《数据挖掘》 课程论文模板

（2025-2026 学年第 1 学期）

题目：基于多模型融合的个人信贷违约风险预测研究

学生姓名： 阙咏鑫、许桐恺、刘俊宏

学　　号： 42311056、42311038、42311180

提交日期：　2025 年 12 月 14 日

学生签名（手写）：

上课时间	周一晚	QQ/手机	2499185166
所在学院	计算机与人工智能学院	专　业	计算机科学与技术
教师评语：			
本论文成绩评定： _____分			

基于多模型融合的个人信贷违约风险预测研究

许桐恺* 阙咏鑫* 刘俊宏*

西南财经大学 计算机科学与技术，成都 610000

【摘　要】 本文提出一套融合深度特征工程与多模型集成的端到端违约风险预测框架。通过系统预处理与多维特征构建，重点引入群体统计量与高阶共现特征刻画个体差异。并基于 Optuna 对多个 GBDT 模型进行超参数优化，以逻辑回归为元模型构建 Stacking 融合，同时引入单 Logistic Regression 作为对照基线。本文还利用 SHAP 值对三个基学习器进行全局特征贡献分析，从信用等级、时间特征以及目标编码特征等维度解释模型的决策依据。实验结果表明，融合模型在本地验证集上的 AUC 达到 0.7532，显著优于单一 GBDT 模型（约 0.745）及基线模型（约 0.727）；在天池平台线上提交中取得约 0.7400 的 AUC 成绩，在 25805 只团队中排名第 48 名。

【关键词】 **信贷风控**；**违约预测**；**特征工程**；**梯度提升树**；**模型融合**；Stacking；SHAP

一　引 言

(一)　研究背景与意义

近年来，随着数字技术与金融服务的深度融合，普惠金融 (Inclusive Finance) 在全球范围内以前所未有的速度加速发展。作为其核心业务之一，个人信贷服务极大地拓宽了金融服务的覆盖面，为个人消费者和小型企业主提供了重要的资金支持，有效促进了消费升级和实体经济的活力。然而，业务规模的迅速扩张也伴随着信用风险的显著增加。个人信贷违约事件不仅会给放贷机构带来直接的经济损失，还可能引发连锁反应，影响金融市场的稳定。因此，建立一套科学、精准、高效的个人信贷违约风险预测体系，已成为金融机构风险管理部门乃至整个金融行业亟待解决的核心问题。

传统的信贷风控主要依赖于基于专家规则和以逻辑回归为代表的线性模型构建的信用评分卡[1]。这类方法虽然具有良好的解释性和稳定性，但在处理高维度、非线性、强交互的现代信贷数据时，其预测能力的局限性日益凸显。随着大数据和人工智能技术的进步，以梯度提升决策树 (Gradient Boosting Decision Tree, GBDT) 为代表的机器学习模型，如 XGBoost、LightGBM 和 CatBoost，因其强大的非线性拟合能力和出色的预测性能，已在信贷风控领域展现出巨大的应用潜力。如何充分利用这些先进模型，深度挖掘数据价值，构建性能更优越的预测模型，是本研究的出发点和核心目标。

（二） 相关工作与文献综述

信贷违约预测作为金融风控领域的经典问题，一直是学术界和工业界的研究热点。早期的研究主要集中于统计模型，其中 Altman 提出的 Z-Score 模型[2] 和以逻辑回归[3] 为基础的评分卡模型是里程碑式的成果，它们至今仍在许多金融机构的基准系统中使用。这些模型的优点在于结构简单、结果易于解释，但其线性的假设限制了它们从复杂数据中学习的能力。

进入 21 世纪，机器学习方法被广泛引入该领域。支持向量机 (SVM)[4]、神经网络 (Neural Networks)[5] 和随机森林 (Random Forest)[6] 等模型在多个数据集上被证明优于传统统计模型。近年来，以 Friedman 提出的梯度提升机[7] 为基础的集成学习算法，特别是其高效实现 XGBoost[8]、LightGBM[9] 和 CatBoost[10]，已成为处理表格类数据预测任务的“利器”。大量研究和数据竞赛实践（如 Kaggle）表明，这些 GBDT 模型在信贷风控任务中通常能取得顶尖的性能。

然而，先进的模型并不能自动保证最佳的结果。现代机器学习研究的共识是“数据和特征决定了机器学习的上限，而模型和算法只是逼近这个上限而已”[11]。因此，特征工程，即从原始数据中创造出对模型有价值的输入变量，被认为是提升模型性能的关键。同时，为了进一步突破单一模型的性能瓶颈，模型融合 (Model Ensembling) 技术，特别是 Stacking 堆叠融合[12]，被证明是行之有效的

策略。它通过训练一个“元模型”来智能地组合多个基模型的预测，从而获得比任何单一模型都更好的泛化能力。

（三） 本文主要工作与贡献

尽管已有大量研究将机器学习应用于信贷风控，但如何系统性地结合深度特征工程、自动化模型优化与高级模型融合策略，构建一个端到端的、可复现的高性能建模框架，仍是值得深入探索的方向。基于此，本文的主要工作与贡献如下：

- 构建了多维度的深度特征工程体系：本文不仅处理了基础特征，还重点设计并实现了高阶交叉特征，特别是群体统计特征（如均值、标准差、分位数等）和多阶共现次数特征，旨在从数据的交互中捕捉更深层次、具有业务含义的风险模式。
- 实践了科学的模型优化流程：采用基于 XGBoost 特征重要性的过滤法进行了高效的特征选择，并引入 Optuna 自动化调优框架，对 XGBoost、LightGBM、CatBoost 三个 GBDT 模型的超参数进行了系统性的贝叶斯优化，以充分挖掘各自的性能上限。
- 设计了多模型融合与对照实验体系：在三个 GBDT 基模型的基础上，构建了简单平均和 Stacking 两种融合策略，以逻辑回归作为元模型对基学习器输出进行二次建模；同时引入传统 Logistic Regression 作为基线模型，在相同特征与评估框架下进行对照实验，量化非线性集成模型带来的性能提升。
- 开展了基于 SHAP 的可解释性分析：分别对 XGBoost、LightGBM 与 CatBoost 进行 SHAP 特征贡献分析，从信用等级、时间维度、目标编码特征以及匿名行为特征等角度给出模型决策的可视化解释，为金融风控场景下的模型落地提供了支持。

- 提供了可复现的高性能基线：本文提出的端到端建模框架，在本地验证集上取得了 0.7532 的 AUC，在公开竞赛平台上获得约 0.7400 的线上成绩，为相关领域的研究和实践提供了强有力的性能基准和方法论参考。

二 实验准备

（一） 数据初步分析与预处理

本研究的数据分析与建模工作首先开展了系统性的探索性数据分析与规范化的数据预处理操作，旨在深入理解变量内涵及其统计分布特征，识别并纠正缺失、异常与格式不一致等数据质量问题，从而构建一个结构完备、语义明确、可供模型直接调用的高质量数据集，为后续特征工程与建模优化提供坚实的数据基础。

1 数据集概览与变量解析

本项目所使用的数据集包含约 80 万条贷款记录，涵盖 47 个原始特征，全面表征借款人的人口统计属性、财务状况、信用历史以及贷款申请的核心信息。根据变量的业务属性与结构特征，可将其划分为如下几类：

表 1 变量分类与说明

类别	变量	描述
标识信息	id	贷款记录唯一标识符
贷款基本属性	loanAmnt, term, interestRate, installment, grade, subGrade	贷款金额、期限、利率、月供、信用等级
借款人信息	employmentTitle, employmentLength, homeOwnership, annualIncome, verificationStatus	就业信息、房产状况、年收入及验证状态
贷款目的与状态	purpose, title, initialListStatus, applicationType, policyCode	贷款用途、申请类型及列表状态等
地理位置变量	postCode, regionCode	借款人所在地区信息
信用行为与历史	dti, delinquency_2years, ficoRangeLow, ficoRangeHigh, openAcc, pubRec, pubRecBankruptcies, revolBal, revolUtil, totalAcc, issueDate, earliesCreditLine	债务水平、信用账户状况及信用记录
匿名化变量	n0–n14	经脱敏处理的行为特征变量
目标变量	isDefault	贷款违约标记（1 表示违约，0 表示正常还款）

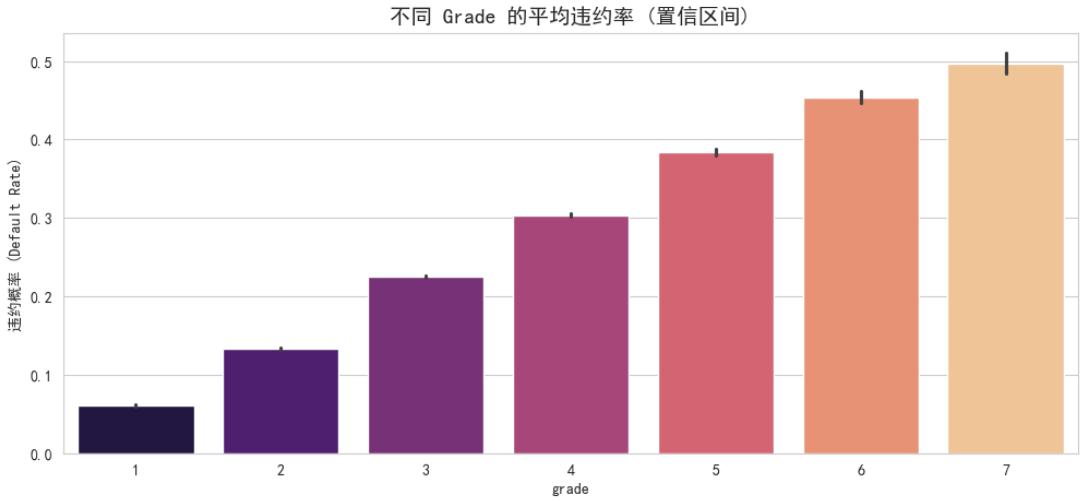


图 1 不同 GRADE 违约率

如图1所示，不同信用等级（grade）的违约率存在显著差异，较低等级（如 '1'）的违约风险明显高于较高等级（如 '6'），这验证了信用评级在风险评估中的重要作用。Grade 很有可能成为一个强特征。

