

# 基于 TOPSIS 的双目标优化风险评估模型

## 摘要

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。本文通过建立数学模型，就实际和附件中的数据信息进行分析，给出对中小微企业最优的信贷策略。

针对问题一，对附件 1 中 123 家企业的信贷风险进行量化分析。首先分析题目给出的信息，找到能够体现企业实力及供求关系稳定性的六个评价指标，即开票频率、销售额环比、净收入率环比、规模、进项发票作废率、销项发票负数率，建立**企业信贷风险综合评估模型**，通过**熵权法**确定每个评价指标的权重，通过**TOPSIS 法**计算出每家企业最终的得分，根据得分排名确定放贷顺序。其次，我们将贷款额度及年利率方案看作一个**双目标优化问题**，即在考虑企业规模及信誉评级的情况下，确定约束条件，设计出一个使客户流失率最小化、银行收益最大化的信贷策略，确定对每个企业的放贷额度及利率。

针对问题二，首先根据问题一中建立的企业信贷风险综合评估模型，给出附件 2 中 302 家企业的最终得分，根据得分确定放贷顺序。另外，分析得到问题一中由信贷风险综合评估模型得出的得分排名与对应信誉评级之间呈强正相关的关系，由此为附件 2 中 302 家企业给出信誉评级，并在问题一的双目标优化模型中引入总贷款额不超过 1 亿元的约束条件，得到一个新的最优信贷策略，确定对每个企业的放贷额度及利率。

针对问题三，在问题二的基础上，考虑一些突发因素，如新冠疫情对不同行业企业的影响。参照《统计上大中小微型企业划分办法(2017)》中将 302 家企业划分成房地产与物业服务、个体经营、工业、建筑业、农林牧渔业、软件和科学信息技术服务业、文化业、销售业、医药业、运输和租赁和商务服务业十种类别，通过查找文献给出新冠疫情对其的影响程度的科学的等级评定。将疫情对各行业负面影响程度作为新的评价指标加入问题一中的信贷风险综合评估模型中，得到每家企业新的最终得分排名，从而确定放款顺序。

最后，给出对每个模型优缺点的评价。

**关键词：**TOPSIS 熵权法 多目标规划 企业信贷风险综合评估模型

## 一、问题重述

### 1.1 问题背景

随着市场经济的发展和金融改革的不断深入,一些大型商业银行为获得更高的经营效益,降低信贷风险,将信贷资金主要投放到了大中型企业,而忽视了对小微企业的信贷支持[1]。近年来,国家和政府致力于支持小微企业发展,解决小微企业信贷难的问题。

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。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。

### 1.2 问题提出

根据以上背景,以及给出的三个附件,需要解决以下问题:

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。

其中,附件 1 给出了 123 家有信贷记录企业的相关数据,包含企业名称、信誉评级、是否违约、进项发票信息、销项发票信息等;附件 2 给出了 302 家无信贷记录企业的相关数据,包含进项发票信息、销项发票信息;附件 3 给出了在不同信誉评级下贷款年利率与客户流失率的关系。

## 二、问题分析

### 2.1 问题一的分析

问题一要求我们根据附件 1 中 123 家有信贷记录企业的信誉评级、违约情况、进销项发票信息等来对企业的信贷风险进行量化分析,并给出对这些企业的信贷策略。首先,实际中银行一般会向实力强、供求关系稳定的企业提供贷款,因此需要找到能够体现企业实力及供求关系的指标。我们将从这两大方面考虑企

业信贷风险的五大评价指标，通过 TOPSIS 法及熵权法建立企业信贷风险综合评估模型，确定每个企业的风险值，以此确定是否放贷以及在年度信贷总额固定时对这些企业的放贷顺序。

其次，根据附件 3 中给出的信息可知，在不同信誉评级下贷款年利率越高，客户流失率越高，从银行的角度考虑，一个较优的信贷策略应该让银行在客户流失率尽可能小的情况下获得尽可能多的收益。同时，不同的企业规模所需贷款额度有所不同，并且银行对信誉评级较好的企业应给予适当的利率优惠。因此，我们可以将信贷策略看作一个双目标优化问题，即在考虑企业规模及信誉评级的情况下，设计出一个使客户流失率最小化、银行收益最大化的信贷策略，确定对每个企业的放贷额度及利率。

### 2.2 问题二的分析

问题二要求我们对附件 2 中 302 家企业的信贷风险进行量化分析，但附件 2 中的 302 家企业并没有信贷记录，因此需要分析第一问中通过企业信贷风险综合评估模型中得出的风险值与信誉评级之间的关系，确定这 302 家企业的信誉评级，并在制定信贷策略时加入银行年度信贷总额 1 亿元的约束条件，从而确定对企业的放贷顺序、放贷额度及利率。

### 2.3 问题三的分析

在实际情况下，企业的生产经营和经济效益可能会受到一些突发因素的影响，而且突发因素往往对不同行业、不同类型的企业会有不同的影响。以新冠病毒疫情为例，其对服务业、旅游业等行业造成了极其负面的影响，但对医药行业、高新技术行业又有一定的推动作用。问题三需要我们查询相关信息，综合考虑附件 2 中各企业的信贷风险和可能发生的突发因素对各企业的影响，给出银行在年度信贷总额为 1 亿元时的信贷调整策略。

## 三、假设与符号

### 3.1 模型假设

- (1) 银行对信誉评级为 D 的企业不予放贷。
- (2) 题目所给数据均真实有效。

### 3.2 符号说明

| 符号    | 含义             |
|-------|----------------|
| $f_i$ | 第 i 家公司的开票频率   |
| $N_i$ | 第 i 家公司的总有效发票数 |
| $t_i$ | 第 i 家公司开票的总天数  |

|                    |                          |
|--------------------|--------------------------|
| $\alpha_{Yk}$      | 第 k 个季度的销售额环比            |
| $Y_k$              | 企业第 k 个季度的销售额            |
| $V_{Yi}$           | 第 i 家公司近 6 个季度的销售额环比平均值  |
| $\beta$            | 第 k 个季度的净收入率             |
| $\alpha_{\beta k}$ | 第 k 个季度的净收入率环比           |
| $V_{\beta i}$      | 第 i 家公司近 6 个季度的净收入率环比平均值 |
| M                  | 银行收益                     |
| $\gamma$           | 客户流失率                    |
| K                  | 银行贷款总额                   |

## 四、问题一的模型建立与求解

### 4.1 问题的分析

问题一要求我们根据附件 1 中 123 家有信贷记录企业的信誉评级、违约情况、进销项发票信息等来对企业的信贷风险进行量化分析，并给出对这些企业的信贷策略。我们从企业实力和供求关系稳定性两大方面考虑企业信贷风险的五大评价指标，通过 TOPSIS 法及熵权法建立企业信贷风险综合评估模型，确定每个企业的风险值，以此确定是否放贷以及在年度信贷总额固定时对这些企业的放贷顺序。

其次，根据附件 3 中给出的信息可知，在不同信誉评级下贷款年利率越高，客户流失率越高，从银行的角度考虑，一个较优的信贷策略应该让银行在客户流失率尽可能小的情况下获得尽可能多的收益。同时，不同的企业规模所需贷款额度有所不同，并且银行对信誉评级较好的企业应给予适当的利率优惠。因此，我们可以将信贷策略看作一个双目标优化问题，即在考虑企业规模及信誉评级的情况下，设计出一个使客户流失率最小化、银行收益最大化的信贷策略，确定对每个企业的放贷额度及利率。

### 4.2 企业信贷风险综合评估模型

#### 4.2.1 合适评价指标的选取

在确定企业信贷风险时，我们主要考虑该企业的实力及供求关系，对于这两类评价指标，我们分别讨论各自对应的影响因素。一是影响公司实力的因素，包括开票频率、销售额环比、净收入率环比、规模；二是影响公司供求关系稳定性的因素，包括进项发票作废率、销项发票负数率。

#### 4.2.2 数据处理

#### 4.2.2.1 开票频率

分析认为，一家企业的有效发票开票频率可以反映这家企业的交易频率，而交易频率可以在一定程度上反映出这家企业的产品质量及综合实力。由此，我们给出开票频率的公式如下，即

开票频率=总有效发票数/总时长

$$f_i = \frac{N_i}{t_i} \quad (1)$$

其中， $f_i$ 表示第  $i$  家公司的开票频率， $N_i$ 表示第  $i$  家公司的总有效发票数， $t_i$ 表示附件中给出的第  $i$  家公司开票的总天数。

#### 4.2.2.2 销售额环比

分析认为，一家企业的销售额环比可以体现企业近期销售情况的变化趋势，进而体现出企业的经营情况、产品质量、技术水平、生产能力，从而反映出该企业的综合实力。从企业近期的销售情况也可以看出企业贷款的用途，若企业销售额一直呈上升趋势，贷款极有可能用于建设工厂、扩充生产线、技术更新等，其违约风险也会较低。因此，我们考虑企业最近 6 个季度的销售额环比，若最新的发票时长不足一个季度，则舍去。因为销售额环比反映的是长期趋势，所以舍去最后不足一季度的数据对整体结果不会产生较大影响。我们用企业最近 6 季度的销售额环比平均值表示企业的近期销售情况，具体公式如下：

$$\alpha_{Yk} = \frac{Y_k - Y_{k-1}}{Y_{k-1}} \quad (2)$$

$$V_{Yi} = \frac{1}{6} \sum_{k=n-5}^n \alpha_{Yik} \quad (3)$$

其中， $\alpha_{Yk}$ 表示第  $k$  个季度的销售额环比， $Y_k$ 表示企业第  $k$  个季度的销售额， $V_{Yi}$ 表示第  $i$  家公司近 6 个季度的销售额环比平均值。

