

“东证期货杯”
全国大学生统计建模大赛
初赛论文



论文名称 互联网金融时代下信用评分体系模型构建

学 校 重庆理工大学

参赛人员 林中飞、赵赞豪、叶 涵、龚乃林

指导老师 赵胜利

日期范围 2017. 12. 20-2018. 02. 28

目录

摘要	1
一、问题描述	2
(一) 案例背景	2
(二) 问题重述	2
二、指标选择	3
(一) 原始数据的导入	3
(二) 变量的预处理	4
(三) 数据横向合并与追加	6
(四) 分类变量的标记(重编码)	8
(五) 缺失值的处理	10
(六) 变量的筛选	11
三、数据描述	12
(一) 单变量统计量	12
(二) 变量相关性判断	13
四、数据准备	15
(一) 数据标准化	15
(二) 降低基数维数	15
(三) 压缩观测	20
(四) 过度抽样	20
五、模型建立	21
(一) 逻辑回归	21
六、模型求解和检验	23
七、模型评价	27
(一) 模型估计	27
(二) 模型评估	28
八、模型结果分析	31
(一) 模型输出(测试集的验证)	31
(二) 策略咨询	33
(三) 将模型转化为评分卡	33
结论与建议	34
参考目录	35
附录	36

表格清单

表 1	训练集 contest_basic_train 中的每个变量的缺失比率.....	4
表 2	数据集 temp1 中缺失比率大的变量.....	6
表 3	数据集 tmp1 中缺失比率大的变量.....	7
表 4	描述统计量.....	12
表 5	变量解释.....	13
表 6	连续变量的相关系数.....	14
表 7	分类变量的相关系数.....	14
表 8	连续变量与分类变量的相关系数.....	15
附表 1	数据集观测变量统计表.....	41
附表 2	训练集 contest_basic_train 中的每个变量的缺失比率.....	42
附表 3	测试集 contest_basic_test 中的每个变量的缺失比率.....	42
附表 4	贷款 contest_ext_crd_cd_ln 中的每个变量的缺失比率.....	43
附表 5	贷款特殊交易 contest_ext_crd_cd_ln_spl 中的每个变量的缺失比率.....	43
附表 6	贷记卡 contest_ext_crd_cd_lnd 中的每个变量的缺失比率.....	44
附表 7	贷记卡透支记录 contest_ext_crd_cd_lnd_ovd 中的每个变量的缺失比...	44
附表 8	报告主表 contest_ext_crd_hd_report 中的每个变量的缺失比率.....	44
附表 9	信用提示 contest_ext_crd_is_creditcue 中的每个变量的缺失比率.....	45
附表 10	透支信息汇总 contest_ext_crd_is_ovdsummary 中的每个变量的缺失比 率.....	45
附表 11	未结清贷款 contest_ext_crd_is_sharedebt 中的每个变量的缺失比率...	45
附表 12	信贷审批查询记录明细 contest_ext_crd_qr_recorddtlinfo 中的每个变量 的缺失比率.....	45
附表 13	查询记录汇总 contest_ext_crd_qr_recordsmr 中的每个变量的缺失比 率.....	46
附表 14	反欺诈 contest_fraud 中的每个变量的缺失比率.....	46

插图清单

图 1	原始数据导入 SAS 系统.....	3
图 2	消除重复记录的 SAS 日志结果.....	5
图 3	分类变量的标记结果图.....	8
图 4	逻辑回归模型的变量数据集.....	22
图 5	逻辑回归建模总体介绍.....	23
图 6	逐步回归选择过程第 0 步和第 1 步.....	24
图 7	逐步回归选择过程第 1 步.....	24
图 8	逐步回归选择汇总报表图.....	25
图 9	参数估计报表图.....	25
图 10	优比估计和预测概率报表图.....	26
图 11	模型估计输出的数据集 case2.train_score_r.....	27
图 12	模型拟合统计量报表图.....	28
图 13	预测准确性报表图.....	29
图 14	模型的 ROC 曲线.....	30
图 15	模型的 KS 曲线.....	31
图 16	测试集目标变量的取值.....	32
图 17	测试集中违约客户个数.....	32
图 18	评分量化结果.....	33

互联网金融时代下信用评分体系模型的构建

摘要

时下，互联网金融已蓬勃兴起，但是也存在用户信用风险和欺诈等问题，那么互联网金融时代下信用评分体系模型构建就显得尤为重要。

本文运用因子分析、逻辑回归等统计方法，对原始数据做了以下步骤的处理：数据导入（业务认识→数据探索→变量选取）、数据清洗（数据集合并→缺失值的处理→衍生变量的产生→变量属性探究→数据抽样处理）、特征工程（降低维数→分箱处理→证据权重转换）、检验和建立模型（变量的相关性和共线性检验→模型的选择→业务约束→模型拟合→模型区分→逻辑回归模型的建立）、模型评估（ROC 曲线→KS 检验）、模型的部署与监控（模型的输出→策略咨询）。

在数据的导入阶段，选取合适的建模变量，详见正文表 5；在数据的清洗阶段，选取抽样样本进行建模处理；在特征工程阶段，运用因子分析方法降低维数；在变量的检验阶段，应用相关性和共线性检验建模变量，紧接着，在模型建立阶段，得出逻辑回归的表达式，具体如下：

$$p = \frac{T}{1 + T}$$

其中， $T = e^{f(x)}$,

$$\begin{aligned} f(b) = & -1.895 - 0.526 * x_1 + 0.2866 * x_3 - 0.0809 * x_6 - 0.1095 * x_8 \\ & - 0.6236 * x_{10} + 0.1861 * x_{11} - 0.1338 * x_{14} - 0.1933 * x_{15}。 \end{aligned}$$

在模型的评估阶段，用 ROC 曲线和 KS 检验评判模型的优劣；在模型的部署与监控阶段中，给出测试集的模型输出和违约客户的筛选情况，预测出客户在开户一定时期内违约的风险概率，有效排除信用不良客户和非目标客户申请。

综上所述，本次评分卡的开发的所有工作已经完成，待后续数据更新后再对评分卡进行调整与优化。

关键词：信用评分；违约预测；反欺诈；风险管理

一、问题描述

（一）案例背景

信用评分是银行三大风险模型之一，通过客户申请时填写以及通过其他渠道查询到的信息，预测将来发生违约、逾期、坏账等的统计概率，例如信用卡公司决定是否向客户发卡、银行决定是否允许信贷审批。

然而，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，在互联网上为有融资需求的客户提供更快捷的金融服务。通过互联网技术，加快业务处理速度，带给用户更好的服务体验。

但是，融资需求的客户对于企业来说一般多数是新客户，互联网金融机构或者企业尚不掌握客户的第一手信息，而仅仅依靠客户申请时填写的信息相对有限，同时也存在着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，引进外部数据、开展第三方机构合作、采购大数据征信与反欺诈产品等成为传统金融产业的刚需，转而促成了大数据征信与相关数据产品的蓬勃发展。

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，能够提高操作的效率，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。因此，建立精准的信用评分体系对于企业有着重要的意义。

（二）问题重述

本文要解决的问题如下：

1. 基于给定数据，运用数据挖掘等方法，构造模型变量；
2. 制定信用规则，建立信用评估模型，预测违约情况；

对于问题一，要根据数据挖掘等方法，提取出构建模型的变量；对于问题二，要建立信用评分模型，然后预测违约情况。

二、指标选择

(一) 原始数据的导入

对于给定的数据：训练集 contest_basic_train、测试集 contest_basic_test、贷款 contest_ext_crd_cd_ln、贷款特殊交易 contest_ext_crd_cd_ln_spl、贷记卡 contest_ext_crd_cd_lnd、贷记卡透支记录 contest_ext_crd_cd_lnd_ovd、报告主表 contest_ext_crd_hd_report、信用提示 contest_ext_crd_is_creditcue、透支信息汇总 contest_ext_crd_is_ovdsummary、未结清贷款 contest_ext_crd_is_sharedebt、信用审批查询记录明细 contest_ext_crd_qr_recorddtinfo、查询记录汇总 contest_ext_crd_qr_recordsmr、反欺诈 contest_fraud，在 SAS 系统中，建立永久逻辑库 case2，然后将给定数据导入到 SAS 系统中。得到结果如下图 1。

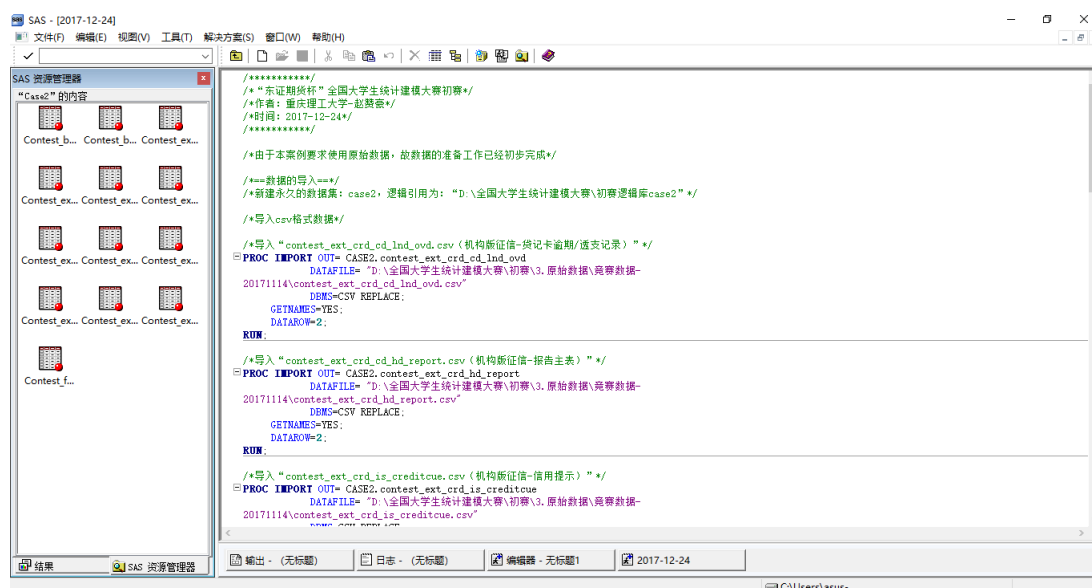


图 1 原始数据导入 SAS 系统

给定数据的导入的 SAS 程序代码详见附录一，然后我们可以比较原始格式的数据和 SAS 数据集中的数据，得到观测和变量是一致的。具体的数据集观测变量统计表见附录二。附录二中给出了每一个给定数据集中的观测数和变量数。

但是，由于实际业务中，软硬件问题可能导致数据存储失败，造成数据缺失；同时，人工录入数据也会有失误的情况，也会造成数据缺失。那么，对数据缺失情况的把握是非常有必要的。

所以，接下来就要对数据进行预处理。

（二）变量的预处理

变量的预处理一般包括两块内容：一是缺失值的处理，而是极值的处理。

1. 缺失值的处理

① 了解缺失变量和缺失情况

对于缺失值，首先要了解总体情况，哪些变量存在缺失，缺失率是多少。一般情况下，缺失率超过了 20% 的变量在分析中应该考虑予以剔除。

将给定的数据：训练集 `contest_basic_train`、测试集 `contest_basic_test`、贷款 `contest_ext_crd_cd_ln`、贷款特殊交易 `contest_ext_crd_cd_ln_spl`、贷记卡 `contest_ext_crd_cd_lnd`、贷记卡透支记录 `contest_ext_crd_cd_lnd_ovd`、报告主表 `contest_ext_crd_hd_report`、信用提示 `contest_ext_crd_is_creditcue`、透支信息汇总 `contest_ext_crd_is_ovdsummary`、未结清贷款 `contest_ext_crd_is_sharedebt`、信贷审批查询记录明细 `contest_ext_crd_qr_recorddtlinfo`、查询记录汇总 `contest_ext_crd_qr_recordsmr`、反欺诈 `contest_fraud` 中的每个数据表格的每个变量的缺失比率用 SAS 系统计算求出。训练集 `contest_basic_train` 中的每个变量的缺失比率详见表 1，所有的给定数据的每个数据表格的每个变量的缺失比率详见附录三，求解缺失比率的 SAS 代码详见附录四。

表 1 训练集 `contest_basic_train` 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_AGENT	0.7016
2	mr_EDU_LEVEL	0.1019333333
3	mr_HAS_FUND	0.0000666667
4	mr_ID_CARD	0
5	mr_IS_LOCAL	0
6	mr_LOAN_DATE	0
7	mr_MARRY_STATUS	0
8	mr_REPORT_ID	0
9	mr_SALARY	0.7045333333
10	mr_WORK_PROVINCE	0.0752666667
11	mr_Y	0

根据表 1，可以知道：训练集 `contest_basic_train` 中，报告 id `REPORT_ID`、婚姻状况 `MARRY_STATUS`、放款时间 `LOAN_DATE`、是否本地籍 `IS_LOCAL`、身份证 `ID_CARD`、Y 目标变量值（是否逾期）这 6 个变量没有缺失值；客户渠道 `AGENT`、教育 `EDU_LEVEL`、是否有公积金 `HAS_FUND`、收入 `SALARY`、工作城市 `WORK_PROVINCE` 这 5 个变量均有缺失值，缺失比率为：客户渠道 `AGENT`(0.70)、教育 `EDU_LEVEL`(0.10)、是否有公积金 `HAS_FUND`(0.00007)、收入 `SALARY` (0.70)、工作城市 `WORK_PROVINCE` (0.08)。

缺失值的产生，有系统的原因和人为原因两种。系统原因是指软硬件的问题导致数据存储失败，从而造成数据缺失；人为原因是指人的操作不当导致的数据缺失，比如数据录入错误，或者市场调查中被访人拒绝回答。

② 缺失值的类型

如何对缺失值进行预处理，取决于缺失值的类型。一般分为三类：

完全随机缺失，指数据的缺失是随机的，不依赖于任何已观察到的数据和未观测到的数据；随机缺失，指数据的缺失是随机的，即该类数据的缺失依赖于已观察到的数据，而不依赖于未观察到的数据；完全非随机缺失，指数据的缺失依赖于未观察到的数据。

③ 缺失值的处理

在了解完变量的缺失情况以及缺失原因和类型后，就要对缺失值进行处理。对于字符型变量，通常不做处理，而是单独为一类进行分析；对于数值型变量，方法有很多种，比如个案删除法、均值填补法和多重填补法。由于还没有形成数据宽表，所以缺失值的处理方法放到数据的横向合并与追加一节介绍。

2. 去除重复数据

数据在提取的过程中由于种种原因会出现数据的重复问题，大量重复的数据会对结果造成显著的影响，因此首先要找出重复记录，明确问题所在，进而消除重复记录或者观测。消除重复记录的结果如下图 2，代码详见附录五。

```
proc sort data=case2.contest_ext_crd_cd_ln_spl nodups;
by report_id;
run;

从数据集 CASE2.CONTEST_EXT_CRD_CD_LN_SPL. 读取了 67725 个观测
503 个重复观测已删除。
数据集 CASE2.CONTEST_EXT_CRD_CD_LN_SPL 有 67222 个观测和 6 个变量。
“PROCEDURE SORT” 所用时间（总处理时间）：
实际时间      0.22 秒
CPU 时间      0.01 秒

proc sort data=case2.contest_ext_crd_qr_recorddtlinfo nodups;
by report_id;
run;

从数据集 CASE2.CONTEST_EXT_CRD_QR_RECORDDTLINFO. 读取了 654329 个观测
1744 个重复观测已删除。
数据集 CASE2.CONTEST_EXT_CRD_QR_RECORDDTLINFO 有 652585 个观测和 4 个变量。
“PROCEDURE SORT” 所用时间（总处理时间）：
实际时间      0.88 秒
CPU 时间      0.39 秒
```

图 2 消除重复记录的 SAS 日志结果

从上述的日志结果来看，贷款特殊交易 contest_ext_crd_cd_ln_spl 数据集中消除了 503 个重复观测，信贷审批查询记录明细 contest_ext_crd_qr_recorddtlinfo

数据集中删除了 1744 个观测。图 2 仅仅只是一部分删除的日志结果，具体情况需运行程序，然后再在日志中观察。

（三）数据横向合并与追加

1. 数据的横向合并与追加

对于数据的横向合并与追加，首先要将关联变量的变量属性变成同样的。比如，我们将贷记卡透支记录 `contest_ext_crd_cd_lnd_ovd` 数据集、信用提示 `contest_ext_crd_is_creditcue` 数据集、透支信息汇总 `contest_ext_crd_is_ovdsummary` 数据集、未结清贷款 `contest_ext_crd_is_sharedebt` 数据集中的 `report_id` 全部变为数值型，然后再进行数据的横向合并。将关联变量的变量属性统一化的 SAS 程序详见附录六。

数据横向合并时，不改变变量，但是根据 `report_id` 的不同划分为数据宽表训练集和数据宽表测试集。经过删除对应观测，得到数据宽表训练集 `train1` 和数据宽表测试集 `test1`，然后再消除重复观测，以及求解出两个数据宽表的变量缺失比率。横向合并以及求解数据宽表变量缺失比率的 SAS 程序详见附录七。

得到数据宽表变量缺失比率大于或者等于 20% 的变量构成的数据集 `temp1` 和数据集 `tmp1`，结果如下表 2 和表 3。数据集 `temp1` 对应数据宽表训练集，数据集 `tmp1` 对应数据宽表测试集。

表 2 数据集 `temp1` 中缺失比率大的变量

	以前的变量名	bilv	beizhu
1	mr_AGENT	0.7393861667	建议删
2	mr_AMOUNT	0.6945355164	建议删
3	mr_COUNT_DW	0.2915716441	建议删
4	mr_HIGHEST_OA_PER_MON	0.2915716441	建议删
5	mr_LAST_MONTHS	0.6945355164	建议删
6	mr_MAX_DURATION	0.2915716441	建议删
7	mr_MONTHS	0.2915716441	建议删
8	mr_MONTH_DW	0.5067476417	建议删
9	mr_SALARY	0.7428676937	建议删
10	mr_balance	0.2706675117	建议删
11	mr_changing_amount	0.5959266301	建议删
12	mr_changing_months	0.5959266301	建议删
13	mr_content	0.5959266301	建议删
14	mr_get_time	0.5959266301	建议删
15	mr_payment_cyc	0.2499962573	建议删
16	mr_sum_dw	1	建议删
17	mr_type_id	1	建议删

表 3 数据集 tmp1 中缺失比率大的变量

	以前的变量名	bilv	beizhu
1	mr_AGENT	0.5127825104	建议删
2	mr_AMOUNT	0.7768117378	建议删
3	mr_COUNT_DW	0.3800155868	建议删
4	mr_EDU_LEVEL	0.2264993265	建议删
5	mr_HIGHEST_OA_PER_MON	0.3800155868	建议删
6	mr_LAST_MONTHS	0.7768117378	建议删
7	mr_MAX_DURATION	0.3800155868	建议删
8	mr_MONTHS	0.3800155868	建议删
9	mr_MONTH_DW	0.5965817202	建议删
10	mr_SALARY	0.5526098845	建议删
11	mr_changing_amount	0.4568548042	建议删
12	mr_changing_months	0.4568548042	建议删
13	mr_content	0.4568548042	建议删
14	mr_get_time	0.4568548042	建议删
15	mr_payment_cyc	0.2205529252	建议删
16	mr_sum_dw	0.9904767482	建议删
17	mr_type_id	0.9904767482	建议删

综合上述的表 2 和表 3, 可以将要删除的变量罗列如下: 客户渠道 AGENT、逾期/透支金额 AMOUNT、贷款逾期笔数 COUNT_DW、贷款单月最高逾期总额 HIGHEST_OA_PER_MON、逾期/透支月数 LAST_MONTHS、最大贷款时长 MAX_DURATION、贷款逾期月份数 MONTHS、逾期/透支月份 MONTH_DW、收入 SALARY、发生金额 changing_amount、变更月数 changing_months、处罚内容 content、信息更新日期 get_time、还款期数 payment_cyc、查询次数 sum_dw、查询类别 type_id。然后, 接着进行缺失率低的变量的缺失值的处理。

2. 数据宽表中缺失值的处理准备

对于字符型变量, 通常不做处理, 而是单独作为一类进行分析; 对于数值型变量, 方法有很多种。例如: 列表删除法、个案删除法、人工填补法、均值填补法、回归填补法、热平台填补法、冷平台填补法、极大似然估计法、期望最大化法、K 最近距离邻法、C4.5 方法、随机回归填补法 (PMM 法)、趋势得分法、马尔可夫链蒙特卡罗法 (MCMC 法)、贝叶斯网络、人工神经网络。

比较常用的有个案删除法、均值填补法以及 MCMC 法。

① 个案删除法

对于任何存在缺失值的变量, 直接删除缺失记录。在缺失值占比非常小的情况下, 这是一种非常简单有效的方法。然而, 该方法以减少样本量来换取数据的完整性, 有很大的局限性。例如, 有时候会丢失隐藏在缺失值当中的信息; 当变量有成百上千时, 因每个变量的少数缺失值而删除掉的记录, 累加起来可能非常

的可观。因此，使用此方法需要特别慎重。

② 均值填补法

针对数值型的变量，使用平均值来填充该变量的缺失值。使用该方法对缺失值进行填补时，不会影响变量的集中趋势，但是会造成变量的方差和标准差变小，给强调数据离散趋势的分析造成影响。

③ 多重填补法

多重填补法由 Rubin 于 1977 年首次提出，现在已经形成了一个比较系统的理论。多重填补法是一种单一的填补方法，不同之处在于对每一个缺失值用多个可能的值填补，以反应缺失值的不确定性，形成多个完整的数据集，然后用针对完整数据集的统计方法来对每一个填补数据集分别进行统计分析，综合形成最后的结果。

需要注意的是，没有哪一种方法是普遍适用的，每种方法都有优缺点，需要结合具体业务和实际情况进行使用。有时候，不需要处理缺失值，就可以建模，因为模型本身对缺失值就有一定的处理能力，比如贝叶斯网络和神经网络。

（四）分类变量的标记（重编码）

我们将数据集中的分类变量进行标记，然后再进行数据的清洗。将分类变量的筛选结果如下图 3。

	REPORT_ID	IS_LOCAL	WORK_PROVINCE	EDU_LEVEL	MARRY_STATUS	HAS_FUND	Y
1	8787	1	320000	4	1	0	0
2	8787	1	320000	4	1	0	0
3	9410	1	120000	9	1	1	0
4	9410	1	120000	9	1	1	0
5	9410	1	120000	9	1	1	0
6	9410	1	120000	9	1	1	0
7	9410	1	120000	9	1	1	0
8	9410	1	120000	9	1	1	0
9	9410	1	120000	9	1	1	0
10	9410	1	120000	9	1	1	0
11	9410	1	120000	9	1	1	0
12	9410	1	120000	9	1	1	0
13	9410	1	120000	9	1	1	0
14	9410	1	120000	9	1	1	0
15	9410	1	120000	9	1	1	0
16	9410	1	120000	9	1	1	0
17	9410	1	120000	9	1	1	0
18	9410	1	120000	9	1	1	0
19	9410	1	120000	9	1	1	0
20	9410	1	120000	9	1	1	0
21	9410	1	120000	9	1	1	0
22	22999	1	.	6	1	0	0
23	22999	1	.	6	1	0	0
24	22999	1	.	6	1	0	0