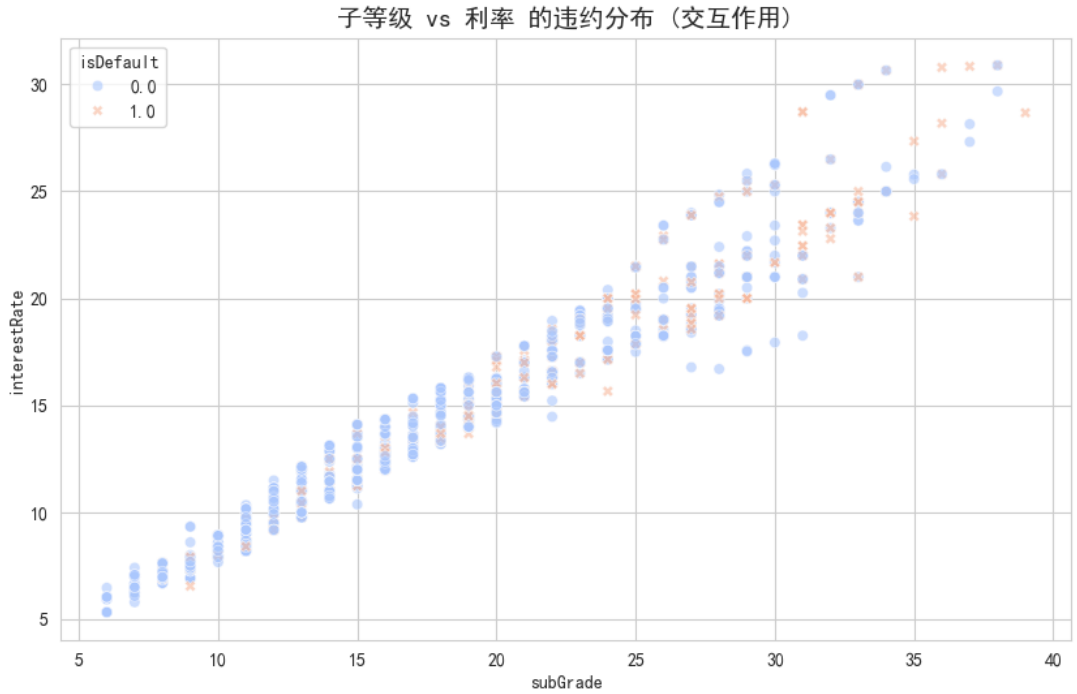


图 2 子等级 vs 利率的违约分布

如图2我们可以发现，违约样本主要集中在右上角（等级高且利率高），说明这两个特征正如预

期工作。

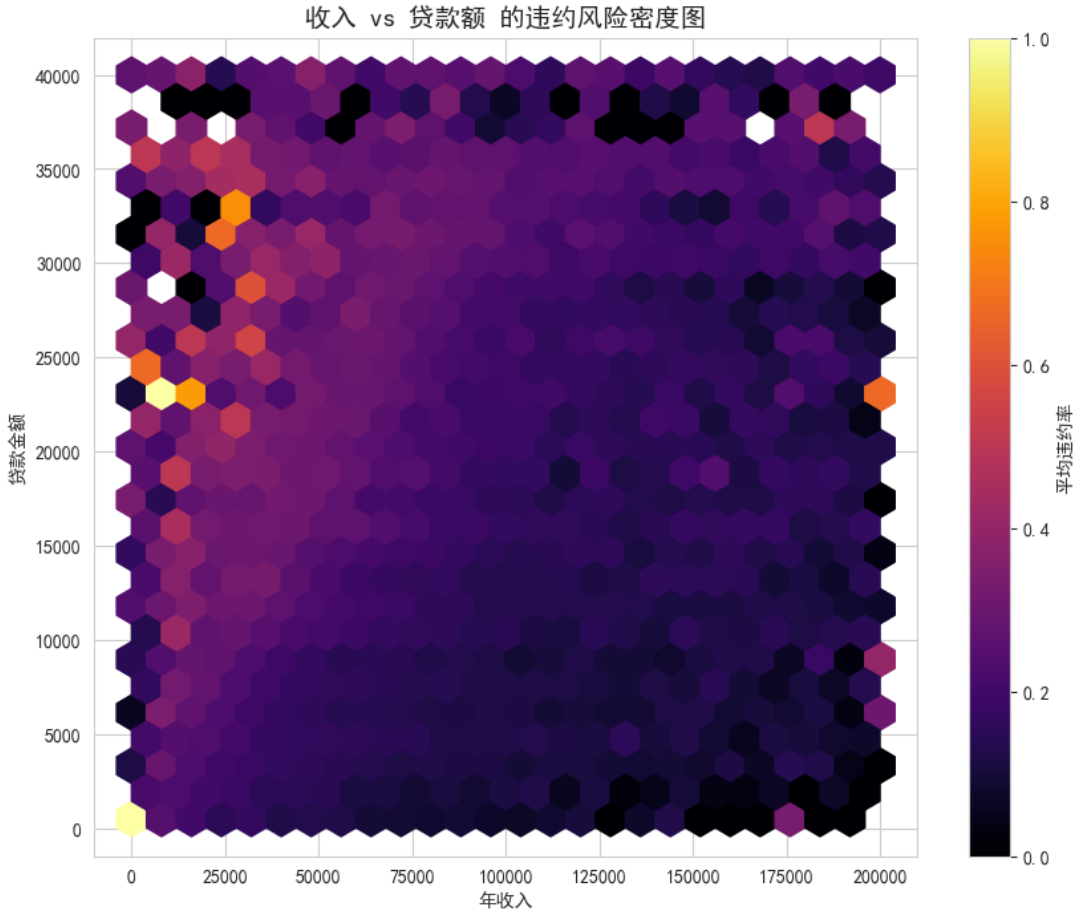


图 3 收入与贷款额的违约风险密度图

如图3所示，低收入且高贷款额的借款人违约风险显著增加，这与债务收入比（dti）的风险评估逻辑一致。为后续特征工程提供了重要参考。

（1）缺失值处理 对于缺失值的处理，通常有使用均值/中位数填充、插值法以及基于模型的预测填充等多种策略。但经过我们讨论研究后，发现在金融信贷领域，缺失值的出现可能本身就携带了一部分信息，我们采用了如下统一的填充方案：

- **数值型变量填充**：针对所有数值型特征（含匿名变量），采用固定值 -999 进行填充。此类极端值设计可被树模型有效识别为“缺失模式”的代理变量，便于建模捕捉其潜在信息。
- **分类变量填充**：对于类别变量，统一使用占位符 MISSING 进行填充。

于是，通过这样的处理，我们把“缺失”也作为一种信息纳入了模型的学习范畴。图 4 展示了数据集中各特征缺失值的百分比分布情况，可以看出部分变量存在较高比例的缺失，提示这些变量在后续建模中可能具有重要作用。

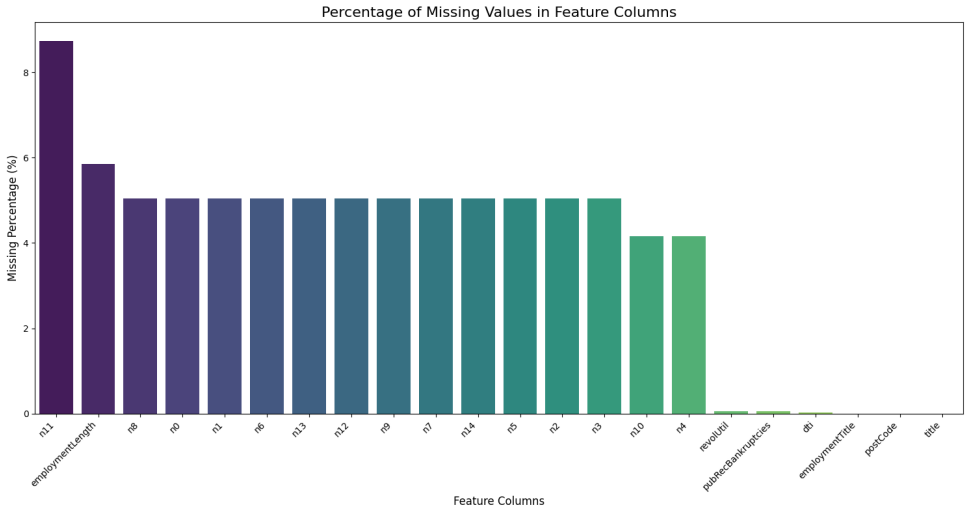


图 4 数据集中特征缺失值百分比分布图

（2）数据类型规范与变量编码

- **时间变量标准化**：将 `issueDate` 与 `earliesCreditLine` 自字符串格式转换为 `datetime64` 类型，便于后续构造衍生时间变量（如账户年龄、放款月份等）。

- **有序变量编码**：

  – `grade`：将信用等级 `'A'–'G'` 编码为 1–7，保留其单调顺序结构；

  – `subGrade`：通过组合编码公式（如 主等级值 `Œ5 + 子等级编号`），实现主等级与子等级顺序的连续性表达；

  – `employmentLength`：对就业年限中的模糊表达（如 `<1 year`, `10+ years`）进行规则解析与清洗，统一转换为浮点数形式（`<1 year` 计作 `0.5`，`>10 year` 计作 `12`），增强其数值可解释性与计算能力。

- **目标编码**：对于名义类别特征（Nominal Categorical Features），尤其是唯一值众多的高基数特征（如 `postCode`, `employmentTitle`, `title`），简单的独热编码会导致维度灾难。为此，我们采用了目标编码（Target Encoding）策略，其核心思想是用类别对应的目标变量 `isDefault` 的均值（即该类别的平均违约率）来代替类别本身，生成与目标高度相关的数值特征。具体实施方法如下：
  - **外部多种子循环**：为保证编码稳定性，交叉验证过程使用多个不同随机种子重复多次。
  - **内部 K-Fold 划分**：在每个种子循环内，将训练集划分为 K 个互斥子集（折），取 $K = 5$。
  - **折外计算**：对每一折进行编码时，使用其余 $K - 1$ 折数据计算类别均值，确保训练样本的编码值不依赖其自身目标标签，防止数据泄露。
  - **验证集与测试集处理**：使用训练集上计算的全局目标均值映射转换验证集和测试集。对于验证集中出现的新类别，填充训练集目标变量的全局均值。

目标编码的数学原理如下：设第 j 个类别特征为 x_j，其取值集合为 $\mathcal{C}_j$，目标变量为 y，全局样本均值为：

$$\mu = \mathbb{E}[y]$$

在训练集上采用 K 折交叉验证。第 k 折中，训练集记为 $D_{\text{train}}^{(k)}$，验证集为 $D_{\text{val}}^{(k)}$。对每个类别 $c \in \mathcal{C}_j$，在第 k 折的训练集上计算目标均值：

$$\mu_j^{(k)}(c) = \frac{\sum_{i \in D_{\text{train}}^{(k)}} \mathbf{1}(x_j^{(i)} = c) \cdot y^{(i)}}{\sum_{i \in D_{\text{train}}^{(k)}} \mathbf{1}(x_j^{(i)} = c)}$$

