中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生数学建模竞赛

学　　校	西安电子科技大学
参赛队号	20107010013
队员姓名	1.刘伟
	2.邢继玮
	3.高尚

中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生
数学建模竞赛

题 目 **降低汽油精制过程中的辛烷值损失模型**

摘 要：

汽油清洁化研究对大气环境有重要影响，而辛烷值（以 RON 表示）是反映汽油燃烧性能的最重要指标。其中重点是降低汽油中的硫、烯烃含量，同时尽量保持其辛烷值。

本文针对以上问题，完成了数据预处理，通过特征选择来寻找该问题的主要建模变量，结合这些变量和 AdaBoost 回归算法建立了辛烷值损失预测模型并完成了模型验证，最后使用该模型进行实际操作方案变量的优化，和模型可视化展示。

针对问题一：问题一是一个数据处理问题，采用**特定的数据整定方法**，对待处理的 2 个 40 组数据进行预处理。本文依据以下准则：删除残缺位点、删除样本中数据全部为空位点、用前后 2 小时数据均值填补空值位点、通过限幅方法剔除部分范围外样本、根据拉依达准则去除异常值，完成了数据预处理，并将其加入附件一中相应的样本号中。

针对问题二：问题二是一个特征选择问题，要求从 367 个变量中进行筛选，降维至 30 个以下作为主要变量。本文首先通过计算**缺失值比率**验证了数据的完整性；接着使用**低方差滤波**去除方差较小的特征变量，根据 0.01 的阈值筛去了 35 个特征变量；然后根据**高相关滤波**去除相关系数大于 0.6 的特征变量后，剩余 50 个变量；最后使用**随机森林**的特征重要性筛选机制得到最重要的 30 个特征变量。

针对问题三：问题三是一个回归预测问题，寻找问题二筛选得到的 30 个主要变量与辛烷值(RON)损失值之间的关系。本文采用了**基于决策回归树的 AdaBoost 回归模型**，在特征选择后的样本上完成模型的训练，以得到辛烷值损失值的预测，完成了训练拟合效果展示；然后本文对模型进行验证，得到 AdaBoost 算法测试得分为 0.82。除此之外，本文绘制了数据集 K-折交叉验证的 MSE 曲线，以上验证体现了该模型的合理性和鲁棒性。

针对问题四：问题四是一个有约束条件的优化问题，在操作变量和产品硫含量的约束条件下，寻找满足辛烷值损失降幅大于 30%的操作条件。本文充分利用主要特征重要性优势来**缩小搜索空间**，保证满足问题约束条件的情况下，对缩减后的空间进行**迭代求解**，得到一定范围内的 325 个样本操作变量的优化结果。

针对问题五：问题五在问题四的基础上，需要先对产品中的硫元素进行估计，再对操作变量进行逐步优化，最后将模型可视化展示，分别绘制汽油辛烷值和产品中硫含量的变化轨迹。本文首先绘制了原料和产品中硫含量的**箱型分析图**，得到了硫含量的大致分布区间，约为 3.2μg/g。再根据**贪婪算法**，对于每个主要操作变量，在保持除该操作变量外其余变量不变的前期下，计算该变量增加和减少每次允许调制幅度值时，辛烷值损失的预测值，选取辛烷值损失降幅最大的变量所对应的操作步骤并执行。最终绘制了主要操作变量优化过程中汽油辛烷值的变化轨迹图。

关键词：数据整定 特征选择 AdaBoost 迭代求解 箱型分析 贪婪算法

目 录

一、问题重述

1.1 问题背景

汽油作为小型车辆的主要燃料，而汽油清洁化研究对大气环境有重要影响，其中重点是降低汽油中的硫、烯烃含量，同时尽量保持其辛烷值。

汽油生产所用到的原油中的重油通常占比 40-60%，而重油中以硫为代表的杂质含量高，难以直接利用。以催化裂化为核心的重油轻质化工艺技术，将重油转化为汽油、柴油和低碳烯烃，超过 70%的汽油是由催化裂化生产得到，因此成品汽油中 95%以上的硫和烯烃来自催化裂化汽油。故必须对催化裂化汽油进行精制处理，以满足对汽油质量要求。

辛烷值（以 RON 表示）是反映汽油燃烧性能的最重要指标。现有技术在对催化裂化汽油进行脱硫和降烯烃过程中，普遍降低了汽油辛烷值。

化工过程的建模一般是通过数据关联或机理建模的方法来实现的，取得了一定的成果。但是由于炼油工艺过程的复杂性以及设备的多样性，它们的操作变量（控制变量）之间具有高度非线性和相互强耦联的关系，而且传统的数据关联模型中变量相对较少、机理建模对原料的分析要求较高，对过程优化的响应不及时，所以效果并不理想。

某石化企业的催化裂化汽油精制脱硫装置运行 4 年，积累了大量历史数据，其汽油产品辛烷值损失平均为 1.37 个单位，而同类装置的最小损失值只有 0.6 个单位。故有较大的优化空间。

1.2 需要解决的问题

依据从催化裂化汽油精制装置采集的 325 个数据样本（每个数据样本都有 354 个操作变量），通过数据挖掘技术来建立汽油辛烷值（RON）损失的预测模型，并给出每个样本的优化操作条件，在保证汽油产品脱硫效果（欧六和国六标准均为不大于 10μg/g，但为了给企业装置操作留有空间，本次建模要求产品硫含量不大于 5μg/g）的前提下，尽量降低汽油辛烷值损失在 30%以上。

问题 1：数据处理：请参考近 4 年的工业数据(见附件一“325 个数据样本数据.xlsx”)的预处理结果，依“样本确定方法”（附件二）对 285 号和 313 号数据样本进行预处理（原始数据见附件三“285 号和 313 号样本原始数据.xlsx”）并将处理后的数据分别加入到附件一中相应的样本号中，供下面研究使用。

问题 2：寻找建模主要变量：由于催化裂化汽油精制过程是连续的，虽然操作变量每 3 分钟就采样一次，但辛烷值（因变量）的测量比较麻烦，一周仅 2 次无法对应。但根据实际情况可以认为辛烷值的测量值是测量时刻前两小时内操作变量的综合效果，因此预处理中取操作变量两小时内的平均值与辛烷值的测量值对应。这样产生了 325 个样本（见附件一）。

建立降低辛烷值损失模型涉及包括 7 个原料性质、2 个待生吸附剂性质、2 个再生吸附剂性质、2 个产品性质等变量以及另外 354 个操作变量（共计 367 个变量），工程技术应用中经常使用先降维后建模的方法，这有利于忽略次要因素，发现并分析影响模型的主要变量与因素。因此，请你们根据提供的 325 个样本数据（见附件一），通过降维的方法从 367 个操作变量中筛选出建模主要变量，使之尽可能具有代表性、独立性（为了工程应用方便，建议降维后的主要变量在 30 个以下），并请详细说明建模主要变量的筛选过程及其合理性。（提示：请考虑将原料的辛烷值作为建模变量之一）。

问题 3：建立辛烷值（RON）损失预测模型：采用上述样本和建模主要变量，通过数

据挖掘技术建立辛烷值（RON）损失预测模型，并进行模型验证。

问题 4：主要变量操作方案的优化：要求在保证产品硫含量不大于 5μg/g 的前提下，利用你们的模型获得 325 个数据样本（见附件一“325 个数据样本数据.xlsx”）中，辛烷值（RON）损失降幅大于 30%的样本对应的主要变量优化后的操作条件（优化过程中原料、待生吸附剂、再生吸附剂的性质保持不变，以它们在样本中的数据为准）。

问题 5：模型的可视化展示：工业装置为了平稳生产，优化后的主要操作变量（即：问题 2 中的主要变量）往往只能逐步调整到位，请你们对 133 号样本（原料性质、待生吸附剂和再生吸附剂的性质数据保持不变，以样本中的数据为准），以图形展示其主要操作变量优化调整过程中对应的汽油辛烷值和硫含量的变化轨迹。（各主要操作变量每次允许调整幅度值Δ见附件四“354 个操作变量信息.xlsx”）。

二、模型假设

假设 1：题目中所提供的工厂采集到的样本数据都真实有效；
假设 2：题目中所提供的样本足够用于模型训练和检验；
假设 3：变量间的关系与样本测量时间无关。

三、符号说明

符号	符号说明
σ	贝塞尔公式计算出的标准差
v	剩余误差，i=1，2，...，n
μ	均值
u^*	归一化后的结果
$S_{x_k}^2$	低方差滤波中的方差值
r	相关系数
D	训练集
ω	训练样本的权值
h_t	第 t 个决策回归树
E_t	训练集上的样本最大误差
e_{ti}	每个样本的相对误差

ε_t	回归误差率
α_t	回归树 h_t 的权重系数
Z_t	规范化因子
$H(x)$	强学习器模型
$f(x)$	训练模型
$x^0(m)$	第 m 个样本主要变量的初始值
$x^k(m)$	第 m 个样本迭代第 k 步时主要变量的取值
Δ	各主要操作变量每次允许调整幅度值

四、模型的建立与求解

4.1 问题 1 的分析与求解

问题一主要目标是数据与处理操作，通过“样本确定方法”（附件二）对附件三提供的原始数据进行数据重整，处理结果填充到附件一相应的样本号中，供后续问题研究使用。

4.1.1 问题分析

附件二提供了数据采集来源信息、数据整定处理操作，以及样本确定方法，附件三中的 285 号和 313 号数据样本作为待处理的原始数据，进行预处理。每个原始数据样本包括 2 小时内共计 40 个时间点的采集数据组，其中每个数据组包括 354 个操作位点。预处理结果即为 285 号和 313 号数据样本的 354 个操作位点数据，并将处理后的数据分别加入到附件一中相应的样本号中，供后续研究使用。

为了得到 285 号和 313 号数据样本的数据预处理结果，需要对每个数据样本的 40 个测量原始数据进行数据整定，具体操作包括如下：

（1）删除残缺位点；

（2）删除 325 个样本中数据全部为空值的位点；

（3）用前后 2 小时数据均值填补空值位点；

（4）总结变量操作范围，通过限幅方法剔除部分范围外样本；

（5）根据拉依达准则（3σ准则）去除异常值。

通过 MATLAB 编程处理，可以完成无效数据的剔除，并填补错误数据。

4.1.2 数据处理

首先通过 MATLAB 编程，找出附件三中 285 号和 313 号数据样本残缺位点和 325 个样本中数据全部为空值的位点，并进行进行删除。根据原始数据变量的操作范围，然后采用最大最小的限幅方法剔除一部分不在此范围的样本。

对于异常值处理，根据拉依达准则（3σ准则），设对被测量变量进行等精度测量，得

到 a_1，a_2，...，a_n，算出其算术平均值 x 及剩余误差 $v_i=a_i - a(i=1，2，...，n)$，并按贝塞尔公式算出标准误差σ，若某个测量值 a_b 的剩余误差 $v_b(1\le b\le n)$，满足$|v_b|=|a_b-a| > 3\sigma$，则认为 a_b 是含有粗大误差值的坏值，应予剔除。贝塞尔公式如式 4-1 所示：

$$\sigma=[\frac{1}{n-1}\sum_{i=1}^{n}v_i^2]^{1/2}=\{[\sum_{i=1}^{n}a_i^2-(\sum_{i=1}^{n}a_i)^2/n]/(n-1)\}^{1/2} \tag{4-1}$$

