**贷款违约预测系统**

1. **项目背景**

贷款违约风险是金融机构面临的主要信用风险之一，不仅影响机构的盈利能力，还可能引发系统性金融风险。随着金融科技的发展，传统基于人工审核的信用评估方法已无法满足现代金融业务的需求。本项目旨在设计并实现一个智能化的贷款违约预测系统，通过分析借款人的多维特征，准确预测其违约概率，帮助金融机构优化信贷决策，降低不良贷款率。

1. **项目步骤**
2. **项目定义和需求分析**
3. **目标设定**

业务上，通过精准识别违约风险，降低坏账损失，优化信贷决策效率。技术上，保障模型准确率、可解释性。系统建设方面，实现实时预测、有效监控和持续迭代。

1. **需求收集**

与小组成员讨论项目需求。确定贷款违约预测项目的范围、各个时间段的任务分配表和资源需求。

1. **数据收集与理解**
2. **数据来源识别**

确定需要的数据类型（结构化数据、非结构化数据）。

确定数据来源（如内部业务系统数据、征信机构数据、税务部门数据）。

1. **数据集简介**

从kaggle网站（https://www.kaggle.com/datasets/liuzhuangzhuang/tianchi?select=testA.csv)上下载有关贷款违约预测的数据集

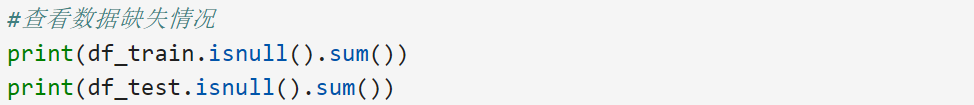
|  |
| --- |
| 2c669a15f2e184689608127dbc6af18 |

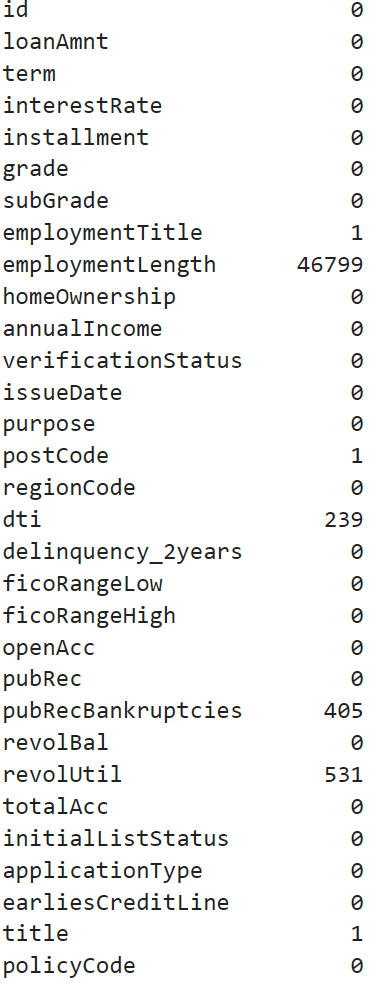
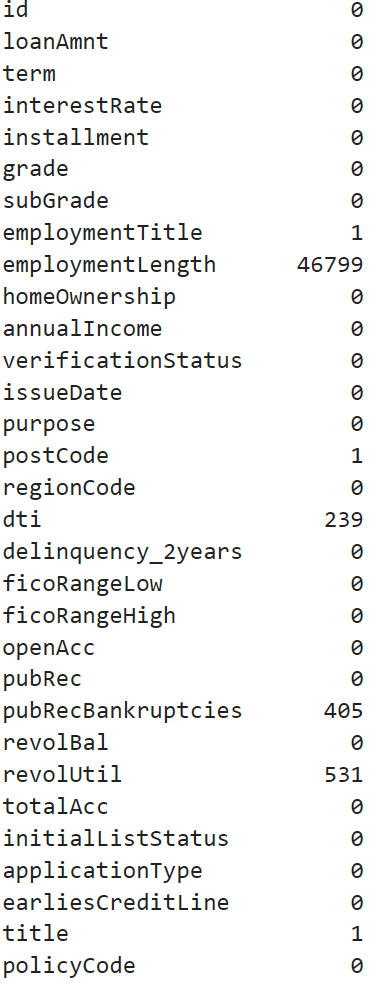
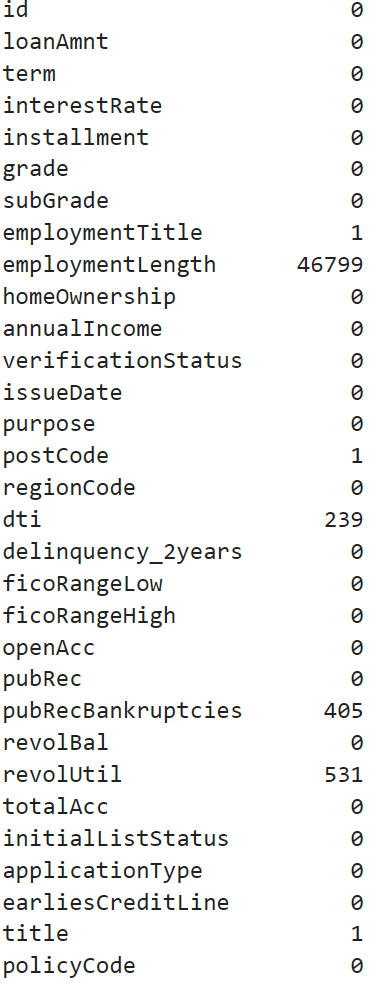
|  |
| --- |
| d4640aedf20e745425361c8117fd141 |

(注：数据集总共有100万 行和32列，其中72万为训练集，8万为验证集，20万为测试集)

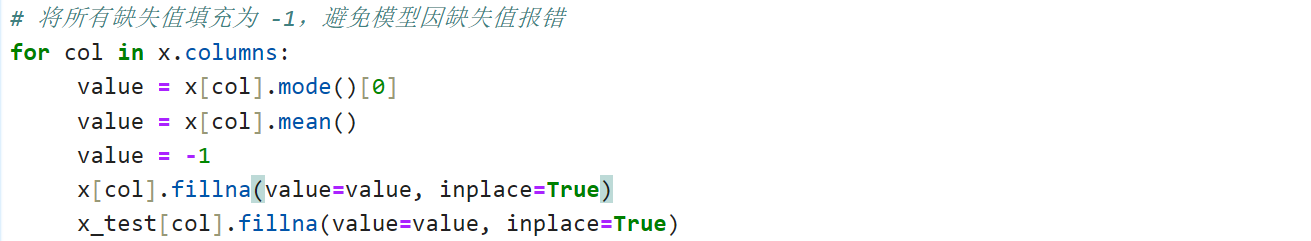
数据集中各列列名含义：

1. id：标识每笔贷款记录的编号。
2. loanAmnt：贷款金额，即借款人所申请的贷款数额。
3. term：贷款期限，以年为单位表示借款人需要偿还贷款的时长。
4. interestRate：贷款利率，反映借款人贷款所需支付的利息比例。
5. installment：每月还款额，借款人每月需偿还的固定金额。
6. grade：贷款等级，根据借款人信用状况等因素划分的评级。
7. subGrade：更细分的贷款等级，对grade的进一步细化。
8. employmentTitle：就业职位，记录借款人的工作岗位。
9. employmentLength：工作年限，借款人当前工作的时长。
10. homeOwnership：住房所有权情况，用数字编码表示借款人是否拥有自有住房等情况。(0表示租房，1表示自有住房但房贷尚未还清，2表示完全自有住房）。
11. annualIncome：借款人的年收入。
12. verificationStatus：收入验证状态，表示借款人的收入是否经过验证以及验证的情况。
13. issueDate：贷款发放日期。
14. isDefault：是否违约，用数字表示借款人是否出现违约情况（如 1 表示违约，0 表示未违约）。
15. purpose：贷款目的，用数字编码表示借款人申请贷款的用途。
16. postCode：邮政编码，用于标识借款人的居住地区。
17. regionCode：地区编码，用于更宏观地划分借款人所在的区域。
18. dti：债务收入比，反映借款人每月债务支出与收入的比例。
19. delinquency\_2years：过去两年的逾期次数，体现借款人过去的信用表现。
20. ficoRangeLow：FICO 信用分下限，FICO是由美国费埃哲公司（Fair Isaac Corporation）开发的一种信用评分系统 。
21. ficoRangeHigh：FICO 信用分上限。
22. openAcc：未结清账户数，借款人当前未还清的账户数量。
23. pubRec：公共记录，用分类编码表示借款人的破产、税收留置权等公共记录信息。
24. pubRecBankruptcies：公共记录中的破产记录。
25. revolBal：信用卡循环余额，借款人信用卡上未还清的余额。
26. revolUtil：信用卡使用比率，即已使用的信用额度与总信用额度的比例。
27. totalAcc：总账户数，借款人名下的所有账户数量。
28. initialListStatus：用数字编码（0、1）表示贷款审核是否通过。
29. applicationType：申请类型，用数字编码（0、1）表示贷款申请是个人申请还是联合申请。
30. earliesCreditLine：最早的信用记录日期，反映借款人开始有信用活动的时间。
31. title：贷款标题，借款人或贷款机构对该笔贷款的简要描述。
32. policyCode：政策代码，与贷款所遵循的政策或规则相关。
33. **数据预处理**
34. **数据清洗**

