赛区评阅编号（由赛区组委会填写）：

**2020年高教社杯全国大学生数学建模竞赛**

**承 诺 书**

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称 “竞赛章程和参赛规则”，可从http://www.mcm.edu.cn下载）。

我们完全清楚，在竞赛开始后参赛队员不能以任何方式，包括电话、电子邮件、“贴吧”、QQ群、微信群等，与队外的任何人（包括指导教师）交流、讨论与赛题有关的问题；无论主动参与讨论还是被动接收讨论信息都是严重违反竞赛纪律的行为。

我们完全清楚，在竞赛中必须合法合规地使用文献资料和软件工具，不能有任何侵犯知识产权的行为。否则我们将失去评奖资格，并可能受到严肃处理。

**我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。**

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从A/B/C/D/E中选择一项填写）：

我们的报名参赛队号（12位数字全国统一编号）：

参赛学校（完整的学校全称，不含院系名）：

参赛队员 (打印并签名) ：1.

2.

3.

指导教师或指导教师组负责人 (打印并签名)：

（指导教师签名意味着对参赛队的行为和论文的真实性负责）

日期： 年 月 日

**（请勿改动此页内容和格式。此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）**

赛区评阅编号： 全国评阅编号：

(由赛区填写） （全国组委会填写）

**2020年高教社杯全国大学生数学建模竞赛**

**编 号 专 用 页**

赛区评阅记录（可供赛区评阅时使用）：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 评  阅  人 |  |  |  |  |  |  |
| 备  注 |  |  |  |  |  |  |

送全国评阅统一编号:

（赛区组委会填写）

**（请勿改动此页内容和格式。此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页。）**

中小微企业的信贷决策

## 摘要

本文针对银行对小微企业的信贷策略展开研究，通过调查研究影响银行贷款风险的各个因素，建立银行信贷风险与企业各项指标间关系的数学模型，并将各个指标进行量化计算，进而为企业科学的信贷策略的制定提供合理科学的参考。本文通过建立一个最为核心的“影响因素层级图”并据此建立各个模型；利用python的Numpy、pandas库对附件中的数据进行挖掘、分析；matplotlib库绘制相关图形；利用Spssau在线网站对数据进行各项科学分析，进而求解各个模型。

问题一中，重点在于建立模型合理评估企业信贷风险以及量化计算各个指标。本文建立“影响因素层级图”，并据此确定了各个指标的相互依赖关系，进一步逐层地建立数学模型。对于信贷风险的量化计算，本文采用了熵权法；并且对于影响信贷风险的三个子指标采用AHP层次分析法进行量化计算。对于最底层的一些指标，如：信誉评级、是否违约等，本文都对它们进行了归一化处理，以便于后续计算。最后，根据企业的信贷风险和附件3中的信息建立银行贷款策略模型并求解。

问题二中，由于缺少信贷记录，而无法直接使用问题一中的模型。需要先根据月均订单量、月均利率、销进作废占比等信息建立信誉预测模型，采用二元logit回归对各指标进行回归分析求解得出回归方程，回归方程用于求解企业的违规概率，通过企业的违约概率来衡量企业的信誉评级，进而求出信誉值。之后再使用问题一中的模型给出贷款策略建议。

问题三中，新增影响指标：突发因素。本文认为突发因素表现在对企业的实力的影响，进而影响了信贷风险。于是我们以新冠疫情为例，收集各行业受到的影响数据，量化该指标；并且对所有的企业分类，用“对行业的影响”近似估计“对具体企业的影响”，从而求出新的企业实力并代入前面的模型。

## 关键词：AHP层次分析；拟合；熵值法；logit回归分析；信贷策略

## 一 问题重述

### 1.1 问题背景

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

某银行对确定要放贷企业的贷款额度为万元；年利率为4%~15%；贷款期限为1年。附件1~3分别给出了123家有信贷记录企业的相关数据、302家无信贷记录企业的相关数据和贷款利率与客户流失率关系的2019年统计数据。该银行请你们团队根据实际和附件中的数据信息，通过建立数学模型研究对中小微企业的信贷策略。

### 1.2 求解问题

本文作者依据附件中的大量数据，并利用python中的Numpy、pandas等工具库对其进行数据挖掘和数据分析，并建立数学模型解决了以下问题：

(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。

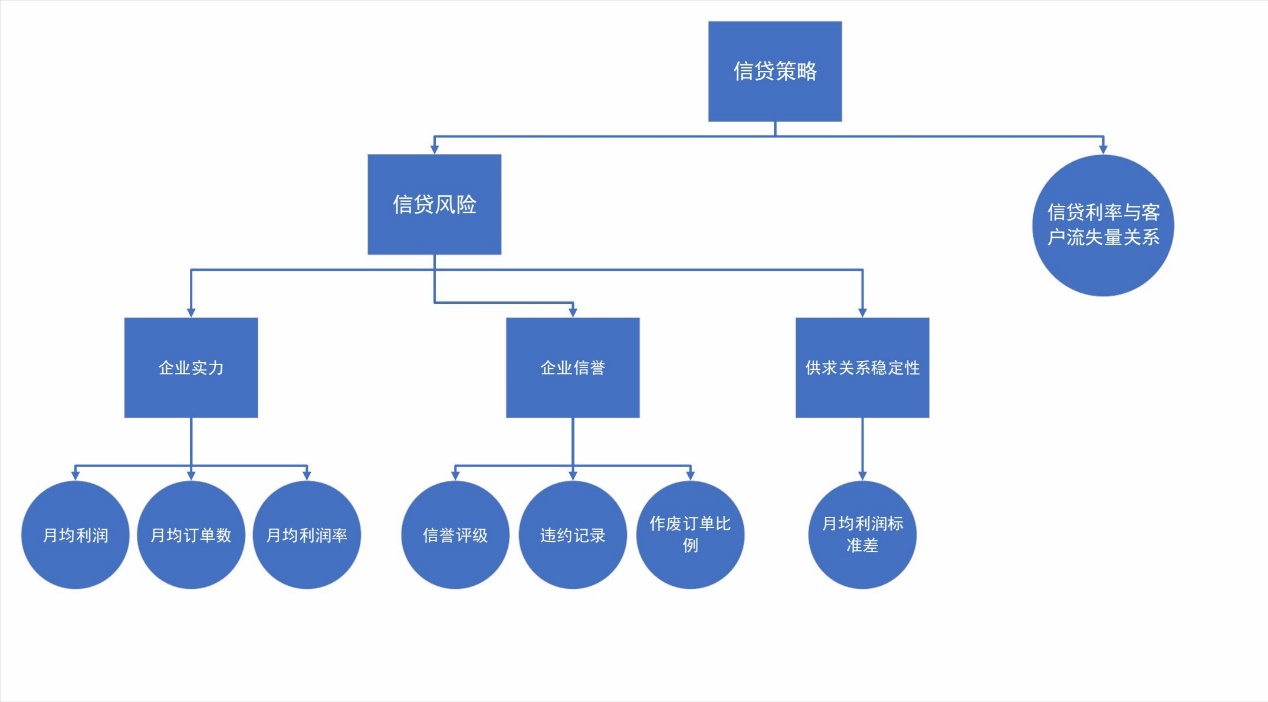
(2) 在问题1的基础上，对附件2中302家企业的信贷风险进行量化分析，并给出该银行在年度信贷总额为1亿元时对这些企业的信贷策略。

