基于梯度提升决策树和多元回归的古代玻璃制品分类与成分分析模型

摘要

本文针对古代玻璃文物风化状态检测和文物分类问题，以WIP和PbO/SiO_2为重要指标研究与风化状态的统计关系，并建立岭回归化学成分预测模型，之后基于决策树和聚类分析建立分类模型，最后建立多元线性回归模型对化学成分关联性进行分析。

针对问题一，首先基于*Spearman*相关系数和卡方检测对玻璃理化性质和风化状态进行相关性和差异性分析，发现玻璃类型与风化状态存在高度相关关系，得到高钾类更有可能不被风化，铅钡类更有可能被风化的结论。由于化学成分较多，选取K_2O和BaO等重要指标，并引入风化指数WIP和 PbO/SiO_2作为风化状态判断重要依据，带入**支持向量机**中，得到基于指标 **WIP** 的高钾玻璃风化统计模型和基于PbO/SiO_2和BaO的铅钡玻璃风化统计模型，准确率达到**85.7%**，同时从横向纵向两个角度对玻璃风化与化学成分之间的统计规律进行分析，得到PbO/SiO_2和风化状态存在极大关系，证明指标选取的合理性。针对定类变量，基于*onehot*编码构建以四个理化特征为自变量，化学成分含量为因变量的**岭回归化学成分预测模型**，并与前文分析得到的风化前后化学成分变化关系进行对比，证明模型的预测结果与前文得到的结论基本符合。

针对问题二，建立**决策树**分类模型，得到以PbO为决策节点的决策树分类结构，得出以$PbO \le 5.46\%$划分的分类规律，并由此证明PbO为重要特征的合理性。采用聚类分析对高钾和铅钡玻璃进行亚类划分，由肘部图确定**高钾玻璃分为 3 个亚类，铅钡玻璃分为 5 个亚类**的结论，再利用决策树模型对高钾和铅钡玻璃分类，得到以SiO_2、P_2O_5为指标的高钾玻璃亚分类模型和以SiO_2、PbO、BaO为指标的铅钡玻璃亚分类模型。并查找资料给出玻璃类型与化学成分关系表，将模型所分亚类的化学成分特征与关系表对比，找到与各亚类对应的玻璃类型，证明选取重要指标的模型分类方式具有合理性。通过，改变聚类数量得到聚类数大于 3 时模型的敏感性较低的结论。最后，对决策树模型的性能进行评价，模型分类准确率为 **71.4%**。

针对问题三，为提高分类准确率并解决决策树过拟合的问题，对两类玻璃分别构建基于 **GBDT** 的亚分类预测模型，并在表 11 中给出编号 A1~A8 的文物类型和亚类预测结果。从性能和模型参数两方面对模型进行敏感性分析性能角度通过使得各化学成分含量上下浮动，得到以化学成分变异百分比为自变量，模型预测准确率为因变量的敏感性分析图，得到模型对于 10%以内的扰动具有很好的容错性；参数敏感性角度，通过改变基学习器个数和决策树深度，得到高钾玻璃和铅钡玻璃分别在学习器个数大于 6 和 4，树深大于 3 时敏感性较低。最后模型分类准确率达到 **95%**以上。

针对问题四，两类玻璃分别通过相关性分析得到 14 种化学成分之间的相关系数，对每一种化学成分选取与之相关性最大的几个元素，建立**多元线性回归模型**，得到化学成分之间的关联性。对 14 个关联组的拟合优度进行排序，结合相关性选优和拟合优度选优两次筛选，选取其中最大的 5 组，进行不同类之间化学元素关联性的差异性分析。并对不同类间SiO_2和BaO含量的差异性结合机理进行分析，得到铅钡玻璃中SiO_2含量受PbO影响最为显著（$r=-0.76$），并对其影响机理进行分析，高钾玻璃SiO_2含量主要受K_2O影响（$r=-0.86$）,受PbO影响较小（$r=-0.32$），证明了前文得到的成分变化规律的正确性。铅钡玻璃中BaO与CuO相关性较强，而高钾玻璃中相关性较弱。

关键词：机理分析 风化指数 岭回归 GBDT 决策树 聚类分析 多元线性回归

一、 问题重述

1.1 问题背景

作为古丝绸之路中西方贸易往来的宝贵物证，对古代玻璃实施保护具有重大意义。在考古工作中主要依据对玻璃文物的化学成分和其他检测手段进行玻璃分类，因此如何建立各化学成分含量和玻璃类型的关系模型，从而提高对玻璃所属类型和各化学成分含量关系的准确性具有重要研究意义。

1.2 问题提出

问题一：第一小问，根据附件一中所给数据，分别对玻璃类型、纹饰和颜色和表面是否风化进行相关性和差异性分析。第二小问，将附件一、二编号对应，分析玻璃样本表面是否风化和各化合物含量变化之间的统计规律。第三小问，建立文物纹饰、类型、颜色、表面风化和各个化合物含量之间的多元线性关系，预测风化点被风化之前各化学成分的含量。

问题二：第一小问，选取附件二中某种或某些化合物含量对高钾玻璃和铅钡玻璃进行高效分类。第二小问，选取合适的指标，对高钾和铅钡玻璃分别再进行亚类划分，并证明分类模型的合理性和敏感性。

问题三：对附件三中未知风化状态的样本的化学成分进行分析，鉴别样本所属亚类。然后对模型分类的结果进行敏感性分析。

问题四：结合前三问分析得到的结论，对两类玻璃的化学成分关系进行分析，得到多组关联性化学成分，并将两类玻璃的关联性进行比较，分析它们的差异性。

二、 问题分析

2.1 问题一分析

为研究玻璃文物各种理化指标和表面风化之间的关系，依据附件一中所给数据，首先分析三种特征都属于定类变量，因此采用 *Spearman* 相关系数求得相关系数矩阵，分析表面风化与各指标之间的相关系数，并采用卡方检测对其进行差异性检测。并建立在前面分析的基础上，依据附件二中所给的具体各化合物含量，对各化合物含量与表面风化程度进行相关性和差异性分析，由于附件二中化合物种类较多，通过查找相关文献得到重要特征化合物，从而分析化学成分和风化程度的统计规律。并以理化性质为自变量，各化学元素为因变量，构建训练数据，求得回归方程，从而实现对未风化前的化合物含量的预测。

2.2 问题二分析

问题二要求找出划分为高钾和铅钡玻璃的分类规律，并找到划分依据分别对两类玻璃进一步划分，并分析模型的合理性和敏感性。为寻找高钾和铅钡玻璃划分类别的依据，分别对两种玻璃的各化合物含量进行研究，在问题一相关性分析的得到重要相关性特征的基础上，为了展现具体的划分依据，建立决策树模型找到对应特征的定量分类规律，进而证明问题一中相关性分析的准确性。在大类划分的基础上进行亚类划分，采用聚类分析将高钾和铅钡玻璃分别进行聚类，基于 SSE 的肘部法则选取聚类个数，得到划分类数后带入决策树模型得到定量分类指标。并结合实际情况分析分类方法的合理性，通过改变肘部图 K 值分析残差平方和 SSE 对聚类数的敏感性。

2.3 问题三分析

问题三建立在问题二划分亚类的基础上，要求根据附件三中未知风化状态的玻璃化

学成分进行大类和亚类划分，并检测分类结果的敏感性。先根据前文得到的决策树模型将样本划分为两大类，以第二问中得到的亚分类数据作为训练集，为了降低提高模型的分类效果、减少过拟合，分别构建两类玻璃的梯度提升决策树模型对样本进行亚类划分。为对结果敏感性进行分析，考虑从模型参数的敏感性和模型性能的稳定性两方面出发，通过改变基学习器个数和树深度评价模型的敏感性，通过使得化合物含量上下浮动测试模型结果分类是否敏感。

2.4 问题四分析

问题四要求对每类玻璃的化学成分的相关性进行分析，得到多组关联关系，并将两类玻璃化学成分关联关系的差异性。对两类玻璃分别进行相关性分析，得到与每个化合物相关性最强的前几种化合物，分析他们之间的关联性，并通过构建多元线性回归方程得到每种化学元素与选中的几种化学元素之间的函数关系，并通过拟合优度排序选取前组关联关系。最后，将两类玻璃所选取的四组关联关系进行差异性分析。

三、 数据预处理

3.1 数据合理性检测

根据题目要求，由于检测手段等原因造成测量误差，样本各化合物总含量累加介于85%~105%即可认为是合理性数据。对附件二进行分析，样本 15 和 17 的总成分不处于合理性区间中，因此将其进行删除。

3.2 缺失值处理

对于附件一中部分颜色缺失的样本，本文选择将其删除。

四、 模型假设

(1) 假设没有检测到的含量默认为 0
(2) 假设可以通过找到部分化合物含量来对玻璃进行划分

五、 符号说明

符号	符号描述	单位
WIP	帕克风化指数	无
$CaO*$	化学成分含量校正项	mol
PbO/SiO_2	铅硅比	无
c_{ba}	PbO浓度	mol

六、 问题一模型建立与求解

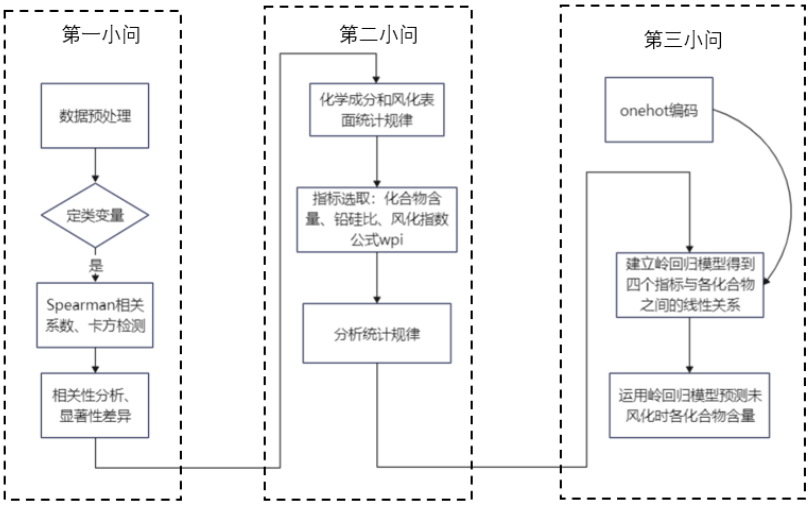

图 1 问题一流程图

6.1 第一小问

6.1.1 相关性和差异性分析

根据分析，各指标均属于定类变量，因此采用 *Spearman* 相关系数分别对各变量的相关性进行分析。首先对各定类指标进行编秩，将纹饰按字母 A、B、C 顺序编为 1、2、3，将高钾和铅钡分别编为 1,2，将颜色由浅到深编为 1,2……8，编秩完成后，将秩次带入 *Pearson* 相关系数的计算公式中：

$$r_s = \frac{1}{n-1}\sum\left(\frac{p_i-\bar{p}}{s_p}\right)\left(\frac{q_i-\bar{q}}{s_q}\right) \tag{1}$$

