

针对新型镇静药物临床实验疗效的分析与预测

摘要

针对某种新型镇静药物的临床实验数据分析是非常必要和重要的。这样的分析可以帮助评估药物的效果、安全性以及剂量效应关系，并为进一步的研发和临床应用提供科学依据。本文将利用**双样本 t 检验**、**BP 神经网络模型**、**逻辑回归模型**等方法对新型镇静药物临床实验疗效进行分析与预测。

针对问题一，分析新药与原药在术中术后不良反应方面有无显著差异，并建立模型对患者术中术后不良反应进行预判。我们选取术中术后不良反应所包含的八个变量作为协变量，将镇静药物种类作为因变量进行**双样本 t 检验**分析，并使用 Python 将分析结果可视化，发现部分不良反应有显著差异，而剩下的不良反应没有显著差异；我们将患者信息和镇静药种类作为协变量，将不良反应作为因变量建立**二元逻辑回归模型**，使用 Python 编程，并使用 ROC 图和 AUC 值作为模型评判指标，结果表明，预测模型较为完善，正确率较高。

针对问题二，分析新药组和原有药组在生命体征方面是否有显著差异，若有显著差异判断是否由新药造成。我们采用不同的权重处理与生命特征值相关的各个属性，并将其结果通过**方差分析**和 **Wilcoson 模型**分析，分别从方差和中位数角度进行差异性比较。对于第二问，我们采用和第一问一样的数据处理方法。然后通过**多元回归模型**和 **Filter 方法**，对它们的差异性进行检验，通过它们的得分来判断哪些属性对生命特征值的影响最大。通过分析我们所得到的结果，我们可以知道新旧药物的类型对生命特征值的影响比较显著。同时，年龄、身高、体重以及是否追加镇静对生命特征值的影响也较大。

针对问题三，对给药后三分钟内的 IPI 数据进行预测。我们分析出 IPI 数据本质是定类数据，于是我们建立 **BP 神经网络模型**，使用 **Matlab** 编程，将用药信息和患者信息作为协变量，将 IPI 数据作为因变量，对题目所给数据进行预测，发现预测结果较为精准，预测正确率合格。另外，为了验证该模型的合理性，我们还使用 **GDBT 模型**进行预测，发现 GDBT 模型预测结果不如 BP 神经网络模型。最后，考虑到影响 IPI 的因素较多，我们认为得到的正确率可以接受。

针对问题四，分析术后满意度可能与哪些因素有关。我们首先将五个术后满意度定类变量处理成一个定量变量，然后使用**斯皮尔曼相关分析法**分析可能影响术后满意度的因素，发现现有数据中时间因素影响力最大，但相关度不高。考虑到题目尚未给出某些可能的影响因素，我们认为该结果是合理的。

最后，我们对模型的**合理性**进行分析，同时，针对各个问题中所构建的模型，均提出评价和改进方案。

关键词： 双样本 t 检验；逻辑回归模型；方差分析；Wilcoson 模型；
BP 神经网络模型；GDBT 模型；斯皮尔曼相关分析法

目录

第1章 问题重述	1
1.1 问题背景	1
1.2 问题提出	1
第2章 问题分析	1
2.1 问题一分析	1
2.2 问题二分析	2
2.3 问题三分析	2
2.4 问题四分析	2
第3章 模型假设	2
第4章 符号说明	3
第5章 模型的建立与求解	3
5.1 问题一的模型的建立与求解	3
5.1.1 双样本 t 检验模型的建立	3
5.1.2 第一部分求解结果	4
5.1.3 二元逻辑回归模型的建立	5
5.1.4 第二部分求解结果	6
5.1.4.1 恶心呕吐预测	7
5.1.4.2 头晕头昏头痛预测	7
5.1.4.3 睡眠乏力预测	8
5.2 问题二模型的建立与求解	8
5.2.1 方差分析及结果分析	8
5.2.2 Wilcoxon 秩和检验	10
5.2.3 多元回归分析	10
5.2.4 filter 分析及结果分析	12
5.3 问题三的模型建立与求解	12
5.3.1 前置分析	12
5.3.2 数据预处理	13
5.3.3 基于 BP 神经网络模型的 IPI 分类预测	14
5.3.4 求解结果展示	16
5.4 问题四的模型建立与求解	19
5.4.1 前置分析	19
5.4.2 数据预处理	20
5.4.3 斯皮尔曼相关性分析模型的引入	20
5.4.4 求解结果分析	21
第6章 模型分析	22
第7章 模型总结与评价	23
7.1 问题一模型的评价与改进	23

7.2 问题二模型的评价与改进	24
7.3 问题三模型的评价与改进	24
7.4 问题四模型的评价与改进	24
第8章 附录(程序所使用的所有代码展示)	25
8.1 问题一代码	25
8.1.1 第一部分的 Python 代码	25
8.1.2 第二部分的 Python 代码	26
8.2 问题二代码	27
8.2.1 第一部分的 Python 代码	27
8.2.2 第二部分的 Python 代码	30
8.3 问题三的 Matlab 代码	32
8.4 问题四的 Python 代码	34

第1章 问题重述

1.1 问题背景

随着现代医学技术的不断进步，新型药物的研究和开发已经成为医药行业的重要领域之一。在新药物研究的过程中，临床实验是至关重要的环节之一，因为它可以提供有关新药物的有效性、安全性和剂量等方面的关键信息。在临床实验中，研究人员会对参与者进行药物治疗，并收集数据来评估药物的效果和安全性。因此，针对某种新型镇静药物的临床实验数据分析是非常必要和重要的，它有助于评估药物的效果、安全性和剂量效应关系，为进一步的研发和临床应用提供科学依据。

1.2 问题提出

在肠胃微创手术中，通常需要使用到局部镇静和镇痛药物。其中，传统的局部镇静药物被称为“B 药”，而某药物研发中心开发了一种新型药物“R 药”。新药物投入使用需要经过两个阶段：生物试验和临床试验。本题所涉及的是新型药物“R 药”在医院进行的非干预性研究。

通过对新型药物和传统镇静药物在临床试验中的表现数据的分析，可以对病患的 IPI 等生命体征、不良反应和满意度等方面进行预估，为药物选择提供了重要参考，并为医师和病患提供了预测的依据。本文基于附件 1 提供的新药物“R 药”和原有药物“B 药”对比的临床实验研究数据及其他附件资料，致力于解决如下问题：

问题一. 是否存在新药组和原有药物组在术中和术后 24 小时内不良反应的显著差异，并且能否建立数学模型进行预测。

问题二. 是否存在新药组和原有药物组在生命体征数据方面的显著差异；若存在，能否确定是由于新药造成还是其他因素造成。

问题三. 尝试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。

问题四. 基于现有数据是否能够找出术后满意度与哪些因素有关，以及它们之间的关系。

第2章 问题分析

2.1 问题一分析

本问第一部分要求判断是否存在新药组（R 药）和原有药物组（B 药）在术中和术后 24 小时内不良反应的显著差异。这需要我们使用合适的统计学方法来分析数据，以确定是否存在新药组和原有药物组之间的显著差异。对于本问题，本文考虑使用**双样本 t 检验**的统计方法进行显著性差异判断，通过 **Python** 语言进行编程，对输出结果进行分析，进而判断出是否存在显著差异。

本问第二部分要求建立数学模型，对患者术中、术后 24h 的不良反应进行预判。本文使用**二元逻辑回归模型**对输入特征进行建模，然后使用已知的训练数据来调整模型参数，以最大化模型对训练数据的正确分类。之后利用二元逻辑回归模型，就可以将患者的基本信息和镇静药物种类输入到模型中，将不良反应（如恶心呕吐等）作为输出标签，从而预测他们是否会出现不良反应。

2.2 问题二分析

本问第一部分要求对 R 药组和 B 药物组在生命体征数据方面的是否存在显著差异进行判断。这需要我们首先对于能够体现生命体征的数据进行处理，通过题目给定的信息，我们能够知道 IPI 指标是刻画生命体征的核心指标，所以在处理的过程中，应突出 IPI 指标的地位，先对数据进行权重处理。对于本问题，本文考虑使用 **Wilcoxon 秩和检验和方差分析法**对显著性差异进行判断，通过 **Python** 语言进行编程，对输出结果进行分析，进而判断出是否存在显著性差异。

本问第二部分要求判断显著差异的产生是由于新旧药物还是其他因素。这需要我们先对数据进行归一化等处理，然后进行**多元回归分析**和 **Filter 方法**，输出各特征值对生命体征值的影响，通过 **Python** 语言进行编程，对输出结果进行分析，从而进行判断。

2.3 问题三分析

本题要求根据患者信息和用药信息对给药后 3 分钟以内的 IPI 数据进行预测。首先需要对数据进行预处理，患者信息可以采用问题一处理后的数据，用药信息应转换成定量数据。接下来通过分析得知 IPI 本质是一个定类数据，所以确定数据类型为定量和定类之间的关系。本题考虑使用 **BP 神经网络模型**，通过 **Matlab** 进行编程，使用已知的数据进行训练来调整模型参数，期望得到最高的正确率，最后我们根据均方根误差的大小判断模型是否合理。

2.4 问题四分析

本问要求找出术后满意度与哪些因素有关，以及它们之间的关系。由于题目中给出的数据不是定量数据，于是我们先对数据进行预处理，将处理后的数据作为新的变量存放起来。然后我们进行假设，先猜测某些因素可能与术后满意度有关，分析后排除掉无关因素，再重新作新的假设。由于本题问的是术后满意度与哪些因素有关，故我们使用斯皮尔曼相关分析进行解决，通过 **Python** 语言编程得到结果，再与 spss 分析得到的结果进行对比，若结果一致则做出结论；不一致则修改变量重新进行分析。

第 3 章 模型假设

1. 假设题目所有数据来源皆真实可靠，题目中假设皆真实合理。
2. 假设样本的数据满足正态分布、样本量足够大，且两个样本的方差大致相等，满足双样本 t 检验要求。
3. 对于每个样本，观测值之间应该是独立的。
4. 对于每个样本，观测值应该是由随机抽样得到的。
5. 数据特征与目标变量之间存在一定的相关性，即特征对目标变量有一定的解释能力。
6. 数据特征之间不能存在过高的相关性，否则会造成冗余特征的存在，影响模型的解释能力和性能。
7. 数据集的大小应该足够大，否则可能会影响特征选择的准确性。

第4章 符号说明

符号	说明
df1	表示存储 B 药组的数据
df2	表示存储 R 药组的数据
arr[i][j]	表示合并两药组数据，便于进行检验
result1	将 df1 中的多种生命值特征综合处理后的结果
result2	将 df2 中的多种生命值特征综合处理后的结果
stat	指的是秩和统计量，表示两个样本的秩和之差，用与检验两个样本的差异是否显著
p	表示检验的显著性水平
f_value	F 统计量，表示组间方差和组内方差的比值
P_value	统计学中的显著性水平，表示得到当前样本结果的概率
x	经过处理得到的可能造成生命特征值显著性的特征的数据化表示
result	表示各种生命特征值经过一定的权重运算后得到的一个有代表性的结果

第5章 模型的建立与求解

5.1 问题一的模型的建立与求解

5.1.1 双样本 t 检验模型的建立

(1) 首先，我们对于新药组和原有药物组两个样本均值是否存在显著差异提出假设。假设表示为：

$$H_0 : \mu_1 = \mu_2$$

$$H_a : \mu_1 \neq \mu_2$$

解释 5.1

其中， H_0 表示原假设， H_a 表示备择假设。 μ_1 表示 B 药样本所属的总体的均值， μ_2 表示 R 药样本所属的总体的均值



(2) 接着，我们收集数据并计算样本均值和标准差：将“B 药”组和“R 药”组分别存储为 $df1$ 和 $df2$ 。将 $df1$ 和 $df2$ 转换为 NumPy 数组，再将它们合并为一个二维数组 $arr[i][j]$ ，以便进行双样本 t 检验。从两个独立样本中各自收集‘术中呛咳’；‘术中体动’；‘术中其他’；‘术后恶心’；‘术后头痛’；‘术后嗜睡’；‘术后腹胀’；‘术后其它’的数据，并计算两个样本的均值和标准差。公式如下所示

$$X_1 = \frac{1}{n_1} \sum_{i=1}^N x_1 \quad (5.1)$$

$$X_2 = \frac{1}{n_2} \sum_{i=1}^N x_2 \quad (5.2)$$

$$S_1 = \sqrt{\frac{\sum_{i=1}^n (x_1 - X_1)^2}{n_1 - 1}} \quad (5.3)$$

$$S_2 = \sqrt{\frac{\sum_{i=1}^n (x_i - X_2)^2}{n_2 - 1}} \quad (5.4)$$

解释 5.2

其中， x_1 和 x_2 分别表示两个样本的观测值， n_1 和 n_2 分别表示两个样本的大小， X_1 和 X_2 分别表示两个样本的均值， S_1 和 S_2 分别表示两个样本的标准差。



