# 数据挖掘课程作业

## 基于 XGBoost 的贷款逾期预测模型构建与应用研究


作 者 姓 名：黄一奇

学科、专业 ：数据科学与大数据技术

学　　号 ：22392023



完 成 日 期：2025-06-30



大连大学

Dalian University

# 摘　要

随着金融市场的快速发展，贷款业务已成为金融机构的核心业务之一，而贷款逾期预测作为信贷风险管理的重要组成部分，对于金融机构来说具有重大意义。逾期贷款不仅会导致金融机构面临经济损失，还可能引发连锁反应，影响整个金融系统的稳定性。因此，如何准确预测和防范贷款逾期风险，是金融机构亟需解决的问题。

本次研究的主要目标是开发一个贷款逾期预测模型。为实现这一目标，利用XGBoost 算法进行模型训练。首先，对金融数据进行数据预处理，包括数据清洗、缺失值和异常值处理；其次，通过 IV 值和随机森林进行特征选择与构造，筛选出对贷款逾期预测具有重要影响的特征变量，并基于原始数据生成新的特征；接着，利用XGBoost 算法进行模型训练，并通过交叉验证和网格搜索等技术对模型参数进行优化；最后，通过准确率、精确率、召回率、F1 分数和 AUC 值等评估指标对模型的预测效果进行全面衡量。

经过详细的实验与分析，本研究成功构建了一个性能良好的贷款逾期预测模型，模型在测试集上取得了较高的 AUC 值（0.7826）。与传统方法相比，本研究采用的XGBoost 算法具有更强的预测能力和更高的模型稳定性，能够有效处理金融数据中的不平衡问题和复杂特征。此外，研究成果不仅为金融机构提供了一种可靠的信贷风险评估工具，也为金融科技领域的进一步研究提供了新的思路和方法，具有重要的理论意义和实践价值。

**关键词**：贷款逾期预测；XGBoost 算法；IV 值；网格搜索

# 目录

# 1 绪论

## 1.1 选题背景

随着金融市场的快速发展，贷款业务已成为金融机构的核心业务之一。贷款逾期预测作为信贷风险管理的重要组成部分，对于金融机构来说具有重大意义[1]。逾期贷款不仅会导致金融机构面临经济损失，还可能引发连锁反应，影响整个金融系统的稳定性。因此，如何准确预测和防范贷款逾期风险，是金融机构亟需解决的问题。特别是在全球经济不确定性增加的背景下，贷款逾期预测对于金融机构来说尤为重要。

近年来，中国政府高度重视金融风险的防控，出台了一系列政策和措施，以加强金融市场的监管和风险管理。例如，中国银保监会发布的《商业银行押品管理指引》强调了商业银行在信贷管理中应加强押品的风险评估和价值管控。此外，随着《征信业管理条例》的实施，个人信用信息的采集和应用更加规范，为贷款逾期预测提供了更为可靠的数据支持。

金融科技的发展为贷款逾期预测提供了新的解决方案。大数据、人工智能和机器学习等技术的应用，使得金融机构能够更有效地处理和分析海量的金融数据，从而提高风险预测的准确性。XGBoost 作为一种高效的机器学习算法，已被广泛应用于各种预测任务中，包括贷款逾期预测。

本研究旨在构建一个有效的预测模型，以预测贷款用户是否会逾期。通过对金融数据的深入分析，将识别影响逾期行为的关键因素，并利用机器学习算法，如 XGBoost，来训练预测模型。模型的准确性和泛化能力将通过一系列评估指标进行衡量，包括准确率、精确率、召回率、F1 分数和 AUC 值等。此外，考虑到金融数据的特点，如数据不平衡和特征多样性，本研究还将探讨如何处理这些挑战，以提高模型的性能。例如，可能会采用数据预处理技术来平衡数据集，或者使用特征选择和构造方法来提取最有价值的信息。

通过本研究，期望能够为金融机构提供一种可靠的工具，以更好地管理信贷风险，提高贷款业务的盈利能力和可持续性。同时，本研究也将为金融科技领域的进一步研究提供参考和启示。在全球金融监管日益严格的背景下，本研究不仅具有理论意义，也具有重要的实践价值。通过提高贷款逾期预测的准确性，金融机构可以更好地遵守监管要求，增强风险抵御能力，促进金融市场的健康发展[2]。

## 1.2 研究意义

本研究聚焦于金融领域中一个至关重要的问题：贷款逾期预测。通过应用数据分析技术和机器学习算法，旨在提高预测模型的准确性和效率，这对于金融机构的风险

管理具有重大意义。准确预测贷款用户的逾期行为可以帮助金融机构提前采取措施，如调整信贷政策或加强催收工作，从而降低逾期率和违约风险，这对于维护金融机构的稳健运营至关重要。此外，本研究的成果能够为金融机构的信贷决策提供强有力的数据支持，帮助它们更合理地分配信贷资源，优化贷款组合，提高资产质量和盈利能力。在更广泛的层面上，通过有效的风险预测和管理，可以减少系统性金融风险，维护金融市场的稳定，对整个金融系统的健康发展具有积极影响。

金融科技的发展为贷款逾期预测提供了新的解决方案。本研究展示机器学习技术在金融风险管理中的应用潜力，推动金融科技在信贷风险评估领域的创新和应用。同时，研究结果也将为监管机构制定相关政策提供参考，帮助它们更好地理解和监管金融市场的风险，提高监管的有效性和针对性。

在学术领域，本研究将丰富金融风险管理和机器学习领域的研究，为后续研究提供新的方法论和实证分析基础。社会层面上，通过提高贷款审批的准确性，可以促进金融包容性，使更多有信用但缺乏传统抵押物的个人或小微企业能够获得贷款，支持社会经济的发展。

本研究还强调了数据在金融决策中的重要性，推动金融机构加强数据收集、处理和分析能力，实现数据驱动的业务模式[3]。总体而言，本研究不仅对金融机构具有直接的应用价值，有助于提升其风险管理能力，而且对监管机构、学术界乃至整个社会都具有深远的影响。通过提高贷款逾期预测的准确性，可以促进金融资源的合理配置，增强金融市场的稳定性，推动金融科技的发展，最终实现金融行业的可持续发展。

## 1.3 研究目标与主要内容

本研究旨在开发一个精确的贷款逾期预测模型，目的是辅助金融机构在贷款审批过程中做出更加明智的决策。研究的核心目标是构建一个基于 XGBoost 算法的预测模型，并通过优化模型参数和进行特征工程来提高模型的预测准确性和泛化能力。此外，研究还旨在通过模型评估不同贷款申请者的风险等级，为金融机构提供风险管理的决策支持，并基于模型预测结果制定贷款审批和风险控制策略。

为了实现这些目标，研究将从文献回顾开始，总结贷款逾期预测领域的研究进展，并明确研究的理论基础和实际应用价值。接着，将进行数据收集与预处理，包括收集贷款数据、清洗数据、处理缺失值和标准化，确保数据的质量和一致性。特征选择与构造是研究的另一个重点，将分析贷款数据，识别影响逾期行为的关键因素，并提取有价值的信息[4]。

在模型开发阶段，将基于 XGBoost 算法构建预测模型，并探索不同的参数配置以优化模型性能。模型评估将使用准确率、精确率、召回率、F1 分数和 AUC 值等指

标进行，同时比较调参前后模型的性能，分析模型的优势和不足。结果分析将深入探讨影响逾期行为的主要因素，并对比不同模型和参数设置的性能，提出改进建议。

研究还将提出基于模型预测结果的策略建议，为金融机构提供贷款审批和风险控制的指导。最后，研究将总结成果，讨论研究的局限性，并展望未来研究方向，为后续研究提供参考。

通过这些研究内容，期望为金融机构提供一个有效的贷款逾期预测工具，帮助它们更好地管理信贷风险，提高贷款业务的盈利能力和可持续性。同时，本研究也将为金融科技领域的研究提供新的视角和方法，推动金融风险预测技术的发展。

# 2 相关工作与理论基础

## 2.1 国内外相关研究现状

在国内外金融科技领域，贷款逾期预测的研究现状呈现出积极的发展态势。国际上，金融机构和学术界都在积极探索如何利用大数据和机器学习技术来提高贷款逾期预测的准确性。由于金融市场的成熟和数据科学的发展，贷款逾期预测的研究较为深入。许多研究集中在开发复杂的机器学习模型，如集成学习方法和深度学习，以及如何通过特征工程来提高模型的预测能力。此外，国际上的研究还关注于模型的可解释性，即如何使模型的决策过程更加透明，以便监管机构和金融机构能够理解和信任模型的预测结果[5]。

在国内，随着金融市场的快速发展和金融科技的兴起，贷款逾期预测的研究也得到了广泛关注。中国的金融机构正在积极采用机器学习技术来优化信贷风险管理。国内的研究者们在借鉴国际先进经验的同时，也在探索适合中国市场特点的预测模型。例如，考虑到中国特有的社会信用体系和金融环境，研究者们可能会更加关注如何利用本地化的数据和特征来提高模型的适用性。此外，国内的研究也在关注如何处理数据不平衡问题，以及如何通过在线学习和实时预测技术来适应金融市场的快速变化。

无论是国内还是国外，贷款逾期预测的研究都在不断进步，研究者们都在努力提高模型的预测准确性、泛化能力和解释性。随着技术的不断发展和金融市场的日益复杂，预计未来这一领域的研究将更加深入，并可能带来新的理论和实践突破。这些研究不仅有助于金融机构更好地管理信贷风险，也为金融监管提供了强有力的工具，以维护金融市场的稳定和健康发展。

## 2.2 数据挖掘相关算法

### 2.2.1 XGBoost

XGBoost（Extreme Gradient Boosting）是一种基于梯度提升（Gradient Boosting）框架的集成学习算法，它在金融风险预测领域表现出色。XGBoost 通过构建多个弱预测模型（通常是决策树），并将它们的结果进行加权合并，以提高整体预测的准确性。该算法的一个关键特点是它能够自动处理各种数据类型，包括分类数据和连续数据，这使得它非常适合处理金融数据集中常见的特征多样性[6]。

XGBoost 还具有正则化功能，这有助于防止模型过拟合，特别是在数据量有限的情况下。此外，XGBoost 提供了丰富的参数供用户调整，以优化模型性能。在本研究中，将利用 XGBoost 的这些优势来构建一个强大的逾期预测模型，并通过调整参数如学习率、树的最大深度、子样本比例等来提高模型的泛化能力。

### 2.2.2 Logistic Regression

逻辑回归是另一种广泛用于二分类问题的经典算法。尽管它在概念上相对简单，但在金融风险预测中仍然非常有效。逻辑回归模型通过估计事件发生的概率来预测结果，它使用逻辑函数（Logistic Function）将线性组合的输入特征映射到 0 和 1 之间的概率值。

逻辑回归的一个主要优点是模型的可解释性强，因为它直接提供了特征对事件发生概率影响的大小和方向。这在金融领域尤为重要，因为监管机构和金融机构通常需要理解模型的决策依据。在本研究中，将使用逻辑回归作为基线模型，与 XGBoost 模型的性能进行比较。此外，逻辑回归还可以帮助识别影响贷款逾期的关键因素，为进一步的特征工程和模型优化提供指导[7]。

# 3 数据分析与处理

## 3.1 数据来源

本数据集来源于 Datawhale——AI 开源学习社区（https://www.datawhale.cn/）。该数据集是社区提供的用于预测贷款用户是否会逾期的样本数据，涵盖了用户贷款行为相关的特征，包括交易金额变化率、贷款逾期次数、申请评分、历史交易金额等 90 个维度。

## 3.2 数据可视化与探索分析

### 3.2.1 整体观察

在对数据进行初步探索时，主要通过 shape()、describe()、自定义的 overall()以及对正负样本的数量进行统计这四种方式进行整体的观察和分析。以下是具体的整体观察和分析内容。

首先通过调用 shape 函数，了解到数据的维度为(4754,90)，可知数据集中共有4754 条记录，以及 90 个特征，共 269120 个数据，文件大小为 1.867MB，数据量级为MB 级。由数据集中有 90 个特征可知，后期需要进行特征选择，以删除一些无关特征，减少模型复杂度并提高模型性能。此外，通过计算 IV（信息价值）值，筛选出了 6 个重要的字段，并进行了相关的数据说明，如表 1 所示。

表 1 主要字段的数据说明

| 属性名称 | 属性说明 | 示例 | IV 值 |
|---|---|---|---|
| historical_trans_amount | 历史交易金额 | 149050 | 53.956445 |
| trans_amount_3_month | 过去 3 个月的交易金额 | 34030 | 42.480944 |
| pawns_auctions_trusts_consume_last_6_month | 过去 6 个月的典当、拍卖、信托消费金额 | 18040 | 32.080975 |
| repayment_capability | 还款能力 | 19890 | 27.977140 |
| consume_mini_time_last_1_month | 过去 1 个月的最小消费时间间隔 | 5 | 22.833834 |
| consfin_avg_limit | 消费金融平均额度 | 1200 | 21.590355 |

接着，绘制了这六个特征的直方图，如图 1。该图中每个子图对应一个特征，横轴是特征值，纵轴是频率。通过这些图，可以观察到每个特征的分布情况：historical_trans_amount（历史交易金额）：分布高度偏斜，大多数交易金额集中在较低的范围内，少数交易金额非常高，呈现出长尾分布。trans_amount_3_month（三个月交易金额）：与历史交易金额类似，也呈现出右偏分布，但分布稍微更集中一些。pawns_auctions_trusts_consume_last_6_month（过去六个月典当、拍卖、信托消费金额）：分布同样偏斜，大多数消费金额较低，少数消费金额较高。repayment_capability（还款能力）：分布较为集中，大多数用户的还款能力在中等范围内，少数用户的还款能力非常高或非常低。consume_mini_time_last_1_month（过去一个月最小消费时间间隔）：分布也偏斜，大多数用户的消费时间间隔较短，少数用户的消费时间间隔较长。consfin_avg_limit（消费金融平均额度）：分布较为集中，大多数用户的平均额度在中等

