CS412: Machine Learning Homework 1
Fashion-MNIST k-NN Classifier Analysis

Kerem Tufan 32554

October 18, 2025

Jupyter Notebook Link

CS412-HW1-KeremTufan.ipynb on Google Colab

1 Overview

This study presents a detailed analysis of k-Nearest Neighbors (k-NN) on the Fashion-MNIST dataset. The dataset consists of 28×28 grayscale images across 10 clothing categories. The objectives of this analysis are:

- Exploration of training, validation, and test splits.
- Hyperparameter tuning (k and distance metric).
- Evaluation of performance using accuracy, precision, recall, F1-score, and confusion matrix.
- Error analysis on misclassified samples.

2 Dataset and Preprocessing

2.1 Data Loading and Splitting

The dataset contains 60,000 training and 10,000 test samples. An 80% training and 20% validation split was performed using stratified sampling (random seed=42). Shapes and labels:

- Training data shape: (48000, 28, 28), Labels: (48000,)
- Validation data shape: (12000, 28, 28), Labels: (12000,)
- Test data shape: (10000, 28, 28), Labels: (10000,)

2.2 Sample Visualization

Figure 1 shows 10 sample images from the training set to illustrate the dataset.

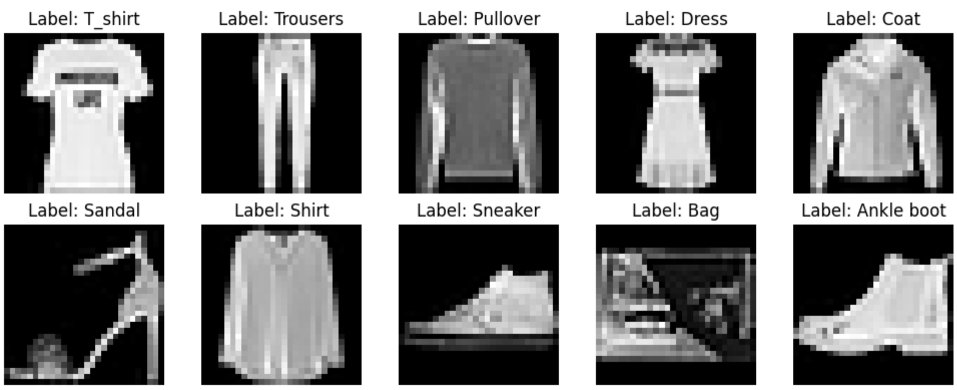

Figure 1: Ten sample images from the training set.

2.3 Data Analysis

Class Distribution: Figure 2 shows the number of samples per class. The dataset is balanced.

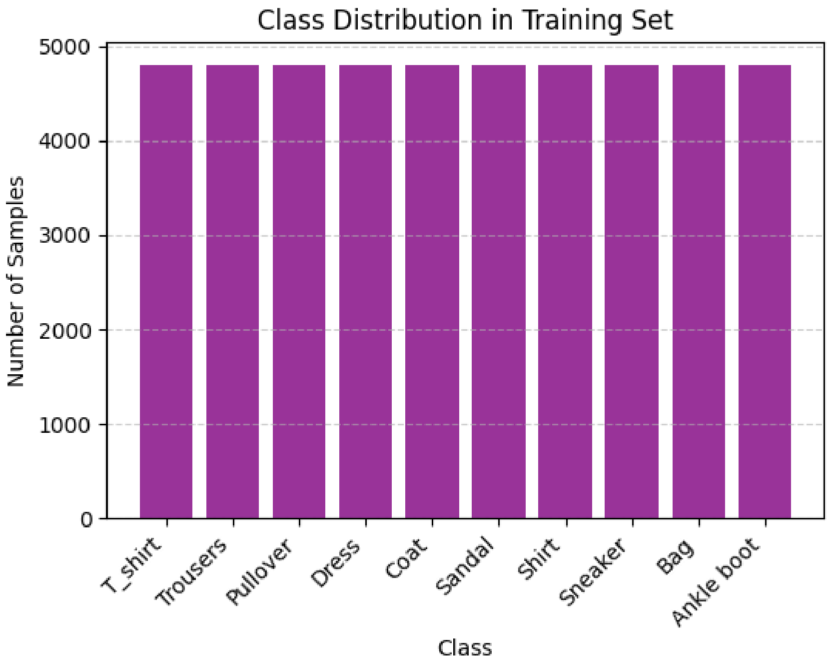

Figure 2: Class distribution in the training set.