图 3 分类变量的标记结果图

有些 SAS 过程不能直接进行处理字符型变量，即使可以处理，过程也比较繁琐，因此在数据分析之前，需要进行编码转换，即重编码。

从上述的标记结果图，可以知道：工作城市和其他的连续变量是存在着缺失值的，下一步应该多缺失值进行填充。分类变量的标记如下表所示。

表 分类变量的标记

变量名及变量的意义	编码号	编码对应的意义
是否本地籍	1	本地籍
	0	非本地籍
教育	1	博士研究生
	2	硕士及以上
	3	硕士研究生
	4	本科
	5	专科
	6	专科以及以下
	7	高中
	8	初中
	9	缺省值
婚姻	1	已婚
	2	未婚
	3	离婚
	4	离异
	5	丧偶
	6	其他
账户状态	1	呆账
	2	冻结
	3	结清
	4	未激
	5	销户
	6	逾期
	7	正常
	8	止付
	9	缺省值
币种	1	澳大利
	2	港元
	3	加拿大
	4	美元
	5	欧元
	6	人民币
	7	日元
	8	瑞士法郎
	9	英镑

	10	缺省值
担保方式	1	保证
	2	抵押担保
	3	农户联保
	4	其他担保
	5	信用/免担保
	6	质押（含保证金）担保
	7	组合（不含保证）担保
	8	组合（含保证）担保
	9	缺省值
还款频率	1	按半年归
	2	按季归还
	3	按年归还
	4	按其他方
	5	按日归还
	6	不定期归
	7	一次性归
	8	一次性归
	9	按月归还
	10	缺省值
五级分类	1	正常
	2	未知
	3	关注
	4	缺省值
卡类型	1	贷记卡
	2	缺省值
查询原因	1	贷款审批
	2	担保资格
	3	信用卡审

上述表格中，可以知道一些分类变量已经被重编码处理。具体如表。

（五）缺失值的处理

下面开始正式的缺失值的处理，用人工填补法、均值填补法和中位数填补法。首先，去除缺失率高的变量 `balance`，`remain_payment_cyc`；然后，对变量 `work_province`，`used_credit_limit_amount`，`latest6_month_used_avg_amount` 这几个变量是进行的均值填补；紧接着，对变量 `edu_level`，`has_fund`，`state`，`currency`，`guarantee_type`，`class5_state`，`credit_limit_amount`，`scheduled_payment_amount`，`actual_payment_amount`，`curr_overdue_cyc`，`curr_overdue_amount`，`share_credit_limit_amount`，`HOUSE_LOAN_COUNT`，`COMMERCIAL_LOAN_COUNT`，

OTHER_LOAN_COUNT 这些变量进行人工填补法；再接着，对变量 used_highest_amount, LOANCARD_COUNT, STANDARD_LOANCARD_COUNT, FINANCE_CORP_COUNT, FINANCE_ORG_COUNT, ACCOUNT_COUNT, CREDIT_LIMIT, MAX_CREDIT_LIMIT_PER_ORG, MIN_CREDIT_LIMIT_PER_ORG, USED_CREDIT_LIMIT, LATEST_6M_USED_AVG_AMOUNT 这些变量进行中位数填补法。

经过一系列的缺失值的填补，最终得到了 34 个变量（包含 report_id 和目标变量 Y）。紧接着，就可以进行下一节的探索性数据分析。

（六）变量的筛选

从一开始，由关联矩阵的 78 个变量，变成现在的 34 个变量，在这个数据清洗的期间，是根据缺失率和其他变量与目标变量的联系强弱来反复循环筛选建模初步所需的变量。

第一个阶段是，根据 78 个变量的缺失率去除了 20 个缺失率高达 20% 的变量，得到 58 个变量；第二个阶段是，根据其他变量与目标变量的联系强弱，又去除了 24 个变量，然后得到 34 个变量。由此，可以进行探索性数据分析。具体可以参见表 5（除了 ID 和响应变量 Y 不含外，其余都包括）。

三、数据描述

（一）单变量统计量

经过一系列的数据清洗和指标选择的过程后，开始进行单变量的描述性统计。首先开始单变量的离散型统计量的统计。具体如下表 4。

表 4 描述统计量

	N	极小值	极大值	均值	标准差
b2	339187	0	1	.56	.496
b3	339187	110000	650000	312767.12	94571.455
b4	339187	1	9	5.70	1.613
b5	339187	1	6	1.43	.749
b6	339187	0	1	.44	.497
b7	339187	1	9	5.95	1.594
b8	339187	1	10	5.62	.972
b9	339187	1	9	4.95	.547
b10	339187	1	10	8.49	1.421
b11	339187	1	4	2.48	1.499
b12	339187	0	3120000	14816.76	41907.624
b13	339187	0	9999	506.86	1163.041
b14	339187	0	9999	930.98	1881.678
b15	339187	0	8646	.03	14.846
b16	339187	0	42891	31.93	1153.868
b17	339187	0	544325	9120.49	17327.100
b18	339187	0	99999	5952.30	9880.442
b19	339187	0	99995	5859.26	9682.966
b20	339187	0	99999	9821.04	13443.173
b21	339187	0	7	.19	.461
b22	339187	0	72	.01	.344
b23	339187	0	2017	20.36	61.463

b24	339187	0	2016	12.90	56.140
b25	339187	0	2016	.53	24.925
b26	339187	1	16	4.60	2.782
b27	339187	1	24	4.64	2.812
b28	339187	0	479500	50.39	2693.426
b29	339187	0	9824000	86998.51	112182.174
b30	339187	0	544325	32810.31	34760.976
b31	339187	0	370652	5047.41	8694.034
b32	339187	0	818635	54213.50	62513.684
b33	339187	0	835431	50092.62	62082.558
有效的 N (列表状态)	339187				

由表 4 描述性统计量表, 可知各个变量的极大值、极小值、均值和标准差。其中, 各个变量的含义由下表 5 中所示, 每个变量都对应着一个建模变量。

表 5 变量解释

b2 IS_LOCAL	b3 WORK_PROVINCE
b4 EDU_LEVEL	b5 MARRY_STATUS
b6 HAS_FUND	b7 state
b8 currency	b9 guarantee_type
b10 payment_rating	b11 class5_state
b12 credit_limit_amount	b13 scheduled_payment_amount
b14 actual_payment_amount	b15 curr_overdue_cyc
b16 curr_overdue_amount	b17 share_credit_limit_amount
b18 used_credit_limit_amount	b19 latest6_month_used_avg_amount
b20 used_highest_amount	b21 HOUSE_LOAN_COUNT
b22 COMMERCIAL_LOAN_COUNT	b23 OTHER_LOAN_COUNT
b24 LOANCARD_COUNT	b25 STANDARD_LOANCARD_COUNT
b26 FINANCE_CORP_COUNT	b27 FINANCE_ORG_COUNT
b28 ACCOUNT_COUNT	b29 CREDIT_LIMIT
b30 MAX_CREDIT_LIMIT_PER_ORG	b31 MIN_CREDIT_LIMIT_PER_ORG
b32 USED_CREDIT_LIMIT	b33 LATEST_6M_USED_AVG_AMOUNT

(二) 变量相关性判断

下面开始进行连续变量的相关性的检验，具体结果如下表 6 所示。

表 6 连续变量的相关系数

	b3	b12	b13	b14	b15	b16	b17	b18	b19	b20	b21	b22	b23	b24	b25	b26	b27	b28	b29	b30	b31	b32	b33
b3	1.00	0.01	0.01	0.00	0.00	0.00	0.02	0.02	0.02	0.02	0.01	0.00	0.01	0.00	-0.01	0.02	0.02	0.00	0.02	0.02	0.01	0.03	0.03
b12	0.01	1.00	0.17	0.09	0.00	-0.01	0.24	0.16	0.16	0.17	0.12	0.01	0.01	0.00	0.00	0.03	0.04	0.00	0.14	0.17	0.07	0.14	0.13
b13	0.01	0.17	1.00	0.58	0.00	0.05	0.40	0.52	0.52	0.52	0.04	0.02	-0.01	0.05	0.01	0.10	0.10	0.03	0.13	0.11	0.04	0.14	0.14
b14	0.00	0.09	0.58	1.00	0.00	-0.01	0.27	0.33	0.33	0.34	0.01	-0.01	-0.07	-0.01	-0.01	0.09	0.09	-0.01	0.08	0.05	-0.01	0.08	0.07
b15	0.00	0.00	0.00	0.00	1.00	0.06	0.01	0.01	0.00	0.00	0.00	0.10	0.06	0.06	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.00
b16	0.00	-0.01	0.05	-0.01	0.06	1.00	0.06	0.12	0.13	-0.02	-0.01	0.44	0.79	0.96	0.40	-0.01	0.01	0.53	0.00	-0.01	0.20	0.01	0.00
b17	0.02	0.24	0.40	0.27	0.01	0.06	1.00	0.58	0.59	0.59	0.07	0.03	0.05	0.05	0.01	0.06	0.06	0.03	0.21	0.27	0.15	0.19	0.19
b18	0.02	0.16	0.52	0.33	0.01	0.12	0.58	1.00	0.92	0.74	0.05	0.05	0.10	0.11	0.03	0.06	0.06	0.05	0.13	0.15	0.15	0.20	0.19
b19	0.02	0.16	0.52	0.33	0.00	0.13	0.59	0.92	1.00	0.74	0.06	0.06	0.11	0.12	0.06	0.06	0.06	0.06	0.14	0.16	0.16	0.20	0.20
b20	0.02	0.17	0.52	0.34	0.00	-0.02	0.59	0.74	0.74	1.00	0.07	-0.01	0.00	-0.02	-0.01	0.08	0.07	-0.01	0.16	0.18	0.11	0.19	0.19
b21	0.01	0.12	0.04	0.01	0.00	-0.01	0.07	0.05	0.06	0.07	1.00	0.00	0.05	0.01	0.00	0.08	0.09	-0.01	0.23	0.19	0.05	0.19	0.19
b22	0.00	0.01	0.02	-0.01	0.10	0.44	0.03	0.05	0.06	-0.01	0.00	1.00	0.49	0.54	0.32	0.00	0.02	0.50	0.01	0.00	0.13	0.01	0.00
b23	0.01	0.01	-0.01	-0.07	0.06	0.79	0.05	0.10	0.11	0.00	0.05	0.49	1.00	0.82	0.36	0.03	0.05	0.49	0.04	0.04	0.19	0.06	0.06
b24	0.00	0.00	0.05	-0.01	0.06	0.96	0.05	0.11	0.12	-0.02	0.01	0.54	0.82	1.00	0.44	0.07	0.09	0.56	0.06	0.03	0.19	0.07	0.06
b25	-0.01	0.00	0.01	-0.01	0.00	0.40	0.01	0.03	0.06	-0.01	0.00	0.32	0.36	0.44	1.00	0.00	0.01	0.33	0.00	-0.01	0.09	0.00	0.00
b26	0.02	0.03	0.10	0.09	0.00	-0.01	0.06	0.06	0.06	0.08	0.08	0.00	0.03	0.07	0.00	1.00	1.00	0.01	0.59	0.43	-0.20	0.66	0.67
b27	0.02	0.04	0.10	0.09	0.00	0.01	0.06	0.06	0.06	0.07	0.09	0.02	0.05	0.09	0.01	1.00	1.00	0.03	0.59	0.44	-0.20	0.66	0.67
b28	0.00	0.00	0.03	-0.01	0.01	0.53	0.03	0.05	0.06	-0.01	-0.01	0.50	0.49	0.56	0.33	0.01	0.03	1.00	0.00	0.00	0.18	0.01	0.00
b29	0.02	0.14	0.13	0.08	0.00	0.00	0.21	0.13	0.14	0.16	0.23	0.01	0.04	0.06	0.00	0.59	0.59	0.00	1.00	0.70	0.03	0.74	0.72
b30	0.02	0.17	0.11	0.05	0.00	-0.01	0.27	0.15	0.16	0.18	0.19	0.00	0.04	0.03	-0.01	0.43	0.44	0.00	0.70	1.00	0.18	0.73	0.72
b31	0.01	0.07	0.04	-0.01	0.02	0.20	0.15	0.15	0.16	0.11	0.05	0.13	0.19	0.19	0.09	-0.20	-0.20	0.18	0.03	0.18	1.00	0.04	0.04
b32	0.03	0.14	0.14	0.08	0.00	0.01	0.19	0.20	0.20	0.19	0.19	0.01	0.06	0.07	0.00	0.66	0.66	0.01	0.74	0.73	0.04	1.00	0.97
b33	0.03	0.13	0.14	0.07	0.00	0.00	0.19	0.19	0.20	0.19	0.19	0.00	0.06	0.06	0.00	0.67	0.67	0.00	0.72	0.72	0.04	0.97	1.00

由连续变量之间的相关系数表 6 可以看出，连续变量 b_{33} 与连续变量 b_{32} 之间的相关性比较强，相关系数 $r=0.97$ ；连续变量 b_{24} 与连续变量 b_{16} 之间也存在较强相关性， $r=0.96$ ；等等。因此，我们从表得出，很多连续变量变量之间的相关性不强，或者说相关性较弱。

下面进行分类变量的相关性检验，分类变量之间的相关系数如表 7 所示：

表 7 分类变量的相关系数

	b2	b4	b5	b6	b7	b8	b9	b10	b11
b2	1.00	0.06	-0.07	-0.04	0.01	0.01	-0.02	-0.03	0.00
b4	0.06	1.00	-0.05	0.07	0.02	0.01	-0.04	-0.02	0.03
b5	-0.07	-0.05	1.00	0.00	-0.01	0.01	0.04	0.04	-0.01
b6	-0.04	0.07	0.00	1.00	0.00	0.01	-0.01	-0.01	0.01
b7	0.01	0.02	-0.01	0.00	1.00	-0.12	0.16	0.00	-0.43
b8	0.01	0.01	0.01	0.01	-0.12	1.00	-0.03	0.00	0.03
b9	-0.02	-0.04	0.04	-0.01	0.16	-0.03	1.00	0.09	-0.08
b10	-0.03	-0.02	0.04	-0.01	0.00	0.00	0.09	1.00	0.03
b11	0.00	0.03	-0.01	0.01	-0.43	0.03	-0.08	0.03	1.00

由分类变量之间的相关系数表 7 可知，所有的分类变量之间的相关性不强或者较弱。

紧接着，进行连续变量和分类变量之间的相关性检验，具体结果如表 8：

表 8 连续变量与分类变量的相关系数

	b3	b13	b14	b15	b16	b17	b18	b19	b20	b21	b22	b23
b2	-0.03	0.00	0.01	0.00	0.01	0.00	-0.01	-0.01	0.00	-0.01	0.01	-0.02
b4	0.04	-0.01	0.03	0.00	0.02	0.00	-0.01	0.00	-0.01	-0.05	0.00	-0.01
b5	-0.02	-0.02	-0.03	0.00	-0.01	-0.03	-0.02	-0.03	-0.03	-0.01	-0.01	0.00
b6	0.01	-0.01	0.00	0.00	0.00	-0.01	0.00	0.01	0.00	-0.05	0.00	-0.01
b7	-0.02	0.29	0.33	0.00	0.02	0.06	0.07	0.05	0.05	-0.03	0.00	-0.16
b8	0.02	0.17	0.19	0.00	0.01	0.20	0.21	0.21	0.24	0.00	0.00	0.06
b9	-0.02	0.04	0.04	0.01	0.20	-0.04	-0.01	-0.01	-0.04	-0.03	0.11	0.15
b10	-0.01	-0.01	-0.01	-0.01	-0.12	-0.03	-0.03	-0.03	-0.01	0.01	-0.06	-0.08
b11	0.01	-0.08	-0.07	0.00	0.03	0.00	0.00	0.00	0.00	0.01	0.02	0.11
b12	0.01	0.17	0.09	0.00	-0.01	0.24	0.16	0.16	0.17	0.12	0.01	0.01

	b24	b25	b26	b27	b28	b29	b30	b31	b32	b33
b2	0.00	0.01	-0.03	-0.03	0.00	-0.02	-0.01	0.00	-0.02	-0.02
b4	0.01	0.00	-0.05	-0.05	0.01	-0.05	-0.02	0.00	-0.05	-0.06
b5	-0.02	0.00	-0.04	-0.04	-0.01	-0.05	-0.05	-0.01	-0.05	-0.05
b6	-0.01	0.00	-0.02	-0.02	0.00	-0.04	-0.03	-0.01	-0.03	-0.03
b7	0.02	0.01	0.08	0.08	0.01	0.03	0.00	-0.04	0.03	0.02
b8	-0.01	0.00	-0.04	-0.04	0.01	-0.04	-0.02	0.02	-0.03	-0.03
b9	0.21	0.09	0.02	0.02	0.12	-0.01	-0.03	0.01	-0.02	-0.02
b10	-0.12	-0.06	-0.02	-0.02	-0.07	-0.03	-0.03	-0.02	-0.03	-0.03
b11	0.03	0.01	-0.02	-0.02	0.02	-0.01	0.01	0.02	0.00	0.01
b12	0.00	0.00	0.03	0.04	0.00	0.14	0.17	0.07	0.14	0.13

从连续变量与分类变量的相关性表 8 可知，相关系数的最大值 $r=0.33$ ，最小值为 $r=-0.16$ ，所以，所有的连续变量与分类变量之间的相关性都不强。

四、数据准备

（一）数据标准化

将建模选定的数据进行标准化，具体的可以见附录十一。

（二）降低基数维数

1.因子分析理论思想

①基本思想

因子分析可以视为主成分分析的一种推广，它的基本思想是：根据相关性大小把变量分组，使得组内的变量相关性较高，但不同组的变量相关性较低，则每组变量可以代表一个基本结构，称为因子，它反映已经观测到的相关性。

②正交因子模型

设 p 维随机向量 $X = (x_1, x_2, \dots, x_p)^T$ 的期望为 $\mu = (\mu_1, \mu_2, \dots, \mu_p)^T$ ，

协方差矩阵为 Σ ，假定 X 线性地依赖于少数几个不可观测的随机变量 $f_1, f_2, \dots, f_m (m < p)$ 和 p 个附加的方差源 $\epsilon_1, \epsilon_2, \dots, \epsilon_p$ ，一般称 f_1, f_2, \dots, f_m 为公因子，称 $\epsilon_1, \epsilon_2, \dots, \epsilon_p$ 为特殊因子或误差。那么，因子模型为：

$$\begin{aligned} x_1 &= \mu_1 + a_{11}f_1 + a_{12}f_2 + \dots + a_{1m}f_m + \epsilon_1 \\ x_2 &= \mu_2 + a_{21}f_1 + a_{22}f_2 + \dots + a_{2m}f_m + \epsilon_2 \\ &\vdots \\ x_p &= \mu_p + a_{p1}f_1 + a_{p2}f_2 + \dots + a_{pm}f_m + \epsilon_p \end{aligned}$$

引入矩阵符号，记

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ & & \ddots & \\ a_{p1} & a_{p2} & \dots & a_{pm} \end{bmatrix}, \quad F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_p \end{bmatrix}$$

那么因子模型可以写为：

$$X = \mu + AF + \epsilon$$

式中， a_{ij} 称为第 i 个变量在第 j 个因子上的载荷，矩阵 A 称为载荷矩阵。

③因子得分

在因子分析中，虽然我们关心模型中载荷矩阵的估计和对公因子的解释，但对于公因子的估计即计算因子得分，有时也是需要的。

给定因子模型 $X = \mu + AF + \epsilon$ ，假定均值向量 μ 、载荷矩阵 A 和特殊方差阵 Φ 已知，把特殊因子 ϵ 看作误差，因为 $\text{Var}(\epsilon_i) = \phi_i (i = 1, 2, \dots, p)$ 未必相等，所以我们用加权最小二乘法估计公因子 F 。

首先将因子模型改写为：

$$X - \mu = AF + \epsilon$$

两边左乘 $\Phi^{-1/2}$ 得

$$\Phi^{-1/2}(X - \mu) = (\Phi^{-1/2}A)F + \Phi^{-1/2}\epsilon$$

记 $X^* = \Phi^{-1/2}(X - \mu)$, $A^* = \Phi^{-1/2}A$, $\epsilon^* = \Phi^{-1/2}\epsilon$, 则上式可以写成

$$X^* = A^*F + \epsilon^*$$

注意到 $E(\epsilon^*) = \Phi^{-1/2}E(\epsilon) = 0$, $\text{Cov}(\epsilon^*) = E(\epsilon^*\epsilon^{*T}) = \Phi^{-1/2}E(\epsilon\epsilon^T)\Phi^{-1/2} = I$, 所以上式是经典的回归模型, 由最小二乘法知 F 的估计为:

$$\begin{aligned}\hat{F} &= (A^{*T}A^*)^{-1}A^{*T}X^* = \left(A^T\Phi^{-1/2}\Phi^{-1/2}A\right)^{-1}A^T\Phi^{-1/2}\Phi^{-1/2}(X - \mu) \\ &= (A^T\Phi^{-1}A)^{-1}A^T\Phi^{-1}(X - \mu)\end{aligned}$$

实际中, A , Φ 和 μ 都是未知的, 通常用它们的某种估计来代替, 比如我们采用正交旋转后的载荷矩阵 A 的估计 \hat{A} , $\hat{\Phi} = \text{diag}(1 - \hat{h}_1^2, 1 - \hat{h}_2^2, \dots, 1 - \hat{h}_p^2)$ 和样本均值 $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$ 分别代替 A , Φ 和 μ , 于是可得对应于 x_j 的因子得分

$$\hat{f}_j = (\hat{A}^T\hat{\Phi}^{-1}\hat{A})^{-1}\hat{A}^T\hat{\Phi}^{-1}(x_j - \bar{X})$$

