

一、引言

电池故障诊断是电池管理系统中一个重要的功能，其旨在通过实时采集电池系统中的电流、电压和SOC等信息，分析和判断电池中是否可能存在某种故障，并及时地提醒管理人员。本实验针对所给的数据集，从多个角度建立电池故障诊断模型，已实现通过input特征变量到output的正确分类。本实验数据集的输入特征变量有13维，分别代表SOC、电压、电流特征等，带判断的电池故障有Y1：容量偏小故障、Y2：内阻偏大故障、Y3：SOC偏小故障。

本实验共计进行了七次计算，采用了五种常用的机器学习分类模型，分别是反向传播神经网络、逻辑回归、K近邻、支持向量机、分类树，其中在反向神经网络上分别进行了整体训练、降维训练和单分类训练。通过上述实验，得到了每个模型在测试集上的分类结果，并通过混淆矩阵、Accuracy、Precision、Recall、F1-score评价指标对模型分类性能做充分分析。

本文第二章重点描述了本文数据集预处理和特征工程工作，这一部分是数据建模的重要先行工作，因为原始数据集中存在无关列等干扰信息，会无意义地增加模型训练时间。同时为了得到更好的分类效果，在特征工程中我们观测了每列特征变量下样本数据的范围，并对所有数据进行归一化处理。针对可能的数据量庞大造成的训练时间过长情况，我们额外地做了一次相关性分析和主成分分析，以分析特征变量和分类标签之间的相关性，并初步对分类逻辑进行探究。

本文第三章简要介绍了本文所用的模型的数理原理，包括反向传递神经网络、逻辑回归、K近邻分类、支持向量机分类、分类决策树等。本章中重点介绍了各模型的核心公式，并介绍了其中影响到分类性能的关键系数。

本文第四章为本文的核心。首先对我们选取的评价指标做简要介绍，接下来介绍了各个模型的训练过程，重点介绍了参数选取方式、测试集带入后的分类结果、混淆矩阵、评价指标和ROC图像。特别地、本文对故障诊断性能最好的反向传递神经网络训练过程做了重点描述，包括各个不同条件下模型的建立与训练、分类结果与模型评价、基于GridSearchCV方法的模型参数优化等。本文选取的模型中最优的神经网络模型分类结果（Accuracy）为：Y1：0.9695、Y2：0.9310、Y3：0.9675。说明模型分类准确性良好，且具有一定的泛化能力。

本文第五章为模型的综合评价与对比，得出结果反向神经网络是训练时间较短，建模性价比最高的模型。同时对该实验做了总结，提出了我对该工作的进一步展望。

二、数据集预处理与特征工程

2.1 数据描述性统计

本实验采用的数据集为**battery_fault_diagnosis**，通过python进行描述性统计，得到如表1和表2所示的统计信息。**battery_fault_diagnosis**包含10000条样本数据，每条样本包括两部分信息，分别是13列特征变量V: *input*和3列分类标签Y: *output*。根据描述性

统计信息，显然可以发现，有四条特征变量（V10，V11，V12，V13）下数据完全一致，因此此四条变量对分类无影响，故在建立数据集时忽略此四条变量。对于分类标签，我们关系的是样本分布是否均匀，以及三条分类之间有无统计上的相关性。观察`output`的分类统计和组合统计，单条分类可以近似得到1:1的分类比，（Y1，Y2，Y3）的组合分类样本数近似等于1250，我们暂时认为三条分类标签之间没有统计上的相关性，该结论有待进一步说明。

本文中认为分类Y=0为无该故障（正例），认为分类Y=1为有该故障（反例）。

表 1: `battery_fault_diagnosis`特征变量input描述性统计

特征变量	V1	V2	V3	V4	V5	V6	V7
样本量	10000	10000	10000	10000	10000	10000	10000
平均值	0.576989	0.005398	0.000499	0.000249	0.000184	0.310626	0.239849
标准差	0.214925	0.000335	0.000031	0.000015	0.000011	0.178094	0.157522
最小值	0.2	0.00416	0.000384	0.000192	0.000142	0.171858	0.121024
25%分度值	0.39	0.00517	0.000478	0.000238	0.000176	0.240449	0.178752
50%分度值	0.58	0.005395	0.000499	0.000248	0.000184	0.261256	0.195654
75%分度值	0.76	0.005619	0.000519	0.000259	0.000191	0.288961	0.218115
最大值	0.95	0.006661	0.000616	0.000307	0.000227	1.637694	1.403971
特征变量	V8	V9	V10	V11	V12	V13	
样本量	10000	10000	10000	10000	10000	10000	
平均值	0.189574	0.210951	3.29	2.59	2.05	2.21	
标准差	0.123417	0.151695	0	0	0	0	
最小值	0.09763	0.104362	3.29	2.59	2.05	2.21	
25%分度值	0.140949	0.152476	3.29	2.59	2.05	2.21	
50%分度值	0.153865	0.166563	3.29	2.59	2.05	2.21	
75%分度值	0.172012	0.186098	3.29	2.59	2.05	2.21	
最大值	1.099346	1.31977	3.29	2.59	2.05	2.21	

表 2: `battery_fault_diagnosis`分类标签output描述性统计

分类标签	Y1	Y2	Y3	
样本量	10000	10000	1000	
分类1样本数	5038	5080	5005	
分类0样本数	4962	4920	4995	
组合(Y1,Y2,Y3)	(0,0,0)	(1,0,0)	(0,1,0)	(1,1,0)
组合样本数	1234	1211	1275	1275
组合(Y1,Y2,Y3)	(0,0,1)	(1,0,1)	(0,1,1)	(1,1,1)
组合样本数	1216	1259	1237	1293

对特征变量进行离群值检验，并观察其数据分布，发现其分布较为均匀，没有特别的离群值。同时由于该数据集分类二值均匀性良好，所以不需要对样本容量做处理。观察特征变量（V1 ~V9）的数量级，能发现部分变量数量级之间相差较大，所以要对各变量做归一化处理。本实验选取的归一化方法为最大最小值归一化，将每一条样本变量值映射到[0,1]之间，原像的最小值映射为0，原像的最大值映射为1，其他值依照原像的大小分布均匀映射。该归一化方法为式2，其中 x^* 为归一化的值， x 为原值。

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

2.2 相关性分析与主成分分析

为了摸清特征变量和分类标签之间的关系，同时希望能进一步降低数据集的维度，所以本实验在建立数据集之前，进行相关性和主成分的特征工程。

首先分析特征变量和分类标签之间的相关性。本实验采用python的corr()函数绘制特征变量V1~V9与分类标签Y1~Y3之间的关联矩阵。通过热力图可以看到，颜色越接近蓝色则正相关性就越强，越接近红色则负相关性越强。关联矩阵相关度默认采用Pearson相关系数计算：

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)}\sqrt{E(Y^2) - E^2(Y)}} \quad (2)$$