范围内，少数用户的平均额度非常高。

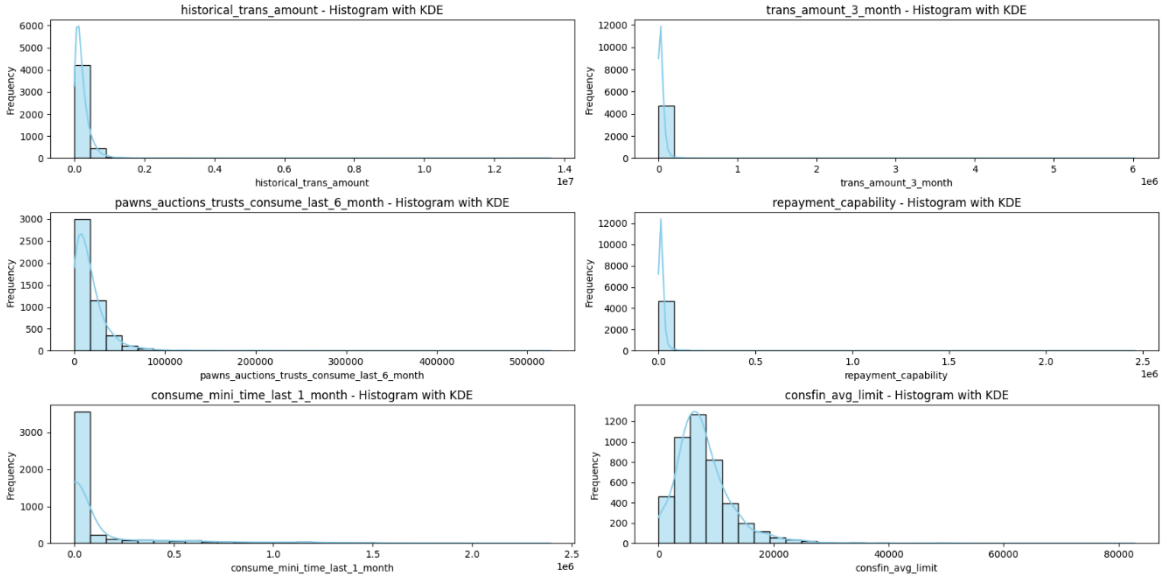


图 1 特征直方图

其次，绘制了这六个特征的散点图矩阵，如图 2，展示了六个特征之间的关系以及它们与标签（status）之间的关系。每个小图展示了两个变量之间的关系，颜色区分了不同的标签类别（0 表示未逾期，1 表示逾期）。

从图中可以分析出，大多数特征之间没有明显的线性关系，但可以观察到一些特征之间存在一定的相关性，例如 historical_trans_amount 和 trans_amount_3_month 之间。通过颜色的分布，可以初步观察到某些特征与标签之间可能存在一定的关系。例如，pawns_auctions_trusts_consume_last_6_month 和 status 之间的关系图中，逾期（红色）和未逾期（蓝色）的分布有一定的区分度。

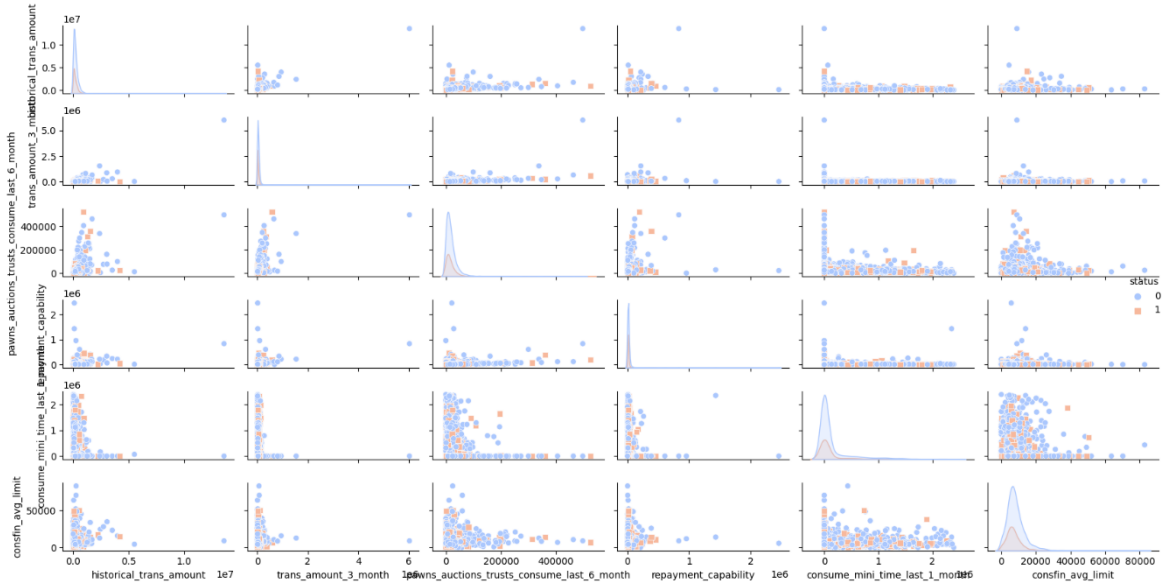

图 2 特征散点图矩阵

然后，绘制了特征相关性热力图，如图 3，展示了六个特征以及标签之间的相关性系数。从图中可以分析出，historical_trans_amount 和 trans_amount_3_month 之间的相关性非常高（0.79），这表明这两个特征可能在某种程度上衡量了相似的经济活动。pawns_auctions_trusts_consume_last_6_month 与 historical_trans_amount 和 trans_amount_3_month 也有较高的相关性（分别为 0.55 和 0.51），这可能反映了用户在不同金融活动中的消费行为具有一定的一致性。其次，大多数特征与标签之间的相关性较低，这表明这些特征可能不是预测逾期的强指标。然而，repayment_capability 与 status 之间的相关性为-0.09，虽然不高，但表明还款能力较低的用户可能更有可能逾期。

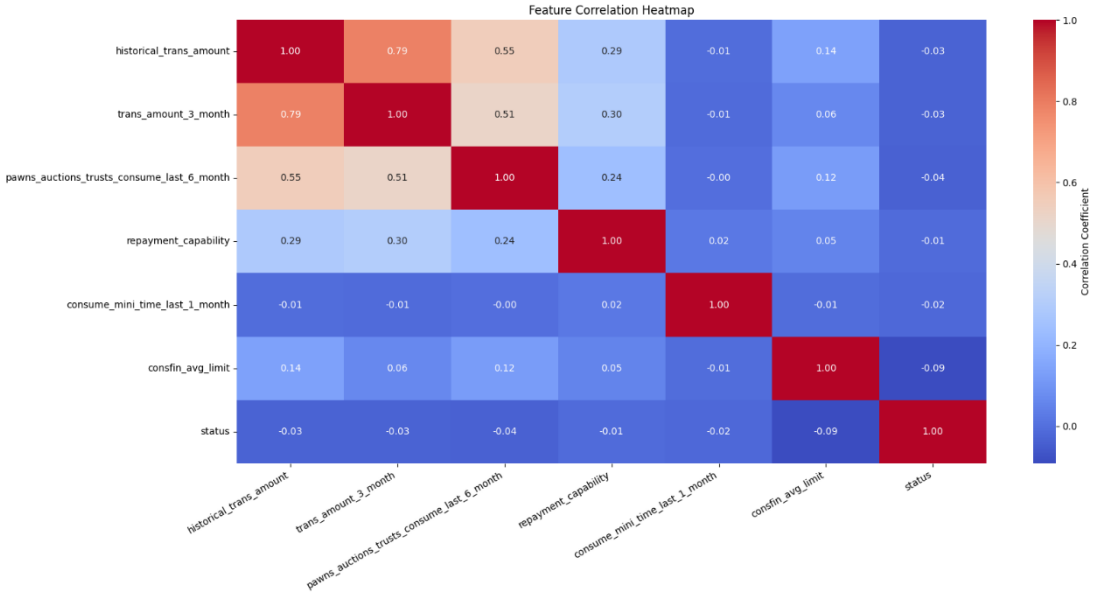


图 3 特征相关性热力图

随后，通过调用 describe 函数对数据进行了描述性统计分析，部分数据的描述性统计分析如图 4 所示。由于特征数量较多，所以难以一次性观察到所有特征的详细统计信息，但可以得知数值型特征共有 83 个。

| | Unnamed: 0 | custid | low_volume_percent | middle_volume_percent | take_amount_in_later_12_month_highest |
|---|---|---|---|---|---|
| **count** | 4754.000000 | 4.754000e+03 | 4752.000000 | 4752.000000 | 4754.000000 |
| **mean** | 6008.414178 | 1.690993e+06 | 0.021806 | 0.901294 | 1940.197728 |
| **std** | 3452.071428 | 1.034235e+06 | 0.041527 | 0.144856 | 3923.971494 |
| **min** | 5.000000 | 1.140000e+02 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 3106.000000 | 7.593358e+05 | 0.010000 | 0.880000 | 0.000000 |
| **50%** | 6006.500000 | 1.634942e+06 | 0.010000 | 0.960000 | 500.000000 |
| **75%** | 8999.000000 | 2.597905e+06 | 0.020000 | 0.990000 | 2000.000000 |
| **max** | 11992.000000 | 4.004694e+06 | 1.000000 | 1.000000 | 68000.000000 |

8 rows × 83 columns

图 4 部分数据的描述性统计分析

为进一步了解数据的特征类型分布，通过自定义的 overall 函数对数据进行了更深入的分析。结果显示，数据集中有 7 列是非数值类型（object 类型），这些列的名称分别为'trade_no'，'bank_card_no'，'reg_preference_for_trad'，'source'，'id_name'，'latest_query_time'，'loans_latest_time'。这些非数值类型的特征可能包含了重要的信息，但需要根据具体分析目标和模型需求，决定是否将其纳入后续的分析流程，或者是否需要对其进行进一步的处理和转换。

最后，统计了正负样本的数量，正样本为 1193，负样本为 3561，比值约为 1:3，呈现出中度的不平衡分类。因此，在后期进行模型的选择和训练时，需要注意以下几点。首先，优先选择对不平衡数据具有较好鲁棒性的模型，如 XGBoost 和支持向量机（SVM）等。其次，在模型训练过程中，为正负样本分配不同的权重，使模型更加关注少数类样本，避免模型过度偏向多数类。最后，避免仅使用准确率（Accuracy）作为评估指标，建议使用召回率（Recall）、精确率（Precision）、F1 分数、ROC-AUC 曲线等指标，以更全面地评估模型性能。通过以上措施，可以有效缓解不平衡分类问题对模型性能的影响，提高模型对少数类的识别能力。

### 3.2.2 缺失值可视化与探索分析

在数据分析领域，缺失值是一个普遍存在的问题，它可能对模型的性能和分析结果的准确性产生负面影响。因此，对缺失值进行可视化和探索性分析是数据预处理过程中的关键步骤。本实验主要通过缺失值矩阵图和缺失值数量柱状图进行可视化。以下是缺失值可视化和探索分析的详细内容。

首先，为了直观地展示数据中的缺失值情况，利用 missingno 库绘制了缺失值矩阵图（如图 5 所示）。

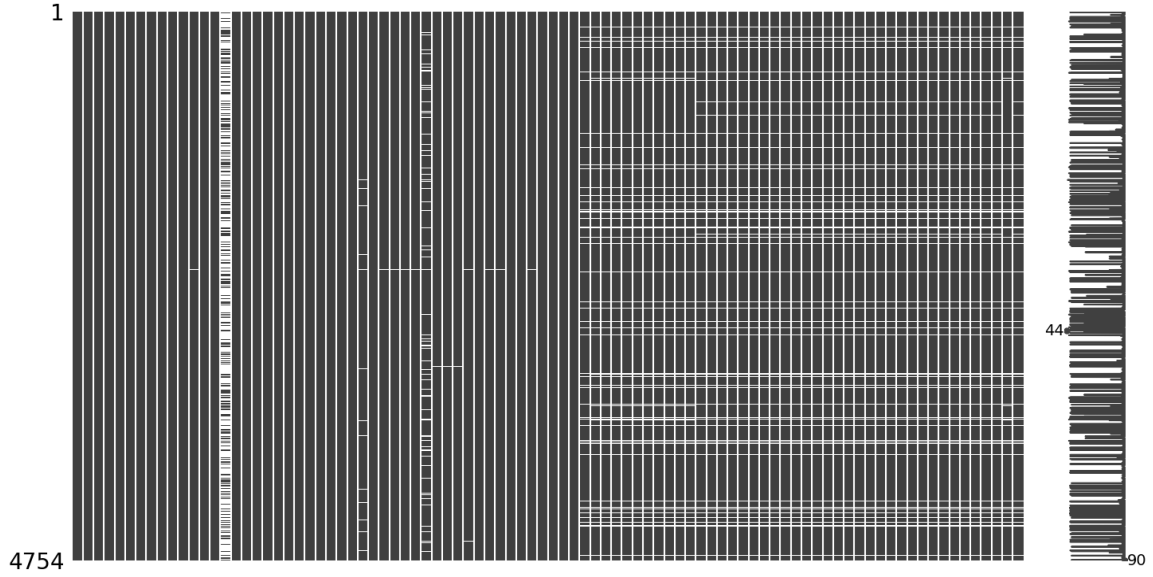


图 5 缺失值矩阵图

通过该图，可以清楚地观察到从右往左数第 15 个特征的缺失值情况较为严重，并且从图的右半部分可以直观地看出部分行数据中连续缺失了一半的数据。

为了进一步识别数据缺失较为严重的特征，采用了 isnull().sum()方法来统计每个特征的缺失值数量，并将其与数据集的总行数进行比较以计算出缺失率。随后，筛选出存在缺失值的列，并按照缺失数量进行降序排序。最终，利用 matplotlib 库绘制了缺失值数量前十的特征的柱状图（如图 6）。

