华东师范大学数据科学与工程学院实验报告

课程名称：统计学习与机器方法	**年级：2019**	**实践成绩：**
指导教师：董启文	**姓名**：周辛娜	**学号**：10195501442
上机实践名称： Project1七种方法解决二分类问题	**上机实践时间**：	2021.12.15

实验数据集：

uci心脏病数据集

数据下载网址：Index of /ml/machine-learning-databases/heart-disease (uci.edu)。共有4个数据库，分别为克利夫兰，匈牙利，瑞士和VA长滩，这里采用的是克利夫兰数据库的实验集。下载的是processed.cleveland.data，由于文件后缀是.data，不是机器学习中常用的.csv，所以后面需要整理数据得到.csv文件。

- Parent Directory
- Index
- WARNING
- ask-detrano
- bak
- cleve.mod
- cleveland.data
- costs/
- heart-disease.names
- hungarian.data
- long-beach-va.data
- new.data
- processed.cleveland.data
- processed.hungarian.data
- processed.switzerland.data
- processed.va.data
- reprocessed.hungarian.data
- switzerland.data

一、探索性数据分析

数据全貌：

	63.0	1.0	1.0.1	145.0	233.0	1.0.2	2.0	150.0	0.0	2.3	3.0	0.0.1	6.0	0
0	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.0	3.0	2
1	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.0	7.0	1
2	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.0	3.0	0
3	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.0	3.0	0
4	56.0	1.0	2.0	120.0	236.0	0.0	0.0	178.0	0.0	0.8	1.0	0.0	3.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
297	45.0	1.0	1.0	110.0	264.0	0.0	0.0	132.0	0.0	1.2	2.0	0.0	7.0	1
298	68.0	1.0	4.0	144.0	193.0	1.0	0.0	141.0	0.0	3.4	2.0	2.0	7.0	2
299	57.0	1.0	4.0	130.0	131.0	0.0	0.0	115.0	1.0	1.2	2.0	1.0	7.0	3
300	57.0	0.0	2.0	130.0	236.0	0.0	2.0	174.0	0.0	0.0	2.0	1.0	3.0	1
301	38.0	1.0	3.0	138.0	175.0	0.0	0.0	173.0	0.0	0.0	1.0	?	3.0	0

302 rows × 14 columns

直接通过pd.read_csv函数读取processed.cleveland.data文件，得到现在的数据全貌。

数据集有302行，14列，每行表示一个病人，13列特征，1列标签，可以看到现在的文档中是没有列名的，就需要加入列名，从官网上下载了heart-disease.names文件，了解了具体每一列的详细信息，从而进行添加。

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.0	3.0	2
1	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.0	7.0	1
2	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.0	3.0	0
3	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.0	3.0	0
4	56.0	1.0	2.0	120.0	236.0	0.0	0.0	178.0	0.0	0.8	1.0	0.0	3.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
297	45.0	1.0	1.0	110.0	264.0	0.0	0.0	132.0	0.0	1.2	2.0	0.0	7.0	1
298	68.0	1.0	4.0	144.0	193.0	1.0	0.0	141.0	0.0	3.4	2.0	2.0	7.0	2
299	57.0	1.0	4.0	130.0	131.0	0.0	0.0	115.0	1.0	1.2	2.0	1.0	7.0	3
300	57.0	0.0	2.0	130.0	236.0	0.0	2.0	174.0	0.0	0.0	2.0	1.0	3.0	1
301	38.0	1.0	3.0	138.0	175.0	0.0	0.0	173.0	0.0	0.0	1.0	?	3.0	0

302 rows × 14 columns

现在是修改之后的数据，由于"目标"字段(num)是指患者中是否存在心脏病。它是介于 0（无存在）到 4 之间的整数值。克利夫兰数据库的实验集中在简单地尝试区分存在（值1，2，3，4）和不存在（值 0）。所以这里1,2,3,4是代表同样的意思，都是患者患病，实质上这就是一个二分类问题，为了方便后续模型处理，所以这里将num不大于0的都处理成1。经过这两个处理后再查看数据是否有缺失值，发现没有缺失值，故进行下一步。

