2023 年武汉理工大学大学生数学建模竞赛

题 目：基于 PSO-XGBoost 的糖尿病预测模型

摘 要：

本文针对题目所提供体检数据集，研究并构建通过体检指标来预测血糖的模型、糖尿病风险的评估模型，期望对糖尿病进行科学有效的干预、预防和治疗，来降低发病率和提高患者的生活质量。

在数据的预处理方面，我们团队进行整体数据集的概述，对缺少值和异常值进行分析和选择，同时利用中间值填充法和**高斯混合模型**来进行数据的完善和舍去。

对于问题一，在进行数据预处理后，我们团队首先利用 Spearman 来分析缺失率过大的指标，排除乙肝指标的影响。然后，我们团队以 CART **回归树**为基学习器的 XGBoost 集成学习算法，再依据R^2进行**特征逐步消除**。最后，选出了十项指标作为特征值：'***天门冬氨酸氨基转换酶**'，'**尿酸**'，'**年龄**'，'**性别**'，'**甘油三酯**'，'**红细胞体积分布宽度**'，'**红细胞平均体积**','**红细胞计数**'，'**血小板平均体积**'，'**血红蛋白**'。

对于问题二，通过对智能群优化算法的比较和选取，我们团队在问题一的模型基础上引入了**智能群优化算法**中的**粒子群算法**来针对数据集进行 XGBoost **算法**参数的调整，并通过与其他预测模型（如‘SVM’、‘BP’、‘KNN’、‘Linear Regression’）进行比较，来验证我们团队模型的合理性。

对于问题三，我们团队根据**聚类分析**将血糖分成了四类，分别为**患糖尿病低风险、患糖尿病中低风险、患糖尿病中高风险**和**患糖尿病高风险**。我们采用了 K-means++**算法**对糖尿病评估模型进行训练，并分别绘制了 K-means++的**肘部图**和 K 值的**轮廓分析图**进行分析：血糖 3.07-5.4 为患糖尿病低风险，5.04-7.04 为患糖尿病中低风险，7.05-10.45 为患糖尿病中高风险，10.45 以上为患糖尿病高风险，与目前医学对糖尿病血糖的界定相符。最后，结合第二问的模型，具体说明**体检数据对糖尿病风险评估的影响**。

对于问题四，我们团队用血糖和体检数据的模型来对附件 2 中数据进行血糖预测，用血糖和糖尿病的模型做出评估。最后，计算出附件 2 的结果比例：**正常血糖为 42/141，具有患糖尿病风险为 74/141，糖尿病为 20/141，严重糖尿病为 5/141**。

我们团队将**群智能优化算法与梯度提升算法相结合**，克服了传统算法精度低、抗噪声能力弱和无法解决特征明显的结构化数据的局限性，提高了血糖预测的合理性和精度。同时，我们对**聚类模型的类别数**进行多方面检验，选取最合适的类别数来评估糖尿病风险，使结果符合医学标准。最后，我们建立了体检数据和糖尿病风险评估的关系，并利用血糖预测模型和评估模型进行双向构建，通过**集成模型**使结果更加符合医学实际。

关键词：XGBoost 算法，PSO 算法，智能群优化算法，Spearman，K-means++算法，CART 回归树，糖尿病预测，高斯混合模型

目录

模型的假设，建立和求解

一．问题重述

1.1 问题的背景

糖尿病是一种代谢性疾病，其特征是患者的血糖长期高于标准值。胰腺无法产生足够的胰岛素或人体无法有效利用所产生的胰岛素时，就会出现糖尿病。糖尿病的临床表现包括频尿、口渴和饥饿感。同时伴随并发症如心血管疾病、中风、慢性肾脏病和足部溃疡等。根据 2021 年 IDF 发布的数据，全球成年糖尿病患者人数达到 5.37 亿（10.5%），约十分之一的成年人受到影响。在过去的 10 年间，中国糖尿病患者人数增加了 56%，其中约 7283 万名患者尚未被确诊，比例高达 51.7%。糖尿病种类主要分为 1 型糖尿病、2 型糖尿病、妊娠糖尿病和其他类型糖尿病。作为一种常见的慢性疾病，糖尿病目前无法根治，需要通过科学有效的干预、预防和治疗，来降低发病率和提高患者的生活质量。

1.2 问题的要求

附件 1 和 2 分别给出了有血糖值的检测数据和无血糖值的检测数据，包含年龄、性别、各项体检数据等 42 个监测指标，包含数值型、字符型、日期型等数据类型。题目需要我们解决以下四个问题：

问题 1：根据附件 1 的检测数据，从 42 个检测指标中筛选出主要变量指标，并说明筛选过程及其合理性。

问题 2：根据附件 1 的检测数据，建立血糖值的预测模型。

问题 3：根据附件 1 的检测数据，对糖尿病的风险进行评估。

问题 4：根据附件 2 的检测数据，对血糖值进行预测和评估糖尿病风险。

二．问题分析

2.1 第一问的分析

首先，要对于数据集进行预处理，对于其中异常值、缺少值的数据进行删除或填充。然后，考虑不同生理指标和血糖值的相关性，并对其重要性进行排序。最后，筛选出相关性强的生理指标作为特征值。

2.2 第二问的分析

在第一问得出的特征值，对数据预处理后的数据集来建立特征值和血糖的关系。通常可以采取：支持向量机回归、决策树回归、随机森林回归、逻辑回归等多种回归模型来拟合特征值和血糖的关系。

2.3 第三问的分析

根据题目文本提供的信息，正常血糖值是指人在空腹状态下血糖值在 3.9～6.1 毫摩尔/升之间。然而，要判断是否存在高血糖，一般需要对人体进行两次重复测量，若两次测量的平均值大于 6.7 毫摩尔/升，则可以诊断为糖尿病。针对不同程度的血糖值，我们团队可以制定一个简单的分类标准，并利用聚类的思想，找出不同程度的血糖值对应患糖尿病的概率风险。此外，我们还可以通过分析血糖和体检数据之间的关系，评估体检数据对患糖尿病风险的影响。

2.4 第四问的分析

在前三问的基础上，可建立出血糖的预测模型和血糖对应患糖尿病概率的评估模型，只需要将附件 2 的数据集利用两种模型来处理，即可得出结论。

三．数据的预处理

数据的预处理在数据分析和机器学习任务中具有重要意义。作为数据处理流程中的关键步骤之一，它对于获取准确、可靠的分析结果和模型预测起着至关重要的

作用。预处理可以提高数据质量，减少噪声，提取关键特征，归一化数据，并降低数据维度等。这些操作为后续的分析和建模提供了更可靠的数据基础，确保了最终结果的准确性和可解释性。因此，在进行数据分析和机器学习任务之前，充分重视数据的预处理是必不可少的。

3.1 数据的概述

我们小组对附件 1 中的数据进行了详细分析，包括每个生理指标的数量、平均值、偏差、最小值、25%、中位数、75%、最大值以及缺失率等指标。

	count	mean	std	min	50%	max	Missing Rate
*r-谷氨酰基转换酶	4671	38.82159	40.45764	6.36	26.19	736.99	0.208975
*丙氨酸氨基转换酶	4671	27.70333	22.54064	0.12	21.53	498.89	0.208975
*天门冬氨酸氨基转换酶	4671	26.85538	13.496	10.04	23.95	434.95	0.208975
*总蛋白	4671	76.78613	4.038641	57.32	76.63	100.41	0.208975
*球蛋白	4671	30.97013	3.578211	7.06	30.8	66.18	0.208975

表 1.部分生理指标（详见附录）

为了更加直观放映出各项生理指标的缺失率，我们小组将附件 1 中的数据进行了具体的可视化，使得缺失率的程度更加直观明显。

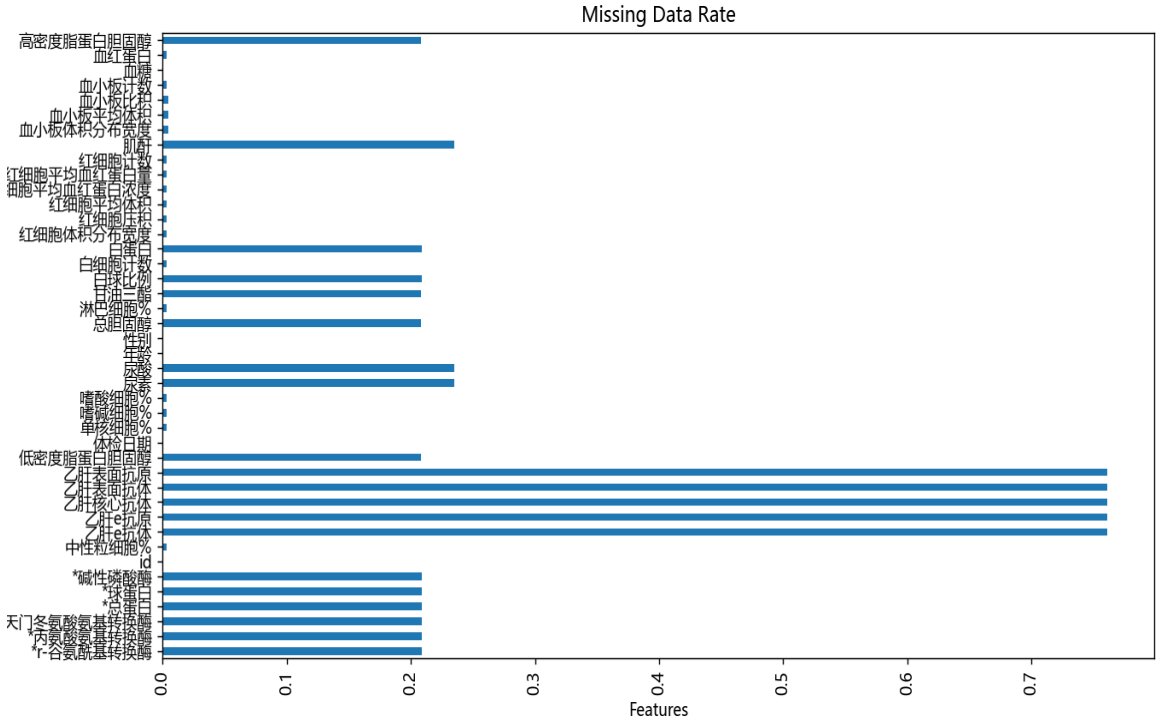


图 1.各项生理指标的缺失率

3.2 数据的缺失值处理

通过对缺失率的柱状图分析，酶、蛋白、醇的生理指标的缺失率在 20%左右，而乙肝相关的指标缺失在 76%左右，这需要我们对其进行筛选。

我们小组从附件 1 中提取出乙肝相关的指标存在的数据，并且利用 Spearman Correlation Heatmap 来放映乙肝相关的指标和血糖的相关性。

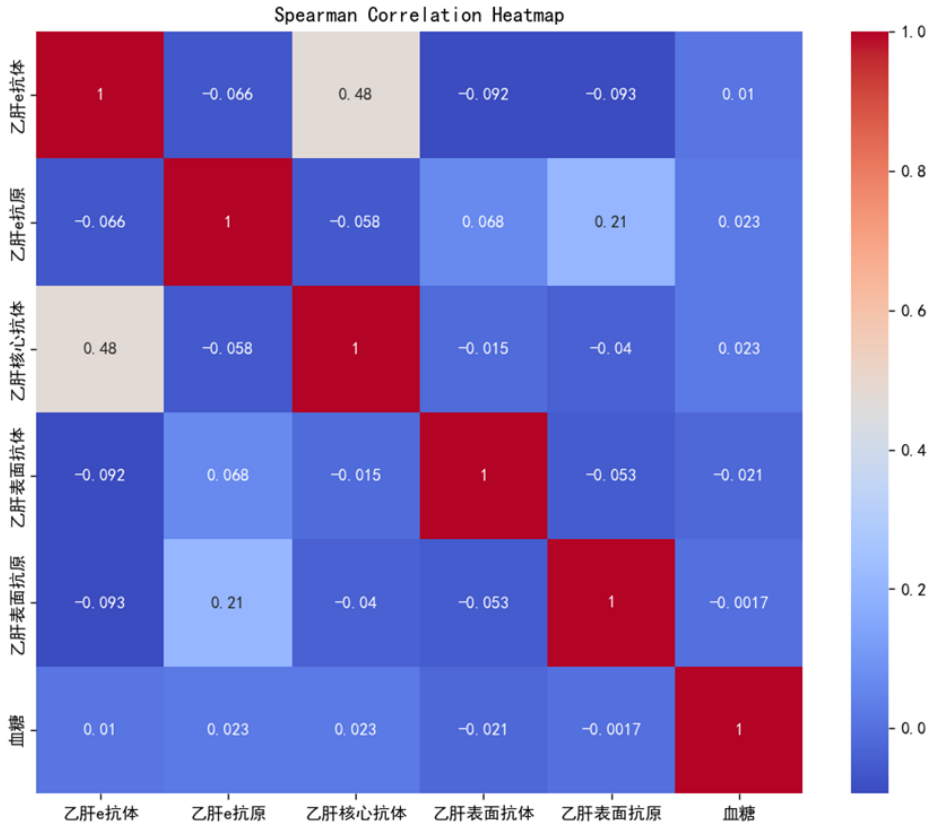


图 2. 乙肝相关指标和血糖的 SCH 图

根据乙肝相关指标和血糖的 SCH 图，我们可以分析这些指标和血糖的相关系数并不是很高，同时模型不可以通过 24%的数据来衡量疾病的基础信息。所以我们团队决定舍弃这五项指标。

