上海大学
Shanghai University

shanghai university

信用卡欺诈预测

范舒舰
23121677

从数据清洗到
机器学习建模

上海大学
Shanghai University

实验流程

1 项目背景与研究目标

明确任务：基于信用卡交易数据，识别潜在欺诈交易并构建预测模型。

→

2 数据理解与数据清洗

认识数据字段、样本分布与质量问题，并完成缺失检查、异常处理和字段规范化。

→

3 数据分析与样本筛选

结合统计结果与业务含义，对原始样本进行筛选，构建更适合建模的数据集。

→

4 EDA发现

通过类别分布、相关性和特征分布图，观察正常交易与欺诈交易的差异特征。

5 特征工程设计

对时间、金额及匿名特征进行处理，并完成特征选择、缩放与样本平衡。

→

6 模型设计与训练过程

选择逻辑回归、随机森林和 XGBoost 等模型，完成参数设置、训练与调优。

→

7 实验结果与模型分析

结合混淆矩阵、Precision、Recall、F1 和 AUC，比较不同模型的效果与特点。

→

8 改进总结

在复现原实验的基础上完成 CPU 规范改进与 GPU 适配，并总结可继续优化的方向。

1

上海大学
Shanghai University

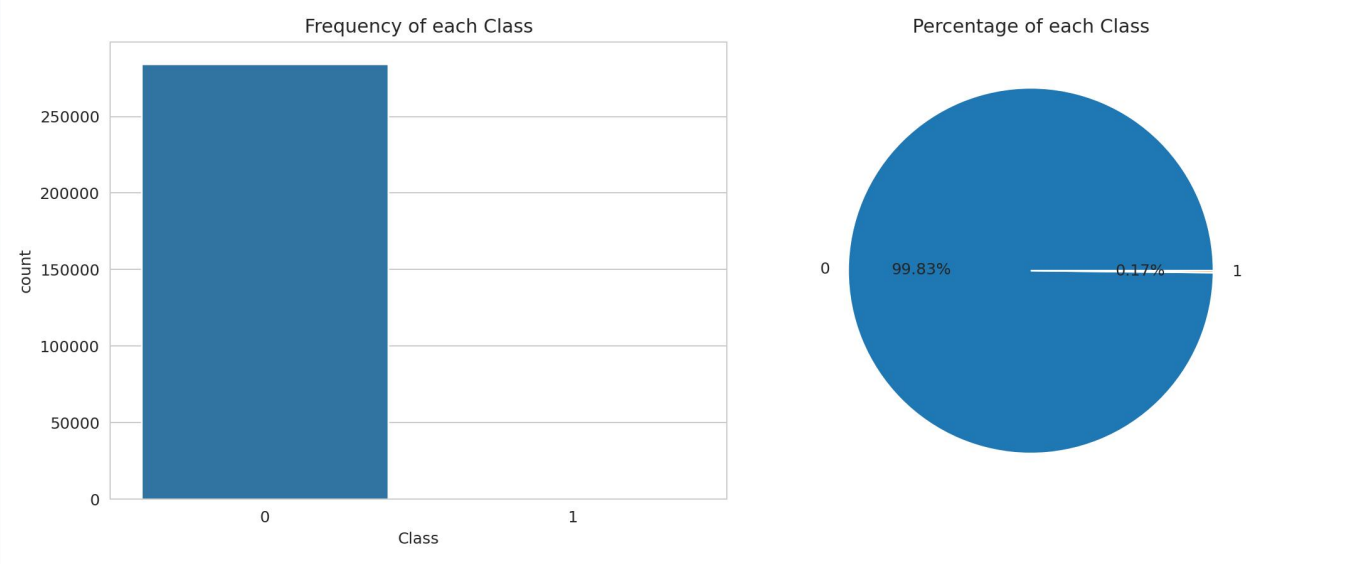
项目背景

项目背景

信用卡交易量巨大、交易频率高，欺诈交易一旦放行，会直接带来资金损失与风控压力。实验目标就是利用历史交易数据，尽早识别高风险交易。

样本背景

本任务具有典型的数据挖掘特征：样本量大、特征多、类别极不平衡，因此既要做数据分析，也要设计合理的训练与评估流程。



总样本	欺诈样本	欺诈占比
284,807	492	0.173%
	极少数类	

1

上海大学 Shanghai University

研究目标

核心任务

输入一笔交易的数值特征，输出其是否为欺诈。任务本质是监督学习中的二分类问题。

实验目标

先复现原始训练思路，再根据实验细节进行规范化改进，最后完成 GPU 适配。

评估目标

从少数类识别效果、整体分类性能和阈值敏感性三个方面评估模型，重点关注 Recall、F1、PR-AUC、ROC-AUC 及不同阈值下混淆矩阵的表现。

指标选择

Accuracy 不能反映少数类识别效果。实验重点放在 Recall、Precision、F1、PR-AUC、ROC-AUC 以及阈值下的混淆矩阵。

风控侧重点

欺诈检测更关注“漏报”和“误报”的平衡。Recall 过低会漏掉风险交易，Precision 过低会增加审核成本。

主要指标

F1

阈值分析

Recall / Precision

曲线分析

ROC-AUC

曲线分析

PR-AUC

2

上海大学
Shanghai University

数据理解

样本量	原始字段	缺失值	金额中位数	金额最大值
284,807	31	0	22.0	25691.16
	含标签列	无空值		

字段结构

- Time：交易时间
- V1–V28：匿名化后的主特征
