

基于链路预测对老龄化的分析

摘要

中小

关键字： 信贷定价 随机森林 朴素贝叶斯

目录

一、问题重述	4
二、模型假设	4
三、符号说明	7
四、问题一——LPR 定价机制下中小商业银行信贷定价模型	8
4.1 问题一的分析	8
4.2 数据预处理	8
4.3 银行贷款年利率与客户流失率	8
4.4 基于随机森林的信贷风险量化模型	9
4.5 信贷策略的确定	9
4.5.1 贷款利率的确定	10
4.5.2 贷款额度的确定	12
五、问题二——基于随机森林的信誉等级评定	12
5.1 问题二的分析	12
5.2 随机森林信誉等级评定	13
5.2.1 模型训练	13
5.2.2 信誉评级判定	15
六、问题三——基于朴素贝叶斯的文本分类	15
6.1 问题三的分析	15
6.2 自然语言处理	15
6.3 朴素贝叶斯文本分类	16
6.3.1 贝叶斯公式	16
6.3.2 词袋法	16
6.3.3 TF 计算方法	16
6.3.4 拉普拉斯平滑	17
6.4 模型求解	17
七、模型的评价与推广	19
7.1 模型评价	19

7.1.1 模型优点	19
7.1.2 模型缺点	19
7.2 模型推广	19
参考文献	20
附录 A 数据预处理以及指标生成-MATLAB 源代码	22
附录 B 随机森林进行信誉等级评定-Python 源代码	27
附录 C 朴素贝叶斯进行文本分类-Python 源代码	28
附录 D 问题二中 302 家企业信誉评级结果	30

一、问题重述

随着贷

$$m_2 \circledcirc m_1$$

二、模型假设

Algorithm 1: 分支限界算法实现旅行商问题

Input: 图的邻接矩阵: G

Output: 最优解和最优值

初始化最小堆;

while 遍历每一层 $E.s < (n-1)$ **do**

if 当前扩展结点是叶结点的父结点 (到了 $n-2$ 层) **then**

if 形成回路且当前值优于最优值或还未产生最优值 **then**

 更新最优解;

 更新最优值;

else

 舍弃需要进一步搜索的节点;

else

for 遍历这个节点的所有扩展节点 **do**

if 这个点可以进入下一个未访问的点 **then**

 更新当前费用;

 更新最小出边费用和;

 更新下界 (限界函数);

if 子树可能含最优解 **then**

 节点插入最小堆;

 完成当前节点的扩展, 清除该节点;

if 最小堆已经空 **then**

 结束循环

 从最小堆中弹出队首节点 E ;

- 所有附件所给的数据真实且合理;
- 不考虑其他主观因素 (例如企业之间的商业竞争、银行之间的利益冲突) 的影响;
- 数据源中其他未列出的评价对企业的信誉评级没有影响;
- 假设同一信用等级企业能够申请的最大信贷额度是一定的;

Algorithm 2: 回溯算法实现 n 皇后问题

Input: 棋盘数组: $Chess[N][N]$, 递归到第 n 层: n

Output: 所有放置皇后的方案和方案总数

if 到了第 $(n + 1)$ 层 (已经成功放置最后一个皇后) **then**

 格式化输出当前解决方案;

 返回上一层;

else

for 遍历这一层的每一个点 **do**

if 这个点满足放置皇后的条件 **then**

 放置皇后;

 进入下一层递归;

 移走皇后;

Algorithm 3: 回溯算法实现旅行售货员问题

Input: 递归到第 n 层: n

Output: 最优解和最优值

if 到了第 n 层 (已经成功访问了所有节点) **then**

if 形成回路 (当前点和上一个点有连线, 且和第 1 个点有连线) **then**

if 当前值优于最优值或还未产生最优值 **then**

 更新最优解;

 更新最优值;

 返回上一层;

else

for 遍历这一层的每一个点 **do**

if 这个点可以进入下一个未访问的点 **then**

 将这个点通过交换的方式存入解数组;

 累加最优值;

 进入下一层递归;

 将这个点通过交换的方式移出解数组;

 还原最优值;

-
- 不考虑企业与银行的历史交易关系, 例如企业在银行的往期存款、企业在银行的入股情况、与银行的合作时长以及在银行抵押优质资产占客户在金融机构抵押的优质资产比率等;
 - 假定贷款年限为一年, 贷款额度为 10-100 万元, 年利率为 4%-15%。

Algorithm 4: 贪心算法实现二元归并树问题

Input: 需要归并的文件长度向量 arr .

Output: 最小带权外部路径长度

对文件长度数据进行从小到大的排序;

while 当前 $vector$ 中还有未进行归并的节点 **do**

 查找 $vector$ 中当前最小的两个节点;

 将这两个节点进行归并, 并打上标记 (赋值为正无穷);

 将归并结果作为新的节点插入 $vector$;

 记录这两个归并节点和新节点的父节点和子节点信息;

 累加移动总量;

Algorithm 5: 动态规划实现矩阵连乘最优化问题

Input: 连乘矩阵行列值数组 p ; 连乘矩阵的个数 n ; 储存分步最优值矩阵的内存 m ; 储存分步最优解矩阵的内存 s .

Output: 最优解

考虑连乘长度为 1 的情况, 将对应矩阵运算次数的最优值都设置为 0;

for 连乘长度, 子问题规模 (从 2 两个开始, 到全部矩阵连乘) **do**

for 连乘开始的位置, 子问题定位 **do**

for 分成两部分的分割位置 **do**

if 这是子问题的最优值 **then**

 在矩阵 m 中进行记录最优值;

 在矩阵 s 中进行记录分割位置;

返回最优值;

Algorithm 6: 分治策略实现棋盘覆盖问题

Input: 棋盘的左上角坐标 (tr, tc) ; 特殊方块的位置 (dr, dc) ; 棋盘大小 $size$.

Output: 使用 L 型骨牌覆盖好的棋盘

骨牌编号 t 随递归次数增加;

棋盘大小 $size$ 随递归次数每次除以 2 增加;

(a. 覆盖左上角子棋盘)

if 特殊方格在左上角子棋盘中 **then**
└ 递归覆盖非特殊方格的其余方格;

else
└ 用 t 号 L 型骨牌覆盖右下角;
└ 递归覆盖其余方格;

(b. 覆盖右上角子棋盘)

if 特殊方格在右上角子棋盘中 **then**
└ 递归覆盖非特殊方格的其余方格;

else
└ 用 t 号 L 型骨牌覆盖左下角;
└ 递归覆盖其余方格;

(c. 覆盖左下角子棋盘)

if 特殊方格在左下角子棋盘中 **then**
└ 递归覆盖非特殊方格的其余方格;

else
└ 用 t 号 L 型骨牌覆盖右上角;
└ 递归覆盖其余方格;

(d. 覆盖右下角子棋盘)

if 特殊方格在右下角子棋盘中 **then**
└ 递归覆盖非特殊方格的其余方格;

else
└ 用 t 号 L 型骨牌覆盖左上角;
└ 递归覆盖其余方格;

三、符号说明

符号	意义	备注
W	银行的放贷盈利	
Q_i	i 类企业的贷款额度	10-100 万元
n_i	i 类企业的个数	
r_0	央行出台的贷款基准利率	
r_i	银行针对 i 类企业的贷款年利率	4%-15%

四、问题一——LPR 定价机制下中小商业银行信贷定价模型

4.1 问题一的分析

对附件 1 中 123 家企业的信贷风险进行量化分析，计算出“进项发票作废比”、“销项发票作废比”、“2016 至 2020 年年支出”、“2016 至 2020 年年收入”、“2016 至 2020 年年利润”、“进项负数发票数量”、“销项负数发票数量”等初级指标，对所给数据先进行升维处理。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。