(3) 计算 t 统计量：通过使用以下公式 5.5 计算 t 统计量：

$$t = \frac{(X_1 - X_2)}{SE} \quad (5.5)$$

$$SE = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}} \quad (5.6)$$

解释 5.3

其中， X_1 和 X_2 分别表示两个样本的均值， SE 表示标准误差。



(4) 最后，我们计算 P 值并做出决策：如果 P 值小于预先设定的显著性水平（通常为 0.05），则拒绝原假设，接受备择假设，即认为两个样本的均值存在显著差异。否则，接受原假设，认为两个样本的均值不存在显著差异。

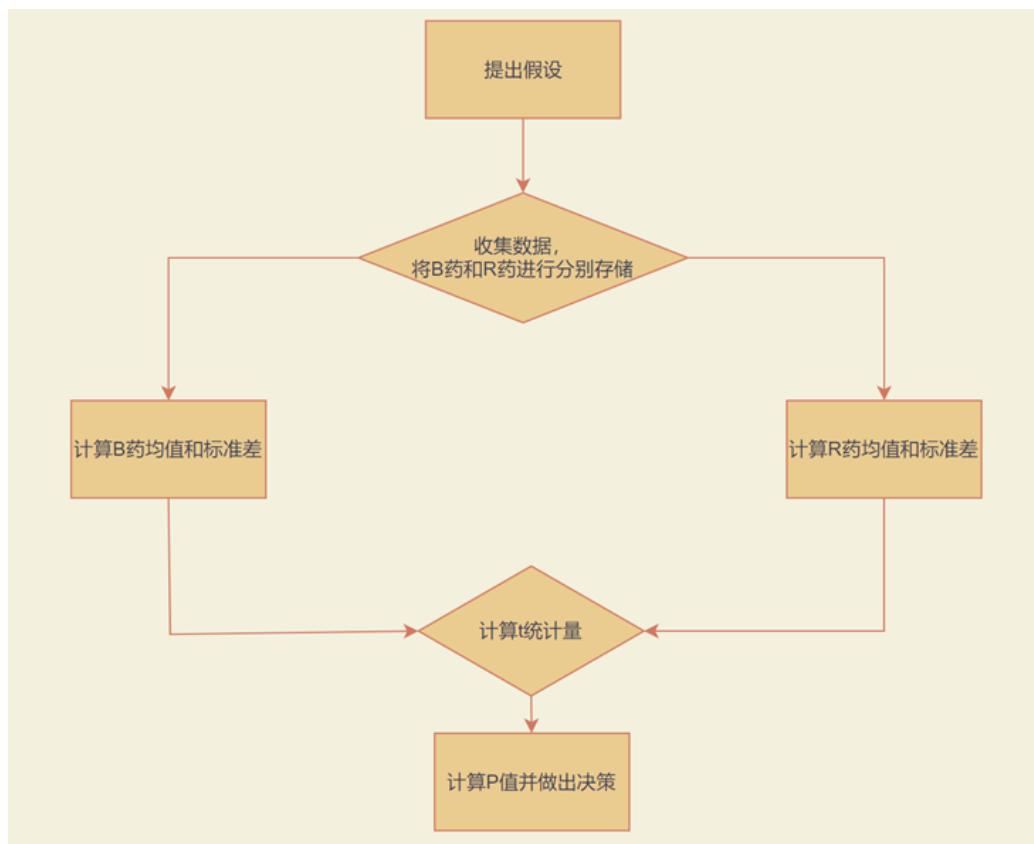


图 5.1: 流程图

5.1.2 第一部分求解结果

根据上述数学模型，我们编写 Python 程序，利用 matplotlib 库绘制一个条形图，横坐标为不同的反应（术中呛咳、术中体动等），纵坐标为对应的 P 值。

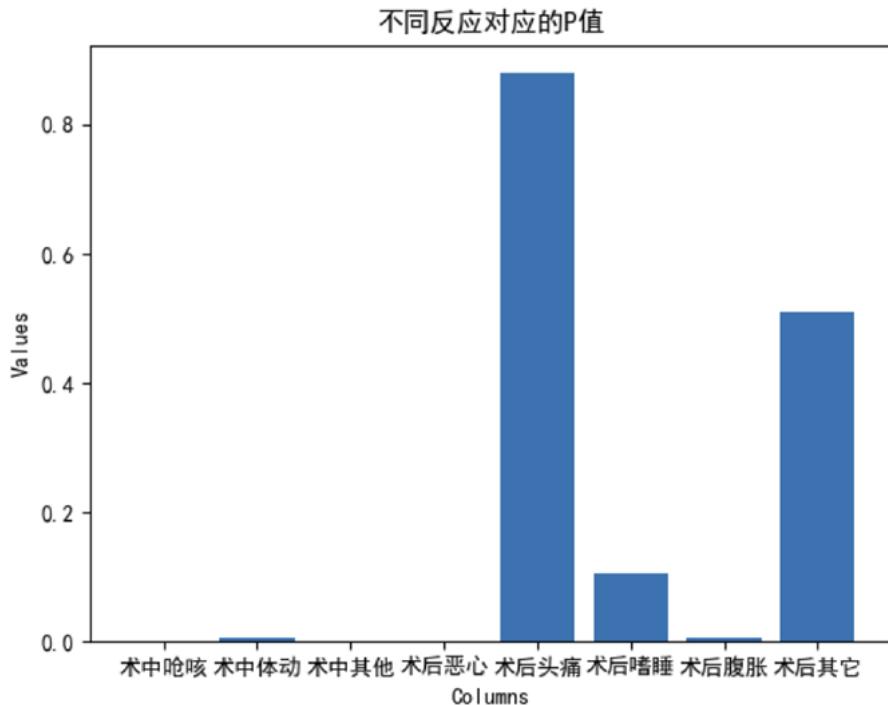


图 5.2: 不同反应对应的 P 值图

结论 5.1

经过以上分析，我们得出如下结论：对于呛咳、体动、术中其他、是否出现了恶心呕吐的情况、有没有出现腹胀腹痛的情况，不同的镇静药名称存在显著性差异，但差异程度呈现为弱程度差异（差异程度较小）。

5.1.3 二元逻辑回归模型的建立

二元逻辑回归中的“二元”指因变量为二分变量，逻辑回归指对目标概率进行对数几率变换。二元逻辑回归是因变量为二分类变量的线性回归分析，要求先将目标概率进行对数几率变换，这样就保证了当概率在(0, 1)取值时，对数几率转换值可以取任意实数，避免了线性概率模型的结构缺陷。

(1) 首先，我们设 Y 为二分因变量，取值“1”代表发生对应的不良反应，“0”代表不发生对应的不良反应，自变量(X_i)为单因子信息量值。记发生不良反应的条件概率为 p，把 p 的某个函数 $f(p)$ 假设为变量的函数形式，进行对数几率变换：

$$f(p) = \ln \frac{p}{1-p} \quad (5.7)$$

(2) 由此可得，二元逻辑回归模型为：

$$\ln \frac{p}{1-p} = \beta_0 + \beta_1 X_1 + \cdots + \beta_i X_i \quad (5.8)$$

解释 5.4

其中： $\beta_0, \beta_1, \dots, \beta_i$ 为逻辑回归系数。

(3) 将公式 5.8 中对 p 求解，即可得到对应不良反应发生的概率：

$$p = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^m \beta_i \times X_i)}} \quad (5.9)$$

根据上述模型的建立，我们编写 Python 程序求解，具体流程如下

1. 读取数据集并去除缺失值
2. 划分训练集和测试集，其中测试集占总数据的 20%。
3. 建立逻辑回归模型，其中 `multi-class='multinomial'` 表示多元逻辑回归模型，`max_iter=3000` 表示最大迭代次数为 3000 次。
4. 使用训练好的模型来预测测试集的结果。
5. 计算模型的准确度
6. 计算预测概率和混淆矩阵
7. 绘制 ROC 曲线并计算 AUC 值

解释 5.5

对于 ROC 曲线，曲线下方的面积 (Area Under Curve, AUC) 值越大，表示模型的分类性能越好，AUC 的取值范围在 0.5 到 1 之间，其中 0.5 表示随机猜测，1 表示完美分类。因此，AUC 值越接近 1，模型的性能越优秀。

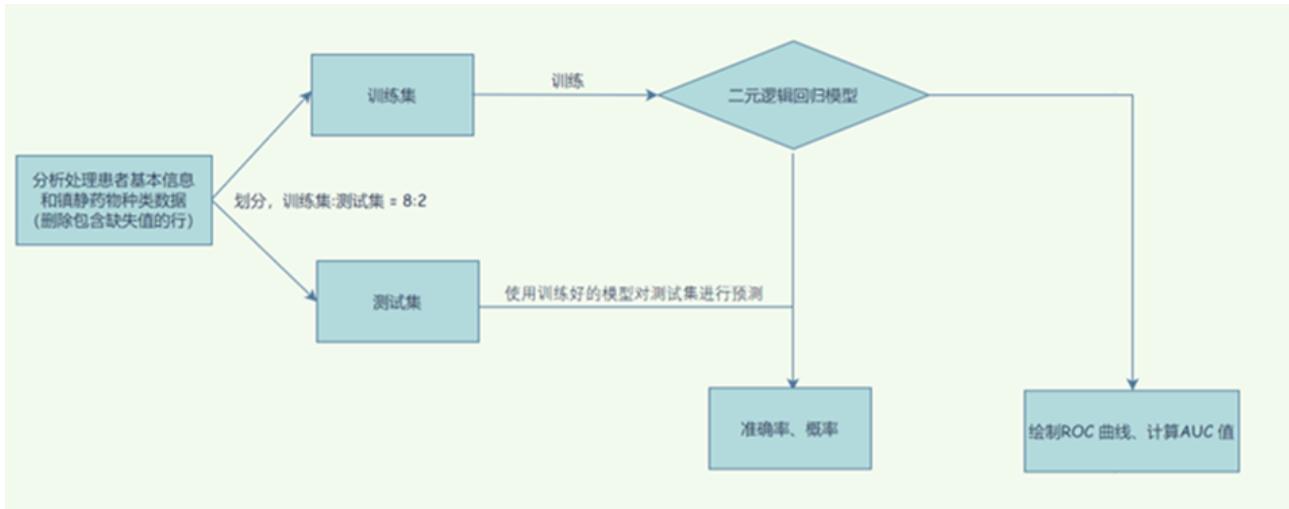


图 5.3: 流程图

5.1.4 第二部分求解结果

根据建立的二元逻辑回归模型及编写的 Python 程序，本文对患者术中、术后 24h 的恶心呕吐、头晕头昏头痛和嗜睡乏力三种不良反应进行预测。

5.1.4.1 恶心呕吐预测

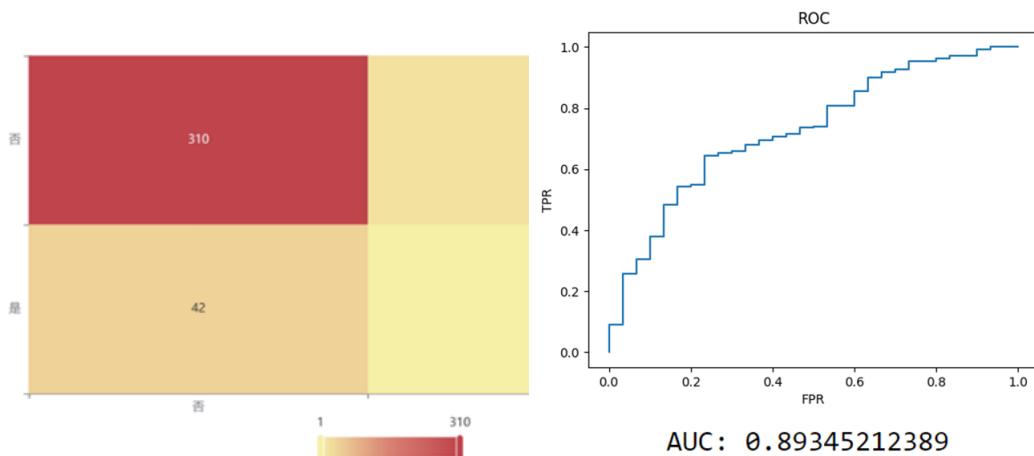


图 5.4: 混淆矩阵热力图、ROC 和 AUC 值图

结论 5.2

通过分析混淆矩阵热力图我们得到如下结论：实际未恶心呕吐预测为未恶心呕吐的人数为 310 人，实际恶心呕吐预测为恶心呕吐的人数为 1 人，预测正确率为 83% 左右。

