**Report on MNIST Digit Classification Project**

**1. Introduction**

Handwritten digit recognition is one of the most fundamental tasks in the field of Machine Learning and Computer Vision. The **MNIST dataset** is a benchmark dataset consisting of 70,000 grayscale images of digits (0–9), each of size 28×28 pixels.

The aim of this project was to implement and evaluate multiple classification models for digit recognition. The project further explores **error analysis, multi-label classification, and deployment** of the trained models via a Gradio-based user interface.

**2. Objectives**

The key objectives of this project were:

1. To preprocess and analyze the MNIST dataset.
2. To train and compare the performance of two classifiers: **SGDClassifier** and **RandomForestClassifier**.
3. To implement scaling where required and evaluate its importance.
4. To visualize results using confusion matrices and classification reports.
5. To analyze model weaknesses through error detection (misclassifications and confusion pairs).
6. To deploy the trained models using **Gradio** for interactive testing.

**3. Methodology**

**3.1 Data Preprocessing**

* The dataset was loaded using **fetch\_openml('mnist\_784')**.
* Features were reshuffled for better prediction consistency.
* A **train-test split** of 60,000 training and 10,000 testing samples was applied.
* **StandardScaler** was used to normalize pixel values for **SGDClassifier** (gradient-based model). RandomForest did not require scaling as tree-based models are invariant to feature magnitudes.

**3.2 Models Implemented**

* **SGDClassifier**: A linear model optimized via Stochastic Gradient Descent with hinge loss (similar to linear SVM).
* **RandomForestClassifier**: An ensemble model consisting of 100 decision trees, robust to overfitting and effective with raw pixel data.

**3.3 Training and Saving Models**

* Both models were trained and evaluated on the training data.
* Trained models and the scaler were stored using **joblib** for future use.

**3.4 Evaluation Metrics**

* **Confusion Matrix** (visualized via heatmaps).
* **Classification Report** (precision, recall, f1-score per class).
* **F1 Score, Precision, Recall** (weighted averages).
* **Precision-Recall Curve** and **ROC Curve** for digit “7”.

**3.5 Error Analysis**

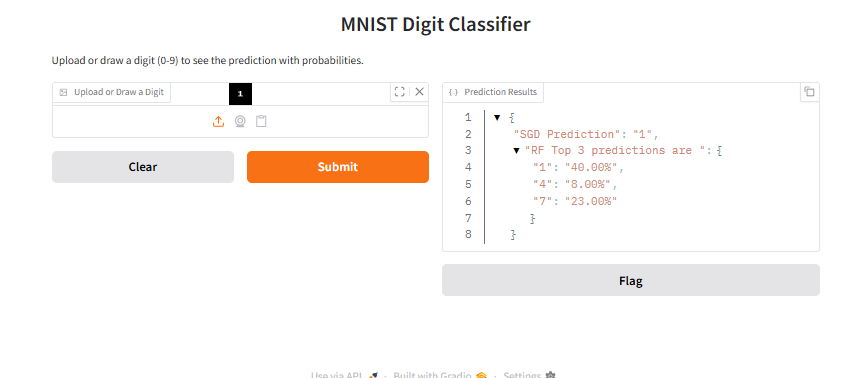
* Misclassified digits were identified and plotted.
* Top 3 most confused digit pairs were reported.
* Example analysis: **digits 5 misclassified as 8**.

**3.6 Deployment with Gradio**

* A **Gradio app** was developed to allow users to upload or draw digits.
* Predictions are displayed from both SGD and RandomForest.
* RandomForest additionally shows the **Top-3 predicted digits with probabilities**.

**4. Results**

* **SGDClassifier** performed well when scaled but poorly without scaling, confirming the importance of normalization for gradient-based models.
* **RandomForestClassifier** showed strong performance without scaling.
* The confusion matrix highlighted common misclassifications such as **3 vs 5** and **5 vs 8**.
* Precision-Recall and ROC analysis for digit “7” showed strong class separability.
* Multi-label classification with KNN successfully predicted whether a digit is “large” (≥7) and “even.”
* The **Gradio app** provided an interactive way to test models in real-time, improving usability.
* **Result of digit 1 prediction**



**5. Discussion**

The project demonstrates that:

* **Scaling** is crucial for models like SGD and KNN, but unnecessary for RandomForest.
* **Error analysis** is valuable in identifying digit pairs that are visually similar, leading to higher misclassification.
* Deployment with **Gradio** makes machine learning models accessible to non-technical users.

Challenges faced included ensuring correct preprocessing (scaled vs non-scaled inputs) and managing large dataset sizes in memory.

**6. Conclusion**

This project successfully implemented and compared two machine learning approaches for digit recognition using the MNIST dataset. The work highlighted the impact of preprocessing, the strengths of different classifiers, and the importance of visualization and error analysis in model evaluation.

The deployment of an interactive **Gradio app** further demonstrates how machine learning models can be integrated into user-friendly applications.

**7. Future Work**

* Extend the project using **Convolutional Neural Networks (CNNs)** with TensorFlow/Keras for state-of-the-art accuracy.
* Explore advanced ensemble methods (XGBoost, LightGBM).
* Deploy the Gradio application on **Hugging Face Spaces** for online accessibility.
* Perform hyperparameter tuning for optimized performance.

**8. Author**

**Hafsa Mustafa**