

# 银行对中小微企业信贷策略模型

## 摘要

本文是对银行贷款策略的研究，通过对企业数据的分析，建立模型评估贷款的风险、为银行提供贷款的策略。

针对问题一中评估风险的问题，我们引入两个一级指标“公司自身信誉评分”、“行业风险评分”来构建风险评估模型。“公司自身信誉评分”包含四个二级指标（信誉等级、公司资金稳定性、公司支票作废比等），运用熵权法求出二级指标对应一级指标“公司信誉评分”的权重，并计算一级指标得分；对于“行业风险评分”的量化，我们查阅相关资料得到行业内企业平均倒闭率，用它来表征行业风险。此后，运用层次分析法求得两个一级指标的权重，采用基于带权重的 TOPSIS 模型计算各个企业贷款风险的得分。

针对问题一中为银行提信贷策略的问题，在我们的策略中，信誉等级为 D 的企业一律不贷款，风险评分未达到所设置的平均风险指数的一律不贷。在剩余的企业中，我们选取了两组指标来确定给各企业的贷款数量。一是评估风险的结果——风险评分，另一个为企业的年利润。我们采用系统聚类的方法，将企业分为合适的类别。根据各类的“可靠程度”确定类与类之间各企业的贷款比例，按照贷款总额固定的原则，求出每类中各个企业所分的贷款比例。对于贷款利率的问题，我们通过拟合的方法得到了不同信誉等级下，利率与客户流失率的关系，建立银行利润最大化模型。求解出不同信誉等级的客户对应的不同利率。

针对问题二，我们创新性地对附件 1 中的数据提取了十八项指标，并通过基于遗传算法的 BP 神经网络模型对指标数据进行训练，经过第一轮训练后筛选出十个指标进行学习。然后将附件二中相应指标数据输入训练好的神经网络模型，求得附件 2 中各企业的虚拟信誉评级。此后，问题二与问题一的求解十分相似，问题一中求解风险评分、贷款额度比例、贷款利率的模型对问题二同样适用，由此可得出银行对不同企业的信贷策略。

针对问题三，我们借鉴林毅夫等学者利用国有企业就业人员占行业就业人员的比重作为衡量行业对劳动力的依赖程度的工具变量的方法，去可以估算不同行业受疫情影响的程度。利用该指数将问题一模型中行业风险评分进行修正，然后再继续套用问题一的信贷模型。我们以突发因素为新冠肺炎为例，求解了在疫情影响下，银行放贷给附件 2 中企业的信贷策略。

行业突发因素影响指数 = (行业所有单位就业人员数 - 行业国有单位就业人员数) / 行业所有单位就业人员数

**关键字：** 熵权法 TOPSIS 模型 系统聚类模型 遗传算法 BP 神经网络 行业突发因素影响指数

## 一、问题重述

现实中，中小微企业规模相对较小，贷款时也缺少抵押资产，所以银行通常要依据信贷政策、企业的交易票据和上下游企业的影响力，给实力强、供求关系稳定的企业提供贷款，同时对信誉高、信贷风险小的企业提供利率优惠。银行在制定信贷策略时会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决定是否放贷以及贷款额度、利率和期限等。

某银行对于确定放贷的企业的贷款额度是 10 到 100 万元；年利率是 4% 到 15%；贷款期限为 1 年。现提供 123 家有信贷企业记录的相关数据、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。基于这些数据，需要建立数学模型以解决以下问题：

- 对 123 家有信贷记录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。
- 在问题一的基础上，对 302 家无信贷记录企业的信贷风险进行量化，并给出该银行在年度信贷总额为 1 亿元时针对这些企业制定的信贷策略。
- 现实中企业的生产经营和经济效益可能会收到一些突发因素影响，同时突发因素对不同行业、不同类别的企业往往会有不同的影响。综合考虑（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

对于问题一，我们需要评估 123 家企业的贷款风险，并给出该银行在年度贷款总额固定时对这些企业的信贷策略。那么我们将面临以下两个问题：

### A. 如何评估风险呢？

针对这一问题，我们认为评估一家公司的信贷风险要从两个方面考虑，一个是公司自身因素，另一个公司所在行业的经济环境。从公司自身因素来说，稳定的现金流水、对订单的履行程度、销售过程中被退款的频率以及银行根据以往贷款记录给予公司的信誉评级等这些指标均可以作为证明该公司具有良好的还贷能力的证据，也即银行贷款给该公司时所面临的风险。另一方面，从行业风险方面看，身处不同行业的公司在进行企业活动时所面临的资金链短缺的风险也是不一样的，对于处于金融业的公司，其面临的行业风险相较于处于制造业的公司要大很多。因此，公司所处的行业类型一定程度上也反映了银行贷款给公司时所面临的风险。

基于以上两个方面的分析，我们建立两个一级指标“公司自身信誉评分”、“行业风险评分”，并将其量化化，综合评估银行贷款给该公司时所面临的风险。

B. 如何根据风险指数等已知数据确定信贷策略呢？

银行的放贷策略分为两个，一个是放贷款数所占总款数的比例，另一个是放贷时的利率。针对确定放贷款数所占总款数比例问题，经过探究发现银行需要在考虑风险的同时，也需要考虑该公司的现金储备量。公司的现金储备量越高表示其应对贷款危机的能力越强，当公司经营不善时，现金储备量大的公司归还贷款的可能性越大。因此，银行放给该公司的贷款也应该越多。针对确定放贷利率的问题，对于不同的信誉等级，其放贷利率应是不同的。银行对某一企业的收益 = 放贷款数 × 放贷利率 × (1 - 该客户流失率)。由题目提供的数据可知，客户流失率与企业的放贷利率是具有一定的函数关系的。我们将上文客户流失率的替换为关于放贷利率的函数，根据收益最大的原则，我们可以求出每一个信誉等级的企业的利率。

**对于问题二**，我们需要对附件 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。在处理这一问题时，302 家企业均无贷款记录，这也就意味着他们没有相应的信誉评级。倘若有信用评级，问题二的求解则与问题一的求解十分相似，问题一中求解风险评分、贷款额度比例、贷款利率的模型对问题二同样适用，由此可得出银行的放贷策略。

然而，附件 2 中 302 家企业无贷款记录，那么如何预测这些无贷款记录企业的信誉评级呢？这是着重解决的问题。我们通过神经网络模型对附件 1 中的数据进行学习，从而给附件 2 中的企业虚拟一个信誉评级，再通过问题一的模型求解放贷策略。

**对于问题三**，我们需要在考虑突发因素影响的情况下，为银行提供合理的信贷策略。突发因素的影响是对行业的冲击或者促进。在现实中，越依赖劳动力的行业受突发因素的影响越大。我们借鉴林毅夫等学者利用国有企业就业人员占行业就业人员的比重，作为衡量行业对劳动力的依赖程度的工具变量的做法，去估算不同行业受突发因素影响的程度，得到行业受突发因素影响的指数（指数越大，受突发因素的影响也就越大）。利用该指数将问题一模型中行业风险评分进行修正，修正方式为将该指数与行业内企业平均倒闭概率做乘积作为新的行业风险评分。我们以突发因素为新冠肺炎为例，求解了在疫情影响下，银行放贷给附件 2 中企业的信贷策略。

### 三、模型假设

- 讨论对象银行为商业银行，以取得最大利益为目标，不带有公益性质。
- 各企业均提供了对应时间范围内完整的交易票据信息。
- 各企业的交易票据信息与具体情况与数据中企业名称体现的行业类别相符合。
- 银行利率和贷款额度的范围，不受特定行业的特殊规定所限制。

#### 四、符号说明

符号	含义
$X_i$	公司销项发票第 $i$ 个月销售的数额
$P(i)$	公司到达第 $i$ 个月产生的累计利润总额
$J_i$	公司进项发票第 $i$ 个月支出的数额
$R^2$	拟合优度
$\alpha$	公司支票作废比
$C_{void}$	公司作废支票数
$C_{sum}$	支票总数
$\beta$	被用户退款数额比例
$M_{back}$	被客户返回商品的金额
$M_{sum}$	卖出的总金额
$f_{risk}^-$	平均风险评分
$W$	二级指标的权重向量
$Q$	一级指标权重向量
$M_i$	第 $i$ 类别公司的贷款数额
$\varepsilon_i$	第 $i$ 类公司中每个公司贷款数占银行贷款总数的比例
$N_i$	第 $i$ 类中公司个数
$M$	该年度银行贷款总额
$a$	为公司贷款额度比例区间的中间值
$r_A$	A 信誉的公司贷款利率；
$\gamma_A$	A 信誉的公司利率 $r_A$ 下的客户流失率

## 五、模型建立及求解

### 5.1 问题一模型的建立与求解

在问题一中，我们需要评估 123 家企业的贷款风险，并给出该银行在年度贷款总额固定时对这些企业的信贷策略。

#### 5.1.1 风险评估指标的选取及量化

为评估企业贷款风险，我们需要建立评价指标以此来构建风险评估体系。在选取指标时，我们采用定性定量分析相结合的原则，通过题目提供及查阅到的相关数据，经过一定的处理和筛选，确定了两个一级指标，四个二级指标。两个一级指标即“自身信誉评分”、“行业风险评分”，四个二级指标即“公司资金稳定性”、“公司支票作废比”、“公司信誉评级”、“被客户退款比例”。指标间的隶属关系如下表：

表 1 Add caption

一级指标	二级指标
自身信誉评分	公司资金稳定性
	公司支票作废比
	公司信誉评级
	被客户退款比例
行业风险评分	行业内企业平均倒闭概率

#### 5.1.2 二级指标含义详解

- 公司资金稳定性  
该指标衡量一个公司的持续造血能力，稳定资金比例越高，企业应对中长期资产负债结构性问题的能力越强，偿还贷款的能力也越强。
- 公司支票作废比  
该指标衡量公司在经济活动中是否能够做到长远发展，在经济活动中是否有出尔反尔的行为，衡量公司在经济活动中的信誉。
- 公司信誉评级

该指标是银行根据以往还贷情况，对公司信誉的一个评级，对公司的贷款信誉具备较强的参考性

- **行业内企业平均倒闭概率**

该指标衡量客户对公司产品的认可程度，衡量企业在消费者中的形象，从消费者的角度看待该公司的经济活动中作风。

### 5.1.3 二级指标的量化

#### A. “公司资金稳定性”的量化

在经营过程中，公司资金为起始资金与利润总额的加和，由于各个公司的起始资金为一个定值，那么公司资金的稳定性也就是利润总额的稳定性。

我们以月为时间单位，通过题目提供的进项发票信息与销项发票信息计算出每月每个企业的月利润，则第  $i$  月产生的累计利润总额为：

$$P(i) = \sum_{i=1}^n (X_i - J_i)$$

$X_i$  为公司销项发票第  $i$  个月销售的数额； $J_i$  为公司进项发票第  $i$  个月支出的数额； $P(i)$  是公司到达第  $i$  个月产生的利润总额。易知，截至到第  $i$  个月的利润总额  $P(i)$  与时间相关，我们将时间序列下的到达每个月的利润总额（离散型变量）进行函数拟合，拟合的结果为  $Y(x)$  通过计算拟合函数各月份的利润总额  $Y(i)$ ，得到两者间的拟合优度如下：

$$R^2 = 1 - \frac{\sum_{j=1}^i (P_i - Y_i)^2}{\sum_{j=1}^i (P_i - \bar{P})^2} = 1 - \frac{RMSE}{Var}$$

