基于随机森林的熵权排序信贷决策模型

摘要

在大众创业，万众创新的时代背景下，中小微企业以初生资本的姿态活跃在市场上，而银行又是中小微企业融资的重要来源，但中小微企业贷款常常遭遇等待时间过长的状况。因此，本文建立合理的数学模型评估中小微企业的信贷风险，对于减少银行审核的人力成本和时间成本具有重要意义。

针对问题一，我们的目标是给出是否放贷、贷款额度、利率等信贷策略。首先我们基于 123 家有信贷记录企业的相关数据，挖掘企业信贷风险评估指标，如企业规模、合作依存关系、企业资金链稳定程度等，并通过**随机森林回归**建立信贷风险量化模型，准确率达到了 87.26%。对于贷款额度的确定，我们使用多项式进行利率和客户流失率的函数拟合，利用蒙特卡洛算法求得最优解，作为企业利率的取值。对于贷款额度，本文将企业月平均流水作为企业贷款期望，同时引入信赖指数作为贷款额度的浮动上下限，并基于多项重要指标，使用**熵权法**分配权重，再通过 TOPSIS **模型**得到合理的企业综合评分，最后选择额度阈值将评分归一化，由此建立信赖指数模型。

针对问题二，本文基于问题一建立的模型，对 302 家无信贷纪录的企业进行信贷风险的量化分析，并给出合理的信贷决策。在受到具体总额度的限制后，本文合理调整信贷策略，对贷款金额进行再分配，由此达到最优的策略模型。

针对问题三，突发因素会即时性地影响各个企业，数值会长期处于混沌状态，针对问题一二而设立的评估模型和策略难以应对突发因素，因此作为应对突发因素的银行人员必须敏锐地觉察到所有企业的状态变化，并以此调整信贷策略达到最优解。本问我们基于已有模型，通过构建抽象影响函数，且将其简化为增长影响系数 γ，再通过国家统计局的数据得到突发因素对不同行业的增长影响系数 γ，代入后求得调整后的最优信贷策略。

关键字： 随机森林　熵权法　TOPSIS　信赖指数

一、 问题重述

1.1 问题背景

中小微企业是由单个或少数人提供资金，通常由业主直接管理的企业，是大众创业，万众创新的重要载体，对国民经济，社会发展以及解决就业等方面具有战略意义。对社会和谐稳定具有极其重要的作用。

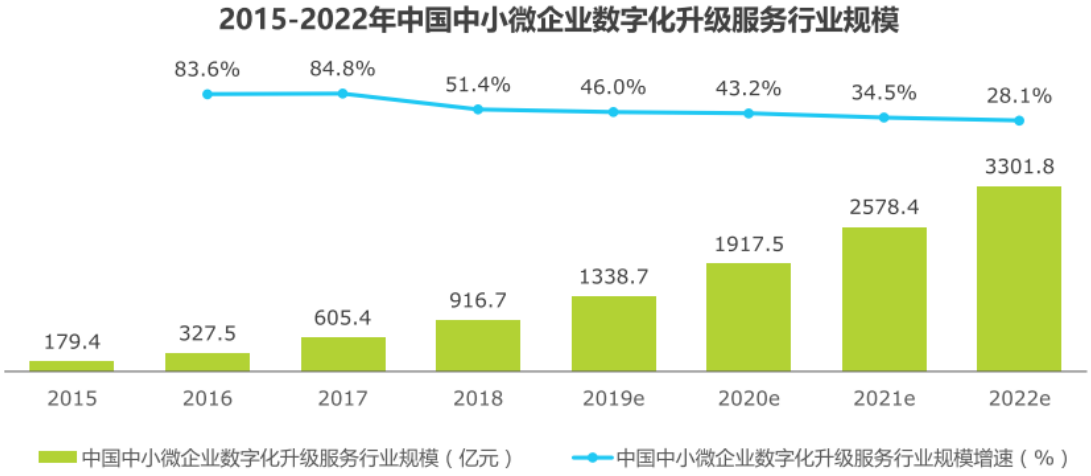


图 **1**　中小微企业数字化升级服务行业规模

但由于中小微企业的底蕴不足，缺少公司担保和抵押财产，以及财务状况不透明等原因，银行只能通过企业的历史交易发票，判断企业的综合实力，发展潜力，财务稳定情况等企业的当前状态，并且以此为依据，判断是否发放贷款以及确定发放贷款额度以及利率。

国别	2016年		2017年		2018年	
	中小企业	总体	中小企业	总体	中小企业	总体
中国	2.60	2.07	2.58	2.05	2.82*	1.87**
美国	1.28	1.57	1.34	1.33	1.41	1.12
巴西	6.7	3.15	5.67	2.99	4.29	2.45
俄罗斯	14.23	6.91	14.93	6.66	12.38	6.51
葡萄牙	17.88	15.85	15.1	13.46	10.38	9.43

数据说明：*为2018年末数据，**为2018年三季度末数据。
数据来源：OECD《中小企业融资2020》。

图 **2**　中小微企业不良信贷

相较于大规模专业化公司，从客观统计规律来看，中小微企业的产生不良信贷行为的可能性更高，加之中小微企业提供材料有限，因此银行会付出更多的人力和时间成本对中小微企业进行综合评估，对于借贷双方都有不必要的时间和成本上的浪费。

若能建立合理的数学模型，通过中小微企业提供的有限信息，判断出企业的综合实力以及信贷风险，从而帮助银行迅速制定针对不同企业的信贷策略，对于银行减少人力资源成本，提高银行运作效率，最大化的规避不良信贷风险具有重要的意义。同时也能提高博弈双方双赢的可能。

1.2 问题要求

基于上述背景我们需要建立数学模型解决以下问题：

(1) 根据附件 1 所提供的 123 家企业数据，分析挖掘出隐藏的指标，并根据已知的信誉评级，探究隐藏指标和信誉评级之间的联系，建立信誉风险量化模型，从而对信誉评级进行量化。根据附件三提供的利率和客户流失率之间的关系，选择合适的函数进行拟合，并求得最优解作为企业的利率。基于影响企业实力的不同因素和量化后的信誉评级建立数学模型，给出最为合理的信贷策略。

(2) 根据附件二给出的 302 家无信贷记录企业的数据，在年度信贷总额为 1 亿元的条件下，根据问题 1 建立的模型，给出针对这些企业的信贷策略。

(3) 考虑突发因素导致的影响，对问题 1 建立的模型进行调整和优化，给出调整后的银行信贷策略

二、 问题分析

本文从银行的角度出发，解决的银行对于中小微企业放贷策略问题，目标是银行利益的最大化，和针对不同企业发放合理的贷款。目前得到的数据是有 123 信贷记录的企业的信誉评级和发票往来，以及 302 家无信贷记录的发票往来，依据这些数据，我们可以挖掘出大量隐藏指标从而对信贷评级进行量化，并结合其他客观因素对企业进行综合评分，建立信贷决策模型。