(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。

## 二 问题分析

### 2.1 问题一的分析

根据题目的要求以及附件中所提供的信息，可以发现此问题中的影响因素较多且具有明显的层次机构，较高层的因素由较低层的因素决定，而最底层的因素，如：企业的信誉评级、企业的销售额等则来自于附件。因此，本文为此问题设计了“影响因素层级图”



图表 1 影响因素层级图

图中圆框内的因素便可以从附件中通过挖掘、分析来得到。因此需要逐层建立数学模型来量化地求出各个企业的信贷风险值，并据此制订银行的信贷策略。

同时，由于此问题数据量非常大，因此本文选择python中的Numpy和pandas库进行数据筛选与分析，并实现各个影响因素的量化。

此外，由于此问题没有统一的量纲，故本文对上述层级图中涉及的全部影响因素都进行“归一化”量化处理，以方便比较与计算，归一化公式：



### 2.2 问题二的分析

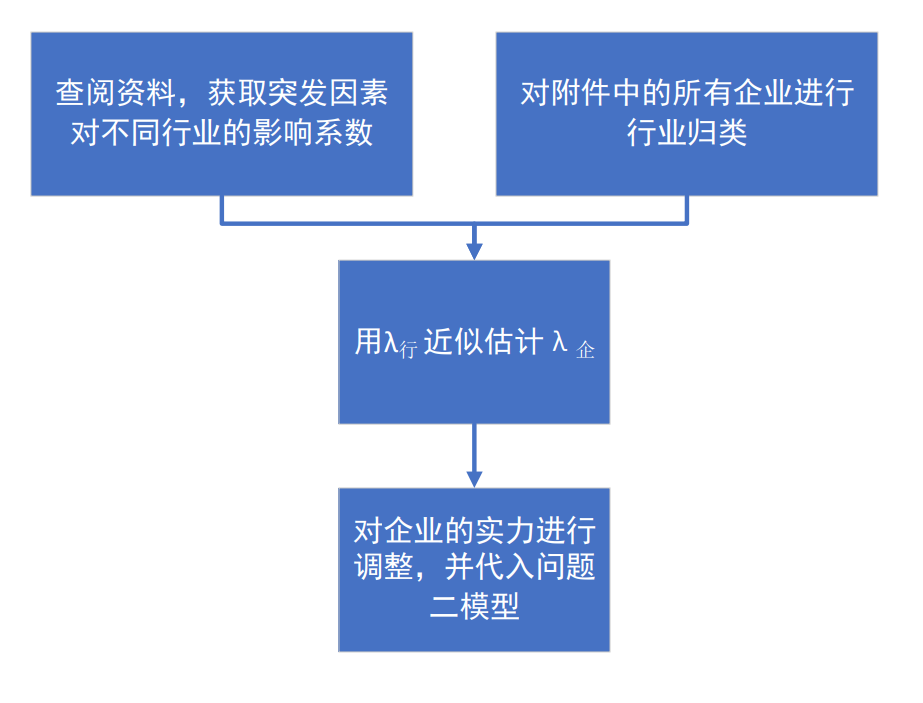
第二问比较第一问差别在于无企业信贷记录的缺少。需要依据其他变量来合理的预测无信贷记录的企业的信誉值。

依据附件1中的数据，对违约与否映射为1和0，再对月均订单数、销进作废发票占比、销进负单占比、月均利润和月均利润额进行回归分析，从而求得回归方程用于预测各个企业的违约率，再根据违约率预测信誉评级，得到这些数据之后再套用问题一中的信誉评价模型算出信誉值，进而可以利用信贷风险评价模型得出信贷风险值，给出信贷策略建议

### 2.3 问题三的分析

此问题其实就是在问题二的基础上对模型进行调整，再多考虑一个影响因素：“突发因素对各企业的影响系数”，记做。本文假定只影响企业实力，据此求出。再将代入问题二的模型中即可得解。

考虑到企业的代号与名称繁杂，直接处理很困难。于是借助概率论与数理统计的知识，本文使用“突发因素对各行业的影响系数”来近似估计。思路图如下



图表 2 问题三的思路

## 三 问题假设

1.假设附件1中数据具有一般代表性

2.假设企业交易发票真实有效

3.假设各个信誉等级的企业占比反应一般水平

4.假设在突发因素发生后，本行业的总体利润率短期内波动，长远来看趋向正常

5.假设企业实力和企业信誉之间没有必然关系

## 四 符号说明

注：下表中所有需要“归一化”的变量都会进行“归一化”处理

|  |  |
| --- | --- |
| 变量 | 说明 |
|  | 银行信贷策略 |
|  | 企业信贷风险值 |
|  | 企业供求稳定性值 |
|  | 企业信誉值 |
|  | 信誉评级值 |
|  | 违约记录值 |
|  | 作废发票比值 |
|  | 企业实力值 |
|  | 月均利润值 |
|  | 月均发票值 |
|  | 月均利润率 |
|  | 熵权法中第个指标的熵值 |
|  | 熵权法中第个指标的熵权 |
|  | 熵权法中的初始矩阵 |
|  | AHP中第个指标的权重 |
|  | 对进行归一化处理后的值 |
| , , | 信誉等级分别为A,B,C的企业的贷款利率 |
| , , | 信誉等级分别为A,B,C的客户流失率 |
|  | 企业代号为的公司可贷款额度 |
|  | 银行贷款收益 |
|  | 突发因素对企业的影响系数 |
|  | 突发因素对行业的影响系数 |

## 五 模型建立与求解

### 5.1 问题一

##### 5.1.1 企业信誉评价模型

(1)建立

在本文设计的“影响因素层级图中”，企业信誉受“信誉评级”、“违约记录”和“作废发票占比”影响，分别记为，且这三个子因素对的影响程度明显不同，且彼此之间的相对重要性也不同，因此本文运用AHP层次分析法将定量分析与定性分析结合起来，通过请教相关领域的专家的经验判断之间的相对重要程度，并合理地给出，从而求得

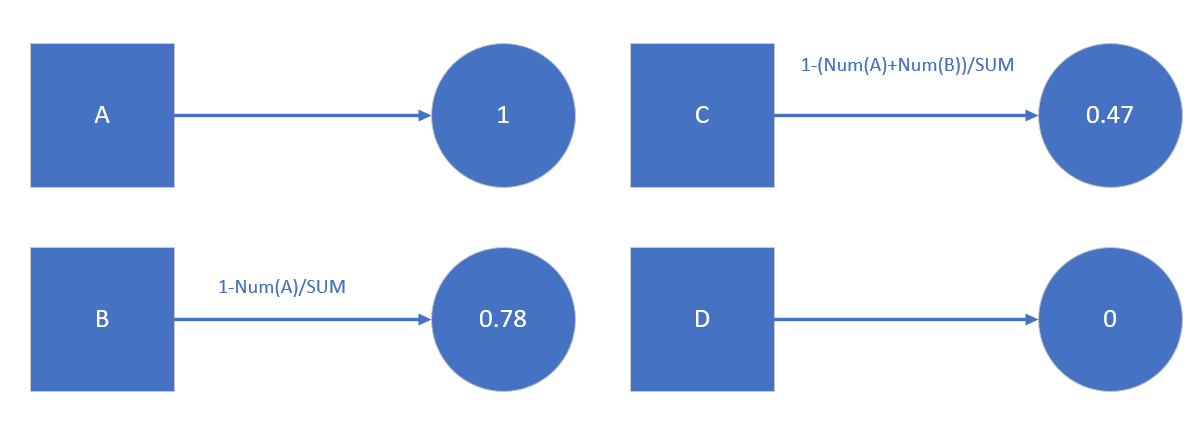


此模型利用权重求出的优劣次序，能够有效地用定量方法求出企业的信誉值。

(2)数据处理

**Step1**

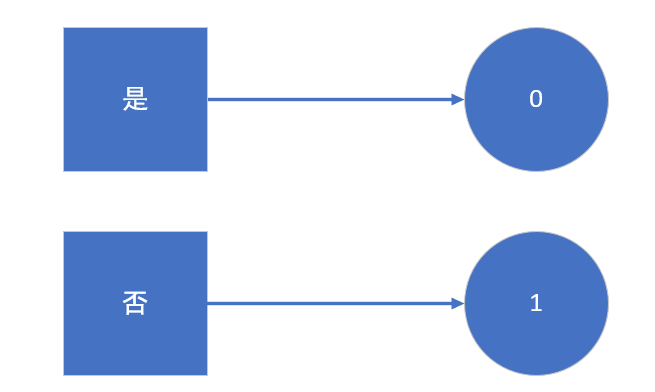
附件1中给出了对各个企业的信誉评级，本文依据各个等级所占的百分比构造一个映射，从而完成对信誉评级的量化