- Amount：交易金额
- Class：标签，1 为欺诈

数据特征

原始业务语义已匿名化，建模重点落在分布分析、相关性分析、降维观察和分类器性能，而不是人工编写复杂规则。

建模含义

高维数值特征 + 极端不平衡标签，是整个实验设计的前提。后续所有清洗、特征工程、模型训练和评估都围绕这两个特点展开。

2

上海大学
Shanghai University

数据清洗

处理目标
完成基础清洗与字段转换，使数据能够进入分类模型。

时间处理
Time 转为 Hour(0~47)，表示两天内的小时位置。

金额与字段
对 Amount 做缩放，并删除部分匿名特征后进入训练。

主要步骤
- 检查缺失值，确认数据集无空值。
- 构造 Hour 字段，用于替代原始 Time。
- 对金额与时间字段做缩放处理。
- 保留建模字段，删除部分变量。

缺失检查	时间特征	金额处理	流程方式
无缺失	**Hour(0~47)**	**直接缩放**	**Notebook 串联**

2

上海大学
Shanghai University

数据清洗

改进目标
在保留原实验主线的基础上，提高特征表达合理性与流程规范性。

时间改进
Hour 改为 HourSin / HourCos，保留周期信息。

金额改进
Amount 先做 log1p，再进入模型。

主要改进
- 保留“数据无缺失”的原始结论。
- 保留全部匿名特征 V1–V28。
- 新增 LogAmount、HourSin、HourCos 等派生特征。
- 缩放与采样只在训练流程内部完成。
- 将 Notebook 串联流程拆为独立脚本，便于复现。

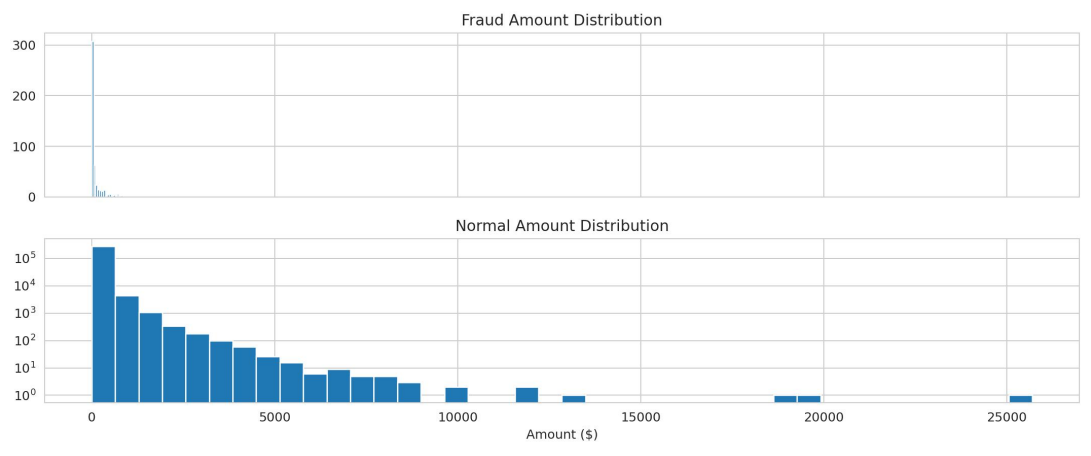
改进后的意义
- 不只是让数据“能训练”，而是让特征表达更合理。
- 减少预处理与训练耦合，提高实验复现性。
- 更适合后续模型比较、阈值评估和 GPU 适配。
- 为更规范的 train / val / test 划分打下基础。

时间特征	金额处理	特征保留	流程方式
HourSin / HourCos	**log1p**	**V1–V28 全保留**	**独立脚本拆分**

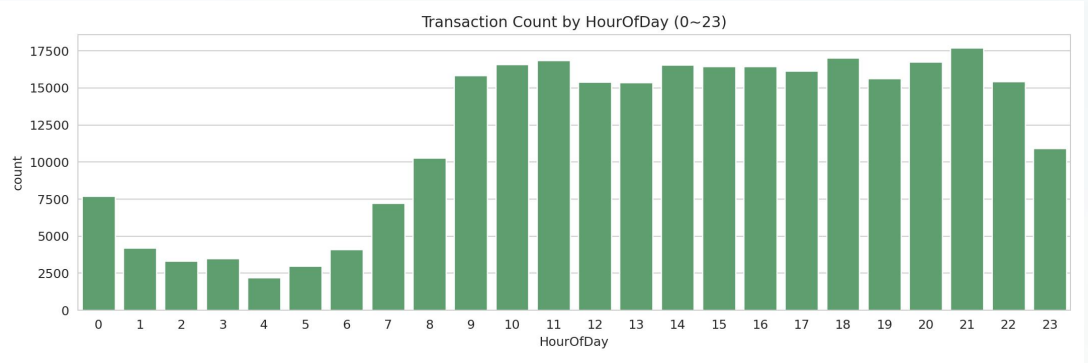
3

上海大学
Shanghai University

数据分析



左图：欺诈样本与正常样本的交易金额分布（对数坐标）



右图：一天 24 小时内的交易次数分布

图表含义

金额分布是否偏态，时间分布是否存在规律。图像本身是观察依据，结论必须从图像中提炼。

金额结论

正常交易金额呈明显长尾分布，低金额交易占绝大多数，少量高金额样本拉长尾部；因此金额特征不能直接按线性尺度理解。

时间结论

交易次数在 24 小时内分布不均，白天和晚间较活跃，凌晨明显下降；时间不是随机噪声，而是具有周期结构的特征。

- 金额分布支撑了后续对 Amount 做 log1p 处理
- 时间分布支撑了后续把 Hour 改写为 HourSin / HourCos
- 提取可用于建模的结构信息

