for classification of potash glass beads unearthed from tombs of Han Dynasty in Guangxi, China. X-Ray Spectrometry 2013, 42(6), 470–479.

［12］Hou, D. J. Preliminary studies of glass production in Chu State. Cultural Relics of

Central China 1989, 4, 24–29.

[13] 2. Gan, F. X.; Cheng, H.; Li, Q. Origin of Chinese ancient glasses— study on the earliest Chinese ancient glasses. Science in China Series E: Technological Sciences 2006, 49(6), 701–713.

八、 附录

附录 1：
介绍：支撑材料的文件列表
Data.xlsx 还原为未风化总表.xlsx 回溯算法从风化还原为未风化所得的预测值.xlsx 多配对样本 Friedman 检验_钾钙类.doc 多配对样本 Friedman 检验_铅钡类.doc Code.ipynb

附录 2：

介绍：风化回溯模型代码，在 jupyter-notebook 平台编写

```
import pandas as pd
import numpy as np

def filter(array,Max,Min,dvalue):
A = np.zeros_like(array,dtype='float64')
for i in range(array.shape[0]):
A[i] = array[i] - dvalue
if A[i] > Max:
A[i] = Max
elif A[i] < Min:
A[i] = Min

return A

df = pd.read_excel("C/风化变未风化.xlsx",sheet_name='铅钡类还原成为风化')
df = df.fillna(0)
df.head()
Indi = pd.read_excel("C/风化变未风化.xlsx",sheet_name='IN',index_col=0)
#K
df_k = pd.read_excel("C/风化变未风化.xlsx",sheet_name='高钾类还原成为风化')
df_k = df_k.fillna(0.0)
df_k
for name in df_k.columns:
df_k[name] = filter(df_k[name].values,Indi[name]['Kmax'],Indi[name]['Kmin'],Indi[name]['K_change'])#Pb
for name in df.columns:
df[name] = filter(df[name].values,Indi[name]['Pbmax'],Indi[name]['Pbmin'],Indi[name]['Pb_change'])
df.to_excel('BeforeWeathering_Pb.xlsx')
```

```
df_k.to_excel('BeforeWeargerubf_K.xlsx')
```

附录 3：

介绍：问题二玻璃分类树模型，在 jupyter-notebook 平台编写

```
import pandas as pd
import numpy as np
#从 sklearn.model_selection 导入 train_test_split 用于数据分割
from sklearn.model_selection import train_test_split
#从 sklearn.preprocessing 中导入 StandardScaler 用于数据标准化
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
import matplotlib.pyplot as plt

data = pd.read_excel("data.xlsx",sheet_name="Sheet1")
data.head()

data = data.fillna(0)
data.head()
y = data['类型'].values
print(y)
x = np.array([])
data_mc = data.drop(['类型'],axis=1)
for i in range(data_mc.shape[0]):
    x=np.append(x,data_mc.iloc[i].values)
x.resize(data_mc.shape)
# 划分训练集测试集
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_sta
te=40)
rf = RandomForestClassifier()
pro_dic = {"n_estimators":np.arange(10,50,10),'max_features':['sqrt', 'log2',
None]}
estimator = GridSearchCV(rf,param_grid=pro_dic,cv=5)
estimator.fit(x_train,y_train)
print("在交叉验证中验证的最好结果: \n", estimator.best_score_)
print("最好的参数模型: \n", estimator.best_estimator_)
print("每次交叉验证后的准确率结果: \n", estimator.cv_results_)
from sklearn.metrics import plot_roc_curve
ax = plt.gca()
rfc_disp = plot_roc_curve(estimator.best_estimator_, x_test, y_test, ax=ax, a
```

```
lpha=0.8)
from sklearn.metrics import roc_curve, auc ###计算 roc 和 auc
from sklearn.metrics import classification_report
y_predict = estimator.best_estimator_.predict(x_test)
ret = classification_report(y_test,y_predict)
print(ret)
#敏感性分析
data_mc.loc['weight'] = rf.feature_importances_
sensitive = pd.DataFrame(data_mc.loc['weight'])
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
for i in np.arange(3,11):
    rf = RandomForestClassifier(n_estimators=10,max_features=i)
    rf.fit(x_train,y_train)
    sensitive.insert(0,i,rf.feature_importances_)
sensitive = sensitive.drop(['weight'],axis=1)
# sensitive.to_excel('sensitive2.xlsx')
order = pd.read_excel("sensitive2.xlsx",sheet_name='Sheet2',index_col=0)
order.plot()
```

