Statistics and Application 统计学与应用, 2019, 8(5), 823-834
Published Online October 2019 in Hans. http://www.hanspub.org/journal/sa
https://doi.org/10.12677/sa.2019.85093

Hans汉斯

Research on Internet Credit Risk Prediction Based on Model Fusion

Hongyan Fei, Hao Huang

School of International Technology & Management, University of International Business and Economics, Beijing
Email: 18340877426@163.com

Received: Oct. 8th, 2019; accepted: Oct. 22nd, 2019; published: Oct. 29th, 2019

Abstract

The prediction of the credit risk of Internet credit is a key factor for the sustainable development of Internet finance. It can accurately estimate the credit risk of borrowers before lending, effectively reducing the possible risk loss of enterprises. With the development of machine learning, the algorithm model of machine learning has been applied more and more in the credit risk of Internet credit. In order to explore the effect of integrating tree model and linear model in the prediction of credit risk of Internet credit, this paper adopts Stacking model fusion method to design the credit risk prediction model, in which the first layer model is random forest, XGBoost and LightGBM and the second layer model is logistic regression, and conducts experiments on the real data of Clap to Borrow. Compared with the performance of the single model on AUC, accuracy and time consuming, the results show that the fused model, although takes longer time, but performs better in terms of AUC and accuracy, which provides a new idea for the construction of financial credit risk prediction model.

Keywords

Logistic Regression, the Credit risk, Random Forests, XGBoost, LightGBM

基于模型融合的互联网信贷信用风险预测研究

费鸿雁，黄　浩

对外经济贸易大学信息学院，北京
Email: 18340877426@163.com

收稿日期：2019年10月8日；录用日期：2019年10月22日；发布日期：2019年10月29日

摘　要

互联网信贷信用风险的预测是互联网金融可持续发展的关键因素，在放贷前准确预估借款人的信用风险，能有效较低企业可能的风险损失。随着机器学习的发展，机器学习的算法模型在互联网信贷信用风险方

文章引用: 费鸿雁, 黄浩. 基于模型融合的互联网信贷信用风险预测研究[J]. 统计学与应用, 2019, 8(5): 823-834.
DOI: 10.12677/sa.2019.85093

面的应用也越来越多。为了探究树模型和线性模型融合在互联网信贷信用风险预测的效果，本文采用Stacking模型融合方法设计了信用风险预测模型，其中第一层模型为随机森林、XGBoost、LightGBM，第二层模型为逻辑回归。并且在拍拍贷的真实数据上进行实验，对比了融合后的模型和单模型在AUC、准确率和耗时上的表现，结果表明融合后的模型虽然耗时长一些，但是在AUC和准确率方面都比单模型的效果要好，为互联网金融信贷风险预测模型的构建提供了一个新的思路。

关键词

逻辑回归，信用风险，随机森林，XGBoost模型，LightGBM模型

1. 引言

网络借贷是一种新型的互联网金融模式，而在如何快速、准确的综合评估借款人的信用风险则成为了互联网金融能否健康可持续的发展的关键因素，这在工业界备受重视，如今在预测信贷信用风险领域有很多学者进行研究，也取得了不错的进展，但是还有待进一步的深入研究[1]。

对用户的信用风险的评估本质是上是一个二分类问题，对此主要有三种解决方法：统计分析、定性分析、人工智能方法[2]。近年来，随着机器学习算法的发展，机器学习在金融信用风险评估领域的应用也越来越广。Malekipirbazari 等人提出了一种基于随机森林的借款人信用评估方法，利用借贷平台 Lending Club 上的真实用户数据进行预测，取得不错的效果[3]。李昕、戴一成基于 BP 神经网络构建了信用风险评估模型，实验结果表明 BP 神经网络具有较高的预测准确率，适合平台和投资者甄选优质借款人[4]。这些机器学习算法模型的单模型应用已经有了广泛的研究，而模型之间的融合还有进一步探讨的空间。2016 年 FaceBook 提出了 GBDT 和逻辑回归相融合模型对点击率进行预测，融合后的模型比单模型效果的提升了 3% [5]。本文将树模型和线性模型融合的方法引入信用风险评估中，基于拍拍贷的数据建立分类预测模型，为企业和投资者提供决策依据，结果表明树模型和线性模型的融合效果比单模型的效果要好。

