


中国研究生创新实践系列大赛

“华为杯”第二十届中国研究生

数学建模竞赛

学　校	北京石油化工学院
参赛队号	23100170006
队员姓名	1.邢晓龙
	2.王利猛
	3.张婧

中国研究生创新实践系列大赛

“华为杯”第二十届中国研究生

数学建模竞赛

题　目：　基于血性脑卒中临床智能诊疗的建模和优化

摘　要：

出血性脑卒中起病急、进展快，预后较差，病死率高，并且可能会给患者遗留较严重的神经功能障碍，为社会及患者家庭带来沉重的健康和经济负担。研究血肿扩张和血肿周围水肿的发生及发展是针对出血性脑卒中后的两个重要关键事件，所以进行早期识别和预测对于改善患者预后、提升其生活质量具有重要意义。为此我们建立模型，对问题进行分析和研究。本文所做的工作可概括以下几点：

问题一：

问题 a，根据题目要求提取 ″表 1″ ，″表 2″，″附表 1″ 的数据。使用″表 2″中的流水号至“附表 1”中查询相应影像检查时间点，结合发病到首次影像时间间隔和后续影像检查时间间隔，可以获得每个病人从发病到各个随访时的**发病小时数(h)**。通过对比患者首次检查的 血肿体积**对比**后续随访**血肿体积的变化，判断血肿扩张是否发生**。如果判断发生血肿扩张，则根据该患者首次检查到该次随访的时间段**建立以发病小时数(h)为自变量,以对应的血肿体积(HM_v)为因变量的线性方程**。然后将血肿扩张发生条件条件（比首次检查绝对体积增加≥6 mL 或相对体积增加≥33%）代入到方程中求解，可以**计算出发生血肿条件具体时间。结果见附件“表 4”C，D 字段。**

问题 b，首先进行数据处理，离散变量进行编码，连续变量进行归一化，然后采用**随机森林（RF）特征重要度、极度梯度提升树（XGBoost）**和**斯皮尔曼相关系数（Spearman ）**进行**特征筛选**，排序投票选出 3 个候选集，对候选集取交集，最终确定高血压病史、糖尿病史、冠心病史、止血治疗等 38 个变量作为发生血肿扩张的自变量。本文构建**决策树、随机森林分类、Adaboost、梯度提升树(GBDT)、ExtraTree** 等十二类算法，以 5 折交叉验证的方式采用**准确率、召回率、精确率、F1** 等多个指标评估， 发现 **ExtraTree 分类模型与 adaboost 分类模型效果最好，使用遗传算法对其参数进行了调优。**调优后进行**模型融合**，根据所有含随访影像检查的患者（sub001 至 sub160）进行分类预测。本文使用组合预测模型预测血肿扩张预测概率，**预测结果见附件“表 4”E 字段。**

问题二：

问题 a 研究发病到首次影像检查时间间隔与水肿体积的关系。构建并训练**基于鲸鱼优化算法**(WOA)**的 ARIMA 时序模型**，用 WOA 算法优化 ARIMA 起始超参数 PDQ 并用 **5 折交叉验证**计算 **RMSE**，评估模型质量，最后计算全体残差。见附件“表 4”F 字段。

问题 b 选择 '发病到首次影像检查时间间隔' 和 '水肿体积 ED_volume' 两列作为聚类特征对患者聚类分组。根据肘部法确定合适的组数为 3。使用 **KMeans, MiniBatchKMeans, SpectralClustering, AgglomerativeClustering, Birch** 共 5 种聚类算法训练聚类模型，而后用**轮廓系数，Calinski-Harabaz Index 和 Davies-Bouldin Index** 作为评估标准，选择 AgglomerativeClustering 聚类算法的结果作为亚组依据。见附件“表 4”H 字段。对每一类亚组都分别使用 1 至 5 次多项式、指数和对数拟合。根据 R^2 系数选择最佳拟合方式即 **5 次多项式**拟合并计算亚组残差。见附件“表 4”G 字段。

问题 c **定义水肿体积进展模式特征为相邻随访记录单位时间内水肿体积变化量**，计算治疗方法与其的**斯皮尔曼相关性**。可以看到，治疗方式的各项特征与进展模式既存在负相关又存在正相关。但是正相关性不大；负相关性最大能达到-0.25。说明治疗有效抑制了水肿体积增加，减缓了体积增长速度。

问题 d 与 c 相似，**计算治疗方式与多次随访的血肿体积和水肿体积的相关性**。水肿和血肿的相关性在(0.5,1)区间内，二者具有很强的正相关性。随着随访次数增加，治疗方式的特征与血肿和水肿由正相关转成负相关，说明治疗效果正在变好，病变区域得到了有效控制。

问题三：

问题 a，根据题目要求提取 ″表 1″字段 E 至 W ，″表 2″和″表 3″ 的首次影像结果数据。对数据进行归一化和编码处理。然后对数据进行数据筛选最终留下 38 个变量。使用筛选后的变量作为因变量，以 90 天 mRs 自变量进行建模。因为 90 天 mRs 为离散等级，因此使用**十二个分类预测模型**进行训练，选出效果优秀的 **LightGBM 分类模型**和**随机森林分类模型**，并使用**粒子群算法**对模型进行**参数调优**。调优后进行两个**模型融合**，使用两个模型的平均值作为输出。最后的预测**结果见附件“表 4”I 字段。**

问题 b 与第一小问类似，将后续随访数据加入到第一小问的数据中。对新加入的数据进行归一化处理以及特征筛选。最后保留下变量。同第一小问，使用新的数据训练**十二个分类预测模型**。选出效果优秀的 **LightGBM 分类模型**和 **BP 神经网络分类模型**。然后使用**模拟退火算法**进行参数调优。调优后进行两个**模型的融合**。最后预测结果**见附件“表 4”J 字段。**

问题 c 对发病特征(血压值)用 **PCA** 降为 2 维；治疗方式相关特征用 PCA 降为 5 维；体积和位置特征用 PCA 降为 5 维；形状及灰度特征用 PCA 降为 2 维。计算可得 PCA 降维后新特征的**贡献率**皆大于 90%。计算 14 维特征与 90 天 mRS 的斯皮尔曼相关性系数矩阵。可见发病特征，形状及灰度和体积及位置与 90 天 mRS 呈现正相关；治疗特征与 90 天 mRS 呈现负相关。说明患者经过治疗后，血肿和水肿减少，90 天 mRS 会降低。若未经过治疗，且伴有高血压，就会很容易扩大血肿和水肿致使病情严重，90 天 mRS 会逐渐升高，危及生命。所以**建议病情早发现早治疗，珍爱生命。**

针对以上问题设计了各种模型的最优参数组合求解算法。

关键词：极度梯度提升树、LightGBM 分类模型、ARIMA 时序模型、模拟退火算法、粒子群算法、鲸鱼优化算法

目录

一、问题重述 5
1.1 问题背景 5
二、问题分析 6
2.1 问题一 6
2.2 问题二 7
2.2.1 问题 a 7
2.2.2 问题 b 8
2.2.3 问题 c 9
2.3 问题三 10
三、模型假设 11
3.1 问题一假设 11
3.2 问题二假设 11
3.3 问题三假设 12
四、符号说明 12
五、问题一模型的建立与求解 13
5.1 问题 a 模型建立和求解 13
5.1.1 数据提取 13
5.1.2 模型建立与求解 13
5.2 问题 b 模型建立和求解 14
5.2.1 数据处理 14
5.2.2 特征筛选 14
5.2.3 对比分析与模型验证 16
5.2.4 参数调优及模型融合求解 17
5.2.5 模型融合 18
六、问题二模型的建立与求解 19
6.1 问题 a 模型的建立与求解 19
6.1.1 模型建立 20
6.1.2 算法实现 24
6.1.3 结果分析 25
6.2 问题 b 模型的建立与求解 26
6.3 问题 c 模型的建立与求解 33
6.4 问题 c 模型的建立与求解 35
七、问题三模型的建立与求解 37
7.1 问题 a 模型建立和求解 37
7.1.1 数据处理 37
7.1.2 模型建立与求解 37
7.1.3 参数调优及模型融合求解 37
7.2 问题 b 建模 39
7.2.1 数据处理 39
7.2.2 模型建立 40
7.2.3 参数调优及模型融合求解 40
7.3 问题 c 算法流程及实现 42
7.3.1 实现流程 42

7.3.2 结果分析 42
八、模型的分析与检验 44
8.1 误差分析 44
8.1.1 问题一的误差分析 44
8.1.2 问题二的误差分析 44
8.1.3 问题三的误差分析 44
九、模型的评价 44
9.1 模型优点 44
9.2 模型缺点 44
9.2.1 问题一的缺点 44
9.2.2 问题二的缺点 45
9.2.3 问题三的缺点 45
十、参考文献 46
十一、附录 47

一、问题重述

1.1 问题背景

出血性脑卒中指非外伤性脑实质内血管破裂引起的脑出血，占全部脑卒中发病率的10-15%。其病因复杂，通常因脑动脉瘤破裂、脑动脉异常等因素，导致血液从破裂的血管涌入脑组织，从而造成脑部机械性损伤，并引发一系列复杂的生理病理反应。出血性脑卒中起病急、进展快，预后较差，急性期内病死率高达 45-50%，约 80%的患者会遗留较严重的神经功能障碍，为社会及患者家庭带来沉重的健康和经济负担。因此，发掘出血性脑卒中的发病风险，整合影像学特征、患者临床信息及临床诊疗方案，精准预测患者预后，并据此优化临床决策具有重要的临床意义。

出血性脑卒中后，血肿范围扩大是预后不良的重要危险因素之一。在出血发生后的短时间内，血肿范围可能因脑组织受损、炎症反应等因素逐渐扩大，导致颅内压迅速增加，从而引发神经功能进一步恶化，甚至危及患者生命。因此，监测和控制血肿的扩张是临床关注的重点之一。此外，血肿周围的水肿作为脑出血后继发性损伤的标志，在近年来引起了临床广泛关注。血肿周围的水肿可能导致脑组织受压，进而影响神经元功能，使脑组织进一步受损，进而加重患者神经功能损伤。综上所述，针对出血性脑卒中后的两个重要关键事件，即血肿扩张和血肿周围水肿的发生及发展，进行早期识别和预测对于改善患者预后、提升其生活质量具有重要意义。

医学影像技术的飞速进步，为无创动态监测出血性脑卒中后脑组织损伤和演变提供了有力手段。近年来，迅速发展并广泛应用于医学领域的人工智能技术，为海量影像数据的深度挖掘和智能分析带来了全新机遇。期望能够基于本赛题提供的影像信息，联合患者个人信息、治疗方案和预后等数据，构建智能诊疗模型，明确导致出血性脑卒中预后不良的危险因素，实现精准个性化的疗效评估和预后预测。相信在不久的将来，相关研究成果及科学依据将能够进一步应用于临床实践，为改善出血性脑卒中患者预后作出贡献。

1.2 问题提出

基于附件数据与出血性脑卒中知识背景，本文需要解决下列三大问题：

问题 1. 血肿扩张风险相关因素探索建模

a) 根据附件“表 1”和“表 2”，判断患者 sub001 至 sub100 发病后 48 小时内是否发生血肿扩张事件。如发生血肿扩张事件，同时记录血肿扩张发生时间到附件“表 4”D 字段。

b）以是否发生血肿扩张事件为目标变量，基于附件“表 1” 前 100 例患者的个人史，疾病史，发病相关、附件“表 2”中其影像检查结果及附件“表 3”其影像检查结果等变量，构建模型预测所有患者（sub001 至 sub160）发生血肿扩张的概率于附件“表 4”E 字段。

问题 2. 血肿周围水肿的发生及进展建模，并探索治疗干预和水肿进展的关联关系

a）根据附件“表 2”前 100 个患者的水肿体积和重复检查时间点，构建一条全体患者水肿体积随时间进展曲线，计算前 100 个患者真实值和所拟合曲线之间存在的残差于附件“表 4”F 字段。

b）探索患者水肿体积随时间进展模式的个体差异，构建不同人群的水肿体积随时间进展曲线，并计算前 100 个患者真实值和曲线间的残差于“表 4”G 字段，所属亚组填写在

H 段。结

c）请分析附件“表 1”中不同治疗方法对水肿体积进展模式的影响。

d）请分析附件“表 1”中血肿体积、水肿体积及治疗方法三者之间的关系以及出血性脑卒中患者预后预测及关键因素探索。

问题 3. 出血性脑卒中患者预后预测及关键因素探索

a）根据附件“表 1”中前 100 个患者个人史、疾病史、发病相关及附件“表 2”、“表 3”中首次影像结果构建预测模型，预测前 100 个患者 90 天 mRS 评分于附件“表 4”I 字段。

b）根据附件“表 1”前 100 个患者所有已知临床、治疗附件“表 2”及“表 3”的影像（首次+随访）结果，预测所有含随访影像检查的患者（sub001 至 sub100,sub131 至 sub160）90 天 mRS 评分于附件“表 4”J 字段。

c）分析出血性脑卒中患者的预后（90 天 mRS）和个人史、疾病史、治疗方法及影像特征（包括血肿/水肿体积、血肿/水肿位置、信号强度特征、形状特征）等关联关系，为临床相关决策提出建议。

二、问题分析

2.1 问题一

问题 a 要进行血肿扩张风险相关因素探索建模，首先需要分析 "表 1" 和 "表 2" 的数据，以确定患者发病后 48 小时内是否发生血肿扩张事件，并记录血肿扩张发生时间。因此需要从"表 1" 中提取字段 "入院首次影像检查流水号" 和 "发病到首次影像检查时间间隔"的数据，从 "表 2" 提取字段 "各时间点流水号" 和 "HM_volume"的数据。同时我们可以可通过流水号至“附表 1-检索表格-流水号 vs 时间”中查询相应影像检查时间点，结合发病到首次影像时间间隔和后续影像检查时间间隔，可以获发病后到该次影像的时间小时数。通过对比首次检查的 "HM_volume"与后续随访 "HM_volume"的变化，判断血肿的发生时间段。在该时间段内对每一个患者建立发病小时数(h)为自变量,对应的 "HM_volume"为因变量的线性方程，初始值为首次影像时间的 HM_volume。最后根据发生血肿条件约束条件，可以使用方程求出发生血肿条件的临界时间。

问题 b 要求依据“表 1” 前 100 例患者（sub001 至 sub100）的个人史，疾病史，发病相关（字段 E 至 W）、“表 2”中其影像检查结果（字段 C 至 X）及“表 3”其影像检查结果（字段 C 至 AG）等变量，构建模型预测所有患者（sub001 至 sub160）发生血肿扩张的概率。，本文从以下三个步骤解决问题二：（1）对附件“表 1”、“表 2”、“表 3”中对数据缺失值较多删除，并且采用割线法对缺失值较少的进行补全；（2）对附件“表 1”、“表 2”、“表 3” 中特征变量进行筛选，采用随机森林（RF）特征重要度、极度梯度提升树（XGBoost）和斯皮尔曼相关系数（Spearman）先剔除冗余的强相关性变量，再按特征重要性排序，然后采用投票方式选出 3 组分别对发生血肿扩张具有显著影响的变量候选集，最后将 3 组候选集取交集后得到最终筛选出的特征变量。（3）十二个模型进行预测对比择优后，选择最优模型-随机森林优化参数干预测发生血肿扩张的概率。

问题一 a、b 的总体思路如图 2.1 所示。

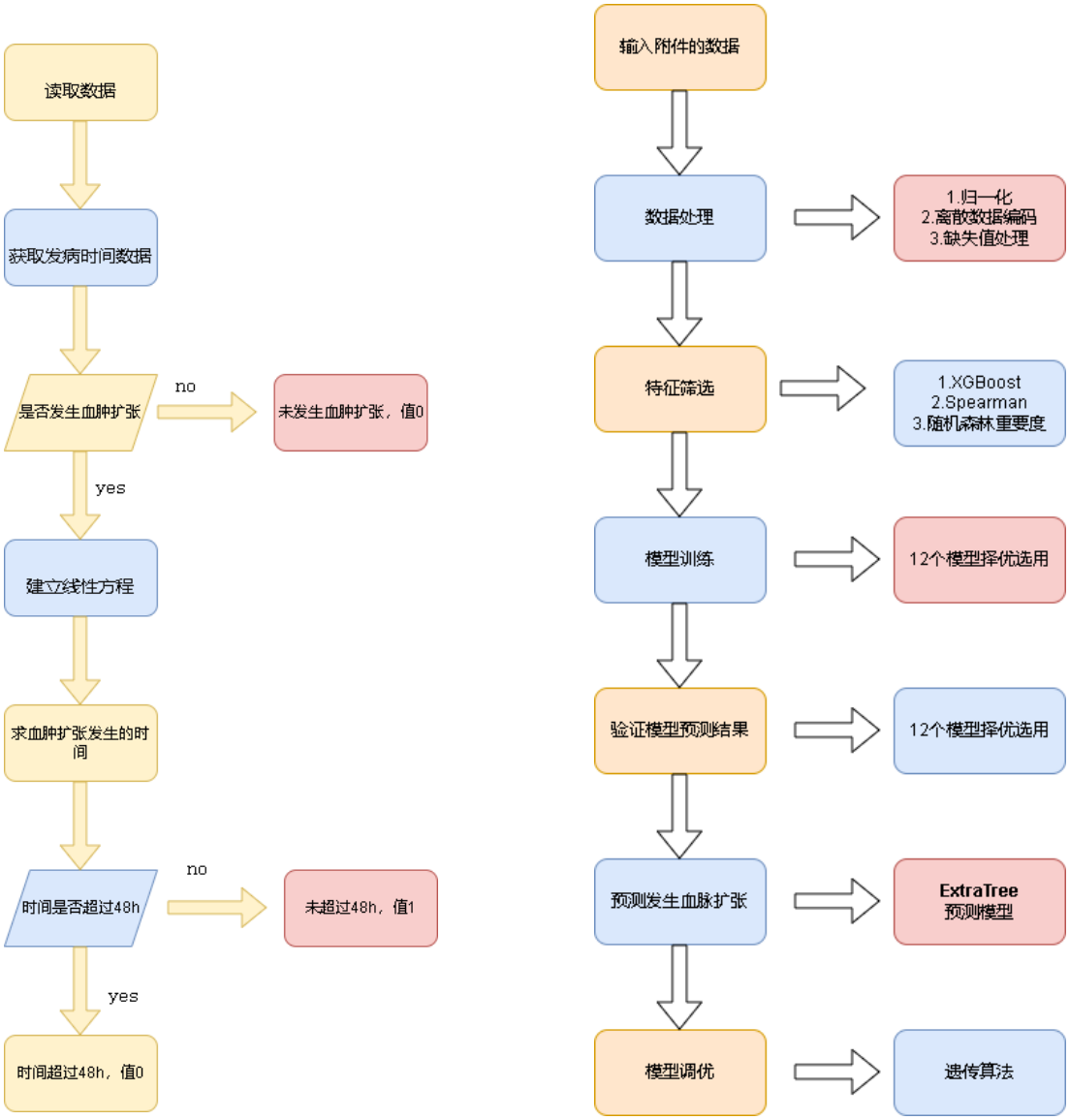


图 2.1 问题一的解题思路（左 a 右 b）

2.2 问题二

2.2.1 问题 a

问题 a 属于模型拟合类型。要求分析全体患者水肿体积与时间之间的关系。难点在于该时间是否应该考虑多次随访记录。假设首次影像检查是在治疗前，第一次随访检查是在治疗后。第一次随访所检测到的水肿体积势必会受到治疗方式等外部因素的影响，无法与时间保持唯一相关。由于每一名患者从发病到首次影像检查时间间隔不尽相同，且相邻随访时间间隔不固定。这就导致即使是在同一个时间点，有的患者可能已经接受过治疗，有的则可能尚未进行首次影像检查。即固定的时间点对不同人是否已经接受过治疗具有歧义性，而是否接受过治疗会直接影响到水肿体积。因为治疗前后水肿体积值不是同一量级，所以无法通过简单的平均去融合固定时间点的水肿体积值。综上，为了避免这种风险，选择研究发病到首次影像检查时间间隔与水肿体积的关系。因为患者从发