数据可视化：

查看各列之间的关系：

由该图可以看到，各列之间还是存在一定的关系的。

将各列之间两两的散点图绘制出来，可以看到各列之间的关系

二、数据预处理

1. 区分定类 定序 定距 定比四种特征，将定类特征由整数编码转为实际对应的字符串

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	67.0	male	asymptomatic	160.0	286.0	fasting blood sugar > 120 mg/dl	showing probable or definite left ventricular ...	108.0	yes	1.5	2.0	3.0	3.0	1
1	67.0	male	asymptomatic	120.0	229.0	fasting blood sugar > 120 mg/dl	showing probable or definite left ventricular ...	129.0	yes	2.6	2.0	2.0	7.0	1
2	37.0	male	non-anginal pain	130.0	250.0	fasting blood sugar > 120 mg/dl	normal	187.0	no	3.5	3.0	0.0	3.0	0
3	41.0	female	atypical angina	130.0	204.0	fasting blood sugar > 120 mg/dl	showing probable or definite left ventricular ...	172.0	no	1.4	1.0	0.0	3.0	0
4	56.0	male	atypical angina	120.0	236.0	fasting blood sugar > 120 mg/dl	normal	178.0	no	0.8	1.0	0.0	3.0	0

2. 在pandas中，离散的定类和定序特征应该是object这样的对象类型，连续的定距和定比特征应该是int64或者是float64这样的浮点数类型，将离散的定类和定序特征转为one-hot独热向量编码。

	age	trestbps	chol	thalach	oldpeak	slope	num	sex_female	sex_male	cp_asymptomatic	...	exang_yes	ca_0.0	ca_1.0	ca_2.0	ca_3.0	ca_?	thal_3.0	thal_6.0	thal_7.0	thal_?
0	67.0	160.0	286.0	108.0	1.5	2.0	1	0	1	1	...	1	0	0	0	1	0	1	0	0	0
1	67.0	120.0	229.0	129.0	2.6	2.0	1	0	1	1	...	1	0	0	1	0	0	0	0	1	0
2	37.0	130.0	250.0	187.0	3.5	3.0	0	0	1	0	...	0	1	0	0	0	0	1	0	0	0
3	41.0	130.0	204.0	172.0	1.4	1.0	0	1	0	0	...	0	1	0	0	0	0	1	0	0	0
4	56.0	120.0	236.0	178.0	0.8	1.0	0	0	1	0	...	0	1	0	0	0	0	1	0	0	0

5 rows × 29 columns

三、划分训练集测试集

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

四、归一化

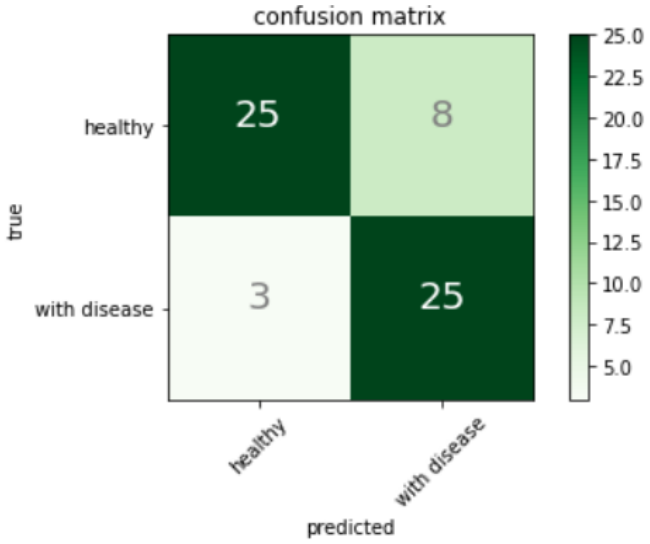
五、构建模型：

1.多层神经感知机MLP：

1) 先进行网格搜索，得到最优超参数：

```
MLPClassifier(hidden_layer_sizes=4, max_iter=1500)
```

2) 在最优模型上得到的混淆矩阵：



3) classification report:

	precision	recall	f1-score	support
healthy	0.89	0.76	0.82	33
with disease	0.76	0.89	0.82	28
accuracy			0.82	61
macro avg	0.83	0.83	0.82	61
weighted avg	0.83	0.82	0.82	61