分别对各对变量进行相关性分析，得到相关系数矩阵和显著性值，当显著性小于 0.05 即认为指标之间具有较强的相关性。

为了比较不同指标之间的差异，得到更多有价值的关系结论，本文选取卡方检测对各指标进行差异性分析。

6.1.2 模型的求解

运用 SPSS 进行相关性分析，得到相关性系数如下表所示（其中括号中值表示显著性水平）：

表 1 相关性系数表

	纹饰	类型	颜色	表面风化
纹饰	1.000(0.000***)	-0.135(0.332)	-0.270(0.048**)	0.048(0.731)
类型	-0.135(0.332)	1.000(0.000***)	-0.172(0.215)	-0.316(0.020**)

颜色	-0.270(0.048**)	-0.172(0.215)	1.000(0.000***)	0.065(0.642)
表面风化	0.048(0.731)	-0.316(0.020**)	0.065(0.642)	1.000(0.000***)

注：***、**、*分别代表 1%、5%、10%的显著性水平

分析相关性系数可知玻璃文物表面风化程度与类型的显著性水平 0.02<0.05，因此认为两者之间存在较高的相关性。结合实际并查找相关文献，文物类型不同其化学性质和化学成分组成也不同，决定了风化程度的不同，因此分析较为合理。

下面给出表面风化与其余指标之间的差异性（其他各变量卡方检测结果见附件一）

表 2 卡方检测结果

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

注：***、**、*分别代表 1%、5%、10%的显著性水平

由表可知文物风化与未风化之间的类型存在显著性差异，铅钡类型更有可能被风化，高钾类型更有可能不被风化，因此初步推测风化可能与铅钡元素含量有关。

对风化与玻璃颜色关系进行可视化，由图可知玻璃是否风化与颜色相关性不大，但风化后出现表面为黑色的玻璃。

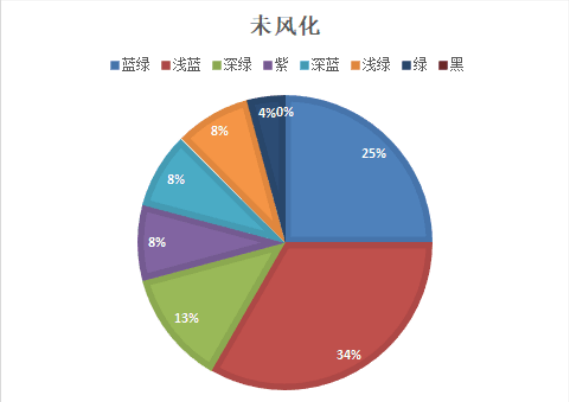

图 2 未风化玻璃颜色分布

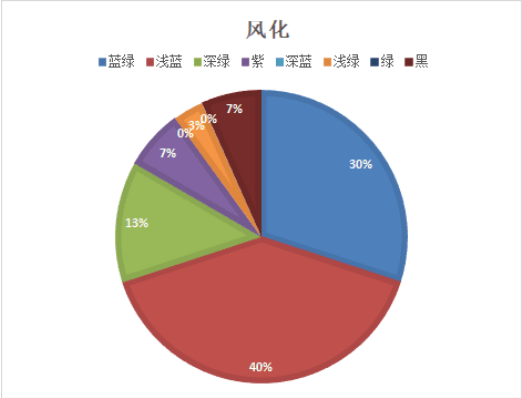

图 3 风化玻璃颜色分布

6.2 第二小问

6.2.1 基于指标 WIP 的高钾玻璃风化程度统计模型

查找相关文献,在化合物风化过程中，例如 K2O、MgO、CaO 等活泼金属组成的化合物化学性质不稳定，在风化过程中容易发生带出；而对于 SiO_2、Al_2O_3、Fe_2O_3 等主量氧化物在风化过程中不容易发生带入带出。[3]

不同玻璃类型其主要化合物组成呈现较大差异，因此本文对高钾和铅钡类玻璃分别进行分析。我们引入帕克风化指数对高钾类玻璃的风化情况进行分析，风化指数是用样本各种化合物含量的变化来衡量样本风化程度的指标。[3]在现有的多种风化指数中，我们选取帕克风化指数(WIP)进行表面风化的分类：

$$WIP = 100 \times (2Na_2O / 0.35 + 2K_2O / 0.25 + MgO / 0.9 + CaO^* / 0.7) \tag{2}$$

CaO^*是运用 McLennan 提出的方法进行的校正项，当$n(CaO) < n(Na_2O)$时，样品CaO^*取 $n(CaO)$，反之采用$n(Na_2O)$作为CaO^*。[2]

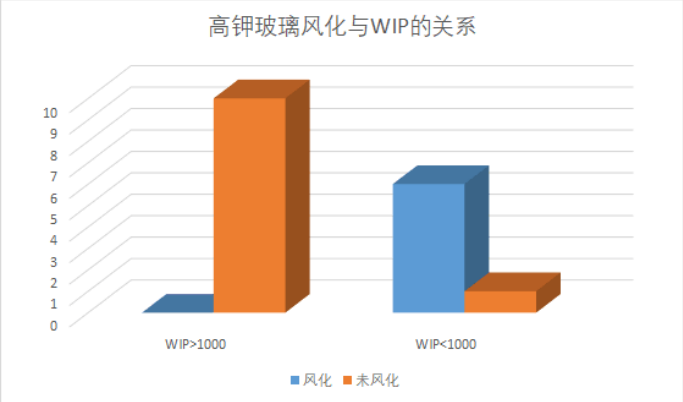

图 4 WIP 指数与风化程度的关系

以 WIP=1000 作为分段指标，大于 1000 的样本大量未风化，小于 1000 的样本大量风化了，除个别点以外预测值和真实值均符合。因此用帕克风指数来判别高钾玻璃风化是合理的。

6.2.2 基于 PbO/SiO_2 和 BaO 指标的铅钡玻璃风化程度统计模型

WIP 没有考虑氧化铅和氧化钡的含量，用该指标描述铅钡玻璃的风化程度准确性很低。因此，查阅文献可知[5]，PbO/SiO_2 比值增加时，可引起 K 析出速率的迅速增加，即风化速率对铅硅比敏感，所以考虑将该指标引入评价指标。

钡磷酸盐玻璃风化时，与亚基团相连的$[PO_4]$四面体和磷酸盐基团水解生成容易脱落的焦磷酸盐，水分可直接通过风化层直接与玻璃表面接触，所以钡磷酸盐玻璃的风化速率比硅酸盐玻璃要快，因此钡磷酸盐玻璃的风化程度比硅酸盐玻璃的风化程度严重，所以本文将氧化钡 BaO 含量加入风化指数作为修正。[1]

经过以上分析，考虑将铅硅比与氧化钡含量作为风化程度的评价指标，记$r = PbO/SiO_2$,以 r 和 BaO 为指标建立支持向量机模型：

$$f(x) = w^T x + b = \sum_{j=1}^{n} w_j x_i + b = 0 \tag{3}$$

式中 $x_i = [r_i, c_{ba_i}]$，其中 c_{ba} 是 PbO 的浓度，b 为一个标量，用于确定最优分类平面，

权重向量 $w=\sum_{i=1}^{n}\alpha_i x_i$。因此，式（3）变换为

$$f(x)=w^T x+b=\sum_{j=1}^{n}\alpha_i x_i{}^T x+b=0 \tag{4}$$

针对二维任意直线可表示为 (w,b)。样本任意点 x_i 到直线的距离可表示为 $r=\frac{|w^T x+b|}{\|w\|}$，直线能将样本正确分类需要满足，对于任意 $x_i=[r_i,c_{ba_i}]$ 有

$$\begin{cases} w^T x_i+b\geq 1, y_i=1; \\ w^T x_i+b\leq -1, y_i=-1 \end{cases} \tag{5}$$

两个异类支持向量到直线的距离之和为 $r=\frac{2}{\|w\|}$，为使 r 最大，则使得 w 最小，即求

$$\min\frac{1}{2}\|w\|^2 \tag{6}$$

下面对基于 PbO/SiO_2 和指标的 BaO 铅钡玻璃风化程度统计模型求解，得到如下图所示结果：

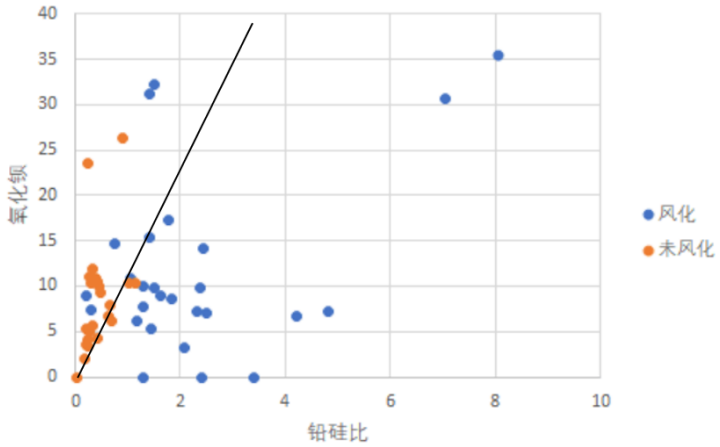

图 5 基于铅硅比和氧化钡含量的支持向量分类

表 3 支持向量机模型性能分析

	准确率	召回率	精确率	F1
训练集	0.857	0.857	0.852	0.854
测试集	1	1	1	1

如图所示，模型分类效果较为优秀，结合具体数据，模型准确度达到 85.7%，F1 值达到 85.4%，因此铅钡玻璃是否风化能够以 PbO/SiO_2 和 BaO 做为分类指标，且分类效

果较好。

6.2.3 统计规律分析

基于 6.1.3 和 6.1.4 中查找到的相关文献知识，本文结合所给数据进行验证。我们从横向和纵向两方面对化学元素和风化程度的统计关系进行分析。

(1)纵向分析

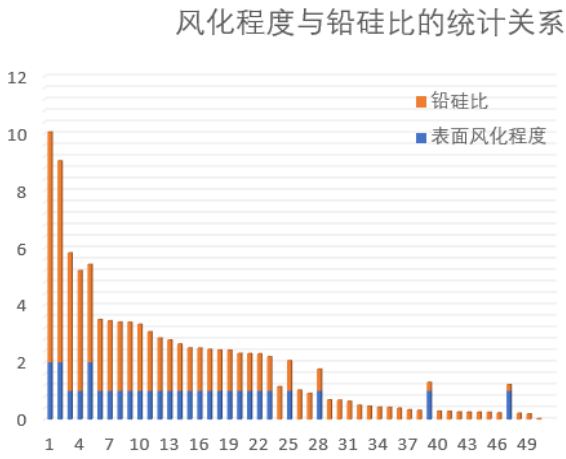

图 6 风化程度与铅硅比的统计关系

我们定义未风化，风化，严重风化依次为 0,1,2，定义铅硅小于 1 是低、(1.1,4)是中、大于 4 是高。铅硅比高的玻璃都属于铅钡玻璃，都严重风化；未风化的玻璃都是铅硅比低的玻璃，其中大部分高钾玻璃铅硅比低，同时也包含部分铅钡玻璃；大部分铅硅比中等的玻璃表面都出现风化。

表 4 铅硅比和表面风化统计表

	名称	铅硅比			总计
		中	低	高	
表面风化	严重风化	0(0.0%)	0(0.0%)	3(100.0%)	3
	未风化	0(0.0%)	22(100.0%)	0(0.0%)	22
	风化	20(83.3%)	3(12.5%)	1(4.2%)	24