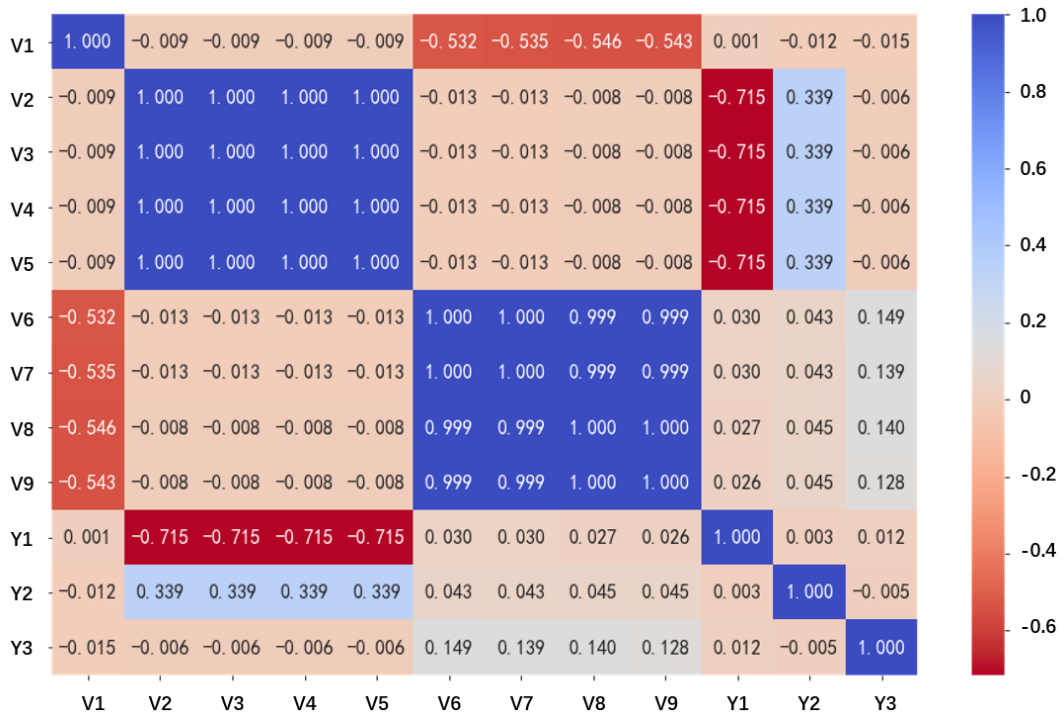


图 1: 数据集关联矩阵（去无关变量）

从中可以看出，相当数量的变量之间存在高度相关性（怀疑是存在线性相关性的一次仿真实验的结果变量），在分类方面，V2、V3、V4、V5与分类Y1表现高度负相关性，与分类Y2表现明显的正相关性，V6，V7，V8，V9与分类Y3表现出一定正相关性。同时分类Y1、Y2、Y3之间并不存在明显相关性，这也验证了前面的想法。

由于变量之间存在着高度相关性，说明其有可能呈线性相关性，这样就可以通过主成分分析（PCA）的方法对数据集降维，以减少训练时间。主成分分析实质上是对特征变量空间的线性变换，其算法大致有四步：1.每一列特征变量作为一个空间维度分量，将每一列特征变量值做均值中心化处理；2.以各列特征变量的均值点为空间的原点，将空间基底以原点为中心旋转，基底线性变换的同时坐标也随之变化；3.找到以相关系数矩阵为线性变换阵的基底，使得坐标分量中尽可能多的分量值最小；4.求出该相关系数矩阵及变换后的特征变量样本序列，去除数量级极小的特征变量。

本实验通过MATLAB实现PCA过程，将数据集特征变量从9为降至6维。得到的变换矩阵 tranM 和降维后的数据集 dataset 的特征变量统计性描述如表3所示。

$$\text{tranM} = \begin{bmatrix} 0.7911 & -0.0291 & 0.6109 & 0.0062 & -0.0010 & 0.0019 & 0.0000 & 0.0000 & 0.0000 \\ 0.0083 & 0.4997 & 0.0130 & -0.0010 & 0.0002 & 0.0002 & -0.1726 & -0.3525 & 0.7719 \\ 0.0083 & 0.4997 & 0.0130 & -0.0010 & 0.0002 & 0.0002 & -0.2234 & -0.5541 & -0.6269 \\ 0.0083 & 0.4997 & 0.0130 & -0.0010 & 0.0002 & 0.0002 & -0.4507 & 0.7340 & -0.0894 \\ 0.0083 & 0.4997 & 0.0130 & -0.0010 & 0.0002 & 0.0002 & 0.8468 & 0.1726 & -0.0556 \\ -0.3002 & -0.0061 & 0.3951 & -0.5779 & -0.4518 & -0.4642 & 0.0000 & 0.0000 & 0.0000 \\ -0.3041 & -0.0064 & 0.3974 & -0.4091 & 0.6282 & 0.4329 & 0.0000 & 0.0000 & 0.0000 \\ -0.3076 & -0.0041 & 0.3913 & 0.3898 & -0.5312 & 0.5639 & 0.0000 & 0.0000 & 0.0000 \\ -0.3108 & -0.0038 & 0.3985 & 0.5886 & 0.3447 & -0.5283 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \quad (3)$$

表 3: dataset 特征变量input描述性统计(训练集和测试集)

特征变量	V_PCA1	V_PCA2	V_PCA3	V_PCA4	V_PCA5	V_PCA6
样本量	8000	8000	8000	8000	8000	8000
平均值	0	0	0	0	0	0
标准差	0.33570	0.26714	0.17988	0.00475	0.00292	0.00060
最小值	-1.50361	-0.98618	-0.30838	-0.04077	-0.01083	-0.00252
25%分度值	-0.16152	-0.18199	-0.13556	-0.00302	-0.00203	-0.00045
50%分度值	0.05199	-0.00439	-0.02664	0.00033	0.00019	-0.00004
75%分度值	0.25014	0.17550	0.11574	0.00313	0.00219	0.00048
最大值	0.48483	1.01241	1.13421	0.01806	0.00755	0.00182

这里只展示对部分数据的降维后统计，因为为了保证验证的效果，避免过拟合，

验证集要单独做处理，不能一起做降维。

2.3 数据集建立

原始数据集经过特征工程之后，就可以做数据集切分了，切分规则是序号为1-6000的数据为训练集，序号6001-8000的数据为验证集，序号为8001-10000的数据为测试集。训练集为模型训练的数据，其直接影响到模型的分类性能；验证集为训练过程中对模型性能验证和优化所用；测试集与模型本身无关，为测试模型分类结果所用。针对模型训练的需要，本实验建立如下数据集。

表 4: *dataset*特征变量input描述性统计(训练集和测试集)

数据集	样本数量	说明
<i>train_x</i>	6000	训练集变量
<i>train_y</i>	6000	训练集分类
<i>vali_x</i>	2000	验证集变量
<i>vali_y</i>	2000	验证集分类
<i>test_x</i>	2000	测试集变量
<i>test_y</i>	2000	测试集变量
<i>train_pcax</i>	6000	降维训练集变量
<i>train_pcal</i>	6000	降维训练集分类
<i>vali_pcax</i>	2000	降维验证集变量
<i>vali_pcal</i>	2000	降维验证集分类
<i>test_pcax</i>	2000	降维测试集变量
<i>test_pcal</i>	2000	降维测试集分类

由于降维过程势必会造成一部分特征信息丢失，所以这里我们设计了两组实验以确定降维是否有效。第一组的训练集-验证集-测试集为原始数据集*battery_fault_diagnosis*删除无关列并经过归一化之后的数据。第二组训练集-验证集-测试集在第一组的基础之上，增加了PCA的一步。对训练集和验证集进行PCA降维，保留前六列数据，测试集特征变量矩阵单独进行数据中心化并与相关矩阵相乘得到PCA后的验证集。

$$test_pcax = test_x \times tranM \quad (4)$$

基于上述工作，实验数据集建立完毕。

三、机器学习分类模型

本章将针对前一章节构建的数据集，训练基于多种分类模型的电池故障诊断模型。本实验采用MATLAB的神经网络工具箱和分类工具箱，编写相关程序进行模型训练和分类验证。本实验采用的模型包括：