2.1 对问题一的分析

针对问题一，我们的目标是对信贷风险进行量化评估，并建立合理的信贷决策模型，针对不同企业制定的贷款额度和利率。

本文将从已有数据中挖掘出公司隐藏指标，如公司规模，公司利润增长率，合作依存关系，资金链流动速度，资金链稳定程度等，找到合理的数学模型建立信誉风险量化模型。

根据附件三的数据，利率和客户流失率拥有部分点的对应关系，本文将利用最小二乘法，用多项式进行函数拟合，使用蒙特卡洛法求得利率和客户流失率的极大值，求得三个评价等级下对应的最佳利率。

我们通过查阅资料得知，企业月平均流水一般是企业总资产的五分之一，而原则上企业贷款额度不能超过企业总资产的 25%, 同时我们将引入公司的信赖指数确定贷款额度的浮动上下限。参考多种合理指标，使用熵权法分配权重，最后使用 TOPSIS 算法确立信赖指数评分模型，选择合理阈值后将评分归一化得到每个企业个性的贷款额度浮动上下限。

2.2 对问题二的分析

针对问题二，此小题的主要任务是利用第小问建立的模型，对 302 家无信贷纪录的企业进行信贷风险的量化分析，并进行合理的信贷决策。

本小问和第一小问的主要区别在于，拥有了具体的年度信贷总额的约束，如果资金无法满足所有公司需求，我们将合理调整信贷决策策略，尽可能使得银行利益最大化。

将银行利益作为目标函数，考虑信贷总额的约束条件，即可以优化模型，进而求得银行信贷的最优策略。

2.3 对问题三的分析

突发事件的发生会造成大量因素的变化，如经济衰退，政府补贴等变量，因此要尽可能将所有发生变化的参数纳入到模型之中，以更好解释突发事件影响的的复杂现实客观世界。

三、 模型的假设和符号说明

3.1 模型假设

(1) 假设一：企业和银行都是经济上的绝对理性法人，以自身利益作为决策依据，进行决策。不会采取激进策略。

(2) 假设二：银行贷款决策时只考虑企业的客观经营状况，忽略个人倾向等主观因素。

(3) 假设三：企业流水符合正常的经济规律，正常贷款需求额度为企业月平均流水。

(4) 假设四：银行的贷款金额充足但有限。

3.2 符号说明

表 1　符号说明

符号	含义	单位
n	贷款公司总数	个
r_i	每个企业的贷款银行利率	
W_i	对应利率的客户流失率	
P	银行获得的总收益	元
M_{1i}	每个企业预期贷款额度	元
M_{2i}	每个企业最终贷款额度	元
H_{score}	信誉风险评分	
H_{Rank}	信誉风险评级	
S_i	每个企业综合评分	

四、 问题一模型的建立与求解

4.1 行为主体的博弈构建

假设博弈双方都是在有限理性基础上，以自身利益为依据进行决策的。对中小微企业来说，有守信和失信两种选择，对银行而言有贷款和不贷款两种选择。

假设博弈初始时，银行发放贷款概率为 P，不发放贷款概率为 1-P；中小微企业守信的概率为 q，不守信的概率为 1-q。首先假设中小微企业在银行选择贷款还是不贷款的情况下都能获得基准收益 b，银行在中小微选择守信还是失信情况下都能获得基准收益 a。

双方的支付矩阵如下表所示。

表 2　支付矩阵

	守信	不守信
贷款	a+g+m, b+n	a+g-c, b+n-h
不贷款	a, b	a, b-h

其中 m 是中小微企业贷款给银行带来的收益，h 为中小微企业失信后带来的信誉损失。g 为政府给中小微企业贷款给银行的补贴，c 为中小微企业失信给银行带来的损失。

4.1.1 企业策略的动态演变

当失信时，适应度为:

$$u_{11} = p(b+n) + (1-p)b$$

当守信时, 适应度为:

$$u_{12} = p(b+n-h) + (1-p)(b-h)$$

平均适应度为:

$$u = qu_{11} + (1-q)u_{12}$$

中小微企业的动态方程为:

$$F(q) = \frac{dq}{dt} = q(1-q)(u_{11} - u_{12})$$

对 q 求一阶导数：

$$F^{'}(q) = (1-2q)h$$

$$F^{'}(0) = h > 0$$

$$F^{'}(1) = -h < 0$$

显然，中小微企业的最优行为决策是守信。

4.1.2 银行策略的稳定性和动态演变

当贷款时，适应度为:

$$u_{21} = q(a+g+m) + (1-q)(a+g-c)$$

当不选择贷款时, 适应度为:

$$u_{21} = qa + (1-q)a$$

平均适应度为:

$$u = pu_{21} + (1-p)u_{22}$$

若 $u_{21} > u_2$ 银行选择贷款可带来高于平均水平的收益，银行贷款的概率 p 会随着时间的增加而增加；当 $u_{21} < u_2$ 时，银行选择贷款的概率 p 会随时间的增加而减少。

银行的动态方程为:

$$F(q) = p(1-p)[q(m+c) + g - c]$$

$$F^{'}(q) = (1-2q)[q(m+c)+g-c]$$

当中小微企业守信概率 $q = \frac{c-g}{m+c}$ 时，$F^{'}(1) = 0$, 此时，银行选择贷款或不贷款的策略效果没有太大区别。

当中小微企业守信概率 $q > \frac{c-g}{m+c}$ 时,$F^{'}(1) < 0$, 此时，贷款是银行的演化稳定策略。

当中小微企业守信概率 $q < \frac{c-g}{m+c}$ 时,$F^{'}(1) > 0$, 此时，不选择贷款是银行的演化稳定策略。显然，中小微企业守信概率越高，银行放贷的意愿越强。

综上，中小微企业的最佳策略选择是守信，此时，银行的最佳策略选择是放款。

4.2 隐藏指标挖掘

对企业的信贷风险进行量化评分是信贷策略的重要指标，信贷风险评估主要和公司的经营状况挂钩。本文根据附件一提供发票数据，深度挖掘出反应公司综合实力，经营状况的隐藏指标，每个指标介绍如下：

4.2.1 年利润增长率

符号表示: a_1, a_2, a_3

指标解释：

这三个变量分别代表企业在 2018 年，2019 年,2020 年的平均利润增长率，用于反映企业在未来发展的潜力。

公式计算：

$$a_1 = \frac{profit_{2018}}{profit_{2017}} \quad a_2 = \frac{profit_{2019}}{profit_{2018}} \quad a_3 = \frac{profit_{2020}}{profit_{2019}}$$

其中 $profit_{2017}$ 是 2017 年的企业利润，$profit_{2018}$ 是 2018 年的企业利润，$profit_{2019}$ 是 2019 年的企业利润，$profit_{2020}$ 是 2020 年的企业利润.