2. 模型介绍

2.1. 逻辑回归模型

逻辑回归属于广义线性回归分析模型，在二分类问题中有广泛的应用，通过 Logistic 函数将目标值 Y 的取值归一化到 0 和 1 之间，使用梯度下降法或拟牛顿法不断的迭代，直到损失函数收敛为止。对于二分类问题，设：

$$P(Y=1|x)=\pi(x),\ P(Y=0|x)=1-\pi(x) \tag{1}$$

则似然函数为：

$$\prod_{i=1}^{N}\left[\pi(x_i)\right]^{y_i}\left[1-\pi(x_i)\right]^{(1-y_i)} \tag{2}$$

对数似然函数为：

$$\begin{aligned} L(w) &= \sum_{i=1}^{N}\left[y_i \log \pi(x_i) + (1-y_i)\log\left(1-\pi(x_i)\right)\right] \\ &= \sum_{i=1}^{N}\left[y_i \log \frac{\pi(x_i)}{1-\pi(x_i)} + \log\left(1-\pi(x_i)\right)\right] \\ &= \sum_{i=1}^{N}\left[y_i (w*x_i) - \log\left(1+\exp(w*x_i)\right)\right] \end{aligned} \tag{3}$$

其中 w 是权重向量， $w*x$ 是 w 和 x 的内积。对 $L(w)$求极大值，得到 w 的估计值[6]。逻辑回归通常采用梯度下降法和拟牛顿法来求解对数似然函数的最优化问题。假设 w 的极大似然估计值为 $\bar{w}$，则逻辑回归模型为：

$$P(Y=1 \mid x) = \frac{e^{\bar{w}x}}{1+\left(e^{\bar{w}x}\right)} \tag{4}$$

$$P(Y=0 \mid x) = \frac{1}{1+\left(e^{\bar{w}x}\right)} \tag{5}$$

2.2. 随机森林模型

随机森林是 Bagging 集成学习模型的代表，以决策树为基学习器构建 Bagging 模型，并且引入特征随机选择的思想，对模型进行训练，输出结果由多棵决策树决定[7]。随机森林的主要流程是：对每棵树而言，随机有放回的抽取 N 个样本作为该树的训练集；如果每个样本的特征数量为 M，则选择一个 m，令 $m \ll M$，每次在 M 个特征中随机抽取 m 个特征进行分裂，分裂点是这个 m 个特征中最优的那个；每棵树都尽可能深的生长，不进行剪枝；对所有决策树的结果进行加总得到输出结果，回归问题使用多数投票方法，回归问题使用取平均值的方法。

2.3. XGBoost 模型

XGBoost 模型是 Chen 等人在 2016 年提出的一种 Boosting 模型[8]，在传统 GBDT 算法的基础上，对损失函数进行二阶泰勒展开，并且加入了正则化项，平衡模型的复杂度和的目标函数的下降速度，能够有效解决过拟合问题。Boosting 的思想是将多个弱学习分类器集合起来形成一个强学习分类器，而 XGBoost 所用到的树模型是 CART 树。XGBoost 通过不断的添加树来分裂特征，每添加一棵树就是在学习一个新函数来拟合上个函数的残差。在对数据进行预测时，就是这个数据的每个特征都会落到一个叶子结点上，最后的输出值就是这些树的分数的和。XGBoost 的算法模型如公式(6)所示，x_i 为第 i 个样本的特征向量，f_k 是一个回归树，F 是回归树的集合。

$$\overline{y_i} = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F \tag{6}$$

目标函数如公式(7)所示，包含了自身的损失函数和政策化惩罚项，其中 f_i 为第 i 棵树； $\overline{y_i}^{(t)}$ 表示 t 棵树模型的组合的预测值。

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \overline{y_i}^{(t)}\right) + \sum_{i=1}^{t} \theta(f_i) \tag{7}$$

XGBoost 一次添加一棵树，整个优化流程如下：

$$\overline{y_i}^{(0)} = 0$$

$$\overline{y_i}^{(1)} = f_1(x_i) = \overline{y_i}^{(0)} + f_1(x_i)$$

$$\overline{y_i}^{(2)} = f_1(x_i) + f_2(x_i) = \overline{y_i}^{(1)} + f_2(x_i)$$

…

$$\bar{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \bar{y}_i^{(t-1)} + f_t(x_i) \tag{8}$$

将公式(8)代入公式(7)得公式(9)，其中 C 为常数：

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \bar{y}_i^{(t-1)} + f_t(x_i)\right) + \theta(f_t) + C \tag{9}$$

