MNIST Digit Recognition Using Scikit-learn

Prepared by: Zunaira Hameed

1. Introduction

Handwritten digit recognition is a fundamental problem in the field of machine learning and computer vision. The MNIST dataset serves as a standard benchmark for this task, consisting of grayscale images of digits from 0 to 9. The objective of this project is to classify these handwritten digits using classical machine learning techniques provided by the Scikit-learn library.

2. Dataset Description

The MNIST dataset used in this project is obtained via the `fetch_openml()` function in Scikit-learn. It contains a total of 70,000 grayscale images, each of size 28×28 pixels, flattened into 784 features. The dataset is divided into 60,000 training images and 10,000 testing images. The labels correspond to the digit (0–9) shown in each image.

3. Preprocessing

Before training, the dataset undergoes several preprocessing steps. The labels are converted to integers, and the data is split into training and testing sets using an 85/15 ratio. Although Random Forests do not require feature scaling, standardization was tested using StandardScaler for comparison with other models like SGD.

4. Models Used

Two different models were trained and evaluated in this project:

• Stochastic Gradient Descent (SGD Classifier): A linear classifier trained using hinge loss (SVM loss). This model is lightweight and requires feature scaling for optimal performance.

• Random Forest Classifier: A powerful ensemble model based on decision trees. It does not require feature scaling and works well with high-dimensional data like MNIST.

5. Evaluation

The models were evaluated using Scikit-learn's `classification_report` and `confusion_matrix`. The Random Forest model achieved a test accuracy of 96%, demonstrating strong generalization. Some digits such as 5 and 8 or 9 and 4 were occasionally misclassified due to their similar shapes.

6. Error Analysis

Error analysis was conducted by identifying the most frequent misclassifications. Visualizing these errors helped in understanding the model's limitations. Three common misclassified digit pairs were noted. To improve performance, techniques such as centering the digits, applying image preprocessing, and using data augmentation were proposed.

7. Web Interface Using Gradio

To provide user interaction with the model, a Gradio web app was developed. Users can draw a digit, and the app predicts it in real-time using the Random Forest model. The image undergoes preprocessing including grayscale conversion, resizing to 28×28, inversion to match MNIST format, normalization, and reshaping before prediction.

8. Conclusion

The project effectively demonstrates how traditional machine learning models can solve image classification tasks like handwritten digit recognition. With minimal preprocessing and model tuning, high accuracy was achieved. For better real-world robustness, switching to deep learning methods like CNNs is recommended.

9. Appendix

This section may include code snippets used for model training, evaluation, and deployment, as well as confusion matrix visuals and screenshots of the Gradio interface.