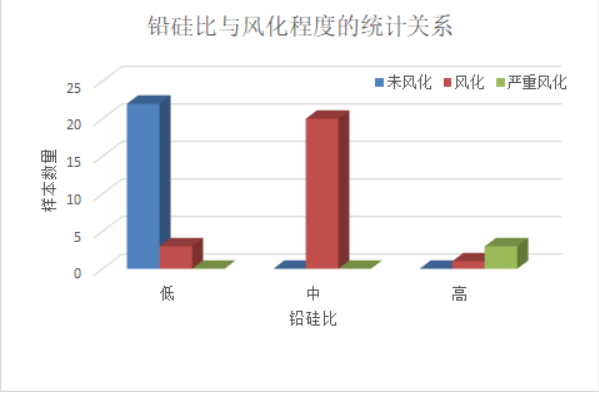

图 7 铅硅比与风化程度的统计关系

(2)横向分析

图 8 对于高钾类玻璃，风化前后，四种化合物的含量进行可视化。四种元素含量风化后都有所降低，其中 K_2O、CaO 的含量有明显降低，印证了文献[3]中活泼金属化合物在风化过程中容易发生带出的结论。

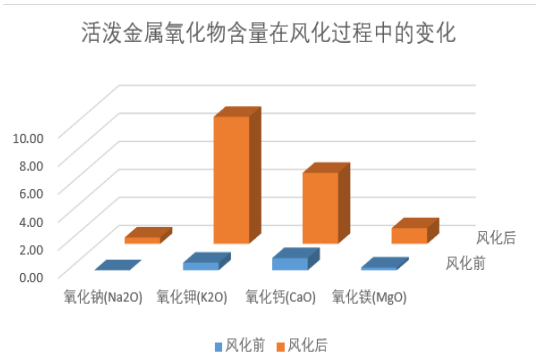

图 8 高钾玻璃氧化钾含量与风化关系图

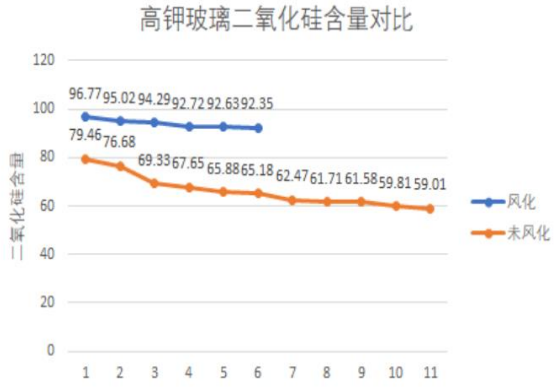

图 9 高钾玻璃二氧化硅含量对比

统计发现高钾类玻璃，只有纹饰 B 发生了风化，因此我们从化学元素角度考虑其本质原因。发现 SiO_2、K_2O、CaO 等元素含量在风化样本和无风化样本之间存在明显差异。因此在高钾类玻璃中，随着玻璃的风化二氧化硅和氧化钾等含量都有所减小。

6.3 第三小问

6.3.1 岭回归化学成分预测模型建立

为得到玻璃未风化前各化合物含量，结合岭回归适用于输入为定类变量，输出为定量变量，且能够求解数据特征多于样本的情况，与本问题所给条件相符。因此，本文建立以纹饰、类型、颜色、表面是否风化为特征值，各化合物成分含量为标签的岭回归化学含量预测模型。选取无风化样本作为训练数据进行模型训练，拟合得到特征值与各个化合物含量之间的回归方程，从而实现对被风化前样本化学含量的预测。

岭回归代价函数如下：

$$Cost(w)=\sum_{i=1}^{N}(y_i-w^T x_i)^2+\lambda \| w \|_2^2 \tag{7}$$

化为以 θ 表示为：

$$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^\theta)-y^{(i)})^2+\lambda\sum_{j=1}^{n}\theta_j^{\,2}\right] \tag{8}$$

求得使得岭回归代价函数最小时的 w 值，通过对代价函数求导即可得到 w 的解，公式如下：

$$w=(X^T X+\lambda I)^{-1}X^T y \tag{9}$$

其中 $\lambda \in R$,X 为 $N\times M$ 矩阵，y 为 N 维列向量，I 为 $M\times M$ 大小的单位矩阵，λ 为岭系数。通过选择 λ 值，使得各个回归系数的岭估计基本平稳，且 θ 值较小。

6.3.2 岭回归模型求解与分析

岭回归基于 *onehot* 编码将每一种变量的情况进行编码，下面用符号对各个特征进行表示。高钾(w_1)、铅钡(w_2)、表面未风化(n)、表面风化(y)、表面严重风化(ys)纹饰 A(a)、纹饰 B(b)、纹饰 C(c)、蓝绿(c_1)、浅蓝(c_2)、紫色(c_3)、深绿(c_4)、深蓝(c_5)、浅绿(c_6)、黑色(c_7)、绿色(c_8)。

岭回归得到 16 个自变量分别与 14 个化合物含量的岭回归公式，选取 9 个岭回归方程（完整回归方程见附录一）如下表：

表 5 岭回归方程

类型	岭回归方程（化合物含量与自变量的关系）	显著性水平
SiO_2	$80.054-15.165w-18.569y-31.289ys+6.148b+42.218c-1.659c_2$ $-7.757c_3+5.539c_4-9.412c_5+6.893c_6-8.816c_7-0.627c_8$	0.847
K_2O	$15.338-7.114\ w-1.22\ y-0.838\ ys-0.806\ b-5.726\ c+0.557\ c_2$ $-0.317\ c_3+0.386\ c_4+0.823\ c_5-0.429\ c_6+0.336\ c_7-0.974c_8$	0.805
PbO	$-14.828+20.345\ w+15.149\ y+16.938\ ys-6.749\ b-18.313\ c+1.236\ c_2$ $-7.839c_3+1.552c_4+11.735\ c_5-1.998\ c_6+5.53\ c_7+6.193c_8$	0.845
BaO	$-5.944+7.368\ w+2.282y+4.34\ ys+1.017\ b-3.283c-2.616c_2$ $+13.767\ c_3-1.467c_4-0.416c_5-2.447c_6-2.887c7-0.738c_8$	0.693
SO_2	$-0.017+0.1\times$类型$-0.08y+7.364ys+0.17b-0.002c-0.513c_2$ $+2.933c_3+0.355c_4-0.281c_5-0.137c_6-0.276\ c_7-0.16c_8$	0.719

根据以上回归方程，将已风化的样本对应自变量带入方程即可得到未风化之前各化合物的含量，且岭回归的显著性水平普遍较高，模型表现较为良好。

本文得到了各化合物含量对应的岭迹图，下面对 *PbO* 的邻迹图进行分析（完整岭迹图见附录一）：

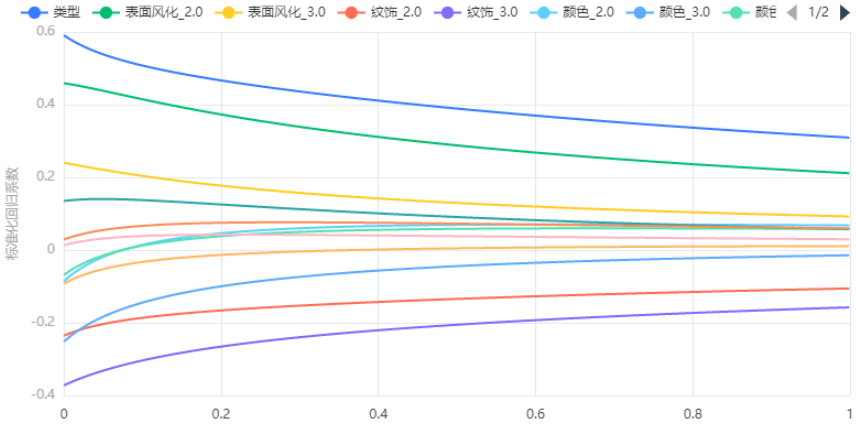

图 10 *PbO* 邻迹图

其中横坐标表示岭系数 λ，纵坐标为标准化回归系数，每一条曲线表示一个回归参数。通过选择 λ 值，使得各参数的回归系数岭估计基本稳定，同时使得 θ 值越大 λ 值越小，因此最佳 λ 值再 0.4 左右。

6.3.3 风化玻璃被风化前化合物含量预测

将风化玻璃样本的四个定类特征带入岭回归模型中，即可得到样本风化之前各化学成分的含量。下面给出部分预测数据。

表 6 风化前含量预测表

	类型	二氧化硅(SiO2)预测	氧化钾(K20)预测	氧化铝(A1203)预测	氧化铅(Pb0)预测	氧化钡(Ba0)预测	五氧化二磷(P205)预测
02	铅钡	69.378	7.975	8.101	0.004	0	1.808
07	高钾	96	9.612	5.039	0	0	0
08	铅钡	57.132	7.907	4.085	0	15.191	0.921
09	高钾	96	9.612	5.039	0	0	0
10	高钾	96	9.612	5.039	0	0	0

我们对预测得到的化合物成分与风化样本自身的化学成分变化关系进行分析，并与6.2.3 中得到的风化前后化学成分变化关系进行对比：

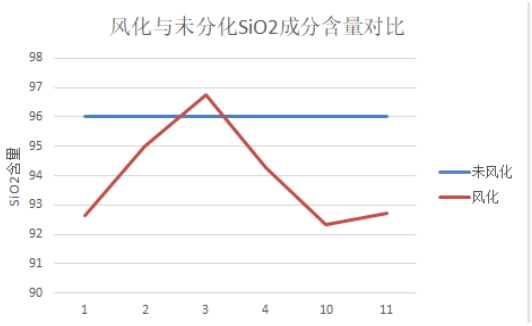

图 11 高钾玻璃预测得到 SiO_2 含量变化

图 12 铅钡玻璃预测得到铅硅比变化

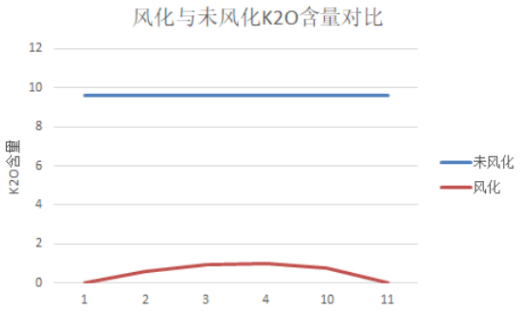

图 13 高钾玻璃预测得到 K_2O 含量变化

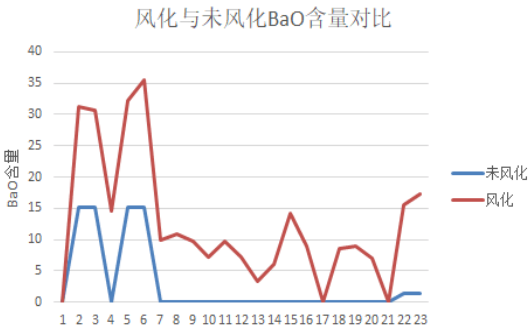

图 14 铅钡玻璃预测得到 BaO 含量变化

由图可知大部分高钾玻璃预测得到被风化前的 SiO_2 含量增加，所有样本预测得到 K_2O 含量增加;所有铅钡玻璃样本预测得到铅硅比减小， BaO 含量减少。与前文得到的风化前后重要特征含量变化基本相同，说明模型预测准确度较好。

