中小微企业的信贷决策问题分析

摘要

当前中小微企业占我国企业比重日益增大，贡献了 60% 的 GDP，解决了 80% 的就业，提供了 50% 的纳税额，但与此同时中小微企业所获得的信贷支持却十分有限，因此从银行角度出发，建立合理的、有助于推进中小企业信贷业务的信贷决策模型意义重大。

针对问题一，本题查阅文献得到当前业内评估信贷风险的主要指标，结合附件一所给数据，决定以财务状况、企业增值税率、上下游企业供求关系、企业信誉四类指标来表征企业的信贷风险。由于表征四类指标的相关特征量并不单一，因此本题对各特征量进行了相关性分析、k—均值聚类、离散化等操作，最终得到了数据质量较好的指标信息。考虑到各指标量级不同，因此本题对各指标进行 [0,1] 区间处理，并最终通过对各指标进行主观权重赋值，运用层次分析法得到了各企业的量化风险模型，求解后发现 $E83$ 企业风险最低，$E103$ 和 $E108$ 企业风险最高。由于题目未给出银行年度信贷总额，因此在模型一中假设银行信贷总额满足附件一中所有企业贷款需求。对于银行信贷决策问题，本题通过构建企业信贷额度模型 (保证银行风险较小) 和基于潜在客户流失率下的贷款利率规划模型 (保证银行收益较大) 来求解银行对企业的信贷策略。对于贷款额度，首先根据企业财务状况灰色预测 2020 年的企业财务情况，以此作为企业理论最大信贷金额的参考依据。同时，为保证银行放贷过程中风险较小，本题设定企业实际信贷额度由信贷风险评分和理论最大信贷金额共同决定。贷款利率规划模型则是根据企业风险评分、企业贷款额度和潜在客户流失率来确定。对于贷款利率，结合实际情况，本题认为银行利率种类一般不超过七种。通过对企业信贷风险评分的 k—均值聚类得到企业理想分类为三类，因此以三类作为银行最佳利率种类对规划模型进行求解，最终得到银行最多获利为 1886384.8 元，最佳利率组合为 [0.0505, 0.0585, 0.0665]，相应的企业贷款额度和利率组合详见附录 2.1。

争对问题二，本题首先分析了附件一二数据的主要差异，对于附件二内缺失的企业信誉等级，本题首先提取附件一企业的财务信息和信誉评级数据，利用决策树算法，解得企业信誉评级与财务信息之间的决策树关系模型。补全附件二企业信誉等级数据后，本题在模型一贷款利率规划模型上，增加了年度贷款额度 1 亿的约束条件，求解得到银行最多获利为 6651140.9 元，累计贷款总额为 97162110.1 元，最佳利率组合为 [0.0625, 0.0665, 0.0825]，相应的企业贷款额度和利率组合详见附录 2.2。

针对问题三，本题在问题二的基础上添加突发事件对各企业的影响这一评价指标，从银行角度对每个企业的借贷额度和利率作出调整。本题以新型冠状肺炎作为为突发事件举例展开研究，通过用五个具体的数字 (-0.5,-0.25,0,0.25,0.5) 量化各行业受疫情影响的正负性质和影响程度，再根据实际情况和银行自身情况对这一评价指标与问题二对各企业的评分评估分数给与合适权重得出最终决定各企业信贷额度和利率的合计分数，同样使用聚类对其进行等级划分，求解得到银行最多获利为 5994629.4 元，累计贷款总额为 96805944.5 元，最佳利率组合为 [0.0505, 0.0585, 0.0625, 0.0665, 0.0825]，相应的企业贷款额度和利率组合详见附录 2.3。

关键字： 银行信贷　规划模型　决策树　层次分析法

一、 问题重述

1.1 问题背景

民营中小企业占我国企业比重日益增大，贡献了 60% 的 GDP，解决了 80% 的就业，提供了 50% 的纳税额。但与此同时其所获得的信贷支持只有 20%，从未有过银行信贷的小企业高达 90%。由此可见，中小企业的借贷需求是远远没有满足的。

近年来，应国家号召，各地方银行纷纷加大了用于中小型企业信贷投资的金额比例，我国中小企业信贷规模不断加大。由于中小型企业发展规模限制，在实际信贷审核中，银行往往通过当前信贷政策、企业的交易发票数据信息以及上下游企业影响力情况对一些自身实力较强且运营稳定的企业提供贷款并在利率方面给予适当优惠。

现已知某银行对贷款企业进行一年短期贷服务的额度为 10 万-100 万，年利率为 4%-15% 。附件 1-3 分别提供了 123 家有信贷记录企业的相关数据、302 家无信贷记录企业的相关数据和贷款利率与客户流失率关系的 2019 年统计数据。请根据微小企业的实力、信誉及风险评估情况，从银行角度出发，给出银行是否放贷及贷款额度、利率和期限等信贷策略。

1.2 问题描述

对上述背景进行量化建模，本文对以下问题做出解答：

问题 1：量化分析附件 1 中各个企业的信贷风险，并依此给出在年信贷总额一定时，银行对这些企业的信贷策略。

问题 2：在问题 1 的基础上，量化分析附件 2 中各个企业的信贷风险，并依此给出在年信贷总额为 1 亿元时，银行对这些企业的信贷策略。

问题 3：综合附件 2 中各个企业的信贷风险和可能的突发因素影响，给出在年信贷总额为 1 亿元时，银行对这些企业的信贷调整策略。

二、 问题分析

2.1 问题一的分析

问题一要求建立关于企业信贷风险的评估模型，并给出对于附件一 123 家企业的贷款策略。

对于信贷风险评估模型，本文首先通过查阅行业文献，得到了用来体现信贷风险的多个特征量，结合附件数据，考虑以财务状况、企业增值税率、上下游企业供求关系、企业信誉四类指标来表征企业的信贷风险。由于表征企业财务状况和上下游供求关系的特征量并不单一，因此仍需对各特征量进行进一步分析。考虑到各类指标量级不同且实际银行贷款主要参照各指标的评级情况，本文考虑对各指标 (或相关表征特征量) 进行 k—均值聚类，将其转化为离散化指标，以便后续运算。

由于附件仅提供了企业基本面数据，且当前业内主要是通过专家调查法来分析放贷策略，因此本文考虑通过层次分析法来构建四类指标间的关系，最终得到 [0,1] 区间上的风险评分，值得注意的是由于四类指标的数据特性，最终得到的评分越高意味着企业风险越低。

对于短期放贷策略，主要考虑是否放贷、贷款金额、贷款利率。对于是否放贷，我们认为在问题一中银行年度信贷总额可以满足附件一所有企业在该银行的贷款需求，则从银行利润最大化的角度出发，本文考虑在降低风险的前提下对所有信誉等级高于 D 的企业给予信贷服务。

关于贷款金额和贷款利率，若对其建立在风险最低前提下的利润最高的多目标规划模型，则可能存在模型较难求解的情况。因此本文考虑将贷款金额和利率分成两个模型依次考虑。通过贷款额度模型保证银行放贷风险较小，通过利率模型保证银行在对应贷款额度模型的利润最高。

对于贷款金额，首先可以根据企业财务状况灰色预测 2020 年的经济情况，以此作为企业理论最大信贷金额的参考依据。同时，考虑到短期贷款风险较大，为保证银行放贷过程中风险较小，本文考虑企业实际信贷额度由信贷风险评分和理论最大信贷金额共同决定。结合附件三数据可知，当利率升高时，银行潜在客户流失率也将升高。因此在得到企业实际信贷额度后，本文考虑对贷款利率建立利润最大的目标规划模型。

2.2 问题二的分析

问题二要求在问题一模型的基础上，对附件二中 302 家无信誉记录的企业进行信贷风险的量化分析，并给出当银行年度信贷总额为 1 亿元时的信贷策略。

由于企业信誉是银行放贷中需要考虑的主要因素，因此对于信誉为 D 的企业，银行将不予贷款，而对于信誉较好的企业，往往能够获得较高的贷款额度和较低的贷款利率。因此本文假设企业的信誉情况能够通过其在附件中的财务信息得到较好的体现。为了确定企业的财务信息与企业信誉评分的关系，本文首先提取附件一企业的财务信息和信誉评级数据，利用决策树算法，解得企业信誉评级与财务信息之间的决策树关系模型。将其移植到附录二的数据中，获得附录二企业的信誉评级，从而使模型一的风险评估模型适用于问题二。

相比于问题一，问题二规定了银行年度信贷总额为 1 亿元，因此需要考虑贷款额度是否会用尽的情况。当附件二企业的贷款需求大于 1 亿元时，银行从风险角度考虑，会首先贷给风险评分较好的企业，因此对于一些风险评分较差的企业，虽然其信誉高于 D，但是仍然有可能无法获得银行贷款。对此需要建立一定贷款排序下的企业贷款额度模型。对于贷款利率，本题仍考虑建立与模型一相同的规划模型。

2.3 问题三的分析

问题三要求考虑突发性因素对于银行信贷策略的影响，对此本文考虑以新冠疫情为例，分析疫情下各行业发展情况，对各行业进行抗疫能力分类，得到引入突发性因素下的风险评估模型，并基于银行风险最低建立企业贷款额度模型。此外，考虑到疫情下国家经济发展受影响，政府呼语广大银行降低贷款门槛，为中小型企业提供贷款福利的大背景下，本文在已保证风险较小的情况下，对受疫情影响明显的企业进行适当的利率优惠，在此基础上建立最大利润的目标规划模型，从而得在到疫情影响下的响应政府号召、放贷风险较低、利润较大的信贷调整策略。

三、 模型假设

1. 假设银行对所有企业收集的信息是真实可用的；
2. 假设企业在贷款期间其利率保持稳定不变；
3. 假设银行的信贷时间为一年且中途不可部分还款，保证企业风险评估以年为单位；
4. 假设本题的风险评估针对企业一年还款能力，不考虑延迟若干年后还款情况；
5. 假设银行信贷需考虑信贷目的是否存在因社会需要导致的政策倾斜；

四、 符号说明

符号	意义
S_i	企业 i 年度总流水
I_i	企业 i 进项金额
O_i	企业 i 销项金额
t_i	第 i 个企业的企业规模分数
$p_(i,2018)$	企业 i 在 2018 年的净利润
θ_i	企业 i 发展潜力分数
G_i	企业 i 的财务影响分数
δ_i	企业 i 税收
$GU\&GD$	上游 & 下游企业稳定系数
ω_i	企业 i 供求关系分数
L_i	企业 i 信贷风险模型评分
M_i	2020 年企业 i 理论最大贷款额度
P_{2020}	2020 年企业预测营收利润
I_{2020}	2020 年企业预测进款额度
β_i	企业 i 信贷额度折扣率
R_i	企业 i 实际贷款额度
MON	银行总收益
l_i	客户流失率
u_i	企业 i 利率
E	企业风险评估序列
X_i	企业 i 疫情风险评分
Y_i	引入疫情风险评分 X_i 后的企业 i 信贷额度折扣率
N_i	引入疫情风险评分 X_i 后的企业 i 信贷利率系数

五、 模型一的建立与求解

5.1 问题一模型的准备

信贷风险影响因素

银行贷款收益取决于贷款利率的高低，同样也受到不良贷款的影响。不良贷款率越低，银行收益越高。查阅文献，本文得到 2013 年到 2018 年建设银行、农业银行、渤海银行的不良贷款率，发现短期贷款的不良贷款率明显高于中长期贷款。

由于短期贷款的资金周期更短，对应相应投资的效益要求更高，其风险也越大。对于中小型企业的风险评估，我们通过查阅资料[1]，得到银行判断贷款可行性的主要因素，基于附件所提供数据，本文将以财务状况、企业增值税率、上下游企业供求关系、企业信誉作为风险评估的参考指标。