同时针对着缺失率在 20%左右的生理指标，我们团队采取同样的方法得出了 Spearman Correlation Heatmap（详见附录）。

图 3. 缺失率 20%的指标和血糖的 SCH 部分图

针对相关性不是很高的*总蛋白、*球蛋白、白球比例、白蛋白和高密度脂蛋白胆固醇，我们团队决定舍去这五项指标。除了舍弃上述 10 项生理指标之外，还需要对于冗余数据的处理。体检日期和 id 与血糖值没有直接的关联，所以这两项数据也需要舍去。

通过上一步对于缺失率过大的特征值进行筛选后，我们需要对于缺失值进行填充。在查询相关文献后，填充缺失值的方法大致有这几种：特殊值填充、平均值填充、中位数填充、K 最近距离邻法(K-means clustering)、回归方程法(Regression)、期望值最大化法（Expectation maximization）等。在考虑到该数据集中存在着部分异常值，有可能存在糖尿病人的某项生理指标远超于正常范围，我们团队决定从血糖正常的数据中提取出各项生理指标的中间值。既能避免血糖异常患者的生理指标过高，又能使得填充的数据更为合理。

特征	补充值	特征	补充值	特征	补充值	特征	补充值
*r-谷氨酰基转换酶	36.39	低密度脂蛋白胆固醇	3.30	尿酸	356.26	红细胞体积分布宽度	12.77
丙氨酸氨基转换酶	26.55	单核细胞%	6.85	年龄	44.08	红细胞压积	0.44
*天门冬氨酸氨基转换酶	26.24	嗜碱细胞%	0.60	总胆固醇	5.17	红细胞平均体积	89.03
*碱性磷酸酶	85.98	嗜酸细胞%	2.02	淋巴细胞%	33.93	红细胞平均血红蛋白浓度	334.76
中性粒细胞%	56.59	尿素	4.91	甘油三酯	1.69	红细胞平均血红蛋白量	29.82
红细胞计数	4.93	血小板体积分布宽度	13.26	血小板比积	0.27	血糖	5.18
肌酐	77.67	血小板平均体积	10.65	血小板计数	255.76	血红蛋白	146.86

表 2.填补缺失值的数据

3.3 数据的异常值处理

该数据中还存在许多异常的生理指标，这些离散的异常指标可能会对后续的模型产生影响。如果数据集的基础特征的分布与推理条件中的先验高斯分布存在较大的差异，或者数据集中的特征不平滑，这会对模型的拟合产生负面影响。同时，一些噪声数据会影响拟合函数的准确性，因此需要设定一定的阈值来筛选离散的异常指标。

我们团队采用了箱型图和高斯混合模型来处理数据集中的异常值。然而，考虑到箱型图的筛选范围较广，可能会删除大量的异常数据。这不仅导致数据集大幅减少，影响后期模型的拟合精度，而且也不符合糖尿病患者应有的高数据指标。相反，利用高斯混合模型，我们可以根据情况设定相应的阈值来控制异常数据的筛选，从而使数据集更加集中，更易于模型拟合。

除上述之外，针对性别这类特征值对于血糖的影响，根据查阅统计资料[1]，对于中年糖尿病患者，多项研究显示男性 2 型糖尿病患病率高于女性。所以，我们团队利用 1 和 0 来分别给男女性别的差异赋值。

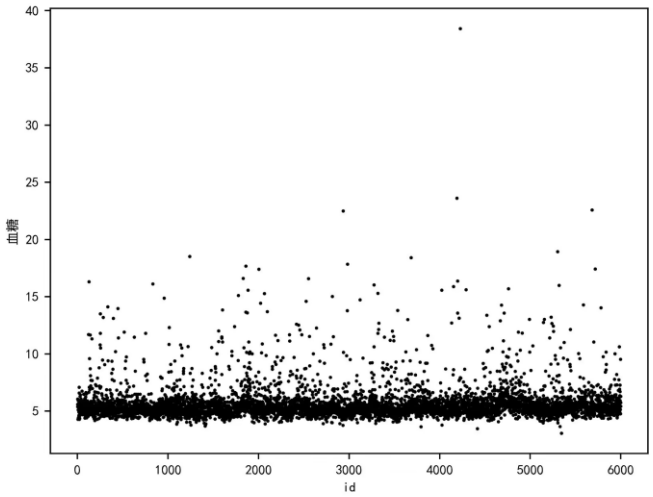


图 4.血糖值的离散程度

四．模型的假设、建立和求解

4.1 第一问的模型

在进行数据预处理后，我们团队得到与血糖值相关性较强的几组生理指标：*r-谷氨酰基转换酶，低密度脂蛋白胆固醇，尿酸，红细胞体积分布宽度，*丙氨酸氨基

转换酶，单核细胞%，年龄，红细胞压积，*天门冬氨酸氨基转换酶，嗜碱细胞%，总胆固醇，红细胞平均体积，*碱性磷酸酶，嗜酸细胞%，淋巴细胞%，红细胞平均血红蛋白浓度，中性粒细胞%　　，尿素，甘油三酯，红细胞平均血红蛋白量，红细胞计数，血小板体积分布宽度，血小板比积，血糖，肌酐，血小板平均体积，血小板计数，血红蛋白，性别。

为了进一步筛选出更为重要的特征值，我们团队打算建立机器学习中决策树模型，利用 XGBoost 算法进行特征值的进一步筛选。

假设已经训练了 K 颗树（即训练了 K 个数据指标），对第 i 个数据的最终预测值为：

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$$

目标函数等于损失函数加控制复杂度：

$$obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

进行叠加式训练：

$$\begin{aligned} \hat{y}_i^{\langle k \rangle} &= f_1(x_i) + f_2(x_i) + f_3(x_i) + \ldots + f_k(x_i) \\ &= \sum_{j=1}^{K-1} f_j(x_i) + f_k(x_i) \\ &= \hat{y}_i^{\langle k-1 \rangle} + f_k(x_i) \end{aligned}$$

目标函数：

$$Minimize: \sum_{i=1}^{n} l(y_i, \hat{y}_i^{\langle k-1 \rangle} + f_k(x_i)) + \Omega(f_K)$$

对目标函数进行泰勒级数逼近：

$$minimize: \sum_{i=1}^{n} [g_i \cdot f_k(x_i) + \frac{1}{2} h_i \cdot f_k^2(x_i)] + \Omega(f_K)$$

接下来我们只需要优化这个目标函数，只需要将其中进行参数化。

符号	说明
$\hat{y}_i$	第i个样本x_i的预测值
y_i	第i个样本x_i的真实值
$f_k()$	第K棵决策树对x_i预测函数
Obj	优化模型的目标函数
$\hat{y}_i^{<k>}$	K个决策树对第i个样本x_i的预测值和
$l(y_i, \hat{y}_i)$	真实值y_i与预测值$\hat{y}_i$的损失函数
g_i, h_i	$l(y_i, \hat{y}_i)$的一阶导数,二阶导函数
$\Omega(f_k)$	K 棵树的复杂度函数

表 3.第一问模型的符号说明

4.2 第二问的模型

近年来,研究者应用逻辑回归(logistic regression, LR)、支持向量机(support vector machine, SVM)、随机森林(random forest, RF)、梯度提升树(gradient boosting decision tree, GBDT)、神经网络(artificial neural network, ANN)等多种机器学习算法，研究构建糖尿病预测模型。其中裴修侗等通过实验表明，SVM 传统算法模型精度较低，利用布谷鸟、灰狼仿生算法优化后精度都有所提升。郭奕瑞等利用 ANN 和 LR 算法建立 2 型糖尿病预测模型，得出 ANN 模型较 LR 模型具有更好的预测效果。余丽玲等融合了自回归积分滑动平均(autoregressive integrated moving average model, ARIMA)和 SVM 两种模型，并将融合方法与 ARIMA 模型、SVM 模型、ANN 模型进行对比，实验表明组合模型的预测精度相比单一预测模型有明显提高。[5]

这些不同类型的集成模型在糖尿病预测方法研究中注入了新鲜血液，使得该项研究有了新的进展。尽管深度学习近些年变得热门起来，但它多用于捕获图像、语音和文本等高维数据，并不适合处理特征明显的结构化数据。极致梯度提升树 XGBoost 算法(Extreme Gradient Boosting) 是适应大规模数据的分布式并行算法，该算法以其高效灵活、准确性高、可移植性强等特点在近几年的 Kaggle 数据挖掘比赛中脱颖而出。[6]

所以我组在第一问的模型上进行正则化参数的优化，能够有效控制模型过拟合。

目标函数的正则化方程：

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

将属于第 j 个叶子节点的所有样本x_i,划分到一个叶子节点的集合I_j中：

$$I_j=\{i|q(x_i) = j\}$$

样本x_i落在第$q(x_i)$叶子节点上，便可以得出$f_t(x_i)$与$w_{q(x_i)}$值相等。

综合第一问模型和第二问正则化参数的具体优化：

$$\mathrm{Obj}(\theta) \approx \sum_{i=1}^{n}\left[g_i\, f_t(x_i) + \frac{1}{2}h_i\, f_t^2(x_i)\right] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 =$$

$$\sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right) w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right) w_j^2\right] + \gamma T.$$

由于$\sum_{i\in I_j} g_i$和$\frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2$是已知的部分，我们记做$G_j$和$H_j$,参数也就是$w_j$。目标函数也就变成了对$w_j$的二次函数最优解问题。我们对其求导，

得到当$w_j = -\frac{G_j}{H_j+\lambda}$取得最小值。

符号	说明
T，t	决策树节点数，决策树的数量
γ, λ	节点数与节点权重相对于复杂度权重
I_j	第 j 个节点的所有样本x_i,划分到一个节点的集合
w	叶子节点的值函数
$q(x_i)$	样本x_i落在的叶子位置
G_j，H_j	$\sum_{i\in I_j} g_i$，$\frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2$

表 4.第二问模型的符号说明

4.3 第三问的模型

我们团队结合题目文本的糖尿病的血糖评估标准和医学文献[1, 2, 3, 4]的血糖标准划分，决定将数据集中的血糖值按照四种类型划分：患糖尿病低风险、患糖尿病中低风险、患糖尿病中高风险和患糖尿病高风险，分别以 0，1，2，3 来代表。

患糖尿病低风险：表示一个人患病的可能性非常小，一般不需要进一步检查和治疗。例如，没有明显家族史、生活方式健康的人可以被认为是低风险人群。

患糖尿病中低风险：表示一个人患病的可能性较小，但仍需要进一步检查和治疗以防止疾病风险进一步增加。例如，存在一些与疾病相关的危险因素，但是这些因素的影响相对较小，例如体重稍微超标或者血糖略高。

患糖尿病中高风险：表示一个人患病的可能性较大，需要进行更为详细的检查以及加强饮食、运动等多方面的预防措施。例如，存在一些较大的与疾病相关的危险因素，例如高胆固醇水平或者高血压等。

患糖尿病高风险： 表示一个人已经处于患某种疾病的高风险状态中，需要立即采取措施进行治疗和管理。例如，存在明显的家族史或者其他严重的危险因素，例如已经被诊断为糖尿病或者心脏病等。

同时针对数据集中部分极端异常的数据，我们也考虑用默认数据来处理这一类噪声点。对于是否患糖尿病的评估，就以聚类过程中分到不同区域来作为风险值的评估。

通过将血糖数据聚类到不同风险的糖尿病区域，我们可以评估高血糖患糖尿病的风险。然而，根据题目要求，我们需要利用体检数据进行糖尿病的风险评估。这实际上要求我们建立体检数据与糖尿病风险之间的关系，包括特征值与血糖的关系以及血糖与患糖尿病风险之间的关联，以进行糖尿病的风险评估。

下面是聚类分析的数学模型：

假设我们有 n 个样本，每个样本具有 d 维特征。我们要将这些样本分为 k 个不同的聚类。

1.初始化：选择 k 个初始质心，可以是随机选择或根据某种启发式方法选择。

2.分配样本：对于每个样本，计算其与 k 个质心之间的距离，将样本分配给距离最近的质心所代表的聚类。

3.更新质心：对于每个聚类，计算属于该聚类的所有样本的均值，将均值作为新的质心。

4.重复步骤 2 和 3，直到满足停止条件，例如达到最大迭代次数或质心变化不大。

4.4 第四问的模型

第四问的模型只需要结合第二问和第三问的模型，对附件 2 中数据集进行预测和评估即可。

算法的设计和实现

一．算法的设计和实现

1.1 第一问算法的设计和实现

XGBoost 算法是一种梯度提升算法，具有出色的预测性能。通过使用 XGBoost 模型进行特征选择，可以利用其强大的学习能力和优化算法，准确评估每个特征的重要性，并选择对目标变量有最大贡献的特征。同时，它还提供特征重要性排名，有助于解释模型的预测结果。通过分析特征重要性，可以理解哪些特征对目标变量的预测起到关键作用，从而提供洞察和解释模型的决策过程。

