上海财经大学

Shanghai University of Finance and Economics

Shanghai University of Finance and Economics

东证期货杯

基于机器学习的

互联网金融借贷风险预测

队长姓名 郭宇

组员姓名 程雷 曹晨舟 陈麒旦 樊忠睿

指导老师 李卫明

参赛学校 上海财经大学

二〇一八年二月

基于机器学习的互联网金融借贷风险预测

摘要

当下互联网金融已蓬勃兴起，呈现出多种多样的业务模式和运行机制。但互联网金融的兴起同时引发了信用风险和用户欺诈等问题，急需通过信用评分模型提高风险控制水平。因此，建立信用体系模型来预测用户违约风险具有重要意义。

本文从传统的 Logistic、集成学习以及模型的 stacking 出发来构建模型，从而预测用户是否违约。首先，本文针对原始数据做了特征工程，其中包含变量衍生、缺失值处理、特征分箱以及类别不均衡处理等，共计生成数据 13 份。其次，本文运用了评分卡模型、GBDT、XGBoost、Random Forest、LGBM 以及模型的 Stacking 来预测用户是否违约，并通过 AUC 值、KS 值以及 F1-Score 三种评价指标来进行比较，选择出了综合效果最优的模型：XGBoost。最后，本文根据效果最优的模型来预测 Test 数据集，并给出每个用户的风险预测。

关键字：特征工程 评分卡 集成学习 Stacking

目录

插图清单

插表清单

一、 背景

当下互联网金融已蓬勃兴起，呈现出多种多样的业务模式和运行机制。金融机构能够突破时间和地域的约束，通过互联网技术加快业务处理速度，在互联网上为有融资需求的客户提供更快捷的金融服务。但互联网金融的兴起同时引发了信用风险和用户欺诈等问题，急需通过信用评分模型提高风险控制水平。

征信机构利用采集到的丰富信息对个人进行综合信用评价。在丰富海量的个人信用历史和信用行为数据基础上，采用数据挖掘方法得出的信用行为模式能够更加准确地预测个人未来的信用表现，提高操作效率，降低授信成本，精确估计消费信贷的风险，是金融机构内部评分不可替代的重要工具。

“大数据时代”下如何利用数据实现实际业务需求，以达到提高业务效率和精确率的目标，是统计人一直所探究的问题。因此，建立精准的信用评分体系预测借贷人信贷风险程度对于企业有着重要的意义。

二、 综述

从本质上来说，进行信用评分实际上是一个数据分组的过程：将贷款人分为“好”与“坏”两个组别，从而判断是否进行借贷。早期的方法中，统计方法和运筹学方法并驾齐驱：统计学方法主要有判别分析、广义回归；运筹学方法主要使用线性规划。随着计算机硬件性能快速提高、统计学习方法快速进步，近年来统计学习方法逐渐占据主流。一个统计学习方法建模的过程通常包含以下几个步骤：特征工程、特征选择、模型建立。

特征工程是对原始变量的信息加工。常用的特征工程有缺失值处理、异常值处理、连续变量离散化（BIN）、变量衍生等等。在本次研究中，考虑到数据的不平衡性，除去上面所述的常规特征工程方法，我们使用了 SMOTE 方法进行数据增强，调整正负样本比例至合适的值，使训练出的模型对正样本（违约）有更强的敏感性。

特征搜索按照搜索策略大致可以分为三种：全局最优、随机搜索和启发式搜索。全局最优方法效果优秀，但因时间复杂度过大，在应用上适用面较小；随机搜索方法可检索出高性能子集，但会存在较高的不确定性；启发式搜索方法快速、便捷，但牺牲了全局最优。每种方法都有优缺，综合考虑，我们选择了启发式搜索方法中的特征重要性排序，选取一定数目的特征变量。

统计学习模型在信用评估建模上的应用繁杂，现常用方法大致可以归为：

传统统计学习类模型：以逻辑回归为例。逻辑回归模型建模过程较为简单，模型预测性能良好。但也因为逻辑回归的简单，其无法模拟变量间非线性复杂联合影响的效应，针对如今日益复杂的金融市场背景，其正逐渐被能体现复杂模式的模型取代。但由于逻辑回归解释性良好，其在统计建模中依然占有一席之地。

集成学习：GBDT、XGBoost、LGBM、Random Forest 等。集成学习通过联合弱监督模型，得到更全面、性能更强大的强监督模型[5]。模型通过联合，可以模拟复杂的借贷影响效应，从而获得更好的预测效果。

深度学习：由于深度学习模型可以自行地模拟任意非线性效应，近期在信用评估应用广泛。但由于其网络内部的“不可见性”，模型的解释性相对较差。

三、 问题分析

(一) 数据来源及介绍

大赛官方给出了某贷款机构的历史业务数据作为原始数据。其中 2 个数据集 CONTEST_BASIC_TEST 和 CONTEST_BASIC_TRAIN 为基本信息表，包含了共 40000 个借贷人的基本信息；1 张 CONTEST_FRAUD 表为欺诈数据集，包含了该 40000 个借贷人是否欺诈的变量。另外十张表为个人征信报告的信息，通过对每个借贷人查询征信报告，可以得到该人的征信数据。

具体官方数据集列表如下，衍生后所有表用 REPORT_ID 作为主键链接到基本信息表。

表 1 原始数据字典

数据集名	中文简介	数据量	变量数
CONTEST_BASIC_TRAIN	训练数据集包含借贷时间、个人信息、是否违约等变量	30000	11
CONTEST_BASIC_TEST	测试数据集包含除“是否违约”外训练集所有变量	10000	10
CONTEST_EXT_CRD_HD_REPORT	征信报告主表，包含报告查询时间、查询原因、查询 ID	40000	4
CONTEST_EXT_CRD_CD_LN	包含每笔贷款截至查询日的基本信息和还款情况，每个人可有多笔贷款	357196	22
CONTEST_EXT_CRD_CD_LND	包含每笔的贷记卡至查询日的基本信息和还款情况，每个人可有多张卡	324299	20
CONTEST_EXT_CRD_IS_CREDITCUE	征信报告信用提示信息，概括了查询个人的基本信用情况	39970	11

CONTEST_EXT_CRD_IS_SHAREDEBT	包含未销户贷记卡或者未结清贷款的基本信息	76246	11
CONTEST_EXT_CRD_IS_OVDSUMMARY	逾期透支信息汇总	76216	6
CONTEST_EXT_CRD_QR_RECORDDTLINFO	审批查询记录明细，即在此查询之前该借贷人的征信被查记录	654329	4
CONTEST_EXT_CRD_CD_LN_SPL	贷款特殊交易，即贷款全部或者部分提前结清/延后结清信息	67725	6
CONTEST_EXT_CRD_CD_LND_OVD	贷记卡逾期/透支记录信息，包含了每个借贷者的逾期记录	199644	4
FRAUD	欺诈标志	40000	2

除了基本信息表的数据之外，其它信息为 40000 借贷者的全量征信数据，一个借贷者可以对应多笔贷款和多张信用卡，因此需要对原始的征信数据进行变量衍生。变量衍生我们将在第四部分特征工程详细介绍。

(二) 建模思路

本文针对官方给定数据，基于特征工程对原始数据进行加工处理后，选择多个模型进行测试，并根据多个评价指标进行模型效果评价。最后依据特定的评价指标选择出效果最好的模型，对题目所给的 Test 数据集中的客户预测是否违约，具体建模流程图如图 1 所示：

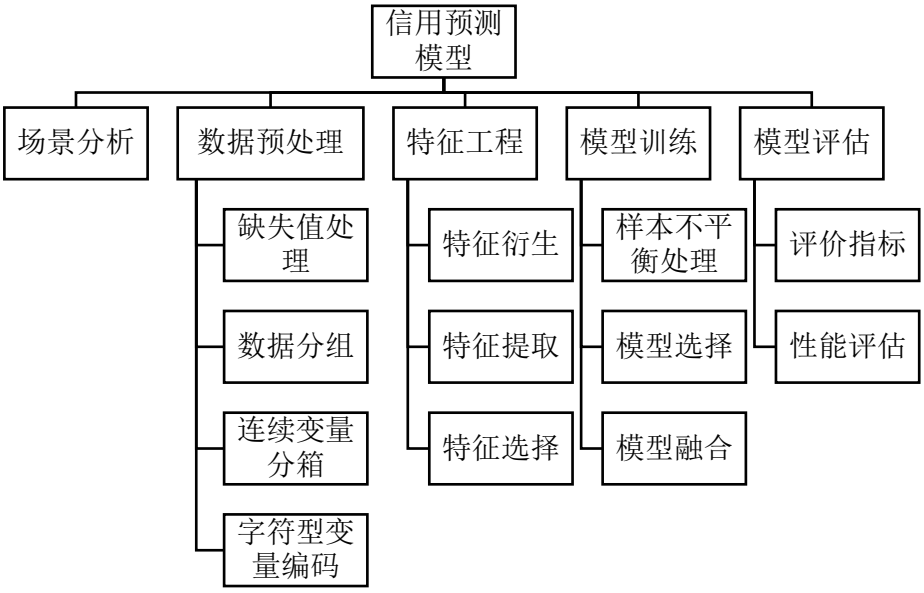


图 1 建模流程图

四、 特征工程

有这么一句话在业界广泛流传：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。那特征工程到底是什么呢？顾名思义，其本质是一项工程活动，目的是最大限度地从原始数据中提取特征以供算法和模型使用，故本文首先讲述我们对原始数据所做的特征工程。

