基于树的影响因素预判与信息挖掘

摘要

肠胃微创手术中需要使用局部的镇静和镇痛药物，现有一种新型药物“R 药”有待非干预性研究。本文基于新型、传统镇静药物在临床试验中的真实表现数据，对 IPI 等生命体征、不良反应和患者满意度进行分析与挖掘。

针对问题一：本文首先对数据进行**清洗、编码与归一化**，接着基于**多变量可视化**分析得出**不同药组关于各不良反应均有显著的差异**，并对不同药组关于各个不良反应分别进行**卡方检验**，得出**不同药组关于术中不良反应均有显著差异，关于术后不良反应中仅“恶心呕吐”和“腹胀腹痛”有显著差异**。关于对不良反应的预判，本文对数据集进行**上采样**，并基于 **K 最近邻**算法建立模型，经测试模型在数据集上的**分类 AUC 正确率均在 0.92 以上**，并作出**混淆矩阵**和 **ROC 图**对模型的具体测试情况进行可视化。

针对问题二：本文首先对数据集进行预处理，接着基于**主成分分析法**在**将数据降维至 15 维**。然后对 15 个主成分进行正态性检验，并分别对主成分进行**独立样本 T 检验**和 **Mann-Whitney U 检验，不同药组仅关于第六个主成分有显著差异**。关于探究造成显著差异的原因，本文以第六主成分为标签、基于 **LightGBM** 建立回归模型，在模型性能优良的前提下进行**树模型的特征重要性评价**，最终认为**造成显著差异的因素由大到小依次是样本的年龄、体重、体重、镇静药名称**。

针对问题三：本文首先进行数据清洗以及特征的选择、缩放、编码，接着类似于问题二对数据进行降维，以避免分类特征造成数据集特征离散。然后分别以六个时间点的 IPI 维标签、基于**岭回归**和**支持向量机**建立回归模型，并通过**加权平均法**对二者进行融合。最后通过 **MAE** 和 **MSE** 对三个模型进行评价，并基于数据扰动对模型进行**灵敏度分析**，结果显示各标签的模型在测试中的 MAE 和 MSE 基本分布在 0.1 到 0.3，模型在经过 30% 内的数据扰动后仍保持测试的 MSE 稳定，故知**模型具有良好的性能与稳定性**。

针对问题四：本文首先基于**斯皮尔曼和肯德尔相关系数**分别对数值特征和分类特征进行相关性分析，得出镇痛药总剂量、moaasjinjing、镇静药的名称、镇静药诱导剂量、镇静药总剂量与术后满意度有相对较大的关联。关于术后满意度具体定量关系的挖掘，本文对数据集进行上采样，并基于**随机森林**建立分类模型，以**准确率、召回率、F1_score 和支持率**评价其在测试集上的性能，结果显示模型**分类准确率为 0.75**。最后导出随机森林模型的**树状图**，并找出模型划分类别的标准，其中涉及 petco2025、镇静药总剂量等，本文**基于树状图绘制一个流程图用于表达其间的定量关系**，此标准即为题设所要求的部分特征与术后满意度的联系。

关键词：主成分分析法；LightGBM；加权平均法；灵敏度分析；随机森林

目录

1 问题重述

1.1 问题背景

新药物研究是临床研究中的关键环节。在肠胃微创手术中，往往需要使用局部的镇静和镇痛药物，传统的镇静药物为“B 药”，某药物研发中心研发了一种新型药物“R 药”，新药物投入使用，通常需要经历生物试验和临床试验两个阶段。

为了解新药物的药性特征，要研究分析病患在术中、术后的不良反应；病患术后的生命体征以及病患的满意度及其相关问题，根据实验数据的分析，对以上方面做出预估，为药物选择提供重要参考，为医师病患提供预判依据。

1.2 问题提出

本题附件 1 中收集了新型药物和传统镇静药物在临床试验中的表现数据，数据包括患者基本信息、术中用药、术中患者身体状态和术后患者反馈信息等，根据附件所给数据信息建立数学模型，解决一下问题：

问题 1：首先根据术中、术后 24 小时不良反应，判断新药组和原药组是否存在显著差异；接着根据患者基本信息和镇静药物种类建立预判患者术中、术后 24 小时是否会出现不良反应的数学模型。

问题 2：首先判断新药组和原药组在生命体征数据方面是否表现出显著差异；若有显著差异，通过模型挖掘造成生命体征数据有显著差异的影响因素。

问题 3：用药后 3 分钟内的 IPI 数据在临床研究具有很大的参考意义，要求根据用药信息和患者信息预测给药后 3 分钟以内的 IPI 数据。

问题 4：要求基于现有数据找出影响术后满意度的因素，并分析影响因素与术后满意度之间的联系。

2 问题分析

本文对于赛题题设，从多个角度对数据集进行统计分析与统计推断，本部分将对四个问题进行简要分析，同时赛题的每个问题的解答完整流程以流程图的形式在附录 A 给出。

2.1 问题一的分析

针对问题一，第一问要求关于术中、术后 24h 不良反应，判断新药组和原有药物组是否存在显著差异。首先对附件 1 中进行数据清洗、特征缩放、特征编码，由于不良反应在编码后均为分类特征，故用基于多变量可视化分析的定性方法和基于卡方检验的定量方法探究不同药组对于不良反应的差异性。

第二问要求建立一个有效的分类模型用于对患者的不良反应进行预判。本文基于经过数据预处理的数据集，首先对标签进行分布分析，并对标签比例严重失衡的数据集进行上采样处理。接着把数据按比例分为训练集和测试集。然后基于决策树对训练集进行特征提取，使用经过训练的决策树模型实现对两大类、8 组不良反应的预判。最后基于混淆矩阵、ROC 图对模型在测试集上的表现进行评价。

2.2 问题二的分析

针对问题二，第一问要求判断新药组和原有药物组在生命体征数据方面是否表现出显著差异。首先对数据进行数据清洗和特征编码。接着基于 Spearman、Kendall 相关性分析计算特征之间的相关性并删除相关性高特征之一，为避免高维数据集造成数据分析困难，基于主成分分析法在保留较高方差解释比例的前提下对数据进行降维。最后对降维后的特征进行正态性检验，分别基于独立样本 t 检验和 Mann-Whitney U 研究不同药组对生命体征数据的差异性。

第二问在第一问的基础上，探究造成新药对生命体征数据显著差异的影响因素。本文基于经过数据预处理的数据集，基于 LightGBM 对以受试者情况和病史为特征空间、以检验出显著差异的生命特征组为标签的数据集进行特征提取，在确保模型性能极好的前提下并利用树模型独有的特征重要性评价功能对各特征对标签的作用效果进行评价，最后利用条形统计图呈现。

2.3 问题三的分析

针对问题三，要求根据用药信息和患者信息预测对给药后 3 分钟以内的 IPI 数据。首先对数据进行数据清洗和特征编码。接着为了避免大量分类特征造成特征空间离散从而影响模型提取特征，基于主成分分析法对特征空间进行降维。然后基于岭回归和支持向量机回归分别对数据集进行特征提取，并用线性加权法对二者结果进行融合。最后为了提高模型的可靠性和准确性，本文基于 MAE、MSE 和数据扰动对模型在测试集上的表现和回归的稳定性进行评价。

2.4 问题四的分析

针对问题四，要求基于现有数据找出与术后满意度有关的因素，本文使用定性方法和定量方法分别进行模型求解。

基于斯皮尔曼、肯德尔相关性分析分别对数值特征和分类特征与术后满意度的相关性进行探究。首先对数据进行数据清洗，对分类特征进行特征编码，对数值特征进行特征缩放。接着对两类特征分别进行相关性分析，求得相关系数。最后用条形统计图对结果进行可视化，得出初步结论。

基于树模型的树状图对术后满意度划分的定量探究。首先对数据进行预处理，并将五维关于术后满意度的特征聚合为一维关于术后满意度的评分作为标签，为避免术后满

意度评分分布不均衡导致模型难以提取特征，对数据集进行上采样。然后按比例将数据集划分为训练集和测试集，并基于随机森林对训练集进行特征提取。最后导出随机森林的树状图，并通过树状图获取分类的具体依据。

3 模型假设

1. 假设本文的数据真实可靠。

2. 假设患者除了使用本题所用药物外，未使用其他药物。

3. 假设患者出现的不良反应是除了环境等客观因素造成的。

4. 本文认为所给的数据不存在过大的人为过失。

5. 假设患者在用药前并未吃其他食物。

6. 本文认为药物在存放过程中成分没有发生改变。

4 符号说明

符号	含义
χ^2	卡方检验的卡方值
O_{ij}	第 i 个样本的第 j 种特征存在数量
E_{ij}	第 i 个样本的第 j 种特征不存在数量
df	特征的自由度
p	样本的 p 值
χ	特征空间
$\overrightarrow{x}_i$	第 i 个样本的特征向量
$L_3(\cdot)$	计算闵可夫斯基距离
c_j	第 j 个分类指标
d_i	第 i 个样本的等级
$H(\overrightarrow{x})$	组合模预测型向量函数
$error_{ij}$	第 i 个特征的第 j 个样本扰动值

5 问题一的模型建立与求解

5.1 数据预处理

5.1.1 数据清洗

为了提高数据的质量，便于更好的进行数据分析，对有缺失值和异常值进行剔除或者插值，通过对附件一的数据进行观察，发现有缺失值，数值特征用均值来填补缺失值。

5.1.2 特征缩放

对于一些需要输入特征进行计算的模型，特征放缩可以让模型更加准确的进行分类或回归预测。在对预判对患者术中、术后 24h 的会出现的不良反应中，为了提高 KNN 模型预测的准确性，先删除与之无关的数据，由于部分特征比例的差别过大，所以先对“从未抽烟”、“偶尔抽烟：每天吸卷烟超过四次”和“经常抽烟：每天吸卷烟一支以上”三个特征聚为“抽烟”和“不抽烟”两类特征。

5.1.3 特征编码

为了便于分析，把分类数据转化为数字数据，本文主要采用独热编码把分类变量转化为数值变量，然后对数值型变量进行归一化，使不同特征之间的值具有可比性，将一些二元分类变量进行二值化（映射为 0 和 1），以便后续建模。对于附件 1 中的数据先提取出术中不良反应（咳嗽、体动、术中其他）的数据和术后 24 小时不良反应的数据（恶心呕吐、嗜睡乏力、头昏头晕头痛、腹胀腹痛、其他不舒服），把出现该情况的数据标为“1”，无的标为“0”。

5.1.4 数据的上采样

对附件一中的数据进行数据可视化发现术中和术后 24 小时未出现不良反应的数量远远超于出现不良反应的数量，故采用 imblearn 库中 RandomOverSampler 函数来进行数据上采样，通过增加少数类样本的数量来平衡数据集中的类别分布。

5.2 药组对于不良反应的差异性探究

依据术中和术后 24h 不良反应，为判断新药组和原有药物组是否存在显著差异，接下来本文主要从定性（数据可视化）和定量（卡方检验）两个方面来进行分析。

5.2.1 基于可视化的多变量分布分析

我们先 python 中的 Seaborn 库对术中不良反应的数据进行数据可视化，来观察在不同镇静药（“R”药和“B”药）使用情况下，手术期间患者出现咳嗽、体动以及其他术中不适症状的情况，得到结果如下图：

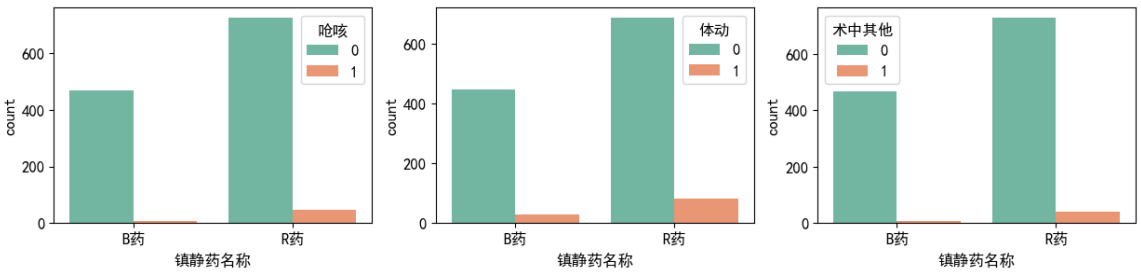


图 1 术中是否出现不良反应数据可视化图

分析图 1 中的术中是否出现三类不良反应的直方图，可以发现从数量上 B 药出现呛咳、体动、术中其他不良反应的数量比 R 药少，尤其是体动方面的不良反应在直观上两要差异最大，其他两种不良反应中 B 药几乎不出现不良反应，可以看到术中使用新药容易出现不良反应。

接下来利用和以上一样的方式对于手术后 24 小时的不良的反应 (恶心呕吐、嗜睡乏力、头昏头晕头痛、腹胀腹痛、其他不舒服) 的数据做出数据可视化图来进一步分析新药组和原有药组之间是否存在差异，得到数据可视化图如下所示：

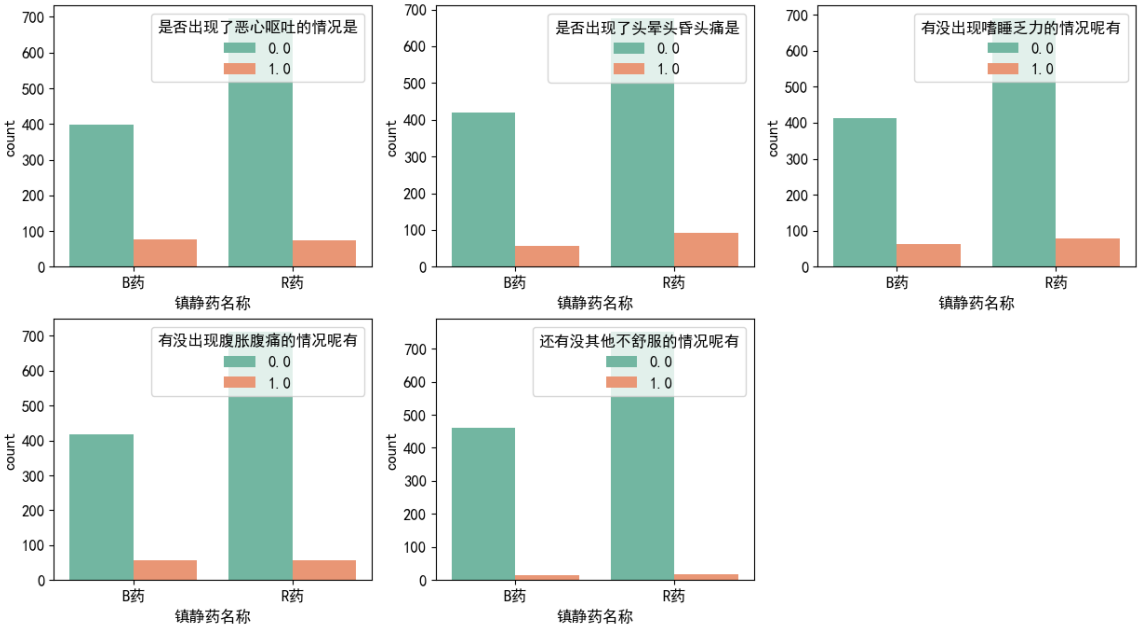


图 2 手术后是否出现不良反应数据可视化图

分析图 2 的术后是否出现三类不良反应的直方图，可以发现从数量上，B 药出现恶心呕吐、嗜睡乏力、腹胀腹痛、其他不舒服的不良反应数量基本与 R 药持平，只有出现头晕头昏头痛的不良反应的数量上少于 R 药。从比例上看，R 药的基数大于 B 药，但是出现恶心呕吐、嗜睡乏力、腹胀腹痛、其他不舒服的不良反应数量却与 B 药持平，可以出在术后使用 R 药不容易出现不良反应。

5.2.2 基于卡方检验的定量差异探究

接下来利用卡方检验分别对手术期间和手术 24 小时后的出现不良反应来定量探究两组药物之间是否具有显著差异，对于分类问题可以列出交叉表，发现总样本大于 40 且每个类别的理论频数大于 5 可用卡方检验：

1. 原假设 H_0：两种药没有显著差异；

2. 备择假设 H_1：两种药有显著差异；

首先计算检验统计量：

$$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{k}\frac{O_{ij} - E_{ij}}{E_{ij}}. \tag{1}$$

