实 验 报 告

实验名称 网贷平台的风控建模与调优

课程名称 网络数据风控技术

专业班级：2022 级计算机科学与技术专业

学生姓名：张宇　　学　　号：2251745

指导教师：叶晨老师　　成　　绩：75.09

实验日期：2024 年 11 月 8 日

同济大学

一、 实验背景

信用贷（原蚂蚁借呗）是支付宝平台提供的小额信用贷款服务。互联网小额信用贷款平台（简称网贷平台）根据风险控制和准入标准对支付宝实名用户进行筛选，为筛选出的优质客户提供不同的借款额度。网贷平台的快速发展得益于互联网平台建立了自建的场景流量信用评分体系，形成了获客和贷款的闭环体验。

本实验需要通过约400个数据维度评估用户当前的信用状态，结合脱敏处理过的样本数据，对贷款风险进行建模，预测用户6个月内的逾期率，为平台提供关键的决策依据。

二、 问题分析与数据准备（10%）

1. 问题分析

在网贷平台的风控建模中，主要目标是通过分析用户的行为、信用特征等约400个数据维度，预测用户未来6个月内的逾期风险。这一预测对于平台的风险控制和贷前决策具有重要意义。具体而言，模型的输出将帮助平台在决策时区分高风险和低风险用户，进而调整贷款额度或拒绝高风险用户的贷款请求。

本实验涉及的问题具有以下特点：

高维数据：数据集包含400多个特征，这些特征反映了用户的不同属性，如账户信息、交易历史、行为数据等。因此，需要进行特征筛选以提高建模效率。

类别不平衡：通常在信用风险预测中，违约用户往往占比少，因此类别不平衡可能会对模型的预测结果产生影响，需考虑平衡策

略。

2. 数据检查

数据准备的第一步是对数据的整体情况进行检查，主要包括数据的维度、数据类型、缺失值和异常值等方面。

（1）维度和数据类型检查：首先检查数据的列数和行数，并查看每个特征的数据类型，以确定哪些是数值型特征、哪些是类别型特征。此步骤有助于后续的特征工程（如数值特征的标准化、类别特征的编码等）。

（2）缺失值检查：由于数据可能存在缺失值，需要统计各列的缺失值数量和比例。针对缺失值较多的列，可能需要进行适当的处理，如删除、填充或插值。

（3）重复值检查：检查数据中是否存在重复行或重复用户数据，以免影响建模效果。通常，重复数据需要被删除。

三、 数据清洗与预处理（20%）

在信用风险预测模型中，数据的质量至关重要。为确保数据的完整性和一致性，数据清洗与预处理包括以下关键步骤：缺失值处理、异常值剔除、标准化和归一化等操作。

1. 缺失值处理

缺失值处理是数据预处理的首要步骤。在本项目中，数据字段多达400个，为提高数据的质量和模型的预测效果，需要对各字段的缺失值情况进行全面检查和多维度处理。

（1）列缺失值比例：首先统计每一列的缺失比例，并设置缺失阈

值。若某列的缺失比例超过 50%，则认为该特征信息不足，可直接删除。

（2）行缺失值比例：统计每一行的缺失值数量，并检查训练集和测试集缺失分布的一致性。若某些样本的缺失值比例较高，可以考虑将其作为离群点处理。

（3）特征重要性筛选：利用 XGBoost 模型对所有特征进行训练，提取特征重要性列表。若缺失值出现在重要性较高的特征中，优先填补该缺失值。具体填充策略包括：数值型特征：使用均值或中位数填补和类别型特征：使用众数填补。

```
import pandas as pd
from sklearn.impute import SimpleImputer
import xgboost as xgb

# 加载数据
data = pd.read_csv('Master_Training_Set.csv', encoding='gbk')

# 删除缺失率超过 50%的列
data = data.loc[:, data.isnull().mean() < 0.5]

# 数值型特征填充
num_imputer = SimpleImputer(strategy="mean")
data_num = data.select_dtypes(include=['float64', 'int64'])
data[data_num.columns] = num_imputer.fit_transform(data_num)

# 类别型特征填充
cat_imputer = SimpleImputer(strategy="most_frequent")
data_cat = data.select_dtypes(include=['object'])
data[data_cat.columns] = cat_imputer.fit_transform(data_cat)
```