5.2 问题一模型的建立

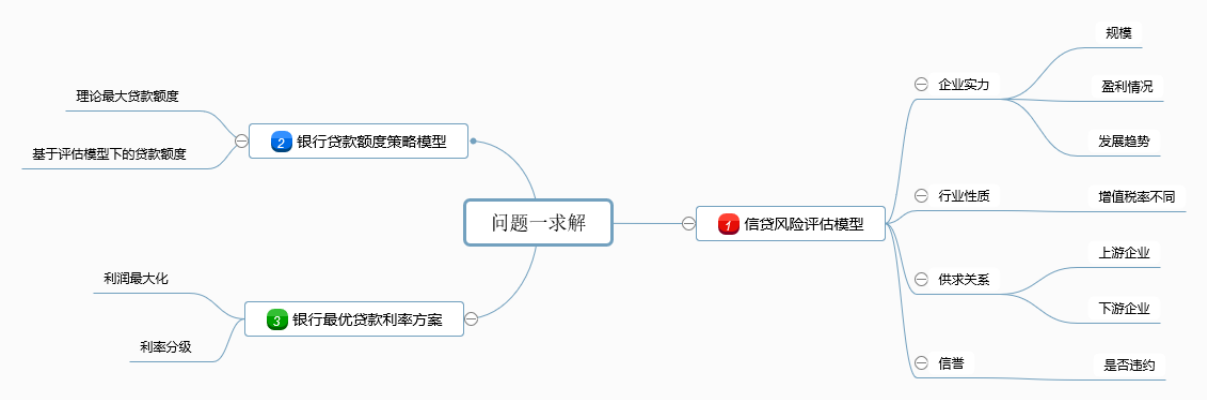

图 1　模型一思维导图

5.2.1 信贷风险量化模型[2]

银行在对一个企业进行风险评估时，根据其企业所属行业和企业的基本账目信息对该企业进行风险量化，根据风险量化的结果来确定是否信贷和信贷的金额和相关利率。结合本文题意和实际情况，考虑从财务状况、企业增值税率、上下游企业供求关系、企业信誉这四个指标对各企业进行信贷风险量化。

基于财务信息的企业实力分析

企业整体实力往往决定了其在与银行的贷款博弈中是否能够占据主动，实力越强的企业往往能够得到银行更多的信任，从而获得很多的贷款额度以及相对较低的利率。本文考虑从企业规模、企业盈利情况和企业发展趋势、业务类型几个角度对企业实力进行分析。

- 企业规模

本文考虑以 2019 年企业年度进项和销项总额作为衡量企业规模的指标。具体计算模型如下：构建对应企业 2019 年 1 月 1 日到 12 月 31 日 (共 365 天) 的日进项金额序列 I_{ij} 和日销项金额序列 O_{ij}。

定义第 i 企业 2019 年度的总流水为 S_i，其表达式为：

$$S_i = \sum_{j=1}^{365}(I_{ij} + O_{ij}) \tag{1}$$

对于所有企业的流水数据集 S，本文考虑通过 k-均值聚类对其进行分类。

通过手肘法确定聚类组数 K，根据距离均值公式可以得到 K 类企业规模数据。本模型通过手肘法和 k-均值聚类得到的分类参数虽然看似忽略了现实中经济理论，实则是充分考虑了所属环境下的各企业情况，须知银行贷款并没有一成不变的策略，一个效益较好的策略往往是基于在一定理论基础下的对经济环境的充分适应调整。本模型正是从这一思想出发，以附件一数据为参考依据得到的充分适用于附件一情况下的分类标准。

在银行得到其相对合理的 K 类企业规模数据，会对所有企业进行企业规模大小的赋分，其保证在统一层次的企业规模的分数是一致的，为使其量化程度较清晰，组间的评分差距相对较大，考虑使用离散的 0-1 间数值进行赋值且按照聚类组数平均将 0-1 划分若干档次。对于已经分成 k 类的数据按照 0，$\frac{1}{k}$，$\frac{2}{k}$，$\cdots$，1 的分数依次赋分，于本文中此种处理数据的方法为 0-1 离散赋值方法，各企业附分后的值为 $s_1^{'}$，$s_2^{'}$，$\cdots$，记第 i 个企业规模的分数为 $s_i^{'}$。

对于本文采取的先进行聚类分级再给定其具体的分数按照一定的分数比例得出最终评分结果的方式，虽然组间可能存在的误差和筛选差异较大，但是该更符合银行对每个企业各部分的评价机制。银行对企业的评价是针对企业每个板块进行等级划分制度，按照其该板块是否通过银行规定的最低下限来决定信贷是否通过和信贷的额度大小，该筛选机制对企业每个方面都由一定的要求，故对每个方面先进行聚类再求分数总和可以有效避免企业存在特殊的短板和弱点造成企业风险较大然而因为其他方面目前现状较好以致在总分上反映信息不全导致银行的判断失误。

- 企业盈利情况

企业盈利情况也是反映企业实力的重要指标。对于企业盈利数据，本文以企业 2016 年 12 月以来到 2020 年初的数据为标准，计算几年来的总盈利情况, 记第 i 个企业总盈利分数为 p_i，并通过手肘法和 k—均值聚类模型进行分类并按照 0-1 离散赋值方法进行赋分，赋分后的第 i 个企业的分数为 $p_i^{'}$。

- 企业发展趋势

企业发展趋势是银行对企业发展潜力的重要评估指标。对于发展趋势，本文考虑以各企业 2019 年整年的发展情况作为指标，计算 2019 年度净利润 $p_(i,2019)$ 与 2018 年度净利润 $p_(i,2018)$ 的比值。即企业发展潜力 θ_i 为：

$$\theta_i = \frac{p_(i,2019)}{p_(i,2018)} \tag{2}$$

考虑到企业发展过程中存在入不敷出的情况，由此争对 2018 年和 2019 年亏损情况，本文建立以下三个关系表达式：

$$\theta_i = \begin{cases} 1 + \frac{p_{(i,2019)}}{q_{(i,2018)}} & 18 \text{ 年亏损 } q_{(i,2018)}，19 \text{ 年盈利 } p_{(i,2019)}； \\ -\frac{q_{(i,2019)}}{p_{(i,2018)}} & 18 \text{ 年盈利 } p_{(i,2018)}，19 \text{ 年亏损 } q_{(i,2019)}； \\ -1 - \frac{q_{(i,2019)}}{q_{(i,2018)}} & 18 \text{ 年盈利 } q_{(i,2018)} \text{ ，} 19 \text{ 年亏损 } q_{(i,2019)} \text{ ；} \end{cases} \tag{3}$$

上述公式中对于企业发展潜力的计算具有一定主观性，主要的理论依据在于：对于转亏为盈的企业，意味着其突破了原有产业环境困境，有了相对良好的盈利模式，从投资角度出发，这类企业具有较好的投资潜力；对于刚刚亏损的企业，其企业发展可能存在一定的困境，但是由于原有模式及市场关系并没有受到不可修复的影响，相对而言仍然具有投资潜力，这一理论与股市中股价降低股民买进思想相同；对于连年亏损企业，其企业发展潜力相对而言最低。

对于已经计算出的第 i 企业发展潜力 θ_i 同样进行聚类分析，对相同类别的企业进行 0-1 离散赋值方法进行赋分，得出赋分后的第 i 个企业关于企业发展潜力的分数为 $\theta_i^{'}$。

- 财务影响的总分数

对于上述企业规模大小，企业盈利情况，企业发展趋势这三个指标共同影响着财务影响总分数的大小，对于这三者对财务影响总分数的贡献程度本题认为其三者并无区别，最后的分数是按照 1：1：1 的比例进行求和计算得出最终的财务影响总分数。记第 i 个企业的财务影响总分数的 G_i 其公式为：

$$G_i = t_i^{'} + p_i^{'} + \theta_i^{'} \tag{4}$$

最后将财务影响总分数进行 0-1 离散赋值方法进行赋分，得到第 i 个企业赋值后的企业财务影响总分数 $G_i^{'}$。

企业增值税率[3]

不同的行业所承受的风险能力是有所差异的，行业增值税率可作为其风险表征的基本量。当行业的的税率越小表明国家的政策越向其倾斜，该行业的风险也就越小, 即该行业的稳定性也就越好。在附件一中税额和金额的比值记为行业的税率。对于企业而言，其可能存在一个企业涉略多个行业，也就是说存在多个不同的税率，考虑求其所有税率的平均值作为该企业的合计税率，记第 i 个企业的合计税收为 δ_i，设一共计算出存在 τ 个税率 δ_{ik}，其计算公式为：

$$\delta_i = \sum_{k=i}^{\tau} \delta_{ik} / \tau \tag{5}$$

得到每个企业的合计税率后按照上述聚类方法后进行 0-1 离散赋值方法进行赋分，最后得出处理赋分后的处在 0-1 之间的合计税率 $\delta_i^{'}$。

上下游企业供求关系

企业的信贷风险除了包括上述的影响因素外还存在非财务影响因素，其表示各企业的营业状况稳定程度以及在银行信贷中信誉的好坏程度。本题考虑使用上下游企业的影响力和其各企业在银行的信誉作为影响因素研究非财务影响因素。

- 上游企业稳定系数

对于一个企业而言，其企业的稳定受其上下游企业的商业结构和数量影响，如果上游企业其服务的企业数量较多，即证明其上游企业的企业综合能力较强和产业链相对较完备，也就意味着上游企业可以保证其供货链的安全性和及时性，这对银行衡量一个企业材料来源的可持续性提供了较为准确的标准。

设一个企业有 N_1 个上游企业对其供货，且这 N_1 个企业不仅对该企业供货，在已知的附件中还可以确定其还对另外 $N-1$ 个企业提供供货服务，即一共对 N 个企业提供供货服务。故定义上游企业稳定系数 GU 来表示上游企业的供货数量，表达式为：

$$GU = \frac{N_1}{N} \tag{6}$$

上式表示其平均每个上游企业所需要供货的企业数量的倒数，当该系数越小时，其代表上游企业平均每个上游企业要供货的企业数量也就越多，即说明其上游企业的能力越强。

- 下游企业的稳定系数

对于一个企业作为供货商时即需要对其下游企业供货，供货的下游企业是企业获利的主要来源，如果下游企业供货数量较少或供货过度集中，此情况可能导致企业未来面临一定的挑战和危险，故企业的下游企业数量适当扩大和占比较大的下游企业份额适当减少有利于企业的稳定和可持续发展。

本题基于企业的稳定性考虑对所有需要供货的下游企业中购货金额前三名的下游企业进行购货金额求和，计算前三名的下游企业的购货金额和该供货企业所有销售额的比值，根据比值的大小就可以明确其安全性。

设企业的下游企业按照购货金额从大到小构成的购货金额数列为：a_1，a_2，$\cdots$，a_n，定义其下游企业的稳定系数 GD 为：

$$GD = (a_1 + a_2 + a_3)/(\sum_{i=1}^{n} a_i) \tag{7}$$

上述定义的下游企业的稳定系数越小，其企业的供货渠道越多且供货金额越分散，对单独购货商的依赖较少，其企业的稳定和可持续发展能力也越强。

- 供求关系分数计算

上述得到各企业的上游企业稳定系数 GU 后，对其进行聚类处理使得稳定系数在整体数据中相近的企业在同一个上游企业稳定的范围中。同上述赋分的方法完全一致，按照 0-1 离散赋值方法进行赋分得出第 i 个企业上游企业稳定系数聚类后得分记作 GUC_i

同理对各企业的下游企业稳定系数也进行 step1 和 step2 操作，得出第 i 个企业下游企业稳定系数聚类后得分，记为 GDC_i。

得出聚类后所给每个企业赋值的关于上下游企业稳定相关分数 GUC_i 和 GDC_i 后，利用这两个离散的分数对其供求关系给出量化的分数。考虑上下游对一个企业的主要程度基本完全一致，故对这两个分数按照 1：1 的分数占比得出第 i 个企业供求关系分数 w_i，公式如下：

$$w_i = GDC_i + GUC_i \tag{8}$$

最后将供求关系分数进行 0-1 离散赋值方法进行赋分，得到第 i 个企业赋值后的企业供求关系分数 w_i'。

信誉情况

- 信誉评级和违约因素

观察附件一中的 123 家企业，发现当一个企业出现违约现象的时候，绝大部分出现违约的企业其信誉评级都为 D，只存在企业 E_{29}, E_{45}, E_{87} 的信誉评级不是 D 却依然可以申请信贷。

即可知在 123 个企业中只存在 3 个企业存在违约行为但是其信誉评级不为 D，由此可以大胆的推测违约行为与信贷评级之间存在较大的相关度。

随后针对企业的信誉评级和是否违约情况进行了相关性分析。由于其均为离散型变量，故 person 相关性系数难以适用。因此运用 SPSS 进行 spearman 相关性系数求解，发现其相关显著性为 0.00，小于阈值 0.01 即可视为拥有显著相关。由此可知企业的信誉评级和是否违约情况拥有显著相关性，故本文除去企业违约情况对企业信誉的影响。

各企业在银行信贷的信誉评级是以 A,B,C,D 四个等级进行划分，其中 D 等级不能申请信贷。该等级体现了银行在信贷方面对企业的信誉评价，这会决定企业在申请信贷时是否通过和借贷金额大小等信息。