问题一要求对已给的 123 企业的信贷风险进行量化分析，并在附件一中给出了这 123 家企业的企业类别、信誉评级、是否违约和进出项发票的信息。

针对信贷风险的量化问题：首先，我们需要对数据进行处理，根据经济学的知识和经验计算出衡量一个企业财务和信用状况的几个指标；接着在第二问中我们利用随机森林方法对各个指标和信用水平的关系式进行确定。

针对银行的信贷策略问题 [1]：策略由是否放贷、放贷额度、利率和期限组成。其中，放贷期限为 1 年；根据信用水平的量化表达式，我们通过分析企业信用水平对银行信贷利率的影响计算出不同信用等级企业的放贷利率，最后在“银行净利润最大”的优化目标下求解不同等级企业的放贷额度。

4.2 数据预处理

题目为我们提供了三个附件，附件中的数据给出了各企业的相关信息：附件 1 是 123 家有信贷记录企业的进销项发票信息；附件 2 是 302 家无信贷记录企业的进销项发票信息；附件 3 是银行贷款年利率与不同信誉评级的客户流失率的关系。对于众多的发票信息，我们需要对数据进行清洗整理，使用 EXCEL 和 MATLAB 软件对数据做了以下预处理：

1. 统计出每个公司进项、销项发票总数、发票作废数，进而计算出进销项发票作废比。
2. 根据开票日期和进、销项发票的税价合计，统计出 2016-2020 年每个公司的收入、支出，进而计算出当年的利润。
3. 我们根据《国民经济行业分类》[2] 对附件 1 中的 123 家企业进行了人工分类。共分为 12 类：餐饮业、城市建设、电器、服务业、工业、建筑与装饰、交通运输、科技密集型产业、零售业、农业、医疗、个体企业。
4. 对数据进行归一转换，并把字符串类型特征“是”、“否”分别用“1”、“0”表示。

4.3 银行贷款年利率与客户流失率

附件 3 中给出了银行贷款年利率与客户流失率的数值关系。其中客户流失率 (Customer Churn Rate, CCR) [3, 4] 不仅与银行设定的贷款年利率有关，也与客户企业的信誉评级有



图 1 初级指标

关。对于信誉评级高的企业，其可以供选取的信贷公司较多，因此会在信贷公司之间给出的年利率进行横向比较。然而，对于信誉评级较低的企业，其可供选择的信贷公司较少。

图 (2) 中展示的是针对 A、B、C 三级企业，银行贷款的不同年利率对客户流失率的关系。在下方的三个子图中，我们对关系曲线进行了多项式拟合 (5 阶)。得到不同信誉等级的企业流失率与利率的关系为：

$$\begin{aligned}
 l_1 &= 141970x^5 - 72481x^4 + 14857x^3 - 1588.2x^2 + 96.825x - 2.1004 \\
 l_2 &= 117499x^5 - 61734x^4 + 1298x^3 - 1414.9x^2 + 87.735x - 1.9255 \\
 l_3 &= 114626x^5 - 56609x^4 + 11257x^3 - 1183.1x^2 + 74.171x - 1.6557
 \end{aligned} \tag{1}$$

其拟合优度 R^2 均大于 0.998，拟合效果良好 [5]。

4.4 基于随机森林的信贷风险量化模型

4.5 信贷策略的确定

信贷策略由是否放贷、贷款额度、贷款利率和期限组成。题中所给的贷款期限为 1 年。根据上文得出的对某一特定企业信贷风险的量化模型，我们认为信用等级为 D 级的企业应不予授予放贷资格。

银行放贷的目的是为了在帮扶中小型企业的同时、实现贷款盈利最大化，由此我们可以得出数学模型：

$$\begin{aligned}
 \max W &= \sum_{i=1}^3 Q_i n_i r_i (1 - l_i(r_i)) \\
 s.t \quad &10 \leq Q_i \leq 100 \quad i = 1, 2, 3 \\
 &Q_1 \leq Q_2 \leq Q_3 = \frac{S - Q_1 n_1 - Q_2 n_2}{n_3}
 \end{aligned} \tag{2}$$

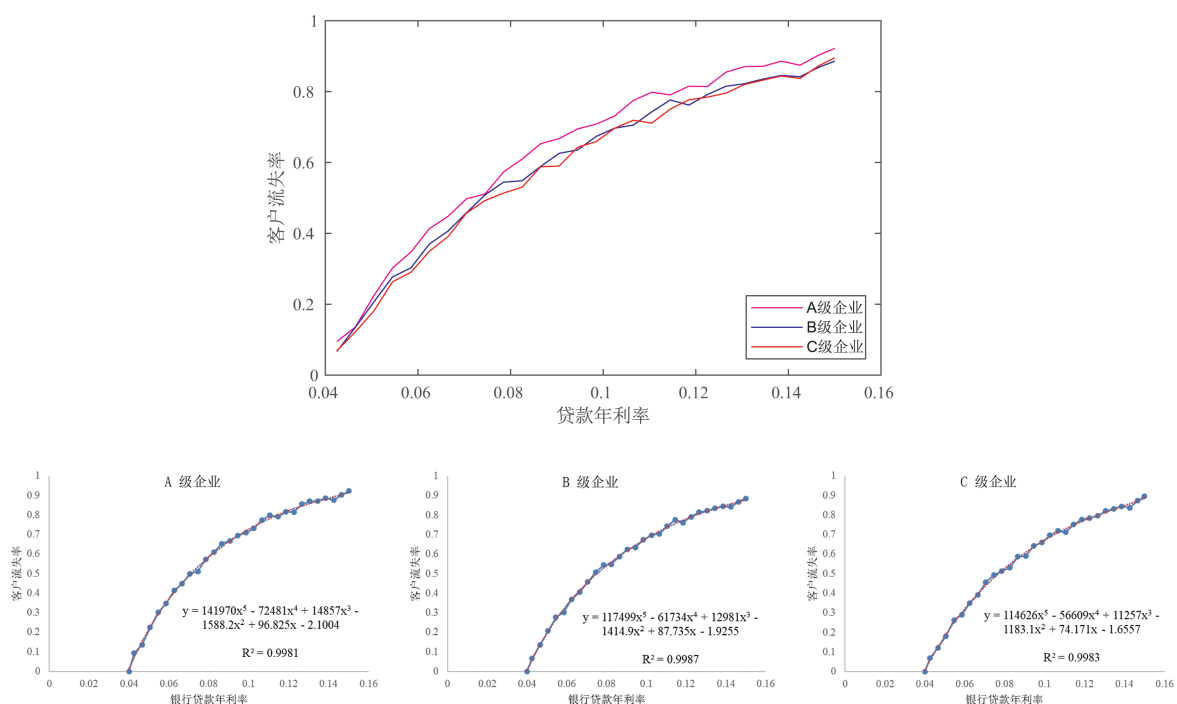


图 2 银行贷款年利率与客户流失率关系

其中, W 为银行盈利; Q_i 为信用等级 i 的企业所得贷款额度; r_i 为 i 类企业的贷款利率; n_i 为 i 类企业的数量; l_i 为 i 类企业在贷款利率为 r_i 情况下的流失率, 为 4.3 中拟合出的关系式。

接下来, 我们通过建立模型对不同信用等级企业的贷款利率 r_i 、贷款额度 Q_i 进行确定。

4.5.1 贷款利率的确定

银行信贷业务的定价主要基于风险量化和会计核算两个角度, 目前业界主要采用的定价模式有风险加点模式、成本加成模式和客户关系模式。[6]