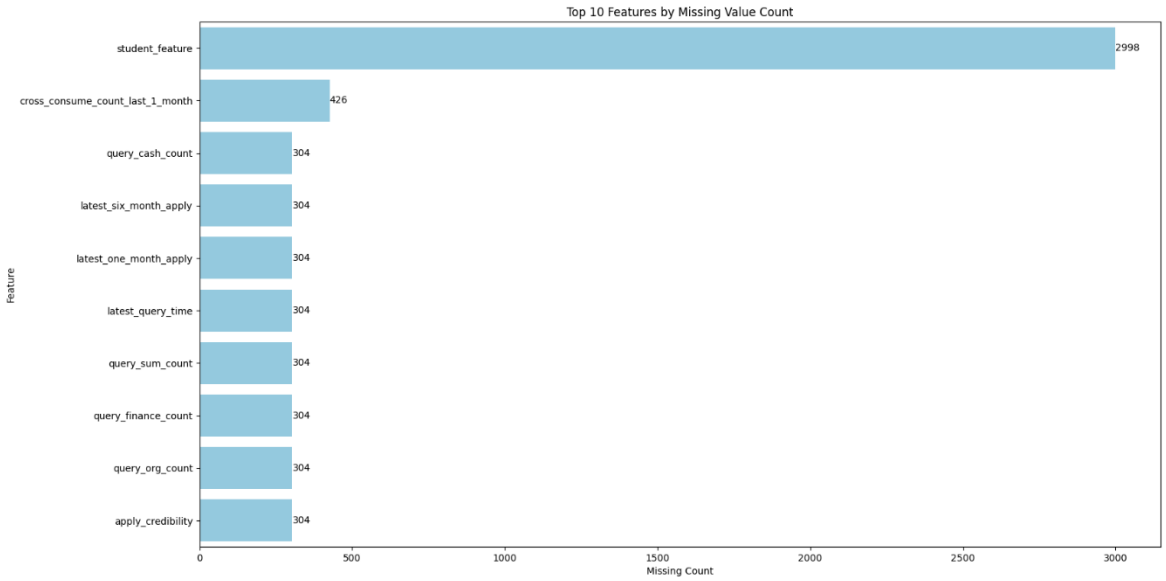


图 6 缺失值数量前十的特征的柱状图

从图中可以发现，student_feature 特征的缺失数量最多，达到了 2998 个，缺失率约为 63%。如果该特征的重要性不高，可以考虑在后续处理中将其删除。至于其他特征，它们的缺失数量都在 500 以下，缺失率低于 10%，可以考虑采用适当的方法进行填充。

### 3.2.3 异常值可视化与探索分析

在数据分析中，异常值的识别和处理是至关重要的，因为它们可能会对模型的预测性能产生显著影响。通过箱线图可视化方法，可以直观地识别数据中的异常值，并进行进一步的探索分析。以下是对异常值可视化和探索分析的详细描述。

由于特征有 90 个，数量很多，不能一一进行异常值的可视化，所以首先通过使用随机森林分类器（Random Forest Classifier）来评估特征的重要性，筛选出了前 8 个最重要的特征，并打印了它们的重要性指标，如表 2。随机森林是一种集成学习方法，它通过构建多个决策树并结合它们的预测结果来提高模型的准确性和稳定性。特征重要性是通过评估每个特征对模型预测准确性的贡献来计算的。

表 2 前 8 个重要特征的重要性指标

| 特征 | 特征重要性指标 |
|---|---|
| trans_fail_top_count_enum_last_1_month | 0.053701 |
| history_fail_fee | 0.045952 |
| loans_score | 0.034630 |
| apply_score | 0.030945 |
| latest_one_month_fail | 0.021894 |
| loans_overdue_count | 0.020574 |
| trans_amount_3_month | 0.019292 |
| max_cumulative_consume_later_1_month | 0.016936 |

接下来，使用箱线图（Box Plot）来可视化这些重要特征的分布情况，分别为每个重要特征绘制了一个箱线图，并将这些图排列在一个 2 行 4 列的网格中，如图 7。箱线图是一种常用的统计图表，用于显示数据的分布情况，包括中位数、四分位数和异常值。

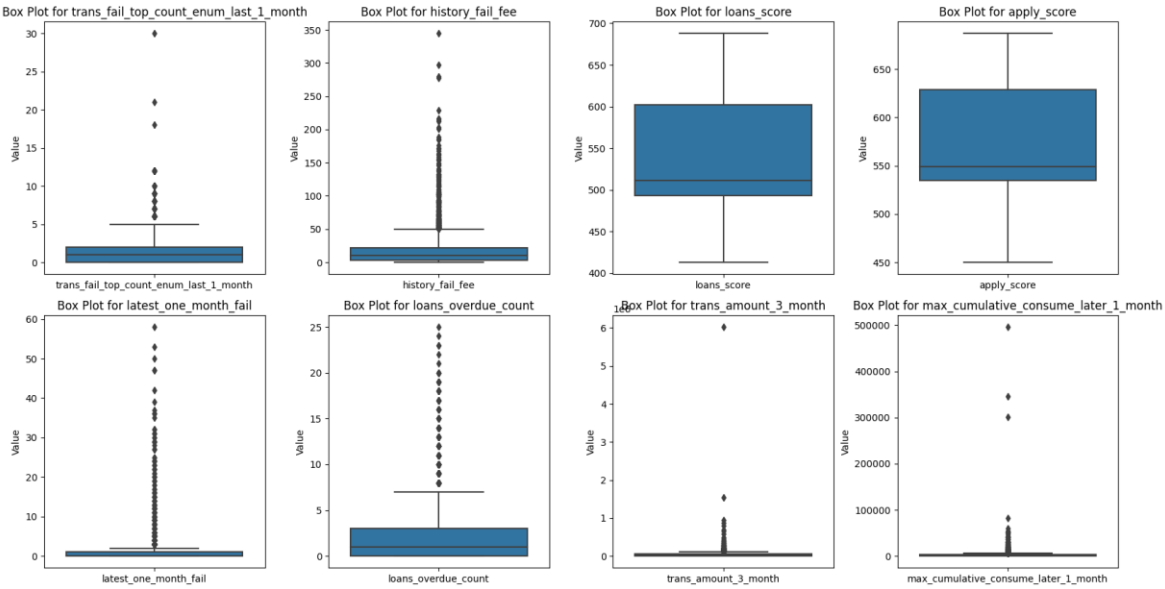


图 7 前 8 个重要特征箱线图

从图中可以观察到以下两点。首先在每个箱线图中，超出箱线范围的点被标记为异常值。例如，在 trans_fail_top_count_enum_last_1_month、history_fail_fee、latest_one_month_fail 和 loans_overdue_count 等特征中，存在一些显著高于其他数据点的异常值。其次，通过比较不同特征的箱线图，可以识别出哪些特征具有更多的异常值或更广泛的数据分布。例如，max_cumulative_consume_later_1_month 特征显示出较大

的数据范围和较多的异常值。

综上，根据异常值的分布和特征的重要性，后续可以决定是否需要对这些异常值进行处理，例如通过删除、替换或使用更复杂的方法（如插值或基于模型的预测）来填充这些异常值。

## 3.3 数据清洗与缺失值处理

### 3.3.1 无关特征删除

在数据预处理阶段，删除无关特征是一个关键步骤，这有助于提高模型的性能并减少计算资源的消耗。特征删除原则为：与标签列无关的特征。具体包括所有用户在该特征上取值相同的特征，以及一些无实际意义的特征，如用户姓名等。这些特征对于模型的预测能力没有贡献，甚至可能引入噪声，影响模型的准确性。

首先，识别并删除了那些在所有用户中取值相同的特征，也就是单一值列。这类特征对于模型学习没有帮助，因为它们不携带任何有助于区分不同类别的信息。通过定义一个函数 same_value_delete 来实现这一点，该函数遍历数据集中的所有列，检查每个特征的值是否全部相同。如果是，那么该特征就会被删除。通过这种方法，从数据集中删除了两个特征：source 和 bank_card_no。

接着，识别并删除了一些无意义的特征。这些特征可能包括用户姓名、交易编号等，它们对于预测任务来说并不重要，甚至可能因为包含敏感信息而需要被排除。具体删除方法是直接通过 drop 方法从数据集中删除了 4 个特征：Unnamed: 0、custid、trade_no 和 id_name。这些特征的删除有助于简化数据集，使得模型能够更加专注于那些真正对预测结果有影响的特征。Unnamed: 0 特征分布直方图如图 8 所示。

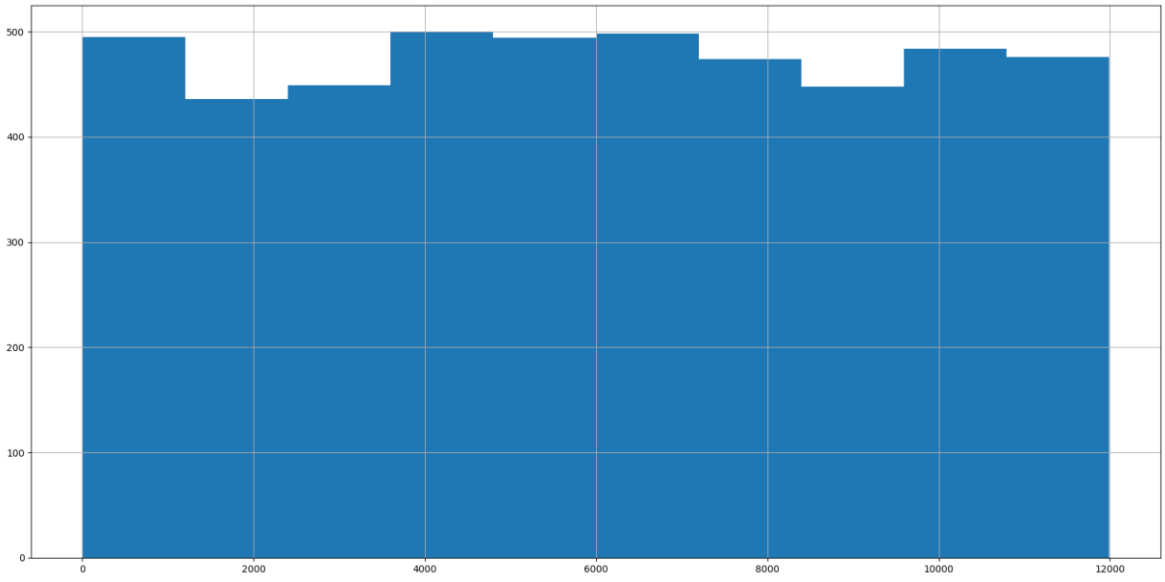


图 8 Unnamed: 0 特征分布直方图

删除 custid、trade_no 和 id_name 这三个特征的原因是基于业务的理解，因为这三个特征分别指消费者账号、交易编号和用户姓名，很明显可以分析出与标签列几乎无相关性。删除 Unnamed: 0 这个特征是由于通过直方图发现该特征分布非常均匀。其次通过类别计数图（如图 9）发现该特征每个样本的取值不一样，和标签列几乎无相关性，可以看成是一个类似于客户 id 的特征，因此删掉。

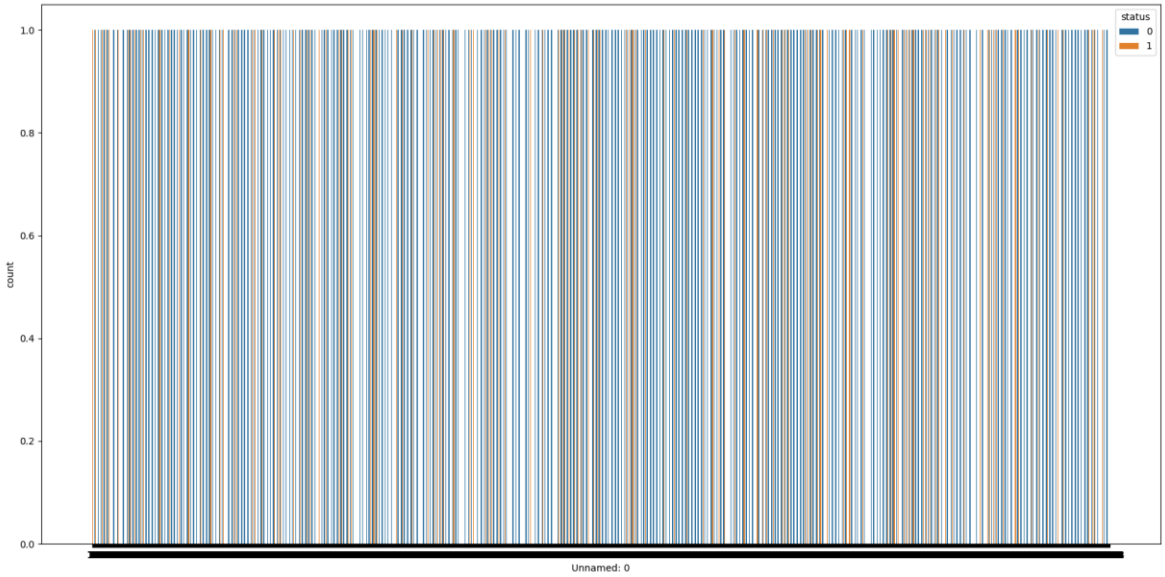


图 9 Unnamed: 0 特征类别计数图

此外，为了方便理解，我还绘制了 trans_fail_top_count_enum_last_1_month 特征的类别计数图（如图 10），可以将图 9 和图 10 进行对比分析。

图 9 展示了 Unnamed: 0 特征的分布情况。从图中可以看出，该特征的值分布非常密集，几乎每个值都有状态为 0 和 1 的样本。这种均匀分布可能意味着 Unnamed: 0 特征对于区分不同状态的样本没有太大的帮助，甚至可能是一个无意义的特征。由于该特征在不同状态的样本中分布相似，它可能不会对模型的预测能力产生显著影响，因此在数据预处理阶段，需要考虑删除这个特征以简化模型并减少噪声。

而从图 10 中，可以观察到 trans_fail_top_count_enum_last_1_month 特征的分布情况。该特征的直方图显示了两个不同状态（0 和 1）的样本分布。状态为 0 的样本在较低的 trans_fail_top_count_enum_last_1_month 值上更为集中，而状态为 1 的样本则在较高的值上分布更多。这表明该特征可能与目标变量 status 有一定的关联。此外，图中还显示出类别不平衡的现象，状态为 0 的样本数量明显多于状态为 1 的样本数量。这种不平衡可能需要在后续的模型训练中进行处理，以避免模型偏向多数类。

