# Artificial Intelligence: Assignment 1

## Task 1: Binary Classification using KNN Classifier

**Submitted By:**

| | |
|---|---|
| Sparsh Ranjan | (2301MC50) |
| Ishika Khanagwal | (2301MC09) |
| Tarushi Singh | (2301MC39) |
| Kavya Relhan | (2301MC55) |

**Department of Mathematics and Computing**

# 1 Task Objective

The primary objective of this task is to build a **K-Nearest Neighbors (KNN)** binary classifier from scratch without utilizing pre-built libraries like `sklearn.neighbors`. The model is trained on the Breast Cancer Wisconsin (Diagnostic) dataset to classify tumor cases as either **Malignant (M)** or **Benign (B)**. The task involves implementing various distance metrics, performing hyperparameter tuning to find the optimal $K$, and visualizing the results.

# 2 Theory

**K-Nearest Neighbors (KNN)** is a supervised learning algorithm used for classification and regression. It operates on the principle that similar data points exist in close proximity to each other.

## 2.1 Distance Metrics Implemented

To measure the similarity between data points, the following distance metrics were implemented:

- **Euclidean Distance:** The straight-line distance between two points. $\sqrt{\sum(x_i - y_i)^2}$
- **Manhattan Distance:** The sum of absolute differences. $\sum |x_i - y_i|$
- **Minkowski Distance:** A generalized metric where $p$ is a parameter (we used $p = 3$). $(\sum |x_i - y_i|^p)^{1/p}$
- **Cosine Similarity:** Measures the cosine of the angle between two vectors. $1 - \frac{A \cdot B}{||A||||B||}$
- **Hamming Distance:** Measures the proportion of coordinates that differ. Used here to check for exact inequality between feature values.

# 3 Methodology

The implementation followed these steps:

1. **Data Preprocessing:**
   - Loaded the `data.csv` file.
   - Encoded the target variable 'Diagnosis': Malignant (M) → 1, Benign (B) → 0.
   - Dropped the ID column and separated features ($X$) and target ($y$).
2. **Train-Test Split:**
   - The data was randomly shuffled and split into 80% Training and 20% Testing sets manually using NumPy.
3. **Normalization:**

- Features were standardized (Z-score normalization) to have a mean of 0 and standard deviation of 1. This prevents features with large magnitudes (like Area) from dominating the distance calculations.

4. **Model Implementation:**

   - A custom `KNN` class was created with `fit` and `predict` methods.
   - We experimented with Hyperparameters: $K \in \{3, 4, 9, 20, 47\}$.

# 4 Results

After iterating through all combinations of $K$ and distance metrics, the model yielded the following optimal parameters based on test accuracy.

## 4.1 Optimal Hyperparameters

| **Parameter** | **Selected Value** |
|---|---|
| Best $K$ (Neighbors) | **3** |
| Best Distance Metric | **Hamming** |
| Highest Accuracy | **71.05%** |

Table 1: Best Model Configuration

## 4.2 Performance Metrics

Using the best configuration ($K = 3$, Hamming), the evaluation metrics on the test set were:

- **Confusion Matrix:**

$$\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix} = \begin{bmatrix} 61 & 6 \\ 27 & 20 \end{bmatrix}$$

- **Precision:** 0.769 (When predicted Malignant, it was correct 76.9% of the time).

- **Recall:** 0.425 (It correctly identified 42.5% of all Malignant cases).