在正态分布中σ代表标准差，μ代表均值。$a=\mu$即为图像的对称轴，3σ原则所规定数值分布与数据占比对应表，如表 4.1 所示。

表 4.1　3σ原则下数值分布与占比对应表

数值分布	在数据中的占比
$(\mu-\sigma, \mu+\sigma)$	0.6826
$(\mu-2\sigma, \mu+2\sigma)$	0.9544
$(\mu-3\sigma, \mu+3\sigma)$	0.9974

可以认为，数据取值几乎全部集中在（$\mu-3\sigma, \mu+3\sigma$)区间内，超出这个范围的可能性仅占不到 0.3%。

附件三中包含 285 号和 313 号两组数据样本，每组数据包含 2 个小时内采集的 40 次数据组，每个数据组包含 354 个操作位点，即 354*40，附件三中第一列是采集时间信息，随后的各个列为各操作位点信息，即每一列为一个变量，剔除每一列中的异常值。

题目基本目标为降低装置产品辛烷值损失，故确定样本的主要依据为样品的辛烷值数据。由于辛烷值的测定数据相对于操作变量数据而言相对较少，而且辛烷值的测定往往滞后，因此确定某个样本的方法为：以辛烷值数据测定的时间点为基准时间，取其前 2 个小时的操作变量数据的平均值作为对应辛烷值的操作变量数据,并将 285 号和 313 号两组数据样本最后的预处理结果分别加入到附件一中相应的样本号中。

4.2 问题 2 的分析与求解

问题二目标是寻找建模主要变量,即将原题附件一中样本数据的 367 个变量进行降维、建模，筛选出主要建模变量。

4.2.1 问题分析

根据实际情况，可以认为辛烷值的测量值是测量时刻前两小时内操作变量的综合效果，即预处理中取操作变量两小时内的平均值与辛烷值的测量值对应得到附件一中提供的 325 个样本。

附件一提供的 325 个样本数据，每个样本数据包含有 7 个原料性质、2 个待生吸附剂性质、2 个再生吸附剂性质、2 个产品性质等变量以及另外 354 个操作变量，共计 367 个变量。为建立降低辛烷值损失模型，首先要对上述附件一提供的 367 个变量降维，筛选出 30 个以下具有代表性、独立性的主要变量。

数据维度的降低方法主要有两种：

（1）仅保留原始数据集中最相关的变量，即特征选择。

（2）寻找一组较小的新变量，其中每个变量都是输入变量的组合，包含与输入变量基本相同的信息（降维）。

根据附件一提供的数据，下面拟采用具体的四步操作对原始的 367 个变量进行降维处理：

（1）缺失值比率(Missing Value Ratio)

（2）低方差滤波(Low Variance Filter)

（3）高相关滤波(High Correlation filter)

（4）随机森林(Random Forest)

具体降维处理流程图，如图 4.1 所示。

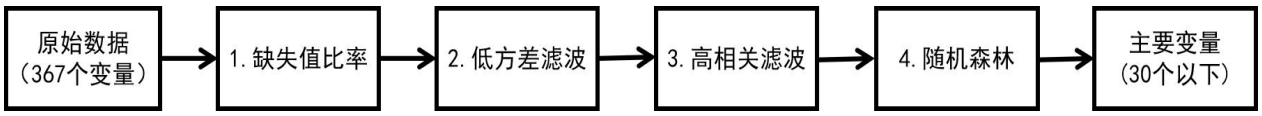

图 4.1 降维处理流程图

4.2.2 缺失值比率(Missing Value Ratio)

降维操作首先进行缺失值比率分析。基于包含缺失值多的数据列包含有用信息的可能性较少。因此，可以将数据列缺失值大于某个阈值的列去掉。阈值越高，降维方法更为积极，即降维越少。

可以通过调用 Pandas 库进行数据分析，Pandas 库是用于数据操纵和分析，建立在 Numpy（以提供高性能的矩阵运算）之上。数据分析结果发现原始数据中无缺失值。

通过 Python 进行缺失值比率分析，其分析结果图如图 4.2 所示。

缺失值比率处理后，所剩变量数为 367。

```
[325 rows x 367 columns]
v0      0.0
v1      0.0
v2      0.0
v3      0.0
v4      0.0
       ...
v362    0.0
v363    0.0
v364    0.0
v365    0.0
v366    0.0
Length: 367, dtype: float64
```

图 4.2 缺失值比率数据分析结果

4.2.3 低方差滤波(Low Variance Filter)

降维操作第二步进行低方差滤波降维，低方差降维假设数据列变化非常小的列包含的信息量少。因此，删除待处理变量中低方差的一些特征,再结合方差的大小进一步分析。

由于方差与数值大小有关，所以低方差降维前需要对数据做归一化处理。归一化相关公式，如式 4-2 所示。

$$u^* = \frac{u - \min(u)}{\max(u) - \min(u)} \tag{4-2}$$

其中，x 表示待处理的变量，x^*表示归一化后的结果。

变量的方差计算公式，如式 4-3 所示。

$$S_j^2 = \frac{1}{N}\sum_{j=1}^{N}(x_{ij} - \mu_i)^2, \quad i = 1,2,...,367 \tag{4-3}$$

其中，i 为变量编号，一共有 367 个变量；j 为样本编号，从 1 到 N；N 为样本的个数（325 个）；x_{ij} 代表第 i 个变量第 j 个样本的具体取值；μ_i 为第 i 个变量 325 个样本的平均值；S_j^2 为第 i 个变量 325 个样本的方差。

若特征方差小，表明某个特征与大多样本的值比较相近；若特征方差大，表明某个特征与很多样本的值都有差别。方差小的数据所含信息较少，这些变量一直处于相对稳定的状态，它们对辛烷值损失的影响很小。

通过 MATLAB 进行归一化处理，并计算方差。在实际低方差降维处理操作中，有效信息与设置的筛选方差阈值呈负相关，有大量数据的归一化方差值集中在 0~0.05 处，根据数据分析，拟将低方差降维方差阈值设置为 0.01。

如图 4.3 所示，是归一化方差散点图，其中深蓝色散点为变量的归一化方差结果，淡红色直线为方差阈值，选取低方差降维方差阈值为 0.01，对所有低于阈值的变量进行滤除操作。

低方差滤波降维处理后，去除掉 35 个变量，所剩变量数为 332。

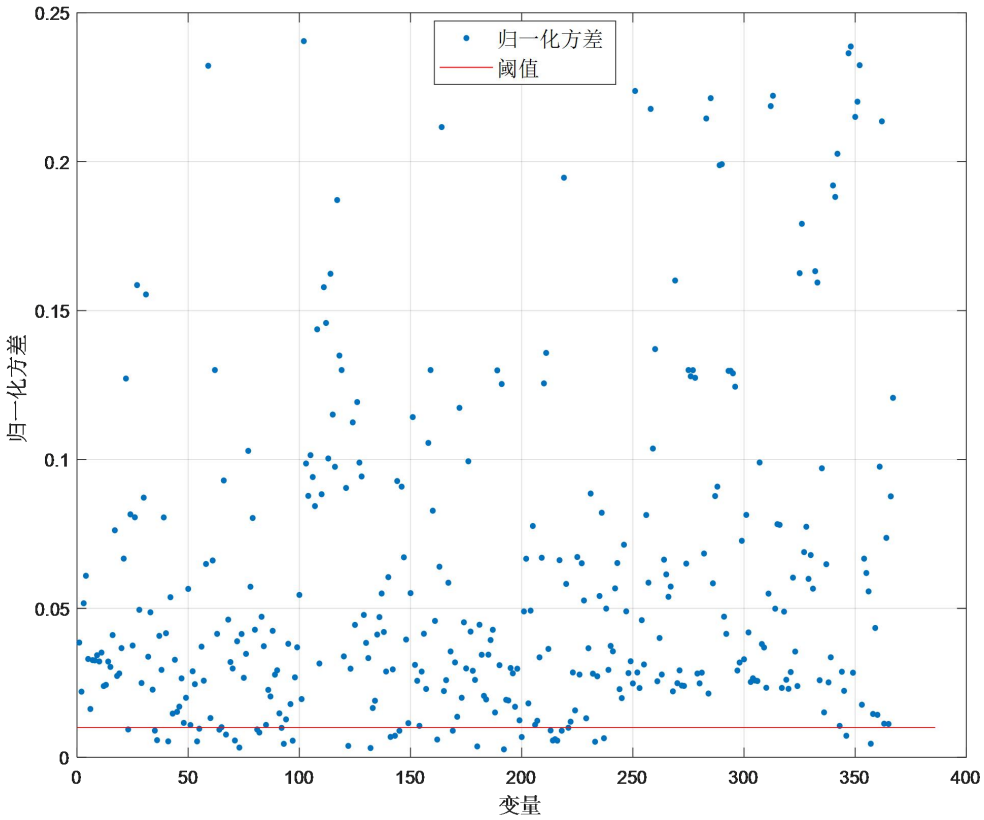

图 4.3 归一化方差散点

4.2.4 高相关滤波(High Correlation filter)

降维操作第三步进行高相关滤波降维，计算皮尔逊相关系数，作为反映变量之间相关关系密切程度的统计指标，其公式，如式 4-4 所示。

$$r=\frac{n\sum x_i x_j-\sum x_i\sum x_j}{\sqrt{n\sum x_i^2-(\sum x_i)^2}\sqrt{n\sum x_j^2-(\sum x_j)^2}},\quad i,j=1,2...,367 \tag{4-4}$$

其中 r 表示相关系数，x_i、x_j 表示两个待处理变量。

相关系数的取值介于-1 与+1 之间，即-1 ≤r≤+1。其性质如下：

（1）当 $r>0$ 时，表示两变量正相关，$r<0$ 时,两变量为负相关；

（2）当$|r|=1$ 时，表示两变量为完全相关，当 $r=0$ 时，表示两变量间无相关关系；

（3）当 $0<|r|<1$ 时，表示两变量存在一定程度的相关。且 r 越接近 1，两变量间线性关系越密切，$|r|$越接近于 0，表示两变量的线性相关越弱。

对相关系数进行三级划分如下：

$|r|<0.4$ 为低度相关；$0.4\le|r|\le0.6$ 为显著性相关；$0.6\le|r|<1$ 为高度线性相关。

根据相关程度划分，如果两个变量之间相关性的绝对值大于 0.6，可以认为这两个变量之间高度相关，可以用一个变量表示另一个变量。为了实现降维处理，应当从这两个变量之间选择一个删去。我们将两个高度相关变量与辛烷值损失之间相关性的绝对值进行比较，选择相关性较小的删去。在实际高相关降维处理操作中，根据具体的数据分析，拟将高相关降维相关性阈值设置为 0.6。