我们的特征工程分为多个部分，分别为变量衍生（如身份证信息挖掘等）、缺失值处理、特征分箱（BIN）、字符变量重编码、类别不均衡处理。

(一) 变量衍生

根据已有征信数据集并结合具体的业务含义，我们衍生出新的可解释性变量。以包含每笔贷款详细状态的数据集 CONTEST_EXT_CRD_CD_LN 为例，由于单个借贷人可有多条贷款记录，则可以对单个借贷人贷款历史数据汇总，得到如最近 3 个月贷款违约笔数、最近 6 个月贷款违约笔数等衍生变量信息。

同理，通过对于各征信数据集进行变量衍生，共得到了 513 个新变量，具体衍生字段及各表变量衍生数目如表 2 所示。

表 2 衍生字典表

原始数据集名	衍生变量	衍生后数据量	衍生变量数
CONTEST_EXT_CRD_HD_REPORT	REPORT_ID（KEY） 本次查询时间 本次查询原因 本次查询机构	40000	4
CONTEST_EXT_CRD_CD_LN	REPORT_ID（KEY） 贷款笔数 贷款总额 未结清贷款笔数 未结清贷款总额 未结清贷款余额 最近 3/6/12 个月贷款违约笔数 …	35594	179
CONTEST_EXT_CRD_CD_LND	REPORT_ID（KEY） 贷记卡张数 贷记卡总额度 未结清贷记卡张数	39858	151

	未结清贷款总额度 未结清贷款余额度 最近 3/6/12 个月贷记卡违约笔数 …		
CONTEST_EXT_CRD_IS_CREDITCUE	REPORT_ID（KEY） 个人住房贷款笔数 个人商用贷款笔数 首张贷记卡发卡至今月份数 …	39970	11
CONTEST_EXT_CRD_IS_OVDSUMMARY	REPORT_ID（KEY） 总逾期贷款笔数 总逾期贷记笔数 贷款总逾期金额 …	25404	13
CONTEST_EXT_CRD_IS_SHAREDEBT	REPORT_ID（KEY） 未结清贷款机构数 未结清贷款总额度 最近六个月未结清贷记卡信用卡使用额度 …	39948	28
CONTEST_EXT_CRD_QR_RECORDDTLINFO	REPORT_ID（KEY） 最近 3/6/18 个月贷款审批、/信用卡审批查询次数 最近 3/6/18 个月贷款审批、/信用卡审批查询机构数 …	39865	59
CONTEST_EXT_CRD_CD_LN_SPL	REPORT_ID（KEY） 全额提前还款笔数 提前还款笔数 提前还款金额 …	14190	37
CONTEST_EXT_CRD_CD_LND_OVD	REPORT_ID（KEY） 贷记卡最近 6/12/24 月发生 30/60/90 天以上逾期数量 贷记卡最近 6/12/24 逾期金额 …	16429	33
FRAUD	REPORT_ID（KEY） 欺诈标志	40000	2

通过衍生后的数据集与 40000 借贷者的基础信息表拼接，最后得到包含 40000 条观测值、528 个变量的宽表作为模型建立的数据基础。

(二) 缺失值处理

缺失值对于模型建立有着巨大影响，如果随意填补了样本缺失的重要变量，则模型效果将大打折扣；如果删除所有含有缺失的样本，则减少了用于模型训练的样本数目。

本文共采用了表 3 的三种方法进行缺失值处理，分别是中位数填充法、赋 0 法、单列类法。

表 3 缺失值处理方法

	填充方法
中位数填充法	把原始数据进行排序后，计算出每个变量的中位数，然后对缺失值进行赋值
赋 0 法	原始数据的缺失有可能是因为部分数据为 0 而没有进行填写，故可以把这一部分数据的缺失当是 0 来处理
单列类法	由于缺失值的不确定性，以及引起每个样本缺失的原因不尽相同，采用单列类法把原始数据的缺失值重新分为一类。本文对重新分为的类赋值为-9。

根据三种不同的方法，我们将缺失数据填补为三个不同的版本进行模型训练，以寻找模型表现最好的填补方式后进行模型部署。

(三) 特征分箱

连续变量在进入模型时，也可以有不同的形式：可以按照其原本数值直接作为模型变量，也可以通过离散化处理以“序”的形式进入模型，即特征分箱。

1. 特征分箱的优点

① 稀疏向量内积乘法运算速度快，计算结果方便存储，容易扩展；

② 离散化后的特征对异常数据有很强的鲁棒性；

③ 逻辑回归属于广义线性模型，表达能力受限；单变量离散化为 N 个后，每个变量有单独的权重，相当于为模型引入了非线性，能够提升模型表达能力，加大拟合；

④ 离散化后可以进行特征交叉，由 M+N 个变量变为 M*N 个变量，进一步引入

非线性，提升表达能力；

2. 特征分箱的方法

基于逻辑回归的评分卡需要进行分箱操作。本文拟采用基于 Best-KS 和最小 Bin 的方法产生分裂点，最后通过 IV 值选择最优分裂点组合。以连续性变量为例，其算法流程如下所示：

连续变量处理步骤

Step1: 分裂点的产生

a.排序

b.计算每一点的 KS 值

c.选取最大的 KS 值对应的特征值，用该特征值将特征分为大于该值和小于该值两端

d.对于每一部分，循环 b、c 步骤，直到满足终止条件

Step2: 分裂的终止条件

a.下一步分箱，最小的箱的占比低于设定的阈值（0.05）

b.下一步分箱后，有一箱的对应的 y 的类别全部为 0 或者 1

c.下一步分箱后，WOE 值不单调

Step3: 选出最优的分裂组合

a.对于产生出来的所有分裂点，我们选出 9 个分裂点来进行分箱操作

b.对产生的所有分裂点组合求 IV 值，选择最优的 IV 值确定分裂点

对于离散程度很高的分类变量，对不同类别先编码，再依据连续变量的方式进行相同的分箱操作。

为了便于业务上的解释，变量进行分箱后，各组的 WOE 值必须为单调的，一个良好的变量分箱后各组 WOE 值如图 2 所示：

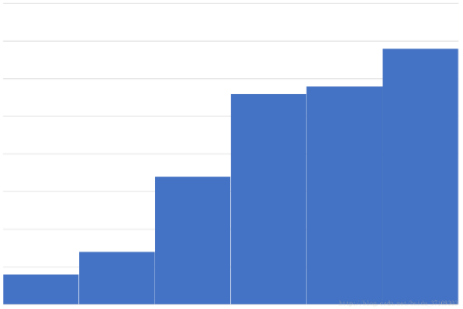


图 2 WOE 单调性示意图

假定该变量被分成了 6 个箱，X 轴为年龄，Y 轴为 WOE 值（“当前分组中响应客户占所有响应客户的比例”和“当前分组中没有响应的客户占所有没有响应的客户的比例”的差异），这样就可以很好地进行解释：年龄越大，则产生坏样本的可能性越大。如果分箱之后 WOE 不单调也不呈现 U 型，那么模型在这个变量上的可解释性就成问题了。所以要注意分箱后该变量 WOE 的单调性。

本文的数据集 Train 分为两部分数据，训练集和测试集。值得注意的是，在特征分箱的过程中，本文不对全部数据进行排序确定分裂点，而是对 Train 中的训练集进行排序确定分裂点，并在测试集上采用相同分割方式进行分箱操作。这样做更加合理，更加符合测试集数据的未知性，从而减少过拟合。

(四)　字符变量重编码

在研究因变量时，解释变量除了数值型变量，还有一些字符型变量，比如性别、婚姻状况、教育程度等。以教育程度为例，如果人为地设定在进入逻辑回归模型时，“初中”用 1 表示，“高中”用 2 表示，“大学”用 3 表示，则相当于默认了在其他各解释变量相同的情况下，教育程度由“初中”变为“高中”跟由“高中”变为“大学”对因变量的影响程度是相当的，这显然不合理。

此时需要引入虚拟变量。如果某个字符型变量有 n 种选择，则将其用哑变量引入模型时，设置 n-1 个哑变量，以避免完全的多重共线性。假如 X1 教育程度为上述 3 种，则引入 2 个哑变量，“初中”则两个变量均为 0，“高中”则仅 X11=1，“大学”则仅 X12=1。

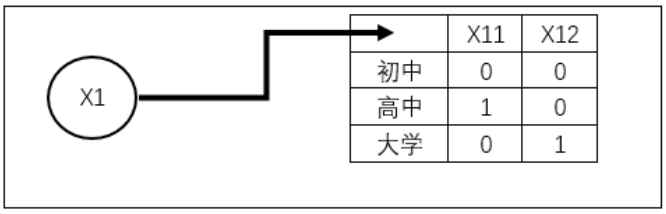


图 3 字符变量重编码示意图

引入哑变量可使线性回归模型变得更复杂，但对问题描述更简明而且更接近现实。相比于作为一个变量进入模型， n-1 个哑变量也更加具有解释意义和实际适用性。

基于此原理，我们采用 SAS 宏程序对提供的 BASIC 表中个人数据的字符型变量均进行重编码处理，生成对应的哑变量备用于逻辑回归。

(五)　类别不均衡处理

此次本赛题逾期用户只占总样本的 6%，因此需要使用算法来解决这一类别不平衡问题。解决类别不平衡问题主要有欠采样和过采样两种方法。欠采样方法牺牲了许多样本信息，不利于模型的优化提升。因此，我们组采用过采样方法。