通过分析 ROC 图和 AUC 值我们发现，AUC 值接近 0.9，说明该预测模型比较完美。

5.1.4.2 头晕头昏头痛预测

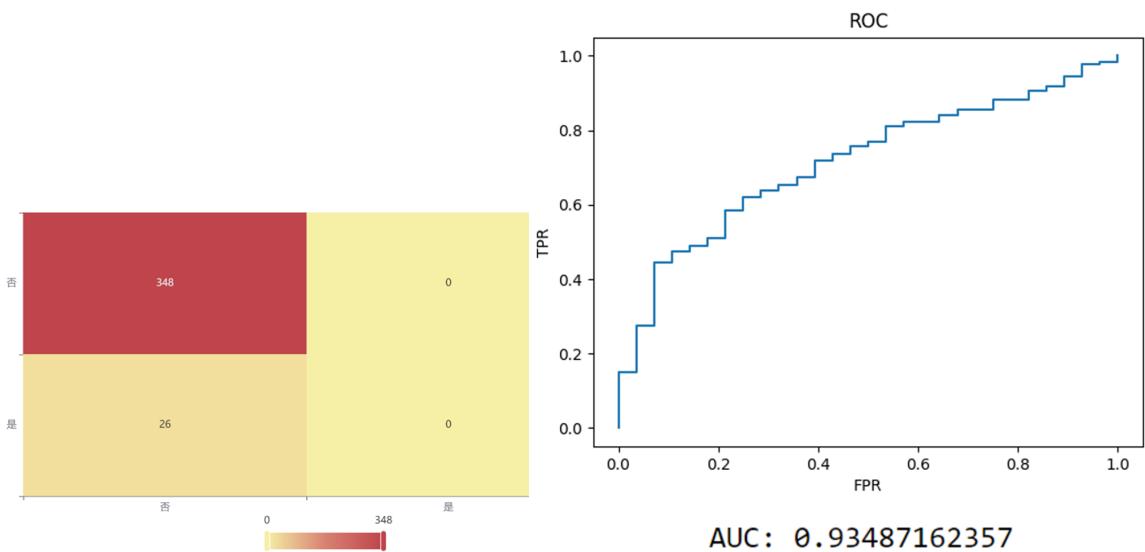


图 5.5: 混淆矩阵热力图、ROC 和 AUC 值图

结论 5.3

通过分析混淆矩阵热力图我们得到结论：实际未头晕头昏头痛预测为未头晕头昏头痛的人数为 348，预测准确率为 0.93 左右，准确率较高。

另外，AUC 值高达 0.935，说明该预测模型接近完美。

5.1.4.3 嗜睡乏力预测

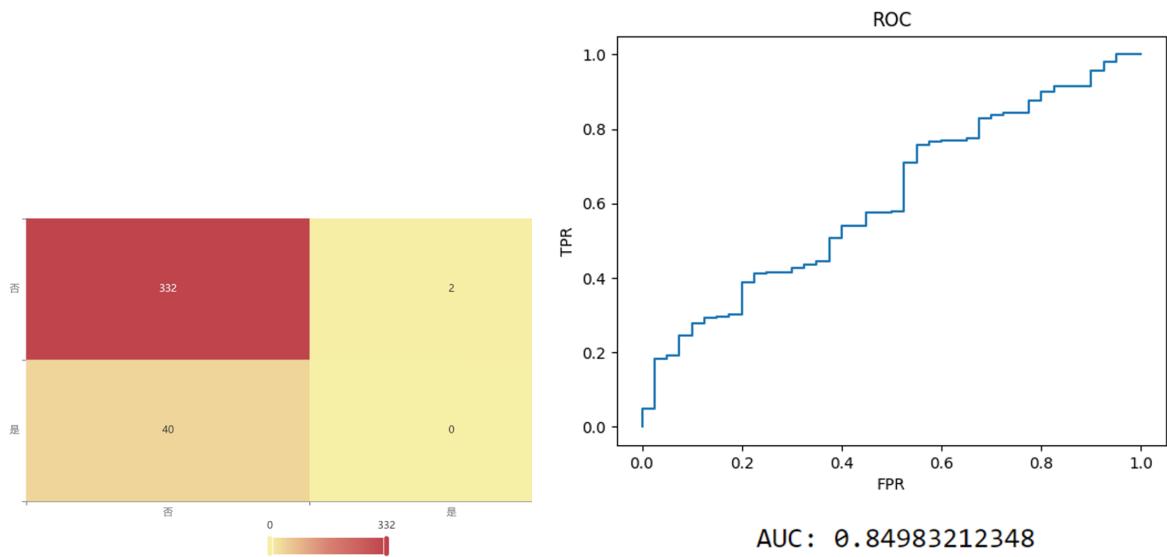


图 5.6: 混淆矩阵热力图、ROC 和 AUC 值图

结论 5.4

通过分析混淆矩阵热力图我们发现，实际未嗜睡乏力预测为未嗜睡乏力的人数为 332，预测正确率接近 0.89，正确率较高。

通过分析 ROC 图和 AUC 值我们发现，AUC 值接近 0.85，说明该预测模型比较完美。

5.2 问题二模型的建立与求解

5.2.1 方差分析及结果分析

具体步骤如下：

- (1) 确定数据类型为定类和定量之间的关系。score 是通过权重计算得到的具有代表性的生命特征值，result1 中是选定的各个与生命特征值相关的属性。
- (2) 确定为多因素方差分析，因为要比较的是多个自变量和一个因变量之间的关系，所以通过多因素方差进行实现。
- (3) 方差分析要求分析的属性满足正态性，所以使用 Shapiro-Wilk 正态性检验方法进行检验。

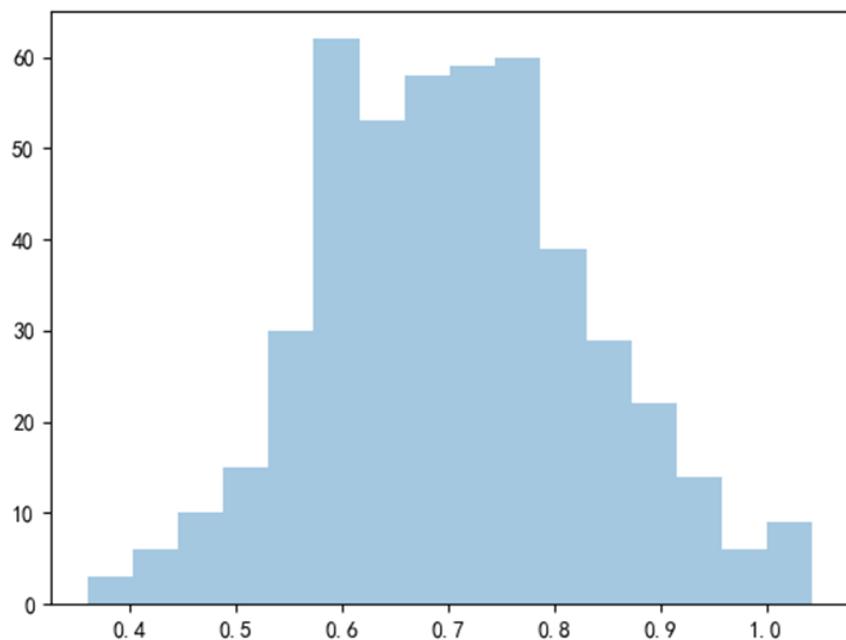


图 5.7: 属性正态性分布图

对 B 药和 R 药的生命体征值做散点图

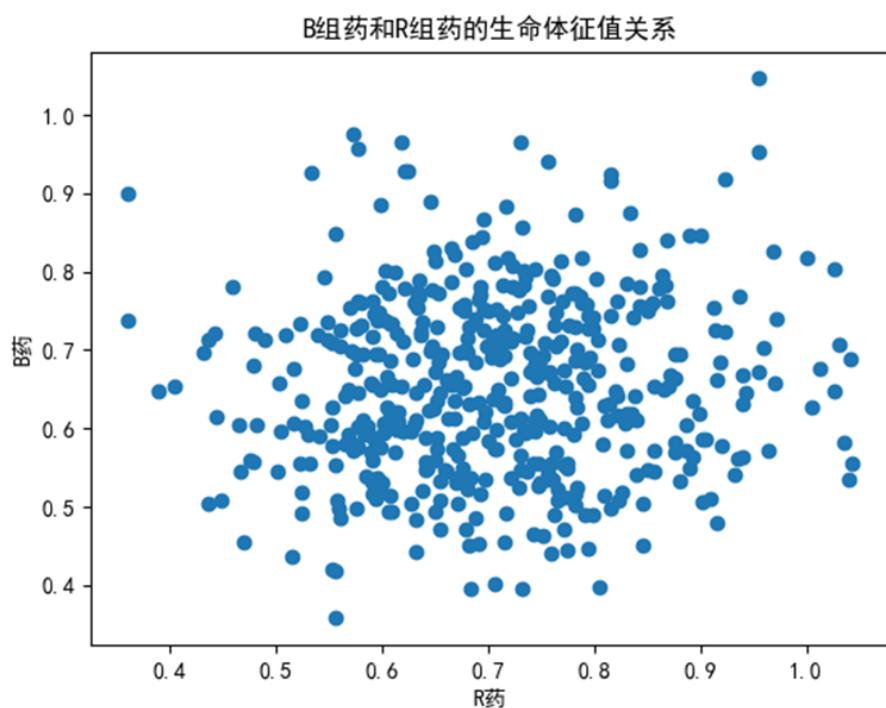


图 5.8: B 药和 R 药的生命体征值散点图

最终得出: F-value: 45.00989020407864 ; p-value: 3.367095594597044e-11

结论 5.5

因为差异性水平小于 0.05，所以认为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。

**5.2.2 Wilcoxon 秩和检验**

Wilcoxon 秩和检验是一种非参数假设检验的方法，用于检验两个相关样本或配对样本之间的差异是否显著。

- (1) 确定零假设和备择假设。零假设指两组样本没有显著差异，备择假设指两组样本存在显著差异。
- (2) 对两组样本进行配对。
- (3) 对配对差值按绝对值大小排序，然后给排名。
- (4) 计算正排名和负排名之和，取较小值作为 Wilcoxon 秩和检验的统计量。
- (5) 根据样本数和显著性水平查找 Wil 秩和检验的临界值。
- (6) 计算检验的 p 值，如果 p 值小于显著性水平，则拒绝零假设，认为两组样本存在显著差异，否则接受零假设，认为两组样本没有显著差异。

最终得出：stat=37046.000；p=0.000

结论 5.6

stat 中存放的是总样本的排名和，因为我们的计算数据量是 476*2，也就是说我们的 stat 量级应该是 476*476，由已知的一个样本的秩和为 37046，所以另一个样本的秩和应该是 22 万左右，故可侧面验证二者的差异显著。

同时，p 值小于显著性水平 0.05，故认为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。

**5.2.3 多元回归分析**

多元回归分析是一种建立自变量和因变量之间关系的统计方法，它可以用来解决一个或多个自变量对因变量的影响程度以及各自变量的相互作用的问题。在处理之前，先对自变量的相关性检验。具体步骤如下：