XGBoost 算法支持两种分裂节点的方法——贪心算法和近似算法。其思想核心是：观察分裂后的收益，我们会发现节点划分不一定会使得结果变好，因为我们有一个引入新叶子的惩罚项，也就是说引入的分割带来的增益如果小于一个阀值的时候，我们可以剪掉这个分割。

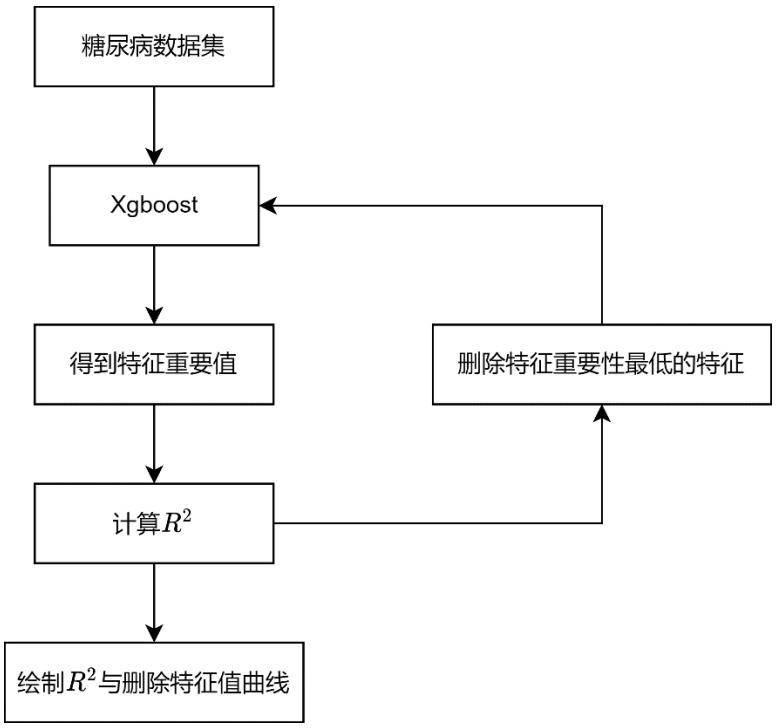


图 5.第一问算法的设计过程

1.2 第二问算法的设计和实现

在参考第一问的模型之后，我们团队加入正则化处理，具体的算法流程图如下：

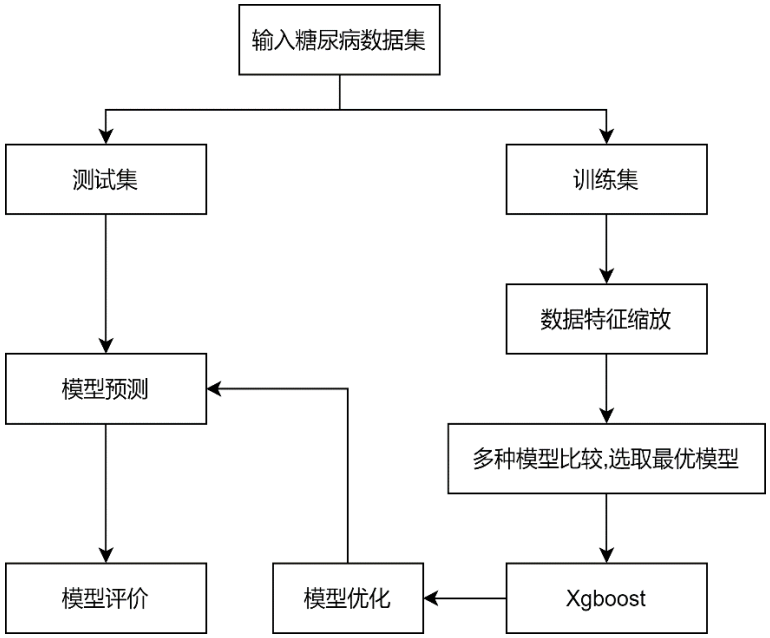


图 6.第二问算法的设计过程

在建立好模型后，最关键的步骤就是对模型进行参数的调整。我们团队根据网络上已有的文献[6]，利用已调好的参数进行测试，发现其拟合结果与文献中并不太一致。其原因在于我们团队的特征值进行过多次筛选，测试所用的特征值数据也少于文献之中的。

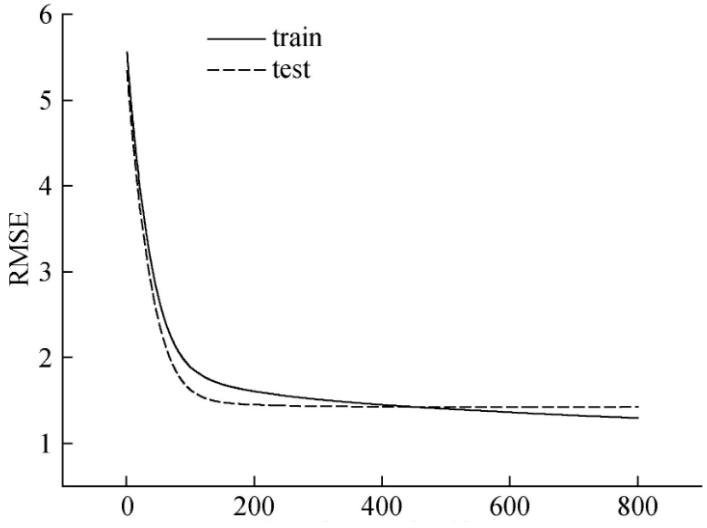


图 7.参考文献中的 RMSE 效果[6, 9]

为了找出适用于我们团队特征值和数据集的 XGBoost 的具体参数,我们团队利用群智能优化算法来进行参数的优化调整。

常用的群智能优化算法有遗传算法[8]、粒子群算法、麻雀搜索算法。这些算法将生物的抽象行为转化为寻优的过程，并且个体的异常不会对整个群体的行为进行影响，具有强大的鲁棒性。我们团队在通过对比三种群智能优化算法的 RMSE 值与迭代次数的关系，发现 PSO(粒子群算法)比其他两种更优。

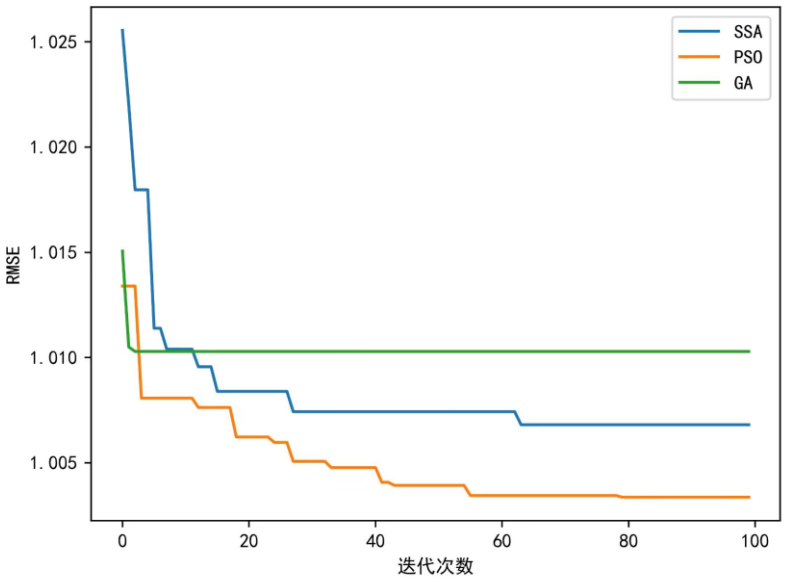


图 8. 群智能优化算法的比较

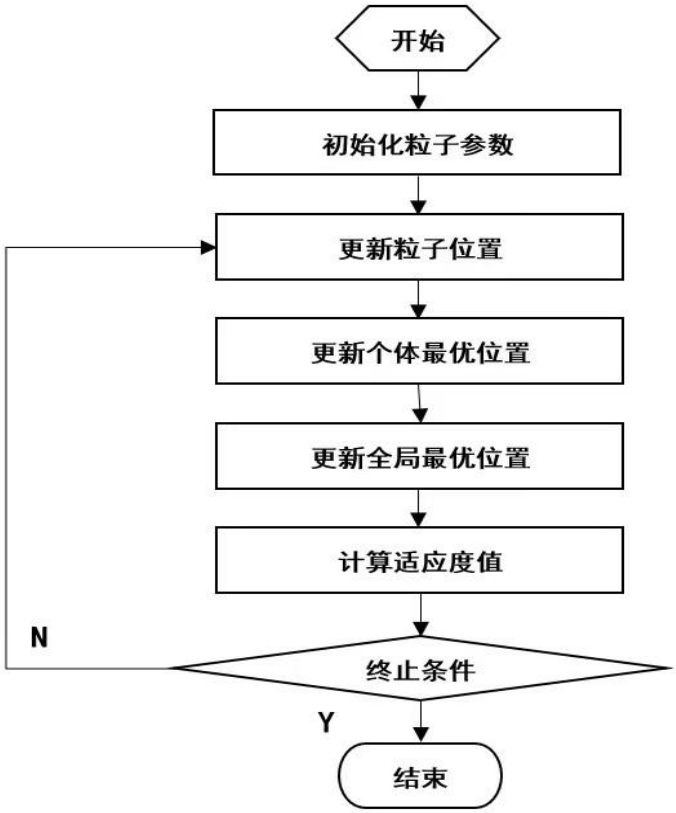


图 9. 粒子群算法（PSO）的实现过程

PSO 启动过程中，需要初始化各项参数，并设置算法停止条件；根据参数来计算每个粒子的适应度值，采取迭代的方法不断优化全局最优以及个体最优适应度。这

就为我们团队的 XGBoost 模型的参数优化起到了很大的作用。

1.3 第三问算法的设计和实现

考虑到题目文本的糖尿病的血糖评估标准和医学文献[1,2,3,4,5]的血糖标准划分，我们团队决定将数据集中的血糖按照四种类型划分：患糖尿病低风险、患糖尿病中低风险、患糖尿病中高风险和患糖尿病高风险，分别以 0，1，2，3 来代表。在设计聚类过程中，选择合适聚类算法是最重要的一步。通过查找资料[7]，聚类算法可分为很多种：划分式聚类算法、基于密度的划分算法、层次化聚类算法、新式方法。

若采取 DBSCAN 这种基于密度聚类算法，它能够将不同的血糖值聚类成四种任意形状大小的区域，同时它的抗噪声性能十分强大，它能够将噪声大的点分类到-1。若采取 agglomerative 这种层次划分算法，能够将不同的血糖值聚类成四种任意形状大小的区域，但是它对于噪声和异常值过于敏感，生成复杂的树状结构，其中包含多个层次和聚类簇。对于大规模数据集，这些结构可能很难解释和解读，特别是当聚类层次非常深或复杂时。

由于我们的血糖的聚类形状应该符合线性划分，同时对于那些噪声过高的数据，我们团队在数据预处理的过程中已经去除，剩下的一些异常的血糖值是符合现实中重症糖尿病人的情况，不应该当作噪声点去除。

综合考虑上述算法的条件，我们团队决定采用 K—means++算法来聚类。

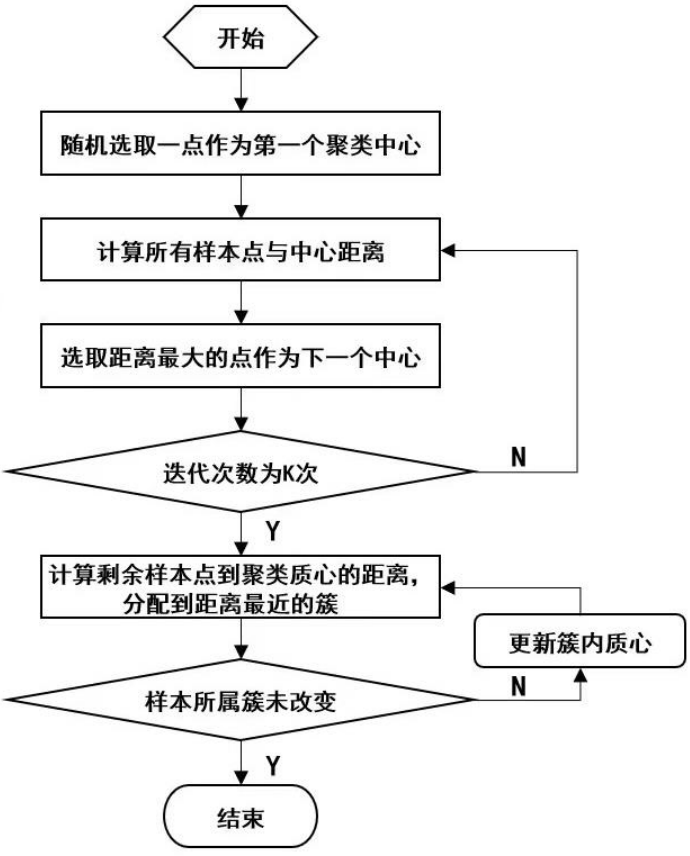


图 10.聚类的算法设计

算法类型	适合的数据类型	抗噪点性能	聚类形状	算法效率
Kmeans	混合型	较差	球形	很高

K-Mean++	混合型	一般	球形	较高
Bi-Kmean++	混合型	一般	球形	较高
DBSCAN	数值型	较好	任意形状	一般
OPTICS	数值型	较好	任意形状	一般
Agglomerativa	混合型	较好	任意形状	较差