4.2.2 企业规模

符号表示：$scale$

指标解释：

此指标代表企业的年平均利润，同时对其取 10 的对数以缩小企业规模之间的巨大客观差距，用于反映企业规模。

公式表示：

$scale = \lg \frac{\sum_{i=2017}^{2020} profit_i}{n}$

其中 $profit_i$ 代表企业的某年的年利润。

4.2.3 进项合作依存关系

符号表示：$cooperate_{input}$

指标解释：

此指标代表公司在进项方面与其最重要的一位长期合作伙伴的密切关系，用其月平均合作次数表示，以此反映长期合作关系上的稳定程度。

公式表示:

$$cooperate_{input} = \frac{frequency_{input}}{month_{input}}$$

其中 fruqency 代表此公司和重要合作伙伴的进货发票次数，month 代表两家公司有过交易行为的月份数。

4.2.4 售项合作依存关系

符号表示：$cooperate_{output}$

指标解释：

此指标代表公司在售项方面与其最重要的一位长期合作伙伴的密切关系，用月平均合作次数表示，用于反映其长期合作关系上的稳定程度。

公式表示:

$$cooperate_{output} = \frac{frequency_{output}}{month_{output}}$$

其中 fruqency 代表此公司和重要合作伙伴的售货发票次数，month 代表两家公司有过交易行为的月份数。

4.2.5 长期合作公司规模

符号表示：$scale_c$

指标解释：

此指标代表其长期合作公司的实力大小，用合作公司的月平均流水表示。其长期合作公司的实力大小，也可反映其公司在合作关系上的稳定程度。

公式表示：

$scale_c = \log \frac{\sum_{i=2017}^{2020} profit_{ci}}{n}$ 其中 $profit_{ci}$ 代表企业的某年的年利润。

4.2.6 发票作废比例

符号表示：$invalid$

指标解释：

此指标代表公司在经营过程中买卖的稳定程度，用作废发票和负数发票之和与有效发票之比表示公司交易过程中的稳定程度。

公式表示：

$$invaild = \frac{num_{invalid}}{num_{valid}}$$

其中 $num_{invalid}$ 是作废发票和负数发票之和，num_{valid} 是有效发票。

4.2.7 资金链稳定程度

符号表示：$stablity$

指标解释：此指标代表公司资金链的稳定程度，用月均开发票次数的方差表示。

公式表示:

$$stablity = \frac{\sum_{i=0}^{n}(y_i - y_{avg})^2}{n}$$

4.2.8 资金链流动速度

符号表示: $speed$

指标解释：

此指标代表资金链的流动速度，用月平均发票金额表示。

公式表示：

$$speed = \frac{\sum_{i=0}^{m} money_i}{month}$$

其中 m 是发票总数，$money_i$ 是每条发票的金额，month 是月份数。

4.2.9 其他指标

由于选取的指标较多，限于篇幅限制，仅对部分复杂指标进行了详细解释和公式说明，其余指标在此项中以表格形式展示。

表 **3**　其他指标

指标	详细解释
进项金额	进项发票的金额总和
销项金额	销项发票的金额总和
进项税额	进项发票的税金总和
销项税额	销项发票的税金总和
进项价税合计	进项金额和进项税额之和
销项价税合计	销项金额和销项税额之和
有效进项发票数	进项发票中有效发票的个数
作废进项发票数	售项发票中有效发票的个数
总发票数	销项发票和进项发票之和
进项平均金额	进项金额除以进项发票个数
售项平均金额	售项金额除以进项发票个数

4.3 信誉风险量化模型

4.3.1 线性回归模型

线性模型是一个通过学习属性的线性组合来进行预测的函数。

$$y = w_0x_0 + w_1x_1 + ... + w_mx_m = w^Tx$$

线性回归可以被看做是样本点的最佳拟合直线。其原理是通过梯度下降法，基于均方误差最小化来求解回归曲线的参数，使得回归曲线到样本点垂直距离（残差或误差）的平方和最小。

通过梯度，我们可以基于代价函数 J(w) 沿梯度方向做一次权重更新:

$$w = w + \Delta w$$

在此，权重增量 Δ w 定义为负梯度和学习速率的乘积:

$$\Delta w = -\eta \Delta J(w) = \eta \sum (y^i - \phi(z^i)x_j^i$$

算法步骤:

1. 初始化模型参数: 学习速率 η , 权重 w；

2. 对每一组训练数据，计算误差，计算代价函数的权重；

3. 更新权重；

4. 重复多次；

最终结果为 R^2 等于-3.93，MSE 等于 6.31，Accuracy 等于 0.225。模型拟合效果较差。

4.3.2 随机森林回归模型

随机森林模型是从决策树模型发展优化而来，是决策树模型的泛化形式。决策树是一种基本的分类器，一般是将特征分为两类，判定样本属于哪个类别的算法，决策树容易出现对训练集的过拟合问题。而一个随机森林模型中有多个决策树，其分类的确定是由多个决策树分类结果的众数决定，可以形象的描述为投票决定。在处理回归问题时，就是把多个决策树的结果进行平均。我们将隐藏指标数据打包为训练数据，将信誉等级量化为连续值后，作为标签向量