Pixel Statistics: The global mean is 72.99 with a standard deviation of 90.06. Per-class mean pixel intensity is reported in Table 1.

Class	Mean Pixel Intensity
T_shirt	82.85
Trousers	56.99
Pullover	96.49
Dress	66.12
Coat	98.10
Sandal	34.80
Shirt	84.83
Sneaker	42.84
Bag	90.17
Ankle boot	76.76

Table 1: Per-class mean pixel intensity.

High-intensity classes (Coat, Pullover, Bag) occupy more pixels, while Sandal and Sneaker occupy fewer pixels. This difference may affect classification difficulty.

2.4 Preprocessing

- Images were flattened from 28×28 to 784-dimensional vectors.

- Pixel values were normalized using `StandardScaler`:
  - Before scaling: mean=72.99, std=90.06
  - After scaling: mean=0.00, std=1.00

3 k-NN Classifier

3.1 Hyperparameter Tuning

Hyperparameter tuning was conducted for $k \in \{1, 3, 5, 7\}$ and distance metrics *euclidean* and *manhattan*. Validation results are summarized in Table 2.

k	Metric	Validation Accuracy	Time (s)
1	Euclidean	0.8521	37.18
3	Euclidean	0.8553	37.33
5	Euclidean	0.8548	37.33
7	Euclidean	0.8562	37.17
1	Manhattan	0.8570	698.95
3	Manhattan	0.8631	721.77
5	Manhattan	0.8661	720.89
7	Manhattan	0.8643	713.18

Table 2: Validation accuracy and timing for different k and distance metrics.

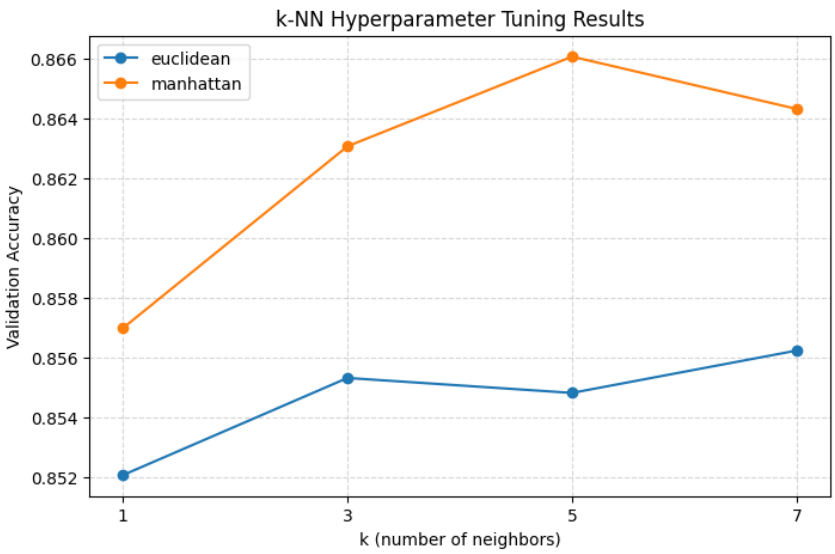

Figure 3: Validation accuracy vs k for Euclidean and Manhattan distance metrics.

Observation: Manhattan distance achieved slightly higher accuracy but required significantly more computation time. The best hyperparameters were $k = 5$ with Manhattan distance.

3.2 Final Model Evaluation

The k-NN model was retrained on the combined training and validation sets and evaluated on the test set:

- Test Accuracy: 0.8613
- Macro Precision: 0.863
- Macro Recall: 0.861
- Macro F1-score: 0.861
- Training time: 0.04s
- Prediction time: 733.33s

Table 3: Per-class performance of the final k-NN model (k=5, Manhattan distance).

