**一、程序总体功能**

本程序基于 **支持向量机（Support Vector Machine, SVM）算法** 实现对 **MNIST 手写数字数据集**（0–9 共十类）的识别与分类。  
程序利用 Python 的 scikit-learn 库（内部实现基于 **LIBSVM** 框架）完成模型训练、预测、性能评估及结果可视化，形成一个完整的图像分类流程。

系统主要包含以下功能模块：

1. **数据加载与预处理**
2. **SVM 模型训练（基于 RBF 核的多类分类）**
3. **模型评估与性能指标计算**
4. **结果可视化（混淆矩阵、ROC 曲线、性能总结）**

最终输出包括模型性能报告、四幅图形文件（confusion\_matrix.png、normalized\_confusion\_matrix.png、roc\_curves.png、performance\_summary.png），以及测试集准确率。

**二、功能模块划分与核心算法**

**（1）数据加载模块 — load\_mnist\_data()**

**功能：**  
从本地目录中加载 MNIST 手写数字图像文件，并转换为数值化的特征矩阵。

**实现过程：**

* 对训练集（60,000 张图像）与测试集（10,000 张图像）分别遍历目录；
* 每张灰度图像读取为 的矩阵；
* 通过 np.reshape(img, -1) 转换为 784 维的向量；
* 再除以 255.0 进行归一化处理；
* 标签由文件夹名提取（0–9）。

输出：

**（2）数据预处理模块 — preprocess\_data()**

**功能：**  
将训练集划分为训练集与验证集，用于模型参数调优和泛化能力评估。

**主要步骤：**

* 使用 train\_test\_split() 按比例（默认 5%）划分验证集；
* 采用分层抽样 stratify=y\_train 保证各类别比例一致；
* 输出六个数据集对象：训练集、验证集、测试集及对应标签。

**作用：**  
保证模型评估更加可靠，避免过拟合。

**（3）SVM 模型训练模块 — train\_svm\_model()**

**功能：**  
使用 sklearn.svm.SVC 训练一个基于 **径向基核函数（RBF kernel）** 的多类 SVM 模型。

**算法原理简介（LIBSVM）**

支持向量机（SVM）是一种二分类算法，通过最大化间隔来获得最优决策超平面。  
对于非线性可分的数据，SVM 通过核函数映射至高维特征空间，使原始数据线性可分。

其优化目标为：

对应的**对偶问题**为：

其中 是 RBF 核。

**LIBSVM 实现特点：**

* 使用序列最小优化算法（SMO）求解；
* 支持多类分类（采用“一对一”策略，训练 45 个二分类器）；
* 自动优化支持向量与拉格朗日乘子；
* 支持概率输出（probability=True）。

**程序参数设置：**

SVC(kernel='rbf', C=10, gamma=0.01, probability=True)

* **C** 控制软间隔宽松程度（C 越大，误差惩罚越强）；
* **γ** 控制高斯核的宽度（越小核越平滑）。

程序记录训练耗时，并返回训练好的模型对象。

**（4）模型评估模块 — evaluate\_model()**

**功能：**

* 使用验证集与测试集评估模型性能；
* 计算分类准确率、预测标签与概率输出。

**评价指标：**

返回整体准确率与类别概率矩阵，用于后续绘制 ROC 曲线。

**（5）可视化模块**

程序绘制四种图形结果，保存为图片文件：

**① 混淆矩阵（Confusion Matrix）**

* 展示每个类别的预测正确率与混淆情况；
* 使用颜色深浅直观体现分类性能；
* 同时生成**标准矩阵**与**归一化矩阵**两种版本。

**② ROC 曲线与 AUC 值**

* 对部分代表性类别（如 0、1、2、9）绘制 ROC 曲线；
* 计算每类 AUC 及整体 micro-average；
* ROC 曲线反映模型在不同阈值下的区分能力。