Smote 算法是用于解决类别不平衡问题的一种过采样方法。其在确保大类样本的信息完整性前提上，增加了小类样本数量，利于模型提升其小类样本分类的准确性。

1.　Smote 的基本思想[4]

Smote 算法是依据两类样本在 p 维空间上分布特征，找出小类样本的分布空间，在小类样本与小类样本之间生成新的小类样本。

为了便于理解，以二维样本空间为例，我们绘制了 Smote 思想图。如图 4 是两类样本的分布情况：

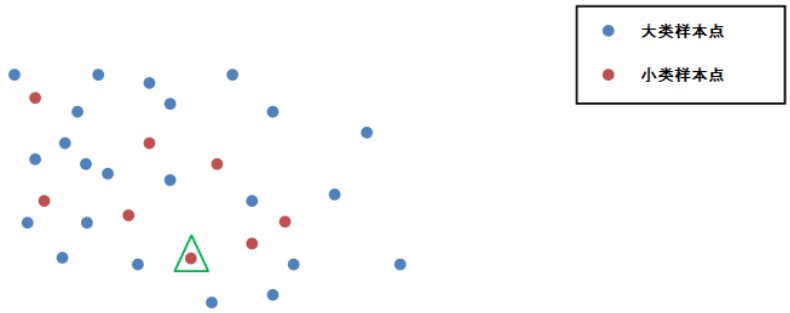


图 4 smote 算法原理展示（1）

首先从小类样本中随机选择一个小类样本点，如图 4 中三角形标注的小类样本点。

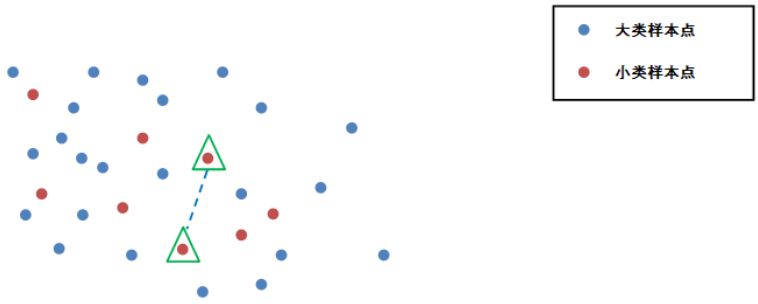


图 5 smote 算法原理展示（2）

然后从该小类样本点近邻的小类样本点中随机选取一点，并连线。如图 5（原理展示 2）中的两个三角形标注的小类样本点及连线。

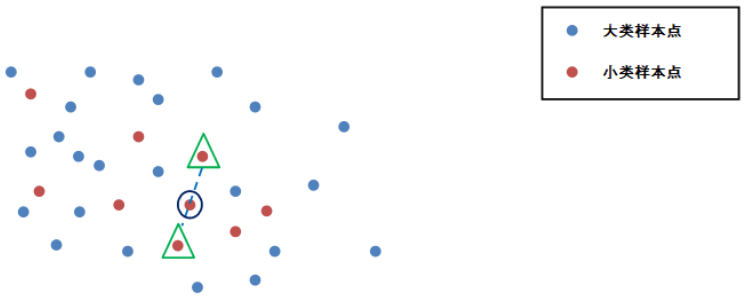


图 6 smote 算法原理展示（3）

最后在小类样本点的连线之间，随机选取一个位置生成一个新的小类样本点。如图 6（原理展示 3）中用圆圈标注的小类样本点。

2. 传统 Smote 算法所面临的问题[3]

一般 Smote 算法的确提升了小类样本的数量，但有时 smote 生成的样本并不能有效反应小类样本分布的性质，可能反而会对模型造成干扰，导致模型分类性能下降。如图 7（原理展示 4）中两个小类样本的分布情况。

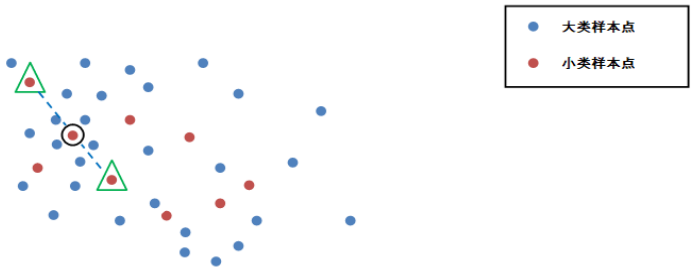


图 7 smote 算法原理展示（4）

对这两个小类样本进行连线，生成的新样本会落在大类样本空间中，新生成的小类样本此时不能代表小类样本的分布性质，此时新生成的样本成为噪音。因此需要对 smote 算法改进以使生成的小类样本能有效反应小类样本分布情况。

我们组的思路是针对新生的小类样本点，将其随机与近邻大类样本连接，其距离定义为 R。以该生成样本点为中心画出以 R 为界的样本空间，记空间中小类样本个数为 s，大类样本个数为 d。若比值 k=d/s 大于一定阈值 α，则生成样本点删除，否则保留。

五、 特征选择

我们基于原始特征生成了排序特征、离散特征等，总计 500 多维。如果这么多维特征全部进入模型，一方面可能会导致维数灾难，另一方面很容易导致过拟合。因此，我们需要做降维处理。

常见的降维方法有 PCA，t-SNE（计算复杂度很高）。我们尝试了 PCA，效果并不好，猜测原因是大多数特征含有缺失值且缺失值个数太多，而 PCA 前提假设数据呈高斯分布，赛题数据很可能不满足。除了采用上述降维算法之外，也可以用特征选择来降低特征维度。特征选择的方法很多：最大信息系数（MIC）、皮尔森相关系数（衡量变量间的线性相关性）、正则化方法（L1，L2）、基于模型的特征排序方法。比较高效的是最后一种方法，即基于学习模型的特征排序方法，这种方法有一个好处：模型学习的过程和特征选择的过程是同时进行的，因此我们采用这种方法。

基于决策树的算法（如 Random Forest，Boosted Tree）在模型训练完成后可以输出特征的重要性。Random Forest 自身所带的特征重要性排序在各类竞赛中备受推崇，是一个很好的选择；同时，XGBoost 是 Boosted Tree 的一种实现，效率和精度都很高，在各类数据挖掘竞赛中也被广泛使用。所以，我们分别采用了 Random Forest 和 XGBoost 输出特征重要性来做特征选择，形成多种组合再通过模型检验来选择效果最优的组合。