XGBoost 的思想是在 $f_t = 0$ 的二阶泰勒展开来求得近似值，引入正则项后，得：

$$Obj^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(X_i) \right] + \theta(f_t) \tag{10}$$

其中：$g_i = \alpha_{\bar{y}_i^{(t-1)}} l\left(y_i, \bar{y}_i^{(t-1)}\right)$，$h_i = \alpha^2_{\bar{y}_i^{(t-1)}} l\left(y_i, \bar{y}_i^{(t-1)}\right)$。

2.4. LightGBM 模型

LightGBM 是由微软亚洲研究院在 2017 年提出的一种 GBDT 框架[5]，相较于传统的 GBDT 算法，LightGBM 的优化主要包括三部分：基于 Histogram 的决策树算法、带有深度限制的 Leaf-wise 的叶子生长策略、直方图做差加速，既提升了算法的效率，又能防止过拟合。

Histogram 算法的基本思想是先把连续的浮点特征值离散化为 k 个整数，同时构造一个宽度为 k 的直方图。在遍历数据的时候，根据离散化后的值作为索引在直方图中累计统计量，当遍历一次数据后，直方图累积了需要的统计量，然后根据脂肪乳的离散值，遍历寻找最优的分割点。

在 Histogram 算法的基础上，LightGBM 进行进一步的优化，即带有深度限制的 Leaf-wise 的叶子生长策略，每次从当前所有叶子结点中，找到分裂增益最大的一个叶子，然后分裂，如此循环。LightGBM 的另一个优化是直方图做差加速，一个叶子的直方图可以由它的父节点的直方图和兄弟节点的直方图做差得到，因此直方图做差仅需遍历直方图的 k 个桶，所以 LightGBM 可以用非常小的代价得到兄弟叶子的直方图，在速度上可以提升一倍。

3. 实验过程

3.1. 实验数据和描述

本实验所采用数据来自拍拍贷“魔镜杯”风控数据大赛，数据集共有三个数据表，分别是用户行为信息表 Master、用户登录数据表 LogInfo、用户信息修改数据表 Userupdate。用户行为信息表 Master 有 5 万多条数据，每个样本包含 228 个特征；用户登录数据表 LogInfo 有 96 万条数据；用户信息修改数据表 Userupdate 有 61 万条数据，数据特征已经进行了脱敏处理。三个表的主要特征如表 1、表 2、表 3 所示：

Table 1. User behavior information table
表 1. 用户行为信息表

特征名称	含义
Idx	每一笔贷款的 unique key，可以与另外 2 个表的 idx 相匹配。
UserInfo_*	借款人特征特征
WeblogInfo_*	网络行为特征
Education_Info*	学历学籍特征
ThirdParty_Info_PeriodN_*	第三方数据时间段 N 特征
SocialNetwork_*	社交网络特征
LinstingInfo	借款成交时间
Target	违约标签(1：贷款违约；0：正常还款)

Table 2. User login data table
表 2. 用户登录数据表

特征名称	含义
Idx	每一笔贷款的 unique key
LogInfo1	操作代码
LogInfo2	操作类别

Table 3. User information update table
表 3. 用户信息修改数据表

特征名称	含义
Idx	每一笔贷款的 unique key
ListingInfo1	借款成交时间
UserupdateInfo1	修改内容
UserupdateInfo2	修改时间

3.2. 数据预处理

1) 数据清洗

数据预清洗主要包括缺失值处理、文本处理、删除常变量。

缺失值处理：用户行为信息表 Master 的行和列的缺失值有不同处理方式。列(属性)统计缺失值比率，缺失值比率为 60%左右的数值型特征用−1 填充，将“是否缺失”看做另一种类别。其他缺失值比率比较小的数值型特征用中值填充，字符型特征用 unknown 填充。行统计每个样本的属性缺失值个数，将缺失值个数按从小到大排序，以序号为横坐标，缺失值个数为纵坐标，画出图 1 散点图，有部分样本缺失值很高，对缺失值超过 120 的样本进行删除。