病到首次影像检查未经过治疗，水肿体积仅由患者个人信息决定，且后者不随时间对水肿体积产生影响，此时水肿体积仅与时间相关。考虑到自变量是时间，可以构建并训练 ARIMA 时序模型，用 WOA 算法优化超参数并用 5 折交叉验证说明模型质量，最后计算残差。

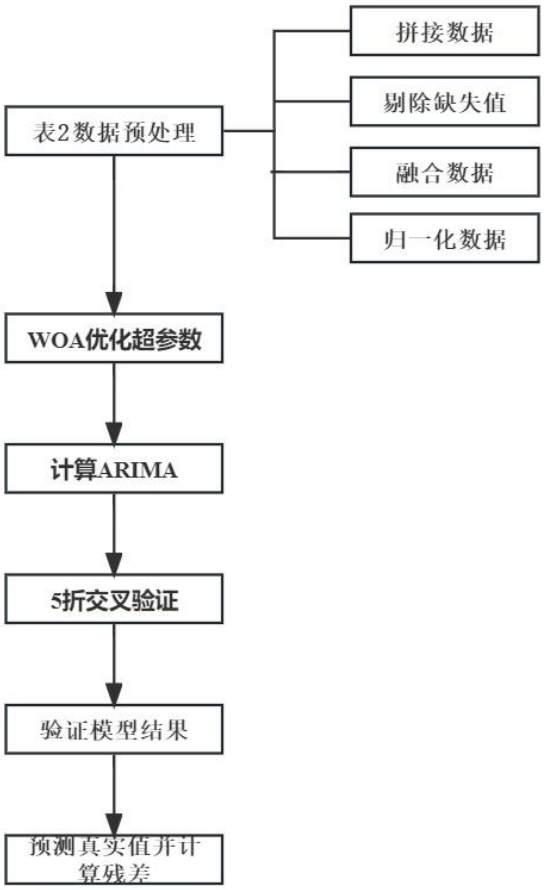


图 2.2 问题二 a 解题思路

2.2.2 问题 b

问题 b 与 a 类似，因而也选择发病到首次影像检查时间间隔作为时序自变量。不同之处在于，需要先选择特征对患者聚类分组，再训练若干模型分别计算亚组残差。首先根据肘部法确定合适的组数，在聚类算法的选择上，使用了 KMeans,MiniBatchKMeans,SpectralClustering,AgglomerativeClustering,Birch 共 5 种聚类算法，同时用轮廓系数，Calinski-Harabaz Index 和 Davies-Bouldin Index 作为评估标准，选择最佳的 AgglomerativeClustering 聚类结果。对每一类亚组都分别使用 1 至 5 次多项式、指数和对数拟合。根据 R^2 系数选择最佳拟合方式并计算亚组残差。

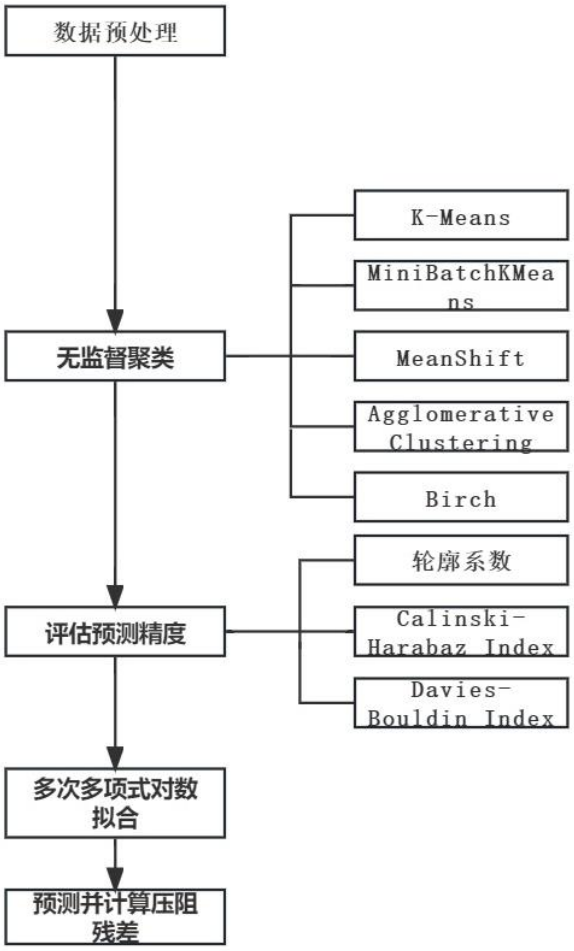


图 2.3 问题二 b 解题思路

2.2.3 问题 c

问题 c 属于相关性分析，难点在于水肿体积进展模式特征的确定。定义相邻随访记录单位时间内水肿体积变化量为此特征，计算治疗方法与其相关性。

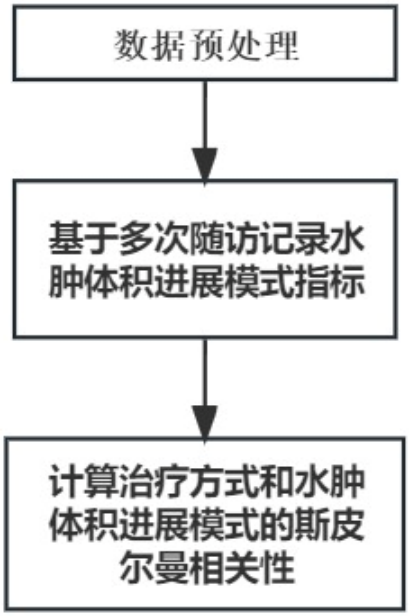


图 2.4 问题二 c 解题思路

问题 d 与 c 相似，需要注意的是应该研究患者接受治疗后，治疗方式与血肿体积和水肿体积的相关性。因而直接计算治疗方式与多次随访的血肿体积和水肿体积的相关性即可，不必考虑首次检查的血肿体积和水肿体积。

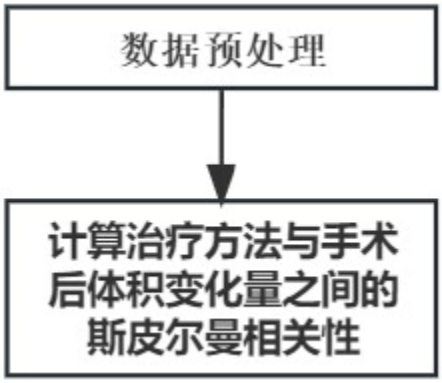


图 2.5 问题二 d 的解题思路

2.3 问题三

问题 a，要求使用请根据前 100 个患者（sub001 至 sub100）个人史、疾病史、发病相关（“表 1”字段 E 至 W）及首次影像结果（表 2，表 3 中相关字段）构建预测模型，预测患者（sub001 至 sub160）90 天 mRS 评分。首先提取表 1 中的字段 E 至 W, 以及表 2 和表 3 中首次影像结果，对数据进行处理，离散数据进行编码，连续数据进行归一化。由于特征过多，需要进行特征筛选，使用为了预测 90 天 mRs，观察 90 天 mRs 为离散整数（0-6）。因此使用分类预测模型进行运算求解。

问题 b，要求是使用前 100 个患者（sub001 至 sub100）所有已知临床、治疗（表 1 字段 E 到 W）、表 2 及表 3 的影像（首次+随访）结果，预测所有含随访影像检查的患者（sub001 至 sub100, sub131 至 sub160）90 天 mRS 评分。首先提取表 1 中的字段 E 至 W, 以及表 2 和表 3 中所有随访的影像结果，对数据进行处理，离散数据进行编码，连续数据进行归一化。由于特征过多，需要进行特征筛选，使用为了预测 90 天 mRs，观察 90 天 mRs 为离散整数（0-6）。因此使用分类预测模型进行运算求解。

问题 c，选择研究发病特征(血压值)、治疗方式相关特征、体积和位置特征以及形状及灰度特征对 90 天 mRS 的相关性。由于特征太多，先对每一类分别进行 PCA 降维重新融合新的特征后再计算斯皮尔曼相关性矩阵。

问题三 a) , b)问题 算法流程：

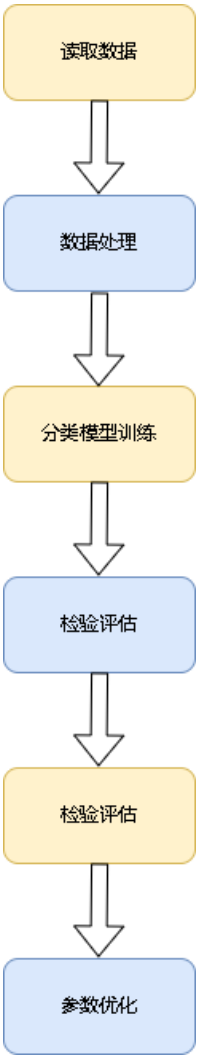


图 2.6 问题 a、b 的算法流程图

三、模型假设

3.1 问题一假设

- 假设 1. 假设所给数据无异常值，数据仅需要归一化处理和编码处理。
- 假设 2. 假设血肿体积在两个随访时间点的时间具有线性关系
- 假设 3. 除建模需要的数据外，每个个体其他的因素都相同。

3.2 问题二假设

- 假设入院首次影像检查是在治疗前，随访 1 影像检查是在治疗后。治疗方式虽然会影响血肿和水肿体积，但患者未治疗前，患者的血肿和水肿体积仅与个人史和距离发病时间长短有关，此时与治疗方式无关。

3.3 问题三假设

➢假设 1. 使用的数据无异常值，仅需要进行归一化和编码处理。
➢假设 2. 使用特征筛选后的，各个因素之间都是独立的。

四、符号说明

本文所使用的符号系统及其解释如表 4.1 所示。

表 4.1 本文所使用的部分符号说明

符号	符号说明
ℓ	损失函数
ta_k	随访数据发病到的k次随访的时间
y_{hm}	变量血肿体积
b_i	公式常数项
$\hat{y_i}$	预测输出
y_i	实际输出
$\Omega(f_k)$	正则化项
θ	模型参数
T	树党数量
N	种群中个体总数
t	当前迭代次数
b	对数螺旋形状的常数
$\mu^{(j)}$	簇的中心
j	簇
μ_i	种子点
K	核函数
h	带宽参数
x	样本点
N(x)	以 x 为中心的领域
x_i	第 i 个样本点
x_j	第 j 个样本点
σ	高斯核函数参数

注：考虑到全文连续性，其他未在表 4.1 中列出的符号将在建模和求解过程中给出解释说明。

五、问题一模型的建立与求解

5.1 问题 a 模型建立和求解

5.1.1 数据提取

（1）因此需要从″表 1″中提取字段 ″入院首次影像检查流水号″ 和 ″发病到首次影像检查时间间隔″的数据。

（2）从 ″表 2″ 提取字段 ″各时间点流水号″ 和 ″HM_volume″的数据。

（3）我们可以可通过流水号至“附表 1-检索表格-流水号 vs 时间”中查询相应影像检查时间点，结合发病到首次影像时间间隔和后续影像检查时间间隔，可以获发病后到该次影像的时间数据。

5.1.2 模型建立

根据发病后到该次影像的时间数据，得知仅前三次随访时间数据存在是在 48 小时以内及附近。因此只用跟据前三次随访的数据进行建模。

（1）判断患者是否发生血肿扩张

使用每个患者后一次的随访血肿体积和后一次的血肿体积进行对比分析判断是否发生血肿扩张，判断条件如下：

$$HM_{V_k}{}^{i} - HM_{V_{k-1}}{}^{i} > 6000 \ or \ \frac{HM_{V_k}{}^{i} - HM_{V_{k-1}}{}^{i}}{HM_{V_{k-1}}{}^{I}} > 0.33 \tag{1}$$

i 是代表不同患者，k 为血肿发生在第几次随访,$\mathrm{HM\_V_k^i}$ 为不同患者和随访次数的血肿数据。

（1）根据患者发生血肿扩张的随访数据以及首检数据建立线性方程

$$Y_{hm} = \left(\frac{HM_{V_k}{}^{i} - HM_{V_1}{}^{i}}{ta_k{}^{i} - ta_1{}^{i}}\right) \times x_{hm} + b_i \tag{2}$$

$\mathrm{ta_1}$和$\mathrm{HM_V_1}$为首次检查的时间和血肿体积,$\mathrm{x_{hm}}$是自变量发病时间。

（2）根据血肿发生条件，使用线性方程求解血肿扩张发生的时间t_{hm}，公式为：

$$f_{hm} = min\left((1+0.33) \times HM_{V_{k-1}}{}^{I}, \ HM_{V_{k-1}}{}^{i} + 6000\right) \tag{3}$$

$$t_{hm} = \mathrm{x_{hm}} = \frac{(f_{hm} - \mathrm{b_i}) \times \left(\mathrm{ta_k^i} - \mathrm{ta_1^i}\right)}{\left(\mathrm{HM_{V_k}^i} - \mathrm{HM_{V_1}^i}\right)} \tag{4}$$

（3）判断t_{hm}是否在 48 小时内，是则符合题目要求判定为 1，否则为 0 公式为：

$$\begin{cases} mubiao = 1 & if \quad t_{hm} < 48 \\ mubiao = 0 & if \quad t_{hm} >= 48 \end{cases} \tag{5}$$

$mubiao$为是否发生血肿扩张，1 为发生，0 为不发生。

5.2 问题 b 模型建立和求解

5.2.1 数据处理

（1）数据编码：编码方法，用于将离散型特征（如分类变量）转换为可以用于机器学习模型的数值型特征。编码将每个类别映射到一个向量，例如其中只有一种分类元素为 1，另一种分类元素为 2 或者其他。将性别数据进行编码，其中女为 1，男为 2.

（2）数据归一化：数据归一化将每个特征的值减去该特征的均值，然后再除以该特征的标准差，以确保每个特征的均值为 0，标准差为 1。这样处理后的数据具有零均值和单位方差，使得数据分布更加标准化。归一化可以使得特征的分布更符合模型的假设，有助于提高模型的稳定性和收敛速度。将数据中“发病到首次影像检查时间间隔”，“年龄”，“HM_volume”等字段进行归一化处理。

（3）数据缺失值处理：处理数据中的缺失值是数据预处理的重要步骤，因为缺失值会影响数据分析和机器学习模型的性能。因为数据量很少，不建议删除的方式继续宁处理。因此使用数字 0，替换数据中的缺失值。

5.2.2 特征筛选

通过对变量进行筛选，找出对影响发生血肿扩张最显著的变量。通过采用随机森林（RF）特征重要度、极度梯度提升树（XGBoost）和斯皮尔曼相关系数（Spearman）3 个方法，可以对所有变量进行近似表达。各变量间的作用可以用相关性来表达，且相关性越大，其作用愈明显。由于传统的特征筛选法很难精确地反映各个变量间的相关性，而且各个方法的原理及衡量结果也不尽相同，所以本文采用了三种包括线性和非线性特征筛选法的综合选取，并将三种方法中得到的变量相关性排序结果进行了综合，从中得到最好的特征变量，并给出了如下的特征筛选模型：

假定在方法 1 至 2 中，所选的变量集为 B1、B2、B3，各个变量集包含多个变量，且根据变量的相关性程度进行排序。选取最佳变量的指标为（1）变量出现频数（愈多愈好）（2）变量排序（愈往前愈好）。由于各项方法的评估结果都存在差异，所以本文选取了人工的投票模型选择变量。最后将 3 个 B5 集合中所得出的变集取交集得出影响发生血肿扩张的“公有变量”为最终结果。

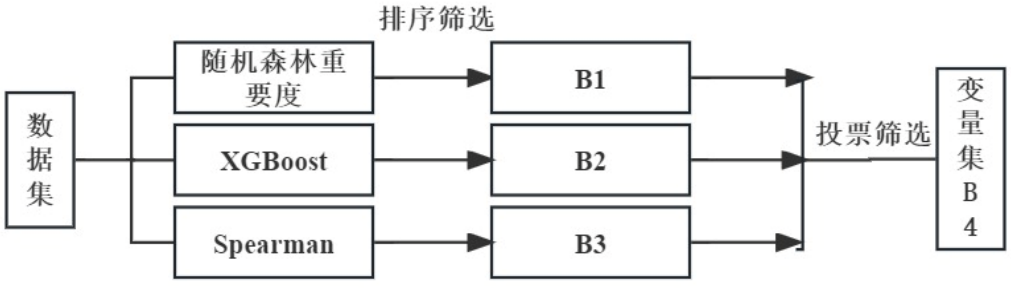


图 5.1 变量筛选流程图

（1）XGBoost

XGBoost 是 boosting 算法的一种实现方式，能够有效减少模型的误差。基本思路为不断生成新的决策树，每棵树都是基于上一颗树和目标值的差值来进行学习，从而降低模

型的偏差。

$$Objective(\theta) = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \Omega(f_k) \tag{6}$$

式中，n 是样本数。i 表示样本的索引。ℓ用于衡量预测值$\hat{y}_i$与实际值y_i 之间的差异，$\Omega(f_k)$用于控制模型的复杂度，避免过拟合。θ包括每棵树的结构和叶子节点上的预测值。

（2）随机森林重要度

随机森林特征重要度的计算通常基于两个主要指标：Gini 重要度和平均减少（Mean Decrease Impurity）。

Gini 重要度是通过评估每个特征在决策树节点上的 Gini 不纯度减少量来计算的。在随机森林中，Gini 重要度是对每个特征的 Gini 减少量的平均值。

计算公式：

$$\text{Gini Importance} = \frac{\sum_{t=1}^{T} \text{Gini(t)} \times \text{Weight(t)}}{\sum_{t=1}^{T} \text{Weight(t)}} \tag{7}$$

式中， Gini(t)是树 t 上的 Gini 不纯度减少量，Weight(t)是树 t 的权重（通常为叶子节点样本数或节点样本数）。

平均减少不纯度衡量了每个特征在每个决策树上降低的不纯度的平均值。该指标用于确定每个特征对模型准确性的贡献。

计算公式：

$$\text{Mean Decrease Impurity} = \frac{\sum_{t=1}^{T} \left(\text{Impurity}_{\text{parent}}(t) - \text{Impurity}_{\text{left}}(t) - \text{Impurity}_{\text{right}}(t) \right)}{T} \tag{8}$$

式中，Impurity_parent (t)表示树 t 的父节点的不纯度，Impurity_left(t)和 Impurity_right(t)分别表示树 t 的左子节点和右子节点的不纯度。

（3）Spearman 相关系数分析

斯皮尔曼相关系数（Spearman Correlation Coefficient）是一种用于度量两个变量之间的单调关系（可能是非线性的）的统计量。它不要求变量是连续的，可以是有序变量或者连续变量。斯皮尔曼相关系数是通过计算变量的秩（rank）来确定相关性的。

公式如下：

$$r_s = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{9}$$

式中 n 是样本数量， r_s=1 表示完全正相关。r_s =−1 表示完全负相关。r_s =0 表示无相关关系

（4）排序，投票，交集

通过对 3 种不同方法筛选出的变量集进行比较，我们可以看出，不同的方法可以得到不同的特征集，但也有部分变量在每个变量集中重复出现。将每个变量集中的变量按照其是否出现分别赋值 1（出现）和 0（未出现），选择那些在多个变量集中值为 1 的变量进入候选集，对发生血肿扩张对应的 3 个候选集取交集，最终交集包含的筛选出的变量如表 5.1 表所示。