附录 4：

介绍：问题二均值漂移算法，在 jupyter-notebook 平台编写

```
import numpy as np
from sklearn.cluster import MeanShift, estimate_bandwidth
from sklearn.datasets import make_blobs
data_Pb_Ba = pd.read_excel("data.xlsx",sheet_name="Pb_Ba")
data_Pb_Ba = data_Pb_Ba.fillna(0)
data_High_K = pd.read_excel("data.xlsx",sheet_name="High_k")
data_High_K = data_High_K.fillna(0)
x = np.zeros((data_Pb_Ba.shape[0],2))
x[:,0] = data_Pb_Ba['氧化铅(PbO)'].values
x[:,1] = data_Pb_Ba['氧化钡(BaO)'].values
bandwidth = estimate_bandwidth(x, quantile=0.4, n_samples=100)
ms = MeanShift(bandwidth=bandwidth, bin_seeding=True)
ms.fit(x)
labels = ms.labels_
cluster_centers = ms.cluster_centers_

labels_unique = np.unique(labels)
n_clusters_ = len(labels_unique)

print("number of estimated clusters : %d" % n_clusters_)
ms.n_features_in_
```

```
DF = data_Pb_Ba[['氧化铅(PbO)','氧化钡(BaO)']]
DF['clusters'] = labels
DF["clusters"] = DF["clusters"].astype(str)
DF =  DF.replace('0','高铅类')
DF = DF.replace('1','均匀类')
DF = DF.replace('2','高钡类')
import plotly.express as px
# DF["clusters"] = DF["clusters"].astype(str)
fig = px.scatter(DF, x="氧化钡(BaO)", y="氧化铅(PbO)",color='clusters',symbol='clusters')
fig.show()
#高钾玻璃
data_High_K.head()
data_High_K = data_High_K.drop(11)
x_k  = np.zeros((data_High_K.shape[0],2))
x_k[:,0] = data_High_K['氧化钾(K2O)']
x_k[:,1] = data_High_K['氧化钙(CaO)']
#模型训练
bandwidth_k = estimate_bandwidth(x_k, quantile=0.45, n_samples=100)
ms_k = MeanShift(bandwidth=bandwidth_k, bin_seeding=True)
ms_k.fit(x_k)
labels_k = ms_k.labels_
cluster_centers_k = ms_k.cluster_centers_

labels_unique_k = np.unique(labels_k)
n_clusters_k = len(labels_unique_k)

print("number of estimated clusters : %d" % n_clusters_k)
ms_k.n_features_in_
DF = data_High_K[['氧化钾(K2O)','氧化钙(CaO)']]
DF.insert(0,'clusters',labels_k)
DF["clusters"] = DF["clusters"].astype(str)
DF =  DF.replace('0','钾钙类')
DF = DF.replace('1','钾硅类')
import plotly.express as px
# DF["clusters"] = DF["clusters"].astype(str)
fig = px.scatter(DF, x='氧化钾(K2O)', y='氧化钙(CaO)',color='clusters',symbol='clusters')
# fig.update_layout(
#     paper_bgcolor='rgba(1,0,0,0)',
#     plot_bgcolor='rgba(0,0,0,0)',
#     legend = dict(bgcolor = 'yellow')
```

```
# )
# fig.update_yaxes(showgrid=True, gridwidth=1, gridcolor='#C0C0C0')
# fig.update_xaxes(showgrid=True)
fig.show()
DF['二氧化硅(SiO2)'] = data_High_K['二氧化硅(SiO2)']
ms_k.cluster_centers_
import plotly.graph_objects as go

PLOT = go.Figure()


for C in list(DF.clusters.unique()):

    PLOT.add_trace(go.Scatter3d(x = DF[DF.clusters == C]['二氧化硅(SiO2)'],
                                y = DF[DF.clusters == C]['氧化钾(K2O)'],
                                z = DF[DF.clusters == C]['氧化钙(CaO)'],
                                 mode = 'markers', marker_size = 8, marker_line_width = 1,
                                name =  str(C)))

PLOT.update_layout(width = 800, height = 800, autosize = True, showlegend = True,
                                scene  =  dict(xaxis=dict(title  =  '二氧化硅(SiO2)%', titlefont_color = 'black'),
                                               yaxis=dict(title  =  '氧化钾(K2O)%', titlefont_color = 'black'),
                                               zaxis=dict(title  =  '氧化钙(CaO)%', titlefont_color = 'black')),
                    font = dict(family = "Gilroy", color  = 'black', size = 12))
```