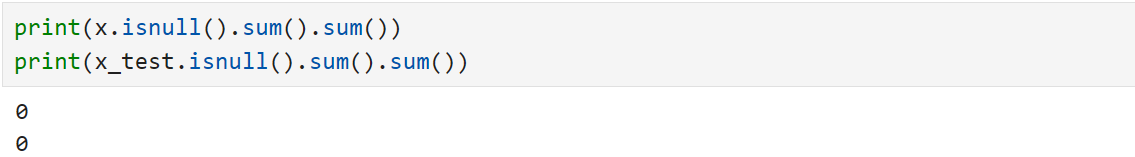




**1.缺失值处理**：先尝试众数填充 ，再尝试均值填充， **最终硬编码为-1，**'employmentLength'和n系列匿名特征均缺失大量数据，考虑以均值填充



最后缺失值处理验证

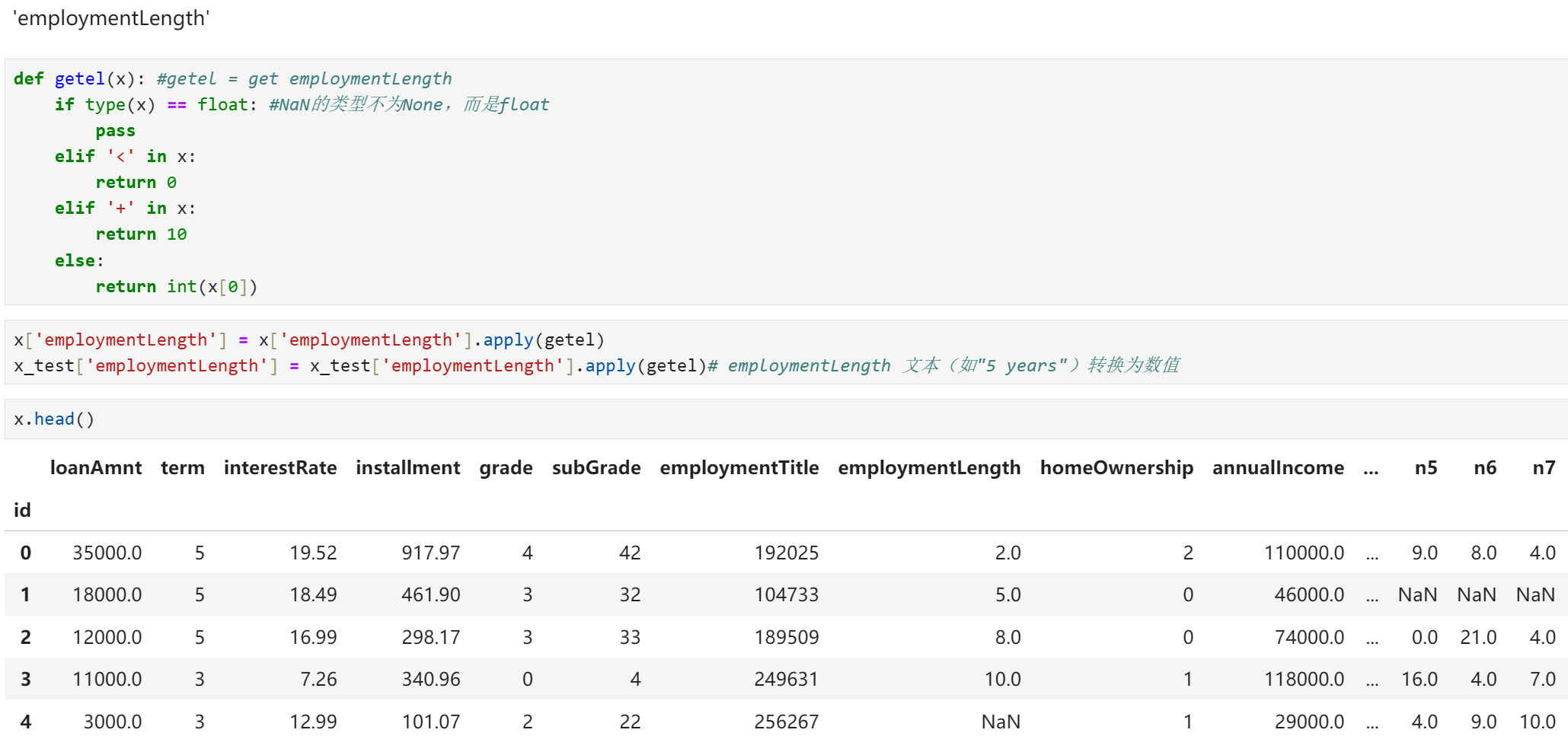


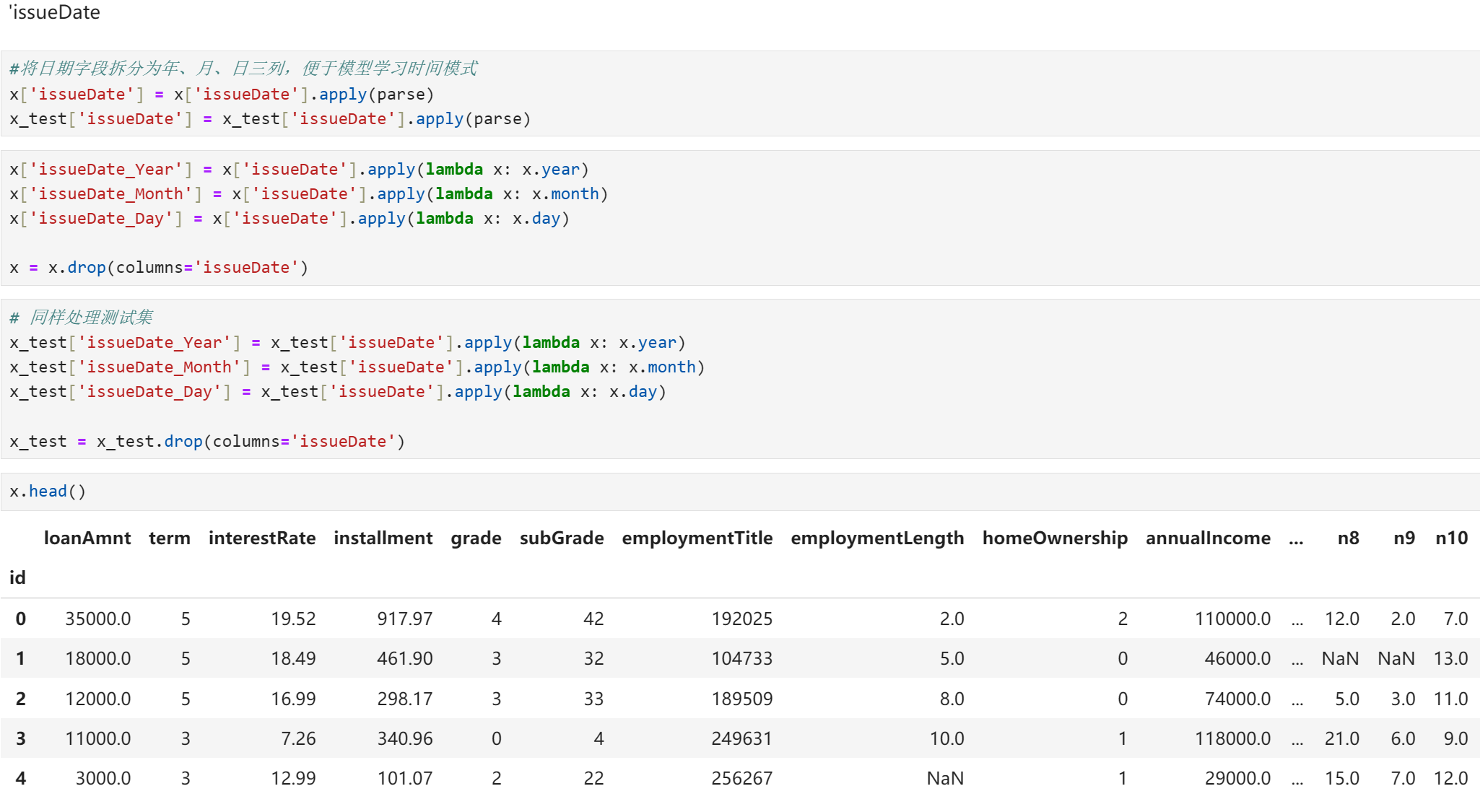
1. **数据转换**

**1.类别数据数值化**：x['subGrade'].apply(lambda x: x[-1])：提subGrade字段的**最后一个字符**（如A1→1、B2→2 ），转为整数。x['grade'] \* 10：用grade（主等级，如1、2）乘以10，放大主等级的权重。相加后，subGrade变为数值（如A1+grade=1→1 + 1\*10 = 11），便于模型处理。

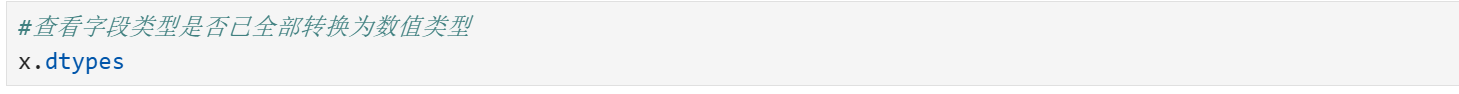


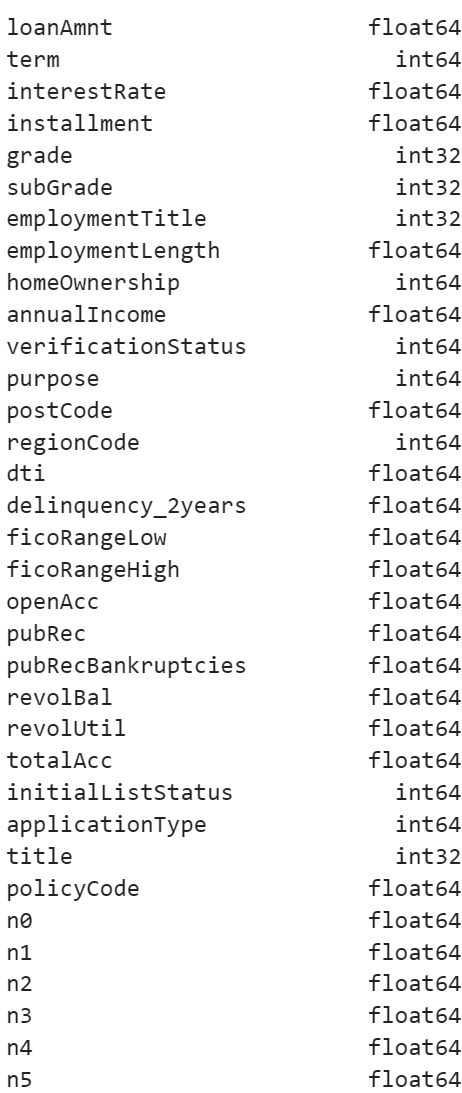
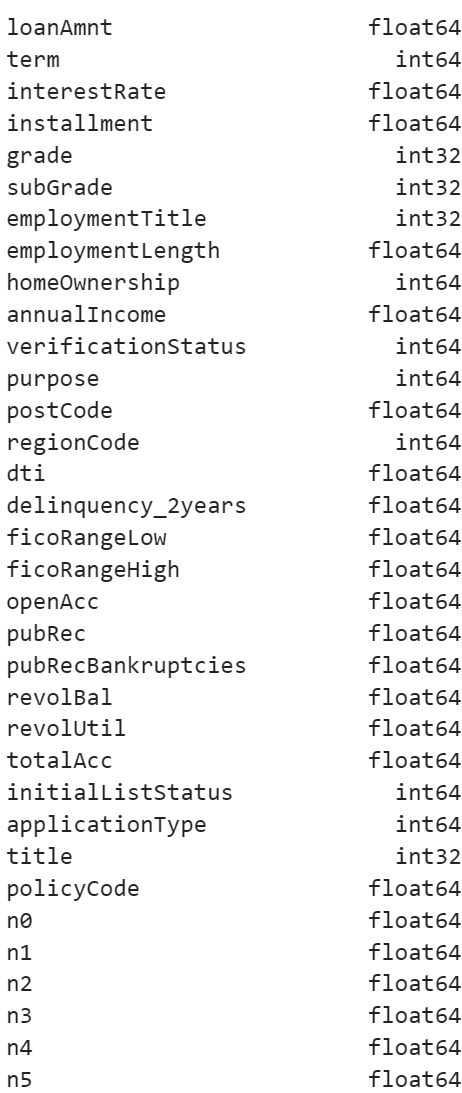
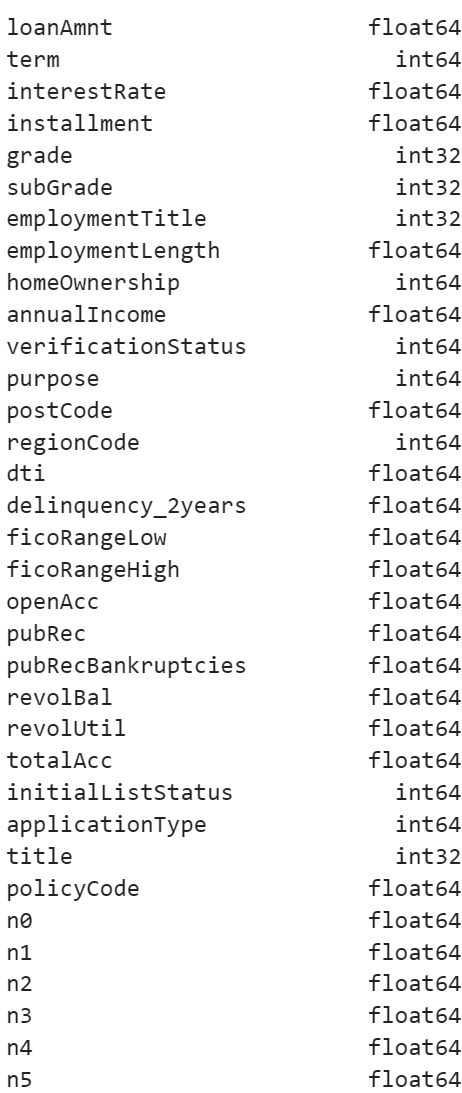












**2.数值数据标准化**：对 “贷款金额” 等可能呈现正态分布的连续型数值变量，使用StandardScaler进行标准化，将数据转换为均值为**0、**标准差为**1**的标准正态分布。对“贷款期限”等具有固定范围的数值变量，使用MinMaxScaler进行归一化，将数据映射到**[0, 1]**区间。



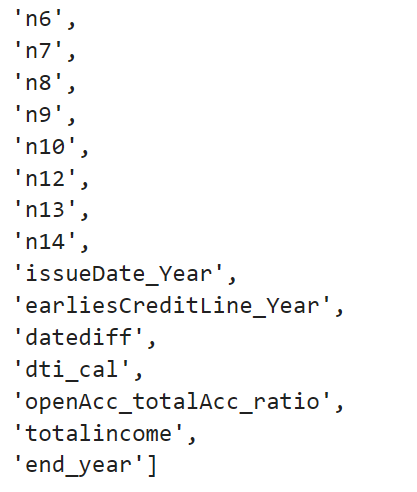
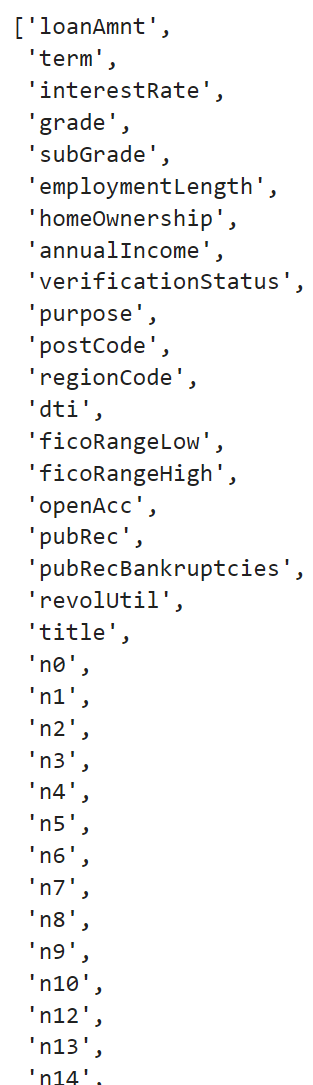
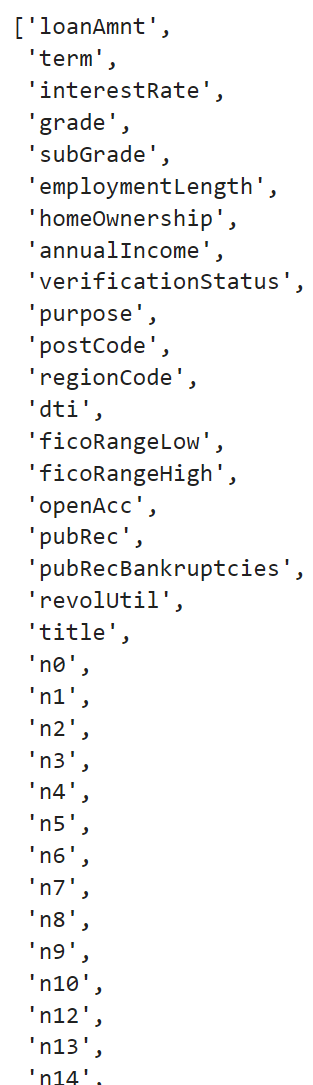