接着计算自由度：

$$df = (m-1) \times (k-1). \tag{2}$$

利用卡方分布的累积分布计算值：

$$p = 1 - F\left(\chi^2, df\right). \tag{3}$$

若检验统计量大于临界值（p 值大于显著性水平 0.05），不能拒绝原假设，两种药没有显著差异；若检验统计量小于临界值（p 值小于显著性水平 0.05），拒绝原假设，两种药有显著差异。

表 1 卡方检验结果

名称	卡方值	p 值	自由度
呛咳	13.514691	0.000236	1
体动	7.278738	0.006977	1
术中其他不良反应	12.815257	0.000343	1
恶心呕吐	11.929730	0.000552	1
嗜睡乏力	0.003899	0.950210	1
头昏头晕头痛	2.333728	0.126598	1
腹胀腹痛	7.536825	0.006045	1
其他不舒服	0.226608	0.634050	1

由表可知，“咳嗽”、“体动”以及“术中其他反应”的卡方检验结果的 p 值均小于 0.05，故认为在术中不良反应关于药物类别具有显著差异；“是否出现恶心呕吐”和“是否出现腹胀腹痛”的情况的卡方检验的结果 p 值小于 0.05，两药物之间有显著差异；“是

否出现头晕头昏头痛”、“是否有嗜睡乏力情况”、“是否有其他不舒服” 情况的 p 值均大于 0.05，故其关于药物类别无显著差异。

5.3 基于 KNN 的不良反应预判

5.3.1 模型建立

题目要求根据患者基本信息和镇静药物种类，对患者术中、术后 24h 的不良反应进行预判。首先先对手术期间和手术后是否产生不良反应进行数据可视化，得到结果如下图所示：

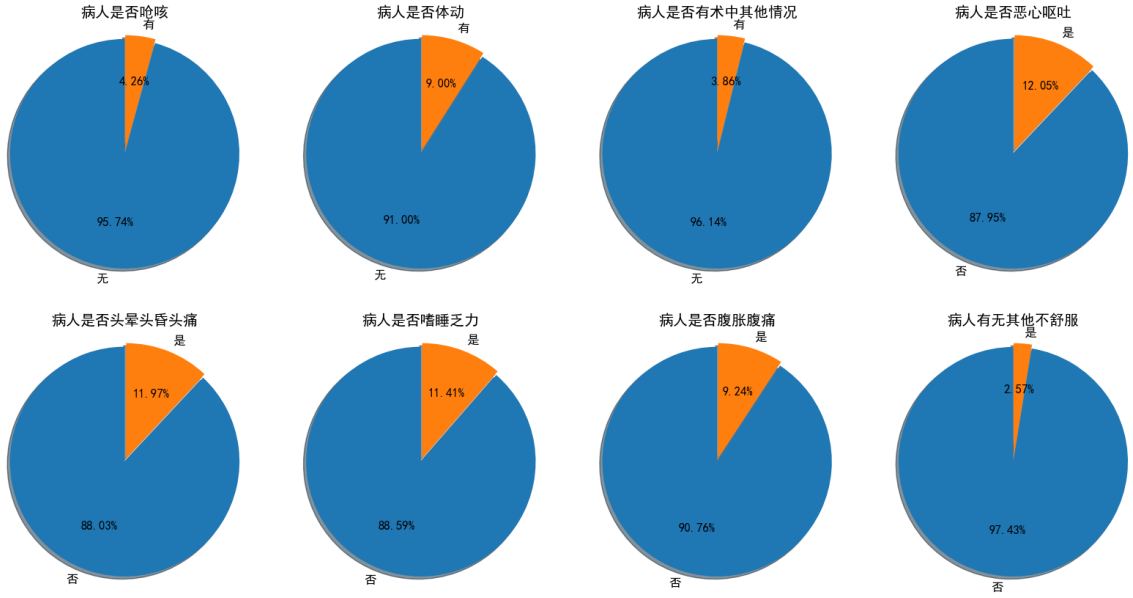


图 3 对不良反应的分布可视化

由上图可知，未出现呛咳、体动、术中其他、恶心呕吐、头晕头昏头痛、嗜睡乏力、腹胀腹痛、其他不舒服的的占比均超过 87%，即术中和术后未出现不良反应远远超过出现不良反应的数量，针对此类情况，对数据进行上采样。设各标签少数类样本数量为 k，则有：

$$P(x|y=1,D_{up})=\frac{\sum_{i=1}^{k}[x_i=x]}{k}. \tag{4}$$

其中 x_i 表示第 i 个采样得到的少数类样本的特征。再将上采样后的数据集划分为训练集和测试集，建立 KNN 模型对训练集进行拟合，再使用测试集来评估模型的性能。

近邻法使用的模型主要有三个基本要素——距离度量，$L_3\left(\vec{x}_i,\vec{x}_j\right)$ 值的选择和分类决策规则决定. 近邻法中，当训练集、距离度量、$L_3\left(\vec{x}_i,\vec{x}_j\right)$ 值及分类决策规则确定后，对于任何一个新的输入实例，它所属的类唯一地确定. 这相当于根据上述要素将特征空间划分为一些子空间，确定子空间里每个点所属的类.

1. **度量距离**

本文使用的主要是闵可夫斯基（Minkowski）距离. 原理如下：设特征空间 χ 是 n 维实数向量空间 $R^n, \overrightarrow{x}_i, \overrightarrow{x}_j \in \chi$.

$$\overrightarrow{x}_i = \left(x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(n)}\right)^T, \overrightarrow{x}_j = \left(x_j^{(1)}, x_j^{(2)}, \cdots, x_j^{(n)}\right)^T, \tag{5}$$

$\overrightarrow{x}_i, \overrightarrow{x}_j$ 的距离定义为：

$$L_3\left(x_i, x_j\right) = \left(\sum_{l=1}^{n} \left|x_i^{(l)}, x_j^{(l)}\right|^3\right)^{\frac{1}{3}}. \tag{6}$$

2. k **的取值**

如果选择的 k 值较小，模型“学习”的近似误差就会减小，但“学习”的估计误差会增大.k 值的减小就意味着整体模型变得复杂，容易发生过拟合. 如果选择的 k 值较大，则相反，一般使用交叉验证确定 k 值。

3. **分类决策规则**

k 近邻法中分类决策规则往往是多数表决，即由输入实例的 k 个邻近的训练实例中的多数类决定输入实例的类. 分类函数为：

$$f : R^n \to \left\{c_1, c_2, \cdots, c_K\right\}. \tag{7}$$

那么误分类的概率是：

$$P\left(Y \neq f\left(X\right)\right) = 1 - P\left(Y \equiv f\left(X\right)\right). \tag{8}$$

对给定的实例 $\overrightarrow{x} \in \chi$，其最近邻的 k 个训练实例点构成的集合 $N_k\left(\overrightarrow{x}\right)$。如果涵盖 $N_k\left(x\right)$ 的区域的类别是 c_j，那么误分类概率是：

$$\frac{1}{k} \sum_{\overrightarrow{x}_i \in N_k\left(\overrightarrow{x}\right)} I\left(y_i \neq c_j\right) = 1 - \frac{1}{k} \sum_{\overrightarrow{x}_i \in N_k\left(\overrightarrow{x}\right)} I\left(y_i = c_j\right). \tag{9}$$

要使误分类率最小即经验风险最小，就要使 $\sum\limits_{\overrightarrow{x}_i \in N_k\left(\overrightarrow{x}\right)} I\left(y_i = c_j\right)$ 最大，所以多数表决规则等价于经验风险最小化。

4. **KNN 的实现—kd 树**

更优化处理一般使用 kd 树。为了更高效地找到 k 个近邻样本，使用 k-d 树来组织训练数据集。kd 树是一种二叉树结构，每个节点代表一个训练样本。通过对训

练样本递归地划分空间，可以构建出一棵 kd 树。在查询测试样本的 k 个近邻时，可以通过遍历 k-d 树来快速找到距离测试样本最近的 k 个训练样本。

因此，k-d 树可以被看作是 k 近邻法模型中实现近邻搜索的数据结构，它可以大大提高模型的预测效率。为了进一步提高模型的准确率，使基于 GridSearchCV 实现带交叉验证的网格搜索对 KNN 模型进行超参数优化。参数选择与最终结果为：

表 2 GridSearchCV 对 KNN 模型调参结果

参数	取值范围	最优结果
algorithm	[auto, ball_tree, kd_tree, brute]	auto
leaf_size	np.arange(10,51,10)	10
weights	[uniform, distance]	uniform
n_neighbors	np.arange(1,11,1)	1
p	[1,2,3]	3

5.3.2 模型求解

基于 Scikit-Learn 库按上文最有参数初始化模型，分别以“呛咳”、“体动”、“术中其他不良反应”、“恶心呕吐”、“嗜睡乏力”、“头昏头晕头痛”、“腹胀腹痛”、“术后其他不舒服”八个为标签生成数据集，按 4：1 的比例划分数据集，并使用训练集对初始化的 KNN 模型进行训练。使用 KNN 模型对测试集的特征空间进行预测，并使用分类器常见评价指标进行评价，以此来判断此模型是否可以用来根据患者基本信息和镇静药物种类，对患者术中、术后 24 小时的不良反应进行预判，得到结果如下表：

表 3 KNN 测试结果评价

标签名称	测试选项	准确率	召回率	F1 分数	支持率
呛咳	无	1.00	0.93	0.96	247
	有	0.93	1.00	0.96	230
体动	无	1.00	0.92	0.96	224
	有	0.93	1.00	0.96	228
术中其他不良反应	无	1.00	0.96	0.98	250
	有	0.93	1.00	0.96	230
恶心呕吐	无	0.98	0.88	0.93	228
	有	0.88	0.98	0.93	210
嗜睡乏力	无	0.99	0.86	0.92	232
	有	0.86	0.99	0.92	207
头昏头晕头痛	无	1.00	0.87	0.93	221
	有	0.88	1.00	0.94	221
腹胀腹痛	无	1.00	0.92	0.96	224
	有	0.93	1.00	0.96	228
术后其他不舒服	无	1.00	0.97	0.99	245
	有	0.97	1.00	0.99	241

为了让结果更直观，本文基于 Scikit-Learn 将测试结果利用混淆矩阵和 ROC 图可视化，并计算出 KNN 在测试集上预测的 AUC 值，以“术后有无其他不舒服”为例的可视化结果如下所示：

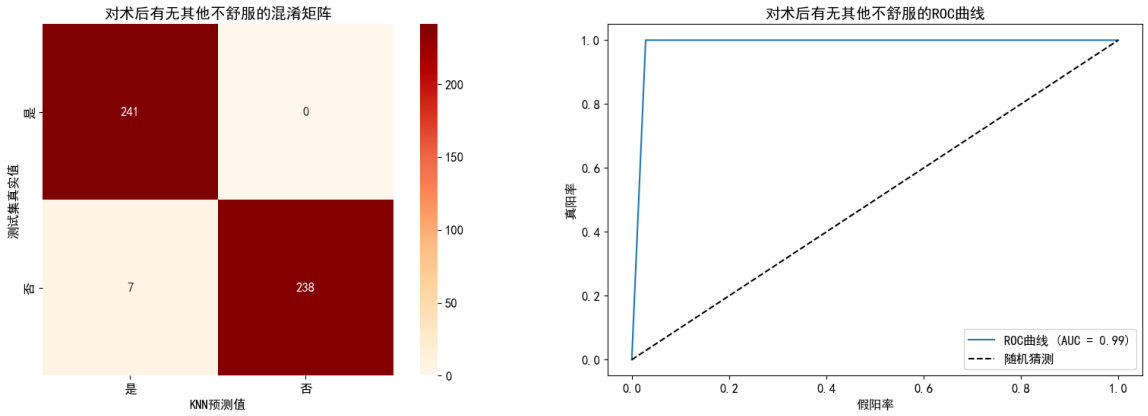


图 4 术后有无其他不舒服热力图和 ROC 曲线

由上图可以看到对角线上的数字相对较大，说明模型的分类效果比较好，但是对于“否”类别的预测效果略微不如“是”类别。另外，热力图的颜色也提供了直观的参考，颜色越深表示数量越多，可以看到”是”类别的预测结果相对比较准确，因此相应的颜

色也比较深；ROC 曲线看真阳率高，综上所述，KNN 模型性能良好。完整的 KNN 测试结果的混淆矩阵、ROC 图详见附录 A。基于测试结果的评价，本文认为此模型可用来预判术中和术后 24 小时的不良反应出现情况。

6 问题二的建模与求解

6.1 数据预处理

与问题一类似地，对于样本缺失值过多的特征 (超过 80%) 进行剔除，共剔除了 20 个特征列，同时对于处理后的数据集中存在缺失值的样本进行剔除共剔除了 547 行。对于余下的样本的 92 个特征进一步的处理，筛选出更有代表性特征。

6.1.1 以相关性筛除特征

对于处理后的生命体征指标，筛选出相关性较大的特征，将它们关联为一种特征能够极大的减少模型中的冗余信息和噪声，删除较大相关性的特征可以帮助提高下面建立模型的泛化能力和解释能力。以相关性来筛除特征，首先需对数据集进行归一化处理，为了避免数值问题，利用 min-max 标准化方法对数据采用 min-max 标准化进行处理，定义数据为：$Z = \{z_1, z_2, \cdots, z_n\}$，min-max 标准化的公式为：

$$z_i^{'} = \frac{z_i - z_{\min}}{z_{\max} - z_{\min}}. \tag{10}$$

处理好的数据表示为：$Z^{'} = \{z_1^{'}, z_2^{'}, \cdots, z_n^{'}\}$。斯皮尔曼相关系数可以根据以下公式进行计算：

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}. \tag{11}$$

其中 n 为样本容量，ρ 为相关系数，d_i 为样本的两个变量中对应的数据次序的等级差（等级差指的是将样本按从小到大排序后规定等级，如果数值相同，则将它们所在的位置取算术平均）。计算不同特征之间的相关系数，筛除掉高度相关的特征，以减少数据集中的冗余信息和降低模型过拟合的风险。本文基于 Pandas 的 corr() 函数来计算相关系数，并认为相关系数的绝对值大于 0.7 是高度相关的特征，计算各个特征之间的相关性这里删除相关性大于 0.7 的特征列（通过相似性处理最终得到 49 个特征代表整个生命体征，总共删除了 43 个特征）。

6.1.2 基于主成分分析法的数据降维

上面虽然得到 49 个生命体征，但是对于分析不同药组之间的具体的显著性差异来说，生命体征特征仍然偏多会导致分析难度大，不利于准确的分析出其中的差异性，需

要对数据特征进行进一步的降维处理，基于主成分分析法的数据降维是一种常见的数据降维技术，通过将多个相关性高的变量合并成少数几个不相关的变量（即主成分），从而降低数据维度，提高数据处理和分析的效率。

本文采用 PCA 降维方法，通过线性变换将原始数据映射到新的空间中，从而得到降维后的数据。在这个新的空间中，原始数据中的冗余信息被消除，只保留了最重要的特征。设筛选后的数据构成一个 $\overrightarrow{x}=(x_1,x_2,\cdots,x_p)^T$ 为一个 p 维随机变量，并假定二阶矩阵存在，记均值向量为：

$$\overrightarrow{\mu}=E\left(\overrightarrow{x}\right). \tag{12}$$

协方差矩阵为：

$$\sum=V\left(\overrightarrow{x}\right). \tag{13}$$

下式进行线性变换：

$$\begin{cases} y_1=a_{11}x_1+a_{12}x_2+\cdots+a_{1p}x_p=\overrightarrow{a}_1^T\overrightarrow{x} \\ y_2=a_{21}x_1+a_{22}x_2+\cdots+a_{2p}x_p=\overrightarrow{a}_2^T\overrightarrow{x} \\ \cdots \\ y_p=a_{p1}x_1+a_{p2}x_2+\cdots+a_{pp}x_p=\overrightarrow{a}_p^T\overrightarrow{x} \end{cases} \tag{14}$$

