第三章 研究过程

3.1 使用工具介绍

本次实习项目，主要基于 Python 环境进行相关分析，在神经网络超参数的寻优过程和图形绘制中使用 Matlab 进行辅助分析，具体而言：

1. 主要使用 sklearn 进行神经网络的搭建和训练；
2. 使用 imblearn 进行非平衡数据的处理，具体使用 SMOTE 算法的过采样；
3. 在测试集与训练集的划分中，为自行实现，保持两个集合中都有一定数目的负类样本；
4. 在特征分析中，使用的是 sklearn 的 RandomForestRegressor 的重要度分析；
5. 在超参数的最优化与图形绘制中，使用 Matlab 进行辅助分析与图形绘制；

3.2 特征选择

使用随机森林特征选择的方法进行分析，可以得到不同特征在分类过程中的不同重要性程度，具体结果如图所示。

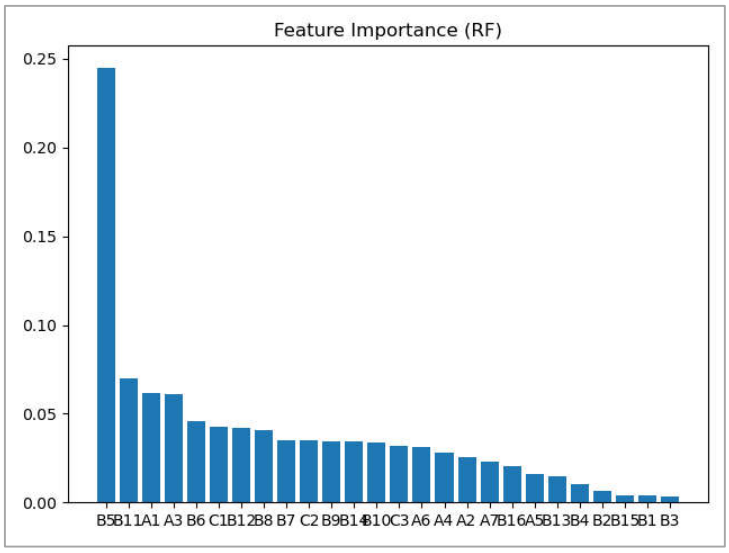


图 3.1 各特征重要性度量

3.3 不平衡数据的处理

本次实验处理非平衡样本，基于 Python 下完成，在测试集与训练集的划分中，通过特殊配置，在保持随机性的同时，使得训练和测试集中都有一定的正、负类样本；对于训练数据集的非平衡现象，借助 imblearn 包，使用 SMOTE 算法对于负类样本进行过采样处理，避免了训练的分类器出现无意义的分类决策（将测试集全判断为正类）。

3.3.1 面对的问题

在该二分类问题中，因为正类样本（指，正常样本）的数目远大负类样本（指，故障样本）的数目，从而带来两个问题：

1. 划分训练集与测试集

   对原始数据进行训练集与测试集的划分，如何保持测试集与训练集中都含有一定的正、负类样本。

2. 非平衡样本的模型训练

   在神经网络训练的过程中，因正类样本的数量较多，如何防止训练所得模型出现无意义分类现象（全判断成正类）。

3.3.2 处理方法

[问题 1] 划分训练集与测试集

对于数据进行划分时，将正类负类按照不同的比例进行划分，基本保证在测试集中负类有 5 个样本，在训练集中负类有 21 个，从而克服在测试集或者训练集中没有负类样本的问题。

指标	正类	负类
总数目	3809	26
测试集比例	30.00%	19.00%
训练集比例	70.00%	81.00%
验证集比例	30.00%	30.00%

注：验证集比例，为在训练集中占比。

表 3.1 训练集与测试集划分

[问题 2] 非平衡样本的模型训练

对于训练数据集的非平衡现象，基于 Python 下，借助 imblearn 包，使用 SMOTE 算法对于负类样本进行过采样处理，避免训练的分类器出现无意义的分类决策（将测试集全判断为正类）。