3

上海大学
Shanghai University

样本筛选

Train	Validation	Test	特征数
170884	56961	56962	31
0.173% 欺诈占比	0.172% 欺诈占比	0.174% 欺诈占比	

原始实验中的做法

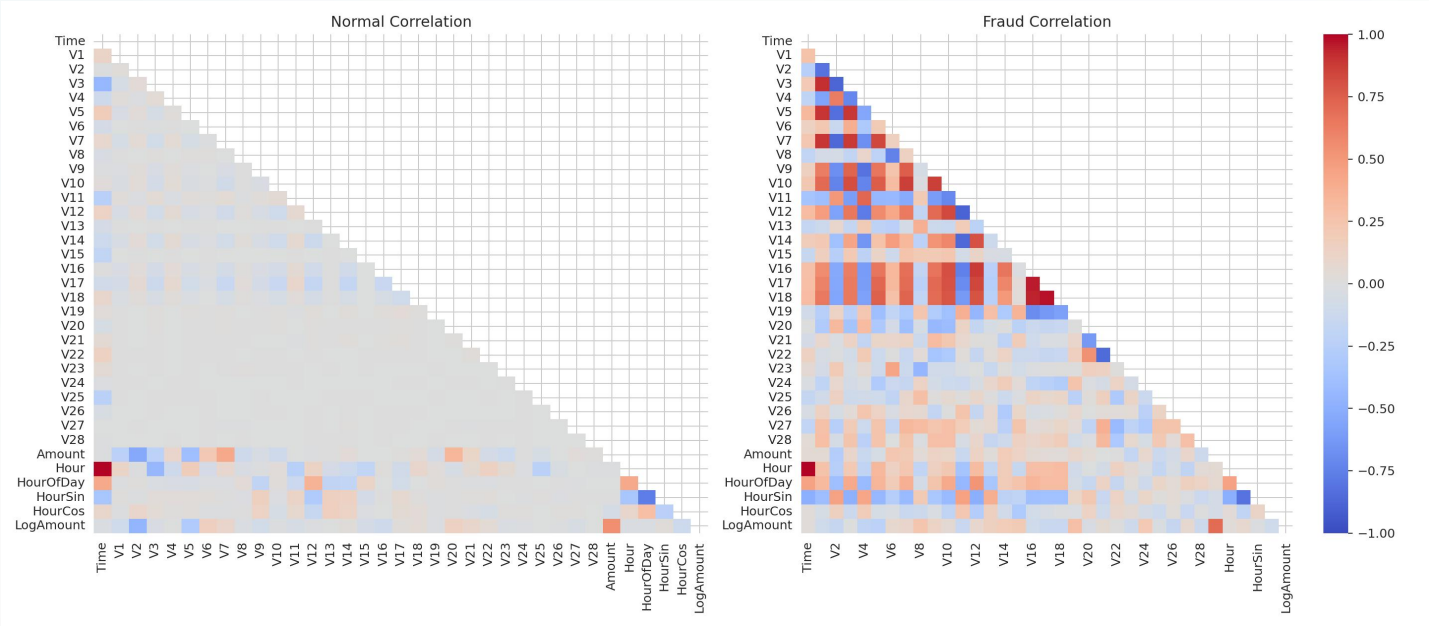
可视化阶段直接抽取 492 条欺诈样本和 492 条正常样本，形成平衡子样本，便于观察类别边界与降维分布。