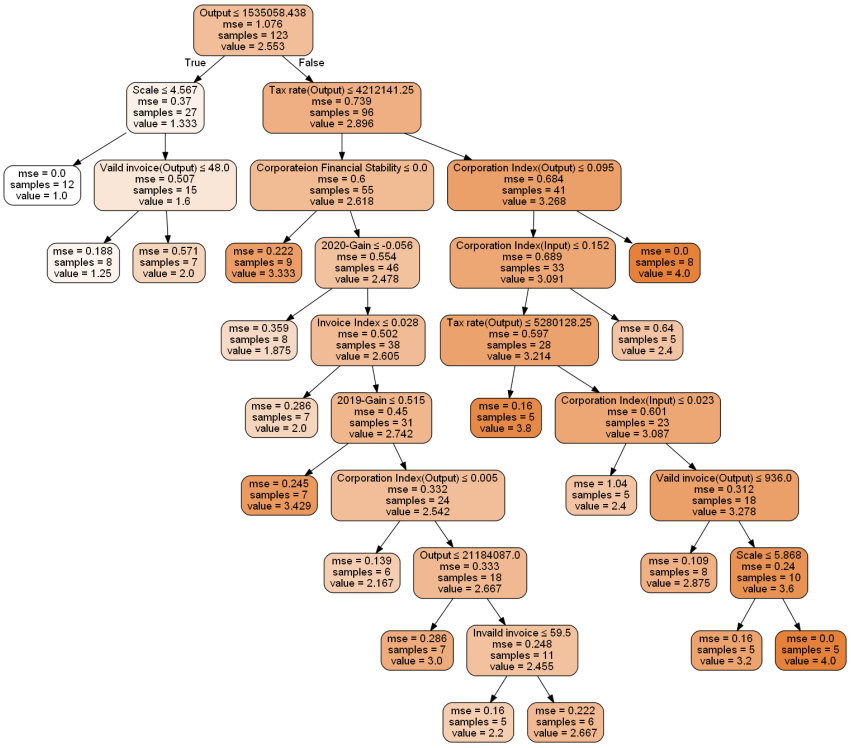


图 **3**　随机森林模型

算法步骤如下：

(1). 从训练集中又放回的随机抽取 $\frac{2}{3}$N 个子样本，之后随机选取 n 个特征进行自主采样作为候选特征。

(2) 再利用子样本数据分别训练，根据决策树信息增益的原则从候选特征中选择每棵树的每个节点，直到所有训练数据子集出现以下三种情形：a. 当前节点的样本数据为同一类；b. 当前特征是空集，或全部样本数据在所有特征上取值相同；c. 节点的样本数据为空。

(3) 重复步骤一和步骤二多次，从而见立 k 颗决策树建立随机森林。

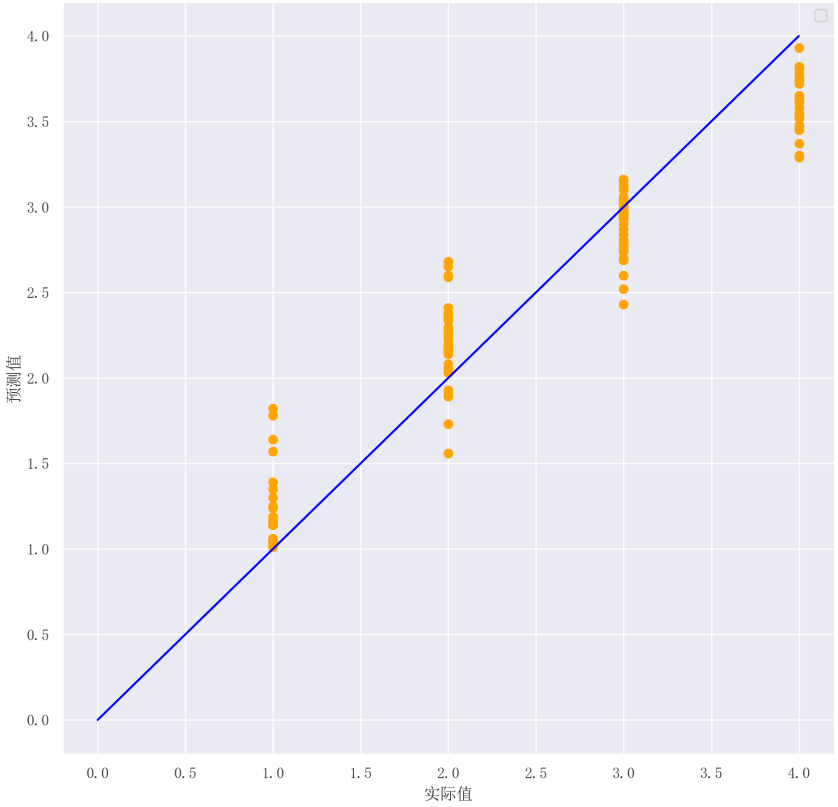


图 **4**　测试集实际结果和预测结果分布图

最终结果为 R^2 等于 0.897，MSE 等于 0.1105，Accuracy 等于 0.872。模型拟合效果较好。

为了排除样本过拟合的情况，我们对测试集进行验证，为减小因样本划分不同而引入的差别，我们采用 k 折交叉验证随机使用不同的划分重复 p 次，最终将 p 次 k 折交叉验证结果的均值作为最终的误差，即进行 pk 次训练/测试。测试集的检验情况如图 13 所示，测试集与预测集的相关系数 Rtest=0.897. 可见我们的模型泛化能力较好，因此我们将随机森林模型运用到后续的测试当中。

表 **4**　模型对比

	R^2	MSE	Accuracy
线性回归模型	-3.93	6.31	0.225
随机森林回归	0.897	0.1105	0.872

由上表可得，随机森林回归模型拟合效果较好，最终选择选择随机森林作为信誉风险量化模型。

4.4 利率的确定和选择

利率的确定和选择是一个相对独立的问题，求解的目标是银行获得最大的利益，即：

$$P = \sum_{i=0}^{n} r_i \times (1 - W_i) \times M_{2i}$$

其中 r_i 是企业利率，W_i 是客户流失率，M_{2i} 是企业最后获得的贷款总额，n 是企业总数，P 是银行获得的总利益。

为了使得 P 最大，我们可以将问题进行拆解，分为求得最大的利益期望和合理的贷款额度。

要求得利益期望的最大值，即求得 $r_i \times (1 - W_i)$ 的最大值，根据附件三给出的客户流失率和利率表格，绘制出两个变量的折线图。

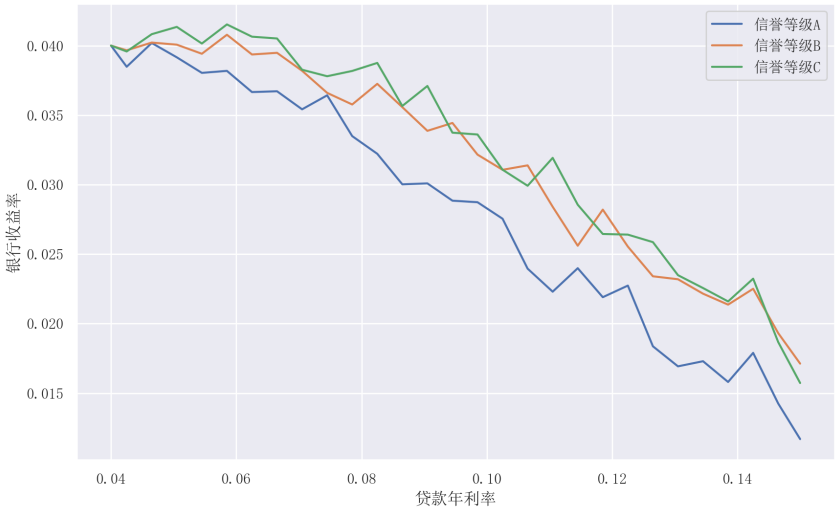


图 **5**　利率和客户流失率折线图

我们可以尝试利用最小二乘法的方法，使用多项式进行函数的拟合。

4.4.1 最小二乘法多项式拟合

我们设利率和客户流失率的函数为:

$$P(x) = a_0 + a_1 x + a_2 x^2 + ... + a_m x^m = \sum_{i=0}^{m} a_i x^i$$

我们的目标是使得误差的平方和最小：

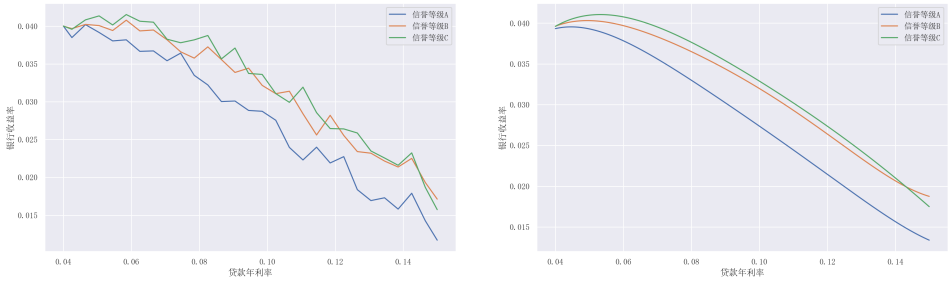
$$I = \sum_{i=0}^{n} [p(x_i) - y_i]^2 = min$$

其中 I 是误差的平方和。

由极值条件可得：

$$\frac{\partial I}{\partial a_j} = 2 \sum_{i=0}^{n} (\sum_{k=0}^{m} a_k x_i^k - y_i) x_i^j = 0, j = 0, 1, 2....m$$

通过简单求解线性方程组的解，可得函数的各项系数。使用 Python 程序进行求解，可以得到函数的拟合曲线及其解析式。



(a) 利率和客户流失率折线图　　**(b)** 利率和客户流失率曲线图

图 **6**　多项式拟合前后对比

信誉评级 A 函数解析式:

$$P(A) = -5095x^4 + 2574x^3 - 519.5x^2 + 52.64x - 1.41$$

信誉评级 B 函数解析式:

$$P(B) = -5963x^4 + 2815x^3 - 530.2x^2 + 51.17x - 1.41$$

信誉评级 C 函数解析式:

$$P(C) = -2202x^4 + 1340x^3 - 320.1x^2 + 38.5x - 1.098$$

函数的评估：

以度量拟合优度的决定系数 R^2 作为函数拟合的评价标准，得到下表：

表 **5**　各信誉评级的 R^2

信誉评级	决定系数 R^2
A	0.997980
B	0.9985847
C	0.998207

因为决定系数越接近 1，拟合效果越好，从上表可以看出，利用最小二乘法的多项式函数拟合效果较好。

4.4.2 目标函数最优解的确定

通过蒙特卡洛法，向 [0.4,0.15] 的区间投入 110000 个点，得到最优解分别是：

信誉评级 A：[0.044599,0.03953877]

信誉评级 B：[0.049379,0.04031107]

信誉评级 C：[0.053225,0.04104998]

综上，我们得出结论，我们给信誉评级为 A 的企业年利率大小为 0.044599，给信誉等级为 B 的企业的年利率为 0.049379，给信誉等级为 C 的企业年利率为 0.053225.

4.5 企业信赖指数评分模型

4.5.1 熵权法

熵值法是计算指标权重的经典算法之一，它是指用来判断某个指标的离散程度的数学方法。离散程度越大，即信息量越大，不确定性就越小，熵也就越小；信息量越小，不确定性越大，熵也越大。根据熵的特性，我们可以通过计算熵值来判断一个事件的随机性及无序程度，也可以用熵值来判断某个指标的离散程度，指标的离散程度越大，该指标对综合评价的影响越大。

我们将已有的指标用一个矩阵 A 表示

$$A = [x_1.....x_m]$$

然后将数据进行归一化处理：

$$x_{ij} = \frac{x_{ij} - min(x_j)}{max(x_j) - min(x_j)}$$

计算第 j 项指标第 i 个所占比重:

$$P_{ij} = \frac{ij}{\sum_1^n x_{ij}}$$

计算第 j 项指标的熵值

$$e_j = -k \times \sum_{1}^{n} P_{ij} \times \log(P_{ij}), k = 1/ln(n)$$

计算第 j 项指标的差异系数

$$g_j = 1 - e_j$$

计算第 j 项指标的权重

$$W_j = \frac{g_j}{\sum_1^m g_j}$$

根据以上步骤得出，每个指标的权重为：

4.5.2 TOPSIS 评分法

其基本原理，是通过检测评价对象与最优解、最劣解的距离来进行排序，若评价对象最靠近最优解同时又最远离最劣解，则为最好；否则为最差。其中最优解的各指标值都达到各评价指标的最优值。最劣解的各指标值都达到各评价指标的最差值。

TOPSIS 法中“理想解”和“负理想解”是 TOPSIS 法的两个基本概念。所谓理想解是一设想的最优的解（方案），它的各个属性值都达到各备选方案中的最好的值；而负理想解是一设想的最劣的解（方案），它的各个属性值都达到各备选方案中的最坏的值。方案排序的规则是把各备选方案与理想解和负理想解做比较，若其中有一个方案最接近理想解，而同时又远离负理想解，则该方案是备选方案中最好的方案。

TOPSIS 模型构建步骤:

对特征矩阵进行规范化处理, 得到规格化向量 r_{ij} , 建立关于规格化向量 r_{ij} 的规范化矩阵

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$$

通过计算权重规格化值 vij, 建立关于权重规范化值 vij 的权重规范化矩阵

$$v_{ij} = w_i r_{ij}, j = 1, 2, 3, ..., n$$

确定理想解和反理想解:

$$A^* = (max_i v_{ij} | j \in J_1), (min_i v_{ij} | j \in J_2)$$

$$A^- = (max_i v_{ij} | j \in J_2), (min_i v_{ij} | j \in J_1)$$

其中,J_1 是收益性指标集, 表示在第 i 个指标上的最优值; J_2 是损耗性指标集, 表示在第 i 个指标上的最劣值。收益性指标越大, 对评估结果越有利; 损耗性指标越小, 对评估结果越有利。反之, 则对评估结果不利.