Class	**Precision**	**Recall**	**F1-score**	**Support**
T-shirt	0.771	0.859	0.813	1000
Trousers	0.992	0.967	0.979	1000
Pullover	0.730	0.795	0.761	1000
Dress	0.898	0.878	0.888	1000
Coat	0.768	0.767	0.767	1000
Sandal	0.990	0.902	0.944	1000
Shirt	0.656	0.577	0.614	1000
Sneaker	0.912	0.960	0.935	1000
Bag	0.980	0.938	0.959	1000
Ankle boot	0.928	0.970	0.949	1000

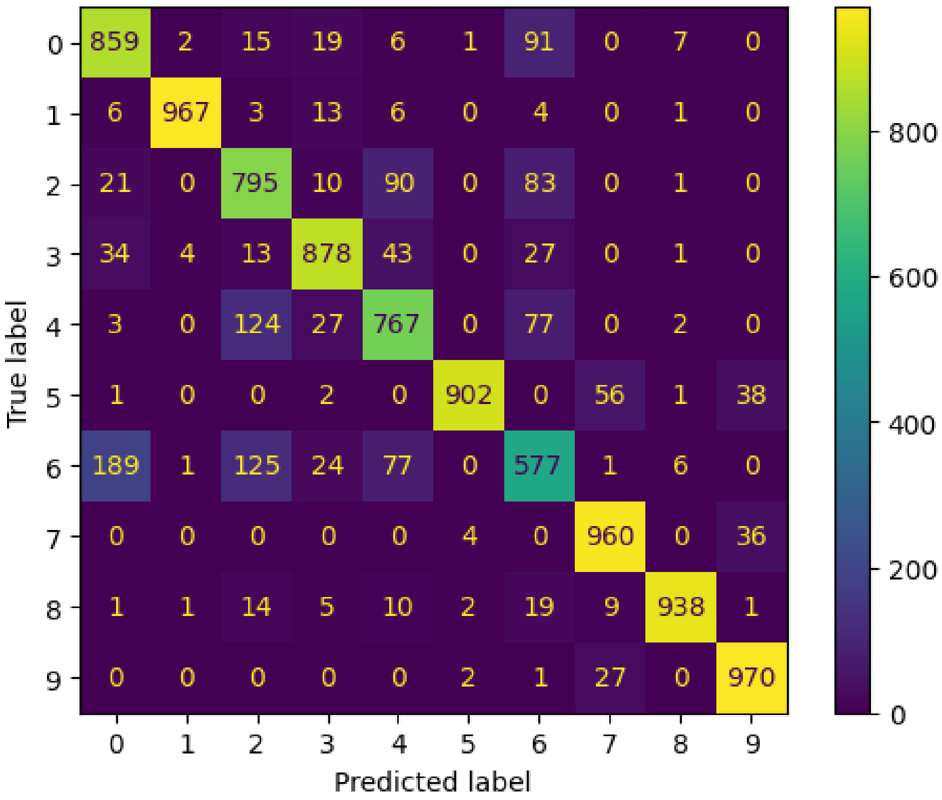

Figure 4: Confusion matrix for the test set.

Observation: From the confusion matrix, Trousers class is predicted most accurately with 967 correct predictions out of 987 samples, while Shirt has the lowest correct predictions (577/1028), indicating difficulty in distinguishing visually similar upper-body garments. Major misclassifications occur between Shirt T-shirt and Coat Pullover, consistent with pixel intensity and silhouette similarity observations.

Lazy Learner Characteristic and Computational Analysis: k-NN is a *lazy learner*, meaning it does not build an explicit model during training. Instead, it simply stores the training data, resulting in an extremely short training time (0.04s in our case). However, during prediction, the model must compute distances between each test sample and all stored training samples, leading to a very long prediction time (733.33s). This large time gap between training and prediction clearly reflects the lazy learning nature of k-NN, where the computational burden is shifted from training to inference.