图表 3 信誉评级量化映射

**Step2**

对于违约记录,因为它与企业信誉值是负相关，故做映射



图表 4 违约记录量化映射

**Step3**

本文认为企业的作废发票比率与企业信誉值负相关，故



###### (3)求解

**Step1**

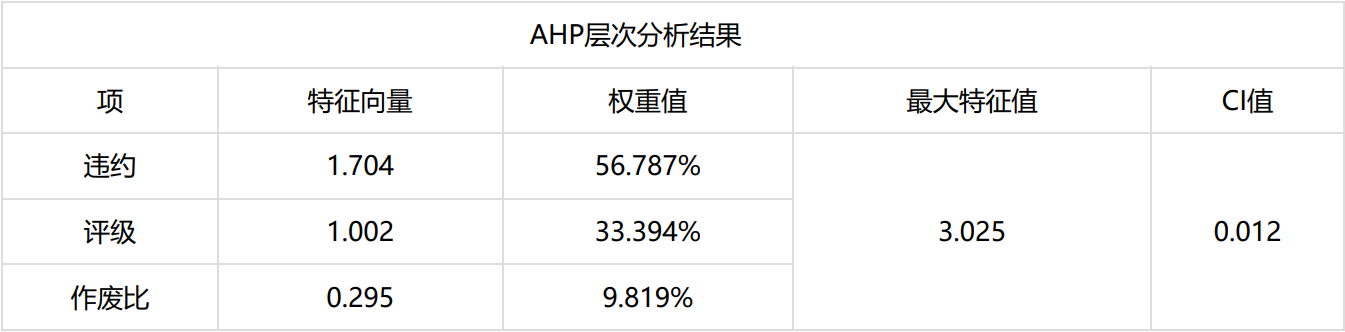
本文咨询了10位领域内专家，确立了此三个因素的相互重要性矩阵



图表 5 信誉影响因素相互重要性矩阵

**Step2**

利用SPSSAU数据科学在线网站进行“AHP层次分析”，即可求得



图表 6 信誉影响因素AHP分析

同时进行重要性矩阵一致性检测



图表 7 矩阵一致性检验结果

**Step3**

利用python中的Numpy和pandas库编程计算出某一个企业信誉值

最后对进行归一化处理即可



##### 5.1.2 企业实力评价模型

###### (1)建立与求解方法

在本文设计的“影响因素层级图中”，企业实力受影响，同样运用运用AHP层次分析法即可求出

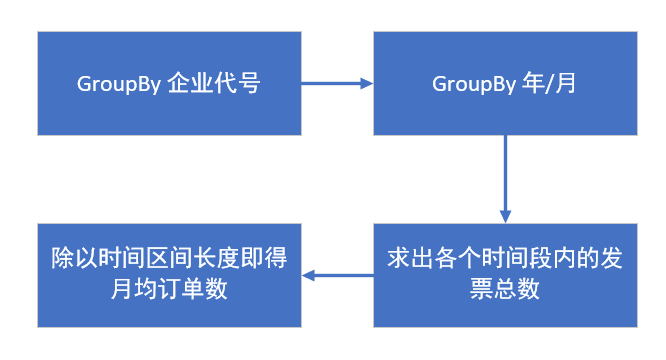


此模型与前文相同，不多赘述。

###### (2)数据处理

此模型的困难之处在于数据处理。衡量企业实力的主要因素有三大因素：月均订单数，月均利润及月均利润率。在python中使用pandas库对数据进行筛选及处理

**Step1**



图表 8 计算月均订单步骤

**Step2 & Step3**

同上步骤即可

##### 5.1.3 企业供求稳定性评价模型

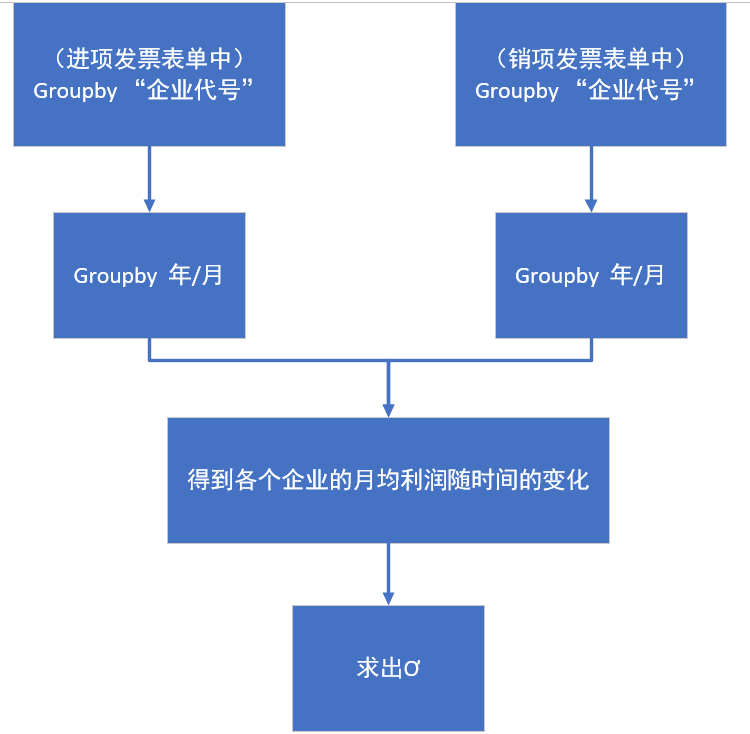
(1)建立

考虑到企业的供求稳定性也是影响信贷风险的重要因素之一，而常用的统计量“标准差”正好可以反应数据的波动情况；此外，月均利润能够很好地反应企业的收益情况。综上，本文使用“企业的月均利润标准差”作为衡量“企业供求稳定性”的指标，且二者成负相关，故





(2)数据处理



图表 9 求月均利润标准差

(3)求解

对经过处理过后的销项和进项数据，用销项的价税合计—进项的价税合计得出利润一列，再对利润一列使用numpy.std()函数求出标准差，日期单位为月。

##### 5.1.4 企业信贷风险评价模型

(1)建立

在我们设计的“印象因素层级图”中，“信贷风险”受“企业信誉”“企业实力”“企业供求稳定性”影响，分别记作，与前文的AHP层次分析法不同，这里的它们彼此之间的相对重要性不容易确定，因此我们采用另一种多指标综合评价方法——熵权法来求。