改进后的做法

建模阶段保持原始不平衡分布，先做 train / validation / test 的分层切分，再只在训练集内部做样本平衡，避免信息泄露。

可视化可以用平衡子样本，测试不能用平衡子样本。只有保留真实类别比例，测试结果才有风控意义。

EDA发现



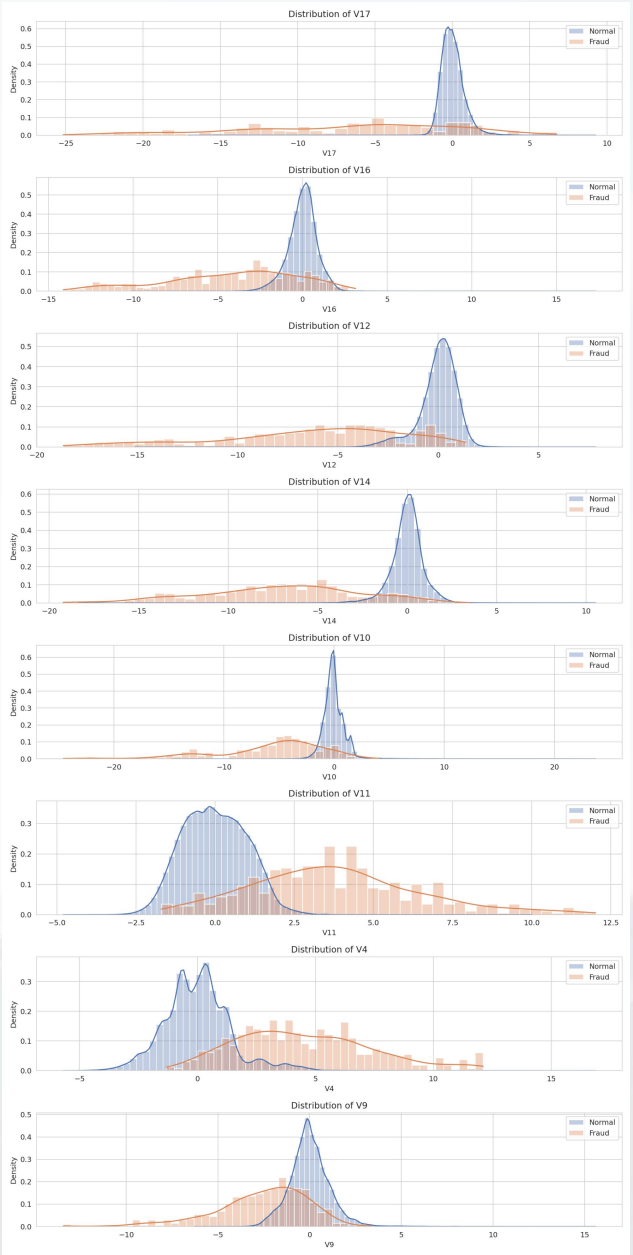
正常样本与欺诈样本的相关性热力图

相关性观察

欺诈样本内部的特征联动结构与正常样本不同。相关性热力图的意义，不在于给出最终规则，而在于说明两类样本并非完全同分布。

分布观察

部分特征在正常样本和欺诈样本上的分布位置、峰值和尾部存在明显差异，这些变量更可能成为有效判别特征。



部分关键特征在两类样本中的分布差异

5

上海大学
Shanghai University

特征工程设计

时间特征

原始实验将 Time 直接改写为 Hour(0~47)。改进版进一步提取 HourSin / HourCos，把“时间点”改成“周期位置”。

金额特征

原始实验对金额直接缩放。改进版先做 log1p，再参与缩放和训练，更适合长尾分布。

特征保留

原始实验删除了部分匿名变量。改进版保留 V1–V28，并在此基础上加入新构造的时间与金额特征。

设计原则

1. 保留有效信息；
2. 降低分布偏态；
3. 避免训练阶段的数据泄露；
4. 让不同模型能在同一套输入上公平比较。

上海大学
Shanghai University

模型选择

逻辑回归

作为基线模型，便于观察线性可分条件下的识别能力，并为阈值分析提供参照。

输入：31 维改进特征

搜索参数：C、penalty

目标：建立基线，并观察阈值变化对 Recall 与 Precision 的影响。

随机森林

利用树模型的非线性划分能力，观察集成树对少数类欺诈样本的识别效果。

搜索参数：n_estimators、max_depth、min_samples_split、min_samples_leaf、max_features

目标：提升非线性识别能力，同时保留一定解释性。

XGBoost

在树模型基础上进一步提升拟合与排序能力，应作为综合表现最强的候选模型。

搜索参数：max_depth、learning_rate、subsample、colsample_bytree、reg_lambda

目标：兼顾排序能力、泛化能力与少数类识别效果。

为什么不是只用一个模型

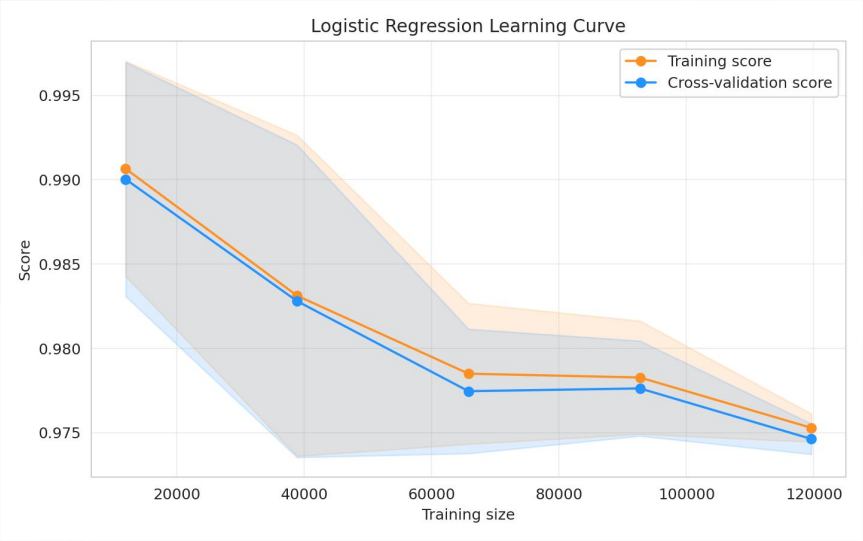
比较不同类型模型在不平衡分类中的表现差异。线性模型、Bagging 树模型和 Boosting 树模型能形成完整对照。