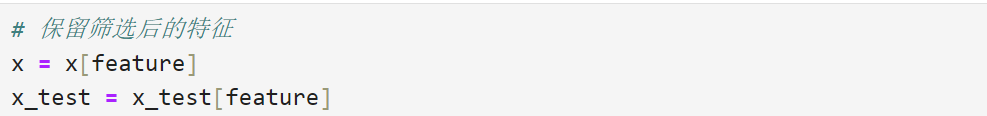
1. **特征选择**

**1.特征提取**：日期字段提取时间特征；算 “收入负债比” 等比率特征；TF-IDF 提取信用报告关键词。

**2.特征筛选**：算相关系数矩阵删冗余特征；结合逻辑回归用 RFE 筛选；LASSO 回归压缩系数为 0 实现特征选择；综合三者结果确定最终特征子集。







1. **探索性数据分析**
2. **探索性数据分析（EDA）**

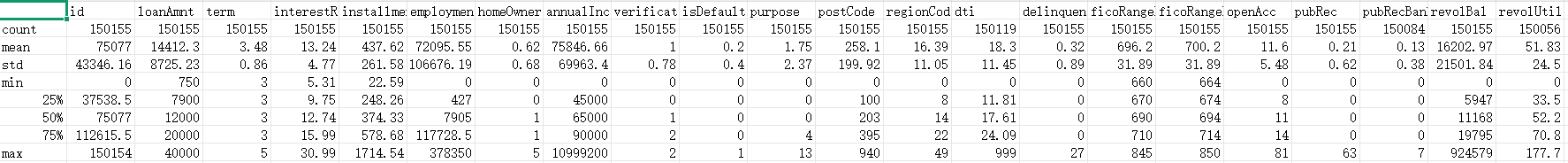
计算统计量（如均值、标准差、分位数）以了解数据的基本分布。

检查特征之间的关系（如相关性矩阵）。



电子计算机

AI 生成的内容可能不正确。



图片包含 图示

AI 生成的内容可能不正确。

1. **相关性分析**

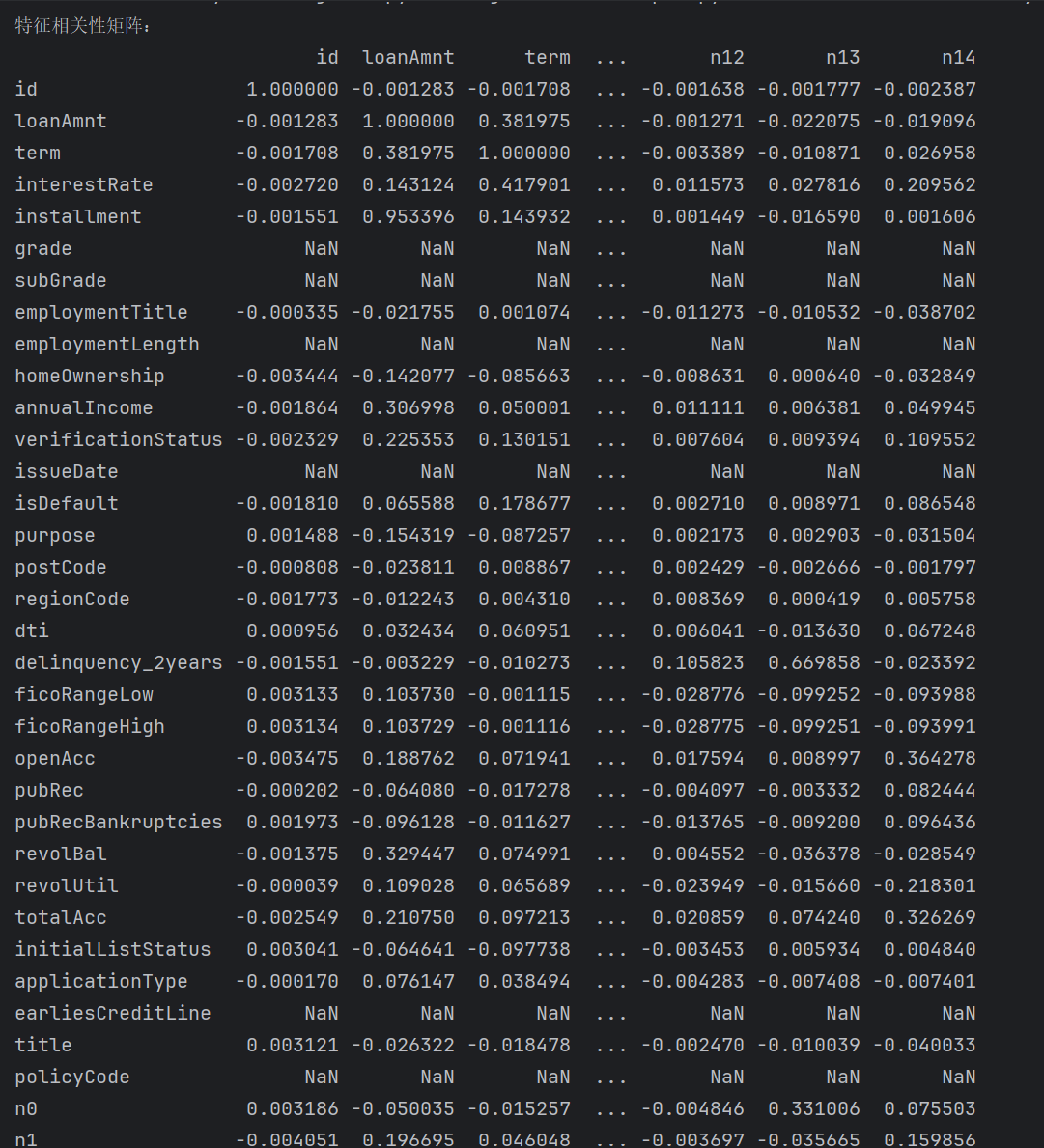
使用直方图或箱线图等了解单个特征的分布。

**数据分布:**

1、均值与标准差：例如 loanAmnt的均值为 14412.30，标准差为 8725.23，标准差相对均值较大，这表明贷款金额数据的离散程度较高，数据可能较为分散。

2、最小值与分位数：loanAmnt 的最小值为 750，25% 分位数为 7900，说明有 25% 的贷款金额低于 7900，同时最小值与均值差距较大，可能存在一些小额贷款拉低了整体数值或者数据中存在异常值影响了均值。

检查特征之间的关系（如相关性矩阵）。



**使用箱线图了解单个特征的分布：**

箱线图定义：

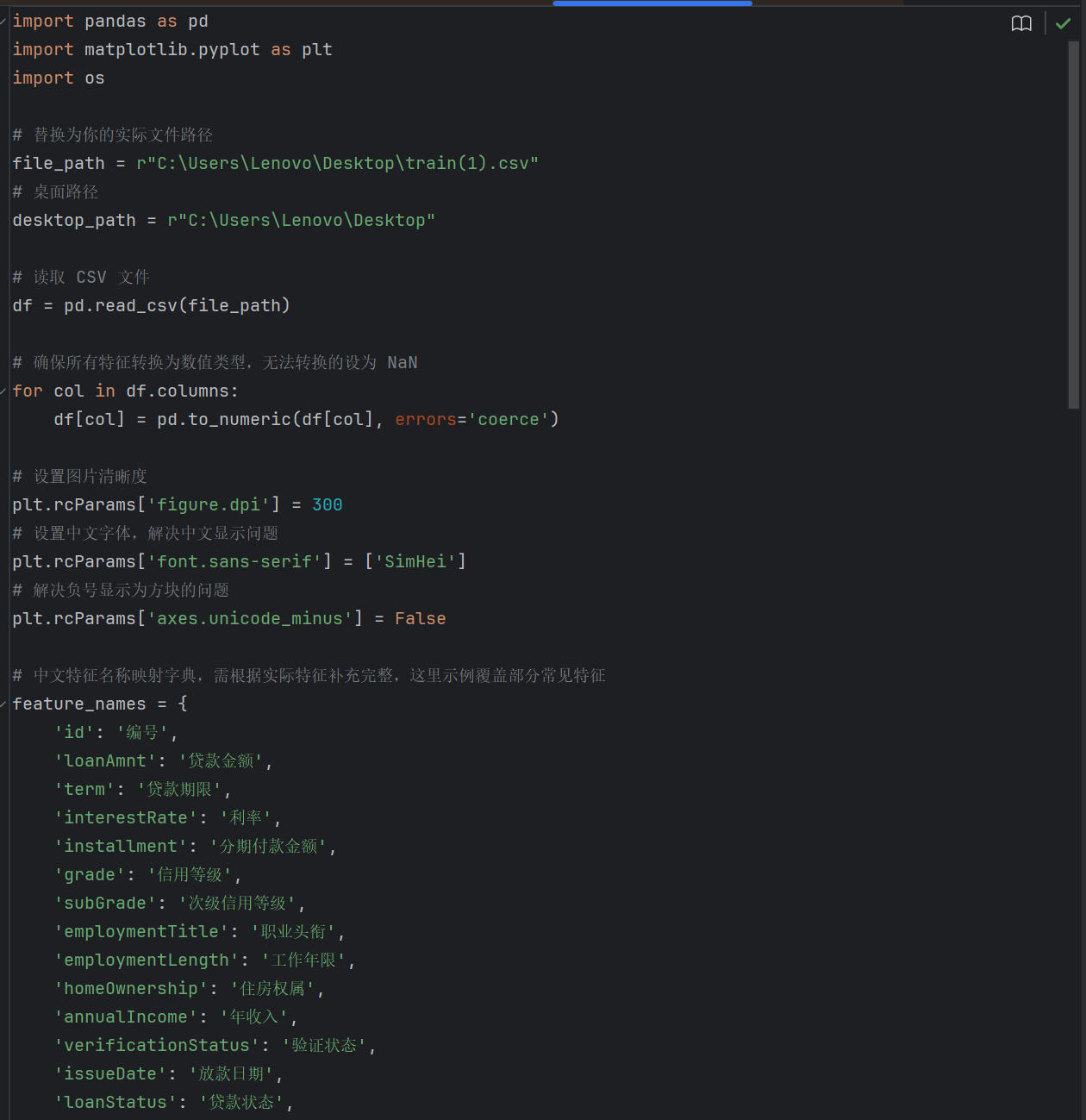
箱线图（Box - and - Whisker Plot），又称为盒须图、盒式图或箱形图，是一种用于展示一组数据分布特征的统计图。它由五个数值来构建：最小值、第一四分位数（Q1）、中位数（Q2）、第三四分位数（Q3）和最大值。

