

2022 第七届“数维杯”大学生数学建模竞赛论文

题 目 银行业中风险预测的相关研究

摘 要

自金融业发展以来，国际银行规模快速扩充，银行业也在不断扩大规模，银行的倒闭会造成严重的经济损失，产生不良的社会影响，必须要在银行倒闭之前对银行实现相关政策，我们应看到虽然针对银行倒闭的相关手段十分健全，但如果能在银行倒闭之前的就能及时预警银行的倒闭就能有效的减轻银行倒闭带来的影响，本文针对银行业的倒闭风险问题，通过对数据的分析以及相关性分析，找出能够影响银行倒闭的关键指标，建立基于神经网络的风险预测模型对银行的倒闭进行预测得到银行的倒闭概率。

针对问题一，我们先对数据进行预处理，然后对处理后的数据中的银行的 64 项指标进行提取，通过查询资料确定了根投入产出关联性较大的几个主要指标为 X_1 、 X_2 、 X_3 、 X_6 ，并利用熵权法求出了主要指标的权重，最后以此为依据构建出了银行效率计算公式，对各银行的效率进行量化和评价，构建出了倒闭和现存银行的效率图，通过这张图分析出了银行倒闭分界线为 0.60090761

针对问题二，我们运用了主成分分析法首先将数据进行标准化处理，然后计算相关系数矩阵、特征值、特征向量、贡献率以及累积贡献率，我们能从中找出 5 个可以反应 64 个指标信息的主要成分，接着根据成分矩阵表我们能得到主成分分析降维后的计算公式以及因子权重，最后可以得到综合得分表来直观展示。

针对问题三，我们在问题一和问题二结论的基础上，针对银行的 64 项指标进行筛选和剔除，通过这 64 项指标的指标之间的相关性以及他们与 class 之间的关系构建出了银行综合评分函数，利用该函数就可以对银行进行量化，然后针对这一量化数据以及筛选出来的指标建立基于神经网络的机器学习模型，就可以建立出银行风险预测模型，利用 matlab 实现了这一模型，模型在支撑材料中。

针对问题四，我们在问题三的基础上，利用问题三的机器学习模型对所有的银行进行预测，得到所有银行的倒闭概率，并根据题目要求计算得到 20 家最具代表性的现存银行以及 20 家最具代表性的倒闭银行，将所有数据保存在“支撑材料”中。

针对问题五，通过第四问综合指标数据我们可以很快地找出相同的银行。

最后，我们评价了三个问题中模型的优缺点，并给出了改进的思路。由于知识体系以及导入了很多数据，以上模型能够很好的对不同银行进行风险预测，具有较好的通用性。

关键词：熵权法 主成分分析 神经网络

目 录

一、问题重述	1
二、问题分析	1
2.1 问题一的分析	1
2.2 问题二的分析	1
2.3 问题三的分析	2
三、模型假设	2
四、符号说明	2
4.1 主要符号与说明	2
五、模型的建立与求解	3
5.1 问题一的分析 and 求解	3
5.1.1 模型一建立前的数据处理	3
5.1.2 基于熵权法的投入产出的银行效率分析模型的建立	3
5.1.3 问题一中的模型的求解	5
5.1.4 第一问的结果	5
5.2 问题二的求解和分析	5
5.2.1 对问题的分析	5
5.2.2 主成分分析模型	6
5.2.3 问题二中的模型的求解	8
5.2.4 因子权重分析	11
5.3 问题三的求解和分析	11
5.3.1 对问题的分析	11
5.3.2 基于神经网络的银行风险预测模型	12
5.3.3 对问题三的模型求解	13

5.4	问题四的求解和分析	16
5.4.1	对问题的分析	16
5.4.2	对问题四模型的建立	16
5.4.3	对问题四的模型求解	16
六、模型的评价与推广模型的评价与推广		16
6.1	模型的评价	16
6.2	模型的、模型的推广	17
参考文献		18

一、 问题重述

国际银行规模快速扩充，银行业的大规模合并、重组现象风起云涌。在经济金融一体化的总体背景下，金融市场交易规模不断扩大，金融性资金交易额大大超过实物交易量。金融全球化导致全球金融市场的竞争激烈，为适应经济金融全球化，银行业加紧调整、兼并、合并。

任务一是从 64 项数据中整理出适合的投入产出数据，评价各银行的效率，提供银行倒闭效率的分界线；任务二是利用该 64 项指标对银行倒闭的原因进行挖掘，提供最为重要的 5 项指标数据及其对应的权重；任务三是对比分析前两个任务中的银行倒闭分析结果，提出精确的倒闭风险预测模型；任务四是从 2021 年银行数据中筛选出最具代表意义的 20 家现存银行和 20 家倒闭银行，预测其它银行的倒闭风险；任务五是分析出 2017 年至 2021 年的银行数据中可能来自同一家银行的数据，结合同一家银行的时间序列数据预测呈现出倒闭的趋势的银行。

二、 问题分析

2.1 问题一的分析

首先处理附件中的 64 项数据，根据数据预处理处理部分缺失的值。通过查阅资料确定银行投入产出的主要指标，得到银行倒闭时的效率分界线。判断可作为银行投入产出的指标是否对银行倒闭有所影响并建立灰色关联度模型，通过计算指标与银行倒闭参数间的关联系数数值即得具体的评价指标。

接着，构建模糊综合评价模型，使用熵权法确定评价指标的权重，计算得到银行效率的综合指数。将综合指数与表格中的类别放至两列，以次类别为参考，统计分析银行效率，并统计不同类型对应的银行效率所在的区间。引入隶属度函数对区间的相交部分进行描述分析，区间的下限即是银行倒闭的分界线。

2.2 问题二的分析

根据问题一对附件 1 数据处理的结果，分析得到影响银行倒闭的主要因素，首先要建立主成分分析模型，通过计算影响银行倒闭的前 5 个成分

即可得到 5 个重要的指标，对 5 个成分所对应的贡献率进行归一化处理，从而得到各成分对应的权重。

2.3 问题三的分析

在问题一、二的基础上得出 64 个指标和银行是否倒闭的关系。首先，将 64 个指标作为训练集建立神经网络模型，并将集输入，训练集的输出集即为各银行的综合得分值；然后通过网络训练，得到神经网络各隐层值和权值；最后计算各银行综合得分的预测值，使用误差来分析模型是否准确。

三、模型假设

1. 模型假设一，假设银行在观测期内不会突然倒闭；
2. 模型假设二；银行不受国际形式的影响；
3. 模型假设三；银行的技术层面一致

四、符号说明

4.1 主要符号与说明

序号	符号	符号说明
1	$X1$	净利润
2	$X2$	总负债
3	$X3$	营运资金
4	$X6$	留存收益
5	X_i	指标
6	Y_i	标准化指标
7	X	数据样本资料
8	m	变量的个数
9	n	样本的个数
10	Z	方差矩阵
11	μ_i	每一列的平均值
12	σ_i^2	每一列的方差
13	z_{ji}	标准化处理后的数据
14	r_{ij}	相关系数
15	R	相关系数矩阵

五、模型的建立与求解

5.1 问题一的分析与求解

5.1.1 模型一建立前的数据处理

首先我们通过预处理使用 python 语言得到了附件 1 中 2017 和 2021 表的数据的 64 个指标之间的相关性如图，根据这 64 个指标的相关性，表 2017 以及表 2021 的相关性数据在支撑材料中，我们先将相关性高的变量进行线性拟合，通过拟合的数据来预测有“?” 的值，最后将所有有“?” 的区域的价值都填上去，得到一个处理好的数据。