观察附件中的信誉等级分布情况，发现其四种等级的分布比例为 21.95% ，30.89% ，27.64% ，19.51% ，其分布为占比基本差异不大，故可以使用同样相互接近的四个具体的数字去代替四个等级将其量化，由于考虑到 D 等级不给考虑放贷即把 D 等级的数字量化为 0，也避免了数字被压缩的情况，故本题使用的是 0，1，2，3 四个具体的数字。

表 1　信誉等级的数字量化

信誉等级	A	B	C	D
数字量化	3	2	1	0

对于信誉情况量化后很显然是四个已经聚类完成的类别，同样采取 0-1 离散赋值方法进行赋分，4 个 0-1 分为段中的数值很明显是 0，1/3，2/3，1。设第 i 个企业的信誉情况按照分段后的数值记为 $\lambda_i^{'}$。

层次分析风险量化

本题上述对财务因素，行业性质，供求关系，信誉情况的具体情况进行了量化赋值，给出每一个因素按照聚类而确定的分数，这四个因素对企业信贷风险的影响程度决定了银行对企业最终的判断，对于每个不同的银行对这四个影响因素的重视程度和判断依据存在一定差异，故这四个因素对于最终风险量化的权值具有一定的主观性，故考虑使用层次分析法对每个影响因素的权值进行确定，最终得到每个企业最后的信贷风险分数。

本题的层次分析法用 1-8 这 8 个数字来衡量两两之间对结果影响的重要程度，在保证一致性可行的原则上得出这四个影响因素财务因素，行业性质，供求关系，信誉情况的权重分别为 v_1，v_2，v_3，v_4，结合上述各影响因素的评分，将所有因素评分按权重线性叠加，记第 i 个企业信贷风险总分数为 L_i，得出最终衡量信贷风险量化的表达式：

$$L_i = v_1 * G_i^{'} + v_2 * \delta_i^{'} + v_3 * w_i^{'} + v_4 * \lambda_i^{'} \tag{9}$$

5.2.2 银行贷款额度策略模型

理论最大信贷额度

银行在信贷金额方面需要对企业进行理论最大信贷额度的判断，结合该企业的资金流通和盈利的相关情况来确定该企业的理论最大信贷额度。由于附件一仅给到 2020 年年初的企业财务数据，由此本题考虑银行贷款时间为 2020 年且不考虑疫情等环境因素的影响。即本题将给出 2020 年银行对各企业的贷款额度数据。

银行根据企业在一年以内最大现金上的可偿还能力角度出发，考虑一年时间内该企业的营收利润来作为该企业的理论最大信贷额度，其好处是该设置几乎不会影响破坏企业正常的运作，但会存在一些突发事件或企业运营的相关问题导致利润出现大幅度波动，这导致由于银行信贷金额过大而企业无法按时偿还。所以若考虑利润存在波动的情况，可以选择从进款额度的角度出发，进款额度反映该企业的现金流准备能力，在利润存在波动的情况下该企业的现金流准备能力可以作为企业最大信贷额度的确定标准，考虑以进款额度的 10% 作为最大信贷额度也很合理且稳定。基于上述情况分析，从银行角度盈利角度可以对企业的未来年营收利润和进款额度的 10% 出发，选择其俩者中的最大值作为最大信贷额度，这更准确的衡量了该企业的真实偿还能力。

为计算各企业理论最大贷款额度，本文定义了 2020 年最大贷款额度的计算公式如下：

$$M_i = \begin{cases} P_{2020} & P_{2020} > 10\% * I_{2020} \\ 10\% * I_{2020} & P_{2020} < 10\% * I_{2020} \end{cases} \tag{10}$$

P_{2020} 为企业在 2020 年预测的营收利润，I_{2020} 为企业在 2020 年预测的进款额度。对于 P_{2020}和I_{2020} 的数据预测，本题考虑利用 2017、2018、2019 年的数据进行短期预测，故通过建立灰色预测模型对 2020 年营收利润和进款额度进行预测 (详细模型内容见附录)

实际最大信贷额度

由上述得出其理论最大信贷额度，其并没有考虑企业的风险评估问题，只是建立在企业偿还能力上的理论计算。对于银行而言，确定其理论最大信贷额度的同时需要对每个企业的风险评估分数进行考虑，对于风险较大的企业不能按照其企业的账面上得出的偿还能力来提供信贷金额，而是根据风险确定给出理论最大信贷额度的折扣率后才得到实际最大信贷额度，这保证银行给出的信贷额度更加安全稳定且更符合企业的偿还能力范围。

本文基于上述的风险评估分数进行归一化处理得到的值为其理论最大信贷额度计算成实际最大信贷额度的折扣值。定义归一化的风险评估分数称为折扣率，第 i 个企业的折扣率记为 β_i，其归一化公式如下：

$$\beta_i = \frac{L_i - \min(L_i)}{\max(L_i) - \min(L_i)} \tag{11}$$

确定每个企业的折扣率 β_i 即可计算出每个企业的实际最大信贷额度，记为 R_i，公式为：

$$R_i = \beta_i * M_i \tag{12}$$

结合本文的规定银行对确定要放贷企业的贷款额度为 $10 \sim 100$ 万元，对于问题一的情况是没有给出具体的银行信贷总额，对于计算后信贷小于 10 万的企业一律按照 10 万的信贷水平给与放贷，信贷大于 100 万的企业一律按 100 万发放信贷，银行真实信贷 R_i 的具体分段函数如下：

$$R_i = \begin{cases} 100 & 100\text{万} < \beta_i * M_i \\ \beta_i * M_i & 10\text{万} \le \beta_i * M_i \le 100\text{万} \\ 10 & \beta_i * M_i < 10\text{万} \end{cases} \tag{13}$$

企业利率分类确定

关于企业信贷时的利率确定问题，从银行的实际角度出发，每一个企业都有其独特的利率是不现实的，故需要对所有的企业进行聚类划分并规定其同一类别的利率是一致的。由于本题是从风险评估为核心考虑因素的角度出发，考虑利用其风险评估得出的量化分数为标准进行聚类分析，理想的情况是各企业依据信贷风险分数将企业分成若干类，银行对同类别中的所有企业实行相同的贷款利率。

本题将企业信贷风险总分数 L_i 进行聚类分析，将企业分类后每一组类的利率完全一致且和信贷风险总分数呈正比例关系，即当信贷风险总分数越高的时候，其银行对其信贷的利率也会越高，其信贷的企业数量也会因此降低。

5.2.3 银行最优贷款利率方案

贷款业务作为银行的主要支撑业务，是银行获利的重要途径之一，银行利率越高，等额贷款下获得的利润则越高。而根据附件三可知，利率升高客户潜在流失率也将升高，从而导致部分预备贷款最终闲置。因此，如何合理设置贷款利率使得银行得以获得更多收益问题，将是我们在调整利率时主要研究的问题。已知对于对信誉高、信贷风险小的企业银行将会给予利率优惠，因此本文在建立利率模型时需要考虑到贷款利率的离散分类问题。

本题希望建立的模型是保证风险最小时银行的收益最大，针对风险最小的目标，由于本题的银行信贷总额没有给出具体的数值且信贷企业数量也较少，本题的总风险是企业可以实际信贷的总额度反映的。对于不确定银行信贷总额且信贷企业数量也较少的情况，本题默认其所有企业的实际信贷金额都可以满足，不存在由信贷资金短缺导致的决策问题，故由上述假设可知：本题的总风险已经确定，只存在以确定利率来保证银行收益最大的目标函数。

首先对 99 家企业 (除去 24 家信誉为 D 的企业) 的风险评估分数进行 K-均值聚类分析，发现可以其聚类数量为 f，对于每个企业都有其对应的信誉等级，按照给定的利率就会存在顾客流失的情况。将企业数据与聚类数据结合，得到第 i 个企业的利率为 u_i。企业的实际信贷额度为 R_i 且该利率下客户的流失率为 l_i，设银行的总收益为 MON，即 MON 的表达式为：

$$MON = \sum_i R_i * u_i * (1 - l_i) \tag{14}$$

由此即可得到规划模型：

$$\max MON = \sum_i R_i * u_i * (1 - l_i) \tag{15}$$

$$s.t. \begin{cases} u_i\text{为待求自变量，} 4\% \leq u_i \leq 15\% \\ 10\text{万} \leq R_i \leq 100\text{万} \end{cases} \tag{16}$$

5.3 问题一模型求解

5.3.1 信贷风险评估模型求解

特征量 K-均值聚类分析为了能更加合理地表示企业的指标，本文利用 sklearn 库对各个特征值进行 K-均值聚类，将连续化的特征值离散化，并用手肘法确定合适的聚类 k 值，部分特征量手肘图如图 2 所示：(完整特征量图片详见附录 1.1)

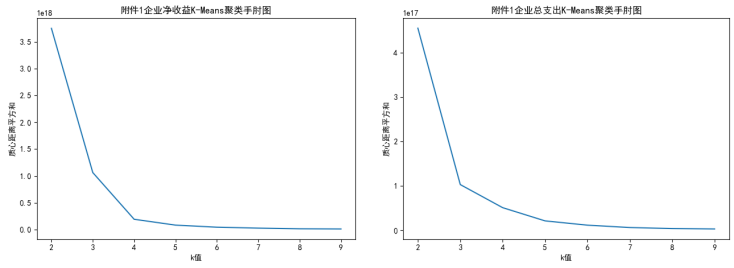

图 2　手肘图

如上图所示，通过手肘法，可以得到各个特征值的组由聚类组数，具体分类情况如下表所示：

表 2　各特征值聚类组数

类数	净收益	总支出	总收入	企业发展趋势	上游客户集中度	下游客户集中度
聚类组数	4	5	5	6	4	4

层次分析法

基于对行业领域、信誉、财务因素、供求关系重要程度的比较，本文得到下表所示数据。

表 3　对比判断优选矩阵

	行业	信誉	财务因素	供求关系
行业	1	1/5	1/7	1/4
信誉	5	1	1/2	2
财务因素	7	2	1	3
供求关系	4	1/2	1/3	1

$$CI = \frac{(\lambda_{max} - m)}{m-1} = \frac{(4.05-4)}{4-1} = 0.017 > 0.01 \tag{17}$$

由 $CI > 0.01$，初步认定信贷风险模型中层次分析法的相对优先顺序无逻辑混乱。

$$CR = \frac{CI}{RT} = \frac{0.017}{0.90} = 0.019 < 0.1 \tag{18}$$

由 $CR < 0.1$ 可知层次分析法得到的矩阵具有较为满意的一致性。

表 4　abc

目标层	准测层	准测层权重
风险评估体系	信誉	0.29
	供求关系	0.17
	财务因素	0.48
	行业	0.06

将行业领域、信誉、财务因素、供求关系依据归一化权重系数表求得信贷风险评估分数结果如下：（其中分数越高，风险越小）

表 5　企业风险归一化评分表

公司编号	归一化风险	公司编号	归一化风险	公司编号	归一化风险
E1	0.5584	E2	0.8827	E3	0.3783
E4	0.3423	E5	0.5214	E6	0.6948
E7	0.9985	E8	0.9521	E9	0.5430

5.3.2 银行贷款额度求解

首先基于企业 2017、2018、2019 三年的进项发票金额和企业利润进行灰色预测，得到 2020 年的相关预测数据。某一预测结果如图 3 所示：

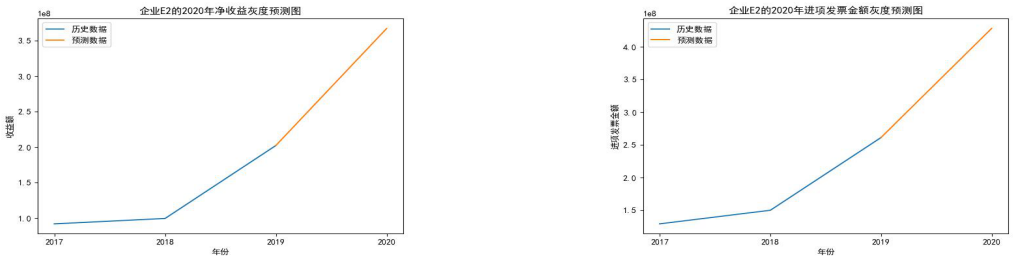

图 3　企业 E_2 相关指标预测

随后依据公式 (15)，结合企业风险评估分数，得到企业相应贷款额度。对各企业贷款额度进行分析，发现有 13.7% 的企业贷款额度为 10 万元，37% 的企业贷款额度为 100 万元，19.3% 的企业信誉等级为 D，不予放贷。