高相关降维处理前原始数据的相关性分析热图，如图 4.4 所示，高相关滤波降维结果的相关分析热图，如图 4.5 所示。

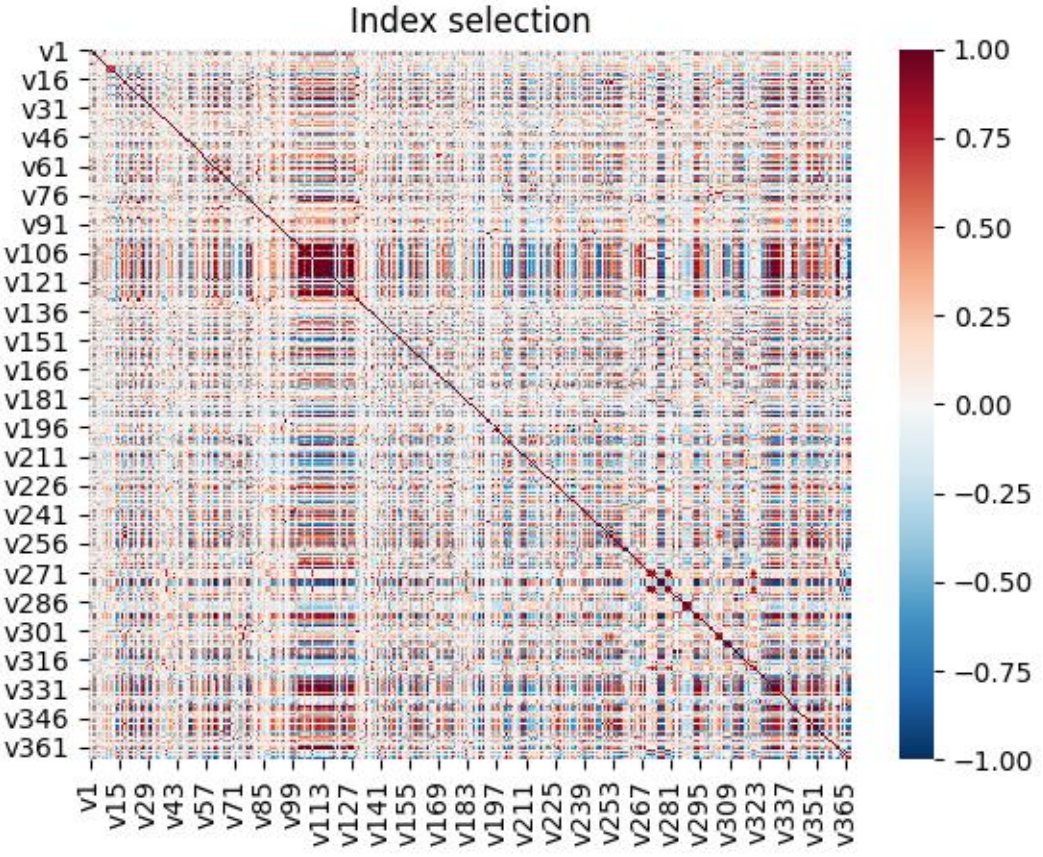

图 4.4 高相关滤波降维前的相关分析热图

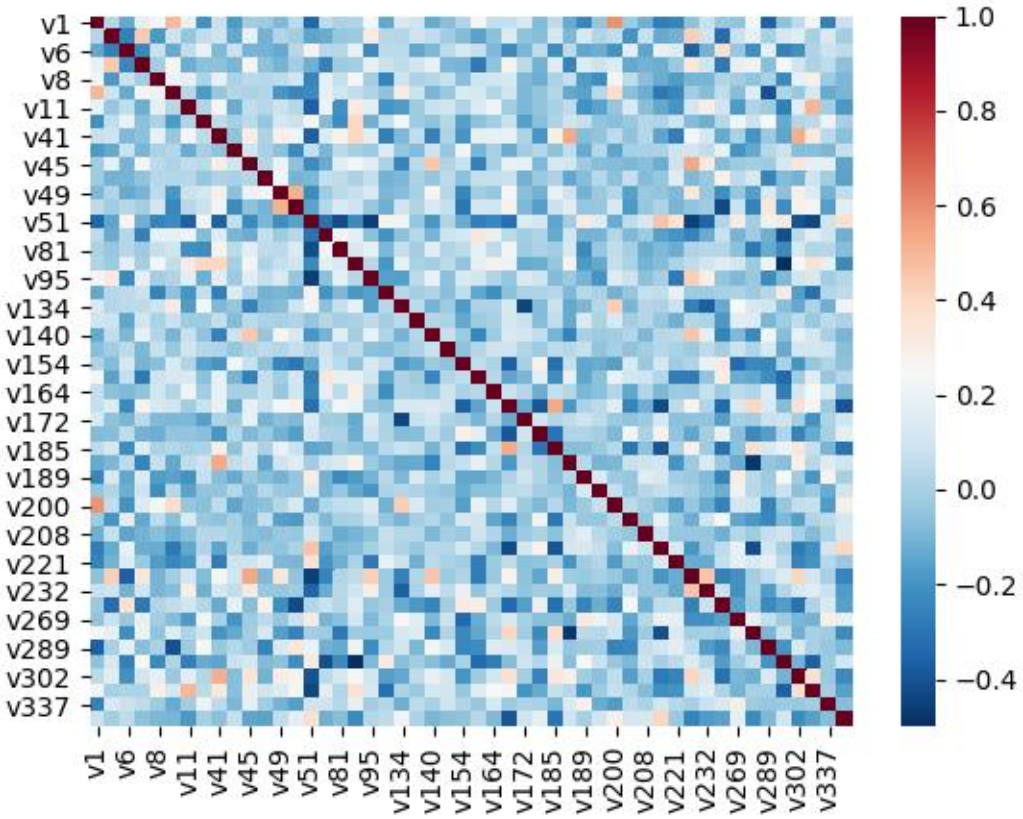

图 4.5 高相关滤波降维结果的相关分析热图

图 4.4 和图 4.5 中，相关分析热图通过颜色深浅度表示相关程度，右侧颜色色谱指示颜色与相关程度数值的对应关系，颜色越浅越靠近白色表示相关程度越低，颜色越深越偏向红色表示正相关程度越大，颜色越深越偏向蓝色表示负相关程度越大。图中深红色对角线为同一变量的相关曲线。

对比高相关降维处理前的原始数据相关性热图（图 4.4）和高相关降维处理结果的相关性热图（图 4.5），从整体上看，可以直观地看出进行高相关降维处理后，滤除了相关性较高的变量。

高相关滤波降维处理后，去除掉 282 个变量，所剩变量数为 50。

4.2.5 随机森林(Random Forest)

降维操作最后一步通过随机森林降维。随机森林，即组合决策树，一种基于 Bagging 的集成学习方法，可以用来分类，在降维中可用于进行特征选择，有助于选择较小的特征子集，对随机森林数据属性的统计评分会揭示出预测能力最好的属性。

随机森林有高的准确率；随机性的引入，使得随机森林不容易过拟合，有很好的抗噪声能力；训练速度快，可以得到变量重要性排序；易实现并行化等优点。

通过随机森林进行特征选择的目标有两个，一是找到与应变量高度相关的特征变量；二是选择出数目较少的特征变量并且能够充分的预测应变量的结果。

实际数据处理操作中步骤如下：

（1）初步估计和排序

A. 对随机森林中的特征变量按照 VI（Variable Importance）降序排序。

B. 确定删除比例,从当前的特征变量中剔除相应比例不重要的指标，从而得到一个新的特征集。

C. 用新的特征集建立新的随机森林,并计算特征集中每个特征的 VI,并排序。

D. 重复以上步骤,直到剩下 *m* 个特征。

（2）根据 1 中得到的每个特征集和它们建立起来的随机森林,计算对应的袋外误差率(OOB err),将袋外误差率最低的特征集作为最后选定的特征集。

通过 Python 进行随机森林操作，筛选出最终的主要变量为 30 个。

如表 4.2 所示，为最终筛选出的 30 个主要变量及其重要性排序对应表。表格第一列为重要性序号（数值越小重要性越高），第二列为对应于原题“附件一：325 个样本数据.xlsx”中的变量排序，第三列为对应的主要变量名称。表 4.2 已按照降维结果主要变量的重要性进行排序。

表 4.2 主要变量筛选结果及其重要性对照表

重要性排序(越小越重要)	主要变量名称	变量序号		重要性排序(越小越重要)	主要变量名称	变量序号
1	辛烷值 RON(原料中)	v2		16	D-113 顶放空线流量	v158
2	0.3MPa 凝结水出装置流量	v51		17	D121 温度	v85
3	D-204 液位	v137		18	密度(20℃) kg/m³	v7
4	E-101D 壳程出口管温度	v134		19	循环水进装置流量	v49
5	非净化风进装置压力	v54		20	焦炭 wt%	v11
6	精制汽油出装置温度	v34		21	芳烃 v%	v5
7	燃料气进装置流量	v45		22	D-109 上部温度	v166
8	再生器顶底差压	v95		23	硫含量 μg/g（原料中）	v1
9	D110 顶底压差	v99		24	溴值 gBr/100g	v6
10	硫含量 μg/g（产品中）	v8		25	D104 压力	v81
11	D203 出口燃料气流量	v140		26	干气出装置流量	v41
12	D123 冷凝水罐液位	v150		27	D-109 吸附剂料位	v164
13	燃料气进装置压力	v44		28	循环水出装置流量	v50
14	1.0MPa 蒸汽进装置温度	v48		29	D-122 入口管温度	v154
15	D-107 底排放滑阀压差	v172		30	D-103 底部液位	v179

如图 4.6 所示，为降维筛选出的 30 个主要变量及其重要性对应图，其中，横坐标表示各主要变量的重要性程度，纵坐标为主要变量序号 v_n（对应于原题“附件一：325 个样本数据.xlsx”中的变量排序），最终筛选出的主要变量实际名称详见论文“附件一：主要变量.xlsx”。

主要变量按重要顺序包括：辛烷值 RON、0.3MPa 凝结水出装置流量、D-204 液位、E-101D 壳程出口管温度、非净化风进装置压力、精制汽油出装置温度、燃料气进装置流量、再生器顶底差压、D110 顶底压差、硫含量（产品中）等。

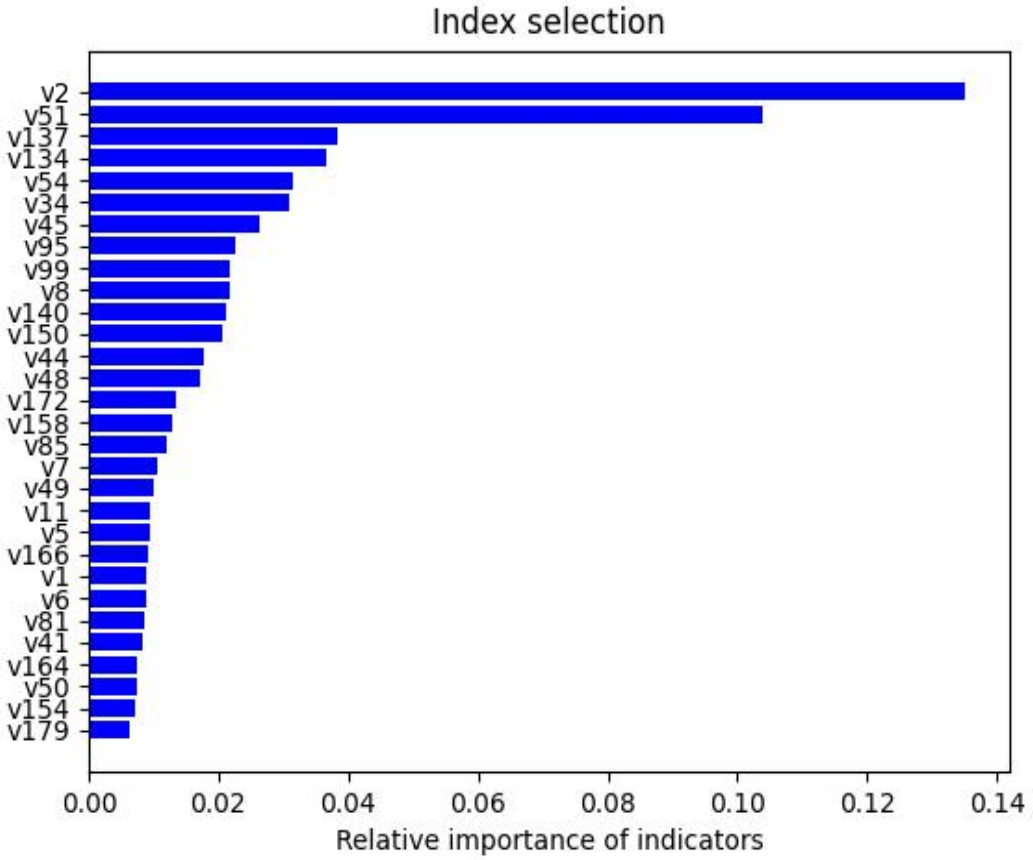

图 4.6 主要变量筛选结果及其重要性条形图

4.2.6 分析总结

根据寻找建模主要变量的目标，即将初始 325 个样本数据的 367 个变量进行降维、建模，筛选出主要建模变量。

降维处理过程中，根据实际待样本进行分析，采用缺失值比率(Missing Value Ratio)、低方差滤波(Low Variance Filter)、高相关滤波(High Correlation filter)，及随机森林(Random Forest)，共四步操作对原始的 367 个变量进行降维处理，最终筛选出如表 4.2 所示的 30 个主要变量，并得到对应的重要性程度，如图 4.6 所示。