#### 4.2.2.3 净收入率环比

与销售额环比类似，一家企业的净收入率环比可以体现出该企业近期的经营情况，从而影响企业的信贷风险。同样，我们考虑企业最近 6 个季度的净收入率环比，若最新的发票时长不足一个季度，则舍去。因为净收入率环比反映的是长期趋势，所以舍去最后不足一季度的数据对整体结果不会产生较大影响。我们用企业最近 6 季度的净收入率环比平均值表示企业的近期盈利情况，具体公式如

下：

$$\text{净收入率}\beta = \frac{\text{销项金额} - \text{进项价税合计}}{\text{进项价税合计}} \quad (4)$$

$$\alpha_{\beta k} = \frac{\beta_k - \beta_{k-1}}{\beta_{k-1}} \quad (5)$$

$$V_{\beta i} = \frac{1}{6} \sum_{k=n-5}^n \alpha_{\beta ik} \quad (6)$$

其中， $\alpha_{\beta k}$ 表示第 k 个季度的净收入率环比， $\beta_k$ 表示企业第 k 个季度的净收入率， $V_{\beta i}$ 表示第 i 家公司近 6 个季度的净收入率环比平均值。

#### 4.2.2.4 规模

分析认为，企业的规模可以体现出该企业的实力，我们将企业规模量化为企业每年的销售额，记为 $h_i$ 。年销售额越大，说明该企业实力越强。

#### 4.2.2.5 进项作废率、销项负数率

由题目中给出的信息可知，进项发票中的作废发票代表该企业进货时，当对方已为该交易活动开具发票后，该企业因故取消该项交易，使发票作废；销项发票中的负数发票代表该企业销售产品时，企业已入账计税后，购方因故发生退货并退款。可以认为，进项发票作废率和销项发票负数率均可反映该企业的供求关系稳定性，进项作废率和销项负数率越低，该企业供求关系越稳定，对应公式如下：

$$\text{进项作废率} = \frac{\text{进项发票中作废发票总数}}{\text{进项发票总数}} \quad (7)$$

$$\text{销项负数率} = \frac{\text{销项发票中负数发票总数}}{\text{销项发票总数}} \quad (8)$$

$$\omega_i = \text{进项作废率} + \text{销项负数率} \quad (9)$$

其中， $\omega_i$ 表示第 i 家公司进项作废率与销项负数率之和。

#### 4.2.2.6 信誉评级及违约情况

若企业曾发生违约，则在原信誉评级上降一级。

#### 4.2.3 基于 TOPSIS 法的综合评估模型

Step 1: 将评价指标转为极大型指标, 得到正向化矩阵

除进项发票作废率和销项发票负数率是极小型指标外, 其余四个指标均为极大型。对评价指标进行正向化处理, 得到的正向化矩阵如下(假设有  $n$  个要评价的对象,  $m$  个评价指标):

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} \quad (10)$$

Step 2: 正向化矩阵标准化

对矩阵  $X$  标准化后的矩阵记为  $Z$ ,  $Z$  中的每一个元素:

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}} \quad (11)$$

其中,  $i = 1, 2, 3, \dots, 123$ ,  $j = 1, 2, 3, 4, 5$ 。

Step 3: 确定理想解和负理想解, 并计算综合评价值

$$\begin{aligned} Z^+ &= \{Z_1^+, \dots, Z_n^+\} \\ &= \{\max(z_{11}, z_{21}, \dots, z_{n1}), \max(z_{12}, z_{22}, \dots, z_{n2}), \dots, \max(z_{1m}, z_{2m}, \dots, z_{nm})\} \end{aligned}$$

$$\begin{aligned} Z^- &= \{Z_1^-, \dots, Z_n^-\} \\ &= \{\min(z_{11}, z_{21}, \dots, z_{n1}), \min(z_{12}, z_{22}, \dots, z_{n2}), \dots, \min(z_{1m}, z_{2m}, \dots, z_{nm})\} \end{aligned}$$

$$\text{定义第 } i \text{ 个评价对象与最大值的距离 } D_i^+ = \sqrt{\sum_{j=1}^m (Z_j^+ - z_{ij})^2}$$

$$\text{定义第 } i \text{ 个评价对象与最小值的距离 } D_i^- = \sqrt{\sum_{j=1}^m (Z_j^- - z_{ij})^2}$$

$$\text{那么, 我们可以计算得出第 } i \text{ 个评价对象未归一化的得分: } S_i = \frac{D_i^-}{D_i^+ + D_i^-}$$

显然,  $0 \leq S_i \leq 1$ , 且  $S_i$  越大  $D_i^+$  越小, 即越接近最大值。我们可以将得分归一化:  $\tilde{S}_i = S_i / \sum_{i=1}^n S_i$ , 这样的话  $\sum_{i=1}^n \tilde{S}_i = 1$ 。

#### 4.2.4 基于熵权法对 TOPSIS 模型的修正

Step 1: 判断输入的矩阵中是否存在负数, 如果有则要重新标准化到非负区间。

对矩阵  $X$  进行一次标准化得到  $\tilde{Z}$  矩阵, 其标准化公式为:

$$\tilde{z}_{ij} = \frac{x_{ij} - \min\{x_{1j}, x_{2j}, \dots, x_{nj}\}}{\max\{x_{1j}, x_{2j}, \dots, x_{nj}\} - \min\{x_{1j}, x_{2j}, \dots, x_{nj}\}} \quad (12)$$

Step 2: 计算第  $j$  项指标下第  $i$  个样本所占的比重, 并将其看作相对熵计算中用到的概率。假设有  $n$  个要评价的对象,  $m$  个评价指标, 且经过了上一步处理得到的非负矩阵为:

$$\tilde{Z} = \begin{bmatrix} \tilde{z}_{11} & \cdots & \tilde{z}_{1m} \\ \vdots & \ddots & \vdots \\ \tilde{z}_{n1} & \cdots & \tilde{z}_{nm} \end{bmatrix} \quad (13)$$

计算概率矩阵  $P$ , 其中  $P$  中的每一个元素  $p_{ij}$  的计算公式如下:

$$p_{ij} = \frac{\tilde{z}_{ij}}{\sum_{i=1}^n \tilde{z}_{ij}} \quad (14)$$

Step 3: 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熵权。对于第  $j$  个指标而言, 其信息熵的计算公式为:

$$e_j = -\frac{1}{\ln n} \sum_{i=1}^n p_{ij} \ln(p_{ij}) \quad (j = 1, 2, \dots, m) \quad (15)$$

信息效用值  $d_j = 1 - e_j$ , 信息效用值越大, 其对应的信息越多。将信息效用值归一化, 我们就能够得到每个指标的熵权:

$$W_j = \frac{d_j}{\sum_{j=1}^m d_j} \quad (j = 1, 2, \dots, m) \quad (16)$$

#### 4.2.4 模型求解

基于熵权法, 我们可以计算出概率矩阵  $P$ , 从而计算出每个指标的信息熵, 进而得到各个指标对应的熵权, 如图表 1 所示:

| 指标 | 进项作废率  | 开票频率   | 销售额环比  | 净收入率环比 | 销项负数率 | 规模     |
|----|--------|--------|--------|--------|-------|--------|
| 熵权 | 0.0086 | 0.1988 | 0.0498 | 0.4245 | 0.004 | 0.3143 |

图表 1 各个指标对应的熵权

计算出各个评价指标对应的熵权后, 我们可以计算出 TOPSIS 模型中各家企



业的最终得分,由于数据过大,这里只截选其中得分排名前五和倒数前五的企业,如图表 2 所示。得分越高的企业违约风险越低,故优先放款。

| 企业代号 | 最终得分   |
|------|--------|
| E96  | 0.0908 |
| E1   | 0.084  |
| E3   | 0.0507 |
| E15  | 0.0371 |
| E2   | 0.0365 |
| ...  | ...    |
| ...  | ...    |
| ...  | ...    |
| E86  | 0.0028 |
| E120 | 0.0028 |
| E112 | 0.0027 |
| E118 | 0.0026 |
| E111 | 0.0018 |

图表 2 各家企业的最终得分

### 4.3 双目标优化下的信贷策略

#### 4.3.1 决策变量的确定

根据题目要求,问题一中银行年度信贷总额固定,确定贷款额和年利率为模型的决策变量,通过调节贷款额度和年利率的大小来对目标进行优化。

##### 4.3.1.1 贷款额度

在确定对企业的放贷额度时,需要综合考虑企业的规模和信誉评级。首先可以按照企业年销售额将企业划分为中型、小型、微型,参考国家财政部《关于印发中小企业划型标准规定的通知》中的划分标准,在本文中规定年销售额低于 10 万元的企业为微型企业,年销售额 10 万元-1000 万元的为小型企业,年销售额高于 1000 万元的为中型企业。对于企业规模较大的企业,可以适当增加贷款额度。综合考虑企业的信誉评级,本文中将贷款额度分为 a、b、c、d 四类,为简化模型,设 a 类贷款额度为 100 万元, d 类贷款额度为 10 万元,信誉评级为 D 的企业不予放贷,所有的微型企业贷款额度均为 d 类,通过改变 b、c 来对目标进行优化。