5.3.3 银行最优贷款利率求解

将企业的信贷风险指数再进行 K-均值聚类，发现聚为三类时聚类效果较好。因此将所有企业依据信贷风险大小分为三类 (详细数据见附件 1 的离散化信息.xls)，每一类

拥有各自的最优利率。为了确定各自的利率，利用贪心算法对每一类风险的企业利率进行假设，在不同的利率条件下，综合利率和企业流失率计算得到银行的最终收益。收益最高的利率方案即为银行的最优利率方案。最终求解得到的银行理论最多获利值为1886384.8 元，对应的三组利率为 0.0505，0.0585 和 0.0665。

5.4 问题一模型检验

根据银行已成熟经验可知，现实中企业信贷金额和利率的确定与企业信誉等级有极高的相关度。故本题建立的企业信贷风险模型得出的信贷风险总分数应该与银行所给的企业信誉等级拥有显著相关性。为了证明两者间确实相关，本文对附件一各个企业的信贷风险指数与企业信誉等级数据进行了 T 检验。最终检验显著性为 0.015，在显著性小于 0.05 时说明两者相关性显著。由此说明本文所建立的模型在实际意义上具有可解释性。

六、 问题二模型建立与求解

6.1 问题二模型建立

相比于问题一，本题的银行和客户基本信息均发生了变化。客户信息由附件一的123 家有信贷记录企业转变为附件二的 302 家无信贷记录企业，即企业信誉这一重要贷款指标需要构建参数进行表征，以保证模型一在问题二的延续适用性; 此外，银行贷款总额度变为了一亿元，即存在额度用完而仍无法满足企业贷款需求的问题，因此需要考虑建立基于额度不足这一约束条件下的银行最小贷款风险和最大盈利的规划模型。

6.1.1 企业信誉等级确定

已知财务信息越好的企业，其信誉往往较高，因此对于企业信誉，本题猜想企业的信誉信息可以通过财务信息来体现。本文依据附件一数据，提取了行业信息、企业利润、支出情况、收入情况、交易取消率、上游企业影响、下游企业影响、企业发展规模，企业发展趋势、有效发票比例等因素。为了确定企业信誉等级与影响因素之间的关系，可以采用决策树、神经网络、回归等模型进行求解。综合考虑到模型的可解释性和准确度，本文采用决策树模型[4] 进行求解。并在附件一 123 家企业的中随机提取 99 家企业信息作为训练集、24 家企业信息作为测试集进行模型的验证。

银行在进行企业信誉等级划分判断时考虑了很多企业及其相关因素，由于本题给出各企业的信息不够完整导致使用决策树的模型对企业信誉等级的模拟正确性相对较低。显然按照一一对应来判断正确率的方法过于苛刻，为使得其模拟的正确率较高，考虑放宽判断正确的标准。规定通过决策树判断的等级是附件一的企业已给出的信誉等级的上下两个等级都认为其符合正确判断标准，例如：实际附件一中的企业信誉等级为 B，那通过决策树模型判断其信誉等级为 A，B，C 均可认为正确，等级 D 的合理等级范围为 C ，D。对于 D 等级如果以 C 同样也为正确答案的情况下会导致其银行从不给予企业信贷到可以信贷，但是 C 等级给予的信贷额度也较少，对结果上影响不大，本文不考虑该情况的影响。

ID3 **算法**

ID3 算法使用信息增益作为其属性选择指标。由信息 (D) 表示的数据分区的熵计算公式如下：

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i) \tag{19}$$

其中数据分区 D 是带有类标记的元组的训练集 p_i 表示任意元组属于分区 D 中的类 C_i 的非零概率。它由以下公式计算:

$$P_i = \frac{|c_{ir}D\ |}{D} \tag{20}$$

此外，为了衡量每个属性中的元素与 D 中的元素之间的关系，我们假定 D 中的元素是基于属性 $A(a_1, a_2, \ldots, a_n)$ 进行划分的。如果 A 中的 n 个不同值都是离散值，则它们可以直接对应于 A 上测试的输出，分区越纯，结果越好。实际上，这些分区通常包含属于不同类的元素衡量获得准确分类所需的信息量的函数是:

$$Info_A(D) = -\sum_{i=1}^{m} \frac{|D_k\ |}{|D\ |} * Info(D_k) \tag{21}$$

将信息增益定义为原始信息需求 $Info(D)$ 与分割后的新信息需求 $Info_A(D)$ 之间的差得到 $Gain(A) = Info(D) - Info_A(D)$。信息增益值越大，期望信息越少，从而分区的纯度越高。

6.1.2 银行贷款额度分配模型

首先本题假设银行提供的一亿元贷款额度无法满足信誉非 D 的所有企业的贷款需求 (模型求解中将验证假设)。为此本题考虑建立如下银行贷款额度分配模型：

首先基于决策树得到的企业信誉表征方案补全附件二 302 家企业的信誉等级 Q。对于信誉为 D 的企业将不提供贷款服务，记信誉等级不为 D 的企业总数为 N。

再将附件二企业数据代入 5.2.1 信贷风险量化模型求解得到 N 家企业的评分降序序列 $R(R_1, R_2, \ldots, R_N)$ 以及对应的企业序列 $E(E_1, E_2, \ldots, E_N)$。由模型一可知，评分越高表示企业风险越小。

有限信贷总额下的银行借款策略

由于贷款金额有限，从贷款风险最小的角度出发，银行将借更多的钱给风险较低的企业，优先满足低风险企业的信贷额度要求。即对于 302 家企业的风险评分序列 R，银行将首先满足排序靠前的企业贷款需求。即对于企业序列 $E(E_1, E_2, \ldots, E_N)$，银行将依次满足各企业所需要的贷款。但银行满足了 E_j 企业的贷款需求时，银行可用贷款余额 Ψ_j(单位：元) 计算公式为：

$$\Psi_j = \Psi_0 - M_1 - M_2 - \cdots - M_j, j\epsilon(1, N) \tag{22}$$

上述公式中的 M_j 序列为企业 E_j 序列所对应的理论最大贷款额度 (该企业数据仍然通过模型一方法灰色预测得到)。

6.1.3 最大利润下的利率规划模型

由于企业进行贷款服务时有一定的概率 (即潜在流失率) 放弃贷款服务，且随着利率的升高这种概率也将越来越高。对于因为客户流失而重新闲置在银行手中的资金，银行将会将其再次分配给其他贷款客户。由于每类客户均存在潜在流失率，为简化模型，本文建立关于年度贷款总额为一亿元的约束条 $\sum_i R_i * (1 - l_i) \leq 100000000$，即任意时期对于所有理论上向银行贷款的企业，它们在对应利率下的获得的贷款总额度和达成贷款合作的概率 (即 1-潜在流失率) 应小于等于一亿元。结合模型一的目标规划模型和问题二引入的年度贷款总额约束条件，模型二得到如下所示的最大利润规划模型：

$$\max MON = \sum_i R_i * u_i * (1 - l_i) \tag{23}$$

$$s.t. \begin{cases} u_i\text{为待求自变量，} 4\% \leq u_i \leq 15\% \\ 10\text{万} \leq R_i \leq 100\text{万} \\ \sum_i R_i * (1 - l_i) \leq 100000000 \end{cases} \tag{24}$$

利用贪婪算法对规划模型求解即可得到银行利率贷款策略。

6.2 问题二模型求解

6.2.1 企业信誉求解

Step1：**初始解生成**

利用机器学习库 sklearn 中的决策树模型对特征量与企业信誉评级数据进行训练。

首先选用增值税率、净收益、总支出、总收入、交易取消率、上游客户集中度、下游客户集中度、企业发展趋势等指标作为特征量，得到各个特征的影响力为：[0.11337454,0 ,0.03798653, 0.17470763, 0.16879765, 0.13231992 ,0.27276071, 0.10005302]。

表 6　决策树结果表一

训练集准确度	0.8673469387755102	训练集 D 等信誉错误率	0.1875
测试集准确度	0.8	测试集 D 等信誉错误率	0.4

分析上述数据，发现准确度较高。但是测试集 D 等信誉错误率偏高。而在问题二模型中，D 等信誉的企业将无法获得贷款，因此该指标较为重要。分析其原因可能是数据冗余度太高，导致决策树中出现影响结果的无关决策。为了减小无关数据对决策树模型的影响，本文进行了无关因素的剔除和决策树的剪枝。

Step2：**相关性分析降维**

将增值税率、净收益、总支出、总收入、交易取消率、上游客户集中度、下游客户集中度、企业发展趋势与信用等级进行相关性分析，发现其结果如下：

表 7　相关性分析矩阵

特征量	增值税率	净收益	总支出	总收入
显著性	0.118	-0.149	-0.339	-0.347
相关性	0.194	0.100	0.000	0.000
特征量	交易取消率	上游客户集中度	下游客户集中度	企业发展趋势
显著性	-0.383	-0.253	-0.431	0.264
相关性	0.000	0.005	0.000	0.003

由于当 $p<0.05$ 时两者相关，当 $p<0.01$ 时两者显著相关，因此在决策树中剔除经营类型和净收益两个变量。

Step3：**决策树剪枝**

重新进行决策树训练，结果如下：选用总支出、总收入、交易取消率、上游客户集中度、下游客户集中度、企业发展趋势、各个特征的影响力为: [0.02613541 0.3233813 0.19081063 0.27167066 0.08810141 0.09990058]。

表 8　决策树结果表二

训练集准确度	0.83673469388	训练集 D 等信誉错误率	0.25
测试集准确度	0.76	测试集 D 等信誉错误率	0.333333333

将剪枝后的决策树利用 graphviz 插件进行可视化，简化后可得到如下图所示结果。

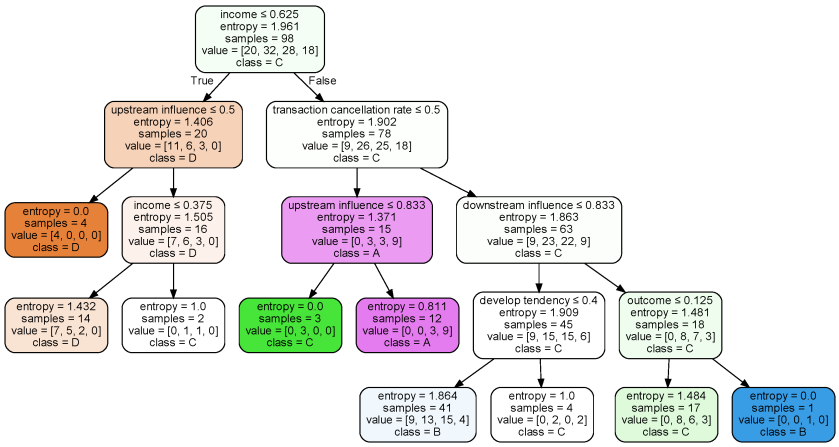

图 4　决策树可视化图

分别对其中一个决策信用为 A 和 D 的节点进行跟踪分析，发现决策为 A 需要收入大于 0.625，交易取消率需要小于 0.5，上游影响力 (越小越好) 需要小于 0.833；而决策

为 D 需要收入小于 0.625，上游影响力 (越小越好) 需要大于 0.5，收入需要小于 0.375。决策条件符合实际情况，说明该决策树模型具有良好的现实意义和可解释性。

Step4：**附件二企业信誉模拟**

基于决策树求解得到的特征量及其权值，可以进一步计算得到附件二企业信誉情况，部分企业信誉情况见下表：

表 9　部分企业信誉信息表

公司编号	经营类型	净收益	总支出	总收入	交易取消率	上游影响力	下游影响力	企业发展趋势	信用等级
E124	0	0	1	1	1	0.75	0.33	0.33	0.67
E126	0	1	0.67	1	1	0	0.33	0.33	0.33
E425	1	0	0	0	0.67	0	0	0.33	0.67

6.2.2 银行贷款策略求解

依据 5.2.2 银行贷款额度策略模型得到在不考虑银行贷款总额度下的企业贷款金额，计算附件二在模型一中的企业贷款总额为 173318492.9 元，远超过 1 亿元，说明当前银行贷款储备无法满足信誉非 D 的所有客户的贷款需求，即 6.1.2 假设成立，银行贷款额度分配模型适用于当前问题。