2. 异常值处理

异常值指偏离正常范围的值，可能会对模型的稳定性产生负面影响。处理异常值的具体步骤如下：

（1）三倍标准差法：对于数值型特征，通过三倍标准差法判断异常值，若某特征的值超出均值三倍标准差的范围，则视为异常。可视化

检查特征分布后，判断是否删除或修正这些异常值。

（2）基于业务规则的异常检测：若有明确的业务规则（例如贷款金额不可为负），可基于规则过滤异常值。

3. 标准化与归一化处理

标准化和归一化是数据预处理中重要的步骤，能够有效提高模型的收敛速度和精度。考虑到特征值的量纲不同，通过标准化或归一化将所有特征缩放至相同尺度，有助于提升模型的表现。

（1）标准化：将数值特征转化为均值为 0、方差为 1 的标准正态分布，适合正态分布的数据。

（2）归一化：将特征值缩放至 0 到 1 之间，适合特征取值范围差异较大的数据。

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler

# 标准化处理
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data.select_dtypes(include=['float64',
'int64']))

# 归一化处理（若适用）
min_max_scaler = MinMaxScaler()
data_normalized =
min_max_scaler.fit_transform(data.select_dtypes(include=['float64',
'int64']))
```

4. 文本处理

在数据中，部分类别型变量为文本信息，且存在大小写不一致、空格等问题，需统一格式进行归一化。

（1）大小写处理：将文本型特征统一转化为小写，避免大小写不一致对模型造成影响。

（2）去除空格：移除特征值中的空格符，统一格式。

（3）城市名处理：将地理信息字段中的“市”字符去除，统一城市名的格式。

```
import re

# 将 UserupdateInfo1 字段统一转为小写
data['UserupdateInfo1'] = data['UserupdateInfo1'].str.lower()

# 去除空格
data['UserInfo_9'] = data['UserInfo_9'].str.strip()

# 城市名统一
data['UserInfo_8'] = data['UserInfo_8'].apply(lambda x: re.sub(r'市$', '', x))
```

四、 特征工程（30%）

特征工程在风控建模中至关重要，通过构建、转换、选择和优化特征，可以显著提升模型的预测能力。在本项目中，特征工程包含地理位置信息、成交时间、类别特征编码、组合特征构建、特征选择和类别不平衡处理等。

1. 地理位置信息处理

由于地理位置信息可能与违约风险相关，因此可以从省份和城市两个维度进行特征构造：

（1）省份级特征：统计每个省份的违约率，并将违约率超过一定阈值的省份标记为高风险省份，生成对应的二值特征（如“是否来自高风险省份”）。

（2）城市等级合并：根据经济发展水平对城市进行分级，例如将一线城市（如北京、上海、广州等）标记为 1，二线城市标记为 2，三线及以下城市标记为 3，避免直接独热编码带来的维度过高问题。

（3）经纬度信息：将地理位置编码为数值型特征，通过引入经纬

度信息反映地域差异，例如北京市的经纬度可表示为纬度 39.92，经度 116.46。

```
# 省份级特征构造
data['is_high_risk_province'] = data['UserInfo_7'].apply(lambda x: 1 if x in ['四川', '湖南', '湖北'] else 0)

# 城市级特征构造
city_map = {'北京': 1, '上海': 1, '广州': 1, '深圳': 1}  # 一线城市示例
data['city_level'] = data['UserInfo_8'].map(city_map).fillna(3)

# 经纬度特征构造
city_coords = {'北京': (39.92, 116.46), '上海': (31.22, 121.48)}
data['latitude'] = data['UserInfo_8'].map(lambda x: city_coords.get(x, (0, 0))[0])
data['longitude'] = data['UserInfo_8'].map(lambda x: city_coords.get(x, (0, 0))[1])
```

2. 特征选择

为避免特征过多导致的模型复杂度增加和过拟合问题，使用 XGBoost 等模型进行特征选择，保留重要特征。

（1）基于模型的特征重要性排序：使用 XGBoost 模型训练完成后，提取特征的重要性，将 Top N 个重要特征用于模型训练。

（2）过滤方法：根据方差筛选，剔除方差低的特征，因为这些特征在数据中区分度低。

```
from xgboost import XGBClassifier

xgb = XGBClassifier()
xgb.fit(data.drop(columns=['target']), data['target'])

