DCU BUSINESS SCHOOL

DCU

Evaluating Machine Learning Classification Algorithms for Heart Disease Prediction: *Performance on Full and Reduced Clinical Datasets*

Module Code and Title: BAA1027 Data Analytics: Machine Learning & Advanced Python

Lecturer: Dr. Michael Farayola

Student name: Tom Blume

Student ID: 46557

Date of Submission: 11.04.2026

Word Count: 2717

Table of Content

List of Abbreviations

CV..... .....Cross-Validation
DT..... ..... Decision Trees
HD ....Heart disease
KNN..... .....K-Nearest Neighbour
LMICs..... ..... Low- And Middle-Income Countries
ML ....Machine Learning
NB..... .....Naive Bayes
NN .... Neural Network
RF ....Random Forest
SVM..... ..... Support Vector Machines
VHD..... ..... Valvular Heart Disease

Abstract

Heart disease (HD) remains the leading cause of mortality worldwide, accounting for an estimated 17.9 million deaths annually and approximately 31% of all global deaths. Early and accurate diagnosis is critical, yet particularly challenging in low- and middle-income countries (LMICs), where shortages of medical equipment, specialist clinicians, and healthcare infrastructure severely limit diagnostic capacity. Machine learning (ML) classification algorithms have emerged as promising tools to support HD prediction; however, their performance under resource-constrained conditions remains underexplored.

This study evaluates four supervised ML classification algorithms: Naive Bayes (NB), K-Nearest Neighbour (KNN), Decision Trees (DT), and Random Forest (RF); applied to a Kaggle heart disease dataset comprising 1,000 patient records and 16 attributes, assessed across a full (15-feature) and a simplified (10-feature) dataset using accuracy, precision, recall, specificity, and F1-score with 5-fold stratified cross-validation (CV).

RF produced the most credible full-dataset results (accuracy: 99.0%, F1: 98.7%). On the simplified dataset, accuracy declined by an average of 9.25%, yet RF retained perfect recall (100%) and DT achieved the highest F1-score of 85.23%, supporting the feasibility of ML-assisted screening in resource-limited environments.

1. Introduction

HD remains the primary cause of death worldwide, with an estimated 17.9 million deaths annually (Shah et al., 2020), representing around 31% of all global deaths (Vincent Paul et al., 2022). Timely diagnosis is therefore critical to avoid life-endangering cardiac events and to initiate appropriate therapeutic interventions at an early stage (Khan et al., 2019). Clinically, HD encompasses a range of conditions, including cardiomyopathy and coronary HD, which together impose a substantial burden on individuals and healthcare systems (Soni et al., 2011).

This burden is particularly pronounced in emerging LMICs, where a vast number of cardiovascular deaths occur (Shah et al., 2020; Vincent Paul et al., 2022). In these settings, precision in prediction is vital to effectively treat heart patients despite limited financial, technological, and human resources (Folorunso et al., 2022). The burden of valvular heart disease (VHD) is especially significant in the Indian subcontinent and Africa, where social and systemic issues further complicate diagnosis and long-term management (Santangelo et al., 2023).

To address these challenges, this paper evaluates four ML classification algorithms using two criteria: ease of use (Criteria A) and good predictive performance as reported in the literature (Criteria B). The analysis is based on a Kaggle HD dataset containing 1,000 records and 15 attributes, divided into a full dataset retaining all input variables and a simplified dataset restricted to features measurable without professional equipment. The objective is to investigate whether ML models trained on simplified data can still deliver sufficiently accurate predictions to support risk stratification in resource-limited environments.