附录 5：

介绍：问题三预测以及画图算法，在 jupyter-notebook 平台编写

```
df_af_wea = pd.read_excel("data.xlsx",sheet_name="3_风化",index_col=0)
df_af_wea = df_af_wea.fillna(0)
df_af_wea.head()
def filter_2(data,alpha,A_change,A_max,A_min):
    new = data - (1+alpha)*A_change
    for i in range(A_max.shape[0]):
        if new[i]>A_max[i]:
            new[i] = A_max[i]
        elif new[i] < A_min[i]:
```

```
            new[i] = A_min[i]
    return new
for i in range(df_af_wea.shape[0]):
    count_y = 0
    count_l = 0
    for alpha in np.arange(-1,1,0.1):

        x_pre = filter_2(df_af_wea.iloc[i].values,alpha,Indi.loc['AV_change'].values,
                Indi.loc['AVmax'].values,Indi.loc['AVmin'].values)
        if estimator.best_estimator_.predict([x_pre])[0] == 1:
            count_y = count_y+ 1
        else:
            count_l = count_l+1
    print(df_af_wea.index[i])
    print("1:",count_y)
    print("0:",count_l)
df_af_wea.iloc[0] = filter_2(df_af_wea.iloc[0].values,0,Indi.loc['Pb_change'].values,
             Indi.loc['Pbmax'].values,Indi.loc['Pbmin'].values)


# In[53]:


df_af_wea.iloc[1] = filter_2(df_af_wea.iloc[1].values,0,Indi.loc['Pb_change'].values,
             Indi.loc['Pbmax'].values,Indi.loc['Pbmin'].values)


# In[54]:


df_af_wea.iloc[2] = filter_2(df_af_wea.iloc[2].values,0,Indi.loc['K_change'].values,
             Indi.loc['Kmax'].values,Indi.loc['Kmin'].values)


# In[55]:


df_af_wea.iloc[3] = filter_2(df_af_wea.iloc[3].values,0,Indi.loc['K_change'].
```

```
values,
                Indi.loc['Kmax'].values,Indi.loc['Kmin'].values)


# In[56]:


df_af_wea.to_excel("After.xlsx")


# In[57]:


# for A5 if High K


# In[58]:


df_af_wea.iloc[1] = filter_2(df_af_wea.iloc[1].values,0,Indi.loc['K_change'].
values,
                Indi.loc['Kmax'].values,Indi.loc['Kmin'].values)
df_af_wea


# ## 预测未分化的种类：


# In[59]:


df_be_wea = pd.read_excel("data.xlsx",sheet_name="3_未风化",index_col=0)
df_be_wea = df_be_wea.fillna(0)


# In[60]:


df_be_wea


# In[61]:
```

```
88.
89. for i in range(4):
90.      print(estimator.best_estimator_.predict([df_be_wea.iloc[i].values])[0])
91.
92.
93. # # 亚分类预测
94.
95. # In[62]:
96.
97.
98. Un_pb = pd.read_excel("Q3.xlsx",sheet_name="Pb",index_col=0)
99. Un_pb = Un_pb.fillna(0)
100.    Un_pb