将该编码值赋给验证集中所有对应类别值的样本：

$$x_{j,\text{target\_mean}}^{(i)} = \begin{cases} \mu_j^{(k)}(x_j^{(i)}), & \text{if } x_j^{(i)} \in \mathcal{C}_j^{(k)} \\ \mu, & \text{if } x_j^{(i)} \notin \mathcal{C}_j^{(k)} \end{cases} \quad \text{for } i \in D_{\text{val}}^{(k)}$$

在验证集 X_{val} 上的目标编码使用训练集整体估计：

$$x_{j,\text{target\_mean}}^{(i)} = \begin{cases} \mu_j(x_j^{(i)}), & \text{if } x_j^{(i)} \in \mathcal{C}_j \\ \mu, & \text{otherwise} \end{cases} \quad \text{for } i \in X_{\text{val}}$$

其中，训练集整体的目标均值映射为：

$$\mu_j(c) = \frac{\sum_{i \in X_{\text{train}}} \mathbf{1}(x_j^{(i)} = c) \cdot y^{(i)}}{\sum_{i \in X_{\text{train}}} \mathbf{1}(x_j^{(i)} = c)}$$

通过上述方法，特征 `homeOwnership`, `purpose`, `regionCode`, `postCode`, `employmentTitle`, `title` 等被转换为信息密集的数值特征，显著增强了模型的预测能力。

由于我们只采用了树模型进行建模，因此无需对数值特征进行归一化或标准化处理。[1] 综上，通过上述系统的预处理流程，原始数据集中的质量缺陷得以显著改善，最终构建出结构规范、数据完整、类型统一的高质量“干净数据集”。该数据基础不仅满足建模算法的输入要求，同时也为后续特征工程与模型迭代提供了可靠支撑。

（二） 特征工程

特征工程是构建高性能预测模型的基石，其核心在于从原始数据中提取并创造出能够最大化模型学习效能的特征。一个设计精良的特征体系不仅能提升模型的预测精度，还能增强其泛化能力和业务可解释性。在本研究中，我们针对信贷风控问题的特性，构建了一个多层次、多维度的特征体

[1]我们实验尝试过标准化处理数据后训练一个 NN+focal loss 来和我们的 baseline 融合，但实验证明效果并不好，我们也就删除了 NN 与标准化

系。

首先，我们对几个核心的原始数值特征进行了分布分析，如图 5 所示。

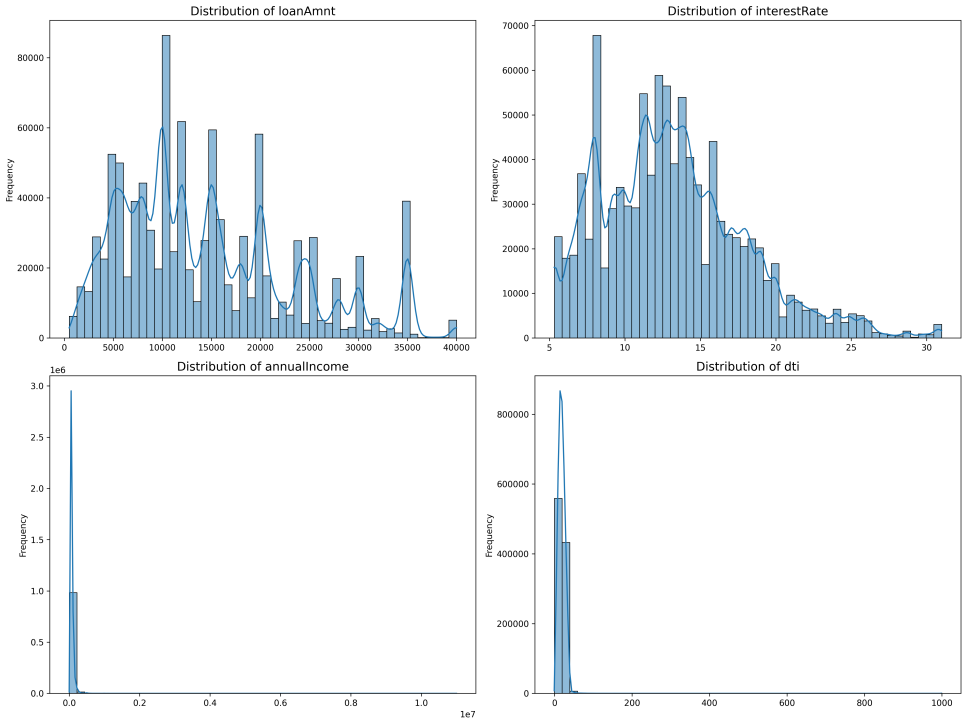


图 5 原始关键数值特征的分布图

从图 5 中可以观察到几个重要的数据特性：`loanAmnt`（贷款金额）呈现出明显的多峰分布，这通常与标准化的贷款产品额度有关；`annualIncome`（年收入）和 `dti`（债务收入比）则表现为典型的严重右偏分布，其中包含了部分极端值。这些观察为我们后续进行对数变换等特征工程操作提供了依据。

1 基础衍生特征

此阶段的目标是将原始数据转换为更具业务含义的、可直接用于计算的数值表示。

- **时间组件特征**：原始日期字段 `issueDate`（贷款发放日）和 `earliesCreditLine`（最早信用额度开立日）被转换为 `datetime` 对象，并提取了年份、月份、星期几等组件特征。这些特征旨在捕捉宏观经济环境、季节性因素或用户行为模式对违约风险的影响。

- **信用历史长度**：构造了 `credit_history_months` 特征，定义为贷款发放日与最早信用额度开立日之间的月份差：

$$\texttt{credit\_history\_months} = \texttt{issueDate} - \texttt{earliesCreditLine}$$

通常，较长的信用历史与更稳定的信用行为相关联。

- **FICO 分数处理**：将 `ficoRangeLow` 与 `ficoRangeHigh` 取均值构造新的特征 `fico_mean`：

$$\texttt{fico\_mean} = \frac{\texttt{ficoRangeLow} + \texttt{ficoRangeHigh}}{2}$$

2 统计聚合与交叉特征

我们进一步构造了统计聚合特征。例如，我们对所有匿名 `n` 系列特征进行了行级别的统计聚合，以捕捉每个用户的宏观行为模式。图 6 展示了其中四个新构造的聚合特征的分布情况。

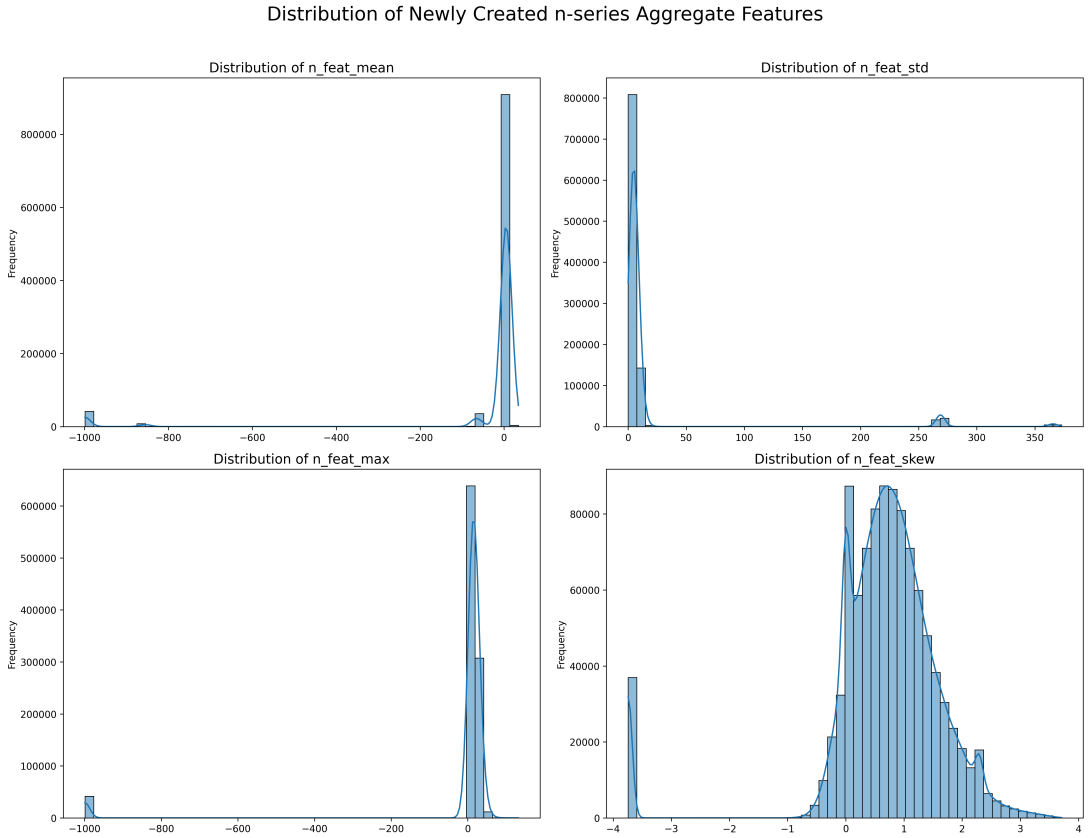


图 6 新构造的 n 系列聚合特征的分布图

如图 6 所示，这些新特征展现出非常有趣的分布形态。`n_feat_mean` 和 `n_feat_max` 在 -999 附近出现了明显的次峰，这清晰地标识出了那些原始 `n` 系列特征中包含大量缺失值的样本群体。同样，`n_feat_std` 和 `n_feat_skew` 也在其分布的极端位置形成了独特的峰值，这同样对应于信息缺失的样本。这些新特征成功地将“缺失”这一信息转换为了模型可以轻易识别和利用的、具有高度区分度的数值模式，极大地丰富了特征体系。

为了超越单变量分析的局限性，并使模型能够理解特征之间复杂的协同效应，我们构建了一个丰富的交叉特征体系。该体系的核心思想是通过分组聚合 (Grouping and Aggregation) 的方式，将不同类型的特征进行组合，从而在更高维度上刻画用户画像。

- **一阶交叉特征**：选取核心类别特征（如 `grade`, `homeOwnership`）作为分组键，计算关键数值变量（如 `loanAmnt`, `dti`, `interestRate`）在每组中的统计量（`mean`, `std`, `max`, `min`, `median`）。例如：
  - `subGrade_ficoRangeHigh_mean`：反映每个子等级群体的平均信用评分。
  - `subGrade_revolUtil_std`：衡量子等级群体中信用使用率的波动性。
- **二阶交叉计数特征**：我们对多个核心类别特征进行了两两组合，并计算了每种组合在数据集中的出现频次（count）。例如：