定义拟合优度  $R^2$  表征  $P$  的数据散点到拟合曲线集中程度，拟合优度越大，散点到拟合曲线的集中程度也越大。说明其越加稳定于拟合曲线周围，可以表征  $P$  的稳定性，也即可以表征公司资金的稳定性。并且由于  $R^2$  是一个无计量单位的数值，它可以排除单位量级对集中程度的干扰。

#### B. “公司支票作废比”的量化

公司支票作废比是衡量公司信誉的重要指标，作废比越高，说明该公司再经营过程中出尔反尔的概率也就越高，信誉则就偏低。其计算公式为：

$$\alpha = \frac{C_{void}}{C_{sum}}$$

$\alpha$  代表公司支票作废比； $C_{void}$  代表作废支票数； $C_{sum}$  代表支票总数

#### C. “公司信誉评级”的量化

公司信用评级是银行根据以往还贷情况，对公司信誉的一个评级，对公司的贷款信誉具备较强的参考性。信誉越高，说明银行贷款风险越小。以下是各个等级对应的分数：

#### D. “被用户退款比例”的量化

表 2 信誉评级对应分数

等级	A	B	C	D
分数	4	3	2	1

“被用户退款比例”表征了客户对公司产品的认可程度，衡量企业在消费者中的形象，其计算公式如下：

$$\beta = \frac{M_{back}}{M_{sum}}$$

$\beta$  代表被用户退款比例； $M_{back}$  被客户返回商品的金额； $M_{sum}$  卖出的总金额

#### E. 行业内企业平均倒闭概率”的量化 [4]

行业内企业平均倒闭概率表征了该行业一段时间内以来的发展前景，倒闭概率越大，则银行放贷给该行业中企业的风险越大。我们通过查阅相关文献，得到下表：

表 3 各行业平均倒闭概率

行业类型	制造业	租赁和商业服务	居民服务和其他服务
平均倒闭概率	2%	0.10%	1%
行业类型	电子信息业	物流业	水利、环境和公共设施管理
平均倒闭概率	4.50%	2%	0.05%
行业类型	交通运输业	销售业	科学研究、技术服务和地质勘查
平均倒闭概率	1%	6%	4.50%
行业类型	批发与零售	建筑业	电力、燃气及水的生产和供应
平均倒闭概率	1%	2.20%	0.40%
行业类型	金融业	房地产业	卫生、社会保障和社会福利
平均倒闭概率	19%	1%	0.05%
行业类型	住宿和餐饮业	个体经营	文化、体育、娱乐业
平均倒闭概率	8.20%	14%	2%

#### 5.1.4 基于熵权法的 TOPSIS 风险评估模型的建立及求解 [3]

我们评估一家公司的信贷风险要从两个方面考虑，一个是公司自身因素，另一个公司所在行业的经济环境。两者之间存在着一定的关联性，能够相互制约，共同作用形成一个有机的整体。

从公司自身因素来说，稳定的现金流水、对订单的履行程度、销售过程中被退款的频率以及银行根据以往贷款记录给予公司的信誉评级等这些指标均可以作为证明该公司具有良好的还贷能力的证据。从银行的角度，可以参考这些方面对公司偿还贷款的能力和意愿进行评估。

另一方面，从行业风险方面看，身处不同行业的公司在进行企业活动时所面临的资金链短缺的风险也是不一样的，对于处于金融业的公司，其面临的行业风险相较于处于制造业的公司要大很多。因此，公司所处的行业类型一定程度上也反映了银行贷款给公司时所面临的风险。

基于以上两个方面考虑，我们建立如下评价体系：

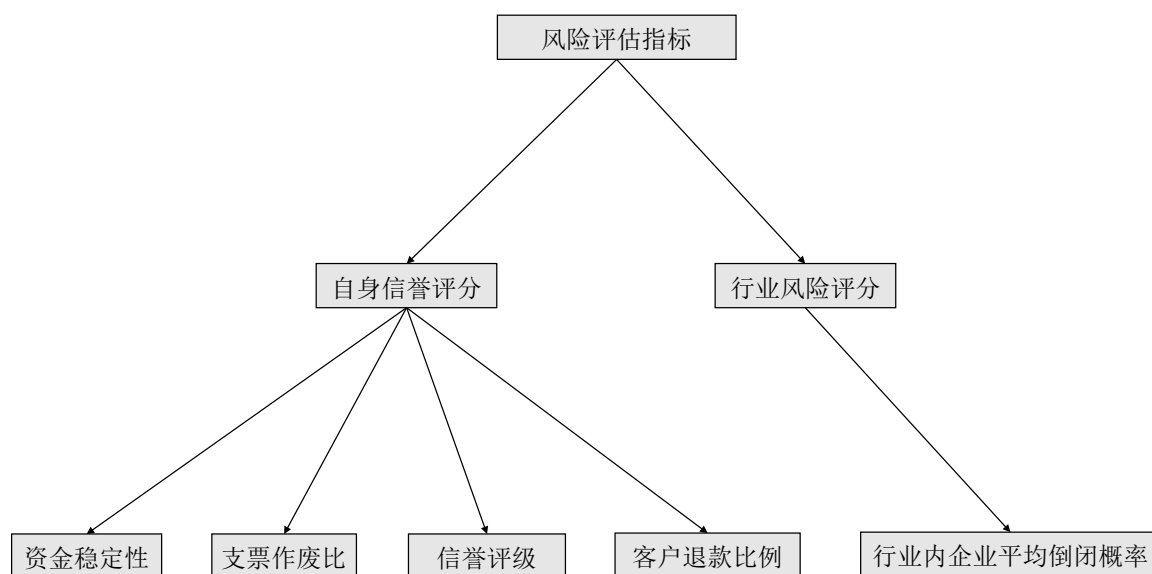


图 1 风险评价体系

由于我们选择了两个一级指标、五个二级指标作为评价依据，这些指标对企业贷款风险的影响存在很大的差异，这样在进行评价时，容易出现仅根据个别指标，使评价带有主观性。为了更客观、科学的进行评价，综合考虑各个指标的情况，我们选择基于熵权法的 TOPSIS 风险评估模型。

熵权法是一种客观赋权，它依据的原理是，指标的变化程度越小，对应的权重应该越低。



TOPSIS 方法是一种多属性决策分析方法。TOPSIS 的基本思想是，应该选择距离正的理想解几何距离最近，并且距离负的理想解几何距离最远的方案。TOPSIS 是一种补偿的集成方案，比较一组候选方案的手段包括确定每一指标的权重，标准化每一指标的得分，计算每一方案与理想方案间的几何距离作为每一指标的最好得分。

以上两个模型的结合使我们的指标权重不再具有盲目性，同时能够更加实际地、科学地对企业贷款风险进行评估。以下是基于熵权法的 TOPSIS 风险评估模型的求解过程：

**Step1. 运用熵权法求取一级指标“自身信誉评分”下四个二级指标的权重**

有 123 个要评价的公司，4 个二级评价指标构成的矩阵如下（ $n=123$ ， $m=4$ ）：

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

1. 设  $x_i$  为一组指标序列，确定正向化矩阵  $X$

极小型指标转化为极大型指标

$$\tilde{x}_i = \max -x$$

中间型指标转化为极大型指标

设最佳的数值为  $x_{\text{best}}$ ，则

$$\tilde{x}_i = 1 - \frac{|x_i - x_{\text{best}}|}{\max \{|x_i - x_{k=x}|\}}$$

区间指标转化为极大型指标

设最佳区间为  $[c, d]$ ，则：

$$M = \max \{a - \min \{x_i\}, \max \{x_i\} - b\}$$

$$\tilde{x}_i = \begin{cases} 1 - \frac{a-x}{M} & x < a \\ 1 & a \leq x \leq b \\ 1 - \frac{x-b}{M} & x > b \end{cases}$$

2. 确定标准化矩阵  $Z$

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{123} x_{ij}^2}}$$

3. 计算概率矩阵  $P$

$$p_{ij} = \frac{\tilde{z}_{ij}}{\sum^n \tilde{z}_{ij}}$$

4. 计算每个指标的信息熵，并计算信息效用值，并归一化得到每个指标的权重。

对于第  $j$  个指标而言：

$$e_j = -\frac{1}{\ln 123} \sum_{i=1}^n p_{ij} \ln(p_{ij}) (j = 1, 2, \dots, 4)$$

$$d_j = 1 - e_j$$

$$W_j = d_j / \sum_{j=1}^m (j = 1, 2, \dots, 4)$$

**Step2.** 将四个二级指标映射到一级指标“自身信誉评分”对于每一个企业而言，设  $z = (z_1, z_2, \dots, z_n)$  为标准化后的二级指标数据， $y$  为一级指标“自身信誉评分”的数据，则：

$$y = \sum_{i=1}^n W_i z_i$$

**Step3.** 运用层次分析法并查阅相关资料两个一级指标“自身信誉评分”、“行业风险评分”进行赋权

$$Q_j = (q_1, q_2)$$

**Step4.** 基于带权重的 TOPSIS 模型用 Python 计算银行对每个企业放贷时的风险评分

我们有 123 个要评价的对象，2 个评价指标的标准化矩阵：

$$Z = \begin{bmatrix} z_{11} & z_{21} \\ z_{21} & z_{22} \\ \vdots & \vdots \\ z_{123\ 1} & z_{123\ 2} \end{bmatrix}$$

1. 求出每一列的最大值和最小值

$$Z^+ = (\max z_{11}, z_{21}, \dots, z_{n1}, \max z_{12}, z_{22}, \dots, z_{n2}, \dots, \max z_{1m}, z_{2m}, \dots, z_{nm})$$

$$Z^- = (\min z_{11}, z_{21}, \dots, z_{n1}, \min z_{12}, z_{22}, \dots, z_{n2}, \dots, \min z_{1m}, z_{2m}, \dots, z_{nm})$$