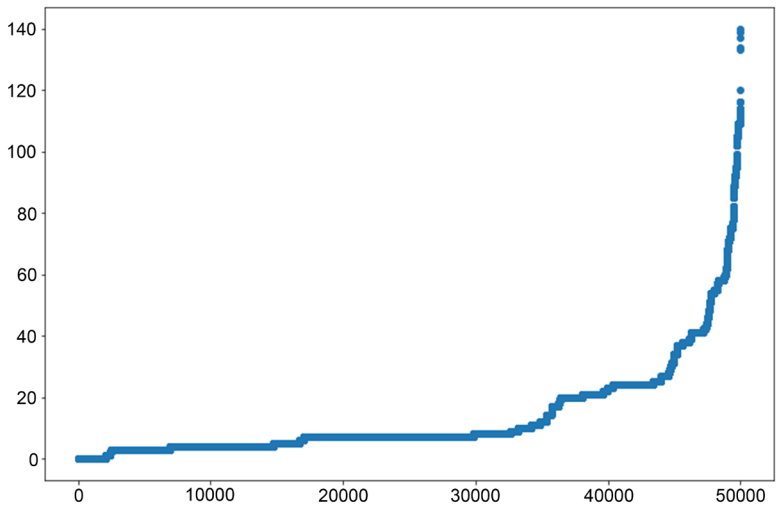

Figure 1. Number of missing value attributes for each sample
图 1. 每一条样本缺失值属性个数

文本处理：对用户行为信息表 Master 中含有空格的文本型特征进行删除空格处理，并统一取值形式(将重庆、重庆市统一为重庆)。对用户信息修改数据表 Userupdate 中的文本型数值统一大小写。

删除常变量：计算用户行为信息表 Master 中数值型特征的方差，删除部分方差较小的特征，如表 4 所示，删除方差小于 0.1 的特征。

Table 4. Sort of variance
表 4. 方差排序

特征	方差	特征	方差
WeblogInfo_49	0.000000	WeblogInfo_28	0.015079
WeblogInfo_44	0.000240	SocialNetwork_16	0.016559
WeblogInfo_41	0.000360	WeblogInfo_50	0.020766
WeblogInfo_46	0.000740	WeblogInfo_38	0.022579
WeblogInfo_55	0.001220	WeblogInfo_13	0.022961
SocialNetwork_1	0.001419	WeblogInfo_34	0.023144
WeblogInfo_43	0.001558	WeblogInfo_45	0.024843
WeblogInfo_47	0.001638	WeblogInfo_57	0.025356
WeblogInfo_52	0.002937	SocialNetwork_15	0.027182
WeblogInfo_58	0.004154	WeblogInfo_29	0.029978
WeblogInfo_40	0.004376	WeblogInfo_25	0.030240
WeblogInfo_32	0.006964	Education_Info5	0.031205
WeblogInfo_31	0.006990	WeblogInfo_56	0.032448
WeblogInfo_10	0.008021	SocialNetwork_2	0.039319
WeblogInfo_23	0.008190	WeblogInfo_11	0.048684
WeblogInfo_54	0.010551	WeblogInfo_48	0.050104
WeblogInfo_35	0.010850	Education_Info1	0.058566
WeblogInfo_53	0.011841	SocialNetwork_14	0.058664
WeblogInfo_37	0.012094	UserInfo_21	0.064936
WeblogInfo_26	0.013549	WeblogInfo_51	0.014910

2) 特征工程

特征工程主要包括成交时间离散化、类别特征变换、特征衍生、特征选择。

成交时间离散化：对用户行为信息表 Master 中的成交时间进行离散化处理。以起始时间为第一周，将日期变量按周离散化。

类别特征变换：统计用户的居住地省份和户籍省份特征的违约率，各取违约率最高的前五名省份进行二值化。之后使用 XGBoost 挑选重要的特征(图 2、图 3、图 4、图 5)，将特征重要性排名前三的城市进行二值化。