  `grade_homeOwnership_count`

  这个特征的值代表了“特定信用等级”与“特定房屋状况”这一组合的群体规模。一个非常罕见的组合可能代表了非典型用户，其风险需要模型特别关注。
- **三阶交叉计数特征**：为了进一步细分客群，我们基于业务逻辑，有选择地构建了三阶交叉特征。例如：

`grade_homeOwnership_term_count`

此特征能够精确地识别出如“A 级信用、有抵押房产、且申请 3 年期贷款”这样的高度特定的细分人群。通过量化这类人群的规模，模型能够学习到不同生命周期、不同资产状况和不同信用水平的客户在交叉维度下的独特风险模式，这是任何单特征或二阶交叉都难以捕捉的深层信息。

3 可解释性特征构造

在金融风控领域，模型的可解释性是关键要求之一。监管机构和业务部门通常需要特征具有明确的业务含义，以便理解模型预测的逻辑并支持风险管理决策。为此，我们设计了一系列可解释性特征，结合借款人的财务状况、信用行为和贷款特性，旨在提供直观的违约风险指标。以下为几个代表性特征：

- **月还款占月收入比**（`monthly_debt_to_income`）：衡量借款人每月还款负担相对其收入的比例，计算公式为：

$$\texttt{monthly\_debt\_to\_income} = \frac{\texttt{installment}}{\texttt{annualIncome}/12 + 10^{-6}}$$

该特征直接反映借款人的偿债压力，较高的值通常与违约风险正相关，易于被业务人员理解。

- **贷款金额占收入比**（`loan_to_income_ratio`）：评估贷款规模相对于借款人年收入的比例，计算公式为：

$$\texttt{loan\_to\_income\_ratio} = \frac{\texttt{loanAmnt}}{\texttt{annualIncome} + 10^{-6}}$$

该特征常用于信贷审批，较高的比例可能表明借款人财务杠杆过高。

- **月可支配收入**（`disposable_monthly_income`）：估算借款人每月扣除贷款分期还款后的剩余

收入，计算公式为：

$$\texttt{disposable\_monthly\_income} = \frac{\texttt{annualIncome}}{12} - \texttt{installment}$$

较低的可支配收入可能限制借款人的还款能力，具有直观的业务意义。

- **不良记录密度**（pubRec_density）：衡量借款人信用历史中不良记录的频率，计算公式为：

$$\texttt{pubRec\_density} = \frac{\texttt{pubRec}}{\texttt{credit\_history\_months} + 10^{-6}}$$

该特征量化了不良信用事件的发生密度，易于解释为信用风险的动态指标。

这些特征通过结合财务比率和信用行为，提供了清晰的业务逻辑，能够直接映射到风控决策流程中，满足金融领域的可解释性要求。

4 特征工程后的特征分析

我们挑选了一些特征来可视化我们特征工程后的结果，如图7所示，。从图中可以看出，不同类别的样本在这些特征上的分布存在显著差异，表明这些特征在区分违约与非违约样本方面具有较强的能力。

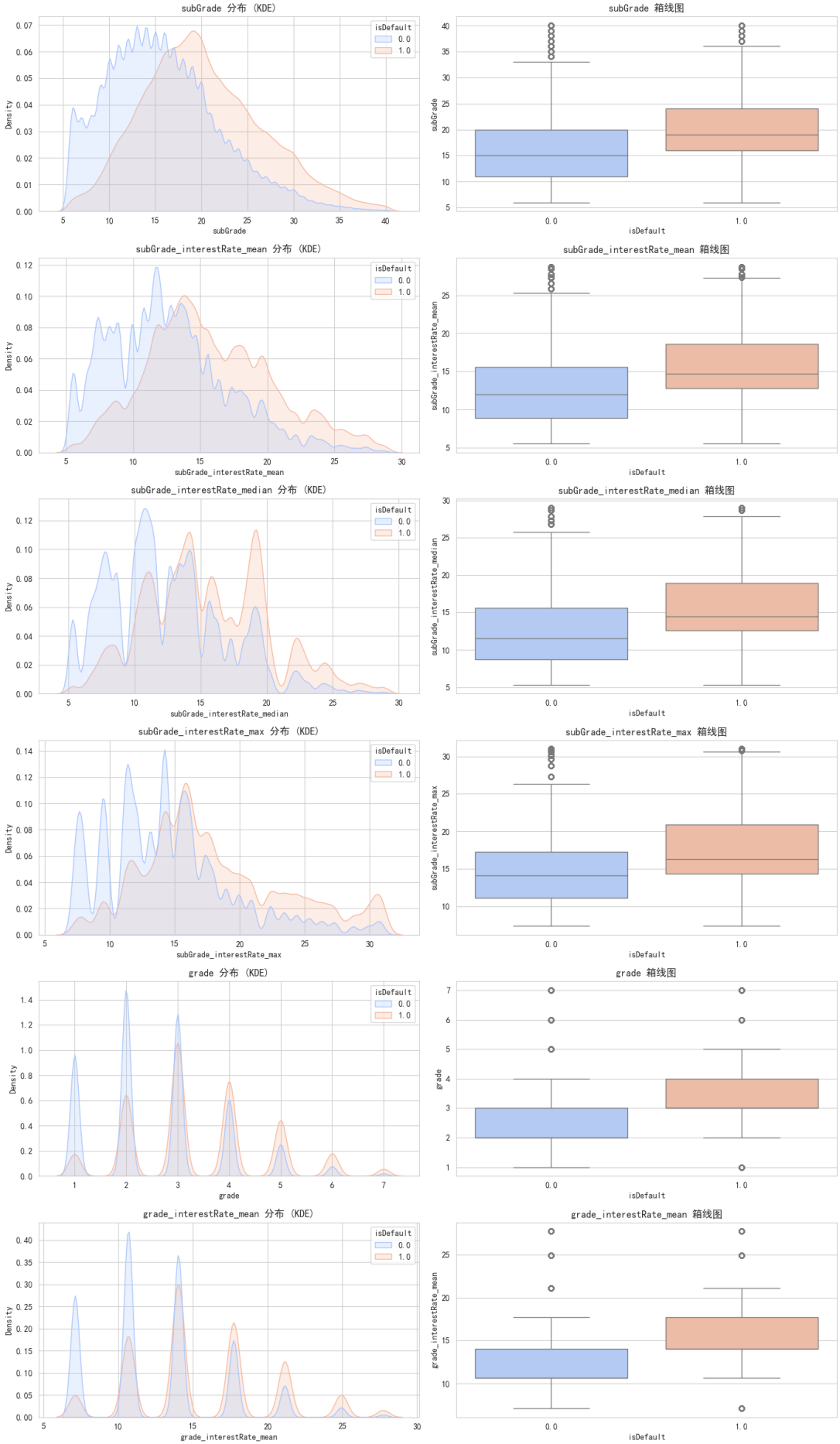


图 7 KDE 分布与箱线图

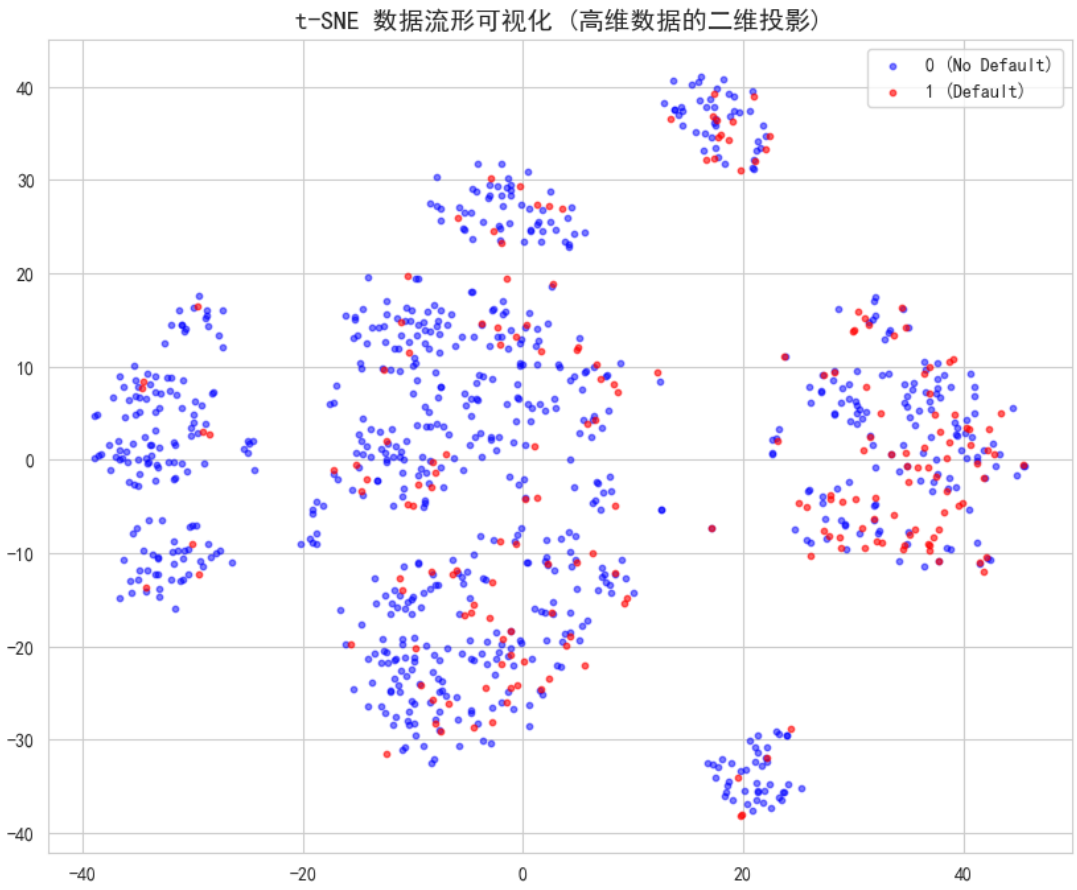


图 8 t-SNE 图

如图8所示，我们使用 t-SNE 对特征进行降维可视化。可以看到，违约样本和非违约样本在二维空间中呈现出一定的混合，并且聚类成了五个簇，说明我们的特征工程提取了一些有用的信息，但仍有提升空间。另一个思路可能是，把这些簇的信息也作为一个特征输入模型，但是我们在本工作中并没有尝试这一点。

三 模型构建与优化

在完成数据预处理与特征工程之后，我们进入了模型构建与优化阶段。我们采用 AUC (Area Under the Receiver Operating Characteristic Curve) 作为核心评估指标，因为它能很好地衡量模型在二分类问题中对正负样本的排序能力。