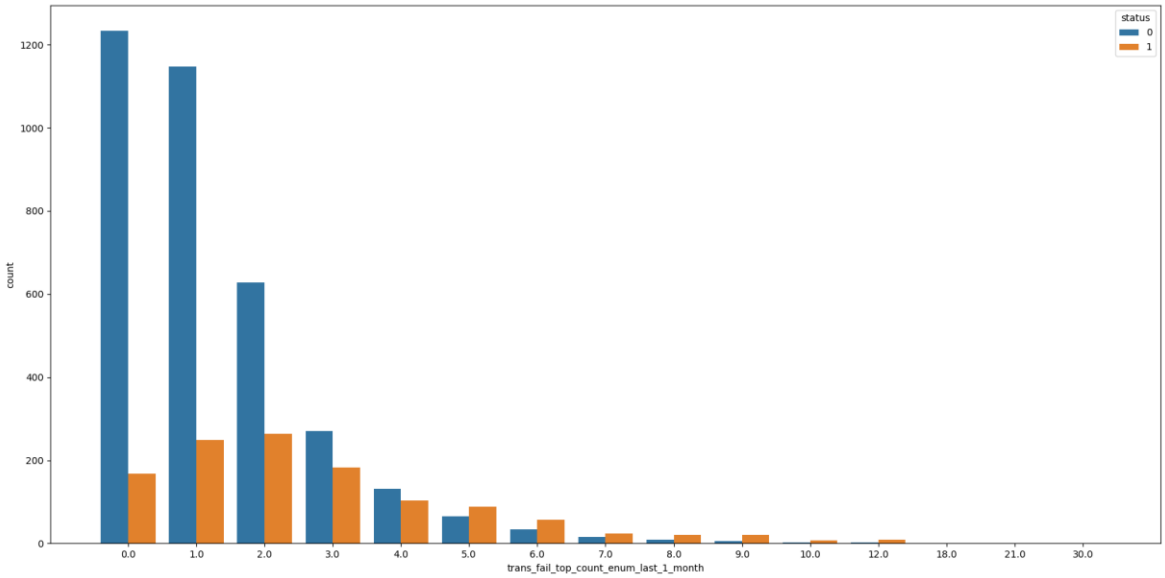


图 10 trans_fail_top_count_enum_last_1_month 特征类别计数图

综上，通过上述步骤，初步对数据集进行了清理，去除了那些无关的特征。这一过程不仅有助于提高模型的预测性能，还能够减少模型训练和预测时所需的计算资源。删除无关特征是数据预处理中的一个重要环节，它为后续的特征选择和模型训练打下了坚实的基础。

### 3.3.2 数据类型转换

在数据类型转换时，通过自定义的 overall（）方法发有三个特征是 object 类型，即非数值类型，这三个特征分别是：reg_preference_for_trad,，latest_query_time 和 loans_latest_time。可以看出后两个特征与时间有关，所以暂时不处理，在特征选择和构造的时候再进行处理。

而第一个特征 reg_preference_for_trad 经过 print（）发现这个特征表示城市级别，并且这些级别之间存在一定的顺序关系。为了更好地利用这一信息，因此决定将这些级别转换为数值。具体来说，就是将“一线城市”赋予数值 5，“二线城市”赋予 4，“三线城市”赋予 3，“境外”赋予 2，“其他城市”赋予 1。这样的转换不仅保留了城市级别的顺序，而且使得数据更容易被模型处理。

综上，通过这种方法，成功地将一个有序的分类变量转换为了数值变量，这有助于提高模型对数据的理解，从而可能提升预测的准确性。

### 3.3.3 缺失值处理

在数据预处理阶段，缺失值的处理是一个关键步骤，它直接影响到模型的性能和预测的准确性。在本次实验中，采取了多种策略来处理缺失值，包括删除缺失率较高

的特征、填补缺失率较低的特征，以及对不同类型特征的缺失值进行针对性处理。

首先，删除缺失率较大的特征，例如 student_feature。由 3.2.2 缺失值可视化与探索分析可以得出，这一特征的缺失值数量较多。接着可视化了类别计数图（如图 11），发现 student_feature 的取值为 2 的只有 2 个样本，其中样本标签一个是 1，一个是 0。因此与标签没有什么相关性，选择将其从数据集中移除。这样的处理有助于减少数据的不完整性，并且避免了在模型训练过程中对这些缺失值进行不准确的填补。

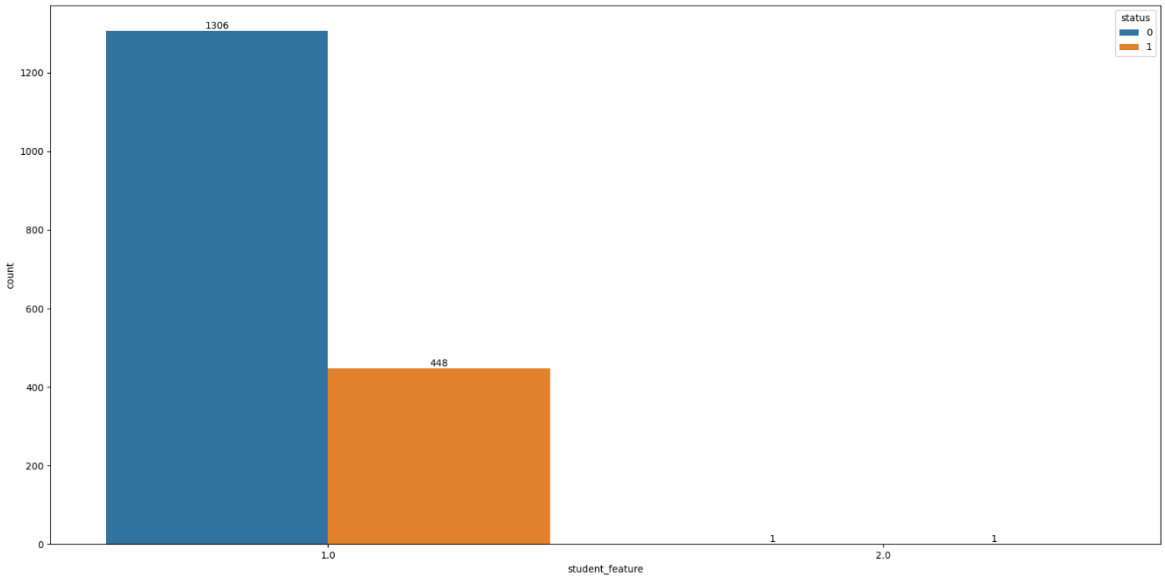


图 11 student_feature 特征类别计数图

接下来，对缺失率较小的特征进行了填补。首先是将这些特征分为三类：时间型特征、类别型特征和数值型特征。然后分别进行缺失值处理，具体处理过程如下。

对于时间型特征。首先通过 head（）方法打印了前五行数据，如表 3 所示。发现 first_transaction_time 为浮点型，所以直接使用中位数来填补 first_transaction_time 的缺失值，然后将日期转换为字符串格式，并从中衍生出年份、月份和星期几等新特征。对于 latest_query_time 和 loans_latest_time，先是并衍生出相应的时间特征，然后使用中位数来填补缺失值。

表 3 前 5 行样本数据

| | first_transaction_time | latest_query_time | loans_latest_time |
|---|---|---|---|
| 0 | 20130817.0 | 2018-04-25 | 2018-04-19 |
| 1 | 20160402.0 | 2018-05-03 | 2018-05-05 |
| 2 | 20170617.0 | 2018-05-05 | 2018-05-01 |
| 3 | 20130516.0 | 2018-05-05 | 2018-05-03 |

| | | | |
|---|---|---|---|
| 4 | 20170312.0 | 2018-04-15 | 2018-01-07 |

对于类别型特征。先通过观察取值和属性名称，挑选类别特征，然后对剩余除时间型特征外的特征进行数据去重，将去重后<15 的特征认为是类别型特征。通过这种方式筛选出了七种特征：regional_mobility 、is_high_user 、avg_consume_less_12_valid_month、top_trans_count_last_1_month、reg_preference_for_trad、railway_consume_count_last_12_month 7 和 jewelry_consume_count_last_6_month 8。最后统计了特征的缺失率，发现特征缺失值都比较少，因此考虑通过计算每个特征的众数（即出现次数最多的值）来填补缺失值。

对于数值型特征时。首先是筛选出除时间型和类别型特征外的数值型特征。接着统计有缺失值的特征，具体包括： cross_consume_count_last_1_month、apply_score、apply_credibility、query_org_count、query_finance_count、query_cash_count、query_sum_count、latest_one_month_apply、latest_three_month_apply 和 latest_six_month_apply。最后根据缺失率选择使用中位数来填补缺失值。中位数是一个稳健的统计量，它不受极端值的影响，适合用于填补数值型数据的缺失值。

综上，通过这些缺失值处理步骤，不仅提高了数据的完整性，还为模型训练准备了更高质量的数据。这些处理方法针对不同类型的特征采取了不同的策略，旨在最大限度地保留数据的信息，同时减少缺失值对模型性能的负面影响。通过这样的数据预处理，可以更有信心地进行后续的模型训练和预测。

### 3.3.4 异常值处理

在数据预处理中，处理异常值是一个重要步骤。异常值是指在数据中明显偏离其他数据点的值，可能由于测量误差、数据录入错误或真实但罕见的事件导致。识别和处理异常值是数据预处理的重要步骤，因为它可以影响模型的训练效果和预测准确性。在本次实验中主要通过绘制箱线图来直观地展示不同量级特征的分布情况，从而帮助识别潜在的异常值。

由于特征数量较多，所以首先根据特征的最大值将特征分为不同的量级范围，如“量级 < 2.5”、“2.5 ≤ 量级 < 20”等。这种分类方法有助于将特征按照其数据范围进行分组，便于后续的可视化分析。在箱线图中，异常值通常表现为超出上边缘（Q3 + 1.5×IQR）和下边缘（Q1 - 1.5×IQR）的点，其中 Q1 和 Q3 分别是第一四分位数和第三四分位数，IQR 是四分位距（Q3 - Q1）。通过观察箱线图中的这些离群点，可以初步判断哪些数据点可能是异常值。具体箱线图的可视化如图 12-15。

从图 12-15 中可以分析得出，在低量级特征箱线图中（图 12），可以看到特征如 is_high_user 和 low_volume_percent 的中位数较低，且存在一些异常值。随着量级的

增加，如在 20 ≤ 量级 < 40 的图中，特征如 trans_top_time_last_1_month 和 consume_top_time_last_1_month 的分布更为分散，中位数在 5 到 10 之间，且异常值的数量增多。

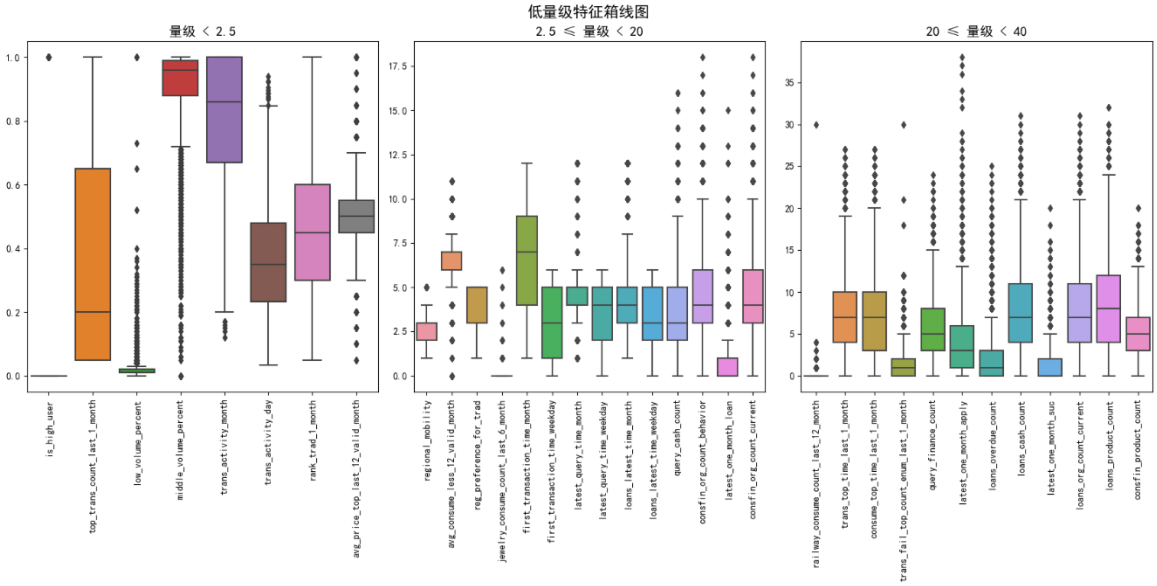


图 12 低量级特征箱线图

中低量级特征箱线图显示了量级在 40 ≤ 量级 < 100 的图中（图 13），特征如 apply_credibility 和 loans_credibility_behavior 的中位数较高，分布相对集中，但仍有异常值存在。在 100 ≤ 量级 < 200 的特征，如 trans_top_time_last_6_month 和 consume_top_time_last_6_month，它们的中位数在 0 到 20 之间，分布较为分散，异常值较多。在 200 ≤ 量级 < 1000 的图中，特征如 trans_days_interval_filter 和 trans_days_interval 的中位数在 100 到 200 之间，分布较为集中，但异常值依然存在。

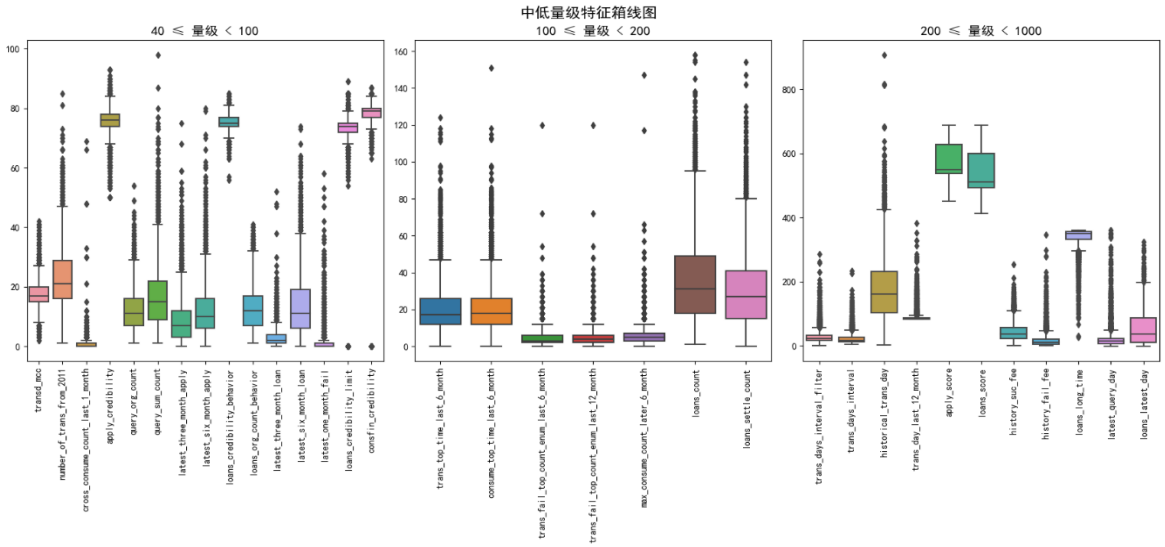


图 13 中低量级特征箱线图

中高量级特征箱线图展示了量级 1000 ≤ 量级 < 2500 的图中（图 14），特征

如 first_transaction_time_year 和 latest_query_time_year 的中位数在 2011 到 2018 之间，分布集中，异常值较少。在 2500 ≤ 量级 < 10000 的特征，如 first_transaction_day 和 loans_credit_limit，中位数在 500 到 2500 之间，分布集中，异常值较少。在 10000 ≤ 量级 < 25000 的图中，特征如 avg_price_last_12_month 和 loans_max_limit 的中位数在 2500 到 5000 之间，分布集中，异常值较少。