##### 4.3.1.2 年利率

根据题目可知,在确定对企业贷款的年利率时,主要考虑企业的信誉评级。因此,设信誉评级为 A、B、C 的企业对应的贷款年利率为  $i_A$ 、 $i_B$ 、 $i_C$ 。由附件 3 给出的信息可知,信誉评级不变时,贷款年利率越高客户流失率越大。银行在做中小微企业信贷业务的实际操作中,必须综合考虑利率和贷款风险的平衡,拓展维护信用高、业务风险低的客户,拒绝信用不佳、风险较高的申请者或给予其部分贷款,尽可能降低信贷业务风险,保障经济利润[2]。由此,本文设信誉评级为 A 的企业贷款年利率为 0.04,通过改变  $i_B$ 、 $i_C$  来对目标进行优化。

#### 4.3.2 目标函数分析

由题目背景可知，要对信贷策略进行优化就要使得银行收益尽可能大，客户流失率尽可能小。据此本文一共确立如下 2 个优化目标。

(1) 银行收益尽可能大：

$$\max M = M_A + M_B + M_C \quad (17)$$

$$M_A = (A_1 \cdot a + A_2 \cdot b + A_3 \cdot d) \cdot i_A \quad (18)$$

$$M_B = (B_1 \cdot b + B_2 \cdot c + B_3 \cdot d) \cdot i_B \cdot (1 - \gamma_B)(1 - k_B) - k_B \cdot (B_1 \cdot b + B_2 \cdot c + B_3 \cdot d) \quad (19)$$

$$M_C = (C_1 \cdot c + C_2 \cdot d + C_3 \cdot d) \cdot i_C \cdot (1 - \gamma_C)(1 - k_C) - k_C \cdot (C_1 \cdot c + C_2 \cdot d + C_3 \cdot d) \quad (20)$$

其中  $\gamma_B$   $\gamma_C$  分别表示利率为  $i_B$   $i_C$  时客户的流失率， $k_B$   $k_C$  分别表示信誉评级为 B C 的客户的过往违约率（由附件 1 得到），D 表示微型企业的数量。

(2) 客户流失率尽可能小：

$$\min \gamma = \gamma_B + \gamma_C \quad (21)$$

#### 4.3.3 约束条件分析

(1) 贷款额度约束

规定 a 类贷款额度为 100 万元，即  $a=100$ ；b 类贷款额度在 50 万元-100 万元之间，即  $50 \leq b < 100$ ；c 类贷款额度在 10 万元-50 万元之间，即  $10 < c \leq 50$ ；d 类贷款额度为 10 万元，即  $d=10$ 。

(2) 各类信誉评级企业贷款收益约束

在银行向企业贷款时，综合考虑各级信誉评级客户过往违约率、客户流失率，确保银行向各级信誉评级客户贷款所得收益为正，即

$$M_B \geq 0 \quad (22)$$

$$M_C \geq 0 \quad (23)$$

#### 4.3.4 建立多目标优化模型

基于以上分析，建立模型如下：

$$\max M = M_A + M_B + M_C$$

$$\min \gamma = \gamma_B + \gamma_C$$

$$s.t. \begin{cases} a = 100 \\ 50 \leq b < 100 \\ 10 < c \leq 50 \\ d = 10 \\ M_B \geq 0 \\ M_C \geq 0 \end{cases}$$

#### 4.3.5 模型的求解

对于多目标规划模型，我们先解单目标规划：

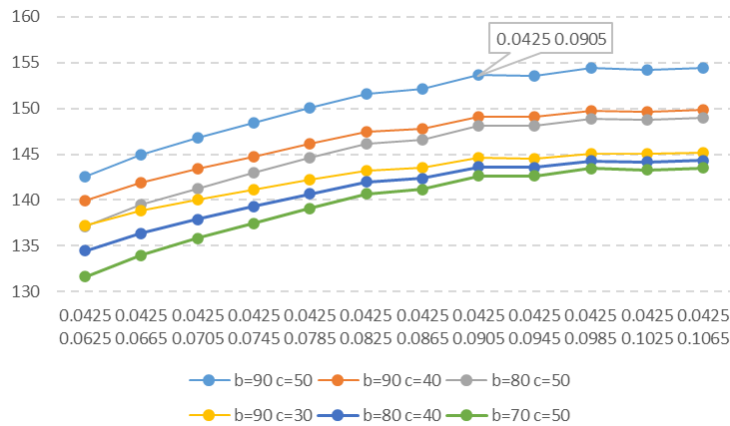
$$\begin{aligned} \max \quad & M = M_A + M_B + M_C \\ \text{s.t.} \quad & \begin{cases} a = 100 \\ 50 \leq b < 100 \\ 10 < c \leq 50 \\ d = 10 \\ M_B \geq 0 \\ M_C \geq 0 \end{cases} \end{aligned}$$

可得 $i_B \geq 0.0270$ ,  $i_C \geq 0.0625$ 。将其作为约束条件，同时固定 $i_B = 0.0425$ , 改变 $b$ 、 $c$ 、 $i_C$ 的值，得到对应的收益 $M$ 如图表3所示：

| b=c   | 0.0425      | 0.0625      | 0.0425      | 0.0665      | 0.0425      | 0.0705      | 0.0425      | 0.0745      | 0.0425      | 0.0785      | 0.0425      | 0.0825      | 0.0425 | 0.0865 | 0.0425 | 0.0905 | 0.0425 | 0.0945 | 0.0425 | 0.0985 | 0.0425 | 0.1025 | 0.0425 | 0.1065 |
|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 90 50 | 142.589859  | 144.9291848 | 146.7579204 | 148.4336331 | 150.0594844 | 151.6084708 | 152.086993  | 153.6072219 | 153.556075  | 154.3792928 | 154.2268161 | 154.4536321 |        |        |        |        |        |        |        |        |        |        |        |        |
| 90 40 | 139.5189661 | 141.8683735 | 143.3923199 | 144.7889971 | 146.1436232 | 147.434452  | 147.8332137 | 149.1000711 | 149.0574467 | 149.7434635 | 149.6180663 | 149.8054129 |        |        |        |        |        |        |        |        |        |        |        |        |
| 80 50 | 137.1193563 | 139.4586452 | 141.2673808 | 142.9633935 | 144.5889448 | 146.1379312 | 146.6164534 | 148.1366823 | 148.0855353 | 148.9087532 | 148.7582765 | 148.9830924 |        |        |        |        |        |        |        |        |        |        |        |        |
| 90 30 | 137.2480363 | 138.8075622 | 140.0267193 | 141.1440611 | 142.227762  | 143.2604196 | 143.5794344 | 144.5929203 | 144.5588223 | 145.1076342 | 145.0073164 | 145.1571937 |        |        |        |        |        |        |        |        |        |        |        |        |
| 80 40 | 134.4494265 | 136.3978339 | 137.9217802 | 139.3184575 | 140.6730836 | 141.9639056 | 142.3626741 | 143.6295315 | 143.586909  | 144.2729239 | 144.1475267 | 144.3348733 |        |        |        |        |        |        |        |        |        |        |        |        |
| 70 50 | 131.6488166 | 133.9801056 | 135.8168412 | 137.4926538 | 139.1184052 | 140.6673915 | 141.1459138 | 142.6661427 | 142.6149957 | 143.4392136 | 143.2877369 | 143.5125528 |        |        |        |        |        |        |        |        |        |        |        |        |
| 90 20 | 134.5771065 | 135.7467509 | 136.6611187 | 137.4991251 | 138.3119007 | 139.0863939 | 139.3256951 | 140.0857695 | 140.060196  | 140.4718049 | 140.3965666 | 140.5099746 |        |        |        |        |        |        |        |        |        |        |        |        |
| 80 30 | 131.7744966 | 133.3370226 | 134.5561797 | 135.6735215 | 136.7572223 | 137.7986799 | 138.1088947 | 139.1223807 | 139.0882827 | 139.6370846 | 139.5367768 | 139.6866541 |        |        |        |        |        |        |        |        |        |        |        |        |
| 70 40 | 128.5778869 | 130.9272943 | 132.4512406 | 133.8479178 | 135.2025439 | 136.4939569 | 136.8921344 | 138.1589919 | 138.1163694 | 138.8023843 | 138.676987  | 138.8643336 |        |        |        |        |        |        |        |        |        |        |        |        |
| 60 50 | 126.178277  | 128.5175659 | 130.3463015 | 132.0223142 | 133.6478655 | 135.1968519 | 135.6753741 | 137.195603  | 137.1444561 | 137.9576739 | 137.8171972 | 138.0420132 |        |        |        |        |        |        |        |        |        |        |        |        |
| 80 20 | 129.105668  | 130.2762113 | 131.1905791 | 132.0285855 | 132.8413611 | 133.6158543 | 133.8551154 | 134.6152299 | 134.5895664 | 135.0012653 | 134.962027  | 135.0394349 |        |        |        |        |        |        |        |        |        |        |        |        |
| 70 30 | 126.306957  | 127.866493  | 129.08564   | 130.2029818 | 131.2866827 | 132.3193403 | 132.6383551 | 133.6518411 | 133.6177431 | 134.166555  | 134.0662372 | 134.2161145 |        |        |        |        |        |        |        |        |        |        |        |        |
| 60 40 | 123.5073472 | 125.4567546 | 126.980701  | 128.3773782 | 129.7320043 | 131.0228263 | 131.4215948 | 132.6884522 | 132.6458298 | 133.3318446 | 133.2064474 | 133.393794  |        |        |        |        |        |        |        |        |        |        |        |        |
| 50 50 | 120.7077374 | 123.0470263 | 124.8757619 | 126.5517746 | 128.1773259 | 129.7263123 | 130.2048345 | 131.7250634 | 131.6739164 | 132.4971343 | 132.3466576 | 132.5714735 |        |        |        |        |        |        |        |        |        |        |        |        |
| 70 20 | 123.6360272 | 124.8056717 | 125.7200395 | 126.5580458 | 127.3708215 | 128.1453147 | 128.3845758 | 129.1446902 | 129.1191168 | 129.5307257 | 129.4554873 | 129.5678953 |        |        |        |        |        |        |        |        |        |        |        |        |
| 60 30 | 120.8364174 | 122.3959433 | 123.6151004 | 124.7324422 | 125.8161431 | 126.8488607 | 127.1678155 | 128.1813014 | 128.1472034 | 128.6960153 | 128.5956975 | 128.7455748 |        |        |        |        |        |        |        |        |        |        |        |        |
| 50 40 | 118.0388076 | 119.986215  | 121.5101613 | 122.9068386 | 124.2614647 | 125.5522867 | 125.9510552 | 127.2179126 | 127.1752901 | 127.851305  | 127.7359078 | 127.9232544 |        |        |        |        |        |        |        |        |        |        |        |        |
| 60 20 | 118.1654876 | 119.335132  | 120.2494988 | 121.0875062 | 121.9002818 | 122.647775  | 122.9140362 | 123.6741506 | 123.6485771 | 124.060188  | 123.9849477 | 124.0973557 |        |        |        |        |        |        |        |        |        |        |        |        |
| 50 30 | 115.3658777 | 116.9254037 | 118.1445608 | 119.2816026 | 120.3458034 | 121.378261  | 121.6972758 | 122.7107618 | 122.6766838 | 123.2254757 | 123.1251579 | 123.2750352 |        |        |        |        |        |        |        |        |        |        |        |        |
| 50 20 | 112.6649479 | 113.8645324 | 114.7789602 | 115.6169666 | 116.4297422 | 117.2042354 | 117.4434965 | 118.203611  | 118.1760375 | 118.5996464 | 118.5144081 | 118.6268816 |        |        |        |        |        |        |        |        |        |        |        |        |