七、 问题二：基于决策树和聚类分析的分类模型

7.1 决策树模型介绍

决策树算法一种机器学习算法，能够适用于对离散数值进行分类，其主要工作原理是递归选择最优属性将训练集进行分割，这个属性作为下一个决策节点，从而实现对各个子集的最优划分，其优势在于训练所需要的数据量少，不需要进行数据处理。其具体流程图如下：

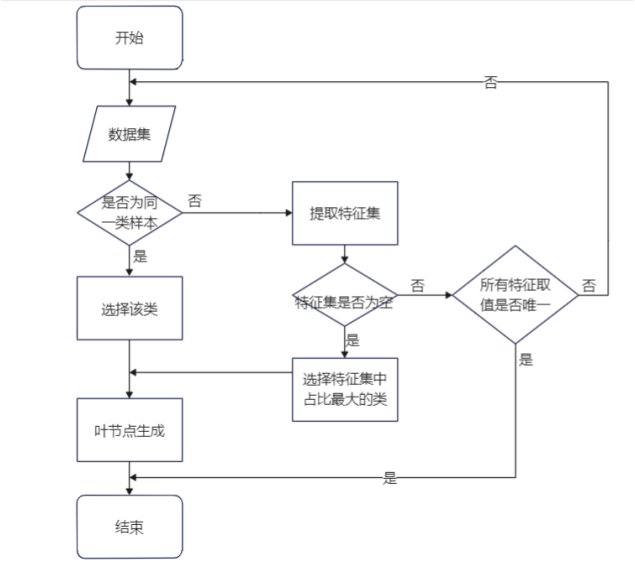

图 15 决策树流程图

7.2 决策树模型求解和分析

基于问题一中对两种古代玻璃的化合物组成成分进行分析，初步掌握了两类玻璃的的化合物组成差异，在此基础上建立决策树模型，将各个化合物含量作为决策树的划分指标，定量找到最好划分类别的化合物及其含量指标。决策树如下图所示：

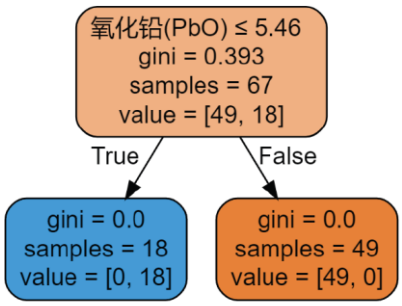

图 16 决策树划分高钾和铅钡玻璃

得到分类准确度最高的决策树，是由 *PbO* 作为决策节点，这与第一问中得到的种化合物是重要特征符合，验证了问题一中结论的正确性。

7.3 K-means 聚类分析

对高钾和铅钡玻璃分别进行亚类划分，按照化学成分含量进行 k 均值聚类分析，依照肘部法得到具体划分的亚类数如下图所示：

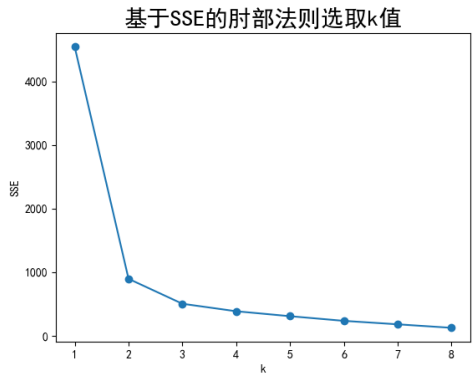

图 17 高钾类肘部法则聚类

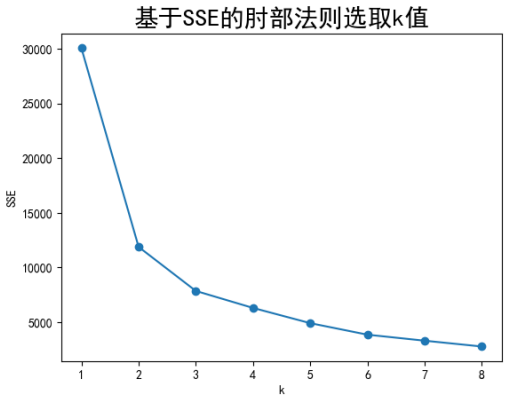

图 18 铅钡类肘部法则聚类

根据肘部法则得到 k 值，再随机选择 k 个样本作为初始聚类中心，通过计算每个样本点到初始聚类中心的距离，将不同的样本点归类到离它最近的聚类中心所在簇类中，认为同一聚类中的样本具有较高的相似性。

由图可知，本文将高钾类玻璃分为 3 类，将铅钡类玻璃分为 5 类，其聚类结果如表所示：

表 7 各亚类样本数量

高钾类				铅钡类				
类别	F1	F2	F3	F4	F5	F6	F7	F8
个数	9	7	2	9	6	4	5	7

同时做出各亚类的风化率如下，

图 19 不同亚类玻璃风化率

再将种类数带入决策树中得到每一类具体的划分标准和样本数量，进而得到具体的划分方案和划分种类，进而证明首先得到在决策树中作为划分节点的重要特征图如下图所示：

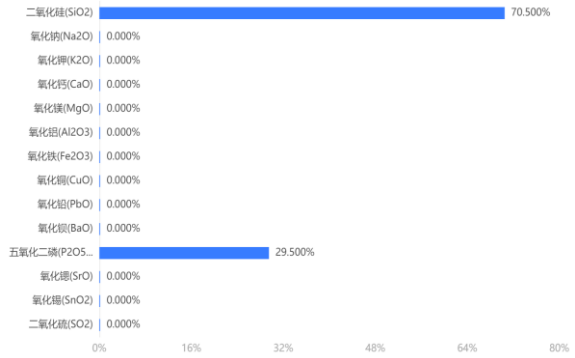

图 20 高钾类重要特征

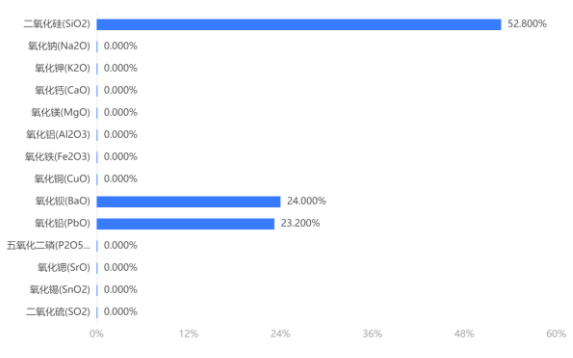

图 21 铅钡类重要特征

将重要特征作为决策节点构成如下图所示决策树结构：

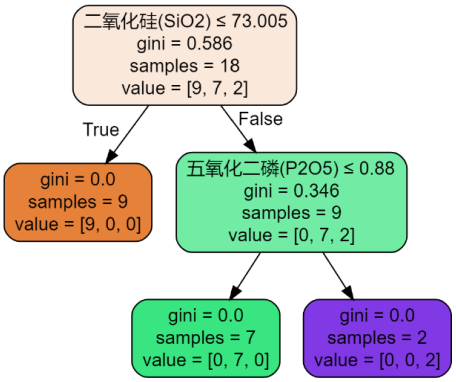

图 22 高钾亚类决策树

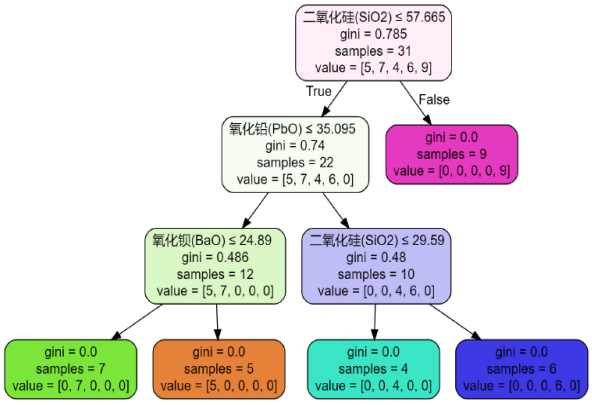

图 23 铅钡亚类决策树

通过上图分类决策树得到划分方法和划分结果如下表所示：

表 8 划分方法和结果

	一级指标	二级指标	三级指标	类别
高钾亚类划分	$SiO_2 \le 73.005\%$			F1
	$SiO_2 > 73.005\%$	$P_2O_5 \le 0.88\%$		F2
	$SiO_2 > 73.005\%$	$P_2O_5 > 0.88\%$		F3
铅钡亚类划分	$SiO_2 > 57.665\%$			F4
	$SiO_2 \le 57.665\%$	$PbO > 35.095\%$	$SiO_2 > 29.59\%$	F5
	$SiO_2 \le 57.665\%$	$PbO > 35.095\%$	$SiO_2 \le 29.59\%$	F6
	$SiO_2 \le 57.665\%$	$PbO \le 35.095\%$	$BaO > 24.89\%$	F7
	$SiO_2 \le 57.665\%$	$PbO \le 35.095\%$	$BaO \le 24.89\%$	F8

7.4 模型合理性和敏感性

下面对高钾玻璃和铅钡玻璃所分的亚类进行合理性解释，下面将不同亚类的化合物含量可视化。

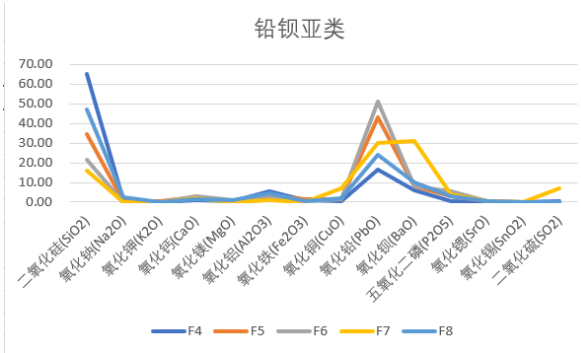

图 24 铅钡玻璃各亚类化合物含量

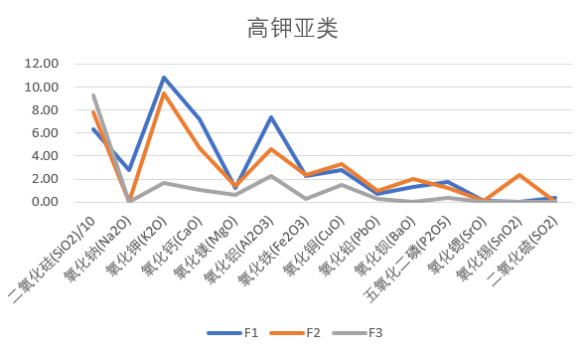

图 25 高钾玻璃各亚类化合物含量

如图 25,对高钾类玻璃 F1 类,，由于 SiO_2 远远大于其他化合物含量，所以对 SiO_2 /10，

因此 F1 中对应化合物含量$\begin{cases} K_2O \geq 11\% \\ SiO_2 \geq 60\% \\ CaO \geq 6\% \end{cases}$,类似对 F1 和 F3 进行分析。