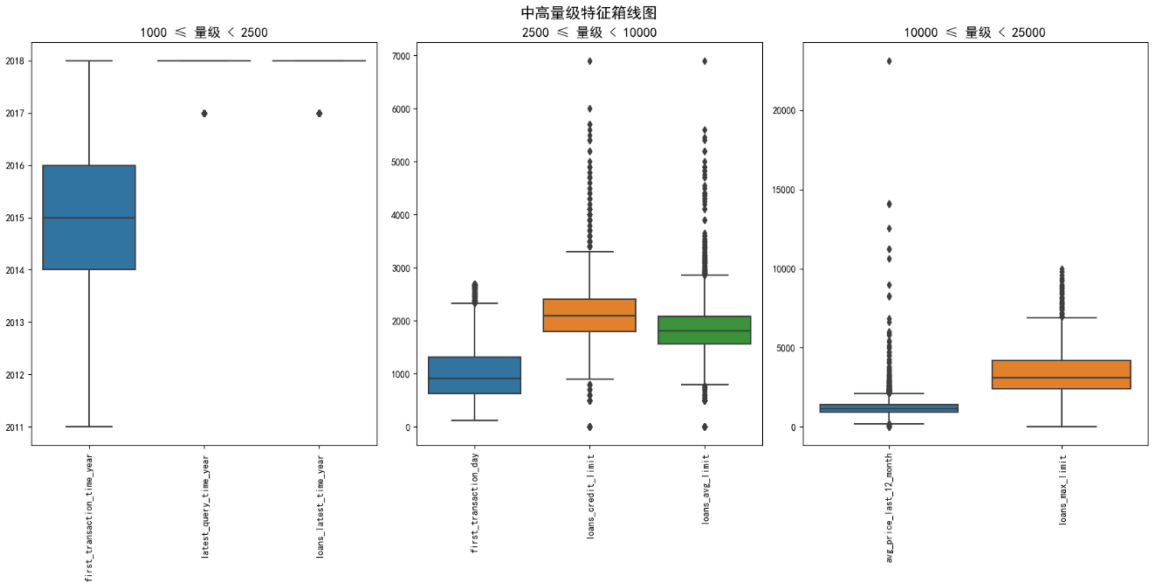


图 14 中高量级特征箱线图

最后，高量级特征箱线图展示了量级 25000 ≤ 量级 < 300000 的图中（图 15），特征如 take_amount_in_latter_12_month_lightest 和 trans_amount_increase_rate_lately 的中位数在 0 到 10000 之间，分布集中，异常值较少。在 300000 ≤ 量级 的特征，如 repayment_capability 和 historical_trans_amount，中位数在 0 到 2000000 之间，分布集中，但存在一些异常值。

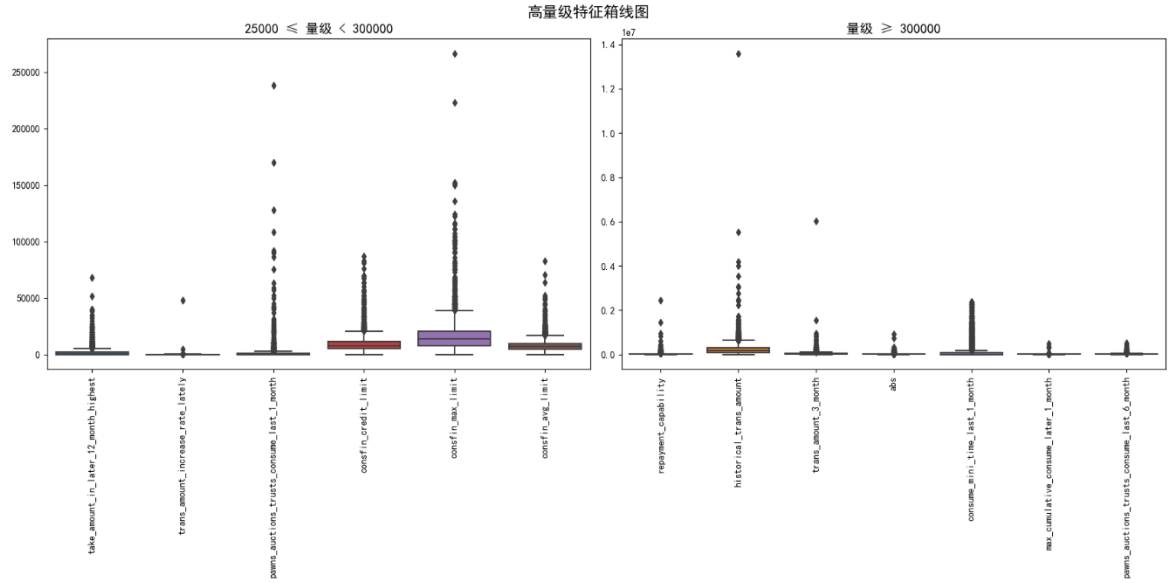


图 15 高量级特征箱线图

综上，这些箱线图揭示了数据的分布特征和异常值的存在，为数据预处理和异常值处理提供了直观的依据。通过观察这些图，可以更好地理解数据的特性，从而结合删除异常值、修正异常值、替换异常值和保留异常值这四种方法来进行异常值的处理。

## 3.4 特征选择与构造

在本次实验中，特征选择的过程采用了两种主要方法：IV 值过滤和随机森林（Random Forest，RF）重要性评分，以筛选出对模型预测能力有显著影响的特征[8]。特征构造在 3.3.3 缺失值处理中，已经对时间型特征进行了特征构造，构造出了年份、月份和星期几等三个新特征。以下是对特征选择的相关内容的叙述。

首先，通过 IV（Information Value）值过滤方法，计算了每个特征与目标变量之间的信息增益，这是一种衡量特征预测能力的指标。IV 值越高，表示该特征对目标变量的区分能力越强。在本次实验中，设定了 IV 值大于 0.05 作为筛选标准，从而得到了一组 IV 值高于 0.05 的特征集合 A。其中部分特征出现了 IV Value 为 inf 的问题，如 abs 和 max_cumulative_consume_later_1_month 等等。这类问题是由于在计算过程中遇到了数学上的问题，导致信息值变得无限大。因此在可视化特征集合 A 时没有对这些特征进行可视化，具体的特征和 IV 值如图 16 所示。

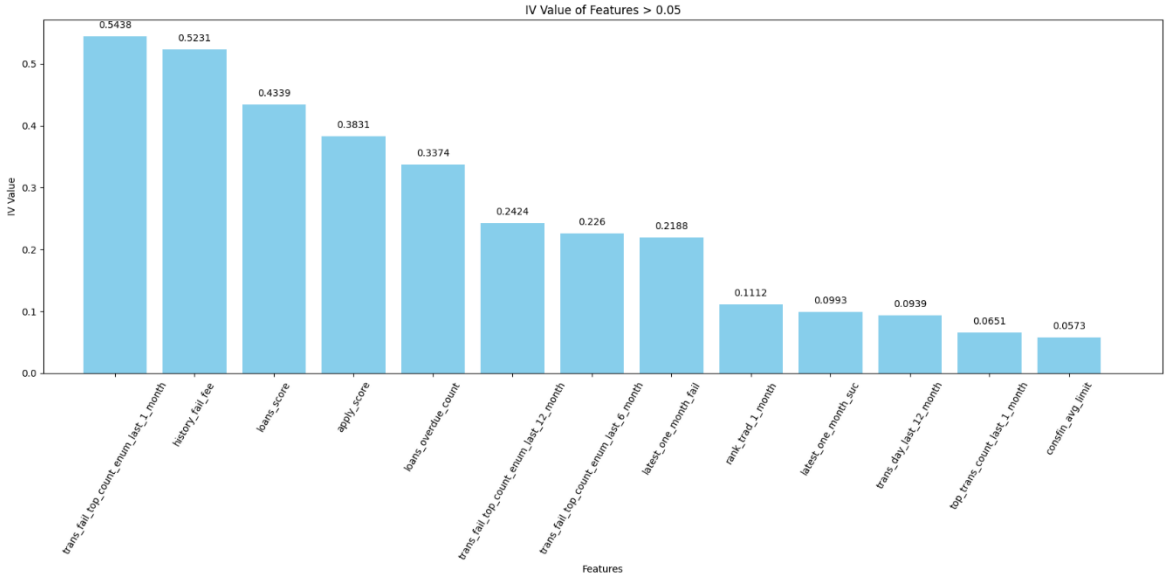


图 16 通过 IV（Information Value）值过滤出的特征

接下来，利用随机森林算法进一步筛选特征。随机森林是一种集成学习方法，通过构建多个决策树并综合其结果来提高模型的预测准确性和稳定性。在训练过程中，随机森林能够评估每个特征对模型预测能力的贡献度，即特征的重要性。根据特征的重要性评分，筛选出了前 20 个最重要的特征，形成了特征集合 B，具体如图 17 所示。

最后，为了合并两组特征集合并筛选出最终的有用特征，对集合 A 和集合 B 进行了并集操作，即选择了同时出现在两个集合中的特征。这样，得到了一个综合了 IV 值过滤和随机森林重要性评分的特征集合，共 29 个特征。通过这种方法，能够确保所选特征既具有较高的预测能力，又在模型中具有重要性，从而提高模型的整体性能。

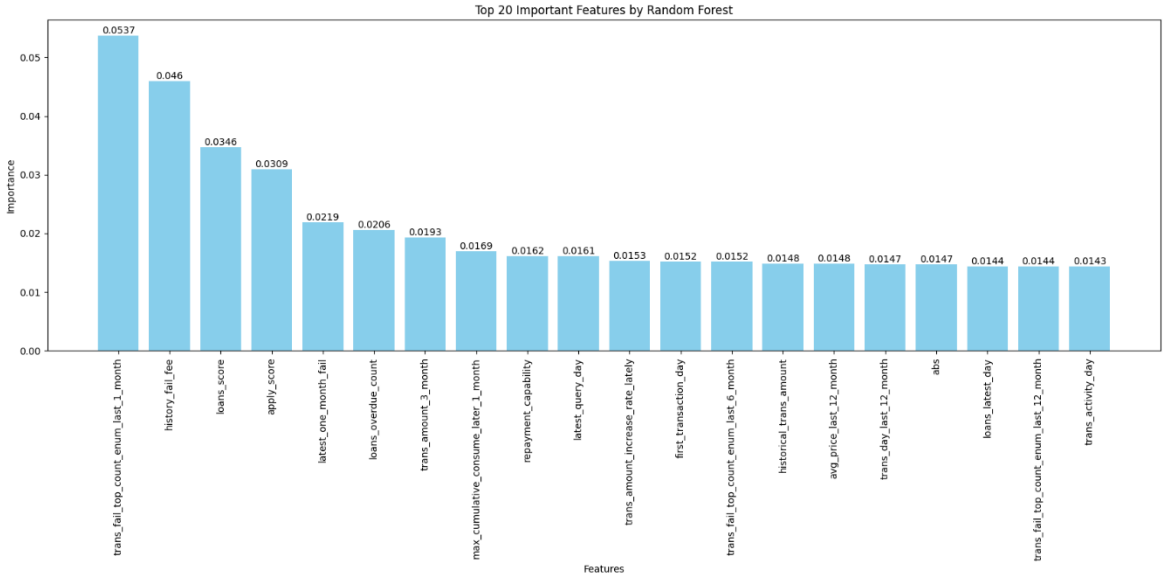


图 17 通过 RF（Random Forest）过滤出的 20 个特征

综上，在得到最终的特征集合后，将其用于构建预测模型。通过这种方式，不仅减少了模型的复杂度，避免了过拟合，还提高了模型的泛化能力。此外，特征选择的过程也有助于更好地理解数据，识别出对目标变量有实质性影响的因素，为后续的模型优化和业务决策提供了有力支持。

# 4 算法详细设计与实现

## 4.1 算法描述

本次实验使用的算法为 XGBoost，全称为 eXtreme Gradient Boosting，即极致梯度提升树，是由华盛顿大学的陈天奇开发的一个开源机器学习库。它在处理大规模数据集、特征工程以及模型性能方面表现出色，被广泛应用于数据挖掘、机器学习竞赛和实际工业场景中[9]。

XGBoost 是 Boosting 算法的其中一种，Boosting 算法的思想是将许多弱分类器集成在一起，形成一个强分类器（个体学习器间存在强依赖关系，必须串行生成的序列化方法）。XGBoost 是一种提升树模型，即它将许多树模型集成在一起，形成一个很强的分类器。其中所用到的树模型则是 CART 回归树模型。

XGBoost 的基本组成元素是：决策树。这些决策树即为“弱学习器”，它们共同组成了 XGBoost，并且这些组成 XGBoost 的决策树之间是有先后顺序的。后一棵决策树的生成会考虑前一棵决策树的预测结果，即将前一棵决策树的偏差考虑在内，使得先前决策树做错的训练样本在后续受到更多的关注，然后基于调整后的样本分布来训练下一棵决策树。XGBoost 会根据预设的迭代次数或模型性能指标（如验证集上的损失函数值）来判断是否终止迭代。当达到迭代次数上限或模型性能不再显著提升时，训练过程结束，最终得到的模型是所有弱学习器的加权组合。下面详细讲解一下 XGBoost 算法的目标函数的推导[10]。