其中风险加点模式是从风险溢价的角度进行定价, 核心思想是: 利率由基准利率和信用风险溢价组成。其代表是 LPR (Loan Prime Rate) [7] 定价机制。

成本加成模式是从会计核算的角度对贷款进行定价, 主要代表是以 RAROC (Risk Adjusted Return on Capital) 模型 [8] 为核心的定价机制, 其核心思想是: 商业银行为了创造利润, 其业务成本 (包括资金成本、运营成本、资本成本等) 与企业违约产生的风险成本应不超过信贷产品价格。该种利润针对的主要对象是资金流较大导致银行业务成本远大于风险成本的的大型企业, 不适用于本题中所给的中小型企业。

客户关系模式即为银行在计算客户信贷额度时, 要将该客户有关的所有收益与费用全部纳入考虑范围, 包括客户在行存款、存款资金投资收益等, 是基于客户与银行的历史交易合作信息的考量, 故不适用于本题。

综上，我们采用风险加点模式中的 LPR 信贷利率计算机制对贷款利率进行计算。LPR 信贷利率计算模型如下：

$$r_i = r_0 + bp_i \quad (3)$$

其中， r_i 为信用等级为 i 类企业的信贷利率； r_0 为基准利率，由央行在当年颁布，经过查询，我们得到 2020 年的 r_0 值为 3.85%； b 风险最高客户的贷款利率与 r_0 之差； p_i 为 i 类企业的风险溢价率。

接下来我们对风险溢价率进行建模计算。

我们认为企业的风险溢价率可以由其失信率近似。企业失信率越高，则其风险溢价率越高，银行提供给企业的贷款利率就越高。因此，我们利用手上的企业票据信息和上下游企业信息对企业的失信率进行度量。

现实中不同银行的风险偏好不同。对于风险厌恶型的银行会要求更高的风险溢价，风险喜好型则相对较低。由于该模型的对象是中小微企业，其受到国家政策的扶持，且融资周期相对大型企业较短，故在模型中我们假定银行为风险中性型。根据冯·诺依曼-摩根斯坦效用函数，风险喜好型银行的效用函数为凸函数。故我们定义 i 类企业的失信率 p_i ：

$$p_i = \frac{1}{1 + e^{-\vec{x}\vec{\beta}}} \quad (4)$$

其中 $\vec{x} = (x_1, x_2, x_3)$ ，为衡量失信率的三个指标构成的三维向量， $\vec{\beta} = (\beta_1, \beta_2, \beta_3)$ 为三个指标的权重向量。

下面我们定义 \vec{x} 。

由于样本数 n_i $i = 1, 2, 3$ 过少，不满足弱大数率的条件，故无法直接用样本失信率代替某信誉等级企业的总体失信率。因此对信用评级为 i 的企业，我们提出三个指标来衡量其失信率以增强其失信率的准确程度，分别为样本失信率 x_{i1} 、进项作废比 x_{i2} 、销项作废比 x_{i3} 。其计算公式如下：

$$\begin{aligned} x_{i1} &= \frac{n_{i1}}{n_1} \\ x_{i2} &= \frac{n_{i2}^*}{n_{i2}} \\ x_{i3} &= \frac{n_{i3}^*}{n_{i3}} \end{aligned} \quad (5)$$

其中 n_{i1} 表示失信 i 类企业数， n_{i2}^* 表示 i 类企业的作废进项发票数， n_{i2} 表示 i 类企业的总进项发票数， n_{i3}^* 表示 i 类企业的作废销项发票数， n_{i3} 表示 i 类企业的总销项发票数。

在得到衡量失信率的三个指标之后，我们利用熵权法确定三个指标的权重。

由于三个指标的实际意义都为某种比例，且都是介于 $[0, 1]$ 之间的数，具有相同的量纲；且样本失信率和两类发票的作废率和企业失信率正相关，即都为同质正项指标，故我们不对指标进行归一化处理。

计算第 j 项指标下第 i 个样本值占该指标的比重 q_{ij} 为：

$$q_{ij} = \frac{x_{ij}}{\sum_{i=1}^3 x_{ij}} \quad i = 1, 2, 3 \quad j = 1, 2, 3 \quad (6)$$

计算第 j 项指标的信息熵冗余度 d_j ：

$$d_j = 1 + \frac{1}{3} \sum_{i=1}^3 q_{ij} \ln q_{ij} \quad j = 1, 2, 3 \quad (7)$$

得出各项指标权重 β_j ：

$$\beta_j = \frac{d_j}{\sum_{j=1}^3 d_j} \quad j = 1, 2, 3 \quad (8)$$

经过计算，得到三个指标的权重如下表所示：

因此最终得到的风险溢价率模型为：

$$p_i = \frac{1}{1 + e^{-(\sum_{j=1}^3 \beta_j \frac{n_{ij}^*}{n_{ij}})}} \quad i = 1, 2, 3 \quad (9)$$

综上，对于信用等级为 i 类的企业，银行提出的贷款利率 r_i 为：

$$r_i = r_0 + \frac{b}{1 + e^{-(\sum_{j=1}^3 \beta_j \frac{n_{ij}^*}{n_{ij}})}} \quad i = 1, 2, 3 \quad (10)$$

其中 $i = 1, 2, 3$ 分别代表 A, B, C 三类企业, r_0 、 b 由当年央行及商业银行颁布决定， β_i 为上述表格中的值。

4.5.2 贷款额度的确定

通过遍历空间中的点求取银行利润 W 的最大值，我们可以得到不同信用等级企业的贷款额度的最优解，如表 (1) 所示。

⑤

五、问题二——基于随机森林的信誉等级评定

5.1 问题二的分析

在问题一的模型基础上，对待不同信誉评级的企业，我们已经具有适合的信贷方案，在问题二中我们需要应用问题一的分级信贷策略。因此，如何利用附件 2 中的数据对

302 家企业进行信誉评级成为了问题二待解决的重点问题，其本质为一个多指标分类问题。

对于附件 2 中数据的预处理与附件 1 类似，通过各个企业的进项发票和销项发票，分析得出该企业的“进项作废比”、“销项作废比”、“进项负数比”、“销项负数比”、“年均利润”等初级指标。使用附件 1 中相同的指标通过随机森林的方法进行模型训练，使用超参数调优的方法选择出最佳的模型参数，并将其应用于附件 2 中数值的处理，得出附件 2 中各企业的信誉评级，并根据信誉评级结合问题一中的分级信贷策略进行信贷分配。

5.2 随机森林信誉等级评定

在机器学习中，随机森林 (Random Forest, RF) 是一个包含多个决策树的分类器，并且其输出的类别是由个别树输出的类别的众数确定 [9]。它利用相同的训练数搭建多个独立的分类模型，然后通过投票的方式，以少数服从多数的原则作出最终的分类决策，因此大大提高了模型的预测精度 [10]。一个标准的决策树会根据每维特征对预测结果的影响程度进行排序，进而决定不同的特征自上而下构建分裂节点的顺序，如此以来，所有在随机森林中的决策树都会受这一策略影响而构建的完全一致，从而丧失多样性。所以在随机森林分类器的构建过程中，每一棵决策树都会放弃这一固定的排序算法，转而随机选取特征。

5.2.1 模型训练

在随机森林的模型训练阶段，使用 Bootstrap 采样方法 [11, 12]，从附件 1 训练数据集中采集多个不同的子训练数据集来依次训练多个不同决策树。这里我们使用附件 1 中随机选取 75% 的数据作为训练集，25% 的数据作为测试集。随机森林的训练流程图展示在图 (3) 中。