图表3 固定 $i_B = 0.0425$ , 改变 $b$ 、 $c$ 、 $i_C$ 的值，对应的收益表

分析图表3中的数据可得，当 $b$ 、 $c$ 、 $i_B$ 的值一定时，信誉评级为C的企业贷款年利率 $i_C$ 与银行收益 $M$ 呈正相关，在其中我们选取收益前六大的数据进行作图，由图表4可以直观地看到，当 $i_C \leq 0.0905$ 时，收益随着 $i_C$ 的增大而增大， $i_C > 0.0905$ 后，收益略有下降，均在154上下波动，考虑到要使银行客户流失率尽可能低，故选择 $i_C = 0.0905$ 。



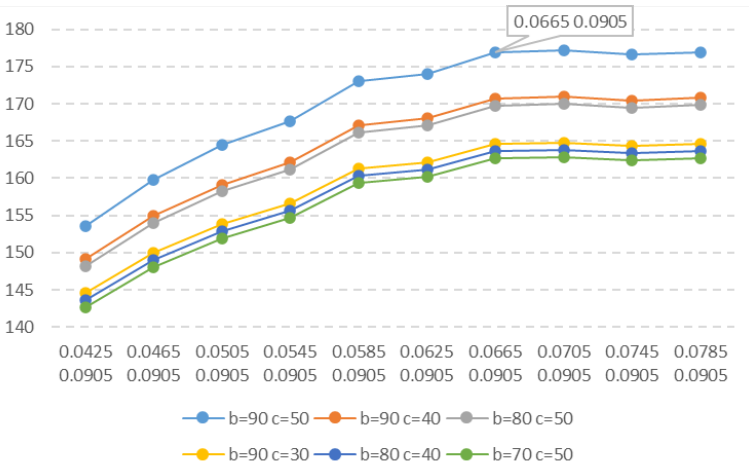
图表4  $b$ 、 $c$ 、 $i_B$ 一定时，收益与 $i_C$ 的关系

固定 $i_C = 0.0905$ , 改变  $b$ 、 $c$ 、 $i_B$  的值, 得到对应的收益  $M$  如图表 5 所示:

| b= | c= | 0.0425 0.0905 | 0.0465 0.0905 | 0.0505 0.0905 | 0.0545 0.0905 | 0.0585 0.0905 | 0.0625 0.0905 | 0.0665 0.0905 | 0.0705 0.0905 | 0.0745 0.0905 | 0.0785 0.0905 |
|----|----|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| 90 | 50 | 153.6072219   | 159.833629    | 164.4467448   | 167.6699887   | 173.0373189   | 174.0740321   | 176.9102778   | 177.2068412   | 176.619658    | 176.9517004   |
| 90 | 40 | 149.1000711   | 154.8817348   | 159.1653424   | 162.1583545   | 167.1423041   | 168.1049663   | 170.738623    | 171.0140033   | 170.4687617   | 170.7770868   |
| 80 | 50 | 148.1366823   | 153.918346    | 158.2019535   | 161.1949657   | 166.1789152   | 167.1415775   | 169.7752341   | 170.0506145   | 169.5053729   | 169.813698    |
| 90 | 30 | 144.5929203   | 149.9298407   | 153.8839399   | 156.6467204   | 161.2472892   | 162.1359004   | 164.5669682   | 164.8211654   | 164.3178655   | 164.6024732   |
| 80 | 40 | 143.629315    | 148.9664518   | 152.9205511   | 155.6833315   | 160.2839004   | 161.1725116   | 163.6035793   | 163.8577766   | 163.3544767   | 163.6390844   |
| 70 | 50 | 142.6661427   | 148.003063    | 151.9571623   | 154.7199427   | 159.3205115   | 160.2091228   | 162.6401905   | 162.8943878   | 162.3910878   | 162.6756956   |
| 90 | 20 | 140.0857695   | 144.9779465   | 148.6025375   | 151.1350862   | 155.3522743   | 156.1668346   | 158.3953134   | 158.6283275   | 158.1669692   | 158.4278597   |
| 80 | 30 | 139.1223807   | 144.0145577   | 147.6391487   | 150.1716974   | 154.3888855   | 155.2034458   | 157.4319245   | 157.6649387   | 157.2035804   | 157.4644709   |
| 70 | 40 | 138.1589919   | 143.0511689   | 146.6757598   | 149.2083086   | 153.4254966   | 154.240057    | 156.4685357   | 156.7015499   | 156.2401916   | 156.501082    |
| 60 | 50 | 137.195603    | 142.08778     | 145.712371    | 148.2449198   | 152.4621078   | 153.2766682   | 155.5051469   | 155.7381611   | 155.2768028   | 155.5376932   |
| 80 | 20 | 134.6152299   | 139.0626635   | 142.3577462   | 144.6600632   | 148.4938706   | 149.23438     | 151.2602697   | 151.4721008   | 151.0526842   | 151.2898573   |
| 70 | 30 | 133.6518411   | 138.0992747   | 141.3943574   | 143.6966744   | 147.5304818   | 148.2709912   | 150.2968809   | 150.508712    | 150.0892953   | 150.3264685   |
| 60 | 40 | 132.6884522   | 137.1358859   | 140.4309686   | 142.7332856   | 146.5670929   | 147.3076023   | 149.3334921   | 149.5453232   | 149.1259065   | 149.3630797   |
| 50 | 50 | 131.7250634   | 136.172497    | 139.4675797   | 141.7698968   | 145.6037041   | 146.3442135   | 148.3701033   | 148.5819343   | 148.1625177   | 148.3996908   |
| 70 | 20 | 129.1446902   | 133.1473805   | 136.1129549   | 138.1850403   | 141.6354669   | 142.3019253   | 144.1252261   | 144.3158741   | 143.9383991   | 144.1518549   |
| 60 | 30 | 128.1813014   | 132.1839917   | 135.1495661   | 137.2216515   | 140.6720781   | 141.3385365   | 143.1618373   | 143.3524853   | 142.9750103   | 143.1884661   |
| 50 | 40 | 127.2179126   | 131.2206029   | 134.1861773   | 136.2582626   | 139.7086892   | 140.3751477   | 142.1984485   | 142.3890964   | 142.0116215   | 142.2250773   |
| 60 | 20 | 123.6741506   | 127.2320975   | 129.8681637   | 131.7100173   | 134.7770632   | 135.3694707   | 136.9901825   | 137.1596474   | 136.824114    | 137.0138526   |
| 50 | 30 | 122.7107618   | 126.2687087   | 128.9047749   | 130.7466285   | 133.8136744   | 134.4060819   | 136.0267937   | 136.1962585   | 135.8607252   | 136.0504637   |
| 50 | 20 | 118.203611    | 121.3168145   | 123.6233724   | 125.2349943   | 127.9185595   | 128.437016    | 129.8551389   | 130.0034206   | 129.709829    | 129.8758502   |