评估口径

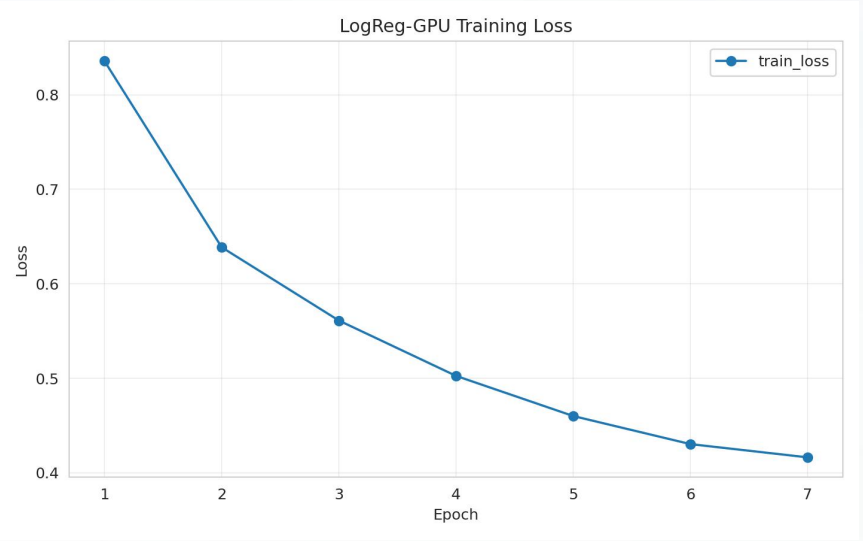
统一关注 Recall、Precision、F1、PR-AUC、ROC-AUC 与阈值下的混淆矩阵。Accuracy 只作为参考，不作为主要结论依据。

上海大学
Shanghai University

训练过程



CPU 版逻辑回归学习曲线



GPU 版逻辑回归训练损失曲线

训练顺序

先完成分层切分，再在训练集内部做平衡处理；验证集用于参数搜索和阈值选择；测试集只做最终一次评估。

阈值选择

本实验没有固定使用 0.5 阈值，而是根据验证集表现选择更合适的阈值，再用于测试集评估。

规范性

与原始实验相比，改进后的训练流程更强调验证集、测试集和训练集的职责分离。

上海大学
Shanghai University

训练结果

原始实验结果

模型训练（baseline）

```
# 模型训练
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression() # 构建逻辑回归分类器
lr.fit(X_train, y_train)

# 测试集预测
y_pred = lr.predict(X_test)

# 模型评估
from sklearn.metrics import confusion_matrix,classification_report
print('<--------Confusion Matrix-------->\n',confusion_matrix(y_test,y_pred))
print('<--------Classification Report-------->\n',classification_report(y_test,y_pred))
```

```
<--------Confusion Matrix-------->
 [[84022  1273]
 [ 5622 79672]]
<--------Classification Report-------->
              precision    recall  f1-score   support

           0       0.94      0.99      0.96     85295
           1       0.98      0.93      0.96     85294

    accuracy                           0.96    170589
   macro avg       0.96      0.96      0.96    170589
weighted avg       0.96      0.96      0.96    170589
```

```
# 构建参数组合
param_grid = {'C': [0.1, 1, 10,100],# 一般经验10倍增加
                           'penalty': [ 'l1', 'l2']}

clf = GridSearchCV(LogisticRegression(),  param_grid, cv=5)
clf.fit(X_train, y_train)
```

GridSearchCV
estimator: LogisticRegression
LogisticRegression

```
GridSearchCV(cv=5, estimator=LogisticRegression(),
             param_grid={'C': [0.1, 1, 10, 100], 'penalty': ['l1', 'l2']})
```

GridSearchCV
estimator: LogisticRegression
LogisticRegression

CPU 改进版测试结果

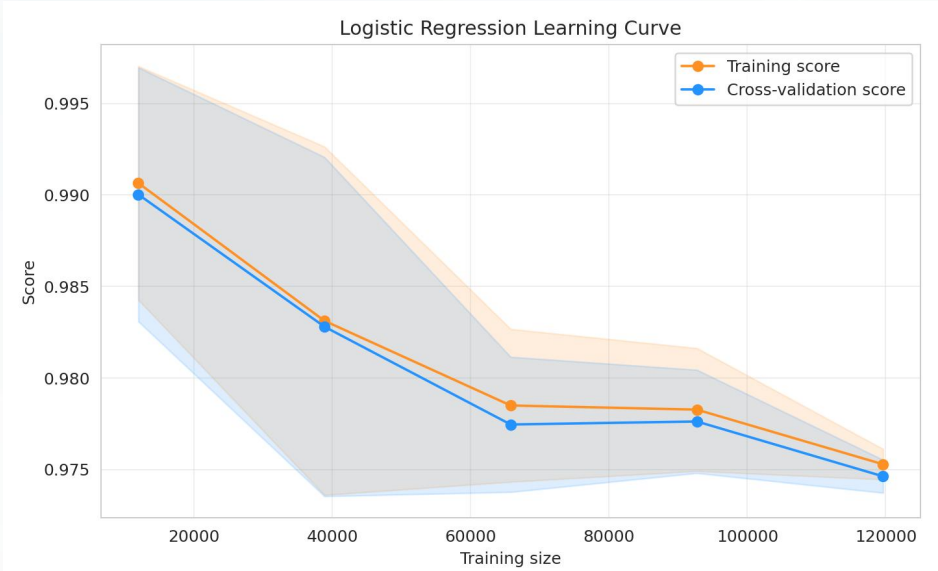
项目	原始实验	CPU 改进版
训练/测试划分	SMOTE 后再切分，训练集 398041，测试集 170589	先切分，测试集保持原始不平衡分布
基线模型	Logistic Regression	Logistic Regression / RandomForest / XGBoost
最优参数	C=100，penalty=l2	LR: C=0.1,l1；RF: 500 trees；XGB: depth=5, lr=0.03
原始实验基线结果	precision=0.98，recall=0.93，f1=0.96	LogReg: recall=0.889，F1=0.435
综合最优	未做统一测试集对比	RandomForest F1=0.890，XGBoost PR-AUC=0.877