2. 求出每一个评价对象与最大值和最小值之间的距离

$$D_i^+ = \sqrt{\sum_{j=1}^m \omega_j (Z_j^+ - z_{ij})^2}$$

$$D_i^- = \sqrt{\sum_{j=1}^m \omega_j (Z_j^- - z_{ij})^2}$$

其中  $\omega_j$  表示第  $j$  个指标的权重 3. 计算出第  $i$  个评价对象未归一化的得分

$$S_i = \frac{D_i^-}{D_i^+ + D_i^-}$$

4. 将所有得分进行归一化处理

Step5. 求解结果及其分析

求得一级指标“自身信誉评分”下四个二级指标的权重为：

$$W = (0.364, 0.193, 0.387, 0.056)$$

求得两个一级指标“自身信誉评分”、“行业风险评分”的权重为：

$$Q = (0.601, 0.399)$$

求得 123 家企业贷款时的风险评分，风险评分越高，贷款风险越小。在以上 123 家企业中，有部分贷款存在违约的状况，针对有违约状况的企业，其风险评分需要修正，通过查阅相关文献，引入修正系数  $r_f = 1.5$ ，即有违约记录的企业，其风险评分为原评分的 1.5 倍。

对于银行对企业是否放贷的决策，我们引入平均风险评分：

$$f_{risk}^- = \frac{100}{n}$$

n 为企业数量

平均风险指数表征在归一百化的风险评分中正常风险的水平。

若  $f_{risk} \leq f_{risk}^-$ ，说明向该公司放贷，风险高于正常水平，选择对该企业不放贷。

若  $f_{risk} > f_{risk}^-$ ，说明向该公司放贷，风险低于正常水平，选择对该企业放贷。

下表为修正后的风险评分，其中，标注红色的是风险评分打不到平均风险评分的企业，决定不对其放贷

标注蓝色的是虽然风险评分达到平均风险评分，但是其信誉等级为 D，决定不对其放贷。

表 4 附件 1 中 123 家企业风险评分

企业代号	风险评分	企业代号	风险评分	企业代号	风险评分	企业代号	风险评分
E59	1.448021	E2	0.945176	E57	0.768566	E27	0.664794
E48	1.411143	E120	0.929558	E72	0.766355	E108	0.662973
E62	1.340398	E95	0.922699	E40	0.763206	E99	0.660828
E65	1.325993	E6	0.922287	E77	0.762510	E109	0.648488
E61	1.320008	E1	0.917002	E87	0.762099	E21	0.645590
E30	1.305397	E18	0.908652	E68	0.761044	E36	0.645430
E54	1.268272	E74	0.898691	E47	0.756611	E5	0.642690

企业代号	风险评分	企业代号	风险评分	企业代号	风险评分	企业代号	风险评分
E51	1.229111	E97	0.886116	E41	0.755953	E71	0.619154
E17	1.165175	E32	0.88082	E60	0.755473	E113	0.618864
E49	1.140178	E93	0.871196	E90	0.750691	E123	0.609019
E67	1.139681	E58	0.869915	E88	0.747523	E92	0.605722
E19	1.128735	E122	0.867098	E94	0.743811	E52	0.602870
E31	1.114574	E38	0.866309	E34	0.733112	E115	0.600872
E25	1.085956	E22	0.862755	E43	0.721654	E56	0.598756
E63	1.083247	E98	0.858974	E89	0.720942	E75	0.598218
E29	1.081920	E8	0.857435	E44	0.718519	E37	0.591236
E119	1.068687	E12	0.847939	E46	0.717941	E112	0.583590
E64	1.038323	E91	0.842618	E78	0.714434	E114	0.571304
E102	1.030024	E83	0.825524	E79	0.706041	E118	0.564321
E42	1.016824	E103	0.823321	E117	0.700692	E53	0.547818
E66	1.014201	E85	0.818764	E116	0.695020	E80	0.540044
E100	1.011421	E76	0.810223	E82	0.694757	E33	0.533628
E50	0.992704	E10	0.809592	E11	0.692319	E86	0.527287
E9	0.988140	E69	0.807914	E121	0.685897	E107	0.524482
E84	0.984259	E104	0.798613	E3	0.679380	E20	0.515188
E24	0.970588	E110	0.797932	E55	0.678744	E111	0.451601
E15	0.955375	E73	0.79081	E4	0.678587	E106	0.435394
E16	0.954720	E28	0.786076	E35	0.676743	E14	0.402459
E13	0.950484	E105	0.782151	E26	0.673915	E23	0.353514
E7	0.950475	E39	0.778447	E101	0.672898	E45	0.291693
E81	0.949122	E70	0.774035	E96	0.664943		

### 5.1.5 基于系统聚类确定贷款比例模型建立及求解

我们认为贷款数额的多少与两个因素有关，分别为“公司实力”及“公司的风险评分”。公司实力可以运用公司年平均利润（2018 和 2019 年利润的均值）来表征。在综合考虑公司实力及风险评分的数据后，我们计划运用聚类算法，将企业分为五个等级，对应五个不同的贷款额度。

系统聚类的算法是通过计算两类数据点间的距离，对最为接近的两类数据点进行组合，并反复迭代这一过程，直至将所有数据点组合成一类，并生成聚类谱系图，最后可以根据肘部法则，来估计出最优的聚类数量。系统聚类的关键在于距离计算公式的选取，我们采用欧氏距离来计算对象与对象之间的距离，即：

$$d(\vec{x}_i, \vec{x}_j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2}$$

采用组间平均连接法来计算类与类之间的距离（两个类共有  $n$  个对象，不计类内对象间的距离），即：

$$D(G_p, G_q) = \frac{\sum_{i=1}^n d_i}{n}$$

因为聚类谱系图并不能自动确定最优的聚类数量，所以我们可以利用肘部法则，通过图形来大致估计出最优的聚类数量。

我们将 123 家企业划分到  $K$  个类中 ( $K \leq n-1$ )，用  $D_k$  表示第  $k$  个类 ( $k = 1, 2, \dots, K$ )，且该类重心的位置记为  $w_k$ ，类中对象的位置记为  $x_i$ ，对象的个数记为  $m_k$ ，那么第  $k$  个类的畸变程度为：（这里的绝对值符号代表距离，下同）

$$j_k = \sum_{i \in D_k}^{m_k} |x_i - w_k|^2$$

则定义所有类的总畸变程度（又称聚合系数）：

$$J = \sum_{k=1}^K j_k = \sum_{k=1}^K \sum_{i \in D_k}^{m_k} |x_i - w_k|^2$$

以聚类的类别数  $K$  为横坐标，聚合系数  $J$  为纵坐标，画出聚合系数折线图，再根据肘部法则，确定最优的聚类数量。以下是利用 SPSS 根据聚类原理求解出的聚类结果：

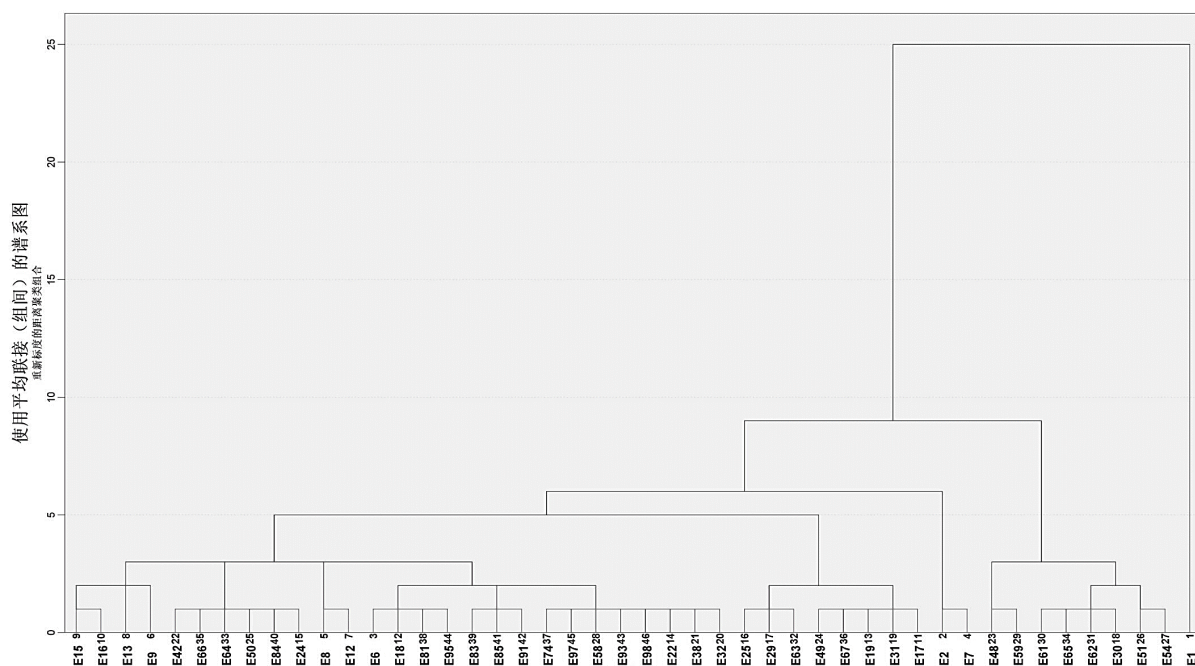


图2 聚类谱系图

由聚合系数折线图可看到，这些企业分为五类最为合适，同时在谱系图中我们看到E1企业最为特殊，被单独归为一类，经过查阅E1企业相关指标，我们发现E1企业每年都在赔钱（即公司实力较差），然而其风险评分却很高，信誉等级为A，没有违约记录。这表明虽然该公司经营状况不佳，但仍然遵守着信用。因此，对于该企业我们建议银行仍选择放贷，但放贷数目为最低限额10万元。

对于其他四类的分类结果，如下表所示：

表5 企业分类结果

第一类	E48, E59, E61, E65, E62, E30, E51, E54
第二类	E2, E7
第三类	E25, E29, E63, E49, E67, E19, E31, E17
	E15, E16, E13, E9, E42, E66, E64, E50, E84, E24
第四类	E8, E12, E6, E18, E81, E95, E83, E85, E91, E74
	E97, E58, E93, E98, E22, E38, E32

表 6 各分类指标的比较

第一类	第三类	第三类	第四类
风险评分较高	风险评分中等	风险评分中等	风险评分较低
公司实力中等	公司实力较高	公司实力中等	公司实力中等

经过比较可知，对于第  $i$  类别公司的贷款数额  $M_i$  应有如下关系：

$$M_1 > M_2 > M_3 > M_4$$

第四类每个公司的贷款数额应为最低，因此，假设以第四类中每个公司贷款所占比例为基准，我们通过查阅相关文献，建立其他三类公司与第四类公司贷款数所占比例的关系如下：

$$\varepsilon_1 = 2\varepsilon_4$$

$$\varepsilon_2 = 1.5\varepsilon_4$$

$$\varepsilon_3 = 1.2\varepsilon_4$$

$$\varepsilon_4 = \varepsilon_4$$

$$N_1\varepsilon_1 + N_2\varepsilon_2 + N_3\varepsilon_3 + N_4\varepsilon_4 = 1$$

