

中小微企业的信贷决策

摘要

中小微企业在国民经济发展中具有重要地位。然而由于中小微企业自身规模小、风险大、管理不够规范与金融机构企业信息不对称等因素，商业银行对其设置了严格的融资约束和限制，普遍存在“融资难”的情况。对于商业银行而言，如何评估中小企业信贷风险，提高风险控制能力，给出合理的信贷政策成为了银行发展信贷业务面临的重要问题。本文针对某一些特定的中小微企业财务数据，从银行的角度对该问题进行了研究。

针对问题一，我们先对附件样本数据进行清洗，提取数据特征。以数据特征因子作为分析项选取合适的因变量进行分层回归，在回归过程中自然地提取出了制约决策的根本影响因素。然后，使用 0-1 变量表征企业是否违约，建立了 Logit 回归模型测量企业守约发生的概率进而对信贷风险进行了概率量化，模型的样本内预测成功率为 97.56%，模型效果良好；然后站在银行的角度通过对企业的实力和供求关系进行回归分析为银行的是否给企业贷款的决策给出依据。接着根据企业体量对企业属性进行分级聚类，结合信贷风险指标建立了以合理贷款额度为目标的贷款额度分配策略。最后根据金额构建最优解方程，当年利率为 0.0465 时得到银行利润的最优解，并按照守约率排名优先满足守约率高的企业贷款申请。

针对问题二，考虑到数据复杂程度以及重要指标信誉评级的缺失，选取了更多的指标来进行研究。剔除掉信誉评级后加入企业供求稳定关系重新进行求解，得出适合附件二企业的信贷风险量化 Logit 模型。根据问题一建立的信贷策略决策模型，依次求出是否贷款、贷款额度、贷款利率。最后提出的策略在贷款总金额为 1 亿元的前提下能满足守约率前 131 家公司的贷款申请。

针对问题三，需要考虑突发因素对银行提供贷款业务的影响，本文以 2020 年新冠疫情为例子，考虑疫情对国家 GDP 的影响，并具体划分到不同的产业。疫情对大部分行业有消极作用，但对医药类行业却有正向促进的作用。因此最终将附件二中 302 家企业划分为 5 个类，分别为第一产业、第二产业、第三产业、个体户、医药类。并赋予不同的权值，并对问题一中信贷风险的输出进行调整。最终成为该突发因素影响下的守约率。再带入贷款额度分配模型与利润最优解方程，最终得到突发因素影响下，当年利率为 0.0465 时得到银行利润的最优解。提出的策略能满足守约率前 124 家公司的贷款申请。

关键词：分层回归 Logit 模型 最优解 特征提取 选择策略

一、 问题重述

1.1 问题背景

在实际中，由于中小微企业规模相对较小，也缺少抵押资产，因此银行通常是依据信贷政策、企业的交易票据信息和上下游企业的影响力，向实力强、供求关系稳定的企业提供贷款，并可以对信誉高、信贷风险小的企业给予利率优惠。银行首先根据中小微企业的实力、信誉对其信贷风险做出评估，然后依据信贷风险等因素来确定是否放贷及贷款额度、利率和期限等信贷策略。

某银行对确定要放贷企业的贷款额度为 10~100 万元；年利率为 4%~15%；贷款期限为 1 年。附件 1~3 分别给出了 123 家有信贷记录企业的相关数据、302 家无信贷记录企业的相关数据和贷款利率与客户流失率关系的 2019 年统计数据。

1.2 问题提出

该银行请你们团队根据实际和附件中的数据信息，通过建立数学模型研究对中小微企业的信贷策略，主要解决下列问题：

(1) 对附件 1 中 123 家企业的信贷风险进行量化分析，给出该银行在年度信贷总额固定时对这些企业的信贷策略。

(2) 在问题 1 的基础上，对附件 2 中 302 家企业的信贷风险进行量化分析，并给出该银行在年度信贷总额为 1 亿元时对这些企业的信贷策略。

(3) 企业的生产经营和经济效益可能会受到一些突发因素影响，而且突发因素往往对不同行业、不同类别的企业会有不同的影响。综合考虑附件 2 中各企业的信贷风险和可能的突发因素（例如：新冠病毒疫情）对各企业的影响，给出该银行在年度信贷总额为 1 亿元时的信贷调整策略。

二、 问题分析

对题目背景中提供的银行信贷策略进行逻辑梳理，我们不难得到如下的信贷策略指标决定关系：企业的实力、供求关系决定了银行是否提供贷款；企业的实力、信誉决定了信贷风险；企业的信誉评级、信贷风险决定了利率。因此银行信贷策略的决定聚焦到了企业的实力、供求关系、信誉评级、信贷风险等指标上。

2.2 问题一的分析

针对问题一，为确定有信贷记录企业信贷风险评估的策略，我们首先需要考虑影响银行做出决策的因素并研究因素对银行决策的作用机理，根据附件一中的数据可以整理出 123 家企业的信誉评级，然后使用 MySQL 分别提取可信企业与失信企业的每一年的发票数据。接着根据附件一中的发票数据利用 Java 语言对数据进行二次清洗，统计出反

应企业的盈利能力、对贷款的偿还能力、成长与发展能力和财务稳定能力的多个隐性指标。接着我们综合考虑各影响指标以及分布情况，兼顾从银行角度出发的收益率和延滞率，确定影响银行信贷策略的主要因子与分布规律。最后建立出包含是否贷款、贷款额度、贷款利率三项因子的综合信贷决策。

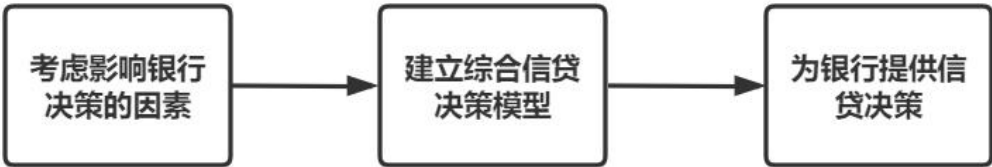


图 2-1 问题一分析

2.2 问题二的分析

针对问题二，为确定无信贷记录企业信贷风险评估的策略，我们需要在第一问建立的信贷策略综合评价体系模型基础上，结合附件二中的实际数据情况，给出银行的信贷风险评价模型，然后带入问题一中求解出的信贷策略模型中得到无信贷记录且年度信贷总额为 1 亿元时的信贷策略。由于在问题一的求解中信誉等级作为因变量参与了信贷风险量化指标的指定，因此在问题二中应剔除信誉项得到适合问题二的信誉风险量化指标模型。

2.3 问题三的分析

针对问题三，以新冠疫情对行业打击为例，收集疫情期间对不同类型企业的影响数据，把影响程度量化。然后针对附件二中的企业进行按类型进行分类，给出每个企业遭受的影响程度的量化参数。接着根据问题二得出的假设企业在未遭受突发因素影响时银行的年度信贷策略，在问题二模型的基础上加入疫情对各个决定银行信贷策略影响的量化指标得出银行的信贷调整策略。

三、模型的假设

1. 假设银行的首要目标是规避放贷风险，并能够根据收益做出最优决策。
2. 所有附件所给的数据真实且合乎常理，不考虑财务造假等商业欺诈行为。
3. 决策因素仅考虑企业经济因素和客观环境因素，不考虑放贷个人倾向等其他因素。
4. 因为我们考虑到题目附件一，二仅提供企业进项发票信息，而公司成本包含但不限于进项成本，还有人工管理费用、水电场地费用、应收应还账款等，所以出于简化模型需要，我们假设企业进项合计额可以最大程度上反映企业的原始资金积累。

四、 符号说明

表 4 符号说明

符号	含义	单位
W	信誉评级	
Y_1	进项总额	元
Y_2	销项总额	元
Y_3	利润总额	元
M	贷款额度	元
S	进项纳税额	元
P	企业守约率	
ux	企业流失率	
r	银行贷款税率	
Ex	附件公司编号	
V_1	销售净利润率	元/y
V_2	营业收入增长率	元/y
V_3	净利润增长率	元/y
Rx	银行盈利总计	元
π	供求稳定系数	

五、 数据清洗

5.1 数据清洗

将数据导入 MySQL 数据库并用可视化工具 SQLyog 作数据预处理，然后剔除作废发票项目数据。由于负数发票需要总计入总额，对冲企业之前开出的销项税发票，所以不在合计中剔除。后续按照公司子项分离出对应的对公流水（交易往来公司，进项总额，销项总额等），统计出每一家公司每一年的进项总额、销项总额、利润总额、进项纳税额、营业收入增长率、净利润增长率、销售净值增长率、供求稳定系数。

表 5-1 指标计算公式

财务分析指标	计算方法
V_1 销售净利润率	利润总额/销项总额
S 进项纳税金额	纳税合计总额
V_2 营业收入增长率	本年营业收入增长额/上年营业收入
V_3 净利润增长率	本年净利润增长额/上年净利润
π 供求稳定系数	供应稳定系数+销售稳定系数

供求稳定系数（ π ）求解过程 如下：

数据集中，按照企业代号排序，以 E1 为例，根据观察，每家公司的开票数额占比最大的前 5 家开票单位

①求出 E1 公司，2016-2020 年五年中每一年的开票数额占比最大的前 5 家开票单位记作 $xyear_1, xyear_2, xyear_3, xyear_4, xyear_5$ 其中 year 为对应的年份，得出该公司（E1）的“老客户”名单。记作 T_{2016in}