# 提取重要性前 50 的特征
feature_importances = pd.Series(xgb.feature_importances_, index=data.drop(columns=['target']).columns)
top_features = feature_importances.nlargest(50).index
data = data[top_features]
```

3. 类别不平衡处理

由于正负样本数量不平衡，直接训练可能导致模型偏向多数类。因

此需进行类别不平衡处理，常用的方法包括：

（1）过采样：对少数类样本进行过采样，例如使用 SMOTE 算法。

（2）代价敏感学习：在模型训练中增加类别权重，使模型更关注少数类样本。

```
from imblearn.over_sampling import SMOTE

# 使用 SMOTE 进行过采样
X, y = data.drop(columns=['target']), data['target']
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)
```

五、建模与优化（30%）

建模与优化是信用风险预测模型开发的核心步骤。为提升模型的预测效果和稳定性，本实验选择了多种机器学习模型进行建模，并采用超参数调优和模型融合策略进一步优化模型性能。具体过程包括逻辑回归、XGBoost、LightGBM 模型的构建与调优，以及通过模型融合提升最终的预测效果。

1. 模型选择

为确保模型能够充分挖掘数据特征，选用了以下几种不同的模型

（1）逻辑回归（Logistic Regression）：适用于二分类问题，作为基准模型，便于与其他复杂模型进行对比。

（2）XGBoost：基于梯度提升的决策树模型，具有较强的非线性特征学习能力和抗过拟合能力。

（3）LightGBM：与 XGBoost 类似，是一种高效的梯度提升算法，在处理大规模数据和高维数据时具有更好的性能。

```
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
```

```
from sklearn.metrics import roc_auc_score, accuracy_score

# 定义模型
lr = LogisticRegression()
xgb = XGBClassifier()
lgb = LGBMClassifier()

# 训练逻辑回归模型
lr.fit(X_train, y_train)
lr_preds = lr.predict(X_test)
lr_auc = roc_auc_score(y_test, lr.predict_proba(X_test)[:, 1])
print(f"Logistic Regression AUC: {lr_auc}")

# 训练 XGBoost 模型
xgb.fit(X_train, y_train)
xgb_preds = xgb.predict(X_test)
xgb_auc = roc_auc_score(y_test, xgb.predict_proba(X_test)[:, 1])
print(f"XGBoost AUC: {xgb_auc}")

# 训练 LightGBM 模型
lgb.fit(X_train, y_train)
lgb_preds = lgb.predict(X_test)
lgb_auc = roc_auc_score(y_test, lgb.predict_proba(X_test)[:, 1])
print(f"LightGBM AUC: {lgb_auc}")
```

2. 模型融合

为了进一步提升模型的泛化能力和稳定性，我们采用模型融合策略，将多个模型的预测结果进行组合，减少单一模型可能出现的偏差。模型融合方法如下：

（1）软投票：结合各模型的预测概率进行加权平均，使得融合模型的预测结果更为稳定。

（2）硬投票：基于多数模型的预测结果，采用少数服从多数的原则。

```
from sklearn.ensemble import VotingClassifier

# 定义 Voting Classifier
voting_model = VotingClassifier(estimators=[
    ('lr', lr), ('xgb', xgb_grid.best_estimator_), ('lgb',
lgb_grid.best_estimator_)
```

```
], voting='soft')

# 训练融合模型
voting_model.fit(X_train, y_train)
voting_preds = voting_model.predict(X_test)
voting_auc = roc_auc_score(y_test, voting_model.predict_proba(X_test)[:, 1])
print(f"Voting Classifier AUC: {voting_auc}")
```

3. 模型评估

在模型训练和优化完成后，使用以下评估指标对模型的预测效果进行综合评估：

（1）AUC-ROC（曲线下面积）：反映模型在不同分类阈值下的表现，是不平衡数据集中常用的评价指标。

（2）F1-Score：综合考虑精确率和召回率，适合类别不平衡问题的评估。

（3）准确率（Accuracy）：衡量模型总体的预测正确率，但在类别不平衡问题中通常不作为主要指标。