同时，在选出的一些变量中，有一些变量并不能使用。所以，我们做了一些特征剔除，如：借贷人的所在省份。

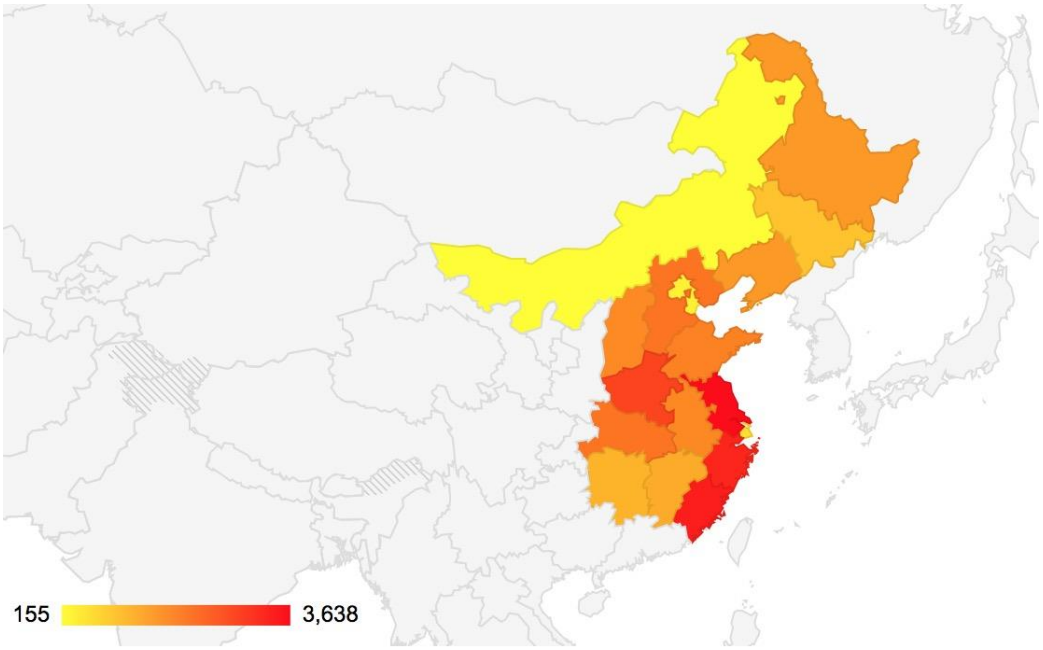


图 8 有标签数据户籍省份热点图

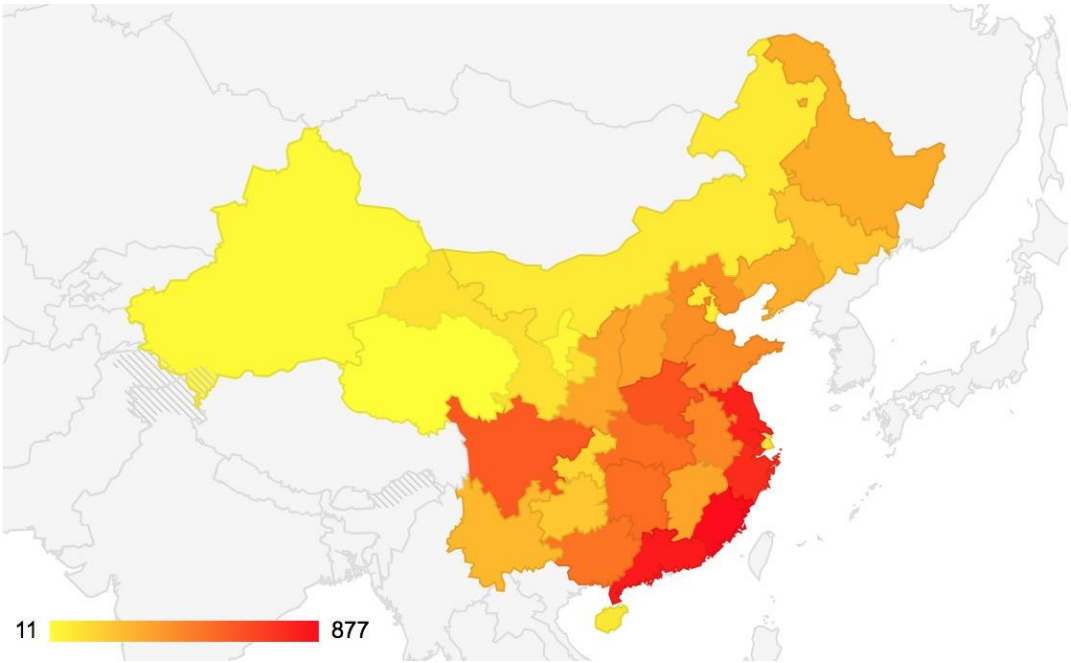


图 9 无标签数据户籍省份热点图

在变量衍生的过程中，有很多可以挖掘的信息，比如可以根据身份证号码识别出某个借款人的户籍省份作为新变量——省份。而出生环境作为影响个人行为和性格特征的重要因素，或许会在一定程度上影响贷款行为。因此，我们首先考虑将省份信息加入模型中。

但是，构建完成的模型中，省份这个变量的重要程度非常高，甚至远远高于央行征信报告数据反应的信息量。因此，将该变量进行详细分析。通过绘制图 8（有标签的借款人）和图 9（无标签的借款人）在户籍省份的热点图可以发现，

两个数据集在该变量上的分布呈现出明显的不一致，有标签的借款人户籍省份的覆盖区域基本集中在东北和华北地区，然而无标签的借款人户籍省份覆盖了除西藏外的全部省份，出现了新的类别。此时由于数据分布的严重差异，若将基于有标签数据建立的模型直接迁移到无标签的模型上是非常不恰当的。

此外，在有标签的数据中，用户分布的省份中近一半以上的省份无任何违约，即在该变量的多个类别下，好用户的纯度为 1，比如山东省。而又因为该变量的信息重要性非常高，此时模型直接可以依据新用户是山东人这一个指标判别其不存在任何违约风险。这无论是基于业务还是基于常理都是十分不合理的。

综上，尽管该变量进入模型后模型表现十分优越，我们也予以剔除。

六、 模型的建立与求解

(一) 建立测试集和训练集

根据题目所给的数据 Train 和 Test 数据，我们通过上述特征工程工作分别把 Train 和 Test 数据衍变为 13 份数据，数据分别对应于不同的缺失值填补方式、smote 比例。然后我们通过把 Train 数据集（带 label）划分为两份数据，一份数据集作为训练集，一份数据集作为测试集。通过不断提高在 Train 里面的测试集的评价指标，以此来达到优化模型的效果。最后用最优化的模型去测试 Test 数据集，得到最终结果。图 10 仅给出了以中位数填补缺失值后的样例。

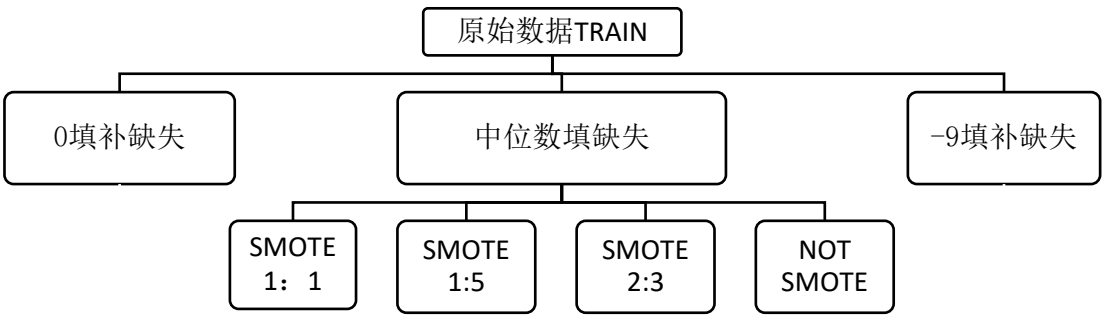


图 10 数据集字典样例图

(二) 算法介绍

1. 逻辑回归与评分卡

在审批消费信贷业务时，信用评分卡被包括商业银行、互联网金融在内的金融行业广泛使用。信用评分卡模型能根据申请人提供的申请信息（如年龄、学历、收入和工作年限等）给出评估其风险高低的分数，这个分数也叫做信用评分。其基本应用是：放贷审批层首先根据风险控制要求设定一个取舍点（cut-off），如果申请人的分数高于该取舍点，则可以获得审批。反之，则该申请人被拒绝。若申请者得到分数处于“灰色”区域，那么信贷员将使用传统方法对该申请者进行评估，并进行进一步调查。

如表 4 所示，某信用评分模型只考虑三个特征因素：年龄、性别和收入。该放贷机构对这些变量做了特殊的数据清理（比如把连续型变量划分为若干个离散的区间(bin)，把水平数太多的离散变量做合并），模型的最终结果以信用评分卡的形式展现。评分模型对不同的因素特征赋予不同的分数，这个分数主要是通过逻辑回归，在考虑如特征因素的预测强度、特征因素间的关系和可操作性等多方面因素之后得到的。可以用分数的总和来度量消费者信用风险的大小，分数高表明该用户违约风险低，分数低的表明风险高。

表 4 考虑三个特征的信用评分卡样例

特征	**品质属性**	**评分**
年龄 1	26 岁以下	100
年龄 2	26-35 岁	120
年龄 3	35-37 岁	185
年龄 4	37 岁以上	225
性别 1	女	180
性别 2	男	90
收入 1	1000 元以下	120
收入 2	1001-3000 元	140
收入 3	3001-5000 元	160
收入 4	5001-10000 元	200
收入 5	10001 元以上	240

放贷机构的风险经理会事先估计出最合适的临界值，假设在这个例子中临界值为 490 分。若该机构迎来了两个申请人甲和乙。甲是月收入为 4000 元的 27 岁女士，则她可获得的分数为 120+180+169=460 分，低于临界值 490 分，因此银行

就拒绝了她的申请。乙是月收入为 11000 元的 55 岁男士，则他得到的评分为 225+90+240=555 分，远远的高于临界值 490 分，因此银行就批准了他的申请。同样本案例中，企业的风控经理可以根据贷款人的特征因素进行信用评分，从而对贷款的申请和风险的控制进行判断。

通过对本文数据变量衍生后的筛选，连续变量基于最优 IV 值离散化，以及特征筛选后，我们进行逻辑回归建模，并通过分数转化得到信用评分卡。全部卡模型在（三）模型求解与评价中展示。

2. GBDT

GBDT 是集成学习 Boosting 家族中一个重要成员，其原理是利用弱学习器进行迭代得到最终强学习器[1]。在 GBDT 的第 T 次迭代中，依据 T-1 次迭代得到的学习器 F（t-1）和损失函数 L（t-1），找到一个 cart 回归树弱学习器 h（t），使得第 T 次迭代的损失函数 L（t）达到尽量小。

二元 GBDT 分类模型伪代码如下[2]：

输入：

训练数据集 D，样本数目 N，步长 v，模型迭代次数 T

输出：

GBDT 提升树$\mathrm{f}_i(x)$

Step1: 初始化$\mathrm{f}_0(x)=\mathrm{argmin}_\beta \sum_{i=1}^{N}\mathrm{L}(\mathrm{y},\beta)$

Step2:

For t=1 to T:

For i=1 to N:

$$\mathrm{z}_i=-\frac{\partial L(\mathrm{y}_j,\mathrm{f}_{i-1}(x))}{\partial \mathrm{f}_{i-1}(x)}$$

$a=argmin_{a,\beta}\sum_{i=1}^{N}[z_i-\beta h_i(x;a)]$

$\beta=argmin_\beta \sum_{i=1}^{N}\mathrm{L}(y_i,\mathrm{f}_{i-1}(x)+\beta h_i(x;a))$

Step3:更新提升树$\mathrm{f}_i(x)=\mathrm{f}_{i-1}(x)+\mathrm{v}\beta h_i(x;a)$

end for

end for

其中损失函数$L(\mathrm{y}_j,\mathrm{f}_{i-1}(x))$在 GBDT 分类模型中使用对数似然损失函数，这里考虑到分类模型预测值不连续的特点。

3. XGBoost

XGBoost 是一种改进的 GBDT 算法，该算法与 GBDT 有很大的区别。GBDT 在优化时只用到一阶导数，XGBoost 则同时用到了一阶导数和二阶导数，同时算法在目标函数里将树模型复杂度作为正则项，用于避免过拟合。