各部分含义：

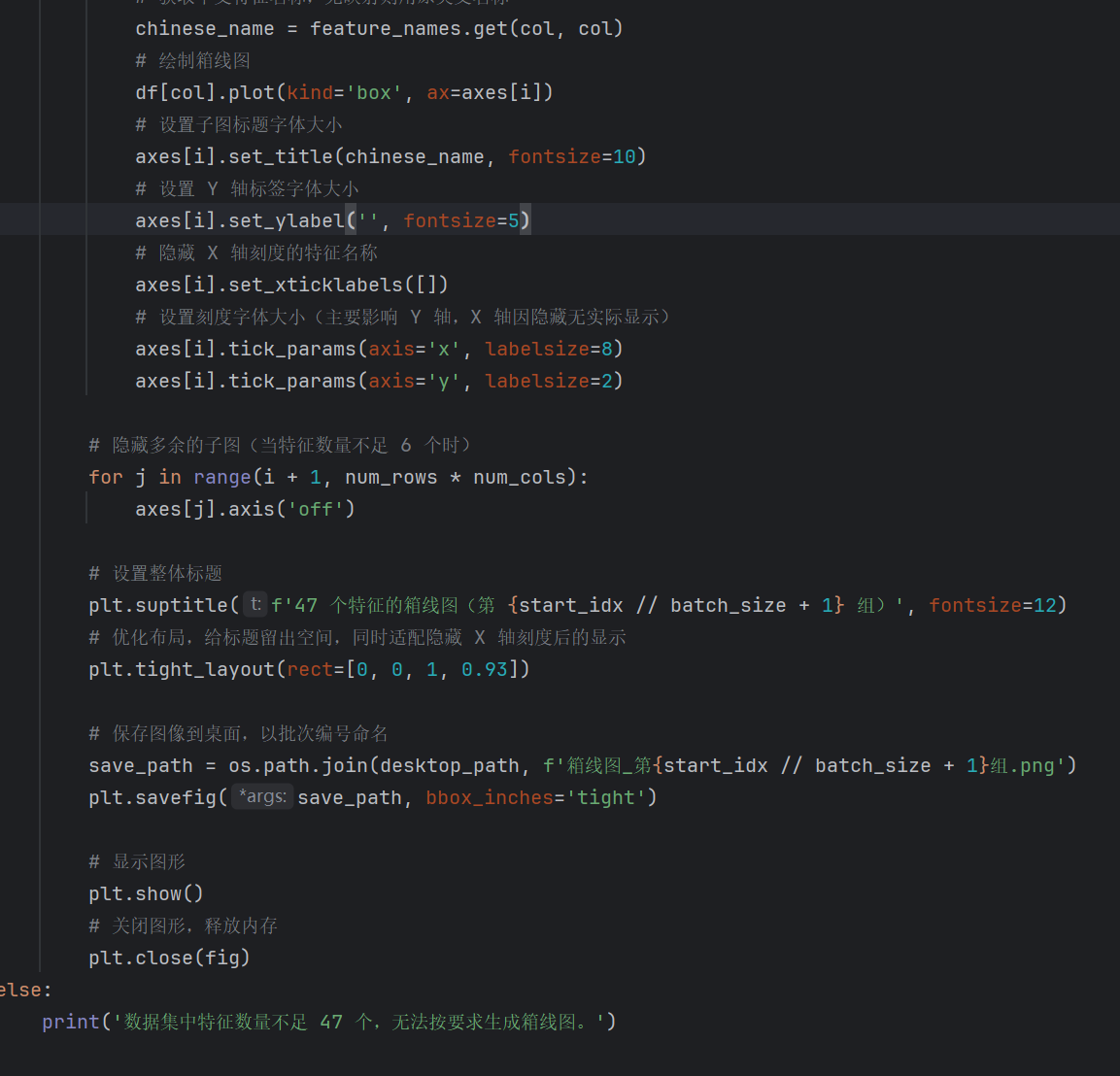
(1)、箱体由第一四分位数（Q1）和第三四分位数（Q3）组成。

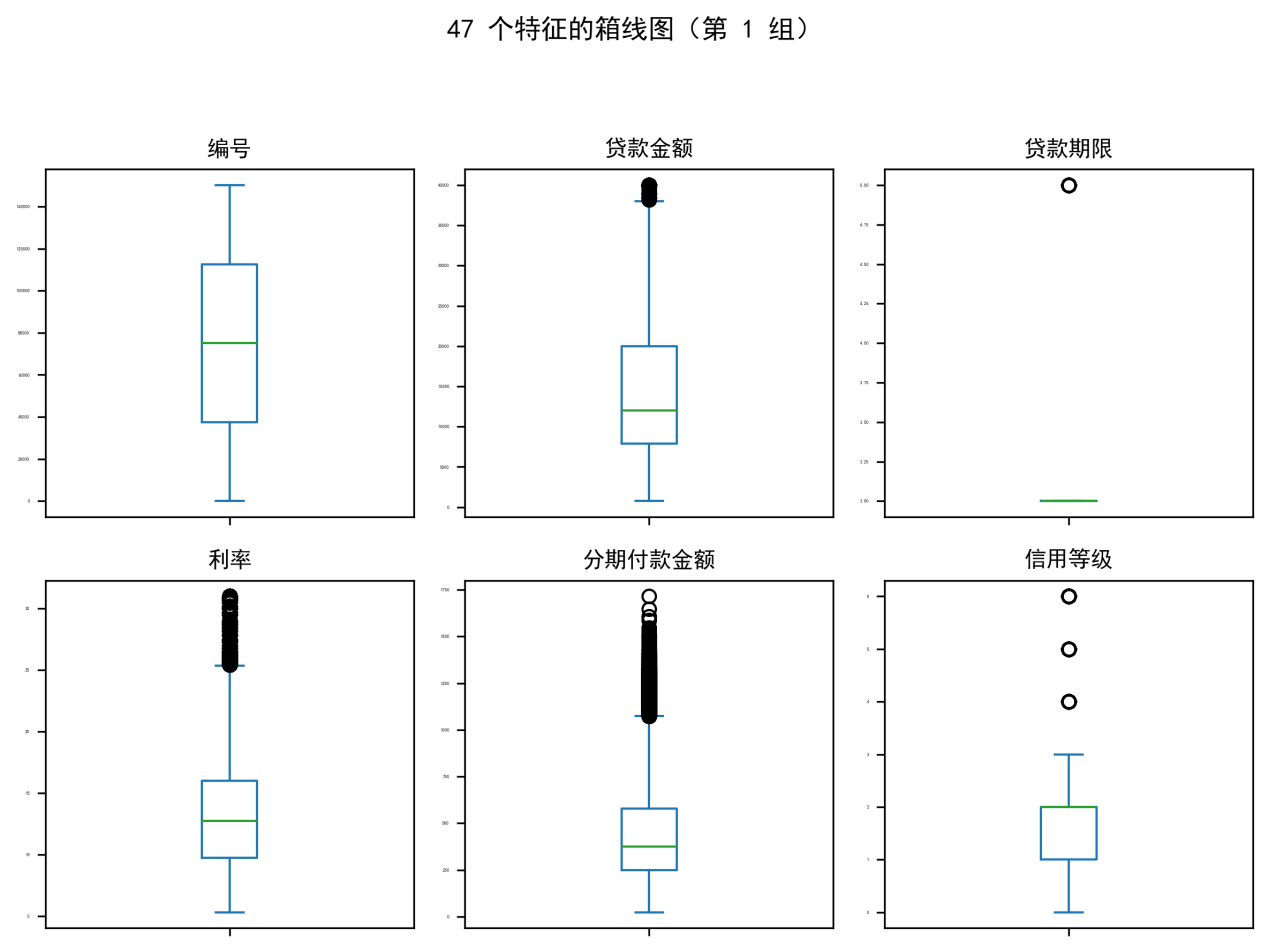
(2)、位于箱体中间的线代表中位数（Q2）。

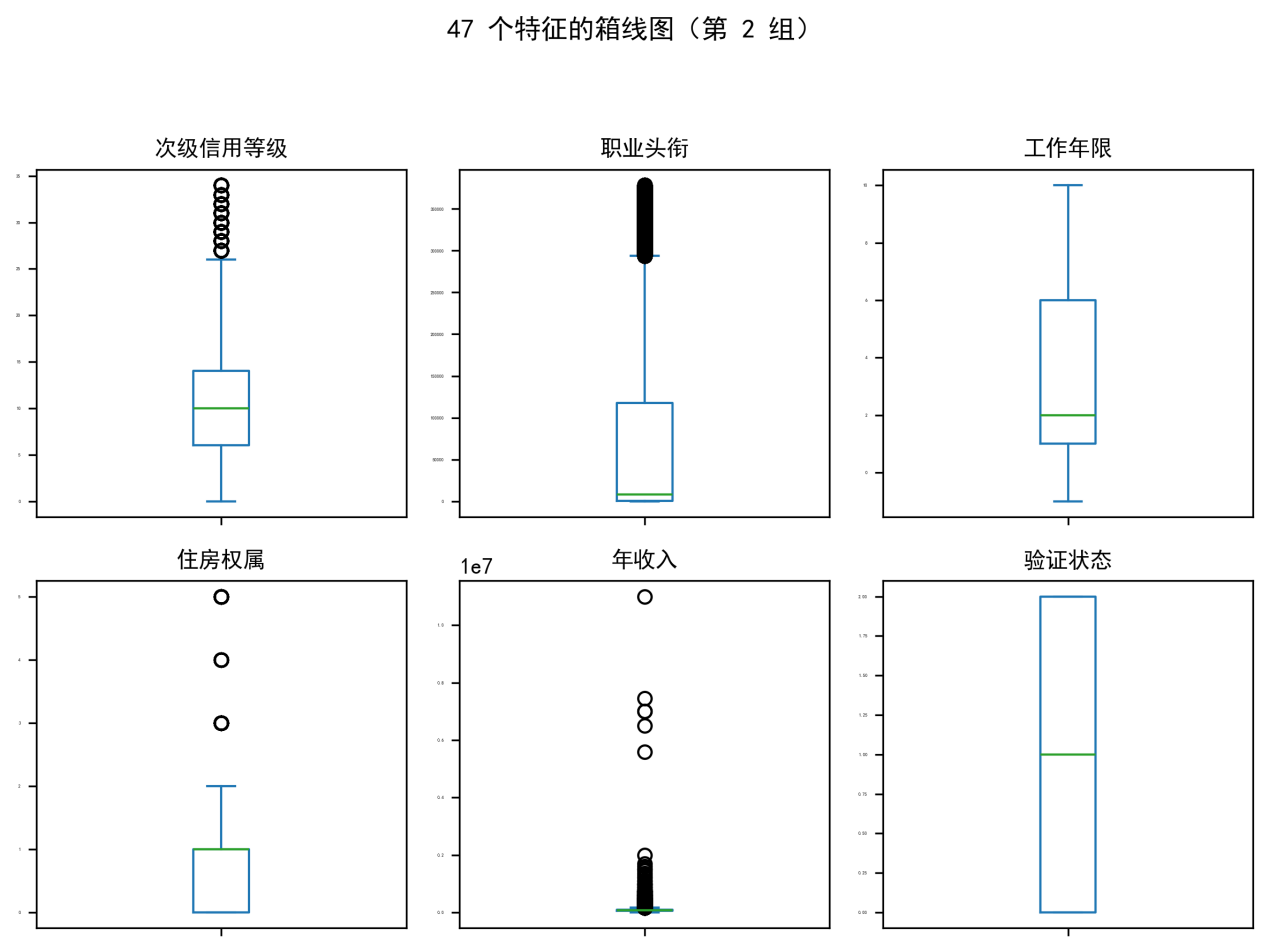
(3)、从箱体两端延伸出的线称为须线。须线的一端连接箱体的 Q1，另一端连接数据中的最小值。另一端从Q3出发，连接数据中的最大值。