4.3 问题 3 的分析与求解

问题三目标是建立辛烷值(RON)损失预测模型，集合样本数据和问题二中筛选出的建模主要变量，建立辛烷值(RON)损失预测模型，并模型验证。

4.3.1 问题分析

经过问题二中的降维处理，已经筛选得到了 30 个主要变量及其重要程度，如表 4.2 所示，问题三的本质是找到 30 个主要变量与辛烷值(RON)损失的关系，这实际上是一个多元回归问题，针对多元回归问题的处理方法有：BP 神经网络、随机森林、AdaBoost 回归算法等。

在 Kaggle 以及阿里云天池竞赛类似问题的解决中，AdaBoost 算法是一种主流的数据分析算法，且均取得了显著的效果。而针对本文问题三，通过对比几种解决方案和效果，在实际处理过程中拟选用 AdaBoost 回归算法进行处理。

AdaBoost 回归算法优点是不易发生过拟合，且在 AdaBoost 的框架下，可以使用各种回归分类模型来构建弱学习器，非常灵活；缺点是对异常样本敏感，异常样本在迭代中可能获得较大权重，影响强学习器的预测准确性。

4.3.2 模型建立与求解

AdaBoost 回归模型是将一些简单的基学习器，通过迭代的方式确定每个基学习器的权重。本文选择的基学习器是决策回归树，因为决策树较为简单且用于回归模型时有一定的效果，我们将其选为本次 AdaBoost 算法基学习器，可以满足求解需求。

AdaBoost 回归算法描述如下：

- 输入：

训练集 $D=\{(x_i,y_i)\}_{i=1}^{m}$，其中 $x_i \in X \subseteq R^n, y_i \in Y$；基学习算法 L；决策回归树的个数 T。

- 过程：

（1）初始化训练样本的权值分布表达式如式 4-5 所示：

$$D_1=(\omega_{11},...,\omega_{1i},...,\omega_{1m}),\ \omega_{1i}=1/m,\ i=1,2,...,m \tag{4-5}$$

（2）对迭代轮次 $t=1,2,...,T$

I. 使用具有当前分布 D_t 的训练数据集训练回归树 $h_t=L(D,D_t)$；

II. 计算训练集上的样本最大误差如式 4-6 所示：

$$E_t=\max|y_i-h_t(x_i)|,\ i=1,2,...,m \tag{4-6}$$

III. 计算每个样本的相对误差：

本问题采用平方误差，则相对误差计算如式 4-7 所示：

$$e_{ti}=\frac{1}{E_t^2}(y_i-h_t(x_i))^2 \tag{4-7}$$

IV. 回归树 h_t 在训练数据集上的回归误差率如式 4-8 所示：

$$\varepsilon_t=\sum_{i=1}^{m}\omega_{ti}e_{ti} \tag{4-8}$$

V. 回归树 h_t 的权重系数计算公式如 4-9 所示：

$$\alpha_t=\frac{\varepsilon_t}{1-\varepsilon_t} \tag{4-9}$$

VI. 更新训练集的样本分布 $D_{t+1}=(\omega_{t+1,1},...,\omega_{t+1,i},...,\omega_{t+1,m})$，计算公式如式 4-10 所示：

$$\omega_{t+1,i}=\frac{\omega_{ti}}{Z_t}\alpha_t^{1-e_{ti}} \tag{4-10}$$

其中，Z_t 是规范化因子计算表达式如式 4-11 所示：

$$Z_t=\sum_{i=1}^{m}\omega_{ti}\alpha_t^{1-e_{ti}} \tag{4-11}$$

（3）构建回归树的线性组合 $\sum_{t}^{T}\alpha_t h_t(x)$，得到最终的强学习器，其表达式如式 4-12 所示：

$$H(x)=\sum_{t=1}^{T}(\ln\frac{1}{\alpha_t})g(x) \tag{4-12}$$

其中，$g(x)$ 是所有 $\alpha_t h_t(x), t=1,2,...,T$ 的中位数。

- 输出：强学习器 $H(x)$

根据 AdaBoost 回归算法流程对问题三进行数据处理分析。

因为样本数量只有 325 个，所以我们考虑采用 K-折交叉验证来完成 AdaBoost 模型的训练和验证，其中 K 选取 5 折。

我们采用的训练集输入特征向量为问题二中降维筛选出的 30 个主要变量。325 个数据样本中每一个样本都包含了 30 个特征向量维度，以及对应的标签值 target，其中标签值为辛烷值(RON)损失。

4.3.3 结果与分析

4.3.3.1 拟合效果

根据所选的 30 个主要变量（特征向量），使用 AdaBoost 回归算法建立了关于辛烷值(RON)损失的模型。

对处理结果进行数据分析，可以得到原始样本值与决策树回归器拟合曲线，如图 4.7 所示，以及原始样本值与 AdaBoost 回归算法拟合曲线，如图 4.8 所示。

在图 4.7 中，蓝色曲线代表原始样本值曲线，橙色曲线代表决策树回归器拟合效果曲线；图 4.8 中，蓝色曲线代表原始样本值曲线，红色曲线代表 AdaBoost 回归算法拟合效果曲线。图 4.7 和图 4.8 的横坐标(data)表示样本序号，纵坐标(target)表示辛烷值(ROM)损失。

对两种拟合曲线图像进行分析，对比图 4.7 和图 4.8 可以得出，与决策树回归器拟合曲线相比，AdaBoost 回归算法拟合曲线有很大的提升；且从图 4.8，对 比原始样本数据曲线和 AdaBoost 回归算法拟合曲线，可以看出 AdaBoost 拟合曲线实际拟合效果较好。

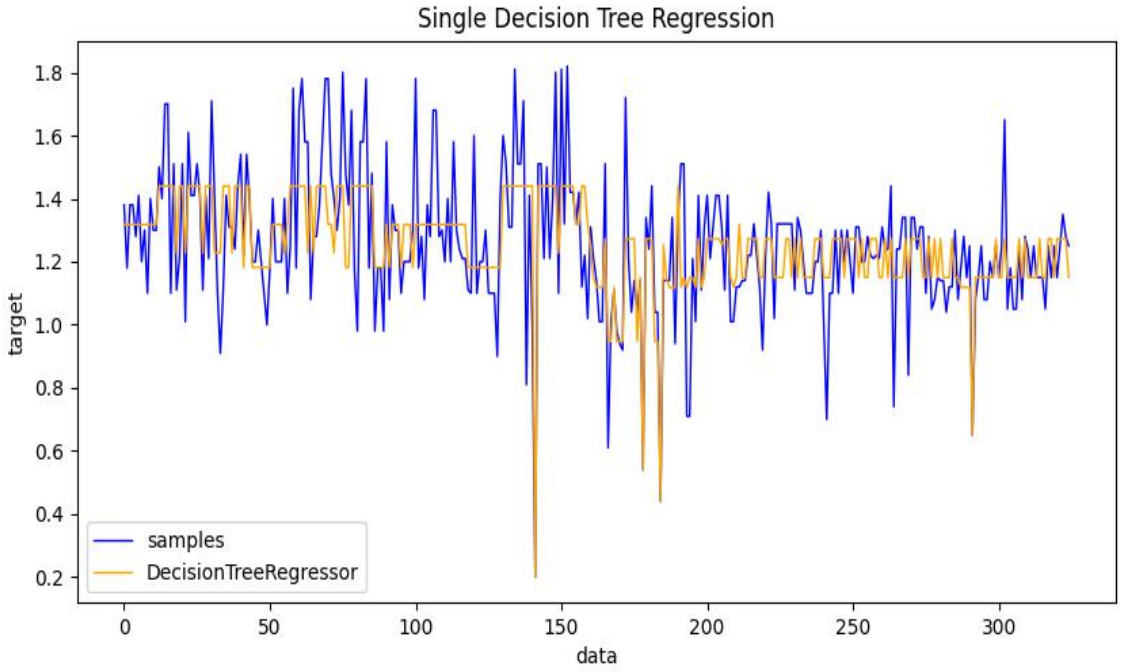

图 4.7 原始样本值与决策树回归器拟合曲线

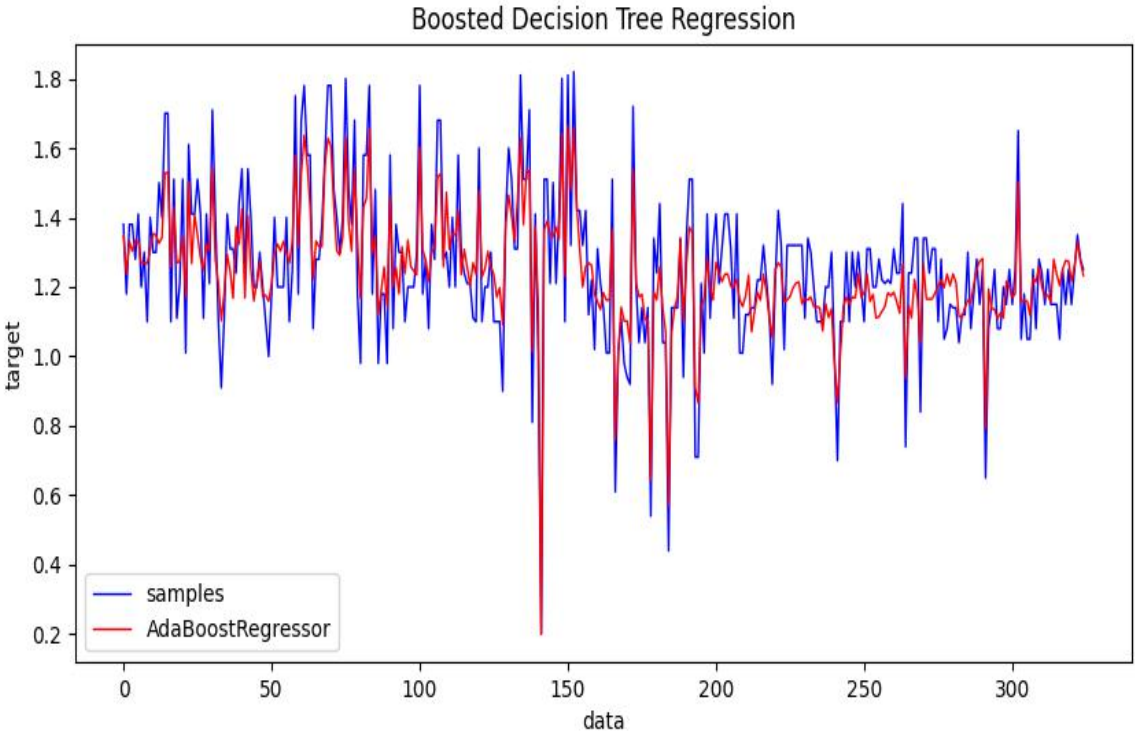

图 4.8 原始样本值与 AdaBoost 回归算法拟合曲线

4. 3. 3. 2 模型检验

如图 4.9 所示，为 K-折交叉验证均方损失图。

在图 4.9 中，横坐标是 AdaBoost 训练尺寸，纵坐标代表均方误差值，红色曲线(e)代表 AdaBoost 回归算法在训练集上经过 5 次分割组合后训练集上得出的均方误差；绿色曲线(b)代表算法在 5 次验证集上得出的均方误差；浅红色区域(s)代表红色曲线(e)均值与标准差的