2. Literature Review

2.1. Machine learning for heart disease prediction

HD prediction research in ML commonly relies on structured clinical datasets originating from publicly accessible repositories. One of the most widely used datasets is the Cleveland HD dataset from the UCI Machine Learning Repository, which contains 303 patient records and focuses on 14 clinically relevant attributes derived from an original set of 76 variables. These attributes include demographic and clinical indicators such as age, blood pressure, cholesterol levels, and electrocardiographic results (Mohan *et al.*, 2019; Tr *et al.*, 2022). Similar dataset configurations are used across multiple studies, demonstrating the importance of standardized benchmark data for comparative evaluation. In addition to the

Cleveland dataset, Vincent Paul *et al.* (2022) also used other databases such as the Hungarian, Switzerland, Long Beach, and VA HD datasets to expand training data and improve model generalisation. Some studies also employ Kaggle-hosted versions of HD datasets that replicate the Cleveland structure with the same 303 instances and 14 variables, enabling consistent experimentation across different ML models (Folorunso *et al.*, 2022).

Numerous studies have applied supervised ML algorithms to these datasets in order to predict HD presence. Common algorithms include NB, DT, KNN, RF and Support Vector Machines (SVM). Comparative experiments have demonstrated that model performance can vary depending on algorithm configuration and preprocessing techniques.

Empirical findings across studies suggest that certain algorithms consistently perform well on HD datasets. KNN has frequently achieved strong accuracy results, particularly when the number of neighbours is optimised (Tr *et al.*, 2022; Khan *et al.*, 2019). Ensemble approaches such as Extra Trees and RF also demonstrate strong predictive performance due to their ability to aggregate multiple decision trees and reduce variance (Folorunso *et al.*, 2022; Tr *et al.*, 2022). Neural network (NN) models combined with feature selection methods have achieved very high accuracy levels, sometimes exceeding 97%, indicating the benefit of dimensionality reduction before model training (Vincent Paul *et al.*, 2022).

Analysis of the broader literature reveals several insights into algorithm behaviour and modelling challenges in HD prediction. High-dimensional clinical data can lead to the "curse of dimensionality," making models more difficult to train effectively (Tr *et al.*, 2022). Small dataset sizes also increase the risk of overfitting when models are trained on limited samples without sufficient validation (Khan *et al.*, 2019). Recent research therefore emphasises explainable ML approaches to identify the most influential clinical attributes while maintaining generalisation capability across different HD datasets (Folorunso *et al.*, 2022).

2.2. Heart disease prediction in developing countries

Health systems in developing countries experience major technological and informational limitations that restrict the adoption of digital healthcare solutions. A key challenge is the lack of interoperability between electronic health record systems, which often rely on centralized databases that are vulnerable to link failures and limited fault tolerance (Rinty *et al.*, 2022). These issues are further intensified by unstable internet connectivity, unreliable electricity supply, and insufficient investment in information and

communication technologies (Rinty *et al.*, 2022). Across numerous LMICs, healthcare professionals are confronted with challenges related to resource shortages, which extend to human capacity. Concurrently, these professionals encounter substantial workloads and protracted patient queues, a phenomenon that is particularly evident in select sub-Saharan African countries. The dearth of cardiac surgeons, with a reported ratio of only one per ten million inhabitants, engenders substantial barriers to the timely detection and management of cardiovascular diseases. (Santangelo *et al.*, 2023; Vincent Paul *et al.*, 2022)

Within such constrained environments, ML approaches for HD prediction have been proposed as supportive tools for clinical decision making, particularly where diagnostic expertise and advanced medical equipment are limited (Folorunso *et al.*, 2022). Studies conducted in LMICs, especially in India, evaluate supervised ML algorithms to improve diagnostic accuracy and support early identification of HD risk (Shah *et al.*, 2020).

However, the development of reliable ML-based HD prediction models in LMICs is further constrained by data-related challenges (Shah *et al.*, 2020). Clinical datasets often contain missing values, noise, and inconsistencies that require extensive preprocessing before modelling can be performed (Shah *et al.*, 2020). Many models demonstrate strong performance primarily on small datasets, indicating limitations in available training data and potential risks of overfitting (Tr *et al.*, 2022). Limited access to diagnostic technologies also results in underreporting of cardiovascular diseases and sparse population-level data in several regions (Tsao *et al.*, 2023).