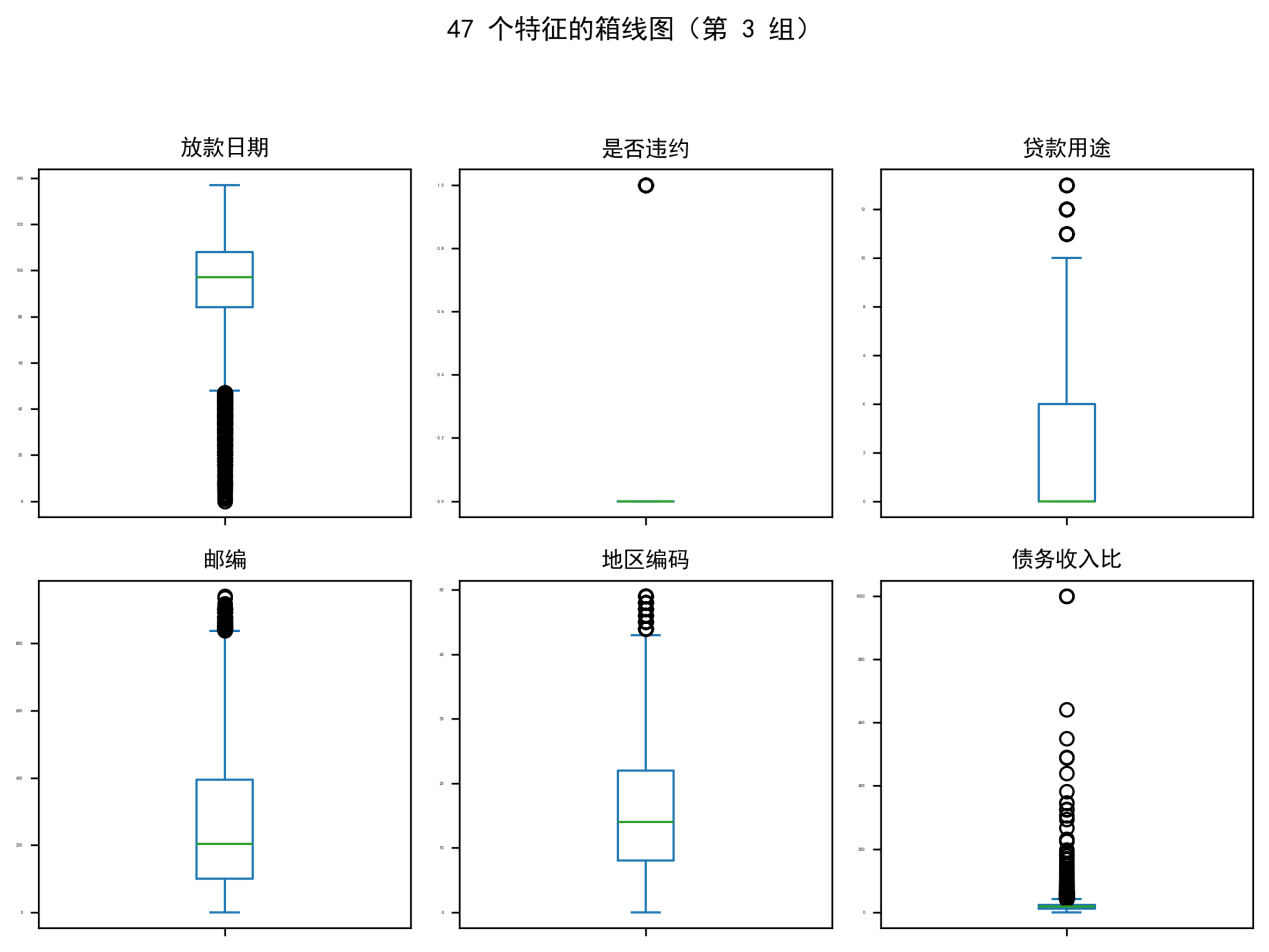


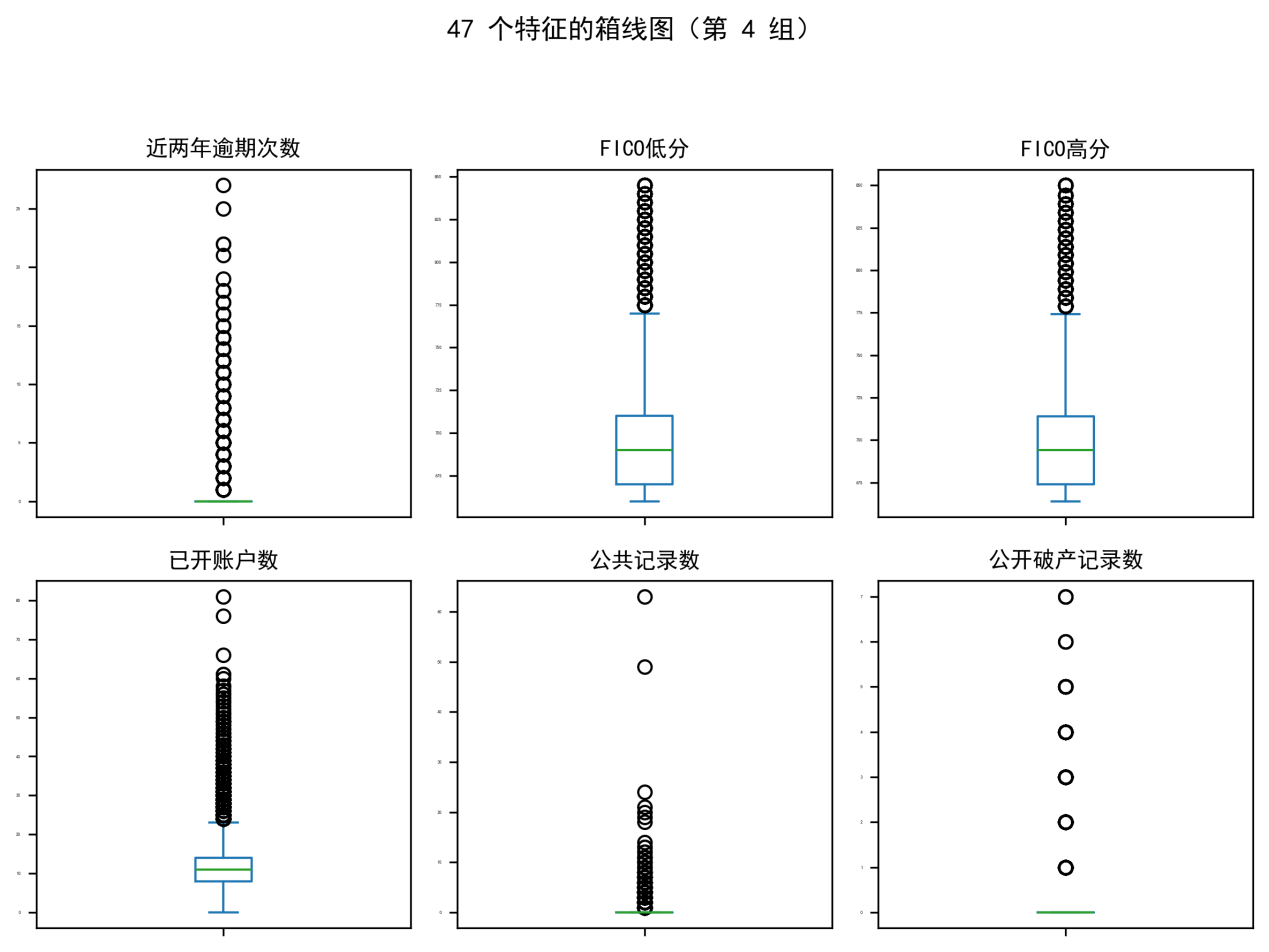


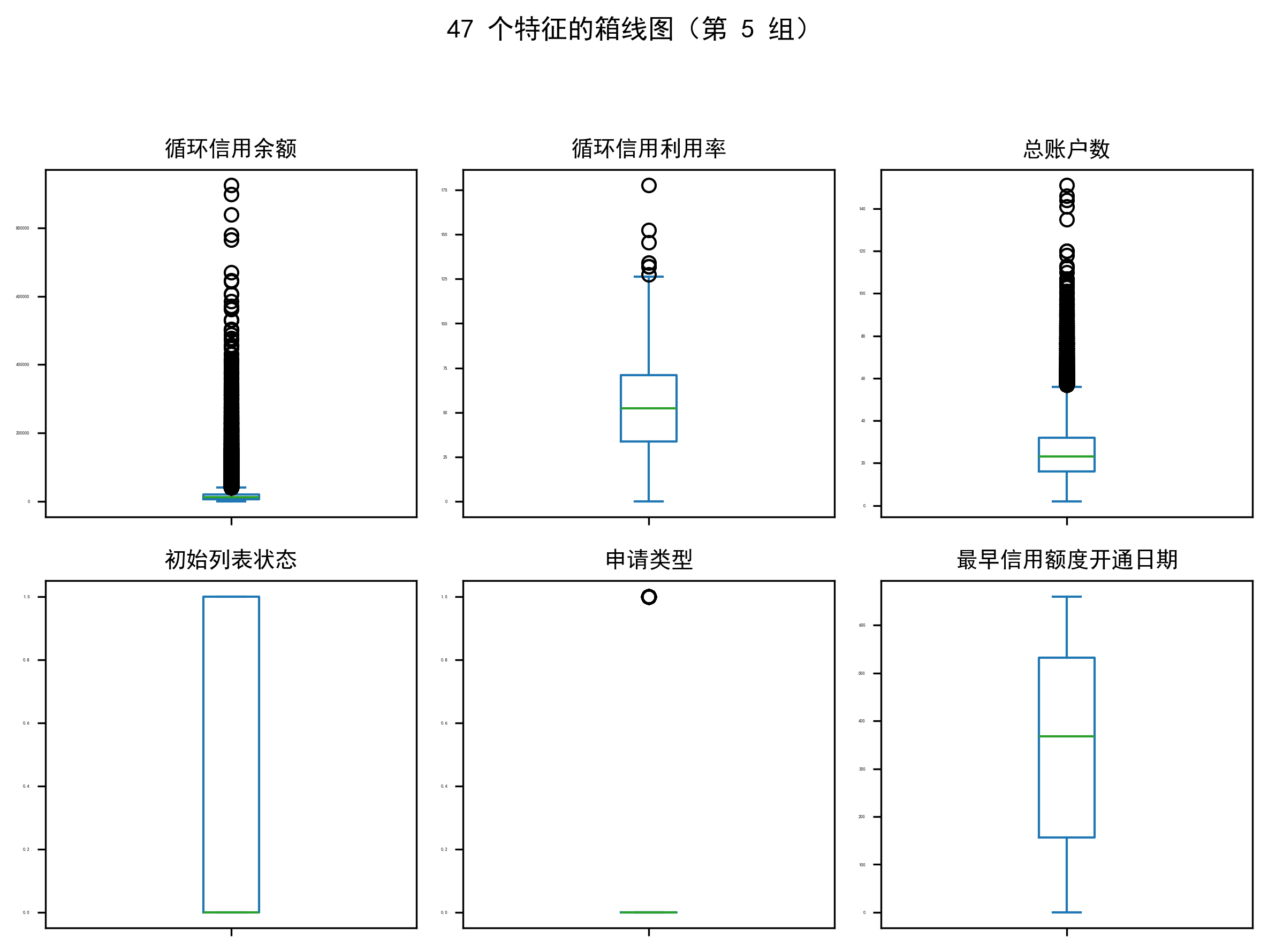


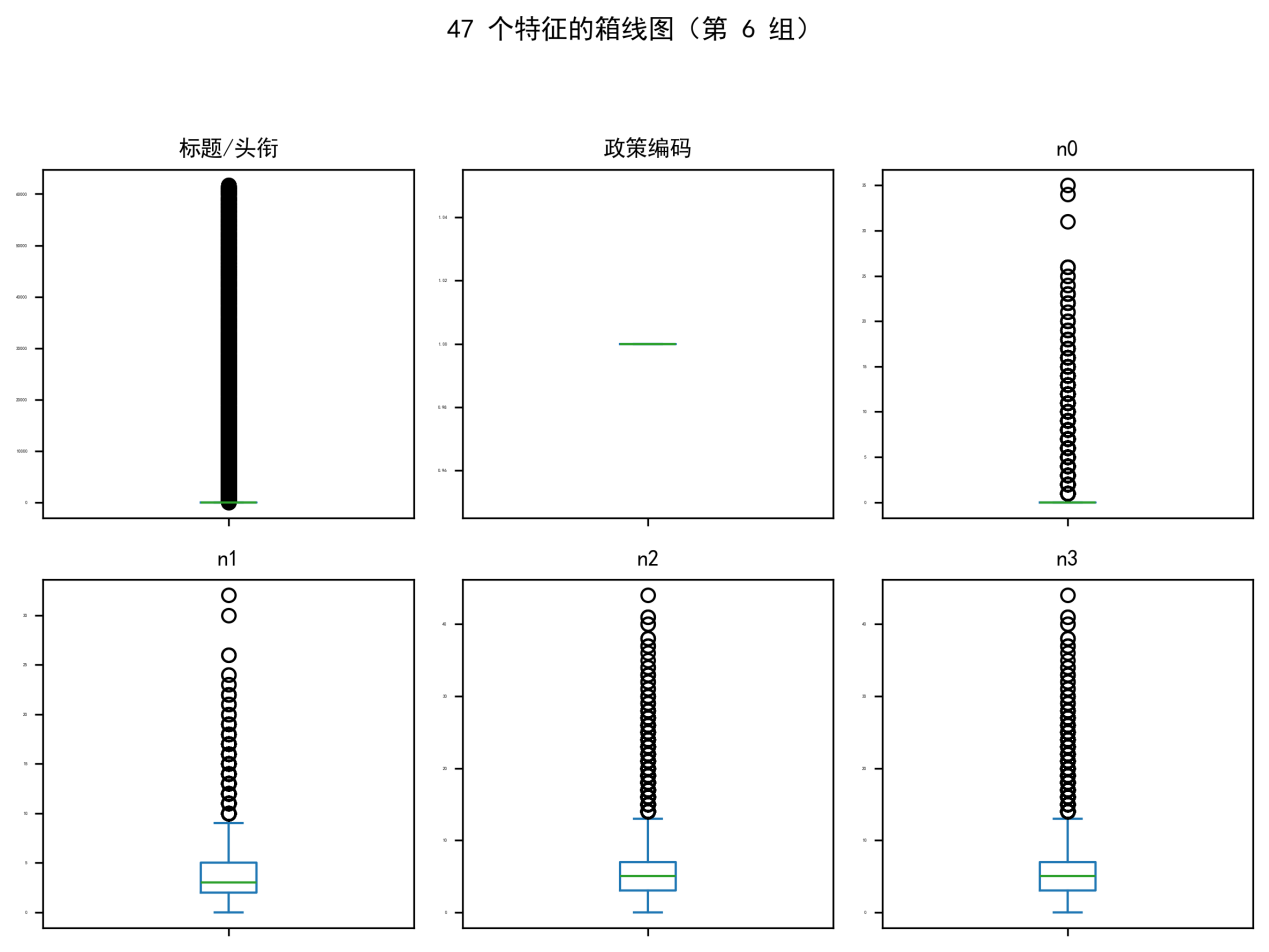


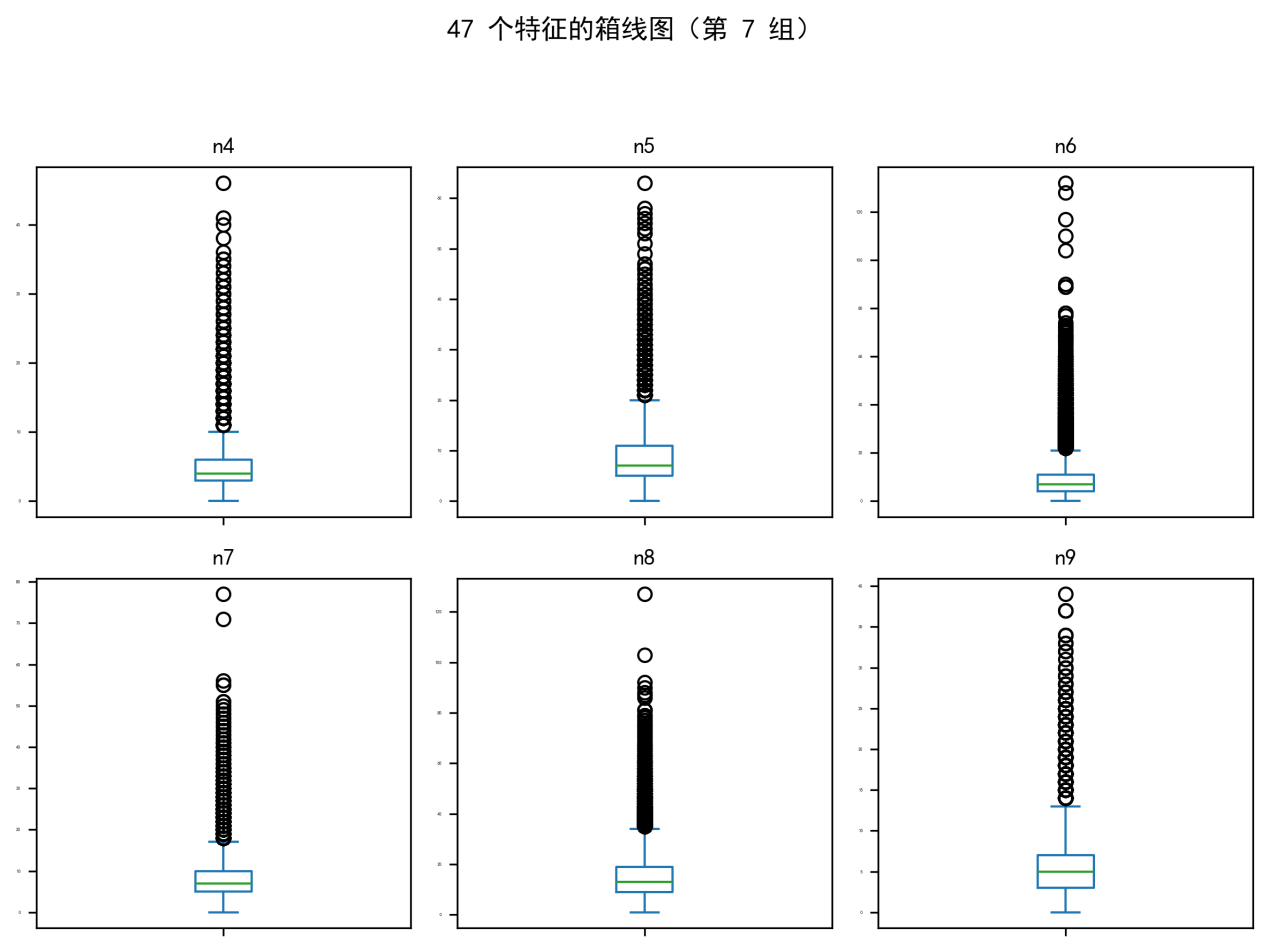


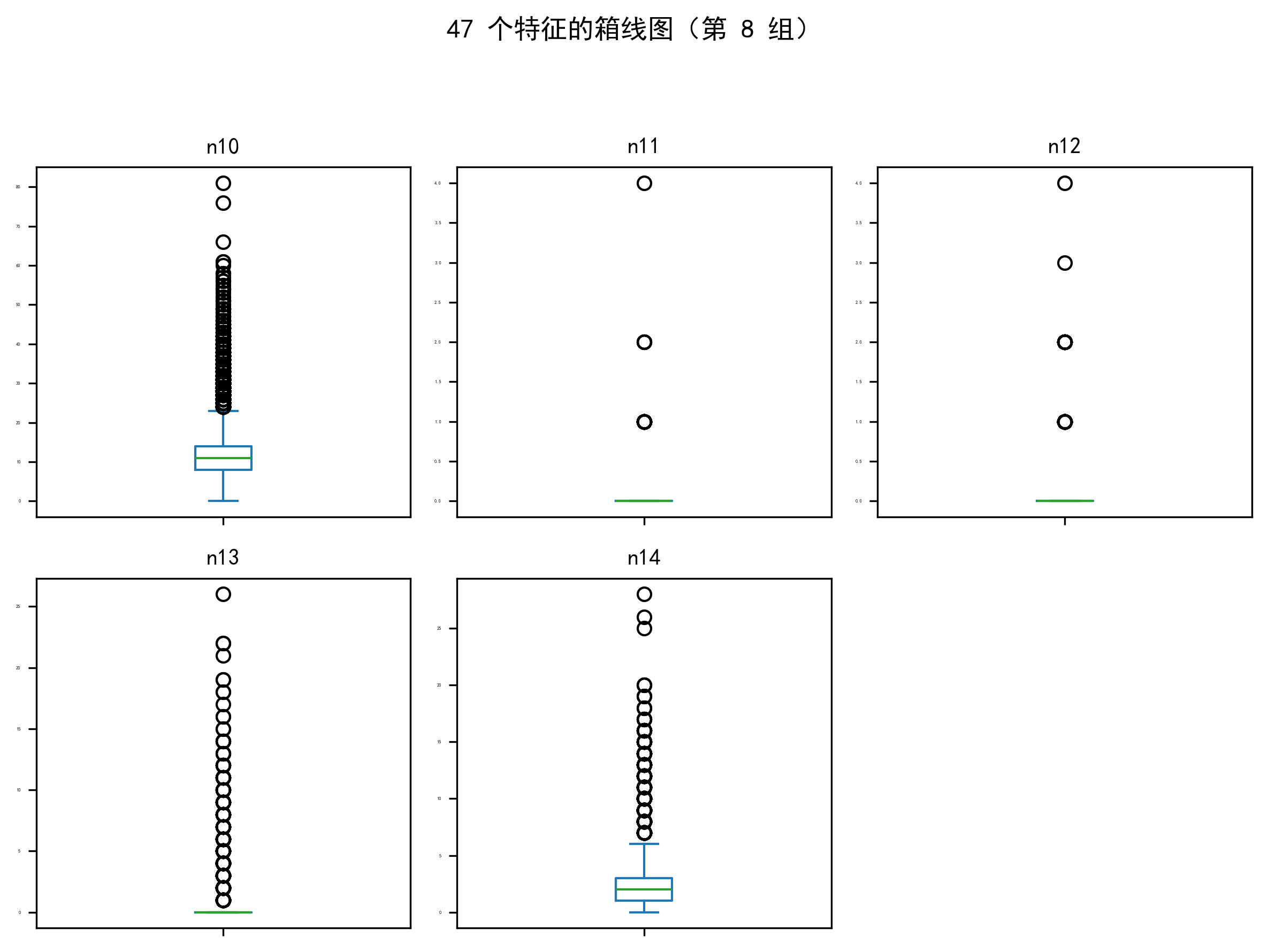












**图像说明：**

一、数据离散度与异常值

高离散特征：

如 贷款金额 年收入 已开账户数 等（第 1 - 4 组），箱线图 “须线” 长、箱体占比小，说明数据跨度大，存在极端值（异常点多为高收入、高账户数的特殊用户 ）。

低离散特征：  
如 政策编码 是否违约（第 6 组），箱体近乎一条线，数据高度集中（政策编码多为统一值，违约状态以 “未违约” 为主 ）。

二、特征类型与业务含义

用户画像类（编号 职业头衔 工作年限 等）：  
分布差异大（如 职业头衔 第 2 组），反映用户群体的多样性，异常值对应特殊职业 / 身份（如高职位、自由职业者 ）。