随机森林的模型是将多个二叉决策树组合而成的，训练一个随机森林便是训练多个二叉决策树。在训练二叉决策树模型的时候需要考虑怎样选择分变量 (特征)、切分点以及怎样衡量一个切分变量、切分点的好坏。针对于切分变量和切分点的选择，采用穷举法，即遍历每个特征和每个特征的所有取值，最后从中找出最好的切分变量和切分点；针对于切分变量和切分点的好坏，一般以切分后节点的不纯度来衡量，即各个子节点不纯度的加权和 $G(x_i, v_{ij})$ ，其计算公式如下

$$G(x_i, v_{ij}) = \frac{n_{left}}{N_s} H(\alpha_{left}) + \frac{n_{right}}{N_s} H(\alpha_{right}) \quad (11)$$

其中， x_i 为某一个切分变量， v_{ij} 为切分变量的一个切分值， n_{left} 、 n_{right} 、 N_s 分别为切分后左子节点的训练样本个数、右子节点的训练样本个数以及当前节点所有训练样

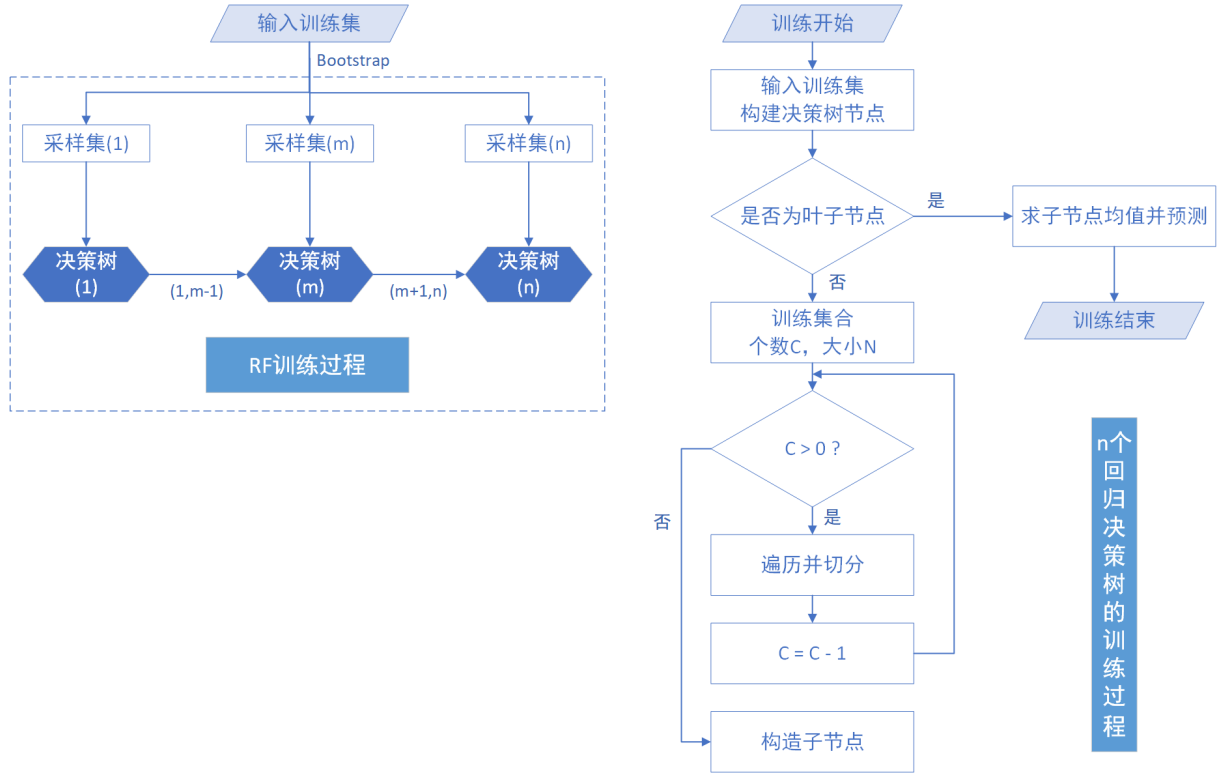


图3 随机森林训练流程图

本个数, α_{left} 、 α_{right} 分为左右子节点的训练样本集合, $H(X)$ 为衡量节点不纯度的函数。本模型中选用的是基尼不纯度 [13], 其定义如下:

$$H(\alpha_m) = \sum_k p_{mk} (1 - p_{mk}) \quad (12)$$

其中, X_m 为当前节点的训练样本集合, p_{mk} 为当前节点训练样本中目标变量取值 $p_{mk} = \frac{N_{mk}}{N_m}$ 出现的概率。

决策树中某一节点的训练过程在数学上等价为即寻找 G 最小的切分变量和切分点优化问题:

$$(x^*, v^*) = \operatorname{argmin}_{x,v} G(x_i, v_{ij}) \quad (13)$$

计算特征重要性 (Feature Importance), 特征的重要性表示特征对预测结果影响程度, 某一特征重要性越大, 表明该特征对预测结果的影响越大, 重要性越小, 表明该特征对预测结果越小。某一节点 k 的重要性为:

$$n_k = w_k * G_k - w_{left} * G_{left} - w_{right} * G_{right} \quad (14)$$

其中, w_k, w_{left}, w_{right} 分别为节点 k 以及其左右子节点中训练样本个数与总训练样本数目的比例, G_k, G_{left}, G_{right} 分为为节点 k 以及其左右子节点的不纯度。知道每一个节点的重要性之后, 即可通过公式 (15) 得出某一特征的重要性。

$$f_i = \frac{\sum_{j \in \text{nodes split on feature } i} n_j}{\sum_{k \in \text{all nodes}} n_k} \quad (15)$$

为了使所有特征的重要性加起来等于 1，需要每一特征的重要性进行归一化，公式如下：

$$f_{ni} = \frac{f_i}{\sum_{j \in \text{all features}} f_j} \quad (16)$$

5.2.2 信誉评级判定

随机森林的预测判定结果是由内部所有二叉决策树的预测结果取平均值得到的。二叉决策树的预测判定过程主要分为以下步骤：

1. 针对某一输入样本，从二叉决策树的根节点起，判断当前节点是否为叶子节点，如果是则返回叶子节点的预测值 (即当前叶子中样本目标变量的平均值)，如果不是则进入下一步；
2. 根据当前节点的切分值进行移动；
3. 循环步骤 2，直到访问到叶子节点，并返回叶子节点的预测值。

将预处理后的附件 2 中的企业指标带入训练后的模型进行信誉评级判定，得出了附件 2 中 302 所企业的信誉评级，其结果展示在附录 D。

六、问题三——基于朴素贝叶斯的文本分类

6.1 问题三的分析

在企业的经营中，一些突发因素是不可避免的。例如 2020 年席卷全球的新冠疫情，让餐饮业、零售业、旅游业以及各种类型的个体企业受到了重大打击，然而对于一些医疗方面的企业反而可能会带来销售额上的大幅上涨。因此，企业的类别对企业应对不同类型的突发因素的表现不同，受其影响也不同。为了可以更加准确的估计企业的盈利额，进而评估企业的信贷风险，我们十分有必要将企业类型考虑到模型中去。附件 1 和附件 2 中给出了企业的名称，但是数据量庞大，在短时间内使用人工分类的可能性很低，且不准确。因此我们使用自然语言处理 (Natural Language Processing, NLP) 的方法 [14–16]，并结合朴素贝叶斯算法 (Naive Bayes, NB) [17]，对所给的公司名称进行分类 [18]，进而在问题一、二的模型基础上进行改进。