$\varepsilon_i$  为第  $i$  类中每个公司贷款数所占银行贷款总数的比例， $N_i$  为第  $i$  类中公司个数

由上述关系可解得各个类别中每个公司贷款数占银行贷款总数的比例，结果如下：

$$\varepsilon_1 = 0.036 \quad \varepsilon_2 = 0.027 \quad \varepsilon_3 = 0.0216 \quad \varepsilon_4 = 0.018$$

### 5.1.6 利润最优化下利率模型建立及求解

在上文确定贷款比例的模型中，我们将贷款额度分为四个等级，每个贷款企业均具有相应的贷款额度比例区间。由题目提供的附件 3 可知对于不同信誉等级的公司其贷款时的利率与客户流失率之间的关系也不一样。如下图所示：

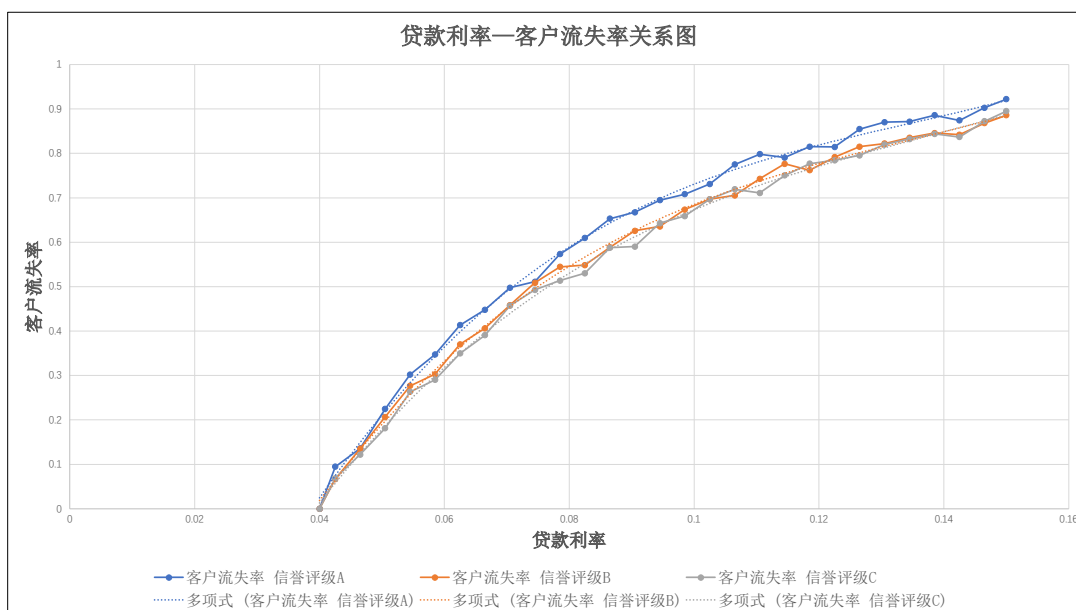


图 3 贷款利率——客户流失率关系图

由上图可以看到贷款利率与客户流失率之间有着确定的关系，设客户流失率与贷款利率的关系为：

$$\gamma = mr^3 + nr^2 + pr + q$$

通过拟合得到不同信誉等级时上式中拟合方程的参数如下：

表 7 不同信誉等级的参数

信誉等级	$m$	$n$	$p$	$q$
A	640.94	-258.57	37.97	-1.1215
B	552.83	-225.05	33.99	-1.016
C	504.72	-207.39	32.157	-0.9735

下文以信誉等级为 A 的企业为例，在保证银行利润最大化的前提下，求解 A 信誉公司的利率。设 A 信誉的某公司贷款额度比例区间为  $[a_1, a_2]$

我们以贷款额度比例区间的中间值  $a = \frac{a_1 + a_2}{2}$  来表示该公司贷款额度所占比例。

则银行贷款给该公司的年利润为：

$$P = M \times a \times r_A \times (1 - \gamma_A)$$

$M$  为该年度银行贷款总额； $a$  为公司贷款额度比例区间的中间值； $r_A$  为 A 信誉的公司贷款利率； $\gamma_A$  为 A 信誉公司  $r_A$  利率下的客户流失率。



目标函数：

$$\max P = M \times a \times r_A \times (1 - \gamma_A)$$

约束条件：

$$s.t. \begin{cases} \gamma = m_A r_A^3 + n_A r_A^2 + p_A r_A + q_A \\ 0.04 \leq r_A \leq 0.15 \\ 0 \leq \gamma_A \leq 1 \end{cases}$$

求解的  $r_A = 0.0469$ ，即信誉评级为 A 的所有公司的利率应为 0.469

同理可求得  $r_B = 0.0517$ ， $r_C = 0.0538$

基于以上三个模型，我们得到了银行针对附录 1 中 123 家的贷款策略（包含每家公司的贷款风险，贷款所占比例，贷款利率），最终结果如下表：

表 8 银行对附件 1 中应放贷企业的信贷策略

企业代号	风险评分	贷款比例	利率	企业代号	风险评分	贷款比例	利率
E1	0.917002	10 万元	0.0469	E49	1.140178	2.16%	0.0538
E2	0.945176	2.70%	0.0469	E50	0.992704	1.80%	0.0538
E6	0.922287	1.80%	0.0469	E51	1.229111	3.60%	0.0517
E7	0.950475	2.70%	0.0469	E54	1.268272	3.60%	0.0469
E8	0.857435	1.80%	0.0469	E58	0.869915	1.80%	0.0517
E9	0.98814	1.80%	0.0469	E59	1.448021	3.60%	0.0469
E12	0.847939	1.80%	0.0517	E61	1.320008	3.60%	0.0517
E13	0.950484	1.80%	0.0469	E62	1.340398	3.60%	0.0517
E15	0.955375	1.80%	0.0469	E63	1.083247	2.16%	0.0517
E16	0.95472	1.80%	0.0469	E64	1.038323	1.80%	0.0469
E17	1.165175	2.16%	0.0469	E65	1.325993	3.60%	0.0517
E18	0.908652	1.80%	0.0469	E66	1.014201	1.80%	0.0517
E19	1.128735	2.16%	0.0469	E67	1.139681	2.16%	0.0517
E22	0.862755	1.80%	0.0469	E74	0.898691	1.80%	0.0517
E24	0.970588	1.80%	0.0469	E81	0.949122	1.80%	0.0469

表 9 银行对附件 1 中应放贷企业的信贷策略（续）

企业代号	风险评分	贷款比例	利率	企业代号	风险评分	贷款比例	利率
E25	1.085956	2.16%	0.0538	E83	0.825524	1.80%	0.0517
E29	1.08192	2.16%	0.0538	E84	0.984259	1.80%	0.0469
E30	1.305397	3.60%	0.0517	E85	0.818764	1.80%	0.0517
E31	1.114574	2.16%	0.0469	E91	0.842618	1.80%	0.0469
E32	0.88082	1.80%	0.0517	E93	0.871196	1.80%	0.0517
E38	0.866309	1.80%	0.0517	E95	0.922699	1.80%	0.0517
E42	1.016824	1.80%	0.0469	E97	0.886116	1.80%	0.0517
E48	1.411143	3.60%	0.0469	E98	0.858974	1.80%	0.0517

## 5.2 问题二模型的建立与求解

问题二中有 302 家无贷款记录的企业，这也就意味着他们没有相应的信誉评级。倘若有信用评级，问题二的求解则与问题一的求解十分相似，问题一中求解风险评分、贷款额度比例、贷款利率的模型对问题二同样适用，由此可得出银行的放贷策略。

然而，附件 2 中 302 家企业无贷款记录，没有相应真实的信誉评级，那么如何预测这些无贷款记录企业的信誉评级呢？这是要着重解决的问题。下文我们将利用神经网络算法，通过训练问题一附件 A 的相关相关指标进行学习，进而预测出 302 家无贷款记录企业的虚拟信誉评级。

### 5.2.1 基于遗传算法的 BP 神经网络模型求解虚拟信用评级 [1]

基于遗传算法优化的 BP 神经网络是在 BP 神经网络的基础上运用遗传算法进行优化。遗传算法可以大大提高 BP 神经网络优化效率以及准确度。此外单纯的 BP 神经网络容易出现局部最优值问题，而经过遗传算法的优化能够很好解决局部最优值问题。

#### 神经网络（Neural Networks, NN）

神经网络是由大量的、简单的处理单元（称为神经元）广泛地互相连接而形成的复杂网络系统，它反映了人脑功能的许多基本特征，是一个高度复杂的非线性动力学习系统。

假设输入层有  $n$  个神经元，隐含层有  $p$  个神经元，输出层有  $q$  个神经元。首先，计算各层神经元的输入和输出。

**Step1.** 首先，计算各层神经元的输入和输出

$$\begin{aligned} hi_h(k) &= \sum_{i=0}^n w_{hi}x_i(k) \quad h = 1, 2, \dots, p \\ ho_h(k) &= f(hi_h(k)) \quad h = 1, 2, \dots, p \\ yi_o(k) &= \sum_{h=0}^p w_{oh}ho_h(k) \quad o = 1, 2, \dots, q \\ yo_o(k) &= f(yi_o(k)) \quad o = 1, 2, \dots, q \end{aligned}$$

**Step2** 利用网络期望输出和实际输出，计算误差函数对输出层的各神经元的偏导数

$$\frac{\partial e}{\partial w_{oh}} = \frac{\partial e}{\partial yi_o} \frac{\partial yi_o}{\partial w_{oh}} = -\delta_o(k)ho_h(k)$$

**Step3** 利用隐含层到输出层的连接权值、输出层的和隐含层的输出计算误差函数对隐含层各神经元的偏导数

$$\begin{aligned} \frac{\partial e}{\partial w_{hi}} &= \frac{\partial e}{\partial hi_h(k)} \frac{\partial hi_h(k)}{\partial w_{hi}} \\ \frac{\partial hi_h(k)}{\partial w_{hi}} &= \frac{\partial (\sum_{i=0}^n w_{hi}x_i(k))}{\partial w_{hi}} = x_i(k) \end{aligned}$$

简化得：

$$\frac{\partial e}{\partial hi_h(k)} = -\delta_h(k)$$

**Step4.** 利用输出层各神经元的和隐含层各神经元的输出来修正连接权值  $w_{oh}(k)$

$$\begin{aligned} \Delta w_{oh}(k) &= -\mu \frac{\partial e}{\partial w_{oh}} = \mu \delta_o(k)ho_h(k) \\ w_{oh}^{N+1} &= w_{oh}^N + \mu \delta_o(k)ho_h(k) \end{aligned}$$

