目 录

[一、 Letter Image Recognition字母图像识别数据集介绍 3](#_Toc27497899)

[1.1 数据集介绍 3](#_Toc27497900)

[1.1.1 简介 3](#_Toc27497901)

[1.1.2 相关信息 3](#_Toc27497902)

[二、 分类模型的建立和评估 5](#_Toc27497903)

[2.1导入数据 5](#_Toc27497904)

[2.1.1导入所需要的库 5](#_Toc27497905)

[2.1.2 导入数据集 5](#_Toc27497906)

[2.1.3 将类的字母替换为数字并对数据集按4：1比例进行分割，建立样本集和测试集 6](#_Toc27497907)

[2.2 建立各种分类模型并采用十折评估模型 6](#_Toc27497908)

[2.2.1 建立多种分类模型并采用缺省参数 6](#_Toc27497909)

[2.2.2 十折评估模型并画折线图 7](#_Toc27497910)

[2.3 建立基于支持向量机的分类模型，参数使用缺省值，对测试集进行分类 8](#_Toc27497911)

[2.3.1 训练数据集并预测 8](#_Toc27497912)

[2.3.2 采用accuracy\_score方法评价模型准确率 8](#_Toc27497913)

[2.3.2 采用classification\_report评价模型 8](#_Toc27497914)

[2.3.3 SVM小结 9](#_Toc27497915)

[三、 分类结果的分析 9](#_Toc27497916)

[四、 总结与展望 10](#_Toc27497917)

[五、 个人感受和反思 11](#_Toc27497918)

[六、 附录 11](#_Toc27497919)

[6.1 完整代码及运行结果 11](#_Toc27497920)

# Letter Image Recognition字母图像识别数据集介绍

## 数据集介绍

### 1.1.1 简介

Letter Image Recognition数据集，由David J. Slate创建、David J. Slate在1991年1月捐献，该数据集的主要描述如下所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Set Characteristics:** | Multivariate | **Number of Instances:** | 20000 | **Area:** | Computer |
| **Attribute Characteristics:** | Integer | **Number of Attributes:** | 16 | **Date Donated** | 1991-01-01 |
| **Associated Tasks:** | Classification | **Missing Values?** | No | **Number of Web Hits:** | 364823 |

过去的用法：-P. W. Frey和D. J. Slate（《机器学习》第6卷第2期，91年3月）：“使用荷兰式自适应分类器的字母识别”。

本文的研究调查了几种方法的能力荷兰式自适应分类器系统的变体，以学习

正确猜测与向量16相关的字母类别从字母的光栅扫描图像中提取的整数属性。获得的最佳精度略高于80％。让人感兴趣的是其他方法对相同数据的处理效果如何。

### 1.1.2 相关信息

目的是识别大量黑白中的每一个矩形像素显示为英语中的26个大写字母之一字母。字符图像基于20种不同的字体，每种字体这20种字体中的字母被随机扭曲以生成一个文件20,000个独特刺激。每种刺激转化为16种原始刺激当时的数值属性（统计矩和边数）缩放以适合从0到15的整数值范围。通常训练前16000个项目，然后使用结果模型预测剩余4000个字母的类别。

属性信息

实例数：20000

属性数量：17（字母类别和16个数字特征）

属性信息：

1. lettr大写字母（从A到Z的26个值）

2.框的x框水平位置（整数）

3.框的y框垂直位置（整数）

4.盒宽（整数）的宽度

5.高箱体（整数）

6. onpix总像素数（整数）

7. x-bar表示框内像素的x of（整数）

8.框内像素的y-bar平均值y（整数）

9. x2bar平均值x方差（整数）

10. y2bar平均值y方差（整数）

11. xybar平均值x y相关性（整数）

12. x \* x \* y的x2ybr均值（整数）

13. x \* y \* y的xy2br平均值（整数）

14. X-ege平均边缘计数从左到右（整数）

15. x-ege与y的xegvy相关（整数）

16. y-ege平均边缘计数从下到上（整数）

17. y-ege与x（整数）的yegvx相关性

缺少属性值：无

类分布：

789 A 766 B 736 C 805 D 768 E 775 F 773 G

734 H 755 I 747 J 739 K 761 L 792 M 783 N

753 O 803 P 783 Q 758 R 748 S 796 T 813 U

764 V 752 W 787 X 786 Y 734 Z

# 分类模型的建立和评估

## 2.1导入数据

### 2.1.1导入所需要的库

import pandas as pd

from sklearn.metrics import accuracy\_score

from sklearn.svm import SVC

from sklearn import metrics

from sklearn.model\_selection import train\_test\_split,cross\_val\_score

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.neural\_network import MLPClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive\_bayes import GaussianNB