图表 5 固定 $i_C = 0.0905$ , 改变  $b$ 、 $c$ 、 $i_B$  的值, 对应的收益表

分析图表 5 中的数据可得, 当  $b$ 、 $c$ 、 $i_C$  的值一定时, 信誉评级为 B 的企业贷款年利率  $i_B$  与银行收益  $M$  呈正相关, 在其中我们选取收益前六大的数据进行作图, 由图表 6 可以直观地看到, 当  $i_B \leq 0.0665$  时, 收益随着  $i_B$  的增大而增大,  $i_B > 0.0665$  后, 收益均在 177 上下波动, 考虑到要使银行客户流失率尽可能低, 故选择  $i_B = 0.0665$ 。



图表 6  $b$ 、 $c$ 、 $i_C$  一定时, 收益与  $i_B$  的关系

最后, 我们固定  $i_B = 0.0665$ 、 $i_C = 0.0905$ , 改变  $b$ 、 $c$  的值, 得到对应的收益如图表 7 所示:

| c\b | 90       | 80       | 70       | 60       | 50       |
|-----|----------|----------|----------|----------|----------|
| 50  | 176.9103 | 169.7752 | 162.6402 | 155.5051 | 148.3701 |
| 40  | 170.7386 | 163.6036 | 156.4685 | 148.3701 | 142.1984 |
| 30  | 164.567  | 157.4319 | 150.2969 | 143.1618 | 136.0268 |
| 20  | 158.3953 | 151.2603 | 144.1252 | 136.9902 | 129.8551 |

图表 7  $i_B$   $i_C$ 一定时, 改变 b、c 的值, 对应的收益表

由图表 7 中的数据可以迅速得出结论, b 类贷款额度和 c 类贷款额度越大, 银行收益越大。由此可确定最优信贷策略, 即  $b=90$ ,  $i_B = 0.0665$ ,  $c=50$ ,  $i_C = 0.0905$ , 具体贷款额度与年利率的策略如图表 8 所示:

| 信誉评级 | 规模 | 贷款额度  | 年利率       |
|------|----|-------|-----------|
| A    | 中型 | a=100 | iA=0.04   |
|      | 小型 | b=90  | iA=0.04   |
|      | 微型 | d=10  | iA=0.04   |
| B    | 中型 | b=90  | iB=0.0665 |
|      | 小型 | c=50  | iB=0.0665 |
|      | 微型 | d=10  | iB=0.0665 |
| C    | 中型 | c=50  | iC=0.0905 |
|      | 小型 | d=10  | iC=0.0905 |
|      | 微型 | d=10  | iC=0.0905 |
| D    | -  | -     | -         |

图表 8 信贷策略中的贷款额度与年利率

## 五、问题二的模型建立与求解

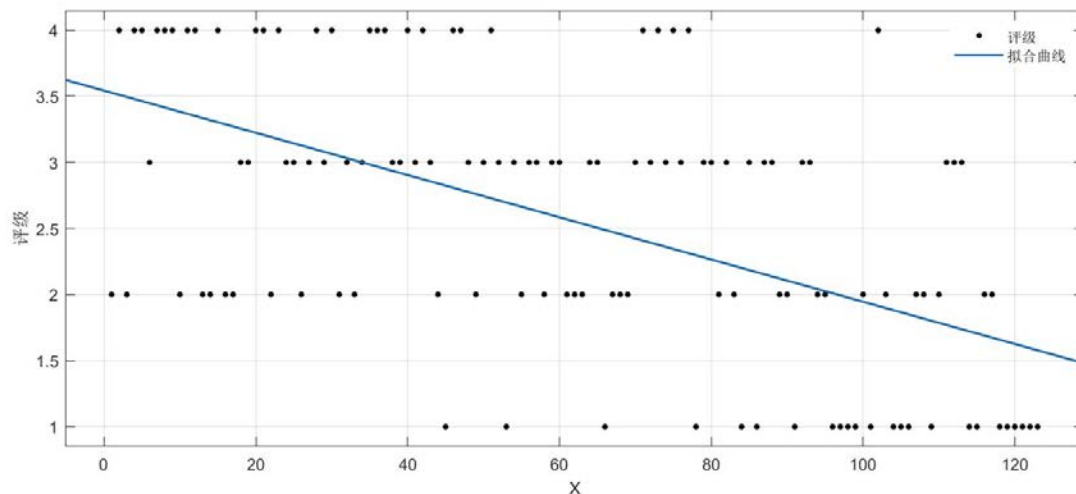
### 5.1 问题的分析

问题二与问题一的差别在于, 附件 2 给出的 302 家企业信息中不包括企业的信贷记录, 即信誉评级和违约情况, 故我们需要分析问题一中模型计算出的企业信贷风险值与信誉评级间的关系, 然后通过计算附件 2 中 302 家企业的信贷风险判断出企业各自的信誉评级。同时, 由于现在已知银行年度信贷总额为 1 亿元, 因此需要在问题一中建立的多目标优化模型中加入这一约束条件, 从而得出新的最优信贷策略。

### 5.2 模型的准备

由问题一中的信贷风险综合评估模型可得, 计算得到的信贷风险与附件 1 中给出的信誉评级的拟合度为分别为信誉评级 A: 0.4444, 信誉评级 B: 0.4474, 信誉评级 C: 0.2671, 信誉评级 D: 0.5417, 并得到 123 家企业风险评级排名与信誉评级的关系如图表 9 所示, 其中 A=4, B=3, C=2, D=1, 横坐标为信贷风险综合评估模型得到的企业风险排名, 排名越高, 风险越小。





图表 9 123 家企业风险评级排名与信誉评级的关系

通过风险评级排名与信誉评级之间的关系，我们可以为附件 2 中 302 家企业进行信誉评级打分，得到风险评级排名前 66 名的企业信誉评级为 A，67-159 名的企业信誉评级为 B，160-243 名的企业信誉评级为 C，244-301 名的企业信誉评级为 D。

### 5.3 信贷风险综合评估模型

在附件 2 中的 302 家企业均得到对应的信誉评级后，将其其他信息带入问题一建立的企业信贷风险综合评估模型中，得到相应的贷款风险评级排序，按照顺序发放贷款，因数据较多，在这里只展示排名前 10 的企业，如图表 10 所示。

| 企业代号 | 最终得分   |
|------|--------|
| E153 | 0.0313 |
| E297 | 0.0236 |
| E193 | 0.0228 |
| E125 | 0.0227 |
| E124 | 0.0215 |
| E329 | 0.019  |
| E199 | 0.0188 |
| E128 | 0.0161 |
| E127 | 0.0153 |
| E139 | 0.0119 |
| ...  | ...    |

图表 10 排名前十的企业最终得分

### 5.4 新双目标优化下的信贷策略

第二问题目规定，该银行年度信贷总额为 1 亿元，将其作为约束条件带入第一问的双目标优化模型中：

$$\begin{aligned} \max \quad & M = M_A + M_B + M_C \\ \min \quad & \gamma = \gamma_B + \gamma_C \end{aligned}$$

$$s. t. \begin{cases} K \leq 10000 \\ 50 \leq b < 100 \\ 10 < c \leq 50 \\ d = 10 \\ M_B \geq 0 \\ M_C \geq 0 \end{cases}$$

其中  $K = A_1 \cdot a + (A_2 + B_1) \cdot b + (A_2 + B_1) \cdot b + (B_2 + C_1) \cdot c + (A_3 + B_3 + C_2 + C_3) \cdot d$ ，解该双目标优化模型，可以得到在银行年度信贷总额为 1 亿元时，最优贷款额度与年利率的策略如图表 11 所示：

| 信誉评级 | 规模 | 贷款额度 | 年利率       |
|------|----|------|-----------|
| A    | 中型 | a=62 | iA=0.04   |
|      | 小型 | b=56 | iA=0.04   |
|      | 微型 | d=10 | iA=0.04   |
| B    | 中型 | b=56 | iB=0.0665 |
|      | 小型 | c=31 | iB=0.0665 |
|      | 微型 | d=10 | iB=0.0665 |
| C    | 中型 | c=31 | iC=0.0905 |
|      | 小型 | d=10 | iC=0.0905 |
|      | 微型 | d=10 | iC=0.0905 |
| D    | -  | -    | -         |

图表 11 年度信贷总额为 1 亿元时的最优信贷策略

## 六、问题三的模型建立与求解

### 6.1 问题的分析

由于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，因此问题三在基于企业信贷综合风险评估模型的基础上新增了一个与突发因素有关的变量进行综合考虑分析。以新冠病毒疫情为例，在问题一的综合评估模型中增加了疫情对行业的负面影响程度的评价指标进行分析。

