DA2401 - Machine Learning Lab

End Semester - MNIST digit classification

28 Oct 2025

We learnt implementation of various algorithms in class, covering methods like Linear Regression, Classification, Clustering and Anomaly Detection. We also learnt ensemble methods like Bagging & Boosting and common training problems like overfitting & the bias - variance trade off.

MNIST database contains images of handwritten digits resized to 28 X 28 pixels, and converted to gray scale, with each pixel ranging from 0 to 255. For each image, this results in 784 numbers (between 0 to 255) and a class label (0 to 9). Ref: `https://www.kaggle.com/datasets/oddrationale/mnist-in-csv`. We will use the same datasets that have been prepared for Assignment 2 - **MNIST_train.csv** with 10,002 entries, **MNIST_validation.csv** with 2499 entries and **MNIST_test.csv** with 2517 entries. Of these, **MNIST_train.csv** and **MNIST_validation.csv** are shared with you for training and testing your models. *DA2401/Assignments/Assignment2-Classification* contains these files. **MNIST_test.csv** is not shared with you, and will be used to assess your model performance.

In this assignment, you will build a multi class classification system to predict the class label (0 to 9). Complete the assignment using plain Python with simple packages like NumPy and SciPy, without using any built in ML packages like sklearn or xgb. Share Python file (.py format), that can be executed and tested.

Here are the set of guidelines for you to consider.

- Use algorithms from the four methods discussed in class - Linear Regression, Classification, Clustering & Anomaly Detection. Exclude methods like Neural Networks that are not covered yet.

- You can use multiple algorithms, try bagging, boosting, stacked models or any other approaches. A single model may give results, but ensemble models are known to improve accuracy. You will have to try ensemble approaches for this exercise.
- All algorithms used must be coded in plain Python with basic packages like Numpy, Scipy. Use of sklearn or xgb or any other built in ML package is not allowed.
- For every algorithm that you use, tune any hyper parameters necessary, evaluate bias - variance trade off, present results and ensure that training is optimal.
- Use this system to predict label of the number (from 0 to 9), using MNIST_train dataset. Use MNIST_validation dataset to evaluate your system performance. Since this is a reasonably balanced dataset, **f1 score** is the metric to maximize. Tabulate the results.
- Keep the training time to less than 5 minutes for the entire system, discuss your steps to achieve this.
- Evaluation Task - TAs will run your system to train on MNIST_train dataset & test on MNIST_test dataset, and assess the training time and accuracy score.

Submissions should contain the following files

1. A Python file (algorithms.py) file with implementation of all algorithms used in your system.
2. Main Python file (main.py) to train the models on the data, and predict the results.
3. PDF report with the following topics -
   (a) A summary of models used & system architecture.

(b) Summary of hyper-parameter tuning and results for each individual model.

(c) Steps taken to optimize system performance and limit run time. Evaluation results on training and validation datasets.

(d) Detailed summary of your thoughts and observations from this exercise.