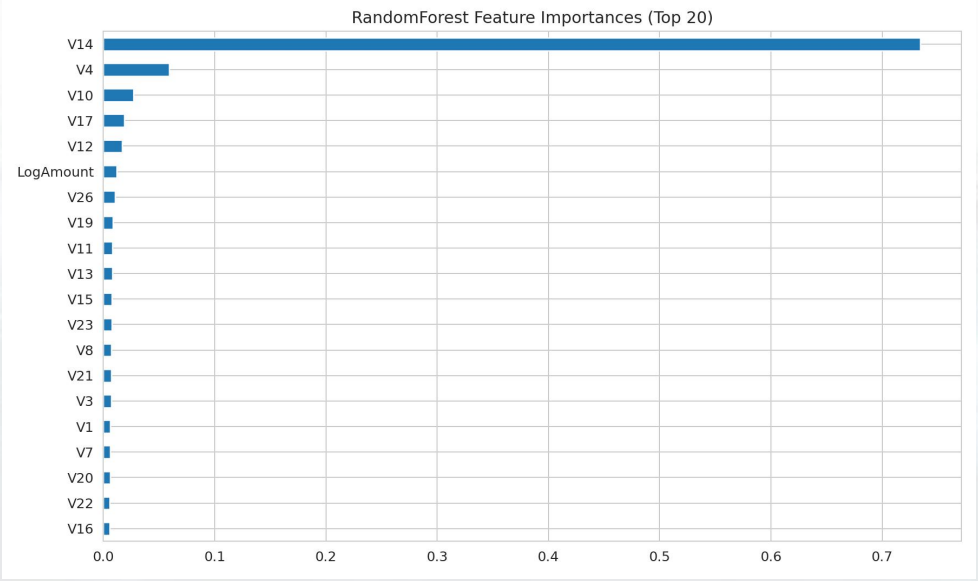
原始实验给出了基线模型与参数搜索结果；CPU 改进版在统一测试集上补充了多模型比较，便于判断改进是否真正有效。

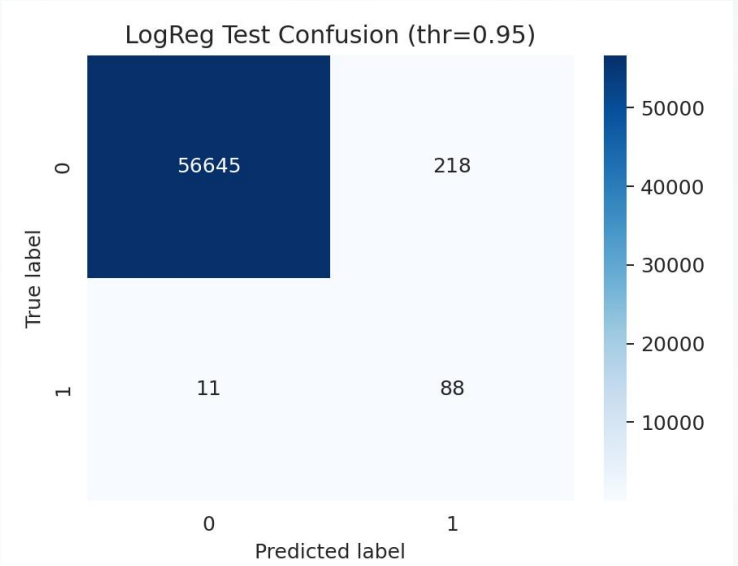
原实验使用了样本平衡处理，评估环境更理想
平衡后的数据更容易提升 F1 和 Recall
本次改进版采用更接近真实场景的测试分布，结果更严格也更可信