表 5.1 保留的变量

HM_volume0	original_shape_Sphericity	高血压病史
HM_MCA R_Ratio	Original_shape_SurfacevolumeRatio	糖尿病史
HM_PCA R_Ratio	NccT_original_firstorder_10Percentile	冠心病史
HM_PCA_L_Ratio	NCCT_original_firstorder_InterquartileRange	止血治疗
ED volume	NCCT_original_firstorder_MeanAbsoluteDeviation	降颅压治疗
ED_MCA_R_Ratio	NCCT_original_firstorder_Mean	降压治疗
ED_MCA_L_Ratio	NCCT_original_firstorder_Median	年龄
original_shape_Elongation	NccT_original_firstorder_Minimum	性别
original_shape_Flatness	NcCT_original_firstorder_Range	脑出血前 mRS 评分
original_shape_Maximum2DDiameterColumn	NCCT_original_firstorder_RobustmeanAbsoluteDeviation	发病到首次影像检查时间间隔
original_shape_Maximum2DDiameterRow	NCCT_original_firstorder_RootmeanSquared	收缩压
original_shape_Waximum2DDiameterSlice	NCCT_original_firstorder_Skewness	舒张压
original_shape_MinorAxisLength	NCCT_original_firstorder_Variance	

5.2.3 对比分析与模型验证

本文对“表 1”、“表 2”和“表 3”的数据共 100 条按 8：2 划分训练集和测试集，统一采用 5 折交叉验证的方式对包括 catboost、 knn、 AdaBoost 等 12 个分类预测模型进行验证，采用**准确率、召回率、精确率、**F1 作为 4 个评价指标来确定最优分类预测模型。

使用筛选后的数据继续宁训练各个模型。各个模型交叉验证的最终实验结果如图 5.11 所示。设置合适的参数对机器学习方法十分重要，为了保证实验结果的客观性和可对比性，这里用到的模型均采用默认参数。

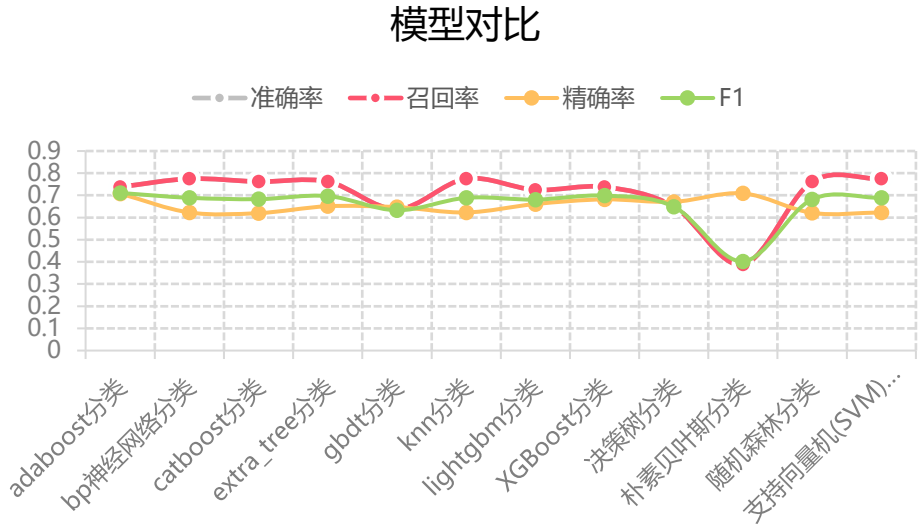


图 5.2 机器学习模型对比图

从图中可以看出 adaboost 分类模型为最优模型，其中 ExtraTrees 分类模型表现也很不错。因此采用两个模型融合的模型 AE_model，对两个模型预测的概率值进行取平均。接下来对的是否发生血肿扩张的概率预测都是用 AE_model 分类来完成。

5.2.4 模型参数调优

（1）遗传启发式算法

遗传算法（Genetic Algorithm, GA）是一种基于自然进化过程的启发式优化算法，主要用于在搜索空间中寻找解决方案的优秀近似或最优解。其基本原理包括选择、交叉和变异。

基本公式：

适应度函数（Fitness Function）：适应度函数 $f(x)$ 用于评估个体 x 的优劣程度，通常与问题的目标函数相关，目标是使适应度最大化（或最小化，根据问题类型而定）。

选择（Selection）：在选择过程中，每个个体$\mathrm{x_j}$被选择的概率$\mathrm{P(x_j)}$ 与其适应度 $f(\mathrm{x_j})$ 成正比。一种常用的选择概率计算方式是按照适应度归一化：

$$\mathrm{P(x_j)} = \frac{f(\mathrm{x_j})}{\sum_{a=1}^{N} f(\mathrm{x_a})} \tag{10}$$

交叉（Crossover）：交叉过程模拟了基因的组合。通过交叉操作，从两个父代个体x_1和 x_2 生成两个子代个体x_1'和x_2'。

变异（Mutation）：变异操作引入随机性和多样性，有助于避免陷入局部最优解。变异操作对个体的某些基因进行变异。

（2）使用遗传算法优化

为了进一步提高回归预测模型的表现，对模型参数的调整十分必要本文将采用遗传算法算法对 ExtraTrees 分类模型中决策树数量、树深度等参数进行调优，对 ababoost 模型中基学习器数量、学习率等参数进行调优。进行 5 折交叉验证的平均结果。

调优后 ExtraTrees 分类模型中参数为表：

表 5.2 调优后 ExtraTrees 分类模型中的参数

参数名	参数值
数据切分	0.8
交叉验证	5
划分时考虑的最大特征比例	None
内部节点分裂的最小样本数	2
叶子节点的最小样本数	1
叶子节点中样本的最小权重	0
树的最大深度	20
叶子节点的最大数量	50
叶子节点的最大数量	50
节点划分不纯度的阀值	0

调优后 ababoost 分类模型中参数为表：

表 5.3 调优后 ababoost 分类模型中的参数

参数名	参数值
数据切分	0.8
交叉验证	5
基分类器数量	200
学习率	0.7

分别将调优后的参数代入到 ExtraTrees 分类模型，ababoost 模型中，再次进行进行 5 折交叉验证。数据如图表 5.4、5.5 所示：

表 5.4 ExtraTrees 分类模型优化参数后的指标

	准确率	召回率	精确率	F1
交叉验证集	0.788	0.788	0.724	0.736

表 5. 5 ababoost 模型优化参数后的指标

	准确率	召回率	精确率	F1
交叉验证集	0.7	0.7	0.625	0.642

5.2.5 模型融合求解

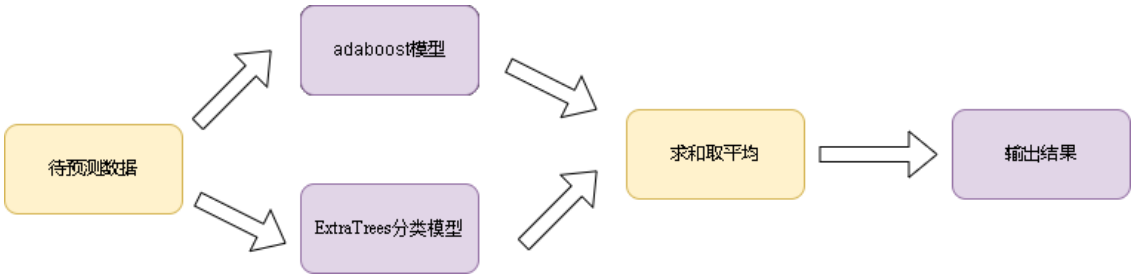


图 5.3 融合示例图

使用融合后的模型对预测数据（sub1-sub160），进行预测。输出为是出现血肿扩张的概率。如表 5.6 所示

表 5.6 发生血肿扩张概率

	首次影像检查流水号	是否发生血肿扩张（1 是，0 否）	血肿扩张时间（单位：小时）
sub001	20161212002136	0	
sub002	20160406002131	0	
sub003	20160413000006	1	12.92
sub004	20161215001667	0	
sub005	20161222000978	1	15.90
sub006	20161110001074	0	
sub007	20161208000139	0	
sub008	20161219000091	1	15.90
sub009	20161031001987	0	
sub010	20161012002008	0	
sub011	20160209000219	0	
sub012	20161031001142	0	
sub013	20161124000397	0	
sub014	20160513001799	0	
sub015	20161013001234	0	
sub016	20161130000004	0	
sub017	20160510002436	1	14.26
sub018	20160602001707	0	

六、问题二模型的建立与求解

问题二由 4 小问组成。

问题 a 自变量确定为时间间隔，尝试构建并训练 ARIMA 时序模型，用 WOA 算法优化超参数并用 5 折交叉验证说明模型质量，最后计算残差。

问题 b 与 a 类似，也选择发病到首次影像检查时间间隔作为时序自变量。然后需要先选择特征对患者聚类分组，再训练若干模型分别计算亚组残差。首先根据肘部法确定合适的组数，在聚类算法的选择上，使用 5 种聚类算法训练聚类模型，同时用 3 种评价指标选择最佳的聚类结果。对每一类亚组都分别使用 1 至 5 次多项式、指数和对数拟合。根据 R^2 系数选择最佳拟合方式并计算亚组残差。

问题 c 属于相关性分析，难点在于水肿体积进展模式特征的确定。定义相邻随访记录单位时间内水肿体积变化量及水肿体积变化速度为此特征，计算治疗方法与其相关性。

问题 d 与 c 相似，需要注意的是应该研究患者接受治疗后，治疗方式与血肿体积和水肿体积的相关性。因而直接计算治疗方式与多次随访的血肿体积和水肿体积的相关性即可。

6.1 问题 a 模型的建立与求解

6.1.1 模型建立

（1）分自回归平移(ARIMA)

问题 a 中，使用差分自回归平移(Autoregressive Integrated Moving Average，ARIMA)模型来研究发病到首次影像检查时间间隔与水肿体积的关系。ARIMA 可以有效地预测时间序列。如果时间序列具有趋势，则对其作差分后变为平稳随机序列，再用平稳时间序列去描述这一随机过程。运用最佳拟合的模型，对过去、现在的时间序列观测值对未来数据进行预测[1]。ARIMA(p,d,q)结构关系如下：

$$\mathrm{ARIMA(p,d,q)}\begin{cases}\mathrm{ARMA(p,q)}\begin{cases}\mathrm{AR(p)}\\ \mathrm{MA(q)}\end{cases}\\ \mathrm{I(d)}\end{cases} \tag{11}$$

模型结合了三种基本方法：

●自回归（AR） – 在自回归的一个给定的时间序列数据在他们自己的滞后值，这是由在模型中的“P”值表示回归的值。

●差分（I-for Integrated）–这涉及对时间序列数据进行差分以消除趋势并将非平稳时间序列转换为平稳时间序列。这由模型中的“d”值表示。如果 d = 1，则查看两个时间序列条目之间的差分，如果 d = 2，则查看在 d = 1 处获得的差分的差分，等等。

●移动平均线（MA）–模型的移动平均性质由“q”值表示，“q”值是误差项的滞后值的数量。

ARIMA 模型对数据的处理方式如下：

第 1 步：检查数据是否平稳且相关

使用 Augmented Dickey-Fuller 单位根测试测试平稳性。对于平稳的时间序列，由 ADF 测试得到的 p 值必须小于 0.05 或 5%。如果 p 值大于 0.05 或 5%，则可以得出结论：时间序列具有单位根，这意味着它是一个非平稳过程。为了将非平稳过程转换为平稳过程，应用差分方法。将差分值形成新的时间序列数据集。可以连续多次应用差分方法，产生“一阶差分”，“二阶差分”等。应用适当的差分顺序（d）使时间序列平稳。

第 2 步：确定 p、d、q

通过自相关函数（ACF）和偏相关函数（PACF）来确定自回归（AR）和移动平均（MA）过程的适当顺序，从而确定 p,d,q。对于 AR 模型，ACF 将以指数方式衰减，PACF 将用于识别 AR 模型的顺序（p）。如果在 PACF 上的滞后 1 处有一个显着峰值，那么就会有一个 1 阶 AR 模型，即 AR（1）。如果在 PACF 上有滞后 1,2 和 3 的显着峰值，那么就会有一个 3 阶 AR 模型，即 AR（3）；对于 MA 模型，PACF 将以指数方式衰减，ACF 图将用于识别 MA 过程的顺序。如果在 ACF 上的滞后 1 处有一个显着的峰值，那么就会有一个 1 阶的 MA 模型，即 MA（1）。如果在 ACF 上的滞后 1,2 和 3 处有显着的峰值，那么就会有一个 3 阶的 MA 模型，即 MA（3）。

第 3 步：估算和预测

一旦确定了参数（p，d，q），就可以估算 ARIMA 模型在训练数据集上的准确性，然后使用拟合模型使用预测函数预测测试数据集的值。

第 4 步：交叉验证

最后，我们交叉检查我们的预测值是否与实际值一致。

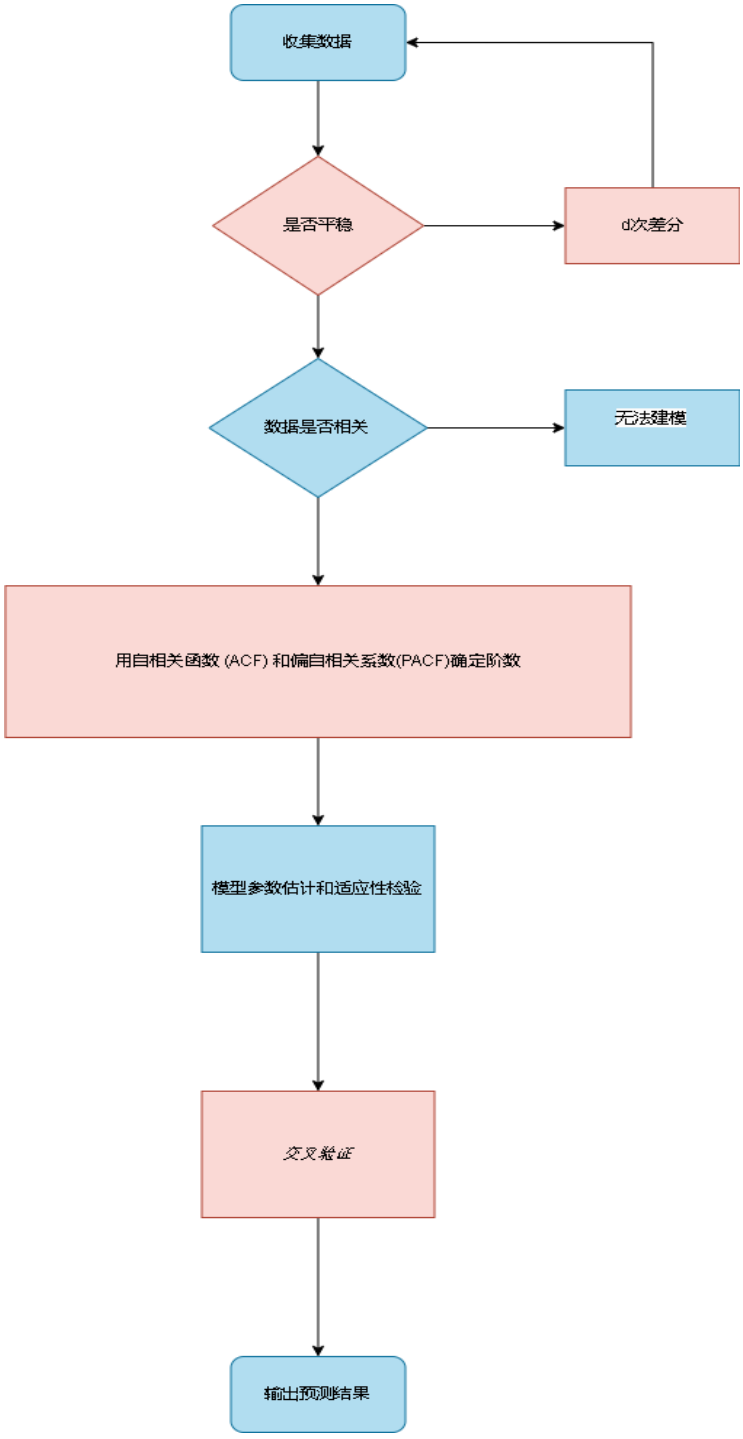


图 6.1 ARIMA 流程图[3]

（2）WOA（鲸鱼优化算法）

WOA（鲸鱼优化算法）旨在解决传统优化算法在处理复杂问题方面的局限性。WOA 主要包括：包围猎物，泡网攻击方法，搜索猎物三步：

● 在包围猎物阶段，WOA 算法将一个人设置为最佳解决方案，其他个体根据最优解更新其位置。此行为由公式 （12）和公式 （13） 表示。[3]

$$D = \left|\vec{C} \cdot X^*(t) - X^*(t)\right| \tag{12}$$

$$X(t-1) = \mathrm{X}^{*}(\mathrm{t}) - \vec{\mathrm{A}} \cdot \mathrm{D} \tag{13}$$

式中: $\vec{A}$、$\vec{C}$、为系数向量，X * (t)为当前最优解的位置向量，X(t)向量为位置向量。

上式中$\vec{A}$、$\vec{C}$计算为:

$$\vec{A} = 2a \cdot \overrightarrow{r_1} - a \tag{14}$$

$$\vec{C} = 2 \cdot \overrightarrow{r_2} \tag{15}$$

式中:a 从 2 逐渐减小到 0,r_1 和 r_2 为[0,1]中的随机向量。

●在气泡网攻击方法阶段，设计了两种方法。收缩包围圈机制:这种方法类似于包围圈猎物阶段，公式没有太大区别，只是$\vec{A}$的取值范围被限制在−1 到 1 之间。螺旋更新位置:通过在鲸鱼(X, Y)和猎物(X*，Y*)位置之间创建一个螺旋方程，数学模型如下:

$$\mathrm{X}(\mathrm{t}-1) = \mathrm{L} \cdot \mathrm{e}^{\mathrm{hd}} \cdot \cos 2\pi d + \mathrm{X}^{*}(\mathrm{t}) \tag{16}$$

式中:L = |X * (t)−X(t)|表示鲸鱼与其猎物之间的距离，d 是在- 1 和 1 之间随机生成的值。

●在自然界中，鲸鱼捕猎时是绕着猎物盘旋，同时不断收缩外壳。因此，WOA 被设计成模拟这种行为，假设选择收缩环绕机制和螺旋更新位置之一的概率为 50%。这种行为的数学模型是:

$$X(t-1) = \begin{cases} \mathrm{X}^{*}(\mathrm{t}) - \vec{\mathrm{A}} \cdot \mathrm{D} & \mathrm{p} < 0.5 \\ \mathrm{L} \cdot \mathrm{e}^{\mathrm{hd}} \cdot \cos 2\pi d + \mathrm{X}^{*}(\mathrm{t}) & \mathrm{p} \geq 0.5 \end{cases} \tag{17}$$