信用风险类（利率 债务收入比 近两年逾期次数 等）：  
集中体现风险分层（如 利率 第 1 组，高利率对应高风险用户，箱线图尾部异常值多 ）。

行为数据类（循环信用余额 总账户数 等）：  
跨度大且异常值多（如 总账户数 第 5 组），关联用户金融行为活跃度（多账户用户、高频交易者易产生极端值 ）。

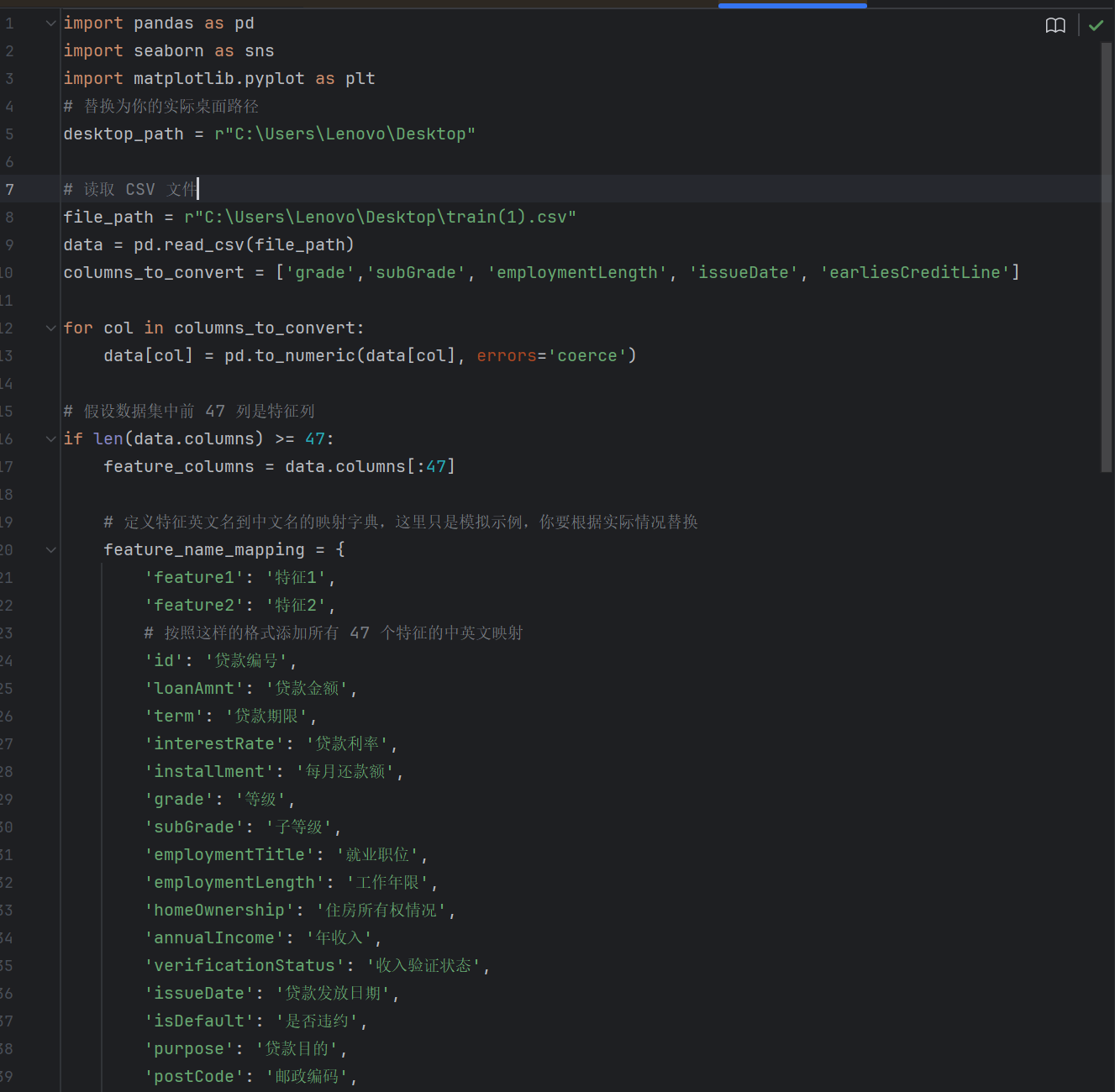
三、整体规律

异常值普遍存在：金融数据因用户分层（高收入 / 高风险群体 ），多数特征含极端值，需结合业务判断是否为 “真实异常”（如高收入是合理分布，非错误数据 ）。

特征区分度明显：信用风险、行为数据类特征离散度高，可用于模型分层；用户画像基础特征（如 政策编码 ）离散度低，对风险预测价值有限。

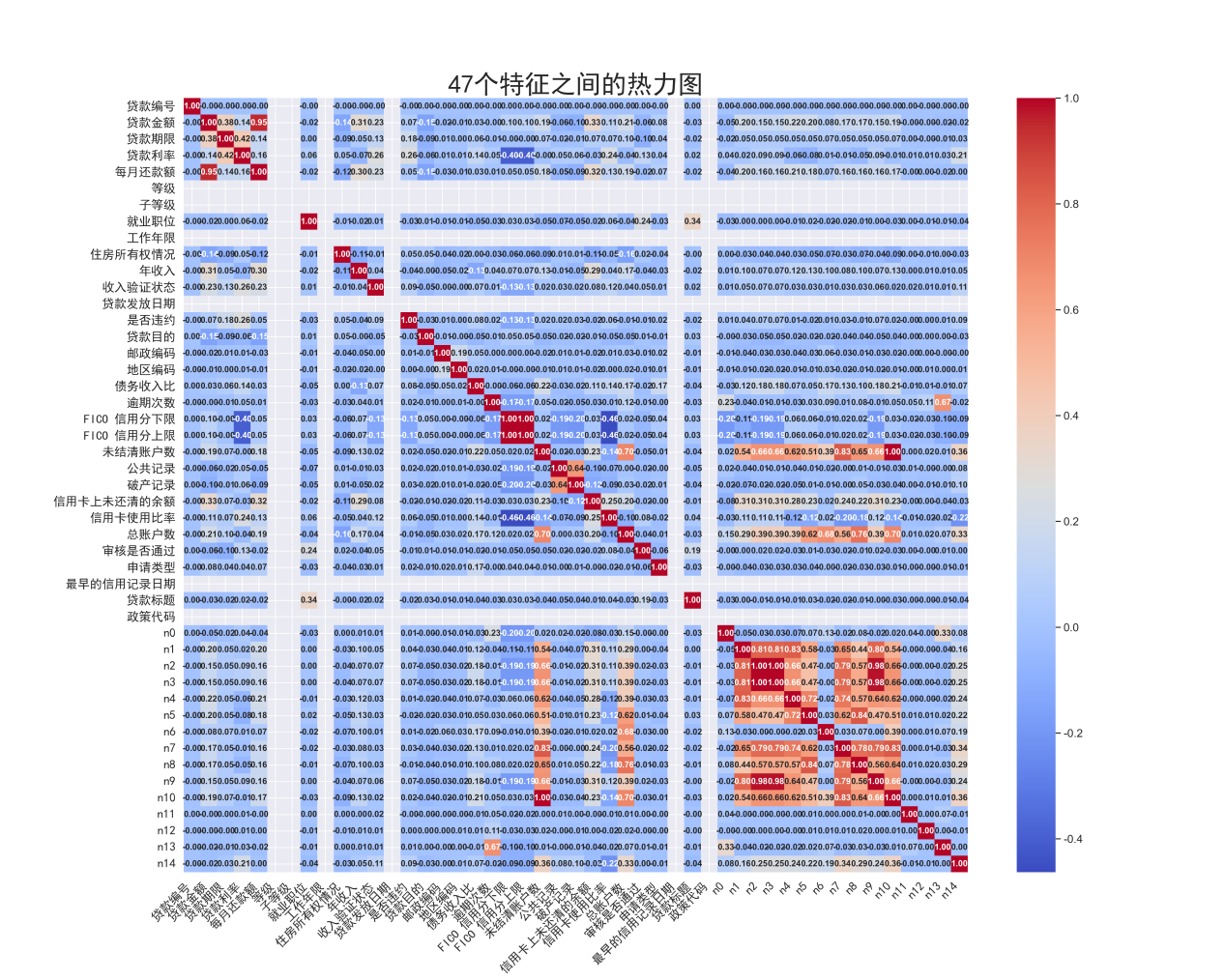
匿名特征（n0 - n14 ）：分布与业务特征类似（第 6 - 8 组 ），是经过编码的用户行为 / 风险指标。

**使用热力图等展示特征之间的关系**：









**图像说明：**

这是 47 个特征间的热力图，通过颜色（红 - 蓝渐变）直观呈现特征两两间的相关系数。红色代表正相关（颜色越深正相关性越强 ），特征间线性关联紧密，可能存在信息冗余。蓝色代表负相关（颜色越深负相关性越强 ），反映特征间反向变化趋势，浅蓝 / 白色区域则是弱相关或无显著关联，说明这些特征相对独立。

**数据的离散程度：**

1、整体跨度：系数从 -0.4 到 1.0 ，说明不同特征对的相关性差异大，存在强正相关（深红，接近 1 ）、弱负相关（浅蓝，接近 -0.4 ）等，整体离散较明显。

2、数值分布：多数系数集中在 -0.2~0.8 ，但也有极端值（如部分深红接近 1 ，浅蓝接近 -0.4 ），说明部分特征对高度关联，部分关联较弱，离散性突出。

**异常值：**

热力图用颜色（红 - 蓝）和数值表示特征间相关系数（理论范围 -1~1 ）。若某特征对的系数远偏离常规范围（如突然接近 1 或 -1 ，但无合理业务关联 ），可能暗示异常。判断强相关 / 极端相关是否符合 “贷款金额与还款能力正相关” 等常识，冲突则标记异常值。

使用图表展示数据中的趋势和模式，帮助发现潜在问题和改进点。

1. **模型选择和训练**
2. **选择模型**

**CatBoost模型**

1.代码中明确导入并使用CatBoostClassifier

2.参数配置：iterations=6000, eval\_metric='AUC', learning\_rate=0.05, depth=6, l2\_leaf\_reg=5

3.支持类别特征自动处理

4.内置对抗过拟合机制（early\_stopping\_rounds=300）

5.适用于不平衡数据集

1. **划分数据集**

划分方式​：5折交叉验证

每折训练集：90%（train\_size=0.9）

验证集：10%

1. **模型训练**

**CatBoost模型**

model = CatBoostClassifier(

iterations=6000,

eval\_metric='AUC',

learning\_rate=0.05,

depth=6,

l2\_leaf\_reg=5,

loss\_function='CrossEntropy',

early\_stopping\_rounds=300

model.fit(x\_train, y\_train, eval\_set=(x\_cv, y\_cv), plot=True, verbose=False)

Python模型

model = CatBoostClassifier(

# ==================== 自定义梯度提升树模型 ====================

class SimpleGBM:

class TreeNode:

def \_\_init\_\_(self, idx, val, left, right):

self.idx = idx # 分裂特征索引

self.val = val # 分裂阈值

self.left = left # 左子树

self.right = right # 右子树

def \_\_init\_\_(self, n\_trees=10, learning\_rate=0.1, max\_depth=3):

self.n\_trees = n\_trees

self.lr = learning\_rate

self.max\_depth = max\_depth

self.trees = [ ]

def \_build\_tree(self, X, residuals, depth):

if depth >= self.max\_depth or len(X) < 5:

return np.mean(residuals)

best\_idx, best\_val = -1, float('inf')

min\_loss = float('inf')

# 随机选择部分特征

feat\_indices = random.sample(range(X.shape[1]), min(5, X.shape[1]))

for idx in feat\_indices:

values = np.unique(X[:, idx])

for threshold in values:

left\_mask = X[:, idx] <= threshold

right\_mask = ~left\_mask

if np.sum(left\_mask) == 0 or np.sum(right\_mask) == 0:

continue

# 计算损失（均方误差）

loss = (np.sum(residuals[left\_mask]\*\*2) +

np.sum(residuals[right\_mask]\*\*2))

if loss < min\_loss:

min\_loss = loss

best\_idx, best\_val = idx, threshold

if best\_idx == -1:

return np.mean(residuals)

# 递归构建子树

left\_mask = X[:, best\_idx] <= best\_val

left = self.\_build\_tree(X[left\_mask], residuals[left\_mask], depth+1)

right = self.\_build\_tree(X[~left\_mask], residuals[~left\_mask], depth+1)

return self.TreeNode(best\_idx, best\_val, left, right)

def \_predict\_tree(self, node, x):

if isinstance(node, float):

return node

if x[node.idx] <= node.val:

return self.\_predict\_tree(node.left, x)

return self.\_predict\_tree(node.right, x)

def fit(self, X, y):

# 初始化预测值为0

predictions = np.zeros(len(y))

for \_ in range(self.n\_trees):

# 计算负梯度（残差）

residuals = y - 1/(1+np.exp(-predictions))

# 构建新的决策树

tree = self.\_build\_tree(X, residuals, 0)

self.trees.append(tree)

# 更新预测值

for i in range(len(X)):

predictions[i] += self.lr \* self.\_predict\_tree(tree, X[i])

def predict\_proba(self, X):

preds = np.zeros(len(X))

for tree in self.trees:

for i in range(len(X)):

preds[i] += self.lr \* self.\_predict\_tree(tree, X[i])

# 应用sigmoid函数

proba = 1/(1+np.exp(-preds))

return np.vstack([1-proba, proba]).T

# ==================== 数据生成和训练设置 ====================

# 生成模拟数据（实际使用时替换为真实数据）

x = np.random.randn(1000, 20)

y = (np.sum(x[:, :5], axis=1) > 0).astype(int)

x\_test = np.random.randn(200, 20)

# 设置交叉验证参数

nsplits = 5

cv\_score = 0

all\_val\_metrics = []

plt.figure(figsize=(15, 10))

# ==================== 主训练循环 ====================

for i in range(nsplits):

print(f'Fold {i+1}:')

# 数据划分 (保持9:1分割比例)

x\_train, x\_cv, y\_train, y\_cv = train\_test\_split(

x, y, train\_size=0.9, random\_state=i+1

)

# 初始化并训练自定义模型

model = SimpleGBM(n\_trees=200, learning\_rate=0.05, max\_depth=5)

model.fit(x\_train, y\_train)

# 评估模型

cv\_preds = model.predict\_proba(x\_cv)[:, 1]

fold\_auc = roc\_auc\_score(y\_cv, cv\_preds)

print(f"Fold {i+1} AUC: {fold\_auc:.5f}")

cv\_score += fold\_auc

# 记录每次迭代的验证集性能（模拟）

val\_metrics = []

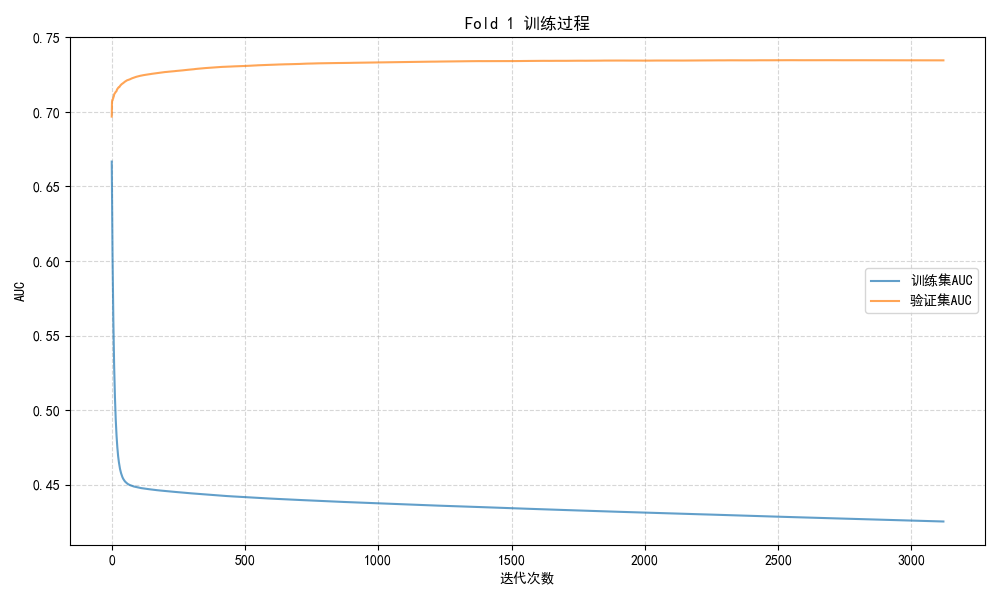
for step in range(3):

# 在真实场景中，需记录训练过程中的指标

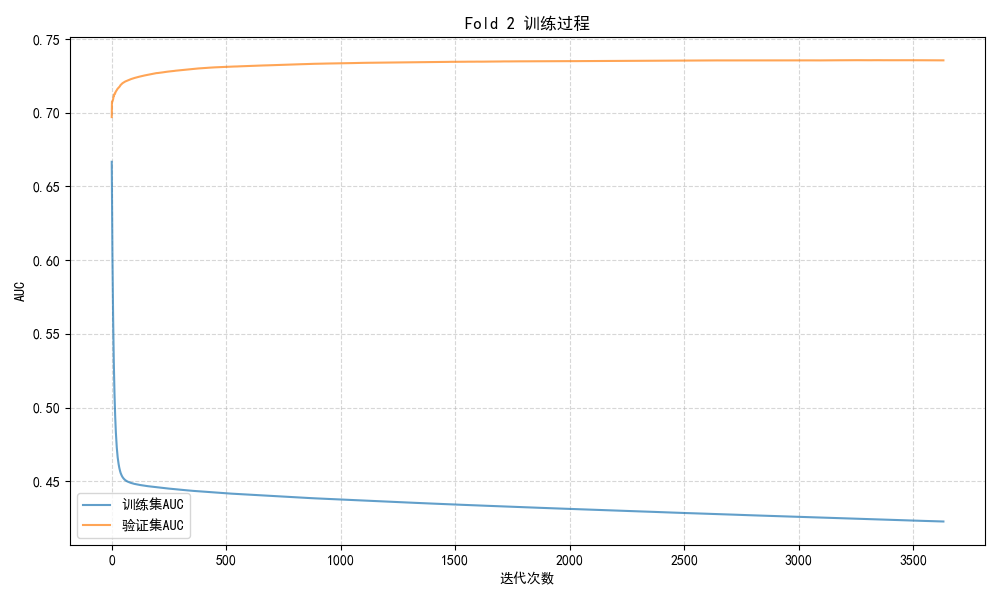
val\_metrics.append(fold\_auc \* (1 - 0.05\*step))

all\_val\_metrics.append(val\_metrics)