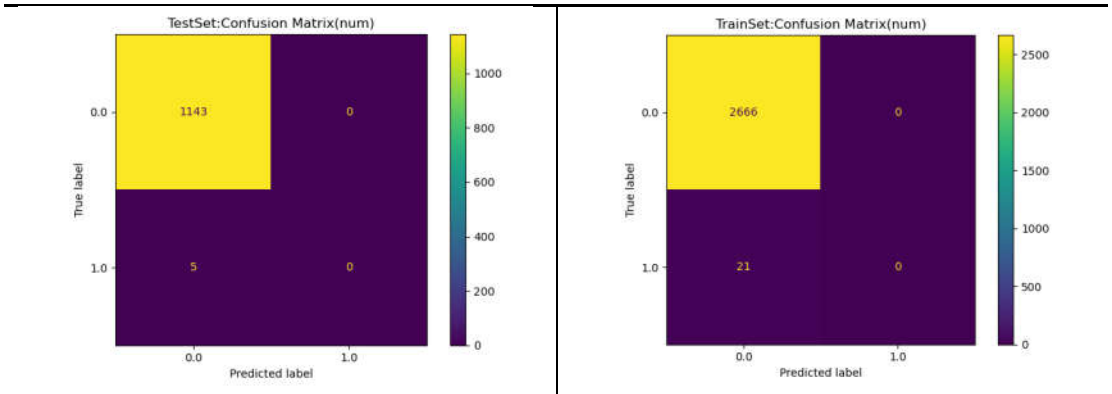
具体而言，在将数据集按照表进行划分后，在训练集合中，采样前后样本占比如表所示。

类比	采样前	采样后
正类占比	99.22%	50.00%
负类占比	0.78%	50.00%

表 3.2 采样前后的正负类占比

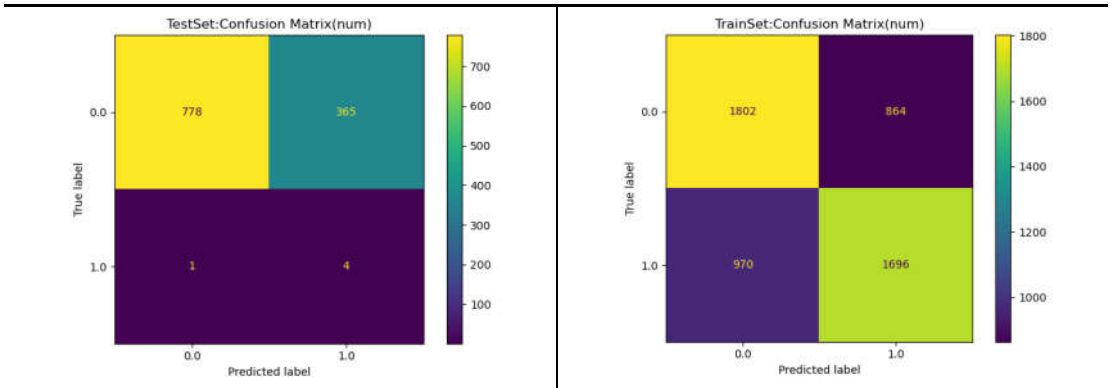
因为神经网络中，根据损失函数进行迭代优化，当训练样本中存在明显的不平衡时容易出现训练得到的模型出现无意义的判断决策，而在使用 STOME 算法将负类样本进行过采样处理后，训练数据中正负类样本各占 50%，从而问题 2 得到解决。

采样前后预测结果对比，用模型对测试集和训练集进行预测，得到混淆矩阵如图。可以看到在没有进行采样前，模型表现出的是无意义的判断，将所有样本均判断为正类，而采样后该问题得到明显改善。