差值范围；浅绿色区域(t)代表绿色曲线(b)均值与标准差的差值范围。均值与标准差的差，表示单测量标准偏差与随机误差，以正态分布曲线作标准，平均值描述样本集中趋势，标准差描述离散趋势。

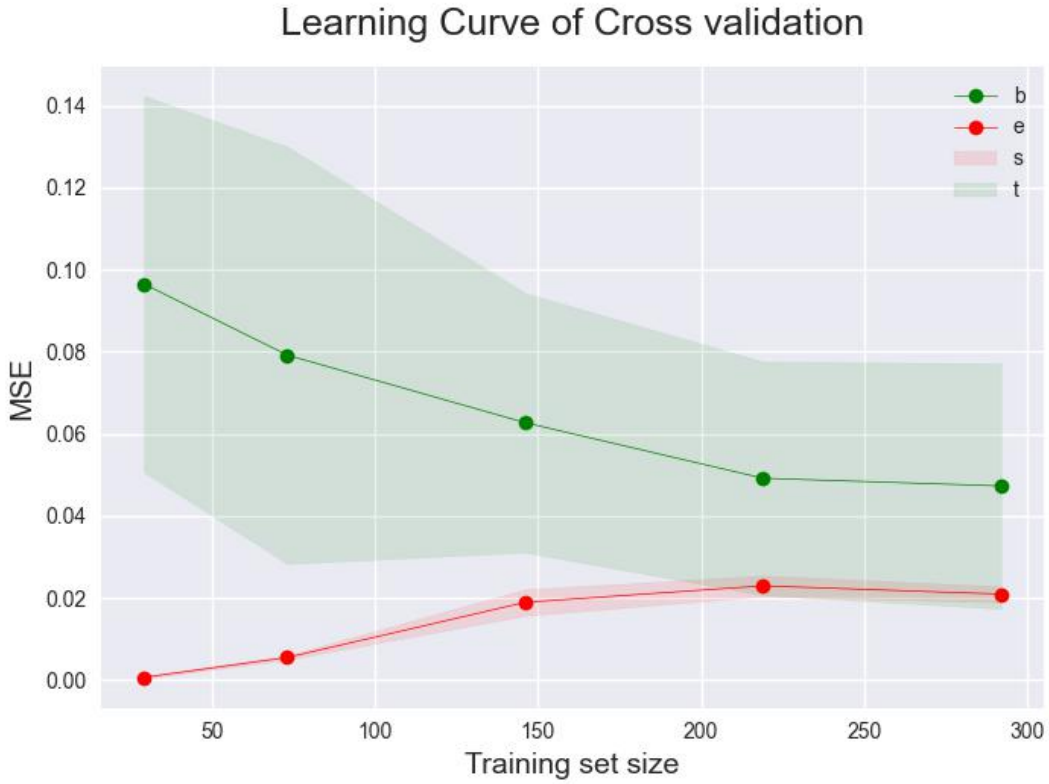

图 4.9 K-折交叉验证均方损失图

K-折交叉验证 5 次辛烷值(RON)损失的均方误差分别为：[0.09055122、0.07816736、0.0607159、0.05257738、0.05102547]，根据实际数据分析，测试集最终 loss（平均均方误差）为 0.066，也就是有 0.24 左右的平均误差（每次训练测试集均包含了 80%数据）。可以得出我们训练后的 AdaBoost 学习器可以对本题的学习数据规律有较好的反映和拟合。

如图 4.10 所示，为 AdaBoost 回归算法验证结果图，显示出 AdaBoost 回归算法在不同基学习器数量情况下的效果。

对图 4.10 进行分析，在使用 AdaBoost 回归算法处理数据过程中，本文得到了使用 1 个基学习器和使用 50 个基学习器组合的 AdaBoost 算法得分(Score)，对模型进行验证并说明。在同样的 train size 条件下，对于不同的 Estimator Num 数目情况，可以看到 Training Score 以及 Testing Score 在 50 个基学习器组合的情况下，分数达到饱和，在 0.7 以上。由此可以验证说明 AdaBoost 回归算法在学习器叠加的情况下可以达到良好的效果。

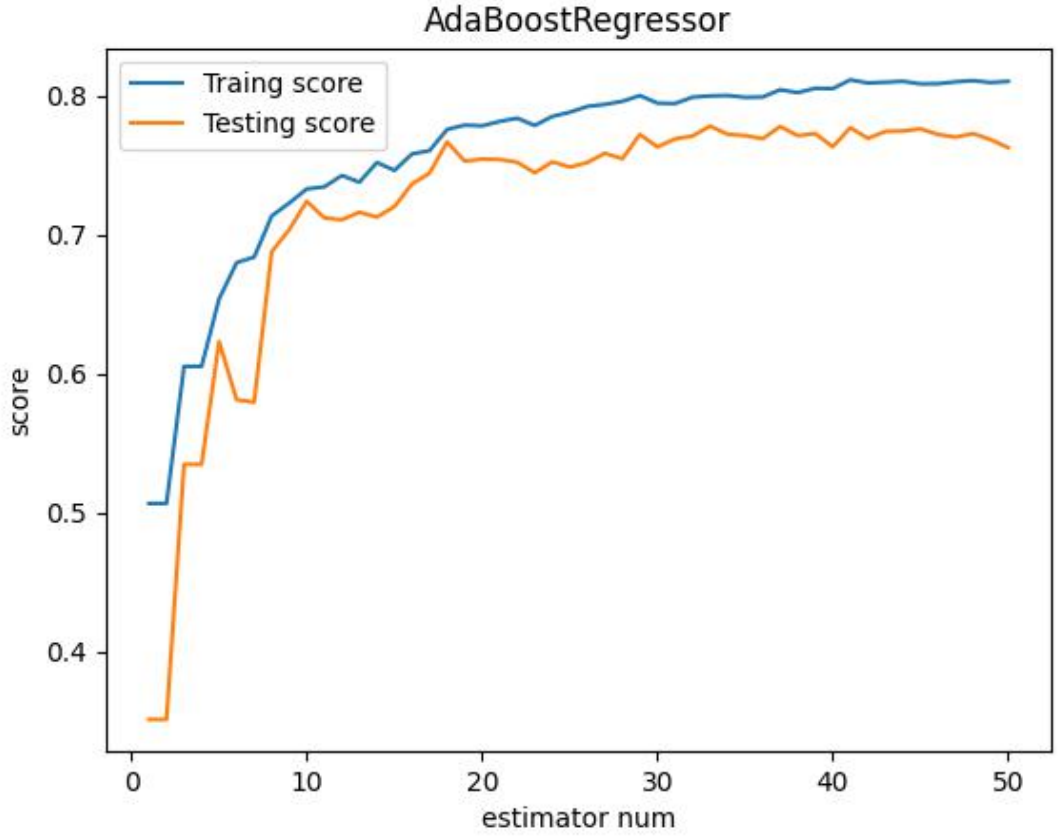

图 4.10 AdaBoost 回归算法验证结果图

4.3.4 分析总结

针对问题三，利用降维筛选出的 30 个主要变量及其重要程度，建立辛烷值（RON）损失预测模型，即找到 30 个主要变量与辛烷值（RON）损失的关系。针对这一多元回归问题根据分析，本文选取了 AdaBoost 回归算法进行处理，阐述了 AdaBoost 回归算法原理及其处理步骤，并选择决策回归树作为基学习器。

最终通过 AdaBoost 回归算法，本文建立了关于辛烷值(RON)损失的模型，并进行了数据分析，通过 K-折交叉验证，证明模型效果良好。

4.4 问题 4 的分析与求解

问题四的目标是对主要变量操作方案进行优化。根据问题三中构建的模型，在限制条件下获得优化后的操作条件。

4.4.1 问题分析

第四题本质是一个变量优化的问题。该问题要求在保证产品硫含量不大于 5μg/g 的前提下，根据问题三建立的主要变量与辛烷值（ROH）损失降幅之间的关系模型，对数据样本中的主要变量进行优化。最终要从 325 个样本数据对应的主要变量数据中，获得辛烷值（RON）损失降幅大于 30%的优化操作条件。

首先要确定基本的目标函数和约束条件，根据题意，首先控制优化过程中原料、待生吸附剂、再生吸附剂的性质保持不变，即表 4.2 主要变量筛选结果中的辛烷值(RON)、硫

含量（产品中）、密度(20℃)、焦炭(wt%)等共 7 个变量保持不变，根据剩余的 23 个可变主要变量进行调节。

针对这种优化问题，一般是沿着模型函数的负梯度方向进行搜索，但是本文选用的模型是基于决策回归树的 AdaBoost 回归算法，此模型实质上是由一系列的分段函数构成的。所以本文基于剩余的 23 个可变主要变量构建搜索空间，优先选取重要度高的特征并结合次要特质进行搜索迭代，目的是在小范围内搜索到满足产品硫含量不大于 5μg/g，且辛烷值（RON）损失降幅大于 30%样本条件的可行解。

4.4.2 模型建立与求解

根据问题的分析思路，可以的到该优化问题的目标函数和约束条件，如式 4-13 所示。

$$\min \ y$$
$$s.t.\begin{cases} y = f(x) \\ x = (x_1, x_2, ..., x_{30})^T \\ \min x_i \le x_i \le \max x_i, \ i = 1,2,...,30 \\ x_8 \le 5 \end{cases} \tag{4-13}$$

其中，y 表示辛烷值(RON)损耗；x_i 表示筛选出的主要变量，如表 4.2 主要变量筛选结果所示；i 表示筛选出的 30 个主要变量序号；区间$[\min \ x_i, \ \max \ x_i]$表示第 i 个主要变量的取值范围，来源于于原题“附件四：354 个操作变量信息.xlsx”；x_8 对应主要变量“硫含量（产品中）”，根据题意产品硫含量取值不大于 5μg/g。

对于第 m 个样本，它的当前状态表达式如式 4-14 所示：

$$x^0(m) = (x_1^0(m), x_2^0(m), ..., x_{30}^0(m))^T, \ m = 0,1,2,...,325 \tag{4-14}$$