- BP神经网络
- 逻辑回归
- K近邻
- 支持向量机
- 分类决策树

本章将简要介绍各分类模型的建模、预测和分类性能评价。

3.1 反向传递神经网络

典型的神经网络模型有三层结构，输入层、输出层和隐藏层。在本实验中为了提高分类准确度，我们设置了两层隐藏层，输入层为训练集的特征变量和偏置节点，输出层为分类结果，隐藏层我们每层设置20个节点。神经网络模拟了人脑中神经元的信息传递过程，信息 X 从输入层经过一次连接矩阵 W_1 和偏置系数 b_1 传递到隐藏层中，再通过连接矩阵 W_2 ， W_3 和偏置系数 b_2 ， b_3 将信息传递至输出层 Y ，最终实现数据的分类。该神经网络模型示意如图2所示。神经网络的三次信息传递过程为：

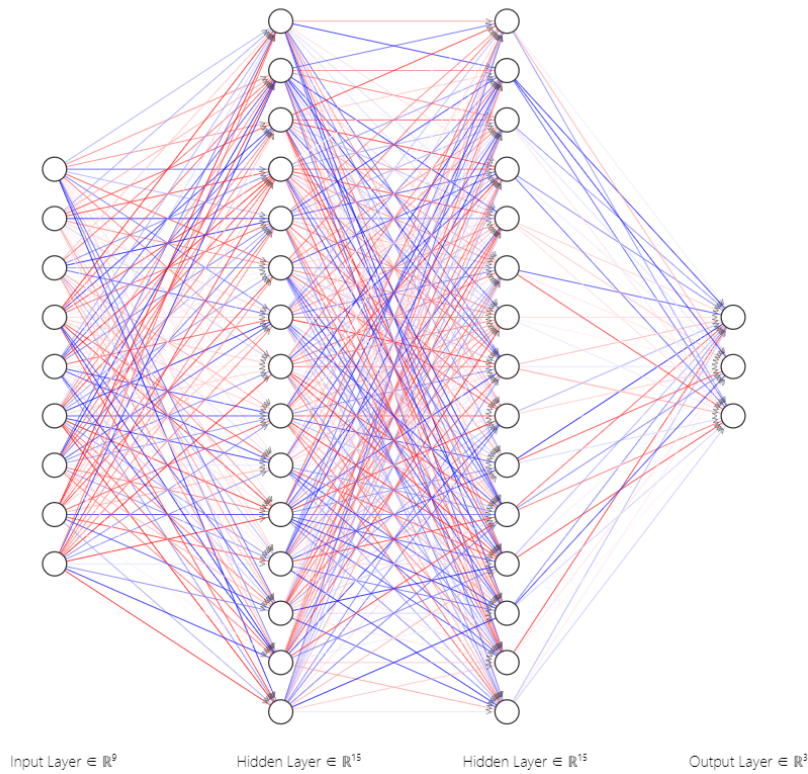


图 2: 神经网络传递模型

$$\begin{aligned}
 H_1 &= X \times W_1 + b_1 \\
 H_2 &= H_1 \times W_2 + b_2 \\
 Y &= H_2 \times W_3 + b_3
 \end{aligned} \tag{5}$$

通过上述的线性方程就可以得到输出 Y 的值，本模型采用双曲正切tan-sigmoid函数作为激活层，其作用是接受向量的值，并将其中的元素压缩到 $[-1,1]$ 内。当输入向量为 N 时，tan-sigmoid函数的输出关系式为：

$$Y = \frac{2}{1 + e^{-2N}} - 1 \quad (6)$$

本实验采用trainlm方法进行反向传播过程，trainlm为Levenberg-Marquardt算法，对于中小型规模的BP神经网络有最快的收敛速度，也是系统默认算法。其是一种利用梯度求最值的算法，形象的说，属于“爬山”法的一种。它同时具有梯度法和牛顿法的优点。该神经网络通过LM实现模型最优化过程。

3.2 逻辑回归

逻辑回归（Logistic Regression）是一种针对离散变量进行预测的模型，可用于金融电信诈骗分类问题中。经典逻辑回归模型的框架如图4所示。逻辑回归的核心是Sigmoid函数，其是以输入值 x 加以权重值，并通过其线性组合来预测输出值 \hat{y} 的传播函数：

$$\hat{y} = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots)}} \quad (7)$$

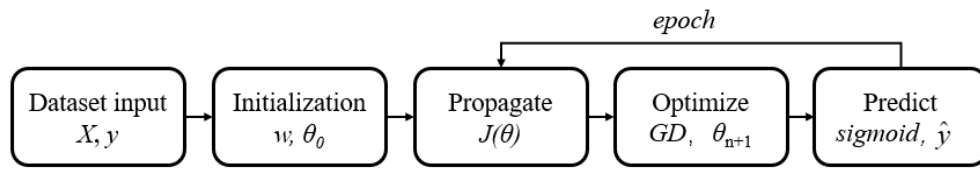


图 3: 逻辑回归模型框架

θ 系数的迭代更新要靠梯度下降获得：

$$\theta_{n+1} = \theta_n + \sigma \times (y - \hat{y}) \times \hat{y}(1 - \hat{y}) \times x \quad (8)$$

在传播和优化过程中采用“损失函数”以衡量模型在拟合训练集时产生的信息损失大小，其定义为：

$$J(\theta) = - \sum_{i=1}^m [y_i \times \log(y_{\theta}(x_i)) + (1 - y_i) \times \log(1 - y_{\theta}(x_i))] \quad (9)$$

同时为了防止过拟合，我们在损失函数中添加了正则化项，以调节模型的拟合程度，真对不同范式类型的罚函数，有两种正则化项定义方法：

$$\begin{aligned} J(\theta)_{L1} &= C \times J(\theta) + \sum_{j=1}^n |\theta_j| \\ J(\theta)_{L2} &= C \times J(\theta) + \sqrt{\sum_{j=1}^n (\theta_j)^2} \end{aligned} \quad (10)$$

3.3 K近邻分类

K近邻（K-Nearest Neighbour）算法核心是从训练集中选取出 k 个与新样本相似度最高的样本，并以此 k 个近邻的类别来确定新样本的类别，其中 k 的大小可以自由选取。通常K近邻算法通过欧氏距离来衡量样本之间的相似度：

$$d = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \quad (11)$$

对于高维样本，两点 $(x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)})$ 和 $(x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)})$ 之间的欧几里得距离为：

$$d = \sqrt{\sum_{i=1}^n (x_i^{(1)} - x_i^{(2)})^2} \quad (12)$$

若训练集共计 m 个样本，我们得到每一个训练样本与新样本的距离序列 $d_i (i = 1, 2, \dots, m)$ 。我们认为两个样本的距离最小时，它们最相似。因此我们从序列 d_i 中选取最小的 k 样本，设其类别为 y_1, y_2, \dots, y_k ，这一列分类结果的众数 l 就是K近邻算法对新样本的分类。K近邻模型的具体实现过程有很多，这里我们最终选取了自动（auto）方法。K近邻的模型框架如图4所示：

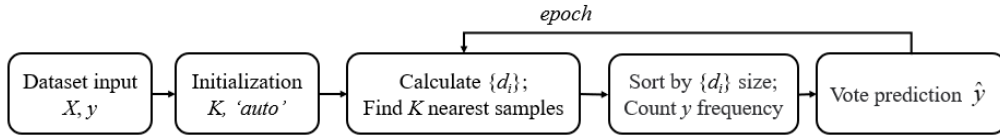


图 4: K近邻模型框架

3.4 支持向量机分类

支持向量机分类（Support Vector Classifion, SVC）的目标是找到一个超平面 $\omega X + b = 0$ ，以此超平面为界，可以将数据集分为两类，分别记为分类标签的1和0。因此，我们需要先找到各个分类的样本点离这个超平面最近的点，使得这个点到超平面的距离最大化。