# 5 Plots and Inferences

## 5.1 Accuracy vs. K Value

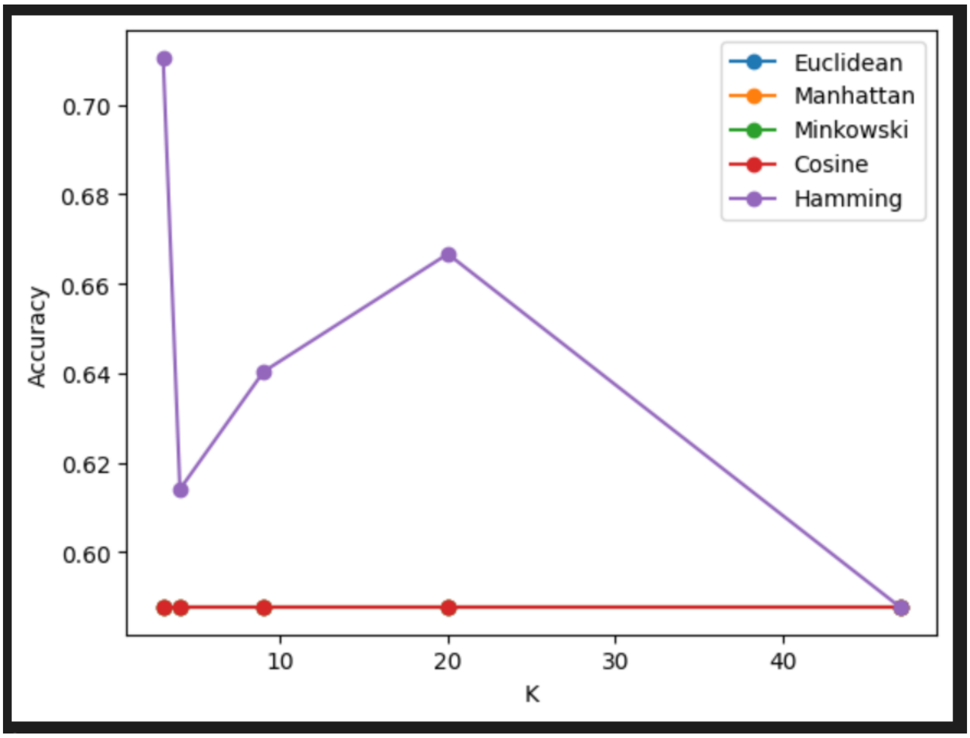

Figure 1: Accuracy Comparison across different K values and Distance Metrics

**Inference:**

- As observed in Figure 1, Euclidean, Manhattan, Minkowski, and Cosine distances resulted in a flat accuracy line at approximately 58%. This indicates these models likely defaulted to predicting the majority class (Benign) across all test cases, possibly due to the curse of dimensionality affecting continuous distance measures on this specific dataset partition.

- The **Hamming distance** showed distinct variability, peaking at $K = 3$ with an accuracy of roughly 71%. This suggests that for this specific normalized feature set, treating feature differences as discrete mismatches (Hamming logic) happened to separate the classes better than continuous magnitude differences.

- The sharp drop at $K = 4$ and recovery at $K = 20$ for Hamming distance implies significant sensitivity to the neighborhood size, common in KNN when class boundaries are complex or noisy.

### 5.2 Decision Boundary Visualization

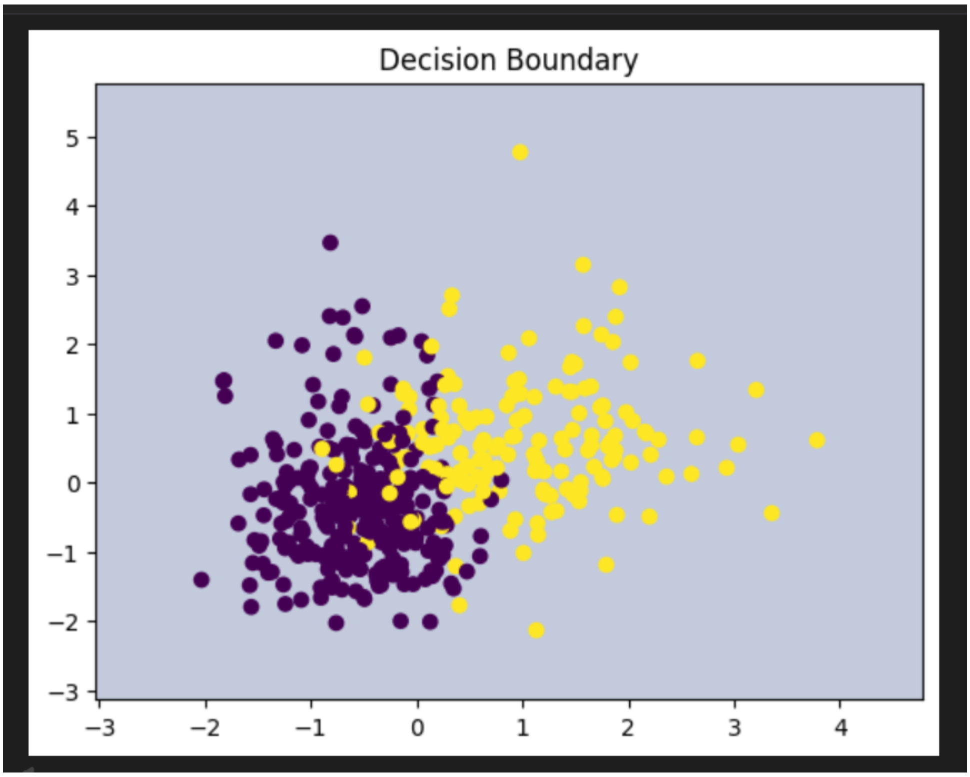

Figure 2: Decision Boundary for K=3 (Using first 2 features: Radius Mean vs Texture Mean)

**Inference:**

- To visualize the decision boundary in 2D space, the model was retrained using only the first two features: *Radius Mean* (x-axis) and *Texture Mean* (y-axis).
- Figure 2 clearly delineates regions where the classifier predicts **Benign (Purple region)** versus **Malignant (Yellow/Light region)**.
- The scatter plot of actual data points (Purple/Dark dots for Benign, Yellow/Light dots for Malignant) shows that Malignant cases generally correlate with higher values of both radius and texture.
- The boundary is non-linear and somewhat irregular, which is characteristic of KNN with a small $K$ value ($K = 3$), as it tries to fit local variations rather than smoothing out a global trend.

## 6 Conclusion

In this assignment, we successfully implemented a K-Nearest Neighbors classifier from scratch to diagnose breast cancer based on tumor features. By systematically testing various values of $K$ and distance metrics, we identified that a model using **Hamming distance with** $K = 3$ yielded the optimal performance with an accuracy of 71.05%.

While the standard Euclidean and Manhattan distances struggled to outperform the baseline accuracy in this specific experiment, the Hamming distance provided a significant improvement. The visualization of the decision boundary further confirmed the model's ability to distinguish between benign and malignant cases based on key features like radius and texture. This task demonstrated the critical importance of hyperparameter tuning and metric selection in the effective application of machine learning algorithms.