表 3.数据聚类方法

1.4 第四问算法的设计和实现

第四问只需要结合第二问和第三问的算法，将附件 2 的数据进行回归预测和分类即可。

结果的分析和检验

一．第一问结果的分析和检验

我们团队通过依次删去特征值，再利用 XGBoost 算法来刻画出R^2。

R^2是一种用于评估回归模型性能的统计指标，也被称为决定系数（coefficient of determination）。它的取值范围在 0 到 1 之间，越接近 1 表示模型对目标变量的解释能力越强，而越接近 0 表示模型解释能力较弱。

通过观察删去特征值数和R^2的图像关系，我们发现在删去特征值数目在 0 到 15 之间，R^2大约在 0.7 左右，说明此时的模型对血糖的解释能力较强；删去特征值数目在 20 到 25 之间是，R^2有一个十分明显的下降幅度，模型的解释能力也在快速下降。所以，我们需要在 15 到 20 的范围之间选取一个最为合理的删去特征数目。

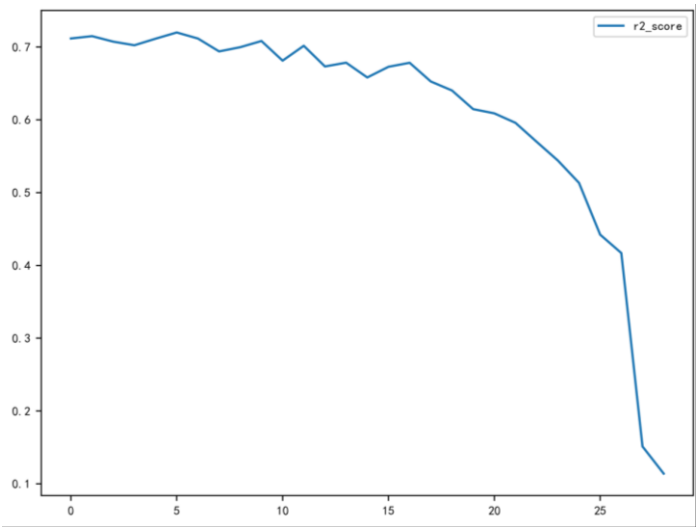


图 11.删去特征值数和R^2的关系

当然，上述图像能够对我们删去特征值的数量指出一个清晰的范围，但是具体的特征值选取还是需要我们衡量特征值的重要度。而我们团队利用 XGBoost 算法，采用 CART 回归树作为基学习器，得出各类特征值的重要程度排行和其 Spearman 系数图像。

通过图像，年龄对于影响血糖的重要程度是最高的。在综合考虑特征值重要性排行和 Spearman 系数的图像，我们选取了十个最为重要的特征值：’*天门冬氨酸氨基转换酶’，’尿酸’，’年龄’，’性别’，’甘油三酯’，’红细胞体积分布宽度’，’红细胞平均体积’,’红细胞计数’，’血小板平均体积’，’血红蛋白’。这十项特征值在重要程度排行之中大于 0.04，在 Spearman 系数图中绝对值大于 0.1 或接近 0.1。

同时在参考部分医学文献[1, 2, 3, 4, 5]时，我们发现上述十项特征值指标在判断病人是否患有糖尿病及其并发症的过程起到一定性作用。

综上所述，我们团队从排除 Spearman 系数过低的乙肝指标，到通过 XGBoost 算法来进一步筛选和强化指标，最后得出了较为合理的十项特征值指标。

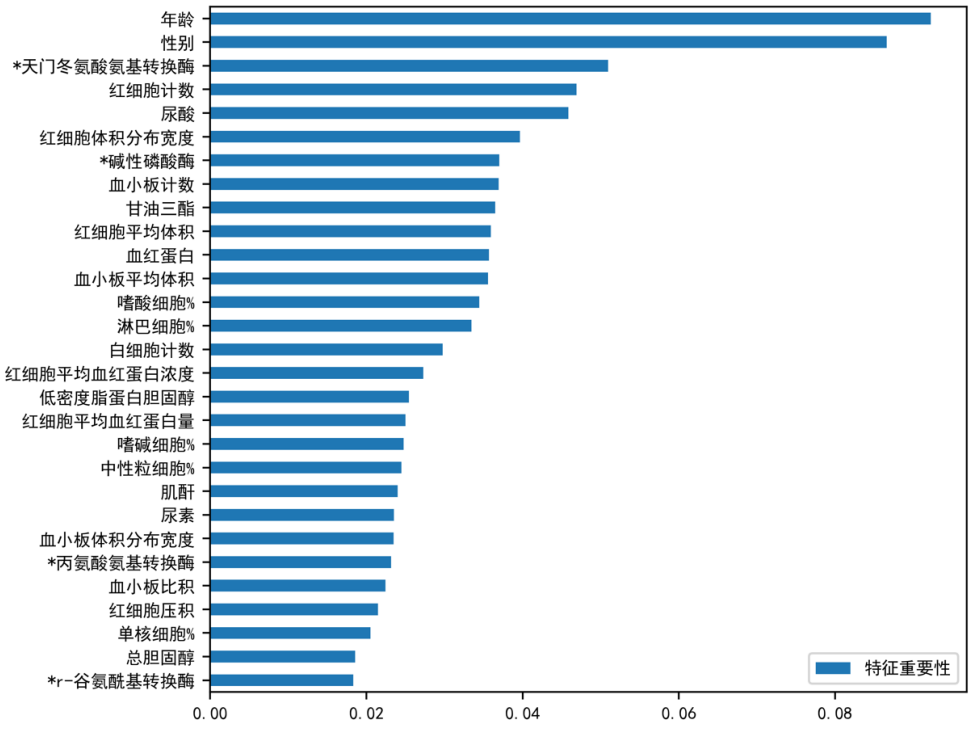


图 12. 特征值重要性排行

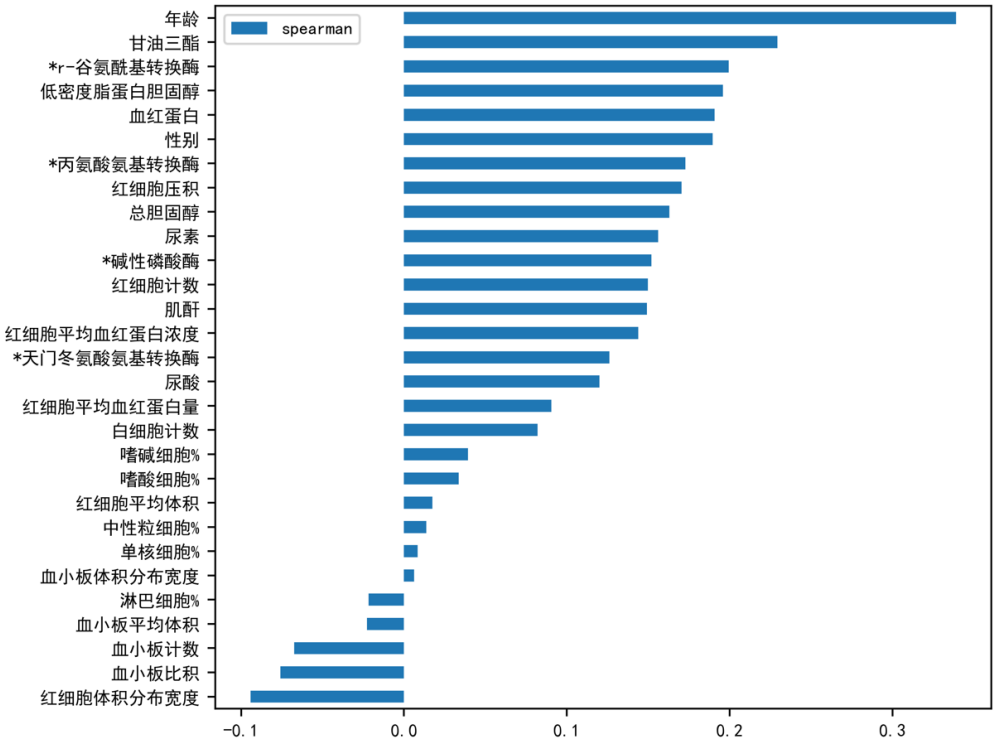


图 13. 特征值的 Spearman 系数

二. 第二问结果的分析和检验

通过观察 PSO 算法的优化参数过程图，我们可以很清楚地看到 PSO 算法在降低

Xgboost 的 RMSE 的效果是十分明显。这也说明，我们团队的基于粒子群优化的 XGBoost 算法拟合出来的模型效果比参考文献[6]上已调好的参数的模型更优。

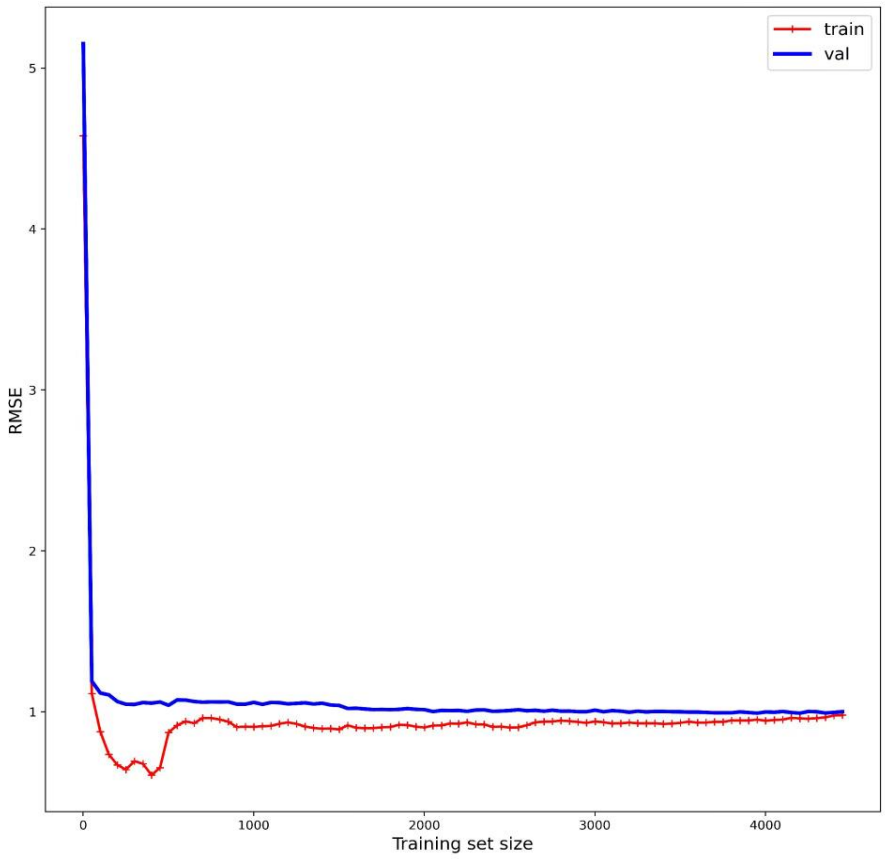


图 14. XGBoost 模型的 RMSE 图

从 XGBoost 模型的 RESM 图中可以看出，模型初期收敛十分快，经过 training set size 的逐步提升，收敛速度明显开始下降，验证集的 RMSE 收敛程度较训练集相比要大。这可能是因为验证集包含了模型在训练过程中未见过的数据，模型在这些数据上的表现相对较差。同时，随着训练集规模的逐步增加，训练集和验证集的 RMSE 快速收敛。这表明模型能够快速学习数据的模式和关联性。

当 training set size 在 500 左右时，测试集 RMSE 曲线趋于平稳，基本不再变化，而训练集 RMSE 仍在缓慢收敛。这可能是因为模型在更多数据上进行训练，仍在逐步提升自己的性能。

当 training set size 达到 4000 之后时，两条曲线之间距离基本保持不变，训练集 RMSE 与测试集 RMSE 十分接近，且两者之间的距离改变趋势基本为 0，这表明模型在这个阶段已经达到了最优效果，并且在更多数据上的训练没有显著改善模型的性能。

随着 training set size 继续增加之后，测试集 RMSE 基本没有变化，训练集 RMSE 依然在以极小的幅度降低。整体来看，模型在训练达到最优效果后呈现平稳状态，没有出现较大的波动，表现出其具有较强的稳定性，这明显说明我们团队建立血糖模型的合理性。