②复盘计算 5 年中，每一个销项公司出现在“老客户”名单的次数。我们定义有以下评分表 5-2。

表 5-2 “老客户”评分表

“老客户”频数	得分
5	50
4	40
3	30
2	20
1	0

③按照上述规则计算出销项发票求关系得分与进项供关系得分，求和得到供应稳定关系量化系数 π 。

$$\pi = f(T_{2016i}, T_{2017i}, T_{2018i}, T_{2019i}, T_{2020i}) + f(T_{2016o}, T_{2017o}, T_{2018o}, T_{2019o}, T_{2020o}) \quad (5-1)$$

π 反映了该企业的供求关系稳定情况， π 越大表示越稳定。

最后，将清洗得到的指标标准化，统一量纲后可参与后续模型的求解。

六、模型的建立与求解

6.1 问题 1 的模型建立与求解

6.1.1 信贷策略影响因素及其影响机理的分析

本问要求对拥有信贷记录的 123 家公司的信贷风险进行量化，同时得出合理的信贷决策。中小微企业信贷风险评价的指标，应该能全面衡量中小微企业所面临的各种风险，反映企业的盈利能力、对贷款的偿还能力、成长与发展能力和财务稳定能力。

首先根据企业财务特点和衡量风险的需要，我们选取了 5 个财务分析指标，衡量企业的盈利能力、对贷款的偿还能力、成长与发展能力以及财务稳定能力。盈利能力是企业有足够的资金偿还贷款的保障，能够直接反映企业的经营状况；偿还能力是企业信用评价的关键因素，直接影响到企业按时偿还贷款的能力；由于中小微企业大多处于成长的初级阶段，经营效率和成长速度都很高，成长与发展能力是考察企业过往发展水平、预测未来规模的重要指标；财务稳定能力反映在企业上下游的供需关系，较高的资金周转效率是企业平稳运营、发展的基础。

接着站在银行的角度，银行以控制风险为出发点，以利润最大化为目标，合理选择目标市场得出具体分配贷款利率和贷款额度的策略。

最后根据问题分析中梳理的信贷策略指标间相互作用关系得到如图 6-1 所示的决策影响因素及其影响机理图。

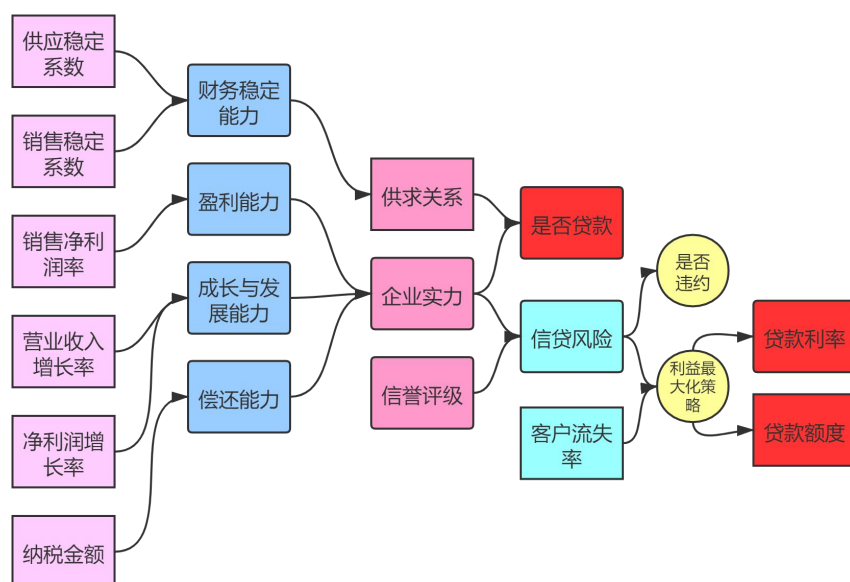


图 6-1 银行信贷策略影响因素及其影响机理图

6.1.2 信贷风险量化评分模型

为了量化信贷风险，我们决定考量公司的企业实力和信誉评级。本问中信誉评级已由银行给出。我们可以通过分析企业的盈利能力、对贷款的偿还能力、成长与发展能力来量化企业实力。最终我们把三大能力映射到四个企业财务分析指标上，得出信贷风险量化评分的决策树，如图 6-2 所示。

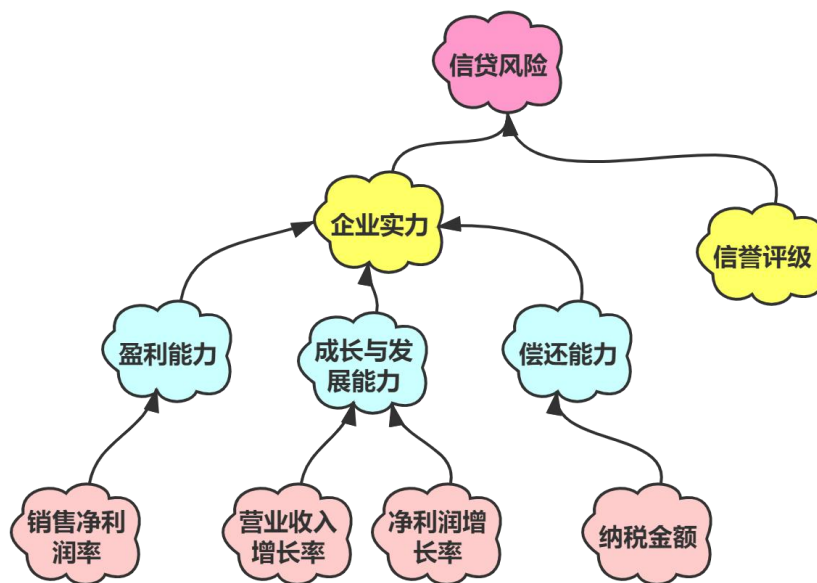


图 6-2 信贷风险量化评分决策树

1. 基于分层回归的信贷风险指标筛选

建立分层回归模型，对信贷风险指标由底层到上层进行重要性检验，总体回归函数的一般形式如下：

$$Y = \beta_1 + \beta_2 x_2 + \beta_3 x_3 + \dots + \beta_k x_k + u_i \quad (6-1)$$

其中， x_i 是解释变量，表示第 i 个分析项；为 β 待估参数， Y 为因变量；

分层回归的核心依旧为回归分析，区别在于分层回归可为多层。每一层均在上一层基础上放入更多项，那放入的更多项是否对模型有解释力度，此为分层回归关心的问题。因此根据图 6-2 将信贷风险的影响因子分为 4 层，如表 6-2 所示。

表 6-2 分层回归的影响因子

层级	分析项
第一层	V ₂ 营业收入增长率
	V ₃ 净利润增长率
第二层	V ₁ 销售净利润率
第三层	S 进项纳税额
第四层	W 信誉评级

上表格中相当于共有 4 个回归分析，将量化后的是否违约参数作为因变量利用 SPSSAU 构建分层回归模型得到如下分层回归估计结果（表 6-3）：

表 6-3 分层回归分析结果

	分层 1	分层 2	分层 3	分层 4
常数	0.781 (20.110**)	0.774 (20.572**)	0.743 (18.709**)	0.335 (6.932**)
V3	0.002 (0.179)	0.002 (0.133)	0.004 (0.334)	0.014 (1.544)
V2	-0.005 (-0.597)	-0.006 (-0.744)	-0.006 (-0.753)	-0.004 (-0.572)
V1		0.061 (3.059**)	0.062 (3.174**)	0.030 (2.088*)
S			0.098 (2.168*)	0.005 (0.144)
W				0.834 (10.472**)
样本量	123	123	123	123
R^2	0.003	0.076	0.111	0.541
调整 R^2	-0.013	0.052	0.081	0.522
F 值	$F(2,120)=0.188,$ $p=0.829$	$F(3,119)=3.253,$ $p=0.024$	$F(4,118)=3.691,$ $p=0.007$	$F(5,117)=27.604,$ $p=0.000$
ΔR^2	0.003	0.073	0.035	0.430
ΔF 值	$F(2,120)=0.188,$ $p=0.829$	$F(1,119)=9.357,$ $p=0.003$	$F(1,118)=4.702,$ $p=0.032$	$F(1,117)=109.660,$ $p=0.000$
因变量：是否违约				
* $p<0.05$ ** $p<0.01$ 括号里面为 t 值				

分层回归用于研究自变量(X)增加时带来的模型变化，通常用于模型稳定性检验，中介作用或者调节作用研究。从上表格可知，本次分层回归分析共涉及 4 个模型。模型 1 中的自变量为净利润平均增长率,营业收入增长率,模型 2 在模型 1 的基础上加入销售净利润率，模型 3 在模型 2 的基础上加入进项纳税金额归一化数据，模型 4 在模型 3 的基础上加入信誉等级，模型的因变量为：违约因子。

针对模型 1，将 V_2 作为自变量，而将是否违约作为因变量进行线性回归分析，从上表可以看出，模型 R 方值为 0.003，意味着 V_3, V_2 可以解释是否违约的 0.3% 变化原因。对模型进行 F 检验时发现模型并没有通过 F 检验($F=0.188, p>0.05$)，也即说明 V_3, V_2 不会对是否违约产生影响关系，因而不能具体分析自变量对于因变量的影响关系，所以在最后挑选显著性指标的时候，我们将剔除净利润平均增长率和营业收入增长率。