2. 因子分析实际操作

在变量的筛选之后, 得到 34 个变量 (包含 report_id 和目标变量 Y), 此时的个案数还有接近三十四万, 而且 report_id 是重复的, 我们需要对个案数进行压缩, 压缩到个案数为三万条, 即一个 report_id 对应一条数据。压缩个案数的同时还需要再一次进行变量压缩, 即降低基数维数, 方法有主成分分析和因子分析, 我们选择了因子分析的办法来处理。三十四万条数据, 每条数据对应的每个公因子上有一个因子得分, 同一个 report_id 对应的每个公因子上有多个, 我们去平均数来作为个案数压缩的方法, 至于变量的降维, 根据累计方差贡献率达到 80% 来选择, 因此选择了 15 个公因子。此时得到的数据进行后续实验操作。具体可见附录十二。

具体地因子得分表达式如下:

$$\begin{aligned}x_1 &= -0.02135 * b_2 + 0.03745 * b_3 - 0.05064 * b_4 - 0.07622 * b_5 - 0.03668 * b_6 + 0.12283 \\ &\quad * b_7 + 0.11572 * b_8 + 0.01013 * b_9 - 0.07147 * b_{10} - 0.02096 * b_{11} \\ &\quad + 0.25166 * b_{12} + 0.47238 * b_{13} + 0.33472 * b_{14} + 0.01715 * b_{15} + 0.19344 \\ &\quad * b_{16} + 0.52411 * b_{17} + 0.57466 * b_{18} + 0.5839 * b_{19} + 0.54733 * b_{20} \\ &\quad + 0.2238 * b_{21} + 0.13889 * b_{22} + 0.21674 * b_{23} + 0.25342 * b_{24} + 0.10214 \\ &\quad * b_{25} + 0.65701 * b_{26} + 0.6616 * b_{27} + 0.14426 * b_{28} + 0.72719 * b_{29} \\ &\quad + 0.69389 * b_{30} + 0.12029 * b_{31} + 0.81093 * b_{32} + 0.80608 * b_{33}\end{aligned}$$

$$\begin{aligned}
x_2 &= 0.01087 * b_2 - 0.00841 * b_3 + 0.03182 * b_4 + 0.01078 * b_5 + 0.00854 * b_6 - 0.03899 \\
&\quad * b_7 + 0.07271 * b_8 + 0.22963 * b_9 - 0.12031 * b_{10} + 0.05965 * b_{11} \\
&\quad - 0.03858 * b_{12} + 0.05086 * b_{13} - 0.01681 * b_{14} + 0.08188 * b_{15} + 0.86326 \\
&\quad * b_{16} + 0.06888 * b_{17} + 0.15623 * b_{18} + 0.17277 * b_{19} + 0.02579 * b_{20} \\
&\quad - 0.06002 * b_{21} + 0.63427 * b_{22} + 0.80288 * b_{23} + 0.86894 * b_{24} + 0.52604 \\
&\quad * b_{25} - 0.27314 * b_{26} - 0.25336 * b_{27} + 0.66877 * b_{28} - 0.24082 * b_{29} \\
&\quad - 0.21319 * b_{30} + 0.29546 * b_{31} - 0.25018 * b_{32} - 0.26001 * b_{33} \\
x_3 &= 0.02195 * b_2 + 0.00262 * b_3 + 0.04987 * b_4 + 0.00659 * b_5 + 0.02996 * b_6 + 0.14524 \\
&\quad * b_7 + 0.32557 * b_8 - 0.09179 * b_9 + 0.02609 * b_{10} - 0.06236 * b_{11} \\
&\quad + 0.15911 * b_{12} + 0.54951 * b_{13} + 0.45043 * b_{14} - 0.01956 * b_{15} - 0.19449 \\
&\quad * b_{16} + 0.51551 * b_{17} + 0.65065 * b_{18} + 0.64577 * b_{19} + 0.64491 * b_{20} \\
&\quad - 0.07012 * b_{21} - 0.17035 * b_{22} - 0.24316 * b_{23} - 0.25529 * b_{24} - 0.13704 \\
&\quad * b_{25} - 0.46334 * b_{26} - 0.46986 * b_{27} - 0.17595 * b_{28} - 0.34563 * b_{29} \\
&\quad - 0.26844 * b_{30} + 0.11112 * b_{31} - 0.37012 * b_{32} - 0.37253 * b_{33} \\
x_4 &= 0.00553 * b_2 - 0.05669 * b_3 - 0.01571 * b_4 + 0.03724 * b_5 + 0.0085 * b_6 + 0.75809 \\
&\quad * b_7 - 0.12892 * b_8 + 0.43491 * b_9 + 0.05414 * b_{10} - 0.57416 * b_{11} \\
&\quad - 0.27822 * b_{12} + 0.2956 * b_{13} + 0.40704 * b_{14} + 0.00409 * b_{15} + 0.05434 \\
&\quad * b_{16} - 0.14702 * b_{17} - 0.08262 * b_{18} - 0.0941 * b_{19} - 0.09281 * b_{20} \\
&\quad - 0.1898 * b_{21} + 0.04287 * b_{22} - 0.11766 * b_{23} + 0.06684 * b_{24} + 0.05338 \\
&\quad * b_{25} + 0.21023 * b_{26} + 0.20992 * b_{27} + 0.04852 * b_{28} - 0.06239 * b_{29} \\
&\quad - 0.18557 * b_{30} - 0.29071 * b_{31} - 0.05976 * b_{32} - 0.07337 * b_{33} \\
x_5 &= -0.16678 * b_2 + 0.01249 * b_3 - 0.09974 * b_4 + 0.16576 * b_5 + 0.08084 * b_6 \\
&\quad - 0.36093 * b_7 + 0.3474 * b_8 + 0.26331 * b_9 + 0.26509 * b_{10} + 0.40173 \\
&\quad * b_{11} - 0.45517 * b_{12} - 0.00098 * b_{13} + 0.01274 * b_{14} - 0.00777 * b_{15} \\
&\quad + 0.00091 * b_{16} - 0.03611 * b_{17} + 0.11904 * b_{18} + 0.11633 * b_{19} + 0.11766 \\
&\quad * b_{20} - 0.24673 * b_{21} - 0.03316 * b_{22} + 0.09313 * b_{23} + 0.00014 * b_{24} \\
&\quad - 0.02136 * b_{25} + 0.25673 * b_{26} + 0.2549 * b_{27} - 0.05068 * b_{28} - 0.09617 \\
&\quad * b_{29} - 0.21722 * b_{30} - 0.41019 * b_{31} - 0.04035 * b_{32} - 0.02635 * b_{33} \\
x_6 &= 0.24108 * b_2 + 0.1318 * b_3 + 0.41614 * b_4 - 0.31513 * b_5 + 0.22354 * b_6 - 0.0286 \\
&\quad * b_7 + 0.11649 * b_8 - 0.42898 * b_9 - 0.40377 * b_{10} + 0.12295 * b_{11} \\
&\quad + 0.25917 * b_{12} + 0.07962 * b_{13} + 0.16868 * b_{14} - 0.00314 * b_{15} + 0.03353 \\
&\quad * b_{16} - 0.05593 * b_{17} - 0.05096 * b_{18} - 0.05198 * b_{19} - 0.02866 * b_{20} \\
&\quad - 0.25009 * b_{21} + 0.06331 * b_{22} + 0.00383 * b_{23} + 0.03954 * b_{24} + 0.06285 \\
&\quad * b_{25} + 0.19487 * b_{26} + 0.19577 * b_{27} + 0.03889 * b_{28} - 0.08046 * b_{29} \\
&\quad - 0.16914 * b_{30} - 0.38576 * b_{31} - 0.06006 * b_{32} - 0.06177 * b_{33} \\
x_7 &= 0.29734 * b_2 + 0.13025 * b_3 + 0.52402 * b_4 - 0.34088 * b_5 + 0.31505 * b_6 + 0.01454 \\
&\quad * b_7 - 0.05591 * b_8 + 0.31834 * b_9 + 0.17285 * b_{10} + 0.04862 * b_{11} \\
&\quad - 0.35745 * b_{12} - 0.09662 * b_{13} - 0.07602 * b_{14} + 0.00008 * b_{15} - 0.00216 \\
&\quad * b_{16} + 0.02101 * b_{17} + 0.04528 * b_{18} + 0.04726 * b_{19} + 0.02757 * b_{20} \\
&\quad - 0.19565 * b_{21} - 0.06343 * b_{22} - 0.04183 * b_{23} - 0.02191 * b_{24} - 0.08097 \\
&\quad * b_{25} - 0.09573 * b_{26} - 0.0977 * b_{27} - 0.03189 * b_{28} + 0.04952 * b_{29} \\
&\quad + 0.16408 * b_{30} + 0.29968 * b_{31} + 0.0824 * b_{32} + 0.08068 * b_{33}
\end{aligned}$$

$$\begin{aligned}
x_8 = & -0.63671 * b_2 + 0.39282 * b_3 + 0.13151 * b_4 + 0.31946 * b_5 + 0.57073 * b_6 \\
& + 0.06614 * b_7 - 0.08169 * b_8 - 0.08487 * b_9 - 0.05989 * b_{10} - 0.11052 \\
& * b_{11} + 0.07425 * b_{12} - 0.02475 * b_{13} - 0.05366 * b_{14} + 0.06082 * b_{15} \\
& + 0.00286 * b_{16} + 0.0138 * b_{17} + 0.00723 * b_{18} + 0.01021 * b_{19} - 0.01415 \\
& * b_{20} - 0.09322 * b_{21} + 0.01246 * b_{22} - 0.00336 * b_{23} - 0.0029 * b_{24} \\
& - 0.02705 * b_{25} - 0.01865 * b_{26} - 0.01709 * b_{27} + 0.01131 * b_{28} + 0.00111 \\
& * b_{29} + 0.02831 * b_{30} + 0.07945 * b_{31} + 0.02336 * b_{32} + 0.01902 * b_{33} \\
x_9 = & 0.02608 * b_2 + 0.02603 * b_3 - 0.01378 * b_4 - 0.05548 * b_5 - 0.05067 * b_6 + 0.00253 \\
& * b_7 + 0.01087 * b_8 - 0.00965 * b_9 + 0.06033 * b_{10} + 0.01326 * b_{11} \\
& + 0.01226 * b_{12} + 0.00526 * b_{13} + 0.01837 * b_{14} + 0.97888 * b_{15} - 0.01591 \\
& * b_{16} + 0.0032 * b_{17} - 0.00123 * b_{18} - 0.01768 * b_{19} - 0.00329 * b_{20} \\
& + 0.01848 * b_{21} + 0.12117 * b_{22} - 0.00305 * b_{23} - 0.0139 * b_{24} - 0.1369 \\
& * b_{25} + 0.00533 * b_{26} + 0.00505 * b_{27} - 0.05941 * b_{28} + 0.00104 * b_{29} \\
& - 0.01049 * b_{30} - 0.0186 * b_{31} - 0.00655 * b_{32} - 0.00667 * b_{33} \\
x_{10} = & -0.06896 * b_2 + 0.81549 * b_3 + 0.00241 * b_4 - 0.23694 * b_5 - 0.40046 * b_6 \\
& + 0.00897 * b_7 + 0.09886 * b_8 + 0.0511 * b_9 + 0.14115 * b_{10} + 0.01766 \\
& * b_{11} - 0.01372 * b_{12} + 0.05231 * b_{13} + 0.09385 * b_{14} - 0.07247 * b_{15} \\
& - 0.0026 * b_{16} - 0.03807 * b_{17} - 0.0525 * b_{18} - 0.0481 * b_{19} - 0.02171 \\
& * b_{20} + 0.22346 * b_{21} + 0.00635 * b_{22} + 0.03034 * b_{23} + 0.00045 * b_{24} \\
& - 0.00333 * b_{25} - 0.00687 * b_{26} - 0.00598 * b_{27} + 0.00904 * b_{28} + 0.00161 \\
& * b_{29} - 0.03544 * b_{30} - 0.03107 * b_{31} - 0.0325 * b_{32} - 0.03056 * b_{33} \\
x_{11} = & 0.01163 * b_2 - 0.16585 * b_3 + 0.36106 * b_4 + 0.18758 * b_5 - 0.11202 * b_6 + 0.04734 \\
& * b_7 + 0.08089 * b_8 - 0.04776 * b_9 + 0.57646 * b_{10} + 0.22667 * b_{11} \\
& + 0.2502 * b_{12} + 0.16864 * b_{13} + 0.2837 * b_{14} - 0.03027 * b_{15} - 0.01914 \\
& * b_{16} - 0.04965 * b_{17} - 0.14663 * b_{18} - 0.14299 * b_{19} - 0.09084 * b_{20} \\
& + 0.34356 * b_{21} + 0.08813 * b_{22} + 0.01201 * b_{23} - 0.00602 * b_{24} + 0.07392 \\
& * b_{25} - 0.02033 * b_{26} - 0.01844 * b_{27} + 0.07671 * b_{28} + 0.03549 * b_{29} \\
& + 0.01246 * b_{30} - 0.04321 * b_{31} - 0.01435 * b_{32} - 0.02027 * b_{33} \\
x_{12} = & 0.31514 * b_2 + 0.12429 * b_3 + 0.20605 * b_4 + 0.59423 * b_5 - 0.07789 * b_6 - 0.02891 \\
& * b_7 + 0.43179 * b_8 + 0.04899 * b_9 - 0.349 * b_{10} - 0.11799 * b_{11} \\
& - 0.1614 * b_{12} - 0.01764 * b_{13} + 0.08022 * b_{14} + 0.03606 * b_{15} + 0.00807 \\
& * b_{16} - 0.01669 * b_{17} - 0.07519 * b_{18} - 0.07992 * b_{19} - 0.04447 * b_{20} \\
& + 0.1224 * b_{21} - 0.04675 * b_{22} + 0.02358 * b_{23} - 0.01033 * b_{24} - 0.05341 \\
& * b_{25} - 0.03852 * b_{26} - 0.03942 * b_{27} - 0.03475 * b_{28} + 0.05203 * b_{29} \\
& + 0.08373 * b_{30} + 0.08881 * b_{31} + 0.03572 * b_{32} + 0.03272 * b_{33} \\
x_{13} = & 0.11477 * b_2 + 0.12081 * b_3 + 0.26737 * b_4 + 0.41381 * b_5 - 0.38673 * b_6 + 0.03689 \\
& * b_7 - 0.45781 * b_8 - 0.06748 * b_9 + 0.15539 * b_{10} + 0.10259 * b_{11} \\
& + 0.12761 * b_{12} - 0.02892 * b_{13} - 0.13557 * b_{14} + 0.00461 * b_{15} + 0.01032 \\
& * b_{16} + 0.06111 * b_{17} + 0.10667 * b_{18} + 0.10173 * b_{19} + 0.05029 * b_{20} \\
& - 0.41614 * b_{21} + 0.01275 * b_{22} - 0.03055 * b_{23} + 0.01076 * b_{24} + 0.00767 \\
& * b_{25} + 0.05376 * b_{26} + 0.05218 * b_{27} + 0.0154 * b_{28} - 0.04556 * b_{29} \\
& - 0.02067 * b_{30} - 0.02977 * b_{31} + 0.00644 * b_{32} + 0.00503 * b_{33}
\end{aligned}$$

$$\begin{aligned}
x_{14} = & 0.37416 * b_2 + 0.16394 * b_3 - 0.09646 * b_4 + 0.12724 * b_5 + 0.31884 * b_6 - 0.01219 \\
& * b_7 - 0.33532 * b_8 + 0.08094 * b_9 + 0.01328 * b_{10} - 0.05983 * b_{11} \\
& - 0.00127 * b_{12} - 0.07644 * b_{13} - 0.22836 * b_{14} + 0.00642 * b_{15} + 0.03174 \\
& * b_{16} + 0.03447 * b_{17} + 0.1472 * b_{18} + 0.14883 * b_{19} + 0.09617 * b_{20} \\
& + 0.47179 * b_{21} - 0.02409 * b_{22} + 0.02534 * b_{23} + 0.03129 * b_{24} + 0.02356 \\
& * b_{25} + 0.04121 * b_{26} + 0.04058 * b_{27} - 0.06444 * b_{28} - 0.02597 * b_{29} \\
& - 0.14118 * b_{30} - 0.30544 * b_{31} - 0.05624 * b_{32} - 0.05155 * b_{33} \\
x_{15} = & 0.3812 * b_2 + 0.20951 * b_3 - 0.43083 * b_4 + 0.07625 * b_5 + 0.25238 * b_6 + 0.00937 \\
& * b_7 + 0.19825 * b_8 - 0.18262 * b_9 + 0.34273 * b_{10} - 0.13634 * b_{11} \\
& + 0.1319 * b_{12} - 0.02553 * b_{13} - 0.00458 * b_{14} - 0.01703 * b_{15} - 0.07222 \\
& * b_{16} + 0.02502 * b_{17} - 0.05658 * b_{18} - 0.05566 * b_{19} - 0.02409 * b_{20} \\
& - 0.342 * b_{21} + 0.13429 * b_{22} - 0.04821 * b_{23} - 0.05417 * b_{24} + 0.10232 \\
& * b_{25} + 0.01483 * b_{26} + 0.01639 * b_{27} + 0.13311 * b_{28} - 0.00558 * b_{29} \\
& + 0.04524 * b_{30} + 0.13436 * b_{31} + 0.02382 * b_{32} + 0.02693 * b_{33}
\end{aligned}$$

（三）压缩观测

因子分析后，将因子得分的数据输出到数据集 `case2.train_yin` 中，得到每一个观测的因子得分。每一个 `report_id` 对应着一个或者多个观测，我们将每一个观测的因子得分按照 `report_id` 的类别进行加总后求平均值。那么，可以成功地压缩观测至 3 万个。具体程序代码详见附录十三。

（四）过度抽样

某些情况下，我们想要的预测事件的发生概率非常低，如邮递营销中潜在客户的响应率、信用卡客户的贷款违约率。如果模型训练时，为了优化总体预测准确率，直接使用原始数据训练时的模型就没有什么用处。本文中，训练集中违约客户 $Y=1$ 的个案数为 1875，占有所有数据的 6.25%，未违约客户的 $Y=1$ 的个案数占的比例为 93.75%，因此我们需要对样本进行过度抽样。具体程序代码详见附录十四。

五、模型建立

（一）逻辑回归

1. 模型理论

假设事件发生的条件概率 $p(y_i = 1|x_i)$ 与 x_i 之间的非线性关系为单调函数是合理的，即随着 x_i 的增加（减少） $p(y_i = 1|x_i)$ 也单调增加，考虑到事件发生的条件概率的值域为 $(0, 1)$ ，因此，这种曲线类似于一个随机变量的累计分布曲线。最常用的分布函数是 logistic 分布。

假设有一个理论上存在的连续随机变量 y_i^* 代表事件发生的可能性，其值域为 $-\infty$ 到 ∞ 。当该变量的值跨越一个临界点 c 时，便导致事件发生了，即 $y_i = 1$ ；否则 $y_i = 0$ ，这里 y_i 是实际观测到的因变量取值。假设随机变量 y_i^* 和自变量 x_i 之间存在线性关系，即

$$y_i^* = \alpha + \beta x_i + \varepsilon_i$$

于是事件发生的概率为 $p(y_i = 1|x_i) = p[(\alpha + \beta x_i + \varepsilon_i) > c] = p[\varepsilon_i > (-\alpha - \beta x_i + c)]$ ，通常假设 ε_i 服从 logistic 分布，根据 logistic 分布的对称性，则有：

$$p[\varepsilon_i > (-\alpha - \beta x_i - c)] = p[\varepsilon_i \leq (\alpha + \beta x_i + c)] = F(\alpha + \beta x_i + c)$$

其中， F 为 ε_i 的累积分布函数，这里就是 logistic 分布的累积分布函数。为了方便表示，可以假设 $c = 0$ ，因此有：

$$p(y_i = 1|x_i) = F(\alpha + \beta x_i)$$

如果假设 ε_i 服从标准 logistic 分布，则累积分布函数可以有一个较为简单的公式：

$$p(y_i = 1|x_i) = p[\varepsilon_i \leq (\alpha + \beta x_i)] = \frac{1}{1 + e^{-(\alpha + \beta x_i)}}$$

这个函数就是说 logistic 函数，它具有 S 型分布。

Logistic 回归分析的目标变量是类别变量，线性回归分析的目标变量是数值变量。Logistic 回归分析的假设条件：

- （1）数据来自随机样本
- （2）自变量之间尽量独立，避免多重共线性

(3) 因变量是自变量的函数

(4) 线性回归模型中要求残差是独立同分布的，在 Logistic 回归中不需要

Logistic 回归中没有关于自变量的分布的假设条件，自变量可以是连续变量，分类变量等。线性回归分析估计参数使用的是最小二乘法或者极大似然法，而 Logistic 回归中只能使用极大似然法。

2.模型训练

选取因子分析之后的 15 个公共因子作为逻辑回归的自变量，然后将目标变量 Y 作为逻辑回归的因变量。从而，建立多元逻辑回归模型。具体建模的展示如下图，模型代码详见附录。