最初的目标函数，设定第 t 个决策树的目标函数公式如下：

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right)\Omega(f_t) \tag{1}$$

根据 Boosting 的原理：第 t 棵树对样本 i 的预测值=前 t-1 棵预测树的预测值 + 第 t 棵树的预测值即：

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i) \tag{2}$$

那么最初的目标函数就可以化为：

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)\right) + \Omega(f_t) \tag{3}$$

根据二阶泰勒展开，那么就有：

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{4}$$

去掉常数项，那么就有：

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{5}$$

综上，就是 XGBoost 算法的目标函数的推导。因为目的是要最小化目标函数，所以那些常数项可以把它们暂时搁置。

## 4.2 模型设计与实现步骤

### 4.2.1 参数设置

在本次实验中，采用网格搜索（Grid Search）对 XGBoost 模型的参数进行了细致的调整，以优化模型的性能，并且选择 roc_auc，即 ROC 曲线下面积用于评估模型性能的指标[11]。网格调参法是一种系统地遍历多个参数组合的方法，它通过交叉验证来确定最佳的参数设置。

在数据分析的 3.2.1 节中，对样本数据进行了整体观察，发现样本中存在明显的类别不平衡问题。具体来说，正样本的数量为 1193 个，而负样本的数量为 3561 个，正负样本的比例大约为 1:3。为了缓解这一问题造成的影响，在训练 XGBoost 模型时将 scale_pos_weight 设置为 3，scale_pos_weight 参数用于调整正样本的权重，以平衡类别不平衡带来的影响。通过这种方式，模型在训练过程中会更加重视正样本，从而提高对正样本的识别能力。接下来，详细介绍调参过程。

首先从学习速率和迭代次数入手，这是模型训练的基础参数[12]。使用 GridSearchCV 工具，将学习速率设置为 0.1，并在 20 到 200 之间以 20 为步长搜索迭代次数。通过 5 折交叉验证，发现当迭代次数为 40 时，模型的 ROC-AUC 评分达到了 0.7931，如图 18。这表明在当前学习速率下，40 次迭代是较为合适的参数组合。

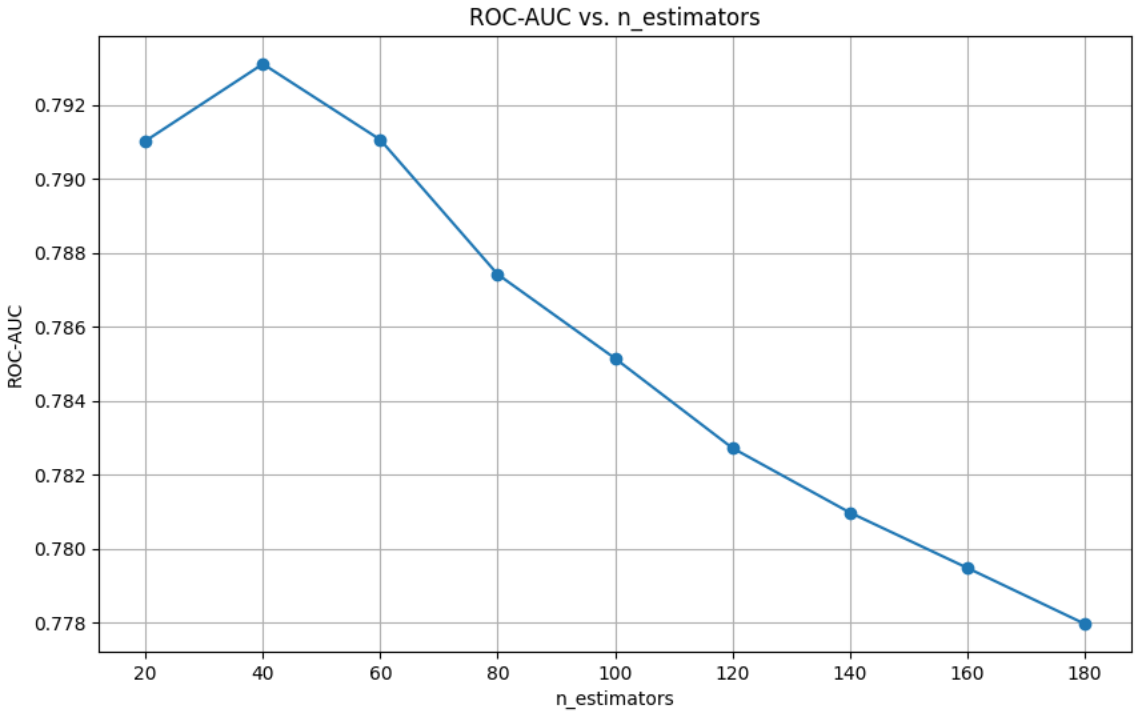


图 18 学习器数量（n_estimators）与 ROC-AUC 得分关系折线图

随后，对树的结构参数进行优化，包括最大深度和最小子节点权重。在保持学习

速率为 0.1 和迭代次数为 40 的基础上，将最大深度（max_depth）的范围设置为 3 到 10（步长为 2），最小子节点权重（min_child_weight）的范围设置为 1 到 11（步长为 2）。通过网格搜索，发现当最大深度为 3，最小子节点权重为 11 时，模型的 ROC-AUC 评分提升至 0.7996，如图 19。这表明这两个参数的调整对模型性能有显著的提升作用。

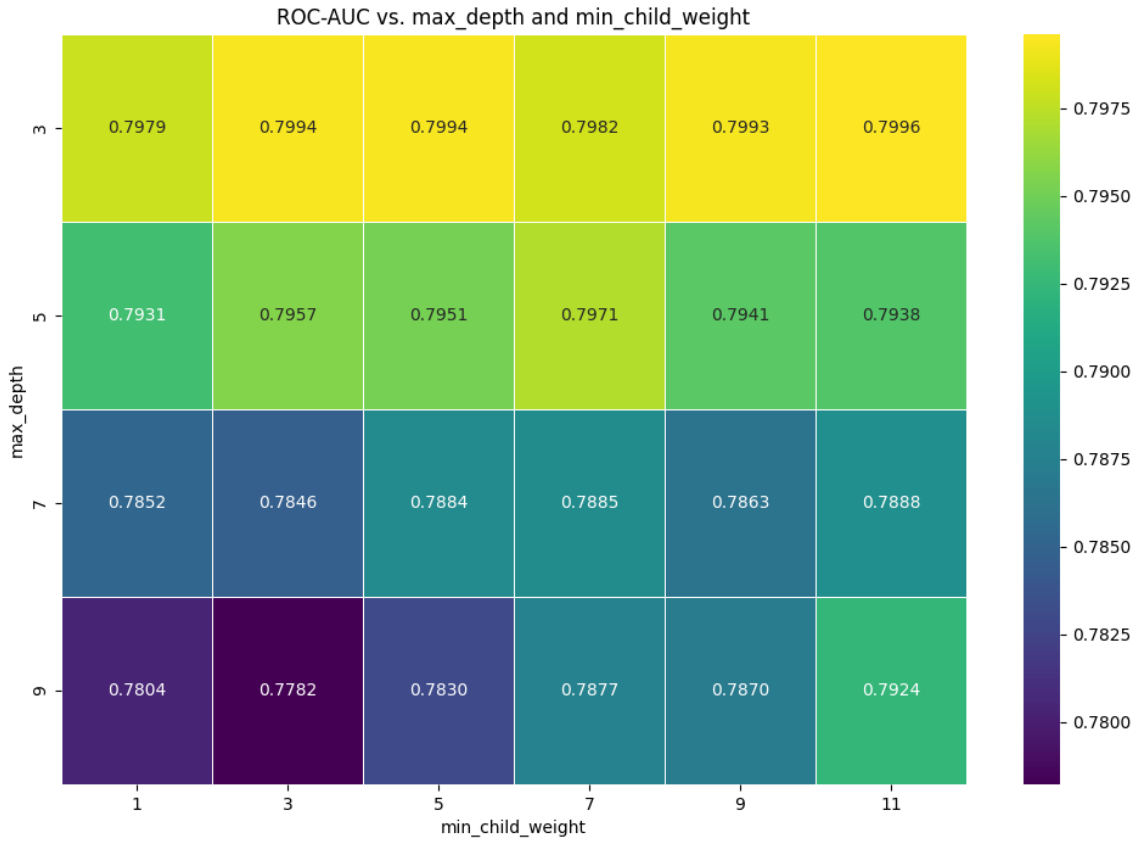


图 19 最大深度和最小子节点权重与 ROC-AUC 得分关系热力图

接下来，对 gamma 参数进行了调优。gamma 参数用于控制节点分裂所需的最小损失函数下降值，从而起到正则化的作用。在保持之前优化的参数不变的情况下，将 gamma 的取值范围设置为 0 到 0.5（步长为 0.1）。通过网格搜索，发现当 gamma 为 0.3 时，模型的性能最佳，ROC-AUC 评分达到了 0.7998，如图 20。这表明适当的 gamma 值可以进一步提升模型的性能。

在调整了主要的树结构相关参数后，进一步优化了子样本采样率（subsample）和列采样率（colsample_bytree）。子样本采样率和列采样率用于控制每棵树使用的样本比例和特征比例，从而减少过拟合。将这两个参数的取值范围分别设置为 0.5 到 1.0（步长为 0.1）。通过网格搜索，发现当子样本采样率为 0.6，列采样率为 0.6 时，模型的性能最佳，ROC-AUC 评分提升至 0.8032，如图 21。这表明调整采样率可以有效提升模型的泛化能力。

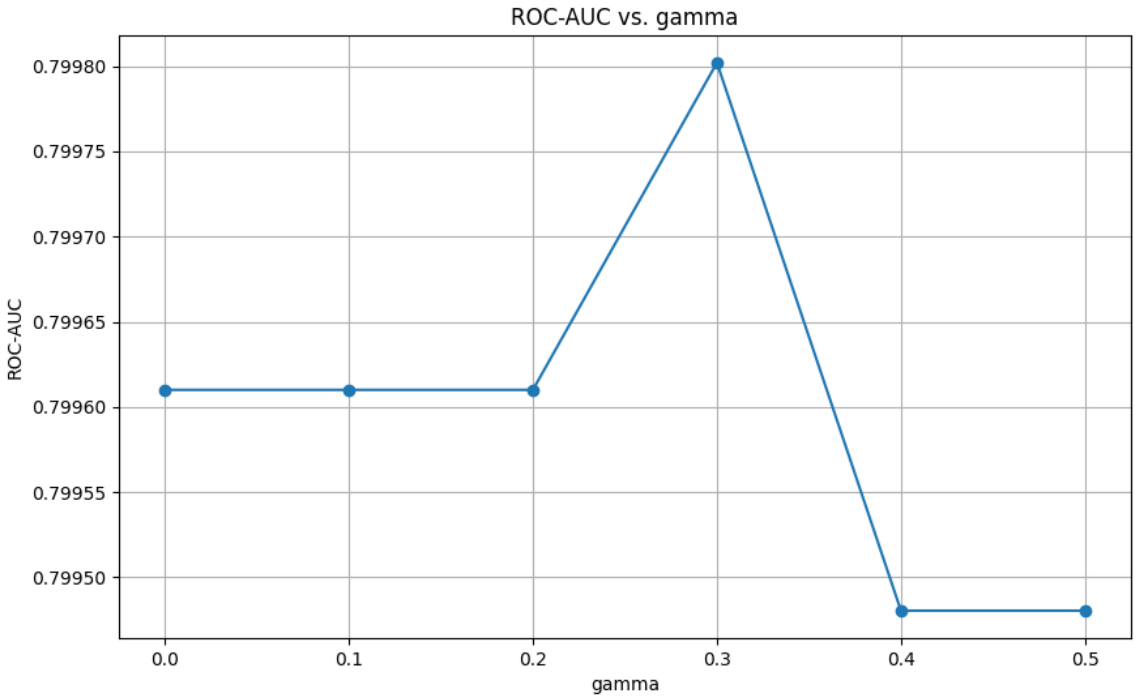


图 20 gamma 参数与 ROC-AUC 得分关系折线图

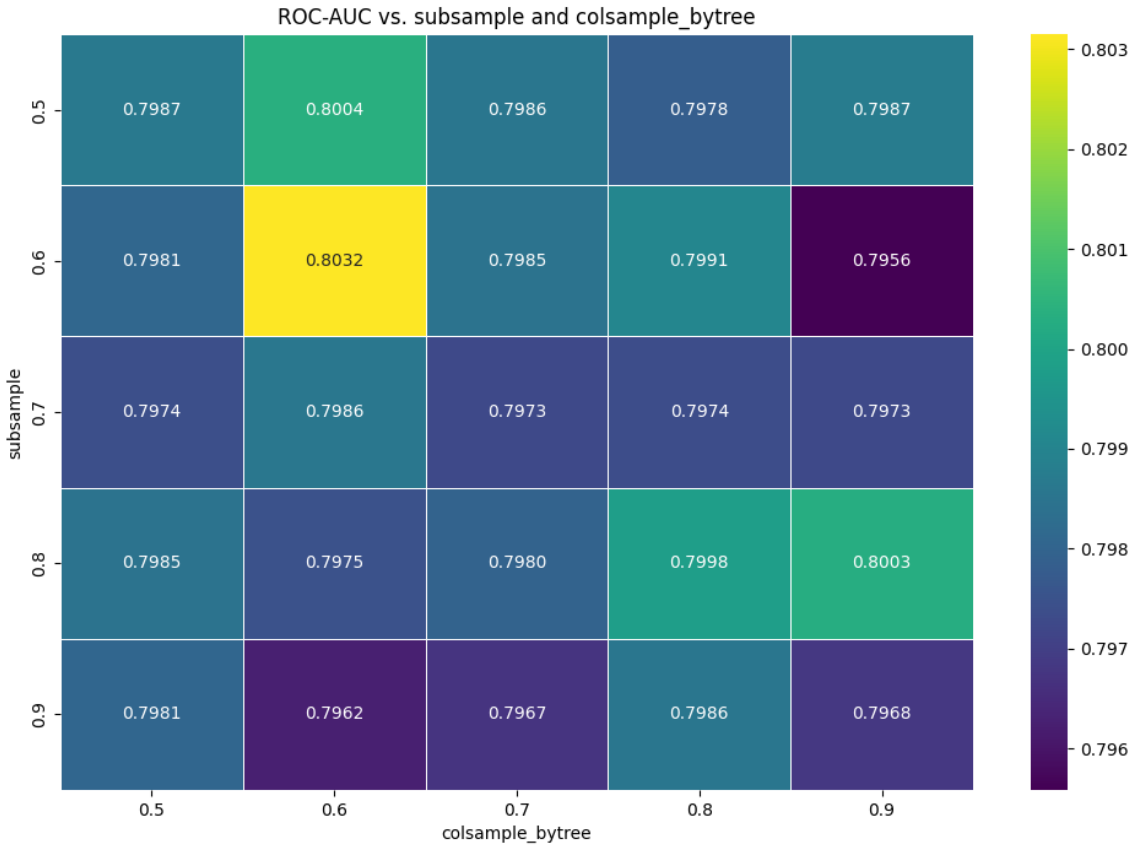


图 21 子样本采样率和列采样率与 ROC-AUC 得分关系热力图

然后，对正则化参数 reg_alpha 和 reg_lambda 进行了调优[13]。正则化参数用于控

制模型的复杂度，防止过拟合。将 reg_alpha 的取值范围设置为 1e-5 到 100（包括 0），reg_lambda 的取值范围设置为 0.2 到 1.0。通过网格搜索，发现当 reg_alpha 为 0.01，reg_lambda 为 0.8 时，模型的 ROC-AUC 评分达到了 0.8040，如图 22。这表明适当的正则化可以进一步提升模型的性能。