计算距离尺度, 即计算每个目标到理想解和反理想解的距离, 距离尺度可以通过 n 维欧几里得距离来计算。

$$S^* = \sqrt{\sum_{j=1}^{n}(V_{ij} - v_j^*)^2}$$

$$S^- = \sqrt{\sum_{j=1}^{n}(V_{ij} - v_j^-)^2}$$

其中，v_j^* 与 v_j^- 分别为第 j 个目标到最优目标及最劣目标的距离，v_{ij} 是第 i 个目标第 j 个评价指标的权重规格化值。S * 为各评价目标与最优目标的接近程度，S * 值越小, 评价目标距离理想目标越近，方案越优。

计算理想解的贴近度 C *

$$C_i^* = \frac{S_i^-}{(S_i^* + S_i^-)}$$

其中，$0 <= C_i^* <= 1$

当 $C_i^* = 0$ 时，该目标为最劣目标，当 $C_i^* = 1$ 时，该目标为最优目标

接着，我们将得到的分数归一化到 [0.8,1.2] 的区间内

$$k = \frac{max - min}{C_{max} - C_{min}}$$

$$S_i = min + k \times (C_i - C_{min})$$

其中 max 和 min 是区间的上下限。最后得出的 S_i 为每个公司的信赖指数，即为贷款的上下限浮动。

$$M_{2i} = M_{1i} \times S_i$$

最终得到的 M_{2i} 即为公司最终的贷款额度。

4.6 综合信贷决策模型

4.6.1 最终模型的确定

综合以上模型和算法，我们可以将随机森林回归模型，利率求解模型，企业信赖指数评分模型进行结构化的规整和组合，得到最终的基于 TOPSIS 的熵权及随机森林法银行信贷决策模型。流程图如下所示。

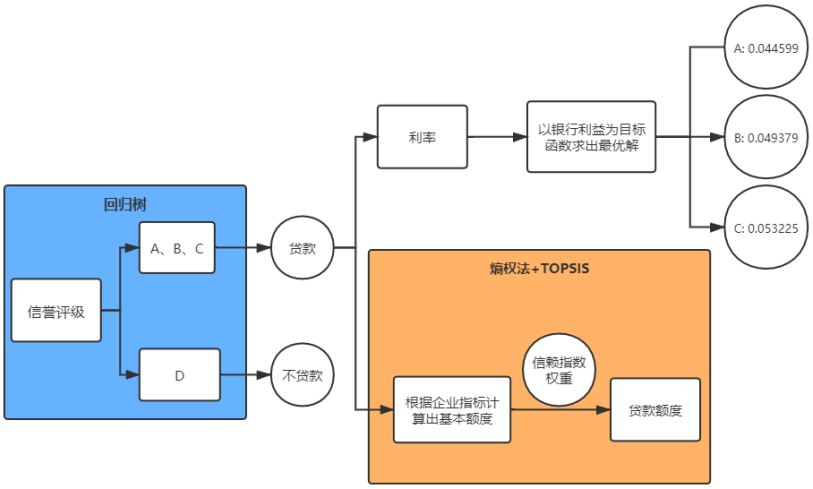


图 7　流程图

该模型的基本流程如下所示:

1. 使用随机森林模型对企业信贷风险进行量化分析，并离散为 A，B，C，D 四个等级，对 D 等级不予贷款，A，B，C 三个等级进一步分析。

2. 根据利率求解模型的结果，给予 A，B，C 三个等级的企业不同的利率优惠。

3. 计算企业的月平均流水作为企业借贷的基本额度。

4. 利用熵权法和 TOPSIS 法求得企业的信赖指数，并依据信赖额度计算贷款额度的浮动程度。

4.6.2 信贷策略结果

根据附件 1 中的数据，将其投入最终模型，可得：

表 **6**　信贷决策部分结果

企业代号	企业名称	企业信赖指数	企业信誉评分	贷款额度
E15	*** 劳务有限公司	0.95	3.48	1000000.0
E16	*** 建筑劳务有限公司	0.95	3.46	169082.0
E37	*** 木业有限公司	0.97	3.1	1000000.0
E38	*** 建设工程有限公司	0.95	3.1	1000000.0
E39	*** 建筑劳务有限公司	0.95	2.81	156277.0
...	...	...	...	...
E50	*** 建筑劳务有限公司	0.95	2.4	926595.0
E109	*** 服饰有限公司	0.95	1.56	0.0
E110	*** 通讯器材经营部	0.96	1.54	0.0
E111	*** 科技有限公司	0.96	1.41	0.0
E112	*** 机械设备有限公司	0.96	1.61	129345.0

备注: 受篇幅限制，完整结果放置于支撑材料。

五、 问题二的求解

问题二相较于问题一，有了具体额度的限制，因此会出现银行贷款资金无法全部满足所有企业需求的情况，此时，我们就要进行贷款额度的内部调整来对模型进行优化。

在阅读相关文献后，我们得知，若企业贷款远高于企业预期贷款，则企业不良信贷率会大大提升。同样，若企业贷款远低于企业预期贷款，则贷款对于企业的发展无法产生有效的影响，即贷款价值较低。

此时若我们将这些影响能力有限的贷款调配到无法得到贷款的企业中，则产生的影响会优于原始状态。

此时我们设最大阈值和最小阈值分别设为 F_{max} 和 F_{min}，覆盖率为 $Cover$，n_{cover} 为被覆盖企业个数，$n_{uncover}$ 为未被覆盖企业个数，f 为企业贷款额度。

目标函数:

$$z = max\ Cover = max\ \frac{n_{cover}}{n_{uncover}}$$

限制条件:

$$\begin{cases} F_{max} > 100 \\ F_{min} < 10 \\ n_{cover}(F_{min} < f < F_{max}) \\ n_{uncover}(f < F_{min}或f > F_{max}) \end{cases} \tag{1}$$

使用线性规划算法，求得最优解为: $F_{max} = 701.4$ 万元,$F_{min} = 5628.4$ 元

此时我们将贷款额度高于 701.4 万的企业，低于 5628.4 元的企业的贷款申请驳回，多余贷款分配至排队序列，使得贷款价值最大化。基于优化后的模型，我们将附件二的数据输入，即可得到最终结果。

表 **7**　问题二决策结果的部分展示

企业代号	企业名称	企业信赖指数	企业信誉评分	贷款额度
E338	*** 商贸有限公司	0.87	1.71	0.0
E339	*** 居益卫浴家俪厂	0.87	1.57	357611.0
E340	*** 物流有限公司	0.87	1.9	0.0
E341	*** 纺织品有限公司	0.87	1.77	240225.0
E342	*** 裕华机械厂	0.87	1.62	313826.0
...	...	...	...	...
E348	*** 酒店管理有限公司	0.89	1.67	118528.0
E349	*** 营销策划广告有限公司	0.87	2.35	0.0
E350	*** 演艺设备有限公司	0.87	1.55	0.0
E351	*** 文化传播有限公司	0.87	1.71	0.0
E352	*** 居装饰工程有限公司	0.86	1.73	1000000.0

六、 问题三的求解

题目认为突发因素将会对中小微企业的生产效益以及资金总规模产生不同程度的影响，且同一突发因素通常来说对于不同领域与不同类别的中小微企业所产生的影响也

不尽相同。对于问题三的分析将以新冠疫情为例，着重围绕新冠病毒疫情对中小微企业的影响，重新调整我们的信贷策略模型。

在这次新冠疫情中，突发的新冠病毒造成诸多工厂的停工停产，同时也激发医疗器械药品的极大需求。因此相同的突发因素会为不同领域，不同类别，亦或是不同规模的中小微企业带来具有个性化的影响。新冠疫情带来的影响是多方面的，为了描述简化多个影响因子，本文使用如下公式进行概括。