式中:p 为 0 ~ 1 之间随机生成的值。

（3）K 折交叉验证（K-Fold）

K-Fold 交叉验证在基于机器学习的应用中经常使用[6]。它有助于比较和选择适合特定预测分析的模型。K-Fold 易于使用，用于枚举各种模型的相对效率[7]。

在本问中，使用 K 折交叉算法来确定参数，遵循以下过程：

● 将收集到的训练数据集分为多个部分，分别设置为“train（1）”、“train（2）”、“train（k-1）”和“train（k）”。

● 在交叉验证过程中，数据集的一部分被设置为测试数据集，而其余的 k-1 部分作为训练数据集（见图 3）。平均预测精度为 RMSE，它是优化目标。

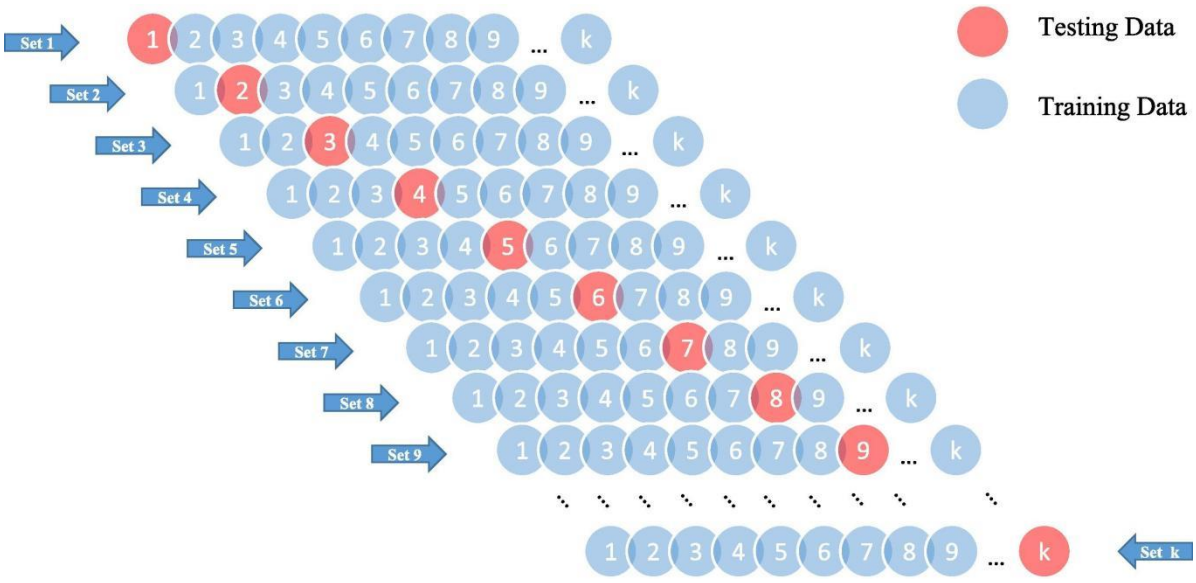


图 6.2 K-Fold 交叉验证程序示意图[7]

6.1.2 算法实现

（1）对表 1 和表 2 的数据预处理。包括以入院首次检查流水号为唯一索引，合并为 1 张表，保留“发病到首次影像检查时间间隔”和“ED_volume”等关键字段；删除缺失值；归一化水肿体积 ED_volume 等。

（2）ARIMA 起始的 pdq 值需要人为指定。恰当的起始值会大幅缩短 ARIMA 模型训练的时间，预测效果更好。本问使用 WOA 算法在规定范围内自适应搜索合适的起始值 pdq。其中，需要用 WOA 优化的目标函数按以下定义：

表 6.1 待优化函数

待优化函数 1

输入： PDQ

输出： mse

```
function ARIMA OPTIMIZER(P, D, Q)
    model = ARIMA(train_data, order = (p, d, q)).fit()
    forecast = model_fit.forecast(steps = len(test_data))
    mse = mean_squared_error(test_data, forecast)
    return mse
```

WOA 超参数设置如下：

表 6.2 WOA 算法超参数设置

捕猎个体	Pdq 最小值	Pdq 最大值	迭代次数	螺旋形状常数
50	(0,0,0)	(2,2,2)	3	0.5

WOA 会输出使得 ARIMA 预测结果的 MSE 最小的 PDQ 值，作为 ARIMA 的最佳起始值。

（3）ARIMA 训练完成后，采用 5 折交叉验证。计算 RMSE 值。

6.1.3 结果分析

表 6.3 每折 WOA 计算的最佳 PDQ 和 ARIMA 的 RMSE 值

	第 1 折	第 2 折	第 3 折	第 4 折	第 5 折
PDQ 最佳参数	112	021	011	022	000
RMSE	0.10	0.14	0.15	0.23	0.19

可见 RMSE 值尚在允许范围内，选择最优的 000 参数。可见原始数据 0 阶差分后的时序图如下：

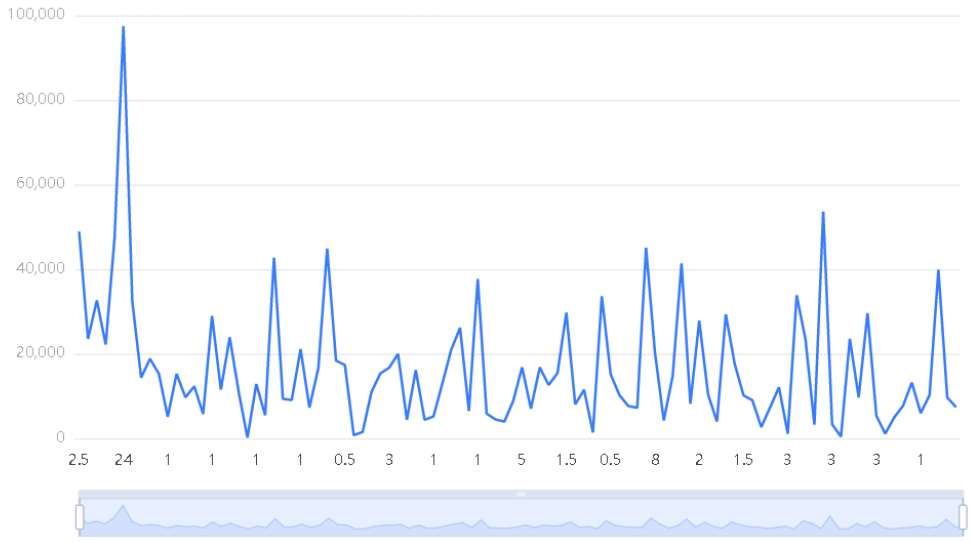


图 6.3 原始数据 0 阶差分后的时序图

ARIMA 的拟合良好。为了更直观看到拟合效果，输出 ARIMA 拟合效果图如下：

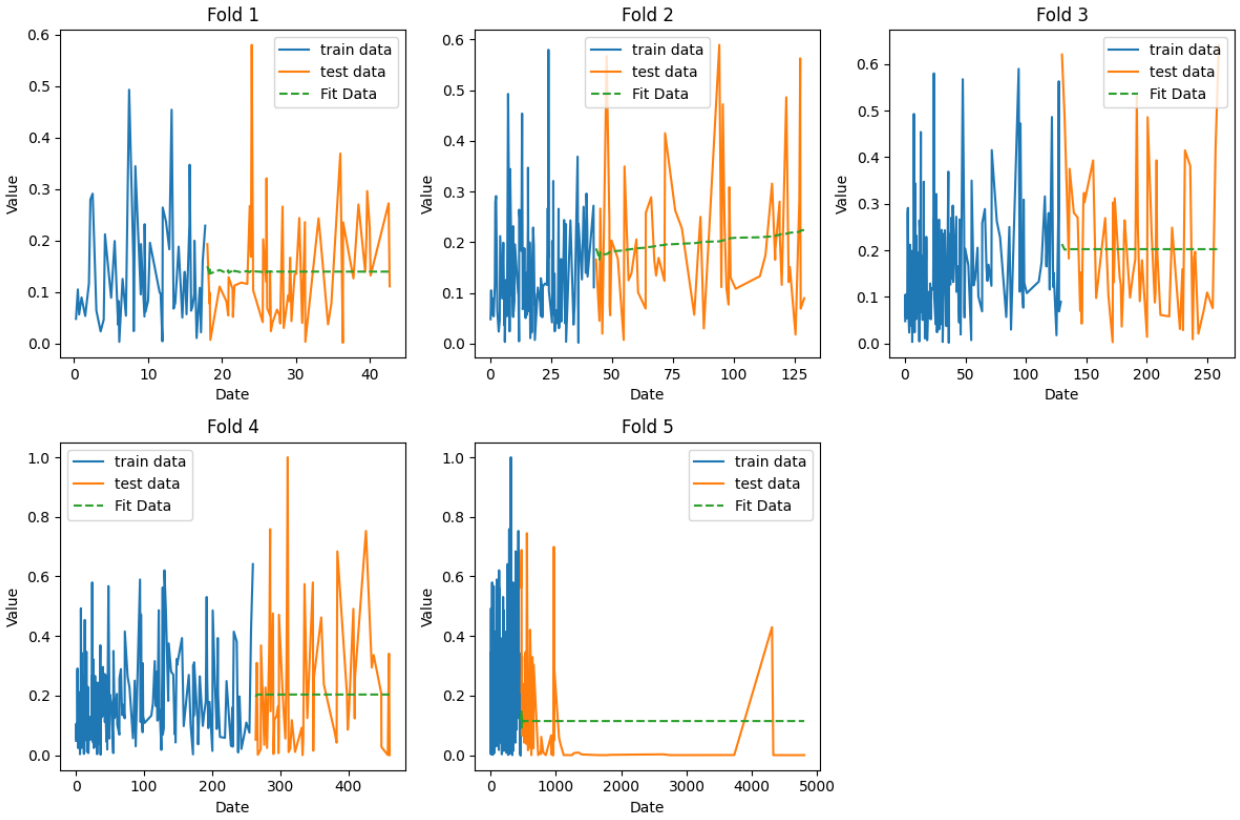


图 6.4 ARIMA 拟合效果

从图中可以看出拟合曲线较为平稳，预测能力尚且在可接受范围内，残差(全体)可信度较高。

6.2 问题 b 模型的建立与求解

6.2.1 模型建立

（1）肘部法

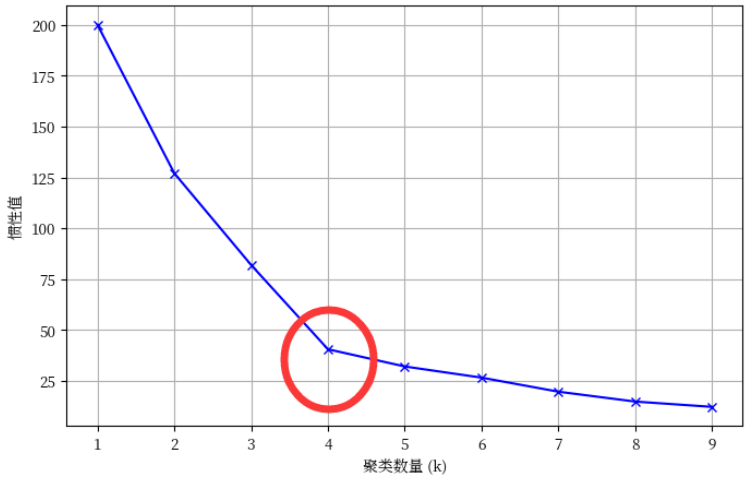


图 6.5 肘部法确定最佳 K 值为 4

“肘”方法 (Elbow method) 用于分类类别 K 值的确定，算法流程如下：

① 对于 n 个点的数据集，迭代计算 k from 1 to n，每次聚类完成后计算每个点到其所属的簇中心的距离的平方和；

② 平方和是会逐渐变小的，直到 k==n 时平方和为 0，因为每个点都是它所在的簇中心本身。

③ 在这个平方和变化过程中，会出现一个拐点也即“肘”点，下降率突然变缓时即认为是最佳的 k 值。

在决定什么时候停止训练时，肘形判据同样有效，数据通常有更多的噪音，在增加分类无法带来更多回报时，我们停止增加类别。如上图最佳分组数为 4。

（2）聚类方法

① K 均值（K-Means）

K 均值（K-Means）是一种常用的聚类算法，其目标是将数据集分成 K 个不同的簇。该算法以簇中心为基础，通过迭代更新簇中心和分配样本到最近的簇来实现聚类。计算公式如下：

$$J = \sum_{i=1}^{n} \left\| x_i - \mu^{ji} \right\|^2 \tag{17}$$

式中对数螺旋形状的常数为初始簇中心,j_i 是样本 j_i 所属的簇。

②均值漂移（MeanShift）

公式如下：

$$K(x, \mu_i) = \frac{1}{h^d} K\left(\frac{\| x - \mu_i \|}{h} \right) \tag{18}$$

$$m(x) = \frac{\sum_{xj\ in\ N(x)} K(x, x_j) \cdot x_j}{\sum_{xj\ in\ N(x)} K(x, x_j)} \tag{19}$$

③谱聚类（SpectralClustering）

● 构建相似度矩阵 w，一般选择高斯核函数计算样本点之间的相似度，公式如下:

$$w_{i,j} = e^{\frac{\| x_i - x_j \|}{2\sigma^2}} \tag{20}$$

➢构建拉普拉斯矩阵 L，一般有两种方式:

i. 随机游走型拉普拉斯矩阵，公式如下:

$$L = D^{\frac{1}{2}} W D^{\frac{1}{2}} \tag{21}$$

式中，D 为度矩阵，其对角线元素为每个样本点的度。

ii.对称型拉普拉斯矩阵，公式如下:

$$L = D - W \tag{22}$$

其中，D 和 W 分别为度矩阵和相似度矩阵。

➢对拉普拉斯矩阵 L 进行特征分解，得到 L 的特征向量矩阵 U。

➢对特征向量矩阵 U 进行 k-means 聚类或者谱聚类,将样本点划分到 k 个簇中。

谱聚类算法的主要思想是将原始数据映射到低维空间中，从而实现聚类。该算法具有较好的性能,并且可以处理非球形簇和噪声数据。

④ 凝聚层次聚类（AgglomerativeClustering）

凝聚层次聚类（AgglomerativeClustering）：是一种基于合并策略的层次聚类算法。它通过逐步合并最相似的簇来构建聚类结构。计算公式如下：

$$\mathrm{d}(\mathrm{C}_{ab},\mathrm{C}) = \max\big(\mathrm{d}(C_a,\mathrm{C}),\mathrm{d}(C_b,\mathrm{C})\big) \tag{23}$$

式中，C、C_a、C_b 为簇，且 C≠C_a、C_b。

⑤ 综合层次聚类（Birch）

BIRCH（Balanced Iterative Reducing and Clustering using Hierarchies）是一种适用于大规模数据集的层次聚类算法，旨在减少内存和计算开销。该算法通过构建一个称为 CF 树（Clustering Feature Tree）的数据结构来实现聚类。计算流程如下：

➢初始化 CF 树：

创建一个空的 CF 树，并设置参数，如簇半径阈值、每个 CF 结点能容纳的最大样本数等。

➢插入样本到 CF 树：

对于每个样本 x_i：

●遍历 CF 树，找到最适合插入样本的叶子结点或子结点。

●如果样本能够插入到某个结点，则更新该结点的 CF 结构。

●如果样本无法插入到任何结点，则创建新的叶子结点，并将样本插入。

➢簇分裂：

根据阈值判断是否需要对某个簇进行分裂，分裂时创建新的 CF 结点。

➢簇合并：根据阈值判断是否需要对某些簇进行合并。

➢聚类结果：遍历 CF 树，得到聚类结果。

BIRCH 算法通过构建 CF 树、插入样本、簇分裂和合并等步骤来实现聚类过程。CF 树结构可以高效地存储和更新聚类特征，使得 BIRCH 算法适用于大规模数据集，并能够生成层次化的聚类结果。

6.2.2 模型评估

（1）轮廓系数（S）

$$\mathrm{S} = \frac{b-a}{\max(\mathrm{a},\mathrm{b})} \tag{24}$$

式中 a 是与它同类别中其他样本的平均距离，b 是与它距离最近不同类别中样本的平均距离。轮廓系数的取值范围是[-1,1]，同类别样本距离越相近不同类别样本距离越远，分数越高，分数越高说明预测越精准。

（2）方差比准则（Calinski-Harabaz Index）

$$s(k) = \frac{tr(B_k)m-k}{tr(W_k)k-1} \tag{25}$$

式中 m 为训练样本数，k 是类别个数，B_k 是类别之间协方差矩阵，W_k 是类别内部数据协方差矩阵，tr 为矩阵的迹。类别内部数据的协方差越小，组与组之间界限不明显。该指标相对于轮廓系数来说，计算速度更快，且当簇密集分离较好时，分数更高。

（3）戴维森堡丁指数（Davies-Bouldin Index）

$$\mathrm{DB}=\frac{1}{n}\sum_{i=1}^{n}\max(j\neq i)\left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right) \tag{26}$$

式中 n 是类别个数，ci 是第 i 个类别的中心，是类别 i 中所有的点到中心的平均距离。算法生成的聚类结果越是朝着类内距离最小（类内相似性最大）和类间距离最大（类间相似性最小）变化，那么 Davies-Bouldin 指数就会越小。