在数据集是线性可分的情况下，数据集应分布于超平面的两侧。那么使得支持向量到超平面的距离最大化，可以使用如下等价的两个优化式表示：

$$\begin{aligned} \max_{\omega, b} \frac{2}{\|\omega\|} \quad & \Leftrightarrow \quad \min_{\omega, b} \frac{\|\omega\|^2}{2} \\ \text{s.t. } y_i(\omega^T X_i + b) & \geq 1, \quad \text{s.t. } y_i(\omega^T X_i + b) \geq 1 \end{aligned} \quad (13)$$

其中， $\omega = \sum_{i=1}^n \alpha_i t_i x_i$ ， x_i 表示数据集特征， t_i 为分类变量。有损失函数：

$$\min_{\alpha} \quad J = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j (x_i^T x_j) - \sum_{i=1}^N \alpha_i \quad (14)$$

通常会用核函数以简化上面损失函数中 $x_i^T x_j$ 内积计算过程，本处选择高斯核函数替换掉原内积项：

$$\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\delta^2}\right) \quad (15)$$

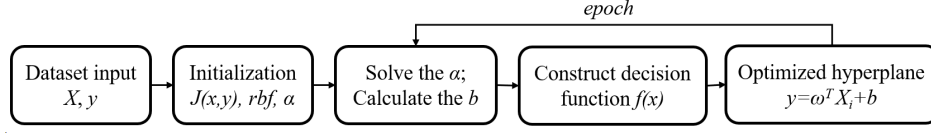


图 5: 支持向量机分类模型框架

3.5 分类决策树

分类决策树（Classification Trees）的表示形式为一颗二叉树。每个节点表示单个输入变量 X 和该变量的分割点（假定变量是数字化的）。树的叶节点包含用于预测的输出变量 y 。决策树的损失函数函数也是决策树的拆分依据，常用的损失函数为 Gini 函数或信息熵（entropy），以此选择成本最高的拆分。Gini系数的定义为：

$$\text{gini}(x) = 1 - \sum_{i=1}^N p_i^2(x) \quad (16)$$

其中 k 为数据集中样本分类的数量， p_i 表示第 i 类样本占总样本 T 的比例。信息熵定义为：

$$E(x) = - \sum_{i=1}^N p(i|x) \log_2 p(i|x) \quad (17)$$

分类决策树的构建过程可以概括损失函数的计算和最佳分割点的递归切分。模型流程框架为：

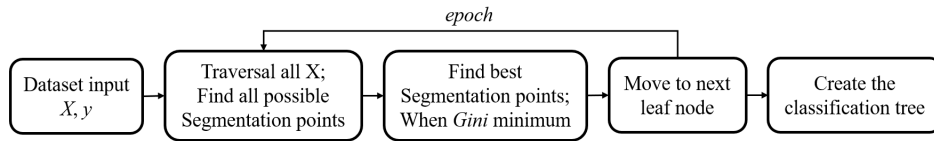


图 6: 分类决策树模型框架

四、模型训练与评估

4.1 分类评价指标

在分类模型中最常用的评价指标是准确率 Accuracy，指正确预测样本占全部预测的比重。但根据实际需求，我们希望设计的故障诊断模型能够尽可能的找到全部故障样本，并且尽可能避免错分类的情况。所以在数据建模中，经常用于判断分类性能的方法为混淆矩阵，混淆矩阵的四个元素为：

- TP (True Positive): 正确的正例, 一个实例是正类并且也被判定成正类;
- FN (False Negative): 错误的反例, 漏报, 本为正类但判定为假类;
- FP (False Positive): 错误的正例, 误报, 本为假类但判定为正类;
- TN (True Negative): 正确的反例, 一个实例是假类并且也被判定成假类。

$$\text{ConfM} = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \quad (18)$$

根据混淆矩阵的定义, 在数据建模的过程中, 经常用到判断分类模型性能的指标有:

- 准确率 (Accuracy) 为预测正确的样本占总的比重, 其定义为:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

- 精确率 (Precision) 为正确预测为正的样本占有所有预测为正的样本的比例, 其定义为:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (20)$$

- 召回率 (Recall) 为正确预测为正占全部实际为正的样本的比例, 其定义为:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (21)$$

- F1-Score, 权衡查准率和召回率的综合指标, 其定义为:

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (22)$$

模型评价可以通过ROC图实现可视化。ROC曲线为假阳性率 (FPR, 负样本中的错判率) 和真阳性率 (TPR, 判对样本中的正样本率) 之间的关系。横坐标为假阳性率, 纵坐标为真阳性率。

$$\text{FPR} = \frac{FP}{FP + TN}, \quad \text{TPR} = \frac{TP}{TP + FN} \quad (23)$$

ROC图中, 曲线越靠近图像的左上方说明该模型的分类性能越好。ROC图中曲线与x轴围城的面积称为AUC, 其值也是越大越好。ROC图像的变量为分类的阈值。

4.2 反向神经网络模型训练

本实验首先构建一包含两隐藏层的反向神经网络模型, 并将训练集带入BP神经网络做初步模型训练。模型各参数值设定如下

表 5: BP神经网络参数

	输入层节点	输出层节点	隐藏层节点	激活函数	最优化方法
模型参数	9	3	[20,20]	tansig	trainlm
	训练次数	学习速率	最小误差	分类置信度	
模型参数	10000	0.01	0.0001	0.5	

该神经网络模型经过训练后，认为分类阈值 $q = 0.5$ ，即输出 $Y \geq 0.5$ 认为分类为1，输出 $Y < 0.5$ 认为分类为0，得到如下的分类混淆矩阵：

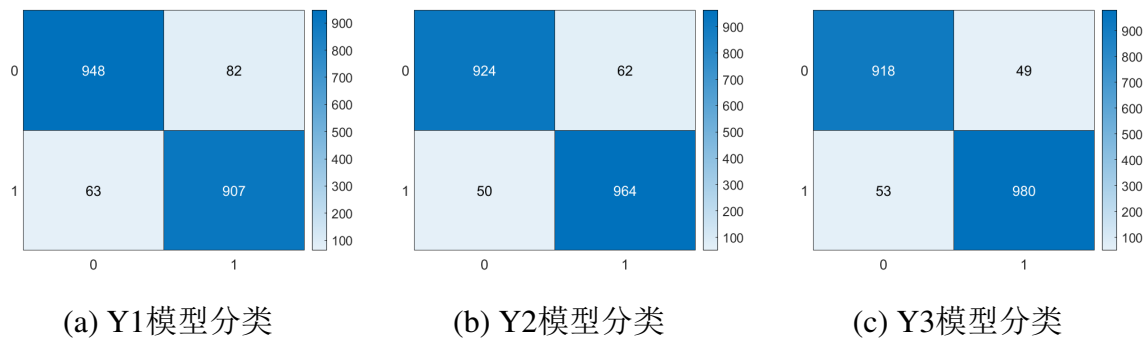


图 7: BP神经网络分类结果

综合三组分类结果，可以得到如下的分类性能评价：

表 6: BP神经网络参数

分类Y1	Accuracy	Precision	Recall	F1-score
评价得分	0.9255	0.9377	0.9204	0.9290
分类Y2	Accuracy	Precision	Recall	F1-score
模型参数	0.9440	0.9487	0.9371	0.9429
分类Y3	Accuracy	Precision	Recall	F1-score
模型参数	0.9490	0.9454	0.9493	0.9474
总体分类	总体正确	总体误判	Accuracy	
模型参数	1658	342	0.8290	