第 i 家企业受到的冲击：

$$\hat{F}_i = \alpha(Enterprise_i) \cdot F_i$$

其中 α 为增长影响函数，$\hat{F}_i$ 为受到影响企业特征，F_i 为受到影响前的企业特征，$Enterprise_i$ 为企业属性。

考虑到因为不同种类、经营领域的企业在新冠病毒疫情第一时段受到的冲击是最直接、最为主要的。因此为了简化模型，我们可以将 $\alpha(Enterprise_{pro})$ 简化为一个影响系数 γ。

即：

$$\hat{F}_i = \gamma_i \cdot F_i$$

其中系数 γ 我们可以通过查询国家的统计数据得到，从国家统计局所发布的统计数据中，我们可以得到如下产业类型的同比增长率：

表 **8**　疫情影响下的各行业同比往年增长率

产业类型	同比往年增长率（γ%）
第一产业（农业）	7.7
第二产业（工业）	-7.4
第三产业（服务业）	0.8
农林牧渔业	9.2
采矿业	-11.9
制造业	-10.2
食品制造业	-9.5
纺织业	-17.4
化学原料和化学制品制造业	-13.6
医药制造业	14.7
有色金属冶炼和压延加工业	-6.5
金属制品业	-15.5
通用设备制造业	-19.6
专用设备制造业	-11.5
汽车制造业	19.9
卫生和社会工作	17.0
电气机械和器材制造业	-14.1
计算机、通信和其他电子设备制造业	10.7
电力、热力、燃气及水生产和供应业	18.0
交通运输、仓储和邮政业	0.9
教育	13.5

我们可以直接将国家统计局中得到的同比往年增长指数作为我们调整企业特征的 γ 指数，最终作用于我们的信贷策略中，从而基于经过调整的信贷策略得到新的企业信用

评级。

表 **9**　医疗行业与建筑行业对比

企业代号	企业名称	冲击前的信誉评级	冲击后的信誉评级
医疗生命科技类企业	-	-	-
E195	*** 医药有限公司	2.76(B)	3.08(B)
E197	*** 医疗设备有限公司	2.90(B)	3.44(B)
E251	*** 医疗器械有限责任公司	2.00(C)	2.72(B)
E261	*** 医疗器械有限公司	1.95(C)	2.71(B)
E379	*** 药业有限公司	1.29(D)	1.33(D)
E398	*** 医疗管理咨询有限公司	1.40(D)	1.46(D)
E420	*** 康药房	1.66(C)	2.11(C)
建筑工程类企业	-	-	-
E135	*** 建设工程有限公司	2.44(C)	2.63(B)
E137	*** 建设工程有限公司	2.22(C)	2.17(C)
E390	*** 建筑工程有限公司	1.61(C)	1.44(D)
E258	*** 建材有限公司	1.60(C)	1.43(D)

由上表可知，医疗生命科技类企业在疫情中因为企业参数的提升，其信誉评级因此提升。而建筑工程类企业则受新冠疫情的影响，导致信誉评级的下降。

由此可见，我们的模型准确地判断了因为企业各项特征增长以及蜕缩而引起的信誉评级改变，表明本文所建立的模型，在面对新冠疫情这一突发因素，有着良好的自我调节性。

七、 模型评价

7.1 优点

1. 对附件的数据进行了深度的挖掘，获得了许多可以反映企业客观情况的隐藏指标，如合作依存关系，长期合作企业的实力，资金链稳定程度，都是具有创新性的指标，

用这些指标构建的随机森林回归模型，能够更加接近客观真实地评估信贷风险。

2. 使用熵权法和 TOPSIS 排序法的企业信赖指数模型，进行了去规模化处理，尽可能消除了企业规模大小对评分的影响，从而提高了模型的准确度。

3. 对于突发因素的处理，本文建立的模型能够很好地兼容突发因素引起的诸多变化，并尽可能将所有产生变化的影响因子纳入模型中，最终给出合理的信贷调整策略。

7.2 模型缺点

1. 对数据的挖掘仍然不够深入，选取的指标不够丰富，部分指标缺乏可解释性。

2. 模型稍显简单，对相对复杂的客观世界进行了部分简化，无法适应更复杂的情况。

3. 模型过于死板，没有考虑加入适量的主观因素，无法学习不同的主观偏向进行个性化决策。

7.3 未来计划

1. 考虑引入更多隐藏指标，增强模型的适应性和可解释性。

2. 尝试引入可以主观改变的参数，增强模型的个性化能力。

参考文献

[1] 苏蕙. 小微企业信贷风险评价模型构建和应用. 经济天地.2019

[2] 谢丽辉. 疫情冲击下商业银行小微企业贷款风险防控策略分析. 经济研究.2020.7

[3] 马腾跃. 对中小微企业贷款延期还本付息加大小微企业信用贷款支持力度. 新闻直播

[4] 王刚. 王彦伟. 尚博文. 新冠肺炎疫情对银行业的影响及政策优化建议.《中国银行业》.2020 年第五期

[5] 李善民. 信用评定、银行贷款决策与农户贷款可得性研究 [A]. 征信.2018 年第 12 期

[6] 国家统计局：2020 年 1—7 月全国固定资产投资下降 1.6%.http://www.xinhuanet.com/fortune/2020-08/14/c_1126367246.htm