其中 $x_1^0(m), x_2^0(m), ..., x_{30}^0(m)$ 分别表示表 4.2 中降维后的三十个主要变量。

根据问题三中我们建立的模型，以 $x^0(m)$ 作为初始状态的辛烷值(RON)损失预测值表达式如式 4-15 所示：

$$y_0 = f(x^0(m)) \tag{4-15}$$

以 y_0 为初始状态，开始寻找满足辛烷值（RON）损失降幅大于 30%的样本对应的主要变量优化后的操作条件，从第 k 步到第 $k+1$ 步的递推应满足辛烷值损失有所降低，如式 4-16 所示。

$$f(x^k(m)) - f(x^{k+1}(m)) > 0 \tag{4-16}$$

按照式 4-16 式中的递推公式，对需要调整的变量进行搜索，当满足辛烷值(RON)损失降幅大于 30%时，即满足式 4-17 时搜索停止，得出结果点列 $x^k(m)$。

$$\frac{f(x^0(m)) - f(x^k(m))}{f(x^0(m))} \ge 30\% \tag{4-17}$$

4.4.3 优化结果与分析

根据上述分析结果和建立的模型求解思路，通过 Python 编程对数据进行处理，并使用算例较强的服务器在可行解的搜索空间内进行问题求解，得到了最终的主要变量优化后操作条件的具体数据。

如表 4.3 所示，为主要变量优化后操作条件部分结果表。

表 4.3 主要变量优化后操作条件部分结果表

主要变量(30 个)	样本 1 处理结果	样本 2 处理结果	...	样本 7 处理结果	...
v1	1.88000000e+02	1.20779735e+02			
v5	2.23700000e+01	3.85826835e+01			
v6	6.14871429e+01	4.52796620e+02			
v7	7.26085714e+02	4.99966625e+01			
v8	3.20000000e+00	3.20000000e+00			
v2	8.92200000e+01	8.93200000e+01			
v11	2.32000000e+00	2.37000000e+00			
v34	3.00000000e+01	3.00000000e+01			
v41	4.30000000e+02	4.30000000e+02			
v44	3.50000000e-01	3.50000000e-01			
v45	3.50000000e+02	3.50000000e+02			
v48	1.50000000e+02	1.50000000e+02			
v49	2.50000000e+02	4.50000000e+02			
v50	2.50000000e+02	2.50000000e+02			
v51	3.00000000e+00	3.00000000e+00			
v54	6.13384630e-01	6.13151955e-01			
v81	2.31469985e+00	2.31618555e+00	…	null	…
v85	3.17132125e+01	3.04542815e+01			
v95	3.62782020e+01	3.58407535e+01			
v99	4.63777335e+01	4.45760290e+01			
v134	1.23650450e+02	1.21358590e+02			
v137	5.28066510e+01	2.88228680e+01			
v140	4.38996390e+02	4.70734730e+02			
v150	5.00321045e+01	5.05015840e+01			
v154	2.32663460e+01	2.11607520e+01			
v158	7.75412900e+01	8.01126835e+01			
v164	9.68019860e-01	3.84858900e+00			
v166	2.48769275e+01	2.82121350e+01			
v172	-3.78947140e-01	-3.79253070e+01			
v179	-7.39462115e-01	-1.2616710e+00			
辛烷值(RON)损失值	1.12342667	1.17824176			

表 4.3 中，展示了部分样本的处理结果及所对应的优化后辛烷值(RON)损失值的预测结果，所展示的样本处理结果为样本 1 处理结果、样本 2 处理结果和样本 7 处理结果。

根据题目要求，要保证产品中硫含量不大于 5μg/g 的前提下，所以本文将硫含量取为定值 3.2μg/g。同时优化过程中原料、待生吸附剂、再生吸附剂的性质保持不变，在此前提下得到了 325 个样本的主要变量优化结果。

结合表 4.2 得出的 30 个主要变量表，对于处理结果进行数据分析，以表 4.3 中样本 1 处理结果数据为例，E-101D 壳程出口管温度(v134)数据应调整为 123.650℃，D203 出口燃料气流量(v140)数据应调整为 438.996Nm3/h，D123 冷凝水罐液位(v150)数据应调整为 50.032%；以表 4.3 中样本 7 处理结果数据为例，处理结果为 null，即没有优化结果，表明无论如何对主要操作变量在取值范围内进行优化都无法达到使辛烷值(RON)损失降幅大于 30%优化目标。其余主要操作变量的调整以及，其他所有样本的处理结果分析，以此类推，不再赘述，完整优化结果表详见“附件：主要变量优化后操作条件结果表.xlsx”。

4.4.4 分析总结

对问题四处理结果总结如下，在 325 个样本中除了少部分样本无法实现使辛烷值(RON)损失降幅大于 30%的优化目标，其余大部分样本均可实现优化目标，且效果良好，对于即使是无法完成优化目标的样本辛烷值(RON)损失降幅也有一定程度的优化，可以帮助相关企业节省成本。

4.5 问题 5 的分析与求解

问题五的目标是模型的可视化展示。需要针对 133 号样本（原料性质、待生吸附剂和再生吸附剂的性质数据保持不变，以样本中的数据为准），以图形展示其主要操作变量优化调整过程中对应的汽油辛烷值和硫含量的变化轨迹。

4.5.1 问题分析

第五题目标是将模型可视化，并按照工厂实际情况对操作变量进行优化调整，以降低产品中辛烷值（RON）损失。其中原料性质、待生吸附剂和再生吸附剂的性质数据保持不变，硫元素含量为了满足国家要求，且工厂实际数据无法测量低于 3.2ug/g 的硫浓度，我们将硫浓度的标准定为 3.2ug/g 保持不变。

针对问题五，其求解过程不同于第四问的情况，不能直接对模型中所有主要操作变量同时进行优化，因为根据题意增加了变量调整限制，即要按照附件四中所限制的操作变量调整幅度逐步进行优化，如果依旧按照第四问的方法求取结果，会使得复杂度过高。

根据分析，问题五的求解过程本文引入了“贪婪算法”的概念。

4.5.2 模型建立与求解

本文为了降低复杂度，获取一个较为满意的优化结果，且同时保证操作方便，采用了贪婪算法对主要操作变量进行迭代优化。计算具体步骤如下：

I. 首先根据样本中各变量具体取值，计算出辛烷值损失的预测值 $y_0 = f(x^0(m))$，其中 $m = 133$ 代表第 133 个样本

II. 对于每个主要操作变量，在保持除该操作变量外其余变量不变的前期下，计算该变量+Δ和-Δ时，辛烷值损失的预测值表达式如式 4-18 和式 4-19 所示：

$$f(x_n^{k+1}(m)+\Delta) = f(x_1^k(m), x_2^k(m), ..., x_n^k(m)+\Delta, ..., x_{30}^k(m)) \tag{4-18}$$

$$f(x_n^{k+1}(m)-\Delta)=f(x_1^k(m),x_2^k(m),...,x_n^k(m)-\Delta,...,x_{30}^k(m)) \tag{4-19}$$

III. 比较 $f(x_n^{k+1}(m)+\Delta)$ 和 $f(x_n^{k+1}(m)-\Delta)$ 的大小，选取较小的作为 $f(x_n^{k+1}(m))$

IV. 对步骤III得到的结果进行比较，选取辛烷值损失降幅最大的变量所对应的操作步骤并执行，其表达式如式 4-20 所示：

$$y_{k+1}=\min\{f(x_1^{k+1}(m)),f(x_2^{k+1}(m)),...,f(x_n^k(m)),...,f(x_{30}^k(m))\} \tag{4-20}$$

V. 重复上述步骤，直至所有操作变量达到取值上下限或无法再降低辛烷值损失，即式 4-21 所示：

$$y_k=\min y \tag{4-21}$$

VI. 根据记录的逐步操作步骤，绘制辛烷值损失降幅曲线。

4.5.3 优化结果与分析

如图 4.11 及 4.12 所示，为原料硫含量以及汽油硫含量的箱型分析图，可以看出原料中的硫含量的分布在 50~400μg/g，而汽油硫含量主要分布在 3.2μg/g，同时存在大量异常点。根据两图的前后对应关系可以推断出，原料中的硫含量在经过加工一般都会稳定在 3.2μg/g。

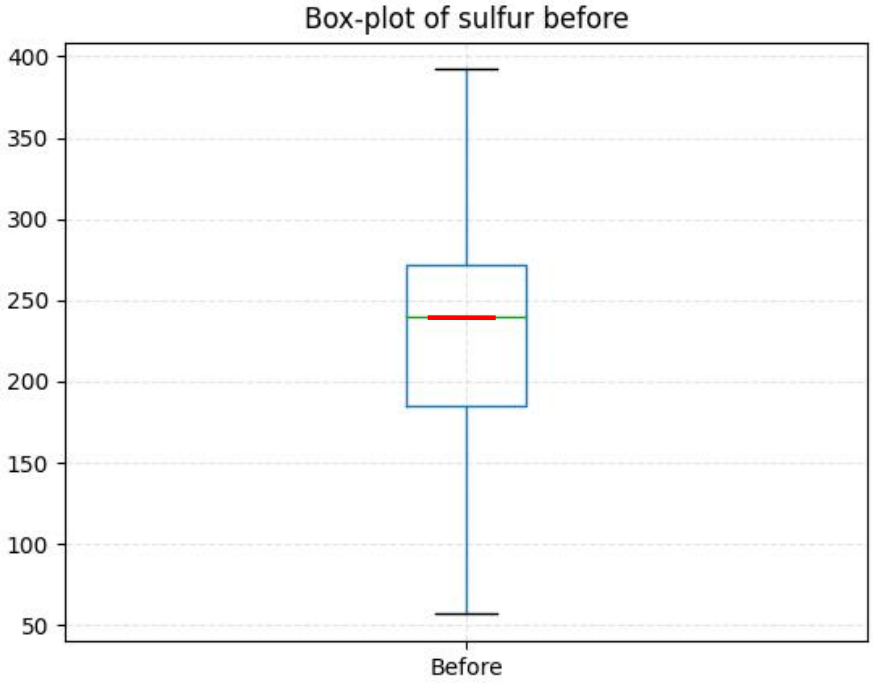

图 4.11 原料硫含量箱型分析图

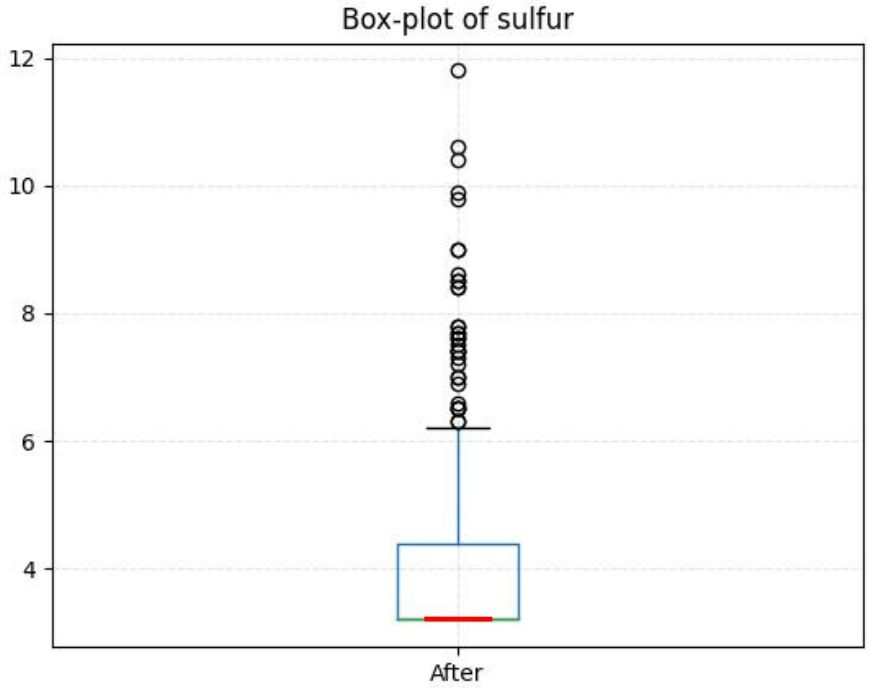

图 4.12 汽油硫含量箱型分析图

所以，根据图 4.13 所示，为主要操作变量优化调整过程中对应的汽油硫含量的变化轨迹图。横坐标表示主要操作变量逐步调节的总步数，纵坐标表示汽油硫含量。由图可知，随着主要操作变量的逐步调节，汽油硫含量总体变化趋势不大，波动较为平稳，维持在中位数在 3.2 的分布上，这满足我们以上的推断。