另分类置信度阈值 q 在0到1之间递进，逐步做分类评价，绘制出BP神经网络的ROC曲线并计算AUC值，得到如下的结果：

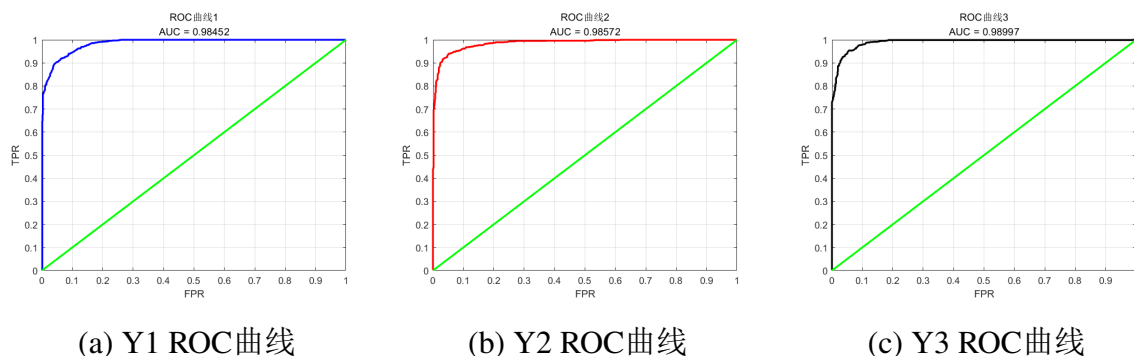


图 8: BP神经网络分类结果-ROC曲线

初步计算得到模型的分类性能较为良好。接下来我们思考，现在的模型是一次性输出全部的特征变量，同时得到所有的分类结果。如果采用三个模型，对三个故障分

别做分类诊断，会不会性能更好一些。接下来我们完全复用上述的方法，重新建立三个模型。下面是单项故障诊断模型的分类结果：

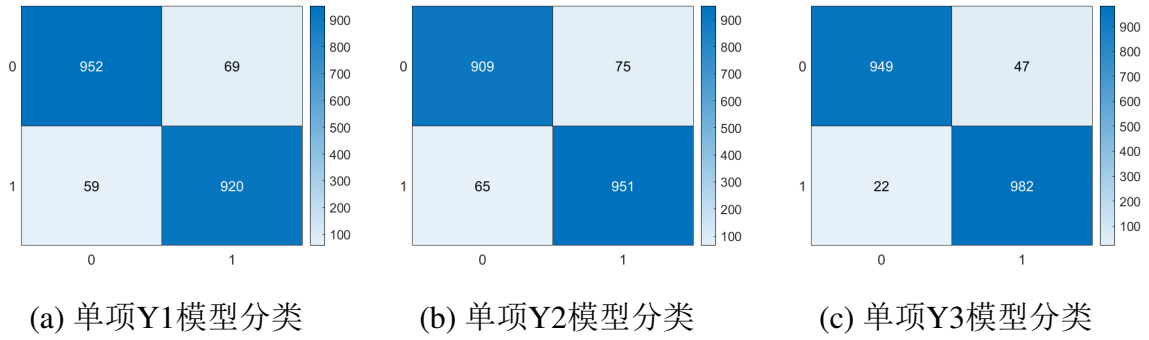


图 9: BP神经网络单项分类结果

综合三组分类结果，可以得到如下的分类性能评价：

表 7: BP神经网络参数-单项故障诊断

分类Y1	Accuracy	Precision	Recall	F1-score
评价得分	0.9515(+0.02)	0.9575(+0.025)	0.9472(+0.025)	0.9523(+0.025)
分类Y2	Accuracy	Precision	Recall	F1-score
模型参数	0.9345(-0.005)	0.9405(-0.01)	0.9262(-0.01)	0.9333(-0.01)
分类Y3	Accuracy	Precision	Recall	F1-score
模型参数	0.96705(+0.015)	0.9701(+0.02)	0.9622(+0.015)	0.9662(+0.02)

绘制三次单项分类的ROC曲线：

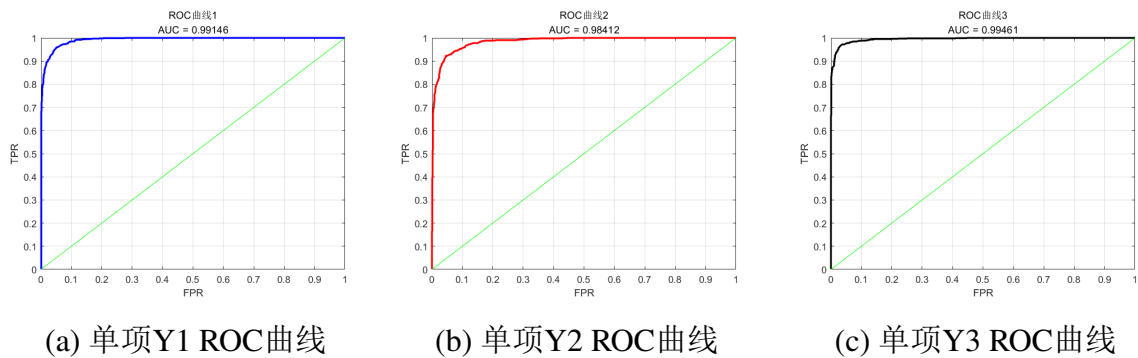


图 10: BP神经网络分类结果-单项故障诊断

简单从分类指标上可以看出，对于单项故障，通过单项分类模型得到的Y1，Y3分类结果要略高与同时分类的模型，但是我们若综合判断三个分类模型分类结果，计算整体的分类性能，发现其准确度Accuracy要远低于同时分类的模型（总体分类正确862，总体分类错误1138）。这是因为训练单一故障诊断时，模型没有对其他故障类型进行

学习，同时由于前文所示的，三个故障之间的相关性很低，而Y1和Y2之间仅有依赖的部分特征变量存在明显的负相关性，所以该方法天然地对整体故障诊断存在偏差。

这一步工作并非无意义，其进一步揭示了特征变量与分类标签之间的相关性，同时整体分类准确度为0.4310并非完全随机分布的0.125，也说明了数据集变量相关性对分类准确度的意义。同时并非所有模型都可以同时做多故障诊断，像是下一节采用的逻辑回归、支持向量机等模型就不适合一次性完成多个分类，所以在下节过程中我们主要采用这种方式进行故障诊断。

接下来基于主成分分析降维后的数据集（train_pcax,train_pcy）进行同表5中所示的模型训练，得到新的分类模型，用降维后的验证集（vali_pcax,vali_pcy）进行模型评价得到如下的分类结果：