6.2 自然语言处理

将基于 Python 文本分析的生态化识别方法引入企业分类。首先进行数据清洗，使用 Python 完成对评论文本的 NLP 程序编写，可以有效分词去停、过滤附件中企业名称中的“*”号。分词主要运用 Pynlpir 模块中的 Segment 方法，此方法当前广泛应用于中文文本分析相关研究领域，具有较高的分词精准性 [19, 20]。具体的自然语言处理步骤展示在 4。



图 4 自然语言处理步骤

6.3 朴素贝叶斯文本分类

以贝叶斯原理为基础的贝叶斯方法，运用了概率统计的知识对样本数据集进行分类。由于其具有坚实的数学理论基础，贝叶斯分类算法的误判率是比较低的。朴素贝叶斯是一个非常简单，但是实用性很强的分类模型。朴素贝叶斯分类器的构造基础就是贝叶斯理论。

6.3.1 贝叶斯公式

贝叶斯公式的表示如下：

$$P(c_i | \beta) = \frac{P(\beta | C_i)P(c_i)}{P(\beta)} \quad (17)$$

其中 c_i 为类别， β 为特征向量。

贝叶斯公式最常用于文本分类，公式 (17) 的左边可以理解为给定一个文本词向量 β ，那么它属于类别 c_i 的概率是多少。公式 (17) 的右边分三部分， $P(\beta | c_i)$ 表示了在给类别的情况下，该文档的词向量的概率，其可以通过条件概率中的重要特性来求解。

6.3.2 词袋法

使用词袋法，且以《国民经济行业分类词库》[2] 训练集中的文本为词汇表，即将训练集中的文本中出现的单词 (不重复) 都统计出来作为词典，那么记单词的数目为 n ，这代表了文本的 n 个维度。

6.3.3 TF 计算方法

使用词频 TF (Term Frequency) 的计算方法，为类别 y_k 每个词汇出现的次数 N_i ，除以公司类别 y_k 中词汇单词出现次数的总数 N ：

$$P_i = \frac{N_i}{N} \quad (18)$$

6.3.4 拉普拉斯平滑

为了避免训练集样本对一些特征的缺失，即某一些特征出现的次数为 0，在计算 $P(X_1, X_2, X_3, \dots, X_n | Y_i)$ 的时候，各个概率相乘最终结果为零，这样就会影响结果。我们对这个概率计算公式做了一个平滑处理：

$$P_i = \frac{N_i + \gamma}{N + \gamma * m} \quad (19)$$

其中 m 为特征词向量的个数, γ 为平滑系数, 当 $\gamma = 1$ 时称为拉普拉斯平滑。

6.4 模型求解

使用《国民经济行业分类词库》[2] 进行训练分类后生成问题一中划分的 12 种企业类型，如图 (5) 所示：餐饮业、城市建设、电器、服务业、工业、建筑与装饰、交通运输、科技密集型产业、零售业、农业、医疗、个体企业。

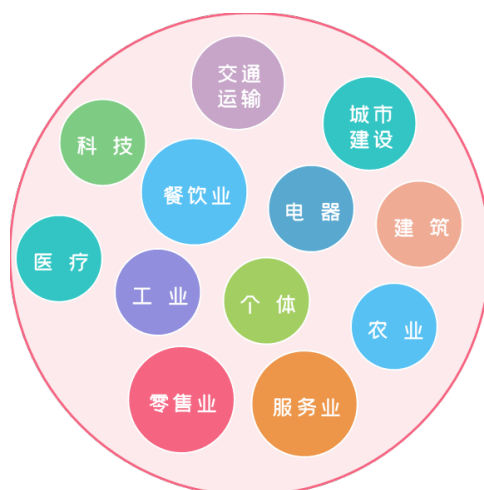


图 5 12 种企业类型

将训练好的模型对附件 1 和附件 2 中的企业进行分类，得到的结果展示在图 (6) 中。由于在问题一的数据预处理中，我们已经人工对 123 家公司进行了分类，将人工分类结果与模型分类结果相比较，符合率达到了 93.47%，说明分类效果很好，可信度较高。

同一种突发因素对不同类型企业的影响程度不同，影响好坏也有区别。例如一些医疗器械类型的企业，在疫情期间首批复工，发展迅速，Choice 数据显示，截至 8 月 19 日，医药生物行业有 157 家上市公司发布 2020 年中报业绩预告，其中，超过 72 家企业业绩同向上升，但疫情期间的零售业、个体和交通运输业等等线下行业的利润严重下降，为了响应国家的号召，顾客疫情期间自我封闭管理，不出门不串门，对零售业和个体来说，这大大降低了其人流量，进而导致货品积压，资金链断裂；对交通运输业来说，客运单位均面临大量的线路取消、变更及旅客票务退改，对票务平台和客服部门带来极大的压力。地震也是不可忽略的突发情况，地震后救治伤员所用的药品，尤其是疫苗和大输液，

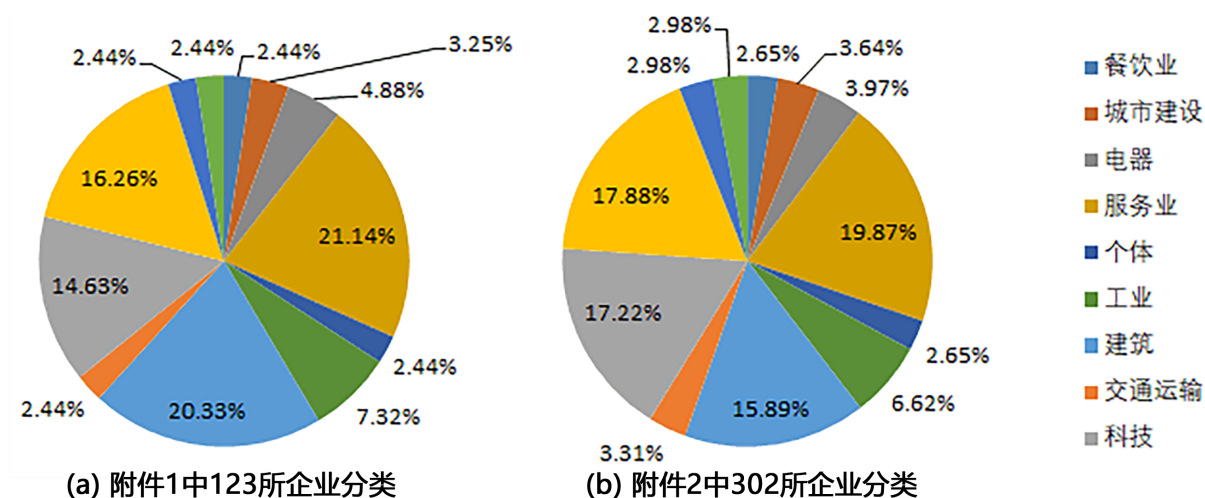


图 6 企业分类结果

使医药行业大大受益，对提供衣食住行的农业与食品饮料行业也有良好的影响，但对于灾区的交通运输、供水供电、石油石化等公用事业公司，则产生偏负面的影响，进而影响到地震所在地的旅游景区。

图 (7) 为 2019—2020 年第一季度共五个季度的农业和零售业总产值柱形图，从图中我们可以看出，受疫情的影响，2020 年第一季度农业、零售业总产值都大幅度下降，打破了持续增加的趋势。