约束条件有：

1. $\overrightarrow{a}_i^T\overrightarrow{a}=a_{1i}^2+a_{2i}^2+\cdots+a_{pi}^2=1\,(i=1,2,\cdots,p)$
2. 当时 $i>1$，$\operatorname{cov}\left(y_i,y_j\right)=0\,(j=1,2,\cdots,i-1)$，即 y_i 与 y_j 不相关。；
3. $\operatorname{var}\left(y_i\right)=\max\limits_{\overrightarrow{a}^T\overrightarrow{a}=1,\operatorname{cov}(y_i,y_j)=0}\operatorname{var}\left(\overrightarrow{a}^T\overrightarrow{x}\right)\,(j=1,2,\cdots,i-1)$

设 $\lambda_1,\lambda_2,\cdots,\lambda_p\,(\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_p\geq0)$ 为 $\sum$ 的特征值，$\overrightarrow{t}_1,\overrightarrow{t}_2,\cdots,\overrightarrow{t}_p$ 为相应的一组正交单位特征向量，$\overrightarrow{x}=(x_1,x_2,\cdots,x_p)^T$ 的主成分就是以 $\sum$ 的特征向量为系数的线性组合，它们互不相关，线性组合的方差为 $\sum$ 的特征值。

当 $\overrightarrow{a}_1=\overrightarrow{t}_1$ 时，$V\left(y_1\right)=\overrightarrow{a}_1^T\sum\overrightarrow{a}_1=\lambda_1$ 达到最大值，所求的 $y_1=\overrightarrow{t}_1^T\overrightarrow{x}$ 就是第一主成分。如果第一主成分所含的信息不够多，不足以代表原始的 p 个向量，则需要再考虑使用 y_2。为了使 y_2 所含的信息与 y_1 不重叠，要求 $\operatorname{cov}\left(y_i,y_j\right)=0$。当 $\overrightarrow{a}_2=\overrightarrow{t}_2$ 时，$V\left(y_2\right)=\overrightarrow{a}_2^T\sum\overrightarrow{a}_2=\lambda_2$ 达到最大值，所求的 $y_2=\overrightarrow{t}_2^T\overrightarrow{x}$ 就是第二主成分。与此类似，可以再定义第三主成分，直至第 p 主成分. 一般来说，$\overrightarrow{x}$ 的第 i 主成分是指约束条件下的 $y_i=\overrightarrow{t}_i^T\overrightarrow{x}$。

记 $\overrightarrow{y}=(y_1,y_2,\cdots,y_p)^T$，主成分向量 $\overrightarrow{y}$ 与原始向量 $\overrightarrow{x}$ 的关系为 $\overrightarrow{y}=T^T\overrightarrow{x}$，其中 $\overrightarrow{T}=\left(\overrightarrow{t}_1,\overrightarrow{t}_2,\cdots,\overrightarrow{t}_p\right)^T$。

第 i 主成分在总方差 $\sum_{i=1}^{p}\lambda_i$ 中的比例 $\lambda_i/\sum_{i=1}^{p}\lambda_i$ 称为主成分的贡献率，第一主成分的贡献率最大，表明它解释原始变量的能力最强，$y_2 \sim y_p$ 的解释能力依次减弱. 主成分分析的目的在于减少变量的个数，因而一般不会使用所有 p 个主成分，忽略一些带有较小方差的主成分不会给总方差带来太大的影响。

前 m 个主成分的贡献率之和在总方差中的比例 $\sum_{i=1}^{m}\lambda_i/\sum_{i=1}^{p}\lambda_i$ 称为主成分 $y_1, y_2, \cdots, y_m$ 的累计贡献率，它表明了 $y_1, y_2, \cdots, y_m$ 解释原始变量的能力。通常去较小（相对于 p）的 m，可使得累计贡献率达到一个较高的百分比，此时 $y_1, y_2, \cdots, y_m$ 可替代 $\sum$，从而达到降维的目的，而信息的损失不多。

PCA 通过构建新的主成分来实现降维，本文基于 Scikit-Learn 库的 PCA 函数实现主成分分析法。首先初始化 pca 模型，并通过参数 n_components 指定降维后的特征维度，当前特征空间共 49 个列向量。接着调用 fit_transform 方法对原始数据进行降维，并基于 matplotlib 绘制主成分方差解释比例图：

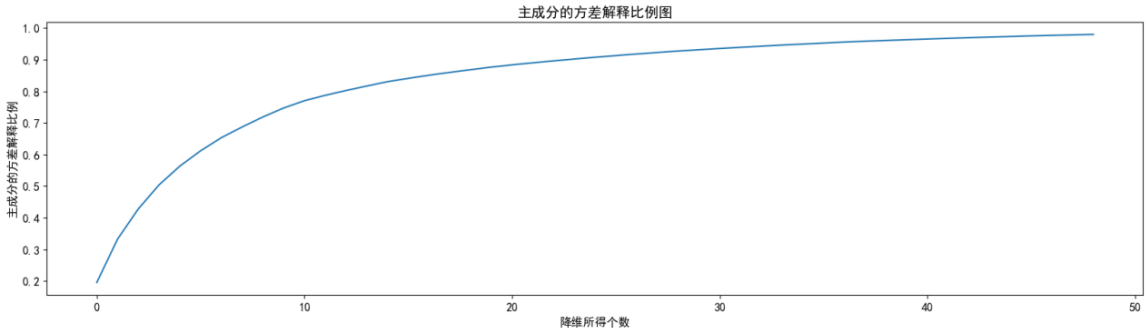


图 5 主成分方差解释比例图

该图显示了每个主成分的方差解释比例。为了使得到的主成分能够较为完整的表示原信息，本文基于图中信息选择将 49 个特征降维成 15 个主成分——显然降维为 15 个主成分时可成分解释比例超过 0.8，丢失的信息较少，故使用这 15 个主成分特征能够较好反映生命指标数据的信息。

由于降维后的数据不再表示原生命体征的意义，其本身不再具有具体的实际意义，却蕴含原生命指标的信息。同时处理得到的主成分不再具有原始数据的列名，因此重新给它们分配了新的列名，将这前 15 个主成分依次命名 $\{1, 2, 3, \cdots, 15\}$，同时与镇静药名称、性别、年龄、身高、体重、是否吸烟、是否酗酒、有无 PONV、有无晕动史 11 个特征构成含有新的数据集。这里欲全面科学的探究出不同药组的差异性情况，因此考虑了患者其他的情况，不失一般性。下面利用这 15 个能够代表原生命体征的主成分直接与不同的药组进行进一步的差异性分析。

6.2 药组对于生命体征数据的差异性探究

利用上面 15 个主成分来研究新药组和原有药物组在生命体征数据方面是否表现出显著差异，本文通过独立样本 t 检验、Mann-Whitney U 检验来研究药组对于生命体征

数据的 15 个主成分是否具有显著性，进而得出结论。

6.2.1 正态性检验

欲进行独立样本的假设检验，需先进行正态性检验。依题意，有：

1. 原假设 H_0：样本数据服从正态分布；

2. 备择假设 H_1：样本数据不服从正态分布；

计算了一个统计量：

$$K^2 = S^2 + V^2. \tag{15}$$

其中，S 是样本偏度，V 是样本峰度.

对基于主成分分析降维出的 15 个主成分特征进行正态性检验，基于 Scipy 库中的 normaltest 函数进行正态性检验。对这 15 个主成分进行正态性检验，通过其返回值 p 值表示数据是否符合正态分布，p 小于 0.05，则拒绝原假设，认为数据不符合正态分布，只能进行 Mann-Whitney U 检验；否则接受原假设，认为数据符合正态分布，可以进行独立样本 t 检验。正态性检验的结果如下表：

表 4 正态性检验结果

正态分布	非正态分布
5、6、7、9、12	1、2、3、4、8、10、11、13、14、15

从上表可以清楚的得到满足正态分布能够进行独立样本检验的主成分为第 5、6、7、9、12 主成分，不满足正态分布进行非参数检验的是第 1、2、3、4、8、10、11、13、14、15。

6.2.2 独立样本的参数、非参数检验

基于以上正态检验的结果对这 15 个特征分别进行参数或非参数检验。对第 5、6、7、9、12 主成分分别进行独立样本 T 检验，先将标签数据根据是否使用镇静药分为两组，然后对这两组数据进行独立样本 t 检验。t 检验用于判断两组数据之间的均值是否存在显著性差异，独立样本 T 检验的原理是基于 T 统计量，利用和 t 值和 p 值来判断这两个样本是否显著不同。

1. 原假设 H_0：样本数据之间没有显著差异性；

2. 备择假设 H_1：样本数据之间有显著差异性.

计算公式如下：

$$t = \frac{\overline{x}_1 - \overline{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}. \tag{16}$$

$$p = 2\left(1 - t_{n_1+n_2-2}(|t|)\right). \tag{17}$$

其中，$\overline{x}_1$ 和 $\overline{x}_2$ 分别表示每个两特征组样本的均值，s_1 和 s_2 分别为两组样本的标准差，n_1 和 n_2 分别为两组样本的样本量，t 为检验统计量。$t_{n_1+n_2-2}(|t|)$ 表示 t 分布的累积分布函数，$|t|$ 表示 t 值的绝对值。

对第 1 、2 、3 、4 、8 、10 、11 、13 、14 、15 主成分用 Mann-Whitney U 进行非参数检验，其原假设为两组独立样本的总体分布相同，备择假设为两组独立样本的总体分布不同。Mann-Whitney U 检验的统计量为 U 值，其 U 值和 p 值其计算公式为：

$$U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1. \tag{18}$$

$$U_2 = n_1 n_2 + \frac{n_2(n_2+1)}{2} - R_2. \tag{19}$$

$$z = \frac{U - \mu_U}{\sigma_U}. \tag{20}$$

$$\mu_U = \frac{n_1 n_2}{2}. \tag{21}$$

$$\sigma_U = \sqrt{\frac{n_1 n_2(n_1+n_2+1)}{12}}. \tag{22}$$

$$p = 2[1 - \Phi(|z|)] \tag{23}$$

其中，n_1 和 n_2 分别为两组样本的样本容量，R_1 和 R_2 分别为两组样本的秩和，μ_U 和 σ_U 分别为 U 的均值和标准差，z 值是标准正态分布的分位数，Φ 表示标准正态分布的累积分布函数，$|z|$ 表示 z 值的绝对值。

本文对于独立样本 t 检验，若 p 值显著小于 0.05，说明两个样本的分布不同，即两组样本存在显著差异。最终检验结果如下表所示：

表 5 独立样本检验的结果

主成分序号	统计量	p	主成分序号	统计量	p
1	5677.0	0.219	9	1.4358	0.152
2	6047.0	0.503	10	5781.0	0.284
3	7725.0	0.063	11	6725.0	0.727
4	5428.0	0.109	12	-0.910	0.363
5	1.31	0.191	13	5944.0	0.409
6	2.313	0.021	14	6118.0	0.573
7	-0.636	0.525	15	7177.0	0.303
8	5658.0	0.209			

根据上表数据显示，第六个主成分的检验所得 p 值为 $0.021(< 0.05)$，故不同药品关于第六个主成分特征具有显著差异。进一步地，接下来将探究造成对于不同药品，生命体征数据的第六个主成分特征存在显著差异的原因。

6.3 基于 LightGBM 的特征重要性探究

为探究新药组和原有药物组在生命体征数据方面表现出显著差异是否能确定是由于新药造成，还是由其他因素造成，下文以 LightGBM 的特征进行重要性探究。

6.3.1 模型建立

LightGBM 是一种基于决策树的集成学习算法，是 GBDT 算法的改进版。相对于传统的 GBDT 算法，LightGBM 具有更快的训练速度、更低的内存消耗以及更高的准确率。可以将 LightGBM 的优化用公式表达，如下式：

$$LightGBM = XGBoost + Histogram + GOSS + EFB. \tag{24}$$

LightGBM 的核心算法可以用如下的公式表示：

$$Obj(\theta) = \sum_{i}^{n} l(y_i, \widehat{y_i}) + \sum_{k} \Omega(f_k). \tag{25}$$

其中，$\Omega(f_k)$ 表示正则化项，$\sum\limits_{k}$ 表示所有树的叶子节点，f_k 表示第 k 棵树。LightGBM 的目标是最小化 $Obj(\theta)$，即同时优化模型的预测精度和模型复杂度，θ 表示模型参数，$l\left(y_i, \widehat{y_i}\right)$ 表示预测值 $\widehat{y_i}$ 与真实值 y_i 之间的损失。

LightGBM 的训练过程可以用下图表示：

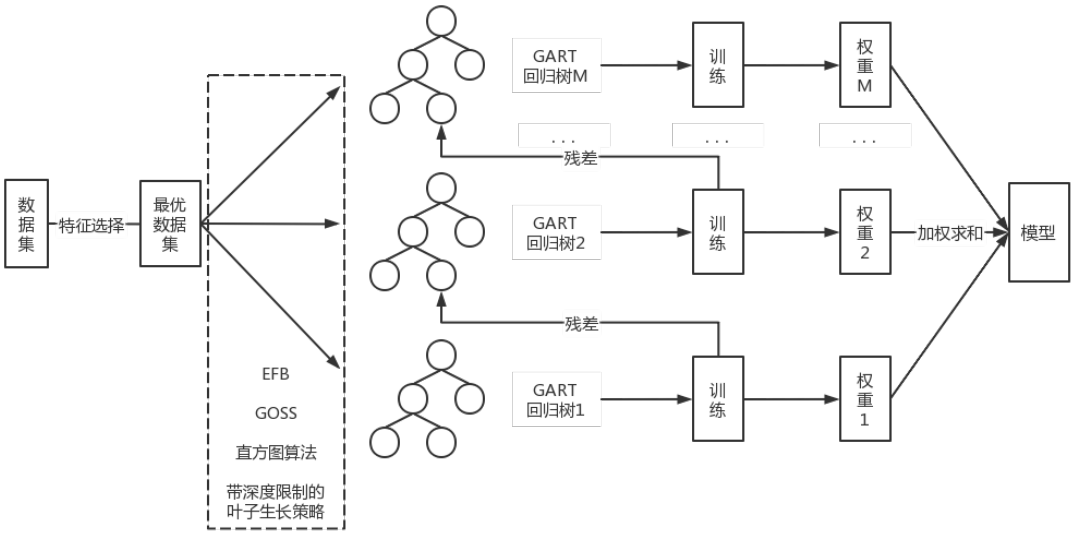


图 6 LightGBM 原理图

首先先对表中数据进行特征放缩和特征编码（具体步骤和问题一中数据预处理相同），然后将数据按照 80% 的比例划分为训练集和测试集。接着，定义模型参数并构建 LightGBM 模型。通过基于 GridSearchCV 的模型调参，通过对 LightGBM 模型的参数进行网格搜索，以找到最优的参数组合来训练模型，进而提高模型的预测性能。设置一些超参数进行调参，如下表所示：

表 6 GridSearchCV 对 LightGBM 调参结果

参数	取值范围	最优结果
max_depth	[4, 7, 10]	7
min_child_samples	[18, 20, 22]	22
n_estimators	[10, 70, 130]	10
num_leaves	[300, 600, 900]	300

基于上表最优参数训练 LightGBM 模型，再计算模型预测结果与实际值之间的均方误差和平均绝对误差，公式原理为：

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - \overline{x}_i|. \tag{26}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x}_i)^2. \tag{27}$$

其中，x_i 为第 i 个样本的实际值，$\overline{x}_i$ 为第 i 个样本的预测值，n 为样本数量，与平均绝对误差相比，均方误差对预测误差较大的样本更加敏感，因为它对误差取平方，使得误

差较大的样本对平均误差的影响更大，并分别输出结果如下：

表 7 LightGBM 评价结果

标签序号	MAE	MSE
第六主成分特征	0.3053	0.1391

对于 MAE 和 MSE 的范围都是 $[0, +\infty)$，当预测值和真实值完全吻合时等于 0，即完美模型；误差越大，该值越大。即这两个值越小模型精度越高。从上表评价指标可以看到 LightGBM 模型的训练效果十分优良，能够充分的得到 11 个患者特征与 15 个主成分的情况。

6.3.2 模型求解

基于上一问新药组和原有药物组关于第六个主成分特征表现出显著差异，为确定是由于新药造成还是其他因素造成，要对不同生命体征指标对应的特征重要性打分。利用上面训练好的 LightGBM 模型得到数据可视化图如下：