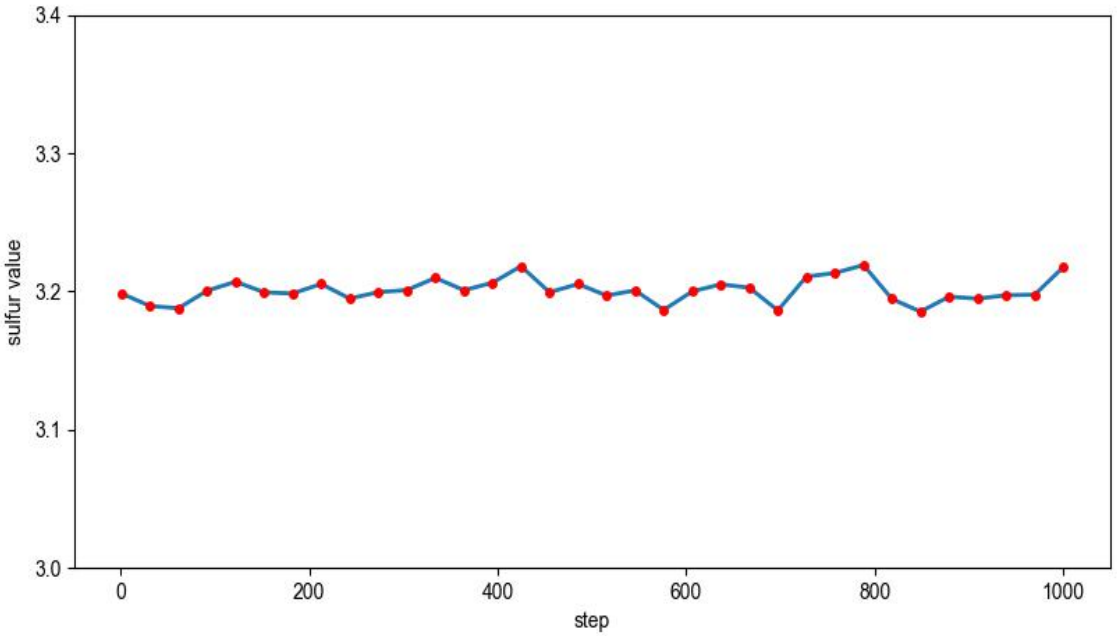

图 4.13 主要操作变量优化调整过程中对应的汽油硫含量的变化轨迹图

接着本文采用贪婪算法对主要操作变量逐步优化调整过程进行了分析，绘制了主要操作变量优化调整过程中对应的汽油辛烷值(RON)的变化轨迹图。

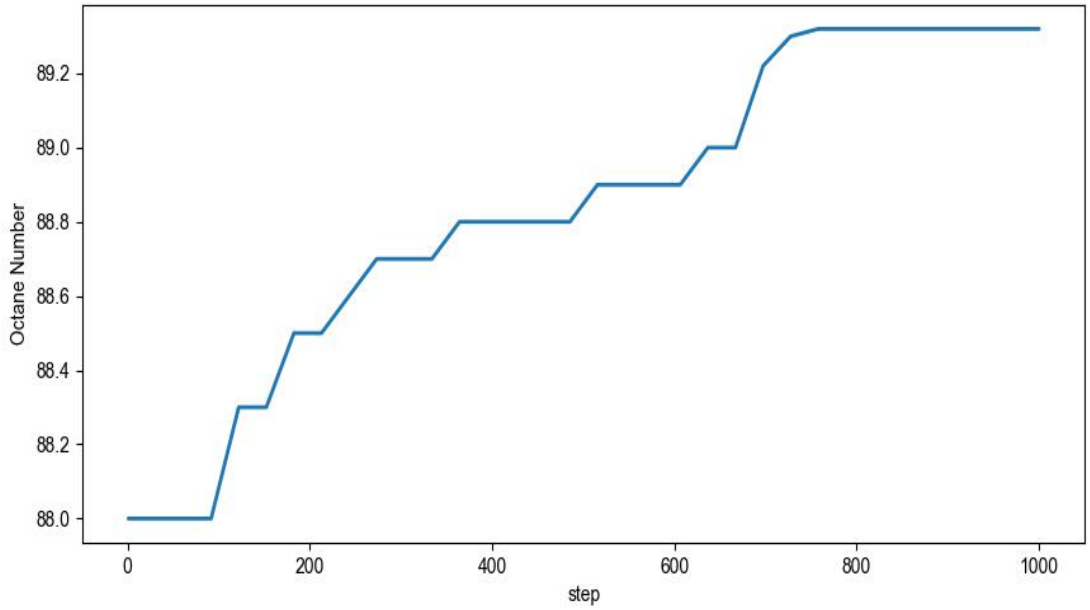

图 4.14 主要操作变量优化调整过程中对应的汽油辛烷值(RON)的变化轨迹图

如图 4.14 所示，为主要操作变量优化调整过程中对应的汽油辛烷值(RON)的变化轨迹图。横坐标表示主要操作变量逐步调节的总步数，纵坐标表示汽油辛烷值(RON)。由图可知，随着主要操作变量的逐步调节，汽油辛烷值(RON)总体呈现上升趋势，并最终达到平稳状态。整个过程中，在某些主要操作变量的特定调节阶段内，汽油辛烷值(RON)变换呈现局部平稳现象，这说明这些阶段内该操作变量的调节对于目标优化的效果不明显，在实际工业应用中应予以考虑分析。

4.5.4 分析总结

对问题五处理结果分析总结如下，为完成问题五模型可视化展示目标，首先绘制了原料及汽油硫含量箱型分析图，得知原料中的硫含量在经过加工一般都会稳定在 3.2μg/g。然后结合上述结论，并采用贪婪算法对主要操作变量逐步优化调整过程进行了分析，绘制了主要操作变量优化调整过程中对应的汽油辛烷值(RON)和硫含量的变化轨迹图。

五、模型的评价和推广

5.1 模型的优点

1. 本文提出了一个完整的数据特征选择流程模型，包括了缺失值补充、方差选择、特征间相关性计算、随机森林特征重要性排序，针对工业建模中的特征缩减和选择问题可以达到很好的效果；

2. 针对辛烷值（RON）损失预测模型，本文选择了 AdaBoost 回归算法作为主要算法，对于数据集进行了 K-折交叉验证以及强学习器的效果验证，验证结果都证明了 AdaBoost 回归算法这一优异的算法对本题目具有良好的建模效果；

3. 针对问题四，面对决策回归树是分段函数，无法通过偏导求解最优下降方向的问题，本文提出了使用问题二求解结果中重要性较高的特征进行全局搜索范围缩减的思想，大致得出了优化的区间

4. 虽然贪婪算法求取的结果不一定是全局最优解，但对于问题五这类复杂度较高的问题，在保证每一步骤最优的前提下，也可以对实际问题做出可观的优化，得到一个相对满意的结果。

5.2 模型的缺点

1. 模型采用 AdaBoost 算法，模型的表示较为复杂，无法通过直接的优化方程进行表示。所以该模型对于一般的机器学习的模型来说，可解释性较高，但是对于数学表征求解来说，可解释性较低。

2. 对于求解一个优化过程，贪婪算法有可能落入局部最优，并不是全局最优，后期仍有算法的改进空间。

5.3 模型的推广

本文采用的算法流程具有通用性，对于其他工业化产品的建模也同样有参考价值，也同样可以通过本文的特征选择、筛选，以及 AdaBoost 学习器的建立，特征缩减搜索空间，以及贪婪算法求解局部最优值，完成常见模型的建立以及简单求解。

针对特征空间，进行搜索是一个考验计算机算力的问题，本文求解时候使用的是实验室多人服务器，面对这类问题性能相对较差，所以针对这样的局限，在大规模工业化环境中可以考虑使用 XGBoost 模型，对数据进行分布式训练，加速模型求解。

参考文献

[1] 陈焕章, 李永丹, 赵地顺,等. 提高 FCC 汽油辛烷值的技术进展[J]. 化工科技市场, 2005(01):25-29.
[2] 高俊, 姚成, 章俊. 人工神经网络用于近红外光谱预测汽油辛烷值[J]. 分析科学学报, 2006(01):76-78.
[3] 王娟, 慈林林, 姚康泽. 特征选择方法综述[J]. 计算机工程与科学, 2005, 27(12):68-71.
[4] 吴晓婷, 闫德勤. 数据降维方法分析与研究[J]. 计算机应用研究, 2009, 26(008):2832-2835.
[5] 张振跃, 查宏远. 线性低秩逼近与非线性降维[J]. 中国科学(A 辑:数学), 2005.
[6] 谭璐. 高维数据的降维理论及应用[D]. 国防科学技术大学, 2005.
[7] 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(01):142-146.
[8] 李欣海. 随机森林模型在分类与回归分析中的应用[J]. 应用昆虫学报(昆虫知识), 2013, 50(004):001190-1197.
[9] Ratsch G . Soft Margins for AdaBoost[J]. Machine Learning, 2001, 42(3):287-320.
[10] Collins M , Schapire R E , Singer Y . Logistic Regression, AdaBoost and Bregman Distances[J]. Machine Learning, 2002, 48(1/2/3):253-285.
[11] 王强. 基于 AdaBoost 回归树的电网基建投资模型研究[D]. 2019.
[12] 高琳, 寇鹏, 高峰, et al. 基于分类型损失的 AdaBoost 回归估计算法[J]. 2009.
[13] 李艳芳, 王钰, 李济洪. 几种交叉验证检验的可重复性[J]. 太原师范学院学报:自然科学版, 2013(4):46-49.
[14] 梅益, 杨幸雨. 一种基于 K 折交叉验证法的支持向量机近似模型优化方法:.
[15] 陈卫东 蔡萌林 于诗源. 工程优化方法[M]. 哈尔滨工程大学出版社, 2006.
[16] 谷良贤, 赵育善. 几种实用的工程优化方法[J]. 机械科学与技术, 1992, 000(004):19-23.
[17] 范淼, 李超. Python 机器学习及实践[M]. 清华大学出版社, 2016.
[18] 张凯姣. 基于 Python 机器学习的可视化麻纱质量预测系统[D].
[19] 包研科, 李娜. 数理统计与 MATLAB 数据处理[M]. 东北大学出版社, 2008.
[20] 李强, 赵伟. MATLAB 数据处理与应用[M]. 国防工业出版社, 2001.