	REPORT_ID	Y	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
1	8787	0	-0.518686055	0.2727815567	0.9706346715	-0.131355683	-0.529768674	-1.02807293	0.305763658	-1.261656891	0.1625936286	0.5811361979
2	9410	0	0.6382173283	-0.484675765	-0.6971111661	1.8015647793	0.8515503607	2.1113101966	0.8399400678	-0.678888811	-0.019338943	-1.952383022
3	22999	0	-0.598835848	0.3410008975	0.7201392305	-0.083627415	-0.584307313	0.6327994156	0.1984075761	-1.03792508	0.0143474573	0.3936388547
4	24820	1	0.6318101462	-0.325827253	-0.182230896	0.2966384051	0.601317346	0.2228369839	0.3605073137	-1.603555084	0.1573041717	-0.238205129
5	25708	0	-0.541876657	0.0429705305	0.2268935442	0.408148774	0.2560405086	0.1286129429	0.2334539668	0.2522848886	-0.024654698	-0.467418024
6	26899	0	-0.568842687	0.0775516124	0.5065696704	0.9433126929	-0.698212979	0.114628868	0.3182647762	-0.999820534	0.1283263423	0.7060542089
7	28675	1	-0.56000095	0.104623768	0.6107531192	0.8435417559	-0.226630703	-0.657391735	-0.265502204	0.8639978644	-0.1759799	-2.285875382
8	41549	0	-0.415006873	0.0273273509	-0.126169312	0.7762898556	-0.334424587	-0.167816182	-0.344796223	-1.107153289	0.1353825535	0.5443023858
9	48614	0	0.0652701481	-0.055918019	0.5716099065	0.6279468958	-0.763338748	0.1402997883	0.5561713327	0.007495183	0.099905663	0.7082036359
10	49099	0	-0.85675702	0.1337775145	0.3124598717	-0.021908387	0.5301121039	-0.493412583	-0.280748146	1.4701502114	-0.091858602	-0.343932585
11	50598	0	-0.316475259	-0.083506006	0.011812376	0.8674955564	-0.370055109	0.2839126951	0.2259285385	-0.987586098	0.1191000039	0.5817936375
12	54038	0	-0.323777373	0.21042304	1.2200191792	0.6220254888	-0.024545497	-0.621939505	-0.768992561	0.5576120591	-0.014168862	0.4830777154
13	65159	0	-0.610521556	0.1246618624	0.5968502196	-0.100412739	0.7077665709	0.2580210295	0.7480738861	0.4793934242	0.0315499404	0.5196956108
14	70476	0	-0.176947034	0.0236573949	0.1893386063	0.3943132526	0.8981692593	-0.315050422	-0.323163208	1.4670189401	-0.101364641	-0.405397882
15	74971	0	-0.752659963	0.2955551359	0.5398654933	-0.047989458	-0.213001318	0.1332440945	0.8303580619	-0.799344139	-0.062165537	-1.92483047
16	88163	1	2.3285129867	-0.952933266	-1.255386351	0.0595951833	0.4835297012	-0.70632565	-0.927163659	-0.221047923	0.1080251192	0.9057646808
17	88203	0	-0.879936923	0.2946740752	0.5411616612	-0.449818707	0.2824240796	0.6798113256	1.3166098171	-0.060062763	0.00899186	-0.160210488
18	90751	0	-1.102453706	0.1502343677	0.1356265191	0.2092748322	0.2890572561	-0.366878438	-0.108783618	-0.099213239	-0.164434981	-2.585002888
19	98638	0	0.0548244887	-0.198123897	-0.096326167	1.4304749059	0.5214281421	-0.462402805	-1.589927618	0.3809687086	0.0378689702	0.5220041093
20	112594	1	-0.369002847	0.1435621785	0.0400718253	-0.42135181	-1.042317403	-0.190724421	1.1450712282	-0.546644455	-0.08313387	-2.023812909
21	113922	0	0.4246275552	-0.256796799	-0.134961325	0.4846799903	0.778984826	1.1619883611	0.8864932764	-0.00015423	0.0651349554	0.155300835
22	116104	0	-0.279089383	0.3185531226	0.6614652947	-1.176771036	0.0968808199	-0.526562047	0.5184430822	-1.213085818	0.1030678085	0.4722849823
23	117735	1	-0.079960474	-0.003364956	-0.189430033	0.6539177333	-0.90741938	1.3412374977	-0.637218567	-1.603680239	-0.142384725	-1.640750194
24	125045	0	0.6312130554	-0.244911237	-0.020938368	0.1449844714	-0.291140353	-0.779192296	0.4725011801	1.4628322795	-0.253744129	-1.377776276
25	126671	0	-0.768488037	0.0848604754	0.2793527384	0.3665370603	-0.356078723	0.1614225465	0.2617478233	-1.067589721	0.1249560427	0.6632508368
26	128098	0	-0.91369236	0.2899453538	0.2194658758	0.205182534	0.2659572707	-0.751877125	-0.665619254	1.4725740403	-0.131088723	-0.367709194
27	128445	0	0.6870671793	-0.266170669	0.2864082246	0.9436672902	-0.287073565	0.1594487169	0.4437681406	1.3803279148	-0.035144345	-0.066872256
28	130668	0	-0.330724242	-0.121791888	-0.170139777	0.4044173501	0.5803525631	0.1906100773	0.0241901496	-1.332817812	0.2043980475	0.7537409423
29	130703	1	-1.146658239	0.2430431729	0.2702647069	-0.470652244	0.4851246622	-1.332264637	-0.064689787	2.4210445128	-0.226791347	-0.919880121
30	134260	0	-0.276176881	-0.085128363	0.0211070469	0.2165123872	0.8484898928	0.2387810179	0.4963566229	1.1091115885	0.036470778	0.2178523067
31	136349	0	-0.151144487	-0.059215161	-0.45317627	-0.08728479	0.778254737	-0.877810125	-1.136552515	0.2801872335	0.0145672079	0.4640192377
32	138676	0	-0.741091345	0.0507898866	0.094590269	0.8036703706	-0.420118819	-0.723087834	-0.676919929	-0.701645256	0.0558217977	0.2929219423
33	138988	0	-0.487200379	-0.007391975	0.0711482226	0.5498557142	0.1742934628	-0.708015987	-1.040881732	0.4917809219	0.0017617663	0.3976195667
34	139114	0	-0.084357481	-0.144748886	-0.230408423	0.1359393745	0.7040121735	0.1878167728	0.0947064126	1.0407927879	-0.015167219	-0.13148712

图 4 逻辑回归模型的变量数据集

上述的 case2.train_yin_ya1 数据集中，变量分别为 report_id, Y, x1-x15。其中，Y 为因变量，x1-x15 为自变量。下面开始进行模型的训练。

我们采用逐步回归的办法，对公共因子 x1-x15 进行逐步回归剔除。最后，我们选择变量 x1, x3, x6, x8, x10, x11, x14, x15 这几个公因子进行建模。

六、模型求解和检验

下面开始用 SAS 系统求解逻辑回归模型，下图为建模的总体介绍，包括建模数据集的名称、响应变量的名称、响应水平数量、模型的种类、优化方法、读入和参与建模的观测、响应概况等。

SAS 系统		
“LOGISTIC” 过程		
模型信息		
数据集	CASE2.TRAIN_YIN_YA2	
响应变量	Y	Y
响应水平数	2	
模型	二元 Logit	
优化方法	Fisher 评分法	

读取的观测数	8625
使用的观测数	8625

响应概况		
有序值	Y	总频数
1	0	6750
2	1	1875

建模的概率为 Y=1。

图 5 逻辑回归建模总体介绍

其中，响应概况表输出了因变量的取值和频数。其中，响应变量取值为 0 的频数为 6750，取值为 1 的频数为 1875。

紧接着输出逐步选择过程的第 0 步的情况，详见下图。

由于使用了 selection=stepwise 选项，因此进行逐步回归的第 0 步的时候，拟合的模型只有截距项，残差的卡方检验是比较了包含全部自变量的模型和当前模型。

残差卡方检验的原假设是全模型和当前模型没有显著性差别。此处的 P 值为 <0.0001，因此，拒绝原假设，说明全模型和当前模型具有显著差别，即全部自

变量中还存有预测价值的变量。



图 6 逐步回归选择过程第 0 步和第 1 步

第一步时 x10 进入模型，首先模型的状态是收敛的，这个结果和变量的选择方法无关，是参数估计使用了迭代法的结果，模型拟合必须达到收敛的状态。

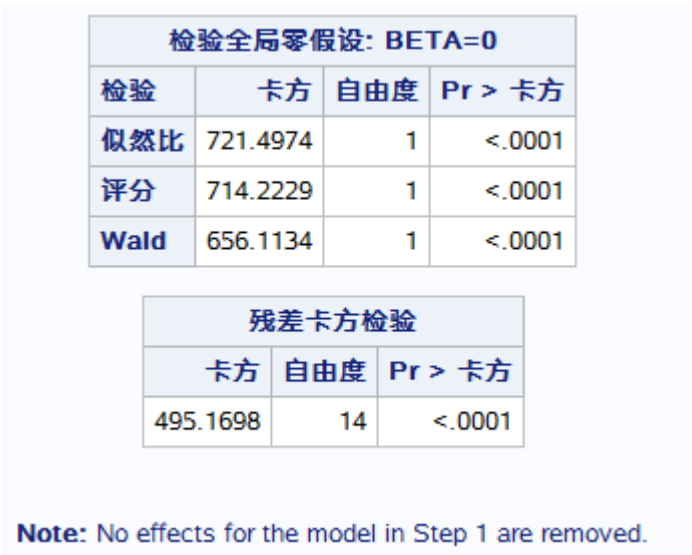


图 7 逐步回归选择过程第 1 步

检验全局零假设，提供三种方法：似然比、评分、wald 检验，这三种方法得到的卡方值均小于 0.05，说明 x10 和截距项是因变量有预测作用的。残差卡方

检验说明剩余自变量中仍有对因变量有预测作用的变量。

那么下面就是重复上述第一步的步骤，直到所有的剩余变量均被迭代找到为止。然后生成逐步选择汇总的报表图，具体详见下图。

逐步选择汇总						
步	效应		自由度	个数	评分 卡方	Wald 卡方
	已输入	已删除				
1	x10		1	1	714.2229	
2	x8		1	2	208.5140	
3	x1		1	3	158.3666	
4	x15		1	4	55.4048	
5	x3		1	5	35.6185	
6	x11		1	6	25.9944	
7	x14		1	7	16.6907	
8	x6		1	8	4.5051	0.0338

图 8 逐步回归选择汇总报表图

通过逐步选择汇总，可以看出该模型共运行 8 步，共有 8 个变量进入模型，剔除了 7 个变量。

下面输出参数估计的值，具体详见下图。

最大似然估计分析					
参数	自由度	估计	标准 误差	Wald 卡方	Pr > 卡方
Intercept	1	-1.8950	0.0418	2055.5285	<.0001
x1	1	-0.5265	0.0483	118.7418	<.0001
x3	1	0.2866	0.0601	22.7794	<.0001
x6	1	-0.0809	0.0381	4.5021	0.0339
x8	1	-0.4095	0.0293	195.6972	<.0001
x10	1	-0.6236	0.0281	491.3813	<.0001
x11	1	0.1861	0.0366	25.8753	<.0001
x14	1	-0.1338	0.0381	12.3068	0.0005
x15	1	-0.1933	0.0369	27.4737	<.0001

图 9 参数估计报表图

从参数估计报表图中，得到：截距项为-1.8950，变量 x1 的系数为-0.5265，

变量 x3 的系数为 0.2866，变量 x6 的系数为-0.0809 等等。那么，我们可以得到逻辑回归的模型为：

$$p = \frac{T}{1 + T}$$

其中： $T = e^{f(x)}$,

$$f(b) = -1.895 - 0.526 * x_1 + 0.2866 * x_3 - 0.0809 * x_6 - 0.1095 * x_8 - 0.6236 * x_{10} + 0.1861 * x_{11} - 0.1338 * x_{14} - 0.1933 * x_{15}$$

下面输出优比估计报表和预测概率报表图，具体如下图。

优比估计			
效应	点估计	95% Wald 置信限	
x1	0.591	0.537	0.649
x3	1.332	1.184	1.498
x6	0.922	0.856	0.994
x8	0.664	0.627	0.703
x10	0.536	0.507	0.566
x11	1.205	1.121	1.294
x14	0.875	0.812	0.943
x15	0.824	0.767	0.886

预测概率和观测响应的关联			
一致部分所占百分比	76.4	Somers D	0.531
不一致部分所占百分比	23.3	Gamma	0.533
结值百分比	0.3	Tau-a	0.181
对	12656250	c	0.766

图 10 优比估计和预测概率报表图

上述输出指标可以模型的预测准确度，具体放入第七节中的模型评估部分，然后再进行详细的介绍。

七、模型评价

（一）模型估计

利用训练集的结果对测试集进行预测，评估测试集的预测效果。下面采用直接打分的方式，输出的数据集 case2.train_score_r 中。具体如下图所示：

选择项的概率	抽样权重	从：Y	到：Y	预测概率： Y=0	预测概率： Y=1
0.24	4.166666667	0	0	0.8397007267	0.1602992733
0.24	4.166666667	0	1	0.4466364496	0.5533635504
0.24	4.166666667	0	1	0.3731767131	0.6268232869
0.24	4.166666667	0	0	0.9237138126	0.0762861874
0.24	4.166666667	0	0	0.8996387809	0.1003612191
0.24	4.166666667	0	0	0.9261142223	0.0738857777
0.24	4.166666667	0	0	0.8561094816	0.1438905184
0.24	4.166666667	0	0	0.8008451847	0.1991548153
0.24	4.166666667	0	0	0.8160576888	0.1839423112
0.24	4.166666667	0	0	0.9096785612	0.0903214388
0.24	4.166666667	0	0	0.8424464699	0.1575535301
0.24	4.166666667	0	0	0.7865390462	0.2134609538
0.24	4.166666667	0	0	0.8638029892	0.1361970108
0.24	4.166666667	0	0	0.8552611018	0.1447388982
0.24	4.166666667	0	0	0.8221567483	0.1778432517
0.24	4.166666667	0	0	0.9246046666	0.0753953334
0.24	4.166666667	0	0	0.8842357706	0.1157642294
0.24	4.166666667	0	0	0.9307088383	0.0692911617
0.24	4.166666667	0	0	0.8757648892	0.1242351108
0.24	4.166666667	0	0	0.8535475424	0.1464524576
0.24	4.166666667	0	0	0.9154970298	0.0845029702
0.24	4.166666667	0	0	0.8039991393	0.1960008607
0.24	4.166666667	0	0	0.9212067342	0.0787932658
0.24	4.166666667	0	0	0.8304818837	0.1695181163
0.24	4.166666667	0	0	0.7579542528	0.2420457472
0.24	4.166666667	0	0	0.8351437628	0.1648562372
0.24	4.166666667	0	0	0.8602226197	0.1397773803
0.24	4.166666667	0	0	0.7496714286	0.2503285714
0.24	4.166666667	0	0	0.8407945453	0.1592054547
0.24	4.166666667	0	0	0.8230692017	0.1769307983
0.24	4.166666667	0	0	0.8660764637	0.1339235363
0.24	4.166666667	0	0	0.9044624833	0.0955375167
0.24	4.166666667	0	0	0.8629121178	0.1370878822

图 11 模型估计输出的数据集 case2.train_score_r

从上述的数据集中，可以知道每一个观测的选择项的概率、抽样权重、响应变量的真实值、响应变量的预测值、预测 Y=0 的概率以及预测 Y=1 的概率。从响应变量的真实值、响应变量的预测值、预测 Y=0 的概率以及预测 Y=1 的概率中，可以得到违约的概率，即过度抽样后样本的违约概率的具体值。此时我们计算得到，Y=1 的有 2061 个，原始样本中 Y=1 的有 1875，而我们预测得到的结果 Y=1 与原来相对应的 Y=1 只有 409 个能够对上，整体的正确判别率为 89.6%。

（二）模型评估

1.拟合优度

在对 Logistic 模型进行拟合优度评价时常用的是 AIC 和 SBC 准则。

AIC 的计算公式为 $AIC = -2 \log L + 2k$ ，其中 L 指似然函数的取值， k 是指参数的个数。

SBC 的计算公式为： $SBC = -2 \log L + k \log n$

对于这两个准则都是取值越小，模型越好，但需要注意的是，AIC 准则和 SBC 准则只适用于同一数据不同模型之间的比较，不适合不同数据模型之间的比较。

模型拟合统计量		
准则	仅截距	截距和协变量
AIC	9033.864	7796.280
SC	9040.927	7859.842
-2 Log L	9031.864	7778.280

图 12 模型拟合统计量报表图

在上述的模型拟合统计量报表图中，我们得到：仅截距的 AIC 值为 9033.864，SC 值为 9040.927， $-2 \log L = 9031.864$ 。

2.预测准确性指标

Logistic 回归模型的因变量只有两种可能值（0 或者 1，发生或者不发生），我们可以按事件是否发生将观测分成两组，每组中各取一条观测，形成一个观测数据对。如果观测到事件发生组的观测条数为 100，观测到事件未发生组的观测条数为 200，则总共有 $100 \times 200 = 20000$ 个观测对。在一个观测数据对中，如果事件发生的观测的预测概率值大于事件未发生的观测的预测概率值，就定义该观测数据对为和谐对；如果事件发生的预测概率值小于事件未发生的观测的概率预测值，就定义该观测数据对为不和谐对。如果一个观测数据对既不是和谐对，也不是不和谐对，也就是说，事件发生的观测的概率预测值等于事件未发生的观测的预测概率值，那就定义该观测数据对为结。相关指标的计算公式如下：

$$\text{Gamma} = \frac{nc - nd}{nc + nd}$$
$$\text{Somers'D} = \frac{nc - nd}{t}$$

$$\text{Tau-a} = \frac{nc - nd}{0.5n(n-1)}$$

$$c = \frac{nc + 0.5t(t - nc - nd)}{t}$$

其中， n 为样本容量， t 为总的观测数据对数， nc 是和谐对的数量， nd 是不和谐数据对的数量。这些指标用于同一组数据的不同模型之间的比较，指标的取值越大，说明模型的预测准确度越高。指标 c 和 Somers'D 在应用于比较 logistic 模型时通常较好。

预测概率和观测响应的关联			
一致部分所占百分比	76.4	Somers D	0.531
不一致部分所占百分比	23.3	Gamma	0.533
结值百分比	0.3	Tau-a	0.181
对	12656250	c	0.766

图 13 预测准确性报表图

从上图可知，和谐对百分比为 76.4%，不和谐对的百分比为 23.3%，结百分比为 0.3%，观测的总对数为 12656250，Somers D 为 0.531，Gamma 为 0.533，Tau-a 为 0.181， c 为 0.766。这里的 c 值代表了使用该模型时，观察到事件发生的观测的预测概率值比观察到事件未发生的观测的预测概率值更大的可能性为 0.766，它是表示模型区分度的指标。

3. ROC 曲线

ROC 曲线是一种有效比较两个或多个二元分类模型的可视化工具。ROC（接受者运行特征）来源于信号检测理论，它显示了给定模型的灵敏性与假正率之间的比较评定。

真正率的增加是以假正率的增加为代价的，ROC 曲线下方的面积就是比较模型准确度的指标和依据，成为 AUC 统计量，或称 C 统计量，面积大的模型对应的模型准确度要高，也就是择优应用的模型，面积越接近于 0.5，对应，模型的准确率就越低。即 ROC 离对角线越近，模型的正确率就越低。

要绘制 ROC 曲线，ROC 的横轴为假正率，纵轴表示真正率。具体绘制时，首先要对模型预测的 $\text{response}=1$ 概率从高到低排序，从左下角开始，在此真正率和假正率都为 0，对每个观测值实际的“正”或“负”进行 ROC 图形的绘制，如果此样本是真正的“正”，则在 ROC 曲线上向上移动并绘制一个点；如果此样本是真正的“负”，则在 ROC 曲线向右移动并绘制一个点。依次对每个观察值重

复这个过程。

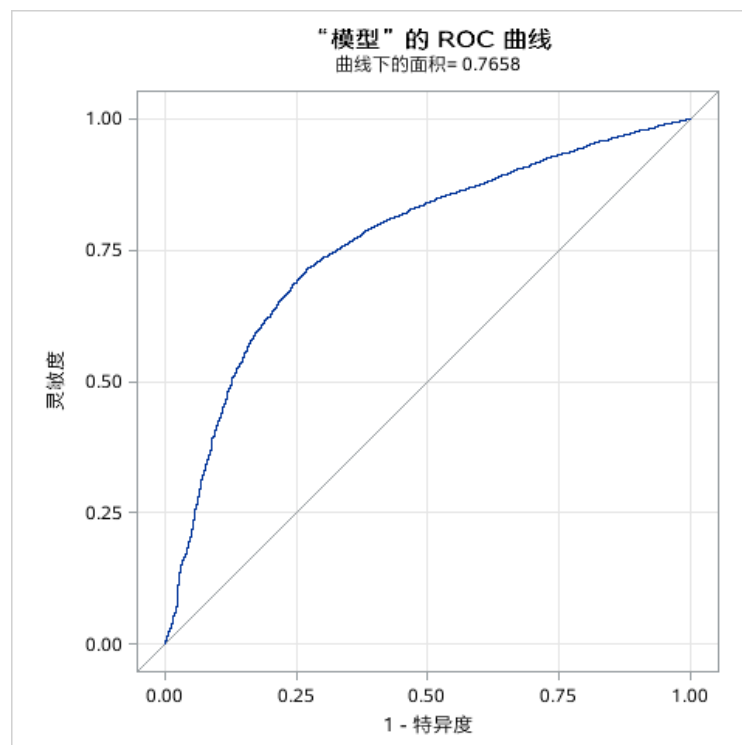


图 14 模型的 ROC 曲线

从 ROC 曲线可以看出，曲线下的面积为 0.7658，大于 0.5，再从曲线与 ROC 离对角线的远近程度可以看出，曲线与 ROC 对角线是比较远的，说明模型是比较好的。

4.KS 曲线

KS 指是一种判断二元分类预测模型准确度的方法，该方法来源于统计学中的 Kolmogorov-Smirnov test，柯尔莫哥洛夫-斯米尔洛夫曲线。KS 统计量是指 KS 曲线中差异的最大值，在评价二元分类模型的预测能力时，越大则区分效果越好，通常来讲大于 0.2 就表示模型有较好的预测性。

绘制过程如下：（1）将总体按照违约概率降序排列；（2）将数据集进行十等分，计算每一份中违约、正常的累计占比；（3）将这两种累计的百分比绘制在同一张图上。

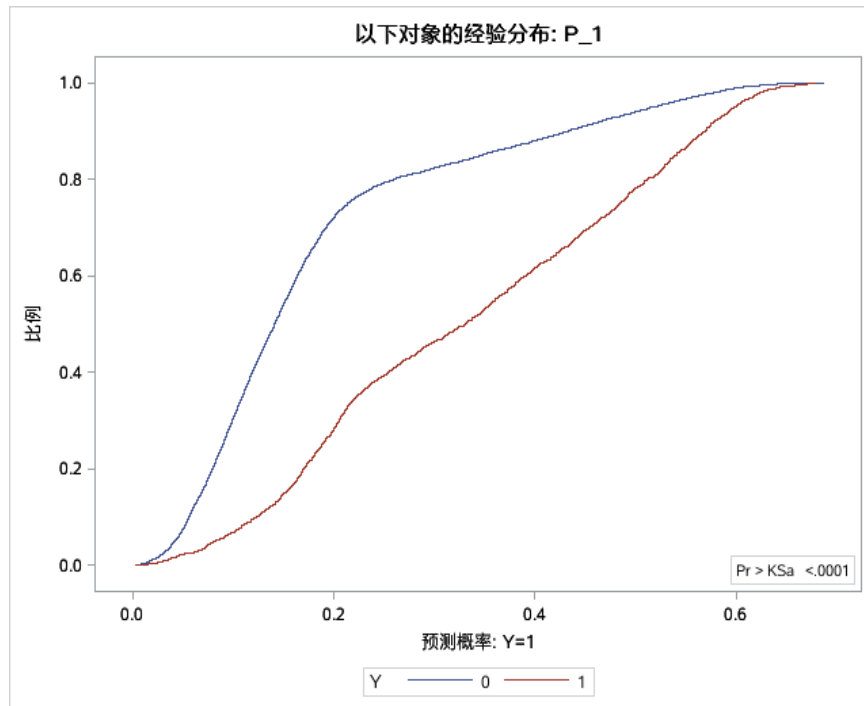


图 15 模型的 KS 曲线

从 KS 统计量的选择标准来看，很多地方的 KS 曲线中差异的最大值，在评价二元分类模型的预测能力时，越大则区分效果越好，通常来讲大于 0.2 就表示模型有较好的预测性。