6.2.3 算法实现

（1）沿用上一小问中预处理的数据。选择 '发病到首次影像检查时间间隔' 和 'ED_volume' 两列作为聚类的特征。

（2）遍历 1 到 10 聚类数量，创建并使用标准化后的数据拟合 KMeans 聚类模型，计算并记录该模型的误差平方和。使用肘部法确定最佳聚类数量为 3。

（3）使用五种聚类模型，分别计算每种模型的三种评估量。选择最佳的聚类模型的输出结果作为划分亚组。

（4）对与每个亚组，采用 1 到 5 阶多项式，指数和对数拟合。根据 R^2 指数确定最佳拟合表达式。

（5）计算亚组残差。

6.2.4 结果分析

聚类组数为 3 时，聚类结果如图：

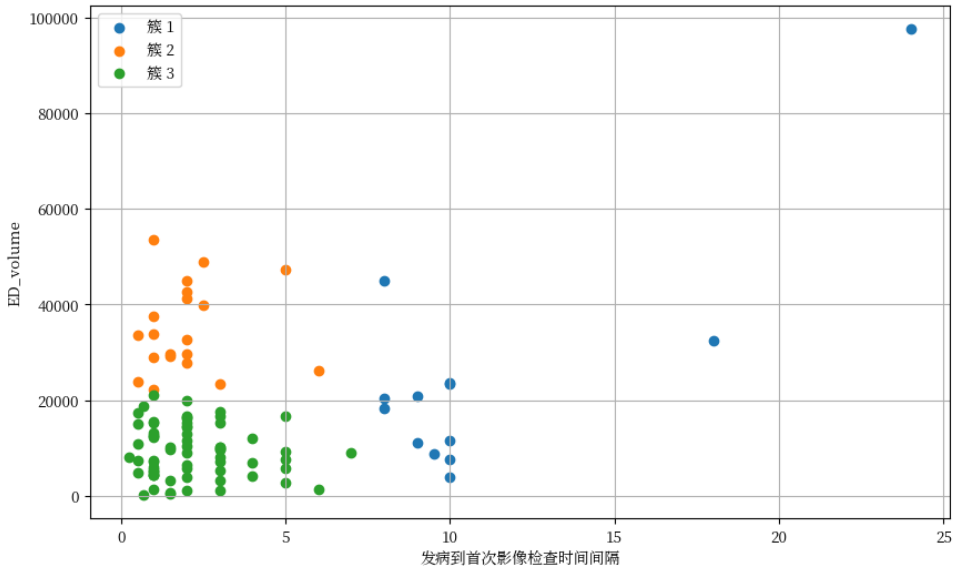


图 6.6 Kmeans 聚类结果

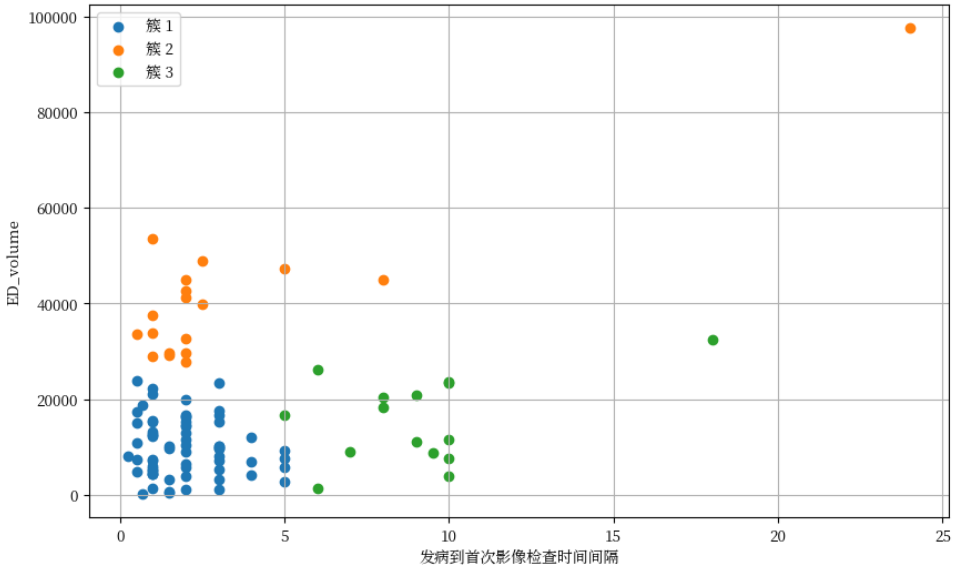


图 6.7 MiniBatchKMeans 聚类结果

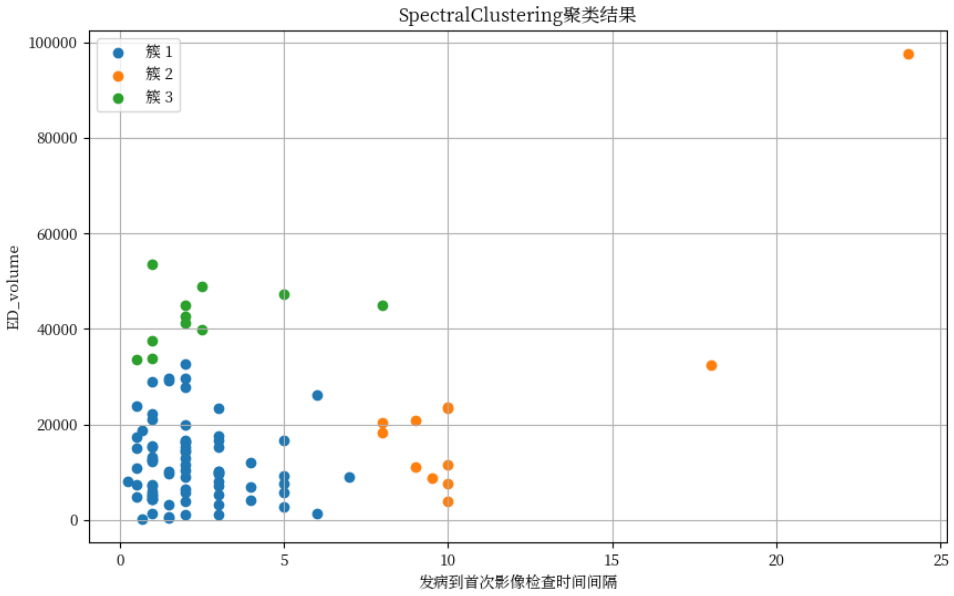


图 6.8 SpectralClustering 聚类结果

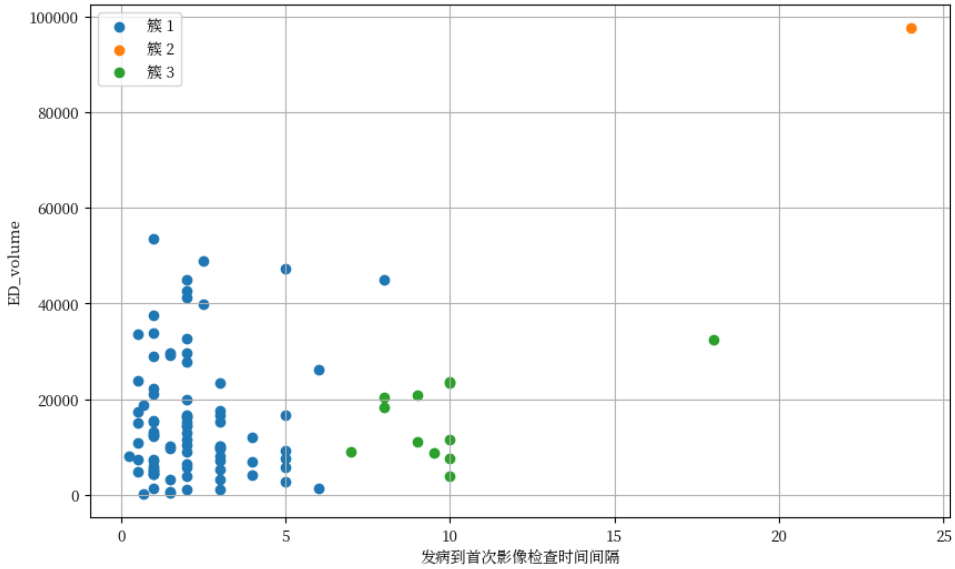


图 6.9 Brich 聚类结果

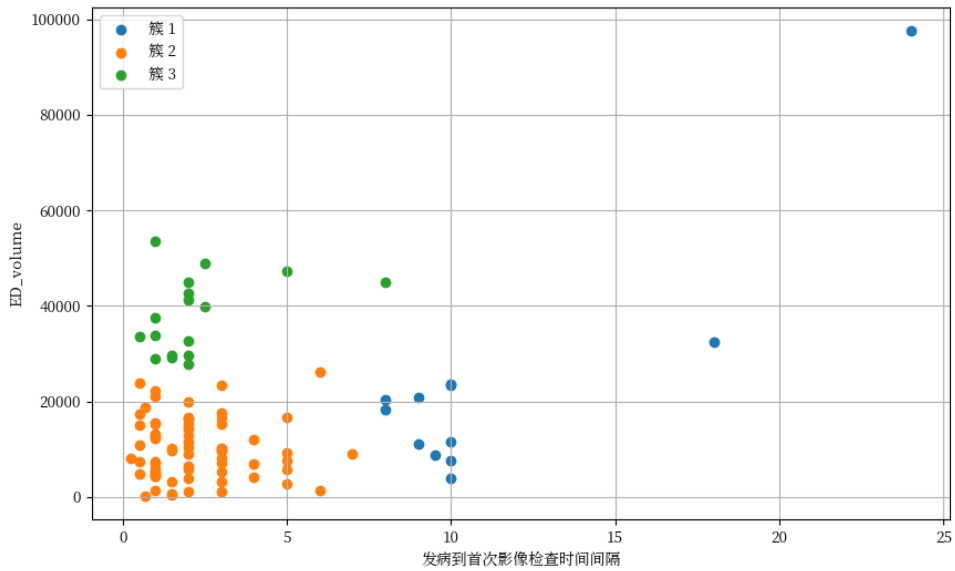


图 6.10 AgglomerativeClustering 的聚类结果

5 种聚类方法的评价指标结果见下表：

表 6.4 聚类方法的评价指标

	KMeans	MiniBatchKMeans	SpectralClustering	Birch	AgglomerativeClustering
轮廓系数	0.53	0.55	0.54	0.51	0.55
Calinski-Harabaz Index	69.75	59.03	59.77	59.06	68.31
Davies-Bouldin Index	0.78	0.79	0.74	0.51	0.74

轮廓系数和 Calinski-Harabaz Index 指标越大越好，Davies-Bouldin Index 越小越好。综

合 5 个算法在 3 个指标的数值，选择 AgglomerativeClustering 聚类算法训练的聚类模型效果最好。对三个亚组分别用多项式，对数和指数拟合，根据 R^2 指标可得用 5 次多项式拟合的效果最好。对三个亚组拟合的多项式分别起为：

y_1=17578495.6051x-2942852.6538x+236062.2113x^3-9010.7326x^4+130.6126x^5-40501858.2321

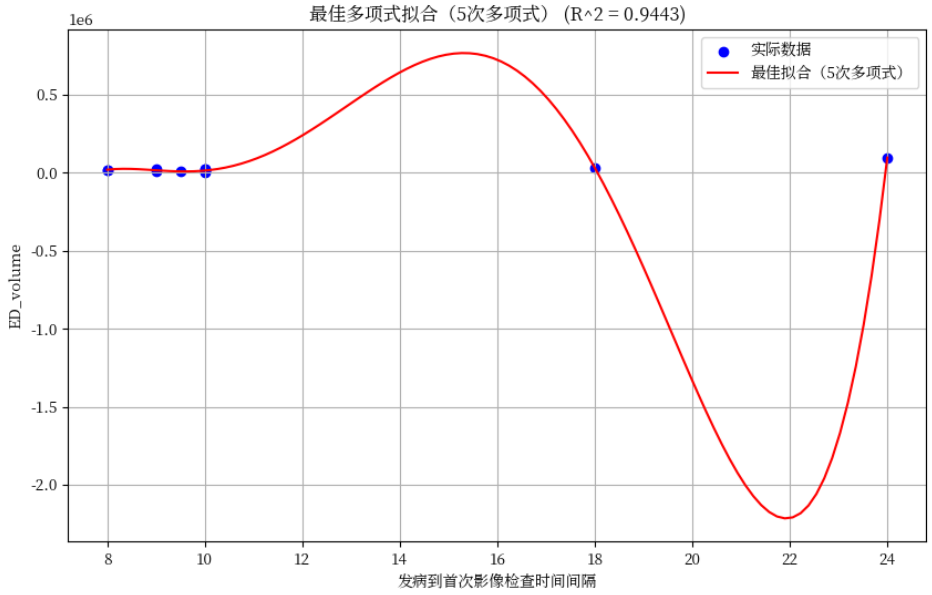


图 6.11 第一个亚组水肿体积随时间进展曲线

y_2= -28117.1960x+24150.6494x^2-8685.0929x^3+1356.2776x^4-75.8713x^5+20603.8526

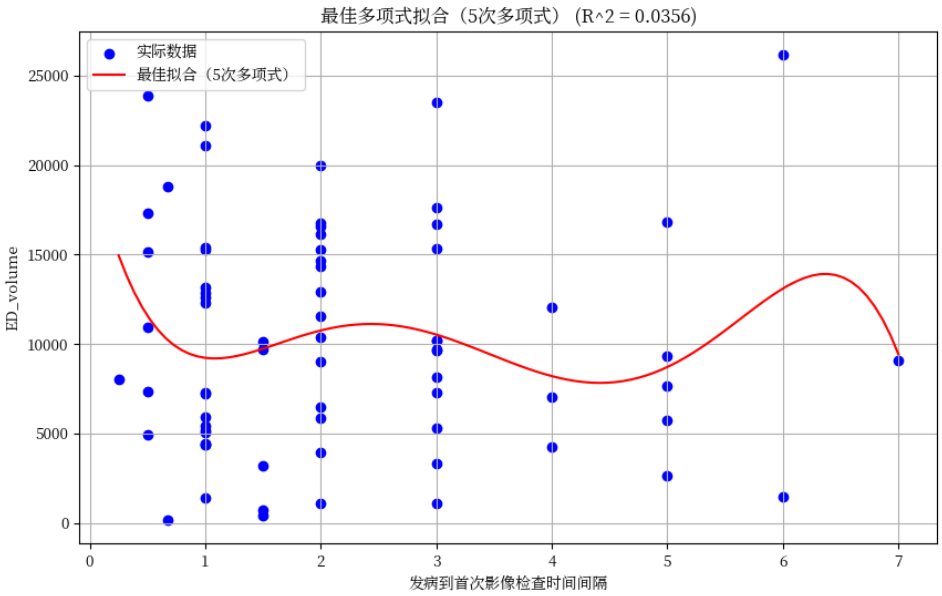


图 6.12 第二个亚组水肿体积随时间进展曲线

y_3=111498.1005x-123342.0362x^2+55080.6033x^3-9801.5828x^4+579.5127x^5+3354.8335

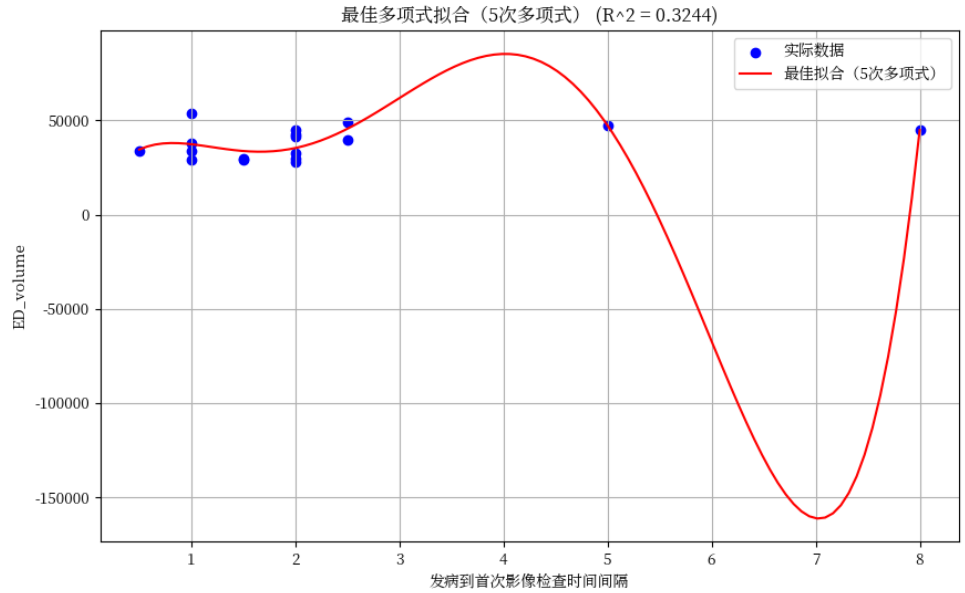


图 6.13 第三个亚组水肿体积随时间进展曲线

用三条拟合曲线分别计算各亚组残差，得到的数据较上问更准确。

6.3 问题 c 模型的建立与求解

6.3.1 模型建立

斯皮尔曼相关系数（spearman）

（1）定义：X 和 Y 为两组数据，其斯皮尔曼（等级）相关系数：

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{29}$$

式中，d_i 为 X_i 和 Y_i 之间的等级差。可以证明：r_s 位于-1 和 1 之间。

6.3.2 算法实现

（1）对表 1，表 2 和附表 1 的数据预处理。包括以入院首次检查流水号为唯一索引，合并为 1 张表，保留首次和每次随访的“HM_volume”，“ED_volume”和时间点等关键字段；删除缺失值等。

（2）构造水肿体积进展模式特征：单位时间内水肿体积的变化量，即水肿体积变化速度。计算方式为相邻两次检查的 ED_volume 与时间差的比值。由于最多有 9 次随访记录（13 次的属于奇异值，已舍去），如图，共可构造 8 组水肿体积进展模式特征 speed1-8。

	入院首次检查流水号	speed1	speed2	speed3	speed4	speed5	speed6	speed7	speed8	脑室引流	止血治疗	降颅压治疗	降压治疗	镇静、镇痛治疗	止吐护胃	营养神经
0	20161212002136	1556.602138	192.579134	204.075610	113.177959	NaN	NaN	NaN	NaN	0	1	1	1	1	1	1
1	20160406002131	-11.406206	92.543804	17.035625	-51.950641	NaN	NaN	NaN	NaN	0	1	1	1	0	1	1
2	20160413000006	1152.675307	281.811128	NaN	NaN	NaN	NaN	NaN	NaN	0	1	1	1	1	1	1
3	20160413000006	1152.675307	281.811128	NaN	NaN	NaN	NaN	NaN	NaN	0	1	1	1	1	1	1
4	20160413000006	1996.676637	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	1	1	1	1	1	1
5	20160413000006	1996.676637	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	1	1	1	1	1	1

图 6.14 8 组水肿体积进展模式特征和治疗方式

（3）计算治疗方式和 speed1-8 的斯皮尔曼相关性系数矩阵。

6.3.3 结果分析

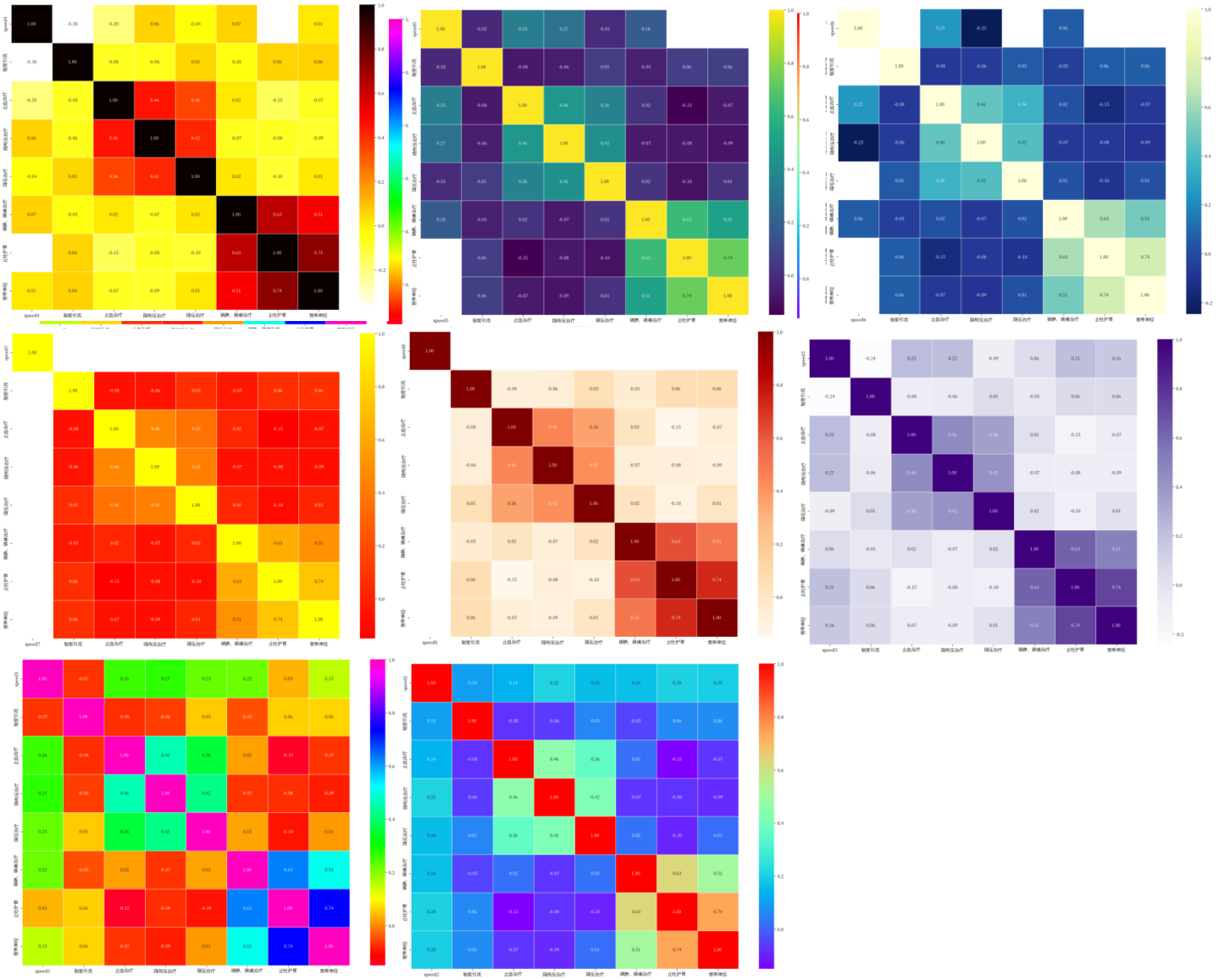


图 6.15 治疗方式和 speed1-8 的斯皮尔曼相关性系数矩阵

可以看到，治疗方式的各项特征与进展模式(水肿体积变化速度)既存在负相关又存在

正相关。但是正相关性不大，在 0 附近；负相关性最大能达到-0.25。说明治疗有效抑制了水肿体积增加，减缓了体积增长速度。这种现象在 speed3 的系数矩阵图比较明显，说明治疗一般会在第 2 次随访之后才会得到较为显著的效果。随诊随访次数越多，相关性越接近 0。很可能是水肿区域已经得到了治疗体积不再增加。