熵权法实际上是对指标重要程度描述的一种方法，该方法以熵理论为核心。事件发生概率越高，相应会表现出更为突出的有序性，而有序程度越高，则其信息熵取值更大，相应的，其权重参数取值更小。基于此，在综合评价中如需应用多个指标，则可借助该方法完成指标赋权；此外，此方法是客观赋值法，即根据指标自身的数据分布特点即可求出结果。

**Step1**

构建一个矩阵



表示个企业的风险值矩阵。

**Step2**

将不同的指标分别归一化处理



得到新矩阵

**Step3**

计算各个指标的熵值







**Step4**

求出指标权重



**Step5**

求出综合评价的风险值



(2)数据处理

求解此模型所需要的三个指标在前面已经求出归一化的值，故此处无需处理

(3)求解

利用前文的熵权法步骤可求出三个指标各自的权重

| 熵值法计算权重结果汇总 | | | |
| --- | --- | --- | --- |
| 项 | 信息熵值e | 信息效用值d | 权重系数w |
| credit归一化 | 0.9934 | 0.0066 | 56.42% |
| 稳定性 | 0.9979 | 0.0021 | 17.91% |
| power | 0.9970 | 0.0030 | 25.67% |

图表10熵权法权重

对，从而计算出风险值



##### 5.1.5 银行信贷利率确定

(1)处理思路

根据附录3中的数据，我们分别可以对信誉等级为A，B，C的三类企业的贷款利率（）和客户流失率（）进行多项式拟合。可通过Numpy库得到n阶拟合函数



再通过计算下列银行放贷收益（）公式，求出在贷款区间4%到15%中的最大收益求出A,B,C三个信贷评级的最佳贷款利率



即 -------公式（1）

对求偏导，求出在4%到15%间的极小值点，若无极小值则直接取区间两端的点。

(2)数据处理

通过pandas库读入数据，将数据转成列表，R列表对应贷款利率，T列表对应客户流失率。

(3)求解

###### Step1

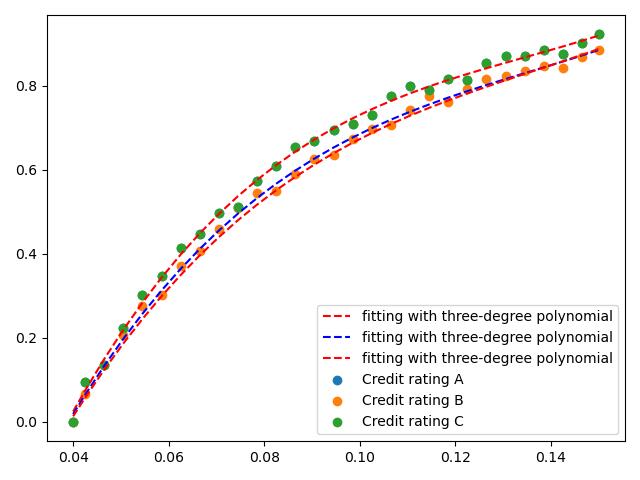
进行数据拟合，得出函数表达式

使用Numpy库中的.polyfit()函数对R列表和T列表进行拟合，最高位阶数为3。用matplotlib库进行绘图，绘制拟合后的函数图像，并打印输出A，B，C三个等级的函数表达式

 ---------公式（2）

 ---------公式（3）

 ---------公式（4）



图表11 三个等级的拟合后函数图像

###### Step2

使用python对上述公式（1）进行求解，将公式（2），（3），（4）代入公式（1），利用循环，取步长为1，区间[0,10000]，归一化处理后就是步长为0.0001，区间[0,1]，最后得出

| 信誉评级 | 最佳贷款利率 | 用户流失率 |
| --- | --- | --- |
| A | 4.70% | 3.95% |
| B | 5.18% | 4.06% |
| C | 5.38% | 4.12% |

图表 12 不同信誉评级的最佳贷款利率

##### 5.1.6 银行信贷策略模型

(1)建立

依照我们之前建立的“影响因素层级图”，企业风险值和“信贷利率与客户流失量的关系”决定银行的信贷策略。

前文已经求出了最佳贷款利率，现在需要确定贷款额度。而贷款额度可以根据企业的风险值在总体的排名百分比来确定，即

贷款额度（）：

 --------------------------公式（1）

注意到题目要求银行的贷款额度必须在区间内，若上述函数求得的不在此区间内，则对其进行调整：





贷款利率根据信誉评级：

| 信誉评级 | 贷款利率 |
| --- | --- |
| A | 4.70% |
| B | 5.18% |
| C | 5.38% |

(2)数据处理

将之前处理好的风险值表格进一步处理，给出银行的放贷总额度，利用上述公式（1）求解即可得出银行给某个具体企业的贷款额度，加上前文已经确定的贷款利率，贷款策略即得解。

### 5.2 问题二

##### 5.2.1 违约预测模型

(1)违约预测

依据附件1中的信息，不同信誉评级的企业中违约的企业的占比不同，以假设附件1中数据具有一般代表性，则可以通过违约占比刻画不同评级企业的违约概率。反之，可以用违约概率来预测信誉评级，再用上述信誉评价模型求出。

| 信誉评级 | 违约数 | 评级数 | 占比 |
| --- | --- | --- | --- |
| A | 0 | 27 | 0 |
| B | 1 | 38 | 1/38 |
| C | 2 | 34 | 1/17 |
| D | 24 | 24 | 1 |

(2)建立

我们需要根据附件1中的相关信息建立起违约预测的模型，考虑附件1中的各个信息：销进项负单占比，销进项作废发票占比，月均利润，月均利润率，月均订单数等为自变量；以违约量化为因变量，违约即为1，无违约则为0。

这正好可以使用二元logit回归模型对各指标进行相关性分析，求得回归系数和p值。

(3)结果分析

从上表10可知，将月均订单数, 销项发票作废比, 进项负单比, 销项负单比, 进项发票作废比,月均利润率, 月均利润共7项为自变量，而将违约量化作为因变量进行二元Logit回归分析，模型公式为：



(其中代表违约量化为1 的概率，1-代表违约量化为0的概率)。

| 二元Logit回归分析结果汇总 | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| 项 | 回归系数 | 标准误 | *z* 值 | *p* 值 | OR值 | OR值95% CI |
| 月均订单数 | -0.125 | 0.044 | -2.802 | 0.005 | 0.883 | 0.809 ~ 0.963 |
| 销项发票作废比 | 5.587 | 2.598 | 2.150 | 0.032 | 266.886 | 1.640 ~ 43427.559 |
| 进项负单比 | -6.920 | 21.572 | -0.321 | 0.748 | 0.001 | 0.000 ~ 2275419859979014.500 |
| 销项负单比 | -2.405 | 5.429 | -0.443 | 0.658 | 0.090 | 0.000 ~ 3770.933 |
| 月均利润率 | -0.001 | 0.002 | -0.609 | 0.543 | 0.999 | 0.995 ~ 1.003 |
| 月均利润 | -0.000 | 0.000 | -2.121 | 0.034 | 1.000 | 1.000 ~ 1.000 |
| 进项发票作废比 | 1.769 | 12.686 | 0.139 | 0.889 | 5.866 | 0.000 ~ 368690085895.586 |
| 截距 | -0.511 | 0.649 | -0.788 | 0.431 | 0.600 | 0.168 ~ 2.138 |
| 因变量: 违约量化 | | | | | | |
| McFadden *R* 方: 0.333 | | | | | | |
| Cox & Snell *R* 方：0.295 | | | | | | |
| Nagelkerke *R* 方：0.454 | | | | | | |