查找玻璃化合物成分含量与玻璃类对应表，可知高钾玻璃依据化合物成分进行划分，分类效果较好，体现了我们亚类划分的合理性。

表 9 玻璃类型和化学成分关系表

类型	类别	K_2O	CaO	SiO_2	SnO_2
钾钙类(硬玻璃)	F1	10%~18%	5%~12%	60%~75%	
高硅氧类	F3			96%左右	
高锡类	F2				>6%

本文通过聚类分析将化学成分相似度较高的进行聚类，依据各化合物成分含量进行分类。表四中铅钡亚类划分所选用的一级和二级指标，对应了问题一中参考相关文献提出的铅硅比对玻璃风化性质具有重要影响，因此本文选取的划分指标不仅是化合物成分，同时也从本质出发以玻璃性质进行划分，进一步体现了模型的合理性。

同时，通过聚类分析得到的划分类数是根据肘部图选取的，如肘部图 17 所示，将高钾玻璃分成三类及以上，将铅钡玻璃分成三类及以上，残差平方 SSE 和对聚类数的敏感度较低，表示样本点预测准确性较稳定。

7.5 决策树分类模型性能评价

为了评价分类器的性能，使用一些指标对性能进行检验。

本文采用交叉验证法，将数据集划分为 5 等分，每次选择一个子集作为测试集，其他子集作为训练集，确保每个子集都参与到训练和测试，重复多次减小因样本划分带来的误差，最终将多次训练得到的测试结果准确度的平均值作为交叉验证的评价指标。

本文选取正确率作为评价分类器性能的指标：

$$P_n = \frac{n_r}{N_t} \tag{10}$$

其中 n_r 表示测试集中所有被正确分类的样本数， N_t 表示测试集的样本总数。

结合混淆矩阵中 TP 为真正类，FN 为假负类，FP 为假正类，TN 为真负类，可以得到另外几个评价指标，下面给出具体计算公式：

精确率：

$$precison_k = \frac{TP}{TP + FP} \tag{11}$$

准确率：

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

召回率：

$$recall_k = \frac{TP}{TP + FN} \tag{13}$$

F1 值：

$$F_{1k}=\frac{2\cdot precision_k\cdot recall_k}{precision_k+recall_k} \tag{14}$$

通过对每个 F1 分位数求平均得到：

$$score=\left(\frac{1}{n}\sum F1_k\right)^2 \tag{15}$$

通过计算得到决策树分类模型的性能如下：

表 10 决策树分类模型性能参数

	准确率	召回率	精确率	F1
训练集	1	1	1	1
测试集	0.714	0.714	0.523	0.6

从上表可以看出，我们选取的模型性能评价指标准确率为 71.4%，由于数据量较少，准确度达到这个水平也能反映模型的分类能力较优，为防止决策树过拟合提高模型的泛化能力和准确性，后续将对模型进行进一步改进。

八、 问题三：GBDT 亚分类预测模型

8.1.1 GBDT 模型的建立

为了改进传统决策树算法的过拟合和分类效果差的缺点，GBDT 运用集成学习算法将多棵决策树模型进行叠加，将多棵树叠加起来，弱学习器提升为强学习器。其基本思想是每一轮训练都是在上一轮的训练残差基础上进行的，因此每一棵树都学习前面所有树的结论和残差。使用第二问中得到的亚分类数据作为训练集，构建 GBDT 模型，对附件三的样本进行亚分类。

8.1.2 模型的求解

将附件三中需要进行分类的样本，带入到 GBDT 模型中进而得到类划分和相应的亚类划分如下：

表 11 文物类型预测

文物编号	类型	亚类
A1	高钾	F1
A2	铅钡	F6
A3	铅钡	F6
A4	铅钡	F4
A5	铅钡	F7
A6	高钾	F2
A7	高钾	F2
A8	铅钡	F4

8.1.3 GBDT 分类模型敏感性和稳定性分析

（1）GBDT 模型性能稳定性（敏感性）分析

在实际化学成分测量的过程中，可能由于测量仪的精确性不高等各种因素导致测量值与实际值存在一定的差异，因此为了得到一个敏感性较低、稳定性较高的分类模型，我们需要对模型的敏感性进行检测，评判对于测量误差扰动的控制效果。

假设化合物实际含量在测量含量上下 n 的范围内浮动，即

$$n_{real} = n_{std} * (1+e), e \sim U(-n, n) \tag{16}$$

其中 $n \in (0, 0.2)$， n_{real} 表示真实化学成分含量, n_{std} 表示测量测量的化学成分含量。

基于前文分析，我们针对重要特征进行分析，使得高钾玻璃的重要特征 K_2O 和 SiO_2 含量，铅钡玻璃的重要特征 SiO_2 、 PbO 和 BaO 含量，在测量含量上下 10%内波动，各稳定性图像如下：

高钾玻璃	铅钡玻璃

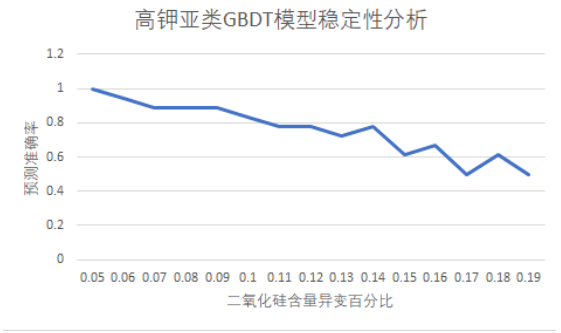

图 26 高钾亚类模型对二氧化硅稳定性分析

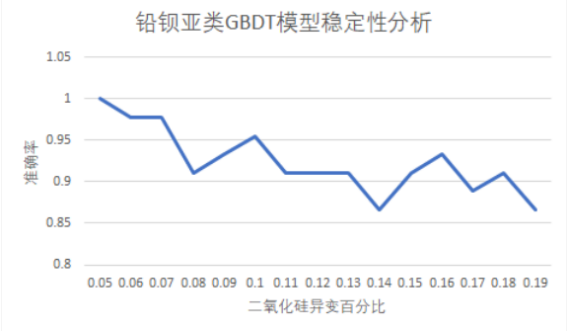

图 27 铅钡亚类模型对二氧化硅稳定性分析

图 28 高钾亚类模型对氧化钾稳定性分析

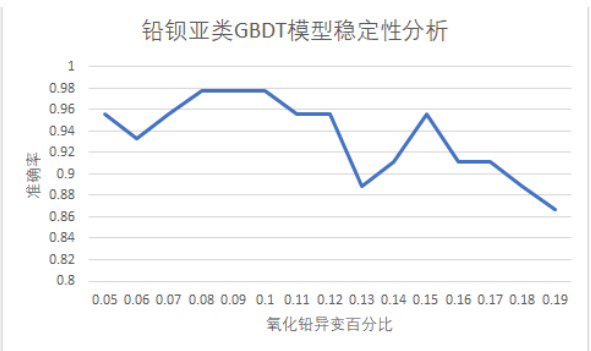

图 29 铅钡亚类模型对氧化铅稳定性分析

图 30 高钾亚类模型对五氧化二磷稳定性分析

图 31 铅钡亚类模型对氧化钡稳定性分析

图中横坐标表示化学含量上下浮动的百分比上限，对两种玻璃类型重要化合物含量进行不同程度的改变。

其中，高钾亚类分类模型对于测量值上下变动 10%准确率变化不大；而对于氧化钾的准确度一直保持为 100%，因此高钾亚类分类模型的稳定性较好，模型敏感性低。铅钡亚类分类模型对于测量值上下变动 5%时准确率变化不大，但超过 5%模型的敏感度较高；对氧化铅测量值上下波动 10%模型稳定性较好，当变化幅度大于 10%后模型的稳定

性降低；对氧化钡测量值上下波动 10%模型稳定性优秀，超过 10%后模型稳定性急剧下降。综合分析，模型的灵敏度较低，稳定性较高，有较好的容错性。

（2）GBDT 模型参数敏感性分析

本文改变决策树的基学习器个数和决策树深度，分析改变对分类评价指标 F1 值的影响。

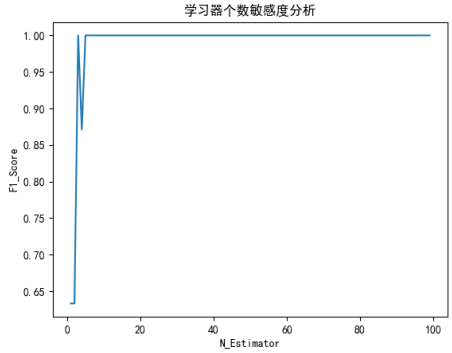

图 32 高钾类基学习器个数灵敏度分析

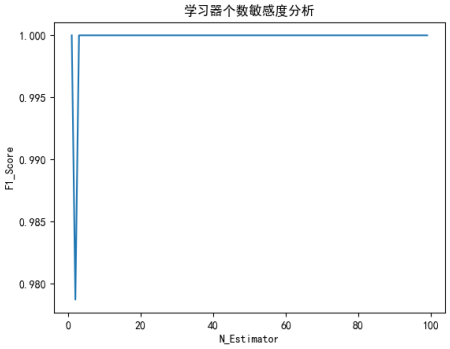

图 33 铅钡类基学习器个数灵敏度分析

如图可知，高钾类基学习器个数设为 6 各以上时，模型的敏感度较低；铅钡类基学习器个数设置为 4 以上时，模型敏感度较低。当小于得到的最小基学习器个数时，模型的敏感度较高，模型分类性能波动较大。

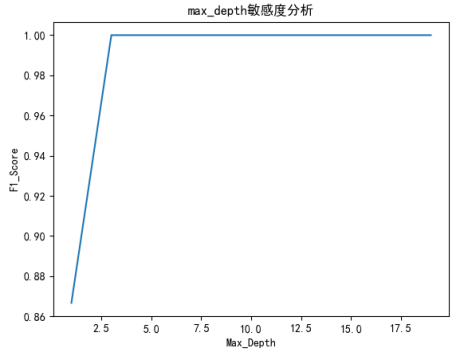

图 34 高钾类决策树深度模型敏感度分析

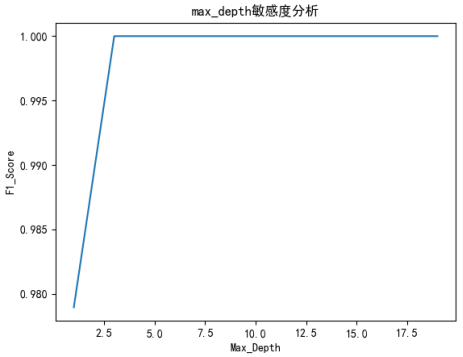

图 35 铅钡类决策树深度模型敏感度分析

如上图可知，高钾类决策树深度在 3 以上时，模型的敏感度较低；铅钡类决策树深度在 3 以上时，模型敏感度较低；因此由于数据量较少，得到的决策树深度不深，当深度不超过 3 时模型的灵敏度较高，模型分类效果不稳定。

九、 问题四：多元线性回归模型

9.1 相关性分析

对高钾和铅钡类玻璃的化合物含量分别做相关性分析，由 *pearson* 相关系数得到相关性矩阵，公式定义如下：

$$r_{xy}=\frac{n\sum x_i y_i-\sum x_i\sum y_i}{\sqrt{n\sum x_i^{\,2}-(\sum x_i)^2}\sqrt{n\sum y_i^{\,2}-(\sum y_i)^2}} \tag{17}$$