- (1) 收集数据：收集一定数量的数据，包括自变量和因变量。本题中就是可能对生命特征值造成影响的各种属性，以及对于生命特征值按照一定的权重将其整合为一列数据。
 - (2) 检查数据：对数据进行清理和预处理，包括缺失值的处理、异常值的处理、数据变换等。
 - (3) 选择模型：根据问题的特点和数据的分布情况，我们选择线性回归模型里面的 OLS 模型。
- 数学表达式如 5.10 所示：

$$y = x_0 + x_1 * a_1 + x_2 * a_2 + \dots + x_k * a_k + \theta \quad (5.10)$$

OLS 模型的主要假设为：

线性性假设：自变量与因变量之间存在线性关系；

独立性假设：样本之间相互独立，误差项之间也相互独立；

方差齐性假设：误差项的方差在各个自变量的取值上是相等的；

正态性假设：误差项服从正态分布。

OLS 模型的求解过程是通过最小化残差平方和来确定自变量的系数，即：

$$\min_{\beta_0, \beta_1, \dots, \beta_k} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \dots - \beta_k x_{ik})^2 \quad (5.11)$$

- (4) 拟合模型：使用拟合方法拟合回归模型，得到模型的系数和截距。

(5) 验证模型：使用统计学方法验证模型的质量和可靠性，包括残差分析、方差分析、回归系数的显著性检验等。残差分析的结果是下表中的 Omnibus, Jarque-Bera, Durbin-Watson，一般希望这些值越小越好。

- (6) 使用模型：使用已经拟合好的模型进行预测和决策，得出结论并作出相应的决策。

OLS Regression Results						
Dep. Variable:	y	R-squared:	0.100			
Model:	OLS	Adj. R-squared:	0.090			
Method:	Least Squares	F-statistic:	10.50			
Date:	Tue, 02 May 2023	Prob (F-statistic):	2.09e-21			
Time:	01:24:23	Log-Likelihood:	957.63			
No. Observations:	1245	AIC:	-1887.			
Df Residuals:	1231	BIC:	-1815.			
Df Model:	13					
Covariance Type:	nonrobust					
coef	std err	t	P> t	[0.025	0.975]	
const	0.8210	0.107	7.705	0.000	0.612	1.030
年龄	-0.0018	0.000	-6.597	0.000	-0.002	-0.001
身高	-0.0011	0.001	-1.587	0.113	-0.002	0.000
体重	0.0016	0.000	4.599	0.000	0.001	0.002
性别 男	0.0037	0.010	0.363	0.717	-0.016	0.024
有无手术史	0.0059	0.012	0.479	0.632	-0.018	0.030
有无既往史	-0.0086	0.010	-0.869	0.385	-0.028	0.011
偶尔吸烟者	0.0032	0.015	0.220	0.826	-0.025	0.032
经常吸烟者	0.0096	0.013	0.740	0.459	-0.016	0.035
是否酗酒	-0.0042	0.011	-0.387	0.699	-0.025	0.017
有无PONV_有	0.0720	0.064	1.126	0.260	-0.053	0.197
有无晕动史_有	-0.0415	0.057	-0.730	0.465	-0.153	0.070
有无追加镇静_有	0.0282	0.006	4.338	0.000	0.015	0.041
镇静药名称_R药	0.0394	0.007	5.935	0.000	0.026	0.052
Omnibus:	4.890	Durbin-Watson:	1.879			
Prob(Omnibus):	0.087	Jarque-Bera (JB):	4.874			
Skew:	0.153	Prob(JB):	0.0874			
Kurtosis:	3.002	Cond. No.	6.17e+03			

图 5.9: 多元回归分析结果图

解释 5.6

Model: 提供了回归模型的基本信息，包括用到的自变量个数、使用的拟合方法和截距的存在与否。

Residuals: 提供了残差相关的统计信息，包括观测值的个数、均值、标准差、最小值、最大值等等。

Coefficients: 提供了每个自变量对因变量的影响程度（回归系数）、标准误差、t 值、对应的 p 值和置信区间等信息。在这里，我们需要关注的是 t 值和 p 值。t 值反映了该自变量对因变量的影响是否显著，t 值越大，说明该自变量的影响越显著。p 值反映了该自变量的系数是否显著不为 0，p 值越小，说明该自变量的系数显著不为 0。

Omnibus: 提供了残差是否正态分布的检验结果，一般希望这个值越小越好。

Durbin-Watson: 提供了残差是否存在自相关的检验结果，一般希望这个值越接近 2 越好。

Jarque-Bera: 提供了残差正态性的检验结果，一般希望这个值越小越好。

Cond. No.: 提供了多重共线性的程度，一般希望这个值小于 10。



结论 5.7

在多元回归分析中，一般采用 t 检验来判断回归系数的显著性水平，对应的 p 值表示该系数对因变量的影响是否显著。通常情况下，若 p 值小于 0.05，就认为该系数对因变量的影响是显著的。此外，还可以通过观察各个变量的置信区间来判断它们的显著性。如果一个变量的置信区间不包括 0，则认为该变量对因变量的影响是显著的。

因此，通过上表，我们可以看出年龄，体重，有无追加镇静以及镇静药名称对生命特征值影响显著。



5.2.4 filter 分析及结果分析

具体步骤如下：

- (1) 通过 SelectKBest 方法选择最相关的 k 个特征，其中 k=3 是通过参数设置的，可以根据实际需求进行调整。
- (2) 使用 fit 方法拟合选择器对象，并在所有特征上计算每个特征的相关得分。将得分保存在 dfscores 变量中，并将特征名称保存在 dfcolumns 变量中。
- (3) 通过 concat 方法将特征名称和得分连接起来，并在连接后的数据帧中添加列名为'Feature' 和'Score'。
- (4) 对连接后的数据帧按照'Score' 列进行降序排列，并输出前 3 个特征的名称和相关得分。输出所有特征的名称和相关得分。并对全部进行图像输出。

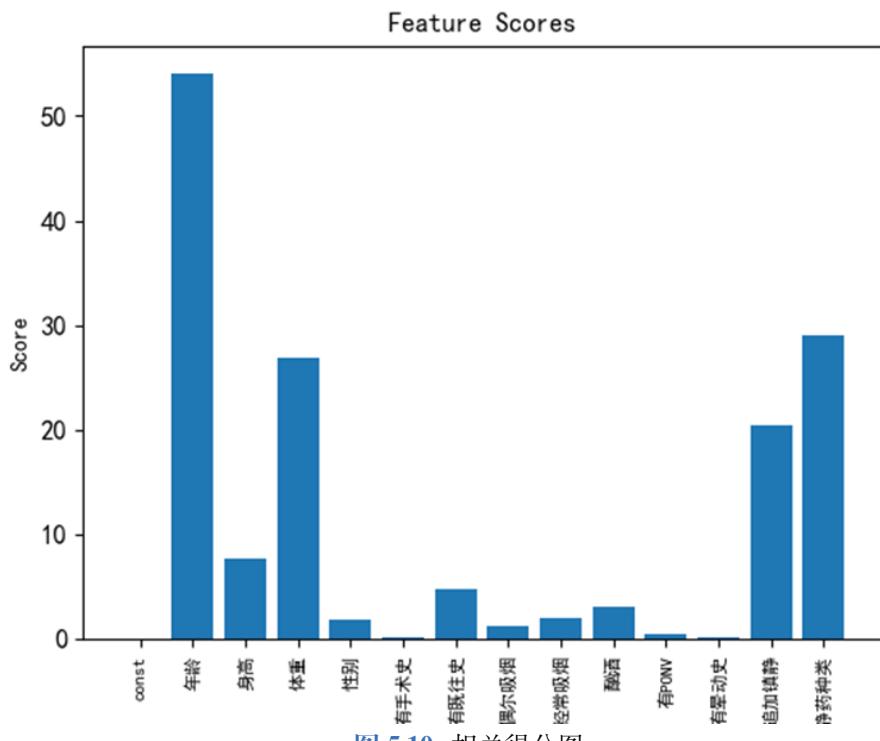


图 5.10: 相关得分图

结论 5.8

在这里得分越高，就表明该属性对生命体征数据方面造成了越大的影响，通过分析可得年龄，身高，体重有无既往史以及有无追加镇静和镇静药的种类对生命特征值有较大的影响

5.3 问题三的模型建立与求解

5.3.1 前置分析

本问我们需要对 IPI 数据训练出一个预测模型。其中，协变量为用药信息和患者信息，因变量为三分钟内的 IPI 数据。我们已经在问题一中对患者信息进行了数据处理，本问可以继续使用问题一中的数据。

对于用药信息：

3. 手术信息采集

镇痛药	诱导剂量: 镇静药诱导剂量 A 药 $7\mu\text{g}/\text{kg}$ 追加剂量: 有无追加镇静无 <input type="checkbox"/> 有 <input checked="" type="checkbox"/> 追加次数: 镇静追加次数 次 每次追加剂量: 镇静药追加剂量 μg 总剂量: 镇静药总剂量 μg
镇静药	诱导剂量分组: 镇痛药诱导剂量 R 药 $0.15\text{mg}/\text{kg}$ <input type="checkbox"/> B 药 $1.5\text{mg}/\text{kg}$ 组 <input checked="" type="checkbox"/> 追加剂量: 有无追加镇痛无 <input type="checkbox"/> 有 <input checked="" type="checkbox"/> 镇痛追加次数: 次 每次追加剂量: 镇 痛药追加剂量 mg 总剂量: 镇痛药总剂量 mg

图 5.11: 用药信息图

根据题目背景:

镇静药物的使用会抑制神经的敏感度, 比如会对呼吸有一定的抑制作用, 从而造成血氧含量下降, 病患的生命体征会有所下降, 其下降的程度对术后恢复效

图 5.12: 题目背景图

镇静药物的使用会使病患的生命体征下降, 故用药信息我们可以只考虑镇静药物的相关信息, 如药物名称、诱导计量、是否追加等等。

对于 IPI 数据, 题目要求我们对用药后三分钟以内的 IPI 值进行预测, 由于用药前 IPI (即 IPI000) 是正常值, 所以我们选取具有代表性的 IPI2 和 IPI3 进行分析和预测。

工作簿	工作表	名称	单元格	值	公式
附件1.xls	胃镜项目全数据		\$DE\$1	IPI7	
附件1.xls	胃镜项目全数据		\$DM\$1	IPI10	
附件1.xls	胃镜项目全数据		\$DU\$1	IPI15	
附件1.xls	胃镜项目全数据		\$EC\$1	IPI20	
附件1.xls	胃镜项目全数据		\$EK\$1	IPIjieshu	
附件1.xls	胃镜项目全数据		\$EM\$1	IPI达到4分时间	
附件1.xls	胃镜项目全数据		\$EN\$1	IPI最低值	

图 5.13: IPI 数据图

结论 5.9

观察 IPI 数据得知: 去掉最后两项 IPI 值后, IPI 本质上是个 14 分类数据, 其并不是定量数据, 因此我们应当建立分类预测模型。

5.3.2 数据预处理

除了 IPI 数据是定类数据外, 其余数据都应处理成定量数据。

对于这三项数据, 我们在 excel 里使用“查找和替换”功能, 将 B 药替换成 0, R 药替换成 1; 将 $1.5\text{mg}/\text{kg}$ 替换成 1.5, $0.15\text{mg}/\text{kg}$ 替换成 0.15; 将“无”替换成 0, “有”替换成 1, 得到处理后的数据如下:

镇静药名称	镇静药诱导剂量	有无追加镇静	镇静药名称	镇静药诱导剂量	有无追加镇静
B 药	1. 5mg/kg	无	0	1. 5	0
B 药	1. 5mg/kg	无	0	1. 5	0
R 药	0. 15mg/kg	有	1	0. 15	1
R 药	0. 15mg/kg	无	1	0. 15	0
R 药	0. 15mg/kg	无	1	0. 15	0
R 药	0. 15mg/kg	有	1	0. 15	1
R 药	0. 15mg/kg	无	1	0. 15	0
R 药	0. 15mg/kg	有	1	0. 15	1
R 药	0. 15mg/kg	无	1	0. 15	0
R 药	0. 15mg/kg	有	1	0. 15	1
R 药	0. 15mg/kg	有	1	0. 15	1
R 药	0. 15mg/kg	无	1	0. 15	0
R 药	0. 15mg/kg	有	1	0. 15	1
R 药	0. 15mg/kg	有	1	0. 15	1
R 药	0. 15mg/kg	无	1	0. 15	0