图表 13 二元logit回归分析结果汇总

根据总结分析可知：销项发票作废比会对违约量化产生显著的正向影响关系，以及月均订单数, 月均利润会对违约量化产生显著的负向影响关系。但是进项负单比, 销项负单比, 月均利润率, 进项发票作废比并不会对违约量化产生影响关系。

| 二元Logit回归预测准确率汇总 | | | | | |
| --- | --- | --- | --- | --- | --- |
|  | | 预测值 | | 预测准确率 | 预测错误率 |
| 0 | 1 |
| 真实值 | 0 | 92 | 4 | 95.83% | 4.17% |
| 1 | 14 | 13 | 48.15% | 51.85% |
| 汇总 | | | | 85.37% | 14.63% |

图表 14 二元logit回归预测准确率汇总

通过模型预测准确率去判断模型拟合质量，从上表11可知：研究模型的整体预测准确率为85.37%，模型拟合情况可以接受。当真实值为0时，预测准确率为95.83%；另外当真实值为1时，预测准确率为48.15%

(4)求解

通过建立模型，可以得到模型公式为：

ln(p/1-p)=-0.511-0.125\*月均订单数 + 5.587\*销项发票作废比-6.920\*进项负单比-2.405\*销项负单比-0.001\*月均利润率-0.000\*月均利润 + 1.769\*进项发票作废比

即：



从而可求：



即为违约概率，再根据图9中给出的违约数占比，可以评定该公司的信誉评级，进而套用上述企业信誉评价模型给出预测的企业信誉值。

### 5.3 问题三

##### 5.3.1 突发因素影响模型

(1)建立

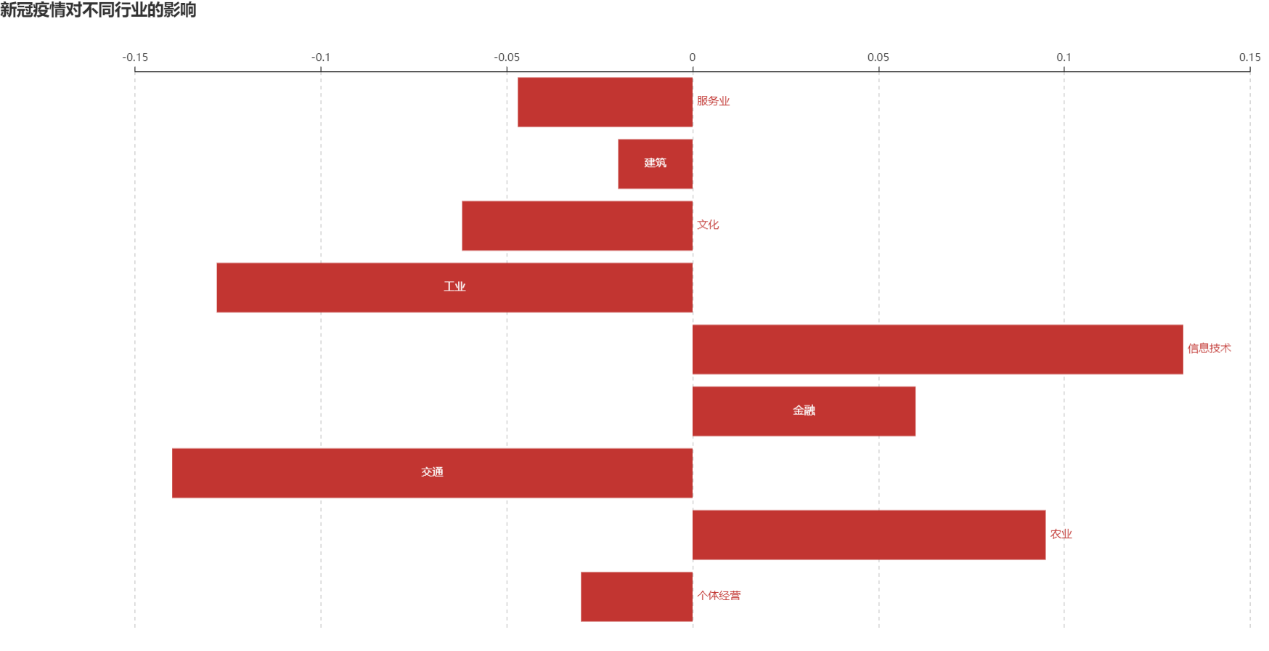
以“突发因素对行业的影响系数”近似估计“突发因素对企业的影响系数”，假设只会对企业实力造成影响，不影响企业信誉和企业稳定性。



再将代入问题二的模型即得解

(2)数据处理

我们在国家统计局官网中搜集新冠疫情期间的大量统计数据，得到对不同行业利润的增减影响情况

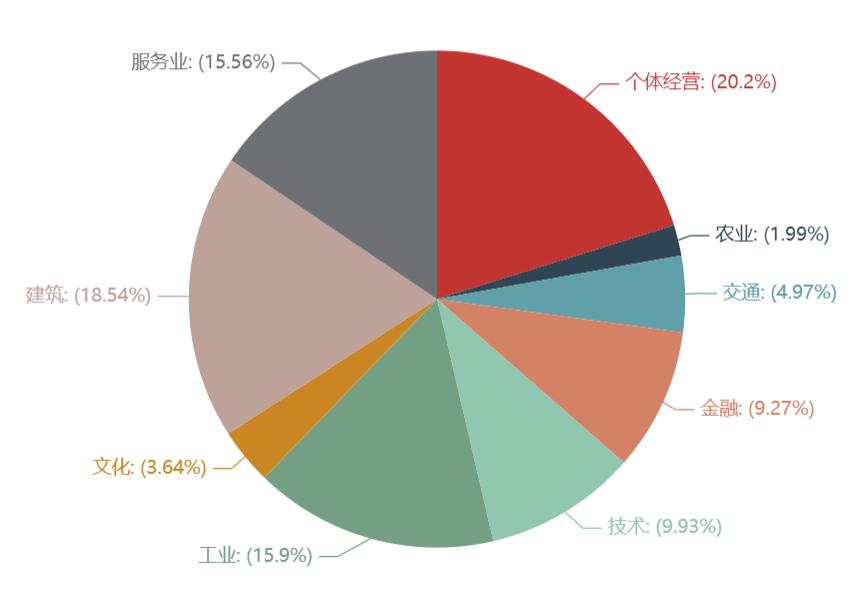


图表15新冠疫情对不同行业的影响

据此求出



同时，对附件中的所有企业进行分类，划归到不同行业：



图表 16附件二中各行业企业的占比

(3)求解

###### Step1

针对某个具体的企业，令



###### Step2

考虑突发因素对不同行业的影响，从而更新具体企业的实力值



###### Step3

将代入问题二的模型中求解即可得到企业的信贷风险数值。综合分析附件二中各企业的占比，在银行信贷对象企业中，个体经营所占的比重最大；但在新冠疫情影响下，个体经营出现衰退趋势。结合模型预测此时若对个体经营行业进行信贷业务，会有较大的违约风险，收益不如预期。

在疫情影响下，信息技术、农业、金融仍然实现了增长。在根据上述模型预测的信贷风险的基础下，对于这三个行业中信贷风险较小的企业优先贷款，并在信贷额度充足的情况下可适当增加额度。

## 六 模型的评价与推广

### 6.1模型评价

优点：

1. 模型根据给出的数据进行详细具体的分析，考虑因素多，经过逐层建模，较为严谨，考虑维度全面，符合实际。
2. 模型的求解采用了python中的科学计算库Numpy、Pandas；以及Visio、Excel、Spssau等专业工具，这些对于处理海量数据、绘制图表、拟合函数作用非常大，有助于模型求解的科学性、准确性以及高效性。
3. 实际应用需求强，可以为银行制定对小微企业的贷款策略提供参考。

