Final Report: SVM for Handwritten Digit Classification

1. Introduction

The objective of this project was to develop a machine learning model using Support Vector Machines (SVM) to classify handwritten digits. The dataset used was the digits dataset from scikit-learn, which contains 8x8 grayscale images of digits ranging from 0 to 9. This report outlines the workflow, insights from misclassification cases, and conclusions about the model's performance.

---

2. Workflow Overview

Step 1: Data Loading and Exploration

- The digits dataset was loaded using scikit-learn's load_digits() function.
- Key statistics and class distributions were examined to understand the dataset:
  - **Number of samples:** 1797
  - **Number of features:** 64 (pixel intensities of 8x8 images)
  - **Number of classes:** 10 (digits 0-9)
- Visualization of a few digit samples provided an intuitive understanding of the dataset.

Step 2: Train-Test Split

- The dataset was split into 80% training and 20% testing using train_test_split with stratification to ensure equal class representation.
- Original indexes were tracked for later analysis of misclassified samples.

Step 3: Data Preprocessing

- The pixel intensity values were scaled using Standard Scalar to improve the convergence of the SVM algorithm.

Step 4: Model Training and Tuning

- An SVM classifier was implemented using scikit-learn's SVC class.
- Hyperparameter tuning was performed using GridSearchCV with 5-fold cross-validation to optimize the following parameters:

  - **Kernel:** Linear, Polynomial, RBF
  - **C (Regularization parameter):** [0.1, 1, 10]
  - **Gamma (for RBF kernel):** ['scale', 'auto']
- The best parameters were selected based on cross-validation scores is {'C': 10, 'degree': 3, 'gamma': 'scale', 'kernel': 'poly'}.

Step 5: Model Evaluation

- The best model from GridSearchCV was evaluated on the test set.
- Metrics used:
  - Accuracy
  - Precision
  - Recall
  - F1-score

Step 6: Misclassification Analysis

- Misclassified samples were identified by comparing the predicted labels with the true labels.
- The original indexes of misclassified rows were mapped back for further analysis.

---

3. Results and Insights

Model Performance

The performance of the best-tuned SVM model is summarized below:

Metric	Score
Accuracy	99%
Precision	99%
Recall	99%
F1-score	99%

Misclassification Analysis

- **Number of Misclassified Samples:** 2 out of 360 test samples.
- **Insights from Misclassified Samples:**
  - The two misclassifications occurred between digits with similar visual structures (6-8) and (7-9)).
  - The misclassifications could be attributed to noise or incomplete digit patterns in the dataset.

Visualizations

- Confusion Matrix: Highlighted the distribution of correct and incorrect predictions.
- Sample Misclassifications: Visualized a few misclassified digits alongside their true and predicted labels for qualitative analysis.

---

4. Conclusions and Recommendations

Conclusions

- The SVM model demonstrated excellent performance on the handwritten digit classification task, achieving over 99% accuracy.
- Misclassification analysis revealed that most errors were due to the inherent similarity between certain digits or noise in the dataset.

Recommendations

1. **Data Augmentation:** Introduce transformations such as rotation, scaling, or adding noise to make the model more robust.
2. **Advanced Models:** Experiment with other classification techniques, such as Convolutional Neural Networks (CNNs), for potentially higher accuracy.
3. **Error Focus:** Use a focused dataset with more samples of commonly confused digits to address specific misclassifications.