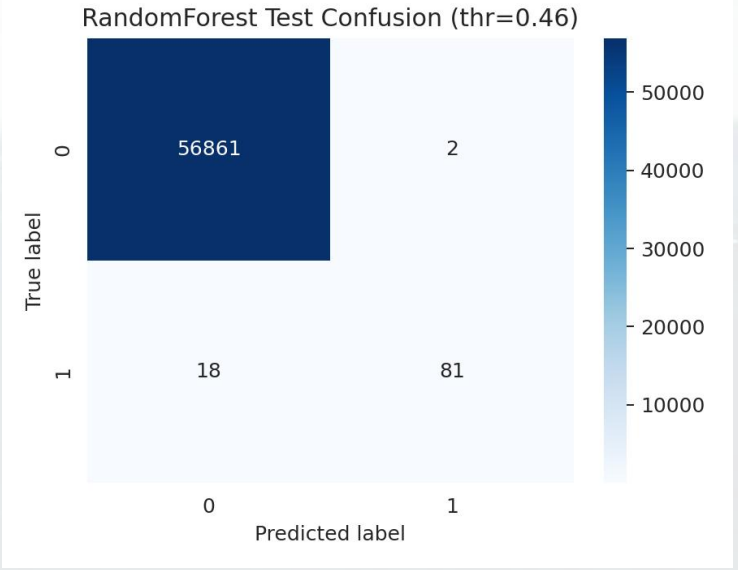
上海大学
Shanghai University

训练结果



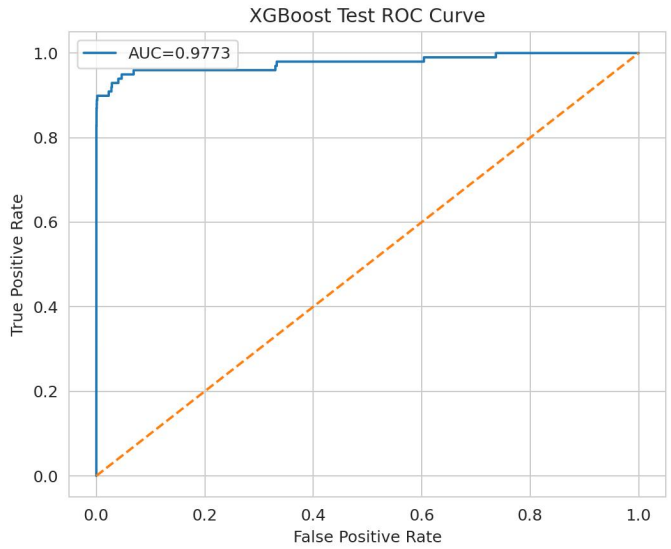


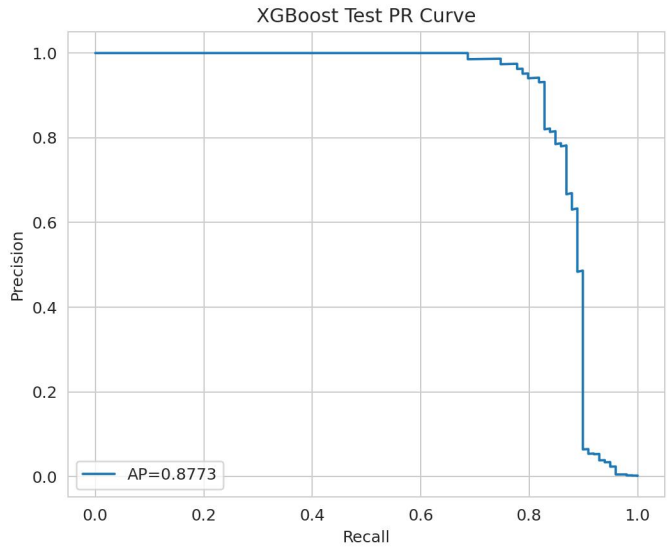




上海大学
Shanghai University

训练结果





阈值
0.19
验证集选择

ROC-AUC
0.977
排序能力

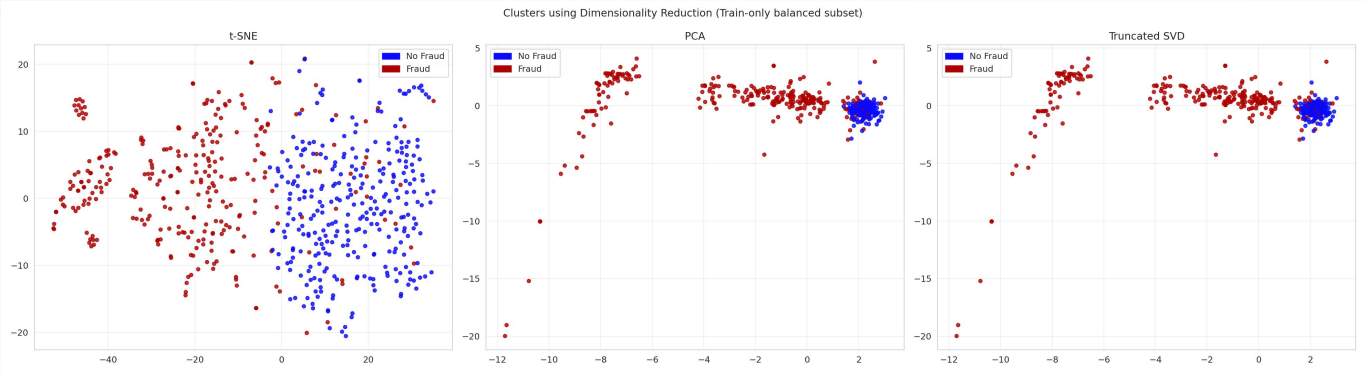
PR-AUC
0.877
少数类优先

F1
0.863
综合表现

- XGBoost 的 ROC-AUC 与 PR-AUC 均为三种模型中最高。
- 在不平衡测试集上，它比逻辑回归更稳，比随机森林更擅长概率排序。

上海大学
Shanghai University

结果分析



降维可视化：平衡子样本在低维空间中已出现可分趋势

模型差异

逻辑回归适合做基线，但对复杂边界的刻画有限；树模型能捕捉非线性关系，因此整体表现更好。

为何逻辑回归 Recall 更高

逻辑回归在阈值调节后更容易追求召回率，但通常会牺牲 Precision。它适合做高召回基线，不一定适合作为最终模型。

为何随机森林 F1 更高

随机森林对局部非线性结构和变量交互的刻画更稳，在当前数据集上实现了较好的误报 / 漏报平衡。

为何 XGBoost 曲线更好

Boosting 方式提升了样本排序能力，因此 PR-AUC、ROC-AUC 更高，适合用作综合表现最强的模型。

上海大学
Shanghai University

GPU改进

改动位置

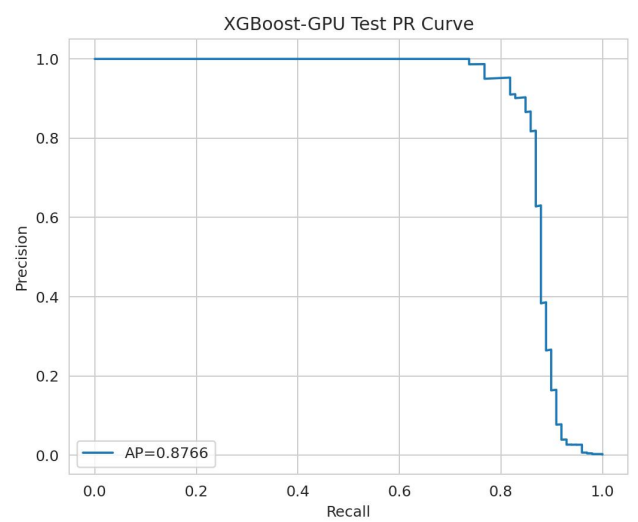