但是有些指标的相关性高的指标也存在丢失数值的情况，我们采用取平均值的办法，将相关性高的一些列指标都填上了数据，支撑材料中的数据就是我们处理好的数据。

5.1.2 基于熵权法的投入产出的银行效率分析模型的建立

银行效率是判别一个是否在正常运作的一个指标，问题一要求从 64 个指标中整理出合适的指标投入产出的数据，选取出能够衡量银行效率的指标，并提供银行倒闭的分界线，我们先对已有的数据进行标准化和归一化，然后利用熵权法对整理出来的指标赋予相应的权值，利用这些权值构建效率函数，利用效率函数对所有的数据样本进行效率评估，最终得到一个银行效率图，根据 class 值的不同我们区分出了破产的和还在运营的银行，利用中值法将银行倒闭效率的分界线求出来。

我们通过根据王新洋等人的研究，我们将 64 个指标中对该问计算相对效率的投入产出的相关指标提取出来，分别是

表 5-1: 投入产出指标划分表

指标类型	指标定义	指标符号
投入指标	净利润	X1
产出指标	总负债	X2
	营运资金	X3
	留存收益	X6

根据上述确定的投入产出指标构建综合评价体系，首先需要各个评价指标的权重，再进行量化，具体的算法如下：

先将各个指标的数据进行标准化处理。一共有 3 个指标，其中

$$X_i = x_1, x_2, \dots, x_n \quad (1)$$

对上述指标标准化后的值为

$$Y_1, Y_2, \dots, Y_n \quad (2)$$

就可以得到标准化公式：

$$Y_{ij} = \frac{x_{ij} - \min(x_i)}{\max(x_i) - \min(x_i)} \quad (3)$$

根据信息论中的熵的定义，一组数据的信息熵为：

$$E_j = -\frac{1}{\ln n} \sum_{i=1}^n P_{ij} \ln P_{ij} \quad (4)$$

其中， $P_{ij} = \frac{Y_{ij}}{\sum_{i=1}^n Y_{ij}}$ ，若 $P_{ij} = 0$ 则 $\lim_{P_{ij} \rightarrow 0} P_{ij} \ln P_{ij} = 0$

根据信息熵的计算公式，计算出各个指标的信息熵为

$$E_1, E_2, \dots, E_n \quad (5)$$

通过信息熵计算各指标的权重：

$$W_i = \frac{1 - E_i}{k - \sum E_i} (i = 1, 2, \dots, k) \quad (6)$$

根据组合权法可得到银行的效率指数为 Z_i ，计算公式为

$$Z_i = \sum_{n=1}^i x_i w_i \quad (7)$$

其中, x_i 为 i 个不同的指标, w_i 为 i 个不同的指标权重, Z_i 为不同银行的效率

通过附件一中的 class 可以知, 可以将所有银行数据分为两个不同的集合, 设 class 值为 0 的集合为 A, class 值为 1 的集合为 B, 根据数学知识易知 $A \cap B$ 的下限为银行倒闭的分界线。

5.1.3 问题一中的模型的求解

根据上述模型的结论, 我们提取出 $X1$ 、 $X2$ 、 $X3$ 、 $X6$ 、 $class$, 我们对这些数据代入上述模型, 通过 excel 文件得到熵权值为

相关指标	权值 W
$X1$	0.33
$X2$	0.32
$X3$	0.32
$X6$	0.02

表 5-2: 相关指标的权值

然后将上述指标的权值带入公式 (7) 中得到:

$$Z_i = 0.33 * x_{1i} + 0.32 * x_{2i} + 0.32 * x_{3i} + 0.02 * x_{6i} \quad (8)$$

其中, $x_{1i} \in X_1$ 、 $x_{2i} \in X_2$ 、 $x_{3i} \in X_3$ 、 $x_{6i} \in X_6$

我们使用 python 所有的 Z_i 算出来, 保存在支撑文件中, 列名为 "score", 然后我们将 Z_i 和 i 对应的 class 值带入到画图函数中, 得到下图

可以看出在观测期内破产的银行与观测期内没破产的银行在图像上有明显的差异, 我可以以 class 值为 0 的那组数据的下界为银行破产效率的分界线为 $Z = 0.60090761$

5.1.4 第一问的结果

5.2 问题二的求解和分析

5.2.1 对问题的分析

根据问题二的分析, 我们首先要得到影响银行倒闭的重要因素, 则必须要确定可以综合反应出 64 个指标所反应的信息的指标。如果用相关性分

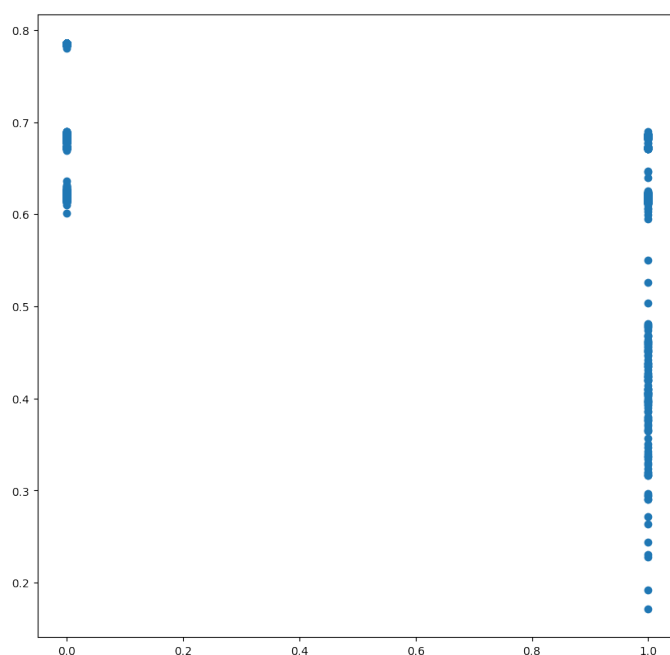


图 5-1: 数据效率评价图

析找出的重要指标会远不止 5 个，所以我们选用了主成分分析法。主成分分析的原理是设法将原来变量重新组合成一组新的相互无关的几个综合变量，同时根据实际需要从中可以取出几个较少的总和变量尽可能多地反映原来变量的信息的统计方法叫做主成分分析或称主分量分析，也是数学上处理降维的一种方法。故此建立主成分分析模型，在对附件一数据降维的同时确定主要因素。

5.2.2 主成分分析模型

(1) 模型的准备

1) 主成分与原始变量之间的关系

主成分是原始变量的线性组合；主成分的数量相对于原始数量更少；主成分保留了原始变量的大部分信息；主成分之间相互独立。

2) 数据处理

对问题一处理后的数据进行带入，然后对数据进行标准化处理。

(2) 模型的建立

1) 根据标准化后的数据集计算协方差矩阵 Z

设数据样本资料 X ，其中有 m 个变量，共 n 个样本。

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \quad (9)$$

计算每一列的平均值 $\mu_i = \frac{1}{n} \sum_{j=1}^n x_{ji}$

每一列的方差 $\sigma_i^2 = \frac{1}{n-1} \sum_{j=1}^n (x_{ji} - \mu_i)^2$

对数据进行标准化处理， $z_{ji} = \frac{x_{ji} - \mu_i}{\sigma_i}$

得到矩阵 Z ：

$$Z = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1m} \\ z_{21} & z_{22} & \dots & z_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \dots & z_{nm} \end{bmatrix} \quad (10)$$

计算相关系数， $r_{ij} = \frac{\sum_{k=1}^n z_{ki} * z_{kj}}{n-1}$, ($i, j = 1, 2, \dots, m$)

得到相关系数矩阵 R

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mm} \end{bmatrix} \quad (11)$$

r_{ij} 即 X 矩阵的第 i 列的样本序列和第 j 列的样本序列之间的相关关系，其值为-1 到 1 之间，且 R 矩阵应该为对称矩阵，即 $r_{ij} = r_{ji}$ 。

相关系数程度区别例如下面两个表所示：

2) 计算 R 的特征值及对应的特征向量

协方差矩阵 Σ 是实对称阵，知其特征值为非负，不妨设其特征值 $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_p \geq 0$ ，它们对应的正交化后的单位特征向量如下：