针对模型 2，其在模型 1 的基础上加入 V_1 后， F 值变化呈现出显著性($p<0.05$)，意味着 V_1 加入后对模型具有解释意义。另外， R 方值由 0.003 上升到 0.076，意味着 V_1 可对是否违约产生 7.3%的解释力度。具体来看， V_1 的回归系数值为 0.061，并且呈现出显著性($t=3.059$ ， $p=0.003<0.01$)，意味着 V_1 会对是否违约产生显著的正向影响关系。

针对模型 3，其在模型 2 的基础上加入 S 后， F 值变化呈现出显著性($p<0.05$)，意味着 S 加入后对模型具有解释意义。另外， R 方值由 0.076 上升到 0.111，意味着 S 可对是否违约产生 3.5%的解释力度。具体来看， S 的回归系数值为 0.098，并且呈现出显著性($t=2.168$ ， $p=0.032<0.05$)，意味着 S 会对是否违约产生显著的正向影响关系。

针对模型 4:其在模型 3 的基础上加入 W 后， F 值变化呈现出显著性($p<0.05$)，意味着 W 加入后对模型具有解释意义。另外， R 方值由 0.111 上升到 0.541，意味着 W 可对是否违约产生 43.0%的解释力度。具体来看， W 的回归系数值为 0.834，并且呈现出显著性($t=10.472$ ， $p=0.000<0.01$)，意味着 W 会对是否违约产生显著的正向影响关系。

通过四个回归模型的分析我们剔除净利润平均增长率，营业收入增长率对违约因子的影，提取出信誉等级，进项纳税金额归一化数据，销售净利润率这三个显著性指标，其中信誉等级对违约因子的影响程度最为显著。

2. 基于 Logit 模型预测企业信贷风险

由附件 1 可知，信贷风险直接体现为企业是否违约，因此我们采用 0-1 变量 Q 表征企业是否违约。若 $Q=1$ ，代表企业不违约，反之 $Q=0$ 则代表该企业违约。

(一) Logit 回归模型

逻辑回归是一种广义的线性回归模型，自变量为一个回归函数 Z ，函数表达式如下：

$$p(z) = \frac{1}{1 + e^{-z}} \quad (6-2)$$

$$z = a + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

函数图像如下：

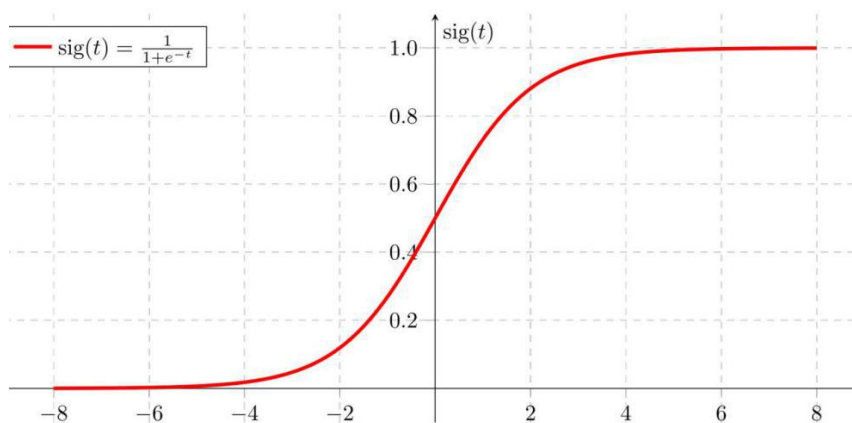


图 6-3 logit 回归函数图像

如图所示，逻辑回归函数是一个单调递增函数，即随着 Z 的增大， P 也不断增大。

现实生活中，许多现象量度并非线性，Logit 模型属于概率模型，能预测某事件发生的概率。因此通过样本企业的财务指标，建立 Logit 模型就可以预测观察期内企业的违约概率，以此来量化信贷风险。在逻辑回归函数的作用下，可以将企业的特征信息综合起来并转化为一个概率值，该值可以给银行预测企业信誉好坏提供了一个直观依据即 $P(Z)$ 值越大，证明该企业的信贷风险越低。

其中，Logit 模型形式定义如下：

$$\text{logit}P = \ln \frac{P}{1-P} \quad (6-3)$$

$$\text{logit}P = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$$

其中 x 是自变量因子，表示第 i 个指标； β 为代求参数， P 为守约状况，为虚拟变量，规定当 P 为 1 时，违约现象不存在。Logit 模型的优点是解决了非线性的问题，他对样本的要求不严苛，不要求样本服从标准正态分布。

（二）模型与检验

在上文经过分层回归分析后我们得到了影响信贷风险的三个显著指标：信誉等级 (W)，进项纳税金额 (S)，销售净利润率 (V_1)。将挑选出的因子与 123 家企业违约现状建立 Logit 分析模型，分析结果如下表 6-4：

表 6-4 二元 Logit 回归模型似然比检验结果表

似然比卡方值	df	p	AIC 值	BIC 值	-2LLNULL 值	-2LLF 值
97.303	3	0.000	40.164	51.413	129.467	32.164

首先对模型整体有效性进行分析，从上表 6-4 可知：此处模型检验的原定假设为：是否放入自变量（W, S, V₁）两种情况时模型质量均一样；这里 *p* 值小于 0.05，因而说明拒绝原定假设，即说明本次构建模型时，放入的自变量具有有效性，本次模型构建有意义。

表 6-5 二元 Logit 回归分析结果汇总表

项	回归系数	标准误	<i>z</i> 值	<i>p</i> 值	OR 值	OR 值 95% CI
W	14.666	3.413	4.297	0.000	2340767.720	2910.735 ~ 1882408713.609
S	1.880	3.304	0.569	0.569	6.553	0.010 ~ 4255.100
V1	0.368	0.435	0.845	0.398	1.445	0.615 ~ 3.391
截距	-3.220	1.080	-2.982	0.003	0.040	0.005 ~ 0.332
因变量：是否违约						
McFadden <i>R</i> 方：0.752						
Cox & Snell <i>R</i> 方：0.547						
Nagelkerke <i>R</i> 方：0.840						

三个变量中销售净利润率 与进项纳税金额归一化数据 在 5%的显著水平下显著性较低，而销售净利润率 与进项纳税金额归一化数据衡量的是企业实力的重要指标。造成这些显著性差异的原因有很多，可能是由于样本数量较少和指标波动性比较大，所以为全面地保留指标信息，我们将销售净利润率 与进项纳税金额归一化数据加入到模型当中。

综上所述，可得 Logit 回归模型：

$$\ln(P/1-P) = -3.220 + 14.666*W + 1.880*S + 0.368*V_1 \quad (6-4)$$

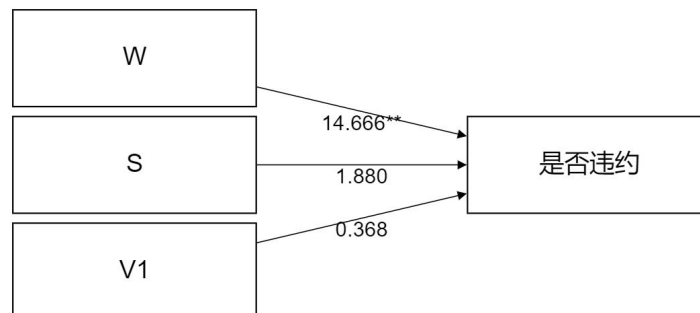


图 6-4 影响因子系数

而违约概率为:

$$P = \frac{e^{(-3.22 + 14.666W + 1.88S + 0.368V_1)}}{1 + e^{(-3.22 + 14.666W + 1.88S + 0.368V_1)}} \quad (6-5)$$

利用上式计算企业的违约概率，即可衡量企业的信贷风险。

要知道模型的有效性需要进行检验，衡量模型的预测能力。一般分为样本内检测和样本外检测。这里我们只进行样本内检测，检验效果如下表。

表 6-6 二元 Logit 回归预测准确率汇总表

		预测值		预测准确率	预测错误率
		0	1		
真实值	0	24	3	88.89%	11.11%
	1	0	96	100.00%	0.00%
汇总				97.56%	2.44%

通过模型预测准确率去判断模型拟合质量，从上表可知：研究模型的整体预测准确率为 97.56%，模型拟合情况良好。当真实值为 0 时，预测准确率为 88.89%；另外当真实值为 1 时，预测准确率为 100.00%。

6.1.3 是否贷款决策模型

类似地，为了量化是否贷款，我们决定考量公司的企业实力和供求关系。企业实力使用 6.1.2 中的评估方法。我们可以通过分析企业的供应稳定系数和销售稳定系数来量化企业供求关系是否稳定。最终我们把公司的企业实力和供求关系整合后得出是否进行信用贷款的决策树，如图 6-5 所示。