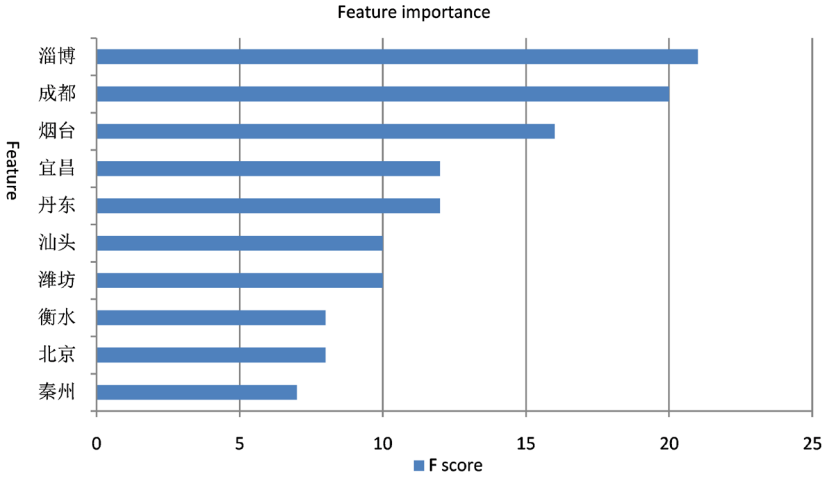

Figure 2. City importance ranking
图 2. 城市重要性排名

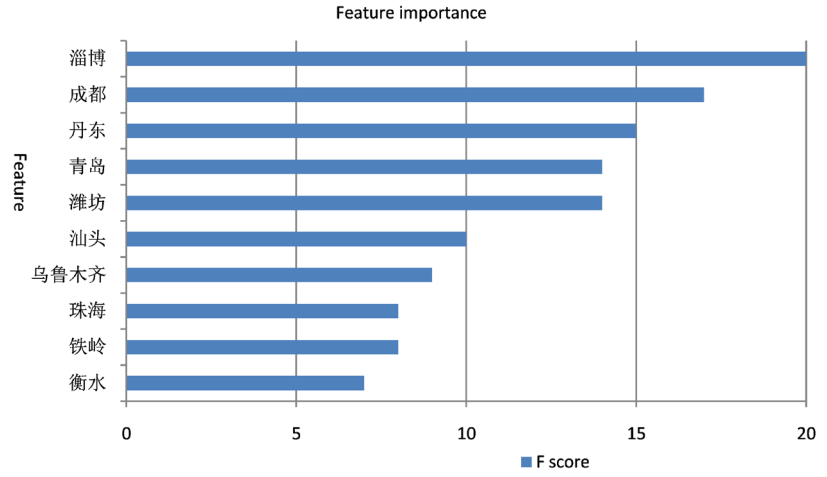

Figure 3. City importance ranking
图 3. 城市重要性排名

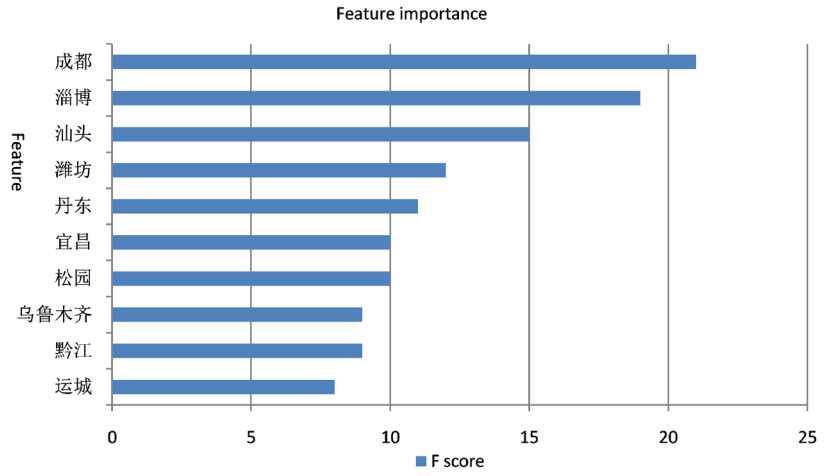

Figure 4. City importance ranking
图 4. 城市重要性排名

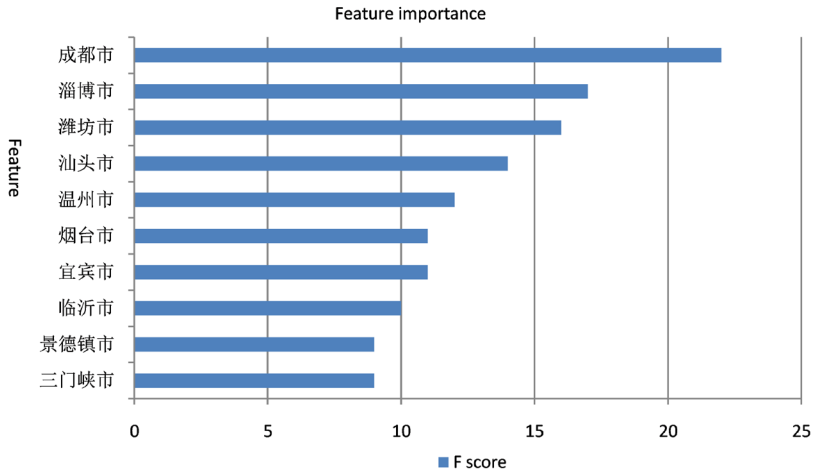

Figure 5. City importance ranking
图 5. 城市重要性排名

特征衍生：统计用户登录数据表 LogInfo 和用户信息修改数据表 Userupdate 中用户登录次数和用户更新信息的次数，并命名为 Log_count 和 Updat_count 加入到用户行为信息数据表 Master 中，新字段的缺失值由 0 填补。户籍省份和居住地省份是否一致衍生出一个新特征，由四个城市特征的非重复计数衍生生成登陆 IP 地址的变更次数。

特征选择：将数据按照 7:3 划分成训练集和测试集，并利用 XGBOOST 筛选特征，训练 10 个模型，并对 10 个模型输出的特征重要性去平均，最后对特征重要性进行归一化，表 5 是重要性排名前 16 的特征(cum_importance：累计重要性；norm_importance：归一化)，并删除重要度小于 0.01 的特征。

Table 5. Feature importance
表 5. 特征重要性

feature	fea_importance	norm_importance	cum_importance
ThirdParty_Info_Period2_6	5.931778e−02	5.931778e−02	0.059318
WeblogInfo_6	5.784138e−02	5.784138e−02	0.117159
ThirdParty_Info_Period1_6	4.975742e−02	4.975742e−02	0.166917
UserInfo_14	4.564730e−02	4.564730e−02	0.212564
ThirdParty_Info_Period3_6	3.624279e−02	3.624279e−02	0.248807
WeblogInfo_15	3.213665e−02	3.213665e−02	0.280943
Weeks	2.972517e−02	2.972517e−02	0.310668