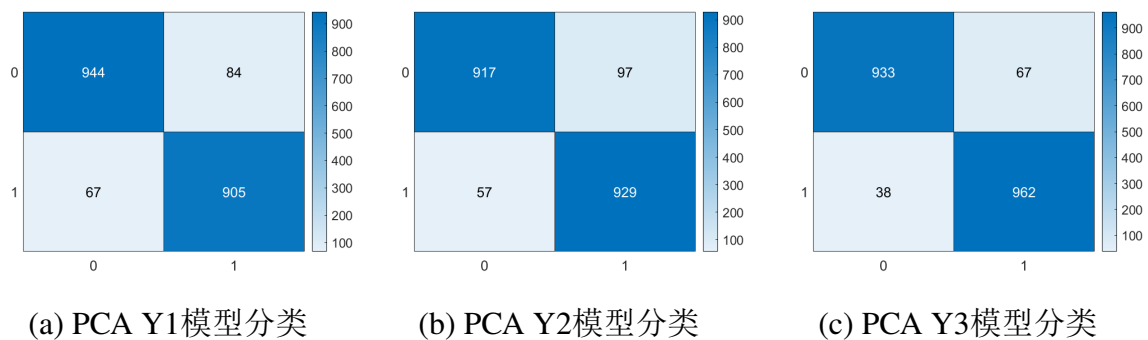


图 11: BP神经网络分类结果-PCA

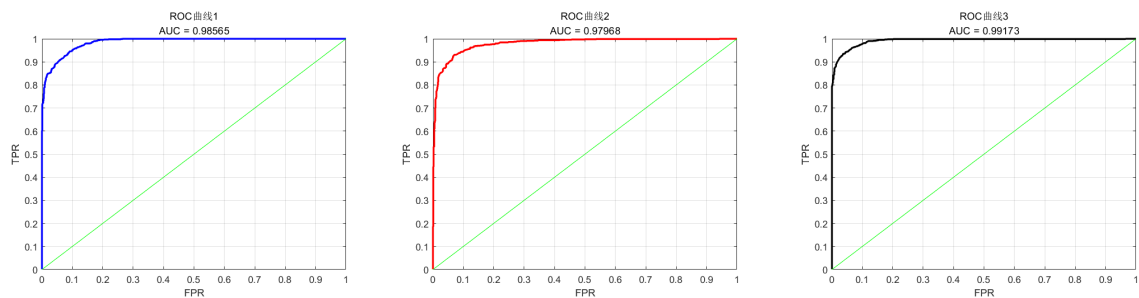
综合三组分类结果，可以得到如下的分类性能评价：

表 8: BP神经网络参数

分类Y1	Accuracy	Precision	Recall	F1-score
评价得分	0.9245	0.9337	0.9183	0.9259
分类Y2	Accuracy	Precision	Recall	F1-score
模型参数	0.9230	0.9415	0.9043	0.9225
分类Y3	Accuracy	Precision	Recall	F1-score
模型参数	0.9475	0.9609	0.9330	0.9467
总体分类	总体正确	总体误判	Accuracy	
模型参数	1610	390	0.8050	

绘制三次单项分类的ROC曲线：

用主成分分析降维的方法构建的数据集再分类性能上没有明显的下降，但是运算的时间和迭代次数缩短了很多（由500余次降低到了80余次）。通过主成分分析降维的方法可以在大批量数据时提高模型训练时间，但是降维的过程必然会删去部分特征变量的信息。



(a) Y1 (PCA) ROC曲线 (b) Y2 (PCA) ROC曲线 (c) Y3 (PCA) ROC曲线

图 12: BP神经网络分类结果-PCA

4.3 GridSearchCV模型参数优化

网格搜索（GridSearch）是一种步进遍历最优参数的优化方法，该方法将所有的待优化参数设定相应的搜索空间，并确定好步长，参数的分布如同网格一样。交叉验证（CrossValidation）指将训练集和验证集随机分成K折，在模型训练的过程中，依次将其中一折作为验证集，将剩下K-1折作为训练集进行训练，总共训练K次模型，最后整合训练的结果。网格搜索的目的是提高分类的准确率，找到最适应的分类模型，交叉验证的目的是防止模型过拟合，提高模型泛化能力。本节将采用GridSearchCV方法对神经网络模型进行模型优化，以找到分类效果最优的分类模型。本节采用的测试集为 $test_x$ 和 $test_y$ 。

本节设置了如下的BP神经网络优化参数：

表 9: BP神经网络参数优化范围

模型参数	交叉验证折数	隐藏层节点	激活函数	最优化方法
	5	[20,20],[30,30]	tansig, ReLU	trainlm, traingd, traingdm