首先将企业根据其贷款风险值分为 3 类，每一类企业拥有其特定的贷款利率。

为了找到限额为 1 亿条件下，银行在限制风险的前提下达到最大收益时的利率组合，本文在模型一的最优贷款利率规划模型的基础上，加入最大贷款总额不超过 1 亿的限制再进行求解。得到适用于问题二的最优贷款利率组合，结果分别为 6.25%,6.65% 和 8.25%，对信贷风险小的企业提供较小的贷款利率。此时银行的交易贷款总额为 97162110.1 元，没有超过 1 亿的限额，且银行能获得最大化的利润，利润值为 6651140.9 元。

七、 问题三模型建立与求解

7.1 问题三模型准备

问题三考虑在突发因素影响的情况下，银行对各企业信贷金额和利率的定价策略调整方案。本题以新型冠状肺炎作为本题的突发影响因素，考虑新型冠状肺炎事件发生对各行业的影响。本文参考国家统计局给定的各行业 2020 年第二季度和上半年国内生产总值（GDP）初步核算结果报告的数据[5]。结合新型冠状肺炎事件发生的时间，由于使用上半年的 GDP 比上年同期增长比例要比使用第二季度更加全面反映行业受该突发事件的影响全貌，故选择使用上半年的 GDP 同上年增长比例数据来进行疫情对行业影响的划分。其国家统计局的部分数据如下所示：

表 10　2020 年上半年国内生产总值 (GDP) 初步核算结果表

行业	上年同期增长 (%)	行业	上年同期增长 (%)
农林牧渔业	1.1	住宿和餐饮业	-26.8
工业	-1.8	金融业	6.6
制造业	-2.5	房地产业	-0.9
交通运输、仓储和邮政业	-5.6	信息传输、软件和信息技术服务业	14.5
批发和零售业	-8.1	租赁和商务服务业	-8.7
建筑业	-1.9	其他服务业	-1.4

由图可知，新型冠状肺炎突发事件的到来对各行业各企业的影响有较大差异，本题考虑将行业划分为五个受突发事件影响剧烈程度的行业类别：突发事件对行业有很较大正影响，有轻微正影响，几乎没影响，有轻微负影响，有较大负影响。同时希望将上述表格中国家给定的行业合理划分到这五个区域之中，考虑以 5% 作为区间跨度，定义同上年上半年增长的比例大于 7.5% 企业称疫情对企业有较大正影响，2.5% 到 7.5% 之间为有轻微正影响，-2.5% 到 2.5% 之间为几乎没影响，-2.5% 到-7.5% 之间为有轻微负影响，小于-7.5% 为有较大负影响。

按上述增长比例进行分类别可得到新型冠状肺炎事件对行业影响的分级，如下表所示：

表 11　行业分类

较大正影响	信息传输，软件和信息技术服务业
轻微正影响	金融业
几乎没影响	农林渔业，工业，批发和零售业，房产产业，其他服务业
轻微负影响	制造业，交通运输，仓储和邮政业，
较大负影响	住宿和餐饮业，批发和零售业，租赁和商务服务业

按照如上标准得到行业划分后，对所要研究的所有企业进行关键词提取后人为将所有企业分到这五个大类之中，划分到同一个大类之中的企业表示它们所受新冠肺炎影响的正负性质和程度是完全一致的。

7.2 问题三模型建立

7.2.1 疫情对行业影响量化

由下表可以看出疫情对各行业的影响程度不同，对于疫情各行业经济的变化情况能够表征该行业对疫情这一突发事件的抵抗能力。对于模型准备中的五个行业分类，本题基于突发事件对其的影响程度进行量化赋值，其正负表示新型冠状肺炎对行业的正负影

响。在赋值时为保证其区间跨度为 1，分别以-0.5，-0.25，0，0.25，0.5 进行赋值，具体赋值对应关系如下：

表 12　行业分类赋值

较大正影响	轻微正影响	几乎没影响	轻微负影响	较大负影响
0.5	0.25	0	-0.25	-0.5

7.2.2 企业实际最大信贷额度

将上述疫情对行业影响量化赋值后作为银行确定其最大信贷额度的一个指标，在问题二中其最大信贷额度是由信贷风险分数计算的折扣率和灰色预测得出的理论最大信贷额度确定的，本题灰色预测得出的理论最大信贷额度同问题二数值上相等，但本题的折扣率银行需要从疫情对行业影响和风险评估分数两个指标进行计算量化，很显然当疫情对行业影响得分数数值越大，证明其其行业运营更加稳定，故银行判断其企业理论最大信贷额度的折扣率的折扣率越大，即尽可能会满足企业的信贷要求；另外当风险评估分数越高，即证明其企业风险越大，这会导致银行给企业的折扣率因为基于风险的考虑会降低，呈负相关。

step1: 其上述的第 i 个企业风险评估总分数 L_i 同问题一，二的方法一样按照聚类方法对其进行离散的 0-1 间数值进行赋值，得到 L_i'。

step2：对第 i 个企业由上述的疫情对行业影响而赋值的行业分类赋权分数为 X_i。其数值已经在跨为 1 的范围区间内，无需进行数据再次赋值。银行对企业信贷的折扣分数 Y_i，其指标权重为 A_1 和 A_2，其具体公式为：

$$Y_i = A_1 * L_i' + A_2 * X_i \tag{25}$$

考虑本题的侧重点和实际情况可知，本题疫情对于行业的影响程度是银行对于目前社会现状作出判断的重要依据，其所占的比重比起风险评估相对较大，考虑到其正负相关性，A_1 设置成-1，A_2 设置成 2。

step3：将折扣分数按聚类方法对其进行离散的 0-1 间数值赋值，得到折扣率 Y_i'

step4：将最大信贷额度与上述的折扣率相乘得到其实际信贷额度。

7.2.3 企业利率分类

本题企业利率的分类不同于之前模型只依据于企业风险评估分数，而是加入行业受疫情的影响程度。如果行业受疫情的负影响越大，在政府和银行角度上需要对行业进行扶持工作，对于受疫情负影响较大的行业，银行对其利率进行调整来帮助企业度过难关，故受疫情的负影响越大其利率就会越低，其在数值上是呈正相关的且利率也会随着风险系数的增大而变大，即在企业利率分类时按照这两个指标的线性相加聚类后来判断其企业利率的类别。衡量利率分类的利率系数 N_i，其与风险评估赋值后的 L_i' 以及疫情对行业影响而赋值的行业分类量化分数为 X_i 之间的关系为：

$$N_i = B_1 * L_i' + B_2 * X_i \tag{26}$$

同样考虑本题的侧重点和实际情况可知，本题疫情对于行业的影响程度是银行对于目前社会现状作出判断的重要依据，其所占的比重比起风险评估相对较大，考虑到其正

负相关性，A_1 设置成 1，A_2 设置成 2，在进行聚类时考虑和起初行业受疫情影响的类别数量一致，这保证最后的企业分层更加偏向于疫情对行业影响这一指标，故将其聚成 5 类且保证每一个类别的税率相等。

7.2.4 信贷策略优化模型

对于企业贷款额度和利率策略的求解，本题考虑用模型一建立的企业贷款额度模型和模型二改进得到的利率规划模型进行求解，唯一不同的是引入了 7.2.2 的折扣分数 Y_i 和 7.2.3 的利率系数 N_i。因此不再对模型进行赘述。

7.3 问题三模型求解

将附件二数据带入求解，解得贷款策略如下表所示。观察可知疫情正影响企业时其额度上升。为了兼顾疫情影响和贷款风险，负影响企业额度下降且利率降低。

表 13　引入疫情风险参数后的银行贷款策略对比

企业对象	疫情影响值	问题二实际额度	问题二贷款利率	问题三实际额度	问题三贷款利率
E153 个体经营	1000000	0.0625	861516.7	0.0505	-0.5
E318 商贸有限公司	168555.2	0.0665	995831.8	0.0505	-0.25
E188 硬质合金有限公司	421648.2	0.0665	850597.7	0.0665	0
E232 投资发展有限公司	100000	0.665	100000	0.665	0.25
E227 安全技术有限公司	218719.7	0.0665	698293.4	0.825	0.5

7.4 问题三灵敏度分析

问题三模型中的企业实际最大信贷额度和企业利率分类中的 A_1，A_2，B_1，B_2 是根据实际情况带有一定主观性的赋值，其由于个人或组织在进行主观做赋值的过程中存在差异导致最终的银行收益存在差异。本题假设 A_2 与 A_1，B_2 与 B_1 在数值上的比值一样的，定义疫情影响权重比为 T 为其数值比值，在其模型建立是其 T 取值为 2，考虑 T 的变化对银行收益的灵敏度分析。考虑在 T 取值为 2 的基础上下浮动 50%, 即 T 的浮动范围为 1-3，取步长为 0.125 分别带入计算得出银行收益，作出所有点的图像如图 5 所示：

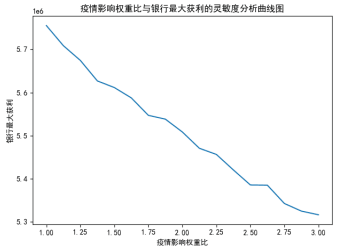

图 5　疫情影响权重比与银行最大获利的灵敏度分析曲线图

由图可知，随着其疫重比的最大也就意味着银行对于疫情对行业影响的重视程度越来越大，对其企业风险评估方面的重视程度也就越来越少，这会导致由于银行存在对目前受疫情负影响程度较大企业的政策倾斜导致牺牲其收益去对企业进行帮助，故其银行收益越来越少，并且可以看出银行收益随着情影响权疫情影响权重比 T 的灵敏度非常明显，故在银行对疫情影响权重比 T 的选择确定时，要合理且谨慎对其确定权重比，过低解决不了社会问题，过高对银行的收益产生巨大影响。

八、 模型评价推广

8.1 模型优点

1. 本文的离散的 0-1 数值赋值方法保证企业的每个评价部分都可达到银行的信贷标准，不存在企业由于存在发展不平衡导致银行判断出现失误。
2. 层次分析法等主观赋值模型使得本文模型的适用性较广泛，变量权值可以按照适时的实际情况变更。
3. 模型考虑风险评价模型指标比较细化，由基本数据推导出的评价指标较为完整。

8.2 模型缺点

1. 本文由于存在离散的 0-1 数值赋值方法，没有严格反映企业与企业之间的细致差别，存在评价误差导致分层较为笼统。
2. 层次分析主观性较强，可能出现评价与实际情况有一定误差风险。
3. 在细化评价指标的同时，指标间的相关指标也在加多，导致某个指标可能过分重叠评价。

8.3 模型推广

本文在构建评价指标时使用的离散的 0-1 数值赋值方法对多角度评价问题具有一定的适用性，特别是适用那些对于不严格等级划分但是要求每个评价部分都要达到一定标准的评价体系，例如对于等级人才的划分，重点城市和旅游景区的等级划分等等。在现实生活中很多情况个人和机构需要结合自身情况对指标进行主观赋值，本文采取的层次分析法是在进行主观赋值时较常用的方法，在实际生活中的应用较为广泛。

参考文献

[1] 王勇军. 商业银行信贷风险识别和控制 [D]. 中国科学院大学 (工程管理与信息技术学院),2016.

[2] 于紫波. 我国商业银行信贷业务风险评价与控制研究 [D]. 天津大学,2015.

[3] 凌洋. H 银行小微企业信贷业务风险管理优化策略的研究 [D]. 安徽财经大学,2019.

[4] 李莉. 数据挖掘技术决策树分类算法 (ID3 算法) 研究 [J]. 电子技术与软件工程,2018(14):181-182.

[5] 秦唏. 新冠疫情对我国若干行业的影响 [J]. 新理财 (政府理财),2020(08):18-20.