Step1: 找出 XGBoost 算法目标函数：

$$J(f_t) = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + f_t + C$$

Step2: 根据 Taylor 展开式：$f(x+\Delta x) \approx f(x) + f'(x)\Delta x + \frac{1}{2}f''(x)\Delta x^2$，同时令：

$$g_i = \frac{\partial L\left(y_i, \hat{y}_i^{(t-1)}\right)}{\partial \hat{y}_i^{(t-1)}}, h_i = \frac{\partial^2 L\left(y_i, \hat{y}_i^{(t-1)}\right)}{\partial \hat{y}_i^{(t-1)}}$$

得到：$J(f_t) \approx \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t-1)} + g_i f_t(x_i) + \frac{1}{2}h_i f_i^2(x_i)\right) + (f_t) + C$

Step3: 找出正则项，决策树的复杂度可考虑叶结点树和叶权值：

$$(f_t) = \gamma^{T_i} + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2,$$

其中 T_t 为叶结点数，w_j 为 j 叶子结点权重。

计算出：

$$J(f_t) \approx \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t-1)} + g_i f_t(x_i) + \frac{1}{2}h_i f_i^2(x_i)\right) + (f_t) + C$$

$$= \sum_{j=1}^{T_t}\left[\left(\sum_{i\in I_j} g_i\right)w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)w_j^2\right] + \gamma^{T_i} + C$$

Step4: 通过定义 $G_j = \sum_{i\in I_j} g_i$，$j = \sum_{i\in I_j} h_i$，计算出最终的目标函数

$$J(f_t) = -\frac{1}{2}\sum_{j=1}^{T_t}\frac{G_j^2}{j+\lambda} + \gamma^{T_i}$$

简化后的 XGBoost 原理[7] [10]为：

其中 Step3 中需要假定某决策树的叶结点数目为 T，每个叶结点的权值为 $\vec{w} = (w_1, w_2 \dots w_T)$。决策树的学习过程，就是构造如何使用特征得到划分，从而得到这些权值的过程。样本 x 落在叶结点 q 中，定义 f 为：$f_t(x) = w_{q(x)}$。

最后利用式上面式子来寻找出一个最优结构的树加入到模型中，通常情况下枚举出所有可能的树结构是不可能的，因此我们使用贪心算法来寻找最优树结构。

4. LGBM

LightGBM 对传统的 GBDT 等提升树进行了优化和改进，使其在准确率和运行速率上得以提升。LGBM 与 XGBoost 在准确率上可以媲美，并且其在运行速率上大大加快。LGBM 基于 XGBoost 的改进与优点如图 11 所示：

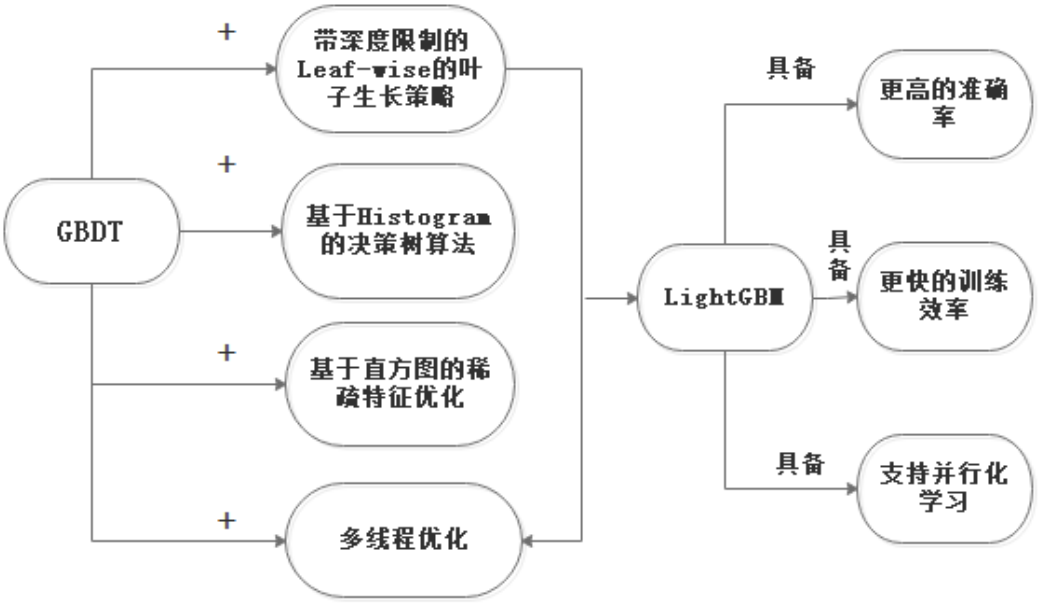


图 11　LGBM 基于 GBDT 的提升图

如图 11 所示，LGBM 在 GBDT 的基础上有两大特点：1、基于 Histogram 的决策树算法；2、带深度限制的 Leaf-wise 的叶子生长策略。其具体改动如下：

① Histogram 算法

直方图算法的基本思想：先把连续的浮点特征值离散化成 k 个整数，同时构造一个宽度为 k 的直方图。遍历数据时，根据离散化后的值作为索引在直方图中累积统计量。当遍历一次数据后，直方图累积了需要的统计量，然后根据直方图的离散值，遍历寻找最优的分割点。

② 带深度限制的 Leaf-wise 的叶子生长策略

Level-wise 过一次数据可以同时分裂同一层的叶子，容易进行多线程优化，也好控制模型复杂度，不容易过拟合。但实际上 Level-wise 是一种低效算法，因为它不加区分地对待同一层的叶子，带来了很多没必要的资源浪费，因为实际上很多叶子的分裂增益较低，没必要进行搜索和分裂[6]。

Leaf-wise 则是一种更为高效的策略：每次从当前所有叶子中，找到分裂增益最大的一个叶子，然后分裂，如此循环。因此同 Level-wise 相比，在分裂次数相

同的情况下，Leaf-wise 可以降低更多的误差，得到更好的精度。

Leaf-wise 可能会长出比较深的决策树，产生过拟合。因此 LightGBM 在 Leaf-wise 之上增加了一个最大深度限制，在保证高效率的同时防止过拟合。

5. Random Forest

随机森林[11]是一种组成式的有监督学习方法。随机森林的思想是 Bagging 多个预测模型，并将模型的结果汇总以提升分类准确率。

随机森林在算法上涉及了对于样本单元和变量进行抽样，从而生成大量决策树的过程。对每一个样本单元来说，所有决策树依次对其进行分类，所有决策树预测分类中的众数类别即被认为是随机森林所预测的这一样本单元的类别。

假设训练集共有 N 个样本单元，M 个类别，则随机森林的操作流程如下：

Step1: 从训练集中随机有放回地抽取 N 个样本单元，生成大量决策树。
Step2: 在每一个节点随机抽取 m<M 个变量，将其作为分割该节点的候选变量。每一个节点处的变量数目应该一致。
Step3: 完整生成所有决策树，无需剪枝（最小节点为 1）。
Step4: 终端节点的所属类别由节点所对应的众树类别决定。
Step5: 对于新的观测点，用所有的树对其进行分类，类别由多数决定原则生成。

6. 模型 stacking

Stacking 是一种集成学习技术，是对已训练的多个模型进一步集成训练的方法。我们先利用原训练数据集进行 XGBoost、GBDT、Random Forest 模型训练，将 XGBoost、GBDT、Random Forest 对测试集预测结果作为下一层模型学习的数据输入，然后在第二层使用 Logistic Regression 进行最终结果的预测输出。

Stacking 流程如下：

输入：训练数据 D
输出：集成分类器 H
Step1：基础模型训练
基于训练数据 D 训练 XGBoost 模型
基于训练数据 D 训练 GBDT 模型
基于训练数据 D 训练 Random Forest
Step2：构建基于基础模型的预测值集合
XGBoost 模型对于测试集数据预测得到 p1
GBDT 模型对于测试集数据预测得到 p2
Random Forest 模型对于测试集数据预测得到 p3
P=(p1,p2,p3)
Step3：生成集成模型
基于预测值集合 P 训练 Logistic 模型 H
Step4：返回模型 H

本文根据上述算法，计算出了每个用户的信用风险，结果部分见模型求解与评价。

(三) 模型的求解与评价

1. 模型评价指标体系介绍

本文针对所给出的众多模型，提出了三种评价指标来评判模型的优异程度，这三种评价指标分别是 AUC 值、KS 值以及 F1 SCORE。

在进行讲述三种评价指标之前，首先需要讲一下预测值与真实值所构成的混淆矩阵，如表 5 所示：

表 5 混淆矩阵

Confusion matrix		真实值	
		positive	negative
预测值	positive	TP	FP
	negative	FN	TN

如表 5 所示，本文把预测出来的正样本分为 TP 和 FP，预测出来的负样本分为 FN 和 TN。其中 TP 表示预测正确的正样本，FP 表示预测错误的正样本；FN 表示预测错误的负样本，TN 表示预测正确的负样本。

下面，本文将在表 5 混淆矩阵的基础上介绍这三种评价指标。

① **AUC**

AUC 可以说是在比赛中二分类最常见的评价指标，在天池和 Kaggle 中二分类问题很多采取该指标。因为很多机器学习的模型对分类问题的预测结果都是概率，如果要计算 accuracy，需要手动设定一个阈值（很多软件默认是 0.5，但实际 0.5 可能不是最优的）。如果预测概率高于这个阈值就判为 1，小于这个阈值就判为 0，这个阈值就很大程度上影响了 accuracy 的计算，而 AUC 就避免了这种计算。另外当样本有偏时比如 1 的比率只有 1%时，accuracy 的区分度就很难体现出来，即使不做模型全判为 0 也有 99%的准确率。