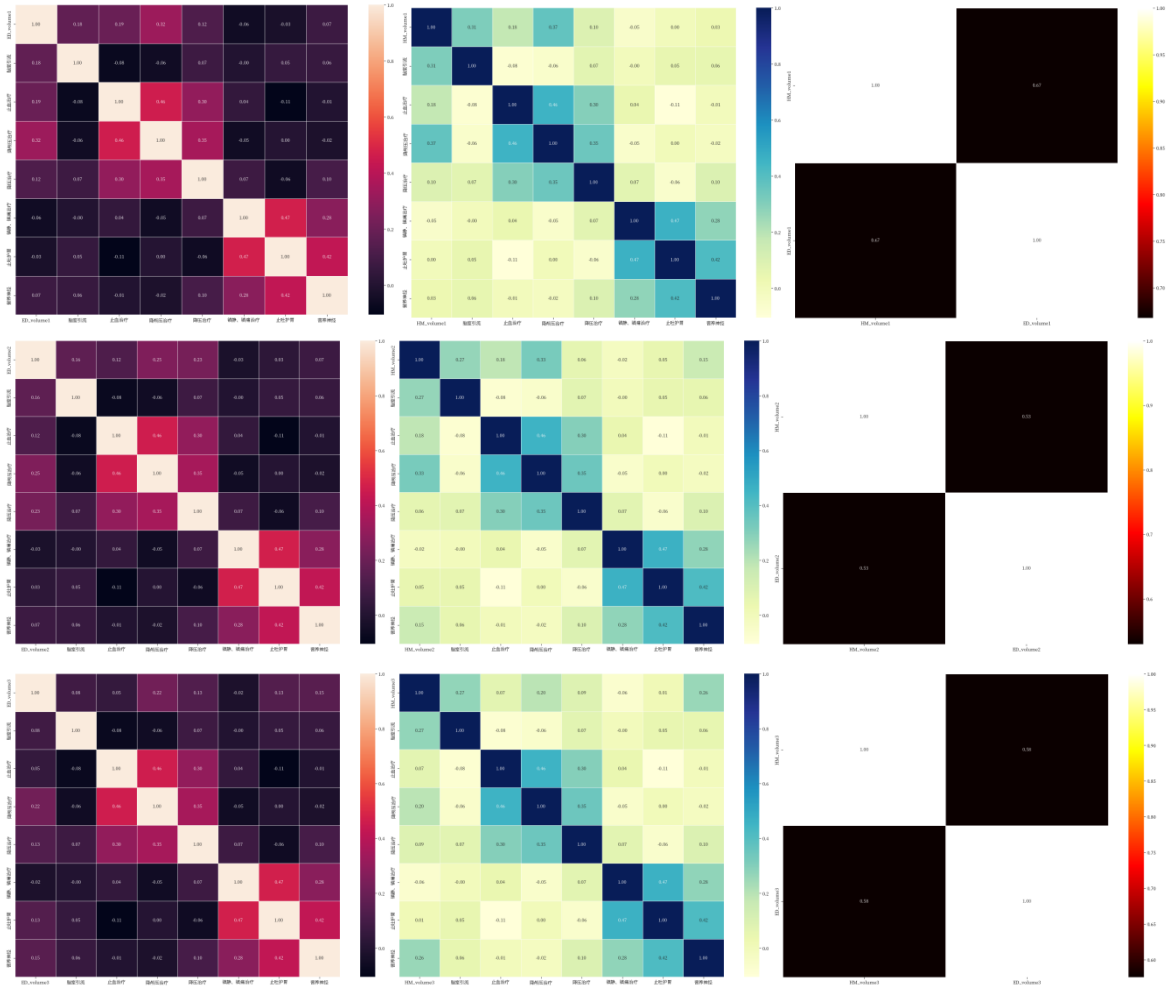
6.4 问题 c 模型的建立与求解

6.4.1 模型建立与算法实现

沿用问题 c 预处理的数据。由于要研究治疗方式与血肿和水肿体积的关系，所以只需要计算治疗方式和每次随访查到的血肿和水肿体积的斯皮尔曼相关性系数矩阵即可。

6.4.2 结果分析

治疗后（随访 1 及以后）治疗方式与血肿和水肿体积的皮尔曼相关性系数矩阵如下，自左到右分别是水肿与治疗方式、血肿与治疗方式和水肿与血肿的相关系数矩阵。自上到下分别第 1 到第 8 次随访。



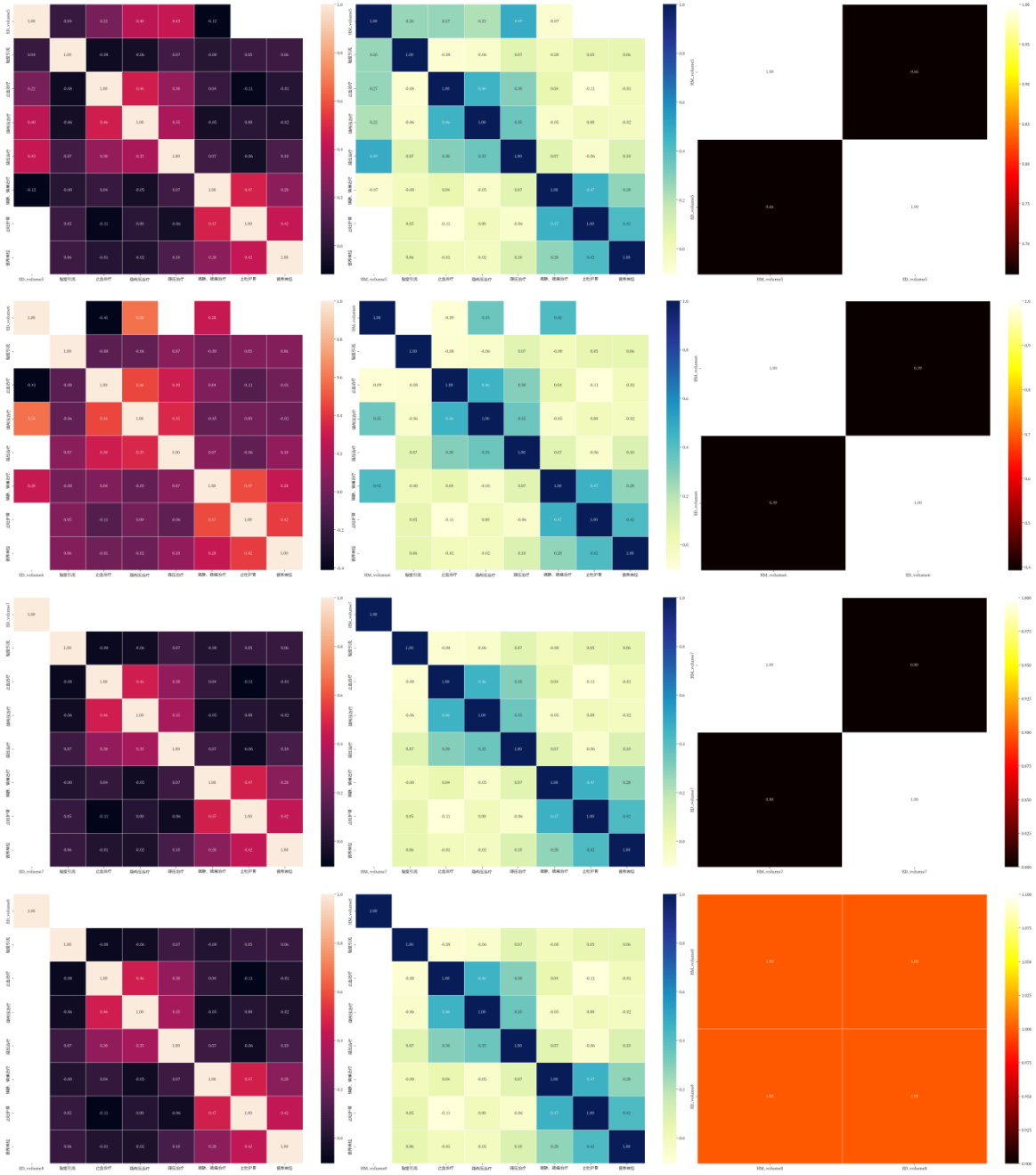


图 6.16 治疗方式与血肿和水肿体积的皮尔曼相关性系数矩阵

水肿和血肿的相关性在(0.5,1)区间内，二者具有很强的正相关性。说明血肿体积增大势必会带来水肿体积随之增大。随着随访次数增加，治疗方式的特征与血肿和水肿由正相关转成负相关，说明治疗效果正在变好，病变区域得到了有效控制。

七、问题三模型的建立与求解

7.1 问题 a 模型建立和求解

7.1.1 数据处理

本小问数据与第一问使用数据基本相同，因此使用第一问已经处理好的数据添加进表 1 中的 90 天 mRs 字段，进行建模分析。

7.1.2 模型建立

本文对“表 1”、“表 2”和“表 3”的数据共 100 条按 8：2 划分训练集和测试集，统一采用 5 折交叉验证的方式对包括 **catboost、 knn、 AdaBoost 等 12 个分类预测模型**进行验证，采用**准确率、召回率、精确率**、F1 作为 4 个评价指标来确定最优分类预测模型。

使用筛选后的数据继续宁训练各个模型。各个模型交叉验证的最终实验结果如图 7.2 所示。设置合适的参数对机器学习方法十分重要，为了保证实验结果的客观性和可对比性，这里用到的模型均采用默认参数。

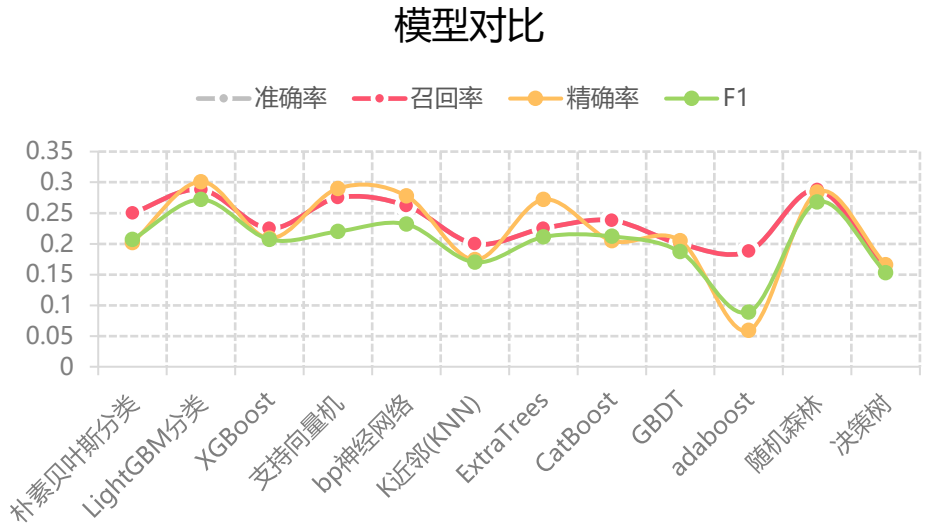


图 7.1 机器学习模型对比示例图

根据评价指标可以得到最优秀的两个模型 LightGBM 分类模型和随机森林分类模型，将两个模型进行融合对两个模型的预测值，进行求平均概率最大的值就是输出结果。

7.1.3 参数调优及模型融合求解

（1）粒子群启发式算法：

粒子群优化算法（Particle Swarm Optimization, PSO）是一种启发式优化算法，模拟了鸟群或鱼群等群体行为，用于在解空间中搜索最优解。该算法通过模拟个体之间的协作和

信息共享来逐步优化目标函数。

➢基本原理：

①初始化： 随机生成一群粒子，每个粒子代表解空间中的一个解，包含位置和速度信息。

②适应度评价： 计算每个粒子的适应度，即目标函数在该位置的取值，用来评估解的优劣。

③更新速度和位置： 根据当前位置、速度和历史最优位置，更新粒子的速度和位置，以寻找更优解。

④更新个体最优解和全局最优解： 更新每个粒子的个体最优解和全局最优解，以便于粒子之间的信息共享。

⑤重复迭代： 重复步骤 2 和 3，直到满足停止准则，例如达到最大迭代次数或适应度达到阈值。

（2）粒子群算法优化模型参数：

为了进一步提高回归预测模型的表现，对模型参数的调整十分必要本文将采用遗传算法算法对 LightGBM 分类模型中决策树数量、树深度等参数进行调优，对随机森林分类模型中基学习器数量、 学习率等参数进行调优。进行 5 折交叉验证的平均结果

随机森林分类模型优化后的参数：

表 7.1 随机森林分类模型优化后的参数表

参数名	参数值
数据切分	0.8
交叉验证	5
节点分裂评价准则	gini
决策树数量	188
有放回采样	true
袋外数据测试	false
划分时考虑的最大特征比例	auto
内部节点分裂的最小样本数	2
叶子节点的最小样本数	1
叶子节点中样本的最小权重	0
树的最大深度	100
叶子节点的最大数量	50
节点划分不纯度的阀值	0

LightGBM 分类模型优化后的参数：

表 7.2 LightGBM 分类模型优化后的参数表

参数名	参数值
数据切分	0.8
交叉验证	5
基学习器	gbdt
基学习器数量	193
学习率	0.1
L1 正则项	0.11
L2 正则项	0.21

样本征采样率	1
树特征采样率	1
节点分裂阈值	0
叶子节点中样本的最小权重	0
树的最大深度	29
叶子节点最小样本数	10

分别将优化后的参数带入到 LightGBM 分类模型和随机森林分类模型，进行 5 折交验证。得到结果如表所示：

表 7.3 随机森林优化参数后的指标

	准确率	召回率	精确率	F1
交叉验证集	0.25	0.25	0.27	0.23

表 7.4 LightGBM 优化参数后的指标

	准确率	召回率	精确率	F1
交叉验证集	0.25	0.25	0.253	0.234

模型融合：

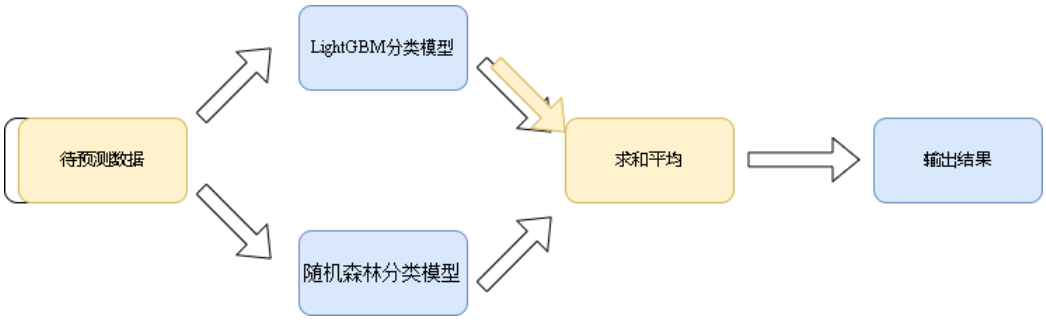


图 7.2 LightGBM 分类模型和随机森林分类模型融合示意图

使用融合后的模型对预测数据（sub1-sub160），进行 90mRS 预测。

7.2 问题 b 建模

7.2.1 数据处理

与第一小问相比，这一问多出了后续随访数据。因此需要对这些数据及西宁数据处理。因为都是连续数据，只需要进行归一化处理。归一化数据：将接收到的数据重新映射到（0，1）。

7.2.2 模型建立

本问对“表 1”、“表 2”和“表 3”的数据共 100 条按 8：2 划分训练集和测试集，统一采用 5 折交叉验证的方式对包括 **catboost、 knn、 AdaBoost 等 12 个分类预测模型**进行验证，采用**准确率、召回率、精确率**、F1 作为 4 个评价指标来确定最优分类预测模型。

使用筛选后的数据继续训练各个模型。各个模型交叉验证的最终实验结果如图 7.4 所示。设置合适的参数对机器学习方法十分重要，为了保证实验结果的客观性和可对比性，这里用到的模型均采用默认参数。

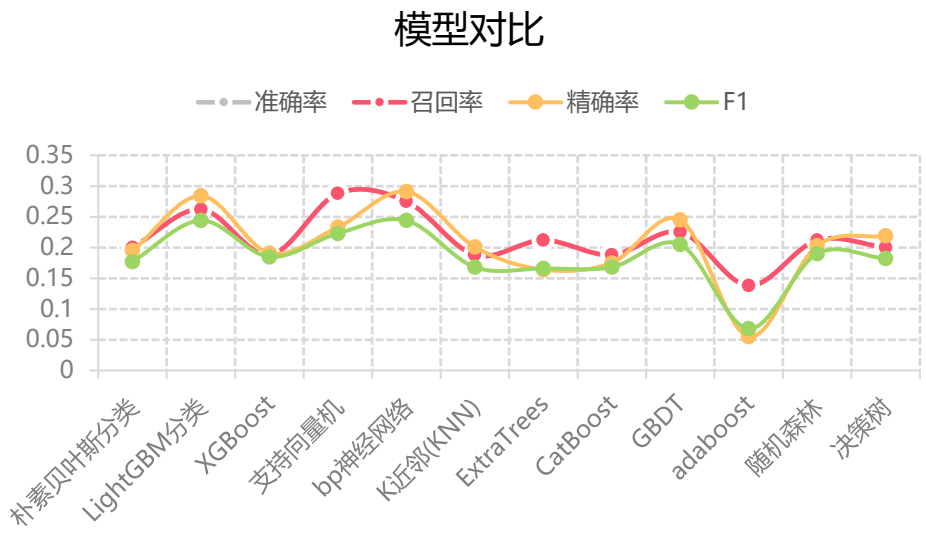


图 7.3 机器学习模型指标对比图

根据模型指标，可以得到最优秀的两个模型 LightGBM 分类模型和 BP 神经网络分类模型，将两个模型进行融合对两个模型的预测值，进行求平均概率最大的值就是输出结果。

7.2.3 参数调优及模型融合求解

（1）模拟退火算法

模拟退火算法（Simulated Annealing, SA）是一种全局优化算法，模拟了固体退火的过程。它通过模拟固体物质加热冷却过程中的行为来搜索问题的解空间，以找到最优解或近似最优解。模拟退火算法允许一定概率接受劣解，以避免陷入局部最优解，具有一定的随机性。

➢基本原理：

模拟退火算法基于固体退火的原理，其中固体物质被加热到高温然后逐渐冷却，使其达到稳定的低能态。这个过程模拟了在解空间中随机搜索的过程，然后逐渐收敛到最优解。

（2）模拟退火算法优化模型参数：

为了进一步提高 回归预测模型的表现，对模型参数的调整十分必要本文将采用遗传算法算法对 LightGBM 分类模型中 决策树数量、树深度等参数进行调优，对 BP 神经网络分类模型中迭代次数、 学习率等参数进行调优。进行 5 折交叉验证的平均结果

调优后 LightGBM 分类模型中的参数为表：

表 7.5 调优后 LightGBM 分类模型中的参数表

参数名	参数值
数据切分	0.9
交叉验证	5
激活函数	tanh
求解器	lbfgs
学习率	0.1
L2 正则项	0.21751059398118222
迭代次数	200
隐藏第 1 层神经元数量	100

调优后 BP 神经网络分类模型中的参数为表：

表 7.6 调优后 LightGBM 分类模型中的参数表

参数名	参数值
数据切分	0.9
交叉验证	5
激活函数	tanh
求解器	lbfgs
学习率	0.1
L2 正则项	0.21751059398118222
迭代次数	200
隐藏第 1 层神经元数量	100