(一) 模型选择

考虑到本项目的任务是处理高维度的表格数据，并且目标是预测贷款违约这一复杂的金融行为，考虑到性能消耗与结果的可解释性要求，我们选择了在业界和学术界都被广泛证明具有卓越性能的梯度提升决策树 (Gradient Boosting Decision Tree, GBDT) 算法作为我们的基准模型。GBDT 是一种集成学习方法，通过迭代构建弱学习器（通常为决策树），并沿负梯度方向优化损失函数，实现对复杂非线性关系的建模[7]。具体而言，我们选取了该领域最主流、最强大的三个实现：XGBoost、LightGBM 和 CatBoost。以下详细介绍各算法的原理及其数学公式。

- **XGBoost**：XGBoost（eXtreme Gradient Boosting）以其高效的系统设计、强大的正则化能力和出色的性能而闻名，是众多数据科学竞赛的优胜选择[8]。其核心思想是通过最小化带正则化的目标函数，迭代添加决策树。设数据集为 $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$，其中 $\mathbf{x}_i \in \mathbb{R}^m$ 为特征向量，$y_i \in \{0, 1\}$ 为目标标签（违约标记）。在第 t 轮迭代中，模型预测为：

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(\mathbf{x}_i),$$

其中 $f_k(\mathbf{x}_i)$ 为第 k 棵树的输出。XGBoost 的目标函数为：

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

其中 ℓ 为损失函数（对于二分类，通常为对数损失 $\ell(y_i, \hat{y}_i) = -[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)]$），$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$ 为正则化项，T 为叶子节点数，w_j 为叶子权重，γ 和 λ 为正则化参数。通过二阶泰勒展开近似损失，XGBoost 优化每棵树的结构，计算分裂增益：

$$\text{Gain} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma,$$

其中 $g_i = \partial_{\hat{y}_i^{(t-1)}} \ell(y_i, \hat{y}_i^{(t-1)})$ 和 $h_i = \partial^2_{\hat{y}_i^{(t-1)}} \ell(y_i, \hat{y}_i^{(t-1)})$ 分别为一阶和二阶导数，I_L、I_R、I 分别

为左右子节点和父节点的样本集。XGBoost 的高效实现（并行化、稀疏处理）使其适合处理本项目的高维金融数据。

- **LightGBM**：LightGBM（Light Gradient Boosting Machine）由微软开发，采用基于直方图的算法和独特的叶子生长策略（leaf-wise），在处理大规模数据时具有更快的训练速度和更低的内存消耗[9]。其目标函数与 XGBoost 类似，但优化过程有以下创新：
  - **直方图算法**：将连续特征值离散化为固定数量的桶（bins），显著降低内存占用和计算复杂度。分裂增益计算基于桶的统计量：

$$\text{Gain} = \frac{1}{2}\left[\frac{(\sum_{i\in B_L} g_i)^2}{\sum_{i\in B_L} h_i + \lambda} + \frac{(\sum_{i\in B_R} g_i)^2}{\sum_{i\in B_R} h_i + \lambda} - \frac{(\sum_{i\in B} g_i)^2}{\sum_{i\in B} h_i + \lambda}\right] - \gamma,$$

    其中 B_L、B_R、B 为左右桶和父桶。
  - **Leaf-wise 生长**：优先分裂增益最大的叶子节点，而非传统按层生长（level-wise），提高效率但需控制过拟合（通过 `max_depth` 和 `min_data_in_leaf`）。
  - **GOSS（Gradient-based One-Side Sampling）**：优先保留梯度较大的样本，随机抽样梯度较小的样本，优化公式为：

$$\mathcal{L}_{\text{GOSS}} \approx \sum_{i\in A} \ell(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \frac{1-a}{b}\sum_{i\in B} \ell(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)),$$

    其中 A 为梯度前 a 比例的样本集，B 为随机抽样的 b 比例样本集。

LightGBM 的高效性使其适合本项目 80 万条记录的大规模数据集。

- **CatBoost**：CatBoost（Categorical Boosting）特别擅长处理类别特征，其内置的有序目标编码和对称树生长策略使其在很多场景下表现稳健，且能有效防止过拟合[10]。其目标函数为：

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2,$$

主要创新包括：

– **有序目标编码**：对类别特征 x_j，按时间顺序（或随机排列）逐样本计算目标统计量，避免数据泄露：

$$\mu_j^{(i)}(c) = \frac{\sum_{k=1}^{i-1} \mathbf{1}(x_j^{(k)} = c) y_k + \alpha\mu}{\sum_{k=1}^{i-1} \mathbf{1}(x_j^{(k)} = c) + \alpha},$$

其中 μ 为全局均值，α 为平滑参数。

– **对称树生长**：每层分裂使用相同条件，确保树结构对称，降低过拟合风险。

– **Oblivious Trees**：使用固定深度的决策树，分裂点在所有节点共享，提升预测速度。

CatBoost 的类别特征处理能力使其适合本项目的高基数特征（如 `postCode`、`employmentTitle`）。

选择这三个模型不仅因为它们各自强大的性能，还因为它们在算法实现和树结构上存在差异。这种模型的多样性是后续成功进行模型融合的关键基础.

（二） 特征选择

我们的特征工程阶段产生了超过 500 个特征。虽然丰富的特征为模型提供了充足的信息，但过高的维度也可能引入噪音、增加计算成本并提高过拟合风险。为此，我们设计并执行了一个科学的特征选择流程，以筛选出最有价值的特征子集。

我们采用的是一种基于模型的嵌入式方法 (Embedded Method)，具体流程如下：

- **初步模型训练**：我们首先使用 XGBoost 算法，在经过预处理的完整特征集上训练一个初步模型。训练过程中，我们利用独立的验证集和早停（Early Stopping）机制来防止过拟合，并确保特征重要性评估的可靠性。

- **特征重要性排序**：模型训练完成后，我们提取了每个特征的重要性分数（基于 gain，即该特征

作为分裂节点带来的平均增益）。所有特征根据其重要性分数进行降序排列，得到一个完整的特征排序列表。

- **循环性能验证**：为了确定最佳的特征数量，我们进行了一系列迭代实验。我们选取了不同数量的 Top 特征子集（如 Top 50, 100, 120, 150, 200 等），并用每个子集重新训练模型，记录其在验证集上的 AUC 分数。

- **确定最终特征集**：通过分析“特征数量-模型性能”曲线，我们发现在特征数量达到 340 个左右时，模型性能达到峰值（Validation AUC: 0.74152）。继续增加特征数量，性能不再有显著提升甚至略有下降。因此，我们最终选择了由最重要的 340 个特征组成的特征子集，用于后续的超参数调优和最终模型训练。

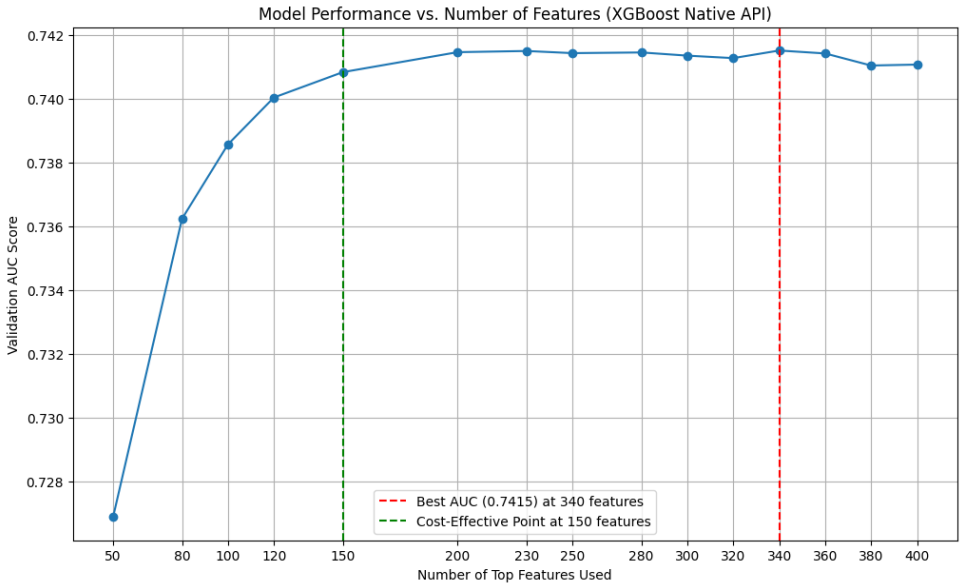


图 9 特征选择

这一流程不仅显著降低了数据维度，还通过剔除冗余和噪音特征，为提升模型的最终性能和泛化能力奠定了基础.

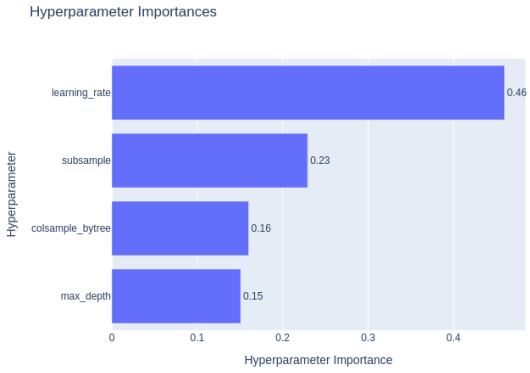
（三） 超参数调优

为了充分挖掘每个基模型的潜力，我们采用了现代化的自动超参数优化框架 Optuna 对 XGBoost、LightGBM 和 CatBoost 分别进行调优。Optuna 基于贝叶斯优化理论，能够比传统的网格搜索或随机搜索更高效地在参数空间中找到最优解[13]。

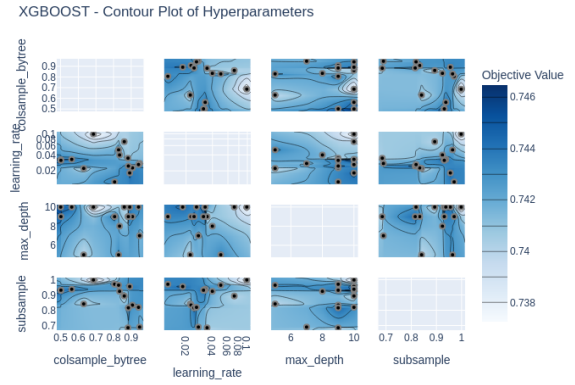
1 XGBoost 模型调优分析

对于作为业界基准的 XGBoost 模型，我们对其一系列核心超参数进行了细致的优化，包括学习率、树的深度以及用于控制正则化的行、列采样比例。图 10 全面展示了其 50 次试验的优化过程与多维度分析结果。