101.
102.
103.    # In[63]:
104.
105.
106.    Un_k = pd.read_excel("Q3.xlsx",sheet_name="K",index_col=0)
107.    Un_k = Un_k.fillna(0)
108.    Un_k
109.
110.
111.    # In[64]:
112.
113.
114.    for i in range(Un_pb.shape[0]):
115.        print(ms.predict([Un_pb.iloc[i].values])[0])
116.
117.
118.    # In[74]:
119.
120.
121.    for i in range(Un_k.shape[0]):
122.        print(ms_k.predict([Un_k.iloc[i].values])[0])
123.
124.
125.    # ## 增加 $a$ 干扰相进行敏感性分析
126.
127.    # In[75]:
128.
```

```
129.
130.    Indi
131.
132.
133.    # In[76]:
134.
135.
136.    # 对于铅钡玻璃需要 PbO 和 BaO 的变化值和上下边界值
137.    Pb_Ba_change = np.array([13.278098,-0.012009])
138.    Pb_Ba_max = np.array([39.220000,26.230000])
139.    Pb_Ba_min = np.array([9.300000,3.420000])
140.
141.
142.    # In[77]:
143.
144.
145.    # 对于高钾玻璃玻璃需要 CaO 和 K2O 的变化值和上下边界值
146.    Ca_K_change = np.array([-8.787500,-4.462500])
147.    Ca_K_max = np.array([14.520000,8.700000])
148.    Ca_K_min = np.array([5.190000,0.000000])
149.
150.
151.    # In[78]:
152.
153.
154.    Un_pb
155.
156.
157.    # In[79]:
158.
159.
160.    def filter_3(data,alpha,A_change,A_max,A_min):
161.        new = data - (alpha)*A_change
162.        for i in range(A_max.shape[0]):
163.            if new[i]>A_max[i]:
164.                new[i] = A_max[i]
165.            elif new[i] < A_min[i]:
166.                new[i] = A_min[i]
167.        return new
168.
169.
170.    # In[80]:
```

```
171.
172.
173.    ms.predict([filter_3(Un_pb.iloc[0].values,0.1,Pb_Ba_change,Pb_Ba_max,Pb_Ba_min)])[0]
174.
175.
176.    # In[81]:
177.
178.
179.    result = np.array([])
180.
181.
182.    # In[82]:
183.
184.
185.    def Sensitive_3(Or_data,Range,Change,Max,Min,Model):
186.        result = np.zeros_like(Range)
187.        j=0
188.        for alpha in Range:
189.            result[j] = Model.predict([filter_3(Or_data,alpha ,Change,Max,Min)])[0]
190.    #         print(result[j])
191.            j=j+1
192.        return result
193.
194.
195.    # In[83]:
196.
197.
198.    Sensitive_3(Un_k.iloc[0].values,np.arange(-0.5,0.5,0.1),Ca_K_change,Ca_K_max,Ca_K_min,ms_k)
199.
200.
201.    # In[84]:
202.
203.
204.    for i in range(Un_pb.shape[0]):
205.        print(Sensitive_3(Un_pb.iloc[i].values,np.arange(-0.5,0.5,0.1),Pb_Ba_change,Pb_Ba_max,Pb_Ba_min,ms) )