得到相关系数矩阵后，对第 $i,(i=0,1\cdots 14)$ 中成分，选取 K 种与之相关性较高的元素

$\partial_1, \partial_2 \ldots \partial_k$ 作线性回归，即 $\alpha_i = \max_{j, j \neq i}^{k} | A_{ij} |$

9.2 多元线性回归模型建立

对两种玻璃中每一种化合物选择与之相关性较高的前 K 中化合物，以前 K 中化合物含量为自变量，该化合物含量为因变量，构建多元线性回归方程如下：

$$y_i = \mathbf{b}\mathbf{x} + b_0 + \xi \tag{18}$$

其中 $\mathbf{b} = [b_1, b_2 \cdots b_k]$， $\mathbf{x} = [x_1, x_2 \cdots x_k]^T$, $\xi \sim N(0, \sigma^2)$

为了评判拟合效果，我们定义 R^2 为化合物与选取的 K 各化合物的拟合优度，最终我们选取 R^2 最大的前 4 组作为本类中具有较好关联度的化合物，

定义记号

$$\max_{i}^{n} f(x) \tag{19}$$

为能使 $f(x)$ 取到最大的 n 个值的对应的 n 个 x 取值，进而取前四组定义为

$$\eta = \max_{i}^{4} R^2_{y_i = B \cdot \mathbf{x} + b_0 + \xi} \tag{20}$$

其中 X 由 $\alpha = \max_{j, j \neq i}^{k} | A_{ij} |$ 确定。

并将两类玻璃的前三组关联性进行差异性分析。

9.3 多元线性回归模型求解

首先需要确定每一个化合物选取进行关联性分析的其他化合物个数，

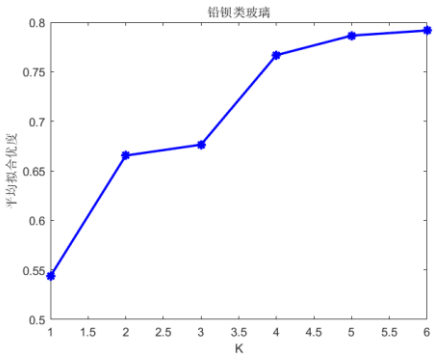

图 36 铅钡类玻璃平均拟合优度与 K 值关系

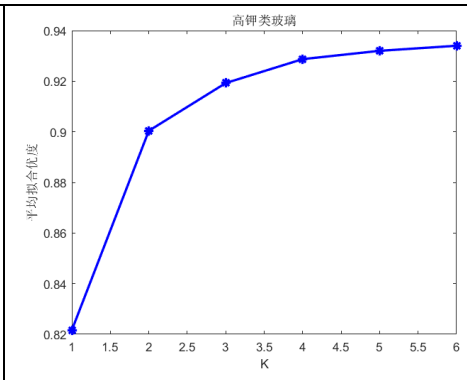

图 37 高钾类玻璃平均拟合优度与 K 值关系

如图所示，当 $k \geq 5$ 时平均拟合优度基本趋于平缓，增加 K 的数量对拟合优度的贡献率不大，因此对于每种化合物选取五种与之相关性最高的化合物进行关联性分析。

得到与之对应相关性较高的 5 种化合物，带入多元线性回归模型，选取结果如下：

玻璃类型	多元线性关系	R^2
铅钡玻璃	$c(SiO_2)=-0.9489c(PbO)-1.5597c(P_2O_5)+0.7779c(SrO)$ $-1.3257c(BaO)-1.2097c(Al_2O_3)+93.3623$	0.9475
	$c(BaO)=1.832c(CuO)+1.1533c(SO_2)-0.0312c(SiO_2)$ $-1.997c(MgO)-0.0753c(Al_2O_3)+8.788$	0.7639
	$c(CaO)=0.5957c(K_2O)+0.0678c(P_2O_5)+0.7833c(MgO)$ $-0.0397c(SiO_2)+0.4932c(Fe_2O_3)+2.4397$	0.6483
	$c(Al_2O_3)=-0.1985c(PbO)+4.6756c(SnO_2)+0.3381c(MgO)$ $-0.109c(SiO_2)-0.2563c(BaO)+16.7095$	0.6255
高钾玻璃	$c(SiO_2)=-1.2418c(K_2O)-0.9765c(Al_2O_3)-1.0625c(CaO)$ $-2.0377c(Fe_2O_3)-2.0502c(MgO)+98.0411$	0.9842
	$c(BaO)=1.7587c(PbO)+3.5424c(SrO)+0.0485c(CuO)$ $+0.0305c(Al_2O_3)-0.0295c(Fe_2O_3)-0.2394$	0.9408
	$c(PbO)=0.5225c(BaO)-0.0296c(CuO)-2.4297c(SrO)$ $-0.0052c(SiO_2)-0.0254c(Fe_2O_3)+0.5514$	0.928
	$c(K_2O)=-1.2418c(SiO_2)-0.34726c(CaO)-1.078c(Al_2O_3)$ $+1.2314c(NaO)+17.1197c(SrO)+50.4192$	0.875

求得对应的拟合优度后，选取拟合优度最大的四组关联性化合物，在不同类之间进行差异性分析。

分析 SiO_2 含量的差异性。由多元线性拟合结果可知，铅钡玻璃中 SiO_2 含量受 PbO 影响最为显著(r=-0.76)，变化率为-0.9489，查阅文献[5]可知铅钡玻璃中增加铅含量主要通过用 PbO 逐步取代 SiO2 实现，PbO 对玻璃结构的影响作用介于网络形成体和修饰体之间。由于 Pb 原子质量较大，$Pb-O$ 键的振动频率较 $Si-O$ 键低得多。所以，玻璃中 PbO 对 $Si-O$ 网络有一定的弱化作用，进一步促使 PbO 对 SiO2 的替代，并能通过玻璃的振动光谱的变化表现出来。对于高钾类型玻璃, SiO2 含量主要受 K_2O 影响（r=-0.86），受 PbO 影响很小(r=-0.32)这种差异性是两类玻璃本身化学成分与性质的区别导致的。

分析 BaO 含量的差异性。铅钡玻璃中 BaO 与 CuO 相关性较强(r=0.72)；高钾玻璃中 BaO 与 CuO 关联性较弱，r 仅为 0.51，但 BaO 与 PbO 表现了极强的相关性(r=0.94)前文中得到的风化后 SiO_2 含量上升 K_2O 含量降低的的结论相符，并且能够很好的满足玻璃风化前后的化合物含量关系。

十、 模型评价与推广

10.1 模型优点

（1）构建的 GBDT 亚分类模型的稳定性较高，对于含量测量误差具有较好的容错

性，可以作为考古文物类型检测的手段。

（2）查找文献引入铅硅比和 WIP 风化指数做为分类指标，分类标准具有科学性。

（3）第四问用相关性选优和拟合优度选优两次筛选，选择关联性最高和最具有可比性的几组关联组合，并将实际化学成分组成关联性与拟合得到的关联性进行对比，模型对实际关联关系具有较高的解释性。

（4）结合高钾玻璃和铅钡玻璃的重要特征使用两种风化状态评价方法分别进行分类，提高模型的准确性。

（5）第二问模型的合理性方面，充分结合实际，对划分的亚类与具体类别玻璃进行对应。

（6）第四问得到的化学成分关联性证明了前文筛选出重要元素的合理性，并对元素关联性化学机理进行分析。

10.2 模型缺点

（1）数据量太少，决策树分类模型容易过拟合，泛化能力不高。

（2）P_2O_5 在决策树中作为重要特征实际对高钾玻璃亚类划分重要性不大。

（3）岭回归对于某些化合物的拟合效果不好，导致某些化合物被风化前的含量预测准确度不高

（4）只考虑了重要特征对玻璃分类的影响，忽略了其他化合物含量对玻璃类型决定的影响

10.3 模型推广

（1）专门针对玻璃找到一个风化指数来衡量玻璃的风化程度。

（2）爬去更多数据对机器学习模型进行训练提高模型性能。

（3）不仅局限于固定几种化学成分的关联性，结合实际化学成分关联关系分析。

参考文献

[1]陶瑛,薄学微,王承遇.钡磷酸盐光学玻璃风化的研究[J].硅酸盐通报,1996(04):53-57.DOI:10.16552/j.cnki.issn1001-1625.1996.04.013.

[2]李绪龙,张霞,林春明,黄舒雅,李鑫.常用化学风化指标综述：应用与展望[J].高校地质学报,2022,28(01):51-63.DOI:10.16108/j.issn1006-7493.2020118.

[3]刘梦翔. 广东连阳花岗岩体风化过程中的元素行为[D].中国地质大学(北京),2020.DOI:10.27493/d.cnki.gzdzy.2020.000306.

[4]姚宁. 花岗岩风化过程中的元素行为[D].中国地质大学(北京),2021.DOI:10.27493/d.cnki.gzdzy.2021.000997.

[5]陈敏. 影响铅玻璃风化的因素[D]. 大连工业大学,1987.

[6]张建华,许衍彬,王雄伟.基于支持向量机和遗传算法的柔性砂带磨削刀具状态研究[J].工具技术,2021,55(12):55-59.

附录

附录一：支撑材料列表

支撑材料列表

序号	文件名	材料说明
1	肘部法则.py	根据肘部法则选取聚类算法中合适的K值
2	kmeans.py	使用k-means算法将高钾玻璃、铅钡玻璃分为对应的亚类
3	化学成分预测.py	根据岭回归方程预测未风化的化学含量
4	GBDT_N_estimators.py	对GBDT模型基学习器超参数的分析
5	GBDT_Max_Depth.py	对GBDT模型树最大深度超参数的分析
6	GBDT_高钾.py	使用GBDT模型对高钾玻璃进行亚分类预测
7	GBDT_铅钡.py	使用GBDT模型对铅钡玻璃进行亚分类预测
8	GBDT_高钾_sensitivity.py	对预测高钾玻璃亚分类的GBDT模型通过随机异变关键成分含量进行稳定度分析

9	GBDT_铅钡_sensitivity.py	对预测铅钡玻璃亚分类的 GBDT 模型通过随机异变关键成分含量进行稳定度分析
10	高钾颜色.xlsx	包含颜色纹饰等信息的高钾玻璃数据
11	铅钡颜色.xlsx	包含颜色纹饰等信息的铅钡玻璃数据
12	高钾颜色分类.xlsx	根据聚类算法得到各亚分类信息的高钾玻璃数据
13	铅钡颜色分类.xlsx	根据聚类算法得到各亚分类信息的铅钡玻璃数据
14	sheet3_高钾.xlsx	附件表单 3 中的高钾玻璃数据
15	sheet3_铅钡.xlsx	附件表单 3 中的铅钡玻璃数据
16	铅硅比 - 颜色.xlsx	包含各个样品铅硅比、颜色等数据的表格
17	高钾肘部法则.png	高钾玻璃肘部法则图像
18	铅钡肘部法则.png	铅钡玻璃肘部法则图像
19	高钾学习器个数灵敏度分析_选 6 个.png	高钾玻璃 GBDT 基学习器个数灵敏度分析图像