分别将优化后的参数带入到 LightGBM 分类模型和 BP 神经网络分类模型，进行 5 折交验证。得到结果如表所示：

表 7.7 LightGBM 分类模型优化参数后的指标

	准确率	召回率	精确率	F1
交叉验证集	0.275	0.275	0.278	0.254

表 7.8 BP 神经网络分类模型优化参数后的指标

	准确率	召回率	精确率	F1
交叉验证集	0.2	0.2	0.264	0.196

模型融合：

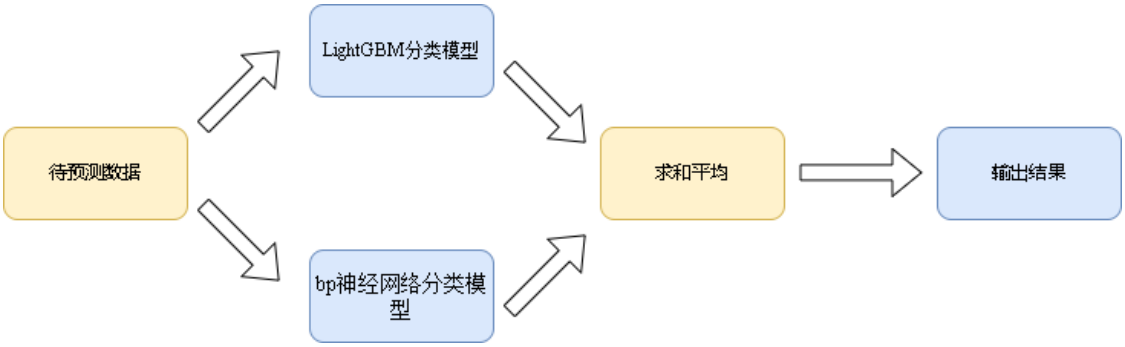


图 7.4 LightGBM 分类模型和 BP 神经网络模型融合示意图

使用融合后的模型对预测数据（sub1-sub160），进行 90mRS 预测。

7.3 问题 c 算法流程及实现

7.3.1 实现流程

（1）数据降维：.对发病特征(血压值)用 PCA 降为 2 维；治疗方式相关特征用 PCA 降为 5 维；体积和位置特征用 PCA 降为 5 维；形状及灰度特征用 PCA 降为 2 维。计算 PCA 降维后贡献率以说明新融合的特征具有代表性。

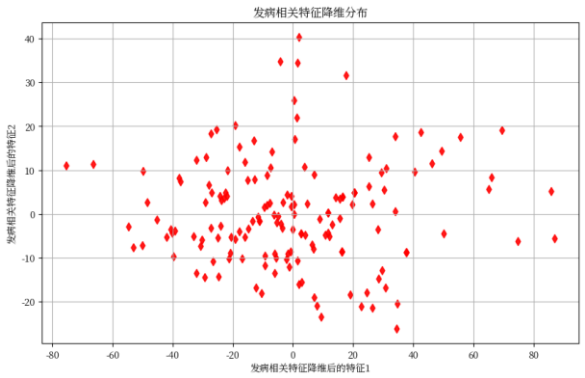
（2）计算 14 维特征与 90 天 mRS 的斯皮尔曼相关性系数矩阵。

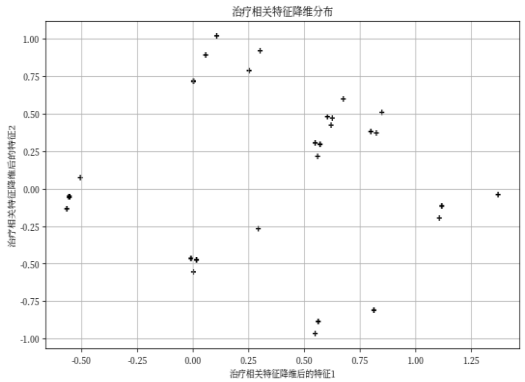
7.3.2 结果分析

表 7.9 PCA 降维后贡献率

	发病特征	治疗方式相关特征	体积和位置特征	形状及灰度特征
贡献率	1	0.91	0.94	0.99

上表说明，经过 PCA 降维后，降维特征最少都带有原特征 90%的信息。降维后的特征可以代替原特征计算相关性。以下是各类特征降维后的 2 维分布。





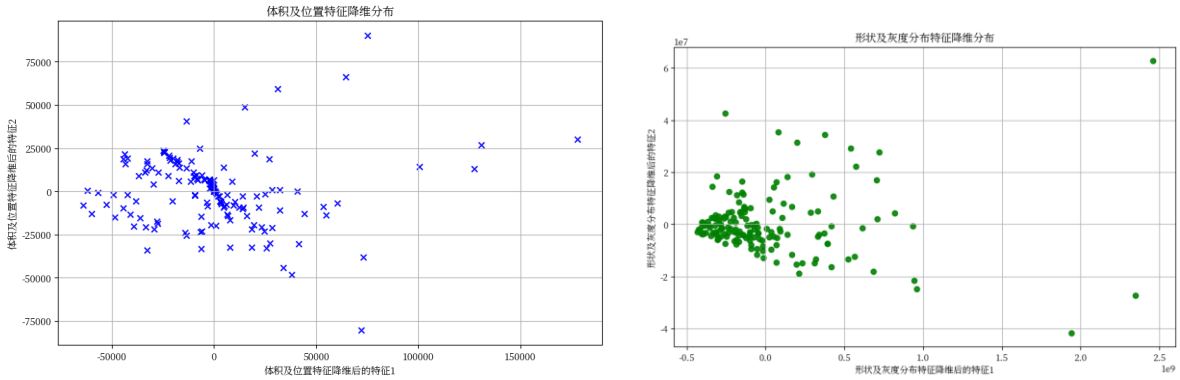


图 7.5 各类特征降维后的 2 维分布图

用降维特征计算相关性矩阵如下：

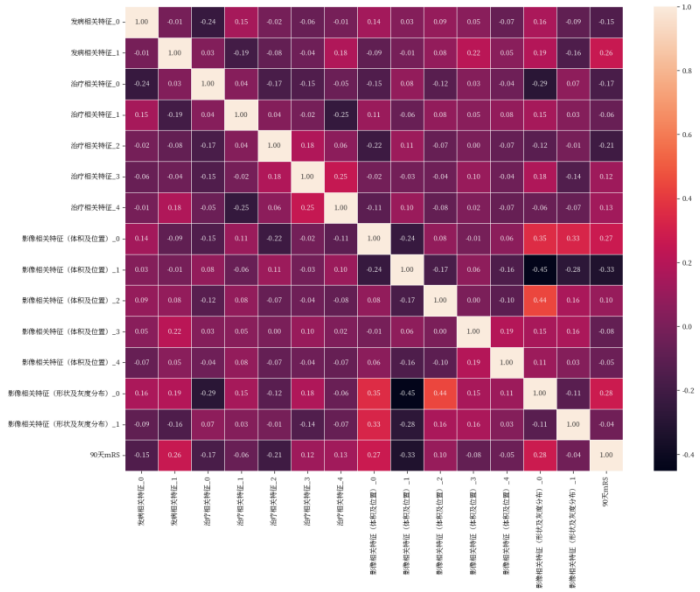


图 7.6 降维特征计算相关性矩阵图

可见发病特征，形状及灰度和体积及位置与 90 天 mRS 呈现正相关；治疗特征与 90 天 mRS 呈现负相关。说明患者经过治疗后，血肿和水肿减少，90 天 mRS 会降低。若未经过治疗，且伴有高血压，就会很容易扩大血肿和水肿致使病情严重，90 天 mRS 会逐渐升高，危及生命。

八、模型的分析与检验

8.1 误差分析

8.1.1 问题一的误差分析

（1）在假设数据中不存在异常值；
（2）使用的机器学习模型不很好的数据。
以上原因可能最终对预测结果有一定的影响。

8.1.2 问题二的误差分析

发病到首次影像检查时间间隔与水肿体积之间相关性很低，随机性很大。而 ARIMA 要求时间序列须具有趋势，所以训练得到的 ARIMA 模型拟合效果不太好。a 问应该再由多项式指对数拟合求残差与 ARIMA 预测的数据对比，选择 RMSE 更小的一组数据作为最终答案。

考虑到模型训练速度以及参数搜索范围， WOA 超参数只选用一组，虽然训练速度提升但是得到的 PDQ 未必是最适合 ARIMA 的。这也会导致 ARIMA 训练效果不好。

8.1.3 问题三的误差分析

在假设数据中不存在异常值。数据使用不当。

九、模型的评价

9.1 模型优点

(1) 算法速度快，响应性好；
(2) 搜寻的结果满足所有约束要求，有较强的的实用性；
(3) 建立了问题之间的联系，使得整个问题之间具备整体性；
(4) 在筛选主要变量的过程中，综合了降维方法的优点，使得筛选效果更佳；

9.2 模型缺点

9.2.1 问题一的缺点

（1）ExtraTrees 分类模型的缺点
① 过拟合风险：ExtraTrees 模型容易受到训练数据的噪声影响，尤其是当基学习器的数量较多时。
② 不稳定性：ExtraTrees 模型在随机性选择特征和阈值的过程中引入了不稳定性，导致同样的数据在不同训练中可能得到不同的结果。
③ 不易解释：由于 ExtraTrees 模型是基于随机特征子集的决策树集成，因此模型的解

释和可解释性相对较差。

④ 超参数选择：需要仔细调整超参数，如基学习器数量、树的深度等，以获得最佳性能。

（2）AdaBoost 分类模型的缺点：

①对噪声和异常值敏感：AdaBoost 对噪声和异常值敏感，可能会导致过拟合，特别是当异常值存在时。

②需要调整弱分类器：AdaBoost 需要选择和调整弱分类器的参数，如弱分类器的数量和类型，这可能需要一定的经验和领域知识。

③可能出现欠拟合： 如果弱分类器选取过于简单或数量不足，可能导致模型欠拟合，性能不佳。

④不适用于高维稀疏数据：AdaBoost 通常在低维数据上效果较好，对于高维稀疏数据可能不如其他模型效果好。

⑤计算开销较大： 在每一轮迭代中，AdaBoost 需要更新样本权重和重新训练弱分类器，导致计算开销相对较大。

9.2.2 问题二的缺点

ARIMA 模型不太适用于单个时间点推理，更适用于未来一段时间的预测；WOA 算法超参数较少，相比于其他启发式算法自由度不高，优化函数效率受限。

9.2.3 问题三的缺点

LightGBM 和 BP 神经网络都是常用于分类问题的模型，但它们各自也有一些缺点。

（1）LightGBM 分类模型的缺点：

①需要调参： LightGBM 有许多超参数需要调整，如学习率、树的深度、叶子节点数等。不合适的参数选择可能导致模型过拟合或欠拟合

②不适用于复杂特征： LightGBM 基于树模型，不擅长处理复杂的特征，尤其是文本或图像数据等高维稀疏特征。

③对异常值敏感： LightGBM 可能对异常值敏感，如果数据中存在异常值，可能会影响模型的性能。

④可能出现过拟合：尽管 LightGBM 有一些正则化参数，但仍有可能在较小的数据集上过拟合。

⑤不稳定性: LightGBM 使用随机采样和分桶技术，因此在不同数据划分下可能得到不同的结果，有一定的不稳定性。

（2）BP 神经网络分类模型的缺点：

①容易陷入局部最小值： BP 神经网络使用梯度下降算法进行优化，容易陷入局部最小值而无法达到全局最优解。

②对初始权重敏感： 初始权重的选择可能影响模型的性能，需要进行适当的初始化。

③需要大量数据和计算资源：BP 神经网络需要大量的训练数据才能获得良好的性能，而且训练过程需要大量的计算资源和时间。

④可能出现过拟合：当网络结构复杂或训练数据不足时，BP 神经网络容易过拟合。

⑤超参数选择困难：BP 神经网络有许多超参数需要调整，如学习率、隐藏层神经元个数、激活函数等，选择合适的参数很具挑战性。

十、参考文献

[1] 童明荣, 薛恒新, 林琳. 基于季节 ARIMA 模型的公路交通量预测. 公路交通科技. 2008(01):124-8.

[2] 吴震亚. 基于 ARIMA 模型的上海虹桥枢纽客流数据分析预测. 交通与运输. 2022;35(S1):315-20.

[3] Luo J, Gong Y. Air pollutant prediction based on ARIMA-WOA-LSTM model. Atmospheric Pollution Research. 2023;14(6):101761.

[4] Mirjalili S, Lewis A. The Whale Optimization Algorithm. Advances in Engineering Software. 2016;95:51-67.

[5] Nadimi-Shahraki MH, Zamani H, Asghari Varzaneh Z, Mirjalili S. A Systematic Review of the Whale Optimization Algorithm: Theoretical Foundation, Improvements, and Hybridizations. Archives of Computational Methods in Engineering. 2023;30(7):4113-59.

[6] Jung Y. Multiple predicting K-fold cross-validation for model selection. Journal of Nonparametric Statistics. 2018;30(1):197-215.

[7] Ling H, Qian C, Kang W, Liang C, Chen H. Combination of Support Vector Machine and K-Fold cross validation to predict compressive strength of concrete in marine environment. Construction and Building Materials. 2019;206:355-63.

[8] Hatwell J, Gaber MM, Atif Azad R M. Ada-WHIPS：解释 AdaBoost 分类及其在健康科学中的应用[J]。BMC 医学信息学与决策，2020, 20(1): 1-25。

Minz A, Mahobiya C. 使用 adaboost 对脑肿瘤类型进行 MR 图像分类[C]//2017 IEEE 第七届国际高级计算会议 (IACC)。IEEE，2017：701-705。

[9] Tripathi A, Kumar K, Misra A, et al. Colon Cancer Tissue Classification Using ML[C]//2023 6th International Conference on Information Systems and Computer Networks (ISCON). IEEE, 2023: 1-6.

[10] Valdes G, Luna J M, Eaton E, et al. MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine[J]. Scientific reports, 2016, 6(1): 37854.

[11] Sun K, He M, Xu Y, et al. Multi-label classification of fundus images with graph convolutional network and LightGBM[J]. Computers in Biology and Medicine, 2022, 149: 105909.

[12] Ji X, Chang W, Zhang Y, et al. Prediction model of hypertension complications based on GBDT and LightGBM[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 1813(1): 012008.

[13] Ramaneswaran S, Srinivasan K, Vincent P M D R, et al. Hybrid inception v3 XGBoost model for acute lymphoblastic leukemia classification[J]. Computational and Mathematical Methods in Medicine, 2021, 2021: 1-10.

[14] Jiang Y Q, Cao S E, Cao S, et al. Preoperative identification of microvascular invasion in hepatocellular carcinoma by XGBoost and deep learning[J]. Journal of cancer research and clinical oncology, 2021, 147: 821-833.

[15] Zhang, Xi., et al. "Predicting SIRS and sepsis development risk using random forest models based on clinical and laboratory features." Journal of translational medicine 16 (2018): 231.

[16] Wang, X., et al. "Predicting breast cancer metastasis risk using K-nearest neighbor method: A pilot study." Anticancer research 35.11 (2015): 5015-5022.

[17] Wang, J., et al. "Diagnostic accuracy of support vector machine-based computer-aided diagnosis for breast masses: A meta-analysis." Breast cancer research and treatment 169.2 (2019): 377-385.

[18] Zhang, H., et al. "A deep neural network-based framework for predicting diabetic retinopathy using fundus images." Journal of clinical medicine 9.4 (2020): 788.

十一、附录

问题二 a 使用 K 折交叉验证基于鲸鱼优化算法的 ARIMA 模型部分核心代码

```
import numpy as np
from statsmodels.tsa.arima.model import ARIMA
from sklearn.model_selection import TimeSeriesSplit
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import math
from pyMetaheuristic.algorithm import whale_optimization_algorithm


n_splits = 5
max_train_size = 360
tscv = TimeSeriesSplit(max_train_size=None,n_splits=n_splits)
ts=result_df

plt.figure(figsize=(12, 8))

mse_scores = []

for i, (train_index, test_index) in enumerate(tscv.split(ts)):
    train_data, test_data = ts.iloc[train_index], ts.iloc[test_index]

    def arima_optimizer(params):
        p, d, q = params
        p=int(p)
        d=int(d)
        q=int(q)
        model = ARIMA(train_data, order=(p, d, q))
        model_fit = model.fit()
        forecast = model_fit.forecast(steps=len(test_data))
        mse = mean_squared_error(test_data, forecast)
        return mse

    parameters = {
    'hunting_party': 50,
    'min_values': (0, 0, 0),
    'max_values': (2, 2, 2),
    'iterations': 2,
    'spiral_param': 0.5,
    'verbose': True
    }

    woa = whale_optimization_algorithm(target_function = arima_optimizer,**parameters)
    best_params = woa[0][:-1]
    p, d, q = best_params
    print("best_params",int(p), int(d), int(q))

    model = ARIMA(train_data, order=(int(p), int(d), int(q)))
    model_fit = model.fit()

    forecast = model_fit.forecast(steps=len(test_data))

    mse = mean_squared_error(test_data, forecast)
    mse_scores.append(mse)

    plt.subplot(2, 3, i + 1)
    plt.plot(train_data, label='train data')
    plt.plot(test_data, label='test data')
    plt.plot(test_data.index, forecast, label='Fit Data', linestyle='--')
    plt.title(f'Fold {i + 1}')
```

```
        plt.xlabel('Date')
        plt.ylabel('Value')
        plt.legend()

    rmse_scores = [math.sqrt(mse) for mse in mse_scores]

    for i, rmse in enumerate(rmse_scores):
        print(f'Fold {i+1}: RMSE = {rmse:.2f}')

    average_rmse = np.mean(rmse_scores)
    print(f'Average RMSE = {average_rmse:.2f}')

    plt.tight_layout()
    plt.show()
```