AUC 的全称是(Area Under then Curve Of ROC)，即 ROC 曲线下方的面积。

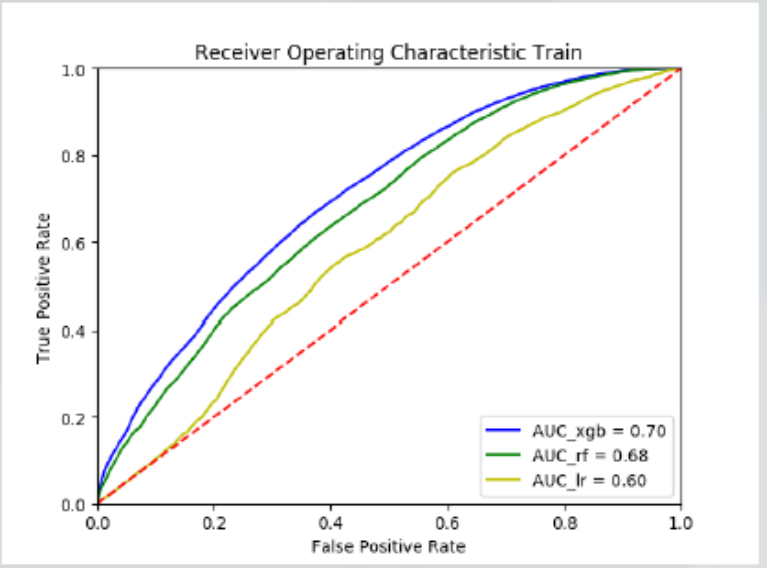


图 11 ROC 曲线示意图

图 11 分别为三个模型在同一个数据上的 ROC 曲线，通常单个的模型 ROC 曲线并不能看出较有效的价值，我们通常会把不同模型画在同一张 ROC 曲线上进行比较。

ROC 曲线的横轴为 False Positive Rate，也叫伪阳率（FPR），即预测错误且实际分类为负的数量与所有负样本数量的比例。纵轴为 True Positive Rate，也叫真阳率（TPR），即预测正确且实际分类为正的数量与所有正样本的数量的比例。其中 $FPR = \frac{FP}{FP + TN}, TPR = \frac{TP}{TP + FN}$。

AUC = 1 是完美分类器，采用这个预测模型时，不管设定什么阈值都能得出完美预测。绝大多数预测的场合，不存在完美分类器；0.5 < AUC < 1，优于随机猜测，这个分类器（模型）妥善设定阈值的话，能有预测价值。

具体 AUC 值大小应根据具体数据和业务进行同类比较，不同的数据之间的 AUC 比较没有任何意义。

② KS

Kolmogorov-Smirnov 两样本检验法简称为 KS 检验法，基于经验分布函数的距离而构造，检验两个累积分布的区分度。传统上说 KS 值就是 KS 检验法的统计量。我们通过预测概率或者分数和真实的 label 就可以计算出 KS 值。

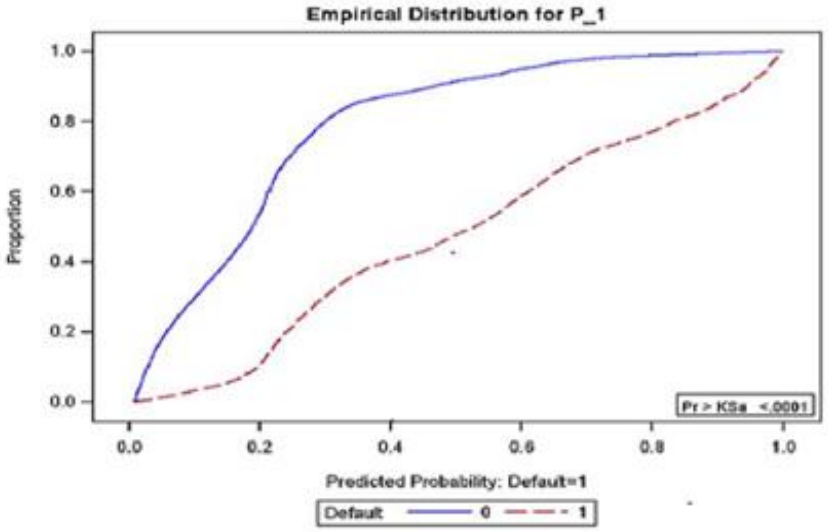


图 12 KS 原理示意图

通过 KS 原理示意图我们可以更加清晰地理解 KS 值是如何计算的。如图 12 所示，红色虚线为样本实际标签为 1 的累积分布，蓝色实线为实际标签为 0 的累积分布。我们先把标签为 0 的和标签为 1 的样本分开，然后将概率从 0-1 取等距的 100 份或者根据所有概率切成相应的份数，计算 0 和 1 的样本中小于等于该概率的样本占该类别总样本的比率即得到两类样本的累计分布，而 KS 即两样本累计分布差的最大值。

KS 的另一种解释为真阳率(TPR)与伪正率(FPR)随阀值变化差的最大值。

通常 KS 的模型区别能力如表 6 所示：

表 6　KS 模型区别能力表

KS 值	模型区别能力
<0.2	无区别能力
0.21-0.40	普通
0.41-0.50	好
0.51-0.60	很好
0.61-0.75	非常好
>0.75	模型可能有问题

③ f1-score

F1-score 实际上是召回率与精确率的加权平均值，因此本文依然借用表 5 混淆矩阵来介绍召回率与精确率，再介绍 F1-score 的计算方式。

召回率：即为在实际为正样本中预测为正确的正样本所占的比例，表示为 Recall（简记为 R）。计算方式为：$Recall = \frac{TP}{TP+FN}$。

精确率：即为在预测为负样本中预测为正确的正样本本所占的比例，表示为 Precision（简记为 P）。计算方式为：$Precision = \frac{TP}{TP+FP}$。

如图 13 所示，随着阈值的变化，召回率和精确率就像假设检验的两类错误一样不能同时提高。因此，我们需要一个指标来调和这两个指标，于是人们就常用 F1-score 来进行表示。

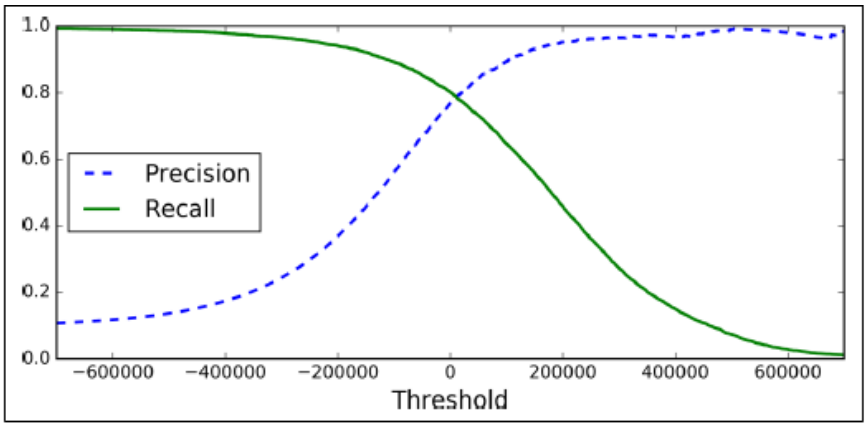


图 13 召回率与精确率图

其 F1-score 的计算方式为：$F1 = \frac{2 \times P \times R}{P + R}$

2. 模型结果及评价

① 评分卡

评价一张评分卡做得好坏与否，我们需要对其进行检验。而在评分卡模型中应用比较广泛的为频率分布，即观察每一组的坏样本占总体坏样本以及占组内坏样本是否单调，且分数越低其坏样本占比越高。其具体检验如表 7 所示：

表 7 评分卡频率分布表

	组内总样本		坏客户			
			组内样本		全部坏客户样本	
分数段	组内样本数	占比	坏样本数	占比	占比	累计占比
516-602	739	9.88%	152	20.57%	32.14%	100.00%
602-620	755	10.09%	91	12.05%	19.24%	67.86%
620-632	751	10.04%	57	7.59%	12.05%	48.63%
632-640	720	9.62%	39	5.42%	8.25%	36.58%
640-649	686	9.17%	39	5.69%	8.25%	28.33%
648-658	757	10.12%	25	3.30%	5.29%	20.08%
658-667	788	10.53%	25	3.17%	5.29%	14.80%
667-678	731	9.77%	23	3.15%	4.86%	9.51%
678-694	783	10.46%	13	1.66%	2.75%	4.65%
694-792	773	10.33%	9	1.16%	1.90%	1.90%
总计	7483	100.00%	473	6.32%	100.00%	100.00%