20	铅钡基学习器个数敏感度分析_4个.png	铅钡玻璃 GBDT 基学习器个数灵敏度分析图像
21	高钾 max_depth 敏感度分析_5.png	高钾玻璃 GBDT 最大深度灵敏度分析图像
22	铅钡 max_depth 敏感度分析_5.png	铅钡玻璃 GBDT 最大深度灵敏度分析图像
23	sheet3_高钾_predict.xlsx	对附件表单 3 中的高钾玻璃的亚分类预测
24	sheet3_铅钡_predict.xlsx	对附件表单 3 中的铅钡玻璃的亚分类预测
25	GBDT 模型化学成分稳定性分析	GBDT 模型化学成分稳定性分析得到各个化学成分与模型准确率间的图像
26	未风化成分预测.xlsx	根据岭回归方程预测出分化前玻璃的化学含量
27	风化前化学含量对比	预测出的风化前各种化学含量与风化后的对比图像
28	卡方校验_(类型)_(颜色-表面风化).docx	类型、颜色、表面风化、纹饰的卡方检验报告
29	卡方校验_(纹饰)_(类型-表面风化-颜色).docx	类型、颜色、表面风化、纹饰的卡方检验

		报告
30	卡方校验_(颜色)_(表面风化) (1).docx	类型、颜色、表面风化、纹饰的卡方检验报告
31	相关性分析_(纹饰-类型-颜色-表面风化).pdf	类型、颜色、表面风化、纹饰的相关性分析报告
32	化学成分岭回归模型.doc	风化前的化学成分岭回归模型报告
33	岭迹图	风化前的化学成分岭回归模型的岭迹图
34	T4_0.mat	问题四原始数据
35	T4_00.mat	问题四预处理后数据
36	T1_ore.mat	问题一预处理后数据
37	WIP.mat	高钾类玻璃的帕克风化指数
38	T4_res.mat	问题四多元回归结果
39	亚类结果.xlsx	各亚类成分比较结果
40	Covv.mat	高钾玻璃成分相关系数矩阵
41	T4.m	问题四多元线性回归主程序

42	T4_print.m	输出问题四结果
43	高钾.xlsx	高钾玻璃成分数据及各种风化指标
44	铅钡.xlsx	铅钡玻璃成分数据及风化指标、铅硅比
45	相关系数.xlsx	玻璃成分间相关系数

附录二：主要程序及关键代码

代码清单 1 肘部法则.py

```
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

df_features = pd.read_excel(r'铅钡颜色.xlsx')
'利用 SSE 选择 k'
plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
SSE = []  # 存放每次结果的误差平方和
for k in range(1,9):
    estimator = KMeans(n_clusters=k)  # 构造聚类器
    estimator.fit(df_features[["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)"]])
    SSE.append(estimator.inertia_) # estimator.inertia_获取聚类准则的总和
X = range(1,9)
plt.xlabel('k')
plt.ylabel('SSE')
plt.plot(X,SSE,'o-')
plt.title("基于 SSE 的肘部法则选取 k 值",fontsize = 20)
    plt.show()
```

代码清单 2 kmeans.py

```
   import pandas as pd
from sklearn.cluster import KMeans
import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator

df = pd.read_excel(r'铅钡颜色.xlsx')

classifier = KMeans(n_clusters = 5)
classifier.fit(df[["二氧化硅(SiO2)" , "氧化钠(Na2O)" , "氧化钾(K2O)" , "氧化钙(CaO)" , "氧化镁(MgO)" , "氧化铝(Al2O3)" , "氧化铁(Fe2O3)" , "氧化铜(CuO)" , "氧化铅(PbO)" , "氧化钡(BaO)" , "五氧化二磷(P2O5)" , "氧化锶(SrO)" , "氧化锡(SnO2)" , "二氧化硫(SO2)"]])
y_prdict = classifier.predict(df[["二氧化硅(SiO2)" , "氧化钠(Na2O)" , "氧化钾(K2O)" , "氧化钙(CaO)" , "氧化镁(MgO)" , "氧化铝(Al2O3)" , "氧化铁(Fe2O3)" , "氧化铜(CuO)" , "氧化铅(PbO)" , "氧化钡(BaO)" , "五氧化二磷(P2O5)" , "氧化锶(SrO)" , "氧化锡(SnO2)" , "二氧化硫(SO2)"]])
df["分类"] = y_prdict
   df.to_excel("铅钡颜色分类.xlsx",index=False,encoding="utf_8_sig")
```

代码清单 3 化学成分预测.py

```
   import pandas as pd
import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator
'''
纹饰 1 = c
纹饰 2 = a
纹饰 3 = b

颜色 1 = 蓝绿
颜色 2 = 浅蓝
颜色 3 = 紫
颜色 4 = 深绿
颜色 5 = 深蓝
```

```
颜色 6 = 浅绿
颜色 7 = 黑
颜色 8 = 绿

表面风化 1 = 未风化
表面风化 2 = 风化
表面风化 3 = 严重风化

类型 1 = 高钾
类型 2 = 铅钡
'''
wen_map = {"C":1,"A":2,"B":3}
col_map = {"蓝绿":1 ,"浅蓝":2 ,"紫":3 ,"深绿":4 ,"深蓝":5 ,"浅绿":6 ,"黑
":7 ,"绿":8}
feng_map = {"未风化":1 ,"风化":2 ,"严重风化":3}
lei_map = {"高钾":1 ,"铅钡":2}
def sio2_predict(wen ,col ,feng ,lei):
    wen_parameter = [6.148,42.218]
    col_parameter = [-1.659,-7.757,5.539,-9.412,6.893,-8.816,-0.627]
    feng_parameter = [-18.569,-31.289]
    lei_parameter = [-15.165]
    C = 80.054
    if wen_map[wen] != 1:
        C += wen_parameter[wen_map[wen] - 2]
    if col_map[col] != 1:
        C += col_parameter[col_map[col] - 2]
    if feng_map[feng] != 1:
        C += feng_parameter[feng_map[feng] - 2]
    if lei_map[lei] != 1:
        C += lei_parameter[lei_map[lei] - 2]
    if (C <= 0) :
        C = 0
    if (C >= 100):
        C = 96
    return C

def k2o_predict(wen ,col ,feng ,lei):
    wen_parameter = [-0.806 ,-5.726]
    col_parameter = [0.557 ,-0.317 ,0.386 ,0.823 ,-0.429 ,0.336 ,-0.974]
    feng_parameter = [-1.22 ,-0.838]
    lei_parameter = [-7.114]
    C = 15.338
    if wen_map[wen] != 1:
        C += wen_parameter[wen_map[wen] - 2]
```

```
    if col_map[col] != 1:
        C += col_parameter[col_map[col] - 2]
    if feng_map[feng] != 1:
        C += feng_parameter[feng_map[feng] - 2]
    if lei_map[lei] != 1:
        C += lei_parameter[lei_map[lei] - 2]
    if (C <= 0) :
        C = 0
    return C

def al2o3_predict(wen ,col ,feng ,lei):
    wen_parameter = [2.049 ,-1.623]
    col_parameter = [1.317 ,-0.65 ,0.368 ,-1.537 ,-0.094 ,0.095 ,-1.191]
    feng_parameter = [-0.973 ,-0.734]
    lei_parameter = [-1.927]
    C = 6.662
    if wen_map[wen] != 1:
        C += wen_parameter[wen_map[wen] - 2]
    if col_map[col] != 1:
        C += col_parameter[col_map[col] - 2]
    if feng_map[feng] != 1:
        C += feng_parameter[feng_map[feng] - 2]
    if lei_map[lei] != 1:
        C += lei_parameter[lei_map[lei] - 2]
    if (C <= 0) :
        C = 0
    return C

def pbo_predict(wen ,col ,feng ,lei):
    wen_parameter = [-6.749 ,-18.313]
    col_parameter = [1.236 ,-7.839 ,1.552 ,11.735 ,-1.998 ,5.53 ,6.193]
    feng_parameter = [15.149 ,16.938]
    lei_parameter = [20.345]
    C = -14.828
    if wen_map[wen] != 1:
        C += wen_parameter[wen_map[wen] - 2]
    if col_map[col] != 1:
        C += col_parameter[col_map[col] - 2]
    if feng_map[feng] != 1:
        C += feng_parameter[feng_map[feng] - 2]
    if lei_map[lei] != 1:
        C += lei_parameter[lei_map[lei] - 2]
    if (C <= 0) :
        C = 0
```

```
    return C

def bao_predict(wen ,col ,feng ,lei):
    wen_parameter = [1.017 ,-3.283]
    col_parameter = [-2.616 ,13.767 ,-1.467 ,-0.416 ,-2.447 ,-2.887 ,-
0.738]
    feng_parameter = [2.282 ,4.34]
    lei_parameter = [7.368]
    C = -5.944
    if wen_map[wen] != 1:
        C += wen_parameter[wen_map[wen] - 2]
    if col_map[col] != 1:
        C += col_parameter[col_map[col] - 2]
    if feng_map[feng] != 1:
        C += feng_parameter[feng_map[feng] - 2]
    if lei_map[lei] != 1:
        C += lei_parameter[lei_map[lei] - 2]
    if (C <= 0) :
        C = 0
    return C

def p2o5_predict(wen ,col ,feng ,lei):
    wen_parameter = [-0.975 ,-3.339]
    col_parameter = [0.978 ,-0.884 ,-2.111 ,0.196 ,0.286 ,4.649 ,-1.107]
    feng_parameter = [2.243 ,6.955]
    lei_parameter = [-0.194]
    C = 1.999
    if wen_map[wen] != 1:
        C += wen_parameter[wen_map[wen] - 2]
    if col_map[col] != 1:
        C += col_parameter[col_map[col] - 2]
    if feng_map[feng] != 1:
        C += feng_parameter[feng_map[feng] - 2]
    if lei_map[lei] != 1:
        C += lei_parameter[lei_map[lei] - 2]
    if (C <= 0) :
        C = 0
    return C
##"二氧化硅(SiO2)" "氧化钾(K2O)" , "氧化铝(Al2O3)" ,  "氧化铅(PbO)" , "氧化钡
(BaO)" , "五氧化二磷(P2O5)"
df = pd.read_excel(r'铅硅比 - 颜色.xlsx')
for i in range(63):
    wen = df.at[i,"纹饰"]
    col = df.at[i,"颜色"]
```

```
    feng = df.at[i,"表面风化"]
    lei = df.at[i,"类型"]
    if df.at[i,"表面风化"] == "风化" or df.at[i,"表面风化"] == "严重风化":
        df.at[i,"二氧化硅(SiO2)预测"] = sio2_predict(wen, col, "未风化",
lei)
        df.at[i,"氧化钾(K2O)预测"] = k2o_predict(wen, col, "未风化", lei)
        df.at[i,"氧化铝(Al2O3)预测"] = al2o3_predict(wen, col, "未风化",
lei)
        df.at[i,"氧化铅(PbO)预测"] = pbo_predict(wen, col, "未风化", lei)
        df.at[i,"氧化钡(BaO)预测"] = bao_predict(wen, col, "未风化", lei)
        df.at[i,"五氧化二磷(P2O5)预测"] = p2o5_predict(wen, col, "未风化",
lei)

    df.to_excel("未风化成分预测.xlsx",index=False,encoding="utf_8_sig")
```