ThirdParty_Info_Period2_3	2.206508e−02	2.206508e−02	0.332734
UserInfo_16	1.897712e−02	1.897712e−02	0.351711
ThirdParty_Info_Period5_1	1.783195e−02	1.783195e−02	0.369543
ThirdParty_Info_Period6_1	1.701123e−02	1.701123e−02	0.386554
ThirdParty_Info_Period5_2	1.570866e−02	1.570866e−02	0.402263
WeblogInfo_2	1.485803e−02	1.485803e−02	0.417121
ThirdParty_Info_Period3_2	1.469115e−02	1.469115e−02	0.431812
ThirdParty_Info_Period3_15	1.421459e−02	1.421459e−02	0.446026
ThirdParty_Info_Period3_5	1.410538e−02	1.410538e−02	0.460132
ThirdParty_Info_Period6_5	1.340561e−02	1.340561e−02	0.473537

3) 样本均衡

在进行模型训练之前要观察一下正负样本的比例，如果样本不均衡时直接使用原样本进行训练，则会使得模型倾向于关注占比高的那一类样本，进而对多数类样本识别率比较高，对少数类样本识别率比较低[9]，所以要进行样本均衡处理。对于不均衡样本，通常有两种方法使得样本均衡，增加正类样本数量的方法被称为过采样，减少负类样本数量的方法被称为欠采样。Chawla 等学者提出了 SMOTE 的方法来解决数据不均衡问题，SMOTE 的思想是对少数类样本的 K 个近邻随机抽取 N 个值进行随机线性插值，进而构成新的少数类样本[10]。

本实验样本中正负比例为 13:1，使用 SMOTE，采取过采样的方法解决类别不平衡问题，平衡后正负样本比例为 1:1。

4. 模型训练

4.1. 模型调参与融合

模型的参数对模型的预测结果的影响很大，本实验使用 Python 的 Hyperopt 包进行模型调参，Hyperopt 通过贝叶斯优化来调整模型的参数，而且相较于基于全局搜索的 GridSearch 方法而言，Hyperopt 速度更快效率更高；Random Search 虽然速度比较快，但是可能遗漏空间上一些比较重要的点，Hyperopt 则精度更高。而且 Hyperopt 支持暴力调参、随机调参等策略，也可结合 MongoDB 进行分布式调参。