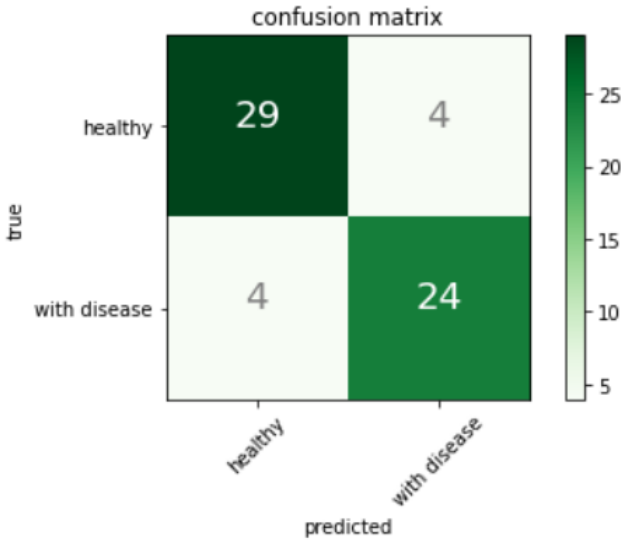
正确率达到82%，表现一般。

2.AdaBoost Classifier：

1) 先进行网格搜索，得到最优超参数：

```
AdaBoostClassifier(learning_rate=0.095)
```

2) 在最优模型上得到的混淆矩阵：



3) classification report:

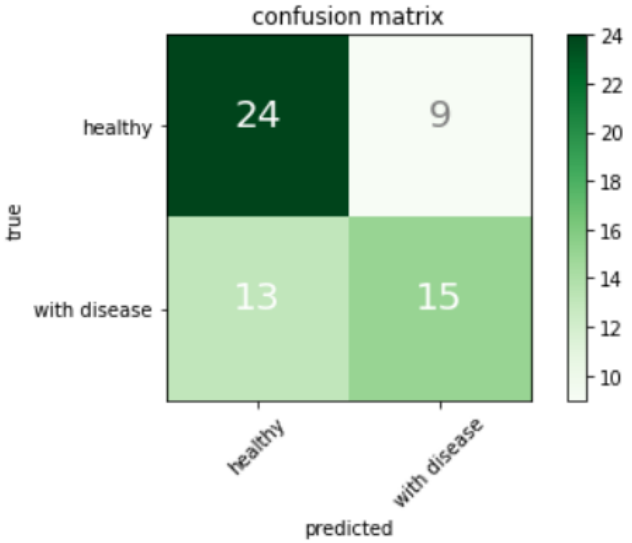
	precision	recall	f1-score	support
healthy	0.88	0.88	0.88	33
with disease	0.86	0.86	0.86	28
accuracy			0.87	61
macro avg	0.87	0.87	0.87	61
weighted avg	0.87	0.87	0.87	61

正确率达到87%，表现较好

3.KNN：

1) 先进行网格搜索，得到最优超参数 k=3

2) 在最优模型上得到的混淆矩阵：



3) classification report:

	precision	recall	f1-score	support
healthy	0.65	0.73	0.69	33
with disease	0.62	0.54	0.58	28
accuracy			0.64	61
macro avg	0.64	0.63	0.63	61
weighted avg	0.64	0.64	0.64	61