缺点：

在建立数学模型时，部分条件考虑理想化，如：假设突发因素只影响企业实力，不影响企业信誉。

### 6.2模型推广

通过分析中小微企业的历史交易记录和信贷记录等数据，可以预测银行信贷风险，也可以对无信贷记录的企业做评估，预测其信誉等级，从而为银行的信贷决策提供科学的依据。

## 七 参考文献

1. 杨蓬勃,张成虎,张湘.基于Logistic回归分析的上市公司信贷违约概率预测模型研究[J].经济经纬,2009(02):144-148.
2. 贾海涛,邱长溶.宏观因素对贷款企业违约率影响的实证分析[J].现代管理科学,2009(02):67-69+72.
3. 高飞.基于Logistic回归模型的商业银行零售贷款客户违约分析——以MSLZ银行为例[J].区域治理,2019(38):204-207.
4. [孙雪梅](http://common.wanfangdata.com.cn/common/getAuthorUrl.do?&authorName=" \t "http://d.wanfangdata.com.cn/periodical/_blank).[浅析商业银行中小企业信贷风险与管理](http://d.wanfangdata.com.cn/periodical/cjj201024021" \t "http://d.wanfangdata.com.cn/periodical/_blank)[J].[财经界（学术）](http://www.wanfangdata.com.cn/perio/detail.do?perio_id=cjj" \t "http://d.wanfangdata.com.cn/periodical/_blank),2010.
5. [张莉](http://common.wanfangdata.com.cn/common/getAuthorUrl.do?&authorName=" \t "http://d.wanfangdata.com.cn/periodical/_blank).[论中小企业贷款风险管理](http://d.wanfangdata.com.cn/periodical/nfjr201404018" \t "http://d.wanfangdata.com.cn/periodical/_blank)[J].[南方金融](http://www.wanfangdata.com.cn/perio/detail.do?perio_id=nfjr" \t "http://d.wanfangdata.com.cn/periodical/_blank),2014,(4).81-82,25.
6. 姜启源，谢金星，叶俊.数学模型(第三版）[M].北京：高等教育出版社，2003.85-130.
7. 戴国强,吴许均.基于违约概率和违约损失率的贷款定价研究[J].国际金融研究,2005(10):43-48.
8. 国家统计局.季度数据.

https://data.stats.gov.cn/easyquery.htm?cn=B01

高英娟.熵权—功效系数法财务风险预警在生物制药行业的应用研究[D].云南大学,2018.