图 5.14: 初始数据图 (左)、替换后数据图 (右)

另外，原始术中存在 IPI 所在列为缺省值的行，我们在 excel 中使用“Ctrl+G”快捷键，点击“定位条件”，将“空值”前面的方框打勾，然后鼠标右键点击任意一个空值单元，以行的方式删除即可。最终得到的不含空值的数据共有 1137 行：

图 5.15: 最终数据图

5.3.3 基于 BP 神经网络模型的 IPI 分类预测

BP 神经网络是前向误差反向传播的神经网络，实质上是一种高度非线性映射，即：输入参数空间映射到输出空间（预报值），包含了各单元之间联结强度的知识网络。

BP 神经网络结构最基本的成分是神经元，是由多个网络层的神经元链接计算组成的，一般包括了输入层、输出层和若干隐藏层，其运行基本逻辑是由权重与偏移项构成线性运算，再作用 Sigmoid 激活函数，得到该神经元连接的下一个神经元上的值，然后根据输出层各个输出神经元误差之和最小化输出层累计误差，采用训练以调整连接权值的方法，反向地去调整连接权重进而达到网络收敛稳定的目的

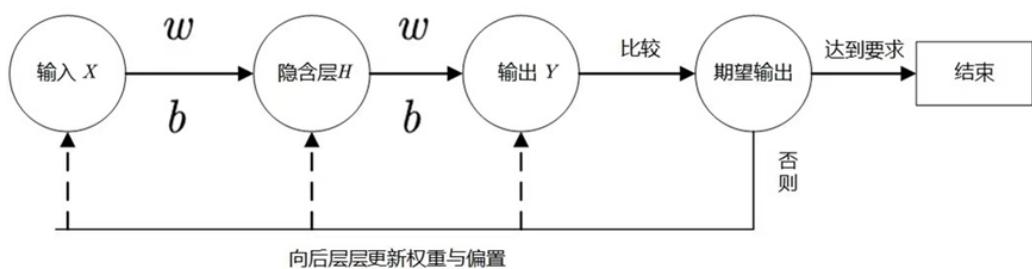


图 5.16: 算法流程图

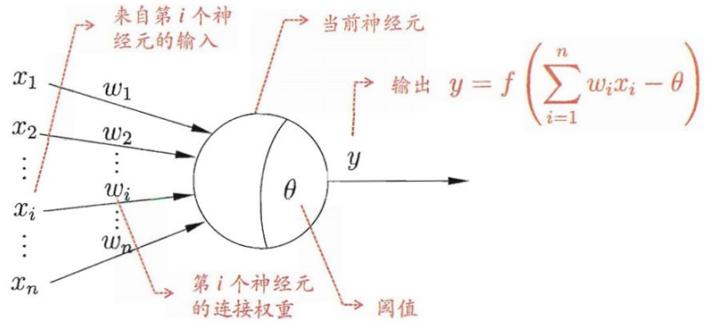


图 5.17: 神经元模型图

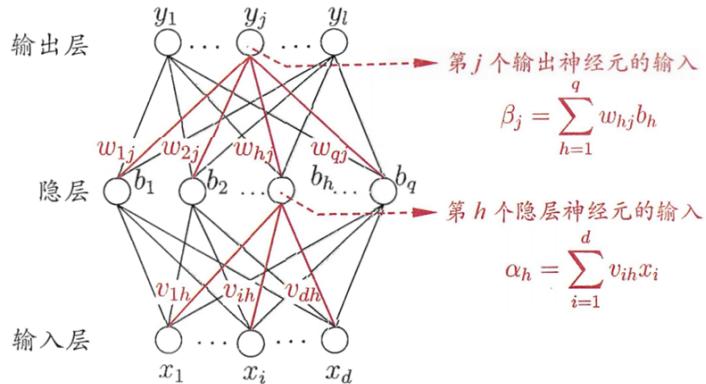


图 5.18: 神经网络基础架构图

正向传播过程：从输入层到隐藏层公式如下：

$$\alpha_h = \sum_{i=1}^d v_{ih} x_i + \theta_h \quad (5.12)$$

从隐藏层到输出层公式如下：

$$\beta_j = \sum_{h=1}^q w_{hj} b_h + \theta_j \quad (5.13)$$

反向传播过程，误差计算公式如下：

$$\beta_j = \sum_{h=1}^q w_{hj} b_h + \theta_j E = \frac{1}{2} \sum_{k=1}^2 (y_k - T_k)^2 \quad (5.14)$$

5.3.4 求解结果展示

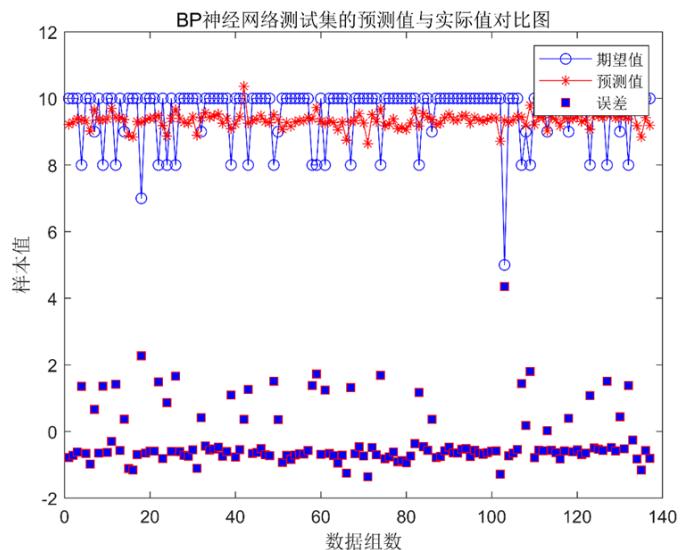


图 5.19: BP 神经网络测试集的预测值与实际值对比图

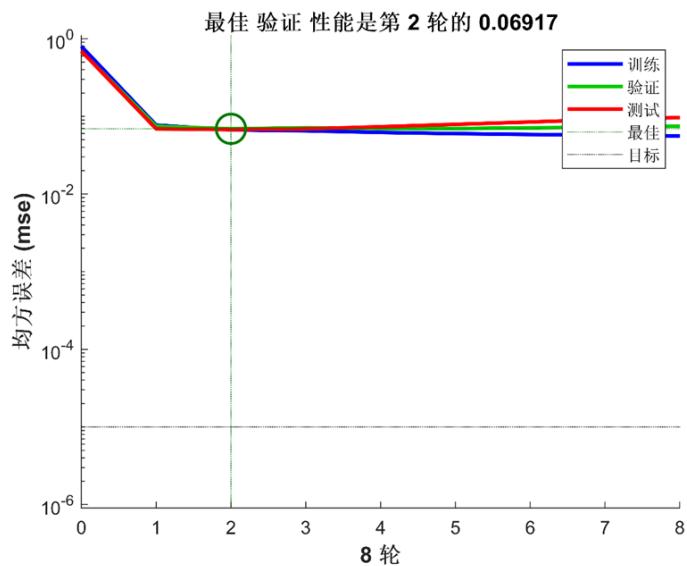


图 5.20: 性能图

-----误差计算-----
 隐含层节点数为 10 时的误差结果如下：
 平均绝对误差 MAE 为： 0.80023
 均方误差 MSE 为： 0.86041
 均方根误差 RMSE 为： 0.92759

图 5.21: MAE、MSE、RMSE 结果图

结论 5.10

可见，均方根误差位于 0.1 之间，可以接受。

	准确率	召回率	精确率	F1
训练集	0.631	0.631	0.399	0.489
测试集	0.784	0.784	0.614	0.689

图 5.22: BP 神经网络预测正确率图

结论 5.11

可以看到，模型预测精度在测试集上达到 68% 左右，表现基本合格。

同时，为了检验结果是否正确，我们还使用了 **GDBT 模型**进行检验，检验结果如下：

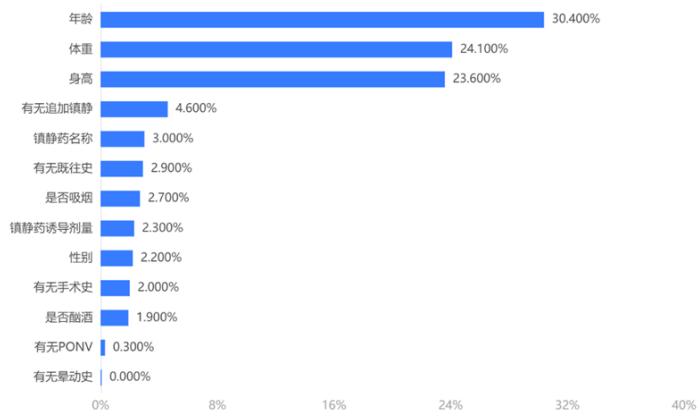


图 5.23: 检验结果图

	准确率	召回率	精确率	F1
训练集	0.999	0.999	0.999	0.999
交叉验证集	0.522	0.522	0.437	0.47
测试集	0.64	0.64	0.627	0.629

图 5.24: GDBT 模型预测正确率图

结论 5.12

由上图知，利用 GDBT 模型进行检验在训练集上精度高达 99%，而在测试集上表现基本合格，与 BP 神经网络预测结果基本一致。

IPI3 的预测思路与 IPI2 一致：

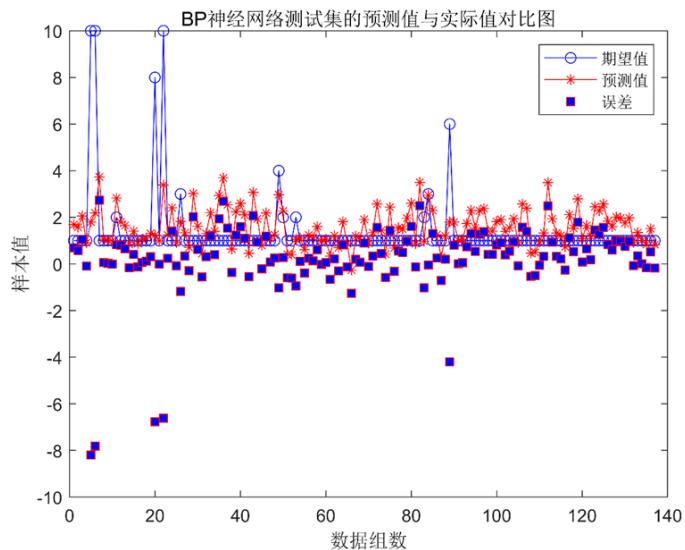


图 5.25: BP 神经网络测试集的预测值与实际值对比图

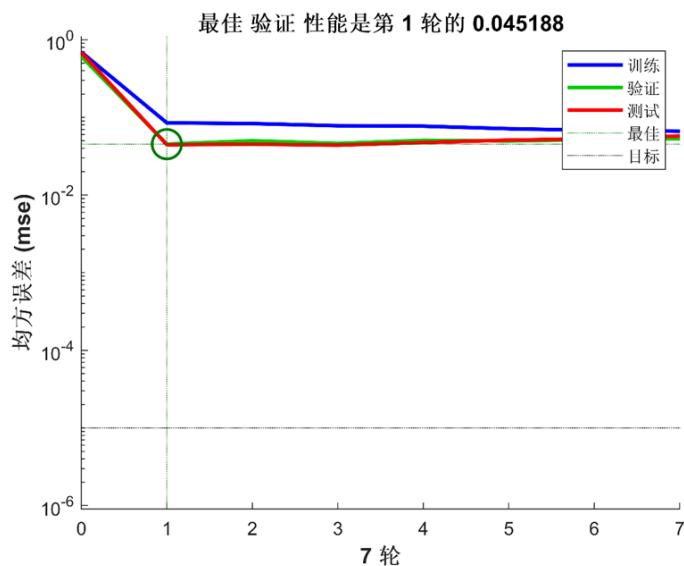


图 5.26: 性能图

```
-->-----误差计算-----<-->
蕴含层节点数为10时的误差结果如下:
平均绝对误差MAE为: 0.67248
均方误差MSE为: 0.76689
均方根误差RMSE为: 0.87572
>>
```

图 5.27: MAE、MSE、RMSE 结果图

预测结果 Y	IPI2
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	10

图 5.28: 预测结果图 (篇幅有限, 这里只截取了前十组数据)