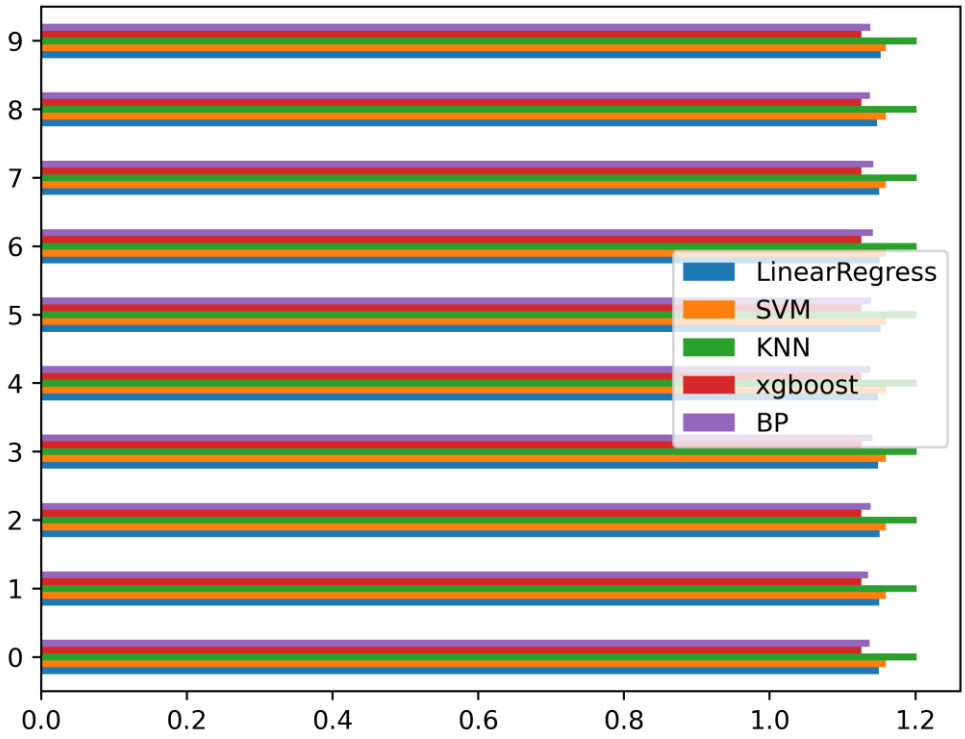


图 15. 不同模型对血糖的预测程度

为了进一步验证我们团队基于粒子群优化参数的 XGBoost 算法模型的合理性和精准性，我们与其他预测模型（如'SVM'、'BP'、'KNN'、'Linear Regression'）相比较。我们发现 XGBoost 算法训练的模型在预测效果方面表现最佳，具有高精度、快速运行和强大稳定性等优势。在糖尿病预测方面，我们的模型展现了可靠性和高效性。同时，针对当前糖尿病预测中数据集维度过高、缺失值较多等特点，传统方法预测精度较低且效率较慢的问题，我们团队构建的基于智能群优化算法的 XGBoost 模型能够自动适应数据集的学习。

三．第三问结果的分析和检验

在正常情况之下，高血糖有较高的风险患糖尿病，为了评估附件 1 的数据集中高血糖患糖尿病的概率，我们团队决定以血糖的数据聚类到患糖尿病不同风险的区域来作为不同血糖患糖尿病的风险评估。同时，题目要求利用体检数据来进行糖尿病的风险评估，这实际上是要求我们通过特征值与血糖的关系、血糖和是否患糖尿病风险的关系来建立出体检数据对糖尿病的风险评估。

首先我们要确定最佳聚类数，为了防止主观确定，我们使用肘部图，由上图可知，随着 k 增长到 4，惯性下降变缓慢，由此可确定聚成 4 类，分别为患糖尿病低风险、患糖尿病中低风险、患糖尿病中高风险和患糖尿病高风险。

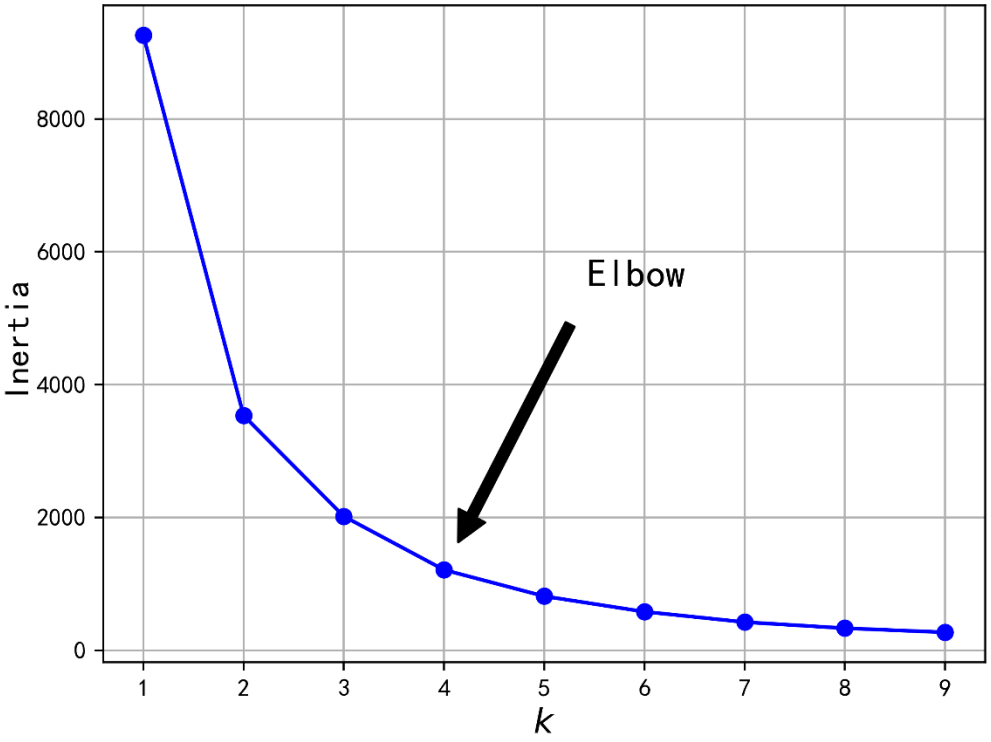


图 16. K-Means++肘部图

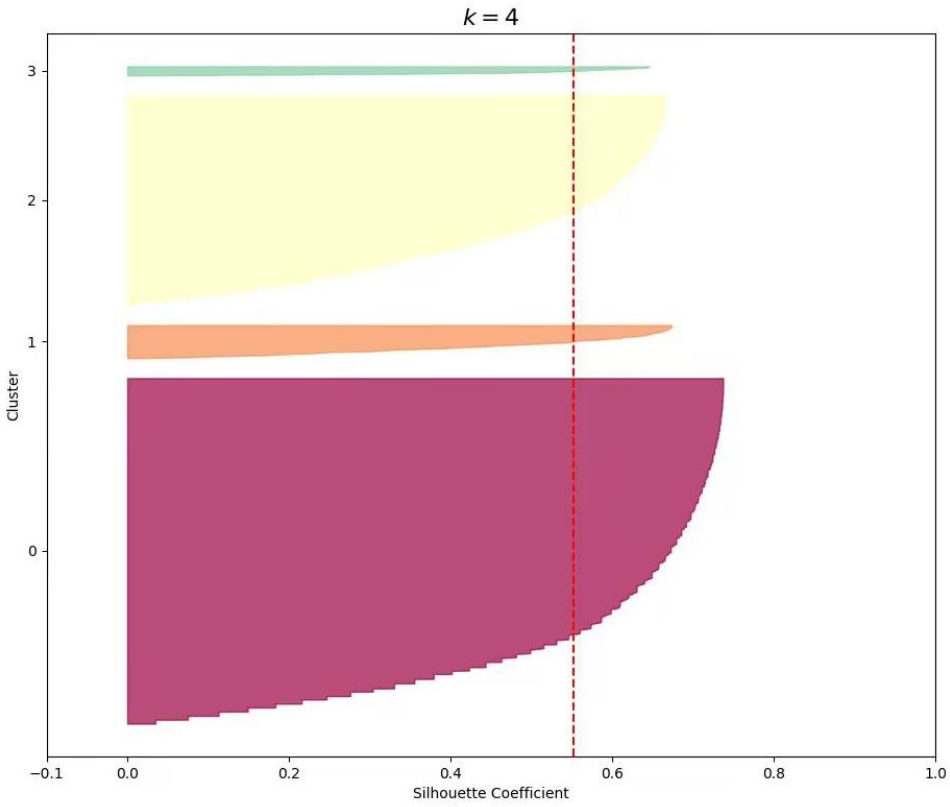


图 17. K 值的轮廓图分析

上图为垂直虚线表示每个集群的轮廓分数。当集群中的大多数实例的系数均低于此分数时，(如果许多实例在虚线附近停止，在其左侧结束)，则该集群比较糟

糕，因为这意味着其实例太接近其他集群了。可以看到，当 k=4 时，集群看起来很好：大多数实例都超出虚线，向右延伸并接近 1.0，说明聚成 4 类是合理的，最终聚类效果如下。

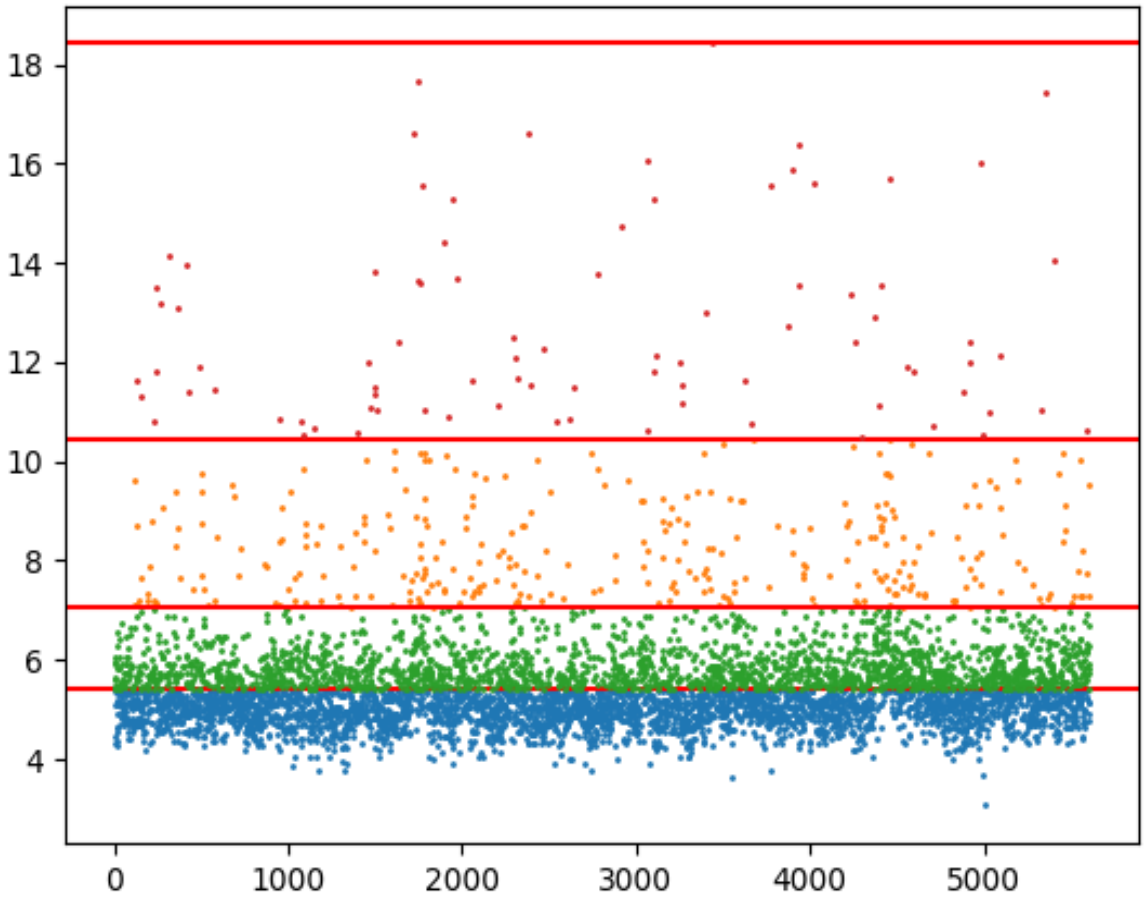


图 18. 聚类效果图

根据聚类效果图，我们可以清晰地发现四种不同地血糖类别：患糖尿病低风险、患糖尿病中低风险、患糖尿病中高风险和患糖尿病高风险。由聚类效果图，我们可以发现血糖 3.07-5.4 为患糖尿病低风险，5.04-7.04 为患糖尿病中低风险，7.05-10.45 为患糖尿病中高风险，10.45 以上为患糖尿病高风险。

由于我们团队使用的数据集是测定的空腹血糖，根据医学资料和专家的建议，空腹血糖在 3.9～6.1 毫摩尔/升是正常人的标准，空腹全血血糖≥6.7 毫摩尔/升时，需要进行 2 次实验来进行诊断，说明这时候已经是中风险，当空腹全血血糖超过 11.1 毫摩尔/升时，表示胰岛素分泌极少或缺乏。因此，空腹血糖显著增高时，不必进行其它检查，即可诊断为糖尿病。我们团队的数据与目前医学对糖尿病血糖的界定相符，说明聚类结果较为准确，具有一定的科学性和参考价值。