### 6.2 模型的准备

在新冠病毒的作用下，今年许多行业均出现了不同程度的打击，本题在此思路，增加这一变量来定量分析疫情对各个行业的负面影响程度。由于附件 2 中的企业拥有各自不同的职能，在疫情下的影响程度不尽相同，因此参照《统计上大中小微型企业划分办法(2017)》中将 302 家企业划分成房地产与物业服务、个体经营、工业、建筑业、农林牧渔业、软件和科学信息技术服务业、文化业、销售业、医药业、运输和租赁和商务服务业十种类别，通过查找文献对其进行科学的等级评定，得到结果如图表 12 所示，其中负数表示疫情对行业有正面影响，0 表示疫情对行业基本无影响，正数表示疫情对行业有负面影响，且数字越大负面影响越大。

| 类型           | 个数 | 疫情对行业的负面影响程度 |
|--------------|----|--------------|
| 房地产与物业服务     | 8  | 1            |
| 个体经营         | 56 | 2            |
| 工业           | 51 | 1            |
| 建筑业          | 52 | 0            |
| 农林牧渔业        | 8  | 2            |
| 软件和科学信息技术服务业 | 33 | -1           |
| 文化业          | 18 | 2            |
| 销售业          | 21 | 2            |
| 医药业          | 6  | -1           |
| 运输业          | 14 | 1            |
| 租赁和商务服务业     | 35 | 2            |

图表 12 疫情对行业的负面影响程度表

### 6.3 模型的建立与求解

在问题一的企业贷款风险综合评估模型中加入疫情对行业负面影响程度的评价指标，通过查询资料及熵权法得到各指标新的熵权（如图表 13 所示），通过 TOPSIS 法得到各家企业新风险评估排序（如图表 14 所示），按照该顺序发放贷款。因该银行年度信贷总额没有改变，故双目标优化下对各信誉评级和规模的企业贷款额与年利率不发生改变，即如图表 11 所示。

| 指标 | 进项作废率   | 开票频率   | 销售额环比   | 净收入率环比   | 销项负数率 | 规模       | 疫情负面影响程度 |
|----|---------|--------|---------|----------|-------|----------|----------|
| 熵权 | 0.00645 | 0.1491 | 0.03735 | 0.318375 | 0.003 | 0.235725 | 0.25     |

图表 13 各指标对应的新熵权

| 企业代号 | 最终得分   |
|------|--------|
| E153 | 0.0232 |
| E297 | 0.0175 |
| E193 | 0.0169 |
| E125 | 0.0168 |
| E124 | 0.0159 |
| E199 | 0.0142 |
| E329 | 0.0141 |
| E128 | 0.0119 |
| E127 | 0.0113 |
| E139 | 0.0088 |
| ...  | ...    |

图表 14 新排名前 10 的企业最终得分

## 七、模型评价

### 7.1 模型一

#### 优点：

1. Topsis 基于检测评价对象与最优解，最劣解的距离来进行排序。在具有多组评价对象时，评价优势明显。
2. 本模型包含了进项作废率，开票频率，销售额环比，净收入率环比，销项负数率以及规模等多种评价角度，使评价效果更为直观，可靠。



3. 熵权法的加入使得不同因素有了权重上的区分, 因此模型在进行评估时能有效突出重点, 降低干扰。

**缺点:**

1. 各评价指标之间存在相关性, 未能充分考虑到评价指标之间的相互作用。
2. 模型参数的设定受人为主观因素的影响, 降低了准确性。

## 7.2 模型二

**优点:**

1. 双目标规划能够很好地解决模型评估中因指标众多而难以讨论的问题。
2. 双目标优化可以遍历所有情况, 从而得到最优策略。

**缺点:**

1. 为了简化讨论, 我们设定 a 类贷款额度为 100 万元, d 类贷款额度为 10 万元, 没有考虑对其进行讨论。
2. 双目标规划得到的最优解并不唯一, 需要进行人工选择, 可能会产生误差。

## 八、参考文献

- [1] 宋用国. H 商业银行小微企业信贷服务优化策略研究[D]. 西南财经大学, 2017.
- [2] 张潇予. 工行吉林省分行小微企业贷款信用风险管理研究[D]. 吉林大学, 2019.
- [3] 费磊. 新冠疫情对银行机构行业贷款质量影响的调查——以日照为例[J]. 黑龙江金融, 2020(03): 36-38.
- [4] 孙俊意. 分析新冠疫情对服务业的影响——参考“非典”疫情[J]. 经济师, 2020(08): 58-60.
- [5] 麻红萍. 新冠肺炎疫情对房地产行业影响及对策研究[J]. 湖北经济学院学报(人文社会科学版), 2020, 17(06): 23-24.
- [6] 李亚光, 李芳芳. 新冠肺炎疫情对我国工业的影响与对策研究[J]. 产业经济评论, 2020(02): 5-12.
- [7] 何旭. 新冠肺炎疫情常态化防控带给建筑业企业的启示[N]. 中国建设报, 2020-08-14(008).
- [8] 方磊. 新冠疫情对个体工商户的影响及金融对策[J]. 银行家, 2020(04): 22-25.
- [9] 李国庆. 从新冠疫情防控看大数据技术的应用价值[J]. 厦门科技, 2020(03): 8-10.
- [10] 孙艳阳, 俞尚茗. 互联网+文化产业的机遇与挑战——基于疫情影响下的文化产业新业态分析[J]. 大众文艺, 2020(14): 229-230.
- [11] 易楚斌. 危中寻机——新冠肺炎疫情下的医药行业发展机遇[J]. 公关世界, 2020(14): 8-10.
- [12] 乔治·华盛顿大学教授、国际中小企业联合会执行董事 艾曼·塔拉比希. 2020 年中小微企业十大发展趋势[N]. 2020-09-04(009).
- [13] 赵树宽, 邵东, 王泷, 李师萌, 李佳逸. 新冠肺炎疫情对吉林省企业的影响及对策建议——基于对吉林省 336 家企业的问卷调查[J]. 吉林大学社会科学学报, 2020, 60(05): 60-70+236. DOI:10.15939/j.jujss.2020.05.jj2.

## 附录

```
1. import xlrd
2. import xlwt
3. excel_path = "附件 2: 302 家无信贷记录企业的相关数据.xlsx"
4. #打开文件 获取 workbook
5. excel = xlrd.open_workbook(excel_path,encoding_override = "utf-8")
6. #返回所有 sheet 对象的 list
7. all_sheet = excel.sheets()#Book(工作簿)对象方法
8. print(all_sheet)
9. for each_sheet in all_sheet:
10.     print(each_sheet)
11.     print("sheet 名称为: ",each_sheet.name)
12. for sheet in all_sheet:
13.     first_row_value = sheet.row_values(1)#打印指定的某一行
14.     print("第一行的数据为:%s" % first_row_value)
15. #进项作废
16. workbook = xlwt.Workbook(encoding='utf-8')
17. worksheet = workbook.add_sheet('Sheet1')
18. sheet = all_sheet[2]
19. waste = 0
20. cnt = 0
21. j = 0
22. waste_pro = []
23. for i in range (1,395175):
24.     tmp = sheet.row_values(i)
25.     if(i != 1 and tmp[0] == sheet.row_values(i - 1)[0]):
26.         cnt = cnt + 1
27.         if(tmp[7] == "作废发票"):
28.             waste = waste + 1
29.     elif(i == 1):
30.         cnt = cnt + 1
31.         if(tmp[7] == "作废发票"):
32.             waste = waste + 1
33.     elif(i == 395175):
34.         cnt = cnt + 1
35.         if(tmp[7] == "作废发票"):
36.             waste = waste + 1
37.     waste_pro.append(waste/cnt)
38. else:
39.     waste_pro.append(waste/cnt)
40.     j = j + 1
41.     cnt = 1
42.     if(tmp[7] == "作废发票"):
```

```

43.         waste = 1
44.     else:
45.         waste = 0
46. for i in range(len(waste_pro)):
47.     worksheet.write(i, 0, waste_pro[i])
48.
49. workbook.save('进项作废_2.xls')
50. #营销额
51. import datetime
52. from xlrd import xldate_as_datetime, xldate_as_tuple
53. workbook = xlwt.Workbook(encoding='utf-8')
54. worksheet = workbook.add_sheet('Sheet1')
55. style = xlwt.XFStyle() # 初始化样式
56. font = xlwt.Font() # 为样式创建字体
57. font.name = 'Times New Roman'
58. font.height = 20 * 11 # 字体大小, 11 为字号, 20 为衡量单位
59. font.bold = True # 黑体
60. font.underline = True # 下划线
61. font.italic = True # 斜体字
62. style.font = font # 设定样式
63. #规模 最后一年的销售额 (销项价税合计)
64. resheet = all_sheet[1]
65. last_year = [] #每家公司有数据的最后一年
66. for i in range(1, 330835):
67.     tmp = resheet.row_values(i)
68.     if(i == 330835):
69.         last_year.append(datetime.datetime(*xldate_as_tuple(tmp[2], 0)))
70.     elif(i != 1 and tmp[0] != resheet.row_values(i - 1)[0]):
71.         last_year.append(datetime.datetime(*xldate_as_tuple(resheet.row_values(i - 1)[2], 0)))
72. #获取每一家公司的最后一年的第一天
73. first_day = []
74. for i in range(0, len(last_year) - 1):
75.     first_day.append(last_year[i] - datetime.timedelta(days = 365))
76. for i in range(0, len(last_year) - 1):
77.     print(first_day[i])
78.     print(" ")
79.     print(last_year[i])
80.     print("\n")
81. #计算每一家公司在最后一年的销售额
82. print(len(last_year))
83. print(len(first_day))
84. summ = []
85. t = 0
86. cnt = 0