八、模型结果分析

（一）模型输出（测试集的验证）

从上述的结果中，可以知道，根据训练集建立的模型有一定的精确度，而且正确率还比较高，误判率比较低。

所以，接下来，进行的是，将测试集中的数据回代我们已经建立好的模型中。

首先，我们要抽取测试集中所有带入模型所需要的变量，然后查看数据集中是否还有字符型的变量，如果有的话，要将字符变量转化为数值型变量。

紧接着，我们对变量处理完了之后，就要接着处理数值变量。在这里，我们首先将重复的观测删掉；然后将观测中的缺失值，按照训练集的建模方法将缺失值进行填补。

对缺失值的填补全部完成之后，方可进行下一步骤。

缺失值全部处理好之后,进行数据的标准化,将我们选取的变量全部标准化,有利于下一阶段的数据分析和处理。

然后,再进行因子分析的降维数的处理。因子分析的时候,公共因子的产生按照训练集的产生方式。然后每一个观测都会有一个因子得分,我们将每一个客户的因子得分加总后求得平均值。那么因子得分的平均值即为衍生出来的新变量。我们将其通过标准化了之后,进行逻辑回归模型的建立。最终根据取值的概率来确定目标变量的取值。目标变量的取值概率和实际值如下图 16 所示。

	x7	x8	x9	x10	x11	x12	x13	x14	x15	f	y	y_hat
1	0.2896219736	-0.948048183	0.1462876901	0.9913315509	-0.179740379	-0.058766168	0.0672300173	0.0047326349	0.8119119747	-1.879345606	0.1324640567	0
2	-1.316666232	0.4023871252	-0.105183401	0.1197762256	-0.218225507	-0.413126169	0.6179548899	-0.425578867	-0.02948141	-1.854781626	0.135124562	0
3	-0.323996865	-0.386508521	7.5582274001	-0.042543709	-0.255233096	0.8209572002	1.5524495391	0.6932166874	0.1089064282	-2.716087557	0.0620307139	0
4	-0.45948501	0.3302755952	-0.154987348	0.0268109421	0.29526233	0.0019102686	1.0268563187	-0.757448086	-0.930266122	-1.615733618	0.1657940971	0
5	-0.470697942	1.609242575	9.3454362754	-1.086310082	-0.242118407	-0.542671496	-0.295298747	0.4268215121	0.0716859878	-1.450821803	0.1853416648	0
6	-1.124551938	0.2531532552	-0.0971003638	0.1832031033	-0.02118468	-0.104611405	0.2752441397	-0.625543281	-0.014937511	-1.512178185	0.1896185049	0
7	-1.639857582	-0.363899367	0.1250563056	0.9551729473	0.3727960963	-1.125282571	-1.522387963	0.6120144426	-0.809517192	-2.3569548	0.0865224565	0
8	-1.080645685	-0.097089312	0.005750511	0.3964565364	-0.639064545	-1.175481455	-0.169658396	-0.994961244	-0.494375954	-2.214747192	0.0849533891	0
9	0.0238156835	-1.488423245	0.2255670789	0.3760729065	0.4502784979	-1.193057737	0.6296627773	0.2289717291	0.8327705011	-1.651581272	0.1608953499	0
10	-0.285352113	-1.324257829	5.586742239	0.1592779554	-0.614741393	-0.267150269	-0.396611473	-0.005973259	0.7652201934	-1.30098711	0.214022239	0
11	-0.462892084	-0.216781032	0.0906532632	0.5509146262	0.6731159545	-1.360895217	0.1002341941	-1.391966211	-0.71425956	-1.650651095	0.161020973	0
12	0.8018679847	0.4794251204	-0.282108565	-1.110450392	-0.611825981	0.4327209712	0.3739064953	1.0060513554	1.6051446445	-2.896569373	0.0528214944	0
13	0.0564088371	0.8722705608	-0.143203605	-0.376093857	-0.746896356	-1.661292646	-1.452230319	-0.316450758	0.7405675438	-1.723839103	0.1513773236	0
14	-0.248159453	-1.898683459	0.102312293	-1.39134624	0.0971552587	-0.9129958	0.0918912341	0.0835511167	-0.178891765	0.1550023457	0.5386731881	1
15	1.9139672204	0.5927779964	-0.398258843	-2.193139964	0.6062596633	-0.748435212	-1.45094717	-2.524441372	-0.408820962	0.0325351657	0.508133074	0
16	-0.077982092	-0.768602229	0.1682056261	1.420305438	-0.44088941	-0.132502894	0.3715939149	0.2209514157	1.1195671732	-0.672171538	0.0247989828	0
17	0.7756696354	-0.17196924	-0.063587446	-0.238965475	0.205603315	-0.234747483	-1.617928963	2.353780711	0.3331953645	-2.754760925	0.0581833335	0
18	0.2789554006	-1.317806491	0.3233354497	1.2566356166	1.5999454533	-0.364450714	0.001875415	-0.746661364	0.775269365	-1.25586745	0.221668109	0
19	0.161173512	-0.024430036	-0.189173033	-0.858287214	-0.617275543	0.3036372625	-0.531568911	0.9352297412	1.8418931729	-1.338386704	0.207754896	0
20	-0.103350629	-1.876949647	-0.083682761	-1.429973562	0.2173450707	-0.043502157	0.168390737	-1.070335054	-0.503325363	0.4906255381	0.6202537821	1
21	-0.157109984	0.846496369	-0.343072895	1.1857984347	-0.538822692	1.1684355299	-0.973702617	-0.874420495	-1.632233125	-1.67867995	0.0403921997	0
22	-1.427276695	0.4072193233	-0.190001579	0.0453036615	-0.93703064	0.33993978	-1.78638179	-1.118369867	0.22854713	-1.426258518	0.193704846	0
23	-0.489953897	-0.134064556	0.0035266234	0.5139781339	0.021143284	-1.417962615	-0.434747718	-1.922064338	0.1890849169	-1.141040497	0.2421293757	0
24	0.2569545671	-0.076780021	-0.244807591	-0.288227707	-0.711251654	0.4719819599	-2.216657016	2.1346650577	-0.549615339	-1.251711231	0.2224040571	0
25	0.2434802648	0.4018476024	-0.164501223	-0.405031602	-0.042076519	0.9557935263	0.125918374	1.0523863377	1.3002060884	-2.350059015	0.0870610817	0
26	-0.820609576	-1.307512896	-0.051492152	0.1439230403	-1.817022814	0.495748637	-0.813114675	-0.119207985	-0.048671612	-1.070782814	0.252524232	0
27	-0.65102547	0.9632794316	-0.285422084	0.251912918	0.624951903	2.0455197584	0.025995359	0.2186215544	0.2291998517	-0.516877454	0.027222225	0
28	0.303322182	1.1967532086	-0.11991472	0.040288261	-0.234231627	-1.576160362	-0.688853421	0.1271086239	0.0954049651	-0.591271819	0.069702268	0
29	0.2026830803	-0.438651509	-0.042503419	0.36575134	0.166749613	1.1223719118	0.891762438	-0.473233088	1.2824214193	-2.13010604	0.1062049553	0
30	0.4388491303	-0.091434244	-0.006867131	0.5922159275	-0.237297536	-0.402512187	-1.944761072	2.4658330963	0.8157929521	-2.811840941	0.066827278	0
31	-0.306235289	1.5094394113	-0.290130397	-0.73394957	-0.00089794	-0.33549452	-0.229170012	-0.091930436	0.469500668	-2.42464458	0.0813114468	0
32	0.3015653205	-1.203702594	0.0718879567	0.0953272167	-0.13515955	-0.459221104	0.2346019234	-0.02269818	0.4659593462	-1.167553337	0.237284289	0
33	1.5768006963	-0.063629411	-0.121364359	-0.680167873	-0.03177596	-0.514177205	-0.301472208	0.6001996825	0.7154516859	-1.924653022	0.1273435718	0
34	-0.356530334	1.4970447956	-0.228594732	0.4297054655	0.3748756371	0.0541628392	-1.883153627	0.9143703394	0.3087139881	-2.281832161	0.0926388324	0

图 16 测试集目标变量的取值

上图给出了测试集中部分的目标变量的取值,但是由于空间的原因,无法展示出所有的观测,但是测试集中违约的客户的个数,用结构查询语言已经求出,具体如下图 17 所示:

```

20 proc sql;
21 create table test_weiyue as select * from case2.test_suoyou_score where y=1;
NOTE: 表 WORK.TEST_WEIYUE 创建完成,有 458 行,19 列。

22 run;
NOTE: PROC SQL 语句被立即执行; RUN 语句无效。
23 quit;
NOTE: "PROCEDURE SQL" 所用时间 (总处理时间):
    实际时间      0.06 秒
    CPU 时间      0.01 秒

```

图 17 测试集中违约客户个数

在上图 17 中,可以知道,通过模型测试集中筛选出来 458 个违约的客户,占了一万个测试集用户的 4.58%。根据上述的训练集中有的 1857 个违约客户,占了三万个训练集中用户的 6.19%。就占比而言,建立的逻辑回归的模型的可靠性还是很好的。上述的编程代码详见附录十七。

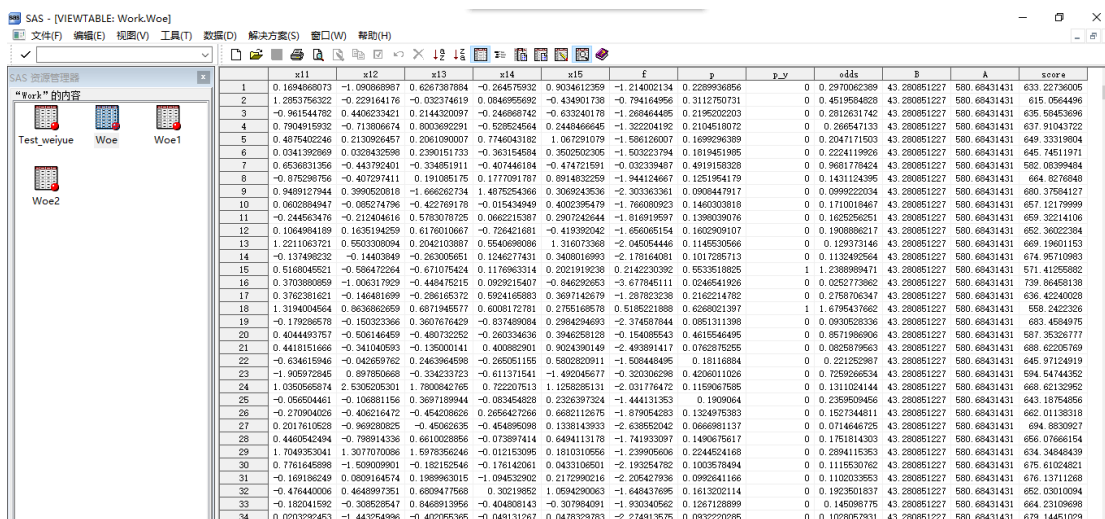
（二）策略咨询

就上述的结果来看，可以对客户实施建立的评分卡模型，拒绝训练集中 1875 个客户的申请贷款的请求，以及拒绝测试集中 458 个客户的贷款需求。

但是，不一定通过本文模型中的预测，就一定能断定所有的客户都是这样的。也不能消除一些客户的误判率，所以最保险的方法是，不断更新客户数据，不断调整模型，优化评分卡。

（三）将模型转化为评分卡

建立评分卡的最终目的是为了建立合适的评分体系和构建具体的量化指标。在下图 18 中，可以看到每一个客户的最终的评分。其实实质上 and 芝麻信用的分数是类似的，但是基础分和评分刻度不相同。



	x11	x12	x13	x14	x15	f	y	p-v	n11	B	A	score
1	0.1694868073	-1.090689987	0.6267387884	-0.264575932	0.9034412359	-1.214002134	0.2289526856	0	0.2970062389	43.280851227	580.68431431	633.22736005
2	1.2853756322	-0.229164176	-0.032374619	0.0846985692	-0.434901738	-0.794164956	0.3112750731	0	0.4519584628	43.280851227	580.68431431	615.05643696
3	-0.961544782	0.4406233421	0.2144320097	-0.246868742	-0.633240178	-1.268464495	0.2195202203	0	0.2812631742	43.280851227	580.68431431	635.58453696
4	0.7804915932	-0.713006674	0.8003692291	-0.528924564	0.2448466645	-1.322204192	0.2104518072	0	0.266547133	43.280851227	580.68431431	637.91043722
5	0.4875402246	0.213026457	0.2061090007	0.774043182	1.067291079	-1.586126007	0.169296389	0	0.2047171503	43.280851227	580.68431431	649.33319804
6	0.0341392089	0.0304323296	0.2390151733	-0.363154564	0.3503032305	-1.502223794	0.1819451985	0	0.2024119928	43.280851227	580.68431431	645.74511971
7	-0.6536831356	-0.443792401	-0.334951911	-0.407446184	-0.474721591	-0.032339487	0.4919158328	0	0.9681778424	43.280851227	580.68431431	582.08399484
8	-0.875286756	-0.407297411	0.191085175	0.1777091787	0.8914832259	-1.944124667	0.1251954179	0	0.1431124395	43.280851227	580.68431431	664.8276848
9	0.9489127944	0.3990520818	-1.666262734	1.4875254366	0.3069243536	-2.303363361	0.0908447917	0	0.0999222034	43.280851227	580.68431431	680.37584127
10	0.0602984947	-0.085274796	-0.422769178	-0.015454949	0.4002395479	-1.766009293	0.1460303818	0	0.1710018467	43.280851227	580.68431431	697.12179999
11	-0.2445653476	-0.212404016	0.5783070725	0.063215387	0.2307242644	-1.616919597	0.1399339076	0	0.1632526251	43.280851227	580.68431431	659.32214106
12	0.1064984189	0.1635194259	0.6176010667	-0.726421681	-0.419392042	-1.65605154	0.1602908107	0	0.1908886217	43.280851227	580.68431431	652.36022384
13	1.2211063721	0.5503308094	0.2042103887	0.5546868066	1.316073368	-2.045054446	0.1145530566	0	0.129373146	43.280851227	580.68431431	669.19601153
14	-0.137498232	-0.14403849	-0.263005651	0.1246277431	0.3408016993	-2.178164081	0.1017285713	0	0.1132492564	43.280851227	580.68431431	674.95710983
15	0.5168045521	-0.586472264	-0.671075424	0.1176963314	0.2021919238	0.2142230392	0.5535618825	1	1.2388989471	43.280851227	580.68431431	571.41255882
16	0.3703880869	-1.006317929	-0.448475215	0.0929215407	-0.846292853	-0.677845111	0.0246541926	0	0.0252772662	43.280851227	580.68431431	739.86459138
17	0.3762361621	-0.146481699	-0.286145372	0.592415683	0.3697142879	-1.287822838	0.2162214782	0	0.2758706347	43.280851227	580.68431431	636.42240028
18	1.3194004564	0.8636862659	0.6871945577	0.6008172781	0.2755168578	0.5186221888	0.6288021397	1	1.6795437662	43.280851227	580.68431431	558.24222326
19	-0.179286578	-0.150323366	0.3607676429	-0.837489084	0.2984294693	-2.374587844	0.0851311398	0	0.0930528336	43.280851227	580.68431431	683.45849475
20	0.4044493757	-0.506146459	-0.480732252	-0.260334636	0.3946258128	-0.154085543	0.4615546495	0	0.9571986906	43.280851227	580.68431431	587.35326777
21	0.4418151666	-0.341040593	-0.135000141	0.400882901	0.924390149	-2.493891417	0.0762875255	0	0.0625879563	43.280851227	580.68431431	688.62205769
22	-0.634615946	-0.042591762	0.2403944598	-0.285651155	0.580232911	-1.508448495	0.18116884	0	0.221252067	43.280851227	580.68431431	645.97124919
23	-1.905972845	0.897850668	-0.334233723	-0.611371541	-1.492045677	-0.320306298	0.4206011026	0	0.7259266534	43.280851227	580.68431431	594.54744352
24	1.0350565874	2.5305205301	1.7800842765	0.722207513	1.258285131	-2.031776472	0.1189067585	0	0.1311024144	43.280851227	580.68431431	686.62132952
25	-0.056504461	-0.106881156	0.3697189944	-0.083454828	0.2326397324	-1.444131353	0.1909064	0	0.2359505456	43.280851227	580.68431431	681.18754956
26	-0.270904028	-0.408216472	-0.454208626	0.3956457266	0.6882112675	-1.879054283	0.1354975383	0	0.1527344811	43.280851227	580.68431431	662.01136318
27	0.2017610528	-0.969208205	-0.403626375	-0.454885096	0.1338143953	-2.638552042	0.068981137	0	0.01744646725	43.280851227	580.68431431	694.58330827
28	0.4460542404	-0.788914336	0.6610028856	-0.073897414	0.6494113178	-1.741933097	0.1490675617	0	0.1751814303	43.280851227	580.68431431	656.07666154
29	1.7049353041	1.3077070086	1.5978356246	-0.012153095	0.1810310556	-1.239905606	0.2244524168	0	0.2894115353	43.280851227	580.68431431	634.34848439
30	0.7761648898	-1.509009901	-0.182125246	-0.176142061	0.0433106501	-2.193254782	0.1003578494	0	0.1115530762	43.280851227	580.68431431	675.61024821
31	-0.169186249	0.0809144574	0.1989963015	-1.094532902	0.2172990216	-2.205427938	0.0992641166	0	0.1102033553	43.280851227	580.68431431	676.13711268
32	-0.476440036	0.4648897351	0.609471596	0.30218952	0.084290663	-1.446437695	0.1613202114	0	0.1923501837	43.280851227	580.68431431	652.03010994
33	-0.182041592	-0.309528547	0.8468913956	-0.404858143	-0.307894091	-1.930340562	0.1267128899	0	0.145098775	43.280851227	580.68431431	664.23109698
34	0.0203292453	-1.443254996	-0.402055365	-0.049131267	0.0478329783	-2.274913575	0.0932220285	0	0.1028057931	43.280851227	580.68431431	679.14451029

图 18 评分量化结果

从上述的评分量化的结果来看，模型已经转化为评分卡的形式了。具体的程序详见附录十八。

至此，我们基本上完成了互联网金融时代下的信用评分体系构建和信用评分模型的构建。

结论与建议

完整的互联网下的金融评分卡的开发与传统的评分卡还是有些许不同。互联网下的金融评分卡开发的流程可以总结为：

数据导入（业务认识→数据探索→变量选取）、数据清洗（数据集合并→离群值的处理→缺失值的处理→衍生变量的产生→变量属性探究→数据抽样处理）、特征工程（降低维数→分箱处理→证据权重转换）、检验和建立模型（变量的相关性和共线性检验→模型的选择→业务约束→模型拟合→模型区分→逻辑回归模型的建立）、模型评估（混淆矩阵→ROC 曲线→KS 检验→VIS）、模型的部署与监控（模型的输出→策略咨询→模型监控→模型的不断优化）。

事实上，互联网下的金融评分卡的开发是一个动态的持续不断的调整优化的过程，但是本篇文章由于原始提供的数据不能得到更新，所以，本文所述的评分卡是一个静态的过程，但正是由于这一个个静态的过程，金融评分卡才会呈现出动态的活力。

本文的特征工程的建立是在建立模型之后的，从而导致了模型相对来说不是那么的精确。但是，考虑到评分卡的实质目的来说，本文的流程和处理方式是正确的可取的。

本文中，由于数据的更新问题导致我们不能处理模型的监控和模型的优化部分。但是，在实际的业务处理中，作为一个合格的数据分析师，本着对职业和对公司负责任的态度，在评分卡已经建立好了之后还必须实时更新更优的评分卡模型，以达到在申请评分之前将所有的反欺诈客户和信誉不好的客户筛选出来。

其实，金融评分卡体系不仅仅是本文申请评分卡这一个。评分卡体系还包括行为评分卡和催收评分卡，本文仅仅只是对申请评分卡做了一个简短的介绍，只是预测客户在开户一定时期内违约的风险概率，有效排除信用不良客户和非目标客户的申请。

其实，建议我们不要只满足与申请评分卡的探究，而要把目光投射到整个互联网金融下的评分卡体系的建立上，通过申请评分卡的建立规律，去探究账户管理前期要建立的行为评分卡和账户管理后期的催收评分卡。

参考目录

- [1] 朱世武.SAS 编程技术教程[M].北京:清华大学出版社,2013.
- [2] 夏坤庄.深入解析 SAS:数据处理、分析优化与商业应用[M].北京:机械工业出版社,2015.
- [3] 谷鸿秋.SAS 编程演义[M].北京:清华大学出版社,2017.
- [4] 陈春宝.SAS 金融数据挖掘与建模系统方法与案例解析[M].北京:机械工业出版社,2017.
- [5] 陈春宝.大数据与机器学习实践方法与行业案例[M].北京:机械工业出版社,2017.
- [6] SAS Institute. SAS 9.4 sql procedure user's guide, third edition [M].SAS Institute, Incorporated, 2015.
- [7] SAS Institute. SAS 9.4 macro language: Reference, second edition [M].SAS Institute, Incorporated, 2014.