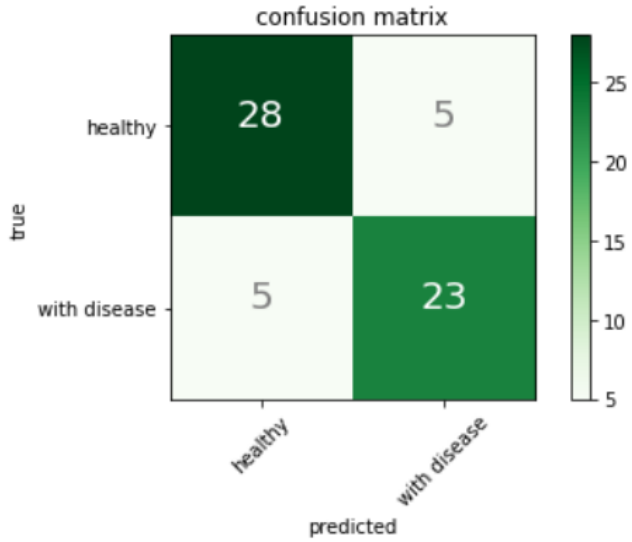
正确率达到64%，表现不佳。

4. Gaussian Naive Bayes：

1) 先进行网格搜索，得到最优超参数：

```
GaussianNB(var_smoothing=0.0001)
```

2) 在最优模型上得到的混淆矩阵：



3) classification report:

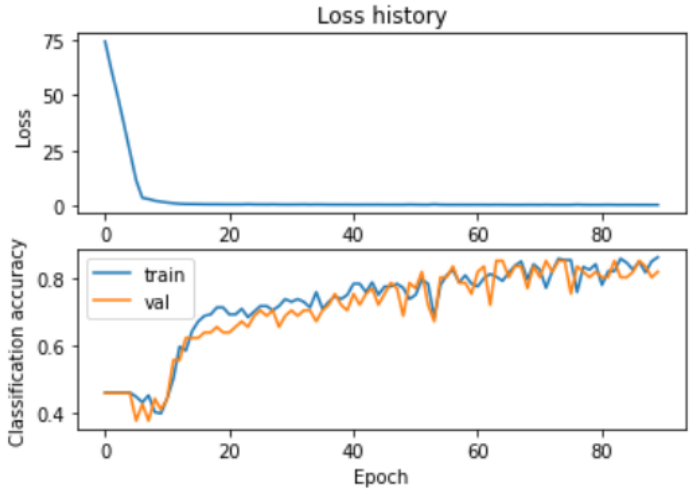
	precision	recall	f1-score	support
healthy	0.85	0.85	0.85	33
with disease	0.82	0.82	0.82	28
accuracy			0.84	61
macro avg	0.83	0.83	0.83	61
weighted avg	0.84	0.84	0.84	61

正确率达到84%，表现较好。

5.Neural network：

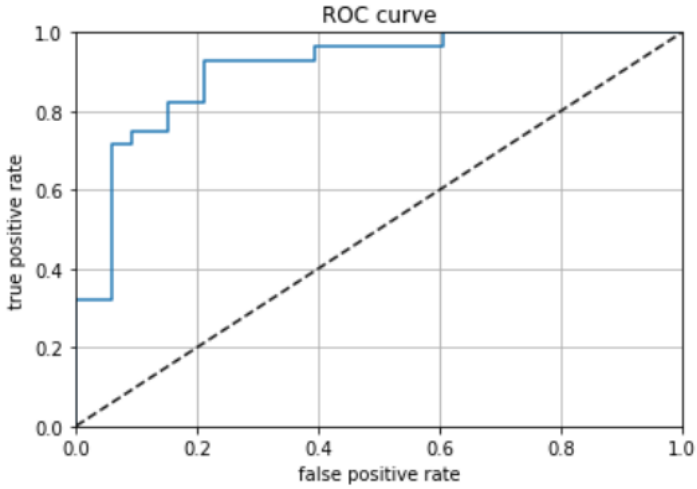
构建了两层神经网络，第一层的神经元个数是128个，第二层的为输出层，神经元个数为1

训练过程中的loss history和classification accuracy:



可以看到学习率设置得比较合理。最后在验证集的准确率有达到80%以上。

得到的ROC曲线:



ROC曲线围成的面积达到了0.9037，表现较好

```
auc(fpr,tpr)
```

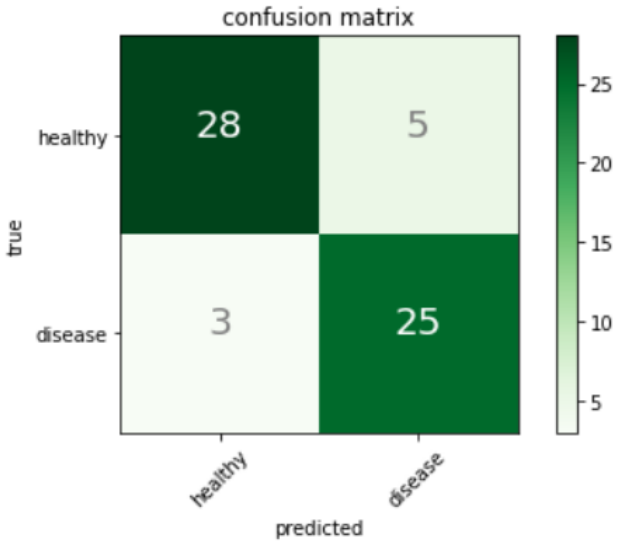
```
0.9036796536796537
```

6.Random forest:

1) 先进行网格搜索，得到最优超参数:

```
RandomForestClassifier(max_depth=6, n_estimators=110, random_state=5)
```

1) 在最优模型上得到的混淆矩阵:



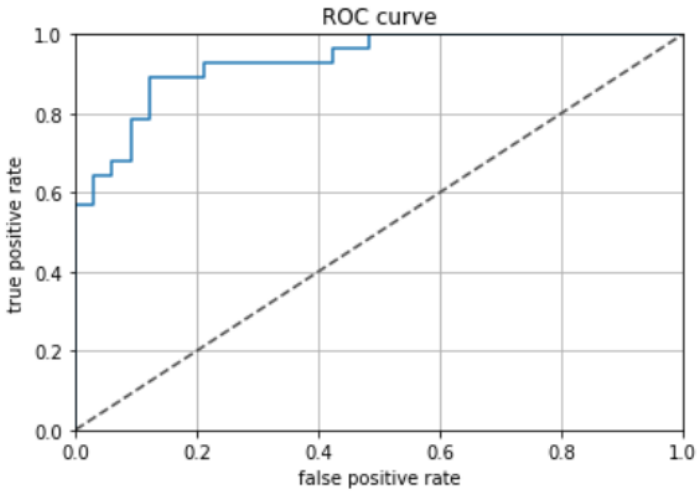
3) classification report:

	precision	recall	f1-score	support
healthy	0.90	0.85	0.88	33
with disease	0.83	0.89	0.86	28
accuracy			0.87	61
macro avg	0.87	0.87	0.87	61
weighted avg	0.87	0.87	0.87	61

正确率达到87%

定量分析：

得到的ROC曲线：



ROC曲线围成的面积：0.9329，表现较好，优于Neural network

```
auc(fpr,tpr)
```

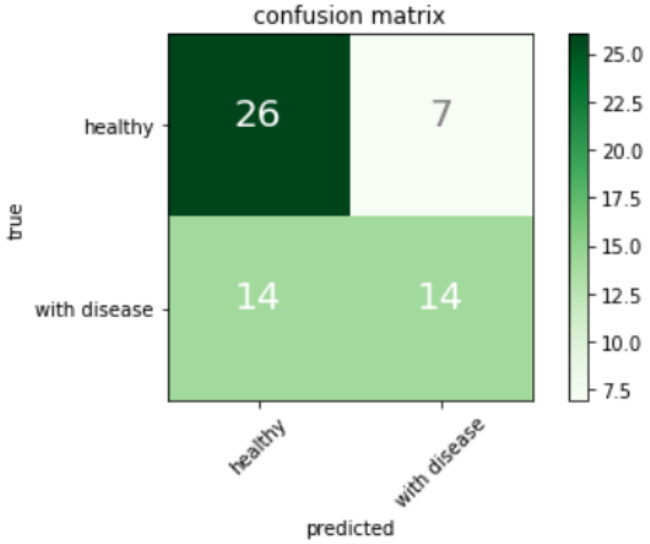
```
0.9329004329004329
```

7.SVM：

1) 先进行网格搜索，得到最优超参数：

```
SVC(gamma=0.0001)
```

2) 在最优模型上得到的混淆矩阵：



3) classification report:

	precision	recall	f1-score	support
healthy	0.65	0.79	0.71	33
with disease	0.67	0.50	0.57	28
accuracy			0.66	61
macro avg	0.66	0.64	0.64	61
weighted avg	0.66	0.66	0.65	61