相关系数 r	相关性
$r > 0$	正线性相关
$r = 0$	线性无关
$r < 0$	正线性相关

表 5-3: 正负相关性

相关系数绝对值	相关性程度
1	完全相关
$[0.8, 1)$	高度相关
$[0.5, 0.8)$	中度相关
$[0.3, 0.5)$	低度相关
$[0, 0.3)$	不相关

表 5-4: 相关性程度大小

$$a_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{p1} \end{bmatrix}; a_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{p2} \end{bmatrix}; \dots; a_p = \begin{bmatrix} a_{1p} \\ a_{2p} \\ \vdots \\ a_{pp} \end{bmatrix} \quad (12)$$

若原先 X 的各个列代表的指标变量,合成向量,记为 $Var = [Var_1, Var_2, \dots, Var_p]^T$, 则有 X 的第 i 个主成分为

$$F_i = (a_i)^T Var = a_{1i} * Var_1 + a_{2i} * Var_2 + \dots + a_{pi} * Var_p \quad (13)$$

3) 计算每个成分的贡献率和累计贡献率

$$j = \frac{\lambda_j}{\sum_{k=1}^p \lambda_k} (j = 1, 2, \dots, n) \quad (14)$$

$$\alpha_p = \frac{\sum_{k=1}^p \lambda_k}{\sum_{k=1}^n \lambda_k} (p \leq n)$$

主成分数量的选取规则是根据累积贡献率确定,一般要求累积贡献率达到 0.85 以上,这样能保证新变量能包括原始变量的绝大多数信息。

5.2.3 问题二中的模型的求解

根据上述所建立的模型求得成分矩阵

成分矩阵 ^a					
	成分				
	1	2	3	4	5
Zscore(VAR00001)	.003	.003	.820	.463	.122
Zscore(VAR00002)	-.003	-.007	.331	-.913	-.045
Zscore(VAR00003)	.006	.010	-.149	.824	.140
Zscore(VAR00004)	.048	.984	.001	-.003	-.004
Zscore(VAR00005)	.000	.001	-.003	.015	.003
Zscore(VAR00006)	.003	.002	.235	.817	.088
Zscore(VAR00007)	.003	.003	.836	-.071	.150
Zscore(VAR00008)	.031	.917	.001	-.002	-.006
Zscore(VAR00009)	-.007	.011	-.115	.129	-.030
Zscore(VAR00010)	.003	.007	-.676	.733	.007
Zscore(VAR00011)	.002	.003	.934	.263	.125
Zscore(VAR00012)	.041	.923	.008	-.009	-.003
Zscore(VAR00013)	.922	-.033	.002	-.003	.005
Zscore(VAR00014)	.003	.003	.836	-.071	.150
Zscore(VAR00015)	.000	-.001	.000	-.001	.000
Zscore(VAR00016)	.034	.978	.002	-.003	-.006
Zscore(VAR00017)	.031	.916	.001	-.002	-.006
Zscore(VAR00018)	.007	.003	.830	-.071	.152
Zscore(VAR00019)	.891	-.009	-.003	.014	.003
Zscore(VAR00020)	.990	-.040	-.001	-.006	-.002
Zscore(VAR00021)	.000	.000	.000	.000	.000
Zscore(VAR00022)	.002	.003	.937	.254	.124
Zscore(VAR00023)	.910	-.013	-.004	.016	.002
Zscore(VAR00024)	.002	.003	.634	.303	.121
Zscore(VAR00025)	.004	.007	-.640	.681	.044
Zscore(VAR00026)	.034	.970	.002	-.003	-.006
Zscore(VAR00027)	.000	.005	.003	.004	.003
Zscore(VAR00028)	.000	.017	-.115	-.111	.959
Zscore(VAR00029)	-.015	-.030	.041	.044	-.006
Zscore(VAR00030)	.984	-.040	-.001	-.008	-.002
Zscore(VAR00031)	.891	-.009	-.002	.013	.002
Zscore(VAR00032)	-.003	-.001	.010	-.064	-.013
Zscore(VAR00033)	.034	.965	.000	.000	-.005
Zscore(VAR00034)	.034	.978	.000	-.003	-.005
Zscore(VAR00035)	.002	.002	.933	.258	.122
Zscore(VAR00036)	-.003	.001	-.917	-.233	-.120
Zscore(VAR00037)	.000	.001	.000	.002	.000
Zscore(VAR00038)	.003	.005	-.660	.733	-.009
Zscore(VAR00039)	.960	-.038	-.001	-.004	-.001
Zscore(VAR00040)	.056	.969	.001	-.005	-.004
Zscore(VAR00041)	.000	.000	.000	.000	.000
Zscore(VAR00042)	.957	-.039	.000	-.003	.000
Zscore(VAR00043)	.987	-.040	-.001	-.006	-.002
Zscore(VAR00044)	.986	-.040	-.001	-.006	-.002
Zscore(VAR00045)	.000	.001	.000	.002	.001
Zscore(VAR00046)	.048	.984	.001	-.004	-.004
Zscore(VAR00047)	.000	-.001	.000	-.017	-.005
Zscore(VAR00048)	.002	.002	.934	.256	.123
Zscore(VAR00049)	-.980	.039	.002	.006	.002
Zscore(VAR00050)	.034	.986	.001	-.004	-.005
Zscore(VAR00051)	-.006	-.009	.146	-.820	-.137
Zscore(VAR00052)	-.001	-.001	.028	-.170	-.032
Zscore(VAR00053)	.001	.017	-.424	-.196	.854
Zscore(VAR00054)	.000	.016	-.391	-.188	.878
Zscore(VAR00055)	.000	.003	.002	.022	.005
Zscore(VAR00056)	-.990	.040	.001	.006	.002
Zscore(VAR00057)	.000	.001	.001	.003	.003
Zscore(VAR00058)	.990	-.040	-.001	-.006	-.002
Zscore(VAR00059)	.000	-.001	.000	-.002	-.002
Zscore(VAR00060)	.000	.000	.000	.000	.000
Zscore(VAR00061)	-.002	.005	-.024	.024	-.016
Zscore(VAR00062)	.931	-.038	.000	-.011	-.003
Zscore(VAR00063)	.033	.967	.000	.001	-.005
Zscore(VAR00064)	.000	.006	-.276	-.162	.913

图 5-2: 成分矩阵图

由图可知,第一个主成分中: $X_{13}, X_{19}, X_{20}, X_{23}, X_{30}, X_{31}, X_{39}, X_{42}, X_{43}, X_{44}, X_{49}, X_{56}, X_{58}$, 较大,说明第一主成分基本反应了这些指标的信息。

第二个主成分中: $X_4, X_8, X_{12}, X_{16}, X_{17}, X_{26}, X_{33}, X_{34}, X_{40}, X_{46}, X_{50}, X_{63}$ 较大,说明了第二主成分基本反应了这些指标的信息。

第三个主成分中： $X_1, X_7, X_{11}, X_{14}, X_{18}, X_{22}, X_{24}, X_{35}, X_{36}, X_{48}$ 较大，说明了第三主成分基本反应了这些指标的信息。

第四个主成分中： $X_2, X_3, X_6, X_{10}, X_{25}, X_{38}, X_{51}$ 较大，说明了第四主成分基本反应了这些指标的信息。

第五个主成分中： X_{53}, X_{64} 较大，说明了第五主成分基本反应了这些指标的信息。

在知道每个主成分所反应的指标后，我们可以得出主成分的贡献率：

总方差解释						
成分	总计	初始特征值		提取载荷平方和		
		方差百分比	累积 %	总计	方差百分比	累积 %
1	12.807	20.011	20.011	12.807	20.011	20.011
2	11.118	17.372	37.383	11.118	17.372	37.383
3	9.447	14.761	52.144	9.447	14.761	52.144
4	5.199	8.123	60.267	5.199	8.123	60.267
5	3.481	5.439	65.706	3.481	5.439	65.706
6	2.120	3.312	69.018			
7	1.692	2.644	71.662			
8	1.504	2.350	74.012			
9	1.438	2.247	76.259			
10	1.405	2.196	78.455			
11	1.049	1.640	80.095			
12	1.028	1.606	81.700			
13	1.000	1.563	83.263			
14	1.000	1.562	84.826			
15	.997	1.558	86.383			
16	.976	1.524	87.908			
17	.957	1.496	89.404			
18	.954	1.490	90.894			
19	.895	1.398	92.292			
20	.849	1.327	93.618			
21	.604	.944	94.562			
22	.570	.890	95.452			
23	.545	.852	96.304			
24			