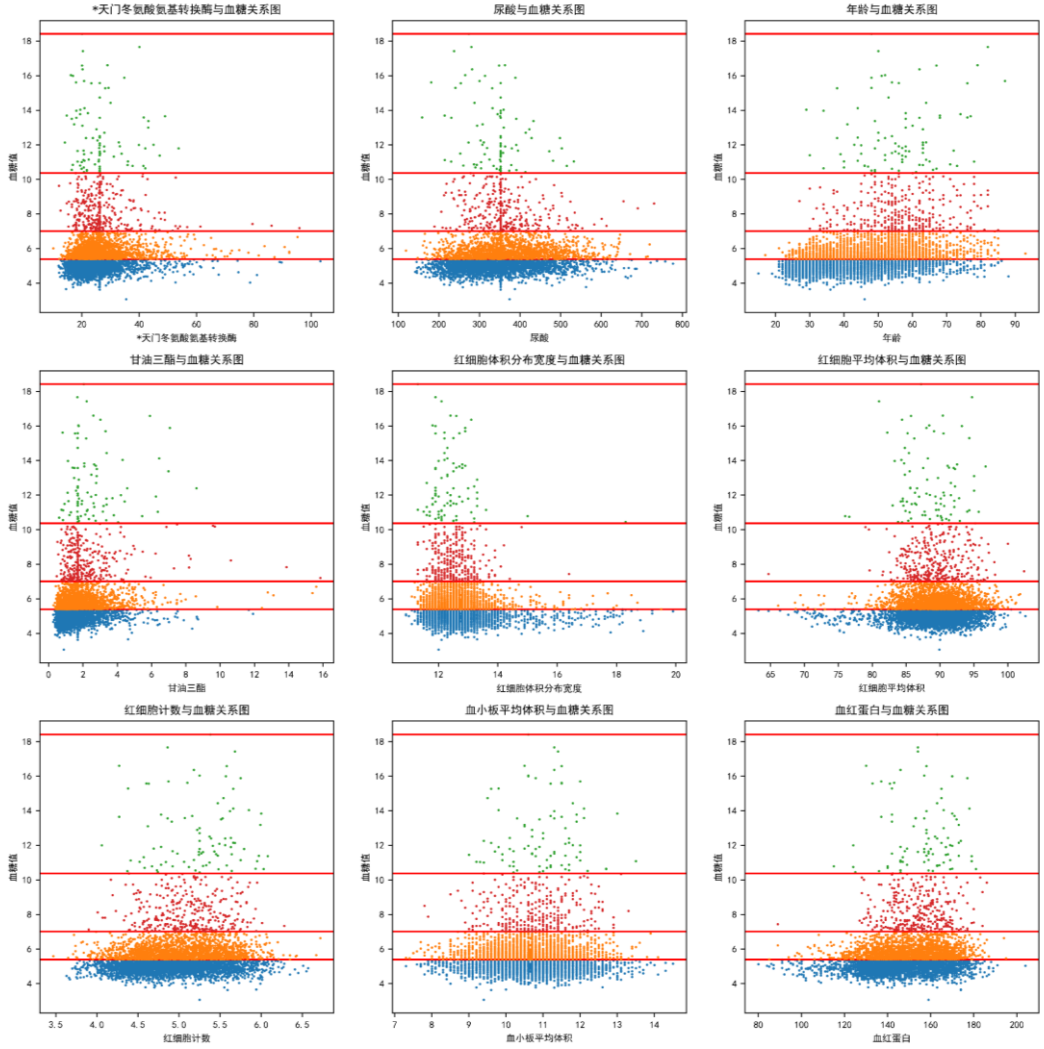


图 19. 体检数据对患糖尿病风险的评估

1. *天门冬氨酸氨基转换酶：在血糖正常的情况下，*天门冬氨酸氨基转换酶含量集中在 20-30 的范围内。随着血糖值的升高，其含量逐渐向高含量处散点化分布。这表明，*天门冬氨酸氨基转换酶的高低与是否患糖尿病的风险有一定关联。根据医学文献[1, 2, 3, 4, 5]的参考资料，*天门冬氨酸氨基转换酶的含量变多往往是由于肝损伤，进而间接性引起血糖变化。因此，*天门冬氨酸氨基转换酶含量大于 40 的人可能是患糖尿病的中低风险。通过综合考虑血糖、*天门冬氨酸氨基转换酶等指标的综合情况，可以更加准确地评估糖尿病的风险水平，为人们提供更好的预防和治疗策略。

2. 尿酸：尿酸的含量与糖尿病的风险水平确实没有直接关联。但是，尿酸的高低与代谢综合征、高血压等多种疾病密切相关。研究表明，尿酸水平升高可能导致胰岛素抵抗，进而增加糖尿病的风险。此外，尿酸水平升高还可能导致内皮功能异常，从而引起心血管疾病和肾脏疾病等并发症。因此，在评估糖尿病的风险时，尿酸水平也应该被综合考虑。

3. 年龄：随着年龄的增加，血糖的含量会逐渐变大，患糖尿病的风险也会逐步增加。这是由于身体机能的下降，导致胰岛素分泌不足，以及胰岛素敏感性降低所致。因此，年龄在 40 岁以上的人需要特别注意控制血糖含量，以预防糖尿病和其它

相关疾病的发生。此外，饮食控制和运动锻炼也是预防糖尿病的重要措施，可以帮助身体更好地利用血糖，降低血糖水平，保持健康的身体状态。

4．甘油三酯：甘油三酯的数值过高，患糖尿病的风险可能越高。您的观点是正确的。甘油三酯是一种血脂，其含量过高会增加患糖尿病的风险。研究表明，甘油三酯的水平升高可能导致胰岛素抵抗，进而增加糖尿病的风险。此外，甘油三酯的水平升高还可能导致心血管疾病等并发症。因此，在评估糖尿病的风险时，甘油三酯水平也应该被综合考虑。

5. 红细胞体积分布宽度：红细胞体积分布宽度的数值越高，患糖尿病的风险可能会更高。红细胞体积分布宽度（RDW）是反映红细胞体积分布的指标，其数值大小与红细胞体积分布的均匀性有关。研究表明，RDW 的数值增加可能与糖尿病的发生和发展相关。当红细胞出现变形或破坏时，其体积分布就会不均匀，从而导致 RDW 的数值升高。这可能是由于糖尿病导致的炎症反应和氧化应激的加剧，进而引起红细胞损伤所致。

6. 红细胞平均体积：红细胞平均体积的含量与糖尿病的风险水平确实没有直接关联。糖尿病患者中，MCV 的水平可能会升高，这可能是由于高血糖导致的红细胞膜的糖基化所致。因此，MCV 的升高可能是糖尿病的一个标志。但是，MCV 的升高并不一定代表患糖尿病的风险更高，因为 MCV 的变化也可能与其他疾病、药物等因素相关。

7. 红细胞计数：红细胞计数和患糖尿病风险无直接关联。研究表明，红细胞计数的变化可能与多种疾病相关，包括糖尿病。在糖尿病患者中，红细胞计数的水平可能会降低，这可能是由于高血糖导致的红细胞寿命缩短所致。因此，红细胞计数的降低可能是糖尿病的一个标志。

8. 血小板平均体积：血小板平均体积和患糖尿病风险无直接关联。研究表明，MPV 的变化可能与多种疾病相关，包括糖尿病。在糖尿病患者中，MPV 的水平可能会升高，这可能是由于高血糖引起的血小板激活所致。高 MPV 与糖尿病的发生和发展密切相关，其水平升高可能会导致糖尿病并发症的发生，如心血管疾病、神经病变等。因此，MPV 可以作为一个糖尿病的生物标志物，但仅凭 MPV 不能确定糖尿病的诊断或风险评估。

9. 血红蛋白：血红蛋白和患糖尿病风险无直接关联。研究表明，血红蛋白的水平与糖尿病的风险并无直接关系。然而，过高或过低的血红蛋白水平可能会提示贫血或失血等情况，这些情况可能与糖尿病并发症的发生有一定的关系。因此，在评估糖尿病的风险时，血红蛋白水平需要与其它指标一起综合考虑，以确定是否存在贫血或失血等因素。

综上论述：*天门冬氨酸氨基转换酶、年龄、甘油三酯、红细胞体积分布宽度这四个体检数据能够对于糖尿病的风险评估提供有力的依据,同时也要综合考虑尿酸、、红细胞体积分布宽度、红细胞平均体积、红细胞计数、血小板平均体积、血红蛋白。

四．第四问结果的分析和检验

根据前 2，3 问建立的模型，我们可以预测附件 2 的血糖并评估糖尿病风险，预测效果见下图。

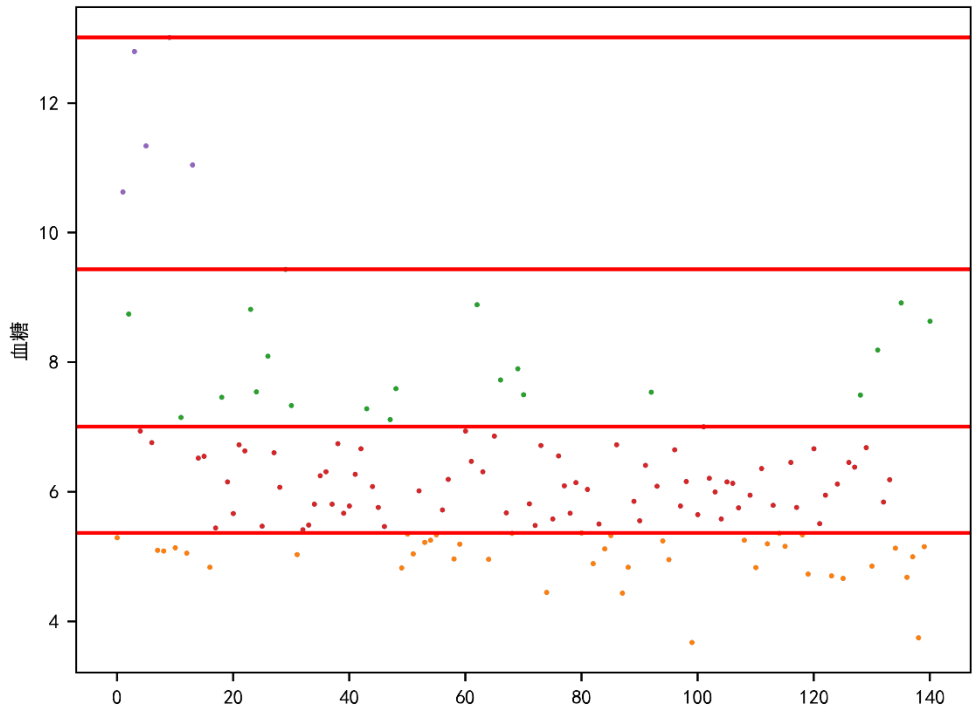


图 20. 附件 2 聚类图

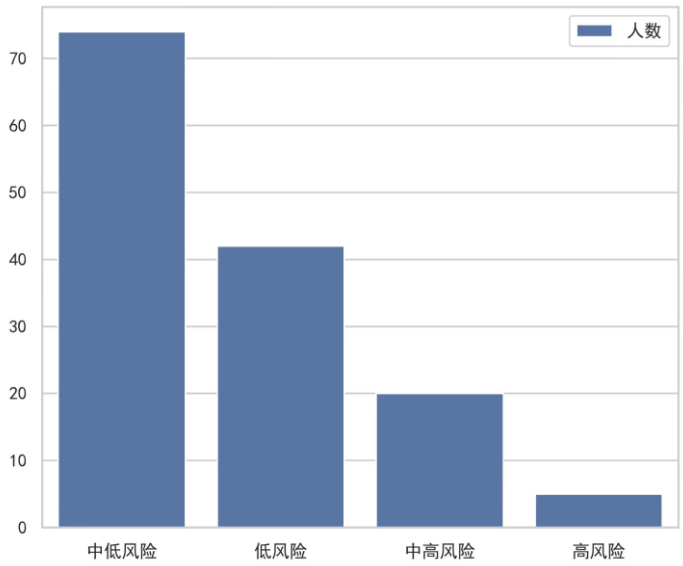


图 21. 附件 2 糖尿病评估图

通过柱状图，我们可以发现：

患糖尿病低风险的人数大约在 40 人左右：表示他们患病的可能性非常小，如果没有明显家族史、生活方式健康的人，一般不需要进一步检查和治疗。

患糖尿病中低风险的人数大约在 70 人左右：表示他们患病的可能性较小，但仍需要进一步检查和治疗以防止疾病风险进一步增加。例如，存在一些与疾病相关的危险因素，但是这些因素的影响相对较小，例如体重稍微超标或者血糖略高。

患糖尿病中高风险的人数大约在 20 人左右：表示他们患病的可能性较大，需要进行更为详细的检查以及加强饮食、运动等多方面的预防措施。例如，去医院对糖

化血红蛋白和餐后血糖进行监测。

患糖尿病高风险的人数大约在 5 人左右： 表示他们已经处于患某种疾病的高风险状态中，需要立即采取措施进行治疗和管理。

模型的优缺点及改进

一. 模型的优缺点

1.1 模型的优点

1.我们模型对缺失值使用中间值进行了填充，对于过于异常值使用了高斯混合模型进行剔除，避免异常值对模型的影响。

2.使用 xgboost 进行递归特征消除，由此进行特征提取，避免维度灾难增加训练和预测的时间和计算成本，降低模型的准确性和可解释性。

3.我们首先选择了预测效果最优的模型进行预测,发现 xgboost 模型效果最好，同时我们使用粒子群算法对传统的 xgboost 进行改进，避免人工调参的耗时耗力，使 xgboost 效果最优。

4.我们使用了 K-Means++算法，该算法大大减少了寻找最优解所需的算法次数，同时增强了算法的鲁棒性与精确性。

1.2 模型的缺点

1.没有进行特征组合，可能会导致部分组合后对血糖预测相关性较强的特征值被舍去，例如乙肝指标。

2.对于部分数据的血糖数据，没有进行平滑化处理，只用了高斯混合模型对其异常值进行排除，可能部分数据存在异常。

二. 模型的改进

1.特征值组合：对于一些缺失过大的特征值数据，我们可以采用特征值组合的方法。特征编码：可以采用不同的编码方式，如二进制编码、序列编码、类别编码等，对原始特征进行编码，生成新的组合特征；特征交叉：在进行特征组合时，可以考虑对不同特征进行交叉，生成新的组合特征。例如，对两个类别特征进行交叉，可以得到新的组合特征，表示这两个特征同时出现的情况；特征扩充：在进行特征组合时，可以考虑对原始特征进行扩充，添加新的特征。例如，对于数值特征，可以通过对其进行对数、平方、倒数等操作，生成新的扩展特征；特征加权：在进行特征组合时，可以对不同特征进行加权，赋予不同的重要性。例如，对于某些特征，可以通过领域专家或机器学习算法确定其权重，进而对特征进行加权组合。

综合考虑以上几种方法的优缺点，根据数据的特点和模型的需求，选择合适的特征组合方法进行改进，以提高模型的性能和泛化能力。[5]

2.平滑处理：平滑化是一种常见的数据预处理方法，用于减少噪声的影响和增强数据的可读性。平滑化的改进方法可以从以下几个方面考虑：加权平均，滑动平均，中位数滤波，基于机器学习的方法，小波变换、卡尔曼滤波等，对数据进行处理，消除噪声并得到平滑后的结果。[5]

参考文献及参考书籍和网站

[1]侣思聪,杨伟,罗鸿宇,马艺欣,赵欢.老年 2 型糖尿病患者衰弱的影响因素分析[J].国际老年医学杂志,2023,44(02):140-144.