问题二 b 代码

```
import pandas as pd
table1 = pd.read_excel("表 1-患者列表及临床信息.xlsx")
table1.head()
table2 = pd.read_excel("表 2-患者影像信息血肿及水肿的体积及位置.xlsx")
table2.head
merged_data = pd.merge(table2, table1[['入院首次影像检查流水号', '发病到首次影像检查时间间隔']], left_on='入院首次影像检查流水号', right_on='入院首次影像检查流水号', how='left')核心
merged_data.drop(3, inplace=True)
merged_data.drop(5, inplace=True)
merged_data.reset_index(inplace=True)
merged_data=merged_data.drop(['index'],axis = 1)
filtered_data = merged_data.loc[merged_data['ID'].str[-3:].astype(int) < 101, ['ID','入院首次影像检查流水号', 'ED_volume', '发病到首次影像检查时间间隔']]
filtered_data
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
import warnings
warnings.filterwarnings('ignore')
! pip install mplfonts
import matplotlib.pyplot as plt
from mplfonts.bin.cli import init
init()
from mplfonts import use_font
use_font('Noto Serif CJK SC')
X = filtered_data['发病到首次影像检查时间间隔'].values.reshape(-1, 1)
y = filtered_data['ED_volume'].values
X_log = np.log(X)
y_log_transformed = np.log(y)
degrees = list(range(1, 11))
models = []
r2_scores = []
for degree in degrees:
    poly_features = PolynomialFeatures(degree=degree, include_bias=False)
    X_poly = poly_features.fit_transform(X)
    model = LinearRegression().fit(X_poly, y)
    y_pred = model.predict(X_poly)
    r2 = r2_score(y, y_pred)
    print(f"{degree}次多项式拟合的 R2 为:{r2}")
    models.append(model)
    r2_scores.append(r2)
    plt.figure(figsize=(10, 6))
    plt.scatter(X, y, color='red', label='实际数据')
    X_range = np.linspace(X.min(), X.max(), 100).reshape(-1, 1)
```

```
    plt.plot(X_range,
model.predict(PolynomialFeatures(degree=d
egree,
include_bias=False).fit_transform(X_range)
), color='purple', label=f'拟合（{degree}次多
项式）')
    plt.xlabel('发病到首次影像检查时间
间隔')
    plt.ylabel('ED_volume')
    plt.legend()
    plt.title(f'拟合（{degree}次多项式）
(R^2 = {max(r2_scores):.4f})')
    plt.grid(True)
    plt.show()
model_exp = LinearRegression()
model_exp.fit(X.reshape(-1, 1), np.log(y))
a_exp, b_exp = model_exp.coef_[0],
model_exp.intercept_
model_log = LinearRegression()
model_log.fit(X_log.reshape(-1, 1),
y_log_transformed)
a_log, b_log = model_log.coef_[0],
model_log.intercept_
x_fit = np.linspace(min(X), max(X), 100)
y_fit_exp = np.exp(a_exp * x_fit + b_exp)
y_fit_log = np.exp(a_log * np.log(x_fit) +
b_log)
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.scatter(X, y, label='原始数据')
plt.plot(x_fit, y_fit_exp, 'r', label='指数拟合
')
plt.legend()
plt.title('指数拟合')
plt.subplot(1, 2, 2)
plt.scatter(X, y, label='原始数据')
plt.plot(x_fit, y_fit_log, 'g', label='对数拟合
')
plt.legend()
plt.title('对数拟合')
plt.tight_layout()
plt.show()
y_pred = model_exp.predict(X.reshape(-1, 1))
r2 = r2_score(np.log(y), y_pred)
models.append(model_exp)
r2_scores.append(r2)
print(f"对数拟合的 R2 为:{r2}")
y_pred = model_log.predict(X_log.reshape(-
1, 1))
r2 = r2_score(y_log_transformed, y_pred)
models.append(model_log)
r2_scores.append(r2)
print(f"指数拟合的 R2 为:{r2}")
best_degree = degrees[np.argmax(r2_scores)]
best_model = models[np.argmax(r2_scores)]
plt.figure(figsize=(10, 6))
plt.scatter(X, y, color='blue', label='实际数
据')
X_range = np.linspace(X.min(), X.max(),
100).reshape(-1, 1)
plt.plot(X_range,best_model.predict(Polyno
mialFeatures(degree=best_degree,
include_bias=False).fit_transform(X_range)
), color='red', label=f'最佳拟合
（{best_degree}次多项式）')
plt.xlabel('发病到首次影像检查时间间隔')
plt.ylabel('ED_volume')
plt.legend()
plt.title(f'最佳多项式拟合（{best_degree}次
多项式） (R^2 = {max(r2_scores):.4f})')
plt.grid(True)
plt.show()
best_degree, max(r2_scores)
y_pred=best_model.predict(PolynomialFeat
ures(degree=best_degree,
include_bias=False).fit_transform(X_range))
residuals = y - y_pre
residuals_df = filtered_data.copy()
residuals_df['残差（全体）'] = residuals
residuals_df[['ID','入院首次影像检查流水
号', 'ED_volume', '发病到首次影像检查时
间间隔', '残差（全体）']]
residuals_df
residuals_df.to_excel('残差（全体）.xlsx')
from sklearn.cluster import KMeans
from sklearn.preprocessing import
```

```
StandardScaler
from scipy.spatial.distance import cdist
from sklearn import metrics
clustering_data = filtered_data[['发病到首次影像检查时间间隔', 'ED_volume']]
scaler = StandardScaler()
clustering_data_scaled = scaler.fit_transform(clustering_data)
inertia = []
K = range(1, 10)
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=42).fit(clustering_data_scaled)
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(8, 5))
plt.plot(K, inertia, 'bx-')
plt.xlabel('聚类数量 (k)')
plt.ylabel('惯性值')
plt.title('肘部法则确定最佳 k 值')
plt.grid(True)
plt.show()
LK = []
CH = []
DB = [
kmeans = KMeans(n_clusters=3, random_state=42).fit(clustering_data_scaled)
labels = kmeans.labels_
filtered_data['Cluster'] = labels
print("Kmeans")
LK.append(metrics.silhouette_score(clustering_data_scaled,labels,metric='euclidean'))
CH.append(metrics.calinski_harabasz_score(clustering_data_scaled,labels))
DB.append(metrics.davies_bouldin_score(clustering_data_scaled,labels))
print(LK)
print(CH)
print(DB)
plt.figure(figsize=(10, 6))
for i in range(3):
    cluster_data = filtered_data[filtered_data['Cluster'] == i]
    plt.scatter(cluster_data['发病到首次影像检查时间间隔'], cluster_data['ED_volume'], label=f'簇{i+1}')
plt.xlabel('发病到首次影像检查时间间隔')
plt.ylabel('ED_volume')
plt.legend()
plt.title('Kmeans 聚类结果')
plt.grid(True)
plt.show()
from sklearn.cluster import AffinityPropagation,MiniBatchKMeans,MeanShift,SpectralClustering,AgglomerativeClustering,DBSCAN,Birch
minikmeans = MiniBatchKMeans(n_clusters=3, random_state=42).fit(clustering_data_scaled)
labels = minikmeans.labels_
filtered_data['Cluster'] = labels
print("MiniBatchKMeans")
LK.append(metrics.silhouette_score(clustering_data_scaled,labels,metric='euclidean'))
CH.append(metrics.calinski_harabasz_score(clustering_data_scaled,labels))
DB.append(metrics.davies_bouldin_score(clustering_data_scaled,labels))
print(LK)
print(CH)
print(DB
plt.figure(figsize=(10, 6))
for i in range(3):
    cluster_data = filtered_data[filtered_data['Cluster'] == i]
    plt.scatter(cluster_data['发病到首次影像检查时间间隔'], cluster_data['ED_volume'], label=f'簇{i+1}')

plt.xlabel('发病到首次影像检查时间间隔')
plt.ylabel('ED_volume')
plt.legend()
plt.title('MiniBatchKMeans 聚类结果')
plt.grid(True)
```

```
plt.show()
from sklearn.cluster import AffinityPropagation,MiniBatchKMeans,MeanShift,SpectralClustering,AgglomerativeClustering,DBSCAN,Birch
sc = SpectralClustering(n_clusters=3, random_state=42).fit(clustering_data_scaled)
labels = sc.labels_
filtered_data['Cluster'] = labels
print("SpectralClustering")
LK.append(metrics.silhouette_score(clustering_data_scaled,labels,metric='euclidean'))
CH.append(metrics.calinski_harabasz_score(clustering_data_scaled,labels))
DB.append(metrics.davies_bouldin_score(clustering_data_scaled,labels))
print(LK)
print(CH)
print(DB)
plt.figure(figsize=(10, 6))
for i in range(3):
    cluster_data = filtered_data[filtered_data['Cluster'] == i]
    plt.scatter(cluster_data['发病到首次影像检查时间间隔'], cluster_data['ED_volume'], label=f'簇{i+1}')
plt.xlabel('发病到首次影像检查时间间隔')
plt.ylabel('ED_volume')
plt.legend()
plt.title('SpectralClustering 聚类结果')
plt.grid(True)
plt.show()
from sklearn.cluster import AffinityPropagation,MiniBatchKMeans,MeanShift,SpectralClustering,AgglomerativeClustering,DBSCAN,Birc
b = Birch(n_clusters=3).fit(clustering_data_scaled)
labels = b.labels_
filtered_data['Cluster'] = labels
print("Birch")
LK.append(metrics.silhouette_score(clustering_data_scaled,labels,metric='euclidean'))
CH.append(metrics.calinski_harabasz_score(clustering_data_scaled,labels))
DB.append(metrics.davies_bouldin_score(clustering_data_scaled,labels))
print(LK)
print(CH)
print(DB)
plt.figure(figsize=(10, 6))
for i in range(3):
    cluster_data = filtered_data[filtered_data['Cluster'] == i]
    plt.scatter(cluster_data['发病到首次影像检查时间间隔'], cluster_data['ED_volume'], label=f'簇{i+1}')

plt.xlabel('发病到首次影像检查时间间隔')
plt.ylabel('ED_volume')
plt.legend()
plt.title('Birch 的聚类结果')
plt.grid(True)
plt.show()
from sklearn.cluster import AffinityPropagation,MiniBatchKMeans,MeanShift,SpectralClustering,AgglomerativeClustering,DBSCAN,Birch
ac = AgglomerativeClustering(n_clusters=3).fit(clustering_data_scaled)
labels = ac.labels_
filtered_data['Cluster'] = labels
print("AgglomerativeClustering")
LK.append(metrics.silhouette_score(clustering_data_scaled,labels,metric='euclidean'))
CH.append(metrics.calinski_harabasz_score(clustering_data_scaled,labels))
DB.append(metrics.davies_bouldin_score(clustering_data_scaled,labels))
print(LK)
print(CH)
```

```
print(DB)
plt.figure(figsize=(10, 6))
for i in range(3):
    cluster_data = filtered_data[filtered_data['Cluster'] == i]
    plt.scatter(cluster_data['发病到首次影像检查时间间隔'], cluster_data['ED_volume'], label=f'簇{i+1}')
plt.xlabel('发病到首次影像检查时间间隔')
plt.ylabel('ED_volume')
plt.legend()
plt.title('AgglomerativeClustering 的聚类结果')
plt.grid(True)
plt.show()
cluster_models = []
cluster_r2_scores = []
dd=pd.DataFrame()
for i in range(3):
    cluster_data = filtered_data[filtered_data['Cluster'] == i]
    X = cluster_data['发病到首次影像检查时间间隔'].values.reshape(-1, 1)
    y = cluster_data['ED_volume'].values
    degrees = list(range(1, 6))
    models = []
    r2_scores = []
    for degree in degrees:
        poly_features = PolynomialFeatures(degree=degree, include_bias=False)
        X_poly = poly_features.fit_transform(X)

        model = LinearRegression().fit(X_poly, y)
        y_pred = model.predict(X_poly)

        r2 = r2_score(y, y_pred)

        models.append(model)
        r2_scores.append(r2)
    best_degree = degrees[np.argmax(r2_scores)]
    best_model = models[np.argmax(r2_scores)]


    coefficients = best_model.coef_
    intercept = best_model.intercept_


    formula_terms = [f"{coefficients[i]:.4f}x^{i+1}" for i in range(len(coefficients))]
    formula = "y = " + " + ".join(formula_terms) + f" + {intercept:.4f}"

    print(formula)


    plt.figure(figsize=(10, 6))
    plt.scatter(X, y, color='blue', label='实际数据')
    X_range = np.linspace(X.min(), X.max(), 100).reshape(-1, 1)
    plt.plot(X_range, best_model.predict(PolynomialFeatures(degree=best_degree, include_bias=False).fit_transform(X_range)), color='red', label=f'最佳拟合（{best_degree}次多项式）')
    plt.xlabel('发病到首次影像检查时间间隔')
    plt.ylabel('ED_volume')
    plt.legend()
    plt.title(f'最佳多项式拟合（{best_degree}次多项式） (R^2 = {max(r2_scores):.4f})')
    plt.grid(True)
    plt.show()


cluster_data['predic']=best_model.predict(Po
```

```
lynomialFeatures(degree=best_degree,
include_bias=False).fit_transform(X))
    try:
        dd=pd.concat([dd,cluster_data])
    except:
        pass

dd=dd.sort_values('ID')
dd


residuals = dd['ED_volume'] - dd['predic']


residuals_df = filtered_data.copy()
residuals_df['残差（亚组）'] = residuals


residuals_df[['ID', 'ED_volume', '发病到首次影像检查时间间隔', '残差（亚组）']]
```

问题二 c、d 代码

```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
warnings.filterwarnings("ignore",
module="matplotlib")

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)


table1 = pd.read_excel("表 1-患者列表及临床信息.xlsx")
table1

table1 = pd.read_excel("表 1-患者列表及临床信息.xlsx")


treatment_columns =
table1.columns[16:23].tolist()


residuals_df.to_excel('残差（亚组）.xlsx')

r2_scores=r2_score(dd['ED_volume'],dd['predic'])
r2_scores


plt.figure(figsize=(10, 6))
plt.scatter(dd['发病到首次影像检查时间间隔'], dd['ED_volume'], color='blue', label='Actual data')
plt.scatter(dd['发病到首次影像检查时间间隔'], dd['predic'], color='red')
plt.xlabel('发病到首次影像检查时间间隔')
plt.ylabel('ED_volume')
plt.legend()

plt.grid(True)
plt.show()



merged_data_corrected = pd.merge(res[['入院首次检查流水号', 'speed1','speed2','speed3','speed4','speed5','speed6','speed7','speed8']],
                                 table1[['入院首次检查流水号'] + treatment_columns],
                                 left_on='入院首次检查流水号', right_on='入院首次检查流水号', how='left')


filtered_data_corrected=merged_data_corrected.head(100)
filtered_data_corrected

filtered_data_corrected.to_excel('治疗方法与水肿体积增长速度关系.xlsx')


! pip install mplfonts
import matplotlib.pyplot as plt
```

```
from mplfonts.bin.cli import init

from mplfonts import use_font
use_font('Noto Serif CJK SC')

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

spearman_corr = filtered_data_corrected[['speed1', '脑室引流','止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="gist_rainbow" ,annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['speed2', '脑室引流','止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="rainbow" ,annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['speed3', '脑室引流','止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="Purples" ,annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['speed4', '脑室引流','止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="hot_r" , annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['speed5', '脑室引流','止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="viridis" ,annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['speed6', '脑室引流','止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="YlGnBu_r" ,annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['speed7', '脑室引流','止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="autumn" ,annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['speed8', '脑室引流','止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经
```

```
']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, cmap="OrRd" ,annot=True, fmt=".2f", linewidths=0.5)

import pandas as pd
table1 = pd.read_excel("表 1-患者列表及临床信息.xlsx")
table1.head()

table2 = pd.read_excel("表 2-患者影像信息血肿及水肿的体积及位置.xlsx")
table2.head()


treatment_columns = table1.columns[16:23].tolist()

merged_data_corrected = pd.merge(table2[['ID', '入院首次检查流水号
','HM_volume1','HM_volume2','HM_volume3','HM_volume4','HM_volume5','HM_volume6','HM_volume7','HM_volume8',

'ED_volume1','ED_volume2','ED_volume3', 'ED_volume4','ED_volume5','ED_volume6', 'ED_volume7','ED_volume8']], table1[['入院首次检查流水号'] + treatment_columns], left_on='入院首次检查流水号', right_on='入院首次检查流水号', how='left')


filtered_data_corrected = merged_data_corrected.loc[merged_data_corrected['ID'].str[-3:].astype(int) < 100]

filtered_data_corrected.head()

spearman_corr = filtered_data_corrected[['ED_volume1', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume1', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="YlGnBu", annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume1', 'ED_volume1']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, cmap="hot", annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['ED_volume2', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume2', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治
```

```
疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="YlGnBu", annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume2', 'ED_volume2']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, cmap="hot", annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['ED_volume3', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume3', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="YlGnBu", annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume3', 'ED_volume3']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, cmap="hot", annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['ED_volume4', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume4', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="YlGnBu", annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume4', 'ED_volume4']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, cmap="hot", annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['ED_volume5', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume5', '脑室引流',
```

```
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="YlGnBu", annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume5', 'ED_volume5']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, cmap="hot", annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['ED_volume6', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume6', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="YlGnBu", annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume6', 'ED_volume6']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, cmap="hot", annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['ED_volume7', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume7', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="YlGnBu", annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume7', 'ED_volume7']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, cmap="hot", annot=True, fmt=".2f", linewidths=0.5)

spearman_corr = filtered_data_corrected[['ED_volume8', '脑室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr, annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume8', '脑
```

```
室引流',
        '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经']].corr(method='spearman')
plt.figure(figsize=(16, 12))
sns.heatmap(spearman_corr,cmap="YlGnBu", annot=True, fmt=".2f", linewidths=0.5)


spearman_corr = filtered_data_corrected[['HM_volume8', 'ED_volume8']].corr(method='spearman')
plt.figure(figsize=(16, 12))
    sns.heatmap(spearman_corr, cmap="hot", annot=True, fmt=".2f", linewidths=0.5)
```

问题三 c 代码

```
    import pandas as pd
    ! pip install mplfonts
    import matplotlib.pyplot as plt
    from mplfonts.bin.cli import init
    # init() # Colab 首次运行时请解除注释 之后请加上注释
    from mplfonts import use_font
    use_font('Noto Serif CJK SC')#指定中文字体

    data = pd.read_excel("merged_df.xlsx")

    onset_related_cols = ['血压_最大值', '血压_最小值']

    treatment_related_cols = [
        '脑室引流', '止血治疗', '降颅压治疗', '降压治疗', '镇静、镇痛治疗', '止吐护胃', '营养神经'
    ]

    volume_and_location_cols = [col for col in data.columns if any(sub in col for sub in ['Hemo', 'ED', 'ACA_L', 'ACA_R', 'MCA_L', 'MCA_R', 'PCA_L', 'PCA_R', 'Pons_Medulla_L', 'Pons_Medulla_R', 'Cerebellum_L', 'Cerebellum_R'])]
    shape_and_gray_distribution_cols = [col for col in data.columns if 'NCCT' in col and col not in volume_and_location_cols]

    df=pd.DataFrame()

    from sklearn.decomposition import PCA
    import matplotlib.pyplot as plt

    onset_data = data[onset_related_cols].fillna(0)
    pca = PCA(n_components=2)
    newX = pca.fit_transform(onset_data)
    print(pca.explained_variance_ratio_.sum())
    plt.figure(figsize=(10, 6))
    plt.scatter(newX[:, 0], newX[:, 1], color='red',alpha=0.9,marker='d')
    plt.xlabel('发病相关特征降维后的特征 1')
    plt.ylabel('发病相关特征降维后的特征 2')
    plt.title('发病相关特征降维分布')

    plt.grid(True)
    plt.show()

    for i in range(2):
        df['发病相关特征_%d'%i]=newX[:, i]


    treatment_data = data[treatment_related_cols].fillna(0)
    pca = PCA(n_components=5)
    newX = pca.fit_transform(treatment_data)
    print(pca.explained_variance_ratio_.su
```

```
m())
    plt.figure(figsize=(10, 6))
    plt.scatter(newX[:, 0], newX[:, 1], color='black',alpha=0.9,marker='+')
    plt.xlabel('治疗相关特征降维后的特征 1')
    plt.ylabel('治疗相关特征降维后的特征 2')
    plt.title('治疗相关特征降维分布')

    plt.grid(True)
    plt.show()

    for i in range(5):
        df['治疗相关特征_%d'%i]=newX[:, i]

    volume_location_data = data[volume_and_location_cols].fillna(0)
    pca = PCA(n_components=5)
    newX = pca.fit_transform(volume_location_data)
    print(pca.explained_variance_ratio_.sum())
    plt.figure(figsize=(10, 6))
    plt.scatter(newX[:, 0], newX[:, 1], color='blue',alpha=0.9,marker='x')
    plt.xlabel('体积及位置特征降维后的特征 1')
    plt.ylabel('体积及位置特征降维后的特征 2')
    plt.title('体积及位置特征降维分布')
    plt.grid(True)
    plt.show()

    for i in range(5):
        df['影像相关特征（体积及位置）_%d'%i]=newX[:, i]

    shape_gray_data = data[shape_and_gray_distribution_cols].fillna(0)
    pca = PCA(n_components=2)
    newX = pca.fit_transform(shape_gray_data)
    print(pca.explained_variance_ratio_.sum())
    plt.figure(figsize=(10, 6))
    plt.scatter(newX[:, 0], newX[:, 1], color='green',alpha=0.9)
    plt.xlabel('形状及灰度分布特征降维后的特征 1')
    plt.ylabel('形状及灰度分布特征降维后的特征 2')
    plt.title('形状及灰度分布特征降维分布')

    plt.grid(True)
    plt.show()

    for i in range(2):
        df['影像相关特征（形状及灰度分布）_%d'%i]=newX[:, i]

    df['90 天 mRS']=data['90 天 mRS']

    import pandas as pd
    import numpy as np
    import seaborn as sns

    spearman_corr = df.corr(method='spearman')
    plt.figure(figsize=(16, 12))
    sns.heatmap(spearman_corr, annot=True, fmt=".2f", linewidths=0.5)
```