图 5-3: 主成分贡献图

根据成分矩阵图和主成分贡献图我们可以计算 5 个主成分中每个指标对应的系数：

分别计算新的特征向量 W_1, W_2, W_3

$$W_1 = \text{VAR00065} / \text{SQR}(12.807)$$

$$W_2 = \text{VAR00066} / \text{SQR}(11.118)$$

$$W_3 = \text{VAR00065} / \text{SQR}(9.447)$$

$$W_4 = \text{VAR00065} / \text{SQR}(5.199)$$

$$W_5 = \text{VAR00065} / \text{SQR}(3.481)$$

通过上述可以计算出主成分分析降维后的公式：

令成分矩阵为 W_i , 则

VAR0006	VAR0006	VAR0006	VAR0006	VAR0006	W1	W2	W3	W4	W5
5	6	7	8	9					
00	02	-12	-11	36	00	01	-04	-05	51
-02	-03	04	04	-01	00	-01	01	02	00
98	-04	00	-01	00	27	-01	00	00	00
89	-01	00	01	00	25	00	00	01	00
00	00	01	-06	-01	00	00	00	-03	-01
03	97	00	00	-01	01	29	00	00	00
03	98	00	00	-01	01	29	00	00	00
00	00	03	26	12	00	00	30	11	07
00	00	-02	-23	-12	00	00	-30	-10	-06
00	00	00	00	00	00	00	00	00	00
00	01	-06	73	-01	00	00	-21	32	00
96	-04	00	00	00	27	-01	00	00	00
06	97	00	-01	00	02	29	00	00	00
00	00	00	00	00	00	00	00	00	00
96	-04	00	00	00	27	-01	00	00	00
99	-04	00	-01	00	28	-01	00	00	00
99	-04	00	-01	00	28	-01	00	00	00
00	00	00	00	00	00	00	00	00	00
05	98	00	00	00	01	30	00	00	00
00	00	00	-02	-01	00	00	00	-01	00
00	00	03	26	12	00	00	30	11	07
-98	04	00	01	00	-27	01	00	00	00
03	99	00	00	-01	01	30	00	00	00
-01	-01	15	-02	-14	00	00	05	-36	-07
00	00	03	-17	-03	00	00	01	-07	-02
00	02	-42	-20	85	00	01	-14	-09	46
00	02	-39	-19	88	00	00	-13	-08	47
00	00	02	01	00	00	00	00	01	00
-99	04	00	01	00	-28	01	00	00	00
00	00	00	00	00	00	00	00	00	00
99	-04	00	-01	00	28	-01	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	01	-02	02	-02	00	00	-01	01	-01
93	-04	00	-01	00	26	-01	00	00	00
03	97	00	00	-01	01	29	00	00	00
00	01	-28	-16	91	00	00	-09	-07	49

图 5-4: 新特征向量图

$$W_i = \begin{bmatrix} w_{1i} \\ w_{2i} \\ \vdots \\ w_{ji} \end{bmatrix} (i = 1, 2, \dots, 5) (j = 1, 2, \dots, 64) \quad (15)$$

各指标矩阵为 $A = [X_1, X_2, \dots, X_j]$

则 $F_i = A * W_i$ 。

综上可得 $F = (0.20011/0.65706) * F_1 + (0.17372/0.65706) * F_2 + (0.14761/0.65706) * F_3 + (0.08123/0.65706) * F_4 + (0.05439/0.65706) * F_5$

5.2.4 因子权重分析

主成分分析的权重计算结果显示，主成分 1 的权重为 0.34、主成分 2 的权重为 0.26、主成分 3 的权重为 0.22、主成分 4 的权重为 0.12、主成分 5 的权重为 0.08。

5.3 问题三的求解和分析

5.3.1 对问题的分析

根据前面两问的数据以及概念，根据题意以及前两问是否倒闭与 64 个指标之间的关系，首先建立神经网络模型以 64 个指标为训练集，输入集以各银行所对应的综合得分为训练集的输出集，通过网络训练从而得到神经

名称 权重	方差解释率	累计方差解释率
主成分 1 0.34	0.20011	0.20011
主成分 2 0.26	0.17372	0.37383
主成分 3 0.22	0.14761	0.52144
主成分 4 0.12	0.08123	0.60267
主成分 5 0.08	0.05439	0.65706

表 5-5: 主成分权重结果

网络各个隐含层的权值，通过预测各银行的综合得分预测值来分析该模型的准确性。

5.3.2 基于神经网络的银行风险预测模型

BP 神经网络通常由三层组成，即输入层、隐含层、输出层。在已明确输出向量的情况下，选择合适的输入向量尤为重要。影响综合得分的主要有 38 个指标，而且 38 个指标又相互影响，在对综合得分进行预测的时候，选择 X_1 、 X_2 、 X_3 等等一系列变量作为输入层，综合得分作为输出层，同样选择单隐含层网络结构。

BP 神经网络预测前要先训练网络，通过训练使网络具有联想记忆和预测能力。BP 神经网络训练首先要进行网络参数的初始化，主要包括输入层和隐含层之间权值的初始化、隐含层和输出层之间权值的初始化以及隐含层和输出层的阈值初始化。通过隐含层、输出层的计算得到网络预测结果和期望结果之间的误差，网络再根据误差结果对初始化权值和阈值进行更新，所以最终确定的权值和阈值会受到其初始权值和阈值选择的影响，进而影响到网络模型的收敛速度和预测精度。遗传算法从随机解出发，通过迭代寻找初始权值和阈值最优解，可以克服 BP 神经网络在自动生成初始权值和阈值的随机性。因此，我们将遗传算法与 BP 神经网络相结合，利用遗传算法优化 BP 神经网络的初始权值和阈值，得到一个更稳定可靠的网络结构。

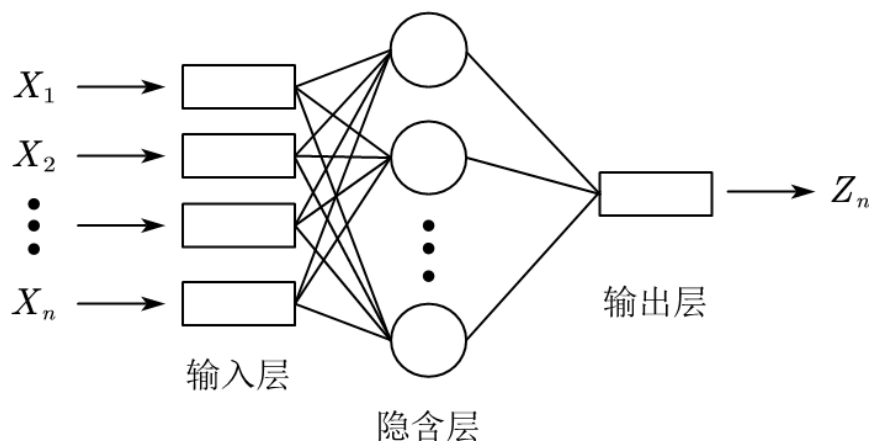


图 5-5: 网络结构

根据上面两问我们可以得出关于银行综合得分的公式：

$$Q_j = \sum_{i=1}^{64} \omega_i x_{ij}, j = 0, 1, 2, \dots, N \quad (16)$$

其中， Q_i 为银行综合评分、 ω_i 为 64 个指标的权重、 x_{ij} 为数据集

下面的权重是我们根据第一二问的模型得出来的结论，上式中的 ω_i 的取值均来自下表

通过上述结论以及公式可以得出，输入层以及输出层的相关指标，以此来建立神经网络预测模型，具体如下：

计算过程如下：

通过上述神经网络的模型，将 38 个银行的指标与银行的综合得分进行训练，令各银行的综合得分值为 Z ，通过上述模型训练出来的神经网络预测综合得分为 Z' ，则相对误差为 δ ：