测试集 Test Set　　训练集 Train Set（采样前）

图 3.2 采样前预测结果（混淆矩阵）



测试集 Test Set	训练集 Train Set（采样后）

图 3.3 采样后预测结果（混淆矩阵）

3.4 网络搭建和参数设置

3.4.1 配置工具与环境

主要使用 Python 下 sklearn 包，使用 MLPClassifier 进行网络的搭建，网络类型为 BP 神经网络。

3.4.2 网络参数设置

因为负类样本并不多，且数据量大小也并不是非常大，从而采用一个隐藏层的网络结构。

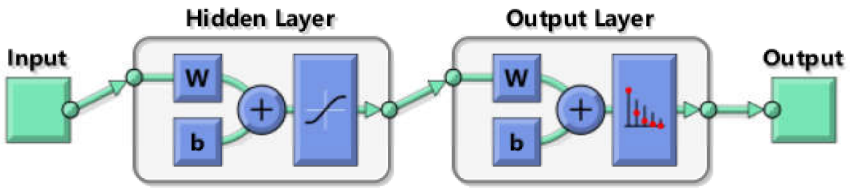


图 3.4 搭建网络结构

具体而言参数设置为：

参数	名称	值
隐藏层节点	hidden_layer_sizes	27
求解器	solver	adam
激活函数	activation	relu
验证集占比	validation_fraction	0.3
学习率	learning_rate	0.0008
损失函数终止条件	tol	1.00E-04

表 3.3 具体网络结构

3.4.3 调参和参数选择

主要调节的参数为隐藏层节点数目和学习率，使用网格搜索法进行调参。具体而言，节点数目从 10 取到 29 个，学习率从 0.1 到 1.0e-05 不等间隔取点 14 个从而构成一个 24×14 的结果矩阵。

值得注意的是，因为数据的不平衡问题，可能出现分类器将样本全部判断为正类的情况。因为负类样本数目较少，当时能准确挑选出负类是非常重要的，从而单纯的将准确率作为挑选参数的指标是不合适的。在这里使用所有样本的总正确率和负类样本正确率的加权平均值作为判断标准，即：

$$criterion = r \times Accuracy\_all + (1-r) \times Accuracy\_positive \tag{1}$$

$$r \in [0,1]$$

Node\NR	1.00E-05	0.0001	0.0003	0.0005	0.0008	0.001	0.005	0.008	0.01	0.02	0.04	0.06	0.08	0.1
10	0.502178	0.502178	0.585976	0.653223	0.666725	0.659321	0.470645	0.503049	0.495645	0.367857	0.462544	0.510889	0.504791	0.497822
11	0.760453	0.531446	0.429355	0.603397	0.586934	0.502178	0.454094	0.502178	0.497822	0.485192	0.379355	0.275958	0.502178	0.502178
12	0.496951	0.546254	0.48162	0.542509	0.486063	0.511324	0.497822	0.497822	0.421777	0.497648	0.28615	0.22953	0.483014	0.497822
13	0.502178	0.548693	0.590767	0.475174	0.5277	0.606272	0.494774	0.523432	0.497822	0.496951	0.527265	0.576394	0.497822	0.502178
14	0.497822	0.351307	0.493031	0.395383	0.520383	0.473606	0.52047	0.502178	0.497822	0.511324	0.37047	0.369599	0.516986	0.502178
15	0.497822	0.497822	0.412282	0.385714	0.492596	0.509408	0.504791	0.442073	0.502178	0.431359	0.507927	0.647213	0.385017	0.502178
16	0.497822	0.366115	0.398955	0.495645	0.4973	0.497822	0.497822	0.497822	0.494338	0.504355	0.444686	0.738676	0.684233	0.490854
17	0.440418	0.475261	0.472561	0.470383	0.53162	0.497822	0.502613	0.497822	0.399303	0.483624	0.305401	0.497822	0.497822	0.497822
18	0.502178	0.440592	0.205401	0.577787	0.476916	0.266812	0.451916	0.497822	0.495645	0.470993	0.421864	0.252178	0.497822	0.497822
19	0.497822	0.379791	0.362718	0.320993	0.502178	0.50392	0.495645	0.516115	0.497822	0.465157	0.428136	0.34216	0.495209	0.502178
20	0.502178	0.557404	0.583537	0.671516	0.643206	0.505226	0.451481	0.490418	0.502178	0.495645	0.462631	0.613328	0.504355	0.502178
21	0.758275	0.535366	0.48824	0.540331	0.493467	0.506359	0.502178	0.502178	0.496516	0.420732	0.423606	0.651132	0.497822	0.502178
22	0.502178	0.304965	0.358101	0.28101	0.518728	0.497387	0.510453	0.497822	0.497387	0.363763	0.40723	0.502178	0.502178	0.497822
23	0.502178	0.576045	0.491115	0.450958	0.444861	0.497822	0.486934	0.487369	0.473345	0.521864	0.40601	0.497822	0.502178	0.497822
24	0.502178	0.502178	0.48284	0.416115	0.490418	0.338937	0.492596	0.497387	0.50784	0.259146	0.403833	0.472822	0.497822	0.497822
25	0.497822	0.497822	0.476568	0.462631	0.491725	0.479094	0.496951	0.398519	0.491725	0.312979	0.364373	0.502178	0.497822	0.497822
26	0.494338	0.305139	0.508101	0.466028	0.429181	0.485192	0.452352	0.496516	0.509582	0.416551	0.438676	0.647213	0.497822	0.497387
27	0.497822	0.592247	0.481707	0.602439	0.740592	0.565941	0.426132	0.588328	0.502178	0.310366	0.458624	0.505226	0.497822	0.502178
28	0.502178	0.320993	0.317422	0.429617	0.502613	0.492596	0.431359	0.494338	0.496516	0.477178	0.462544	0.472561	0.502178	0.502178
29	0.48824	0.423868	0.436237	0.48824	0.489983	0.493902	0.377091	0.493467	0.502178	0.349826	0.361324	0.497822	0.502178	0.502178