附录

附录清单

附录一：原始数据的导入.....	37
附录二：数据集观测变量统计表.....	41
附录三：给定数据的每个数据表格的每个变量的缺失比率.....	42
附录四：求解缺失比率的程序代码.....	47
附录五：消除重复记录的程序代码.....	62
附录六：将关联变量的变量属性统一化.....	63
附录七：横向合并以及求解数据宽表变量缺失比率.....	65
附录八：求解相关系数的代码.....	69
附录九：分类变量的标记.....	70
附录十：缺失值的再处理.....	75
附录十一：变量的标准化.....	78
附录十二：因子分析过程.....	79
附录十三：数据的水平压缩.....	80
附录十四：过度抽样.....	81
附录十五：逻辑回归.....	82
附录十六：模型评估.....	82
附录十七：测试集测试模型.....	83
附录十八：将模型转化为评分卡.....	92

附录一：原始数据的导入

```
/******/  
/* “东证期货杯”全国大学生统计建模大赛初赛*/  
/*作者：重庆理工大学-赵赞豪*/  
/*时间：2017-12-24*/  
/******/  
  
/*由于本案例要求使用原始数据，故数据的准备工作已经初步完成*/  
  
/*==数据的导入==*/  
/*新建永久的数据集：case2，逻辑引用为：“D:\全国大学生统计建模大赛\初赛逻辑库case2”  
*/  
  
/*导入csv格式数据*/  
  
/*导入 “contest_ext_crd_cd_lnd_ovd.csv（机构版征信-贷记卡逾期/透支记录）” */  
PROC IMPORT OUT= CASE2.contest_ext_crd_cd_lnd_ovd  
    DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-  
20171114\contest_ext_crd_cd_lnd_ovd.csv"  
    DBMS=CSV REPLACE;  
    GETNAMES=YES;  
    DATAROW=2;  
RUN;  
  
/*导入 “contest_ext_crd_cd_hd_report.csv（机构版征信-报告主表）” */  
PROC IMPORT OUT= CASE2.contest_ext_crd_hd_report  
    DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-  
20171114\contest_ext_crd_hd_report.csv"  
    DBMS=CSV REPLACE;  
    GETNAMES=YES;  
    DATAROW=2;  
RUN;  
  
/*导入 “contest_ext_crd_is_creditcue.csv（机构版征信-信用提示）” */  
PROC IMPORT OUT= CASE2.contest_ext_crd_is_creditcue  
    DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-  
20171114\contest_ext_crd_is_creditcue.csv"  
    DBMS=CSV REPLACE;  
    GETNAMES=YES;  
    DATAROW=2;  
RUN;
```

```

/*导入“contest_ext_crd_is_ovdsummary.csv（机构版征信-逾期（透支）信息汇总）”
*/
PROC IMPORT OUT= CASE2.contest_ext_crd_is_ovdsummary
    DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_ext_crd_is_ovdsummary.csv"
    DBMS=CSV REPLACE;
    GETNAMES=YES;
    DATAROW=2;
RUN;

/*导入“contest_ext_crd_is_sharedebt.csv（机构版征信-未销户贷记卡或者未结清贷
款）”*/
PROC IMPORT OUT= CASE2.contest_ext_crd_is_sharedebt
    DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_ext_crd_is_sharedebt.csv"
    DBMS=CSV REPLACE;
    GETNAMES=YES;
    DATAROW=2;
RUN;

/*导入tsv格式数据*/

/*导入“contest_basic_test.tsv（基础表-训练集）”*/
PROC IMPORT OUT= CASE2.contest_basic_test
    DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_basic_test.tsv"
    DBMS=TAB REPLACE;
    GETNAMES=YES;
    DATAROW=2;
RUN;

/*导入“contest_basic_train.tsv（基础表-训练集）”*/
PROC IMPORT OUT= CASE2.contest_basic_train
    DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_basic_train.tsv"
    DBMS=TAB REPLACE;
    GETNAMES=YES;
    DATAROW=2;
RUN;

/*导入“contest_basic_train.tsv（基础表-测试集）”*/
PROC IMPORT OUT= CASE2.contest_basic_train
    DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_basic_train.tsv"

```

```

        DBMS=TAB REPLACE;
GETNAMES=YES;
DATAROW=2;
RUN;

/*导入“contest_ext_crd_cd_ln.tsv（机构版征信-贷款）”*/
PROC IMPORT OUT= CASE2.contest_ext_crd_cd_ln
        DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_ext_crd_cd_ln.tsv"
        DBMS=TAB REPLACE;
        GETNAMES=YES;
        DATAROW=2;
RUN;

/*导入“contest_ext_crd_cd_ln_spl.tsv（机构版征信-贷款特殊交易）”*/
PROC IMPORT OUT= CASE2.contest_ext_crd_cd_ln_spl
        DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_ext_crd_cd_ln_spl.tsv"
        DBMS=TAB REPLACE;
        GETNAMES=YES;
        DATAROW=2;
RUN;

/*导入“contest_ext_crd_cd_lnd.tsv（机构版征信-贷记卡）”*/
PROC IMPORT OUT= CASE2.contest_ext_crd_cd_lnd
        DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_ext_crd_cd_lnd.tsv"
        DBMS=TAB REPLACE;
        GETNAMES=YES;
        DATAROW=2;
RUN;

/*导入“contest_ext_crd_qr_recorddtlinfinfo.tsv（机构版征信-信贷审批查询记录明
细）”*/
PROC IMPORT OUT= CASE2.contest_ext_crd_qr_recorddtlinfinfo
        DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_ext_crd_qr_recorddtlinfinfo.tsv"
        DBMS=TAB REPLACE;
        GETNAMES=YES;
        DATAROW=2;
RUN;

/*导入“contest_ext_crd_qr_recordsmr.tsv（机构版征信-查询记录汇总）”*/
PROC IMPORT OUT= CASE2.contest_ext_crd_qr_recordsmr

```

```

        DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_ext_crd_qr_recordsmr.tsv"
        DBMS=TAB REPLACE;
        GETNAMES=YES;
        DATAROW=2;
RUN;

/*导入“contest_fraud.tsv（机构版征信-查询记录汇总）”*/
PROC IMPORT OUT= CASE2.contest_fraud
        DATAFILE= "D:\全国大学生统计建模大赛\初赛\3.原始数据\竞赛数据-
20171114\contest_fraud.tsv"
        DBMS=TAB REPLACE;
        GETNAMES=YES;
        DATAROW=2;
RUN;

```

附录二：数据集观测变量统计表

附表 1 数据集观测变量统计表

数据集的 意义	数据集名称	观测数	变量数
训练集	contest_basic_train	30,000	11
测试集	contest_basic_test	10,000	10
贷款	contest_ext_crd_cd_ln	357,196	22
贷款特殊 交易	contest_ext_crd_cd_ln_spl	67,725	6
贷记卡	contest_ext_crd_cd_lnd	324,229	20
贷记卡透 支记录	contest_ext_crd_cd_lnd_ovd	199,644	4
报告主表	contest_ext_crd_hd_report	40,000	4
信用提示	contest_ext_crd_is_creditcue	39,970	11
透支信息 汇总	contest_ext_crd_is_ovdsummary	76,212	6
未结清贷 款	contest_ext_crd_is_sharedebt	76,246	11
信用审批 查询记录 明细	contest_ext_crd_qr_recorddtlinfo	654,329	4
查询记录 汇总	contest_ext_crd_qr_recordsmr	760	3
反欺诈	contest_fraud	40,000	2

附录三：给定数据的每个数据表格的每个变量的缺失比率

附表 2 训练集 contest_basic_train 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_AGENT	0.7016
2	mr_EDU_LEVEL	0.1019333333
3	mr_HAS_FUND	0.0000666667
4	mr_ID_CARD	0
5	mr_IS_LOCAL	0
6	mr_LOAN_DATE	0
7	mr_MARRY_STATUS	0
8	mr_REPORT_ID	0
9	mr_SALARY	0.7045333333
10	mr_WORK_PROVINCE	0.0752666667
11	mr_Y	0

附表 3 测试集 contest_basic_test 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_AGENT	0.5126
2	mr_EDU_LEVEL	0.2535
3	mr_HAS_FUND	0.1077
4	mr_ID_CARD	0
5	mr_IS_LOCAL	0.0276
6	mr_LOAN_DATE	0
7	mr_MARRY_STATUS	0.0001
8	mr_REPORT_ID	0
9	mr_SALARY	0.553
10	mr_WORK_PROVINCE	0.0667

附表 4 贷款 contest_ext_crd_cd_ln 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_actual_payment_amount	0
2	mr_balance	0
3	mr_class5_state	0
4	mr_credit_limit_amount	0
5	mr_curr_overdue_amount	0
6	mr_curr_overdue_cyc	0
7	mr_currency	0
8	mr_end_date	0
9	mr_finance_org	0
10	mr_guarantee_type	0
11	mr_loan_id	0
12	mr_open_date	0
13	mr_payment_cyc	0.1655169711
14	mr_payment_rating	0
15	mr_payment_state	0
16	mr_recent_pay_date	0
17	mr_remain_payment_cyc	0
18	mr_report_id	0
19	mr_scheduled_payment_amount	0
20	mr_scheduled_payment_date	0
21	mr_state	0
22	mr_type_dw	0

附表 5 贷款特殊交易 contest_ext_crd_cd_ln_spl 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_changing_amount	0
2	mr_changing_months	0
3	mr_content	0
4	mr_get_time	0
5	mr_report_id	0
6	mr_type_dw	0

附表 6 贷记卡 contest_ext_crd_cd_lnd 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_actual_payment_amount	0
2	mr_cardtype	0
3	mr_credit_limit_amount	0
4	mr_curr_overdue_amount	0
5	mr_curr_overdue_cyc	0
6	mr_currency	0
7	mr_finance_org	0
8	mr_guarantee_type	0
9	mr_latest6_month_used_avg_amou	0
10	mr_loancard_id	0
11	mr_open_date	0
12	mr_payment_state	0
13	mr_recent_pay_date	0
14	mr_report_id	0
15	mr_scheduled_payment_amount	0
16	mr_scheduled_payment_date	0
17	mr_share_credit_limit_amount	0
18	mr_state	0
19	mr_used_credit_limit_amount	0
20	mr_used_highest_amount	0

附表 7 贷记卡透支记录 contest_ext_crd_cd_lnd_ovd 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_AMOUNT	0.1047965378
2	mr_LAST_MONTHS	0.1047965378
3	mr_MONTH_DW	0
4	mr_REPORT_ID	0

附表 8 报告主表 contest_ext_crd_hd_report 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_QUERY_ORG	0
2	mr_QUERY_REASON	0
3	mr_REPORT_CREATE_TIME	0
4	mr_REPORT_ID	0

附表 9 信用提示 contest_ext_crd_is_creditcue 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_ANNOUNCE_COUNT	0
2	mr_COMMERCIAL_LOAN_COUNT	0
3	mr_DISSENT_COUNT	0
4	mr_FIRST_LOANCARD_OPEN_MONTH	0
5	mr_FIRST_LOAN_OPEN_MONTH	0
6	mr_FIRST_SL_OPEN_MONTH	0
7	mr_HOUSE_LOAN_COUNT	0
8	mr_LOANCARD_COUNT	0
9	mr_OTHER_LOAN_COUNT	0
10	mr_REPORT_ID	0
11	mr_STANDARD_LOANCARD_COUNT	0

附表 10 透支信息汇总 contest_ext_crd_is_ovdsummary 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_COUNT_DW	0
2	mr_HIGHEST_OA_PER_MON	0
3	mr_MAX_DURATION	0
4	mr_MONTHS	0
5	mr_REPORT_ID	0
6	mr_TYPE_DW	0

附表 11 未结清贷款 contest_ext_crd_is_sharedebt 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_ACCOUNT_COUNT	0
2	mr_BALANCE	0.5575767909
3	mr_CREDIT_LIMIT	0
4	mr_FINANCE_CORP_COUNT	0
5	mr_FINANCE_ORG_COUNT	0
6	mr_LATEST_6M_USED_AVG_AMOUNT	0
7	mr_MAX_CREDIT_LIMIT_PER_ORG	0.4424232091
8	mr_MIN_CREDIT_LIMIT_PER_ORG	0.4424232091
9	mr_REPORT_ID	0
10	mr_TYPE_DW	0
11	mr_USED_CREDIT_LIMIT	0.4424232091

附表 12 信贷审批查询记录明细 contest_ext_crd_qr_recorddtinfo 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_querier	0
2	mr_query_date	0
3	mr_query_reason	0
4	mr_report_id	0

附表 13 查询记录汇总 contest_ext_crd_qr_recordsmr 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_report_id	0
2	mr_sum_dw	0
3	mr_type_id	0

附表 14 反欺诈 contest_fraud 中的每个变量的缺失比率

	以前的变量名	bilv
1	mr_REPORT_ID	0
2	mr_Y_FRAUD	0.246425

附录四：求解缺失比率的程序代码

```
/******/  
/* “东证期货杯”全国大学生统计建模大赛初赛*/  
/*作者：重庆理工大学-赵赞豪*/  
/*时间：2018-1-21*/  
/******/  
  
/*缺失值的处理*/  
  
/*训练集*/  
data _null_;  
set case2.contest_basic_train nobs=total;  
call symput('total',total);  
stop;  
run;  
  
proc contents noprint data=case2.contest_basic_train  
out=contest_basic_train;  
run;  
  
proc sql noprint;  
select distinct name into: var_lst separated by ' ' from  
contest_basic_train;  
quit;  
  
proc sql noprint;  
select count(distinct name) into: var_qty from contest_basic_train;  
quit;  
  
proc sql noprint;  
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from  
contest_basic_train;  
quit;  
  
%macro missing();  
data temp1;  
set case2.contest_basic_train nobs=nobs;  
%do i=1 %to &var_qty;  
if missing(&&var&i) then do;  
m_&&var&i+1;  
end;  
mr_&&var&i=m_&&var&i/&total;  
keep mr_&&var&i;  
%end;
```

```

if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp1 out=temp1;
run;

data temp1;
set temp1;
rename COL1=bilv;
run;
quit;

/*测试集*/
data _null_;
set case2.contest_basic_test nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_basic_test
out=contest_basic_test;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_basic_test;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from contest_basic_test;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_basic_test;
quit;

%macro missing();
data temp2;
set case2.contest_basic_test nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;

```

```

m_&&var&i..1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp2 out=temp2;
run;

data temp2;
set temp2;
rename COL1=bilv;
run;
quit;

/*贷款*/
data _null_;
set case2.contest_ext_crd_cd_ln nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_ext_crd_cd_ln
out=contest_ext_crd_cd_ln;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_ext_crd_cd_ln;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from contest_ext_crd_cd_ln;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_ext_crd_cd_ln;
quit;

```

```

%macro missing();
data temp3;
set case2.contest_ext_crd_cd_ln nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp3 out=temp3;
run;

data temp3;
set temp3;
rename COL1=bilv;
run;
quit;

/*贷款特殊交易*/
data _null_;
set case2.contest_ext_crd_cd_ln_spl nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_ext_crd_cd_ln_spl
out=contest_ext_crd_cd_ln_spl;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_ext_crd_cd_ln_spl;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from
contest_ext_crd_cd_ln_spl;
quit;

```

```

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_ext_crd_cd_ln_spl;
quit;

%macro missing();
data temp4;
set case2.contest_ext_crd_cd_ln_spl nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp4 out=temp4;
run;

data temp4;
set temp4;
rename COL1=bilv;
run;
quit;

/*贷记卡*/
data _null_;
set case2.contest_ext_crd_cd_lnd nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_ext_crd_cd_lnd
out=contest_ext_crd_cd_lnd;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_ext_crd_cd_lnd;

```

```

quit;

proc sql noprint;
select count(distinct name) into: var_qty from contest_ext_crd_cd_lnd;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_ext_crd_cd_lnd;
quit;

%macro missing();
data temp5;
set case2.contest_ext_crd_cd_lnd nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp5 out=temp5;
run;

data temp5;
set temp5;
rename COL1=bilv;
run;
quit;

/*贷记卡透支记录*/
data _null_;
set case2.contest_ext_crd_cd_lnd_ovd nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_ext_crd_cd_lnd_ovd
out=contest_ext_crd_cd_lnd_ovd;

```



```

run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_ext_crd_cd_lnd_ovd;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from
contest_ext_crd_cd_lnd_ovd;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_ext_crd_cd_lnd_ovd;
quit;

%macro missing();
data temp6;
set case2.contest_ext_crd_cd_lnd_ovd nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp6 out=temp6;
run;

data temp6;
set temp6;
rename COL1=bilv;
run;
quit;

/*报告主表*/
data _null_;
set case2.contest_ext_crd_hd_report nobs=total;

```

```

call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_ext_crd_hd_report
out=contest_ext_crd_hd_report;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_ext_crd_hd_report;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from
contest_ext_crd_hd_report;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_ext_crd_hd_report;
quit;

%macro missing();
data temp7;
set case2.contest_ext_crd_hd_report nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp7 out=temp7;
run;

data temp7;
set temp7;
rename COL1=bilv;

```

```

run;
quit;

/*信用提示*/
data _null_;
set case2.contest_ext_crd_is_creditcue nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_ext_crd_is_creditcue
out=contest_ext_crd_is_creditcue;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_ext_crd_is_creditcue;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from
contest_ext_crd_is_creditcue;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_ext_crd_is_creditcue;
quit;

%macro missing();
data temp8;
set case2.contest_ext_crd_is_creditcue nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

```

```

proc transpose data=temp8 out=temp8;
run;

data temp8;
set temp8;
rename COL1=bilv;
run;
quit;

/*透支信息汇总*/
data _null_;
set case2.contest_ext_crd_is_ovdsummary nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_ext_crd_is_ovdsummary
out=contest_ext_crd_is_ovdsummary;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_ext_crd_is_ovdsummary;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from
contest_ext_crd_is_ovdsummary;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_ext_crd_is_ovdsummary;
quit;

%macro missing();
data temp9;
set case2.contest_ext_crd_is_ovdsummary nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;

```

```

%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp9 out=temp9;
run;

data temp9;
set temp9;
rename COL1=bilv;
run;
quit;

/*未结清贷款*/
data _null_;
set case2.contest_ext_crd_is_sharedebt nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_ext_crd_is_sharedebt
out=contest_ext_crd_is_sharedebt;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_ext_crd_is_sharedebt;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from
contest_ext_crd_is_sharedebt;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_ext_crd_is_sharedebt;
quit;

%macro missing();
data temp10;
set case2.contest_ext_crd_is_sharedebt nobs=nobs;

```

```

%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp10 out=temp10;
run;

data temp10;
set temp10;
rename COL1=bilv;
run;
quit;

/*信贷审批查询记录明细*/
data _null_;
set case2.contest_ext_crd_qr_recorddtlinfo nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_ext_crd_qr_recorddtlinfo
out=contest_ext_crd_qr_recorddtlinfo;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_ext_crd_qr_recorddtlinfo;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from
contest_ext_crd_qr_recorddtlinfo;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from

```

```

contest_ext_crd_qr_recorddtlinfo;
quit;

%macro missing();
data temp11;
set case2.contest_ext_crd_qr_recorddtlinfo nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp11 out=temp11;
run;

data temp11;
set temp11;
rename COL1=bilv;
run;
quit;

/*查询记录汇总*/
data _null_;
set case2.contest_ext_crd_qr_recordsmr nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_ext_crd_qr_recordsmr
out=contest_ext_crd_qr_recordsmr;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from
contest_ext_crd_qr_recordsmr;
quit;

proc sql noprint;

```

```

select count(distinct name) into: var_qty from
contest_ext_crd_qr_recordsmr;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_ext_crd_qr_recordsmr;
quit;

%macro missing();
data temp12;
set case2.contest_ext_crd_qr_recordsmr nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp12 out=temp12;
run;

data temp12;
set temp12;
rename COL1=bilv;
run;
quit;

/*反欺诈*/
data _null_;
set case2.contest_fraud nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.contest_fraud out=contest_fraud;
run;

proc sql noprint;

```



```

select distinct name into: var_lst separated by ' ' from contest_fraud;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from contest_fraud;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
contest_fraud;
quit;

%macro missing();
data temp13;
set case2.contest_fraud nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp13 out=temp13;
run;

data temp13;
set temp13;
rename COL1=bilv;
run;
quit;
/*删除work逻辑库中的所有数据集和宏程序*/
/*
proc catalog catalog=work.sasmacr force kill;
run;
quit;
proc datasets library=work nolist nodetails kill;
run;
quit;

*/

```

附录五：消除重复记录的程序代码

```
/*排序,并且删除重复观测*/
proc sort data=case2.contest_basic_train nodups;
by report_id;
run;
proc sort data=case2.contest_basic_test nodups;
by report_id;
run;
proc sort data=case2.contest_ext_crd_cd_ln nodups;
by report_id;
run;
proc sort data=case2.contest_ext_crd_cd_ln_spl nodups;
by report_id;
run;
proc sort data=case2.contest_ext_crd_cd_lnd nodups;
by report_id;
run;
proc sort data=case2.contest_ext_crd_cd_lnd_ovd nodups;
by report_id;
run;
proc sort data=case2.contest_ext_crd_hd_report nodups;
by report_id;
run;
proc sort data=case2.contest_ext_crd_is_creditcue nodups;
by report_id;
run;
proc sort data=case2.contest_ext_crd_is_ovdsummary nodups;
by report_id;
run;
proc sort data=case2.contest_ext_crd_is_sharedebt nodups;
by report_id;
run;
proc sort data=case2.contest_ext_crd_qr_recorddtlinfinfo nodups;
by report_id;
run;
proc sort data=case2.contest_ext_crd_qr_recordsmr nodups;
by report_id;
run;
proc sort data=case2.contest_fraud nodups;
by report_id;

run;
```

附录六：将关联变量的变量属性统一化

```
/*更改report_id的属性*/

/*转化变量的属性*/

data case2.contest_ext_crd_cd_lnd_ovd;
set case2.contest_ext_crd_cd_lnd_ovd;
report_id1=input(report_id,12.);
drop report_id;
run;

data case2.contest_ext_crd_cd_lnd_ovd;
set case2.contest_ext_crd_cd_lnd_ovd;
report_id=report_id1;
run;

data case2.contest_ext_crd_cd_lnd_ovd;
set case2.contest_ext_crd_cd_lnd_ovd;
drop report_id1;
run;

/*****/
data case2.contest_ext_crd_is_creditcue;
set case2.contest_ext_crd_is_creditcue;
report_id1=input(report_id,12.);
drop report_id;
run;

data case2.contest_ext_crd_is_creditcue;
set case2.contest_ext_crd_is_creditcue;
report_id=report_id1;
run;

data case2.contest_ext_crd_is_creditcue;
set case2.contest_ext_crd_is_creditcue;
drop report_id1;
run;

/*****/

data case2.contest_ext_crd_is_ovdsummary;
set case2.contest_ext_crd_is_ovdsummary;
report_id1=input(report_id,12.);
```

```

drop report_id;
run;

data case2.contest_ext_crd_is_ovdsummary;
set case2.contest_ext_crd_is_ovdsummary;
report_id=report_id1;
run;

data case2.contest_ext_crd_is_ovdsummary;
set case2.contest_ext_crd_is_ovdsummary;
drop report_id1;
run;
/*****/

data case2.contest_ext_crd_is_sharedebt;
set case2.contest_ext_crd_is_sharedebt;
report_id1=input(report_id,12.);
drop report_id;
run;

data case2.contest_ext_crd_is_sharedebt;
set case2.contest_ext_crd_is_sharedebt;
report_id=report_id1;
run;

data case2.contest_ext_crd_is_sharedebt;
set case2.contest_ext_crd_is_sharedebt;
drop report_id1;
run;
/*****/

proc sort data=case2.contest_ext_crd_cd_lnd_ovd;
by report_id;
run;
proc sort data=case2.contest_ext_crd_is_creditcue;
by report_id;
run;
proc sort data=case2.contest_ext_crd_is_ovdsummary;
by report_id;
run;
proc sort data=case2.contest_ext_crd_is_sharedebt;
by report_id;
run;