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings("ignore")

### 2.1.2 导入数据集

x = pd.read\_csv('./letter-recognition.data',header=None)

2.1.3 画图查看每个类数据条数

letter = list('ABCDEFGHIJKLMNOPQRSTUVWXYZ')

Letter\_Sum = []

for i in letter:

Letter\_Sum.append(x[x[0]==i].shape[0])

plt.bar(letter, Letter\_Sum)

plt.show()

### 2.1.3 将类的字母替换为数字并对数据集按4：1比例进行分割，建立样本集和测试集

j=0

for i in letter:

x.replace(i,j,inplace=True)

j+=1

x\_class = x.iloc[:,:1]

x\_train = x.iloc[:,1:]

x\_tain,x\_test,\

class\_train,class\_test,= train\_test\_split(x\_train, x\_class,test\_size=0.1,random\_state=30)

## 2.2 建立各种分类模型并采用十折评估模型

### 2.2.1 建立多种分类模型并采用缺省参数

svm=SVC()

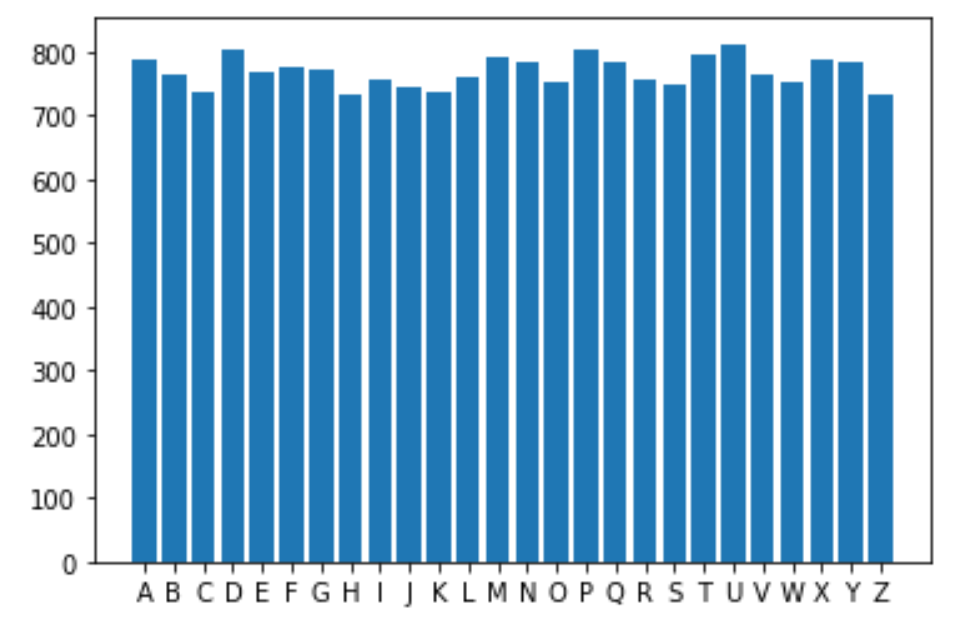
mlp=MLPClassifier()

knn=KNeighborsClassifier()

forest=RandomForestClassifier()

tree=DecisionTreeClassifier()

gass=GaussianNB()



### 2.2.2 十折评估模型并画折线图

svm\_scores = cross\_val\_score(svm,x\_tain,class\_train,cv=10,scoring='accuracy')

mlp\_scores = cross\_val\_score(mlp,x\_tain,class\_train,cv=10,scoring='accuracy')

knn\_scores = cross\_val\_score(knn,x\_tain,class\_train,cv=10,scoring='accuracy')

forest\_scores = cross\_val\_score(forest,x\_tain,class\_train,cv=10,scoring='accuracy')

tree\_scores = cross\_val\_score(tree,x\_tain,class\_train,cv=10,scoring='accuracy')