GPU 版主要放在训练阶段：逻辑回归改为 PyTorch 训练，树模型使用支持 CUDA 的实现，其他数据理解与清洗环节保持一致。

主要效果

在不改变实验主线的前提下，训练时间由原先约 1 小时缩短到约 3 分钟，便于快速复现实验和重复调参。

结果变化

GPU 适配的目的不是“换模型”，而是“换训练方式”。因此重点比较的是加速效果，同时确认测试指标没有出现明显退化。



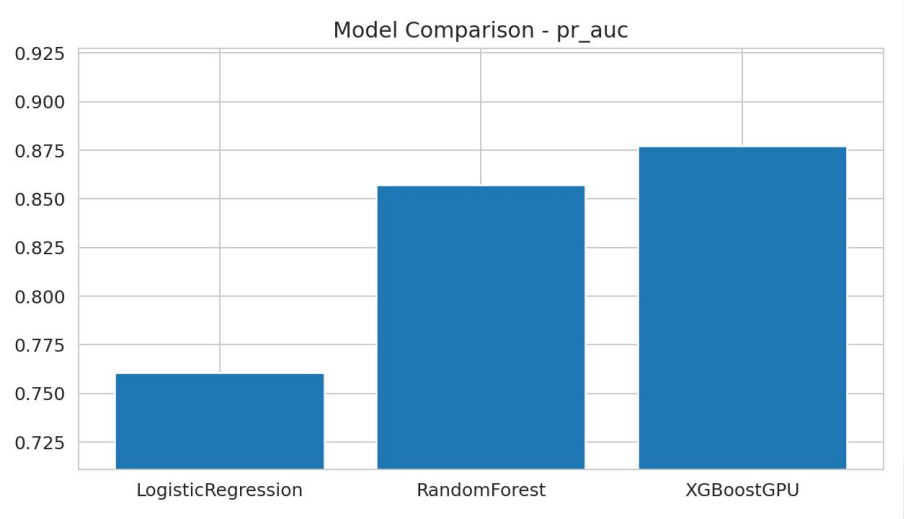
GPU 版 XGBoost Precision-Recall 曲线

- GPU 改进只占整个实验的一部分，不替代前面的数据分析、特征工程和模型评估。
- 实验主体仍然是建模逻辑；GPU 的作用是提高训练效率和复现效率。

上海大学
Shanghai University

训练结果

逻辑回归 F1	随机森林 F1	XGBoost F1	GPU XGBoost F1
0.435	**0.890**	**0.863**	**0.880**
Recall 0.889	Precision 0.976	PR-AUC 0.877	PR-AUC 0.877



结果概括

随机森林在测试集上取得最高 F1；XGBoost 在 ROC-AUC 与 PR-AUC 上表现更强；GPU 版 XGBoost 维持了接近的识别效果，并显著缩短训练时间。

展望

更丰富的特征

继续引入更精细的时间窗口统计、金额行为统计或用户级聚合特征，提升模型对异常模式的刻画能力。

更充分的模型比较

增加 LightGBM、CatBoost 或神经网络表格模型，与现有三类模型进行更系统的对比。

更接近业务场景

进一步考虑代价敏感学习、阈值动态调整和审核成本约束，使实验结果更贴近风控部署需求。

shanghai university

上海大学
Shanghai University

谢谢！

上善若水 海纳百川
大道明德 学用济世