```

附录七：横向合并以及求解数据宽表变量缺失比率

```
/*数据合并*/

/*训练集的数据合并*/
data case2.train;
merge case2.contest_basic_train
      case2.contest_ext_crd_cd_ln
      case2.contest_ext_crd_cd_ln_spl
      case2.contest_ext_crd_cd_lnd
      case2.contest_ext_crd_cd_lnd_ovd
      case2.contest_ext_crd_hd_report
      case2.contest_ext_crd_is_creditcue
      case2.contest_ext_crd_is_ovdsummary
      case2.contest_ext_crd_is_sharedebt
      case2.contest_ext_crd_qr_recorddtlinf
      case2.contest_ext_crd_qr_recordsmr
      ;
by report_id;
run;

/*测试集的数据合并*/
data case2.test;
merge case2.contest_basic_test
      case2.contest_ext_crd_cd_ln
      case2.contest_ext_crd_cd_ln_spl
      case2.contest_ext_crd_cd_lnd
      case2.contest_ext_crd_cd_lnd_ovd
      case2.contest_ext_crd_hd_report
      case2.contest_ext_crd_is_creditcue
      case2.contest_ext_crd_is_ovdsummary
      case2.contest_ext_crd_is_sharedebt
      case2.contest_ext_crd_qr_recorddtlinf
      case2.contest_ext_crd_qr_recordsmr
      ;
by report_id;
run;

/*训练集中删除不必要的观测*/
proc sql;
create table case2.train1 as select * from case2.train where loan_date~=. ;
quit;
/*测试集中删除不必要的观测*/
```

```

proc sql;
create table case2.test1 as select * from case2.test where loan_date~=.;
quit;

/*排序,并且删除重复观测*/
proc sort data=case2.train1 out=train1 nodups;
by report_id;
run;

proc sort data=case2.test1 out=test1 nodups;
by report_id;
run;

/*训练集的缺失状况*/
data _null_;
set case2.train1 nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.train1 out=train1;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from train1;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from train1;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
train1;
quit;

%macro missing();
data temp;
set case2.train1 nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i+1;
end;
mr_&&var&i=m_&&var&i/&total;

```

```

keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp out=temp;
run;

data temp;
set temp;
rename COL1=bilv;
run;
quit;

/*测试集的缺失状况*/
data _null_;
set case2.test1 nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.test1 out=test1;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from test1;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from test1;
quit;

proc sql noprint;
select distinct name into: var1 -: %cmpres(var%eval(&var_qty+0)) from
test1;
quit;

%macro missing();
data tmp;
set case2.test1 nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;

```

```

m_&&var&i..1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=tmp out=tmp;
run;

data tmp;
set tmp;
rename COL1=bilv;
run;
quit;

/*选定删除变量*/
data temp;
set temp;
if bilv>=0.2 then do beizhu='建议删';
end;
run;

data tmp;
set tmp;
if bilv>=0.2 then do beizhu='建议删';
end;

run;

proc sql;
create table temp1 as select * from temp where beizhu='建议删';
run;

proc sql;
create table tmp1 as select * from tmp where beizhu='建议删';
run;

quit;
/*删除work逻辑库中的所有数据集和宏程序*/

proc catalog catalog=work.sasmacr force kill;

```



```

run;
quit;

proc datasets library=work nolist nodetails kill;
run;
quit;

/*删除缺失率高的变量*/
data case2.train2;
set case2.train1;
drop agent amount count_dw highest_oa_per_mon last_months max_duration
months
      month_dw salary changing_amount changing_months content get_time
payment_cyc sum_dw type_id;
run;

data case2.test2;
set case2.test1;
drop agent amount count_dw highest_oa_per_mon last_months max_duration
months
      month_dw salary changing_amount changing_months content get_time
payment_cyc sum_dw type_id;
run;

```

附录八：求解相关系数的代码

```

a<-read.csv("C:/Users/Administrator/Desktop/train(quan1).csv",header=
T)
b3<-a[,3]
b2<-a[,12:33]
b4<-cbind(b3,b2)
bxiang<-round(cor(b4),2)
bxiang
b2<-a[,2]
c2<-a[,4:11]
c3<-cbind(b2,c2)
cxiang<-round(cor(c3),2)
cxiang
d1<-a[,2:33]
dxiang<-round(cor(d1),2)
e1<-dxiang[1:11,3:32]

```

```

e1
b2<-dxiang[1,]
e3<-dxiang[3:11,]
e4<-rbind(b2,e3)
e4
b3<-e4[,2]
e5<-e4[,12:32]
e6<-cbind(b3,e5)
e6
max(e6)
min(e6)

```

附录九：分类变量的标记

```

/*****
/***** 全国大学生统计建模大赛 *****/
/***** 时间：2018年1月26号 *****/
/***** 作者：赵赞豪 *****/
/**** 分类变量的标记 ****/
data train2;
set case2.train2;

/*分类变量：是否本地籍*/
if IS_LOCAL='本地籍' then do IS_LOCAL=1;end;
if IS_LOCAL='非本地籍' then do IS_LOCAL=0;end;

/*分类变量：教育*/
if EDU_LEVEL='博士研究生' then do EDU_LEVEL=1;end;
if EDU_LEVEL='硕士及以上' then do EDU_LEVEL=2;end;
if EDU_LEVEL='硕士研究生' then do EDU_LEVEL=3;end;
if EDU_LEVEL='本科' then do EDU_LEVEL=4;end;
if EDU_LEVEL='专科' then do EDU_LEVEL=5;end;
if EDU_LEVEL='专科及以下' then do EDU_LEVEL=6;end;
if EDU_LEVEL='高中' then do EDU_LEVEL=7;end;
if EDU_LEVEL='初中' then do EDU_LEVEL=8;end;
if EDU_LEVEL=' ' then do EDU_LEVEL=9;end;

/*分类变量：婚姻*/
if MARRY_STATUS='已婚' then do MARRY_STATUS=1;end;

```

```

if MARRY_STATUS='未婚' then do MARRY_STATUS=2;end;
if MARRY_STATUS='离婚' then do MARRY_STATUS=3;end;
if MARRY_STATUS='离异' then do MARRY_STATUS=4;end;
if MARRY_STATUS='丧偶' then do MARRY_STATUS=5;end;
if MARRY_STATUS='其他' then do MARRY_STATUS=6;end;

```

/*分类变量：账户状态*/

```

if state='呆账' then do state=1;end;
if state='冻结' then do state=2;end;
if state='结清' then do state=3;end;
if state='未激' then do state=4;end;
if state='销户' then do state=5;end;
if state='逾期' then do state=6;end;
if state='正常' then do state=7;end;
if state='止付' then do state=8;end;
if state=' ' then do state=9;end;

```

/*分类变量：贷款机构（不处理）*/

/*分类变量：贷款种类细分（不处理）*/

/*分类变量：币种*/

```

if currency='澳大利' then do currency=1;end;
if currency='港元' then do currency=2;end;
if currency='加拿大' then do currency=3;end;
if currency='美元' then do currency=4;end;
if currency='欧元' then do currency=5;end;
if currency='人民币' then do currency=6;end;
if currency='日元' then do currency=7;end;
if currency='瑞士法' then do currency=8;end;
if currency='英镑' then do currency=9;end;
if currency=' ' then do currency=10;end;

```

/*分类变量：担保方式*/

```

if guarantee_type='保证' then do guarantee_type=1;end;
if guarantee_type='抵押担保' then do guarantee_type=2;end;
if guarantee_type='农户联保' then do guarantee_type=3;end;
if guarantee_type='其他担保' then do guarantee_type=4;end;
if guarantee_type='信用/免担保' then do guarantee_type=5;end;
if guarantee_type='质押（含保证金）担保' then do guarantee_type=6;end;
if guarantee_type='组合（不含保证）担保' then do guarantee_type=7;end;
if guarantee_type='组合（不含 ' then do guarantee_type=7;end;
if guarantee_type='组合（含保证）担保' then do guarantee_type=8;end;
if guarantee_type=' ' then do guarantee_type=9;end;

```

```

/*分类变量：还款频率*/
if payment_rating='按半年归' then do payment_rating=1;end;
if payment_rating='按季归还' then do payment_rating=2;end;
if payment_rating='按年归还' then do payment_rating=3;end;
if payment_rating='按其他方' then do payment_rating=4;end;
if payment_rating='按日归还' then do payment_rating=5;end;
if payment_rating='不定期归' then do payment_rating=6;end;
if payment_rating='一次性归' then do payment_rating=7;end;
if payment_rating='一次性归' then do payment_rating=8;end;
if payment_rating='按月归还' then do payment_rating=9;end;
if payment_rating=' ' then do payment_rating=10;end;

/*分类变量：五级分类*/
if class5_state='正常' then do class5_state=1;end;
if class5_state='未知' then do class5_state=2;end;
if class5_state='关注' then do class5_state=3;end;
if class5_state='NULL' then do class5_state=4;end;
if class5_state=' ' then do class5_state=4;end;

/*分类变量：卡类型*/
if cardtype='贷记卡' then do cardtype=1;end;
if cardtype=' ' then do cardtype=2;end;
if cardtype='NULL' then do cardtype=2;end;

/*分类变量：查询原因*/
if QUERY_REASON='贷款审批' then do QUERY_REASON=1;end;
if QUERY_REASON='担保资格' then do QUERY_REASON=2;end;
if QUERY_REASON='信用卡审' then do QUERY_REASON=3;end;

run;

data test2;
set case2.test2;

/*分类变量：是否本地籍*/
if IS_LOCAL='本地籍' then do IS_LOCAL=1;end;
if IS_LOCAL='非本地籍' then do IS_LOCAL=0;end;

/*分类变量：教育*/
if EDU_LEVEL='博士研究生' then do EDU_LEVEL=1;end;
if EDU_LEVEL='硕士及以上' then do EDU_LEVEL=2;end;
if EDU_LEVEL='硕士研究生' then do EDU_LEVEL=3;end;
if EDU_LEVEL='本科' then do EDU_LEVEL=4;end;

```

```

if EDU_LEVEL='专科'      then do EDU_LEVEL=5;end;
if EDU_LEVEL='专科及以下' then do EDU_LEVEL=6;end;
if EDU_LEVEL='高中'      then do EDU_LEVEL=7;end;
if EDU_LEVEL='初中'      then do EDU_LEVEL=8;end;
if EDU_LEVEL=' '         then do EDU_LEVEL=9;end;

```

/*分类变量：婚姻*/

```

if MARRY_STATUS='已婚' then do MARRY_STATUS=1;end;
if MARRY_STATUS='未婚' then do MARRY_STATUS=2;end;
if MARRY_STATUS='离婚' then do MARRY_STATUS=3;end;
if MARRY_STATUS='离异' then do MARRY_STATUS=4;end;
if MARRY_STATUS='丧偶' then do MARRY_STATUS=5;end;
if MARRY_STATUS='其他' then do MARRY_STATUS=6;end;

```

/*分类变量：账户状态*/

```

if state='呆账' then do state=1;end;
if state='冻结' then do state=2;end;
if state='结清' then do state=3;end;
if state='未激' then do state=4;end;
if state='销户' then do state=5;end;
if state='逾期' then do state=6;end;
if state='正常' then do state=7;end;
if state='止付' then do state=8;end;
if state=' '    then do state=9;end;

```

/*分类变量：贷款机构（不处理）*/

/*分类变量：贷款种类细分（不处理）*/

/*分类变量：币种*/

```

if currency='澳大利' then do currency=1;end;
if currency='港元'   then do currency=2;end;
if currency='加拿大' then do currency=3;end;
if currency='美元'   then do currency=4;end;
if currency='欧元'   then do currency=5;end;
if currency='人民币' then do currency=6;end;
if currency='日元'   then do currency=7;end;
if currency='瑞士法' then do currency=8;end;
if currency='英镑'   then do currency=9;end;
if currency=' '      then do currency=10;end;

```

/*分类变量：担保方式*/

```

if guarantee_type='保证' then do guarantee_type=1;end;
if guarantee_type='抵押担保' then do guarantee_type=2;end;

```

```

if guarantee_type='农户联保' then do guarantee_type=3;end;
if guarantee_type='其他担保' then do guarantee_type=4;end;
if guarantee_type='信用/免担保' then do guarantee_type=5;end;
if guarantee_type='质押（含保证金）担保' then do guarantee_type=6;end;
if guarantee_type='组合（不含保证）担保' then do guarantee_type=7;end;
if guarantee_type='组合（不含 ' then do guarantee_type=7;end;
if guarantee_type='组合（含保证）担保' then do guarantee_type=8;end;
if guarantee_type=' ' then do guarantee_type=9;end;

```

/*分类变量：还款频率*/

```

if payment_rating='按半年归' then do payment_rating=1;end;
if payment_rating='按季归还' then do payment_rating=2;end;
if payment_rating='按年归还' then do payment_rating=3;end;
if payment_rating='按其他方' then do payment_rating=4;end;
if payment_rating='按日归还' then do payment_rating=5;end;
if payment_rating='不定期归' then do payment_rating=6;end;
if payment_rating='一次性归' then do payment_rating=7;end;
if payment_rating='一次性归' then do payment_rating=8;end;
if payment_rating='按月归还' then do payment_rating=9;end;
if payment_rating=' ' then do payment_rating=10;end;

```

/*分类变量：五级分类*/

```

if class5_state='正常' then do class5_state=1;end;
if class5_state='未知' then do class5_state=2;end;
if class5_state='关注' then do class5_state=3;end;
if class5_state='NULL' then do class5_state=4;end;
if class5_state=' ' then do class5_state=4;end;

```

/*分类变量：卡类型*/

```

if cardtype='贷记卡' then do cardtype=1;end;
if cardtype=' ' then do cardtype=2;end;
if cardtype='NULL' then do cardtype=2;end;

```

/*分类变量：查询原因*/

```

if QUERY_REASON='贷款审批' then do QUERY_REASON=1;end;
if QUERY_REASON='担保资格' then do QUERY_REASON=2;end;
if QUERY_REASON='信用卡审' then do QUERY_REASON=3;end;

```

run;

```

PROC EXPORT DATA= WORK.Train2
    OUTFILE= "C:\Users\asus-\Desktop\train.csv"
    DBMS=CSV REPLACE;
    PUTNAMES=YES;

```

```

RUN;

PROC EXPORT DATA= WORK.Test2
    OUTFILE= "C:\Users\asus-\Desktop\test.csv"
    DBMS=CSV REPLACE;
    PUTNAMES=YES;
RUN;

```

附录十：缺失值的再处理

```

proc sql;
create table case2.train_d1 as select
REPORT_ID,
IS_LOCAL,
WORK_PROVINCE,
EDU_LEVEL,
MARRY_STATUS,
HAS_FUND,
Y,
state,
currency,
guarantee_type,
payment_rating,
class5_state,
credit_limit_amount,
balance,
remain_payment_cyc,
scheduled_payment_amount,
actual_payment_amount,
curr_overdue_cyc,
curr_overdue_amount,
share_credit_limit_amount,
used_credit_limit_amount,
latest6_month_used_avg_amount,
used_highest_amount,
HOUSE_LOAN_COUNT,
COMMERCIAL_LOAN_COUNT,
OTHER_LOAN_COUNT,
LOANCARD_COUNT,
STANDARD_LOANCARD_COUNT,

```

```

FINANCE_CORP_COUNT,
FINANCE_ORG_COUNT,
ACCOUNT_COUNT,
CREDIT_LIMIT,
MAX_CREDIT_LIMIT_PER_ORG,
MIN_CREDIT_LIMIT_PER_ORG,
USED_CREDIT_LIMIT,
LATEST_6M_USED_AVG_AMOUNT

from case2.train_d;
run;
quit;

/*删除重复值*/
proc sort data=case2.train_d1 nodups;
by report_id;
run;

/*查询数据集case2.train_d1中的变量的缺失比率*/
data _null_;
set case2.train_d1 nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.train_d1 out=train_d1;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from train_d1;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from train_d1;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
train_d1;
quit;

%macro missing();
data temp1;
set case2.train_d1 nobs=nobs;

```



```

%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp1 out=temp1;
run;

data temp1;
set temp1;
rename COL1=bilv;
run;
quit;

/*找出缺失比率大于0.2的变量，输出到数据集queshi中*/
proc sql;
create table queshi as select * from temp1 where bilv>=0.2;
run;
quit;

/*****
***          缺失值的处理          ***
*****/

data case2.train_d1;
set case2.train_d1;
drop balance remain_payment_cyc;
if work_province=. then do work_province=330100;end;
if edu_level=. then do edu_level=9;end;
if has_fund=. then do has_fund=0;end;
if state=. then do state=9;end;
if currency=. then do currency=10;end;
if guarantee_type=. then do guarantee_type=9;end;
if class5_state=. then do class5_state=4;end;

```

```

if credit_limit_amount=. then do credit_limit_amount=0;end;
if scheduled_payment_amount=. then do scheduled_payment_amount=0;end;
if actual_payment_amount=. then do actual_payment_amount=0;end;
if curr_overdue_cyc=. then do curr_overdue_cyc=0;end;
if curr_overdue_amount=. then do curr_overdue_amount=0;end;
if share_credit_limit_amount=. then do share_credit_limit_amount=0;end;
if used_credit_limit_amount=. then do
used_credit_limit_amount=2767;end;
if latest6_month_used_avg_amount=. then do
latest6_month_used_avg_amount=2913;end;
if HOUSE_LOAN_COUNT=. then do HOUSE_LOAN_COUNT=0;end;
if COMMERCIAL_LOAN_COUNT=. then do COMMERCIAL_LOAN_COUNT=0;end;
if OTHER_LOAN_COUNT=. then do OTHER_LOAN_COUNT=0;end;

run;

proc stdize data=case2.train_d1 reponly method=median
out=case2.train_d2;
var used_highest_amount LOANCARD_COUNT STANDARD_LOANCARD_COUNT
FINANCE_CORP_COUNT FINANCE_ORG_COUNT ACCOUNT_COUNT
CREDIT_LIMIT
MAX_CREDIT_LIMIT_PER_ORG
MIN_CREDIT_LIMIT_PER_ORG
USED_CREDIT_LIMIT
LATEST_6M_USED_AVG_AMOUNT;
run;

proc sort data=case2.train_d2 nodups;
by report_id;
run;

```

附录十一：变量的标准化

```

/*将数据标准化处理*/
proc standard data=case2.train_d2 out=case2.train_std mean=0 std=1;
var
IS_LOCAL
WORK_PROVINCE
EDU_LEVEL
MARRY_STATUS
HAS_FUND
state

```

```

currency
guarantee_type
payment_rating
class5_state
credit_limit_amount
scheduled_payment_amount
actual_payment_amount
curr_overdue_cyc
curr_overdue_amount
share_credit_limit_amount
used_credit_limit_amount
latest6_month_used_avg_amount
used_highest_amount
HOUSE_LOAN_COUNT
COMMERCIAL_LOAN_COUNT
OTHER_LOAN_COUNT
LOANCARD_COUNT
STANDARD_LOANCARD_COUNT
FINANCE_CORP_COUNT
FINANCE_ORG_COUNT
ACCOUNT_COUNT
CREDIT_LIMIT
MAX_CREDIT_LIMIT_PER_ORG
MIN_CREDIT_LIMIT_PER_ORG
USED_CREDIT_LIMIT
LATEST_6M_USED_AVG_AMOUNT
;

run;

```

附录十二：因子分析过程

```

/*因子分析*/
proc factor data=case2.train_std out=case2.train_yin nfactors=15 score
scre;
var
IS_LOCAL
WORK_PROVINCE
EDU_LEVEL
MARRY_STATUS
HAS_FUND

```

```

state
currency
guarantee_type
payment_rating
class5_state
credit_limit_amount
scheduled_payment_amount
actual_payment_amount
curr_overdue_cyc
curr_overdue_amount
share_credit_limit_amount
used_credit_limit_amount
latest6_month_used_avg_amount
used_highest_amount
HOUSE_LOAN_COUNT
COMMERCIAL_LOAN_COUNT
OTHER_LOAN_COUNT
LOANCARD_COUNT
STANDARD_LOANCARD_COUNT
FINANCE_CORP_COUNT
FINANCE_ORG_COUNT
ACCOUNT_COUNT
CREDIT_LIMIT
MAX_CREDIT_LIMIT_PER_ORG
MIN_CREDIT_LIMIT_PER_ORG
USED_CREDIT_LIMIT
LATEST_6M_USED_AVG_AMOUNT
;
run;

```

附录十三：数据的水平压缩

```

/*数据的水平压缩*/
proc sql;
create table case2.train_yin_ya as select report_id , y ,
avg(factor1) as x1,
avg(factor2) as x2,
avg(factor3) as x3,
avg(factor4) as x4,
avg(factor5) as x5,

```

```

avg(factor6) as x6,
avg(factor7) as x7,
avg(factor8) as x8,
avg(factor9) as x9,
avg(factor10) as x10,
avg(factor11) as x11,
avg(factor12) as x12,
avg(factor13) as x13,
avg(factor14) as x14,
avg(factor15) as x15
from case2.train_yin group by report_id;
run;

quit;

```

附录十四：过度抽样

```

/****过度抽样*****/
proc sort data=case2.train_yin_ya out=case2.train_yin_ya1 nodups; /*将
case2.train_yin_ya中的重复观测删除，并且输出到case2.train_yin_ya1数据集*/
by report_id;
run;

proc sort data=case2.train_yin_ya1 out=case2.train_yin_ya2; /*将数据集
case2.train_yin_ya1按y排序，输出到数据集case2.train_yin_ya2中*/
by y;
run;

proc surveyselect data=case2.train_yin_ya2 out=case2.train_yin_ya2
seed=1234 samprate=(0.24,1); /*对数据集case2.train_yin_ya2进行过度抽样*/
strata y; /*y为分层抽样类别的变量*/

run;

```

附录十五：逻辑回归

```
/*进行逻辑回归*/
proc logistic data=case2.train_yin_ya2 out=case2.logi ;/*对数据集
case2.train_yin_ya2进行逻辑回归，将结果输出到case2.logi数据集中*/
model y(event="1")= x1-x15/selection=stepwise ;/*逐步回归剔除变量*/
score data=case2.train_yin_ya2 out=case2.train_score_r ;/*模型估计*/

run;
```

附录十六：模型评估

```
proc logistic data=case2.train_yin_ya2 plots=(roc);
model y=x1 x3 x6 x8 x10 x11 x14 x15 ;
run;
```

```
proc npar1way data=case2.train_score_r edf;
class y;
var P_1;
run;
```

```
/*计算所有的预测值y*/
data case2.train_suoyou_score;
set case2.train_yin_ya1;
f= -1.8950 -0.5265*x1 + 0.2866*x3 -0.0809*x6 -0.4095*x8 -0.6236*x10 +
0.1861*x11 -0.1338*x14 -0.1933*x15;
p=exp(f)/(1+exp(f));
if p>=0.5 then do p_y=1;end;
if p<0.5 then do p_y=0;end;
run;
```