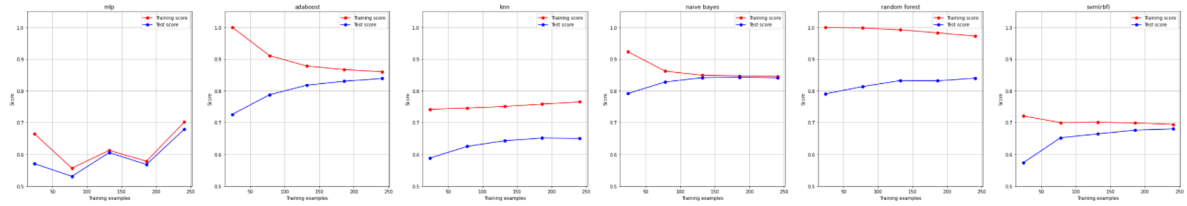
正确率达到66%，表现在该数据集上表现不佳

六、模型进行汇总与比较

1.用时比较

```
mlp用时:00:18:013573s
adaboost用时:00:06:187602s
knn用时:00:01:113120s
naive bayes用时:00:00:650259s
random forest用时:00:10:567751s
svm(rbf)用时:00:00:692116s
```

2. learning curve比较：



可以看到，在运行时间上，naive bayes用时最短，其次是svm(rbf)，knn，adaboost和random forest用时比较短，最长的是mlp。

朴素贝叶斯可以在很少的样本上获得不错的结果，不仅用时比较短，而且在learning curve中的表现也还不错。朴素贝叶斯计算速度远远胜过svm，random forest这样复杂的模型。朴素贝叶斯的分类效果其实不如其他分类器，贝叶斯天生学习能力比较弱。随着训练样本量的逐渐增大，贝叶斯和adaboost的训练准确率却逐渐下降，这证明样本量越大，贝叶斯需要学习的东西越多，对训练集的拟合程度也越来越差，反而比较少量的样本可以让贝叶斯还有较高的训练准确率。

再看过拟合的问题，首先可以观察到，所有模型在样本量很少的时候都是处于过拟合的状态，即在训练集上表现好，测试集上表现糟糕，但随着样本的逐渐增多，过拟合问题就逐渐消失了，不过每个模型的处理手段不同。比较强大的分类器，如svm，random forest是依靠快速升高模型在测试集上的表现来减轻过拟合问题，朴素贝叶斯不同，是依赖训练集上的准确率下降，测试集上的准确率上升来解决过拟合的问题。

接下来，再看每个算法在测试集上的拟合结果，即泛化误差的大小，随着训练样本数量的上升，所有模型的测试表现都上升了，但svm，knn，mlp在测试集上的表现远远不如adaboost，naive bayes和random forest。

3. 模型分类效果不好的原因：

1.超参数设置不合理
2.特征工程没做好
3.模型本身不适合

经过此次实验可以看到，svm，knn，mlp表现不佳，由于在实验过程中，采用了网格搜索交叉验证的方式选取了模型的超参数，所以应该从第二个和第三个原因去看待，这次实验是没有做特征选择的，应该是特征工程这一块还可以进一步优化，此外，也可能是svm，knn，mlp本身对于该数据集的表现能力就不强。

七、总结

本次实验采用克利夫兰数据库的实验集的uci心脏病数据集，采用七种方法实现了对于数据的分类，通过探索性数据分析、数据可视化、划分训练集测试集、归一化、网格搜索交叉验证确定模型最优参数、比较最后的运行时间、learning curve以及最后分类的准确性。在本数据集上，adaboost，naive bayes和random forest表现较优。