/gass\_scores = cross\_val\_score(gass,x\_tain,class\_train,cv=10,scoring='accuracy')

plt.figure(figsize=(16, 12))

f, ax = plt.subplots(3,2)

ax[0][0].plot(svm\_scores)

ax[0][1].plot(mlp\_scores)

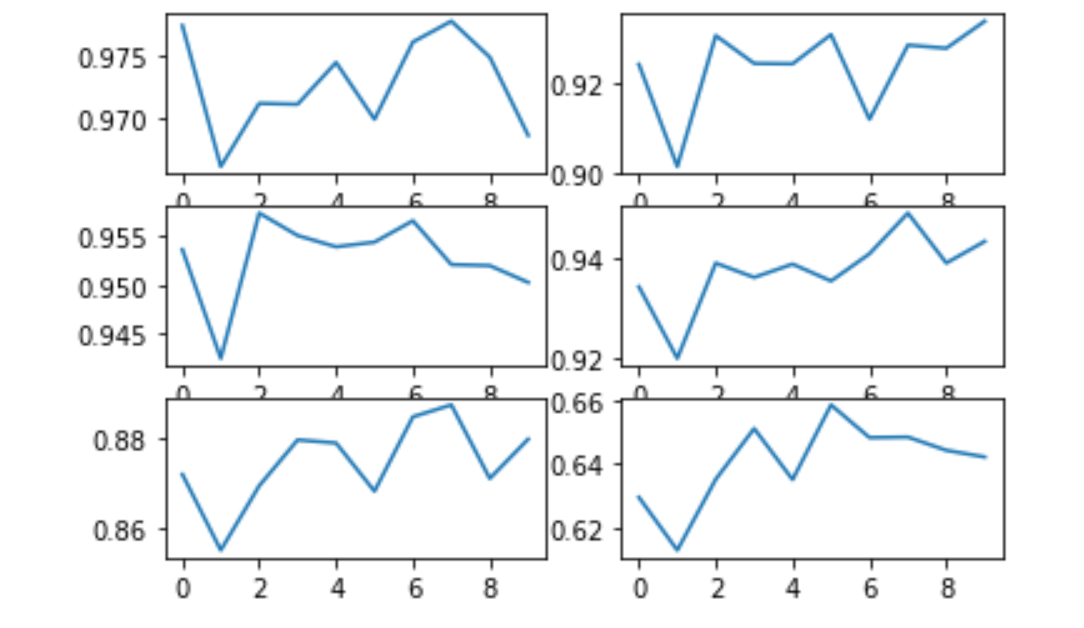
ax[1][0].plot(knn\_scores)

ax[1][1].plot(forest\_scores)

ax[2][0].plot(tree\_scores)

ax[2][1].plot(gass\_scores)

plt.show()

2.2.3 根据折线图分析模型优劣

从折线图中可以看出SVC模型表现最好，并且通过十折算法可以得出此数据集为线性的，故选用SVC模型进行训练。

## 2.3 建立基于支持向量机的分类模型，参数使用缺省值，对测试集进行分类

### 2.3.1 训练数据集并预测

svm=SVC().fit(x\_tain,class\_train)

Data\_target\_pred=svm.predict(x\_test)