附录十七：测试集测试模型

/*测试集依据建立好的模型预测响应变量的值*/

```
proc sql;
create table case2.test_d1 as select
report_id,
IS_LOCAL,
WORK_PROVINCE,
EDU_LEVEL,
MARRY_STATUS,
HAS_FUND,
state,
currency,
guarantee_type,
payment_rating,
class5_state,
credit_limit_amount,
scheduled_payment_amount,
actual_payment_amount,
curr_overdue_cyc,
curr_overdue_amount,
share_credit_limit_amount,
used_credit_limit_amount,
latest6_month_used_avg_amount,
used_highest_amount,
HOUSE_LOAN_COUNT,
COMMERCIAL_LOAN_COUNT,
OTHER_LOAN_COUNT,
LOANCARD_COUNT,
STANDARD_LOANCARD_COUNT,
FINANCE_CORP_COUNT,
FINANCE_ORG_COUNT,
ACCOUNT_COUNT,
CREDIT_LIMIT,
MAX_CREDIT_LIMIT_PER_ORG,
MIN_CREDIT_LIMIT_PER_ORG,
USED_CREDIT_LIMIT,
LATEST_6M_USED_AVG_AMOUNT
from case2.test_d;
run;
quit;
```

/*查看数据集中是否还有字符型变量*/

```

proc contents data=case2.test_d1;
run;

/*首先，删除重复观测*/
proc sort data=case2.test_d1 nodups;
by report_id;
run;

/*然后，将字符型变量变成数值型变量*/
data case2.test_d2;
set case2.test_d1;
actual_payment_amount1=input(actual_payment_amount,comma8.);
curr_overdue_cyc1=input(curr_overdue_cyc,comma8.);
curr_overdue_amount1=input(curr_overdue_amount,comma8.);
latest6_month_used_avg_amount1=input(latest6_month_used_avg_amount,comma8.);
scheduled_payment_amount1=input(scheduled_payment_amount,comma8.);
used_credit_limit_amount1=input(used_credit_limit_amount,comma8.);
used_highest_amount1=input(used_highest_amount,comma8.);
run;

data case2.test_d2;
set case2.test_d2;
drop
actual_payment_amount
curr_overdue_cyc
curr_overdue_amount
latest6_month_used_avg_amount
scheduled_payment_amount
used_credit_limit_amount
used_highest_amount;
run;

data case2.test_d2;
set case2.test_d2;
rename
actual_payment_amount1=actual_payment_amount
curr_overdue_cyc1=curr_overdue_cyc
curr_overdue_amount1=curr_overdue_amount
latest6_month_used_avg_amount1=latest6_month_used_avg_amount
scheduled_payment_amount1=scheduled_payment_amount
used_credit_limit_amount1=used_credit_limit_amount
used_highest_amount1=used_highest_amount
;

```



```
run;
```

```
/*然后，再将缺失值补全，方法按照训练集的方法*/
```

```
/*人工填补*/
```

```
data case2.test_d3;  
set case2.test_d2;  
if IS_LOCAL=. then do IS_LOCAL=0;end;  
if MARRY_STATUS=. then do MARRY_STATUS=6;end;  
if payment_rating=. then do payment_rating=10;end;  
if edu_level=. then do edu_level=9;end;  
if has_fund=. then do has_fund=0;end;  
if state=. then do state=9;end;  
if currency=. then do currency=10;end;  
if guarantee_type=. then do guarantee_type=9;end;  
if class5_state=. then do class5_state=4;end;  
if credit_limit_amount=. then do credit_limit_amount=0;end;  
if scheduled_payment_amount=. then do scheduled_payment_amount=0;end;  
if actual_payment_amount=. then do actual_payment_amount=0;end;  
if curr_overdue_cyc=. then do curr_overdue_cyc=0;end;  
if curr_overdue_amount=. then do curr_overdue_amount=0;end;  
if share_credit_limit_amount=. then do share_credit_limit_amount=0;end;  
  
if HOUSE_LOAN_COUNT=. then do HOUSE_LOAN_COUNT=0;end;  
if COMMERCIAL_LOAN_COUNT=. then do COMMERCIAL_LOAN_COUNT=0;end;  
if OTHER_LOAN_COUNT=. then do OTHER_LOAN_COUNT=0;end;
```

```
run;
```

```
/*中位数*/
```

```
proc stdize data=case2.test_d3 reponly method=median out=case2.test_d3;  
var
```

```
work_province  
used_credit_limit_amount  
latest6_month_used_avg_amount  
  
used_highest_amount  
LOANCARD_COUNT STANDARD_LOANCARD_COUNT  
FINANCE_CORP_COUNT  
FINANCE_ORG_COUNT  
ACCOUNT_COUNT  
CREDIT_LIMIT  
MAX_CREDIT_LIMIT_PER_ORG
```

```

MIN_CREDIT_LIMIT_PER_ORG
USED_CREDIT_LIMIT
LATEST_6M_USED_AVG_AMOUNT;
run;

/*查看数据集case2.test_d3中的缺失情况*/
data _null_;
set case2.test_d3 nobs=total;
call symput('total',total);
stop;
run;

proc contents noprint data=case2.test_d3 out=test_d3;
run;

proc sql noprint;
select distinct name into: var_lst separated by ' ' from test_d3;
quit;

proc sql noprint;
select count(distinct name) into: var_qty from test_d3;
quit;

proc sql noprint;
select distinct name into: var1 -:%cmpres(var%eval(&var_qty+0)) from
test_d3;
quit;

%macro missing();
data temp1;
set case2.test_d3 nobs=nobs;
%do i=1 %to &var_qty;
if missing(&&var&i) then do;
m_&&var&i..+1;
end;
mr_&&var&i=m_&&var&i/&total;
keep mr_&&var&i;
%end;
if _N_=nobs then output;
run;
%mend;
%missing;

proc transpose data=temp1 out=temp1;

```

```

run;

data temp1;
set temp1;
rename COL1=bilv;
run;
quit;

/*紧接着，将数据标准化*/
proc standard data=case2.test_d3 out=case2.test_std mean=0 std=1;
var
IS_LOCAL
WORK_PROVINCE
EDU_LEVEL
MARRY_STATUS
HAS_FUND
state
currency
guarantee_type
payment_rating
class5_state
credit_limit_amount
scheduled_payment_amount
actual_payment_amount
curr_overdue_cyc
curr_overdue_amount
share_credit_limit_amount
used_credit_limit_amount
latest6_month_used_avg_amount
used_highest_amount
HOUSE_LOAN_COUNT
COMMERCIAL_LOAN_COUNT
OTHER_LOAN_COUNT
LOANCARD_COUNT
STANDARD_LOANCARD_COUNT
FINANCE_CORP_COUNT
FINANCE_ORG_COUNT
ACCOUNT_COUNT
CREDIT_LIMIT
MAX_CREDIT_LIMIT_PER_ORG
MIN_CREDIT_LIMIT_PER_ORG
USED_CREDIT_LIMIT

```

```

LATEST_6M_USED_AVG_AMOUNT
;
run;

/*再然后，将进行因子分析*/

/*将原始变量重新命名*/
data case2.test_std1;
set case2.test_std;
rename
IS_LOCAL=b_2
WORK_PROVINCE=b_3
EDU_LEVEL=b_4
MARRY_STATUS=b_5
HAS_FUND=b_6
state=b_7
currency=b_8
guarantee_type=b_9
payment_rating=b_10
class5_state=b_11
credit_limit_amount=b_12
scheduled_payment_amount=b_13
actual_payment_amount=b_14
curr_overdue_cyc=b_15
curr_overdue_amount=b_16
share_credit_limit_amount=b_17
used_credit_limit_amount=b_18
latest6_month_used_avg_amount=b_19
used_highest_amount=b_20
HOUSE_LOAN_COUNT=b_21
COMMERCIAL_LOAN_COUNT=b_22
OTHER_LOAN_COUNT=b_23
LOANCARD_COUNT=b_24
STANDARD_LOANCARD_COUNT=b_25
FINANCE_CORP_COUNT=b_26
FINANCE_ORG_COUNT=b_27
ACCOUNT_COUNT=b_28
CREDIT_LIMIT=b_29
MAX_CREDIT_LIMIT_PER_ORG=b_30
MIN_CREDIT_LIMIT_PER_ORG=b_31
USED_CREDIT_LIMIT=b_32
LATEST_6M_USED_AVG_AMOUNT=b_33
;
run;

```

```

data case2.test_std1;
set case2.test_std1;
x_1=-0.02135*b_2+0.03745*b_3-0.05064*b_4-0.07622*b_5-0.03668*b_6+0.12
283*b_7+0.11572*b_8+0.01013*b_9-0.07147*b_10-0.02096*b_11+0.25166*b_1
2+0.47238*b_13+0.33472*b_14+0.01715*b_15+0.19344*b_16+0.52411*b_17+0.
57466*b_18+0.5839*b_19+0.54733*b_20+0.2238*b_21+0.13889*b_22+0.21674*
b_23+0.25342*b_24+0.10214*b_25+0.65701*b_26+0.6616*b_27+0.14426*b_28+
0.72719*b_29+0.69389*b_30+0.12029*b_31+0.081093*b_32+0.80608*b_33;
x_2=0.01087*b_2-0.00841*b_3+0.03182*b_4+0.01078*b_5+0.00854*b_6-0.038
99*b_7+0.07271*b_8+0.22963*b_9-0.12031*b_10+0.05965*b_11-0.03858*b_12
+0.05086*b_13-0.01681*b_14+0.08188*b_15+0.86326*b_16+0.06888*b_17+0.1
5623*b_18+0.17277*b_19+0.02579*b_20-0.06002*b_21+0.63427*b_22+0.80288
*b_23+0.86894*b_24+0.52604*b_25-0.27314*b_26-0.25336*b_27+0.66877*b_2
8-0.24082*b_29-0.21319*b_30+0.29546*b_31-0.25018*b_32-0.26001*b_33;
x_3=0.02195*b_2+0.00262*b_3+0.04987*b_4+0.00659*b_5+0.02996*b_6+0.145
24*b_7+0.32557*b_8-0.09179*b_9+0.02609*b_10-0.06236*b_11+0.15911*b_12
+0.54951*b_13+0.45043*b_14-0.01956*b_15-0.19449*b_16+0.51551*b_17+0.6
5065*b_18+0.64577*b_19+0.64491*b_20-0.07012*b_21-0.17035*b_22-0.24316
*b_23-0.25529*b_24-0.13704*b_25-0.46334*b_26-0.46986*b_27-0.17595*b_2
8-0.34563*b_29-0.26844*b_30+0.11112*b_31-0.37012*b_32-0.37253*b_33;
x_4=0.00553*b_2-0.05669*b_3-0.01571*b_4+0.03724*b_5+0.0085*b_6+0.7580
9*b_7-0.12892*b_8+0.43491*b_9+0.05414*b_10-0.57416*b_11-0.27822*b_12+
0.2956*b_13+0.40704*b_14+0.00409*b_15+0.05434*b_16-0.14702*b_17-0.082
62*b_18-0.0941*b_19-0.09281*b_20-0.1898*b_21+0.04287*b_22-0.11766*b_2
3+0.06684*b_24+0.05338*b_25+0.21023*b_26+0.20992*b_27+0.04852*b_28-0.
06239*b_29-0.18557*b_30-0.29071*b_31-0.05976*b_32-0.07337*b_33;
x_5=-0.16678*b_2+0.01249*b_3-0.09974*b_4+0.16576*b_5+0.08084*b_6-0.36
093*b_7+0.3474*b_8+0.26331*b_9+0.26509*b_10
+0.40173*b_11-0.45517*b_12-0.00098*b_13+0.01274*b_14-0.00777*b_15+0.0
0091*b_16-0.03611*b_17+0.11904*b_18+0.11633*b_19+0.11766*b_20-0.24673
*b_21-0.03316*b_22+0.09313*b_23+0.00014*b_24-0.02136*b_25+0.25673*b_2
6+0.2549*b_27-0.05068*b_28-0.09617*b_29-0.21722*b_30-0.41019*b_31-0.0
4035*b_32-0.02635*b_33;
x_6=0.24108*b_2+0.1318*b_3+0.41614*b_4-0.31513*b_5+0.22354*b_6-0.0286
*b_7+0.11649*b_8-0.42898*b_9-0.40377*b_10
+0.12295*b_11+0.25917*b_12+0.07962*b_13+0.16868*b_14-0.00314*b_15+0.0
3353*b_16-0.05593*b_17-0.05096*b_18-0.05198*b_19-0.02866*b_20-0.25009
*b_21+0.06331*b_22+0.00383*b_23+0.03954*b_24+0.06285*b_25+0.19487*b_2
6+0.19577*b_27+0.03889*b_28-0.08046*b_29-0.16914*b_30-0.38576*b_31-0.
06006*b_32-0.06177*b_33;
x_7=0.29734*b_2+0.13025*b_3+0.52402*b_4-0.34088*b_5+0.31505*b_6+0.014
54*b_7-0.05591*b_8+0.31834*b_9+0.17285*b_10+0.04862*b_11-0.35745*b_12
-0.09662*b_13-0.07602*b_14+0.00008*b_15-0.00216*b_16+0.02101*b_17+0.0

```

$4528*b_{18}+0.04726*b_{19}+0.02757*b_{20}-0.19565*b_{21}-0.06343*b_{22}-0.04183$
 $*b_{23}-0.02191*b_{24}-0.08097*b_{25}-0.09573*b_{26}-0.0977*b_{27}-0.03189*b_{28}$
 $+0.04952*b_{29}+0.16408*b_{30}+0.29968*b_{31}+0.0824*b_{32}+0.08068*b_{33};$
 $x_8=-0.63671*b_2+0.39282*b_3+0.13151*b_4+0.31946*b_5+0.57073*b_6+0.06$
 $614*b_7-0.08169*b_8-0.08487*b_9-0.05989*b_{10}-0.11052*b_{11}+0.07425*b_{12}$
 $-0.02475*b_{13}-0.05366*b_{14}+0.06082*b_{15}+0.00286*b_{16}+0.0138*b_{17}+0.0$
 $0723*b_{18}+0.01021*b_{19}-0.01415*b_{20}-0.09322*b_{21}+0.01246*b_{22}-0.00336$
 $*b_{23}-0.0029*b_{24}-0.02705*b_{25}-0.01865*b_{26}-0.01709*b_{27}+0.01131*b_{28}$
 $+0.00111*b_{29}+0.02831*b_{30}+0.07945*b_{31}+0.02336*b_{32}+0.01902*b_{33};$
 $x_9=0.02608*b_2+0.02603*b_3-0.01378*b_4-0.05548*b_5-0.05067*b_6+0.002$
 $53*b_7+0.01087*b_8-0.00965*b_9+0.06033*b_{10}+0.01326*b_{11}+0.01226*b_{12}$
 $+0.00526*b_{13}+0.01837*b_{14}+0.97888*b_{15}-0.01591*b_{16}+0.0032*b_{17}-0.00$
 $123*b_{18}-0.01768*b_{19}-0.00329*b_{20}+0.01848*b_{21}+0.12117*b_{22}-0.00305*$
 $b_{23}-0.0139*b_{24}-0.1369*b_{25}+0.00533*b_{26}+0.00505*b_{27}-0.05941*b_{28}+$
 $.00104*b_{29}-0.01049*b_{30}-0.0186*b_{31}-0.00655*b_{32}-0.00667*b_{33};$
 $x_{10}=-0.06896*b_2+0.81549*b_3+0.00241*b_4-0.23694*b_5-0.40046*b_6+0.0$
 $0897*b_7+0.09886*b_8+0.0511*b_9+0.14115*b_{10}$
 $+0.01766*b_{11}-0.01372*b_{12}+0.05231*b_{13}+0.09385*b_{14}-0.07247*b_{15}-0.0$
 $026*b_{16}-0.03807*b_{17}-0.0525*b_{18}$
 $-0.0481*b_{19}-0.02171*b_{20}+0.22346*b_{21}+0.00635*b_{22}+0.03034*b_{23}+0.00$
 $045*b_{24}-0.00333*b_{25}-0.00687*b_{26}-0.00598*b_{27}+0.00904*b_{28}+0.00161*$
 $b_{29}-0.03544*b_{30}-0.03107*b_{31}-0.0325*b_{32}-0.03056*b_{33};$
 $x_{11}=0.01163*b_2-0.16585*b_3+0.36106*b_4+0.18758*b_5-0.11202*b_6+0.04$
 $734*b_7+0.08089*b_8-0.04776*b_9+0.57646*b_{10}+0.22667*b_{11}+0.2502*b_{12}$
 $+0.16864*b_{13}+0.2837*b_{14}-0.03027*b_{15}-0.01914*b_{16}-0.04965*b_{17}-0.14$
 $663*b_{18}-0.14299*b_{19}-0.09084*b_{20}+0.34356*b_{21}+0.08813*b_{22}+0.01201*$
 $b_{23}-0.00602*b_{24}+0.07392*b_{25}-0.02033*b_{26}-0.01844*b_{27}+0.07671*b_{28}$
 $+0.03549*b_{29}+0.01246*b_{30}-0.04321*b_{31}-0.01435*b_{32}-0.02027*b_{33};$
 $x_{12}=0.31514*b_2+0.12429*b_3+0.20605*b_4+0.59423*b_5-0.07789*b_6-0.02$
 $891*b_7+0.43179*b_8+0.04899*b_9-0.349*b_{10}$
 $-0.11799*b_{11}-0.1614*b_{12}-0.01764*b_{13}+0.08022*b_{14}+0.03606*b_{15}+0.00$
 $807*b_{16}-0.01669*b_{17}-0.07519*b_{18}-0.07992*b_{19}-0.04447*b_{20}+0.1224*b_{21}$
 $-0.04675*b_{22}+0.02358*b_{23}-0.01033*b_{24}-0.05341*b_{25}-0.03852*b_{26}-$
 $0.03942*b_{27}-0.03475*b_{28}+0.05203*b_{29}+0.08373*b_{30}+0.08881*b_{31}+0.03$
 $572*b_{32}+0.03272*b_{33};$
 $x_{13}=0.11477*b_2+0.12081*b_3+0.26737*b_4+0.41381*b_5-0.38673*b_6+0.03$
 $689*b_7-0.45781*b_8-0.06748*b_9+0.15539*b_{10}+0.10259*b_{11}+0.12761*b_{12}$
 $-0.02892*b_{13}-0.13557*b_{14}+0.00461*b_{15}+0.01032*b_{16}+0.06111*b_{17}+0.$
 $10667*b_{18}+0.10173*b_{19}+0.05029*b_{20}-0.41614*b_{21}+0.01275*b_{22}-0.0305$
 $5*b_{23}+0.01076*b_{24}+0.00767*b_{25}+0.05376*b_{26}+0.05218*b_{27}+0.0154*b_{28}$
 $-0.04556*b_{29}-0.02067*b_{30}-0.02977*b_{31}+0.00644*b_{32}+0.00503*b_{33};$
 $x_{14}=0.37416*b_2+0.16394*b_3-0.09646*b_4+0.12724*b_5+0.31884*b_6-0.01$
 $219*b_7-0.33532*b_8+0.08094*b_9+0.01328*b_{10}-0.05983*b_{11}-0.00127*b_{12}$
 $-0.07644*b_{13}-0.22836*b_{14}+0.00642*b_{15}+0.03174*b_{16}+0.03447*b_{17}+0.$

```

1472*b_18+0.14883*b_19+0.09617*b_20+0.47179*b_21-0.02409*b_22+0.02534
*b_23+0.03129*b_24+0.02356*b_25+0.04121*b_26+0.04058*b_27-0.06444*b_2
8-0.02597*b_29-0.14118*b_30-0.30544*b_31-0.05624*b_32-0.05155*b_33;
x_15=0.3812*b_2+0.20951*b_3-0.43083*b_4+0.07625*b_5+0.25238*b_6+0.009
37*b_7+0.19825*b_8-0.18262*b_9+0.34273*b_10-0.13634*b_11+0.1319*b_12-
0.02553*b_13-0.00458*b_14-0.01703*b_15-0.07222*b_16+0.02502*b_17-0.05
658*b_18-0.05566*b_19-0.02409*b_20-0.342*b_21+0.13429*b_22-0.04821*b_
23-0.05417*b_24+0.10232*b_25+0.01483*b_26+0.01639*b_27+0.13311*b_28-0
.00558*b_29+0.04524*b_30+0.13436*b_31+0.02382*b_32+0.02693*b_33;
;
run;
quit;

/*将数据水平压缩*/
proc sql;
create table case2.test_yin as select report_id ,
avg(x_1) as x1,
avg(x_2) as x2,
avg(x_3) as x3,
avg(x_4) as x4,
avg(x_5) as x5,
avg(x_6) as x6,
avg(x_7) as x7,
avg(x_8) as x8,
avg(x_9) as x9,
avg(x_10) as x10,
avg(x_11) as x11,
avg(x_12) as x12,
avg(x_13) as x13,
avg(x_14) as x14,
avg(x_15) as x15
from case2.test_std1 group by report_id;
run;
quit;

/*选取建模变量,并且标准化*/
proc sql;
create table case2.test_yin_ya as select report_id, x1, x2, x3, x4, x5,
x6, x7, x8, x9, x10, x11, x12, x13, x14, x15 from case2.test_yin;
run;
quit;

proc standard data=case2.test_yin_ya out=case2.test_yin_ya mean=0 std=1;
var x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15;

```

```

run;

/*计算所有的预测值y*/
data case2.test_suoyou_score;
set case2.test_yin_ya;
f= -1.8950 -0.5265*x1 + 0.2866*x3 -0.0809*x6 -0.4095*x8 -0.6236*x10 +
0.1861*x11 -0.1338*x14 -0.1933*x15;
p=exp(f)/(1+exp(f));
if p>=0.5 then do y=1;end;
if p<0.5 then do y=0;end;
run;

proc sql;
create table test_weiyue as select * from case2.test_suoyou_score where
y=1;
run;
quit;

/*删除work逻辑库中的所有数据集和宏程序*/
proc catalog catalog=work.sasmacr force kill;
run;
quit;

proc datasets library=work nolist nodetails kill;
run;
quit;

```

附录十八：将模型转化为评分卡

```

/*将模型转化为评分卡*/
data woe;
set case2.train_suoyou_score;
odds=p/(1-p);
B=30/log(2);
A=750+B*log(1/50);
score=A-B*log(odds);

run;

```