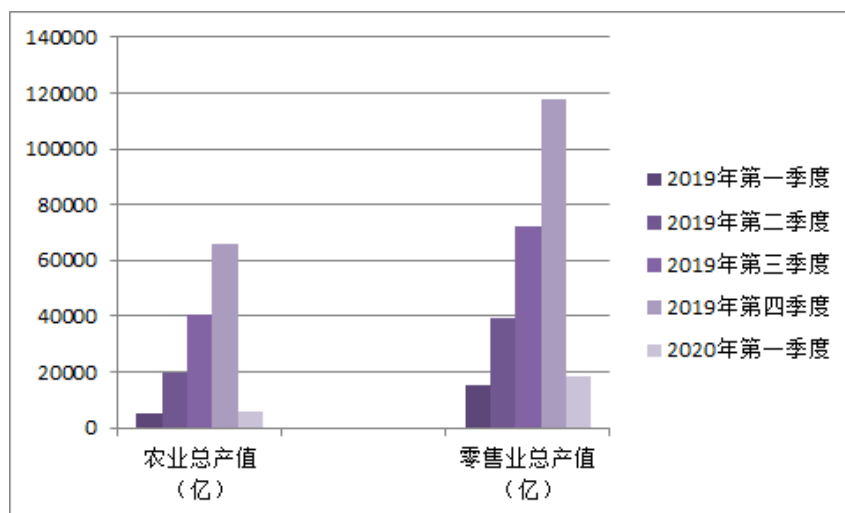


图 7 2019—2020 年第一季度共五个季度的农业和零售业总产值柱形图

七、模型的评价与推广

7.1 模型评价

7.1.1 模型优点

- 对附件中的数据进行了有效的数据清洗和数据预处理，使结果更加真实可靠，提高模型求解速度；
- 本文在正确、清晰地分析了题意的基础上，建立了合理、科学的企业信誉评级模型，为 302 家无信贷记录企业的分析提供了条件；
- 在假设的基础上建立了银行利润模型，巧妙地解决了不同信誉等级企业的不确定性；
- 建立的规划模型能与实际紧密联系，结合实际情况对问题进行求解，使得模型具有很好的通用性和推广性；
- 随机森林的算法能够有效地运行在大数据集上，且能够处理具有高维特征的输入样本，不需要降维。
- 计算采用专业的数学软件 (MATLAB)，可信度较高。

7.1.2 模型缺点

- 本文将每个中小型企业都看作孤立个体，未考虑企业之间的相互影响，会与真实的结果产生偏差，也不能体现不同规模企业在授信额度上的差异；
- 本题是在理想的状态下求解的，现实的很多情况并没有考虑进去，中小型企业信贷的概率还可能受到国家经济形势与政策等多方面的影响；
- 由于只有企业的发票数据，缺失其他股份情况、负债情况、固定资产情况和与银行关系的信息，故只能根据企业的现金流数据对其违约风险进行分析量化；
- 授信额度侧重考虑银行收益最大化，对贷款额是否能满足企业的具体需求没有覆盖。

7.2 模型推广

本文通过一些理想化的假设，建立了银行利润模型和对中小型企业信誉评级的模型，适用领域广泛，实现了对数据属性不同、要求形式多样、限制条件多维情况下的分配方案的统筹分析，在提升资源利用率、节约生产成本、提高经济效益方面有着积极作用。

参考文献

- [1] 杨伟歧 (2011) 商业银行中小企业信贷策略研究. PhD thesis, 山东大学
- [2] 国内-国家标准-国家市场监督管理总局 CN-GB (2017) 国民经济行业分类 pp GB/T 4754-2017
- [3] Huang J, He L (2018) Application of improved pso - bp neural network in customer churn warning. Procedia Computer Science 131:1238-1246
- [4] 张卫 (2018) 关于顾客流失率的研究. 中国市场 000(35):138-139
- [5] 程维虎 (2000) 拟合优度检验的回归分析方法及其应用. 北京工业大学学报 (02):79-84
- [6] 马春明 (2014) 基于财务报表视角的邮政储蓄银行哈尔滨分行信贷风险管理研究. PhD thesis, 哈尔滨商业大学
- [7] 鲍毅莺 (2020) 贷款基础利率 lpr 改革对商业银行经营成本的影响分析. 时代金融 (上旬) 000(005):84-85
- [8] 李海涛, 高谦, 潘越 (2004) Raroc-银行绩效风向标. 银行家 000(008):114-115
- [9] 姚登举, 杨静, 詹晓娟 (2014) 基于随机森林的特征选择算法. 吉林大学学报 (工学版) (1):142-146
- [10] 李欣海 (2013) 随机森林模型在分类与回归分析中的应用. 应用昆虫学报 (昆虫知识) 50(004):001190-1197
- [11] 刘建, 吴翊, 谭璐 (2006) 对 bootstrap 方法的自助抽样的改进. 数学理论与应用 (1):69-72
- [12] 高子林, 熊江, 潘勇, 李洪兵 (2017) 基于 bootstrap 采样粒子滤波的 wirelesshart 时间同步算法. 计算机应用研究 34(3):832-834
- [13] 孙傲, 赵礼峰 (2019) 基于信息增益和基尼不纯度的 k 近邻算法. 计算机技术与发展 029(009):51-54,116
- [14] 陈肇雄, 高庆狮 (1989) 自然语言处理. 计算机研究与发展 (11):1-16
- [15] 张钺 (2007) 自然语言处理的计算模型. 中文信息学报
- [16] 李舟军, 范宇, 吴贤杰 (2020) 面向自然语言处理的预训练技术研究综述. 计算机科学 047(003):162-173

- [17] 林江豪, 阳爱民, 周咏梅, 陈锦, 蔡泽键 (2012) 一种基于朴素贝叶斯的微博情感分类. 计算机工程与科学 34(009):160–165
- [18] 王焱尧 (2011) 基于机器学习的经济行业分类方法研究. PhD thesis, 哈尔滨工业大学
- [19] 李筱瑜 (2019) 基于新词发现与词典信息的古籍文本分词研究. 软件导刊 18(4):66–69
- [20] Xiao-Yu LI (2019) Study on word segmentation in ancient texts based on neologism discovery and dictionary information. Software Guide