```

```

87. for i in range(1,330835):
88.     tmp = resheet.row_values(i)
89.     if(cnt < len(first_day) and datetime.datetime(*xldate_as_tuple(tmp[2],0)) <= last_year[cnt] a
        nd datetime.datetime(*xldate_as_tuple(tmp[2],0)) >= first_day[cnt]):
90.         t = t + tmp[6]
91.         if(datetime.datetime(*xldate_as_tuple(tmp[2],0)) == last_year[cnt]):
92.             cnt = cnt + 1
93.             summ.append(t)
94.             t = 0
95. print(summ)
96. for i in range(len(summ)):
97.     worksheet.write(i, 0, summ[i])
98. workbook.save('营销额_2.xls')
99. #开票频率
100. workbook = xlwt.Workbook(encoding='utf-8')
101. worksheet = workbook.add_sheet('Sheet1')
102. resheet = all_sheet[1]#访问第 1 个 sheet(销项)
103. cnt = 0 #某公司的开票总数
104. delta = 0 #最后时间-最初时间
105. frequency = [] #开票频率
106. start_time = datetime.datetime(*xldate_as_tuple(resheet.row_values(1)[2],0))
107. end_time = start_time
108. j = 1
109. for i in range(1,330835):
110.     tmp = resheet.row_values(i)
111.     if(i == 1):
112.         cnt = cnt + 1
113.     elif(tmp[0] != resheet.row_values(i-1)[0] or i == 330835):
114.         end_time = datetime.datetime(*xldate_as_tuple(resheet.row_values(i-1)[2],0))
115.         delta = (end_time - start_time).days
116.         frequency.append(cnt/delta)
117.         worksheet.write(j,0,frequency[j - 1])
118.         j = j + 1
119.         cnt = 1
120.         delta = 0
121.         start_time = datetime.datetime(*xldate_as_tuple(tmp[2],0))
122.         end_time = start_time
123.     else:
124.         cnt = cnt + 1
125. workbook.save('开票频率_2.xls')
126. #销项负数
127. import xlrd
128. import xlwt
129. excel_path = "附件 2: 302 家无信贷记录企业的相关数据.xlsx"

```

```

130. #打开文件 获取 workbook
131. excel = xlrd.open_workbook(excel_path,encoding_override = "utf-8")
132. #返回所有 sheet 对象的 list
133. all_sheet = excel.sheets()
134. workbook = xlwt.Workbook(encoding='utf-8')
135. worksheet = workbook.add_sheet('Sheet1')
136. sheet = all_sheet[1]
137. waste = 0
138. cnt = 0
139. j = 0
140. waste_pro = []
141. for i in range (1,330835):
142.     tmp = sheet.row_values(i)
143.     if(i != 1 and tmp[0] == sheet.row_values(i - 1)[0]):
144.         cnt = cnt + 1
145.         if(tmp[6] < 0):
146.             waste = waste + 1
147.         elif(i == 1):
148.             cnt = cnt + 1
149.             if(tmp[6] < 0):
150.                 waste = waste + 1
151.         elif(i == 330835):
152.             cnt = cnt + 1
153.             if(tmp[6] < 0):
154.                 waste = waste + 1
155.             waste_pro.append(waste/cnt)
156.         else:
157.             waste_pro.append(waste/cnt)
158.             j = j + 1
159.             cnt = 1
160.             if(tmp[6] < 0):
161.                 waste = 1
162.             else:
163.                 waste = 0
164. for i in range(len(waste_pro)):
165.     worksheet.write(i, 0,waste_pro[i])
166.
167. workbook.save('销项负数_2.xls')
168. #销售环比
169. import xlrd
170. import xlwt
171. import datetime
172. from xlrd import xldate_as_datetime, xldate_as_tuple
173. excel_path = "附件 2: 302 家无信贷记录企业的相关数据.xlsx"

```

```

174. excel = xlrd.open_workbook(excel_path,encoding_override = "utf-8")
175. workbook = xlwt.Workbook(encoding='utf-8')
176. worksheet = workbook.add_sheet('Sheet1')
177. all_sheet = excel.sheets()
178. resheet = all_sheet[1]
179. last_year = [] #每家公司有数据的最后一年
180. for i in range(1,330835):
181.     tmp = resheet.row_values(i)
182.     if(i == 330835):
183.         last_year.append(datetime.datetime(*xldate_as_tuple(tmp[2],0)))
184.     elif(i != 1 and tmp[0] != resheet.row_values(i - 1)[0]):
185.         last_year.append(datetime.datetime(*xldate_as_tuple(resheet.row_values(i - 1)[2],0)
        ))
186. #获取每一家公司的最后一年的第一天
187. first_day = []
188. for i in range(0,len(last_year) - 1):
189.     first_day.append(last_year[i] - datetime.timedelta(days = 365))
190. #销售环比
191. ratio = [] #存一个公司最后四个季度的环比均值
192. cnt = 0 #标记在 last_year 和 first_day 数组内的进程
193. sum1 = 0
194. sum2 = 0
195. sum3 = 0
196. sum4 = 0
197. for i in range(1,330835):
198.     tmp = resheet.row_values(i)
199.     if(i != 1 and tmp[0] != resheet.row_values(i - 1)[0]):
200.         result = 0
201.         if(sum1 == 0 and sum2 == 0 and sum3 == 0):
202.             result = 0
203.         elif(sum1 == 0 and sum2 == 0):
204.             result = (sum4 - sum3)/sum3
205.         elif(sum1 == 0 and sum3 == 0):
206.             result = (sum3 - sum2)/sum2
207.         elif(sum2 == 0 and sum3 == 0):
208.             result = (sum2 - sum1)/sum1
209.         elif(sum1 == 0):
210.             result = ((sum3 - sum2)/sum2 + (sum4 - sum3)/sum3)/2
211.         elif(sum2 == 0):
212.             result = ((sum2 - sum1)/sum1 + (sum4 - sum3)/sum3)/2
213.         elif(sum3 == 0):
214.             result = ((sum2 - sum1)/sum1 + (sum3 - sum2)/sum2)/3
215.         else:
216.             result = ((sum2 - sum1)/sum1 + (sum3 - sum2)/sum2 + (sum4 - sum3)/sum3)/3

```

```

217.         ratio.append(result)
218.         cnt = cnt + 1
219.         sum1 = 0
220.         sum2 = 0
221.         sum3 = 0
222.         sum4 = 0
223.         if(cnt < len(first_day)):
224.             if((datetime.datetime(*xldate_as_tuple(tmp[2],0)).month - first_day[cnt].month)/3 =
= 0): #第一季度
225.                 sum1 = sum1 + tmp[6]
226.                 elif((datetime.datetime(*xldate_as_tuple(tmp[2],0)).month - first_day[cnt].month)/3
== 1):
227.                     sum2 = sum2 + tmp[6]
228.                     elif((datetime.datetime(*xldate_as_tuple(tmp[2],0)).month - first_day[cnt].month)/3
== 2):
229.                         sum3 = sum3 + tmp[6]
230.                         elif((datetime.datetime(*xldate_as_tuple(tmp[2],0)).month - first_day[cnt].month)/3
== 3):
231.                             sum4 = sum4 + tmp[6]
232.         for i in range(0,len(ratio) - 1):
233.             worksheet.write(i, 0,ratio[i])
234.         workbook.save('销售环比_2.xls')
235.         #利润环比
236.         import xlrd
237.         import xlwt
238.         import datetime
239.         from xlrd import xldate_as_datetime, xldate_as_tuple
240.         import numpy
241.         excel_path = "附件 2: 302 家无信贷记录企业的相关数据.xlsx"
242.         excel = xlrd.open_workbook(excel_path,encoding_override = "utf-8")
243.         workbook = xlwt.Workbook(encoding='utf-8')
244.         worksheet = workbook.add_sheet('Sheet1')
245.         all_sheet = excel.sheets()
246.         resheet = all_sheet[1]
247.         sheet = all_sheet[2]
248.         last_year = [] #每家公司有数据的最后一年
249.         for i in range(1,330835):
250.             tmp = resheet.row_values(i)
251.             if(i == 330835):
252.                 last_year.append(datetime.datetime(*xldate_as_tuple(tmp[2],0)))
253.             elif(i != 1 and tmp[0] != resheet.row_values(i - 1)[0]):
254.                 last_year.append(datetime.datetime(*xldate_as_tuple(resheet.row_values(i - 1)[2],0)
))
255.         #获取每一家公司的最后一年的第一天