$$\delta = \frac{|Z - Z'|}{Z} \times 100\% \quad (17)$$

5.3.3 对问题三模型求解

通过上述神经网络模型预测各个银行的综合分数，我们使用 matlab 程序编写该程序，详细程序可见附件以及支撑材料，我们设置各项参数如下：

相关指标	权值 W	相关指标	权值 W
X1	0.03	X33	0.00
X2	0.03	X34	0.01
X3	0.03	X35	0.01
X4	0.00	X36	0.00
X5	0.00	X37	0.02
X6	0.01	X38	0.00
X7	0.00	X39	0.00
X8	0.01	X40	0.01
X9	0.00	X41	0.01
X10	0.00	X42	0.01
X11	0.00	X43	0.16
X12	0.01	X44	0.00
X13	0.01	X45	0.00
X14	0.00	X46	0.01
X15	0.01	X47	0.00
X16	0.01	X48	0.00
X17	0.01	X49	0.00
X18	0.00	X50	0.01
X19	0.00	X51	0.01
X20	0.00	X52	0.01
X21	0.06	X53	0.02
X22	0.01	X54	0.02
X23	0.00	X55	0.01
X24	0.01	X56	0.00
X25	0.01	X57	0.00
X26	0.01	X58	-0.01
X27	0.01	X59	0.02
X28	0.02	X60	0.00
X29	0.03	X61	0.00
X30	0.01	X62	0.00
X31	0.00	X63	0.01
X32	0.01	X64	0.02

表 5-6: 相关指标的权值

经过一段时间的训练后得到如图所示的结果，我们可以看到该神经网络的相关参数变化，如下

最终我们得到了训练好的模型与原数据的图片，可见我们的模型在一

相关参数	设置值
<i>net.trainParam.show</i>	10000
<i>net.trainParam.Lr</i>	0.05
<i>net.trainParam.epochs</i>	50000
<i>net.trainParam.goal</i>	0.78×10^{-3}

表 5-7: 机器学习相关参数表

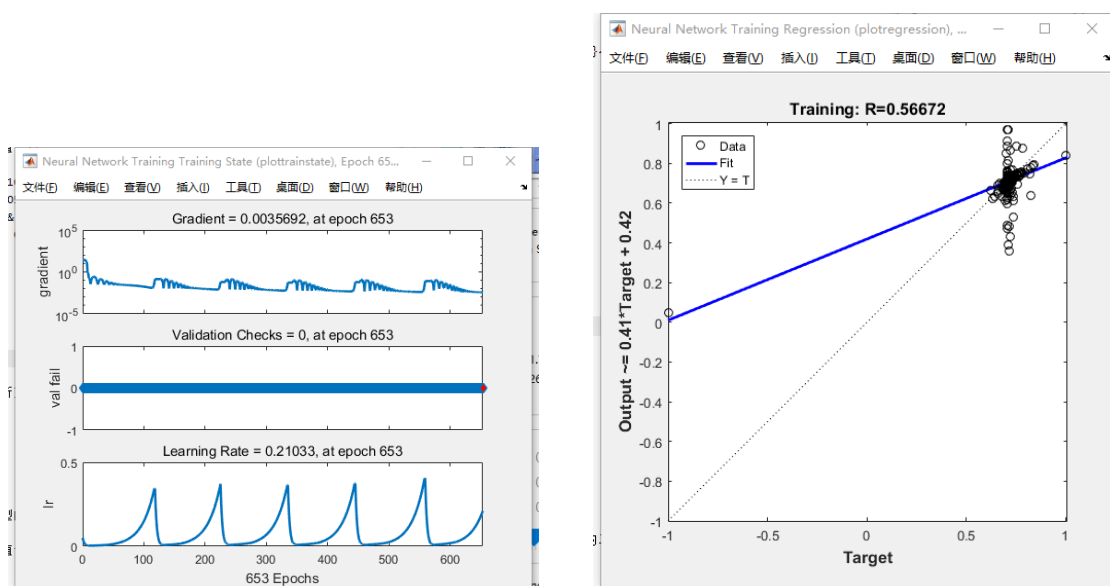


图 5-6: 相关参数变化

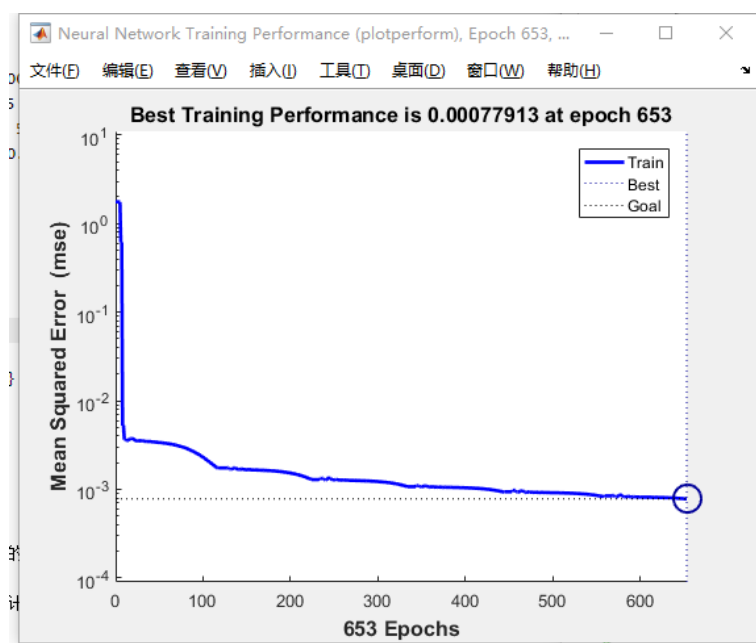


图 5-7: Epochs 与误差的关系

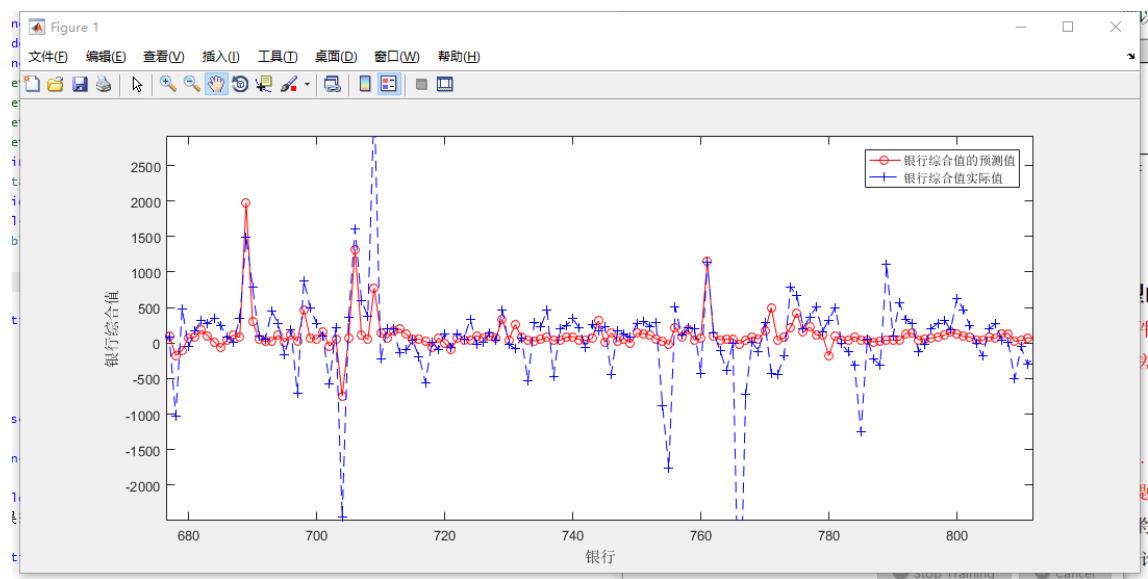


图 5-8: 预测值与真实值之间的函数图像

定范围内是准确的，但并不是, 我们根据问题一以及问题二建立的倒闭银行的分界线，通过上述的预测模型就可以实现对银行的风险预测，当银行的预测值超过问题一所给出的分界线，就可以使用该模型对银行的倒闭风险进行评估，可以很好的适用于各类银行的风险检测。