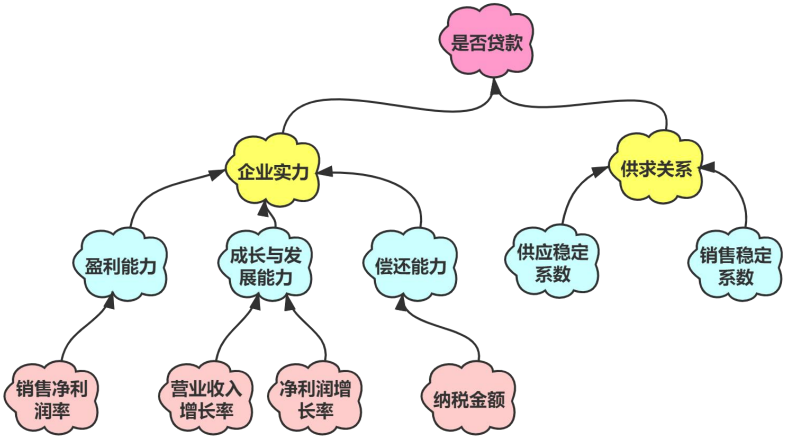


图 6-5 是否贷款决策树

由题目可知，银行对信誉评级为 D 的企业原则上不予放贷，因此类比于信贷风险量化模型采用 0-1 模型将信誉评级为 A,B,C 的企业定义为 1，评级为 D 的企业定义为 0。对其建立 Logit 模型得到实际模型为：

$$\ln(P/1 - P) = -0.127 + 10.281 * S + 0.447 * V_1 + 0.013 * \pi$$

(6-6)

表 6-7 Hosmer-Lemeshow 拟合度检验表

χ^2	自由度 df	p 值
11.171	8	0.192

Hosmer-Lemeshow 拟合度检验用于分析模型拟合优度情况，从上表 6-7 可知：此处模型检验的原定假设为：模型拟合值和观测值的吻合程度一致；这里 p 值大于 0.05,因而说明接受原定假设，即说明本次模型通过 HL 检验，模型拟合优度较好。

利用此模型对银行是否对这 123 家企业提供贷款进行预测得出是否提供贷款的置信度，预测效果如下：

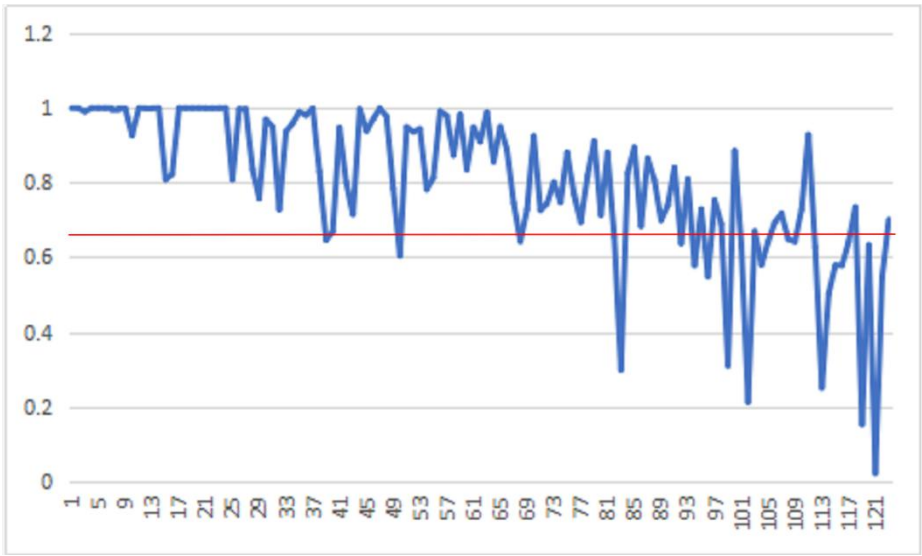


图 6-6 银行对企业是否贷款的置信度分布图

由上图可以看出以 0.65 的置信度为边界对银行是否对企业提供贷款具有良好的效应。当企业的贷款置信度低于 0.65，银行可拒绝对此企业提供贷款。此模型得出的拒绝率为 19.512%，元数据仅在信誉评级作用下得出的拒绝率为 18.699%，具有较为良好合理的拟合效果。

6.1.4 贷款额度模型

贷款额度模型主要综合企业实力、信誉评级和客户流失率决策。因为信贷风险是由企业实力和信誉等级共同生成影响的，考虑到公司的体量，偿还能力等经济指标。因此

在此基础上建立基于信贷风险与客户流失率的贷款额度模型。在已经确定给该企业提供贷款的前提下，该企业的财务规模将直接决定银行的贷款额度。我们选取进项纳税额来直接反应企业的财务规模，首先得出了这 123 家企业的规模信息，如下图 6-7。

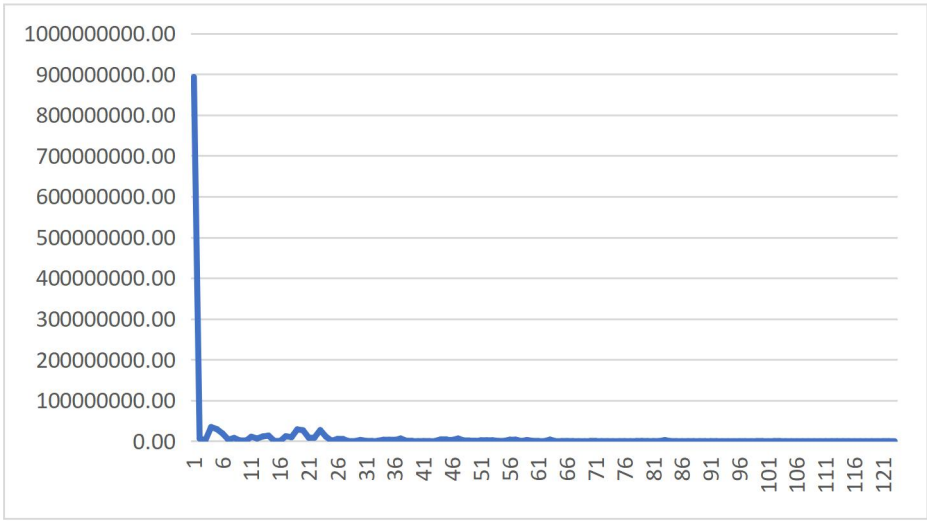


图 6-7 123 家企业进项税额分布情况^①

其中，横坐标为企业标号，纵坐标为企业进项纳税额

由图 6-7，显然可见存在异常极大值，由于该银行对确定要放贷企业的贷款额度为 10~100 万元,因此对企业财务体量明显大于 100 万以及明显小于 10 万的企业进行剔除，以进项纳税额 100 万作为上确界，1 万作为下确界进行筛选，得到的结果如下图 6-8。

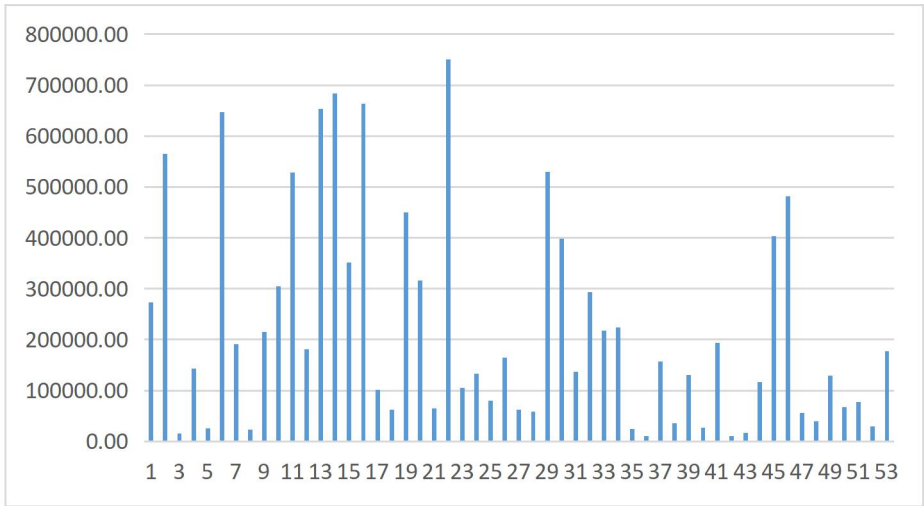
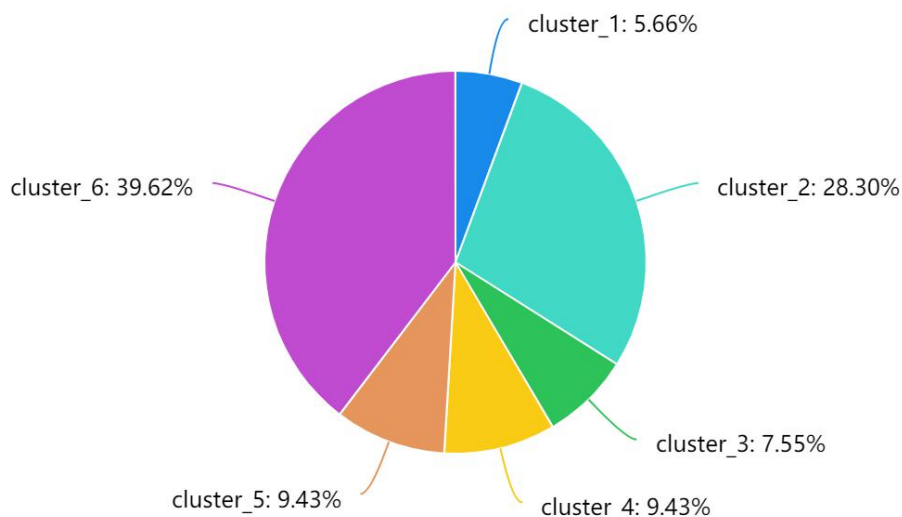


图 6-8 筛选后企业进项税额分布情况图^②

“物以类聚，人以群分”。为了区分贷款额度，我们按照企业体量对公司进行类群划分，把筛选出的进项纳税额数据进行尽可能细分聚类。K-Mean 算法将这 123 家企业的财务体量实力划分六大类，最终的效果如下图 6-9。