$\mu$  是设置的学习率

**Step5** 利用隐含层各神经元的和输入层各神经元的输入修正连接权

$$\begin{aligned} \Delta w_{hi}(k) &= -\mu \frac{\partial e}{\partial w_{hi}} = \delta_h(k)x_i(k) \\ w_{hi}^{N+1} &= w_{hi}^N + \mu \delta_h(k)x_i(k) \end{aligned}$$

同时用 Levenberg-Marquardt 算法对神经网络算法进行修正。

**Step6.** 最后用误差函数  $e$  计算全局误差

$$e = \frac{1}{2} \sum_{o=1}^q (d_o(k) - yo_o(k))^2$$

## 遗传算法（GA）

遗传算法模拟自然界遗传机制和生物进化论而成的一种并行随机搜索最优化方法。（具有“生存 + 检测”的迭代过程的搜索算法）基于自然界“优胜劣汰，适者生存”的生物进化原理引入优化参数形成的编码串联群体中，按照所选择的适应度函数并

通过遗传中的选择、交叉和变异对个体进行筛选，使适应度值好的个体被保留，适应度差的个体被淘汰，新的群体既继承了上一代的信息，又优于上一代。反复循环，直至满足条件。

- 种群初始化

个体编码方法为实数编码，每隔个体均为一个实数串，由输入层和隐含层连接权值、隐含层阈值、隐含层与输出层连接权值以及输出层阈值 4 个部分组成。个体包含了神经网络全部的权值和阈值，在网路结构一直的情况下，就可以构成一个结构、权值、阈值确定的神经网络。

- 适应度函数

根据个体得到 BP 神经网络的初始权值和阈值，用训练数据训练 BP 神经网络后预测系统输出，把预测输出个期望的输出之间的误差绝对值和 E 作为个体适应度值 F 计算公式为：

$$F = k \left( \sum_{i=1}^n asb(y_i - o_i) \right)$$

式中， $n$  为网络输出节点数； $y_i$  为 BP 神经网络第  $i$  个节点的期望输出， $o_i$  为第  $i$  个节点的实际输出； $k$  为系数。

- 选择操作

遗传算法选择操作有轮盘赌法、锦标赛法等多种方法。选择轮盘赌法时，即基于适应度比例的选择策略，每个个体  $i$  的选择概率  $p_i$  为：

$$f_i = \frac{k}{F_i} p_i = \frac{J_i}{\sum_{j=1}^N f_j}$$

式中， $F_i$  为个体  $i$  的适应度值，由于适应度值越小越好，所以在个体选择前对适应度求倒数； $k$  为系数； $N$  为种群个体数目。

- 交叉操作

由于个体采用实数编码，所以交叉操作方法采用实数交叉法，第  $k$  个染色体  $a_k$  和第  $l$  个染色体  $a_l$  在  $j$  维的交叉操作方法如下：

$$akj = akj(1 - b) + aljbalj = alj(1 - b) + akjb$$

式中， $b$  是  $[0, 1]$  之间的随机数。

- 变异操作选取第  $i$  个个体的第  $j$  个基因  $a_{ij}$  进行变异，变异操作方法如下：

$$a_{ij} = \begin{cases} a_{ij} + (a_{ij} - a_{max})^* f(g) & r > 0.5 \\ a_{ij} + (a_{min} - a_{ij})^* f(g) & r \leq 0.5 \end{cases}$$

式中， $a_{max}$  为基因  $a_{ij}$  的上界； $a_{min}$  为基因的下界；是一个随机数； $g$  为当前迭代次数； $G_{max}$  是最大的进化次数； $r$  为  $[0, 1]$  间的随机数。

预测信用评级的求解

神经网络的实现需要一定数据支持其完成训练，附件 1 中的企业信用评级较为完备，具备一定的代表性。因此，我们选择附件 1 中相关指标的数据作为神经网络进行学习的数据。

表 10 神经网络模型训练指标

训练指标	进项单位代号计数	进项作废发票数	进项作废发票数量比
	销项作废发票数	进项有效发票数	进项退款发票数
	销项有效发票数	销项作废发票数量比	销项作废发票金额比
	年平均利润	销项退款发票数	销项退款金额比
	年平均购买额	年平均销售额	拟合优度
	进项作废发票金额比	进项退款金额比	销项单位代号计数

在第一轮学习的过程中，我们筛选出最有价值的 10 个指标作为以后主要学习的因素。以下是十个主要指标

表 11 神经网络主要学习指标

训练时主要指标	进项作废发票数	进项作废发票数量比	进项作废发票金额比
	进项退款发票数	进项退款金额比	销项作废发票数量比
	销项作废发票金额比	销项退款发票数	销项退款金额比
	拟合优度		

基于遗传算法优化的 BP 神经网络在训练时误差（损失函数）及结果准确率随迭代次数的变化如下图：

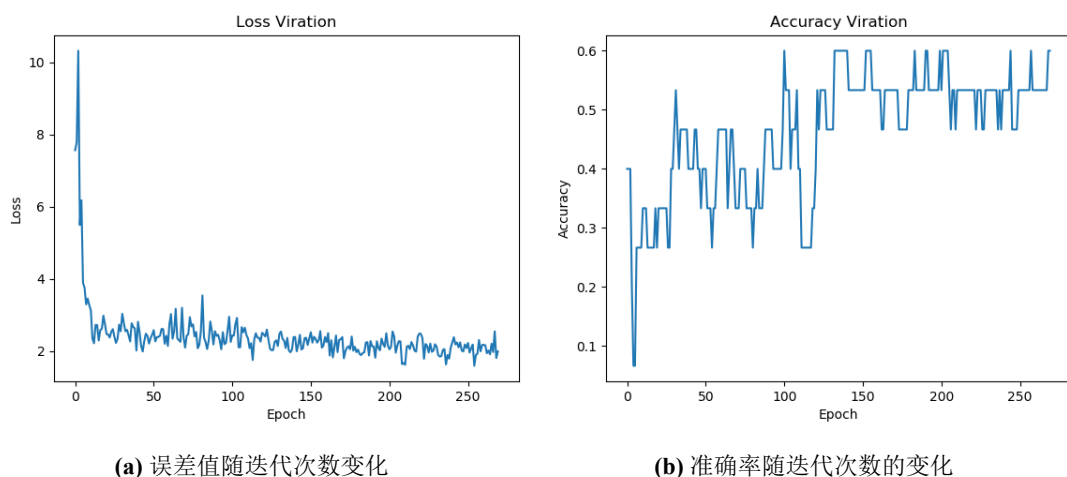


图 4 学习过程中误差和准确率的变化

随着迭代代数的增加，全局误差值逐渐趋近至最初的 2% 左右，结果准确率达到 60%。误差曲线到后面基本偏低且不变，说明权值和阈值已经趋于最优解。

我们将附件 2 各企业指标数据带入到神经网络模型中，最终预测到各企业的虚拟信誉评级，最终结果如下图（因篇幅有限，正文中只放部分预测虚拟信誉评级，全部结果见附录表 16）：

表 12 神经网络模型预测到的附录 2 中企业的虚拟信誉

企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分
E124	A	E144	A	E164	B	E184	C	E204	D	E224	A
E125	B	E145	D	E165	A	E185	B	E205	A	E225	A
E126	A	E146	A	E166	B	E186	C	E206	C	E226	D
E127	C	E147	A	E167	B	E187	C	E207	A	E227	A
E128	C	E148	C	E168	B	E188	A	E208	A	E228	A
E129	B	E149	A	E169	A	E189	A	E209	A	E229	B
E130	D	E150	B	E170	D	E190	A	E210	C	E230	B
...	...	...	...	...	...	...	...	...	...	...	...

5.2.2 附件 2 中企业放贷策略的确定

当附件 2 中的企业具备了信誉评级，那问题二的求解与问题一的求解一般无二，以下是运用问题一中模型对具备信誉评级的问题二求解的过程。

A. 附件 2 中各企业的风险评分

对于附件 2 中的 302 家企业，其平均风险评分为  $f_{risk}^- = \frac{100}{n} = \frac{100}{302} = 0.03125$  风险评分达不到  $f_{risk}^-$  的企业均会被剔除，银行不放贷给该公司。此外，虚拟信誉等级为 D 的企业同样被剔除。得到剔除不符合条件企业后的风险评分表如下（因篇幅有限，正文中只放部分企业的风险评分，完整的风险评分请见附录表 18）：

表 13 附件 2 被放贷的企业的风险评分

企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分
E134	0.41	E180	0.42	E229	0.38	E284	0.36	E330	0.34	E371	0.39
E135	0.45	E181	0.35	E230	0.4	E289	0.45	E333	0.42	E372	0.42
E136	0.35	E182	0.37	E233	0.38	E290	0.41	E334	0.4	E374	0.4
E137	0.42	E183	0.42	E234	0.37	E291	0.35	E335	0.35	E376	0.4
E140	0.42	E184	0.35	E243	0.37	E292	0.36	E336	0.38	E377	0.36
E141	0.37	E185	0.42	E245	0.36	E293	0.36	E339	0.35	E380	0.37
E142	0.4	E186	0.36	E246	0.45	E295	0.34	E340	0.39	E382	0.42
E143	0.34	E188	0.39	E249	0.33	E296	0.34	E341	0.34	E383	0.36
E144	0.44	E189	0.4	E250	0.43	E297	0.38	E342	0.4	E385	0.38
E146	0.38	E190	0.46	E252	0.42	E298	0.37	E344	0.39	E387	0.42
E147	0.39	E192	0.35	E253	0.38	E300	0.33	E345	0.37	E389	0.38
E149	0.43	E195	0.35	E254	0.42	E301	0.46	E347	0.41	E390	0.4
E150	0.4	E196	0.39	E256	0.37	E302	0.39	E348	0.33	E391	0.42
E151	0.4	E197	0.43	E257	0.45	E303	0.41	E349	0.41	E395	0.4
...	...	...	...	...	...	...	...	...	...	...	...