为了生动形象地表现坏样本占组内样本的比率是否单调，本文画了 Lift 图，如图 14 所示：

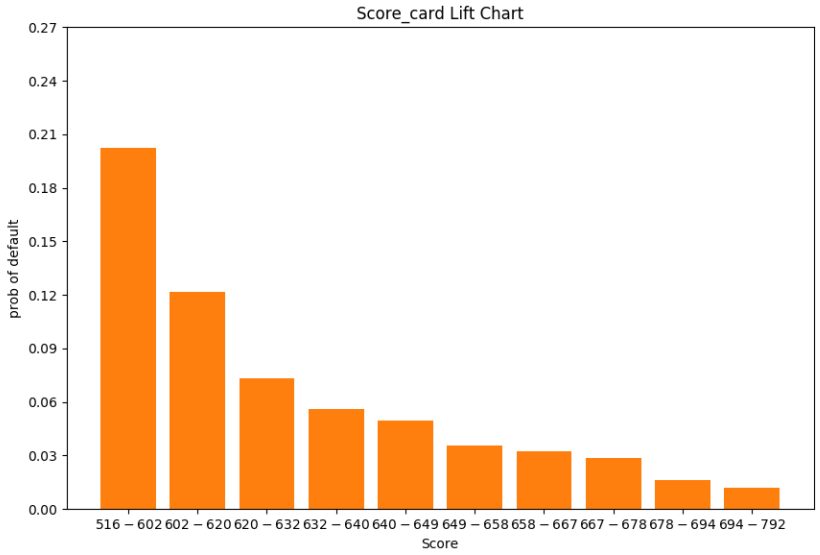


图 14 评分卡 Lift 图

实际上，在实际互联网借贷业务中，逻辑回归即可给出单个借贷者是否违约的预测结果。但是，评分卡作为基于逻辑回归的推广，可以给出每个用户具体的信用分数。因此，在给定分数阈值后，不仅可以判别是否放贷，还可以根据得分多少给出放贷额度。基于此出发点，我们将评分卡单独列出结果。

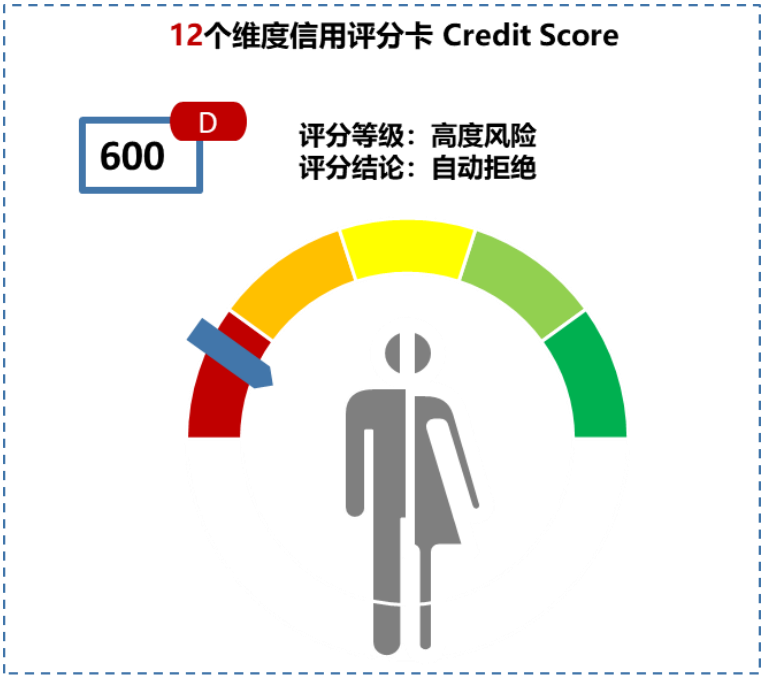
通过 SAS 编程选取 IV 值较大的特征进行聚类，每个聚类类别选取最大 IV 值的特征进入最终模型，建立了信用评分卡如表 8 所示。

表 8 本案例信用评分卡

基础分 632					
AC_NCA_O12_CNT_SUM（开卡大于一年未销户卡数）					
区间	0	1	2-3	4	>4
分数	-18	-10	2	10	21
HOUSE_LOAN_COUNT（房贷数量）					
区间	0	1	>2		
分数	1	-8	-17		
LOAN_BUS_CNT_SUM(经营性贷款笔数)					
区间	<10	10-21	>22		
分数	0	35	37		
LOAN_PREPAY_PART_CNT(部分提前还款贷款笔数)					
区间	0	1	2	3	>4
分数	-1	19	5	18	28
NO_QRORG_LOAN_3M(最近三个月贷款查询机构数)					
区间	0	1	2	3-4	5
分数	10	3	0	-5	-15
NO_QRORG_LOAN_6M(最近六个月贷款查询机构数					
区间	0	1	3-5	6-8	>9
分数	28	13	-3	-22	-42
CREDIT_LIMIT_CC(贷记卡额度总额)					
区间	<23440	23440-132827	132827-207055	202755-363323	>353323
分数	-6	3	7	14	21
CREDIT_LIMIT_LN(贷款合同金额总额)					
区间	<71184	71184-108512	108512-437519	437519-649334	>649334
分数	-3	14	16	26	35
LATEST_6M_USED_AVG_AMOUNT_LN(最近 6 个月贷款平均使用额度)					
区间	<1456	1456-6221	6221-10854	10854-13104	>13104
分数	-3	6	9	7	9

EDU_LEVEL(学历)					
分类	本科	高中	缺失	硕士及以上	专科及以下
分数	4	0	-1	16	-1
SALARY1（收入）					
分类	S1	S2	S3	S4	缺失
分数	-29	-17	-2	4	8
Agent(获客来源)					
分类	A1	A2	A3	A4	A5
分数	-12	-1	4	-7	-7

根据上述评分卡模型，信用评分卡的基础分为 632 分，各个样本的总分通过基础分再加上各特征的分数得到。由此，可以根据借贷人的信用评分以评判其信用状况，从而决定是否给予贷款。如果按照组内坏样本占比对于表 7 的分数段每两组设定为一个信用等级，还可以得到用户对应的评级。



如某用户根据 632 的基础分加其在各个特征上的得分后为 600 分，则其评价等级为高度风险，系统自动拒绝贷款请求。

② **模型评价**

通过特征工程，我们将 Train 和 Test 数据衍变为 13 份数据，数据分别对应于不同的缺失值填补方式、smote 比例。然后通过 Python 编程计算出每一份数据下的 AUC 值从而选出对应于最大 AUC 值的数据集，如表 9 所示：

表 9 最优数据集汇总表

算法	最优 AUC	缺失填补	SMOTE
Logistic/评分卡	0.737	0	无
GBDT	0.763	0	无
XGBoost	0.777	0	无
LGBM	0.769	0	1:5
Random Forest	0.757	-9	1:5
Stacking	0.748		

为了模型评价的完整性，本文又根据表 9 所得到的最优数据集计算出其 KS 及 F1 SCORE，为不同决策目标和决策偏好下的决策者提供参考依据。汇总后的模型评价指标表如表 10 所示：

表 10 最优模型结果汇总表

	模型名称	AUC	KS	F1 SCORE
传统统计模型	逻辑回归+评分卡	0.737	0.360	0.470
集成学习模型	GBDT	0.763	0.387	0.499
	XGBoost	0.777	0.426	0.491
	LGBM	0.769	0.404	0.493
	Random Forest	0.757	0.392	0.484
	模型 stacking	0.748	0.378	0.512

从表 10 可知，AUC 值及 KS 值最高的模型均为 XGBoost，而依据 F1 Score 进行评判，最好的模型则为 Stacking 后的模型。为了更生动地展现模型表现，我们将不同模型的三种评价指标曲线进行了可视化，如图 15、图 16 及图 17 所示。