1

图 6-9 企业聚类类别汇总图

紧接着根据六类对应的进项税金额进行分析，统计了每一类对应的进项金额分布情况，从而进一步归分合并为 A、B、C 三大等级如下图 6-10。其中 A 等级对应较高的贷款额度，B 等级对应中等的贷款额度，C 等级对应较低的贷款额度。

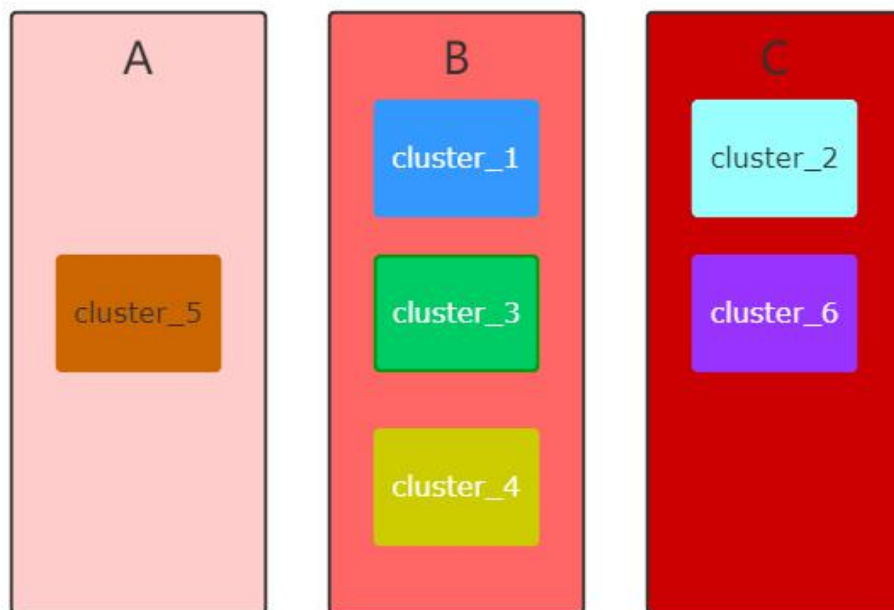


图 6-10 企业聚类迭代后类别示意图

人为地，我们把 10 万~100 万的贷款区间细化为 3 个等级分别对应 A、B、C 三种贷款额度等级，如表 6-8。

①②横坐标表示企业编号，纵坐标表示进项税额。

表 6-8 贷款额度分级表

企业额度等级	企业数量	进项税额分布情况	银行额度分级
A	5	64 万~75 万	60 万-100 万
B	11	30 万~56 万	30 万-60 万
C	36	1 万~22 万	10 万-30 万

银行在给定企业贷款额度除了考虑企业的体量，另一个决定性的因素就是企业的守约率。因此对于进项税额在 1 万~10 万的数据我们可以得出如下的贷款额度求解方式。

$$M = \begin{cases} 100 * P, & A \\ 60 * P, & B \\ 30 * P, & C \end{cases} \quad (6-7)$$

其中 M 为企业对应的贷款额度， P 为企业对应的守约率。

综合上述分析，我们可以得到求解每个企业贷款额度的步骤：

第一步，将进项税额大于 100 万的企业贷款额度设为 100 万，将进项税额小于 1 万的企业贷款额度设为 0；

第二步，根据（1）式求出进项税额在 1 万~100 万之间企业的贷款额度；

第三步，将信誉评级为 D 的企业贷款额度设为 0；

第四步，将企业贷款置信度小于 0.65 的企业贷款额度设为 0；

第五步，将企业违约概率明显较大的企业贷款额度设为 0；

由此我们就得出了 123 家有信贷记录公司的贷款额度分配，其中有 33 家由于信誉较低或市值过低企业不予以贷款额度分配。

6.1.5 贷款利率模型

从 6.1.2 信贷风险量化评分模型我们可以得出量化出每一家企业的失信概率。根据资料显示，银行的利率一般不会波动变化，而是有一定的预收益阈值。利率的选择应是对大小公司统一，因此需要考虑利率对不同信贷级别的公司流失率影响。每一种利率下银行盈利可表示为下列式子：

$$Rx = Nx * M * r * (1 - ux) * P, \quad x = (a, b, c) \quad (6-7)$$

$$r = \text{Max}(Ra(r), Rb(r), Rc(r))$$

其中, Ra , Rb , Rc 分别为信誉等级为 A 、 B 、 C 的公司的银行盈利总计。 M 为公司的贷款额度。 ux 为银行 r 税率下, 信贷等级为 X 的公司流失率。 P 为信贷风险量化评分模型求出的守约率。

最后应选择的是能得出 $Max(Ra(r), Rb(r), Rc(r))$ 的利率值 r 。由下图 6-11, 可得出获益最大的银行年利率是选择 0.0465。

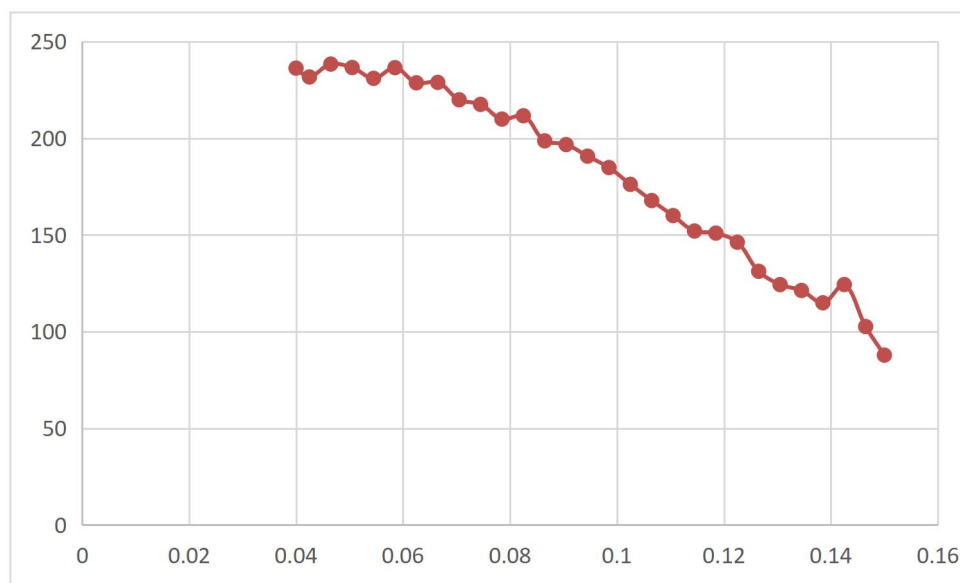


图 6-11 银行年利率与预期收益关系图^③

6.2 问题 2 的模型建立与求解

6.2.1 剔除信誉评级后的信贷风险量化评分模型



图 6-12 企业信贷风险评定影响因子

相较于问题一种求解信贷风险的方法稍有不同的地方是剔除了信誉评级增加了供求稳定系数作为自变量, 同样采用分层回归分析剔除掉次要因子后构建 Logit 模型得到了附件 2 中信贷风险量化的预测模型

$$\ln(P/1-P) = -0.285 + 6.648S + 0.398V_1 + 0.070V_3 + 0.013\pi \quad (6-8)$$

根据附录一的数据进行样本内校验, 得到模型的预测准确率为研究模型的整体预测准确率为 82.93%, 模型拟合情况可以接受。同时经过 HL 模型检验得出拟合优度较好。

因此附件 2 中 302 家企业的信贷风险量化模型为：

$$P = \frac{e^{(-0.285 + 6.648S + 0.398V_1 + 0.070V_3 + 0.013\pi)}}{1 + e^{(-0.285 + 6.648S + 0.398V_1 + 0.070V_3 + 0.013\pi)}} \quad (6-9)$$

6.2.2 贷款额度的分配

表 6-9 贷款额度分级表

企业额度等级	企业数量	进项税额分布情况	银行额度分级
A	17	60 万~90 万	50 万-100 万
B	40	20 万~50 万	20 万-50 万
C	57	1 万~12 万	10 万-20 万