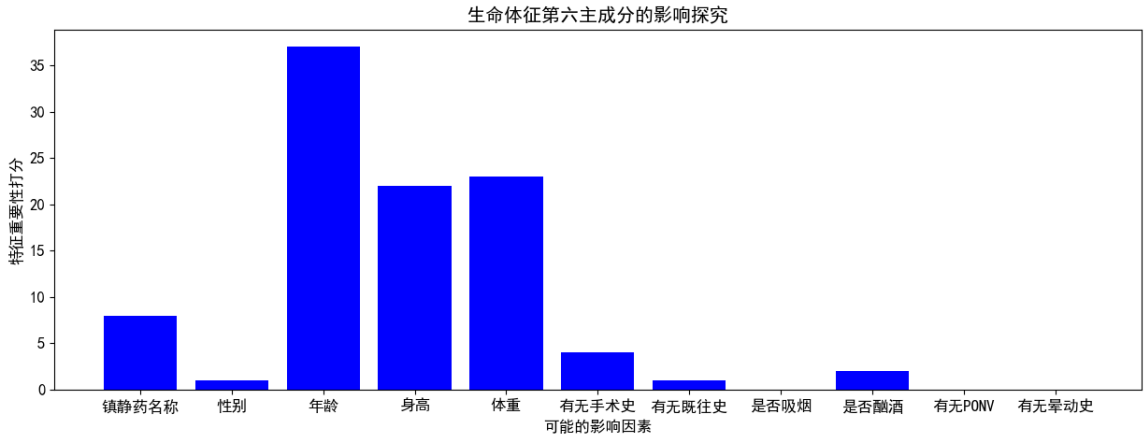


图 7 生命体征的显著主成分对应的特征重要性打分

分析上图，横坐标为可能的影响因素 (即特征名)，纵坐标为特征重要性打分，通过条形图的形式展示了每个特征对应的重要性分值，从上图发现镇静药的类别在不同因素的影响中重要性程度并不高，而是与个人的年龄、身高、体重这些客观因素的关联性更大，对于性别、有无既往史、有无手术史、是否酗酒关联性很小，对是否吸烟、有无晕动史、有无 ponv 之间甚至没有关联。这较为直观的解释了不同药组在生命体征第六主成分上存在显著差异的原因，但从整体上分析新药物对总的生命体征的影响是不大的，甚至比不上客观因素（年龄、身高、体重）的影响，这与实际情况也是吻合的，不同的药物对于生命体征的存在一定的可控的影响。这个结论表明生命体征数据表现出的差异和新药物没有特别强的关联性，新药物是具有一定的可信度的，可以正常使用，同时新药和原有药物之间也没有很大的差异性。

7 问题三的建模与求解

7.1 数据预处理

7.1.1 数据清洗与特征编码

本问要求对根据用药信息和患者信息对给药后 3 分钟以内的 IPI 数据进行预测，通过观察附件 2 发现只需要提取出 IPI005、IPI1、IPI015、IPI2、IPI025、IPI3 的相关数据，并从原始数据集中筛选出了需要的特征作为特征空间——包括性别、年龄、身高、体重、有无手术史、是否吸烟、是否酗酒、镇静药名称、镇静药诱导剂量、有无追加镇静、镇静药总剂量、镇痛药总剂量。

与上文数据预处理类似地，基于 Pandas 查找缺失值并删除一些对模型预测没有意义的特征，包括手术说明、既往史说明、ASA 评分等，之后删除少量含有缺失值的行。对特征空间中的数值特征进行特征缩放中的数据归一化，以消除其量纲；对分类特征进行编码以转化为离散特征——这些特征包括性别、有无手术史等。

7.1.2 数据降维

由于数据集中有大量的分类特征导致数据集整体较为离散，模型难以充分提取特征，且容易过拟合。使用主成分分析法对处理后数据进行降维，当前数据共有 14 列，通过将方差解释比例可视化可得下图：

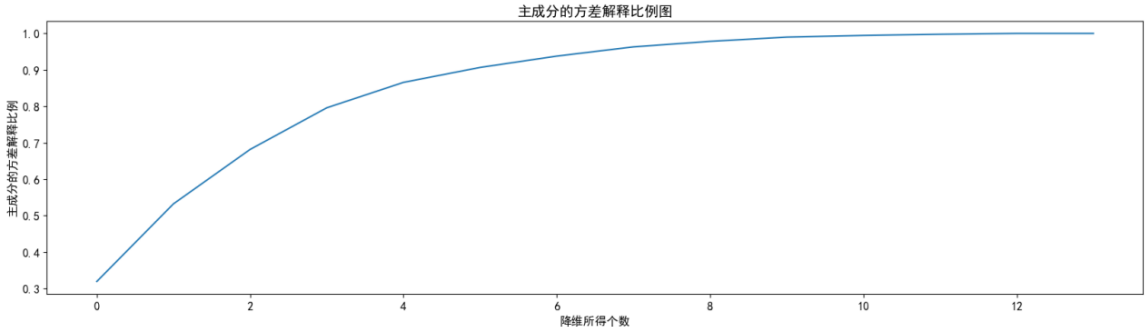


图 8 累积方差解释比例图

通过上图发现前四个主成分即可表示样本 80% 的信息，因此保留了 4 个主成分，对 4 个主成分特征重新导入到数据集，便于后面利用监督学习中的回顾方法。

7.2 岭回归与支持向量机回归的加权平均预测

7.2.1 模型建立

（一）岭回归模型

1. 岭回归

设有一个线性回归模型：

$$y = \overrightarrow{X}\overrightarrow{\beta} + \overrightarrow{\varepsilon}. \tag{28}$$

其中，$\overrightarrow{X}$ 是 $n \times p$ 的自变量矩阵，$\overrightarrow{\beta}$ 是 p 维系数向量，y 是 n 维因变量向量，$\overrightarrow{\varepsilon}$ 是 n 维误差向量。

引入 L2 正则化项后，岭回归的目标函数为：

$$\min_{\overrightarrow{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \overrightarrow{x}_i^T \overrightarrow{\beta} \right)^2 + \lambda \sum_{j=1}^{p} \overrightarrow{\beta}_j^2 \right\}. \tag{29}$$

其中，λ 是正则化强度超参数，控制正则化项的权重大小。岭回归的解可以用闭式解表达式表示：

$$\widehat{\overrightarrow{\beta}} = \left(\overrightarrow{X}^T \overrightarrow{X} + \lambda \overrightarrow{I} \right)^{-1} \overrightarrow{X}^T y. \tag{30}$$

其中，$\overrightarrow{I}$ 是 $p \times p$ 的单位矩阵。当 $\lambda = 0$ 时，岭回归就退化成了普通的线性回归；当 λ 取值较大时，正则化项的影响就越大，模型的系数就越接近于 0。

2. **支持向量机模型**

对于数据集 $D = \{(\overrightarrow{x}_1, y_1), (\overrightarrow{x}_2, y_2), \cdots, (\overrightarrow{x}_m, y_m)\}$，，得到一个回归模型 $f(\overrightarrow{x})$ 与 y 尽可能接近.SVR 问题可形式化为：

$$f(\overrightarrow{x}) = \overrightarrow{\omega}^T \Phi(\overrightarrow{x}) + b. \tag{31}$$

$$\overrightarrow{\omega} = \sum_{i=1}^{m} (\widehat{\alpha}_i - \alpha_i) \Phi(\overrightarrow{x}_i). \tag{32}$$

$$f(\overrightarrow{x}) = \sum_{i=1}^{m} (\widehat{\alpha}_i - \alpha_i) \kappa(\overrightarrow{x}, \overrightarrow{x}_i) + b. \tag{33}$$

其中 $\kappa(\overrightarrow{x}_i, \overrightarrow{x}_j) = \Phi(\overrightarrow{x}_i)^T \Phi(\overrightarrow{x}_j)$ 为核函数.$\overrightarrow{\omega}, b$ 为模型参数，$\widehat{\alpha}_i \geq 0, \alpha_i \geq 0$ 是拉格朗日乘子. 利用高斯核函数求解，公式为：

$$\kappa(\overrightarrow{x}_i, \overrightarrow{x}_j) = \exp\left(-\frac{\|\overrightarrow{x}_i - \overrightarrow{x}_j\|}{2\sigma^2} \right). \tag{34}$$

7.2.2 模型求解

首先基于 Scikit-Learn 建立三分钟内各时间点的 IPI 数值的两种预测模型——岭回归模型和 SVR 模型，设定岭回归模型在 i 样本的预测值为 $h_1(\overrightarrow{x}_i)$，SVR 模型的预测值为 $h_2(\overrightarrow{x}_i), i = 1, 2, \cdots, m$，构建集成学习器 H 包含两个基学习器h_1, h_2。

对于题目中由已知的数据集进行建模,分别需要对六个标签——IPI005、IPI1、IPI015、IPI2、IPI025 以及 IPI3 进行回归分析。加权平均法是一种常见的回归任务的模型融合方法，利用加权平均法进行结合，其中原理如下：

$$\omega_i = \frac{MSE_i}{MSE_1 + MSE_2}, i = 1, 2. \tag{35}$$

$$H(\overrightarrow{x}) = \frac{1}{2}\sum_{i=1}^{2} \omega_i h_i(\overrightarrow{x}). \tag{36}$$

其中,ω_1, ω_2 分别表示两个模型的权重,通过计算两种模型集成后组成的最优模型 $H(\overrightarrow{x})$。

对于两个基学习器和基于加权平均法的融合模型，本文分别在测试集上进行测试，基于 MAE 和 MSE 对模型泛化能力进行评价，所得如下：

表 8 三种模型在测试集上的测试评价

标签名称	模型	MAE	MSE
IPI005	岭回归	0.1447	0.0423
	SVR	0.1418	0.0417
	模型融合	0.1432	0.0419
IPI1	岭回归	0.2376	0.1020
	SVR	0.2272	0.1153
	模型融合	0.2322	0.1066
IPI015	岭回归	0.3669	0.1630
	SVR	0.3567	0.1808
	模型融合	0.3570	0.1651
IPI2	岭回归	0.2666	0.1103
	SVR	0.2363	0.1174
	模型融合	0.2509	0.1101
IPI025	岭回归	0.1680	0.0679
	SVR	0.1663	0.0682
	模型融合	0.1671	0.0678
IPI3	岭回归	0.1810	0.0798
	SVR	0.1788	0.0810
	模型融合	0.1797	0.0800

分析上表可知，基于三种模型在测试集上的评价指标，对于这六种标签预测评价指标都接近 0，说明这三种模型的预测效果都十分优良。从标签种类分析，可以发现对于标签 IPI005 的预测效果最佳，对于标签 IPI015 的预测效果最差。不同标签预测效果整体上的优良性从最优到优的排序为 IPI005、IPI025、IPI3、IPI1、IPI2、IPI015。

分析三种模型发现无论是岭回归模型还是 SVR 模型都不能保证是性能最优，而将模型融合可以弥补单个模型的不足，提高准确率。同时提升模型的鲁棒性，降低由于数据的随机性导致的模型波动，得到的预测结果更准确，泛化能力更强，模型稳定性更强。模型融合是提高预测准确性、提高模型鲁棒性、降低模型波动性的一种有效方法。

7.2.3 灵敏度分析

为了评估两个不同的模型（Ridge 回归和支持向量机）对输入数据的敏感性，这里通过施加一些高斯噪声（从 0 到 0.5 的比例），原理为

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{37}$$

其中，x 表示随机变量的取值，μ 表示期望值，σ 表示标准差。改变测试数据集的特征，来模拟模型的偏差，以确定模型的鲁棒性。然后根据分段函数：

$$error_{ij}=\begin{cases}0 & ,(0,c)\\ \overline{y}_{ij}+f(x_j) & ,(c,0.5)\end{cases} \tag{38}$$

其中 $\overline{y}_{ij}$ 为第 i 个特征的第 j 个的样本值，$f(x_j)$ 表示加入的高斯噪声，c 表示噪音阈值。重新预测，并计算预测结果与真实结果之间的均方误差。下文以 IPI3 为例对模型灵敏度进行分析（其余见附件），得到数据可视化图如下所示：

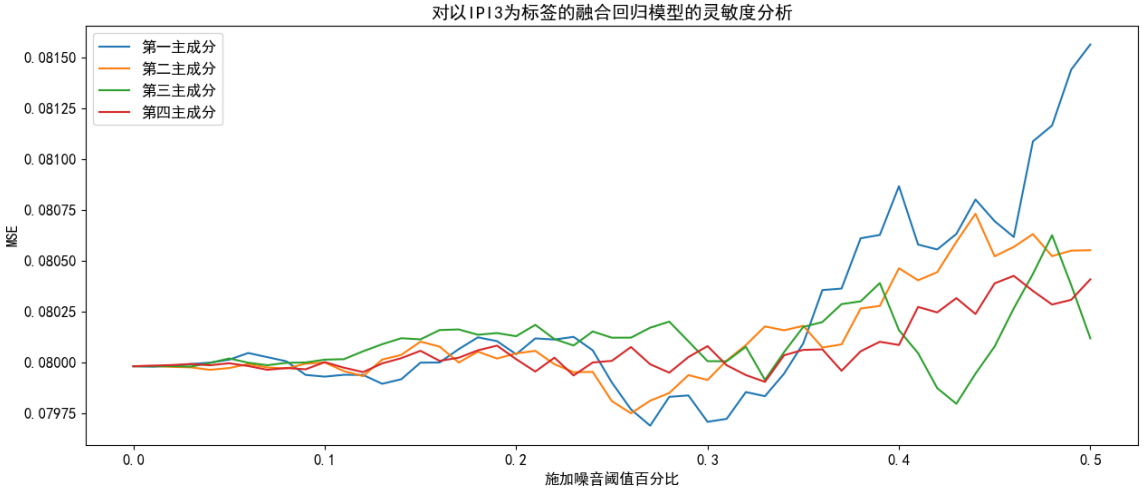


图 9 对以 IPI3 为标签的融合回归模型的灵敏度分析

从上图可以看出，对第一主成分、对第二主成分、对第三主成分、对第四主成分分别不断施加噪音，可以直观地看出 MSE（均方差）值在这个区间波动，四个特征在施加

噪音百分比在 30% 以内时 MSE 均无明显波动，说明该模型具有一定的可靠性和稳定性性能优良。另外五个标签的灵敏度分析图详见附录 A。

8 问题四的建模与求解

8.1 相关性分析

为了找出与术后满意度相关的特征，我们将以术后满意度为标签来计算各个特征的相关性，使用 Spearman 和 Kendall 相关性分析，Spearman 相关系数公式已给出，介绍 Kendall 原理：

设有两个随机变量 $\overrightarrow{X} = (x_1, x_2, ..., x_n), \overrightarrow{Y} = (y_1, y_2, ..., y_n)$，将它们分别排名得到两个的排名向量 $\overrightarrow{X} = (r_1, r_2, ..., r), \overrightarrow{Y} = (s_1, s_2, ..., s_n)$，得到相关系数:

$$\tau = \frac{2}{n(n-2)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{sgn}(x_i - x_j)\text{sgn}(y_i - y_j). \tag{39}$$

其中，表示 $\text{sgn}(x)$ 的 x 符号系数，当 $x > 0$ 时，$\text{sgn}(x) = 1$，当 $x = 0$ 时，$\text{sgn}(x) = 0$，当 $x < 0$ 时，$\text{sgn}(x) = -1$。分别计算得到的相关性如下图所示：