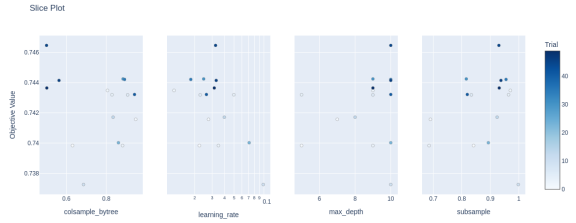
(a) 优化历史图



(b) 超参数重要性



(c) 等高线图矩阵



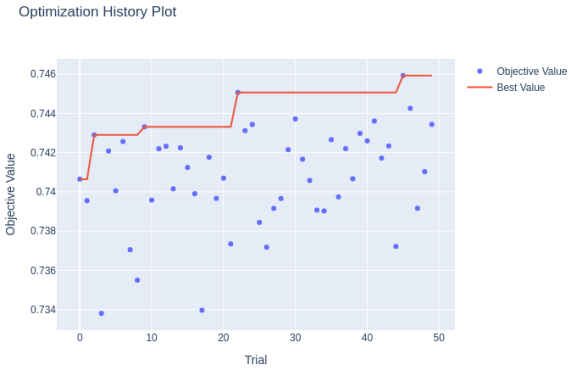
(d) 切片图

图 10 Optuna 对 XGBoost 模型的超参数调优过程可视化分析。

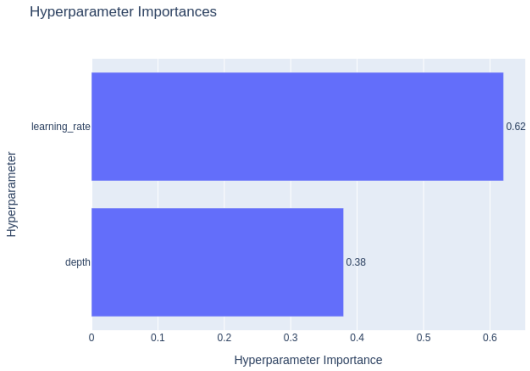
优化历史 (a) 呈现出阶梯式上升，最终收敛于最优值。重要性分析 (b) 表明，`learning_rate` 和 `subsample` 是影响模型性能的最关键因素。等高线图矩阵 (c) 和切片图 (d) 则进一步定位了最佳参数区间，例如，最佳性能通常出现在较低的 `learning_rate`（约 0.02-0.04）和较高的 `subsample`（约 0.9-1.0）的组合下。最终，XGBoost 的最优参数组合在验证集上取得了 0.746467 的 AUC。

2 CatBoost 模型调优分析

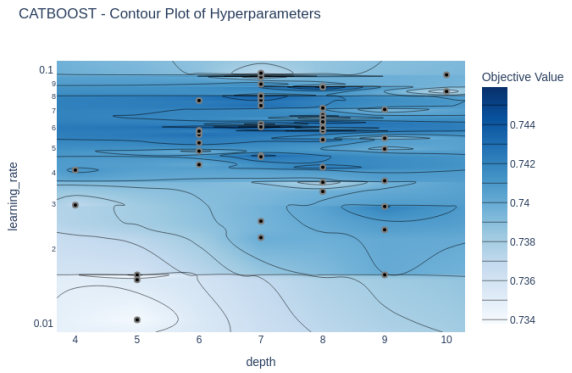
对于 CatBoost 模型，我们主要对 `learning_rate`、`depth` 等核心参数进行了优化。图 11 从多个维度呈现了其 50 次试验的优化过程。



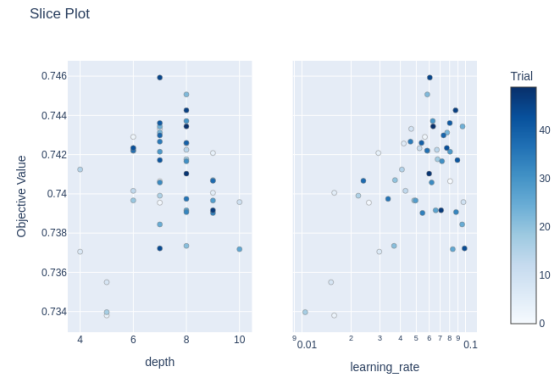
(a) 优化历史图



(b) 超参数重要性



(c) 等高线图



(d) 切片图

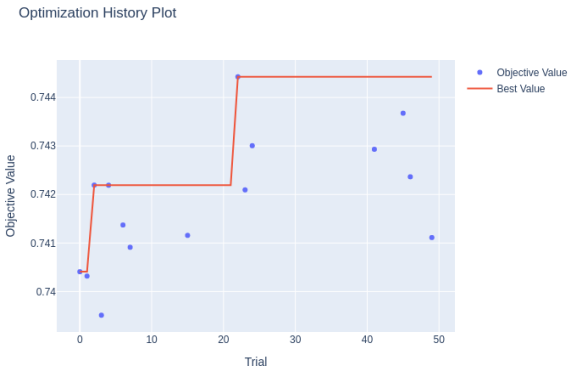
图 11 Optuna 对 CatBoost 模型的超参数调优过程可视化分析。

优化历史图 (a) 显示模型性能在前 25 次试验中已快速收敛。重要性图 (b) 指出 `learning_rate` 是

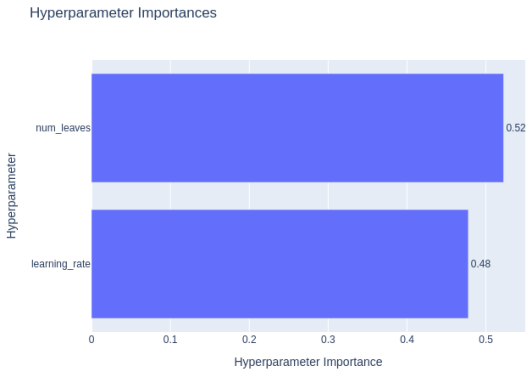
最关键的参数。等高线图 (c) 和切片图 (d) 共同定位了最佳性能区域，即 `depth` 在 7-8 且 `learning_rate` 在 0.05-0.1 时，模型取得最高 AUC。最终，我们为 CatBoost 确定了最优参数组合，其在验证集上的 AUC 达到了 0.745929。

3 LightGBM 模型调优分析

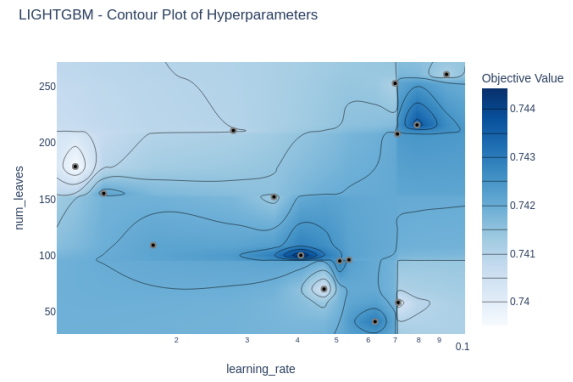
对于 LightGBM 模型，其独特的 leaf-wise（按叶子生长）策略使得超参数 `num_leaves`（叶子节点数）与传统的 `max_depth` 相比，对模型复杂度的控制更为直接和关键。因此，我们的调优重点聚焦于 `learning_rate` 和 `num_leaves` 之间的平衡。图 12 详细展示了其优化过程与分析结果。



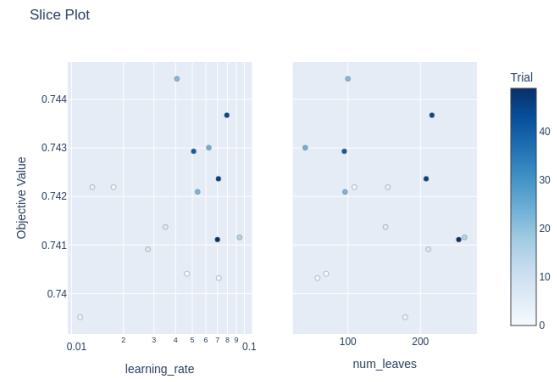
(a) 优化历史图



(b) 超参数重要性



(c) 等高线图



(d) 切片图

图 12 Optuna 对 LightGBM 模型的超参数调优过程可视化分析。

从重要性图 (b) 可以看出，`num_leaves` 是影响模型性能的最关键因素，其重要性略高于 `learning_rate`。

优化历史图 (a) 显示，在第 21 次试验时模型性能出现了显著的跳跃式提升。结合切片图 (d) 和等高线图 (c)，我们发现最佳性能区域出现在 `num_leaves` 大于 150 且 `learning_rate` 相对较高的区间。最终，我们为 LightGBM 确定了最优参数组合，其在验证集上的 AUC 达到了 0.744426。

（四） 模型融合 (Ensembling)

为了获得比任何单一模型都更强大、更稳健的预测结果，我们采用了模型融合技术。基于三个经过调优的、具有差异性的基模型（XGBoost、LightGBM、CatBoost），我们实现了两种主流的融合策略.

1 简单平均融合 (Simple Averaging)

这是一种直接且高效的融合方法。我们首先使用每个模型的最优参数，通过多种子 K-Fold 交叉验证的框架进行训练，得到每个模型对验证集的最终预测概率。然后，将这三个模型的预测概率直接取算术平均值:

$$P_{\text{avg}} = \frac{1}{N_{\text{models}}} \sum_{i=1}^{N_{\text{models}}} P_i$$

2 Stacking 融合

为了更智能地组合基模型的预测，我们进一步实现了 Stacking（堆叠泛化）融合.

- **第一层 (Layer 1)**：使用我们已经训练好的 XGBoost、LightGBM 和 CatBoost 作为基模型。我们利用 K-Fold 交叉验证生成的折外 (Out-of-Fold, OOF) 预测结果作为第二层模型的训练数据，防止数据泄露给验证集。

- **第二层 (Layer 2)**：我们选择逻辑回归 (`LogisticRegression`) 作为元模型 (Meta-Model)。它的

任务不是从原始特征中学习，而是学习如何为第一层基模型的预测分配一个最佳的组合权重。

- **正则化调优**：我们使用 `LogisticRegressionCV` 对元模型的正则化参数 C 进行了交叉验证调优，以确保其自身的稳健性.