模型经过优化后，选择出应用在测试集上分类性能最好的模型。筛选出的最优参数组合为：

表 10: BP神经网络最优参数

模型参数	交叉验证折数	隐藏层节点	激活函数	最优化方法
	5	[20,20]	ReLU	trainlm

将测试集带入模型得到该组合的分类性能为：

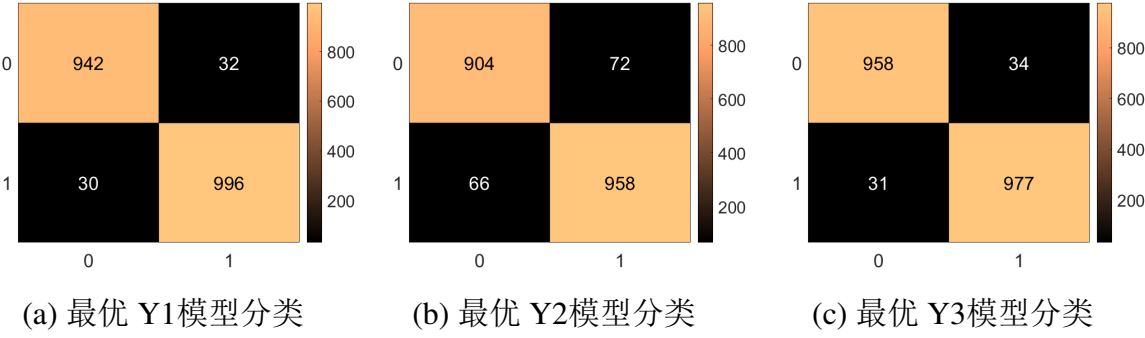


图 13: BP神经网络分类结果-最优化

将测试集带入模型，综合三组分类结果，可以得到如下的分类性能评价：

表 11: BP神经网络分类评价-最优化

	Y1分类	Y2分类	Y3分类
Accuracy	0.9695	0.9310	0.9675

绘制三次单项分类的ROC曲线：

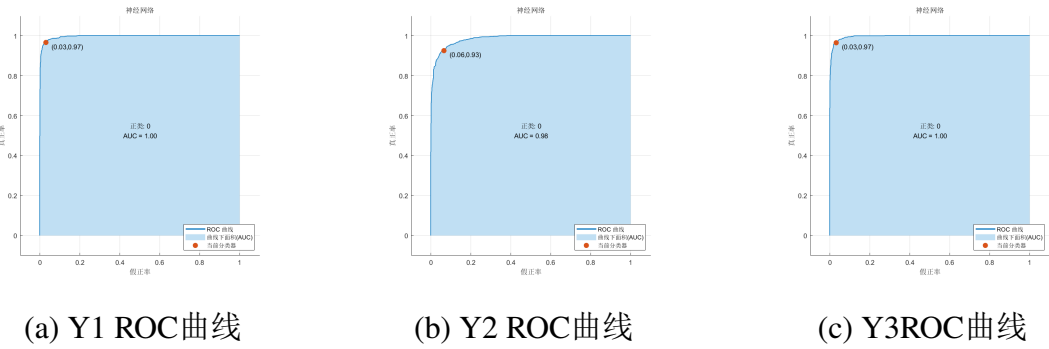


图 14: BP神经网络分类结果-最优化

比较4.2中第一次优化模型与本次最优模型的分类得分：

表 12: BP神经网络优化性能比较

	Y1分类	Y2分类	Y3分类
原始Accuracy	0.9255	0.9440	0.9490
最优Accuracy	0.9695	0.9310	0.9675

通过优化，我们能看到优化模型在分类性能上有一定的提升。

4.4 其他分类模型训练

本实验采用MATLAB分类工具箱对数据集进行多种不同分类模型的训练，并分布对模型进行打分评价。

4.4.1 逻辑回归模型训练

逻辑回归是最简单的分类模型，将训练集和验证集合并后进行五折交叉验证训练，得到基于逻辑回归的分类模型结果和评分：

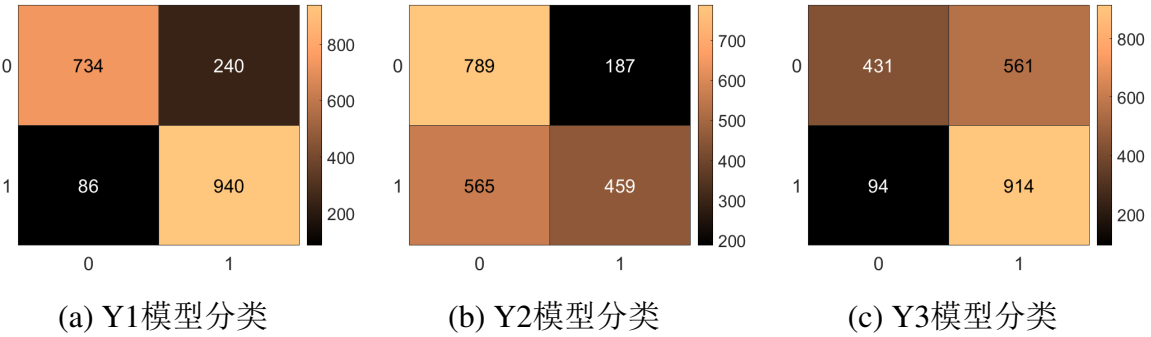


图 15: 逻辑回归分类结果

将测试集带入模型，综合三组分类结果，可以得到如下的分类性能评价：

表 13: 逻辑回归模型分类评价

	Y1分类	Y2分类	Y3分类
Accuracy	0.8370	0.6240	0.6725

绘制三次单项分类的ROC曲线：

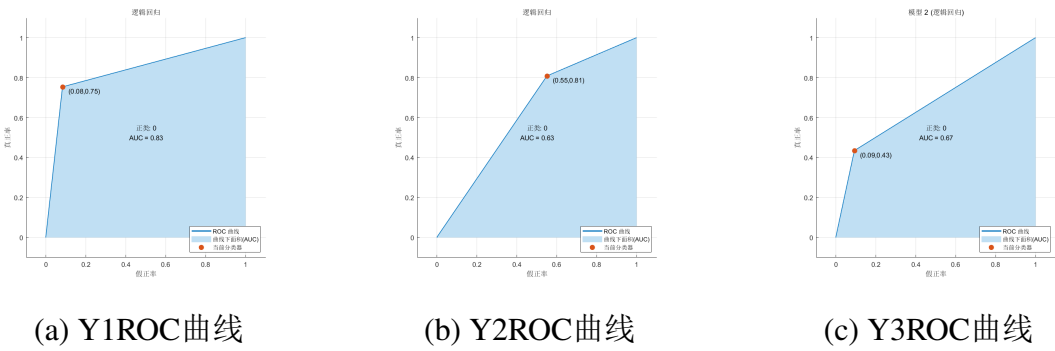


图 16: 逻辑回归分类结果 - ROC曲线

尽管上述结果是多次训练后的结果，但是可以看出逻辑回归的分析并不理想。

4.4.2 K近邻模型训练

本节采用K近邻方法进行分类模型训练，模型参数设置如表14所示：

表 14: K近邻模型参数

模型参数	临近点个数	距离度量方法	距离权重
	5	Euclidean	等距离

将训练集和验证集合并后进行五折交叉验证训练，得到基于K近邻的分类模型结果和评分：

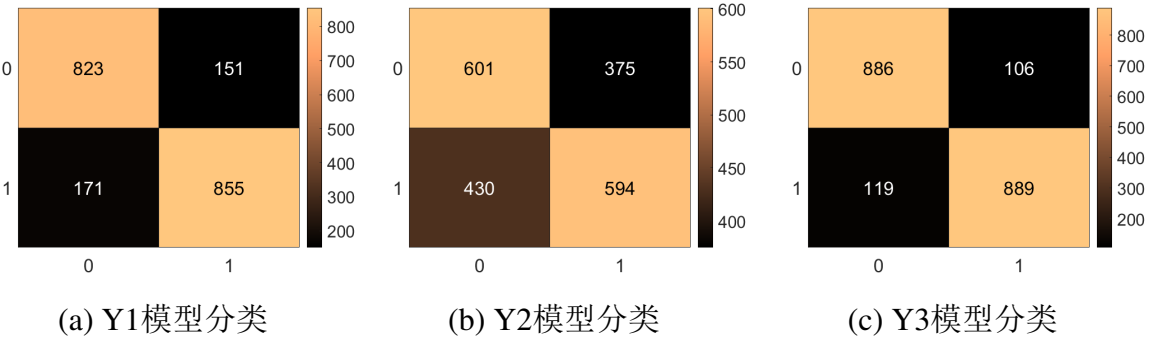


图 17: K近邻模型分类结果

将测试集带入模型，综合三组分类结果，可以得到如下的分类性能评价：

表 15: K近邻模型分类评价

	Y1分类	Y2分类	Y3分类
Accuracy	0.8390	0.5975	0.8875

绘制三次单项分类的ROC曲线：

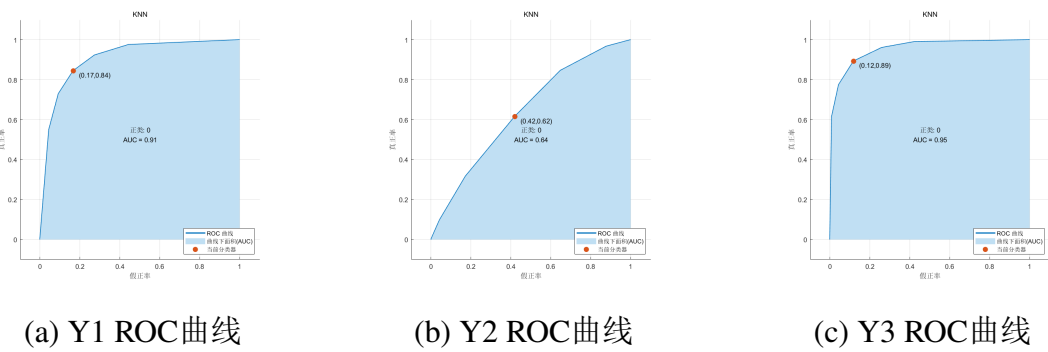


图 18: K近邻分类结果-ROC曲线

4.4.3 支持向量机模型训练

本节采用支持向量机方法进行分类模型训练，模型参数设置如表16所示：

表 16: 支持向量机模型参数

模型参数	核函数	多类方法	核尺度
	高斯核	一对一	自动

将训练集和验证集合并后进行五折交叉验证训练，得到基于支持向量机的分类模型结果和评分：

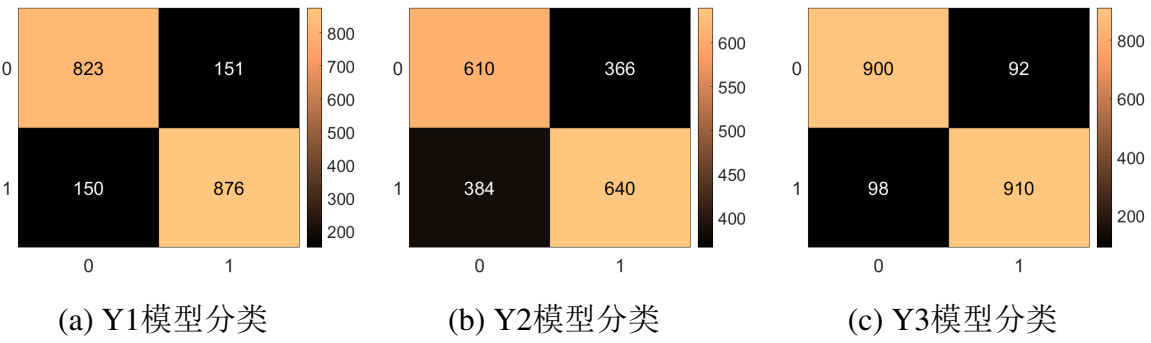


图 19: 支持向量机分类结果

将测试集带入模型，综合三组分类结果，可以得到如下的分类性能评价：

表 17: 支持向量机分类评价