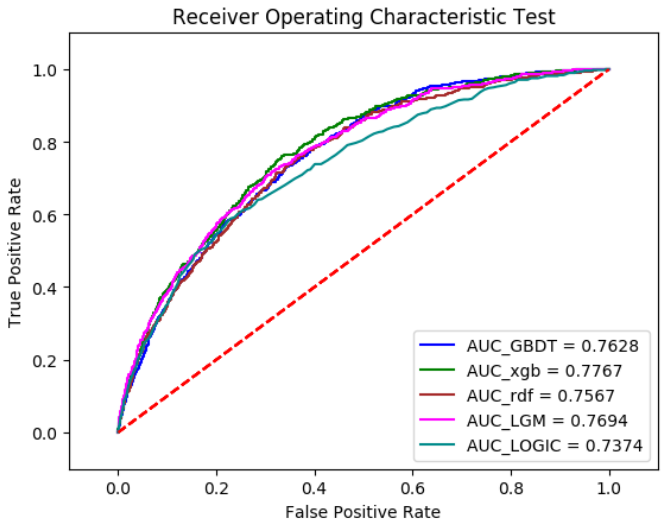


图 15 模型 AUC 横向比较图

AUC 即 ROC 曲线下的面积，图 15 表明：XGBoost 的 ROC 曲线下面积最大，即其 AUC 值也是最大的，为 0.7767。

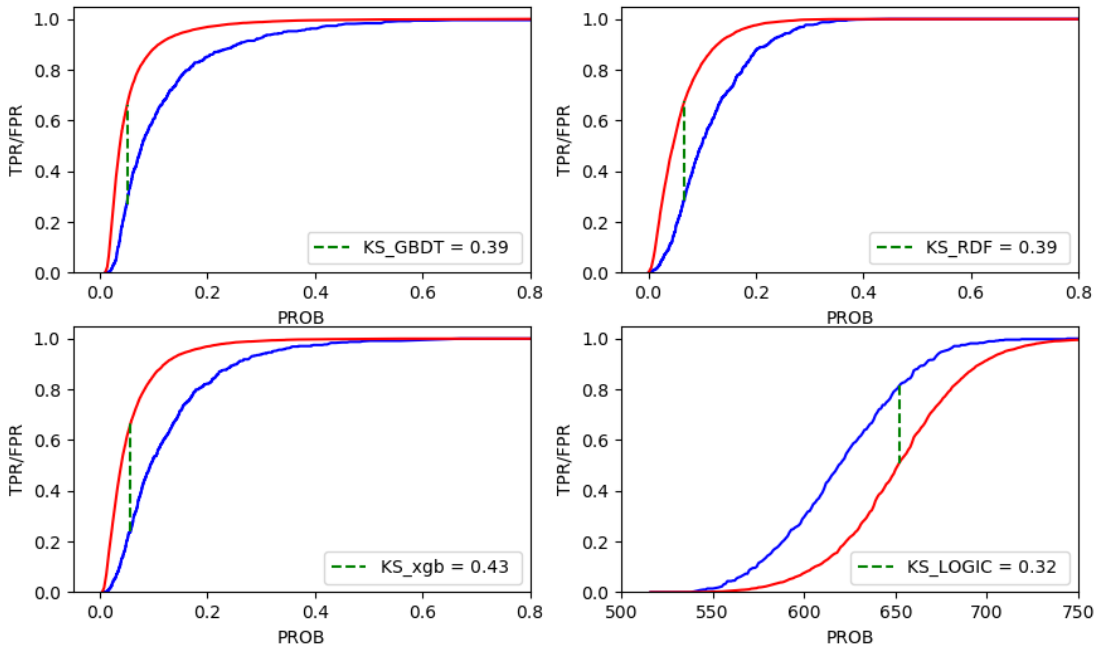


图 16 模型 KS 横向比较图

KS 值即真正率和假正率累计分布的最大差值，图 16 表明：若依据 KS 值为评价指标，XGBoost 有最好的模型表现。

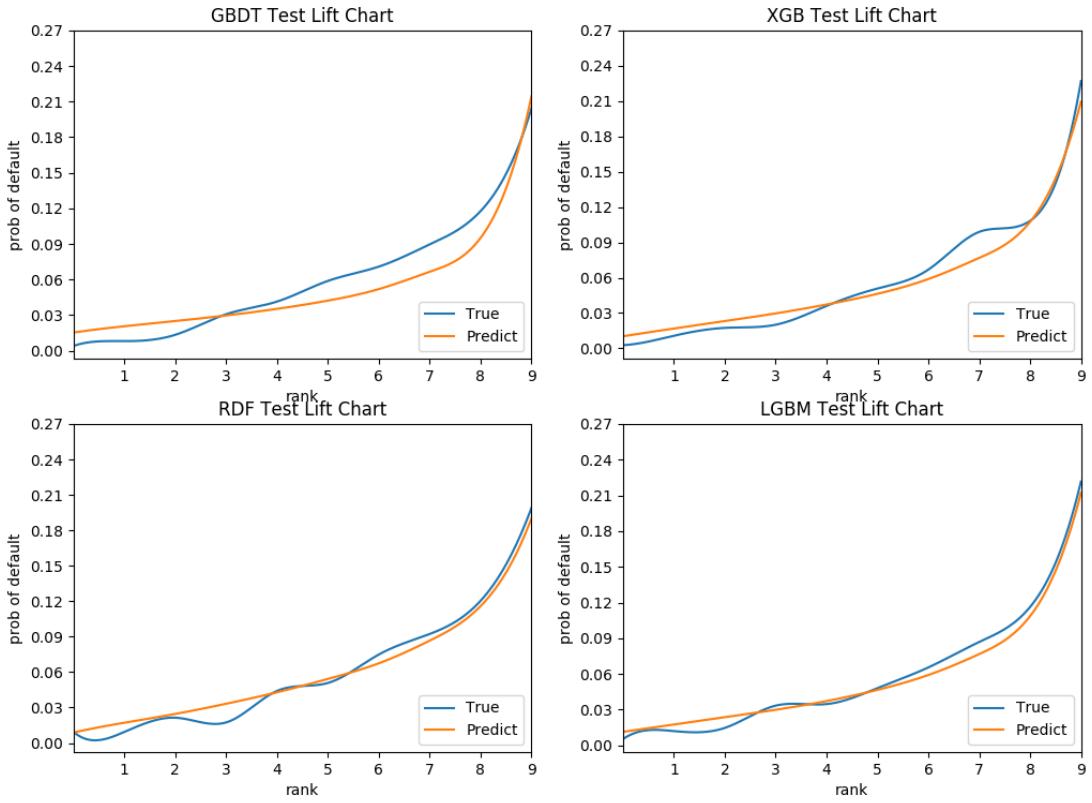


图 17 模型 Lift 横向比较图

模型提升图反映了与随机选择相比运用模型的效果提升。提升曲线左下角和右上角的垂直距离越大，说明模型对于好/坏用户的区分越明显。同时，真实和预测的两条线距离越近则模型拟合效果越好。图 17 表明，XGBoost 和 LGBM 基于随机选择的模型提升较突出。

根据所求出的各个模型的指标值，又加入模型的可解释性和响应速度来评判传统统计模型、集成学习模型以及模型 Stacking 的优异。对此，本文将用高、中、低三种程度进行度量，具体评判如表 11 所示：

表 11 模型评价表

	模型名称	准确性	可解释性	预测性	响应速度
传统统计模型	逻辑回归+评分卡	低	高	中	快
集成学习模型	GBDT	中	低	中	慢
	XGBoost	高	低	高	慢
	LGBM	高	低	高	中
	Random Forest	低	低	低	中
	模型 stacking	中	低	高	慢

综上所述：

从预测准确的角度出发，我们认为 XGBoost 模型在多方面的性能上更加优越，最终选择 XGBoost 模型来对 Test 数据集进行预测。

但从模型的可解释性和响应速度来看，经典的基于逻辑回归的评分卡模型更加适合强监管要求的银行和 IT 架构能力较弱的互联网企业。

企业的风控官可以根据表 11 的汇总结果基于不同的决策偏好进行选择。

(四) 预测结果

根据六-（三）所得出的结论，本文将采用 XGBoost 模型对官方给定的 Test 集进行预测。选取的对应数据集为用 0 填补缺失值、不进行 smote 操作的数据集。

我们根据上述最优算法，利用 Python 编程计算出 Test 集各个 ID 对应的预测结果如表 12 所示。此处仅列出部分结果，全部 ID 对应违约预测结果见附件。

表 12 预测结果表

ID	预测概率	预测值
86974	0.196149006	0
215505	0.077890366	0
323370	0.242814735	0
339404	0.029220032	0
456711	0.118882343	0
471856	0.044119358	0
525937	0.046847727	0
641138	0.068195231	0
662570	0.537365854	1
…	…	…
9206459	0.028659698	0
9232320	0.023501797	0

七、 结论和建议

(一) 创新点和局限性

创新点：本文在进行建模之前进行了大量的特征工程的工作，尤其是变量衍生方面，我们衍生出 500 多个具有实际业务含义的新变量，为模型建立提供了丰富的原始数据基础。又通过缺失值处理、连续变量离散化等操作加强了数据的规

范性和可读性，降低了过拟合的风险，并且提升了模型的效果。在进行逻辑回归时，并非单纯选定 IV 值高的变量构建评分卡，而是对变量聚类后并在同类型的变量中选取信息贡献最大的。在模型方面，本文不仅使用了传统的逻辑回归和评分卡模型，而且使用了多个集成学习模型，并在参数训练时采取了十折交叉验证。最后，本文还使用了模型 stacking 的方法来进行预测比较。

局限性：由于作者自身电脑设备运行速度不够快，本文没有考虑各个特征的交叉项，这可能会造成模型的最终效果有所减弱。

(二) 展望

根据题目所给出的数据以及条件，本文认为还可以使用半监督模型来进行模型训练和预测，这样可能会在最优的 Test 数据集上有更优异的效果。再者，根据本文所提出的局限性，我们认为可以用深度学习的工具去解决。即使交叉项再多一些，深度学习的运行效率仍然足够，以解决我们自身电脑设备运行速度不够的缺点。

参考文献

[1] Llew Mason, Jonathan Baxter, Peter Bartlett, Marcus Frean, Boosting Algorithms as Gradient Descent, [J].1999

[2] Si Si, Huan Zhang, S.Sathiya Keerthi, Dhruv Mahajan, Inderjit S.Dhilon, Cho-Jui Hsieh, Gredient Boosted Decision Trees for High Dimensional Sparse Output, [J].2017

[3] Jaesub Yun, Jihyun Ha, Jong-Seok Lee, Automatic Determination of Neighborhood Size in SMOTE, [J].2016

[4] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer, SMOTE:Synthetic Minority Over-sampling Technique, [J].2002

[5] Guolin Ke, Qi Meng ,Thomas Finley, Taifeng Wang: Lightgbm A Highly Efficient Gradient Boosting Decision Tree, Microsoft Research,2017

[6] Microsoft Corporation: Lightgbm Document ,May 07,2017

[7] Chen, Tianqi: "XGBoost: A Scalable Tree Boosting System" ,Carlos Guestrin (2016).

[8] 丁伟, 刘星海, 韩涵, 基于大数据技术的手机用户画像与征信研究[J]. 邮电设计艺术, 2016(6)

[9] 沈金波, 用户画像在互联网金融中的应用[J], 现代商业, 2017

[10] 王重仁, 韩东梅, 基于社交网络分析和 XGBoost 算法的互联网客户流失预测研究, 微型机与应用[J], 2017.12

[11] 卡巴科弗, 高涛, 肖楠, 陈钢, R 语言实战, 2013.01

附录

本次比赛所使用代码 python1500ine、SAS2000line，以下为部分重要 CODE，其余代码可参阅附件的 CODE。衍生后数据集同样因为过大，以附件形式给出。