结论 5.13

考虑到术后恢复水平所涉及到的因素比较复杂, 需要大量数据进行精确预测, 题目数据量较少, 所以能达到 68% 的准确度算是表现良好。

5.4 问题四的模型建立与求解

5.4.1 前置分析

对于术后满意度可能与哪些因素有关, 我们首先从内部机理分析的角度去进行探究。根据题目给出的:

4. 术后满意度与很多因素有关, 包括护理、身体恢复程度等等, 甚至有一些因素无法观测到。基于现有数据是否能够找出术后满意度与哪些因素有关? 有怎样的关系。

3) 病患的满意度及其相关问题

术后病患的满意度也需要作为药物评价和医生用药的重要参考。相关问题中还包括很多非常重要和待研究的问题, 如用药费用、用药剂量等。受限相关条件本次研究回避这类问题。

图 5.29: 题目背景分析

而护理方面题目没有给出相关数据。

另外, 文中说明不考虑用药费用和用药剂量的因素, 据此我们进行初步选取, 其他还可能与术后满意度相关的因素有:

1. 有无追加药物以及药物总剂量
2. IPI 最低值 (IPI 作为生命体征的核心指标, 其值能反映病患的恢复水平, 而身体恢复程度是影响病患满意度的因素之一)
3. 对医生的满意度 (对医生的满意度评分越低, 则术后满意度评分可能就越低)

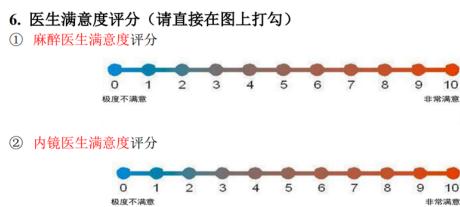


图 5.30: 医生满意度评分图

4. 术中、术后不良反应 (病人术中、术后表现出的不良反应很有可能影响到病人对术后满意度的评分)

5.4.2 数据预处理

由于题目给出的满意度数据不是 10 分制数据，而是“非常满意”这种定类数据。因此我们需要先把定类数据转为定量数据，我们使用区间均值 1 3 5 7 9 代表这五个评级，并将结果保存。

同理，对于医生满意度数据，我们应将他们转换为定量数据

麻醉医生满意度	内镜医生满意度
9分	8分
9分	9分
8分	7分
8分	8分
7分	7分
8分	7分
9分	9分
7分	8分
7分	8分
7分	7分
5分	6分
7分	8分
5分	6分

麻醉医生满意度	内镜医生满意度
9	8
9	9
8	7
8	8
7	7
8	7
9	9
7	8
7	8
7	7
5	6
7	8
5	6

图 5.31: 初始数据图 (左)、替换后数据图 (右)

同时，“是否恶心呕吐”等术中、术后不良反应也是定类数据，但在前面的分析中我们已将它们转为“0、1”定量数据，故在这里不需要对它们进行数据处理。

5.4.3 斯皮尔曼相关性分析模型的引入

皮尔逊相关系数用来度量两个变量间的相关程度，取值介于-1 与 1 之间。

两个变量间的皮尔逊相关系数定义为两个变量间的协方差和标准差的商：

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (5.15)$$

解释 5.7

上式定义了总体相关系数，常用希腊小写字母 ρ 作为代表符号。



估算样本的协方差和标准差，可以使用英文小写字母 r 表示：

$$r_{xy} = \frac{\sum x_i y_i - n \bar{x} \bar{y}}{(n-1)s_x s_y} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \quad (5.16)$$

解释 5.8

式中 (x_i, y_i) ($i=1, 2, \dots, n$) 表示变量 x 、 y 的 n 对研究值， \bar{x} 、 \bar{y} 为研究变量的平均值。可以使用 r_{xy} 研究两个变量之间的相关性质与相关程度。当 $r_{xy}=0$ 时， x 与 y 不相关；当 $0 < r_{xy} < 1$ 时， x 和 y 正相关；当 $-1 < r_{xy} < 0$ 时， x 和 y 负相关。



然而皮尔逊相关系数只适用于实验数据来自与正态分布的总体，适用范围有限。于是我们引出了斯皮尔曼相关系数。

斯皮尔曼相关分析的基本思想是：分别对两个变量 x 、 y 做秩变换，然后按皮尔逊相关分析的方法计算 RX

和 R_Y 的相关性。斯皮尔曼相关系数具体计算公式如下：

$$r_S = \frac{\sum (R_X - R'_X)(R_Y - R'_Y)}{\sqrt{\sum (R_X - R'_X)^2 \sum (R_Y - R'_Y)^2}} = \frac{\sum R_X R_Y - \frac{(\sum R_X)(\sum R_Y)}{n}}{\sqrt{\left(\sum R_X^2 - \frac{(\sum R_X)^2}{n}\right) \left(\sum R_Y^2 - \frac{(\sum R_Y)^2}{n}\right)}} \quad (5.17)$$

分析步骤

- 1、先检验 XY 是否存在统计上的显著关系 ($P < 0.05$)
- 2、分析相关系数的正负以及相关程度
- 3、对分析结果进行总结

5.4.4 求解结果分析

满意度实际评分	0.048(0.099**)	0.195(0.000*)	0.012(0.671)	0.008(0.788)	0.073(0.012**)	0.041(0.157)	0.108(0.000*)	0.066(0.023**)	0.035(0.221)	0.206(0.000*)	0.126(0.000*)	0.08(0.006**)	0.097(0.001**)	1(0.000***)	0.16(0.000**)		
有无追加镇静药总剂量		镇静药总剂量		有无追加镇痛		IPI最低值		麻醉医生满意度		内镜医生满意度		呛咳		体动		术中其他	
是否出现了恶心呕吐的情况是																	
是否出现了头晕头痛的情况是																	
有没出现嗜睡乏力的情况呢有																	
有没有出现腹胀腹痛的情况呢有																	
满意度实际评分																	
镇痛药总剂量																	

图 5.32: 满意度实际评分图

解释 5.9

黄色网格中括号外为相关系数数值，括号内为显著性水平 (P 值)



如图，我们将术后满意度转换成了满意度实际评分。观察上图可以发现，满意度实际评分与药物总剂量、术中术后不良反应的 P 值呈现显著性，说明术后满意度与这些因素相关。但是相关系数均较小（小于 0.3），说明术后满意度与这些因素呈现弱相关性。于是我们尝试更改影响因素，去掉相关系数较小的变量，并加入新的变量进行检验。

首先去掉“有无追加 xx”、“IPI 最低值”、“医生满意度”、“体动”、“术中其他”、“嗜睡乏力”以及“腹胀腹痛”这几个相关系数小于 0.1 的变量，然后加入与时间有关的的变量，如：进镜时间、出镜时间、睁眼时间等变量，最后得到检验结果如下图：

满意度实际评分	0.249(0.000*)	0.17(0.001**)	0.231(0.000*)	0.091(0.082*)	0.221(0.000*)	0.283(0.000*)	0.276(0.000*)	0.295(0.000*)	0.291(0.000*)	0.315(0.000*)	0.287(0.000*)	0.277(0.000*)	1(0.000***)		
镇静药总剂量		呛咳		是否出现了恶心呕吐的情况是		是否出现了头晕头痛的情况是		镇痛药总剂量		开始给药时间		最后一次给药时间		IPI 达到 4 分时间	
睡眠时间															
进镜时间															
出镜时间															
出 PACU															
满意度实际评分															

图 5.33: 最终检验结果图

结论 5.14

可以发现，修改变量后相关系数明显提高，其中进镜时间已超过 0.3，说明术后满意度和进镜时间低度相关；另外，“开始给药时间”、“最后一次给药时间”、“IPI 达到 4 分的时间”以及“睁眼时间”的相关系数接近 0.3，可以认为术后满意度与它们低度相关。



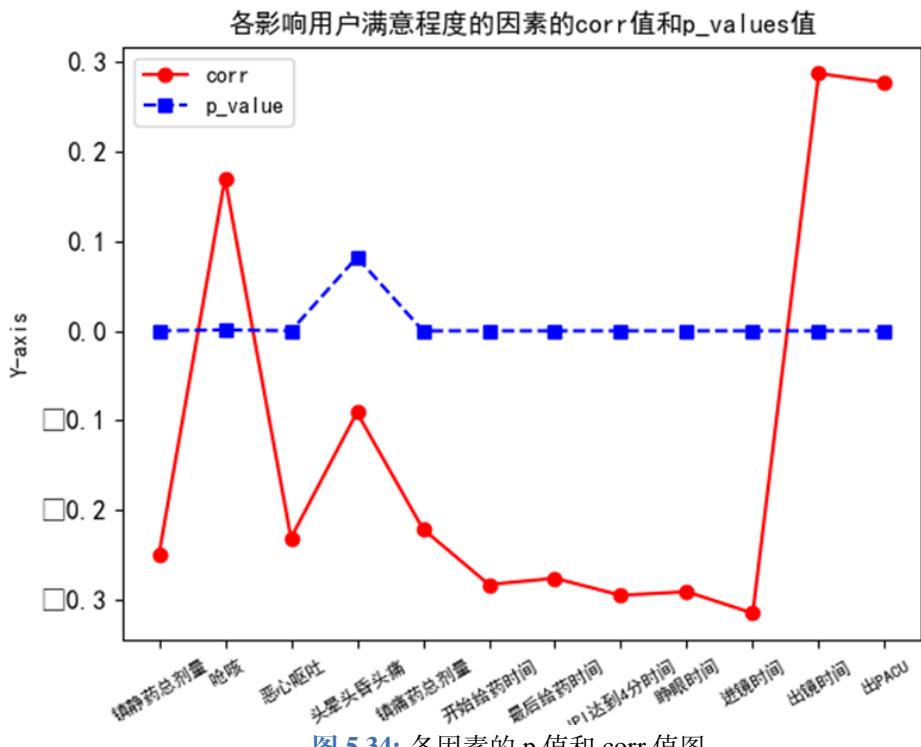


图 5.34: 各因素的 p 值和 corr 值图

模型评估: 本模型探究了影响术后满意度的因素，其中主要是时间变量对满意度影响最大。本模型的优点是能直观分析相关程度。

第 6 章 模型分析

对于问题一模型的合理性进行分析：

通过下面的一种反应的 ROC 图的分析我们能够得到：预测正确的有 332，样本总量是 374，预测的准确率是在 86.45% 左右，预测的效果较好。体现出了预测模型的合理性。

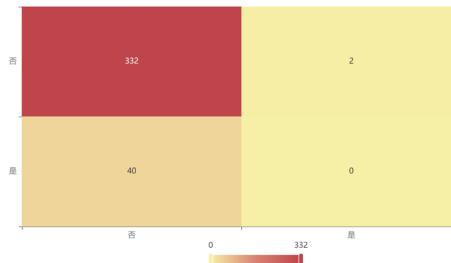


图 6.1: ROC 图

对于问题二模型的合理性进行分析：

在问题二中，我们使用了多元回归模型进行求解。在进行多元回归分析时，属性之间最好具有一定的相关性。通常情况下，属性之间的相关系数不应太大（一般要小于 0.7），但也不应过小（一般要大于 0.3）。此外，多元回归分析还要注意避免多重共线性问题，即属性之间存在高度相关性的情况，这会影响回归系数的精确度和可解释性。因此，在进行多元回归分析之前，先对属性的相关性进行检验。