```
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix

# 计算评估指标
accuracy = accuracy_score(y_test, voting_preds)
f1 = f1_score(y_test, voting_preds)
conf_matrix = confusion_matrix(y_test, voting_preds)

print(f"Accuracy: {accuracy}")
print(f"F1-Score: {f1}")
print("Confusion Matrix:")
print(conf_matrix)
```

六、 结论和心得体会（8%）

（实验结果，并对实验现象及处理方法、实验中存在的问题等进行分析和讨论，对实验的进一步想法或改进意见。）

1. 结论

通过本次实验，我们成功构建了一个用于信用风险预测的模型，能够较好地识别出高风险的贷款用户，为网贷平台的风险控制和贷前决策提供了重要支持。实验中，我们使用逻辑回归、XGBoost 和 LightGBM 等多种模型进行建模，最终通过模型融合提升了预测效果。以下是实验的主要结论：

（1）数据清洗和特征工程至关重要：在实验中，缺失值处理、异常值剔除、归一化以及地理位置信息、成交时间等特征的构造，对模型的预测性能有显著影响。合理的数据预处理和特征工程为模型提供了高质量的输入数据。

（2）模型融合提升了模型的稳定性和预测精度：通过逻辑回归、XGBoost、LightGBM 的软投票融合方法，有效降低了单一模型的偏差，使得模型能够更稳健地应对数据集中的噪声和不平衡问题。

2. 心得体会

在本次实验中，我深刻体会到了数据清洗、特征工程、模型选择和优化的重要性，以下是一些具体的心得体会：

（1）数据预处理是模型效果的基础：实验过程中，数据的质量直接影响了模型的表现。在数据清洗阶段，通过合理的缺失值处理和异常值剔除，避免了不完整或噪声数据对模型的干扰。这让我意识到，投入时间进行数据清洗是非常值得的。

（2）特征工程提升了模型对数据的理解能力：特征工程是本实验的关键步骤，通过构建新的组合特征（如信用评分与收入比值）和地理位置信息的分级处理，模型能够更全面地捕捉数据中的隐含信息。好的特征工程可以让模型更好地理解数据的结构和规律，从而提升模型的预

测效果。

（3）模型选择和调优需要根据实际数据特点：实验过程中，不同模型在初始状态下的表现有所差异，但通过超参数调优后，XGBoost 和 LightGBM 这类基于树的模型表现出更强的非线性学习能力,适合用于复杂的高维数据。合理的模型选择和调优策略能够显著提高模型的效果和鲁棒性。

（4）模型融合有效提升模型的稳健性：在类别不平衡的数据集中，单一模型可能会存在偏差，通过 Voting 融合策略能够将多个模型的优点结合起来，从而增强模型对高风险用户的识别能力。模型融合在一定程度上提高了模型的泛化能力。

3. 改进与展望

虽然本次实验的模型在性能上取得了良好效果，但仍有改进空间：

（1）尝试更复杂的深度学习模型：未来可以尝试使用神经网络或深度学习模型（如 LSTM、GRU 等），特别是对具有时间序列特征的数据，可以进一步提升模型的表现。

（2）引入更多外部数据和特征：例如，增加用户的社交行为数据、第三方信用数据等，能够使模型更加全面，提升预测的准确性。

（3）自动化特征工程和调参：可以尝试使用 AutoML 工具，进行自动化特征选择和超参数调优，以节省手动调试的时间和精力。

总的来说，本次实验让我在数据预处理、特征工程、模型选择与优化等方面积累了宝贵的经验，为今后进行更复杂的机器学习项目打下了坚实的基础。希望在未来的实践中，能够继续优化和提升，为网贷平台的风险控制提供更加先进和高效的技术支持。

七、 参考文献：（2%）

[1]于晓虹，楼文高. 基于随机森林的P2P网贷信用风险评价、预警与实证研究[J]. 金融理论与实践，2016(2):6.

[2]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(6)

[3]薛琦，罗鄂湘. 基于机器学习的银行个人信用风险评估研究[J]. 建模与仿真，2023，12(4)

[4]周炜堉，龚平. 基于主成分分析的信用评分模型研究[J]. 统计学与应用，2024，13(3)

[5]梁龙跃，王浩竹. 基于图卷积神经网络的个人信用风险预测[J]. Journal of Computer Engineering & Applications，2023，59(17)

[6]彭润泽. 基于 Stacking 集成学习算法的个人信用评估模型[J]. Statistics and Application，2017，6