5.4 问题四的求解和分析

5.4.1 对问题的分析

根据问题三所建立的模型，我们可以对所有银行的风险进行评估，如何根据综合分数给所有银行进行分析，得到最具有代表性的 20 家现存银行以及 20 家倒闭银行。

5.4.2 对问题四模型的建立

沿用问题三中的模型

5.4.3 对问题四的模型求解

通过对模型的求解以及数据的处理得到支撑材料中问题四文件夹的相关数据，

六、 模型的评价与推广模型的评价与推广

6.1 模型的评价

1. 优点

(1) 问题求解之前就对数据进行了预处理, 针对存在问题的数据进行了科学的补全, 而不是将数据空着, 能将建模的总体思路以及建模过程中的一些细节描述出来。

(2) 在问题三中, 根据问题一以及问题二的结论为问题三中的主要指标的确定提供了基础, 从前往后, 模型比较有说服力。

(3) 运用多种数学软件 (如 MATLAB、SPSS) 以及编程语言 (如 Python 等), 取长补短, 使计算结果更加准确、明晰。

2. 缺点

(1) 问题一中的投入产出指标通过查阅文献得到, 应该使用灰度关联模型对所有指标的关联度进行分析, 得到的结果将会更加明确且科学。

(2) 问题三中机器学习模型的准确度还不够。

6.2 模型的、模型的推广

1. 问题二中建立的模型在现实生活中可以帮助人们快速对指标进行相关性分析。

2. 基于问题三建立的模型, 可以根据银行的经营情况得到该银行的倒闭风险概率。

参考文献

- [1] 余燕丽. 我国商业银行风险现状及对策研究 [D]. 武汉理工大学,2003.
- [2] 温秀玲. 我国商业银行风险、收益对效率影响的实证分析 [D]. 吉林大学,2013.
- [3] Charlotte P. Alhadeff, David A. Alhadeff, Recent Bank Mergers, The Quarterly Journal of Economics, Volume 69, Issue 4, November 1955, Pages 503–532,
- [4] Scientific Platform Serving for Statistics Professional 2021. SPSSPRO. (Version 1.0.11)[Online Application Software]. Retrieved from <https://www.spsspro.com>.
- [5] Kar Yan Tam, Melody Y. Kiang, (1992) Managerial Applications of Neural Networks: The Case of Bank Failure Predictions. Management Science 38(7):926-947.
- [6] 王新洋. 我国商业银行的效率评价 [D]. 天津财经大学,2016.
- [7] 吴静鸣. 我国商业银行风险的早期预警模型研究 [D]. 厦门大学,2003.

附录

程序一：Python 银行效率求解：

```
{
  "cells": [
    {
      "cell_type": "code",
      "execution_count": 2,
      "metadata": {},
      "outputs": [
        {
          "data": {
            "text/html": [
              "<div>\n",
              "<style scoped>\n",
              "    .dataframe tbody tr th:only-of-type {\n",
              "        vertical-align: middle;\n",
              "    }\n",
              "\n",
              "    .dataframe tbody tr th {\n",
              "        vertical-align: top;\n",
              "    }\n",
              "\n",
              "    .dataframe thead th {\n",
              "        text-align: right;\n",
              "    }\n",
              "</style>\n",
              "<table border='1' class='dataframe'>\n",
              "  <thead>\n",
              "    <tr style='text-align: right;'>\n",
              "      <th></th>\n",
              "      <th>X1</th>\n",
              "      <th>X2</th>\n",
              "      <th>X3</th>\n",
              "      <th>X4</th>\n",
              "      <th>X5</th>\n",
              "      <th>X6</th>\n",
              "      <th>X7</th>\n",
              "      <th>X8</th>\n",
              "      <th>X9</th>\n",
              "      <th>X10</th>\n",
              "      <th>...</th>\n",
            ]
          }
        ]
      }
    }
  ]
}
```

```

"      <th>X57</th>\n",
"      <th>X58</th>\n",
"      <th>X59</th>\n",
"      <th>X60</th>\n",
"      <th>X61</th>\n",
"      <th>X62</th>\n",
"      <th>X63</th>\n",
"      <th>X64</th>\n",
"      <th>class</th>\n",
"      <th>score</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>0.088238</td>\n",
"      <td>0.55472</td>\n",
"      <td>0.01134</td>\n",
"      <td>1.02050</td>\n",
"      <td>-66.5200</td>\n",
"      <td>0.342040</td>\n",
"      <td>0.109490</td>\n",
"      <td>0.577520</td>\n",
"      <td>1.08810</td>\n",
"      <td>0.320360</td>\n",
"      <td>...</td>\n",
"      <td>0.275430</td>\n",
"      <td>0.91905</td>\n",
"      <td>0.002024</td>\n",
"      <td>7.2711</td>\n",
"      <td>4.7343</td>\n",
"      <td>142.760</td>\n",
"      <td>2.5568</td>\n",
"      <td>3.25970</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>-0.006202</td>\n",
"      <td>0.48465</td>\n",
"      <td>0.23298</td>\n",
"      <td>1.59980</td>\n",
"      <td>6.1825</td>\n",
"      <td>0.000000</td>\n",

```

```

"      <td>-0.006202</td>\n",
"      <td>1.063400</td>\n",
"      <td>1.27570</td>\n",
"      <td>0.515350</td>\n",
"      <td>...</td>\n",
"      <td>-0.012035</td>\n",
"      <td>1.00470</td>\n",
"      <td>0.152220</td>\n",
"      <td>6.0911</td>\n",
"      <td>3.2749</td>\n",
"      <td>111.140</td>\n",
"      <td>3.2841</td>\n",
"      <td>3.37000</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>0.130240</td>\n",
"      <td>0.22142</td>\n",
"      <td>0.57751</td>\n",
"      <td>3.60820</td>\n",
"      <td>120.0400</td>\n",
"      <td>0.187640</td>\n",
"      <td>0.162120</td>\n",
"      <td>3.059000</td>\n",
"      <td>1.14150</td>\n",
"      <td>0.677310</td>\n",
"      <td>...</td>\n",
"      <td>0.192290</td>\n",
"      <td>0.87604</td>\n",
"      <td>0.000000</td>\n",
"      <td>8.7934</td>\n",
"      <td>2.9870</td>\n",
"      <td>71.531</td>\n",
"      <td>5.1027</td>\n",
"      <td>5.61880</td>\n",
"      <td>0</td>\n",
"      <td>2</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>-0.089951</td>\n",
"      <td>0.88700</td>\n",
"      <td>0.26927</td>\n",

```



```

"      <td>1.52220</td>\n",
"      <td>-55.9920</td>\n",
"      <td>-0.073957</td>\n",
"      <td>-0.089951</td>\n",
"      <td>0.127400</td>\n",
"      <td>1.27540</td>\n",
"      <td>0.113000</td>\n",
"      <td>...</td>\n",
"      <td>-0.796020</td>\n",
"      <td>0.59074</td>\n",
"      <td>2.878700</td>\n",
"      <td>7.6524</td>\n",
"      <td>3.3302</td>\n",
"      <td>147.560</td>\n",
"      <td>2.4735</td>\n",
"      <td>5.92990</td>\n",
"      <td>0</td>\n",
"      <td>3</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>0.048179</td>\n",
"      <td>0.55041</td>\n",
"      <td>0.10765</td>\n",
"      <td>1.24370</td>\n",
"      <td>-22.9590</td>\n",
"      <td>0.000000</td>\n",
"      <td>0.059280</td>\n",
"      <td>0.816820</td>\n",
"      <td>1.51500</td>\n",
"      <td>0.449590</td>\n",
"      <td>...</td>\n",
"      <td>0.107160</td>\n",
"      <td>0.77048</td>\n",
"      <td>0.139380</td>\n",
"      <td>10.1180</td>\n",
"      <td>4.0950</td>\n",
"      <td>106.430</td>\n",
"      <td>3.4294</td>\n",
"      <td>3.36220</td>\n",
"      <td>0</td>\n",
"      <td>4</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>...</th>\n",