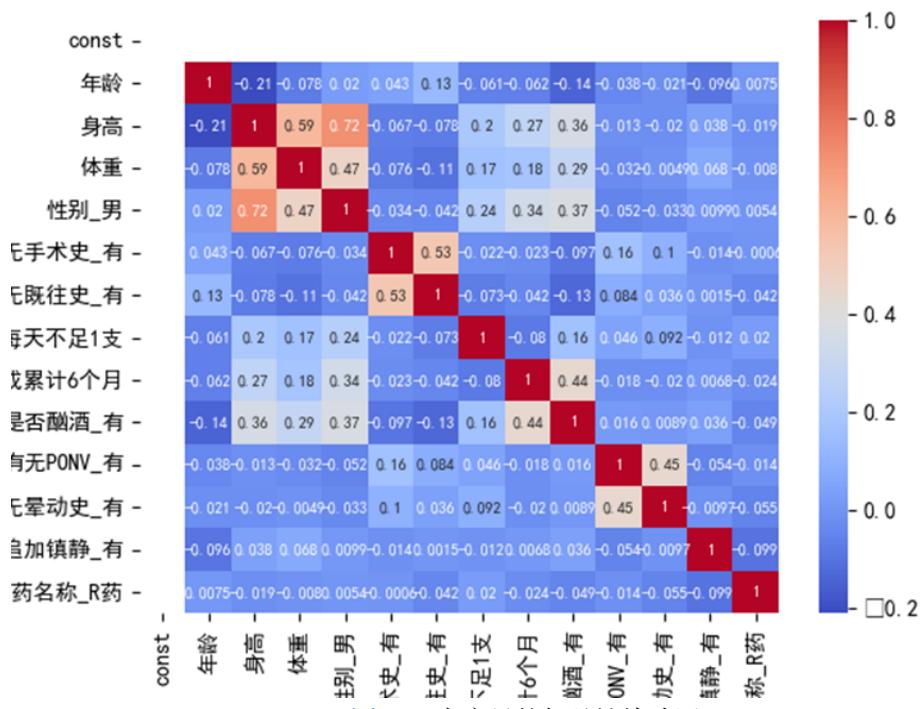


图 6.2: 自变量的相关性检验图

通过对图中的数据分析可得，属性相关性中，性别与身高相关性较强，故我们在分析的时候，只选取其中的一个作为分析的属性。因此，我们的样本数据是满足多元回归模型的要求的。因此，模型的选择具有一定的合理性。

第 7 章 模型总结与评价

7.1 问题一模型的评价与改进

问题一第一部分我们一开始用的卡方检验，发现检验结果不尽人意，于是我们将术中和术后不良反应转换成定量数据，进行双样本 T 检验。从结果来看，双样本 T 检验效果明显好于卡方检验，借助 Python 的特性我们还将结果可视化。

问题一第二部分我们通过不断调整传入的数据组数，建立的二元逻辑回归模型的预测精度越来越高。另外我们还使用 ROC 图检验方法检验该模型是否高效、完美。经过增加训练次数等一系列改进措施，我们训练出了比较完美的模型。

改进方案 7.1

双样本 T 检验对数据量的需求比较大，可以通过增加有效数据等操作进一步改进。对于二元逻辑回归模型，可以通过增加训练次数、调整参数等方法进行改进。



7.2 问题二模型的评价与改进

对于问题二数据处理，我们首先依据材料中给的 IPI 值是核心部分，所以在求解中，我们希望能够凸显 IPI 的同时，将其他的属性也能够加上。因此，我们采用了主成分分析的方法，在进行了数据归一化处理之后，为 IPI 值赋予了更高的权重，但是最终结果并不尽如人意，在我们首次赋权中，我们将 IPI 值与其他值赋予 5:1 的比重，计算出的相关性达到了 0.83，之后，进行了 10:1 的权重赋值，相关性为 0.9，因此，我们猜测是由于属性值过多，导致主成分分析时，IPI 值很难体现核心地位。因此，我们采用了直接赋权的方式，放弃了主成分分析方法。从结果来看，直接赋权比主成分分析法的效果要好很多，保留了 IPI 的影响。

在进行数据的处理后，我们首先用了典型相关分析进行两个样本的比较，得到的结论是二者高度的负相关。最后，我们意识到典型相关分析更适用于相关性的分析求解中，而不是差异性。因此，我们从不同的角度进行了差异性的判断，我们先从中位数方面入手判断，采用 Wilcoxon 秩和检验判断差异性，得到的结论是有显著差异；之后我们又从方差的角度进行方差分析判断，得到的结果依旧是具有显著差异。据此，判断两样本具有显著差异。

最后寻找对生命特征值造成显著差异的原因时：我们采用了多元回归分析和 Filter 分析方法，通过二者的结论的比较，我们可以看出，得到的结论大致相同。

改进方案 7.2

有关多元回归模型的改进：我们可以通过残差分析，残差分析是一种用于检验回归模型是否符合统计假设的方法。它通过分析模型的残差（即预测值与真实值之间的差异），来检验回归模型是否存在异方差、线性关系偏离等问题。如果残差分析发现回归模型存在问题，就需要对模型进行修正或改进。



7.3 问题三模型的评价与改进

本文采用 BP 神经网络模型对结果进行分类预测，算法简单容易操作，同时合理处理数据对算法进行进一步简化。考虑到术后恢复水平所涉及到的因素比较复杂，需要大量数据进行精确预测，题目数据量较少，所以能达到 68% 的准确度算是表现良好。

改进方案 7.3

BP 神经网络模型参数的设置以及隐藏层的数量对结果影响较大，所以应当不断调试，寻找参数与隐藏层的最优组合以达到最好的训练效果。

除此之外，sigmoid 激活函数的选择也会对预测结果产生较大影响，在设计模型时也应当将其考虑在内。



7.4 问题四模型的评价与改进

本文采用斯皮尔曼相关分析模型对可能的因素进行相关性分析，考虑到可能的影响因素有多个，我们首先选取一部分进行分析，并留下其中相关且相关系数较大的变量，然后分析余下部分。我们最终分析出时间因素与术后满意度相关，但相关度不高，但考虑到影响术后满意度的因素较多，甚至有些因素题目尚未给出，所以我们认为这个结果能够接受。

改进方案 7.4

增大数据量、考虑更多变量可以作为该模型的改进方案。



第8章 附录(程序所使用的所有代码展示)

8.1 问题一代码

8.1.1 第一部分的 Python 代码

```
import numpy as np
import pandas as pd
from scipy.stats import chi2_contingency
from scipy.stats import ttest_ind
import matplotlib.pyplot as plt
# 加载数据
df = pd.read_excel(io="for2.xls",usecols="EW,EX,EY,FA,FG,FM,FR,FW,Q")

groups = df.groupby('镇静药名称')
for name, group in groups:
    if(name == "B药"):
        df1 = group
    else:
        df2 = group
df1=df1.drop(columns='镇静药名称',axis = 1)
df2=df2.drop(columns='镇静药名称',axis = 1)
# 将 DataFrame 转换为 NumPy 数组
arr1 = df1.to_numpy()
arr2 = df2.to_numpy()

# 将两个数组组合成一个二维数组
arr = np.vstack((arr1, arr2))
...
# 进行卡方检验
chi2, pval, dof, expected = chi2_contingency(arr)
print('P value is ' + str(pval))
...
# 进行双样本t检验
t, p = ttest_ind(arr1, arr2)
print("t值为:", t)
print("P值为:", p)
x_labels = ['术中呛咳', '术中体动', '术中其他', '术后恶心', '术后头痛', '术后嗜睡', '术后腹胀', '术后其它']

plt.rcParams['font.sans-serif'] = ['SimHei'] # 设置中文字体
plt.rcParams['font.family'] = 'sans-serif' # 设置全局字体

# 绘制条形图
fig, ax = plt.subplots()
ax.bar(x_labels, p)
```

```
# 设置标题和轴标签
ax.set_title('不同反应对应的P值')
ax.set_xlabel('Columns')
ax.set_ylabel('Values')

# 显示图像
plt.show()
```

8.1.2 第二部分的 Python 代码

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegressionCV
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score

df = pd.read_excel('2.xlsx')
# print(df.loc[197])
# 去掉含有缺失值的行
df = df.drop(197)

# X是协变量，Y是因变量
X = df[['No', 'Sex', 'Age', 'Height', 'Weight', 'Pill']]
Y = df['Sleepy']
# print(Y.head())

# 划分训练集、测试集，训练集：测试集 = 8:2
X_train,X_test,Y_train,Y_test = train_test_split(X, Y, test_size=0.2)
# print(X_train.head())
model = LogisticRegressionCV(multi_class='multinomial', max_iter=3000).fit(X_train,Y_train)

# 预测结果
Y_pred = model.predict(X_test)
arr1 = pd.DataFrame()
arr1['prediction'] = list(Y_pred)
arr1['faction'] = list(Y_test)
print(arr1.head())

# 预测的准确度
score = accuracy_score(Y_pred, Y_test)
print("accuracy:", score)

# 预测概率
y_pred_proba = model.predict_proba(X_test)
arr2 = pd.DataFrame(y_pred_proba, columns=['NoLossProba', 'LossProba'])
```

```

print(arr2.head())
print("\n")

# 混淆矩阵
m = confusion_matrix(Y_test, Y_pred)
arr3 = pd.DataFrame(m, index=['0FactNoLoss', '1FactLoss'], columns=['0PredNoLoss', '1PredLoss'])
print(arr3)
print("\n")

# ROC曲线
fpr, tpr, thres = roc_curve(Y_test, y_pred_proba[:,1])
arr4 = pd.DataFrame()
arr4['threshold'] = list(thres)
arr4['FPR'] = list(fpr)
arr4['TPR'] = list(tpr)
print(arr4.head())
print("\n")
plt.plot(fpr, tpr)
plt.title('ROC')
plt.xlabel('FPR')
plt.ylabel('TPR')
# plt.show()

```

8.2 问题二代码

8.2.1 第一部分的 Python 代码

```

#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；如果有显著差异，能
#否确定是由于新药造成，还是由其他因素造成
#思路
#先用主成分分析法对生命体征的判别条件等进行数据降维
#然后用典型相关分析进行判别是否有显著差异

from sklearn.decomposition import PCA
from sklearn.preprocessing import MinMaxScaler
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cross_decomposition import CCA
import scipy.stats as stats
from scipy.stats import wilcoxon
from scipy.stats import shapiro

plt.rcParams['font.sans-serif'] = ['SimHei'] # 设置中文字体
plt.rcParams['font.family'] = 'sans-serif' # 设置全局字体
# 加载数据
df = pd.read_excel(io="附件1.xls",usecols="AE:AL,AO:BJ,BM:BZ,CC:DN,EE:EL,Q")

```

```

#df = pd.read_excel(io="附件1.xls",usecols="CQ:CW,Q")
groups = df.groupby('镇静药名称')
for name, group in groups:
    if(name == "B药"):
        df1 = group
    else:
        df2 = group

# 对属性进行归一化处理
scaler = MinMaxScaler()

df1=df1.drop(columns='镇静药名称',axis = 1)
# 用均值填充空值
df1 = df1.fillna(df1.mean())
print(df1.isnull().any())
normalized_df1 = pd.DataFrame(scaler.fit_transform(df1), columns=df1.columns)
print(normalized_df1)

normalized_df1["IPI00"] = 5*normalized_df1[['IPI00']]
normalized_df1["IPI005"] = 5*normalized_df1[['IPI005']]
normalized_df1["IPI1"] = 5*normalized_df1[['IPI1']]
normalized_df1["IPIjinjing"] = 5*normalized_df1[['IPIjinjing']]
normalized_df1["IPI015"] = 5*normalized_df1[['IPI015']]
normalized_df1["IPI2"] = 5*normalized_df1[['IPI2']]
normalized_df1["IPI025"] = 5*normalized_df1[['IPI025']]
normalized_df1["IPI3"] = 5*normalized_df1[['IPI3']]

normalized_df1["IPI5"] = 5*normalized_df1[['IPI5']]
normalized_df1["IPI7"] = 5*normalized_df1[['IPI7']]
normalized_df1["IPI10"] = 5*normalized_df1[['IPI10']]

normalized_df1["IPIjieshu"] = 5*normalized_df1[['IPIjieshu']]

num_cols1 = normalized_df1.shape[1]
#print(num_cols)
result1 = np.sum(normalized_df1*(float(1)/num_cols1),axis = 1)
print(result1)

df2=df2.drop(columns='镇静药名称',axis=1)
# 用均值填充空值
df2 = df2.fillna(df2.mean())
print(df2.isnull().any())
normalized_df2 = pd.DataFrame(scaler.fit_transform(df2), columns=df2.columns)

```

