**测试数据集：**

测试所用数据集采用的是scikit-learn 官方例子中采用的脸部数据集Olivetti，其采用函数sklearn.datasets.fetch\_olivetti\_faces(data\_home=None, shuffle=False, random\_state=0, download\_if\_missing=True)获得。

其官方介绍为数据集包含400张照片，每张照片为64X64数据矩阵，将64X64矩阵存储为1行数据即4096，因此总的数据大小为400X4096.target为400个标签，总共有40个人的照片，每个人有10张照片。

我们选取的数据集维度为（400，4096），内置400张人脸图片，其中每张图片将其延展特征数为4096，维度相当之大，符合我们的降维需求。

我们使用工具将该数据进行了可视化，由于没有指定色彩，这里是绿色的彩底：

图片包含 表格

描述已自动生成

**对人脸数据降纬度的目的：**

（1）在原始的高维空间中，包含冗余信息和噪声信息，会在实际应用中引入误差，影响准确率；而降维可以提取数据内部的本质结构，减少冗余信息和噪声信息造成的误差，提高应用中的精度。

（2）同时，如果特征维度特别高，模型训练的时间过大，导致耗费资源，如果降低一些维度，这样可以在不降低过高的精度的前提下获得更高的性能。

**系统结构：**

本系统一共包括5个函数，分别为main、plot、Model\_Selection、 Prove、K-Fold。

其中，main方法是系统的主方法，其主要功能是对用PCA降维方法对人脸数据降维处理进行演示，实现了PCA降维数据算法的全过程。依次demo便于我们直观的探究影响其结果的因素。

系统内置的plot函数是利用matplotlib库对数据集进行可视化处理。

本文设计的系统设置了三个方法来探究评估三种因素对降维精度的影响，进而对主框架进行优化，三个方法分别为：

Model\_Selection 、Prove、K-Fold

其中model\_selection中选取了LinearModel、DecisionTree、SVM、RandomForest、DNN模型用于评估数据集在这几个模型上的表现，从而通过数据分析找出表现最优的模型。

Prove方法使用Model\_selection中选取出的最优模型进行降维，验证评估将数据降维到不同特征数的得分情况，用于查找最优的降维特征数。

K-Fold方法利用K折验证，探究将数据分别进行4、5、6、7、8折验证，针对上述定义指标评估模型的稳定性，对模型进行优化。

Data文件夹中存放的是olivetti\_py3.pkz原始数据集以及系统简要的说明文档。

**图示

描述已自动生成**

**系统实现流程：**

（1）调用plot方法对原始数据集进行可视化；

（2）Main方法对PCA降维方法进行展示（选取支持向量机的分类模型对数据降维至50维）；

（3）Model\_selection方法比较LinearModel、DecisionTree、SVM、RandomForest、DNN5种模型对数据集的表现结果，选取出最优模型；

（4）Prove方法选取上一步挑选出的最优模型，验证评估将数据降维到不同特征数的得分情况，并查找最优的降维特征数。

（5）K-Fold方法利用交叉验证的方法选取出最优K折，对模型进行进一步优化。

**模型影响：**

为了验证降维后的数据集在不同机器学习模型上的表现，本文设置了model\_selection

函数来对数据集进行性能评估，我们选取了LinearModel、DecisionTree、SVM、RandomForest、DNN这几种模型。本文自定义的函数实现计算分类问题常用的评估指标分别是精确率、特异性、召回率。

Model\_selection方法运行结果如图4.1，该方法的结果展示了五种方法的评估指标，本文制表对其进行比较：

|  |  |  |  |
| --- | --- | --- | --- |
| 评估指标 | Accuracy | Specificity | Recall |
| LinearModel | 1.0 | 1.0 | 1.0 |
| DecisionTree | 0.66666666666 | 0.4 | 1.0 |
| SVM | 1.0 | 1.0 | 1.0 |
| RandomForest | 0.88888888888 | 0.8 | 1.0 |
| DNN | 0.66666666666 | 0.4 | 1.0 |

表格

描述已自动生成

模型评估结果 模型评估结果比较

比较后发现，支持向量机在三个指标上所表现出的性能要明显优于其他训练模型，因此选择支持向量机作为PCA降维的训练模型是比较优秀的选择。

**纬度影响：**

Prove方法将数据集从4096降到低纬度并通过图像可视化出结果。

降维精度得分部分展示如图4.2所示，可以发现模型在训练集上的精度有了一定的提升，这说明我们的训练数据中存在一定噪音，会导致模型的过度拟合，当我们使用PCA降维后，这些噪音数据会被过滤掉，这时就会使训练集和测试集的分布一致，进而提高模型的精度，减小误差。

手机屏幕截图

中度可信度描述已自动生成

本文将数据进行PCA降维的结果通过工具展现，下图4.3的横坐标代表数据进行PCA之后的维度，纵坐标是模型在测试集上的精度，这里我们可以发现数据在维度小于50的时候，随着特征数的增多，模型的精度越来越高，这说明维度越低，数据囊括的信息越少，也就导致模型的训练不足。

但是发现当数据特征在100维左右时，模型的精度随着维度的增加没有大幅度的提升，说明我们将数据降维到100维度表现，模型性能提高了，而且精度表现良好。

当数据达到270维左右，模型的精度有所下降，这就说明此时数据间出现了噪音，导致模型在训练期间同时将噪音也进行了训练，这就会导致降低模型的泛化性能。

因此本文认为50度是最优降维特征维数。

图表

描述已自动生成

数据降至各维度的精度表现

**K折影响：**

K-Fold方法将K折交叉验证用于模型调优，找到使得模型泛化性能最优的超参值。

验证数据取自训练数据，但不参与训练，这样可以相对客观的评估模型对于训练集

之外数据的匹配程度。

本文使用4，5，6，7，8折对模型进行调优，通过精确率、特异性、召回率进行评估。

|  |  |  |  |
| --- | --- | --- | --- |
| 评估指标 | Accuracy | Specificity | Recall |
| 4折 | 0.86388888888888 | 0.6375 | 1.0 |
| 5折 | 0.90777777777778 | 0.77666666666666 | 1.0 |
| 6折 | 0.92314814814814 | 0.81388888888888 | 1,0 |
| 7折 | 0.90238095238095 | 0.76904761904761 | 1.0 |
| 8折 | 0.93099747474747 | 0.82916666666666 | 1..0 |

K-Fold交叉验证

通过K-Fold方法发现，K为8时模型的泛化性能最好，最稳定，性能最优。