附录

附录 1：问题二降维处理用到了低方差降维和高相关降维的方法，其 MATLAB 源代码如下：

```
clear all
data = data_m;

%%
%低方差降维
data_nor = zeros(325,368);
for i = 1:325
    for j = 1:368
    data_nor(i,j)=(data(i,j)-min(data(:,j)))/(max(data(:,j))-min(data(:,j)));
    end
end

variable_var = zeros(1,368);
for i = 1:368
    variable_var(i) = var(data_nor(:,i));
end

for i=[1:9,11:368]
    if variable_var(i)<=0.01
        data(:,i) = zeros(325,1);
    end
end

after_var_dis_count = 0;
var_dis_number = zeros(200,1);
for i=[1:9,11:368]
    if(data(:,i) == zeros(1,325))
        after_var_dis_count = after_var_dis_count + 1;
        var_dis_number(after_var_dis_count) = i;
    end
end

%%
%相关性降维
coff_all = zeros(368,368);
for i = 1:368
    for j = 1:368
        coff_all(i,j) = corr(data(:,i), data(:,j));
    end
end
```

```
for i = [1:9,11:368]
    for j = [1:9,11:368]
        if(abs(coff_all(i,j))>0.6&&i~=j)
            if(abs(coff_all(i,10))>=abs(coff_all(j,10)))
                data(:,j)=zeros(1,325);
                continue;
            elseif(abs(coff_all(i,10))<abs(coff_all(j,10)))
                data(:,i)=zeros(1,325);
                continue;
            end
        end
    end
end

after_coff_count = 0;
after_coff_number = zeros(200,1);
for i=[1:9,11:368]
    if(data(:,i)~= zeros(1,325))
        after_coff_count = after_coff_count + 1;
        after_coff_number(after_coff_count) = i;
    end
end
```

附录 2：问题二体降维处理用到了缺失值比率和随机森林的方法，其 Python 源代码如下：

```
## 缺失值比率
import pandas as pd
data = pd.read_excel("F:\\shumo\\1.xlsx", sheet_name="Sheet1")
null_ratio = data.isnull().sum()/len(data)*100
print(null_ratio)

## 随机森林
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor

datafile = u'F:\\shumo\\1.xlsx'
data = pd.read_excel(datafile)

data_fea = data.iloc[:,1:]#取数据中指标所在的列
data_s = data.iloc[:,0]
```

```
model = RandomForestRegressor(random_state=1, max_depth=10)
data_fea = data_fea.fillna(0)#随机森林只接受数字输入，不接受空值、逻辑值、文字等类型
data_fea=pd.get_dummies(data_fea)
model.fit(data_fea,data_s)

#根据特征的重要性绘制柱状图
features = data_fea.columns
importances = model.feature_importances_
# 因指标太多，选取前 10 个指标作为例子
indices = np.argsort(importances[0:30])

plt.title('Index selection')
plt.barh(range(len(indices)), importances[indices], color='b', align='center')
plt.yticks(range(len(indices)), [features[i] for i in indices])
plt.xlabel('Relative importance of indicators')

plt.show()
```

附录 3：问题二高相关降维结果数据分析时，通过相关性分析热图进行效果分析，其 Python 代码如下：

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor

datafile = u'F:\\shumo\\1.xlsx'
data = pd.read_excel(datafile)
data_fea = data.iloc[:,1:]#取数据中指标所在的列
data_s = data.iloc[:,0]

sns.heatmap(abs(data_fea.corr()), cmap='PuBu')

plt.show()
```

附录 4：问题三 AdaBoost 回归算法的学习的过程，Python 代码如下：

```
import pandas as pd
import numpy as np
from sklearn import metrics
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import learning_curve
```

```
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeRegressor
from sklearn import datasets,ensemble
datafile = u'F:\\shumo\\1.xlsx'
data = pd.read_excel(datafile)

data_fea                                                                    =
data[['v2','v50','v44','v179','v164','v54','v150','v85','v34','v134','v81','v8','
v45','v140','v6','v51','v41','v1','v137','v154','v95','v11','v158','v172','v48','
v7','v166','v5','v99','v49']]#取数据中指标所在的列
data_s = data.iloc[:,0]
data_fea = (data_fea - data_fea.min()) / (data_fea.max() - data_fea.min())

data_test = data.sample(n=None, frac=0.2, replace=True, axis=0)

# 1
fs0 = 'dat/iris_'
print('\n#1 init, fs0,', fs0)

x_train = data_fea
y_train = data_s

x_test                                                                      =
data_test[['v2','v50','v44','v179','v164','v54','v150','v85','v34','v134','v81','
v8','v45','v140','v6','v51','v41','v1','v137','v154','v95','v11','v158','v172','v
48','v7','v166','v5','v99','v49']]#取数据中指标所在的列
y_test = data_test.iloc[:,0]
x_test = (x_test - x_test.min()) / (x_test.max() - x_test.min())

print('\n#2 model')
mx1 = DecisionTreeRegressor(max_depth=4)
mx1.fit(x_train, y_train)
rng = np.random.RandomState(1)
mx2= ensemble.AdaBoostRegressor(DecisionTreeRegressor(max_depth=4),
                        n_estimators=300, random_state=rng)
mx2.fit(x_train, y_train)
mx3= MLPRegressor()
mx3.fit(x_train, y_train)

x_size = range(0,len(x_train))
y_pred1 = mx1.predict(x_train)
y_pred2 = mx2.predict(x_train)
y_pred3 = mx3.predict(x_train)
print(len(x_size))
```

```
print('-'*80)
print(len(y_pred1))
plt.plot(x_size, y_train, c="b",label="samples",linewidth=1)
plt.plot(x_size, y_pred1, c="orange",label="DecisionTreeRegressor", linewidth=1)


plt.xlabel("data")
plt.ylabel("target")
plt.title("Single Decision Tree Regression")
plt.legend(loc="best")
plt.show()
```

附录 5：问题四 325 个样本的各个操作变量的 Python 代码如下：

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn import datasets,ensemble
from sklearn.model_selection import train_test_split

datafile = u'F:\\shumo\\1.xlsx'
data = pd.read_excel(datafile)


X=data[['v1','v5','v6','v7','v8','v2','v11','v34','v41','v44','v45',
'v48','v49','v50','v51','v54','v81','v85','v95','v99','v134','v137',      'v140',
'v150','v154', 'v158', 'v164', 'v166', 'v172', 'v179']]
y = data.iloc[:,0]
regr=ensemble.AdaBoostRegressor()
regr.fit(X,y)
X2 =X

print(data)
print('*'*100)
my_target = regr.predict(X)
print(my_target)

print('*'*100)
target_last = 10
target_new = 0
features_new = []
features_new2 = []
ratio = 0
ratio2222 = 0.1
```

```
# 34
for z in range(0,len(X2)):
    print("now sample is ",z)
    for q in range(30,45,5):
        if ratio >ratio2222:
            break
        # 41
        if q == 35:
            if ratio >ratio2222:
                break
        for w in range(430,1500,200):
            if ratio >ratio2222:
                break
            # 44
            if w ==630:
                if ratio >ratio2222:
                    break
            for e in np.arange(0.35,0.55,0.5):
                # 45
                if ratio >ratio2222:
                    break
                for r in range(350,600,100):
                    # 48
                    if ratio >ratio2222:
                        break
                    for t in range(150,250,50):
                        # 49
                        if ratio >ratio2222:
                            break
                        for y in range(250,900,100):
                            # 50
                            if ratio >ratio2222:
                                break
                            for u in range(250,600,100):
                                # 51
                                if ratio >ratio2222:
                                    break
                                for i in range(3,6500,500):
                                    if ratio >ratio2222:
                                        break
                                    for x in range(0, 7):
                                        features_new.append(X2.iloc[z,x])
```

```
                                features_new.append(q)
                                features_new.append(w)
                                features_new.append(e)
                                features_new.append(r)
                                features_new.append(t)
                                features_new.append(y)
                                features_new.append(u)
                                features_new.append(i)
                                for c in range(15,30):
                                    features_new.append(X2.iloc[z,c])

                                features_new = np.array(features_new)
                                features_new = features_new.reshape(1,-1)
                                target_new = regr.predict(features_new)

                                ratio                                                 =
(data.iloc[z,0]-target_new)/data.iloc[z,0]
                                if ratio > ratio2222:
                                    print('-'*3)
                                    print("final target is :",target_new)
                                    print("final feature is :",features_new)
                                    empty = pd.DataFrame()
                                    kkk = pd.DataFrame(features_new)
                                    print(type(kkk))
                                    empty = pd.concat([empty, kkk])
                                    print("sample index is :", z)
                                    print('-'*3)
                                    features_new = []


                                features_new = []
    if ratio >ratio2222:
        ratio = 0
        print(empty)
        continue

empty.to_excel('F:\\shumo\\output.xlsx')
```

附录 6：问题五求解单个变量对于 RON 的影响的 Python 代码如下：

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn import datasets,ensemble
```

```
from sklearn.model_selection import train_test_split

datafile = u'F:\\shumo\\1.xlsx'
data = pd.read_excel(datafile)
# data=data.apply(lambda x: (x - np.min(x)) / (np.max(x) - np.min(x)))
X_1 = pd.read_excel(u'F:\\shumo\\q51.xlsx').iloc[:,:-1]
X_2 = pd.read_excel(u'F:\\shumo\\q52.xlsx').iloc[:,:-1]
X_3 = pd.read_excel(u'F:\\shumo\\q53.xlsx').iloc[:,:-1]
X_4 = pd.read_excel(u'F:\\shumo\\q54.xlsx').iloc[:,:-1]
X_5 = pd.read_excel(u'F:\\shumo\\q55.xlsx').iloc[:,:-1]
X_6 = pd.read_excel(u'F:\\shumo\\q56.xlsx').iloc[:,:-1]
X_7 = pd.read_excel(u'F:\\shumo\\q57.xlsx').iloc[:,:-1]
X                                   =                                   data[['v2',
'v50','v44','v179','v164','v54','v150','v85','v34','v134','v81','v8','v45','v140',
'v6','v51','v41','v1','v137','v154','v95','v11','v158','v172','v48','v7','v166','
v5','v99','v49']]#取数据中指标所在的列
y = data.iloc[:,0]
print(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

regr=ensemble.AdaBoostRegressor()
# regr=MLPRegressor()
regr.fit(X,y)
## 绘图
fig=plt.figure()
ax=fig.add_subplot(1,1,1)
#     ax.plot(X_1[['v51']],     regr.predict(X_1),marker     =     'o',linewidth=1,
markerfacecolor='r',label='v51',markersize = 5)
#  ax.plot(X_2[['v137']],  regr.predict(X_2),linewidth=1,  label='v137',marker  =
'o',markerfacecolor='r',markersize = 5)
#  ax.plot(X_3[['v134']],  regr.predict(X_3),linewidth=1,  label='v134',marker  =
'o',markerfacecolor='r',markersize = 5)
#   ax.plot(X_4[['v54']],   regr.predict(X_4),linewidth=1,   label='v54',marker   =
'o',markerfacecolor='r',markersize = 5)
#   ax.plot(X_5[['v34']],   regr.predict(X_5),linewidth=1,   label='v34',marker   =
'o',markerfacecolor='r',markersize = 5)
#   ax.plot(X_6[['v45']],   regr.predict(X_6),linewidth=1,   label='v45',marker   =
'o',markerfacecolor='r',markersize = 5)
ax.plot(X_7[['v95']],    regr.predict(X_7),linewidth=1,    label='v95',marker    =
'o',markerfacecolor='r',markersize = 5)


ax.set_xlabel("the value of var")
ax.set_ylabel("RON Loss")
ax.legend(loc="best")
```

```
ax.set_title("RON curve")


plt.show()
```