- 第一：进项税额在 1 万~100 万之间的数据分析。
第二：根据问题一中建立的贷款额度划分模型估计出这 303 家企业的贷款额度。

6.2.3 贷款利率的分配

通过 6.1.5 的贷款利率模型可得出对这 302 家企业贷款，要使银行利润最高，应选择利率为 0.0465。

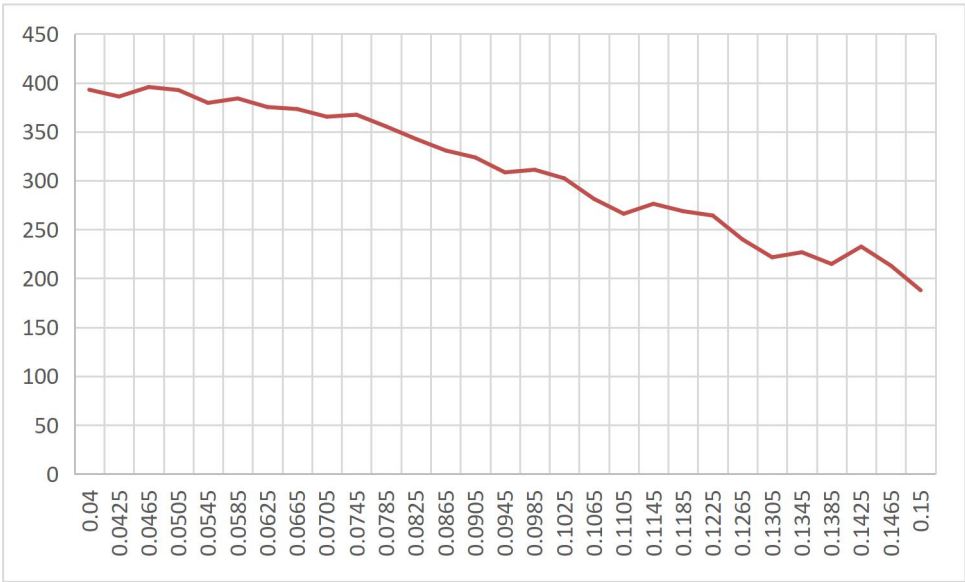


图 6-13 银行年利率与预期收益关系图

6.2.4 总额度分配

按照风险量化指标排序，该模型下可满足排名前 131 的企业的贷款。其中贷款额度为模型 6.2.2 分配出的额度。这 131 家公司，详细结果在支撑材料中展示。

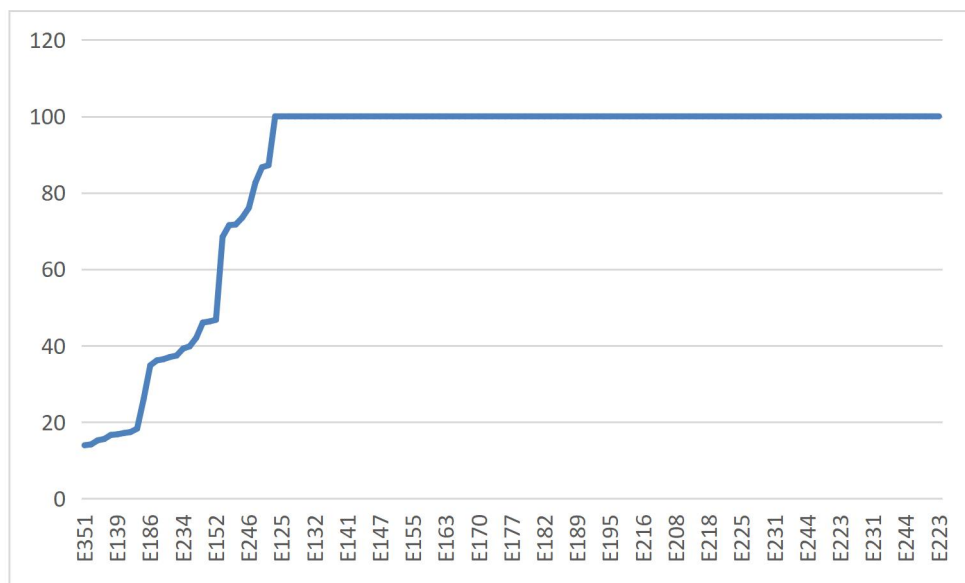


图 6-14 一亿元分配结果图

6.3 问题 3 的模型建立与求解

针对问题 3，这里选取的是新冠病毒疫情对各企业产生了一定的影响，根据文献^[6]，中国人均 GDP 在 2020 年第一季度下降了 6.8%。这个突发情况对不同行业有着不同的影响。其中传统服务业增加值大幅下降，下降了 6%。但对于新兴服务业与金融业住房开发却分别增加了 13.2%，与增加了 6.3%。

6.3.1 加入疫情指标的信贷风险量化评分模型

中国 2020 年第一季度 GDP 总体下降了 6.8%，区分到三大产业，分别为第一产业下降 3.2%，第二产业 9.6%，第三产业下降 5.2%。但由于是疫情，所以对医药企业销售有明显的提高。个体户影响指标取 GDP 的总体反映值。因此需要先对附件 2 中的企业进行分类。分为第一产业、第二产业、第三产业、个体户、医药相关五种公司类型。

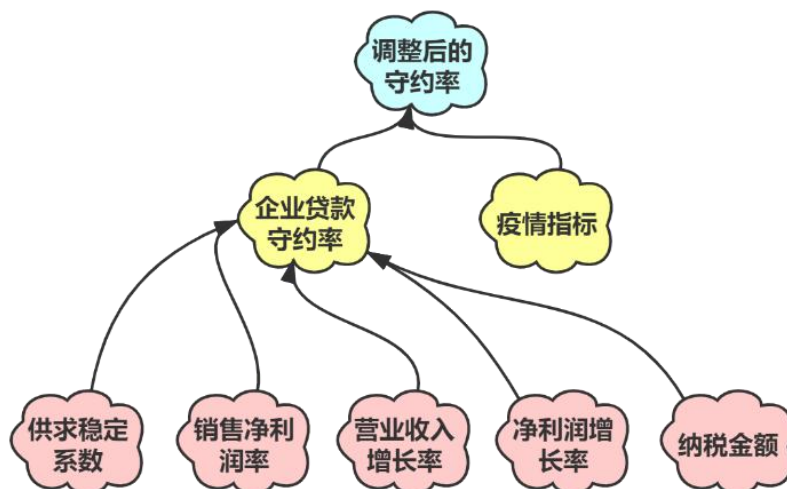


图 6-15 加入疫情指标的信贷风险量化评分决策树

表 6-10:疫情加权指标图

类别	权值 J
第一产业	0.968
第二产业	0.914
第三产业	0.958
个体户	0.6
医药相关	1.1

通过上述分析得出调整后的企业信贷违约率为:

$$P_{调} = \frac{e^{(-0.285 + 6.648S + 0.398V_1 + 0.070V_3 + 0.013\pi)}}{1 + e^{(-0.285 + 6.648S + 0.398V_1 + 0.070V_3 + 0.013\pi)}} * J \quad (6-10)$$

6.3.2 贷款利率的分配

按照 6.1.5 模型可得出以下结果，带入贷款利率模型得出最佳年利率为 0.0465。

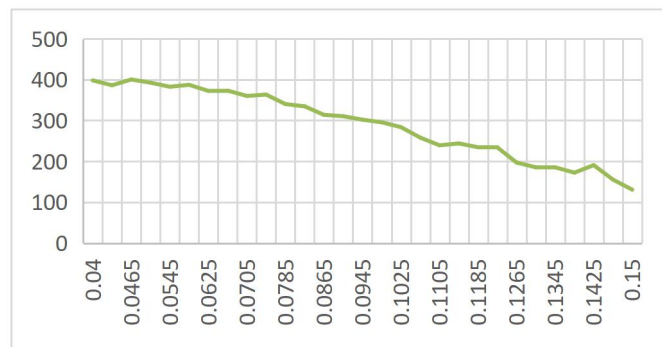


图 6-16

6.3.3 贷款策略求解

根据最 6.3.1 与 6.1.4 按照风险量化指标排序，该模型下可满足排名前 124 的企业贷款。其中贷款额度为模型 6.2.2 分配出的额度。这 124 家公司，详细结果在附录中展示。

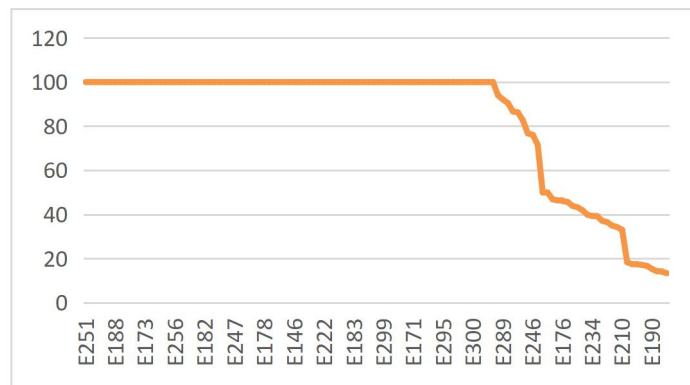


图 6-17 一亿元分配结果图

七、模型的优缺点

7.1 模型的优点

- (1) 建立的信誉评级模型在有信誉评级的情况下预测效果十分优良，同时无信誉评级的情况下适应性也较强，对银行的信贷策略制定具有明显的建设性。
- (2) 建立的利率给定模型很好地反映实际情况，实用性强。
- (3) 综合考虑了多种影响银行信贷决策的显著因素和随机因素，从银行的切身利益出发，深度还原了银行做决策时的考虑路线。

7.2 模型的缺点

- (1) 贷款额度模型存在较大主观决策性。
- (2) 数据量相对缺乏，对模型的检验存在一定的局限。

参 考 文 献

- [1] 张金贵 侯 宇. 基于 Logit 模型的中小企业信贷风险实证分析. 江苏科技大学经济管理学院. 2014.
- [2] Allison P D. Logistic Regression Using the SAS System: Theory and Application[M]. 1999.
- [3] 威廉 H • 格林. 经济计量分析[M]. 北京: 中国社会科学出版社, 1998.
- [4] 单良 乔杨. 数据化风控. 信用评分建模教程. 北京: 电子工业出版社, 2018.
- [5] https://zhuanlan.zhihu.com/p/104599677?utm_source=wechat_session&utm_medium=social&utm_oi=995795697253425152&utm_campaign=shareopen
- [6] 许亚婷, 许宪春, 余航, 杨业伟. 如何准确理解中国 2020 年一季度 GDP 增长数据[J/OL]. 经济学报 1-19[2020-09-13]. <https://doi.org/10.16513/j.cnki.cje.20200811.001>.