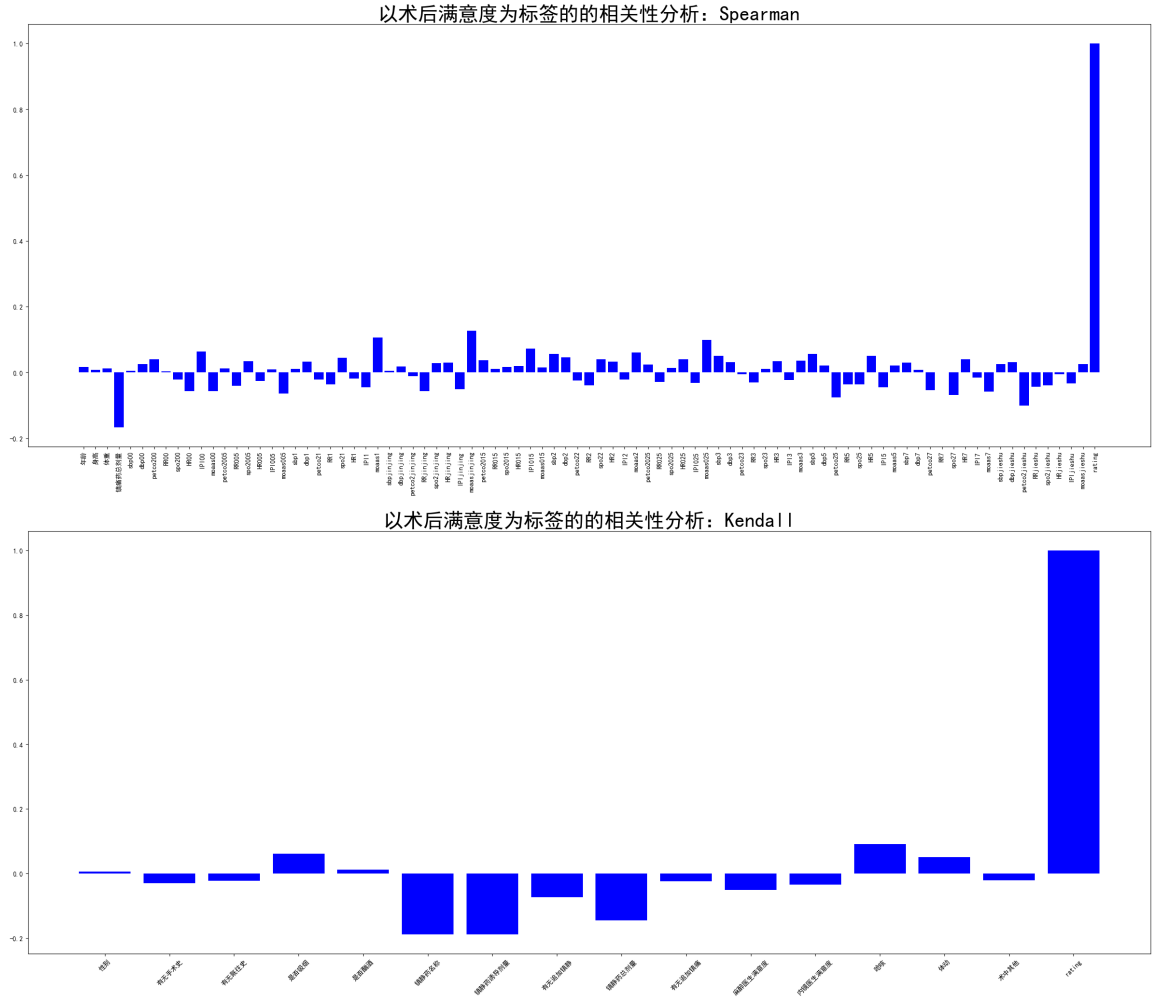


图 10 以术后满意度为标签的相关性分析

由上图可知，第一个图展示的是使用 Spearman 相关性分析得到的各个特征与标签之间的相关系数，而第二个图展示的是使用 Kendall 方法得到的相关系数，条形图的 x 轴是各个特征的名称，y 轴则是相应的相关系数。分析上图，由上分图关于数值型特征与术后满意度的 Spearman 相关系数得到相关性分析可以看出镇痛药的总剂量对于术后满意度的不良影响最大，这与现实情况是吻合的。而 moaasjinjing 与术后满意度的正向影响相关性最大，其中关于 moaas 的含量与术后满意的都有正相关性的关系，综合考虑，相关性的绝对值都不大于 0.2。从统计学上分析存在的相关性并不大，但是相对而言镇痛药的对术后满意度的不良影响的相关性较大。由第二张图关于分类特征与术后满意度的 Kendall 相关性分析可以看出镇静药名称、镇静药诱导剂量、镇静药总剂量对于术后满意度的不良影响最大，而其他因素对于术后满意度的正向影响相关性都比较弱，综合考虑，相关性的绝对值都不大于 0.2。从统计学上分析存在的相关性并不大，但是相对而言镇静药的名称、镇静药诱导剂量、镇静药总剂量对术后满意度的不良影响的相关性较大。综上，镇痛药和镇静药各种因素对于术后满意程度都有不利的影响，而 moaas 的含量对于术后满意度的影响是有利的。

通过这个图可以直观地比较不同特征与标签之间的相关性强度，但是由于术后满意

度标签分布不均衡，可能会影响模型提取特征的效果，进而影响其准确率和稳定性，需要进一步优化模型。

8.2 基于随机森林的分类标准挖掘

为进一步探究出与术后满意度有关的因素，并挖掘出其间的定量关系，本文基于随机森林算法建立模型

8.2.1 模型建立

随机森林是一种集成学习方法，由多棵决策树组成。其公式可以分为两部分：随机森林的生成和随机森林的预测。随机森林的生成过程如下：

1. 从原始数据集中采样出 n 个样本作为训练集，采用 bootstrap 技术进行采样，即每次从原始数据集中随机抽取一个样本并将其放回，重复 n 次得到大小为 n 的采样集。
2. 从所有特征中随机选择 m 个特征，其中 $m \ll d$，d 为原始特征的总数。
3. 利用上述采样得到的数据集和特征集构建一棵决策树，具体建树过程可以使用 ID3、C4.5 或 CART 等决策树算法。
4. 重复步骤前三步 T 次，得到 T 棵决策树，这些决策树构成了随机森林。

随机森林的预测步骤：

1. 对于每个测试样本，对随机森林中的每棵决策树进行预测，得到预测结果。
2. 对 T 个预测结果进行投票，将得票最多的类别作为随机森林的最终预测结果。

随机森林的预测公式可以表示为：

$$\widehat{y} = \arg\max_y \sum_{i=1}^{T} I\left(\widehat{y}_i = y\right). \tag{40}$$

其中，$\widehat{y}$ 表示随机森林的预测结果，$\widehat{y}_i$ 表示第 i 棵决策树的预测结果，T 表示随机森林中的决策树数量，y 表示所有可能的类别，$I(\cdot)$ 表示指示函数，当条件成立时取值为 1，否则取值为 0。

8.2.2 模型求解

通过对标签进行分布分析，得出如下条形统计图：

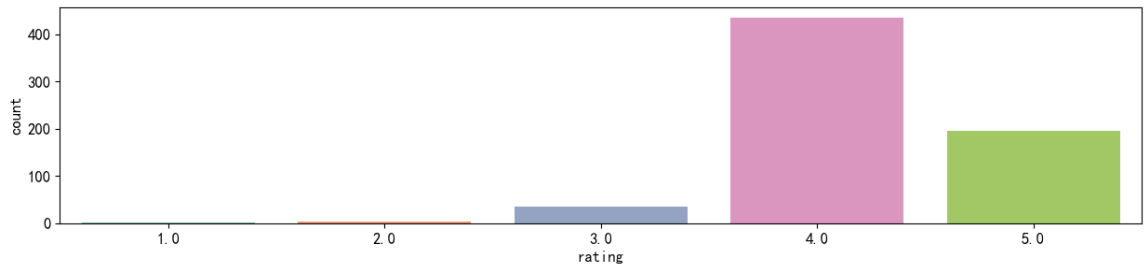


图 11 术后 24 小时的分布分布图

通过上图可以看出，本次分类任务的数据集标签严重失衡，类似问题一中的处理，使用 imblearn 库中的 RandomOverSampler 方法对数据进行上采样，并将处理后的数据集分为训练集和测试集，其中测试集占比 0.2，再用随机森林分类器进行训练和预测。基于 Scikit-Learn 将随机森林模型初始化，并使用训练集对其进行训练，最终分类器在测试集上的评估结果如下表所示：

表 9 对随机森林的术后 24 小时满意度预测评价

类别	准确率	召回率	F1 分数	支持率
非常满意	0.56	0.60	0.58	97
满意	0.50	0.17	0.25	89
一般	0.61	1.00	0.76	76
不满意	0.95	1.00	0.98	83
非常不满意	1.00	1.00	1.00	91

通过这个评价结果可以看出，该随机森林模型性能优良，预测结果可行度高。基于此背景，本文欲通过树模型独有的树状图探究分类的标准，从而了解与满意度有关的因素，并可以如题设所要求挖掘出具体的关系。通过 plot_tree 函数导出 max_depth=3 时的树状图如下：

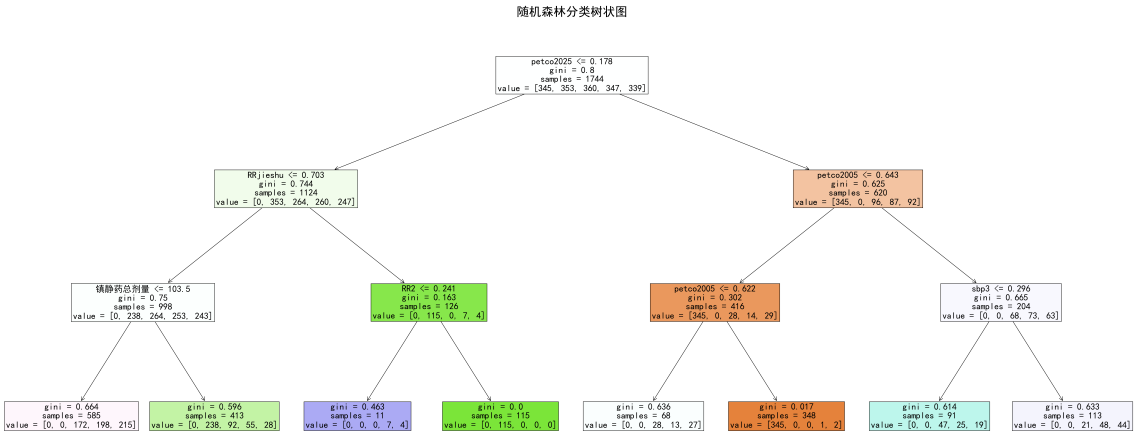


图 12 随机森林树状图导出

通过此图形，本文明确地得到：决策树所挖掘的数据内在的分类标准与每个特征指标之间存在的具体的数值关系，依据随机森林的原理，经过训练的树的每一层都会据特征空间的若干个特征进行分裂，从而模拟一种分类的效果。据图得到的结果如下：

表 10 树状图信息挖掘

层数	叶片编号	参考特征	分裂依据
1	1	petco2025	petco2025≤ 0.178
2	1-1	RRjieshu	RRjieshu≤ 0.703
	1-2	petco2005	petco2005≤ 0.643
3	1-1-1	镇静药总剂量	镇静药总剂量 ≤ 103.5
	1-1-2	RR2	RR2≤ 0.241
	1-2-1	petco2005	petco2005≤ 0.622
	1-2-2	sbp3	sbp3≤ 0.296

通过上表可以看出树状图的每个叶片所蕴含的信息，关键的信息在于随机森林选取的叶片分裂的标准——这有助于本文建立术后满意度与影响因素的定量联系。通过该图可以得知，随机森林选取特征空间中的“petco2025”、“RRjieshu”、“petco2005”、“镇静药总剂量”、“RR2”、“petco2005”以及“sbp3”作为用于分类的特征，并按照严格的定量标准对每层叶片进行分裂。

然而从对随机森林的评价结果可以看出：尽管本文建立的随机森林模型在五分类任务整体准确率达到了 0.75，然而从五个类别分别来看有两个类别正确率仅不足六成，故需考察底层叶片的具体信息，并通过 Gini 指数来严格判断给出的分类标准是否可信，以此来证明所建立的联系的严谨性，底层叶片信息如下：

表 11 底层叶片的具体信息

叶片编号	分裂结果	Gini 指数	主观可信度
1-1-1-1	{0,0,172,198,215}	0.664	差
1-1-1-2	{0,238,92,55,28}	0.596	差
1-1-2-1	{0,0,0,7,4}	0.463	良
1-1-2-2	{0,115,0,0,0}	0	优
1-2-1-1	{0,0,28,13,27}	0.636	差
1-2-1-2	{345,0,0,1,2}	0.017	优
1-2-2-1	{0,0,47,25,19}	0.614	差
1-2-2-2	{0,0,21,48,44}	0.633	差

Gini 指数是一种衡量节点纯度的指标，其体现的即为叶片分裂的可参考程度。对于五分类的随机森林模型，本文主观地根据八个底层叶片的 Gini 指数取值归类为“优”、“良”、“差”三类，舍弃对 Gini 指数表现较差的分类标准的参考，最终选取有效分类标准如下：

表 12 有效联系挖掘

特征名称	分类标准	具体判别类别
分裂组 1	petco2025≤ 0.178 RRjieshu≤ 0.703 RR2≤ 0.241	“不满意”、“满意”、“非常满意”
分裂组 2	petco2025≥ 0.178 petco2005≤ 0.643 petco2005≤ 0.622	“非常不满意”

基于此表，关于各特征与术后满意度的联系尽收眼底，本文通过一个简易流程图来解释术后满意度与“petco2025”，“RRjieshu”，“RR2”，“petco2005”的联系：

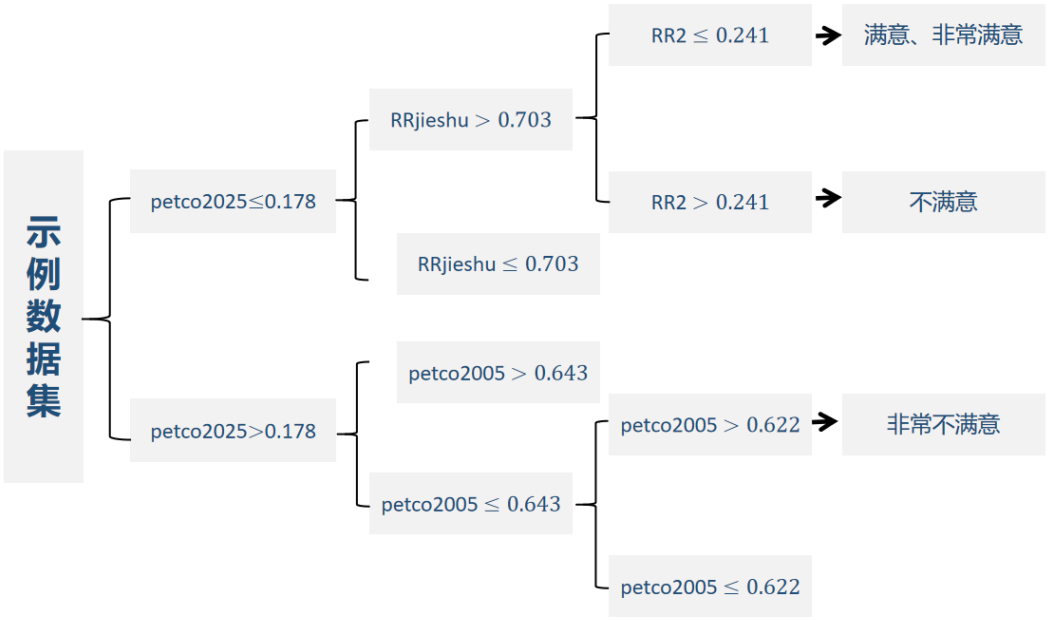


图 13 术后满意度与特征的定量联系

这也进一步对分类的指标有了具体化的体现，说明上面的简单的相关性分析往往只能摸索出浅层次的结论，本文选用的随机森林算法具有深度挖掘类别与特征之间联系的能力，且具有推广价值。

9 模型评价和改进

9.1 模型优点

1. 本文使用特征编码将数据集中的类别特征巧妙转化为数值特征，便于分析特征，以便提高模型的精度。
2. 本文充分考虑到变量与变量之间的相关性，使用主成分分析法对原有的数据进行降维，可以使得特征的选择更加客观。
3. 在问题三中通过施加噪音对模型的灵敏度进行分析，可以使得模型评价更为客观。
4. 对于回归任务利用线性加权的得到更加精确的回归模型，极大的提高了模型的精度和科学性。
5. 对于问题四不仅找到与术后满意度关联的因素，还进一步利用随机森林挖掘出了关联因素的分类规则。