模型融合是构建并结合多个学习器来完成学习任务，模型融合的方法主要有 Blending 和 Stacking。Stacking 是一种分层模型集成框架。以两层为例，第一层由多个基学习器组成，其输入原始训练集，第二层的模型以第一层基学习器的输出作为特征加入训练集进行再训练，从而得到完整的 stacking 模型。Blending 和 Stacking 大致相同，但是 Blending 的主要区别在于训练集不是通过 K-Fold 的 CV 策略得到预测值从而生成第二阶段模型的特征，而是将 K-Fold CV 换成 HoldOut CV。但是 Blending 可能会产生过拟合问题，Stacking 使用多次的 CV 会比较稳健，所以本实验选用 Stacking 的方法进行模型融合。

海量的离散特征 + 逻辑回归模型，因其较高的精度和较少的运算开销在业界广为使用。但是逻辑回归无法捕捉到非线性特征对标签的影响，因而提升逻辑回归精度的有效方法时构造有效的交叉特征。2014 年 FaceBook 提出了树模型 GBDT 和逻辑回归的融合模型[8]，利用 GBDT 构造有效的交叉特征，从根节点到叶子结点的路径代表部分特征组合的一个规则，提升树将连续特征转化为离散特征，明显提升了线型模型的精度。

所以本实验也采用树模型和线性模型相结合的方法进行模型融合。本实验的第一层模型是 XGBoost、随机森林和 LightGBM，对着三个模型进行调参后找到最优参数值，将其产生的预测值作为新特征，输入第二层模型逻辑回归模型中，进而对数据进行预测。

4.2. 模型评价标准

分类模型常用的评价标准有召回率、精确率、F1 分数、准确率、AUC。召回率又被称为查全率，是正确预测的样本中实际为正的样本的占比。精确率又被成为查准率，是预测为正的样本中实际为正的样本占比。召回率和精确率是互相矛盾的，一个取值高，另一个取值则会低，需要根据不同的情况选择使哪个指标更高。F1 分数是精确率和召回率的调和，同时兼顾了精确率和召回率，在多分类任务中，F1 分数是最常用的指标。准确率表示的正确预测的样本数占总数的比例。AUC 是接受者操作特征曲线下的面积，常用来评估二分类模型。

本文主要采用 AUC 和准确率分数为模型的主要评价标准。相较于召回率、精确率、F1 分数、准确率，AUC 在二分类任务中应用的更多。因为一般分类模型输出的都是概率，这时就需要设置一个阈值来

进行分类，阈值的大小则会对召回率、精确率、F1 分数、准确率产生影响，而 AUC 则没有这个问题。AUC 和准确率有时会产生矛盾，这时则优先选择 AUC，因为准确率是基于较佳的截断值进行计算的，但是这个较佳的截断值并不是总体分布的最佳截断值，只是某个随机样本的一个属性指标，而 AUC 基于所有可能的截断值进行计算，所以更稳健[11]。

5. 实验结果

5.1. 模型调参结果

使用 Hyperopt 对 XGBoost、随机森林、LightGBM 三个算法模型分别进行调参，调参前后对比如表 6 所示，可以发现调参后模型的 AUC 和准确率值均有所提升。其中随机森林调参后 AUC 提升，准确率却降低，因 AUC 相对准确率来说更稳健，所以两者冲突的时候以 AUC 为准，因此随机森林的性能在调参后是有所提升的。

Table 6. Compare before and after adjusting parameters
表 6. 调参前后对比

模型	调参前		调参后	
	AUC	准确率	AUC	准确率
XGBoost	0.6736	0.8473	0.7220	0.9098
随机森林	0.6329	0.8918	0.6950	0.8358
LightGBM	0.7028	0.8866	0.7278	0.9073

XGBoost、随机森林、LightGBM 三个算法模型的最优参数分别如表 7、表 8、表 9 所示：

Table 7. The optimal parameter of XGBoost
表 7. XGBoost 最优参数

参数	最优值
learning_rate	0.0512
n_estimators	411
max_depth	16
min_child_weight	6
subsample	0.55

Table 8. The optimal parameter of random forests
表 8. 随机森林最优参数

参数	最优值
max_depth	18
max_features	3
n_estimators	16

Table 9. The optimal parameter of LightGBM
表 9. LightGBM 最优参数

参数	最优值
bagging_fraction	0.55
learning_rate	0.0511
max_depth	18
num_leaves	85
num_trees	584