附录 A 数据预处理以及指标生成–MATLAB 源代码

```
%% 进项发票总数和作废数量统计
type=cost{1,1};
sumType=0;
numType=1;
numNone=0;
for i = 1:210947          % 遍历所有数据
    if cost{i,1}==type    % 如果还是这家企业
        sumType=sumType+1; % 企业的发票总数+1
        if cost{i,4}=="作废发票"
            numNone=numNone+1;
        end
    else
        disp(type);      % 监督进程
        Rate{numType,5}=sumType; % 将上一类的总数写入Rate文件
        Rate{numType,6}=numNone; % 将上一类的"作废发票"数量写入Rate文件
        numType=numType+1;
        type=cost{i,1};
        sumType=1;
        numNone=0;
    end
    if i==210947 % 特殊处理最后一类
        disp(type); % 监督进程
        Rate{numType,5}=sumType; % 将上一类的总数写入Rate文件
        Rate{numType,6}=numNone; % 将上一类的"作废发票"数量写入Rate文件
    end
end

%% 销项发票总数和作废数量统计
type=income{1,1};
sumType=0;
numType=1;
numNone=0;
for i = 1:162484          % 遍历所有数据
    if income{i,1}==type  % 如果还是这家企业
        sumType=sumType+1; % 企业的发票总数+1
        if income{i,4}=="作废发票"
            numNone=numNone+1;
        end
    else
        disp(type);      % 监督进程
        Rate{numType,7}=sumType; % 将上一类的总数写入Rate文件
        Rate{numType,8}=numNone; % 将上一类的"作废发票"数量写入Rate文件
        numType=numType+1;
    end
end
```

```

type=income{i,1};
sumType=1;
numNone=0;
end
if i==162484 % 特殊处理最后一类
disp(type); % 监督进程
Rate{numType,7}=sumType; % 将上一类的总数写入Rate文件
Rate{numType,8}=numNone; % 将上一类的"作废发票"数量写入Rate文件
end
end

%% 年成本
resCostYear=zeros(123,5); % 2016~2020
type=cost{1,1}; % 当前企业名称
numType=1; % 当前企业序号
t=strsplit(cost{1,2}, "-"); % 分隔字符串
currentYear=t(1); % 当前年
profitYear=0; % 本年利润

for i = 1:210947
% 时间转换
t=strsplit(cost{i,2}, "-");
if cost{i,1}==type % 如果还是这家企业
if t(1)==currentYear % 进行年份累加
if cost{i,4}~="作废发票" % 只计算有效发票
profitYear=profitYear-cost{i,3};
end
else % 如果到下一个年了
resCostYear(numType,str2double(t(1))-2015)=profitYear; % 储存数据，存在第type行，第numYear列
disp(str2double(t(1))-1);
currentYear=t(1); % 更新当前年
profitYear=0; % 更新本年利润
if cost{i,4}~="作废发票" % 只计算有效发票
profitYear=profitYear-cost{i,3};
end
end

else % 如果换了一个企业
disp(type); % 监督进程
resCostYear(numType,str2double(t(1))-2015)=profitYear; % 先把本企业的上一个年的数据储存
numType=numType+1; % 更新企业序号
type=cost{i,1}; % 更新企业名称
currentYear=t(1); % 更新当前年
profitYear=0; % 更新本年利润
if cost{i,4}~="作废发票" % 只计算有效发票

```

```

profitYear=profitYear-cost{i,3};
end
end

if i==210947 % 特殊处理最后一类
disp(type); % 监督进程
resCostYear(numType,str2double(t(1))-2015)=profitYear; % 先把上一个企业的最后一个月数据储存
end
end

%% 年收入
resInYear=zeros(123,5); % 2016~2020
type=income{1,1}; % 当前企业名称
numType=1; % 当前企业序数
t=strsplit(income{1,2},'-'); % 分隔字符串
currentYear=t(1); % 当前年
profitYear=0; % 本年利润

for i = 1:162484
% 时间转换
t=strsplit(income{i,2},'-');
if income{i,1}==type % 如果还是这家企业
if t(1)==currentYear % 进行年份累加
if income{i,4}~='作废发票' % 只计算有效发票
profitYear=profitYear+income{i,3};
end
else % 如果到下一个年了
resInYear(numType,str2double(t(1))-2015)=profitYear; %
    储存数据，存在第numType行，第str2double(t(1))-2015列
disp(str2double(t(1))-1); % 监督进程
currentYear=t(1); % 更新当前年
profitYear=0; % 更新本年利润
if income{i,4}~='作废发票' % 只计算有效发票
profitYear=profitYear+income{i,3};
end
end

else % 如果换了一个企业
disp(type); % 监督进程
resInYear(numType,str2double(t(1))-2015)=profitYear; % 先把本企业的上一个年的数据储存
numType=numType+1; % 更新企业序数
type=income{i,1}; % 更新企业名称
currentYear=t(1); % 更新当前年
profitYear=0; % 更新本年利润
if income{i,4}~='作废发票' % 只计算有效发票
profitYear=profitYear+income{i,3};

```



```

end
end

if i==162484 % 特殊处理最后一类
disp(type); % 监督进程
resInYear(numType,str2double(t(1))-2015)=profitYear; % 先把上一个企业的最后一个月数据储存
end
end

%% 年利润
resProfit=resInYear+resCostYear;

%% 进项负数发票数量
type=cost{1,1};
numType=1;
numNone=0;
for i = 1:210947 % 遍历所有数据
if cost{i,1}==type % 如果还是这家企业
if cost{i,3}<0
numNone=numNone+1;
end
else
disp(type); % 监督进程
Rate{numType,5}=numNone; % 将上一类的"作废发票"数量写入Rate文件
numType=numType+1;
type=cost{i,1};
numNone=0;
end
if i==210947 % 特殊处理最后一类
disp(type); % 监督进程
Rate{numType,5}=numNone; % 将上一类的"作废发票"数量写入Rate文件
end
end

%% 销项负数发票数量
type=income{1,1};
numType=1;
numNone=0;
for i = 1:162484 % 遍历所有数据
if income{i,1}==type % 如果还是这家企业
if income{i,3}<0
numNone=numNone+1;
end
else

```

```

disp(type);          % 监督进程
Rate{numType,5}=numNone; % 将上一类的"作废发票"数量写入Rate文件
numType=numType+1;
type=income{i,1};
numNone=0;
end
if i==162484 % 特殊处理最后一类
disp(type);          % 监督进程
Rate{numType,5}=numNone; % 将上一类的"作废发票"数量写入Rate文件
end
end

%% 违约率的三个参数
resBreak=zeros(4,3);
RankArr=["A" "B" "C" "D"];
for rank=1:4
sumBreak=0;
sumRankType=0;
sumObs=0;
sumNeg=0;
for i = 1:123
if Break{i,4}==RankArr(rank)
sumRankType=sumRankType+1;
sumObs=sumObs+Break{i,6}+Break{i,7};
sumNeg=sumNeg+Break{i,8}+Break{i,9};
if Break{i,5}==1
sumBreak=sumBreak+1;
end
end
end
resBreak(rank,1)=sumBreak/sumRankType;
resBreak(rank,2)=sumObs/sumRankType/2;
resBreak(rank,3)=sumNeg/sumRankType/2;
end

%% 年均利润
res=zeros(123,1);
for i=1:123
p=0;
num=0;
if Total{i,16}~=0
p=p+Total{i,16};
num=num+1;
end
if Total{i,17}~=0

```

```

p=p+Total{i,17};
num=num+1;
end
if Total{i,18}~=0
p=p+Total{i,18};
num=num+1;
end
if Total{i,19}~=0
p=p+Total{i,19};
num=num+1;
end
if Total{i,20}~=0
p=p+Total{i,20};
num=num+1;
end
res(i,1)=p/num;
end

```

附录 B 随机森林进行信誉等级评定–Python 源代码

```

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction import DictVectorizer
from sklearn.externals import joblib
from sklearn.ensemble import RandomForestClassifier
import pandas as pd

def decision():
    """
    随机森林进行信誉等级评定
    :return: acc--模型准确率
    """
    # 获取数据
    data_orig = pd.read_csv("data.csv")
    # 处理数据, 找出特征值和目标值
    x = data_orig[['是否违约(0-否, 1-是)', '进项作废比', '销项作废比', '进项负数比', '销项负数比',
                  '年均利润']]
    y = data_orig['信誉评级']
    # 分割数据集到训练集合测试集
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
    # 进行处理(特征工程)特征->类别->one_hot编码
    dic = DictVectorizer(sparse=False)
    x_train = dic.fit_transform(x_train.to_dict(orient="records"))
    print(dic.get_feature_names())
    x_test = dic.transform(x_test.to_dict(orient="records"))