表 3.4 Criterion 计算结果

可以画出在 $r=\frac{1}{2}$ 时，的判断标准的图像与单纯的正确率图像进行对比，图中标识的 Accuracy 表示(1)式中的 criterion.

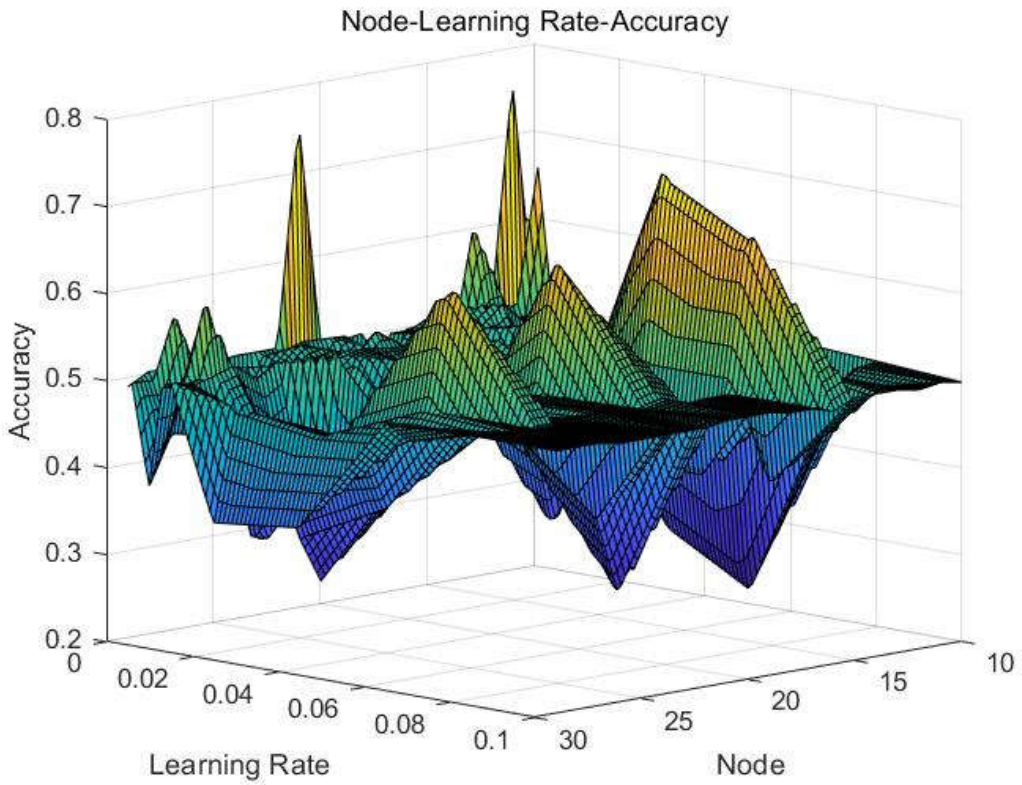


图 3.5 基于 Criterion 的参数选择（r=0.5）

可以看到有几个峰值，不妨每个峰值都去检查一遍，排除无意义的分类判断，从而可以找到$[Node=27, LearningRate=0.0008]$为最优参数。

在这个参数条件下 criterion 值为 0.740592334，而总体准确率 Accuray_all 为 68.12%，负类正确率 Accuracy_positive=0.8，即整体而言是一个非常好的参数。