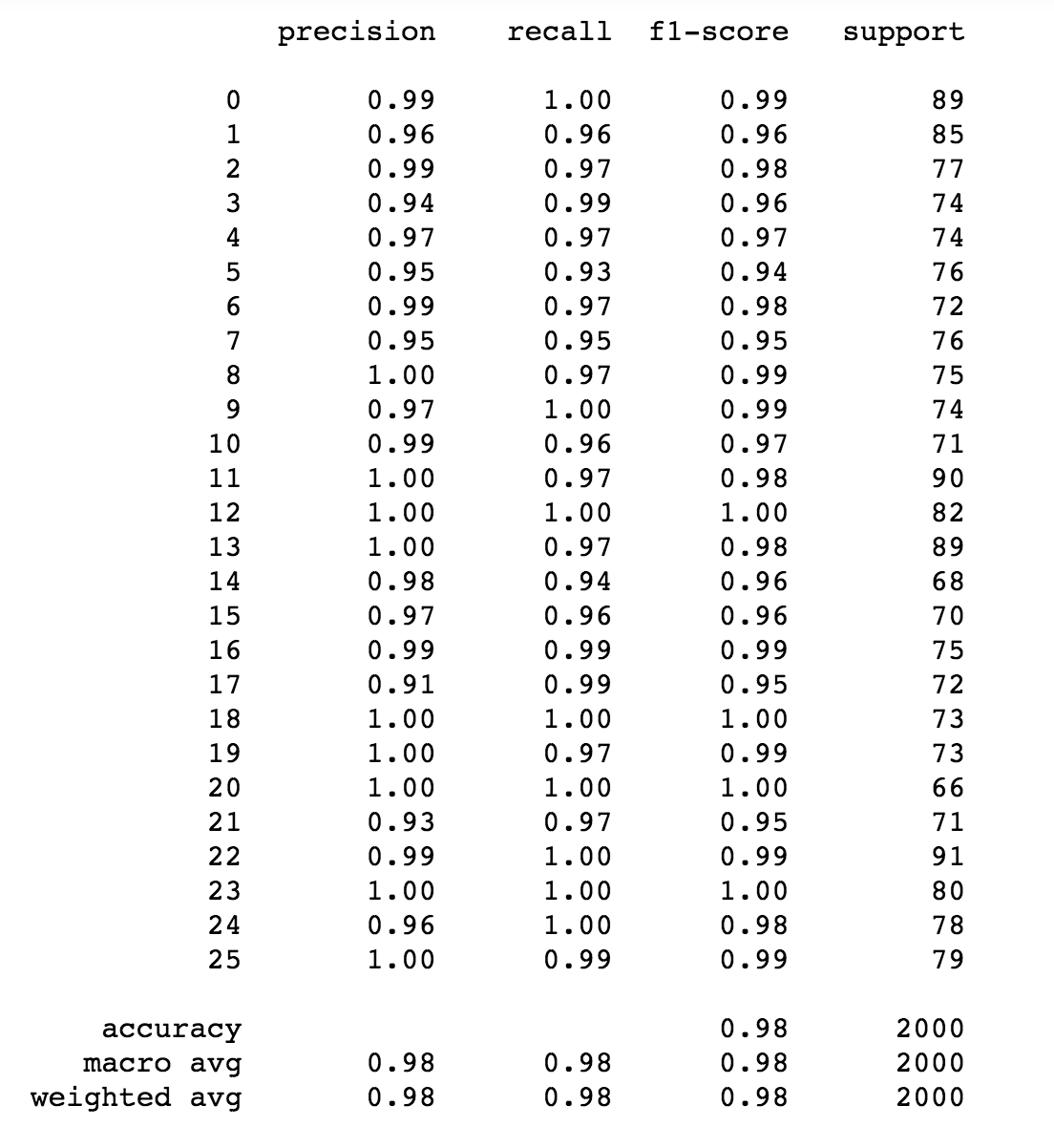
### 2.3.2 采用accuracy\_score方法评价模型准确率

print('准确率为：',accuracy\_score(class\_test,Data\_target\_pred))



### 2.3.2 采用classification\_report评价模型

print(metrics.classification\_report(class\_test,Data\_target\_pred))



### 2.3.3 SVM小结

此数据集时间比较早，数据集较为规范，使用支持向量机模型有较高的准确率，分类效果好，同时也可以说明此数据集为线性数据集，难度不大。

# 分类结果的分析

使用matplotlib绘制ROC曲线

from sklearn.metrics import roc\_curve

fpr,tpr,thresholds= roc\_curve(class\_test,Data\_target\_pred,pos\_label=2)

