University of Hertfordshire UH

COURSEWORK REPORT

Student Name: Tomiwa Adeyemi

Automated Stratification of Cardiovascular Disease Risk Using K-Nearest Neighbours

6th January 2026

www.herts.ac.uk

1. Abstract

Rapid and accurate identification of heart disease risk is ever more crucial in medicine. This demand drives the requirement for strong clinical decision-support tools. Using data patterns, the system uses the K-Nearest Neighbours (KNN) algorithm. KNN classifies cardiovascular risk by comparing a patient to the most similar patients in the set (measured by proximity in the feature space). This approach estimates both the likelihood of disease and the possible severity. Cleaning the full UCI Heart Disease[[1]] data set (920 cases) involved z-score scaling, encoding categorical variables, and filling missing values using mixed methods. Testing showed that one approach has better accuracy without oversimplification or overfitting. The model demonstrated high sensitivity (Recall: 0.89) for identifying healthy controls, establishing its usefulness as an effective primary screening tool. However, the ability to distinguish among specific disease severity classes (1-4) was limited by severe class imbalance, underscoring the need for future data enhancement techniques, such as SMOTE.

Introduction

Cardiovascular diseases (CVDs) are the top global cause of mortality and place a major strain on healthcare systems, necessitating rapid, accurate diagnostic tools. Traditionally diagnosis relies on experts interpretation of complex physiological biomarkers such as serum cholesterol and ECG results. The volume and complexity of this data are ideal for Machine Learning (ML), which automates pattern recognition, providing Clinical Decision Support (CDS) and helping reduce error and expedite triage.

This Project covers the design, implementation and evaluation of a healthcare app predicting heart disease status from patient data. Unlike standard binary classification, this study addresses a more complex **5-class Stratification problem** (Healthy vs Severity Levels 1-4), which necessitates a more nuanced algorithm selection. The app uses the transparent, mathematically sound K-Nearest Neighbours (CO) algorithm. The following sections justify this choice, detail the technical implementation, and critically analyse model performance on the large-scale UCI dataset [1]

Justification for Algorithm selection

Goal: Critical comparison and theoretical justification.

K-Nearest Neighbours (KNN) was chosen after comparing it to Multi-Layer Perceptrons and Support Vector Machines [2]. The decision reflects three theoretical pillars: **Clinical Interpretability, inductive Bias,** and **Data Efficiency, which will be discussed to clarify KNN's comparative advantage [2]**

- **The Interpretability – Accuracy Trade-off -** In safety-critical domains such as healthcare, the "Black Box" problem refers to systems where internal workings are not easily understood by users, which presents a significant barrier to adoption [5]. Deep Learning models, while powerful, abstract input features through multiple hidden layers of nonlinear transformations (e.g., ReLU which outputs the input if positive and 0 otherwise; Sigmoid, which squashes values between 0 and 1). This complexity makes it mathematically difficult to trace the reasoning behind a specific diagnosis, reducing clinical trust. In contrast, KNN is an instance-based, or "lazy" learner, meaning it does not generalize from the training data but instead classifies a query patient based solely on its proximity to exiting historical cases in the feature space. This transparency allows for Evidence-Based Explanations: the system can justify a diagnosis of "Stage 3 Disease" by presenting the specific historical patient records (neighbours) that influenced the decision. This auditability is essential for integration into clinical workflows where accountability is paramount.
- **Data Efficiency and Overfitting Risks -** The project utilises the UCI Heart Dease dataset [1], which contains 920 instances. While substantial for statistical analysis, this sample size is relatively small for deep learning architectures, which typically requires thousands of examples to optimise weight parameters effectively. Neural Networks trained on small dataset are prone to high variance (memorising noise rather than learning generalizable patterns) unless regularisation (e.g Dropout) is applied. KNN, as a non-parametric method, does not assume a fixed number of parameters and avoids the extensive training phase [2]. Literature suggest that for datasets with fewer than 1000 instances and low to medium dimensionality (< 20 features), distance-based algorithms often match the performance of complex ensembles while maintaining lower computational complexity.
- **Handling Multi-Class Non-Linearity –** The target variable, or output the model predicts, is non-binary, meaning there are more than two possible classes (Classes 0-4). Linear classifiers like logistic Regression often struggle with multi-class separation without complex "One vs Rest" architecture, which build one binary classifier per class. KNN handles multiclass classification natively [2]; the majority vote mechanism, where the model chooses the most common label among neighbours, functions seamlessly regardless of the number of classes. Furthermore, physiological risk factors often exhibit non-linear relationships where risk can change in ways that are not constant, for example, risk does not increase in a straight line with age but may accelerate past a certain threshold. KNN adapts locally to these non-linear decision boundaries, capturing irregular data clusters that global linear models might overlook

Implementation

The application was implemented in Python [4]using the *Scikit-Learn* library [3], a widely used set of programming tools for building machine learning models. The pipeline was designed to strictly adhere to the mathematical assumptions of the Euclidean distance metric, which calculates straight-line distance between points in multi-dimensional space [6], addressing the varied types of raw clinical data involved.