附录 A 支撑材料清单

1. python 程序代码源文件：
   question1.py;
   question2.py;
   question3.py;
2. 根据附件进行程序处理得到的企业连续和离散化数据信息和信贷风险：
   附件一数据分析结果.xls;
   附件二数据分析结果.xls;
   附件 1 的离散化信息.xls;
   附件 2 的离散化信息.xls;
   企业各年收入支出信息.xls;
3. 程序解得的银行贷款策略：
   问题一贷款策略.xls;
   问题二贷款策略.xls;
   问题三贷款策略.xls;
4. 问题二决策树模型示意图：tree2.png;

附录 B 部分表格图片结果展示

附录 2.1

表 14　问题一贷款策略

公司编号	实际额度	贷款利率	公司编号	实际额度	贷款利率
E1	1000000	0.0585	E14	1000000	0.0585
E2	1000000	0.0665	E15	1000000	0.0585
E3	1000000	0.0585	E16	1000000	0.0585
E4	1000000	0.0585	E17	1000000	0.0665
E5	1000000	0.0585	E18	1000000	0.0585
E6	1000000	0.0665	E19	1000000	0.0665
E7	1000000	0.0665	E20	1000000	0.0585
E8	1000000	0.0665	E21	1000000	0.0585
E9	1000000	0.0585	E22	1000000	0.0665
E10	1000000	0.0585	E23	1000000	0.0585
E11	1000000	0.0585	E24	1000000	0.0665

附录 2.2

表 15　问题二贷款策略

公司编号	实际额度	贷款利率	公司编号	实际额度	贷款利率
E138	1000000	0.0625	E170	1000000	0.0665
E139	1000000	0.0625	E171	1000000	0.0585
E140	1000000	0.0625	E172	1000000	0.0585
E141	0	0	E173	881318.7665	0.0585
E142	1000000	0.0625	E174	1000000	0.0625
E143	1000000	0.0665	E175	1000000	0.0665
E144	1000000	0.0625	E176	1000000	0.0585
E145	1000000	0.0585	E177	428512.8323	0.0585
E146	883475.3928	0.0625	E178	1000000	0.0585

附录 2.3

表 16　问题三贷款策略

公司编号	实际额度	贷款利率	公司编号	实际额度	贷款利率
E179	1000000	0.0705	E192	1000000	0.0705
E180	1000000	0.0665	E193	738428.0501	0.0625
E181	359311.6898	0.0545	E194	1000000	0.0545
E182	1000000	0.0705	E195	1000000	0.0585
E183	1000000	0.0705	E196	1000000	0.0705
E184	1000000	0.0665	E197	0	0
E185	1000000	0.0705	E198	948385.7491	0.0665
E186	1000000	0.0545	E199	1000000	0.0545
E187	100000	0.0625	E200	319957.9168	0.0545
E188	850597.6535	0.0665	E201	1000000	0.0665