## B. 附件 2 中各企业放贷的款数

我们根据问题一中的聚类模型算出每类企业应分到贷款占总款数的百分比，因此，  
企业贷款款数 = 款数所占百分比 \* 1 亿元。

公司类别	公司代号	贷款比例/%	贷款金额/元
第一类	E144, E149, E185, E197	0.871	871000
第二类	E151, E168, E134	0.848	848000
第三类	E160, E215	0.824	824000
第四类	E135, E190	0.801	801000
第五类	E140, E161	0.771	771000
第六类	E186, E199	0.754	754000
第七类	E166, E176	0.73	730000
第八类	E143	0.707	707000
第九类	E163, E210, E234	0.683	683000
第十类	E141, E152	0.659	659000
第十一类	E312	0.636	636000
第十二类	E154, E324	0.612	612000
第十三类	E310, E301, E257, E289, E225, E246, E214	0.589	589000
第十四类	E171, E212	0.565	565000
第十五类	E223, E418, E315, E321, E146, E298, E345, E380, E182, E314, E317, E415, E293, E422, E357, E399, E178, E355, E383, E278, E328, E377, E227, E284, E282, E313, E282, E203, E220, E256, E221, E243, E245, E289, E195, E208, E274, E192, E184	0.542	542000
第十六类	E288, E328, E330, E323, E361, E348, E365, E417, E418, E300, E351, E369, E410, E363, E368, E295, E341, E307, E296, E136, E339, E406, E414, E335, E360, E181, E291, E249	0.518	518000
第十七类	E142, E230, E196, E305, E233, E147, E189, E329, E150, E172, E216	0.495	495000
第十八类	E366, E391, E359, E372, E382, E387, E349, E400, E180, E311, E290, E303, E260, E279, E201, E261, E333, E174, E259, E252, E250, E266, E165, E228, E179, E209, E137, E254, E183, E169, E258	0.471	471000

## C. 附件 2 中各企业放贷的利率

在问题一中的我们根据建立了以信誉评级为基础的最优化利率模型，可以注意到，利率的多少只与信誉评级有关，因此，

信誉评级为 A 的所有公司的利率应为  $r_A = 0.0469$ ;

信誉评级为 B 的所有公司的利率应为  $r_B = 0.0517$ ;

信誉评级为 C 的所有公司的利率应为  $r_C = 0.0538$

## 5.3 问题三模型的建立及求解

现实中企业的生产经营和经济效益可能会收到一些突发因素影响 [7]，同时突发因素对不同行业、不同类别的企业往往会有不同的影响。因此，如何评估突发因素对行业经营的影响是问题三的核心。我们希望能够建立模型将突发因素对行业的经营进行量化，根据量化结果对问题一中的一级指标“行业风险评分”进行修正，进而求解银行新的放贷策略。



5.3.1 突发因素对行业影响的量化

突发因素（如新冠肺炎）一般都会使劳动力短缺。在现实中，越依赖劳动力的行业可认为受突发因素影响越大。一般来说，国有企业劳动力较为稳定，受突发因素冲击的影响较小。我们借鉴林毅夫等学者利用国有企业就业人员占行业就业人员的比重作为衡量行业对劳动力的依赖程度的工具变量的方法，去可以估算不同行业受疫情影响的程度，得到行业疫情影响指数，影响指数越高，越容易受到冲击。

行业突发因素影响指数 = (行业所有单位就业人员数 - 行业国有单位就业人员数) / 行业所有单位就业人员数

5.3.2 新冠肺炎影响下银行对附件 2 中 302 家企业的信贷策略

A. 行业新冠肺炎影响指数的获取

通过查阅疫情期间各类行业的劳动力数据，并借鉴相关论文 [5]，《疫情对重点行业、重点地区的影响与建议》一文中提到了用该方法得到的新冠肺炎对各行业的影响指数，如下表所示：

表 14 行业受新冠肺炎影响指数

行业类型	电力、燃气及水的生产和供应	电子信息业	金融业
受新冠肺炎影响指数	57.6	86.6	88.7
行业类型	居民服务和其他服务	房地产业	销售业
受新冠肺炎影响指数	61.3	96	96.3
行业类型	水利、环境和公共设施管理	批发和零售	交通运输业
受新冠肺炎影响指数	30.4	96.03	76.3
行业类型	文化、体育、娱乐业	个体经营	物流业
受新冠肺炎影响指数	22.1	98.5	76.3
行业类型	科学研究、技术服务和地质勘查	建筑业	住宿和餐饮
受新冠肺炎影响指数	25.7	93	87
行业类型	租赁和商业服务	制造业	
受新冠肺炎影响指数	56.9	99.5	

## B. 对一级指标“行业风险评分”的修正

在问题一的模型中，我们采用行业企业平均倒闭率来表征行业风险评分，倒闭率越高，则行业风险评分越高。在本问中，由于突发因素对行业的冲击，导致了仅用企业平均倒闭率来衡量“行业风险评分”不再合理，必须将影响指数考虑进去。为此，我们设立如下修正方案：

行业风险评分 = 行业平均倒闭率 × 该行业受新冠肺炎影响指数

修正后的结果如下（由于篇幅有限，仅展示电子信息业企业的修正数据）：

表 15 电子信息业行业风险评分的修正

企业代号	企业类型	平均倒闭概率	修正后行业风险评分
E143	电子信息业	4.5	389.7
E162	电子信息业	4.5	389.7
E222	电子信息业	4.5	389.7
E263	电子信息业	4.5	389.7
E275	电子信息业	4.5	389.7
E307	电子信息业	4.5	389.7
E317	电子信息业	4.5	389.7
E328	电子信息业	4.5	389.7
E345	电子信息业	4.5	389.7
E365	电子信息业	4.5	389.7
E368	电子信息业	4.5	389.7
E369	电子信息业	4.5	389.7
E371	电子信息业	4.5	389.7
E381	电子信息业	4.5	389.7
E392	电子信息业	4.5	389.7
E401	电子信息业	4.5	389.7
E410	电子信息业	4.5	389.7

### C. 在新冠疫情影响下，银行对附件 2 中企业放贷的款数和利率

#### 放贷款数的求解

修正行业风险评分后，将附件 2 中的企业数据再次嵌套问题一种的求解模型得到企业的分类及每类企业放贷的款数为：

类别	公司代号	贷款比例/%	贷款金额
第一类	E406、E415、E418、E185、E267	0.909	909091
第二类	E279、E304、E394、E395、E361、E370、E372、E382、E389、E398、E405	0.861	861244
第三类	E331、E333、E335、E349、E353、E354	0.813	813397
第四类	E233、E254、E257、E260、E265、E269、E281、E351、E377、E396、E177、E180、E189、E247	0.766	765550
第五类	E313、E340、E387、E391、E397、E296、E297、E298、E301、E305、E310、E311、E314、E329	0.718	717703
第六类	E201、E258、E234、E278、E336	0.670	669856
第七类	E343、E299、E134、E171、E190、E212、E219、E183、E192、E203、E206、E210、E214、E215、E216、E218、E250、E251、E252、E256、E261、E166、E302、E312、E315、E325、E400、E402、E344、E347、E362、E364、E379、E380	0.622	622010
第八类	E383、E417、E136、E145、E387、E422、E135、E137、E140、E355、E357、E358、E359、E374、E375	0.574	574163
第九类	E157、E174、E188、E195、E197、E224、E326、E334、E339、E341、E342、E150、E414、E176、E182、E196、E204、E213、E221、E223、E227、E243、E248、E271、E321、E416、E419、E148、E173、E178、E229、E144、E225、E246、E165、E167、E194、E220、E228、E259、E282、E303、E356、E360、E363、E376、E385、E390、E399、E407、E266、E276、E277、E283、E284、E286、E287、E290、E292、E293、E295、E403、E408、E409、E411、E421、E423、E142、E152、E163、E172、E249、E366、E289	0.526	526316
第十类		0.478	478469

图 5 疫情下，银行对附件 2 中各类企业的放款数量

#### 利率的求解

在问题一中的我们根据建立了以信用评级为基础的最优化利率模型，可以注意到，利率的多少只与信用评级有关，因此，

信用评级为 A 的所有公司的利率应为  $r_A = 0.0469$ ;

信用评级为 B 的所有公司的利率应为  $r_B = 0.0517$ ;

信用评级为 C 的所有公司的利率应为  $r_C = 0.0538$

## 六、模型不足与改进

### 6.1 不足之处

- 使用直线方程的拟合优度来表征公司资金稳定性，但并没有区分企业是在盈利还是用亏损，且不能保证所有的企业资金都在某一条直线上波动。模型中使用层次分析法来确定一级指标的权重，但该方法具有一定的主观性，故结果也有一定的误差。
- 虽然通过使用熵权法，避免了对二级指标的主观赋权，但也受到熵权法法的局限，即我们仅凭数据变化确定权重，得出的权重一定程度上不能完全反映实际状态。
- 在确定利率时，根据已知现有数据，贷款年利率和客户流失率之间的关系仅与信誉评级有关,这是明显不符合常理的，利率应由多个指标共同决定，我们考虑不够全面。
- 由于时间有限，没有尝试更多的机器学习算法；该模块先后尝试了岭回归分类器，xgboost 分类模型，多层非线性神经网络模型等，但仍可以比较更多的算法。此外，模型还可以进行更大程度、更综合性的集成；本次先使用了递归特征消除算法筛选了一部分指标传入神经网络进行训练，是初步的模型集成，还可以扩大考虑范围，提升综合度。

### 6.2 对应改进方法

- 资本负债比率 [6] 是衡量企业财务稳定性的重要指标。然而但根据现有数据，我们无法得到每个企业的资产负债比例，该方法虽然更加契合实际，但在本文中没有应用。
- 为了避免主观赋权和熵权法所带来的误差，可以将该模型交给该银行的决策人员，由实际决策人员来决定一级指标和二级指标的权重，这样得到的权重可以避免与实际脱轨。
- 可以通过查阅相关文献，引入更多能够影响贷款年利率的指标，并综合考虑它们对贷款年利率的影响，如企业的类别和企业规模等。
- 在相同数据集上尝试更多的，例如可以尝试随机森林分类算法，支持向量机分类算法等更多模型在数据集上实验并进行比较；进行更大规模，更精密的模型集成；使用半监督学习，可以使用无标签数据进行训练来缓解训练样本量少的缺陷。

## 七、模型推广与运用

本模型可以根据企业很多的内外部因素，综合评估了各类型中小微企业的信誉水平与企业实力，所以本模型最直接的推广就是现实中银行进行无抵押贷款的场景；同时，本模型也可以用来反向评估银行的现有信贷策略，或者应用于对银行年收益的预测。企业的信誉水平与实力一个较长时间内经营状况的与盈利能力反映，所以该模型有机会应用于对股市的分析等类似场景；甚至模型评估结果也在一定程度上蕴含着行业的前景。

## 参考文献

- [1] 孙小那. 基于 BP 神经网络对信贷客户逾期的分析与预测 [D]. 大连理工大学,2018.
- [2] Dokkyun Yi, Jaehyun Ahn, Sangmin Ji. An Effective Optimization Method for Machine Learning Based on ADAM. 2020, 10(3)
- [3] 苏敏艳, 王紫红, 郑慧凌, 张娜娜, 高山. 基于熵权-TOPSIS 法的江苏省卫生资源配置地区差异研究 [J]. 卫生软科学,2020,34(09):62-65.
- [4] 范宏, 盛婉琴, 王直杰. 基于数据的银行信贷行业的信用风险研究 [J]. 经济数学,2020,37(2):1-8.
- [5] 王汉儒. 疫情对重点行业、重点地区的影响与建议 [J]. 海外投资与出口信贷,2020(03):20-22.
- [6] 中国) 卢颖, 高山, 高凯丽, 周莎. 财务管理 [M]. 北京理工大学出版社, 2019.08.
- [7] 许艺君. 疫情对中小微企业的影响及应对策略 [J]. 经济管理文摘,2020(16):35-37.