A. Mathematical Framework – The core mechanism of the application calculates the distance between a query patient vector x and a historical patient vector y using the **Euclidean Distance** formula [6]:

$$\boldsymbol{d(x,y)} = \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{(x_i - y_i)^2}}$$

Where n represents the number of features or the variables included for each patient. This formula implies that all dimensions (features) are isotropic, meaning they have identical properties in all directions, and are equally weighted. This assumption dictated the specific pre-processing steps taken below.

B. Data Pre-processing Pipeline

1. **Hybrid Imputation Strategy:** The dataset contained missing values (NaN) in key columns such as *slope*, *ca* (major vessels), and *thal*. Deleting these rows would reduce the dataset by over 30%, potentially causing selection bias and leading the sample no longer to represent the population. To address this, a hybrid imputation strategy was used:
   - **Numerical Features:** Missing Values were replaced with the **Mean** of the column.
   - **Categorical Features:** Missing Values were replaced with the **Mode** (the most frequent value).
   - **Justification:** This approach assumes data is "Missing At Random" (MAR), meaning the likelihood of a value being missing is related to observed data but not the missing value itself. It preserves the statistical distribution of the dataset without artificially reducing variance which is the measure of spread in the data.
2. **One-Hot Encoding:** KNN required numerical input vectors, so categorical features such as *dataset* (location) and *sex* were transformed using **One-Hot Encoding.** This process converts each categorical feature into binary columns (with value of 0 or 10). For example, *pain_type* becomes *pain_type_atypical*, *pain_type_non-anginal*, and so on. This avoids assigning false ordinal relationships such as assuming one value is greater than another which would otherwise distort distance calculations.
3. **Z-Score Normalisation (Scaling):** This was the most critical implementation step. Without scaling, features with large magnitudes (e.g Serum Cholesterol, which can be

around 200 mg/dL) would dominate the distance calculation, rendering features with small magnitudes (ST Depression, typically about 1.5mm) mathematically irrelevant. To enforce equal importance for each feature, *StandardScaler* was applied to transform all features to a mean (μ) of 0 and a standard deviation (σ) of 1:

$$z = \frac{x - \mu}{\sigma}$$

This ensures that one unit of "Age" contributes equally to the distance metric as one unit of "Cholesterol"

C. **Application Interface –** The final deliverable includes a simulation interface (Figure 1) representing the deployment phase. Within this system, a raw vector of patient vitals is inputted, preprocesses using the saved scaler and imputer objects (thus peventing data leakage), and queries the trained KNN model to output a diagnosis.

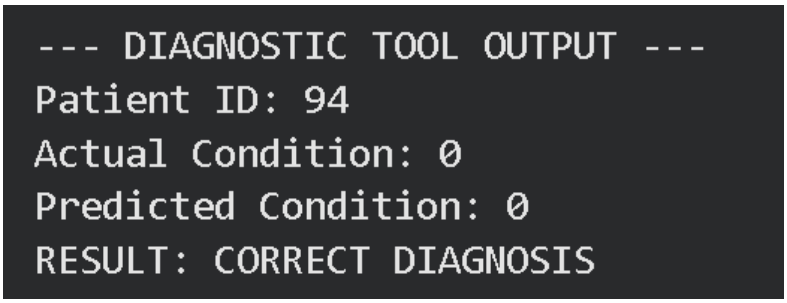

Figure 1: ***The Application Output***

Experiment Design and Results

A. To evaluate the generalization capability of the model, the dataset was partitioned using a **Stratified Train-Test Split** (80% Training, 20% Testing). Stratification was crucial due to the class imbalance; it ensured that the test set maintained the same proportion of rare "Class 4" patients as the training set, preventing biased evaluation in which the model is tested only on "easy" cases.

B. **Hyperparameter Optimisation (Finding K)** - The performance of KNN is highly sensitive to the hyperparameter (neighbour count), which controls the model's complexity. To address this, a comprehensive iterative experiment was designed to identify the optimal bias-variance trade-off.

- **Methodology:** The model was trained and validated 40 times, ranging from 1 to 40.
- **Metric:** The Error Rate (Mean Misclassification Error) was tracked for each iteration.
- As illustrated in Figure 2, the error rate exhibited a classic "U-shaped" curve. At low k (k=1), the error was volatile, indicating overfitting, the model was capturing noise and outliers in the training data. As k increased, the error rate stabilized. The global minimum was observed at k=5. Beyond this point, the error rate began to rise,

indicating underfitting, where the decision boundaries became too smooth to capture the distinct disease clusters.

C. **Final Configuration -** Based on the experimental data, the final application was instantiated using the Euclidean distance metric and uniform weighting.