八、 附件

部分代码:

```
data1 = pd.read_csv('credit_history_enterprise_information.csv')
data2 = pd.read_csv('credit_history_input_invoice.csv')
```

```
data3 = pd.read_csv('credit_hidata1['信誉评级'].replace(['A', 'B', 'C', 'D'], [4.0, 3
data1['是否违约'].replace(['是', '否'], [1, 0], inplace = True)
data1.head()story_output_invoice.csv')
data2['parse_time'] = pd.to_datetime(data2['开票日期'], format = '%Y/%m/%d')
data2.head()
data3['parse_time'] = pd.to_datetime(data3['开票日期'], format = '%Y/%m/%d')
data3.head()
data1['企业代号'].unique()
suming = data2.groupby(['企业代号']).sum()['金额']
print(suming.sort_values(ascending = False))
suming = data3.groupby(['企业代号']).sum()['金额']
print(suming.sort_values(ascending = False))
data2['year'] = data2['parse_time'].dt.year
data2.groupby(['企业代号', 'year']).mean()
data3['year'] = data3['parse_time'].dt.year
data3.groupby(['企业代号']).count().sort_values(by = 'year', ascending = False)
data2['year'].unique()
data2[data2['year'] == 2020]

enterprise_interest = []

for i in data1['企业代号'].unique() :

temp = [i]

enterprise_data2 = data2[data2['企业代号'] == i]

enterprise_data3 = data3[data3['企业代号'] == i]

normalization = enterprise_data2['金额'].sum() - enterprise_data3['金额'].sum()

for j in [2016, 2017, 2018, 2019, 2020] :

mths = len(enterprise_data2[enterprise_data2['year'] == j]['parse_time'].dt.month.uni
```

```
meaning = enterprise_data2[enterprise_data2['year'] == j]['金额'].sum() - enterprise_
if(mths) : meaning /= mths
meaning = 0 if meaning is np.nan else meaning
temp.append(meaning)
scale = np.mean(np.abs(np.array(temp[1:]))[np.array(temp[1:]) != 0.0])
temp.append(np.log10(np.mean(scale)))
temp.append(data1[data1['企业代号'] == i]['购方合作密切指数'].sum() / normalization)
temp.append(data1[data1['企业代号'] == i]['销方合作密切指数'].sum() / normalization)
temp.append(data1[data1['企业代号'] == i]['作废发票/有效发票'].sum())
temp.append(data1[data1['企业代号'] == i]['月交易频率'].sum() / normalization)
temp.append(data1[data1['企业代号'] == i]['企业资金链稳定程度'].sum())
temp.append(data1[data1['企业代号'] == i]['进项金额'].sum() / normalization)
temp.append(data1[data1['企业代号'] == i]['销项金额'].sum() / normalization)
temp.append(data1[data1['企业代号'] == i]['进项税额'].sum() / normalization)
temp.append(data1[data1['企业代号'] == i]['销项税额'].sum() / normalization)
temp.append(data1[data1['企业代号'] == i]['进项价税合计'].sum() / normalization)
temp.append(data1[data1['企业代号'] == i]['有效进项发票数'].sum() / normalization)
temp.append(data1[data1['企业代号'] == i]['有效销项发票数'].sum() / normalization)
```

```
temp.append(data1[data1['企业代号'] == i]['作废进项发票数'].sum() / normalization)

enterprise_interest.append(np.array(temp))

print(enterprise_interest)

enterprise_interest_gain = []

for i in enterprise_interest :

temp = []

for j in range(2, 6) :
if float(i[j - 1]) == 0 :
temp.append(0.0)
continue
temp.append(float(i[j]) / float(i[j - 1]))

for k in range(6, len(i)) :

temp.append(float(i[k]))

print(temp)

enterprise_interest_gain.append(temp)

enterprise_interest_gain = np.array(enterprise_interest_gain)

for i in range(0, len(enterprise_interest_gain[0])) :

enterprise_interest_gain[:, i][enterprise_interest_gain[:, i] == 0.0] = np.median(ent

for i in enterprise_interest_gain :
print(i)
```

```
train_feat = []

feat_name = ['2017-Gain','2018-Gain','2019-Gain','2020-Gain', 'Scale', 'Corporation I
'Trade Frequency(per month)', 'Corporateion Financial Stability', 'Input', 'Output',

for i in range(0, len(np.array(enterprise_interest)[:, 0])) :

temp = np.zeros(shape = len(feat_name) + 1)

temp[:len(feat_name)] = enterprise_interest_gain[i]

temp[-1] = data1[data1['企业代号'] == np.array(enterprise_interest)[i, 0]]['信誉评级']

train_feat.append(np.array(temp))

train_feat = np.array(train_feat)

print(train_feat)
```

```
import pandas as pd
import numpy as np
import re


def entropy_weight(X):
"""

Entropy method

Args:
X: Features

Returns:
weight: A buffered writable file descriptor
"""
```

```
X = np.array(X)
P = X / X.sum(axis=0)  # 归一化
E = np.nansum(-P * np.log(P) / np.log(len(X)), axis=0)  # 计算熵值
weight = (1 - E) / (1 - E).sum()

return weight  # 计算权系数


def clean(string):
return string.replace('.csv', '')


def topsis(data, weight=None):
# 归一化
data = data / np.sqrt((data ** 2).sum())

# 最优最劣方案
Z = pd.DataFrame([data.min(), data.max()], index=['负理想解', '正理想解'])

# 距离
weight = entropy_weight(data) if weight is None else np.array(weight)
Result = data.copy()
Result['正理想解'] = np.sqrt(((data - Z.loc['正理想解']) ** 2 * weight).sum(axis=1))
Result['负理想解'] = np.sqrt(((data - Z.loc['负理想解']) ** 2 * weight).sum(axis=1))

# 综合得分指数
Result['综合得分指数'] = Result['负理想解'] / (Result['负理想解'] + Result['正理想解']
Result['排序'] = Result.rank(ascending=False)['综合得分指数']

return Result, Z, weight


def count_months(bill_date_list):
count_list = []
for bill_date in bill_date_list:
```

```
count_list.append(re.findall(r'^\d{4}.\d{1,2}', bill_date)[0])

return len(list(set(count_list)))


def NormMinandMax(npdarr, min=0, max=1):
arr = npdarr.flatten()
Ymax = np.max(arr)  # 计算最大值
Ymin = np.min(arr)  # 计算最小值
k = (max - min) / (Ymax - Ymin)
last = min + k * (arr - Ymin)

return last


df = pd.read_csv('../C/credit_history_enterprise_information.csv')

df = df.iloc[:, [4, 5, 6, 7, 8]]

CHII_list = ['E{}.csv'.format(str(codename)) for codename in range(1, 124)]

amount_list = []
for each_enterprise in CHII_list:
df_input = pd.read_csv('../C/credit_history_enterprise_input_invoice/{}'.format(each_
df_output = pd.read_csv('../C/credit_history_enterprise_output_invoice/{}'.format(eac
billing_date = list(df_input['开票日期'].values) + list(df_output['开票日期'].values)
amount_list.append((sum(df_output['价税合计'].values) - sum(df_input['价税合计'].value

list_ = []
for i in range(123):
list_.append(df.iloc[i, :].values / amount_list[i])


data = pd.DataFrame(list_, columns=['作废发票/有效发票', '购方合作密切指数', '销方合作
raw_score = topsis(data)[0]['综合得分指数'].values.reshape(-1, 1)
```

```
for index, i in enumerate(topsis(data)[0]['综合得分指数'].values.reshape(-1, 1)):
if i == max(topsis(data)[0]['综合得分指数'].values.reshape(-1, 1)):
raw_score[index] = np.mean(raw_score)
if i == min(topsis(data)[0]['综合得分指数'].values.reshape(-1, 1)):
raw_score[index] = np.mean(raw_score)

result = pd.DataFrame(NormMinandMax(raw_score, min=0.8, max=1.2), columns=['企业信赖指

result['企业代号'] = list(map(clean, CHII_list))

result.to_csv('credit_hitsoty_corporate_trust_degree.csv', index=False, encoding='utf
```