## TOPSIS 模型的 python 代码

```
import pandas as pd
import sys
import numpy as np
import pandas as pd
import math as math

def topsis(data,weights,b):

    dataset=pd.DataFrame(data)
    row_count=dataset.shape[0]
    column_count=dataset.shape[1]

    #Get the root of sum of square values
    for i in range(0,column_count) :
        sum=0
        for j in range(0,row_count):
            sum=sum + (dataset.iloc[j,i])*(dataset.iloc[j,i])
        sum=math.sqrt(sum)

    #Get the column wise values normalized
    for j in range(0,row_count):
        k=float(weights[i])/(sum)
        dataset.iloc[j,i]=float(dataset.iloc[j,i]*k)

    # Get the V+ AND V- values
    Vp=[]
    Vn=[]
    for i in range(0,column_count):
        if(b[i]=='p'):
            Vp.append(max(dataset.iloc[:,i]))
            Vn.append(min(dataset.iloc[:,i]))
        if(b[i]=='n'):
            Vn.append(max(dataset.iloc[:,i]))
            Vp.append(min(dataset.iloc[:,i]))

    # Get S+ & S- Values
    Sp=[]
    Sn=[]
    for i in range(0,row_count):
        p=0
        n=0
        for j in range(0,column_count):
            p=p+(dataset.iloc[i,j]-Vp[j])*(dataset.iloc[i,j]-Vp[j])
            n=n+(dataset.iloc[i,j]-Vn[j])*(dataset.iloc[i,j]-Vn[j])
        p=math.sqrt(p)
```

```

n=math.sqrt(n)
Sp.append(p)
Sn.append(n)

Sp=np.array(Sp)
Sn=np.array(Sn)

p=[]
for i in range(0,row_count):
    t=(Sn[i])/(Sn[i]+Sp[i])
    p.append(t)

p_temp=p.copy()
rank=[]
for i in range(0,row_count):
    rank.append(0)
m=1
for i in range(0,row_count):
    ff=max(p_temp)
    sj=p_temp.index(ff)
    p_temp[sj]=-1
    rank[sj]=m
    m=m+1

dataset['topsis score']=p;
dataset['rank']=rank;
print(dataset)

def main():
    print(sys.argv)
    weights=[float(i) for i in sys.argv[2].split(',')]
    data=pd.read_csv(sys.argv[1]).values
    b=sys.argv[3].split(',')
    topsis(data,weights,b)

if __name__=="__main__":
    main()

```

## 风险评估模型中获取二级指标数据的 python 程序

```

import pandas as pd
import numpy as np
import datetime
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

```

```

def gtmonthlist(stdate, enddate):
    month_num = 12 * (int(enddate/100) - int(stdate/100)) + (enddate-int(enddate/100)*100) -
        (stdate-int(stdate/100)*100) + 1
    month_list = []
    year = int(stdate/100)
    month = (stdate-int(stdate/100)*100)
    for m in range(month_num):
        month_list.append(year*100 + month)
        month += 1
        if month == 13:
            month = 1
            year += 1
    return month_list, month_num

dfi = pd.read_excel('data/data2.xlsx',sheet_name=1)
dfo = pd.read_excel('data/data2.xlsx',sheet_name=2)

dfi.drop(index=dfi[dfi['发票状态'].isin(['作废发票'])].index, inplace = True)
dfo.drop(index=dfo[dfo['发票状态'].isin(['作废发票'])].index, inplace = True)

dfigrp = dfi.groupby(['企业代号'])
dfogrp = dfo.groupby(['企业代号'])

rlist = []
mlist = []
plist = []
udlist = []

for u in range(123, 123+302):

    ckey = 'E'+str(u+1)

    cpni = dfigrp.get_group(ckey)
    cpni = cpni.reset_index(drop=True)
    cpno = dfogrp.get_group(ckey)
    cpno = cpno.reset_index(drop=True)

    mcpni = cpni.groupby(lambda x: (cpni['开票日期'].iloc[x]).year*100 +
        (cpni['开票日期'].iloc[x]).month)['金额'].sum()
    mcpno = cpno.groupby(lambda x: (cpno['开票日期'].iloc[x]).year*100 +
        (cpno['开票日期'].iloc[x]).month)['金额'].sum()

    iidx = list(mcpni.index.get_level_values(0).values)
    oidx = list(mcpno.index.get_level_values(0).values)

    mthlisti = gtmonthlist(iidx[0], iidx[-1])[0]

```



```

mthlisto = gtmonthlist(oidx[0], oidx[-1])[0]

# wi = []
for num in mthlisti:
    if not num in oidx:
        # wi.append(num)
        newo = pd.DataFrame([0.0], index=[num])
        mcpno = mcpno.append(newo)
        oidx.append(num)
    if not num in iidxi:
        newi = pd.DataFrame([0.0], index=[num])
        mcpni = mcpni.append(newi)
        iidxi.append(num)
    # mcpni.drop(labels = wi, inplace = True)

for num in mthlisto:
    if not num in iidxi:
        newi = pd.DataFrame([0.0], index=[num])
        mcpni = mcpni.append(newi)
        iidxi.append(num)
    if not num in oidx:
        newo = pd.DataFrame([0.0], index=[num])
        mcpno = mcpno.append(newo)
        oidx.append(num)
    # mcpno.drop(labels = wo, inplace = True)

mcpni.sort_index(ascending=True)
mcpno.sort_index(ascending=True)

mc = pd.concat([mcpni, mcpno], axis = 1)
mc.columns = ['minput', 'moutput']
mc.eval('pft = moutput - minput', inplace=True)

yt = np.array(mc['pft'])
y = []
tmp = 0
for i in range(len(yt)):
    tmp = tmp + yt[i]
    y.append(tmp)

X = []
for i in range(len(y)):
    X.append(i)
X = np.reshape(X, (len(X), 1))

reg = linear_model.LinearRegression()
reg.fit(X, y)

```

```

mse = np.sqrt(mean_squared_error(y, reg.predict(X)))
r2s = r2_score(y, reg.predict(X))

rlist.append(r2s)
mlist.append(mse)
plist.append(mc['pft'].mean()*12)
print(ckey)
print(r2s)
print(mse)

ud = {'企业代号':ckey, 'r2s':r2s, 'RSME':mse, '年平均利润':mc['pft'].mean()*12,
      '年平均购买额':mc['minput'].mean()*12, '年平均销售额':mc['moutput'].mean()*12}
udlist.append(ud)

nprlist = np.array(rlist)
npmlist = np.array(mlist)
npplist = np.array(plist)
savedata = pd.DataFrame(columns=['r2s', 'RMSE', 'MYP'])
savedata['r2s'] = nprlist
savedata['RMSE'] = npmlist
savedata['MYP'] = npplist
savedata.to_excel('r2s2.xls')

sl = pd.DataFrame(udlist)
sl.to_excel('newr2s2.xls')

```

## 计算二级指标资金稳定性的代码

```

import pandas as pd
import numpy as np
import datetime
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

def gtmonthlist(stdate, enddate):
    month_num = 12 * (int(enddate/100) - int(stdate/100)) + (enddate-int(enddate/100)*100) -
        (stdate-int(stdate/100)*100) + 1
    month_list = []
    year = int(stdate/100)
    month = (stdate-int(stdate/100)*100)
    for m in range(month_num):
        month_list.append(year*100 + month)
        month += 1
    if month == 13:
        month = 1

```

```

year += 1
return month_list, month_num

dfi = pd.read_excel('data/data2.xlsx', sheet_name=1)
dfo = pd.read_excel('data/data2.xlsx', sheet_name=2)

dfi.drop(index=dfi[dfi['发票状态'].isin(['作废发票'])].index, inplace = True)
dfo.drop(index=dfo[dfo['发票状态'].isin(['作废发票'])].index, inplace = True)

dfigrp = dfi.groupby(['企业代号'])
dfogrp = dfo.groupby(['企业代号'])

rlist = []
mlist = []
plist = []
udlist = []

for u in range(123, 123+302):

    ckey = 'E'+str(u+1)

    cpni = dfigrp.get_group(ckey)
    cpni = cpni.reset_index(drop=True)
    cpno = dfogrp.get_group(ckey)
    cpno = cpno.reset_index(drop=True)

    mcpni = cpni.groupby(lambda x: (cpni['开票日期'].iloc[x]).year*100 +
        (cpni['开票日期'].iloc[x]).month)['金额'].sum()
    mcpno = cpno.groupby(lambda x: (cpno['开票日期'].iloc[x]).year*100 +
        (cpno['开票日期'].iloc[x]).month)['金额'].sum()

    iidx = list(mcpni.index.get_level_values(0).values)
    oidx = list(mcpno.index.get_level_values(0).values)

    mthlisti = gtmonthlist(iidx[0], iidx[-1])[0]
    mthlisto = gtmonthlist(oidx[0], oidx[-1])[0]

    # wi = []
    for num in mthlisti:
        if not num in oidx:
            # wi.append(num)
            newo = pd.DataFrame([0.0], index=[num])
            mcpno = mcpno.append(newo)
            oidx.append(num)
        if not num in iidx:
            newi = pd.DataFrame([0.0], index=[num])
            mcpni = mcpni.append(newi)

```

```

iidx.append(num)
# mcpni.drop(labels = wi, inplace = True)

for num in mthlisto:
    if not num in iidx:
        newi = pd.DataFrame([0.0], index=[num])
        mcpni = mcpni.append(newi)
        iidx.append(num)
    if not num in oidx:
        newo = pd.DataFrame([0.0], index=[num])
        mcpno = mcpno.append(newo)
        oidx.append(num)
# mcpno.drop(labels = wo, inplace = True)

mcpni.sort_index(ascending=True)
mcpno.sort_index(ascending=True)

mc = pd.concat([mcpni, mcpno], axis = 1)
mc.columns = ['minput', 'moutput']
mc.eval('pft = moutput - minput', inplace=True)

yt = np.array(mc['pft'])
y = []
tmp = 0
for i in range(len(yt)):
    tmp = tmp + yt[i]
y.append(tmp)

X = []
for i in range(len(y)):
    X.append(i)
X = np.reshape(X, (len(X), 1))

reg = linear_model.LinearRegression()
reg.fit(X, y)
mse = np.sqrt(mean_squared_error(y, reg.predict(X)))
r2s = r2_score(y, reg.predict(X))

rlist.append(r2s)
mlist.append(mse)
plist.append(mc['pft'].mean()*12)
print(ckey)
print(r2s)
print(mse)

ud = {'企业代号': ckey, 'r2s': r2s, 'RSME': mse, '年平均利润': mc['pft'].mean()*12,
      '年平均购买额': mc['minput'].mean()*12, '年平均销售额': mc['moutput'].mean()*12}