```



```

"    </tr>\n",
"    <tr>\n",
"        <th>26638</th>\n",
"        <td>-0.013359</td>\n",
"        <td>0.58354</td>\n",
"        <td>-0.02265</td>\n",
"        <td>0.92896</td>\n",
"        <td>-42.2320</td>\n",
"        <td>-0.013359</td>\n",
"        <td>-0.015036</td>\n",
"        <td>0.562890</td>\n",
"        <td>0.98904</td>\n",
"        <td>0.328470</td>\n",
"        <td>...</td>\n",
"        <td>-0.040671</td>\n",
"        <td>1.01110</td>\n",
"        <td>0.805920</td>\n",
"        <td>10.5990</td>\n",
"        <td>7.1740</td>\n",
"        <td>94.092</td>\n",
"        <td>3.8792</td>\n",
"        <td>1.75720</td>\n",
"        <td>1</td>\n",
"        <td>26638</td>\n",
"    </tr>\n",
"    <tr>\n",
"        <th>26639</th>\n",
"        <td>0.006338</td>\n",
"        <td>0.50276</td>\n",
"        <td>0.43923</td>\n",
"        <td>1.87360</td>\n",
"        <td>9.7417</td>\n",
"        <td>0.006338</td>\n",
"        <td>0.012022</td>\n",
"        <td>0.983560</td>\n",
"        <td>1.00830</td>\n",
"        <td>0.494490</td>\n",
"        <td>...</td>\n",
"        <td>0.012817</td>\n",
"        <td>0.99174</td>\n",
"        <td>0.000000</td>\n",
"        <td>10.4700</td>\n",
"        <td>6.0759</td>\n",
"        <td>51.019</td>\n",
"        <td>7.1542</td>\n",

```

```

"      <td>62.00100</td>\n",
"      <td>1</td>\n",
"      <td>26639</td>\n",
"    </tr>\n",
"  <tr>\n",
"    <th>26640</th>\n",
"    <td>-0.041643</td>\n",
"    <td>0.84810</td>\n",
"    <td>-0.12852</td>\n",
"    <td>0.57485</td>\n",
"    <td>-121.9200</td>\n",
"    <td>0.000000</td>\n",
"    <td>-0.036795</td>\n",
"    <td>0.179010</td>\n",
"    <td>0.42138</td>\n",
"    <td>0.151820</td>\n",
"    <td>...</td>\n",
"    <td>-0.274290</td>\n",
"    <td>0.98788</td>\n",
"    <td>3.593100</td>\n",
"    <td>39.7030</td>\n",
"    <td>3.1420</td>\n",
"    <td>261.850</td>\n",
"    <td>1.3939</td>\n",
"    <td>0.51005</td>\n",
"    <td>1</td>\n",
"    <td>26640</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>26641</th>\n",
"    <td>0.014946</td>\n",
"    <td>0.94648</td>\n",
"    <td>0.03211</td>\n",
"    <td>1.03630</td>\n",
"    <td>-20.5810</td>\n",
"    <td>0.000000</td>\n",
"    <td>0.015260</td>\n",
"    <td>0.056357</td>\n",
"    <td>2.96940</td>\n",
"    <td>0.053341</td>\n",
"    <td>...</td>\n",
"    <td>0.280210</td>\n",
"    <td>0.97443</td>\n",
"    <td>1.179200</td>\n",
"    <td>15.0360</td>\n",

```

```

"      <td>4.1741</td>\n",
"      <td>108.640</td>\n",
"      <td>3.3599</td>\n",
"      <td>35.11800</td>\n",
"      <td>1</td>\n",
"      <td>26641</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"<p>26642 rows x 66 columns</p>\n",
"</div>"
],
"text/plain": [
"      X1      X2      X3      X4      X5      X6
X7  \\\n",
"0      0.088238  0.55472  0.01134  1.02050  -66.5200  0.342040  0.109490
\n",
"1      -0.006202  0.48465  0.23298  1.59980   6.1825  0.000000 -0.006202
\n",
"2      0.130240  0.22142  0.57751  3.60820  120.0400  0.187640  0.162120
\n",
"3      -0.089951  0.88700  0.26927  1.52220  -55.9920 -0.073957 -0.089951
\n",
"4      0.048179  0.55041  0.10765  1.24370  -22.9590  0.000000  0.059280
\n",
"...      ...      ...      ...      ...      ...      ...
... \n",
"26637  0.018371  0.47410 -0.13619  0.60839  -18.4490  0.018371  0.018371
\n",
"26638 -0.013359  0.58354 -0.02265  0.92896  -42.2320 -0.013359 -0.015036
\n",
"26639  0.006338  0.50276  0.43923  1.87360   9.7417  0.006338  0.012022
\n",
"26640 -0.041643  0.84810 -0.12852  0.57485 -121.9200  0.000000 -0.036795
\n",
"26641  0.014946  0.94648  0.03211  1.03630  -20.5810  0.000000  0.015260
\n",
"\n",
"      X8      X9      X10   ...      X57      X58      X59
X60  \\\n",
"0      0.577520  1.08810  0.320360   ...  0.275430  0.91905  0.002024
7.2711 \n",
"1      1.063400  1.27570  0.515350   ... -0.012035  1.00470  0.152220
6.0911 \n",
"2      3.059000  1.14150  0.677310   ...  0.192290  0.87604  0.000000

```

```

8.7934  \n",
      "3      0.127400  1.27540  0.113000  ... -0.796020  0.59074  2.878700
7.6524  \n",
      "4      0.816820  1.51500  0.449590  ...  0.107160  0.77048  0.139380
10.1180  \n",
      "...      ...      ...      ...      ...      ...      ...
...  \n",
      "26637  0.972030  1.01210  0.460840  ...  0.039866  0.98809  0.274140
73.5050  \n",
      "26638  0.562890  0.98904  0.328470  ... -0.040671  1.01110  0.805920
10.5990  \n",
      "26639  0.983560  1.00830  0.494490  ...  0.012817  0.99174  0.000000
10.4700  \n",
      "26640  0.179010  0.42138  0.151820  ... -0.274290  0.98788  3.593100
39.7030  \n",
      "26641  0.056357  2.96940  0.053341  ...  0.280210  0.97443  1.179200
15.0360  \n",
      "\n",
      "      X61      X62      X63      X64  class  score  \n",
      "0      4.7343  142.760  2.5568  3.25970      0      0  \n",
      "1      3.2749  111.140  3.2841  3.37000      0      1  \n",
      "2      2.9870   71.531  5.1027  5.61880      0      2  \n",
      "3      3.3302  147.560  2.4735  5.92990      0      3  \n",
      "4      4.0950  106.430  3.4294  3.36220      0      4  \n",
      "...      ...      ...      ...      ...      ...  \n",
      "26637  79.2370   31.268  11.6730  5.14890      1  26637  \n",
      "26638   7.1740   94.092   3.8792  1.75720      1  26638  \n",
      "26639   6.0759   51.019   7.1542  62.00100      1  26639  \n",
      "26640   3.1420  261.850   1.3939   0.51005      1  26640  \n",
      "26641   4.1741  108.640   3.3599  35.11800      1  26641  \n",
      "\n",
      "[26642 rows x 66 columns]"
    ]
  },
  "execution_count": 2,
  "metadata": {},
  "output_type": "execute_result"
}
],
"source": [
  "import pandas as pd\n",
  "import matplotlib.pyplot as plt\n",
  "from seaborn import heatmap\n",
  "import numpy as np\n",
  "import matplotlib.pyplot as plt\n",