9.2 基于 Voting 的模型推广

对于问题一中的分类任务，本文考虑模型的数学可解释性，同时鉴于经过上采样和适当的数据预处理、特征工程处理后的二分类问题应当有优异的性能，未使用一些复杂的机器学习算法，取而代之的是兼具较好性能和数学可解释性的 K 最近邻算法。然而题设背景为临床医学，这无疑要求本文提供一个尽可能最优的不良反应预判模型——尽管这个模型数学可解释性可能不强。

对于分类任务来说，学习器从类别标记集合中预测出一个标记，最常见的组合策略是使用投票法，记学习器 h_i 在样本 $\vec{x}$ 上预测输出为一个 N 维向量：

$$\left(h_i^1\left(\vec{x}\right), h_i^2\left(\vec{x}\right), \cdots, h_i^N\left(\vec{x}\right)\right). \tag{41}$$

其中 $h_i^j\left(\vec{x}\right)$ 是 h_i 在类别标记上的输出。

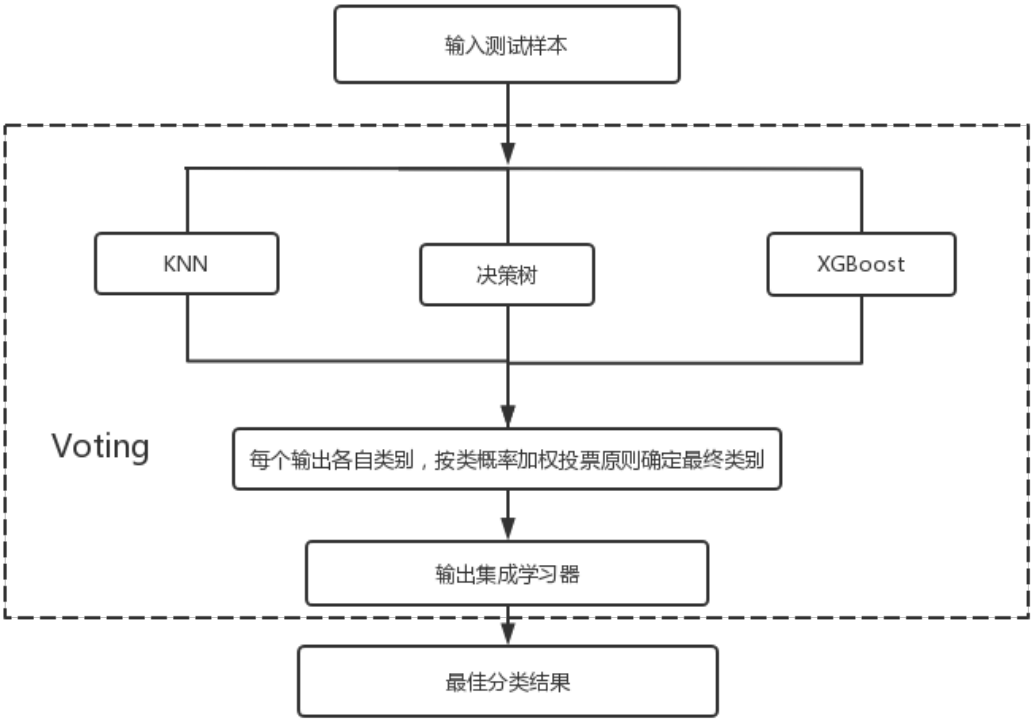


图 14 Voting 原理图

对于分类模型，本文欲基于模型融合算法 VotingClassifier 将多个性能优良的基分类器融合在一起。下文中的分析与评价均以“呛咳”为标签的分类任务为例。本文选取 7 个经典的分类模型，经过训练集训练后在测试集上测试，以 Accuracy_score 为初步的评价指标，结果如下：

表 13 分类器泛化能力评价——以 Accuracy_score 为指标

分类器	Accuracy_score
逻辑回归	0.646
线性判别分析	0.63941
K 最近邻	0.90985
决策树	0.97694
朴素贝叶斯	0.61635
支持向量机	0.66247
XGBoost	0.97904

基于上面的初步测试，本文选择决策树、K 最近邻、XGBoost、Catboost 作为基分类器，通过 Voting 中的软投票对四个分类器进行融合，结果如下：

表 14 Voting 与基分类器性能对比

分类器	Accuracy_score
XGB	0.97904
KNN	0.90985
DT	0.97694
Voting	0.98113

进一步地，为了更直观地展示 Voting 的泛化能力，本文通过混淆矩阵和 ROC 图对 Voting 的性能进行可视化：

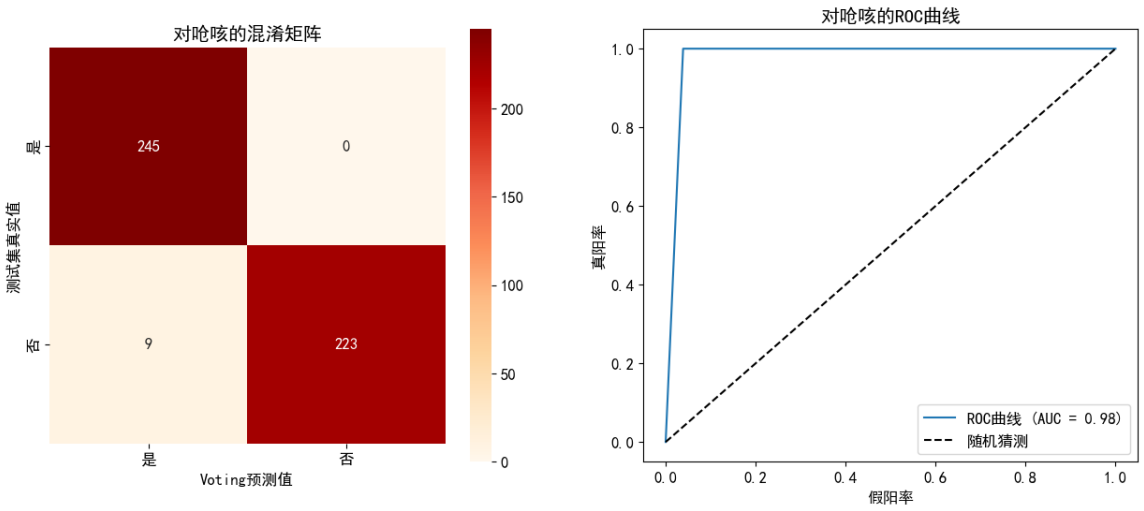


图 15 Voting 的混淆矩阵与 ROC 图

可以见得 Voting 的分类性能要比 KNN 更好，将其应用在临床医学中才更符合人工智能服务人类的初衷。

10 参考文献

[1] 郭躬德, 黄杰, 陈黎飞. 基于 KNN 模型的增量学习算法 [J]. 模式识别与人工智能,2010,23(05):701-707.

[2] 申晴, 张连增. 一种新的银行信用风险识别方法：SVM-KNN 组合模型 [J]. 金融监管研究,2020,(07):23-37.

[3] Benarafa Halima,Benkhalifa Mohammed,Akhloufi Moulay. WordNet Semantic Relations Based Enhancement of KNN Model for Implicit Aspect Identification in Sentiment Analysis[J]. International Journal of Computational Intelligence Systems,2023,16(1).

[4] 王大鹏, 闫肃, 王楠, 等. 基于卡方检验和秩和检验的智慧消防行业分析 [J]. 消防科学与技术, 2022, 41(11): 1594.

[5] 王皓辰, 张长伦, 黎铭亮. 基于深度学习的点云上采样算法研究 [J]. Journal of Image and Signal Processing, 2023, 12: 21.

[6] 梁胜杰, 张志华, 崔立林. 主成分分析法与核主成分分析法在机械噪声数据降维中的应用比较 [J]. 中国机械工程,2011,22(01):80-83.

[7] 曹前. 基于二阶多项式回归和权重主成分分析法的多光谱降维算法研究 [J]. Optical Technique, 2023, 49(2): 250-256.

[8] Wan Minghua,Wang Xichen,Tan Hai,Yang Guowei. Manifold Regularized Principal Component Analysis Method Using L2,p-Norm[J]. Mathematics,2022,10(23).

[9] 张凯, 张科. 基于 LightGBM 算法的边坡稳定性预测研究 [J]. 中国安全科学学报, 2022, 32(7): 113.

[10] Yang Qiang,Feng Yan,Guan Li,Wu Wenyu,Wang Sichen,Li Qiangyu. X-Band Radar Attenuation Correction Method Based on LightGBM Algorithm[J]. Remote Sensing,2023,15(3).

[11] Anand L.,Mewada Shivlal,Shamsi WameedDeyah,Ritonga Mahyudin,Aflisia Noza,KumarSarangi Prakash,NdoleArthur Moses. Diagnosis of Prostate Cancer Using GLCM Enabled KNN Technique by Analyzing MRI Images[J]. BioMed Research International,2023,2023.

[12] 张晓辉, 李莹, 王华勇, 等. 应用特征聚合进行中文文本分类的改进 KNN 算法 [J]. 东北大学学报: 自然科学版, 2003, 24(3): 229-232.

[13] Roshanfekr Behnam,Amirmazlaghani Maryam,Rahmati Mohammad. Learning graph from graph signals: An approach based on sensitivity analysis over a deep learning framework[J]. Knowledge-Based Systems,2023,260.

[14] 彭高辉, 王志良. 数据挖掘中的数据预处理方法 [J]. 华北水利水电学院学报, 2008 (6): 61-63.

[15] ParedesSalazar Enrique A,CalderónCárdenas Alfredo,Varela Hamilton. Sensitivity Analysis in the Microkinetic Description of Electrocatalytic Reactions.[J]. The journal of physical chemistry. A,2022,126(17).

[16] 吴庶宸, 戚宗锋, 李建勋. 基于深度学习的智能全局灵敏度分析 [J]. 上海交通大学学报, 2022, 56(7): 840.

[17] 任家东, 刘新倩, 王倩, 何海涛, 赵小林. 基于 KNN 离群点检测和随机森林的多层入侵检测方法 [J]. 计算机研究与发展,2019,56(03):566-575.

[18] Li Zhenglei,Chen Yu,Tao Yan,Zhao Xiuge,Wang Danlu,Wei Tong,Hou Yaxuan,Xu Xiaojing. Mapping the personal PM2.5 exposure of China's population using random forest[J]. Science of the Total Environment,2023,871.

[19] 张著英, 黄玉龙, 王翰虎. 一个高效的 KNN 分类算法 [J]. 计算机科学,2008,(03):170-172.

[20] 方匡南, 吴见彬, 朱建平, 谢邦昌. 随机森林方法研究综述 [J]. 统计与信息论坛,2011,26(03):32-38.