```

```

udlist.append(ud)

nprlist = np.array(rlist)
npmlist = np.array(mlist)
npplist = np.array(plist)
savedata = pd.DataFrame(columns=['r2s', 'RMSE', 'MYP'])
savedata['r2s'] = nprlist
savedata['RMSE'] = npmlist
savedata['MYP'] = npplist
savedata.to_excel('r2s2.xls')

sl = pd.DataFrame(udlist)
sl.to_excel('newr2s2.xls')

```

## 神经网络模型中首轮学习筛选决定性更高的指标

```

import torch
import torch.nn as nn
import torch.nn.functional as F

class MyClassifier(nn.Module):
    def __init__(self):
        super(MyClassifier, self).__init__()
        self.fc1 = nn.Linear(10, 50)
        self.relu1 = nn.ReLU()
        self.fc2 = nn.Linear(50, 100)
        # self.dropout = nn.Dropout(0.4)
        self.relu2 = nn.ReLU()
        self.fc3 = nn.Linear(100, 4)

    def forward(self, x):
        x = x.view(x.size(0), -1)
        x = self.fc1(x)
        x = self.relu1(x)
        x = self.fc2(x)
        # x = self.dropout(x)
        x = self.relu2(x)
        x = self.fc3(x)
        return x

    def predict(self, x):
        pred = F.softmax(self.forward(x))
        ans = []
        for t in pred:
            if t[0] > t[1]:
                ans.append(0)
            else:

```

```
ans.append(1)
return torch.tensor(ans)
```

## 计算二级指标资金稳定性

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression, RidgeClassifier
from sklearn.feature_selection import RFE

df = pd.read_excel('data/clsdata.xls')
y = df['信誉评级']
df.drop(['企业代号', '信誉评级', 'Unnamed: 0'], axis = 1, inplace = True)
X = df

X_train, X_valid, y_train, y_valid = train_test_split(X, y, train_size=0.8, test_size=0.2,
random_state=0)
lr = RidgeClassifier(alpha=1.5)
rfe = RFE(estimator=lr, n_features_to_select=10, step=1)
rfe.fit(X_train, y_train)
y_pred = rfe.predict(X_valid)
newX = rfe.transform(X)
print(newX)
print(type(newX))
print(rfe.ranking_)
newX = rfe.transform(X)
np.save("newX.npy",newX)

print("Accuracy : %.4g" % accuracy_score(y_valid, y_pred))
print("T-Accuracy : %.4g" % accuracy_score(y_train, rfe.predict(X_train)))
```

表 16 神经网络模型预测到的附录 2 中企业的虚拟信誉

公 司  代 号	虚 拟  信 誉	公 司  代 号	虚 拟  信 誉	公 司  代 号	虚 拟  信 誉	公 司  代 号	虚 拟  信 誉	公 司  代 号	虚 拟  信 誉	公 司  代 号	虚 拟  信 誉	公 司  代 号	虚 拟  信 誉	公 司  代 号	虚 拟  信 誉
E124	A	E144	A	E164	B	E184	C	E204	D	E224	A	E244	A	E264	D
E125	B	E145	D	E165	A	E185	B	E205	A	E225	A	E245	C	E265	A
E126	A	E146	A	E166	B	E186	C	E206	C	E226	D	E246	A	E266	A
E127	C	E147	A	E167	B	E187	C	E207	A	E227	A	E247	D	E267	D
E128	C	E148	C	E168	B	E188	A	E208	A	E228	A	E248	C	E268	C
E129	B	E149	A	E169	A	E189	A	E209	A	E229	B	E249	B	E269	C
E130	D	E150	B	E170	D	E190	A	E210	C	E230	B	E250	A	E270	A
E131	C	E151	B	E171	A	E191	B	E211	A	E231	B	E251	B	E271	A
E132	A	E152	C	E172	B	E192	B	E212	A	E232	D	E252	A	E272	A
E133	B	E153	B	E173	A	E193	A	E213	D	E233	B	E253	B	E273	C
E134	B	E154	B	E174	A	E194	D	E214	A	E234	C	E254	B	E274	C
E135	A	E155	D	E175	C	E195	C	E215	B	E235	C	E255	A	E275	D
E136	C	E156	D	E176	A	E196	A	E216	B	E236	A	E256	C	E276	D
E137	A	E157	D	E177	D	E197	A	E217	C	E237	B	E257	A	E277	B
E138	B	E158	D	E178	A	E198	A	E218	D	E238	A	E258	A	E278	C
E139	D	E159	B	E179	A	E199	C	E219	A	E239	C	E259	A	E279	A
E140	B	E160	A	E180	B	E200	A	E220	C	E240	B	E260	A	E280	C
E141	A	E161	B	E181	B	E201	A	E221	A	E241	C	E261	A	E281	B
E142	B	E162	C	E182	B	E202	C	E222	C	E242	D	E262	A	E282	C
E143	B	E163	C	E183	B	E203	C	E223	A	E243	B	E263	A	E283	B
E284	C	E304	B	E324	A	E344	B	E364	D	E384	C	E404	B	E424	C

表 17 神经网络模型预测到的附录 2 中企业的虚拟信誉（续）

公 司 代 号	虚 拟 信 誉	公 司 代 号	虚 拟 信 誉	公 司 代 号	虚 拟 信 誉	公 司 代 号	虚 拟 信 誉	公 司 代 号	虚 拟 信 誉	公 司 代 号	虚 拟 信 誉	公 司 代 号	虚 拟 信 誉	公 司 代 号	虚 拟 信 誉
E285	A	E305	B	E325	D	E345	B	E365	C	E385	B	E405	B	E425	C
E286	C	E306	B	E326	C	E346	B	E366	B	E386	B	E406	C		
E287	B	E307	B	E327	D	E347	B	E367	C	E387	B	E407	C		
E288	B	E308	B	E328	B	E348	A	E368	B	E388	B	E408	D		
E289	A	E309	C	E329	B	E349	B	E369	C	E389	C	E409	B		
E290	B	E310	A	E330	A	E350	D	E370	B	E390	B	E410	C		
E291	B	E311	B	E331	D	E351	C	E371	A	E391	B	E411	B		
E292	C	E312	A	E332	B	E352	B	E372	B	E392	D	E412	C		
E293	B	E313	C	E333	B	E353	B	E373	B	E393	B	E413	C		
E294	C	E314	C	E334	B	E354	D	E374	B	E394	B	E414	C		
E295	C	E315	C	E335	C	E355	B	E375	D	E395	B	E415	C		
E296	C	E316	A	E336	B	E356	B	E376	B	E396	B	E416	B		
E297	B	E317	A	E337	C	E357	C	E377	C	E397	C	E417	C		
E298	C	E318	C	E338	B	E358	B	E378	B	E398	B	E418	C		
E299	D	E319	D	E339	C	E359	B	E379	D	E399	C	E419	B		
E300	C	E320	B	E340	B	E360	B	E380	B	E400	B	E420	C		
E301	A	E321	B	E341	C	E361	A	E381	D	E401	C	E421	B		
E302	B	E322	A	E342	B	E362	D	E382	B	E402	C	E422	C		
E303	B	E323	B	E343	D	E363	B	E383	B	E403	B	E423	C		



表 18 附件 2 被放贷的企业的风险评分

企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分
E134	0.41	E180	0.42	E229	0.38	E284	0.36	E330	0.34	E371	0.39
E135	0.45	E181	0.35	E230	0.4	E289	0.45	E333	0.42	E372	0.42
E136	0.35	E182	0.37	E233	0.38	E290	0.41	E334	0.4	E374	0.4
E137	0.42	E183	0.42	E234	0.37	E291	0.35	E335	0.35	E376	0.4
E140	0.42	E184	0.35	E243	0.37	E292	0.36	E336	0.38	E377	0.36
E141	0.37	E185	0.42	E245	0.36	E293	0.36	E339	0.35	E380	0.37
E142	0.4	E186	0.36	E246	0.45	E295	0.34	E340	0.39	E382	0.42
E143	0.34	E188	0.39	E249	0.33	E296	0.34	E341	0.34	E383	0.36
E144	0.44	E189	0.4	E250	0.43	E297	0.38	E342	0.4	E385	0.38
E146	0.38	E190	0.46	E252	0.42	E298	0.37	E344	0.39	E387	0.42
E147	0.39	E192	0.35	E253	0.38	E300	0.33	E345	0.37	E389	0.38
E149	0.43	E195	0.35	E254	0.42	E301	0.46	E347	0.41	E390	0.4
E150	0.4	E196	0.39	E256	0.37	E302	0.39	E348	0.33	E391	0.42
E151	0.4	E197	0.43	E257	0.45	E303	0.41	E349	0.41	E395	0.4
E152	0.37	E199	0.34	E258	0.42	E304	0.4	E351	0.34	E398	0.41
E154	0.35	E201	0.43	E259	0.43	E305	0.39	E352	0.41	E399	0.36
E160	0.43	E203	0.37	E260	0.43	E307	0.34	E353	0.4	E400	0.41
E161	0.4	E206	0.35	E261	0.42	E310	0.46	E355	0.36	E403	0.4
E163	0.36	E209	0.42	E263	0.39	E311	0.42	E356	0.41	E405	0.41
E165	0.42	E210	0.36	E265	0.38	E312	0.43	E357	0.36	E406	0.35
E166	0.38	E212	0.46	E266	0.43	E313	0.36	E358	0.4	E409	0.4
E167	0.39	E214	0.45	E268	0.34	E314	0.37	E359	0.42	E410	0.34

表 19 附件 2 被放贷的企业的风险评分（续）

企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分	企业 代号	风险 评分
E168	0.41	E215	0.41	E269	0.36	E315	0.37	E360	0.35	E414	0.35
E169	0.43	E216	0.41	E274	0.35	E317	0.37	E361	0.34	E415	0.37
E171	0.46	E220	0.36	E277	0.4	E321	0.37	E363	0.35	E417	0.34
E172	0.4	E221	0.37	E278	0.36	E323	0.34	E365	0.33	E418	0.34
E174	0.43	E223	0.37	E279	0.43	E324	0.37	E366	0.42	E419	0.36
E176	0.39	E225	0.45	E281	0.39	E326	0.36	E368	0.34	E420	0.38
E178	0.35	E227	0.36	E282	0.36	E328	0.34	E369	0.34	E421	0.38
E179	0.42	E228	0.43	E283	0.39	E329	0.4	E370	0.39	E422	0.36