且在表中，该参数的四周其 criterion 值都处于下降状态，从而可以认为，这个参数是调整到了一个最优值的（局部）。且验证其他局部最优的可行性后，发现这个参数是最好的，可以近似认为该参数点是在搜索范围内的最优点。

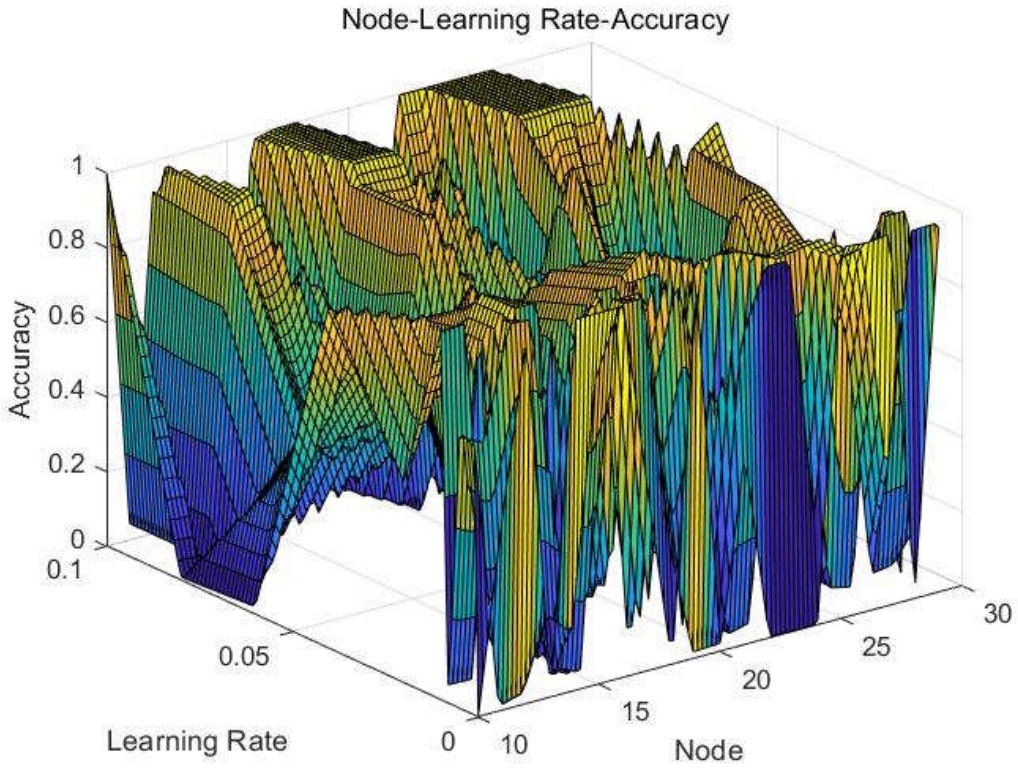


图 3.6 基于 Accuray 的参数选择（r=1）

可以看出在使用 $r=1$ 时，仅依靠 Accuray_all 进行选择是非常困难的，存在大量的无意义的判断的参数组合。

3.5 模型训练及其结果

在 3.5 中已经找到了最优的参数设置 $[Node=27, LearningRate=0.0008]$，在该参数设置下，完成模型的训练和结果的预测，同时对于测试集数据进行相应预测得到相关结果和指标。