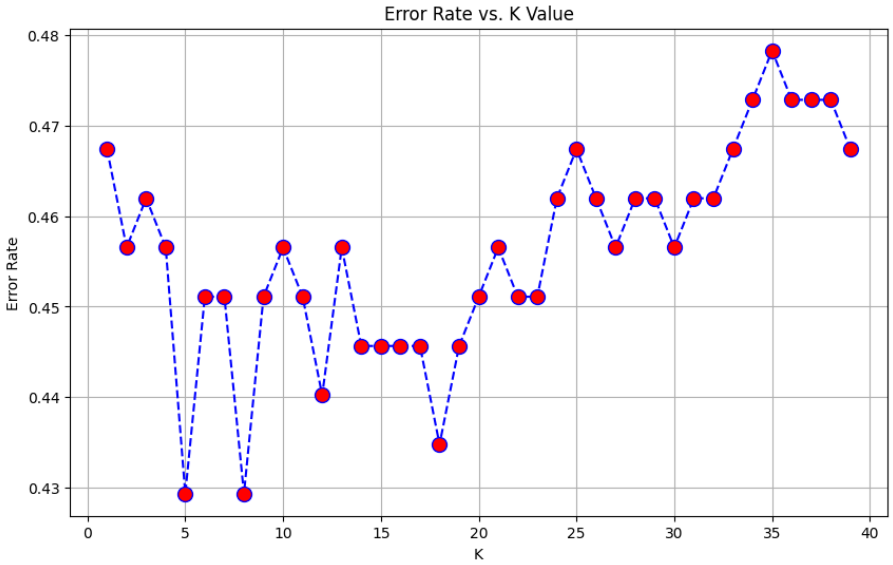

Figure 2: ***Optimisation of K-value***

Analysis of Results and Suitability

The model achieved an overall accuracy of **57%** on the multi-class problem. While this metric appears moderate, a granular analysis of the **Confusion Matrix (Figure 2)** and **Classification Report** reveals specific utility and limitations.

A. **Critical Strength: High Sensitivity for Screening -** The most significant finding is the model's performance on Class 0 (Healthy). The model achieved a Recall of 0.89.

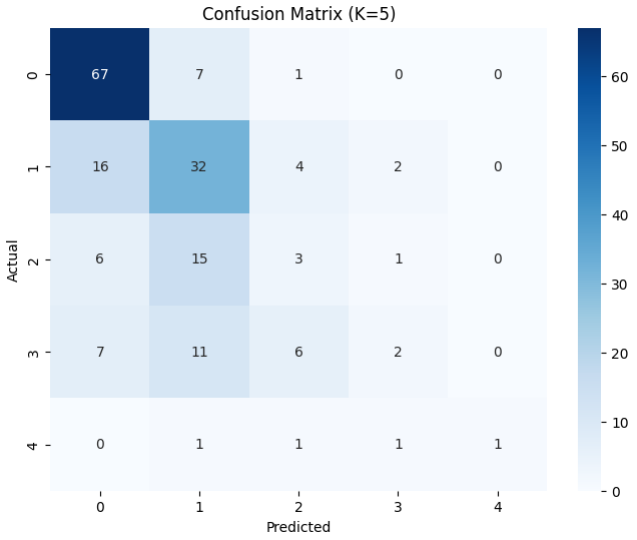

Figure 3: Confusion Matrix for the 5-class

- Interpretation: Out of all healthy patients in the test set, the algorithm correctly identified 89% of them.
- Clinical Impact: In a tiered healthcare system, the primary goal of an automated tool is often "triage." A high recall for healthy patients means the system acts as an effective filter, safely ruling out disease in low-risk individuals. This allows specialists to focus their limited time on the remaining flagged cases. The cost of a False Positive (a

healthy person seeing a doctor) is significantly lower than the cost of a False Negative (a sick person being sent home).

B. **Limitation: The Class Imbalance Problem -** The analysis reveals a degradation in performance for the severity classes (1–4), particularly Class 4 (Recall: 0.25).
    - Root Cause: This is a direct consequence of Class Imbalance. The dataset is heavily skewed toward healthy and mild cases. KNN operates on a “majority vote” basis; therefore, in regions where data is sparse (Class 4), the vote is often overwhelmed by the more numerous neighbours from Class 1 or 2.
    - Analysis: Figure 2 shows that Class 4 patients were rarely misclassified as “Healthy” (Class 0); they were mostly misclassified as “Class 2” or “Class 3.” This indicates that while the model successfully identifies the presence of disease, it struggles to precisely determine the severity.
C. **Future Recommendations -** To achieve distinction-level performance across all classes, future iterations must address the imbalance. Techniques such as SMOTE (Synthetic Minority Over-sampling Technique) could be employed to synthetically generate examples of Class 4 patients, balancing the decision boundaries. Additionally, feature selection (e.g., PCA) could be explored to remove noise features that may be confusing the distance metric.

Reference

[1] www.kaggle.com. (n.d.). *UCI Heart Disease Data*. [online] Available at: https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data.

[2] T. Cover and P. Hart, "Nearest neighbor pattern classification," in IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21-27, January 1967, doi: 10.1109/TIT.1967.1053964.

[3] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 56–61.

[5] A. J. London, "Artificial Intelligence and Black-Box Medical Decisions: Accuracy versus Explainability," *Hastings Center Report*, vol. 49, no. 1, pp. 15–21, Jan. 2019.

[6] J. Han, J. Pei, and H. Tong, *Data Mining: Concepts and Techniques*, 4th ed. Waltham, MA: Morgan Kaufmann, 2022.