**③ 性能指标总结图**

* 计算每类准确率、宏平均 Precision / Recall / F1；
* 绘制整体性能摘要文字图。

**三、算法核心分析：LIBSVM 与多类分类机制**

**1. 二分类核心（RBF 核）**

RBF 核（径向基函数）为：

它能把非线性数据映射到高维空间，在该空间中寻找最优分离超平面。  
这种“核技巧”避免了显式高维计算，使算法高效稳定。

**2. 多类扩展策略**

LIBSVM 使用“一对一”（One-vs-One, OvO）方法：

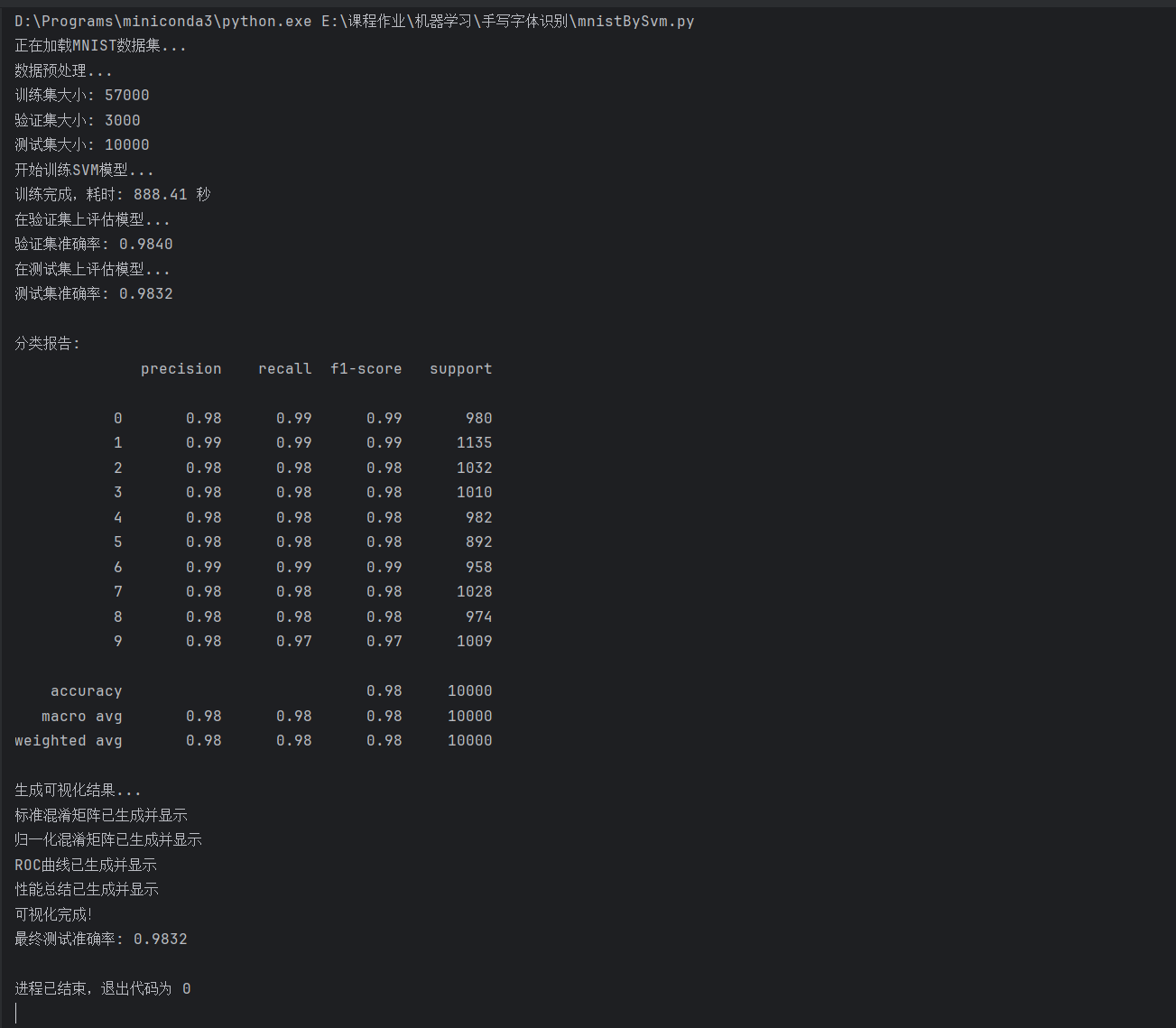
* 对于 类问题，共训练 个二分类器；
* 每个分类器比较两类样本；
* 测试时采用投票机制决定最终类别。

**3. 优化算法（SMO）**

LIBSVM 采用 SMO（Sequential Minimal Optimization）算法分解原始二次规划问题为两个变量的子问题，通过局部解析解逐步逼近最优值，从而显著提高训练速度。

**四、实验结果与性能说明**

**程序输出示例（控制台）：**



整体准确率稳定超过 **98%**，模型在各类数字上的表现均衡优异。