3.5.1 损失函数

在使用 $[Node=27, LearningRate=0.0008]$ 时，可以得到需要训练 53 轮，可以得到相应训练过程中的损失函数的变化。

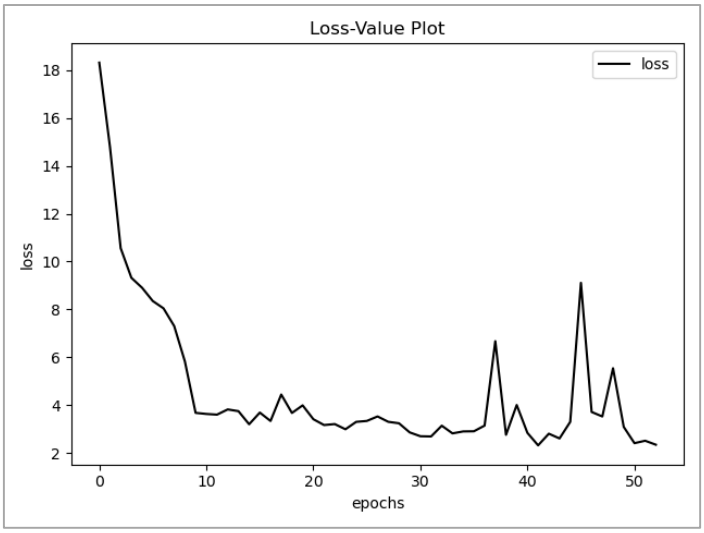


图 3.7 训练过程的损失函数值

可以看到损失函数值基本上一直处于下降的状态，也尝试了更小的终止条件，发现没有什么改变，而使用该 tol= 0.0001，是一个比较好的值。

3.5.2 ROC 曲线

在使用 $[Node = 27, LearningRate = 0.0008]$ 时，将训练所得模型用于 TestSet（测试集）的预测，可以得到相应的 ROC 曲线和 AUC 值，其中 $AUC = 0.74$。

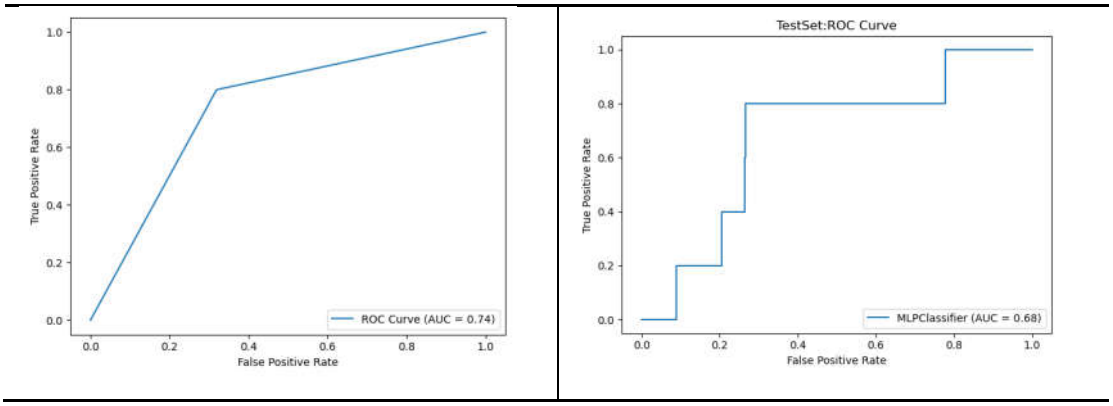


图 3.8 测试集上的 ROC 曲线

3.5.3 混淆矩阵

因为负类样本数目很少，从而在测试集和训练集上，都值得看一下模型在其上的判断情况，一是为了避免无意义的判断（全判断为某一类），二是为了看看模型对于负类样本的判断能力如何（这是该项目也非常关注的点）。

在使用[$Node=27, LearningRate=0.0008$]时，从而可以对于 TestSet、TrianSet（过采样前）和 TrianSet（过采样后）三组数据再绘制它们的混淆矩阵。