SPSSAU数据科学在线网站——AHP层次分析法<https://spssau.com/front/spssau/helps/weights/ahp.html>

## 八 附录

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T1-1 | 文件名称 | delete\_D.py |
| 说明 | 对信息进行筛选，剔除信誉D级的企业 | | |
| # 剔除信誉D级企业  import pandas as pd  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\1.xlsx', '销项发票信息', usecols=(0, 7))  D = pd.read\_excel('C:\\Users\\10049\\Desktop\\1.xlsx', '企业信息', usecols=(0, 2))  delete = []  for i in range(123):  if D.iloc[i, 1] == 'D':  delete.append(i)  D = D.drop(labels=delete, axis=0)  print(D)  D.set\_index('企业代号', inplace=True)  data\_1 = data.value\_counts()  list\_1 = []  list\_2 = []  for i in D.index.to\_list():  print(i)  list\_1.append(i)  E = data\_1.loc[i]  print(E)  if len(E) > 1:  list\_2.append(E.iloc[1] / (E.iloc[0] + E.iloc[1]))  else:  list\_2.append(0 / E.iloc[0])  S1 = pd.Series(list\_2, index=list\_1)  print(S1)  S1.to\_excel('C:\\Users\\10049\\Desktop\\2.xlsx') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T1-2 | 文件名称 | select\_1.py |
| 说明 | 计算供求稳定性，计算利润 | | |
| # 供求稳定、筛选、利润  import numpy as np  import pandas as pd  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\f2.xlsx', '进项发票信息', usecols=(0, 2, 6, 7))  sale = pd.read\_excel('C:\\Users\\10049\\Desktop\\f2.xlsx', '销项发票信息', usecols=(0, 2, 6, 7))  D = pd.read\_excel('C:\\Users\\10049\\Desktop\\1.xlsx', '企业信息', usecols=(0, 2))  # 剔除D评级企业  delete = []  for i in range(123):  if D.iloc[i, 1] == 'D':  delete.append(D.iloc[i, 0])  D.set\_index('企业代号', inplace=True)  D = D.drop(labels=delete, axis=0)  data.set\_index('发票状态', inplace=True)  data\_2 = data.drop(labels='作废发票', axis=0)  data\_2.set\_index('企业代号', inplace=True)  data\_1 = data\_2.drop(labels=delete, axis=0)  # data\_1.to\_excel('C:\\Users\\10049\\Desktop\\2.xlsx')  sale.set\_index('发票状态', inplace=True)  sale\_2 = sale.drop(labels='作废发票', axis=0)  sale\_2.set\_index('企业代号', inplace=True)  sale\_1 = sale\_2.drop(labels=delete, axis=0)  # sale\_1.to\_excel('C:\\Users\\10049\\Desktop\\3.xlsx')  l1 = []  l2 = []  l3 = []  ser = pd.Series([' '])  for i in D.index.to\_list():  a = data\_1.loc[i]  b = sale\_1.loc[i]  if type(a) == type(ser):  a = a.to\_frame()  a = pd.DataFrame(a.values.T, columns=a.index)  if type(b) == type(ser):  b = b.to\_frame()  b = pd.DataFrame(b.values.T, columns=b.index)  a\_g = a.groupby('开票日期').sum()  b\_g = b.groupby('开票日期').sum()  l\_a = a\_g.index.to\_list()  l\_b = b\_g.index.to\_list()  time\_1 = min(l\_a[0], l\_b[0])  time\_2 = max(l\_a[-1], l\_b[-1])  pda = pd.date\_range(start=time\_1, end=time\_2, freq='M')  # pd = pd.date\_range(start=time\_1, end=time\_2)  a\_t = a\_g.reindex(pda, fill\_value=0)  b\_t = b\_g.reindex(pda, fill\_value=0)  #  for j in pda:  j = str(j)[0:7]  l1.append(i)  l2.append(j)  print(i, ' ', j)  l3.append((b\_t.loc[j].sum() - a\_t.loc[j].sum()).iloc[0])  print((b\_t.loc[j].sum() - a\_t.loc[j].sum()).iloc[0])  data\_out = pd.DataFrame({  '企业代号': l1,  '开票时间（月）': l2,  '销进差': l3  })  print(data\_out)  data\_out.describe()  # data\_out.to\_excel('C:\\Users\\10049\\Desktop\\盈利趋势.xlsx')  l1 = []  l2 = []  l3 = []  data\_out.set\_index('企业代号', inplace=True)  for i in data\_out.index.unique():  a = data\_out.loc[i, '销进差']  b = a.describe()  l1.append(i)  l2.append(a.std())  for i in l2:  l3.append((i-np.min(l2))/(np.max(l2)-np.min(l2)))  data\_o = pd.DataFrame({  '企业代号': l1,  '标准差': l2,  '标志差（归一化）': l3  })  print(data\_o)  data\_o.to\_excel('C:\\Users\\10049\\Desktop\\4.xlsx', sheet\_name='Sheet2') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T1-3 | 文件名称 | norm.py |
| 说明 | 进行数据归一化 | | |
| # 归一化、最终策略  import numpy as np  import pandas as pd  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\建模文件\\risk2.xlsx', 'Sheet1')  # 数据归一化处理  min = data['credit'].min()  max = data['credit'].max()  l1 = []  for i in data['credit']:  l1.append((i - min) / (max - min))  ser = pd.Series(l1)  data.loc[:, 'credit归一化'] = ser  data.set\_index('企业代号', inplace=True)  print(data)  data.to\_excel('C:\\Users\\10049\\Desktop\\建模文件\\risk2.xlsx', 'Sheet1') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T1-4 | 文件名称 | power\_compute.py |
| 说明 | 计算企业实力相关的月均订单数，月均利润，月均利润率 | | |
| # 利润、利润率、订单数  import pandas as pd  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\1.xlsx', '进项发票信息', usecols=(0, 2, 6, 7))  sale = pd.read\_excel('C:\\Users\\10049\\Desktop\\1.xlsx', '销项发票信息', usecols=(0, 2, 6, 7))  D = pd.read\_excel('C:\\Users\\10049\\Desktop\\1.xlsx', '企业信息', usecols=(0, 2))  # 剔除D评级企业  delete = []  for i in range(123):  if D.iloc[i, 1] == 'D':  delete.append(D.iloc[i, 0])  D.set\_index('企业代号', inplace=True)  D = D.drop(labels=delete, axis=0)  data.set\_index('发票状态', inplace=True)  data\_2 = data.drop(labels='作废发票', axis=0)  data\_2.set\_index('企业代号', inplace=True)  data\_1 = data\_2.drop(labels=delete, axis=0)  sale.set\_index('发票状态', inplace=True)  sale\_2 = sale.drop(labels='作废发票', axis=0)  sale\_2.set\_index('企业代号', inplace=True)  sale\_1 = sale\_2.drop(labels=delete, axis=0)  average\_list = []  average\_sale = []  average\_saleRa = []  average\_cov = []  name = []  for i in D.index.to\_list():  a = data\_1.loc[i]  b = sale\_1.loc[i]  if type(a) == pd.Series:  a = a.to\_frame()  a = pd.DataFrame(a.values.T, columns=a.index)  if type(b) == pd.Series:  b = b.to\_frame()  b = pd.DataFrame(b.values.T, columns=b.index)  a\_g = a.groupby('开票日期').sum()  b\_g = b.groupby('开票日期').sum()  l\_a = a\_g.index.to\_list()  l\_b = b\_g.index.to\_list()  time\_1 = min(l\_a[0], l\_b[0])  time\_2 = max(l\_a[-1], l\_b[-1])  time\_3 = (time\_2 - time\_1).days / 30  name.append(i)  average\_list.append(len(b) / time\_3)  sale\_money = b\_g.sum() - a\_g.sum()  average\_sale.append(sale\_money.iloc[0])  average\_saleRa.append(sale\_money.iloc[0] / a\_g.sum().iloc[0])  data\_out = pd.DataFrame({  '企业代号': name,  '月均订单数': average\_list,  '月均利润': average\_sale,  '月均利润率': average\_saleRa,  })  # print(data\_out)  # print(data\_out['月均订单数'].corr(b['月均利润']))  data\_out.to\_excel('C:\\Users\\10049\\Desktop\\5.xlsx') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T1-5 | 文件名称 | function\_fit.py |
| 说明 | 多项式拟合 | | |
| # 多项式拟合  import matplotlib.pyplot as plt  import numpy as np  import pandas as pd  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\f3.xlsx', sheet\_name='Sheet1')  # print(data)  # print('协方差:\n', data.cov(), '\n')  # print('相关系数:\n', data.corr(), '\n')  x = data.loc[1:, '贷款年利率'].to\_list()  y = data.loc[1:, '客户流失率'].to\_list()  # y1 = data.loc[1:, 'Unnamed: 2'].to\_list()  # y2 = data.loc[1:, 'Unnamed: 3'].to\_list()  a = np.polyfit(x, y, 3)  # a1 = np.polyfit(x, y1, 3)  # a2 = np.polyfit(x, y2, 3)  b = np.poly1d(a)  # b1 = np.poly1d(a1)  # b2 = np.poly1d(a2)  c = b(x)  # c1 = b1(x)  # c2 = b2(x)  plt.scatter(x, y, marker='o', label='Credit rating A')  plt.plot(x, c, ls='--', c='blue', label='fitting with three-degree polynomial')  plt.legend()  # plt.scatter(x, y1, marker='o', label='Credit rating B')  # plt.plot(x, c1, ls='--', c='red', label='fitting with three-degree polynomial')  # plt.legend()  #  # plt.scatter(x, y, marker='o', label='Credit rating C')  # plt.plot(x, c2, ls='--', c='green', label='fitting with three-degree polynomial')  # plt.legend()  #  # plt.show()  print('red: ', b, '\n')  # print('blue: ', b1, '\n')  # print('green: ', b2, '\n') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T1-6 | 文件名称 | strategy.py |