附录

附件清单:

附录 1: 数据清洗 java 源程序

附录 2: 数据清洗-MySQL 源程序

附录 3: 求解银行在不同年利率下的总收入 matlab 代码

附录 1 数据清洗-java 源程序

```
1. import com.mapper.JinxiangMapper;
2. import org.apache.ibatis.annotations.Param;
3. import org.junit.jupiter.api.Test;
4. import org.springframework.beans.factory.annotation.Autowired;
5. import org.springframework.boot.test.context.SpringBootTest;
6. import java.util.*;
7. @SpringBootTest
8. class ApplicationTests {
9.     @Autowired
10.     JinxiangMapper jinxiangMapper;
11.
12.     @Test
13.     void fiss() {
14.         for(int i = 304 ; i<426;i++){
15.             jinxiangMapper. countnumin2016(i);
16.             jinxiangMapper. countnumin2017(i);
17.             jinxiangMapper. countnumin2018(i);
18.             jinxiangMapper. countnumin2019(i);
19.             jinxiangMapper. countnumin2020( i);
20.             jinxiangMapper. countnumout2016(i);
21.             jinxiangMapper. countnumout2017( i);
22.             jinxiangMapper. countnumout2018( i);
23.             jinxiangMapper. countnumout2019(i);
24.             jinxiangMapper. countnumout2020(i);
25.
26.             jinxiangMapper. countnumin2016sui(i);
27.             jinxiangMapper. countnumin2017sui(i);
28.             jinxiangMapper. countnumin2018sui(i);
29.             jinxiangMapper. countnumin2019sui(i);
30.             jinxiangMapper. countnumin2020sui(i);
31.
32.             jinxiangMapper. countnumout2016sui(i);
33.             jinxiangMapper. countnumout2017sui(i);
34.             jinxiangMapper. countnumout2018sui(i);
35.             jinxiangMapper. countnumout2019sui(i);
```

```

36.         jinxiangMapper. countnumout2020sui(i);
37.     }
38. }
39.
40. @Test
41. void contextLoads() {
42.
43.     for(int i =312;i<317;i++) {
44.         ArrayList<String> id_2016 = new ArrayList<>();
45.         ArrayList<String> id_2017 = new ArrayList<>();
46.         ArrayList<String> id_2018 = new ArrayList<>();
47.         ArrayList<String> id_2019 = new ArrayList<>();
48.         ArrayList<String> id_2020 = new ArrayList<>();
49.         List<String> sellid_2016 = jinxiangMapper.getdifferntsellid(i, 2016);
50.         List<String> sellid_2017 = jinxiangMapper.getdifferntsellid(i, 2017);
51.         List<String> sellid_2018 = jinxiangMapper.getdifferntsellid(i, 2018);
52.         List<String> sellid_2019 = jinxiangMapper.getdifferntsellid(i, 2019);
53.         List<String> sellid_2020 = jinxiangMapper.getdifferntsellid(i, 2020);
54.         if(sellid_2016.size()!=0)
55.             id_2016 = gettop5(sellid_2016,i,id_2016,2016);
56.         if(sellid_2017.size()!=0)
57.             id_2017 = gettop5(sellid_2017,i,id_2017,2017);
58.         if(sellid_2018.size()!=0)
59.             id_2018 = gettop5(sellid_2018,i,id_2018,2018);
60.         if(sellid_2019.size()!=0)
61.             id_2019 = gettop5(sellid_2019,i,id_2019,2019);
62.         if(sellid_2020.size()!=0)
63.             id_2020 = gettop5(sellid_2020,i,id_2020,2020);
64.
65.         int countrating = Countrating(id_2016, id_2017, id_2018, id_2019, id_2020);
66.         jinxiangMapper.updateinrate(i, countrating);
67.         System.out.println(countrating);
68.
69.     }
70. }
71.
72. public ArrayList<String> gettop5(List<String> sellid_2016 ,int i,ArrayList<String>
    id_2016,int year){
73.     if(sellid_2016.size()==0){
74.         sellid_2016.add("0");
75.         sellid_2016.add("0");
76.         sellid_2016.add("0");
77.         sellid_2016.add("0");
78.         sellid_2016.add("0");
79.     }

```

```
80.     long temp_2016 = 0;
81.     SortedSet ss_2016=new TreeSet();
82.     HashMap<Object, Object> tempres_2016 = new HashMap<>();
83.     for (String s : sellid_2016) {
84.         try{
85.             temp_2016 = jinxiangMapper.gettotlabysellid(s,i,year);
86.         }catch (Exception e)
87.         {
88.             temp_2016=0;
89.             continue;
90.         }
91.         ss_2016.add(temp_2016);
92.         tempres_2016.put(temp_2016,s);
93.     }
94.     List <Long> templist_2016 = new ArrayList<Long>(ss_2016);
95.     Long f1=01;
96.     Long f2=01;
97.     Long f3=01;
98.     Long f4=01;
99.     Long f5=01;
100.    if(templist_2016.size()==1){
101.        f1 = templist_2016.get(templist_2016.size()-1);//第一大
102.    }if(templist_2016.size()>=2){
103.        f2 = templist_2016.get(templist_2016.size() - 2);//第二大
104.    }
105.    if(templist_2016.size()>=3){
106.        f3 = templist_2016.get(templist_2016.size() - 3);//第三大
107.    }
108.    if(templist_2016.size()>=4){
109.        f4 = templist_2016.get(templist_2016.size() - 4);//第三大
110.    }
111.    if(templist_2016.size()>=5){
112.        f5 = templist_2016.get(templist_2016.size() - 5);//第三大
113.    }
114.
115.    String f1id = (String)tempres_2016.get(f1);
116.    String f2id = (String)tempres_2016.get(f2);
117.    String f3id = (String)tempres_2016.get(f3);
118.    String f4id = (String)tempres_2016.get(f4);
119.    String f5id = (String)tempres_2016.get(f5);
120.    if(f1id!=null){
121.        id_2016.add(f1id);
122.    }
123.    if(f2id!=null){
124.        id_2016.add(f2id);
```



```
125.     }
126.     if(f3id!=null){
127.         id_2016.add(f3id);
128.     }
129.     if(f4id!=null){
130.         id_2016.add(f4id);
131.     }
132.     if(f5id!=null){
133.         id_2016.add(f5id);
134.     }
135.     return id_2016;
136. }
137.
138. public int Countrating(List<String> sellid_2016,List<String> sellid_2017,
139. List<String> sellid_2018,List<String> sellid_2019,List<String> sellid_2020){
140.     SortedSet ss=new TreeSet();
141.     for (String s : sellid_2016) {
142.         ss.add(s);
143.     }
144.     for (String s : sellid_2017) {
145.         ss.add(s);
146.     }
147.     for (String s : sellid_2018) {
148.         ss.add(s);
149.     }
150.     for (String s : sellid_2019) {
151.         ss.add(s);
152.     }
153.     for (String s : sellid_2020) {
154.         ss.add(s);
155.     }
156.     List <String> tempreslut = new ArrayList<String>(ss);
157.     HashMap<String, Integer> tempmap = new HashMap<>();
158.     for (String s : tempreslut) {
159.         tempmap.put(s,0);
160.     }
161.
162. //     开始正式遍历
163.     for (String s : sellid_2016) {
164.         tempmap.put(s,tempmap.get(s)+1);
165.     }
166.     for (String s : sellid_2017) {
167.         tempmap.put(s,tempmap.get(s)+1);
168.     }
169.     for (String s : sellid_2018) {
```

```

170.         tempmap.put(s,tempmap.get(s)+1);
171.     }
172.     for (String s : sellid_2019) {
173.         tempmap.put(s,tempmap.get(s)+1);
174.     }
175.     for (String s : sellid_2020) {
176.         tempmap.put(s,tempmap.get(s)+1);
177.     }
178.     System.out.println(tempmap);
179.     int final_re = 0;
180.
181.     for (Integer value : tempmap.values()) {
182.         if(value==0)
183.             final_re = final_re +0;
184.         if(value==5)
185.             final_re = final_re +50;
186.         else if(value==4)
187.             final_re = final_re +40;
188.         else if(value==3)
189.             final_re = final_re +30;
190.         else if(value==2)
191.             final_re = final_re +20;
192.
193.     }
194.     return final_re;
195. }
}

```

附录 2 数据清洗-MySQL 源程序

```

1.  <?xml version="1.0" encoding="UTF-8"?>
2.  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
3.  <mapper namespace="com.free.wislab.mapper.JinxiangMapper">
4.
5.      <!-- 通用查询映射结果 -->
6.      <resultMap id="BaseResultMap" type="com.free.wislab.pojo.Jinxiang">
7.          <result column="id" property="id" />
8.          <result column="fapiao_num" property="fapiaoNum" />
9.          <result column="fapiao_date" property="fapiaoDate" />
10.         <result column="sell_id" property="sellId" />
11.         <result column="total" property="total" />
12.         <result column="suie" property="suie" />
13.         <result column="sum" property="sum" />
14.         <result column="state" property="state" />
15.     </resultMap>