A 附录：本文全部解答过程的流程图

A.1 第一题流程图

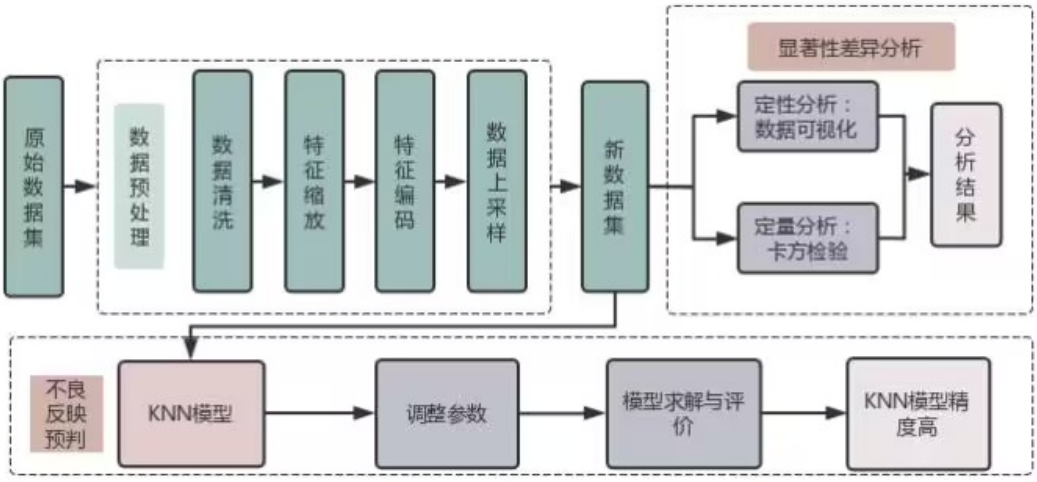


图 16 问题一全过程流程图

A.2 第二题流程图

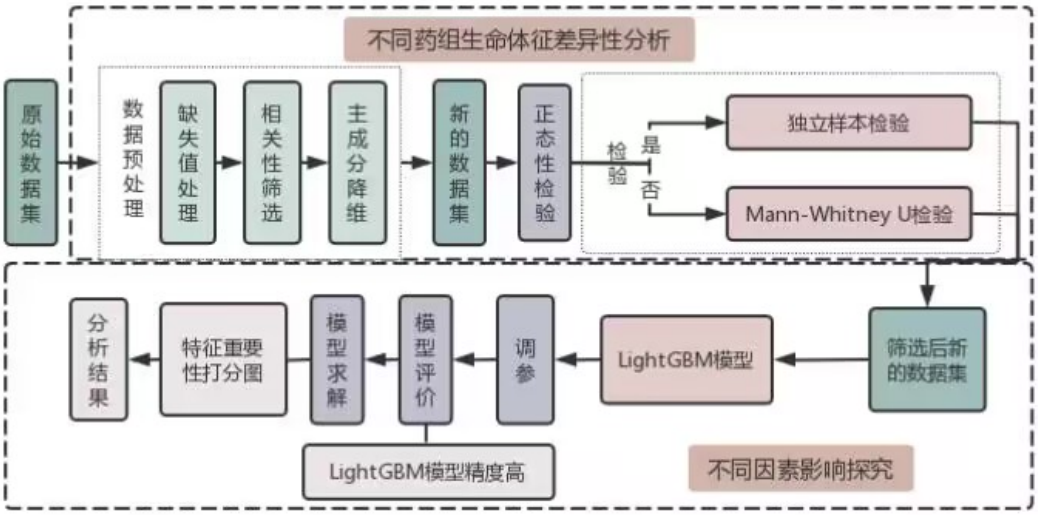


图 17 问题二全过程流程图

A.3 第三题流程图

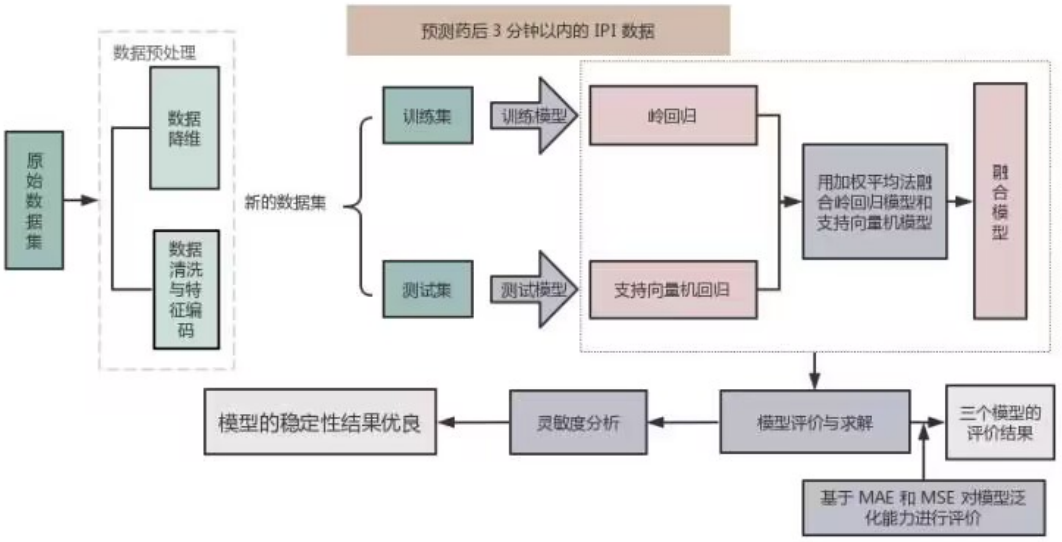


图 18 问题三全过程流程图

A.4 第四题流程图

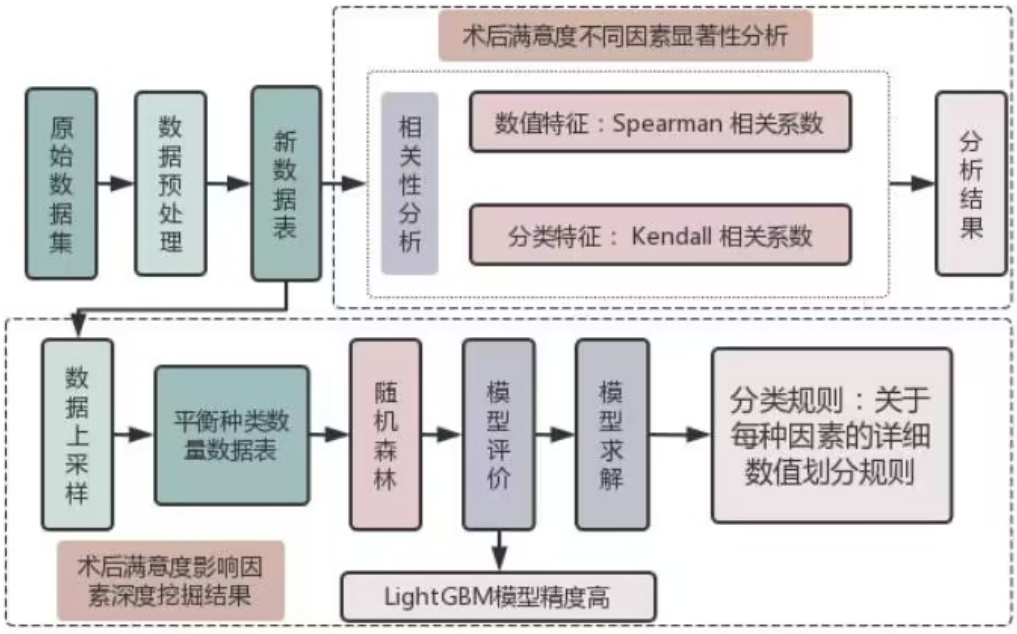


图 19 问题四全过程流程图

B　附录：图表

B.1　完整 KNN 测试的混淆矩阵、ROC 图

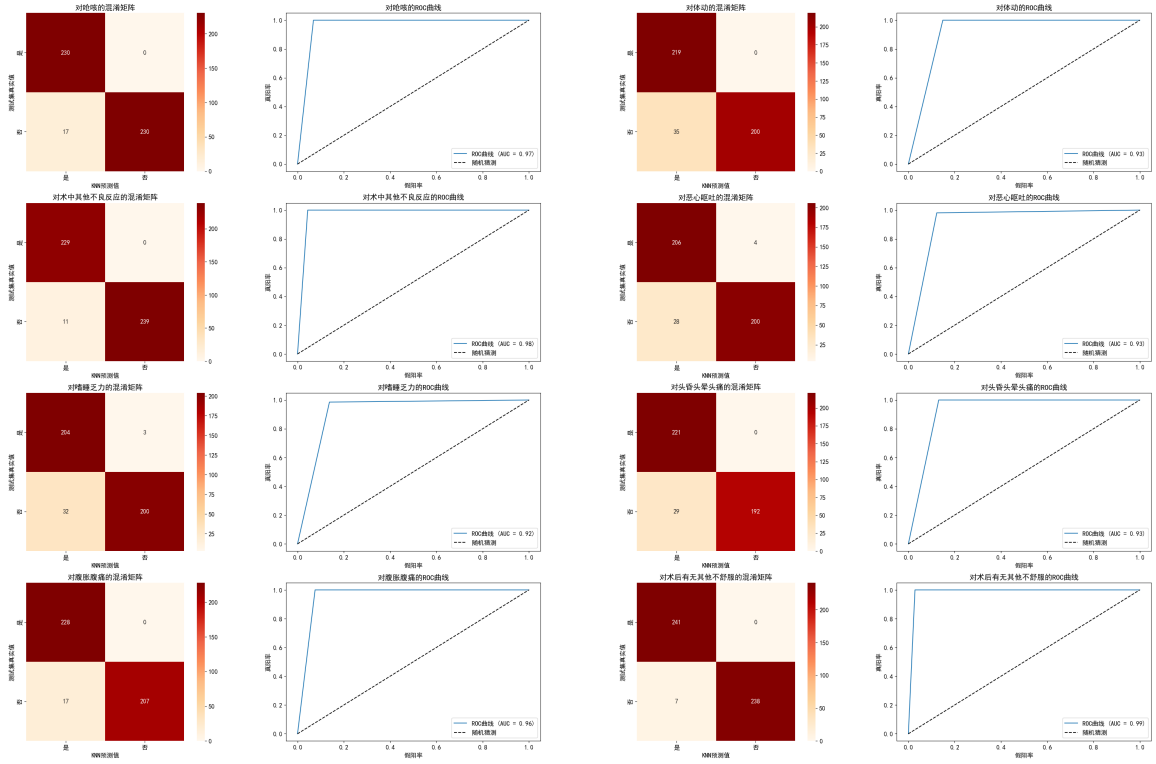
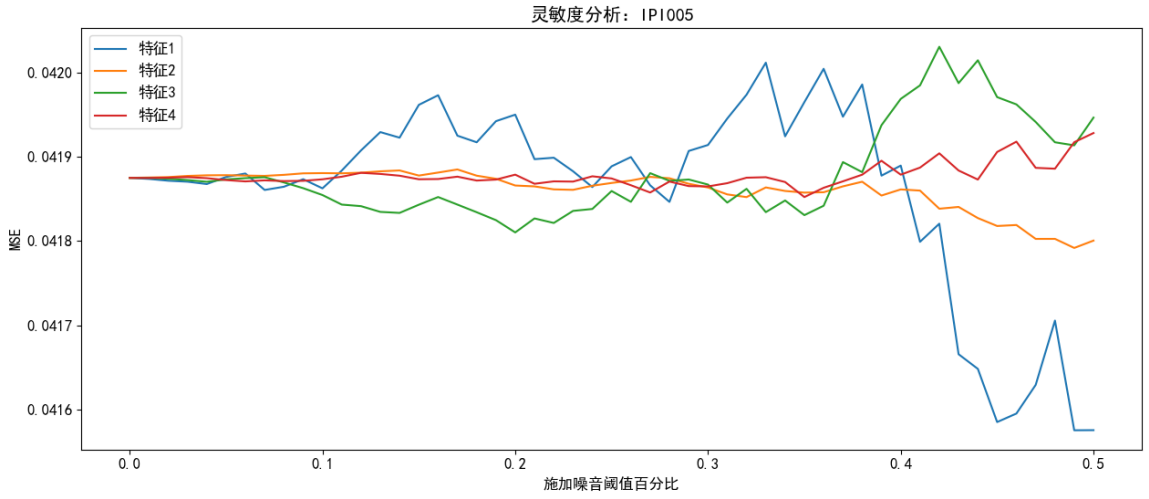
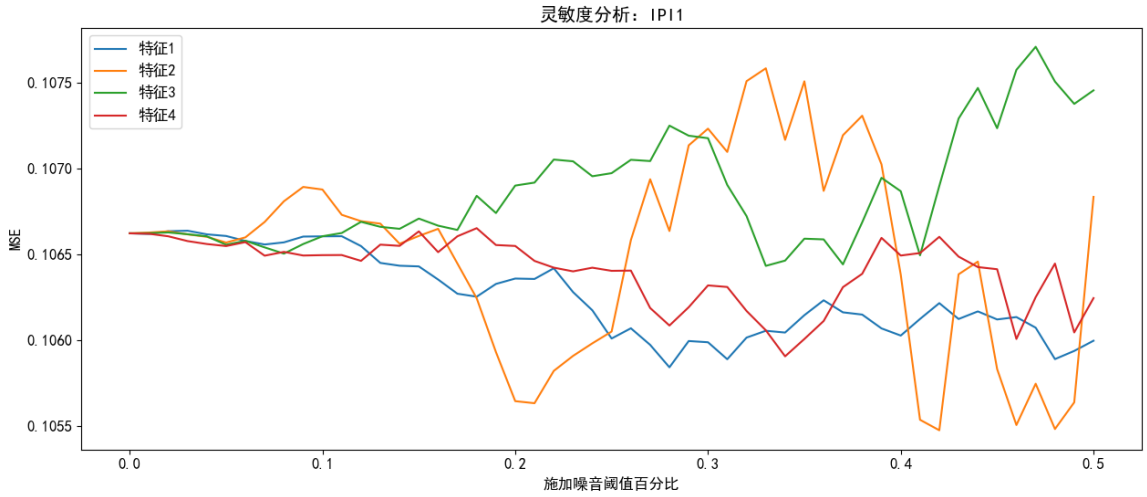
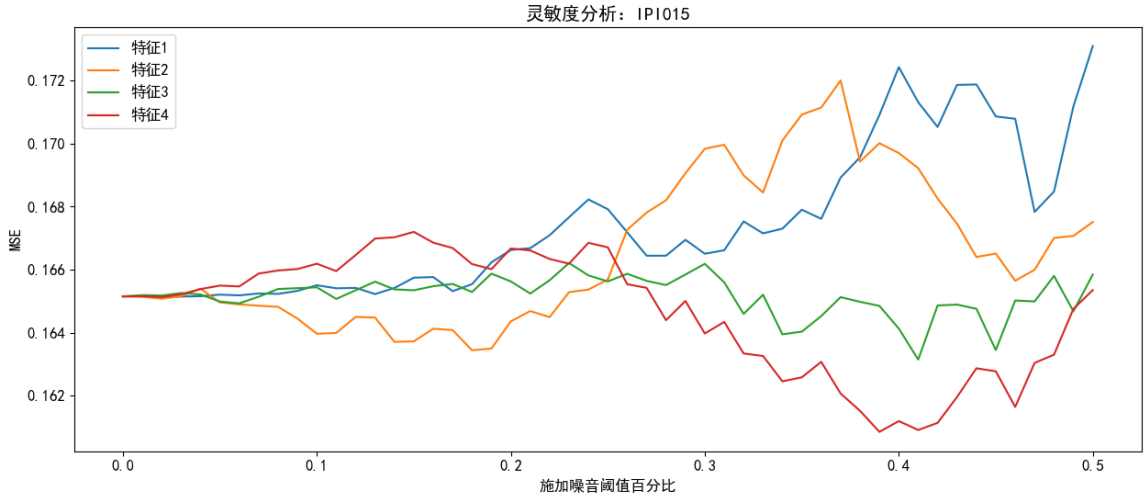


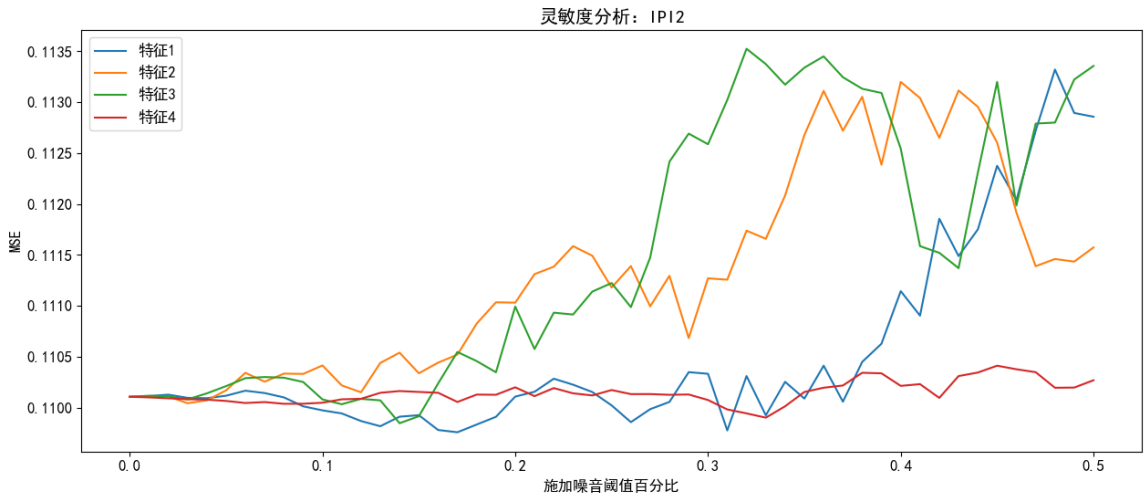
图 20　完整 KNN 测试结果的混淆矩阵和 ROC 图

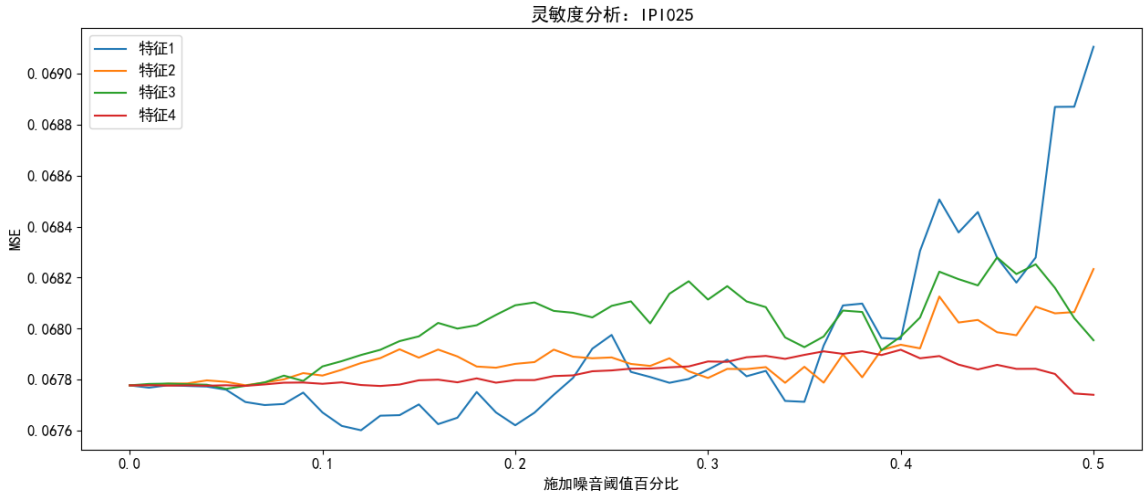
B.2　六次回归的灵敏度分析图

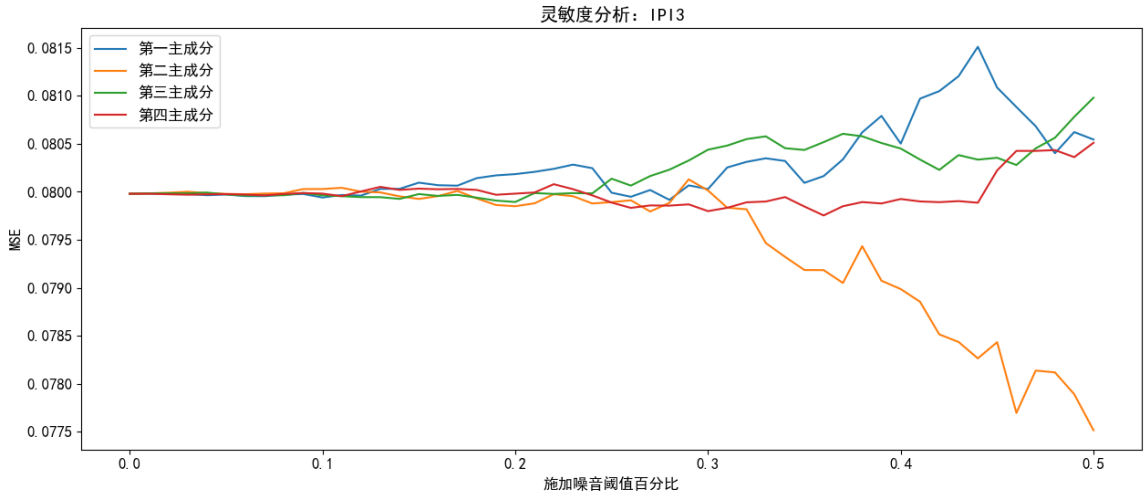



灵敏度分析：IPI1
特征1
特征2
特征3
特征4
MSE
0.1075
0.1070
0.1065
0.1060
0.1055
0.0
0.1
0.2
0.3
0.4
0.5
施加噪音阈值百分比