```

```

"from scipy import optimize as op\n",
"\n",
"a = pd.read_csv('123456.CSV')\n",
"data1 = a.loc[:, ['X1' ]].values\n",
"data2 = a.loc[:, ['X2 ']].values\n",
"data3 = a.loc[:, ['X3 ']].values\n",
"data4 = a.loc[:, ['X6 ']].values\n",
"data5 = a.loc[:, ['class']].values\n",
"a\n"
]
},
{
"cell_type": "code",
"execution_count": 17,
"metadata": {},
"outputs": [
{
"data": {
"image/png": "iVBORw0KGgoAAAANSUhEUgAAABQAAAQCAQMAAABWU0U7AAAABGdBTUEAALGPC9h7kAAAAABJRU5ErkJggg=="
},
"image/svg+xml": "iVBORw0UgAAABQAAAQMAAABWU0U7AAAABGdBTUEAALGPC9h7kAAAAABJRU5ErkJggg=="
},
"text/plain": [
"<Figure size 1000x1000 with 1 Axes>"
]
},
"metadata": {
"needs_background": "light"
},
"output_type": "display_data"
},
{
"name": "stdout",
"output_type": "stream",
"text": [
"[0.60090761]\n"
]
}
],
"source": [
"Y = [i for i in range(len(data1))]\n",
"\n",
"for i in range(len(data1)):\n",
"    data1[i]=(data1[i]-np.min(data1))/(np.max(data1)-np.min(data1))\n",
"    data2[i]=(data2[i]-np.min(data2))/(np.max(data2)-np.min(data2))\n",
"    data3[i]=(data3[i]-np.min(data3))/(np.max(data3)-np.min(data3))\n",
"    data4[i]=(data4[i]-np.min(data4))/(np.max(data4)-np.min(data4))\n",
"\n",
]

```

```

        "\n",
        "for i in range(len(data1)):\n",
        "    Y[i] = data1[i]*0.33 + data2[i]*0.32 + data3[i]*0.34 +data4[i]*0.02\n",
        "\n",
        "type(Y)\n",
        "# 计算P值\n",
        "\n",
        "plt.figure(figsize=(10, 10), dpi=100)\n",
        "plt.scatter(data5,Y)\n",
        "plt.show()\n",
        "\n",
        "num=1\n",
        "for i in range(len(data1)):\n",
        "    if data5[i]==0 and Y[i]<num:\n",
        "        num=Y[i]\n",
        "\n",
        "print(num) \n"
    ]
},
{
    "cell_type": "code",
    "execution_count": 51,
    "metadata": {},
    "outputs": [],
    "source": [
        "ngData=pd.read_csv('123456.csv')\n",
        "\n",
        "ngList=[]\n",
        "ngData['score']=Y\n",
        "ngData.to_csv('123456.csv',index=False)\n"
    ]
}
],
"metadata": {
    "interpreter": {
        "hash": "4442a059423b1d8cb2d566f1d7a9e596fd1852f4b9b9e9d6f69b72f942b31330"
    },
    "kernelspec": {
        "display_name": "Python 3.9.7 ('base')",
        "language": "python",
        "name": "python3"
    },
    "language_info": {
        "codemirror_mode": {
            "name": "ipython",

```



```

        "version": 3
    },
    "file_extension": ".py",
    "mimetype": "text/x-python",
    "name": "python",
    "nbconvert_exporter": "python",
    "pygments_lexer": "ipython3",
    "version": "3.9.7"
},
"orig_nbformat": 4
},
"nbformat": 4,
"nbformat_minor": 2
}

```

程序二：Matlab 基于神经网络的机器学习训练模型

```

%%%%%38,öÖ,±êµÄÑµÁ · %^-
%%%%%Êä³ö1,öÖ,±êµÄÑµÁ · %^-

x=A';
y=B';

testx ; %%%Ô²²âÊ±ðèÒªµÄÊäÈëÖµ

n=38 %%%×Ô±äÁ¿µÄ,öÊý
m=1 %%%Òò±äÁ¿µÄ,öÊý

p=x; %½«ËúÓÐ×Ô±äÁ¿°Ë²¿µÄµÊäÈëÊý%Ý%00ó
t=y; %½«ËúÓÐÒò±äÁ¿°Ë²¿µÄ±êÊý%Ý%00ó
[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t); % ¶ÔÓÚÊäÈë%00ó°ËÊä³ö%00ó%t%00ó¹éÒ»»-´!Àí
u=ones(n,1);
dx=[-1*u,1*u]; %¹éÒ»»-´!Àí°ó×íð;ÖµÎª-1£¬×í´óÖµÎª1
%BPÍØÄÇÑµÁ ·
net=newff(dx,[n,15,m],{'tansig','tansig','purelin'},'traingdx'); %½«ÄÇÄÊðÍ£¬²¿ÓÀÏÝ¶ÈÎÀµ ·

net.trainParam.show=10000; %1000ÄÖ»ðÏÊ%Ò»´Î%á¹û
net.trainParam.Lr=0.05; %Ñ§Ï°Ëú¶ÈÎª0.05
net.trainParam.epochs=50000; %×í´óÑµÁ · ÄÖ»ðÎª50000´Î
net.trainParam.goal=0.78*10^(-3); %ù · %Ïó²í
net=train(net,pn,tn); %¿ÊÑµÁ · £¬ÄäÖðpn,tn · Ò±ðÏÊäÈëÊä³öÑù±%
%ÀúÓÀÏÊý%Ý%00óBPÍØÄÇ · ÄÖæ
an=sim(net,pn); %ÓÀÑµÁ · °ÄµÄÄÊðÍ%00ó · ÄÖæ

```

```

a=postmnmx(an,mint,maxt); % °Ñ · ÂÕæµÃµµµÃÊý%Ý»¹ÔÎ°ÔÊµÃÊý

x=1:length(y);
yucey1=a(1,:); %%% Ö,±êAµÃÔ²²âÖµ
figure ;
subplot(1,1,1);plot(x,y(1,:), 'r-o',x,yucey1, 'b--+') %»æÖÀòòÖ,±êAµÃ¶Ô±ÊÍ%£»
legend('ÒøÐÐ×Ô°ÏÖµµÃÔ²²âÖµ','ÒøÐÐ×Ô°ÏÖµÊµ%ÊÖµ');
xlabel('ÒøÐÐ');ylabel('ÒøÐÐ×Ô°ÏÖµ');

% µ±ÓÃÑµÁ · °ÃµÃÍøÂç¶¶ÔÐÂÊý%Ýpnew%øÐÐÔ²²âÊ±£¬Ò²Ó!×÷ÏàÓ!µÃ'!Àí£°

pnew=testx; %ÊäÊë×Ô±äÁ;µÃ²ÎÊý£¬¬Á;Ò»ÐÐ±íÊ%Ò»,ö×Ô±äÁ;£¬¬ÁÐÊý±íÊ%Ô²²âµÃ,öÊý

pnewn=tramnmx(pnew,minp,maxp); %ÀûÓÃÔÊ%ÊäÊëÊý%ÝµÃ¹éÒ»»²ÎÊý¶¶ÔÐÂÊý%Ý%øÐÐ¹éÒ»»²£»
anewn=sim(net,pnewn); %ÀûÓÃ¹éÒ»»²°óµÃÊý%Ý%øÐÐ · ÂÕæ£»
anew=postmnmx(anewn,mint,maxt) %°Ñ · ÂÕæµÃµµµµÃÊý%Ý»¹ÔÎ°ÔÊµµÃÊýÁ;%¶£»
Y1=anew(1,:); %%%Òò±äÁ;AµÃÔ²²âÖµ
% ÏÔÊ%Ô²²âÖµ
disp(' Ô²²âÖµ£° ')

```