```

normalized_df2["IPI00"] = 5*normalized_df2[['IPI00']]
normalized_df2["IPI005"] = 5*normalized_df2[['IPI005']]
normalized_df2["IPI1"] = 5*normalized_df2[['IPI1']]
normalized_df2["IPIjinjing"] = 5*normalized_df2[['IPIjinjing']]
normalized_df2["IPI015"] = 5*normalized_df2[['IPI015']]
normalized_df2["IPI2"] = 5*normalized_df2[['IPI2']]
normalized_df2["IPI025"] = 5*normalized_df2[['IPI025']]
normalized_df2["IPI3"] = 5*normalized_df2[['IPI3']]

normalized_df2["IPI5"] = 5*normalized_df2[['IPI5']]

normalized_df2["IPI7"] = 5*normalized_df2[['IPI7']]
normalized_df2["IPI10"] = 5*normalized_df2[['IPI10']]

normalized_df2["IPIjieshu"] = 5*normalized_df2[['IPIjieshu']]

num_cols2 = normalized_df2.shape[1]
#print(num_cols)
result2 = np.sum(normalized_df2*(float(1)/num_cols2),axis = 1)
score = result2[1:476]

# 绘制散点图
plt.scatter(score, result1)

# 设置图表标题和轴标签
plt.title('B组药和R组药的生命体征值关系')
plt.xlabel('R药')
plt.ylabel('B药')

# 显示图表
plt.show()
'''

# 计算Pearson相关系数
corr = score.corr(result1)
print(corr)
'''

# 进行Wilcoxon秩和检验
stat, p = wilcoxon(score,result1)
# 输出检验结果
print('stat=%f, p=%f' % (stat, p))
if p > 0.05:
    print('两个样本数据没有显著差异')
else:
    print('两个样本数据有显著差异')
# 进行正态性检验
stat, p = shapiro(score)

# 绘制概率密度图

```

```

sns.distplot(score, kde=False)
plt.show()
# 判断p值是否小于0.05, 若小于0.05则认为不符合正态分布
if p < 0.05:
    print("不符合正态分布")
else:
    print("符合正态分布")
# 进行ANOVA分析,方差分析
f_value, p_value = stats.f_oneway(score, result1)
# 输出结果
print("F-value:", f_value)
print("p-value:", p_value)

```

8.2.2 第二部分的 Python 代码

```

import pandas as pd
import statsmodels.api as sm
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from statsmodels.graphics.regressionplots import plot_partregress
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_regression

# 加载数据
df = pd.read_excel('附件1.xls')

# 将自变量和因变量分离出来
X = df[['年龄', '性别', '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', '有无追加镇静', '镇静药名称']]
y = pd.read_excel(io="附件1.xls",usecols="AE:AL, AO:BJ, BM:BZ, CC:DN, EE:EL")

# 对属性进行归一化处理
scaler = MinMaxScaler()
# 用均值填充空值
y = y.fillna(y.mean())
print(y.isnull().any())
normalized_y = pd.DataFrame(scaler.fit_transform(y), columns=y.columns)
#print(normalized_y)
normalized_y["IPI00"] = 5*normalized_y[['IPI00']]
normalized_y["IPI005"] = 5*normalized_y[['IPI005']]
normalized_y["IPI1"] = 5*normalized_y[['IPI1']]
normalized_y["IPIjinjing"] = 5*normalized_y[['IPIjinjing']]
normalized_y["IPI015"] = 5*normalized_y[['IPI015']]
normalized_y["IPI2"] = 5*normalized_y[['IPI2']]
normalized_y["IPI025"] = 5*normalized_y[['IPI025']]
normalized_y["IPI3"] = 5*normalized_y[['IPI3']]

```

```

normalized_y["IPI5"] = 5*normalized_y[['IPI5']]
normalized_y["IPI7"] = 5*normalized_y[['IPI7']]
normalized_y["IPI10"] = 5*normalized_y[['IPI10']]

normalized_y["IPIjieshu"] = 5*normalized_y[['IPIjieshu']]
#print(normalized_y)
num_cols = normalized_y.shape[1]
#print(num_cols)
result = np.sum(normalized_y*(float(1)/num_cols),axis = 1)
#print(result)

weights = { 'sbp00':'','dbp00':'','petco200':'','RR00':'','spo200':'','HR00':'','IPI00' moaas00
            sbp005 dbp005 petco2005 RR005 spo2005 HR005 IPI005 moaas005 sbp1 dbp1 petco21 RR1 spo21 HR1 IPI1
            moaas1 sbpjining dbpjining petco2jining RRjining spo2jining HRjining IPIjining moaasjining
            sbp015 dbp015 petco2015 RR015 spo2015 HR015 IPI015 moaas015 sbp2 dbp2 petco22 RR2 spo22 HR2 IPI2
            moaas2 sbp025 dbp025 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 sbp10 dbp10 petco210 RR10 spo210 HR10 IPI10 moaas10 sbp15 dbp15 petco215 RR15 spo215 HR15
            IPI15 moaas15 sbp20 dbp20 petco220 RR20 spo220 HR20 IPI20 moaas20 sbpjieshu dbpjieshu petco2jieshu
            RRjieshu spo2jieshu HRjieshu IPIjieshu moaasjieshu
        }

# 将分类变量进行虚拟变量转换
X = pd.get_dummies(X, columns=['性别', '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', '有无追加镇静', '镇静药名称'], drop_first=True)
X = X.fillna(X.mean())
#print(X.isnull().any())
# 添加截距项
X = sm.add_constant(X)

# 计算相关系数矩阵
corr_matrix = X.corr()

plt.rcParams['font.sans-serif'] = ['SimHei'] # 设置中文字体
plt.rcParams['font.family'] = 'sans-serif' # 设置全局字体

# 可视化相关矩阵
#sns.set(font_scale=0.4) # 设置字体大小
#sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', annot_kws={"size": 7})
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.xticks(fontsize=8, rotation=30)
plt.yticks(fontsize=12)
#plt.title('PCA Loadings Plot')
plt.show()
...
# 进行多元回归分析
model = sm.OLS(result, X).fit()

# 输出回归系数和显著性检验结果

```

```

print(model.summary())

```
#filter
特征选择
best_features = SelectKBest(score_func=f_regression, k=3)
fit = best_features.fit(X, result)
输出得分和选择的特征
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
feature_scores = pd.concat([dfcolumns, dfscores], axis=1)
feature_scores.columns = ['Feature', 'Score']
print(feature_scores.nlargest(3, 'Score'))
print(feature_scores['Feature'])
feature_scores.loc[:, 'Feature'] = ['const', '年龄', '身高', '体重', '性别', '有手术史', '有既往史', '偶尔吸烟',
 '经常吸烟', '酗酒', '有PONV', '有晕动史', '追加镇静', '镇静药种类']

绘制条形图
plt.bar(feature_scores['Feature'], feature_scores['Score'])
plt.title('Feature Scores')
plt.xlabel('Feature Names')
plt.ylabel('Score')
plt.xticks(fontsize=8, rotation=90)
plt.yticks(fontsize=12)
plt.show()

```

## 8.3 问题三的 Matlab 代码

```

clear
clc;
load Data1.mat

%% 读取数据
input = S1(:, 2:14);
output = S1(:, 15);

%% 设置训练数据和测试数据
% 注意将指标变为列向量

input_train = input(1:1000, :)';
output_train = output(1:1000, :)';
input_test = input(1001:1137, :)';
output_test = output(1001:1137, :)';

% 节点个数
% 输入层节点个数
inputnum = 7;
% 输出层节点个数

```

```

outputnum = 8;
% 隐藏层节点个数
hiddennum = 10;

%% 训练样本数据归一化
[inputn, inputps] = mapminmax(input_train);
[outputn, outputps] = mapminmax(output_train);

%% 构建BP神经网络
net = newff(inputn, outputn, hiddennum,{'tansig', 'purelin'},'trainlm');
% 输入层到中间层的权值
W1 = net.iw{1,1};
% 中间各层神经元阈值
B1 = net.b{1};
% 中间层到输出层的权值
W2 = net.lw{2,1};
% 输出层各神经元阈值
B2 = net.b{2};

%% 网络参数配置
net.trainParam.epochs = 1000;
net.trainParam.lr = 0.01;
net.trainParam.goal = 0.00001;

%% 神经网络训练
net = train(net, inputn, outputn);

%% 测试样本归一化
inputn_test = mapminmax('apply', input_test, inputps);

%% BP神经网络预测
an = sim(net, inputn_test);

%% 预测结果反归一化与误差计算
test_simu = mapminmax('reverse', an, outputps);
error = test_simu-output_test;

%% 真实值与预测值误差比较
figure('units', 'normalized', 'Position', [0.119 0.2 0.38 0.5])
plot(output_test, 'bo-')
hold on
plot(test_simu, 'r*-')
hold on
plot(error, 'sque', 'MarkerFaceColor','b')
legend('期望值', '预测值', '误差')
xlabel('数据组数')
ylabel('样本值')
title('BP神经网络测试集的预测值与实际值对比图')

```

```
[c,l] = size(output_test);
MAE1 = sum(abs(error))/l;
MSE1 = error*error'/l;
RMSE1 = MSE1^(1/2);

disp(['-----误差计算-----'])
disp(['隐含层节点数为',num2str(hiddennum),'时的误差结果如下：'])
disp(['平均绝对误差MAE为：',num2str(MAE1)])
disp(['均方误差MSE为：',num2str(MSE1)])
disp(['均方根误差RMSE为：',num2str(RMSE1)])
```

## 8.4 问题四的 Python 代码

```
from scipy.stats import spearmanr
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
读取Excel表格
df = pd.read_excel('第四问满意度实际评分数据.xlsx',usecols="Q,W,Y,AB,CU,EU,EV,EW,EX,EY,EZ,FA,FB,FC,ES,
ET")

X = pd.get_dummies(df, columns=['有无追加镇静',''呛咳','有无追加镇痛','体动','术中其他','是否出现了恶心呕
吐的情况是','是否出现了头晕头昏头痛是','有没出现嗜睡乏力的情况呢有','有没出现腹胀腹痛的情况呢有','还
有没其他不舒服的情况呢有'], drop_first=True)
X = X.fillna(X.mean())
X=X.drop(columns='满意度实际评分',axis = 1)
X['满意度实际评分'] = df['满意度实际评分']
print(X)
plt.rcParams['font.sans-serif'] = ['SimHei'] # 设置中文字体
plt.rcParams['font.family'] = 'sans-serif' # 设置全局字体
print(X)
计算每个特征与目标变量之间的Spearman相关系数和p-value
corrs = []
p_values = []
创建一个2行3列的子图，并设置子图大小为(10, 6)
fig, axes = plt.subplots(2, 2, figsize=(10, 6))
i = 0
j = 0

for col in X.columns:
 if col != '满意度实际评分':

 corr, p_value = spearmanr(X[col], X['满意度实际评分'])
 corrs.append(corr)
 #print(corr)
 p_values.append(p_value)
 #print(p_value)
```

```

if(col == '麻醉医生满意度' or col == '是否出现了头晕头昏头痛是_是' or col == '是否出现了恶心呕吐的情况是_是' or col == '有无追加镇痛_有'):
 # 绘制箱线图
 print(col)
 print(X['满意度实际评分'])

 a = pd.DataFrame({col: X[col], '满意度实际评分': X['满意度实际评分']})
 axes[i,j].boxplot(data = a, x = col)
 axes[i,j].set_title(col+'Box plot')
 if(j == 1):
 i = 1
 j = -1
 j+=1

调整子图之间的间距和周围的边距
fig.tight_layout(pad=2, w_pad=1, h_pad=1)
plt.show()

将结果存储在一个DataFrame中
results = pd.DataFrame({'Feature': X.columns[:-1], 'Correlation': corrs, 'P-value': p_values})

x_values = ['镇静药总剂量', '呛咳', '恶心呕吐', '头晕头昏头痛', '镇痛药总剂量', '开始给药时间', '最后给药时间',
 'IPI达到4分时间', '睁眼时间', '进镜时间', '出镜时间', '出PACU']
y1 = [-0.249, 0.17, -0.231, -0.091, -0.221, -0.283, -0.276, -0.295, -0.291, -0.315, 0.287, 0.277]
y2 = [0.000, 0.001, 0.000, 0.082, 0.000, 0.00, 0.000, 0.00, 0.000, 0.000, 0.00, 0.00]

绘制折线图
plt.plot(x_values, y1, color='red', linestyle='-', label='corr', marker='o')
plt.plot(x_values, y2, color='blue', linestyle='--', label='p_value', marker='s')

添加标签和标题
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('各影响用户满意程度的因素的corr值和p_values值')
plt.xticks(fontsize=8, rotation=30)
plt.yticks(fontsize=12)

添加图例
plt.legend()

显示图表
plt.show()

```