Artificial Intelligence – Assignment 1 Report

Karanveer Gautam Sharma (2301MC10)
Prakhar Palod (2301MC17)
Prashant Nigam (2301MC18)

January 21, 2026

1 Introduction

This assignment explores the K-Nearest Neighbors (KNN) algorithm by implementing it from scratch and applying it to two contrasting problems:

1. Binary classification on structured medical data.
2. Multi-class classification on high-dimensional image data (CIFAR-10).

The objective is to analyze the performance, scalability, and limitations of KNN rather than merely achieving high accuracy.

2 Task 1 – Binary Classification (Breast Cancer Dataset)

2.1 Dataset and Preprocessing

The **Breast Cancer Wisconsin** dataset contains 569 samples with 30 continuous features extracted from FNA images.

- Non-informative columns (id, empty column) were removed.
- Labels were encoded as: **Malignant → 1**, **Benign → 0**.
- Feature standardization was applied to prevent large-scale features (e.g., area, perimeter) from dominating distance calculations.
- The data was split into 80% training and 20% testing sets.

2.2 Experimental Setup

- **K values tested:** 3, 4, 9, 20, 47
- **Distance metrics:** Euclidean, Manhattan, Minkowski, Cosine, Hamming

2.3 Results

The best performance was achieved using **K = 9** with **Euclidean distance**.

Final Test Performance

- **Accuracy:** 97.36%
- **Precision:** ≈ 0.975
- **Recall:** ≈ 0.951

Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	TP = 39	FN = 2
Actual Negative	FP = 1	TN = 72

Table 1: Confusion Matrix for best performing model (K=9, Euclidean)

Euclidean and Minkowski distances consistently outperformed others, while Hamming distance performed poorly due to its incompatibility with continuous features.

2.4 Key Inferences (Task 1)

- KNN performs very well on low-dimensional, well-structured tabular data.
- Feature scaling is essential for distance-based algorithms.
- K controls the bias–variance trade-off: very small K overfits, very large K underfits.

3 Task 2 – Multi-Class Classification (CIFAR-10)

3.1 Dataset and Representation

CIFAR-10 consists of 32×32 RGB images across 10 classes.

- Each image was flattened into a 3072-dimensional vector.
- **Training samples:** 5000
- **Test samples:** 500

3.2 Computational Reality of KNN (Corrected Calculation)

KNN computes distances at prediction time. For one full experiment with:

- Train $N = 5000$
- Test $M = 500$

- Features $D = 3072$

The distance computations required are:

$$500 \times 5000 \times 3072 = 7.68 \times 10^9 \text{ operations}$$

Including 5 distance metrics and 5 K values:

$$7.68 \times 10^9 \times 25 \approx 1.92 \times 10^{11} \text{ operations}$$

This results in $\approx$ **192** billion operations. In pure Python, this corresponds to many hours of runtime, explaining why execution became impractical.

3.3 Observed Results (Task 2)

- **Best accuracy:** $\approx 30\%$
- Strong confusion between visually similar classes (cat–dog, deer–horse).
- KNN struggled to form meaningful neighborhoods in raw pixel space.

3.4 Why GPU Did Not Help

Despite GPU availability, runtime did not improve because:

- The implementation relied on Python loops.
- GPUs only accelerate vectorized / CUDA-based operations.
- Simply enabling GPU does not speed up non-vectorized code.

3.5 Key Inferences (Task 2)

- KNN does not scale to high-dimensional image data.
- The *Curse of Dimensionality* weakens distance-based similarity.
- Computational cost dominates practical usability.
- Raw pixel distances are insufficient for image classification.

4 Overall Learnings

- KNN is simple but computationally expensive.
- Distance-based learning works best on low-dimensional data.
- Feature representation is more important than the classifier itself.
- Poor results can still provide strong algorithmic insights.
- Algorithm failure reveals why modern models use learned features.

5 Conclusion

This assignment demonstrated both the strengths and limitations of KNN. While highly effective for structured medical data, KNN becomes computationally infeasible and conceptually weak for image classification tasks such as CIFAR-10. Understanding why an algorithm fails is as valuable as understanding when it succeeds.