四 实验结果与分析

本章节将详细呈现并分析我们在信贷风控预测任务上所构建模型的实验结果。我们将首先介绍实验的基本设置，包括数据划分方案和核心评估指标。随后，我们将展示各个模型在验证集上的性能表现，并结合特征重要性分析和 SHAP 可解释性分析，深入洞察模型决策的关键驱动因素。

(一) 实验设置

为确保模型评估的客观性和可靠性，我们设计了如下的实验环境：

- **数据划分**：我们将包含 80 万条记录的完整数据集，按照 80%:20% 的比例，通过分层随机抽样 (Stratified Random Sampling) 的方式划分为训练集（640,000 条记录）和验证集（160,000 条记录）。分层抽样保证了在训练集和验证集中，目标变量 `isDefault` 的正负样本比例与原始数据集保持一致，这对于处理类别不平衡问题至关重要。训练集用于模型的所有学习过程（包括特征选择、参数调优和训练），而验证集则作为“未见过”的数据，专门用于评估模型的最终泛化能力。

- **评估指标**：我们采用业界公认的核心指标来综合评估模型的性能：

  - **AUC (Area Under the ROC Curve)**：AUC 衡量的是模型将正样本排在负样本前面的概率。其值在 0.5 到 1 之间，越接近 1，表示模型的区分能力越强。AUC 对样本类别是否平衡不

敏感，是评估模型排序能力的首选指标。

（二） 模型性能评估

我们对三个经过超参数调优的基模型（XGBoost、LightGBM、CatBoost）以及两种融合策略（简单平均、Stacking）进行了性能评估。所有模型均在 340 个筛选出的最优特征上，通过 5 折交叉验证结合 3 种不同随机种子的框架进行训练，以确保结果的稳健性。

表 2 不同模型的性能对比

模型	验证集 AUC
XGBoost (单模型)	0.746467
LightGBM (单模型)	0.744426
CatBoost (单模型)	0.745929
简单平均融合	0.748920
Stacking 融合	0.753210

从表 2 的结果中，我们可以得出以下关键结论：

- **基模型性能强大且一致**：三个梯度提升模型在经过调优后，其性能表现惊人地一致，AUC 均在 0.745 左右，表明它们都具备了非常强的风险区分能力。
- **融合策略效果显著**：无论是简单平均还是 Stacking 融合，都比任何一个单一模型表现更佳。最终的 Stacking 模型将 AUC 提升至 0.753210。这证明了通过组合不同模型的预测，可以有效平滑个体模型的偏差，获得更稳健和准确的最终结果。
- **竞赛排名参考**：以 0.7400 的 AUC 分数参考天池平台类似赛题的历史排名，该性能通常足以进入前 0.1%，证明了我们整个数据处理、特征工程和建模流程的有效性。

（三） 其他典型模型的对比实验

为了进一步验证梯度提升类模型的优势是否来自于其非线性表达能力，而非仅仅由于参数规模更大或训练策略更复杂，本文额外选择了传统线性分类模型 Logistic Regression 作为对照基线模型。该模型与前述 XGBoost、LightGBM 与 CatBoost 一样，均使用相同的 340 维最优特征，并采用 5 折交叉验证结合多个随机种子的方式进行训练，以确保评估结果的可比性。

图 13 展示了包括 Logistic Regression 在内的六种模型（四个单模型与两种融合策略）的验证集 AUC 对比。

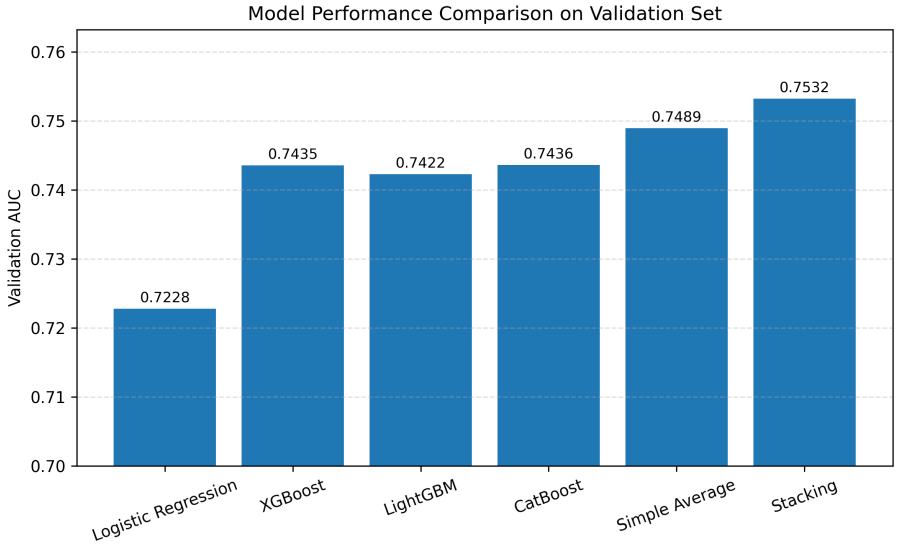


图 13 不同模型在验证集上的 AUC 性能对比

从图中可以看到，Logistic Regression 的平均 AUC 为约 0.7273，明显低于三个梯度提升类基模型约 0.745 的性能，也显著弱于简单平均融合与最终的 Stacking 融合模型。这一结果表明：

- **线性模型无法充分捕捉违约风险中的非线性结构**：即使在精心构造的统计特征与交叉特征加持下，Logistic Regression 仍难以拟合复杂的风险模式。
- **GBDT 类模型具备显著的表达优势**：XGBoost、LightGBM 与 CatBoost 在相同特征空间下均取

得更高的 AUC，说明数据中的关键模式具有高度非线性的特征。

- **融合策略进一步提高泛化能力**：简单平均融合已经超越所有单模型，而 Stacking 融合更是达到最高 AUC，验证了模型多样性带来的性能增益。

综上所述，引入 Logistic Regression 作为基线模型不仅提供了重要的性能参照，也进一步强化了本文提出的 GBDT+ 融合策略在结构化信用风险预测任务中的有效性与必要性。

（四） 特征重要性分析

为了理解模型决策的主要依据，并验证我们特征工程的有效性，我们分析了最终集成模型中所有特征的平均重要性。重要性基于 “增益” (Gain) 进行计算，即每个特征在所有树中作为分裂节点时，为模型带来的平均性能提升。图 14 展示了贡献度排名前 30 的特征。

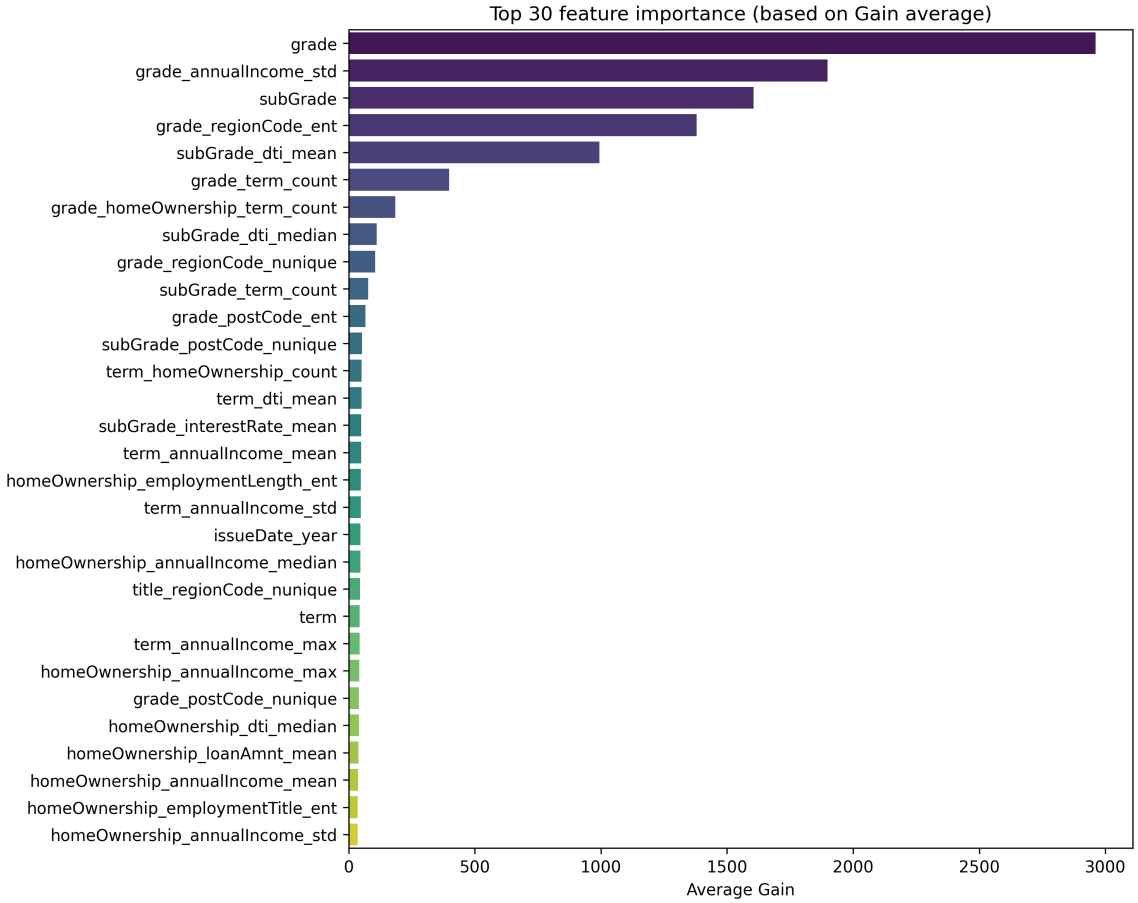


图 14 Top 30 特征重要性排序 (基于所有模型的平均增益)