206.
207.
208.    # In[85]:
```

```
    for i in range(Un_k.shape[0]):
        print(Sensitive_3(Un_k.iloc[i].values,np.arange(-0.5,0.5,0.1),Ca_K_change,Ca_K_max,Ca_K_min,ms_k) )


    # ## Plot

    # In[86]:


    Indi


    # In[87]:


    Un_pb


    # In[88]:


    from plotly.subplots import make_subplots
    import plotly.graph_objects as go
    fig = make_subplots(rows=2, cols=3,horizontal_spacing=0.1)

    xx = np.arange(-0.5,0.5,0.1)
    for i in range(3):
        y_1 = Sensitive_3(Un_pb.iloc[i].values,np.arange(-0.5,0.5,0.1),Pb_Ba_change,Pb_Ba_max,Pb_Ba_min,ms)
        y_1 = y_1.astype('int').astype('str')
        y_1[y_1=='0'] = '高铅类'
        y_1[y_1=='1'] = '均匀类'
        y_1[y_1=='2'] = '高钡类'
        fig.add_trace(
            go.Scatter(x=xx, y=y_1,name=Un_pb.index[i]),
            row=1, col=i+1
        )

    for i in range(3,5):
```

```
        y_1 = Sensitive_3(Un_pb.iloc[i].values,np.arange(-0.5,0.5,0.1),Pb_Ba_change,Pb_Ba_max,Pb_Ba_min,ms)
        y_1 = y_1.astype('int').astype('str')
        y_1[y_1=='0'] = '高铅类'
        y_1[y_1=='1'] = '均匀类'
        y_1[y_1=='2'] = '高钡类'
        fig.add_trace(
            go.Scatter(x=xx, y=y_1,name=Un_pb.index[i]),
            row=2, col=i-2

        )

    fig.update_layout(height=500, width=600, title_text="敏感性分析")
    fig.show()


    # In[89]:


    Un_k


    # In[90]:


    fig = make_subplots(rows=2, cols=2,horizontal_spacing=0.1)

    for i in range(2):
        y_1 = Sensitive_3(Un_k.iloc[i].values,np.arange(-0.5,0.5,0.1),Ca_K_change,Ca_K_max,Ca_K_min,ms_k)
        y_1 = y_1.astype('int').astype('str')
        y_1[y_1=='0'] = '钾钙类'
        y_1[y_1=='1'] = '钾硅类'
        fig.add_trace(
            go.Scatter(x=xx, y=y_1,name=Un_k.index[i]),
            row=1, col=i+1
        )

    for i in range(2,4):
        y_1 = Sensitive_3(Un_k.iloc[i].values,np.arange(-0.5,0.5,0.1),Ca_K_change,Ca_K_max,Ca_K_min,ms_k)
        y_1 = y_1.astype('int').astype('str')
```

```
        y_1[y_1=='0'] = '钾钙类'
        y_1[y_1=='1'] = '钾硅类'
        fig.add_trace(
            go.Scatter(x=xx, y=y_1,name=Un_k.index[i]),
            row=2, col=i-1

        )

    fig.update_layout(height=500, width=600, title_text="敏感性分析")
    fig.show()
```