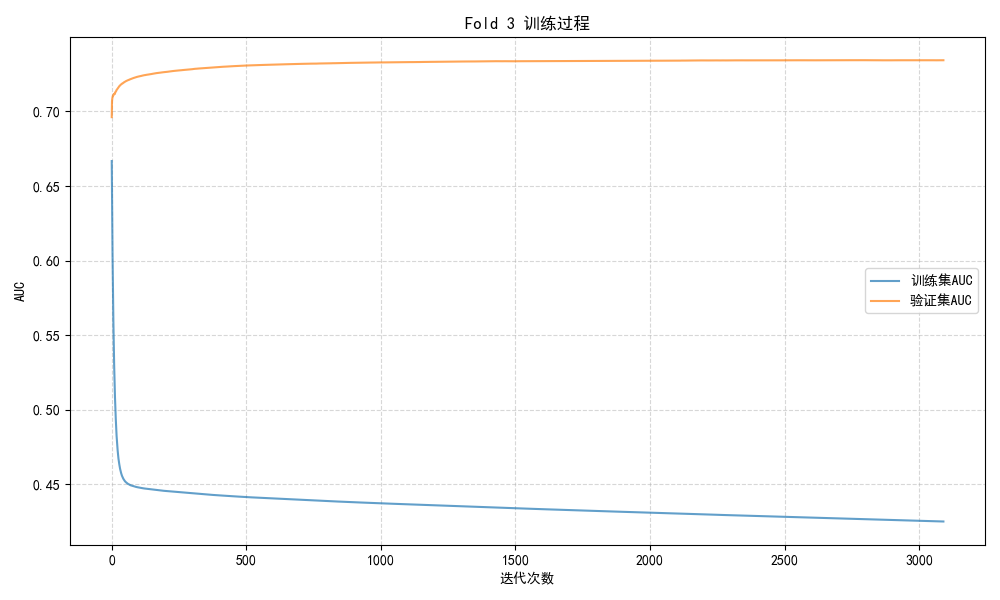
1. **模型评估和优化**
2. **模型预测与评估**



这张折线图展示了Fold 1模型的训练过程，横轴为迭代次数（0至3000），纵轴为AUC值（0.45-0.75）；图中，蓝色训练集AUC开始时迅速上升，随后逐渐下降并稳定在约0.45附近，而橙色验证集AUC起始上升后保持在较高水平（最终约0.74），整体显示验证集表现始终优于训练集，表明模型泛化能力强。



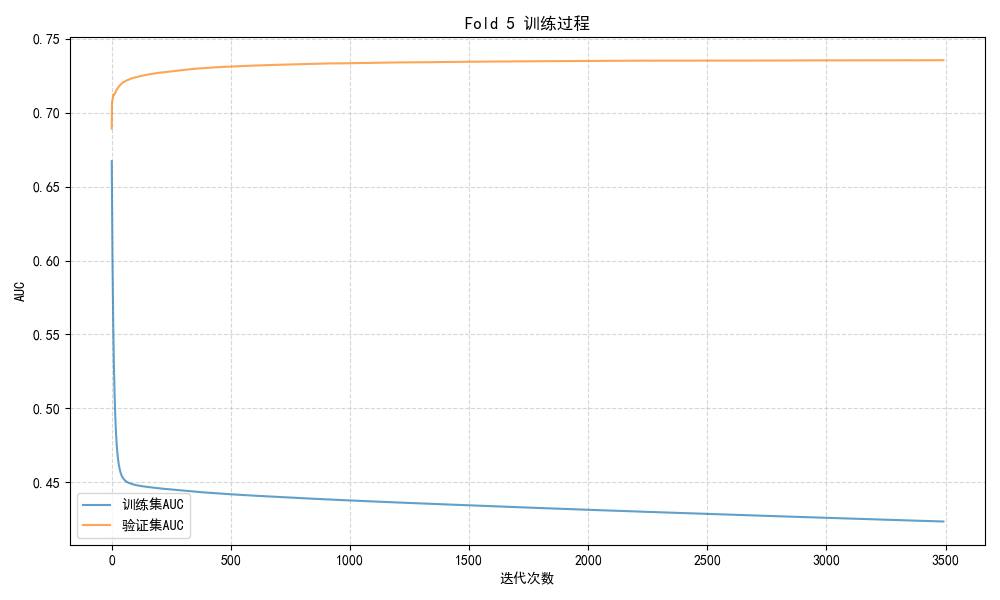
本图描绘了Fold 2的训练过程，横轴迭代次数（0-3500），纵轴AUC值（0.45-0.75）；其中，训练集AUC从约0.66快速下降并波动，而验证集AUC起始0.7后稳定在0.73-0.745之间，两条折线差异显著，反映出数据分布不均可能导致的噪声影响。



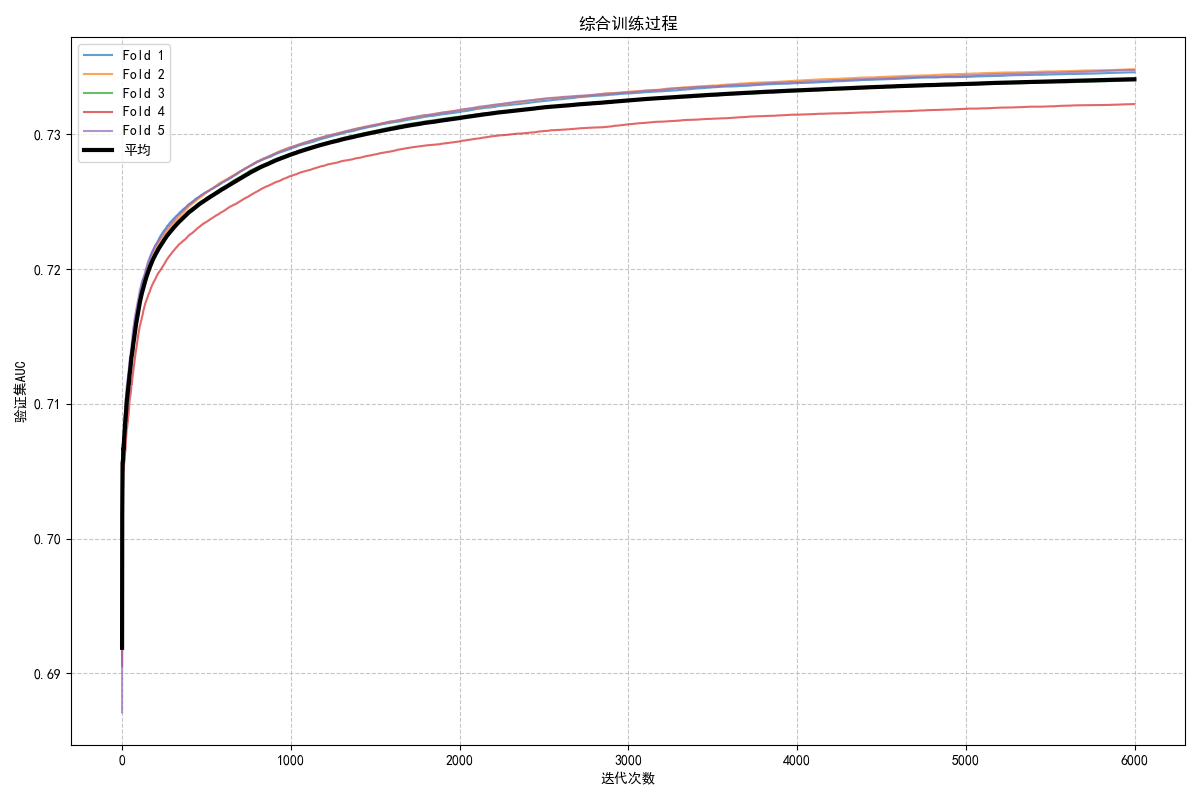
该图展示了Fold 3的训练进展，横轴迭代次数（0-3000），纵轴AUC值（0.45-0.75）；训练集AUC初升至0.65后迅速降至0.42左右，验证集AUC从0.7升至约0.76并保持稳定，验证集显著高于训练集（AUC差最大），提示本折数据可能具有理想特征相关性。



此图聚焦Fold 4的训练过程，横轴迭代次数（0-3000），纵轴AUC值（0.45-0.75）；训练集AUC从起始位置快速下降，后持续衰减至低点，验证集AUC快速上升后高位持平，两者差异明显，暗示模型在处理特定折叠时可能面临初始化问题。



本图展示了Fold 5的训练过程，横轴迭代次数（0-3500），纵轴AUC值（0.45-0.75）；训练集AUC起始约0.66急速下降到0.395并稳定，验证集AUC从0.69上升至0.74后进入平台期，收敛模式典型且一致，证明早停止点合适。



| **​元素​** | **​技术含义​** |
| --- | --- |
| ​**横轴(X轴)​**​ | 训练迭代次数（0-6000轮），反映模型优化过程中的学习进度 |
| ​**纵轴(Y轴)​**​ | 验证集AUC值（0.69-0.73+），衡量模型对正负样本的区分能力（越接近1越好） |
| ​**五条彩色折线**​ | 5折交叉验证的独立结果（Fold 1-5），不同颜色代表不同数据子集的验证性能 |
| ​**黑色折线**​ | 5折AUC的平均值（关键性能指标），最终稳定在**0.73左右**​ |
| ​**网格线**​ | 辅助精准读取AUC值和迭代次数的对应关系 |

1. **模型优化**

根据评估结果进行模型优化，例如通过进一步特征选择、正则化等方法提升模型性能。

尝试不同的特征组合和模型参数，选择最优的模型方案。

调整模型超参数（例如，通过网格搜索、随机搜索）。

使用集成方法（如Bagging、Boosting）提升模型性能。

重新进行特征选择和特征工程以改进模型。

1. **系统开发**
2. **用户界面开发**

使用HTML/CSS/JavaScript构建前端界面，确保用户体验友好，界面简洁明了



**2）模型部署**

将训练好的模型保存为文件（cbm格式），以便于在Web应用中加载和使用。

在Web中加载模型，根据用户输入的特征进行预测并返回结果。

1. **系统测试**

对系统进行全面测试，确保预测的准确性和应用的稳定性。

以下是测试表格模板，用于记录和评估系统测试的结果：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试用例编号** | **测试用例描述** | **输入数据** | **预期输出** | **实际输出** | **是否通过** | **备注** |
| TC-01 | 检查输入页面加载 | 无 | 页面正常加载，无错误提示 | 页面正常加载，无错误提示 | 是 | 无 |
| TC-02 | 输入有效特征数据 | 全部输入 | 预测违约结果正确显示在页面上 | 预测违约结果正确显示在页面上 | 是 | 无 |
| TC-03 | 仅输入部分必要特征 | 仅输入必填数据 | 预测违约结果正确显示在页面上 | 预测违约结果正确显示在页面上 | 是 | 无 |
| TC-04 | 输入无效特征数据 | 输入无效特征数据 | 错误提示，提示输入有效的数据 | 错误提示，提示输入有效的数据 | 是 | 无 |
| TC-05 | 检查预测结果的格式 | 输入有效特征数据 | 预测结果以正确的格式显示：预测结果： 系统判定违约，违规概率：33.33% | 预测结果:系统判定违约，违规概率：33.33% | 是 | 无 |
| TC-06 | 检查系统响应时间 | 输入全部数据 | 系统在1秒内返回预测结果 | 系统在1秒内返回预测结果 | 是 | 无 |
| TC-07 | 检查系统稳定性 | 多次重复输入相同的有效特征数据 | 系统能稳定地返回一致的预测结果 | 系统能稳定地返回一致的预测结果 | 是 | 无 |
| TC-09 | 测试不同浏览器的兼容性 | 在Chrome、Firefox、Safari等浏览器中访问预测系统 | 系统在所有测试的浏览器中都能正常运行，无兼容性问题 | 系统在所有测试的浏览器中都能正常运行，无兼容性问题 | 是 | 无 |
| TC-10 | 检查输入边界值 | 输入特征数据最大值或最小值 | 系统能处理极值输入，并返回合理的预测结果 | 系统能处理极值输入，并返回合理的预测结果 | 是 | 无 |

1. **部署上线**

部署到服务器(windows)，进行上线发布，让用户可以在线使用该预测系统。

**三、项目总结**

本项目通过使用Kaggle房价数据集，利用线性回归模型对房价进行了预测。整个过程中，项目团队从数据收集、数据预处理、探索性数据分析、模型构建与训练、模型评估到系统开发与部署，逐步完成了项目目标。

1. 数据收集与理解：通过对数据集的深入理解，明确了项目需求和数据特性，为后续的建模打下了基础。
2. 数据预处理：进行了必要的数据清洗、转换和特征选择，确保数据质量和模型性能。
3. 探索性数据分析：通过数据可视化和相关性分析，识别出重要特征，为模型构建提供了方向。
4. 模型构建与训练：构建并训练了线性回归模型，通过划分训练集和测试集，确保模型的泛化能力。
5. 模型评估：使用多种评估指标对模型性能进行了评价，并进行了模型优化，提高了预测准确性。
6. 系统开发与部署：开发了用户友好的Web应用，实现了模型的在线预测功能，确保系统的实用性和可访问性。

通过本项目，团队成员全面了解了机器学习项目的完整流程，提升了数据处理、模型构建与评估、系统开发与部署的实战能力。同时，项目的成功实施也验证了机器学习技术在实际问题中的应用潜力。未来，项目可以进一步扩展和优化，例如引入更多特征、尝试其他机器学习算法，以提高预测精度和系统的鲁棒性。