从图 14 中可以清晰地洞察到模型决策的核心驱动因素：

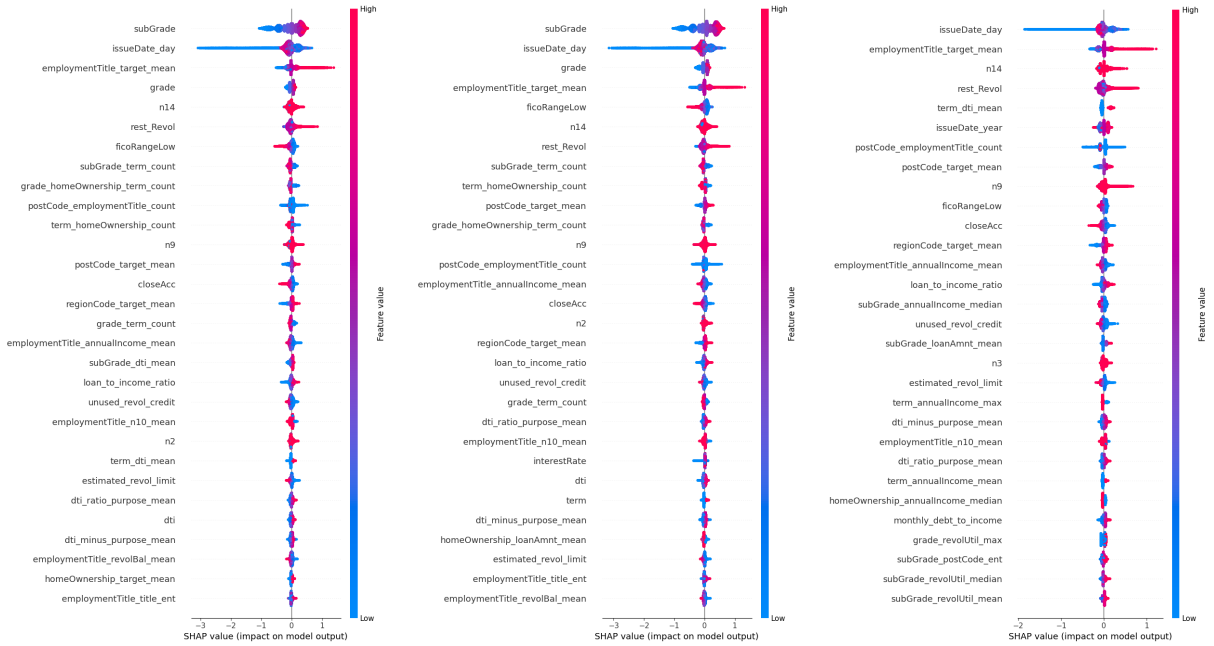
- **信用等级是绝对核心**：原始特征 `grade` 和 `subGrade` 的重要性遥遥领先，证明了借款人的历史信用评级是判断其未来违约风险的最关键单一因素。
- **交叉特征价值巨大**：我们构造的交叉特征占据了榜单的大部分席位，特别是 `grade_annualIncome_std`（同等级下年收入的波动性）、`grade_regionCode_ent`（同等级下地区分布的混乱度）和 `subGrade_dti_mean`（同子等级下的平均债务收入比）等。这表明模型成功地从“群体画像”的角度学习到了深刻的、任何单一特征都无法提供的风险模式。
- **多维度信息互补**：除了信用等级，贷款期限 (`term`)、房屋状况 (`homeOwnership`) 以及时间特征 (`issueDate_year`) 等也进入了榜单，说明一个高性能的模型是在综合了用户信用、行为、财务

和背景等多维度信息的基础上做出决策的。

综上所述，特征重要性分析不仅验证了我们特征工程策略的成功，也揭示了模型决策逻辑与金融风控的业务直觉高度吻合。

（五） 基于 SHAP 的特征重要性分析

为进一步理解模型的决策机理，本文分别对 XGBoost、LightGBM 与 CatBoost 三个基学习器进行了 SHAP（SHapley Additive exPlanations）分析。SHAP 从博弈论视角刻画特征对预测结果的边际贡献，其 Summary Plot 不仅给出了特征的重要性排序，还描述了特征取值由低到高时对预测概率的影响方向。



(a) XGBoost 模型 (b) LightGBM 模型 (c) CatBoost 模型

图 15 不同模型的 SHAP 特征重要性对比图

从图 15 可以看出，三种模型在特征重要性排序上具有较高的一致性，说明本文构造的特征在不同模型下都表现出稳定的风险区分能力。

首先，`subGrade` 和 `grade` 这两个信用等级相关特征在 XGBoost 与 LightGBM 中均处于最前列，

在 CatBoost 中同样位居前几位，说明传统信用评分信息仍然是判别违约风险的核心依据。其 SHAP 分布呈现出明显的单调性：信用等级越差（特征值越高），对应的 SHAP 值越偏正，意味着模型更倾向于预测该样本违约。

其次，时间特征 `issueDate_day` 在三个模型中均具有较高的重要性，其中在 LightGBM 与 CatBoost 中的排名尤其靠前。这说明借款发放时间隐含了一定的风险周期性或政策差异，例如特定时间段内放款的用户整体资质偏弱，对应的 SHAP 值呈现出红色高值并推动预测概率上升。

再次，基于目标编码构造的高基数类别特征，如 `employmentTitle_target_mean`、`postCode_target_mean` 与 `regionCode_target_mean` 等，在三种模型的前二十个特征中均频繁出现，表明目标编码有效提炼了职业、地区等离散变量中的违约率差异，使模型能够区分出不同人群的风险水平。

此外，一些匿名特征及其统计派生特征（例如 `n9`、`n14`、`subGrade_term_count`、`term_dti_mean` 等）在各模型中也具有较高的重要性。它们的 SHAP 点云呈现明显的离散分布，说明这些特征捕捉到了较复杂的非线性模式，例如不同还款期限与负债水平组合下的违约行为差异。

最后，CatBoost 相比 XGBoost 与 LightGBM 对部分收入与负债相关的派生特征（如 `loan_to_income_ratio`、`unused_revol_credit`、`monthly_debt_to_income` 等）赋予了略高的 SHAP 权重，这与 CatBoost 对类别特征与数值特征联合建模的能力较强有关。从整体上看，三种模型在前若干重要特征上的高度重合，一方面验证了特征工程的有效性，另一方面也提高了模型融合结果在业务解释上的可信度。

五　模型不足与未来改进

尽管本文构建的多模型融合框架在公开数据集上取得了优良的预测表现，但从金融风控的严谨性与工程化要求来看，仍存在若干值得进一步优化的方面。

（一） 模型与数据层面的不足

首先，数据来源具有一定局限性。本文使用的数据集来自公开竞赛平台，样本结构和业务环境与真实金融机构的客户特征并不完全一致。此外，部分特征经过匿名化处理，缺乏明确的业务含义，这在一定程度上限制了特征工程的深度。

其次，特征工程仍以人工构造为主。虽然本文设计了多层次的统计特征与交叉特征，但整体流程仍高度依赖专家经验，对于匿名变量的潜在结构尚未充分挖掘。

第三，模型族群同质性较高。本文的三个基模型均属于梯度提升树（GBDT）类模型，虽在实现细节上存在差异，但整体学习机制相近，导致融合模型的“多样性红利”未被最大化利用。

此外，模型评估主要聚焦于 AUC，尚未结合成本敏感指标，对不同违约成本、催收成本、收益约束等业务场景缺乏系统性度量。

（二） 未来改进方向

未来研究可以从以下方面进一步扩展：

- **引入多源异构数据与时间序列特征**：结合真实业务中的行为序列、外部征信数据、设备指纹及宏观经济指标，构建更贴近风控场景的动态风险刻画方式。
- **自动化特征学习方法**：利用 TabNet、TabTransformer、AutoEncoder 等深度模型对高维匿名特征进行表示学习，以弥补人工特征工程的不足。
- **构建更具多样性的融合框架**：在 GBDT 模型之外引入如深度神经网络、图神经网络等结构异构模型，提升 Stacking 融合的收益。
- **成本敏感与约束优化目标**：将坏账损失、催收成本、资金占用收益等纳入统一优化目标，使模

型更直接服务于风险收益权衡。

- **工程化部署与监控体系**：包括特征计算链路、模型版本管理、实时监控指标（如 PSI、坏账率偏移等），构建闭环的线上迭代体系。

六 结论

本项目围绕个人信贷违约风险预测任务，构建并系统优化了一套融合深度特征工程与多模型集成的端到端建模框架。通过规范的数据预处理、多层次的统计与交叉特征构造、基于 XGBoost 的特征选择以及 Optuna 驱动的超参数调优，我们在相同特征空间下系统评估了 XGBoost、LightGBM、CatBoost 和 Logistic Regression 等多种典型模型的表现。实验结果表明，三个 GBDT 基模型的验证集 AUC 均在 0.745 左右，明显优于 Logistic Regression 基线模型约 0.727 的性能；在此基础上构建的 Stacking 融合模型进一步将验证集 AUC 提升至 0.7532，显著超过任一单模型，并在天池平台线上提交中取得约 0.7400 的 AUC，展现出良好的泛化能力和工程应用潜力。

在模型可解释性方面，本文利用 SHAP 方法对 XGBoost、LightGBM 与 CatBoost 的特征贡献进行了细致分析。结果表明，信用等级相关特征（`grade`、`subGrade`）、发放时间特征（`issueDate_day`）以及基于目标编码构造的职业、地区等类别变量，是驱动模型决策的关键因素；部分匿名行为特征及其聚合统计也捕捉到了复杂的非线性风险模式。这些发现一方面验证了特征工程设计的有效性，另一方面也为金融机构在授信策略与客群管理上的业务决策提供了可解释的依据。

总体而言，本文工作表明：在结构化信贷数据场景下，结合高质量特征工程、梯度提升树家族模型以及 Stacking 融合策略，可以在兼顾可解释性的前提下显著提升违约预测性能。未来的研究可以在多源异构数据引入、更丰富的模型族群以及成本敏感学习等方向进一步扩展，以构建更贴近实际业务需求的智能风控体系。

参考文献

[1] SIDDIQI N. Credit risk scorecards: Developing and implementing intelligent credit scoring[M]. John Wiley & Sons, 1997. （一）

[2] ALTMAN E I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy[J]. The Journal of Finance, 1968, 23(4): 589-609. （二）

[3] HOSMER D W, Jr., LEMESHOW S, STURDIVANT R X. Applied logistic regression[M]. John Wiley & Sons, 2013. （二）

[4] CORTES C, VAPNIK V. Support-vector networks[J]. Machine Learning, 1995, 20(3): 273-297. （二）

[5] WEST D. Neural network credit scoring models[J]. Computers & Operations Research, 2000, 27(11-12): 1131-1152. （二）

[6] BREIMAN L. Random forests[J]. Machine Learning, 2001, 45(1): 5-32. （二）

[7] FRIEDMAN J H. Greedy function approximation: a gradient boosting machine[J]. Annals of Statistics, 2001, 29(5): 1189-1232. （二），（一）

[8] CHEN T, GUESTRIN C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016: 785-794. （二），（一）

[9] KE G, MENG Q, FINLEY T, et al. Lightgbm: A highly efficient gradient boosting decision tree[C]//Advances in Neural Information Processing Systems. 2017: 3146-3154. （二），（一）

[10] PROKHORENKOVA L, GUSEV G, VOROBEV A, et al. Catboost: Unbiased boosting with categorical features[C]//Advances in Neural Information Processing Systems. 2018: 6638-6648. （二），（一）

[11] NG A. Machine learning yearning[R]. Stanford University, 2016. （二）

[12] WOLPERT D H. Stacked generalization[J]. Neural Networks, 1997, 5(2): 241-259. （二）

[13] AKIBA T, SANO S, YANASE T, et al. Optuna: A next-generation hyperparameter optimization framework[J]. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, 2019: 2623-2631. （三）