附录 6：

介绍：铅钡类玻璃化学成分——正态性检验结果

变量名	样本量	平均值	标准差	偏度	峰度	S-W 检验	K-S 检验
二氧化硫(SO2)	49	0.8	3.139	4.473	19.619	0.275(0.000***)	0.499(0.000***)
氧化锡(SnO2)	49	0.058	0.213	4.773	25.716	0.307(0.000***)	0.506(0.000***)
氧化钠(Na2O)	49	0.904	1.813	2.299	5.069	0.58(0.000***)	0.405(0.000***)
氧化铝(Al2O3)	49	3.668	3.009	2.211	5.531	0.761(0.000***)	0.155(0.169)
二氧化硅(SiO2)	49	38.876	18.646	0.147	-1.019	0.961(0.102)	0.104(0.629)
氧化锶(SrO)	49	0.348	0.264	0.768	0.612	0.934(0.008***)	0.104(0.626)
氧化钙(CaO)	49	2.05	1.635	0.745	-0.291	0.92(0.003***)	0.147(0.221)
氧化镁(MgO)	49	0.646	0.63	0.763	0.7	0.865(0.000***)	0.235(0.007***)
氧化钾	49	0.173	0.276	2.744	9.28	0.654(0.000***)	0.264(0.002***)

(K2O)							
氧化铅(PbO)	49	33.349	14.947	0.39	-0.622	0.962(0.116)	0.09(0.785)
氧化铁(Fe2O3)	49	0.656	0.948	2.067	5.316	0.732(0.000***)	0.245(0.005***)
五氧化二磷(P2O5)	49	3.293	3.909	1.097	0.21	0.816(0.000***)	0.223(0.013**)
氧化钡(BaO)	49	10.49	8.331	1.612	2.253	0.815(0.000***)	0.253(0.003***)
氧化铜(CuO)	49	1.88	2.47	2.286	5.256	0.692(0.000***)	0.274(0.001***)

注：***、**、*分别代表 1%、5%、10%的显著性水平

图表说明：

上表展示了定量变量二氧化硫(SO2)、氧化锡(SnO2)、氧化钠(Na2O)、氧化铝(Al2O3)、二氧化硅(SiO2)、氧化锶(SrO)、氧化钙(CaO)、氧化镁(MgO)、氧化钾(K2O)、氧化铅(PbO)、氧化铁(Fe2O3)、五氧化二磷(P2O5)、氧化钡(BaO)、氧化铜(CuO)的描述性统计和正态性检验的结果，包括均值、标准差等，用于检验数据的正态性。

附录 7：

介绍：高钾玻璃化学成分——正态性检验结果

变量名	样本量	平均值	标准差	偏度	峰度	S-W 检验	K-S 检验
二氧化硅(SiO2)	17	67.473	6.974	1.604	3.244	0.849(0.010**)	0.219(0.341)
氧化钠(Na2O)	17	0.736	1.052	1.668	1.924	0.71(0.000***)	0.339(0.030**)
氧化钾(K2O)	17	9.88	2.206	0.109	0.71	0.959(0.606)	0.186(0.539)
氧化钙(CaO)	17	5.369	2.574	-1.033	0.653	0.878(0.029**)	0.22(0.331)

氧化锶(SrO)	17	0.044	0.039	0.599	-0.351	0.869(0.021**)	0.231(0.278)
氧化镁(MgO)	17	1.071	0.585	-0.353	-0.528	0.939(0.310)	0.156(0.749)
二氧化硫(SO2)	17	0.108	0.152	1.549	1.298	0.709(0.000***)	0.339(0.030**)
五氧化二磷(P2O5)	17	1.42	1.192	1.846	3.347	0.757(0.001***)	0.303(0.070*)
氧化铝(Al2O3)	17	6.645	2.132	0.477	0.163	0.96(0.641)	0.142(0.834)
氧化锡(SnO2)	17	0.208	0.563	3.927	15.846	0.383(0.000***)	0.449(0.001***)
氧化铁(Fe2O3)	17	1.906	1.378	1.413	4.642	0.816(0.003***)	0.217(0.350)
氧化铅(PbO)	17	0.377	0.464	1.93	3.46	0.715(0.000***)	0.353(0.021**)
氧化钡(BaO)	17	0.518	0.734	2.236	6.161	0.702(0.000***)	0.28(0.114)
氧化铜(CuO)	17	2.404	1.457	0.273	-0.543	0.968(0.776)	0.168(0.660)

注：***、**、*分别代表 1%、5%、10%的显著性水平

图表说明：

上表展示了定量变量二氧化硅(SiO2)、氧化钠(Na2O)、氧化钾(K2O)、氧化钙(CaO)、氧化锶(SrO)、氧化镁(MgO)、二氧化硫(SO2)、五氧化二磷(P2O5)、氧化铝(Al2O3)、氧化锡(SnO2)、氧化铁(Fe2O3)、氧化铅(PbO)、氧化钡(BaO)、氧化铜(CuO)的描述性统计和正态性检验的结果，包括均值、标准差等，用于检验数据的正态性。