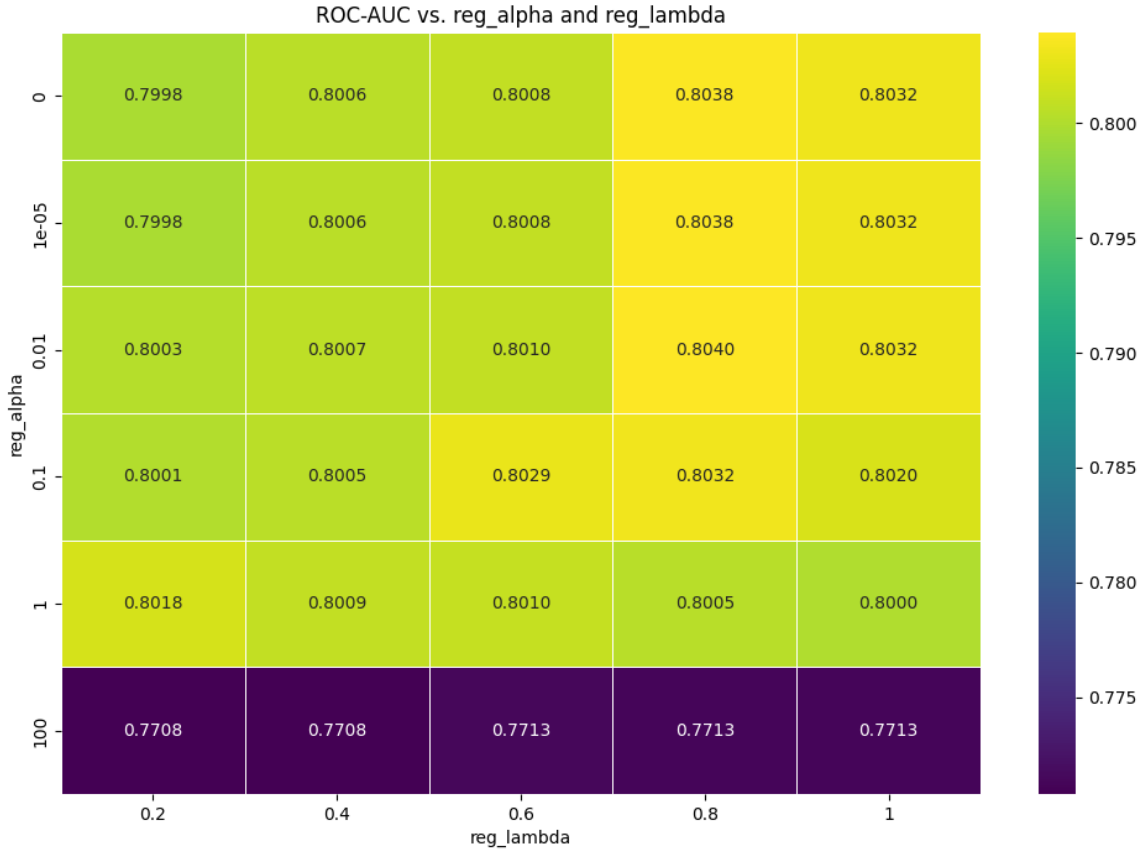


图 22 reg_alpha 和 reg_lambda 与 ROC-AUC 得分关系热力图

最后，尝试降低学习速率并重新调整迭代次数，以进一步优化模型性能。将学习速率降低到 0.01，并将迭代次数的范围设置为 160 到 200（步长为 5）。然而，结果表明当迭代次数为 195 时，模型的 ROC-AUC 评分仅为 0.7970，低于之前的学习速率为 0.1 时的最佳性能。因此，决定不采用这种降低学习速率的调整策略。

综上，通过上述一系列的参数调整，最终确定了模型的最佳参数组合，从而显著提升了模型的性能。网格调参法在这一过程中发挥了关键作用，它系统地探索了参数空间，找到了最优的参数配置。最终，模型的 ROC-AUC 评分达到了 0.8040，这表明参数调整策略是有效的。

### 4.2.2 算法实现流程

在实现 XGBoost 算法的过程中，首先进行数据准备。这包括从文件中加载经过数据预处理、特征选择和构造后的特征数据以及对应的标签数据。为了消除不同特征

之间的量纲影响并提高模型的收敛速度和预测准确性，使用 StandardScaler 对特征数据进行标准化处理[14]。

完成数据准备后，进行数据集划分，即将数据集划分为训练集和测试集。按照 70%训练集和 30%测试集的比例进行划分，以评估模型的泛化能力。

接下来是模型初始化阶段，根据经验和先前的调参结果设置 XGBoost 模型的参数。这些参数包括学习速率、迭代次数、最大深度、子节点最小权重、Gamma 值、子样本比例、列采样比例、正则化参数、目标函数、线程数、正样本权重和随机种子等。

最后，进行模型训练。在这一阶段，使用训练集数据训练 XGBoost 模型，模型通过迭代优化目标函数，学习从输入特征到输出标签的映射关系。为了达到这个目标，精确贪心算法会在所有特征 (features) 上，枚举所有可能的划分(splits)。为了更高效，该算法必须首先根据特征值对数据进行排序，以有序的方式访问数据来枚举 Gain 公式中的结构得分 (structure score) 的梯度统计 (gradient statistics)[15]。XGBoost 算法伪代码如下：

**Algorithm 1** Exact Greedy Algorithm for Split Finding

**Require:** $I$, instance set of current node
**Require:** $d$, feature dimension
gain $\leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i$, $H \leftarrow \sum_{i \in I} h_i$
**for** $k = 1$ to $m$ **do**
    $G_L \leftarrow 0$, $H_L \leftarrow 0$
    **for** $j$ in sorted($I$, by $x_{jk}$) **do**
        $G_L \leftarrow G_L + g_j$, $H_L \leftarrow H_L + h_j$
        $G_R \leftarrow G - G_L$, $H_R \leftarrow H - H_L$
        score $\leftarrow \max(\text{score}, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
    **end for**
**end for**
**Ensure:** Split with max score

## 4.3 实验平台与开发环境

实验平台与开发环境如表 4 所示：

表 4 实验平台与开发环境

| 名称 | 配置信息 |
| --- | --- |
| 操作系统 | Windows 11 |
| 开发语言 | Python 3.8.1 |
| CPU | 12th Gen Intel(R) Core(TM) i7-12700H |

| | |
|---|---|
| GPU | GeForce RTX 3050(4G) |
| 内存 | 16.0 G |

# 5 实验结果与分析

## 5.1 实验结果展示

在本次实验中，对 XGBoost 模型进行了调参，并评估了其在训练集和测试集上的性能。下表是模型的关键性能指标：

表 5 调参后的模型各个性能指标

| 性能指标 | 训练集 | 测试集 |
|---|---|---|
| accuracy | 0.7692 | 0.7155 |
| precision | 0.5275 | 0.4549 |
| recall | 0.7578 | 0.6602 |
| f1_score | 0.6220 | 0.5386 |
| auc | 0.8495 | 0.7826 |

从表格中可以看出，模型在训练集上的表现优于测试集，这可能暗示了模型存在轻微的过拟合现象。下面具体分析各个性能指标：

1） 准确率（Accuracy）：训练集的准确率为 76.92%，而测试集的准确率为 71.55%。这表明模型能够较好地识别正负样本，但在测试集上的泛化能力稍弱。

2）精确率（Precision）：训练集的精确率为 52.75%，测试集的精确率为 45.49%。精确率的下降表明模型在测试集上对正样本的识别能力有所降低。

3）召回率（Recall）：训练集的召回率为 75.78%，测试集的召回率为 66.02%。召回率的下降意味着模型在测试集上错过了更多的正样本。

4）F1 分数（F1 Score）：训练集的 F1 分数为 62.20%，测试集的 F1 分数为 53.86%。F1 分数是精确率和召回率的调和平均，其下降进一步证实了模型在测试集上的性能下降。

5）AUC（Area Under Curve）：训练集的 AUC 值为 0.8495，测试集的 AUC 值为 0.7826。AUC 值的下降表明模型在测试集上的区分能力有所减弱。

为了更直观地展示模型的性能，绘制了 ROC 曲线，如下图 23 所示：

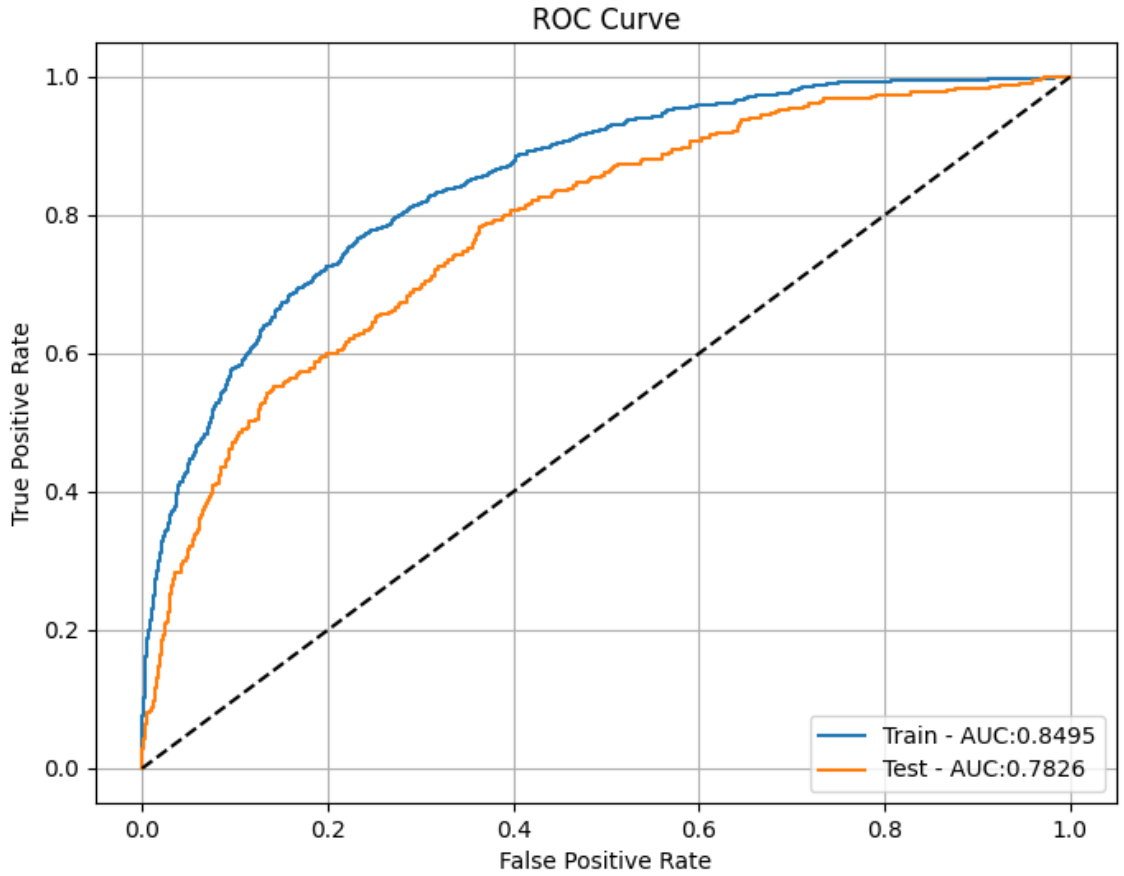


图 23 调参后模型的 ROC 曲线图

从 ROC 曲线可以看出，训练集的曲线更接近左上角，表明模型在训练集上具有更高的真阳性率和更低的假阳性率。测试集的曲线虽然也表现良好，但略逊于训练集。

综上所述，XGBoost 模型在训练集上的表现优于测试集，显示出一定的过拟合迹象。尽管如此，模型在测试集上仍然保持了相对合理的性能。未来的工作可以集中在进一步优化模型参数，以及尝试正则化技术或集成学习方法来提高模型的泛化能力。此外，可以考虑进行更多的特征工程或数据增强，以进一步提升模型性能。

## 5.2 结果分析与讨论

在本次实验中，对 XGBoost 模型进行了调参，并通过各个性能指标和 ROC 曲线，与调参前的模型性能进行了对比分析。以下是详细的结果分析与讨论（不同参数设置的比较见 4.2.1 参数设置）：

从下表 6 中可以分析得出，调参前，模型在训练集上的表现是完美的，所有指标均为 1.0000，这表明模型在训练集上完全拟合了数据。然而，在测试集上，模型的表现远不如训练集，准确率、精确率、召回率和 F1 分数均显著下降，AUC 值也相对较低。这表明模型在训练集上过拟合严重，泛化能力不足。