| 说明 | 计算供求稳定性 | | |
| # 最终策略  import numpy as np  import pandas as pd  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\建模文件\\risk2.xlsx', 'Sheet1')  # 最终贷款策略计算  data.set\_index('企业代号', inplace=True)  max\_xa = 0.047  max\_xb = 0.0518  max\_xc = 0.0538  for i in data.index:  cred = data.loc[i, 'risk']  s = data['risk'].sum()  if data.loc[i, '信誉评级'] == 'A':  data.loc[i, '信赖程度'] = (1 - cred)/s  data.loc[i, '贷款利率'] = max\_xa  if data.loc[i, '信誉评级'] == 'B':  data.loc[i, '信赖程度'] = (1 - cred)/s  data.loc[i, '贷款利率'] = max\_xb  if data.loc[i, '信誉评级'] == 'C':  data.loc[i, '信赖程度'] = (1 - cred)/s  data.loc[i, '贷款利率'] = max\_xc  if data.loc[i, '信誉评级'] == 'D':  data.loc[i, '信赖程度'] = 0  print(data.loc[:, '信赖程度'])  print(data)  data.to\_excel('C:\\Users\\10049\\Desktop\\建模文件\\risk2.xlsx', 'Sheet1') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T1-7 | 文件名称 | compute\_2.py |
| 说明 | 计算作废比 | | |
| # 作废比  import numpy as np  import pandas as pd  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\1.xlsx', '进项发票信息', usecols=(0, 2, 6, 7))  sale = pd.read\_excel('C:\\Users\\10049\\Desktop\\1.xlsx', '销项发票信息', usecols=(0, 2, 6, 7))  data.set\_index('企业代号', inplace=True)  sale.set\_index('企业代号', inplace=True)  def search\_t(data\_f, l1, l2):  for i in data\_f.index.unique():  l1.append(i)  data\_1 = data\_f.loc[i]  b = 0  for j in data\_1['发票状态']:  if '作废发票' == j:  b += 1  l2.append(b / len(data\_1))  l\_1 = []  l\_2 = []  search\_t(data, l\_1, l\_2)  print(l\_1)  print(l\_2)  data\_out = pd.DataFrame({  '企业': l\_1,  '进项发票作废比': l\_2,  })  print(data\_out)  l\_3 = []  search\_t(sale, l\_1, l\_3)  ser = pd.Series(l\_3)  data\_out['销项发票作废比'] = ser  print(data\_out)  data\_out.to\_excel('C:\\Users\\10049\\Desktop\\建模文件\\f1作废比.xlsx') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T2-1 | 文件名称 | Select\_2.py |
| 说明 | 计算供求稳定性（含信誉D级企业） | | |
| # 筛选、稳定性、标准差  import numpy as np  import pandas as pd  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\f2.xlsx', '进项发票信息', usecols=(0, 2, 6, 7))  sale = pd.read\_excel('C:\\Users\\10049\\Desktop\\f2.xlsx', '销项发票信息', usecols=(0, 2, 6, 7))  data.set\_index('发票状态', inplace=True)  data\_1 = data.drop(labels='作废发票', axis=0)  data\_1.set\_index('企业代号', inplace=True)  sale.set\_index('发票状态', inplace=True)  sale\_1 = sale.drop(labels='作废发票', axis=0)  sale\_1.set\_index('企业代号', inplace=True)  l1 = []  l2 = []  l3 = []  ser = pd.Series([' '])  for i in data\_1.index.unique():  a = data\_1.loc[i]  b = sale\_1.loc[i]  if type(a) == type(ser):  a = a.to\_frame()  a = pd.DataFrame(a.values.T, columns=a.index)  if type(b) == type(ser):  b = b.to\_frame()  b = pd.DataFrame(b.values.T, columns=b.index)  a\_g = a.groupby('开票日期').sum()  b\_g = b.groupby('开票日期').sum()  l\_a = a\_g.index.to\_list()  l\_b = b\_g.index.to\_list()  time\_1 = min(l\_a[0], l\_b[0])  time\_2 = max(l\_a[-1], l\_b[-1])  pda = pd.date\_range(start=time\_1, end=time\_2, freq='M')  # pd = pd.date\_range(start=time\_1, end=time\_2)  a\_t = a\_g.reindex(pda, fill\_value=0)  b\_t = b\_g.reindex(pda, fill\_value=0)  #  for j in pda:  j = str(j)[0:7]  l1.append(i)  l2.append(j)  # print(i, ' ', j)  l3.append((b\_t.loc[j].sum() - a\_t.loc[j].sum()).iloc[0])  # print((b\_t.loc[j].sum() - a\_t.loc[j].sum()).iloc[0])  data\_out = pd.DataFrame({  '企业代号': l1,  '开票时间（月）': l2,  '销进差': l3  })  # print(data\_out)  l1 = []  l2 = []  l3 = []  data\_out.set\_index('企业代号', inplace=True)  for i in data\_out.index.unique():  a = data\_out.loc[i, '销进差']  b = a.describe()  l1.append(i)  l2.append(a.std())  for i in l2:  l3.append((i-np.min(l2))/(np.max(l2)-np.min(l2)))  data\_o = pd.DataFrame({  '企业代号': l1,  '标准差': l2,  '标志差（归一化）': l3  })  print(data\_o)  data\_o.to\_excel('C:\\Users\\10049\\Desktop\\f2供应稳定.xlsx', sheet\_name='Sheet2') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T2-2 | 文件名称 | compute\_3.py |
| 说明 | 计算负单比 | | |
| # 负单比  import numpy as np  import pandas as pd  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\1.xlsx', '进项发票信息', usecols=(0, 2, 6, 7))  sale = pd.read\_excel('C:\\Users\\10049\\Desktop\\1.xlsx', '销项发票信息', usecols=(0, 2, 6, 7))  data.set\_index('发票状态', inplace=True)  data = data.drop(labels='作废发票', axis=0)  sale.set\_index('发票状态', inplace=True)  if '作废发票' in sale.index.to\_list():  sale = sale.drop(labels='作废发票', axis=0)  data.set\_index('企业代号', inplace=True)  sale.set\_index('企业代号', inplace=True)  def search\_t(data\_f, l1, l2):  for i in data\_f.index.unique():  l1.append(i)  data\_1 = data\_f.loc[i]  b = 0  print(data\_1)  if type(data\_1) == pd.Series:  if data\_1.loc['价税合计'] < 0:  l2.append(1)  else:  l2.append(0)  else:  for j in data\_1.loc[:, '价税合计']:  if j < 0:  b += 1  l2.append(b / len(data\_1))  l\_1 = []  l\_2 = []  search\_t(data, l\_1, l\_2)  print(l\_1)  print(l\_2)  data\_out = pd.DataFrame({  '企业代号': l\_1,  '进项负单比': l\_2,  })  print(data\_out)  l\_3 = []  search\_t(sale, l\_1, l\_3)  ser = pd.Series(l\_3)  data\_out['销项负单比'] = ser  print(data\_out)  data\_out.to\_excel('C:\\Users\\10049\\Desktop\\建模文件\\f1负单比.xlsx') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T2-3 | 文件名称 | Predict\_credict.py |
| 说明 | 违规预测与信誉预测 | | |
| # 违规预测  import pandas as pd  import math  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\建模文件\\预测违约率.xlsx')  input\_1 = -6.920  output\_1 = -2.405  average\_ra = -0.001  average\_mon = -0.000  average\_number = -0.125  input\_2 = 1.769  output\_2 = 5.587  data.loc[:, '违约预测'] = input\_1 \* data['进项负单比'] + output\_1 \* data['销项负单比'] + average\_ra \* data['月均利润率'] + \  average\_number \* data['月均订单数'] + input\_2 \* data['进项发票作废比'] + output\_2 \* data['销项发票作废比'] - 0.511  # average\_num = -0.127  # out\_put = 5.902  #  # data.loc[:, '违约预测'] = average\_num \* data['月均订单数'] + out\_put \* data['销项发票作废比'] - 0.808  l1 = []  l2 = []  for i in data['违约预测']:  p = math.exp(i) / (1 + math.exp(i))  l1.append(p)  if p > 1 / 17:  l2.append("D")  elif p > 1 / 38:  l2.append('C')  elif p > 1 / 10000:  l2.append('B')  else:  l2.append('A')  data.loc[:, '违规概率'] = pd.Series(l1)  data.loc[:, '预测信誉评级'] = pd.Series(l2)  print(data)  data.to\_excel('C:\\Users\\10049\\Desktop\\建模文件\\f1预测违约.xlsx') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T3-1 | 文件名称 | power3\_compute.py |
| 说明 | 换算利润 | | |
| # 换算利润  import pandas as pd  import numpy as np  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\建模文件\\power\_f2.xlsx')  data\_1 = pd.read\_excel('C:\\Users\\10049\\Desktop\\建模文件\\新冠对行业的影响.xlsx')  data.set\_index('企业代号', inplace=True)  data\_1.set\_index('行业类别')  def fun (b):  data.loc[i, '月均订单数'] = data.loc[i, '月均订单数'] \* b  data.loc[i, '月均利润'] = data.loc[i, '月均利润'] \* b  for i in data.index:  a = data.loc[i, '行业类别']  if a in data\_1.index:  fun(data\_1.loc[a, '最终系数'])  data.to\_excel('C:\\Users\\10049\\Desktop\\建模文件\\power\_f3.xlsx') | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 程序编号 | T3-2 | 文件名称 | check.py |
| 说明 | 检查贷款策略分配的额度 | | |
| # 贷款额度检查  import pandas as pd  data = pd.read\_excel('C:\\Users\\10049\\Desktop\\建模文件\\第二问最终结果.xlsx')  data['贷款额度(万元)'] = data['信赖程度'] \* 10000  if data['贷款额度(万元)'].max() > 100:  print('存在贷款额度超额情况')  if data['贷款额度(万元)'].min() < 10:  print('存在贷款额度不足')  data.to\_excel('C:\\Users\\10049\\Desktop\\建模文件\\第二问最终结果.xlsx') | | | |