CS412: Machine Learning Homework 1

Deadline: October 18, 2025

Submission Guidelines

- **Jupyter Notebook:** Must include all code cells and outputs. (The notebook will *not* be re-run during grading.)
- **PDF Report:** Provide a clear and detailed summary of methodology, experiments, analysis, and conclusions.
- **File Naming:**
  - `CS412-HW1-YourName.ipynb`
  - `CS412-HW1-YourName.pdf`
- **Late Policy:** -10 points per day, up to 3 days.

1 Overview

In this assignment, you will perform a deep dive into *k-Nearest Neighbors (k-NN)* using the **Fashion-MNIST** dataset, which is more challenging than digit recognition due to class similarity (e.g., Shirt vs. T-shirt/Top). The goals are to explore train-val-test split, hyperparameter tuning, distance metrics, error analysis, and visualization techniques to understand how k-NN behaves on complex image data.

2 Dataset and Preprocessing

Fashion-MNIST consists of 28×28 grayscale images across 10 clothing categories. Pixel values range from 0 to 255.

Figure 1: Fashion-MNIST Dataset

2.1 Data Loading

1. To download the Fashion-MNIST dataset, you will use the Keras library. The dataset comprises 60,000 training samples and 10,000 test samples. You will split the training data into two sets: a development set for training your models and a validation set for testing the performance of your models during development.

2. Split the dataset:

   - **Train-Val:** Reserve 20% of the training data for validation and use the remaining 80% for training your models. Before splitting the data, ensure that it is shuffled to maintain the representativeness of both the training and validation sets. Use a fixed random seed (random_state=42) to ensure that dataset splits are fully reproducible across runs. Make sure that your datasets are balanced, use stratify.
   - **Test:** Use the provided test set (10,000 samples) without modification.

3. Print the shapes of the training, validation, and test sets to verify the split.

2.2 Data Analysis

Perform the following exploratory analysis prior to preprocessing:

1. **Class Distribution:** Plot a bar chart showing the number of samples per class.

2. **Pixel Statistics:** Report the global mean and standard deviation of pixel values, and the per-class mean pixel intensity. Interpret these findings briefly in your report.

3. **Visualization:** Display at least one sample image for each class.

2.3 Preprocessing

1. Normalize pixel values using `sklearn.preprocessing.StandardScaler` and compare the before after mean/std values.

2. Reshape (flatten) data from 3D to 2D for Scikit-learn.

3 k-NN Classifier

3.1 Hyperparameter Tuning

1. Tune the number of neighbors $k \in \{1, 3, 5, 7\}$.

2. Compare distance metrics: **Euclidean** and **Manhattan**.

3. Evaluate each configuration on the validation set and record validation accuracy. Note that to find the best parameters you need to use the combinations of the parameters eg. (k=3, metric="euclidean")

4. Plot validation accuracy versus k for each distance metric on the same figure. Clearly label axes and provide a legend.

3.2 Final Model

1. Using the best hyperparameters, retrain k-NN on the concatenated training and validation sets.
2. Evaluate on the test set and report:
    - Overall **Accuracy**
    - **Precision**, **Recall**, and **F1-score** (macro-averaged)
    - **Confusion matrix** (include the figure and discuss which classes are better or worse classified)
3. Observe the training and prediction times for the k-NN model, to be able to answer characteristics of a k-NN model. (HINT: Think about why k-NN is a "lazy learner".)

4 Error Analysis

1. Identify the top 3 most confused class pairs (e.g., Pullover vs. Coat).
2. For each pair, display 5 random misclassified examples (include predicted vs. true labels).
3. Provide a short discussion on why these confusions occur (e.g., visual similarity, texture, silhouette).

Final Report Checklist

Your PDF report should include:

- A clear overview of methodology, including data analysis and preprocessing.
- Justification for hyperparameter and distance metric choices, the validation curve figure.
- Comprehensive test results (accuracy, macro precision/recall/F1) and a confusion matrix figure with discussion.
- Error analysis with the top confused pairs and example visuals.
- Computational analysis with timing results and discussion of trade-offs.
- The shareable link to your Jupyter Notebook at the **top** of the document.