Trade-off Discussion: While Manhattan distance produced slightly higher accuracy than Euclidean distance, it required nearly 20 times more computation time (700s vs. 37s). This highlights a fundamental trade-off in k-NN: using more computationally intensive distance metrics can improve accuracy marginally but may not be practical for large-scale or real-time applications.

4 Error Analysis

The top three misclassified class pairs were identified:

- Shirt T-shirt: 189 errors
- Shirt Pullover: 125 errors

- Coat Pullover: 124 errors

Example misclassified images:

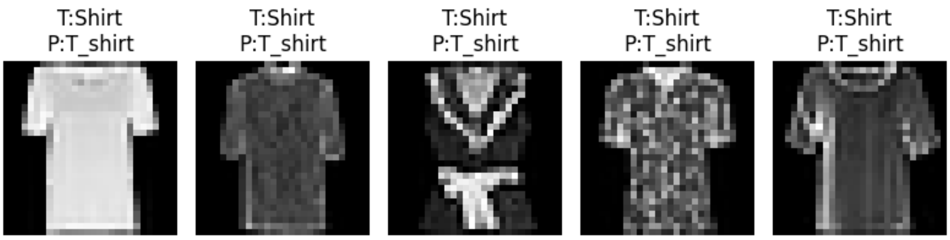

Figure 5: True: Shirt, Predicted: T-shirt

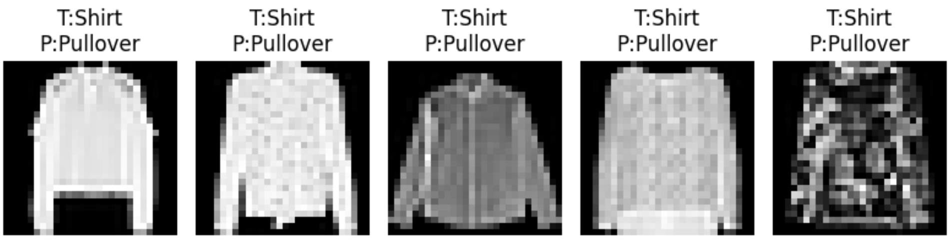

Figure 6: True: Shirt, Predicted: Pullover

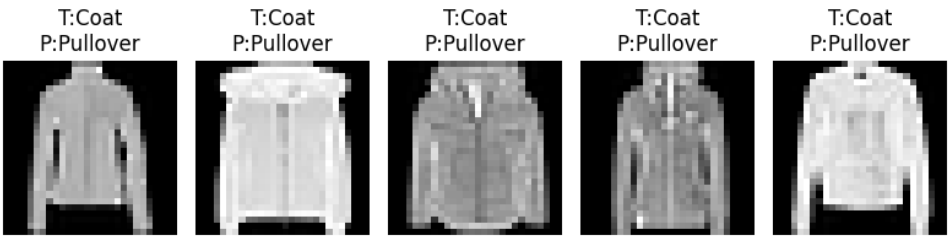

Figure 7: True: Coat, Predicted: Pullover

Discussion: Most misclassifications occurred due to visual similarity in silhouette and pixel intensity among certain classes. For example, the top misclassified pairs—Shirt T-shirt, Shirt Pullover, and Coat Pullover—illustrate this clearly. Shirt and T-shirt have very similar upper-body silhouettes and comparable pixel densities, with only minor differences in collar or sleeve details, which makes them easy to confuse. Similarly, Pullover and Coat share overlapping shapes and textures, leading to frequent misclassification. On the other hand, classes such as Bag or Ankle boot are more visually distinct, with unique contours and pixel distributions, which explains the near-perfect classification for these categories. This analysis highlights that k-NN primarily struggles when classes have subtle visual differences that are not easily captured by raw pixel vectors.

5 Conclusion

- Manhattan distance provides slightly higher accuracy at the cost of computation.

- Pixel intensity and silhouette similarity are key determinants of class-level performance.
- The k-NN algorithm is a lazy learner: training is fast while prediction is computationally intensive.
- Misclassifications are concentrated in visually similar classes.