```

```

# 随机森林进行预测 （超参数调优）
rf = RandomForestClassifier(n_jobs=-1)
param = {"n_estimators": [50, 120, 200, 300], "max_depth": [5, 8, 15, 25, 30, 50]}
# 网格搜索与交叉验证
gc = GridSearchCV(rf, param_grid=param, cv=2)
gc.fit(x_train, y_train)
acc = gc.score(x_test, y_test)
print("准确率: ", acc)
print("查看选择的参数模型: ", gc.best_params_)
joblib.dump(gc, 'rf.pkl')
return acc

if __name__ == "__main__":
while 1:
accuracy = decision()
if accuracy > 0.7:
break

# 获取数据
data = pd.read_csv("dataTest.csv")
# 处理数据，找出特征值
x = data[['是否违约(0-否, 1-是)', '进项作废比', '销项作废比', '进项负数比', '销项负数比',
          '年均利润']]
# 加载模型
new_rf = joblib.load('rf.pkl')
print(new_rf.predict(x))

```

附录 C 朴素贝叶斯进行文本分类—Python 源代码

```

from sklearn.datasets import fetch_20newsgroups
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report

def naviebayes():
    """
    朴素贝叶斯进行文本分类
    :return: None
    """
    news = fetch_20newsgroups(subset='all')
    # 进行数据分割
    x_train, x_test, y_train, y_test = train_test_split(news.data, news.target, test_size=0.25)
    # 对数据集进行特征抽取

```

```

tf = TfidfVectorizer()
# 以训练集中的词的列表进行每个词的重要性统计
x_train = tf.fit_transform(x_train)
print(tf.get_feature_names())
x_test = tf.transform(x_test)
# 进行朴素贝叶斯算法的预测
mlt = MultinomialNB(alpha=1.0)
print(x_train.toarray())
mlt.fit(x_train, y_train)
y_predict = mlt.predict(x_test)
print("预测的公司类别为: ", y_predict)
# 得出准确率
print("准确率为: ", mlt.score(x_test, y_test))
print("每个类别的精确率和召回率: ", classification_report(y_test, y_predict,
    target_names=news.target_names))
return None

if __name__ == "__main__":
    naviebayes()

```

附录 D 问题二中 302 家企业信誉评级结果

Algorithm 7: 分治策略递归实现斯特拉森矩阵乘法

Input:

pM1——矩阵 1;

nLeftIndex1, nTopIndex1——矩阵 1 左上角索引点（相对于源矩阵 pM1）;

nTotalCol1——矩阵 1 实际使用的列数;

pM2——矩阵 2;

nLeftIndex2, nTopIndex2——矩阵 2 左上角索引点（相对于源矩阵 pM2）;

nTotalCol2——矩阵 2 实际使用的列数;

nCount——方阵阶数 n;

pResult——存放运算结果矩阵的内存空间;

Output: 运算结果矩阵

if 当前矩阵为 2 阶方阵 **then**

└ 直接进行矩阵乘法运算;

else

└ 将两个矩阵分别拆分成 4 个大小相等的子矩阵:

└ 将矩阵 1 拆分为 $\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$; 将矩阵 2 拆分为 $\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$;

└ 递归利用斯特拉森矩阵乘法得到 7 个小矩阵:

$$M_1 = A_{11} (B_{12} - B_{22})$$

$$M_2 = (A_{11} + A_{12}) B_{22}$$

$$M_3 = (A_{21} + A_{22}) B_{11}$$

$$M_4 = A_{22} (B_{21} - B_{11})$$

$$M_5 = (A_{11} + A_{22}) (B_{11} + B_{22})$$

$$M_6 = (A_{12} - A_{22}) (B_{21} + B_{22})$$

$$M_7 = (A_{11} - A_{21}) (B_{11} + B_{12})$$

└ 使用矩阵加减法计算得出结果矩阵的四部分:

$$C_{11} = M_4 + M_5 - M_2 + M_6$$

$$C_{12} = M_1 + M_2$$

$$C_{21} = M_3 + M_4$$

$$C_{22} = M_5 + M_1 - M_3 - M_7$$

$$\text{合并结果矩阵: } Result = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

└ 释放内存;

表 1 信贷定价模型

信誉评级	贷款额度 (万元)	年利率
A	100	4%
B	67	7%
C	10	14%
D	0	-

E124	E125	E126	E127	E128	E129	E130	E131	E132	E133
B	B	B	C	B	B	C	B	B	B
E134	E135	E136	E137	E138	E139	E140	E141	E142	E143
B	B	C	B	A	C	B	C	B	A
E144	E145	E146	E147	E148	E149	E150	E151	E152	E153
B	B	B	A	C	C	B	C	A	C
E154	E155	E156	E157	E158	E159	E160	E161	E162	E163
A	A	B	B	B	B	B	C	B	B
E164	E165	E166	E167	E168	E169	E170	E171	E172	E173
B	B	B	B	C	C	C	B	B	A
E174	E175	E176	E177	E178	E179	E180	E181	E182	E183
B	B	B	B	B	B	A	B	C	B
E184	E185	E186	E187	E188	E189	E190	E191	E192	E193
C	A	A	B	C	C	B	A	B	B
E194	E195	E196	E197	E198	E199	E200	E201	E202	E203
C	A	B	C	B	C	B	B	C	B
E204	E205	E206	E207	E208	E209	E210	E211	E212	E213
B	A	C	C	B	B	B	B	C	B
E214	E215	E216	E217	E218	E219	E220	E221	E222	E223
B	C	B	B	B	B	A	C	B	B
E224	E225	E226	E227	E228	E229	E230	E231	E232	E233
C	B	B	B	B	B	B	B	A	C

E234	E235	E236	E237	E238	E239	E240	E241	E242	E243
C	C	B	B	B	B	B	B	B	C
E244	E245	E246	E247	E248	E249	E250	E251	E252	E253
B	B	B	B	B	C	B	B	C	B
E254	E255	E256	E257	E258	E259	E260	E261	E262	E263
C	B	A	B	A	B	B	B	C	B
E264	E265	E266	E267	E268	E269	E270	E271	E272	E273
B	B	B	C	C	B	B	B	B	C
E274	E275	E276	E277	E278	E279	E280	E281	E282	E283
C	B	B	C	B	C	C	C	B	C
E284	E285	E286	E287	E288	E289	E290	E291	E292	E293
B	B	C	B	B	B	B	B	C	C
E294	E295	E296	E297	E298	E299	E300	E301	E302	E303
B	B	B	C	B	B	B	B	B	B
E304	E305	E306	E307	E308	E309	E310	E311	E312	E313
B	C	B	C	B	B	B	B	A	B
E314	E315	E316	E317	E318	E319	E320	E321	E322	E323
B	A	B	B	B	B	C	B	A	B
E324	E325	E326	E327	E328	E329	E330	E331	E332	E333
B	A	C	C	B	C	C	B	C	B
E334	E335	E336	E337	E338	E339	E340	E341	E342	E343
C	C	C	C	B	C	B	C	B	C

E344	E345	E346	E347	E348	E349	E350	E351	E352	E353
A	C	B	C	A	C	B	B	B	B
E354	E355	E356	E357	E358	E359	E360	E361	E362	E363
C	C	C	A	B	C	C	C	B	B
E364	E365	E366	E367	E368	E369	E370	E371	E372	E373
B	B	C	B	B	C	C	C	B	C
E374	E375	E376	E377	E378	E379	E380	E381	E382	E383
C	C	C	C	B	B	B	B	C	B
E384	E385	E386	E387	E388	E389	E390	E391	E392	E393
C	C	C	C	C	B	B	B	C	C
E394	E395	E396	E397	E398	E399	E400	E401	E402	E403
B	B	B	C	C	C	C	C	B	C
E404	E405	E406	E407	E408	E409	E410	E411	E412	E413
B	C	B	B	C	C	A	B	C	C
E414	E415	E416	E417	E418	E419	E420	E421	E422	E423
C	C	C	C	C	C	B	C	C	B
E424	E425								
B	B								