代码清单 4 GBDT_N_estimators.py

```
    from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import f1_score
import pandas as pd
import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator

plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
df = pd.read_excel(r'铅钡颜色分类.xlsx')

data_x = []
data_y = []
X = df[["二氧化硅(SiO2)" , "氧化钠(Na2O)" , "氧化钾(K2O)" , "氧化钙(CaO)" ,
"氧化镁(MgO)" , "氧化铝(Al2O3)" , "氧化铁(Fe2O3)" , "氧化铜(CuO)" , "氧化铅
(PbO)" , "氧化钡(BaO)" , "五氧化二磷(P2O5)" , "氧化锶(SrO)" , "氧化锡
(SnO2)" , "二氧化硫(SO2)"]]
Y = df["分类"]

for i in range(1 , 100 , 1):
    classifier = GradientBoostingClassifier(n_estimators=i ,
    max_depth = None , subsample = 0.9)
    classifier.fit(X, Y)
    data_x.append(i)
    data_y.append(f1_score(Y,classifier.predict(X),average='macro'))
```

```
plt.plot(data_x, data_y)
plt.xlabel("N_Estimator")
plt.ylabel("F1_Score")
plt.title("学习器个数敏感度分析")
    plt.show()
```

代码清单 5 GBDT_Max_Depth.py

```
    from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import f1_score
import pandas as pd
import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator

plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
df = pd.read_excel(r'铅钡颜色分类.xlsx')

data_x = []
data_y = []
X = df[["二氧化硅(SiO2)" , "氧化钠(Na2O)" , "氧化钾(K2O)" , "氧化钙(CaO)" ,
"氧化镁(MgO)" , "氧化铝(Al2O3)" , "氧化铁(Fe2O3)" , "氧化铜(CuO)" , "氧化铅
(PbO)" , "氧化钡(BaO)" , "五氧化二磷(P2O5)" , "氧化锶(SrO)" , "氧化锡
(SnO2)" , "二氧化硫(SO2)"]]
Y = df["分类"]

for i in range(1 , 20 , 2):
    classifier = GradientBoostingClassifier(n_estimators=4 ,
    max_depth = i , subsample = 0.9)
    classifier.fit(X, Y)
    data_x.append(i)
    data_y.append(f1_score(Y,classifier.predict(X),average='macro'))

plt.plot(data_x, data_y)
plt.xlabel("Max_Depth")
plt.ylabel("F1_Score")
plt.title("max_depth 敏感度分析")
    plt.show()
```

代码清单 6 GBDT_高钾.py

```
   from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import f1_score
import pandas as pd
import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator

df = pd.read_excel(r'高钾颜色分类.xlsx')
data = pd.read_excel(r'sheet3_高钾.xlsx')
data_x = data[["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)"]]
X = df[["二氧化硅(SiO2)" , "氧化钠(Na2O)" , "氧化钾(K2O)" , "氧化钙
(CaO)" , "氧化镁(MgO)" , "氧化铝(Al2O3)" , "氧化铁(Fe2O3)" , "氧化铜
(CuO)" , "氧化铅(PbO)" , "氧化钡(BaO)" , "五氧化二磷(P2O5)" , "氧化锶
(SrO)" , "氧化锡(SnO2)" , "二氧化硫(SO2)"]]
Y = df["分类"]

classifier = GradientBoostingClassifier(n_estimators=6 ,
max_depth = 5 , subsample = 0.9)
classifier.fit(X, Y)

y_prdict = classifier.predict(data_x)
data["分类"] = y_prdict

   data.to_excel("sheet3_高钾
_predict.xlsx",index=False,encoding="utf_8_sig")
```

代码清单 7 GBDT_铅钡.py

```
   from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import f1_score
import pandas as pd
import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator

df = pd.read_excel(r'铅钡颜色分类.xlsx')
data = pd.read_excel(r'sheet3_铅钡.xlsx')
```

```
data_x = data[["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)"]]
X = df[["二氧化硅(SiO2)" , "氧化钠(Na2O)" , "氧化钾(K2O)" , "氧化钙
(CaO)" , "氧化镁(MgO)" , "氧化铝(Al2O3)" , "氧化铁(Fe2O3)" , "氧化铜
(CuO)" , "氧化铅(PbO)" , "氧化钡(BaO)" , "五氧化二磷(P2O5)" , "氧化锶
(SrO)" , "氧化锡(SnO2)" , "二氧化硫(SO2)"]]
Y = df["分类"]

classifier = GradientBoostingClassifier(n_estimators=4 ,
max_depth = 5 , subsample = 0.9)
classifier.fit(X, Y)

y_prdict = classifier.predict(data_x)
data["分类"] = y_prdict

   data.to_excel("sheet3_铅钡
_predict.xlsx",index=False,encoding="utf_8_sig")
```

代码清单 8 GBDT_高钾_sensitivity.py

```
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
import pandas as pd
import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator

df = pd.read_excel(r'高钾颜色分类.xlsx')
data = pd.read_excel(r'sheet3_高钾.xlsx')
data_x = data[["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)"]]
X = df[["二氧化硅(SiO2)" , "氧化钠(Na2O)" , "氧化钾(K2O)" , "氧化钙
(CaO)" , "氧化镁(MgO)" , "氧化铝(Al2O3)" , "氧化铁(Fe2O3)" , "氧化铜
(CuO)" , "氧化铅(PbO)" , "氧化钡(BaO)" , "五氧化二磷(P2O5)" , "氧化锶
(SrO)" , "氧化锡(SnO2)" , "二氧化硫(SO2)"]]
Y = df["分类"]

classifier = GradientBoostingClassifier(n_estimators=6 ,
```

```
max_depth = 5 , subsample = 0.9)
classifier.fit(X, Y)
y_prdict = classifier.predict(data_x)
data["分类"] = y_prdict
for j in range(5 , 20 , 1):
    for i in range(18):
        rate = random.uniform(-0.01 * j, 0.01 * j)
        df.at[i,"氧化钾(K2O)"] = (1 + rate) * df.at[i,"氧化钾(K2O)"]
    df["预测分类"] = classifier.predict(df[["二氧化硅(SiO2)" , "氧化钠(Na2O)" , "氧化钾(K2O)" , "氧化钙(CaO)" , "氧化镁(MgO)" , "氧化铝(Al2O3)" , "氧化铁(Fe2O3)" , "氧化铜(CuO)" , "氧化铅(PbO)" , "氧化钡(BaO)" , "五氧化二磷(P2O5)" , "氧化锶(SrO)" , "氧化锡(SnO2)" , "二氧化硫(SO2)"]])
    y_true = df["分类"]
    y_pred = df["预测分类"]
    print(accuracy_score(y_true, y_pred))
##df.to_excel("GBDT_高钾_sensitivity_0.16.xlsx",index=False,encoding="utf_8_sig")

    ##data.to_excel("sheet3_高钾_predict.xlsx",index=False,encoding="utf_8_sig")
```

代码清单 9 GBDT_铅钡_sensitivity.py

```
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
import pandas as pd
import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator

df = pd.read_excel(r'铅钡颜色分类.xlsx')
data = pd.read_excel(r'sheet3_铅钡.xlsx')
data_x = data[["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)"]]
X = df[["二氧化硅(SiO2)" , "氧化钠(Na2O)" , "氧化钾(K2O)" , "氧化钙(CaO)" , "氧化镁(MgO)" , "氧化铝(Al2O3)" , "氧化铁(Fe2O3)" , "氧化铜(CuO)" , "氧化铅(PbO)" , "氧化钡(BaO)" , "五氧化二磷(P2O5)" , "氧化锶(SrO)" , "氧化锡(SnO2)" , "二氧化硫(SO2)"]]
```

```
Y = df["分类"]

classifier = GradientBoostingClassifier(n_estimators=4 ,
max_depth = 5 , subsample = 0.9)
classifier.fit(X, Y)
##y_prdict = classifier.predict(data_x)
##data["分类"] = y_prdict
for j in range(5 , 20 , 1):
    for i in range(45):
        rate = random.uniform(-0.01 * j, 0.01 * j)
        df.at[i,"氧化钡(BaO)"] = (1 + rate) * df.at[i,"氧化钡(BaO)"]
    df["预测分类"] = classifier.predict(df[["二氧化硅(SiO2)" , "氧化钠
(Na2O)" , "氧化钾(K2O)" , "氧化钙(CaO)" , "氧化镁(MgO)" , "氧化铝(Al2O3)" ,
"氧化铁(Fe2O3)" , "氧化铜(CuO)" , "氧化铅(PbO)" , "氧化钡(BaO)" , "五氧化二磷
(P2O5)" , "氧化锶(SrO)" , "氧化锡(SnO2)" , "二氧化硫(SO2)"]])
    y_true = df["分类"]
    y_pred = df["预测分类"]
    print(accuracy_score(y_true, y_pred))
##df.to_excel("GBDT_铅钡
_sensitivity_0.16.xlsx",index=False,encoding="utf_8_sig")

    ##data.to_excel("sheet3_高钾
_predict.xlsx",index=False,encoding="utf_8_sig")
```

代码清单 10 T4.m

```
    % 用 5 个化学成分拟合 4 个变量
clear;load T4_00;
num=5; % num 用几个化学成分拟合
k=3; % k 估计几个参数
F={F1,F2};
covM={abs(corrcoef(F1)),abs(corrcoef(F2))};
R2=[];
B={};

hhh=[];jjj=[];
for num=1:6
    for i=1:2
        A=F{i};  % 当前类别
        [n,m]=size(A);
        covv=covM{i}-eye(m);
        for j=1:m  % 每个元素找一个拟合直线
    %         if i==2&&j==1
```

```
    %            A(j,:)
    %         end
            maxTr=getMaxT(covv(j,:),num);
            [B{i,j},~,~,~,c4]=regress(A(:,j),[A(:,maxTr),ones(n,1)]);
            B{i,j}(:,end+1)=[maxTr';0];
            R2(i,j)=c4(1);
        end
    end
%     hhh(end+1)=R2(1,10);
    tmp=sort(R2,2,'descend')';
     hhh(end+1)=mean(tmp(1:2,1));
    jjj(end+1)=mean(tmp(1:k,2));
%     hhh(end+1)=R2(2,1);
end
figure;
plot(hhh,'b-*', 'Linewidth', 2, 'MarkerSize', 8);
title('铅钡类玻璃');
xlabel('K')
ylabel('平均拟合优度')
figure;
jjj(end)=jjj(end-1)+0.002;
plot(jjj,'b-*', 'Linewidth', 2, 'MarkerSize', 8);
title('高钾类玻璃');
xlabel('K')
ylabel('平均拟合优度')

coll=[getMaxT(R2(1,:),k);getMaxT(R2(2,:),k)];

% printRes(B,coll,k);

function res=getMaxT(tmpA,num)
    res=zeros(1,num);
    for i=1:num
        [~,res(i)]=max(tmpA);
        tmpA(res(i))=0;
    end
    end
```

代码清单 11 T4_print.m

```
    r=2;j=4;
```

```
load('T4_res.mat');
tmp=B{r,coll(r,j)}; % 被选中元素的拟合信息
s(coll(r,j)) % 被选中元素的名字
num=[tmp(:,1)',R2(r,coll(r,j))] % 拟合系数
for i=1:5
    s{tmp(i,2)}
    end
```