5.2. 不同算法模型结果对比分析

本实验对比了 XGBoost、随机森林、LightGBM、逻辑回归，以及模型融合的性能，如表 10 所示：

Table 10. Performance comparison of different models
表 10. 不同模型性能对比

模型	AUC	准确率	耗时
XGBoost	0.7220	0.9098	23.00
随机森林	0.6950	0.8358	1.72
LightGBM	0.7278	0.9073	11.79
逻辑回归	0.6813	0.6993	0.95
模型融合	0.7537	0.9180	43.77

由表 10 可知，四个单模型和融合模型的 AUC 排名由高到低是融合模型、LightGBM、XGBoost、随机森林、逻辑回归。在模型融合之前，对单模型的调参提升了单模型的 AUC，而经过 Stacking 的融合后，信用风险预测模型的 AUC 和准确率得到进一步提升，融合后的模型 AUC 比表现最差的逻辑回归提升了 0.0724，准确率提升了 0.2187；比表现最好的 LightGBM 的 AUC 提升了 0.0319，准确率提升了 0.0107。

融合后的模型虽然在 AUC 和准确率上都有所提升，但是确实耗时最长的，耗时 43.77，比四个单模型中耗时最长的 XGBoost 的耗时还要多近一倍的时间。这是因为模型的复杂度越高，耗时也就越长。在本实验中模型融合的耗时还是可以承受的，在实际应用中面对海量信息如果耗时太长则需要权衡一下耗时和预测准确率之间的关系。

6. 总结

本文针对金融领域的信用风险评估问题，建立了信用风险评估模型，以随机森林、XGBoost、LightGBM 为第一层模型，逻辑回归为第二层模型，使用 Stacking 的融合方法进行模型融合。最后结果表明模型融合的效果要优于单模型，但是耗时也更长。本文的创新点之一在于数据预处理方面，本文对原始特征进行挖掘，使得特征和预测目标之间的联系更明显一些，在实际应用中，可以考虑根据现有特征衍生出一些在区分高风险用户方面效果更好的特征；其次是树模型和线性模型的融合，将 Bagging 类算法的代表——随机森林、Boosting 算法的代表——XGBoost、以及 XGBoost 的改进算法 LightGBM 和逻辑回归进行了融合，并且最终取得融合效果较好。

基金项目

国家重点研发计划资助(National Key R&D Program of China)，项目编号：2017YFB1400700。

参考文献

[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): 53-58.

[2] Redmond, U. and Cunningham, P. (2013) A Temporal Network Analysis Reveals the Unprofitability of Arbitrage in the Prosper Marketplace. *Expert Systems with Applications*, **40**, 3715-3721. https://doi.org/10.1016/j.eswa.2012.12.077

[3] Malekipirbazari, M. and Aksakalli, V. (2015) Risk Assessment in Social Lending via Random Forests. *Expert Systems with Applications*, **42**, 4621-4631. https://doi.org/10.1016/j.eswa.2015.02.001

[4] 李昕, 戴一成. 基于 BP 神经网络的 P2P 网贷借款人信用风险评估研究[J]. 武汉金融, 2018(2): 33-37.

[5] Ke, G.L., Meng, Q., Finley, T., Wang, T.F., Chen, W., Ma, W.D., Ye, Q.W. and Liu, T.-Y. (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, **30**, 3149-3157.

[6] 李航. 统计学习方法[M]. 北京: 清华大学出版社, 2012: 78-79.

[7] Verikas, A., Gelzinis, A. and Bacauskiene, M. (2011) Mining Data with Random Forests: A Survey and Results of New Tests. *Pattern Recognition*, **44**, 330-349. https://doi.org/10.1016/j.patcog.2010.08.011

[8] Chen, T. and Guestrin, C. (2016) XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, August 13-17, 2016, 785-794. https://doi.org/10.1145/2939672.2939785

[9] Sun, Y., Wong, A.K.C. and Kamel, M.S. (2009) Classification of Imbalanced Data: A Review. *International Journal of Pattern Recognition and Artificial Intelligence*, **23**, 687-719. https://doi.org/10.1142/S0218001409007326

[10] Chawla, N.V., Bowyer, K.W., Hall, L.O., *et al.* (2002) SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, **16**, 321-357. https://doi.org/10.1613/jair.953

[11] Ling, C.X., Huang, J. and Zhang, H. (2003) AUC: A Better Measure than Accuracy in Comparing Learning Algorithms: Advances in Artificial Intelligence. 16*th Conference of the Canadian Society for Computational Studies of Intelligence*, *AI* 2003, Halifax, 11-13 June, 2003.