发现模型在三个数据集上的表现都还可以接受，各类的正确率都有 60%以上，可以认为模型在整体数据上的表现都是可以接受的，且对于负类样本也有一定的区分度。

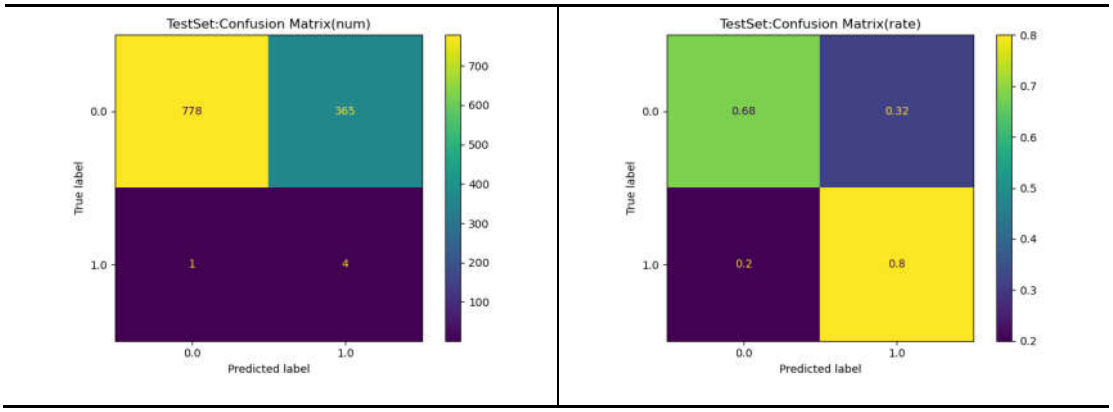


图 3.9 在测试集的混淆矩阵

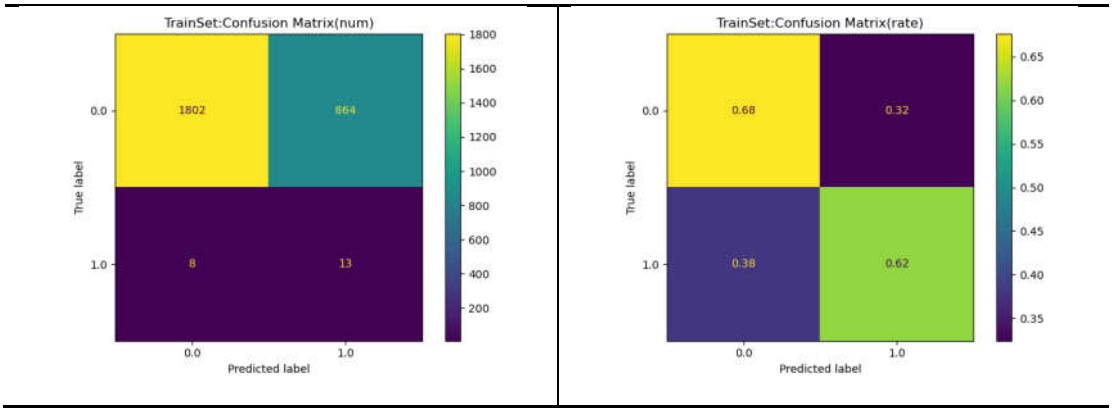


图 3.10 在训练集（采样前）的混淆矩阵

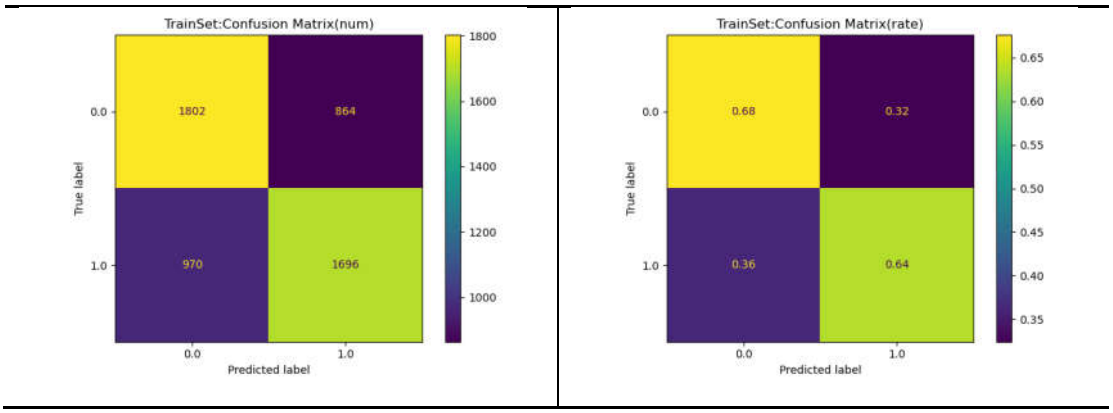


图 3.11 在训练集（采样后）的混淆矩阵

3.6 其他模型评价指标

除上述指标外，还计算了其他指标，如 F1 值、AP 值、AUC 值、fpr 值、tpr 值，具体见下表。

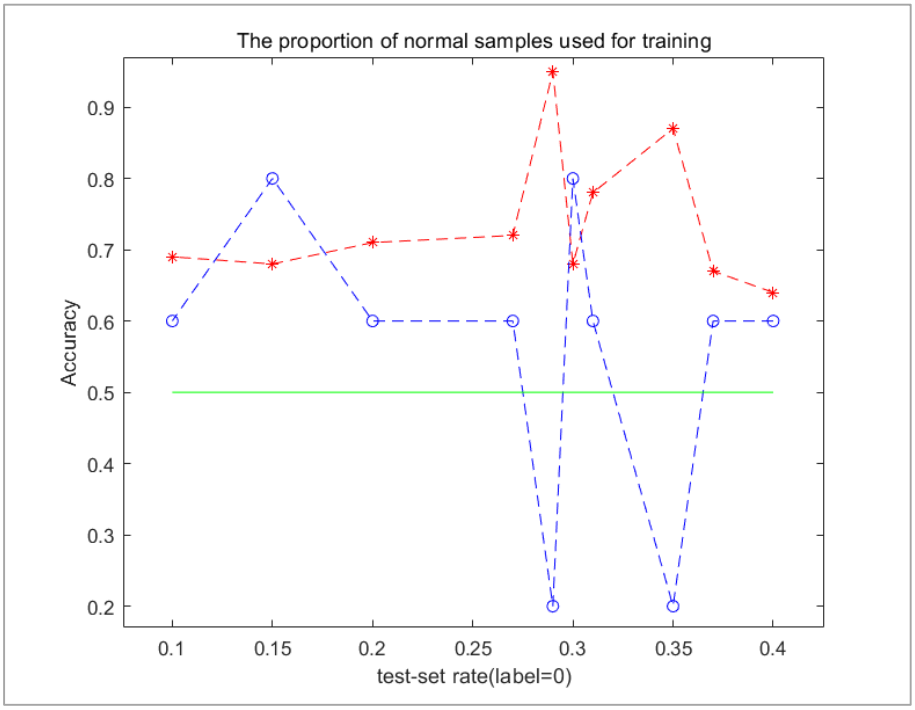
指标	Accuracy	F1	AP	fpr	tpr	AUC
值	68.12%	0.02139	0.009311	0.319335	0.8	0.74033

表 3.5 其他模型指标

3.7 排除训练集划分的偶然性

为了排除因为训练集因为恰好把合适数据划分入内，而表现很好的可能，测试多种不同的测试集划分比例，分别得到它们的整体正确率 Accuracy 和对于负类 Accuracy_positive 判断的准确度，将它们绘制在图上。

可以看到它们除了个别点外，其他点的两个准确度都是在 60%以上的水平，可以认为模型的表现并不是因为训练集恰好的划分导致的，训练中使用的训练集比例为 $testset\_rate = 0.3$。



注:蓝色为负类样本正确率，红色为正类样本正确率。

图 3.12 不同比例训练集划分的准确率