[2]梁珊珊,周智华,李成程,陈慧靖,周尚成.1990—2019 年中国糖尿病疾病负担及发病预测分析[J].中国全科医学,2023,26(16):2013-2019.

[3]李惠珍,常可亭,梁欢欢等.2 型糖尿病高危人群糖尿病防治素养现状及影响因素分析[J].河南医学研究,2023,32(03):462-467.

[4]唐丽敏.2 型糖尿病患者血糖管理行为及血糖控制现状的研究[J].婚育与健康,2023,29(05):181-183.

[5]宁莉燕,陈建荣,董建成,苏建彬.基于特征选择和神经网络的糖尿病预测模型研究[J].医学信息学杂志,2023,44(02):47-51.

[6]曲文龙,李一漪,周磊.XGBoost 算法在糖尿病血糖预测中的应用[J].吉林师范大学学报(自然科学版),2019,40(04):118-125.

[7]戴金. 改进 K-MEANS 算法及在Ⅰ型糖尿病血糖值的聚类应用[D].北京交通大学,2011.

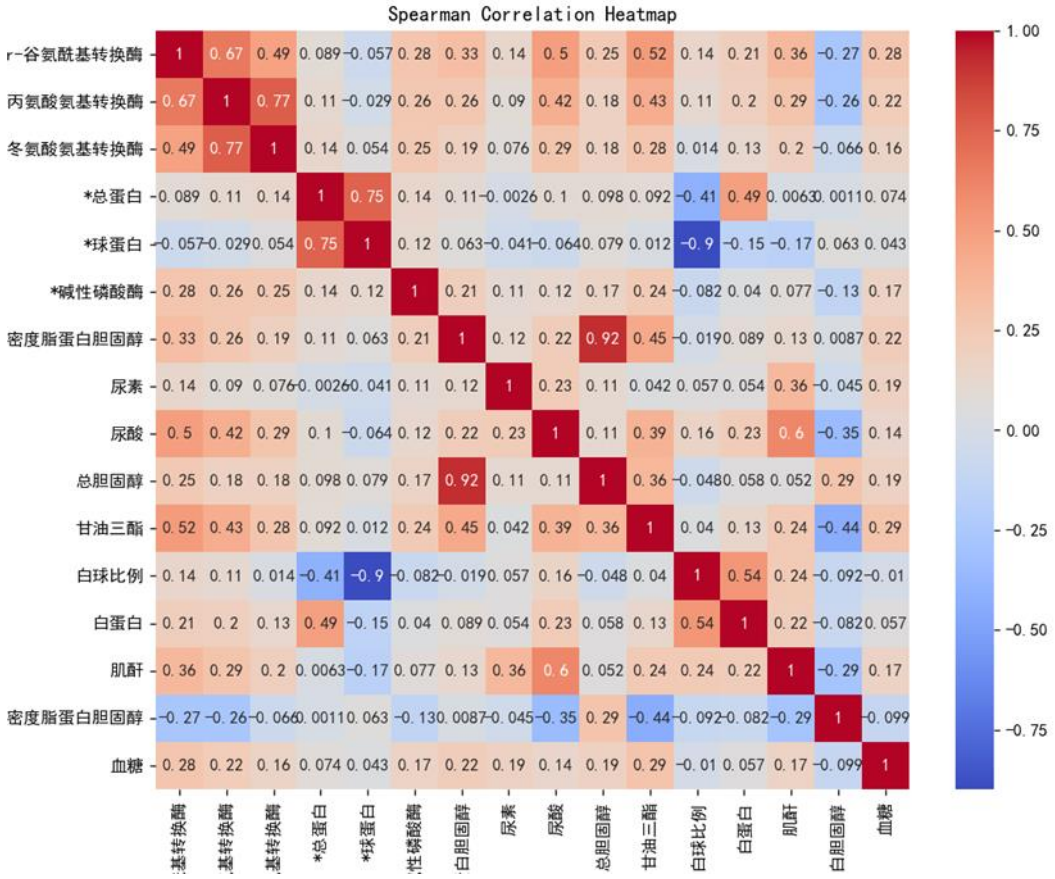
[8]张春富,王松,吴亚东,王勇,张红英.基于 GA_Xgboost 模型的糖尿病风险预测[J].计算机工程,2020,46(03):315-320.

[9]王誉霖. 基于群群智能优化算法和 XGBoost 的血糖预测模型研究[D].北京化工大学,2021.DOI:10.26939/d.cnki.gbhgu.2021.001402.

附件

一. 图片附件

	count	mean	std	min	25%	50%	75%	max	Missing Rate
*r-谷氨酰基转换酶	4671	38.82159	40.45764	6.36	17.81	26.19	44.005	736.99	0.208975445
*丙氨酸氨基转换酶	4671	27.70333	22.54064	0.12	15.17	21.53	32.415	498.89	0.208975445
*天门冬氨酸氨基转换酶	4671	26.85538	13.496	10.04	20.26	23.95	29.31	434.95	0.208975445
*总蛋白	4671	76.78613	4.038641	57.32	74.19	76.63	79.55	100.41	0.208975445
*球蛋白	4671	30.97013	3.578211	7.06	28.585	30.8	33.19	66.18	0.208975445
*碱性磷酸酶	4671	87.56194	25.54226	22.98	70.46	84.61	100.395	374.32	0.208975445
id	5905	2999.831	1732.707	1	1499	3005	4501	6000	0
中性粒细胞%	5885	56.73424	7.787093	14.4	51.6	56.8	62	88.5	0.00338696
乙肝e抗体	1412	1.754154	0.814114	0	1.29	1.65	2.09	7.17	0.76088061
乙肝e抗原	1412	0.072467	0.527305	0.01	0.01	0.04	0.08	17.52	0.76088061
乙肝核心抗体	1412	1.875664	1.55579	0	1.1	1.645	2.15	17.09	0.76088061
乙肝表面抗体	1412	7.030984	8.258835	0	0.22	3.27	12.2075	42.49	0.76088061
乙肝表面抗原	1412	0.91926	5.601976	0	0.01	0.05	0.09	44.35	0.76088061
低密度脂蛋白胆固醇	4678	3.367285	0.860512	0.56	2.76	3.31	3.9075	8.46	0.207790008
单核细胞%	5885	6.859405	1.566637	3.1	5.8	6.7	7.7	23.2	0.00338696
嗜碱细胞%	5885	0.603076	0.291919	0	0.4	0.6	0.7	3.5	0.00338696
嗜酸细胞%	5885	2.039014	1.698953	0	0.9	1.6	2.6	22.5	0.00338696
尿素	4518	4.989303	1.30049	1.5	4.09	4.875	5.72	13.39	0.23488569
尿酸	4518	355.4415	96.27574	118.67	284.4825	346.82	415.3575	776.59	0.23488569
年龄	5905	45.69145	12.99644	3	35	45	54	93	0
总胆固醇	4678	5.234427	1.026691	1.85	4.53	5.15	5.83	20.46	0.207790008
淋巴细胞%	5885	33.76554	7.246735	7.5	28.8	33.6	38.5	76.3	0.00338696
甘油三酯	4678	1.84341	1.77177	0.27	0.97	1.43	2.16	41.57	0.207790008
白球比例	4671	1.501259	0.219323	0.52	1.36	1.49	1.62	7.12	0.208975445
白细胞计数	5885	6.590641	1.610935	2.8	5.47	6.38	7.46	21.06	0.00338696
白蛋白	4671	45.816	2.613351	29.54	44.13	45.8	47.56	54.08	0.208975445
红细胞体积分布宽度	5885	12.73949	1.018096	10.9	12.2	12.6	13	23.8	0.00338696
红细胞压积	5885	0.440781	0.043308	0.239	0.41	0.44	0.473	0.599	0.00338696
红细胞平均体积	5885	89.08331	4.463317	59	86.9	89.3	91.7	113	0.00338696
红细胞平均血红蛋白浓度	5885	335.3589	11.41958	262	329	336	342	462	0.00338696
红细胞平均血红蛋白量	5885	29.88999	1.998685	16	29.1	30	31	44.7	0.00338696
红细胞计数	5885	4.955582	0.503138	3.01	4.58	4.93	5.32	6.85	0.00338696
肌酐	4518	78.39721	13.84088	39.43	67.75	77.8	87.45	177.42	0.23488569
血小板体积分布宽度	5878	13.31075	2.174959	8	11.7	12.9	14.6	25.3	0.004572396
血小板平均体积	5878	10.65553	0.985757	7.1	10	10.6	11.3	15.2	0.004572396
血小板比积	5878	0.268217	0.062572	0.042	0.23	0.26	0.3	1.52	0.004572396
血小板计数	5885	253.3172	60.99067	37	213	248	289	1271	0.00338696
血糖	5905	5.634579	1.527637	3.07	4.93	5.3	5.77	38.43	0
血红蛋白	5885	147.9695	16.55361	65	136	148	161	204	0.00338696
高密度脂蛋白胆固醇	4678	1.390549	0.315187	0.54	1.17	1.35	1.57	5.28	0.207790008