调参后，模型在训练集上的性能有所下降，准确率降至 0.7692，精确率降至 0.5275，召回率降至 0.7578，F1 分数降至 0.6220，AUC 值降至 0.8495。尽管如此，测试集上的性能却有所提升，召回率提升至 0.6602，F1 分数提升至 0.5386，AUC 值提升至 0.7826。这表明调参后的模型在测试集上的泛化能力得到了改善。

表 6 调参前与调参后的模型各个性能指标

| 性能指标 | 调参前 | | 调参后 | |
|---|---|---|---|---|
| | 训练集 | 测试集 | 训练集 | 测试集 |
| accuracy | 1.0000 | 0.7582 | 0.7692 | 0.7155 |
| precision | 1.0000 | 0.5223 | 0.5275 | 0.4549 |
| recall | 1.0000 | 0.4568 | 0.7578 | 0.6602 |
| f1_score | 1.0000 | 0.4874 | 0.6220 | 0.5386 |
| auc | 1.0000 | 0.7442 | 0.8495 | 0.7826 |

通过绘制 ROC 曲线（如图 24），可以直观地评估模型在不同阈值下的表现。ROC 曲线展示了模型的真阳性率（True Positive Rate）与假阳性率（False Positive Rate）之间的关系，而 AUC 值则量化了模型的整体性能，其值越接近 1 表示模型性能越好。

从图中可以看出，使用默认参数的模型在训练集上达到了完美的 AUC 值 1.0000，但在测试集上的 AUC 值下降到了 0.7521，这表明模型可能在训练集上过拟合了。相比之下，经过调参后的模型在训练集上的 AUC 值略有下降，为 0.8495，而在测试集上的 AUC 值提升至 0.7826，显示出更好的泛化能力。此外，从 ROC 曲线的形状来看，调参后的模型在测试集上的曲线更接近左上角，这进一步证实了其区分正负样本的能力有所增强。尽管调参后的模型在训练集上的性能有所下降，但这种权衡是值得的，因为它提高了模型在未见数据上的表现。然而，尽管调参后的模型在测试集上有所改进，但仍然存在提升空间。特别是在精确率和 F1 分数方面，模型还有待进一步优化。未来的工作可能包括继续调整参数、进行特征工程或尝试不同的模型架构，以进一步提升模型性能。总结来说，适当的参数调整对于提高 XGBoost 模型的泛化能力至关重要。通过对比分析，发现调参后的模型在测试集上的表现优于默认参数模型，这表明调参策略是有效的。尽管如此，模型仍有改进的余地，需要在未来的研究中继续探索和优化。

最后绘制了前 20 个最重要的特征的特征重要性直方图，如图 25。从图中可以看出，“trans_fail_top_count_enum_last_1_month”是最重要的特征，其重要性得分接近 0.12，远高于其他特征。紧随其后的是“loans_overdue_count”和“loans_score”，这两个特征的重要性得分也相对较高，但与第一个特征相比仍有较大差距。其他特征的

重要性得分逐渐递减，显示出它们对模型预测的影响较小。

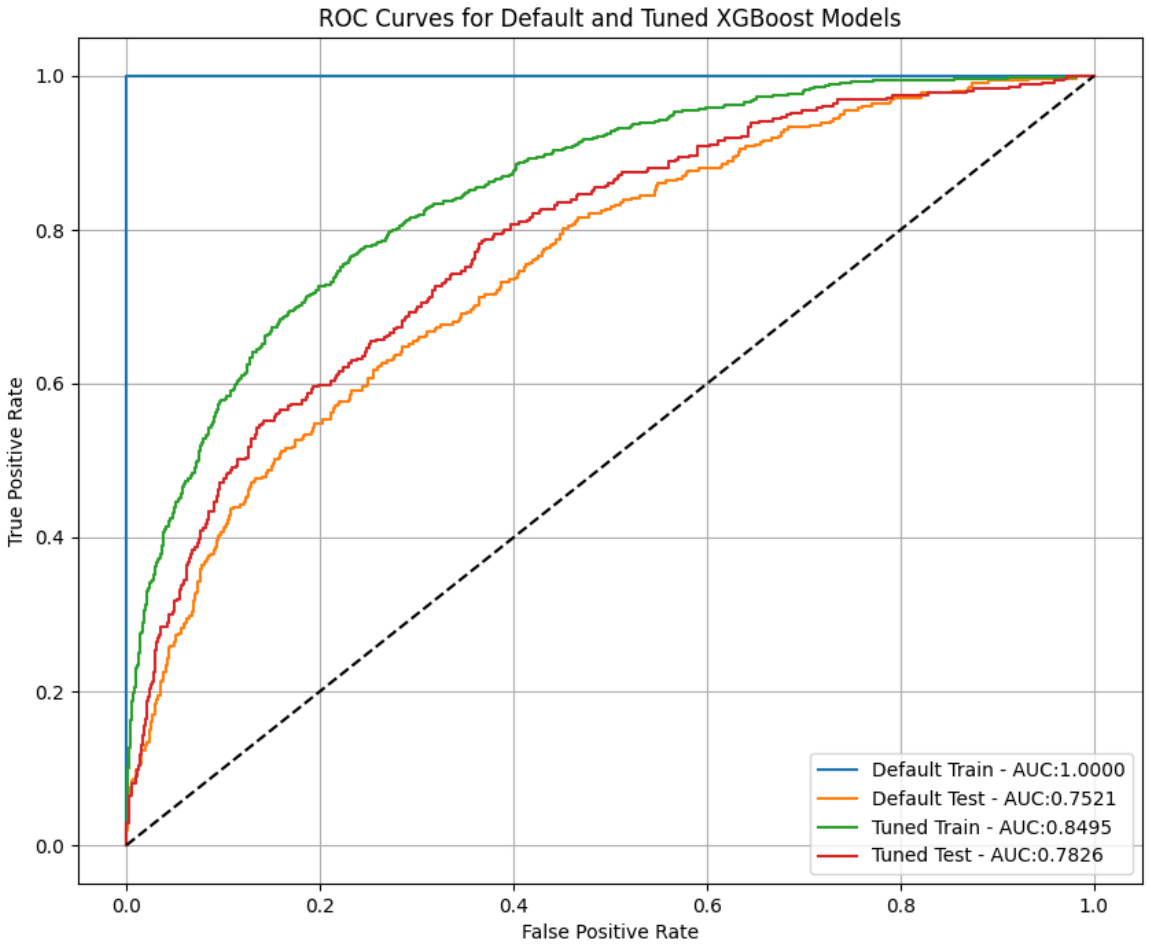


图 24 调参前与调参后模型的 ROC 曲线图

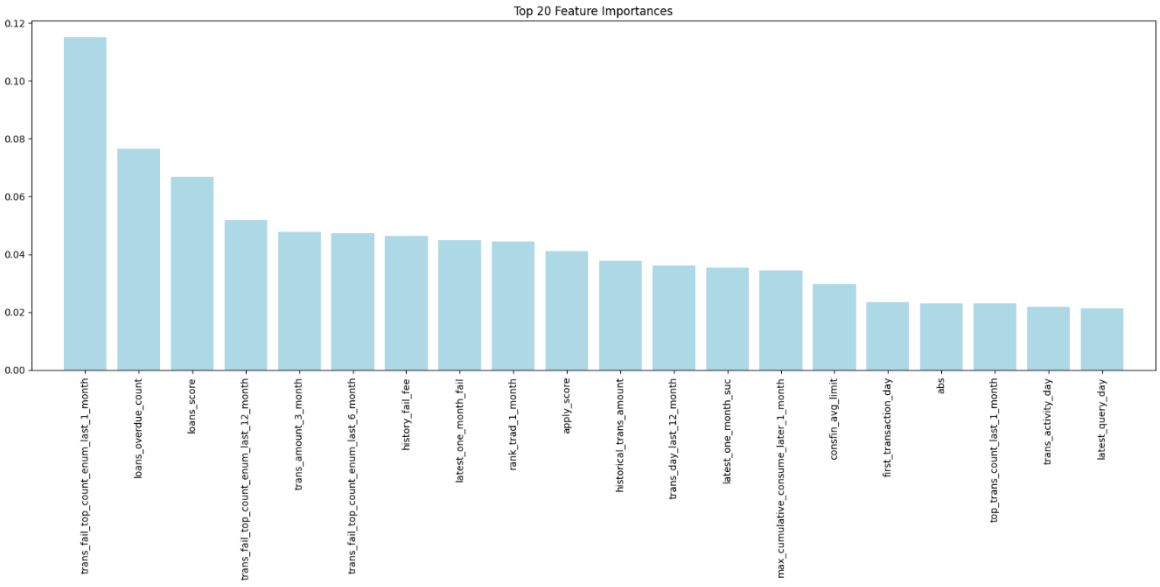


图 25 调参前与调参后模型的 ROC 曲线图

# 6 总结与展望

## 6.1 工作总结

本次研究聚焦于贷款逾期预测问题，通过构建基于 XGBoost 算法的预测模型，旨在为金融机构提供一种有效的信贷风险评估工具。在研究过程中，首先对贷款逾期预测的背景和意义进行了深入分析，明确了该问题在金融风险管理中的重要性。通过对国内外相关研究的综述，总结了当前领域内的研究进展，并确定了本研究的理论基础和方法路径。

在数据处理阶段，对数据进行了详细的预处理，包括数据清洗、缺失值处理、异常值处理以及特征选择与构造。通过 IV 值过滤和随机森林重要性评分相结合的方法，筛选出了对模型预测能力有显著影响的特征，为后续的模型训练奠定了坚实基础。在模型设计与实现阶段，详细介绍了 XGBoost 算法的基本原理，并通过网格搜索法对模型参数进行了细致的调整，优化了模型的性能。最终，通过一系列评估指标对模型的性能进行了全面分析，验证了模型的有效性和泛化能力。

总体而言，本次研究成功构建了一个基于 XGBoost 的贷款逾期预测模型，并通过实验验证了其在处理不平衡数据和复杂特征时的优势。研究不仅为金融机构提供了可靠的信贷风险评估工具，也为金融科技领域的进一步研究提供了新的思路和方法。通过本研究，展示了机器学习技术在金融风险管理中的应用潜力，推动了金融科技在信贷风险评估领域的创新和应用。

## 6.2 存在不足和展望

尽管本次研究取得了一定的成果，但仍存在一些不足之处。首先，模型在测试集上的表现虽然有所提升，但仍显示出一定的过拟合迹象，尤其是在精确率和 F1 分数方面仍有改进空间。其次，虽然通过特征选择和参数优化提高了模型的性能，但在处理高维数据时，模型的训练时间较长，计算效率有待进一步提高。此外，由于数据集的局限性，模型的泛化能力可能受到一定影响，尤其是在面对不同地区或不同类型的金融机构时。

针对上述不足，未来的研究可以从以下几个方向进行改进和拓展。首先，可以尝试引入更多的特征工程方法，如特征交叉、降维技术等，以进一步提升模型的性能和泛化能力。其次，可以探索使用更先进的集成学习算法或深度学习模型，以更好地处理复杂的金融数据。此外，结合领域知识，对模型的可解释性进行深入研究，将有助于金融机构更好地理解和应用模型结果。最后，扩大数据集的规模和多样性，进行跨地区、跨行业的模型验证，将进一步提升模型的实用性和可靠性。

在未来的研究中，还将关注模型的实时性和动态性，探索如何在金融市场的快速变化中实时更新模型，以适应不断变化的风险环境。同时，结合监管政策的变化，进一步优化模型的合规性和稳定性，为金融机构提供更加全面和可靠的信贷风险管理解决方案。

# 参考文献


[1] 张利斌,吴宗文.基于 XGBoost 机器学习模型的信用评分卡与基于逻辑回归模型的对比[J].中南民族大学学报(自然科学版),2023,42(06):846-852.DOI:10.20056/j.cnki.ZNMDZK.20230616.

[2] 贾颖,赵峰,李博,等.贝叶斯优化的 XGBoost 信用风险评估模型[J].计算机工程与应用,2023,59(20):283-294.

[3] 胡越,王桑原,覃浩恒,等.基于双重 XGBoost 模型的农产品期货波动率预测——以玉米期货为例[J].系统管理学报,2023,32(02):332-342.

[4] 刘洪波,刘俊莹.我国房地产企业的信用风险评价研究[J].征信,2023,41(03):66-72.

[5] 顾天下,刘勤明.面向高维和不平衡数据的供应链金融信用评价[J].计算机应用研究,2022,39(11):3396-3401.DOI:10.19734/j.issn.1 001-3695.2022.04.0174.

[6] 顾天下,刘勤明,叶春明.基于 BO-XGBoost 与集成学习方法的供应链金融信用评价研究[J].上海理工大学学报,2023,45(01):95-102.DOI:10.13255/j.cnki.jusst.20211027005.

[7] 肖艳丽,向有涛.金融市场极端风险状态预测模型及其应用[J].金融发展研究,2022,(03):8-17.DOI:10.19647/j.cnki.37-1462/f.2022.03.002.

[8] 王言,周绍妮,石凯.国有企业并购风险预警及其影响因素研究——基于数据挖掘和 XGBoost 算法的分析[J].大连理工大学学报(社会科学版),2021,42(03):46-57.DOI:10.19525/j.issn1008-407x.2021.03.006.

[9] 张雷,王家琪,费职友,等.基于 RF-SMOTE-XGboost 下的银行用户个人信用风险评估模型[J].现代电子技术,2020,43(16):76-81.DOI:10.16652/j.issn.1004-373x.2020.16.020.

[10] 陈荣荣,詹国华,李志华.基于 XGBoost 算法模型的信用卡交易欺诈预测研究[J].计算机应用研究,2020,37(S1):111-112+115.

[11] 周永圣,崔佳丽,周琳云,等.基于改进的随机森林模型的个人信用风险评估研究[J].征信,2020,38(01):28-32.

[12] 夏利宇,张勇,鲁强,等.结合 XGBoost 算法和 Logistic 回归的信用评级方法[J].征信,2019,37(11):56-59.

[13] 刘志惠,黄志刚,谢合亮.大数据风控有效吗?——基于统计评分卡与机器学习模型的对比分

析[J].统计与信息论坛,2019,34(09):18-26.
[14]王燕,郭元凯.改进的 XGBoost 模型在股票预测中的应用[J].计算机工程与应用,2019,55(20):202-207.
[15]廖文雄,曾碧,梁天恺,等.面向高维数据的个人信贷风险评估方法[J].计算机工程与应用,2020,56(04):219-224.