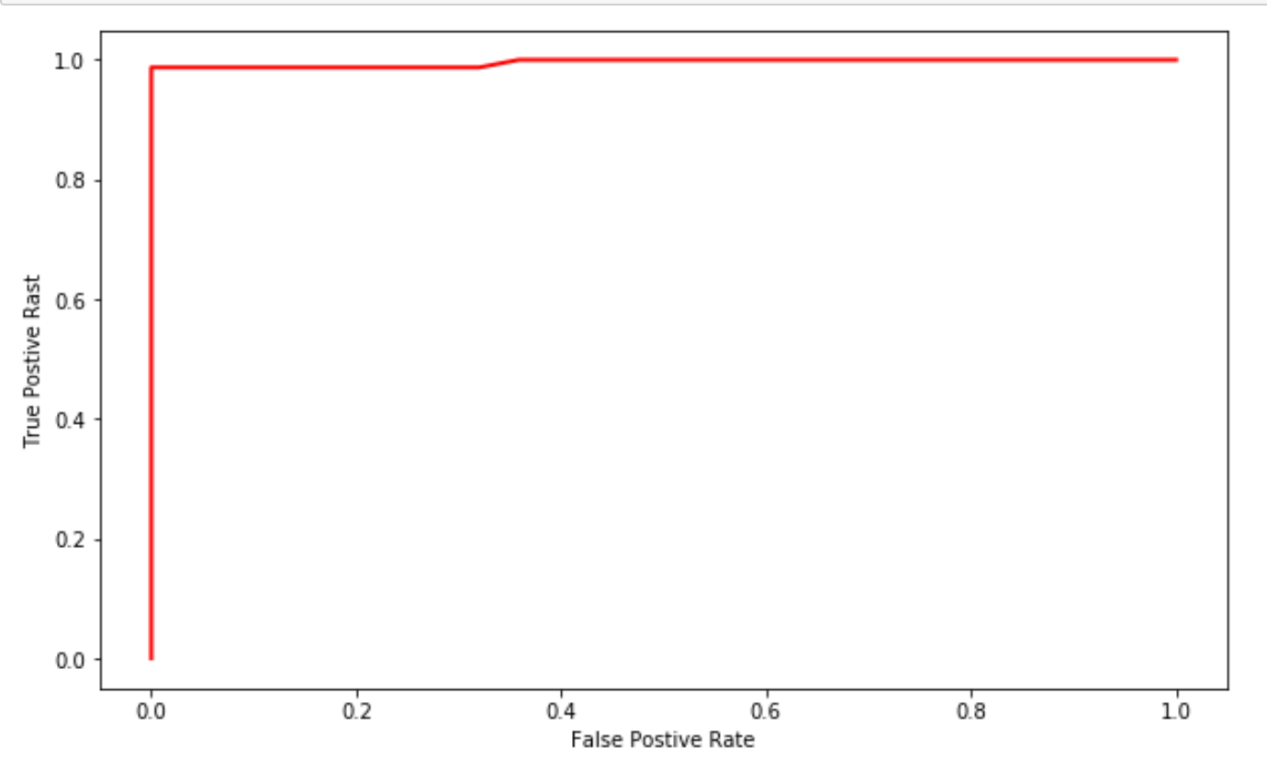
plt.figure(figsize=(10,6))

plt.xlabel('False Postive Rate')

plt.ylabel('True Postive Rast')

plt.plot(fpr,tpr,linewidth=2,linestyle="-",color='r')

plt.show()



# 总结与展望

在此数据集中使用了多种模型进行分类，分类效果都较好，其中支持向量机模型优于其他模型，所以最终选用支持向量机模型进行分类学习，最终有着98%的准确率。

但仍然存在以下问题，后续将继续完善和改进：

1. 对数据集中属性了解不够透彻，打算翻译数据集下面作者的论文
2. 有个别数字预测的准确度微低

# 个人感受和反思

这是我做的第二个数据集了，同第一个数据集相比，这个数据集比较简单，做一个调包侠也可以有很高的准确率，但这意义不大。所以我在此数据集中使用了交叉验证，通过实验，更加深刻的理解了这些验证方法的意义。

同时，如果可能的话在后期会继续研究这个数据集，挖掘出更加有用的信息，规则，通过实验提高自己的能力。

# 附录

## 6.1 完整代码及运行结果