二．源代码附件

附录 1
介绍：python 预处理数据

```
import pandas as pd
# 读取 CSV 文件，每个生理指标的数量、平均值、偏差、最小值、25%、中位数、75%、最大值以及缺失率等指标
data = pd.read_csv('csv 文件/附件 1：有血糖值的检测数据.csv',encoding='gbk')
# 使用 describe 函数获取统计信息
statistics = data.describe()
# 计算每个特征列的缺失率
missing_rate = 1 - data.count() / len(data)
# 添加缺失率列
statistics.loc['Missing Rate'] = missing_rate
# 导出结果到 Excel 文件
statistics.to_excel('result.xlsx')

# 缺失率的可视化
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
# 读取 CSV 文件
```

```
data = pd.read_csv('csv 文件/附件 1：有血糖值的检测数据.csv',encoding='gbk')
# 计算每列数据的缺失率
missing_rate = 1 - data.count() / len(data)
# 绘制每个特征值的缺失率柱状图
plt.figure(figsize=(20, 20))
missing_rate.plot(kind='barh')
plt.title('Missing Data Rate')
plt.xlabel('Features')
plt.ylabel('Missing Data Rate')
plt.xticks(rotation=90)
plt.show()
#Spearman 系数的热力图反映乙肝指标
import pandas as pd
# 读取原始 CSV 文件
data = pd.read_csv('csv 文件/附件 1：有血糖值的检测数据.csv',encoding='gbk')
# 计算每个特征值的缺失率
missing_values = data.isnull().sum() / len(data)
# 找到缺失率大于 50%的特征值列表
filtered_features = missing_values[missing_values > 0.5].index
# 筛选出符合条件的行
filtered_data = data[data[filtered_features].notnull().any(axis=1)]
# 导出新的 CSV 文件
filtered_data.to_csv('附件 1.乙肝抗体的数据.csv', index=False)

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# 读取新的 CSV 文件
data = pd.read_csv('附件 1.乙肝抗体的数据.csv')
filtered_features = ['乙肝 e 抗体', '乙肝 e 抗原', '乙肝核心抗体', '乙肝表面抗
体', '乙肝表面抗原']
selected_columns = list(filtered_features) + ['血糖']
new_data = data[selected_columns]
# 计算特征值和血糖之间的 Spearman 相关系数
correlation_matrix = new_data.corr(method='spearman')
plt.rc('font', family = 'SimHei', size = 10)
plt.rc('axes', unicode_minus = False)
# 绘制热力图
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Spearman Correlation Heatmap')
plt.show()
missing_values = data.isnull().sum() / len(data)
# 找到缺失率%的特征值列表
```

```
filtered_features = missing_values[(missing_values > 0.1) &
(missing_values < 0.5)].index
# 筛选出符合条件的行
filtered_data = data[data[filtered_features].notnull().any(axis=1)]
# 导出新的 CSV 文件
filtered_data.to_csv('附件 1.缺少率为 20%的数据.csv', index=False)

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# 读取新的 CSV 文件
data = pd.read_csv('附件 1.缺少率为 20%的数据.csv')
# 读取原始 CSV 文件
data_old = pd.read_csv('csv 文件/附件 1：有血糖值的检测数
据.csv',encoding='gbk')
# 计算每个特征值的缺失率
missing_values = data_old.isnull().sum() / len(data_old)
filtered_features = missing_values[(missing_values > 0.1) &
(missing_values < 0.5)].index
selected_columns = list(filtered_features) + ['血糖']
new_data = data[selected_columns]
# 计算特征值和血糖之间的 Spearman 相关系数
correlation_matrix = new_data.corr(method='spearman')
plt.rc('font', family = 'SimHei', size = 10)
plt.rc('axes', unicode_minus = False)
# 绘制热力图
plt.figure(figsize=(10, 8))
#Spearman 系数的热力图反映缺失 20%的指标
# 读取原始 CSV 文件
data = pd.read_csv('csv 文件/附件 1：有血糖值的检测数据.csv',encoding='gbk')
# 计算每个特征值的缺失率

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Spearman Correlation Heatmap')
plt.show()
#正常血糖的数据集提取
import pandas as pd
lower_limit = 3.9
upper_limit = 6.1
# 读取输入文件
data = pd.read_csv('csv 文件/附件 1：有血糖值的检测数据.csv',encoding='gbk')
# 提取符合条件的行
filtered_data = data[(data['血糖'] >= lower_limit) & (data['血糖'] <=
upper_limit)]
```

```
# 将符合条件的行保存到输出文件
filtered_data.to_csv('附件 1.血糖正常的数据.csv', index=False)
data_new = pd.read_csv('附件 1.血糖正常的数据.csv')
statistics = data_new.describe()
statistics.to_excel('正常血糖.xlsx')

#高斯混合模型处理异常值
import pandas as pd
import pylab as plt
import warnings
warnings.filterwarnings('ignore')
plt.rc('font', family = 'SimHei', size = 10)
plt.rc('axes', unicode_minus = False)
data = pd.read_excel('处理完全的数据.xlsx')
#男女 0-1
data['性别'][data['性别'] == '男'] = 1
data['性别'][data['性别'] == '女'] = 0

#删除性别？？
data = data.drop(labels=572, axis = 0)
import numpy as np
from sklearn.mixture import GaussianMixture
# 构造 GMM 模型
gmm = GaussianMixture(n_components=4)
# 拟合数据集
gmm.fit(data)
# 计算每个数据点属于每个高斯分布的概率值
scores = gmm.score_samples(data)
# 设置阈值，判断异常值
threshold = np.percentile(scores, 5)
data_ = data[scores > threshold]
data_.to_excel('高斯混合模型.xlsx')
```

附录 2

介绍：python 编写 XGBoost 筛选特征值

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import r2_score
import xgboost
import numpy as np
import pylab as plt
plt.rc('font', family = 'SimHei', size = 8)
```

```
plt.rc('axes', unicode_minus = False)
data_ = pd.read_excel('高斯混合模型.xlsx', index_col=0)
#x, y
y = data_['血糖']
x = data_.drop(labels=['血糖'], axis = 1).values
#画出 xgboost 的特征重要性图
xgb_reg = Pipeline([
    ('std_scaler', StandardScaler()),
    ('xgb_reg', xgboost.XGBRegressor(max_depth = 4))
])
xgb_reg.fit(x, y)
pd.DataFrame(xgb_reg['xgb_reg'].feature_importances_,
index=data_.columns.drop(labels = '血糖'), columns=['特征重要性
']).sort_values(by = ['特征重要性']).plot.barh()
plt.tight_layout()
plt.savefig('xgboost 的特征重要性图.png', dpi = 500)
#xgboost+特征删除
r2 = []
del_col = data_.columns.drop('血糖')
del_column = []
x_ = x
for i in range(0, 29):
    xgb_reg = Pipeline([
        ('std_scaler', StandardScaler()),
        ('xgb_reg', xgboost.XGBRegressor(max_depth = 4))
    ])
    xgb_reg.fit(x_, y)
    r2.append(r2_score(y, xgb_reg.predict(x_)))
    del_num = np.argmin(xgb_reg['xgb_reg'].feature_importances_)
    del_col = np.delete(del_col, del_num)
    del_column.append(del_col)
    x_ = np.delete(x_, del_num, axis = 1)
#绘制删除特征数量与 R^2 的图片
#这样可以确定要删多少特征
r2 = np.array(r2)
import pylab as plt
plt.figure(figsize=(8, 8))
pd.DataFrame(r2, columns=['r2_score']).plot.line()
plt.tight_layout()
plt.savefig('删除特征数量与 R^2(可以确定要删多少特征).png', dpi = 500)
print(del_column[18])
#spearman 系数
data_ = data_.astype('float64')
plt.figure(figsize=(10, 10))
```

```
corr = data_.corr(method='spearman')['血糖'].drop(labels = ['血糖'], axis
= 0)
pd.DataFrame(corr.values, index=corr.index,
columns=['spearman']).sort_values(by=['spearman']).plot.barh()
plt.tight_layout()
plt.savefig('Spearman.png', dpi = 500)
```

附录 3

介绍：python 编写 基于 PSO 优化的 Xgboost 预测

```
import pandas as pd
data = pd.read_excel('../Data/高斯混合模型.xlsx')[['*天门冬氨酸氨基转换酶',
'尿酸', '年龄', '性别', '甘油三酯', '红细胞体积分布宽度', '红细胞平均体积','
红细胞计数', '血小板平均体积', '血红蛋白', '血糖']]
x = data.iloc[:,:-1]
y = data.iloc[:,-1]
from sklearn.preprocessing import MinMaxScaler
x = MinMaxScaler().fit_transform(x)
#划分训练集与测试集
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y)
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error
import xgboost
ret = []
for i in range(0, 10):
    tmp = []
    for model in [LinearRegression(), SVR(), KNeighborsRegressor(),
xgboost.XGBRegressor(min_child_weight = 10, eta = 0.05, colsample_bytree
= 0.5, n_estimators = 136), MLPRegressor()]:
        model.fit(x_train, y_train)
        tmp.append(mean_squared_error(y_test, model.predict(x_test)) **
0.5)
    ret.append(tmp)
import numpy as np
import pylab as plt
pd.DataFrame(np.array(ret), index = range(0, 10),
columns=['LinearRegress', 'SVM', 'KNN', 'xgboost', 'BP']).plot.barh()
plt.savefig('模型比较.png', dpi = 500)
import xgboost
from sko.PSO import PSO
```

```
def f(params):
    model = xgboost.XGBRegressor(min_child_weight = int(params[0]), eta =
params[1], colsample_bytree = params[2], n_estimators = int(params[3]))
    model.fit(x_train, y_train)
    rmse = mean_squared_error(y_test, model.predict(x_test)) ** 0.5
    print(rmse)
    return rmse
pso = PSO(func=f, dim = 4, lb = [5, 0.01, 0, 100], ub = [15, 0.10, 1,
1000], pop=10, max_iter=30)
pso_ret = pso.run()
print('粒子群算法最优参数', pso_ret.gbest_x
def plot_learning_curves(model, X, y):
    X_train, X_val, y_train, y_val = train_test_split(X, y,
test_size=0.2, random_state=10)
    train_errors, val_errors = [], []
    for m in range(1, len(X_train), 50):
        model.fit(X_train[:m], y_train[:m])
        y_train_predict = model.predict(X_train[:m])
        y_val_predict = model.predict(X_val)
        train_errors.append(mean_squared_error(y_train[:m],
y_train_predict))
        val_errors.append(mean_squared_error(y_val, y_val_predict))

    plt.plot(np.sqrt(train_errors), "r-+", linewidth=2, label="train")
    plt.plot(np.sqrt(val_errors), "b-", linewidth=3, label="val")
    plt.legend(loc="upper right", fontsize=14)   # not shown in the book
    plt.xlabel("Training set size", fontsize=14) # not shown
    plt.ylabel("RMSE", fontsize=14)              # not shown
plot_learning_curves(xgboost.XGBRegressor(min_child_weight =
int(pso_ret.gbest_x[0]), eta = pso_ret.gbest_x[1], colsample_bytree =
pso_ret.gbest_x[2], n_estimators = int(int(pso_ret.gbest_x[3]))), x, y)
plt.savefig('学习曲线.png', dpi = 500)
```

附录 4

介绍：python 编写 K-Means 聚类

```
import pandas as pd
data = pd.read_excel('高斯混合模型.xlsx')[['*天门冬氨酸氨基转换酶', '尿酸',
'年龄', '性别', '甘油三酯', '红细胞体积分布宽度', '红细胞平均体积',
       '红细胞计数', '血小板平均体积', '血红蛋白', '血糖']]

#画出聚类后的图
import pandas as pd
from sklearn.cluster import KMeans
```

```
import warnings
import pylab as plt
warnings.filterwarnings('ignore')
plt.rc('font', family = 'SimHei', size = 10)
plt.rc('axes', unicode_minus = False)

kmeans_per_k = [KMeans(n_clusters=k, random_state=42).fit(data.iloc[:,-
1].values.reshape(-1, 1))
                for k in range(1, 10)]
inertias = [model.inertia_ for model in kmeans_per_k]
plt.plot(range(1, 10), inertias, "bo-")
plt.grid()
plt.xticks(range(1, 10, 1))
plt.xlabel("$k$", fontsize=14)
plt.ylabel("Inertia", fontsize=14)
plt.annotate('Elbow',
             xy=(3, inertias[2]),
             xytext=(0.55, 0.55),
             textcoords='figure fraction',
             fontsize=16,
             arrowprops=dict(facecolor='black', shrink=0.1)
            )
plt.savefig('肘部图.png', dpi = 500)

md = KMeans(4).fit(data.iloc[:,-1].values.reshape(-1, 1))
for i in range(0, 4):
    plt.plot(data.iloc[:,-1][md.labels_ == i], 'o', markersize = 1)
plt.savefig('聚类效果.png', dpi = 500)

from sklearn.metrics import silhouette_samples
from matplotlib.ticker import FixedLocator, FixedFormatter
import pylab as plt
import matplotlib as mpl
from sklearn.metrics import silhouette_score
import numpy as np

X = data.iloc[:,-1].values.reshape(-1, 1)
kmeans_per_k = [KMeans(n_clusters=k, random_state=42).fit(X)
                for k in range(1, 10)]


silhouette_scores = [silhouette_score(X, model.labels_)
                     for model in kmeans_per_k[1:]]
plt.figure(figsize=(11, 9))
```

```
for k in ([4]):

    y_pred = kmeans_per_k[k - 1].labels_
    silhouette_coefficients = silhouette_samples(X, y_pred)

    padding = len(X) // 30
    pos = padding
    ticks = []
    for i in range(k):
        coeffs = silhouette_coefficients[y_pred == i]
        coeffs.sort()

        color = mpl.cm.Spectral(i / k)
        plt.fill_betweenx(np.arange(pos, pos + len(coeffs)), 0, coeffs,
                          facecolor=color, edgecolor=color, alpha=0.7)
        ticks.append(pos + len(coeffs) // 2)
        pos += len(coeffs) + padding

    plt.gca().yaxis.set_major_locator(FixedLocator(ticks))
    plt.gca().yaxis.set_major_formatter(FixedFormatter(range(k)))
    if k in (3, 5):
        plt.ylabel("Cluster")

    if k in (5, 6):
        plt.gca().set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1])
        plt.xlabel("Silhouette Coefficient")
    else:
        plt.tick_params(labelbottom=False)

    plt.axvline(x=silhouette_scores[k - 2], color="red", linestyle="--")
    plt.title("$k={}$".format(k), fontsize=16)
plt.savefig('聚类检验图')
```

附录 5

介绍：python 编写 第四问预测

```
import pandas as pd
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
data = pd.read_csv('附件 2：无血糖值的检测数据.csv', encoding='gbk')[['*天
门冬氨酸氨基转换酶', '尿酸', '年龄', '性别', '甘油三酯', '红细胞体积分布宽度
', '红细胞平均体积',
```

```
       '红细胞计数', '血小板平均体积', '血红蛋白']]
data = pd.read_excel('第四问.xlsx', index_col=0)
data = data.fillna(data.mean())
import joblib
from sklearn.preprocessing import MinMaxScaler
x = MinMaxScaler().fit_transform(data)
model = joblib.load('xgboost.pkl')
data['血糖'] = model.predict(x)
data.to_excel('第四问.xlsx')
#男女0-1
data['性别'][data['性别'] == '男'] = 1
data['性别'][data['性别'] == '女'] = 0
import pylab as plt
plt.plot(data['血糖'], 'o', markersize = 1)

import joblib
model = joblib.load('k-means.pkl')

labels_ = model.predict(data['血糖'].values.reshape(-1, 1))
for i in range(4):
    plt.plot(data[labels_ == i]['血糖'], 'o', markersize = 1)
    print()
    plt.axhline(data[labels_ == i]['血糖'].max(), c = 'r')
```