```

```
16.
17. <!-- 通用查询结果列 -->
18. <sql id="Base_Column_List">
19.     id, fapiao_num, fapiao_date, sell_id, total, suie, sum, state
20. </sql>
21. <update id="countnum" >
22.     UPDATE qiyedata SET y2017in = (SELECT SUM(SUM) FROM jinxiang
23.     WHERE id =CONCAT("E",#{e})
24.     AND state = "有效发票"
25.     AND fapiao_date LIKE "%2017%") WHERE id = CONCAT("E",#{e})
26. </update>
27.
28. <update id="countnumin" >
29.     UPDATE qiyedata SET y2016in = (SELECT SUM(SUM) FROM jinxiang
30.     WHERE id =CONCAT("E",#{e})
31.     AND state = "有效发票"
32.     AND fapiao_date LIKE CONCAT("%",#{year},"%")) WHERE id = CONCAT("E",#{e})
33. </update>
34. <update id="countnumin2016" >
35.     UPDATE qiyedata SET y2016in = (SELECT SUM(SUM) FROM jinxiang
36.     WHERE id =CONCAT("E",#{e})
37.     AND state = "有效发票"
38.     AND fapiao_date LIKE "%2016%" )WHERE id = CONCAT("E",#{e})
39. </update>
40. <update id="countnumin2017" >
41.     UPDATE qiyedata SET y2017in = (SELECT SUM(SUM) FROM jinxiang
42.     WHERE id =CONCAT("E",#{e})
43.     AND state = "有效发票"
44.     AND fapiao_date LIKE "%2017%" )WHERE id = CONCAT("E",#{e})
45. </update>
46. <update id="countnumin2018" >
47.     UPDATE qiyedata SET y2018in = (SELECT SUM(SUM) FROM jinxiang
48.     WHERE id =CONCAT("E",#{e})
49.     AND state = "有效发票"
50.     AND fapiao_date LIKE "%2018%" )WHERE id = CONCAT("E",#{e})
51. </update>
52. <update id="countnumin2019" >
53.     UPDATE qiyedata SET y2019in = (SELECT SUM(SUM) FROM jinxiang
54.     WHERE id =CONCAT("E",#{e})
55.     AND state = "有效发票"
56.     AND fapiao_date LIKE "%2019%" )WHERE id = CONCAT("E",#{e})
57. </update>
58. <update id="countnumin2020" >
59.     UPDATE qiyedata SET y2020in = (SELECT SUM(SUM) FROM jinxiang
60.     WHERE id =CONCAT("E",#{e})
```

```

61.         AND state = "有效发票"
62.         AND fapiao_date LIKE "%2020%" ) WHERE id = CONCAT("E",{e})
63.     </update>
64.
65.     <update id="countnumout2016" >
66.         UPDATE qiyedata SET y2016out = (SELECT SUM(total) FROM xiaoxiang
67.         WHERE id =CONCAT("E",{e})
68.         AND state = "有效发票"
69.         AND fapiao_date LIKE "%2016%" )WHERE id = CONCAT("E",{e})
70.     </update>
71.     <update id="countnumout2017" >
72.         UPDATE qiyedata SET y2017out = (SELECT SUM(total) FROM xiaoxiang
73.         WHERE id =CONCAT("E",{e})
74.         AND state = "有效发票"
75.         AND fapiao_date LIKE "%2017%" )WHERE id = CONCAT("E",{e})
76.     </update>
77.     <update id="countnumout2018" >
78.         UPDATE qiyedata SET y2018out = (SELECT SUM(total) FROM xiaoxiang
79.         WHERE id =CONCAT("E",{e})
80.         AND state = "有效发票"
81.         AND fapiao_date LIKE "%2018%" )WHERE id = CONCAT("E",{e})
82.     </update>
83.     <update id="countnumout2019" >
84.         UPDATE qiyedata SET y2019out = (SELECT SUM(total) FROM xiaoxiang
85.         WHERE id =CONCAT("E",{e})
86.         AND state = "有效发票"
87.         AND fapiao_date LIKE "%2019%" )WHERE id = CONCAT("E",{e})
88.     </update>
89.     <update id="countnumout2020" >
90.         UPDATE qiyedata SET y2020out = (SELECT SUM(total) FROM xiaoxiang
91.         WHERE id =CONCAT("E",{e})
92.         AND state = "有效发票"
93.         AND fapiao_date LIKE "%2020%" )WHERE id = CONCAT("E",{e})
94.     </update>
95.     <update id="countnumin2016sui">
96.         UPDATE qiyedata SET y2016in_sui = (SELECT SUM(suie) FROM jinxiang
97.         WHERE id =CONCAT("E",{e})
98.         AND state = "有效发票"
99.         AND fapiao_date LIKE "%2016%" )WHERE id = CONCAT("E",{e})
100.    </update>
101.    <update id="countnumin2017sui">
102.        UPDATE qiyedata SET y2017in_sui = (SELECT SUM(suie) FROM jinxiang
103.        WHERE id =CONCAT("E",{e})
104.        AND state = "有效发票"
105.        AND fapiao_date LIKE "%2017%" )WHERE id = CONCAT("E",{e})

```

```
106. </update>
107. <update id="countnumin2018sui">
108.     UPDATE qiyedata SET y2018in_sui = (SELECT SUM(suie) FROM jinxiang
109.     WHERE id =CONCAT("E",#{e}))
110.     AND state = "有效发票"
111.     AND fapiao_date LIKE "%2018%" )WHERE id = CONCAT("E",#{e})
112. </update>
113. <update id="countnumin2019sui">
114.     UPDATE qiyedata SET y2019in_sui = (SELECT SUM(suie) FROM jinxiang
115.     WHERE id =CONCAT("E",#{e}))
116.     AND state = "有效发票"
117.     AND fapiao_date LIKE "%2019%" )WHERE id = CONCAT("E",#{e})
118. </update>
119. <update id="countnumin2020sui">
120.     UPDATE qiyedata SET y2020in_sui = (SELECT SUM(suie) FROM jinxiang
121.     WHERE id =CONCAT("E",#{e}))
122.     AND state = "有效发票"
123.     AND fapiao_date LIKE "%2020%" )WHERE id = CONCAT("E",#{e})
124. </update>
125. <update id="countnumout2016sui">
126.     UPDATE qiyedata SET y2016out_sui = (SELECT SUM(suie) FROM xiaoxiang
127.     WHERE id =CONCAT("E",#{e}))
128.     AND state = "有效发票"
129.     AND fapiao_date LIKE "%2016%" )WHERE id = CONCAT("E",#{e})
130. </update>
131. <update id="countnumout2017sui">
132.     UPDATE qiyedata SET y2017out_sui = (SELECT SUM(suie) FROM xiaoxiang
133.     WHERE id =CONCAT("E",#{e}))
134.     AND state = "有效发票"
135.     AND fapiao_date LIKE "%2017%" )WHERE id = CONCAT("E",#{e})
136. </update>
137. <update id="countnumout2018sui">
138.     UPDATE qiyedata SET y2018out_sui = (SELECT SUM(suie) FROM xiaoxiang
139.     WHERE id =CONCAT("E",#{e}))
140.     AND state = "有效发票"
141.     AND fapiao_date LIKE "%2018%" )WHERE id = CONCAT("E",#{e})
142. </update>
143. <update id="countnumout2019sui">
144.     UPDATE qiyedata SET y2019out_sui = (SELECT SUM(suie) FROM xiaoxiang
145.     WHERE id =CONCAT("E",#{e}))
146.     AND state = "有效发票"
147.     AND fapiao_date LIKE "%2019%" )WHERE id = CONCAT("E",#{e})
148. </update>
149. <update id="countnumout2020sui">
150.     UPDATE qiyedata SET y2020out_sui = (SELECT SUM(suie) FROM xiaoxiang
```

```

151.         WHERE id =CONCAT("E",#{e})
152.         AND state = "有效发票"
153.         AND fapiao_date LIKE "%2020%" )WHERE id =  CONCAT("E",#{e})
154.     </update>
155.     <update id="updateinrate">
156.         UPDATE qiyedata SET out_rate = #{e2} WHERE id =  CONCAT("E",#{e})
157.     </update>
158.     <select id="getdifferenttsellid" resultType="java.lang.String">
159.         SELECT DISTINCT sell_id FROM xiaoxiang WHERE id = CONCAT("E",#{e}) and state
        = "有效发票"and fapiao_date like CONCAT("%",#{year},"%") ;
160.     </select>
161.     <select id="gettottlabysellid" resultType="java.lang.Long">
162.
163.         select sum(sum) from xiaoxiang where id =CONCAT("E",#{e}) and state = "有效发
        票" and sell_id=#{sellid} and fapiao_date like CONCAT("%",#{year},"%") ;
164.     </select>
165.
166.
</mapper>

```

附录 3 求解银行在不同利率下的总收入 matlab 代码

```

1. qiye = load ('matlab3.mat') ;
2. bank = load ('bank.mat') ;
3. data = qiye.newdata3;
4. bankdata = bank.bank;
5. C= rand(301,29);
6.
7. for bank_id = 1:29
8.     for ii= 1:301
9.         % 等级
10.         area = data(ii,2);
11.         if(area == 1)
12.             money = data(ii,5)*bankdata(bank_id,1)*(1-bankdata(bank_id,2))*data(ii,4);
13.         end
14.         if(area == 0.67)
15.             money = data(ii,5)*bankdata(bank_id,1)*(1-bankdata(bank_id,3))*data(ii,4);
16.         end
17.         if(area == 0.33)
18.             money = data(ii,5)*bankdata(bank_id,1)*(1-bankdata(bank_id,4))*data(ii,4);
19.         end
20.         if(area == 0)
21.             money = 0;

```

```
22.     end
23.     C(ii,bank_id)= money;
24.     end
25. end
```