	Y1分类	Y2分类	Y3分类
Accuracy	0.8495	0.6250	0.9050

绘制三次单项分类的ROC曲线：

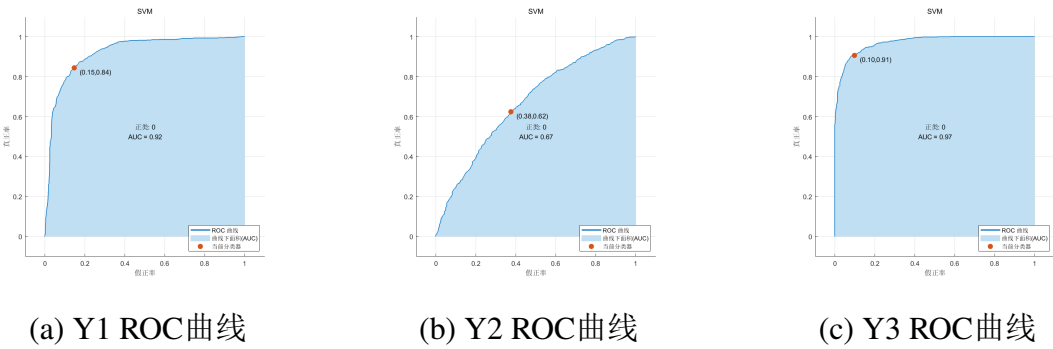


图 20: 支持向量机分类结果 - ROC曲线

4.4.4 分类决策树模型训练

本节采用分类决策树方法进行分类模型训练，模型参数设置如表18所示：

表 18: BP神经网络参数

模型参数	最大分裂数	分裂准则	最大树深度
	10	基尼系数	10

将训练集和验证集合并后进行五折交叉验证训练，得到基于分类决策树的分类模型结果和评分：

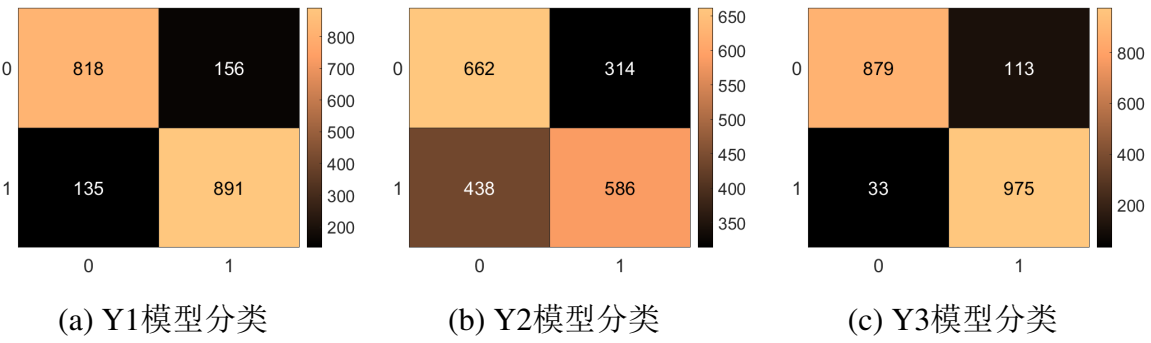


图 21: 分类结果

将测试集带入模型，综合三组分类结果，可以得到如下的分类性能评价：

表 19: 支持向量机分类评价

	Y1分类	Y2分类	Y3分类
Accuracy	0.8545	0.6240	0.9050

绘制三次单项分类的ROC曲线：

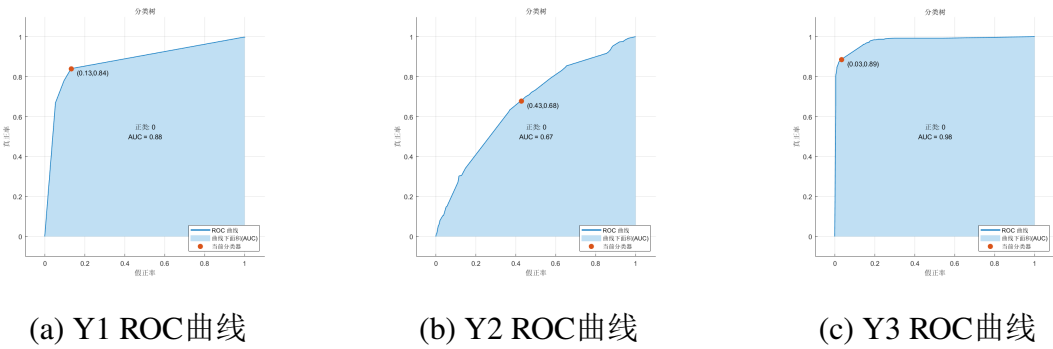


图 22: 分类决策树分类结果 - ROC曲线

本节中共计进行了四组实验，所选用的模型在部分分类问题上取得了不错的成果。但是模型对故障Y1和故障Y3的分类效果较好，但是在Y2上分类性能都不甚理想。

五、模型测试与评价

5.1 模型测试得分

本文中一共做了八组模型训练，将其分类性能做比较得到如下的分类结果表格

表 20: 模型测试得分

Accuracy	Y1分类	Y2分类	Y3分类
反向神经网络整体训练	0.9255	0.9440	0.9490
反向神经网络单次训练	0.9515	0.9345	0.9671
反向神经网络降维训练	0.9245	0.9230	0.9475
反向神经网络优化训练	0.9695	0.9310	0.9675
逻辑回归	0.8370	0.6240	0.6725
K近邻	0.8390	0.5975	0.8875
支持向量机	0.8495	0.6250	0.9050
分类树	0.8545	0.6240	0.9050

从性能评价中可以得到，反向神经网络优化训练的模型结果分类性能最好，这说明了神经网络是一种性价比最高的分类模型。

5.2 实验结论

通过本次的实验，我充分认识到了基于数据建模方法的故障诊断实验流程，重点学习了如何用MATLAB的代码和工具箱的方法设计机器学习模型。本实验主要的工作有三部分，分别是前序性工作数据集预处理与特征工程、主体工作机器学习分类模型训练与测试分类、后续工作模型评估与模型优化。本文中涉及到的五个机器学习方法都属于较为基础的算法，其中分类性能最好的方法为反向传播神经网络模型，因为其高超的分类能力与较强的泛化能力，所以神经网络模型及其衍生出来的各个算法，是现在研究的热点。

由于机能限制，部分算法是无法满足本次实验的多目标分类的要求的，在本实验中采用了分开单项训练的方式，但是这样对整体的分类能力高度以来各分类目标之间和各特征变量之间的相关系数。而本问题中各目标之间的相关系数较低，所以分开训练的结果在合起来之后的性能及其有限。

这里提出一种可能的下一步工作方案，建立一个组合模型投票机制，通过先期训练大量的单独分类的模型，并将其集成为一个联合投票模型。当输入一组特征变量之后，该模型做出多组分类结果，并将至与正确的分类标签做对比，计算各个结果的正确与错误率，依次为对该模型分类性能反馈依据，并通过该反馈信息，对错误的单项分类器做输出惩罚，对分类正确的单项分类器做输出奖励（即对总结果的影响做不同的权重赋值），并依次不断迭代和训练，以训练得到最终的联合分类模型。这样就可以通过单项分类模型组合得到整体分类性能也良好的模型。