```

```

256. first_day = []
257. for i in range(0, len(last_year) - 1):
258.     first_day.append(last_year[i] - datetime.timedelta(days = 365))
259. #利润率环比
260. sell_list = [[0]*303 for i in range(4)]#销项金额
261. buy_list = [[0]*303 for i in range(4)]#进项价税合计
262. cnt = 0
263. ratio_get = []
264. for i in range(1, 330835):#销项金额
265.     tmp = resheet.row_values(i)
266.     if(i != 1 and tmp[0] != resheet.row_values(i - 1)[0]):
267.         cnt = cnt + 1
268.         if(cnt < len(first_day)):
269.             if((datetime.datetime(*xldate_as_tuple(tmp[2], 0)).month - first_day[cnt].month)/3 =
= 0): #第一季度
270.                 sell_list[0][cnt] = sell_list[0][cnt] + float(tmp[4])
271.                 elif((datetime.datetime(*xldate_as_tuple(tmp[2], 0)).month - first_day[cnt].month)/3
== 1):
272.                     sell_list[1][cnt] = sell_list[1][cnt] + float(tmp[4])
273.                     elif((datetime.datetime(*xldate_as_tuple(tmp[2], 0)).month - first_day[cnt].month)/3
== 2):
274.                         sell_list[2][cnt] = sell_list[2][cnt] + float(tmp[4])
275.                         elif((datetime.datetime(*xldate_as_tuple(tmp[2], 0)).month - first_day[cnt].month)/3
== 3):
276.                             sell_list[3][cnt] = sell_list[3][cnt] + float(tmp[4])
277.         cnt = 0
278.     for i in range(1, 395175):
279.         tmp = sheet.row_values(i)
280.         if(i != 1 and tmp[0] != sheet.row_values(i - 1)[0]):
281.             cnt = cnt + 1
282.             if(cnt < len(first_day)):
283.                 if((datetime.datetime(*xldate_as_tuple(tmp[2], 0)).month - first_day[cnt].month)/3 =
= 0): #第一季度
284.                     buy_list[0][cnt] = buy_list[0][cnt] + float(tmp[6])
285.                     elif((datetime.datetime(*xldate_as_tuple(tmp[2], 0)).month - first_day[cnt].month)/3
== 1):
286.                         buy_list[1][cnt] = buy_list[1][cnt] + float(tmp[6])
287.                         elif((datetime.datetime(*xldate_as_tuple(tmp[2], 0)).month - first_day[cnt].month)/3
== 2):
288.                             buy_list[2][cnt] = buy_list[2][cnt] + float(tmp[6])
289.                             elif((datetime.datetime(*xldate_as_tuple(tmp[2], 0)).month - first_day[cnt].month)/3
== 3):
290.                                 buy_list[3][cnt] = buy_list[3][cnt] + float(tmp[6])
291.     for i in range(0, 302):

```



```

292.         if(buy_list[0][i] == 0 and buy_list[1][i] == 0 and buy_list[2][i] == 0 and buy_list[3][
           i] == 0):
293.             result = 0
294.         elif(buy_list[0][i] == 0 and buy_list[1][i] == 0 and buy_list[2][i] == 0):
295.             result = (sell_list[3][i] - buy_list[3][i])/buy_list[3][i]
296.         elif(buy_list[0][i] == 0 and buy_list[1][i] == 0 and buy_list[3][i] == 0):
297.             result = (sell_list[2][i] - buy_list[2][i])/buy_list[2][i]
298.         elif(buy_list[0][i] == 0 and buy_list[2][i] == 0 and buy_list[3][i] == 0):
299.             result = (sell_list[1][i] - buy_list[1][i])/buy_list[1][i]
300.         elif(buy_list[1][i] == 0 and buy_list[2][i] == 0 and buy_list[3][i] == 0):
301.             result = (sell_list[0][i] - buy_list[0][i])/buy_list[0][i]
302.         elif(buy_list[0][i] == 0 and buy_list[1][i] == 0):
303.             result = ((sell_list[2][i] - buy_list[3][i])/buy_list[2][i]+(sell_list[3][i] - buy_
               list[3][i])/buy_list[3][i])/2
304.         elif(buy_list[0][i] == 0 and buy_list[2][i]== 0):
305.             result = ((sell_list[1][i] - buy_list[1][i])/buy_list[1][i]+(sell_list[3][i] - buy_
               list[3][i])/buy_list[3][i])/2
306.         elif(buy_list[0][i] == 0 and buy_list[3][i]==0):
307.             result = ((sell_list[1][i] - buy_list[1][i])/buy_list[1][i]+(sell_list[2][i] - buy_
               list[2][i])/buy_list[2][i])/2
308.         elif(buy_list[1][i] == 0 and buy_list[2][i]==0):
309.             result = ((sell_list[0][i] - buy_list[0][i])/buy_list[0][i]+(sell_list[3][i] - buy_
               list[3][i])/buy_list[3][i])/2
310.         elif(buy_list[1][i] == 0 and buy_list[3][i]==0):
311.             result = ((sell_list[0][i] - buy_list[0][i])/buy_list[0][i]+(sell_list[2][i] - buy_
               list[2][i])/buy_list[2][i])/2
312.         elif(buy_list[2][i] == 0 and buy_list[3][i]==0):
313.             result = ((sell_list[0][i] - buy_list[0][i])/buy_list[0][i]+(sell_list[1][i] - buy_
               list[1][i])/buy_list[1][i])/2
314.         elif(buy_list[0][i] == 0):
315.             result = ((sell_list[1][i] - buy_list[1][i])/buy_list[1][i]+(sell_list[2][i] - buy_
               list[2][i])/buy_list[2][i]+(sell_list[3][i] - buy_list[3][i])/buy_list[3][i])/3
316.         elif(buy_list[1][i] == 0):
317.             result = ((sell_list[0][i] - buy_list[0][i])/buy_list[0][i]+(sell_list[2][i] - buy_
               list[2][i])/buy_list[2][i]+(sell_list[3][i] - buy_list[3][i])/buy_list[3][i])/3
318.         elif(buy_list[2][i] == 0):
319.             result = ((sell_list[1][i] - buy_list[1][i])/buy_list[1][i]+(sell_list[0][i] - buy_
               list[0][i])/buy_list[0][i]+(sell_list[3][i] - buy_list[3][i])/buy_list[3][i])/3
320.         elif(buy_list[3][i] == 0):
321.             result = ((sell_list[1][i] - buy_list[1][i])/buy_list[1][i]+(sell_list[2][i] - buy_
               list[2][i])/buy_list[2][i]+(sell_list[0][i] - buy_list[0][i])/buy_list[0][i])/3
322.         else:

```

```

323.         result = ((sell_list[0][i] - buy_list[0][i])/buy_list[0][i]+(sell_list[1][i] - buy_
        list[1][i])/buy_list[1][i]+(sell_list[2][i] - buy_list[2][i])/buy_list[2][i]+(sell_list[3][i] -
        buy_list[3][i])/buy_list[3][i])/4
324.         ratio_get.append(result)
325.     for i in range(0,len(ratio_get) - 1):
326.         worksheet.write(i, 0,ratio_get[i])
327.     workbook.save('利润环比_2.xls')
328.     #税率
329.     import xlrd
330.     import xlwt
331.     excel_path = "附件 2: 302 家无信贷记录企业的相关数据.xlsx"
332.     #打开文件 获取 workbook
333.     excel = xlrd.open_workbook(excel_path,encoding_override = "utf-8")
334.     #返回所有 sheet 对象的 list
335.     all_sheet = excel.sheets()#Book(工作簿)对象方法
336.     #税率
337.     workbook = xlwt.Workbook(encoding='utf-8')
338.     worksheet = workbook.add_sheet('Sheet1')
339.     sheet = all_sheet[1]
340.     tax = 0 #计算税
341.     get = 0 #价税合计
342.     ratio = [] #税率
343.     for i in range (1,330835):
344.         tmp = sheet.row_values(i)
345.         if(i != 1 and tmp[0] != sheet.row_values(i - 1)[0]):
346.             ratio.append(tax/get)
347.             tax = float(tmp[5])
348.             get= float(tmp[6])
349.         elif(i == 1):
350.             tax = tax + float(tmp[5])
351.             get= get + float(tmp[6])
352.         elif(i == 330835):
353.             tax = tax + float(tmp[5])
354.             get= get + float(tmp[6])
355.             ratio.append(tax/get)
356.         else:
357.             tax = tax + float(tmp[5])
358.             get= get + float(tmp[6])
359.     for i in range(len(ratio)):
360.         worksheet.write(i, 0,ratio[i])
361.
362.     workbook.save('税率_2.xls')

```

Java:

1. ##多目标规划

```

2. package demo;
3.
4. public class i {
5.     public static void main(String[] args) {
6.         double ib = 0.0785;
7.         double ic = 0.0905;
8.         double x = 0.455591163;
9.         double y = 0.409902955;
10.        double m = ib*37/38-0.02632;
11.        double n = ic*32/34-0.05882;
12.        for(int b = 50; b <= 90; b = b + 10 ) {
13.            for(int c = 50; c >= 20; c = c - 10) {
14.                System.out.println(20*100*0.04+7*b*0.04+19*b*x*m+19*c*x*m+
15.                    17*c*y*n+(16+1)*10*y*n);
16.            }
17.        }
18.    }
19. }

```