附录 2.4

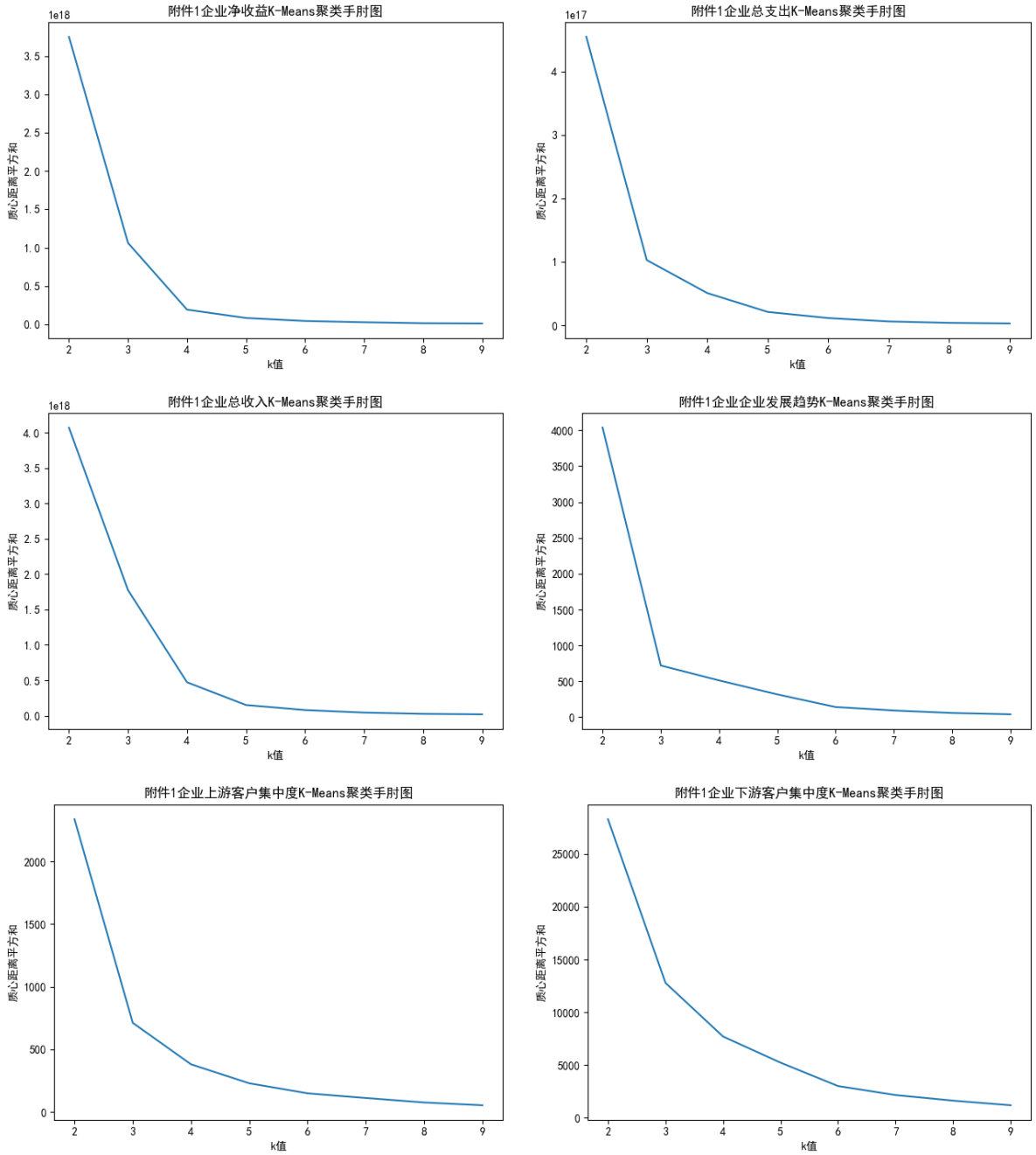

图 6　各特征量手肘图汇总

附录 C 代码

```
# 运行环境: Windows 10 专业版 , Python3.7 64bit , Pycharm 2020.2.1 专业版
# 预装软件: graphviz

import matplotlib.pyplot as plt, numpy as np, scipy.stats as stats, pydotplus, xlrd, xlwt
from sklearn import tree
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
```

```
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

addfiles = ['附件1：123家有信贷记录企业的相关数据.xlsx',
    '附件2：302家无信贷记录企业的相关数据.xlsx',
    '附件3：银行贷款年利率与客户流失率关系的统计数据.xlsx'
, '附件一数据分析结果.xls', '附件二数据分析结果.xls', '企业各年收入支出信息.xls',
    '附件1的离散化信息.xls', '附件2的离散化信息.xls']


# 读取文件
def loadxls(file, sheet=0, str=0, stc=0, enc=None, enr=0):
re = []
d = xlrd.open_workbook(addfiles[file - 1], 'rb')
table = d.sheets()[sheet] # 通过索引顺序获取
nrows = table.nrows
for i in range(str, nrows - enr):
re.append(table.row_values(i, start_colx=stc, end_colx=enc))
return re


# 保存文件
def savexls(d, file, cols):
wb = xlwt.Workbook(encoding='utf-8')
ws = wb.add_sheet('Sheet1')
ws.write(0, 0, label=file)
for i, t in enumerate(cols):
ws.write(1, i, label=cols[i])
for i, t in enumerate(d):
for j, k in enumerate(d[0]):
ws.write(i + 2, j, label=d[i][j])
wb.save(file + '.xls')


# 依据税率和企业类型区分信贷政策
def brandclu(file):
clus = [['3', '劳务', '建筑', '个体经营', '影', '管理', '五金', '设计服务', '维修'], # 3%
['6', '房地产', '教育', '发展', '科学', '策划', '质量', '设计', '传播', '广告', '投资', '代理',
    '印务', '餐饮', '服务',
'律师', '事务', '实业', '物资', '运', '物流', '汽贸', '快递', '环保', '地质', '灾害', '土地',
    '生态', '猕猴桃', '园艺',
'园林'], # 6%
['9', '图书', '石化', '天然气', '装饰', '农业', '调味品', '蔬菜', '建设'], # 9%
['13', '建材', '包装', '材', '木业', '纸业', '塑胶', '经营', '童装', '纺织品', '生活用品', '鞋',
    '服饰', '体育', '家居',
'卫浴', '门窗', '设备', '空调', '家电', '器材', '电器', '机电', '花', '轮胎', '化工', '食品'], #
    13%
['16', '电子', '技术', '机械', '工贸', '车', '工程', '网络', '文化', '贸易', '电气', '合金',
    '塑料', '商贸', '医疗', '药',
'科技']] # 16%
for i, t in enumerate(file):
flag = False
for k, c in enumerate(clus):
if flag:
break
for j in c:
```

```
if j in t[1]:
if len(file[i]) == 2:
file[i].extend(['', ''])
else:
file[i][2] = (ord(file[i][2]) - ord("A")) / 3
if file[i][3] == '是':
file[i][3] = 1
else:
file[i][3] = 0
file[i].append(int(clus[k][0]))
flag = True
break
return file


# 分析企业交易信息
def get_tradeinfo(compinf, buyfile, cellfile):
labels = ["A", "B", "C", "D"]
fracs = [27, 38, 34, 24]
plt.pie(x=fracs, labels=labels, autopct="%0.2f%%")
plt.title('附件一企业信誉评级分布')
plt.show()

# [交易数 交易额 取消率]
def getinfo(file):
inf = [[0, 0, 0]]
comp = 0
comptitle = file[0][0]
for i in file:
if i[0] != comptitle:
inf[comp][2] = inf[comp][2] / inf[comp][0]
comp += 1
comptitle = i[0]
inf.append([0, 0, 0])
inf[comp][0] += 1
inf[comp][1] += float(i[4])
if i[7] == '作废发票':
inf[comp][2] += 1
inf[comp][2] = inf[comp][2] / inf[comp][0]
return inf

buyinf = getinfo(buyfile)
cellinf = getinfo(cellfile)
for i in range(len(buyinf)):
compinf[i].extend(
[cellinf[i][1] - buyinf[i][1], buyinf[i][1], cellinf[i][1], 100 * (buyinf[i][2] +
    cellinf[i][2]) / 2])
return compinf


# 分析上下游影响度
def get_influ(compinf, buyfile, cellfile):
guests = np.zeros((2, 42000), np.int16)
bef = 0
for i in buyfile:
if i[7] == '有效发票' and (bef == 0 or bef != i[3]):
bef = i[3]
```

```
guests[0][int(i[3][1:])] += 1
comp = 0
comptitle = buyfile[0][0]
buyinf = [0]
brands = set()
for i in buyfile:
if i[7] != '有效发票':
continue
if i[0] != comptitle:
for j in brands:
buyinf[comp] += guests[0][j]
buyinf[comp] = len(brands) / buyinf[comp] * 100
comp += 1
comptitle = i[0]
buyinf.append(0)
brands = set()
brands.add(int(i[3][1:]))
for j in brands:
buyinf[comp] += guests[0][j]
buyinf[comp] = len(brands) / buyinf[comp] * 100

comp = 0
comptitle = cellfile[0][0]
cellinf = [0]
guests = np.zeros((1, 42000), np.float)
for i in cellfile:
if i[7] != '有效发票':
continue
if i[0] != comptitle:
t = 0
for j in range(3):
t += np.max(guests[0])
d = np.where(guests[0] == np.max(guests[0]))
guests[0][d[0][0]] = 0

cellinf[comp] = t / cellinf[comp] * 100
comp += 1
comptitle = i[0]
cellinf.append(0)
guests = np.zeros((1, 42000), np.float)
guests[0][int(i[3][1:])] += float(i[4])
cellinf[comp] += float(i[4])
t = 0
for j in range(3):
t += np.max(guests[0])
d = np.where(guests[0] == np.max(guests[0]))
guests[0][d[0][0]] = 0

cellinf[comp] = t / cellinf[comp] * 100
for i, t in enumerate(compinf):
compinf[i].extend([buyinf[i], cellinf[i], (buyinf[i] + cellinf[i]) / 2])
return compinf


# gm模型灰度预测各个企业的今年收益率
class GrayFore():
def __init__(self, d):
```

```
self.d = d
self.fore = d.copy()

def build_model(self):
X_0 = np.array(self.fore)
X_1 = np.zeros(X_0.shape)
for i in range(X_0.shape[0]):
X_1[i] = np.sum(X_0[0:i + 1])
Z_1 = np.zeros(X_1.shape[0] - 1)
for i in range(1, X_1.shape[0]):
Z_1[i - 1] = -0.5 * (X_1[i] + X_1[i - 1])

B = np.append(np.array(np.mat(Z_1).T), np.ones(Z_1.shape).reshape((Z_1.shape[0], 1)), axis=1)
Yn = X_0[1:].reshape((X_0[1:].shape[0], 1))

B = np.mat(B)
Yn = np.mat(Yn)
a_ = (B.T * B) ** -1 * B.T * Yn

a, b = np.array(a_.T)[0]

X_ = np.zeros(X_0.shape[0])

def f(k):
return (X_0[0] - b / a) * (1 - np.exp(a)) * np.exp(-a * (k))

self.fore.append(f(X_.shape[0]))

def forecast(self, time=1):
for i in range(time):
self.build_model()
return self.fore.copy()


# 问题一：企业基本信息获取代码
def ques1_1():
comp_inf1 = brandclu(loadxls(1, 0, 1))
comp_inf2 = brandclu(loadxls(2, 0, 1))
buyfile1 = loadxls(1, 1, 1)
cellfile1 = loadxls(1, 2, 1)
buyfile2 = loadxls(2, 2, 1)
cellfile2 = loadxls(2, 1, 1)
comp_inf1 = get_tradeinfo(comp_inf1, buyfile1, cellfile1)
comp_inf2 = get_tradeinfo(comp_inf2, buyfile2, cellfile2)
comp_inf1 = get_influ(comp_inf1, buyfile1, cellfile1)
comp_inf2 = get_influ(comp_inf2, buyfile2, cellfile2)
if 1:
savexls(comp_inf1, '附件一数据分析结果',
['公司编号', '公司名称', '信用等级', '是否违约', '经营类型(税额)', '净收益', '总支出', '总收入',
    '交易取消率',
'上游客户集中度', '下游客户集中度', '平均客户集中度'])
savexls(comp_inf2, '附件二数据分析结果',
['公司编号', '公司名称', '信用等级', '是否违约', '经营类型(税额)', '净收益', '总支出', '总收入',
    '交易取消率',
'上游客户集中度', '下游客户集中度', '平均客户集中度'])
```

```
# 问题一：企业还款额度
def ques1_2():
# 提取年份信息
comp_inf = [loadxls(4, 0, 2), loadxls(5, 0, 2)]
buyfile = [loadxls(1, 1, 1), loadxls(2, 2, 1)]
cellfile = [loadxls(1, 2, 1), loadxls(2, 1, 1)]
prof = []
for i in range(426):
prof.append([0, 0, 0, 0, 0, 0])
for now_comp in range(2):
for i, t in enumerate(buyfile[now_comp]):
if t[7] != '有效发票':
continue
year = None
if float(t[2]) >= 42736 and float(t[2]) < 43101:
year = 0
elif float(t[2]) >= 43101 and float(t[2]) < 43466:
year = 1
elif float(t[2]) >= 42466 and float(t[2]) < 43831:
year = 2
if year == None:
continue
prof[int(t[0][1:])][year] += float(t[4])
for i, t in enumerate(cellfile[now_comp]):
if t[7] != '有效发票':
continue
year = None
if float(t[2]) >= 42736 and float(t[2]) < 43101:
year = 0
elif float(t[2]) >= 43101 and float(t[2]) < 43466:
year = 1
elif float(t[2]) >= 42466 and float(t[2]) < 43831:
year = 2
if year == None:
continue
prof[int(t[0][1:])][3 + year] += float(t[4])
for i, t in enumerate(comp_inf[now_comp]):
pro18 = prof[i + 1][4] - prof[i + 1][1]
pro19 = prof[i + 1][5] - prof[i + 1][2]
if pro18 <= 0:
if pro19 <= 0:
comp_inf[now_comp][i][-1] = (-1 - pro19 / pro18)
else:
comp_inf[now_comp][i][-1] = (1 + pro19 / pro18)
else:
if pro19 <= 0:
comp_inf[now_comp][i][-1] = (-pro19 / pro18)
else:
comp_inf[now_comp][i][-1] = (pro19 / pro18)
if 1:
savexls(comp_inf[0], '附件一数据分析结果',
['公司编号', '公司名称', '信用等级', '是否违约', '经营类型(税额)', '净收益', '总支出', '总收入',
    '交易取消率',
'上游客户集中度', '下游客户集中度', '企业发展趋势'])
savexls(comp_inf[1], '附件二数据分析结果',
['公司编号', '公司名称', '信用等级', '是否违约', '经营类型(税额)', '净收益', '总支出', '总收入',
    '交易取消率',
```

```
'上游客户集中度', '下游客户集中度', '企业发展趋势'])
savexls(prof[1:], '企业各年收入支出信息', ['17支出', '18支出', '19支出', '17收入', '18收入',
    '19收入'])


# 问题一：企业情况聚类与风险评估确定
def ques1_3():
# 手肘法
comp_inf = [loadxls(4, 0, 2), loadxls(5, 0, 2)]
cho_col = [5, 6, 7, 8, 9, 10, 11]
best = [4, 5, 5, 4, 4, 4, 6,
4, 4, 4, 4, 5, 4, 4]
col_name = ['净收益', '总支出', '总收入', '交易取消率', '上游客户集中度', '下游客户集中度',
    '企业发展趋势']
for now_comp in range(2):
for col in cho_col:
d = np.zeros((len(comp_inf[now_comp]), 1))
for i, t in enumerate(comp_inf[now_comp]):
if int(comp_inf[now_comp][i][4]) == 3:
comp_inf[now_comp][i][4] = 0
elif int(comp_inf[now_comp][i][4]) == 6:
comp_inf[now_comp][i][4] = 0.25
elif int(comp_inf[now_comp][i][4]) == 9:
comp_inf[now_comp][i][4] = 0.5
elif int(comp_inf[now_comp][i][4]) == 13:
comp_inf[now_comp][i][4] = 0.75
elif int(comp_inf[now_comp][i][4]) == 16:
comp_inf[now_comp][i][4] = 1
d[i][0] = t[col]
cons = [0, 0, 0, 0, 0, 0, 0, 0]
for i in range(2, 10):
clf = KMeans(n_clusters=i)
clf = clf.fit(d)
cons[i - 2] = clf.inertia_

if best[now_comp * 7 + col - 5] != 0:
clf = KMeans(n_clusters=best[now_comp * 7 + col - 5])
clf = clf.fit(d)
lab = clf.labels_
cen = clf.cluster_centers_
cen = [float(i) for i in cen]
t = cen.copy()
t.sort()
d = {}
for i, k1 in enumerate(t):
for j, k2 in enumerate(cen):
if k1 == k2:
d[i] = j
break

for i, t in enumerate(lab):
comp_inf[now_comp][i][col] = d[t] / (best[now_comp * 7 + col - 5] - 1)
else:
plt.plot(list(range(2, 10)), cons)
plt.xlabel('k值')
plt.ylabel('质心距离平方和')
plt.title('附件' + str(now_comp + 1) + '企业' + col_name[col - 5] + 'K-Means聚类手肘图')
```

```
plt.show()
# 计算风险
wei = [0.29, 0, 0.06, -0.12, -0.12, -0.12, 0, 0.145, 0.145, -0.12]
mi = 1
ma = 0
for i in range(1):
for j, t1 in enumerate(comp_inf[i]):
sco = 0
for k, t2 in enumerate(t1[2:12]):
sco += wei[k] * t2
if sco > ma:
ma = sco
if sco < mi:
mi = sco
comp_inf[i][j].append(sco)
for j, t1 in enumerate(comp_inf[i]):
comp_inf[i][j].append(1 - (comp_inf[i][j][12] - mi) / (ma - mi))
savexls(comp_inf[now_comp], '附件' + str(now_comp + 1) + '的离散化信息',
['公司编号', '公司名称', '信用等级', '是否违约', '经营类型(税额)', '净收益', '总支出', '总收入',
    '交易取消率',
'上游客户集中度', '下游客户集中度', '企业发展趋势', '信贷风险', '归一化风险'])


# 问题一：企业最大贷款额度与贷款利率确定
def ques1_4():
# 20年现金流预测
prof = loadxls(6, 0, 2, 0, 6)
for i, t in enumerate(prof):
t1 = [t[3] - t[0], t[4] - t[1], t[5] - t[2]]
t2 = [t[3], t[4], t[5]]
mod1 = GrayFore(t1)
mod2 = GrayFore(t2)
if i == 1:
t = mod1.forecast()
plt.plot(list(range(2017, 2020)), t[:3])
plt.plot(list(range(2019, 2021)), t[2:])
plt.title('企业E' + str(i + 1) + '的2020年净收益灰度预测图')
plt.xlabel('年份')
plt.ylabel('收益额')
plt.xticks(list(range(2017, 2021)))
plt.legend(['历史数据', '预测数据'])
plt.show()
t = mod2.forecast()
plt.plot(list(range(2017, 2020)), t[:3])
plt.plot(list(range(2019, 2021)), t[2:])
plt.title('企业E' + str(i + 1) + '的2020年进项发票金额灰度预测图')
plt.xlabel('年份')