灵敏度分析：IPI015
特征1
特征2
特征3
特征4
MSE
0.172
0.170
0.168
0.166
0.164
0.162
0.0
0.1
0.2
0.3
0.4
0.5
施加噪音阈值百分比


灵敏度分析：IPI2
特征1
特征2
特征3
特征4
MSE
0.1135
0.1130
0.1125
0.1120
0.1115
0.1110
0.1105
0.1100
0.0
0.1
0.2
0.3
0.4
0.5
施加噪音阈值百分比





C 附录：代码

附录代码仅为重要功能实现部分，完整代码参照提交的附件，附件内代码均为 ipynb（Jupyter Notebook），路径均为相对路径，可以直接运行。

C.1 卡方检验

```
# 创建一个交叉表格，以Sex和Category作为行列索引
cross_table = pd.crosstab(df1['镇静药名称'], df1['呛咳'])

# 进行卡方检验，并返回卡方值、p值、自由度和期望值等相关信息
chi2, p_value, dof, expected = stats.chi2_contingency(cross_table)

# 输出检验结果
# print(f"卡方值: {chi2}")
```

```
# print(f"p值: {p_value}")
# print(f"自由度: {dof}")
# print(f"期望值: \n{expected}")

if p_value < 0.05:
    print('关于呛咳，两种药有显著差异')
else:
    print('关于呛咳，两种药没有显著差异')
```

C.2 KNN 模型建立与评价

```
# ----------------------------------------------
# 模型准备
# ----------------------------------------------
# 特征空间:独热编码
X = pd.get_dummies(df2[['性别','年龄','身高','体重','有无手术史','有无既往
    史',
          '是否吸烟','是否酗酒','镇静药名称']])

# 数据归一化
model = MinMaxScaler()
X[['年龄','身高','体重']] = model.fit_transform(df2[['年龄','身高','体重'
    ]])

# 字符串换成数字
df2["呛咳"] = df2["呛咳"].map({"有": 1, "无": 0})
df2["体动"] = df2["体动"].map({"有": 1, "无": 0})
df2["术中其他"] = df2["术中其他"].map({"有": 1, "无": 0})
df2["是否恶心呕吐"] = df2["是否恶心呕吐"].map({"是": 1, "否": 0})
df2["是否头晕头昏头痛"] = df2["是否头晕头昏头痛"].map({"是": 1, "否": 0})
df2["是否嗜睡乏力"] = df2["是否嗜睡乏力"].map({"是": 1, "否": 0})
df2["是否腹胀腹痛"] = df2["是否腹胀腹痛"].map({"是": 1, "否": 0})
df2["有无其他不舒服"] = df2["有无其他不舒服"].map({"是": 1, "否": 0})



# ----------------------------------------------
```

```
# 上采样
# ------------------------------------------------
# 创建上采样对象
ros = RandomOverSampler(random_state=42)
# 对数据进行上采样
X_resampled, y_resampled = ros.fit_resample(X, df2.呛咳)

# 定义参数搜索空间
param_grid = {
    'n_neighbors': np.arange(1, 11),
    'weights': ['uniform', 'distance'],
    'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
    'leaf_size': np.arange(10, 51, 10),
    'p': [1, 2, 3]
}

# ------------------------------------------------
# GridSearchCV调参
# ------------------------------------------------
# 初始化KNN模型
knn = KNeighborsClassifier()

# 使用网格搜索进行调参
grid_search = GridSearchCV(knn, param_grid=param_grid, cv=5)
grid_search.fit(X_resampled, y_resampled)

# 输出最佳参数和最佳得分
print("Best parameters: {}".format(grid_search.best_params_))
print("Best cross-validation score: {:.2f}".format(grid_search.
    best_score_))

# ------------------------------------------------
# 以“呛咳”为例的KNN预测实现
# ------------------------------------------------
# 创建上采样对象
ros = RandomOverSampler(random_state=42)
# 对数据进行上采样
```

```
X_resampled, y_resampled = ros.fit_resample(X, df2.呛咳)

# 划分数据集
X1_train, X1_test, y1_train, y1_test = train_test_split(X_resampled,
    y_resampled, test_size=0.2, random_state=0)

Model = KNeighborsClassifier(algorithm='auto', leaf_size=10,
                             n_neighbors=1, p=3, weights='uniform')
y1_pred = Model.fit(X1_train, y1_train).predict(X1_test)

# 模型评价
print("评估数据结果打印:\n", classification_report(y1_test, y1_pred))

# ---------------------------------------------
# 混淆矩阵与ROC曲线
# ---------------------------------------------
mat1 = confusion_matrix(y1_test, y1_pred,
                labels=[1, 0])
fpr1, tpr1, thresholds1 = roc_curve(y1_test, y1_pred)
auc1 = roc_auc_score(y1_test, y1_pred)

plt.subplot(1, 2, 1)
sns.heatmap(mat1, annot=True, square="equal", cmap="OrRd", fmt="d",
    xticklabels=["是", "否"],
    yticklabels=["是", "否"])
plt.xlabel("KNN预测值")
plt.ylabel("测试集真实值")
plt.title("对呛咳的混淆矩阵")

plt.subplot(1, 2, 2)
# 绘制ROC曲线和y=x的对角线
plt.plot(fpr1, tpr1, label='ROC曲线 (AUC = {:.2f})'.format(auc1))
plt.plot([0, 1], [0, 1], 'k--', label='随机猜测')
plt.xlabel('假阳率')
plt.ylabel('真阳率')
plt.title('对呛咳的ROC曲线')
plt.legend()
```

C.3 主成分分析法

```
# ------------------------------------------
# 先试探，用可视化图确定具体降维多少
# ------------------------------------------
# 创建PCA对象并进行降维
pca = PCA(n_components=49)
df_try = pca.fit_transform(df3[['sbp00', 'dbp00', 'petco200', 'RR00', '
    spo200', 'HR00', 'IPI00', 'moaas00',
    'petco2005','RR005','spo2005','HR005','IPI005','moaas005','sbp1','dbp1
        ',
    'petco21','RR1','spo21','HR1','IPI1','moaas1','sbpjinjing','dbpjinjing
        ','petco2jinjing',
    'RRjinjing','spo2jinjing','HRjinjing','IPIjinjing','moaasjinjing','
        petco2015',
    'RR015','spo2015','HR015','IPI015','moaas015','sbp2','dbp2','petco22',
        'RR2','spo22',
    'HR2','IPI2','moaas2','petco2025','RR025','spo2025','HR025','IPI025',
    'moaas025','sbp3','dbp3','petco23','RR3','spo23','HR3','IPI3','moaas3'
        ,'sbp5',
    'dbp5','petco25','RR5','spo25','HR5','IPI5','moaas5','sbp7','dbp7','
        petco27',
    'RR7','spo27','HR7','IPI7','moaas7','sbpjieshu',
    'dbpjieshu','petco2jieshu','RRjieshu','spo2jieshu','HRjieshu','
        IPIjieshu']])

# 绘制主成分方差解释比例图
plt.figure(figsize=(20, 5))
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('降维所得个数')
plt.ylabel('主成分的方差解释比例')
plt.title('主成分的方差解释比例图')
plt.show()

# ------------------------------------------
# 主成分分析法实现数据降维
# ------------------------------------------
# 创建PCA对象并进行降维
```

```
pca = PCA(n_components=10)
df4 = pd.DataFrame(pca.fit_transform(df3.drop(['镇静药名称','性别','年龄',
    '身高',
                                                  '体重','有无手术史','有无既往史','
                                                     是否吸烟',
                                                  '是否酗酒','有无PONV','有无晕动史'
                                                     ], axis=1)))
```

C.4 独立样本检验

```
# ------------------------------------------------
# 进行正态性检验
# ------------------------------------------------
stat, p = normaltest(df[1])
if p < 0.05:
    print('参数检验')
else:
    print('非参数检验')

# ------------------------------------------------
# 独立样本t检验
# ------------------------------------------------
# 将标签数据根据性别分成两组
e1 = df.loc[df['镇静药组'] == 1, 'label1']
e2 = df.loc[df['镇静药组'] == 0, 'label1']

# 进行独立样本t检验
fvalue, pvalue = ttest_ind(e1, e2)

# 输出结果
print("F值为: ", fvalue)
print("p值为: ", pvalue)

# ------------------------------------------------
# Mann-Whitney U
# ------------------------------------------------
# 将标签数据根据性别分成两组
```

```
e1 = df.loc[df['镇静药组'] == 1, 'label2']
e2 = df.loc[df['镇静药组'] == 0, 'label2']

# 进行Mann-Whitney U检验
fvalue, pvalue = mannwhitneyu(e1, e2)

# 输出结果
print("F值为: ", fvalue)
print("p值为: ", pvalue)
```

C.5 LightGBM 建模与特征重要性分析

```
# ------------------------------------------------
# 划分数据集
# ------------------------------------------------
X1_train, X1_test, y1_train, y1_test = train_test_split(X, df[3],
    train_size=0.8, random_state=1)



# ------------------------------------------------
# GridSearchCV调参
# ------------------------------------------------
model = lgb.LGBMRegressor(learning_rate=0.1)
param = {
        "max_depth":[4, 7, 10],
        "num_leaves":[300, 600, 900],
        "n_estimators":[10, 70, 130],
        'min_child_samples': [18, 20, 22],
        'min_child_weight':[0.001, 0.002]
        }

grid_search3 = GridSearchCV(model, n_jobs=-1, param_grid=param, cv=5,
    scoring="neg_mean_squared_error")
grid_search3.fit(X1_train, y1_train)
grid_search3.best_estimator_, grid_search3.best_score_

X1_train, X1_test, y1_train, y1_test = train_test_split(X, df[3],
```

```
    train_size=0.8, random_state=1)


# ------------------------------------------
# LightGBM模型建立
# ------------------------------------------
model1 = lgb.LGBMRegressor(max_depth=7, min_child_samples=22,
    n_estimators=10,num_leaves=300)
y1_pred = model1.fit(X1_train, y1_train).predict(X1_test)


# ------------------------------------------
# LightGBM模型评价
# ------------------------------------------
# Mean Absolute Error（平均绝对误差）
print(mean_absolute_error(y1_test, y1_pred))

# Mean Squared Error（均方误差）
print(mean_squared_error(y1_test, y1_pred))

names = ['镇静药名称','性别','年龄','身高','体重','有无手术史','有无既往史'
    ,
        '是否吸烟','是否酗酒','有无PONV','有无晕动史']

# ------------------------------------------
# LightGBM特征重要性打分
# ------------------------------------------
plt.figure(figsize=(15, 5))
plt.title("生命体征第三主成分的影响探究")
plt.xlabel("可能的影响因素")
plt.ylabel("特征重要性打分")
plt.bar(names, model1.feature_importances_, color='b')
plt.show()
```

C.6 回归器的模型融合与灵敏度分析

```
# ------------------------------------------
```

```
# 岭回归的实现
# ------------------------------------------
# 模型初始化
ridge = Ridge(alpha=1)
# 模型训练、模型预测
y1_pred1 = ridge.fit(X1_train, y1_train).predict(X1_test)
# Mean Squared Error（均方误差）
MSE_1 = mean_squared_error(y1_test, y1_pred1)
# Mean Absolute Error（平均绝对误差）
MAE_1 = mean_absolute_error(y1_test, y1_pred1)

print(MSE_1)
print(MAE_1)

# ------------------------------------------
# 支持向量机的实现
# ------------------------------------------
# 支持向量机
svr = SVR(kernel='rbf')
# 模型训练、模型预测
y1_pred2 = svr.fit(X1_train, y1_train).predict(X1_test)
# Mean Squared Error（均方误差）
MSE_2 = mean_squared_error(y1_test, y1_pred2)
# Mean Absolute Error（平均绝对误差）
MAE_2 = mean_absolute_error(y1_test, y1_pred2)

print(MSE_2)
print(MAE_2)

# ------------------------------------------
# 加权平均法的实现
# ------------------------------------------
w1 = 1 / MSE_1
w2 = 1 / MSE_2
w1_normalized = w1 / (w1 + w2)
w2_normalized = w2 / (w1 + w2)
```

```
y1_pred = w1_normalized * y1_pred1 + w2_normalized * y1_pred2

print(mean_squared_error(y1_test, y1_pred))
print(mean_absolute_error(y1_test, y1_pred))


# ------------------------------------------------
# 灵敏度分析
# ------------------------------------------------
def analysis(X1_test):
    # 接受过噪音的X_test得分情况
    score_1, score_2, score_3, score_4 = [], [], [], []

    for j in range(0, 4):

        # 备份新的X_test用于噪音处理
        df1 = np.array(X1_test.copy())
        # 打印X_test的维数，方便后面做循环
        m, n = np.array(X1_test).shape
        # 噪音
        error = np.ones(shape=(m, 1))

        # 按0到0.5的比例对X_test进行噪音处理
        for i in np.linspace(0, 0.5, 51):

            # 施加对应比例噪音并添加到X_test上，产生新的测试集特征df2
            error[:, 0] = np.random.uniform(-i * df1[:, j], i * df1[:, j])
            df1[:, j] = df1[:, j] + error[:, 0]
            df2 = pd.DataFrame(df1)
            df2.columns = ['特征1', '特征2', '特征3', '特征4']

            # 构建完数据后预测、打分
            y_pred1 = pd.DataFrame(ridge.predict(df2))
            y_pred2 = pd.DataFrame(svr.predict(df2))
            MSE_1 = mean_squared_error(y1_test, y_pred1)
            MSE_2 = mean_squared_error(y1_test, y_pred2)
```

```
            w1 = 1 / MSE_1
            w2 = 1 / MSE_2
            w1_normalized = w1 / (w1 + w2)
            w2_normalized = w2 / (w1 + w2)

            y_pred = w1_normalized * y_pred1 + w2_normalized * y_pred2

            if j == 0:
                score_1.append(mean_squared_error(y1_test, y_pred))
            elif j == 1:
                score_2.append(mean_squared_error(y1_test, y_pred))
            elif j == 2:
                score_3.append(mean_squared_error(y1_test, y_pred))
            else:
                score_4.append(mean_squared_error(y1_test, y_pred))

    return score_1, score_2, score_3, score_4

score_1, score_2, score_3, score_4 = analysis(X1_test)

plt.figure(figsize=(15, 6))
plt.rcParams["font.sans-serif"] = ["SimHei"]
plt.rcParams['font.size'] = 12 # 字体大小
plt.rcParams['axes.unicode_minus'] = False # 正常显示负号
plt.xlabel("施加噪音阈值百分比")
plt.ylabel("MSE")

plt.plot(np.linspace(0, 0.5, 51), score_1, label="特征1")
plt.plot(np.linspace(0, 0.5, 51), score_2, label="特征2")
plt.plot(np.linspace(0, 0.5, 51), score_3, label="特征3")
plt.plot(np.linspace(0, 0.5, 51), score_4, label="特征4")

plt.title("灵敏度分析: IPI005")
plt.legend()
plt.show()
```

C.7 随机森林的建模与树状图导出

```
# ---------------------------------------------
# 模型准备
# ---------------------------------------------
X = df1.drop("rating", axis=1)
y = df1.rating



# ---------------------------------------------
# 上采样
# ---------------------------------------------
# 创建上采样对象
ros = RandomOverSampler(random_state=42)
# 对数据进行上采样
X_resampled, y_resampled = ros.fit_resample(X, y)

X_train, X_test, y_train, y_test = train_test_split(X_resampled,
   y_resampled, test_size=0.2, random_state=0)



# ---------------------------------------------
# 随机森林建模与评价
# ---------------------------------------------
Model = RandomForestClassifier(max_depth=3, n_estimators=1000)
y_pred = Model.fit(X_train, y_train).predict(X_test)

# 打印分类模型最好的评价系统
print("评估数据结果打印:\n", classification_report(y_test, y_pred))

# ---------------------------------------------
# 树状图导出
# ---------------------------------------------
plt.figure(figsize=(20, 10))
estimator = Model.estimators_[5]
estimator.fit(X_train, y_train)
plot_tree(estimator, filled=True, max_depth=3, feature_names=X.columns)
```

```
plt.title("随机森林分类树状图")
plt.show()
```