plt.ylabel('进项发票金额')
plt.legend(['历史数据', '预测数据'])
plt.xticks(list(range(2017, 2021)))
plt.show()
prof[i].append(mod1.forecast()[3] * 0.5)
prof[i].append(mod2.forecast()[3] * 0.05)
prof[i].append(max([prof[i][-1], prof[i][-2]]))

savexls(prof, '企业各年收入支出信息', ['17支出', '18支出', '19支出', '17收入', '18收入',
    '19收入', '预测收益', '预测进款额度', '最大额度'])
```

```
# 最大偿还力度预测
compinf = loadxls(7, 0, 2, 0, 14)
for i, t in enumerate(compinf):
compinf[i].append(prof[i][-1])
compinf[i].append(float(prof[i][-1]) * float(compinf[i][-2]))
if compinf[i][-1] > 1000000:
compinf[i][-1] = 1000000
elif compinf[i][-1] < 100000:
compinf[i][-1] = 100000

# 风险聚类
clf = KMeans(n_clusters=3)
d = np.zeros((len(compinf), 1))
for i, t in enumerate(compinf):
d[i][0] = t[12]
clf = clf.fit(d)
lab = clf.labels_
num = [0, 0, 0]
cen = clf.cluster_centers_
cen = [float(i) for i in cen]
t = cen.copy()
t.sort()
d = {}
for i, k1 in enumerate(t):
for j, k2 in enumerate(cen):
if k1 == k2:
d[i] = j
break
for i, t in enumerate(compinf):
compinf[i].append(int(d[lab[i]]))
num[d[lab[i]]] += 1
labels = ["A", "B", "C"]
plt.pie(x=num, labels=labels, autopct="%0.2f%%")
plt.title('第一问各企业分类饼状图')
plt.show()

# 确定三类企业的各自利率
cost = loadxls(3, 0, 2)
bestrate = [0, 0, 0]
bestpro = 0
for rate1 in range(0, 27):
for rate2 in range(rate1 + 1, 28):
for rate3 in range(rate2 + 1, 29):
rate = [rate1, rate2, rate3]
sumpro = 0
for i in compinf:
if round(i[2]) == 1:
continue
sumpro += i[15] * cost[rate[i[16]]][0] * (1 - cost[rate[i[16]]][3 - round(3 * i[2])])
if sumpro > bestpro:
bestpro = sumpro
rate = [cost[i][0] for i in rate]
bestrate = rate
print('最佳利率为:', bestrate)
print('最多获利为:', bestpro)
```

```
d = []
for i in compinf:
if i[2] == 1:
e = 0
l = 0
else:
e = i[15]
l = bestrate[2 - i[16]]
d.append([i[0], i[1], e, l])

savexls(compinf, '附件1的离散化信息',
['公司编号', '公司名称', '信用等级', '是否违约', '经营类型(税额)', '净收益', '总支出', '总收入',
    '交易取消率',
'上游客户集中度', '下游客户集中度', '企业发展趋势', '信贷风险', '归一化风险', '最大额度',
    '实际额度', '风险聚类'])
savexls(d, '问题一贷款策略', ['公司编号', '公司名称', '实际额度', '贷款利率'])


# 问题二：决策树代码
def ques2_1():
comp_inf1 = loadxls(7, 0, 2, 2)
comp_inf2 = loadxls(8, 0, 2)
grades = 'D', 'C', 'B', 'A'
all_f = ['broken', 'business type', 'profit', 'outcome', 'income', 'transaction cancellation
    rate',
'upstream influence', 'downstream influence', 'develop tendency']
data = np.array(comp_inf1)
for i, t, in enumerate(data):
data[i][0] = np.round(t[0] * 3)

def gettree(cho, maxstep, minsplit, flag):
d = data[:, cho]
f = []
for i in cho:
f.append(all_f[i - 1])

# 生成训练集与测试集
# fr, to = 92, 106
# c = list(range(0, fr + 1))
# c.extend(list(range(to - 1, 123)))
# xtes = d[c, :]
# ytes = data[c, 0]
# xtra, ytra = d, data[:, 0]
xtra, xtes, ytra, ytes = train_test_split(d, data[:, 0], test_size=0.2)
clf = tree.DecisionTreeClassifier(criterion='entropy'
, max_depth=maxstep
, min_samples_split=minsplit
)
clf.fit(xtra, ytra)

# 决策树示意图
file = tree.export_graphviz(clf, out_file=None, feature_names=f, class_names=grades,
filled=True, rounded=True, special_characters=True)
f = pydotplus.graph_from_dot_data(file)
f.write_pdf('tree' + str(flag) + '.pdf')

print('各个特征的影响力:', clf.feature_importances_)
```

```
# 训练集验证
answer = clf.predict(xtra)
ytra = ytra.reshape(-1)
num = 0
corr = 0
realcorr = 0
dincorr = 0
dnum = 0
for i, t in enumerate(answer):
num += 1
if t == 3:
dnum += 1
if t != ytra[i]:
dincorr += 1
if abs(t - ytra[i]) <= 1:
corr += 1
if abs(t - ytra[i]) == 0:
realcorr += 1
print('训练集准确度:', corr / num)
if dnum!=0:
t=dincorr / dnum
else:
t=0
print('训练集D等信誉错误率:', t)
print('训练集平均偏差:', np.mean(np.abs(answer - ytra)) / 3)

# 测试集验证
answer = clf.predict(xtes)
ytes = ytes.reshape(-1)
num = 0
corr = 0
dincorr = 0
dnum = 0
for i, t in enumerate(answer):
if t == 3:
dnum += 1
if t != ytes[i]:
dincorr += 1
num += 1
if abs(t - ytes[i]) <= 1:
corr += 1
print('测试集准确度:', corr / num)
if dnum!=0:
t=dincorr / dnum
else:
t=0
print('测试集D等信誉错误率:', t)
print('测试集平均偏差:', np.mean(np.abs(answer - ytes)) / 3)
return clf

# 初步实现决策树
cho1 = [2, 3, 4, 5, 6, 7, 8, 9]
clf = gettree(cho1, maxstep=6, minsplit=10, flag=1)
# 降维简化数据
rer = []
rep = []
```

```
for col in range(2, 10):
r, p = stats.pearsonr(data[:, 0], data[:, col])
rer.append(r)
rep.append(p)
print('各列相关度为:', end='')
for i in rer:
print('%.3f' % (i), end='\t')
print(' ')
print('各列显著性为:', end='')
for i in rep:
print('%.3f' % (i), end='\t')
print(' ')
# 剔除
cho2 = [4, 5, 6, 7, 8, 9]
gettree(cho2, maxstep=4, minsplit=15, flag=2)
# 预测
d = []
for i in comp_inf2:
t = []
for j in i[2:]:
if j == '':
t.append(0)
else:
t.append(float(j))
d.append(t)
data = np.array(d)
d = data[:, cho1]
ans = clf.predict(d)
for i, t in enumerate(ans):
comp_inf2[i][2] = t / 3
comp_inf2[i][3] = 0
savexls(comp_inf2, '附件2的离散化信息',
['公司编号', '公司名称', '信用等级', '是否违约', '经营类型(税额)', '净收益', '总支出', '总收入',
    '交易取消率',
'上游客户集中度', '下游客户集中度', '企业发展趋势', '信贷风险', '归一化风险', '最大额度',
    '实际额度', '风险聚类'])


def ques2_2():
comp_inf = loadxls(8, 0, 2)
prof = loadxls(6, 0, 2)
# 计算风险与最大额度
wei = [0.29, 0, 0.06, -0.12, -0.12, -0.12, 0, 0.145, 0.145, -0.12]
mi = 1
ma = 0
for i, t1 in enumerate(comp_inf):
sco = 0
for k, t2 in enumerate(t1[2:12]):
sco += wei[k] * t2
if sco > ma:
ma = sco
if sco < mi:
mi = sco
comp_inf[i][12] = sco
for i, t1 in enumerate(comp_inf):
comp_inf[i][13] = (1 - (comp_inf[i][12] - mi) / (ma - mi))
comp_inf[i][14] = float(prof[i + 123][-1])
```

```
comp_inf[i][15] = comp_inf[i][14] * float(comp_inf[i][13])
if comp_inf[i][15] > 1000000:
comp_inf[i][15] = 1000000
elif comp_inf[i][15] < 100000:
comp_inf[i][15] = 100000

# 风险聚类
clf = KMeans(n_clusters=3)
d = np.zeros((len(comp_inf), 1))
for i, t in enumerate(comp_inf):
d[i][0] = t[12]
clf = clf.fit(d)
lab = clf.labels_
num = [0, 0, 0]
cen = clf.cluster_centers_
cen = [float(i) for i in cen]
t = cen.copy()
t.sort()
d = {}
for i, k1 in enumerate(t):
for j, k2 in enumerate(cen):
if k1 == k2:
d[i] = j
break
for i, t in enumerate(comp_inf):
comp_inf[i][16] = (int(d[lab[i]]))
num[d[lab[i]]] += 1
labels = ["A", "B", "C"]
plt.pie(x=num, labels=labels, autopct="%0.2f%%")
plt.title('第二问各企业分类饼状图')
plt.show()

# 确定三类企业的各自利率
cost = loadxls(3, 0, 2)
bestrate = [0, 0, 0]
bestpro = 0
bestmon = 0
for rate1 in range(0, 27):
for rate2 in range(rate1 + 1, 28):
for rate3 in range(rate2 + 1, 29):
rate = [rate1, rate2, rate3]
sumpro = 0
summon = 0
for i in comp_inf:
if round(i[2]) == 1:
continue
sumpro += i[15] * cost[rate[i[16]]][0] * (1 - cost[rate[i[16]]][3 - round(3 * i[2])])
summon += i[15] * (1 - cost[rate[i[16]]][3 - round(3 * i[2])])
if summon <= 1e8 and sumpro > bestpro:
bestpro = sumpro
rate = [cost[i][0] for i in rate]
bestrate = rate
bestmon = summon
print('最佳利率为:', bestrate)
print('最多获利为:', bestpro)
print('交易贷款总额为:', bestmon)
```

```
d = []
for i in comp_inf:
if i[2] == 1:
e = 0
l = 0
else:
e = i[15]
l = bestrate[2 - i[16]]
d.append([i[0], i[1], e, l])

savexls(d, '问题二贷款策略', ['公司编号', '公司名称', '实际额度', '贷款利率'])
savexls(comp_inf, '附件2的离散化信息',
['公司编号', '公司名称', '信用等级', '是否违约', '经营类型(税额)', '净收益', '总支出', '总收入',
    '交易取消率',
'上游客户集中度', '下游客户集中度', '企业发展趋势', '信贷风险', '归一化风险', '最大额度',
    '实际额度', '风险聚类'])


# 计算考虑疫情影响和政府扶持下的各企业分数
def ques3_1(b):
comp_inf = loadxls(8, 0, 2)
n = len(comp_inf[0])
clus = [['-0.5', '劳务', '个体经营', '影', '五金', '设计服务', '策划', '设计', '传播', '广告',
    '印务', '餐饮', '服务', '图书', '经营',
'童装', '纺织品', '生活用品', '鞋', '服饰', '体育', '家居', '食品', '工贸', '文化', '贸易'],
['-0.25', '物资', '运', '物流', '汽贸', '快递', '商贸'],
['0', '建筑', '管理', '维修', '教育', '质量', '律师', '事务', '实业', '环保', '地质', '灾害',
    '土地', '生态', '猕猴桃', '园艺',
'园林', '石化', '天然气', '装饰', '农业', '调味品', '蔬菜', '建设', '房地产', '建材', '包装',
    '材', '木业', '纸业', '塑胶',
'卫浴', '门窗', '设备', '空调', '家电', '器材', '电器', '机电', '花', '轮胎', '化工', '机械',
    '车', '工程', '电气', '合金',
'塑料'],
['0.25', '投资', '代理'],
['0.5', '发展', '科学', '电子', '技术', '网络', '医疗', '药', '科技']
]
for i, t in enumerate(comp_inf):
for x in range(22 - len(comp_inf[i])):
comp_inf[i].append(0)
flag = False
for k, c in enumerate(clus):
if flag:
break
for j in c:
if j in t[1]:
comp_inf[i][17] = float(clus[k][0])
comp_inf[i][19] = (b * comp_inf[i][17] - comp_inf[i][13] + 0.5) / (b + 1) + 0.5
comp_inf[i][18] = (b * comp_inf[i][17] + comp_inf[i][13] - 0.5) / (b + 1) + 0.5
comp_inf[i][20] = comp_inf[i][14] * float(comp_inf[i][19])
if comp_inf[i][20] > 1000000:
comp_inf[i][20] = 1000000
elif comp_inf[i][20] < 100000:
comp_inf[i][20] = 100000
flag = True
break
# 风险聚类
clf = KMeans(n_clusters=5)
```

```
d = np.zeros((len(comp_inf), 1))
for i, t in enumerate(comp_inf):
d[i][0] = t[18]
clf = clf.fit(d)
lab = clf.labels_
num = [0, 0, 0, 0, 0]
cen = clf.cluster_centers_
cen = [float(i) for i in cen]
t = cen.copy()
t.sort()
d = {}
for i, k1 in enumerate(t):
for j, k2 in enumerate(cen):
if k1 == k2:
d[i] = j
break
for i, t in enumerate(comp_inf):
comp_inf[i][21] = (int(d[lab[i]]))
num[d[lab[i]]] += 1
labels = ["A", "B", "C", "D", "E"]
plt.pie(x=num, labels=labels, autopct="%0.2f%%")
plt.title('第三问各企业分类饼状图')
plt.show()

# 确定三类企业的各自利率
cost = loadxls(3, 0, 2)
bestrate = [0, 0, 0, 0, 0]
bestpro = 0
bestmon = 0
for rate1 in range(0, 25):
for rate2 in range(rate1 + 1, 26):
for rate3 in range(rate2 + 1, 27):
for rate4 in range(rate3 + 1, 28):
for rate5 in range(rate4 + 1, 29):
rate = [rate1, rate2, rate3, rate4, rate5]
sumpro = 0
summon = 0
for i in comp_inf:
if round(i[2]) == 1:
continue
sumpro += i[20] * cost[rate[i[21]]][0] * (1 - cost[rate[i[21]]][3 - round(3 * i[2])])
summon += i[20] * (1 - cost[rate[i[21]]][3 - round(3 * i[2])])
if summon <= 1e8 and sumpro > bestpro:
bestpro = sumpro
rate = [cost[i][0] for i in rate]
bestrate = rate
bestmon = summon
print('最佳利率为:', bestrate)
print('最多获利为:', bestpro)
print('交易贷款总额为:', bestmon)

d = []
for i in comp_inf:
if i[2] == 1:
e = 0
l = 0
else:
```

```
e = i[20]
l = bestrate[4 - i[21]]
d.append([i[0], i[1], e, l])

if b==2:
savexls(d, '问题三贷款策略', ['公司编号', '公司名称', '实际额度', '贷款利率'])
savexls(comp_inf, '附件2的离散化信息',
['公司编号', '公司名称', '信用等级', '是否违约', '经营类型(税额)', '净收益', '总支出', '总收入',
    '交易取消率', '上游客户集中度',
'下游客户集中度', '企业发展趋势', '信贷风险', '归一化风险', '最大额度', '实际额度', '风险聚类',
    '影响值', '分数1', '分数2',
'考虑疫情的额度', '分数聚类'])
return bestpro


ques1_1()
ques1_2()
ques1_3()
ques1_4()
ques2_1()
ques2_2()
ques3_1(2)
# 灵敏度分析
i = 1
lmd = []
x = []
while (i <= 3):
print(i)
x.append(i)
lmd.append(ques3_1(i))
i += 0.125
plt.plot(x, lmd)
plt.xlabel('疫情影响权重比')
plt.ylabel('银行最大获利')
plt.title('疫情影响权重比与银行最大获利的灵敏度分析曲线图')
plt.show()
```