2.3. Synthesis

Numerous studies identify several easily measurable attributes that are strongly associated with HD, including behavioural, physiological, biological, demographic measures (Vincent Paul *et al.*, 2022; Tsao *et al.*, 2023). These attributes function as predisposing factors that contribute to HD development and are frequently used in ML models to identify correlations between risk indicators and disease occurrence (Shah *et al.*, 2020). As demonstrated in the study by Mohan *et al.* (2019), clinical observations indicate that the probability of the occurrence of HD is reduced in females in comparison with males.

In developing countries, where diagnostic resources and specialised medical infrastructure are limited, the strong association between easily measurable attributes and HD risk supports simplified predictive approaches. Feature selection techniques demonstrate that reduced datasets containing key attributes can still provide meaningful predictive insights,

motivating comparisons between full clinical datasets and reduced datasets to evaluate model robustness and support more efficient allocation of healthcare resources (Vincent Paul *et al.*, 2022).

3. Methodology

3.1. Dataset and Data Preparation

The dataset used is the "Heart Disease prediction" dataset by Rashad Mammadov from the Kaggle repository, containing 1,000 patient records and 16 attributes (see Appendix, Dataset). Among these, 15 attributes serve as input features (X) and one attribute represents the target variable (Y), indicating presence or absence of HD. The dataset contains no missing values; therefore, no imputation was required. An 80–20 train-test split was applied, with 5-fold stratified CV during training.

Categorical and ordinal variables were encoded prior to model development: binary attributes via label encoding (0/1), ordinal variables via ordinal encoding to preserve category order, and categorical variables without ordinal structure via one-hot encoding. Numerical features were standardised to ensure equal contribution during model training. (see Appendix, Python Code - Section 2)

3.2. Classification Algorithms

Four supervised ML classifiers were selected based on ease of use and documented predictive performance. NB was chosen as a probabilistic baseline due to its minimal complexity and consistent inclusion across the literature as a benchmark model (Folorunso et al., 2022; Khan et al., 2019). KNN was selected for its well-documented performance on HD datasets, with fine-tuning focused on the number of neighbours, distance metric, and weighting scheme (Tr et al., 2022; Khan et al., 2019). DT was included for interpretability, with key parameters being maximum depth and minimum sample thresholds to manage overfitting. RF was selected as the most performance-driven algorithm, with tuning centred on the number of estimators, tree depth, and feature selection per split (Folorunso et al., 2022; Tr et al., 2022). All hyperparameters were set to conservative values to prioritise generalisability given the dataset size of 1,000 records. (see Appendix, Python Code - Section 3)

3.3. Full vs. Simplified Dataset

To investigate ML performance under resource-constrained conditions, the dataset was divided into a full and a simplified version. The full dataset retains all 15 input features. The simplified dataset excludes five attributes requiring professional equipment or clinical expertise: Cholesterol and Blood Sugar (laboratory blood analysis), Diabetes (formal clinical diagnosis), Exercise-Induced Angina (supervised stress testing), and Chest Pain Type (clinician classification). The remaining ten features: Age, Gender, Blood Pressure, Heart Rate, Smoking, Alcohol Intake, Exercise Hours, Family History, Obesity, and Stress Level; are retained, as these can be measured or self-reported without specialist intervention, making them realistically accessible in LMIC settings. (see Appendix, Python Code - Section 2)

Both datasets are evaluated using the same four classification algorithms under identical conditions. The results of both datasets are then compared directly to evaluate the trade-off between data availability and predictive performance, and to assess whether a simplified feature set can still support meaningful HD risk stratification in resource-limited environments.

3.4. Evaluation Metrics

Evaluation of ML models for HD prediction relies on standard classification metrics derived from confusion matrices, including accuracy, precision, recall, specificity, and F1-score, with CV techniques applied to ensure reliable model validation and reduce sampling bias (Folorunso et al., 2022; Tr et al., 2022). In the clinical context of HD prediction, recall is arguably the most critical metric: a false negative, failing to identify a patient with HD, carries significantly greater risk than a false positive. Precision measures the proportion of positive predictions that are genuinely positive, with low precision generating unnecessary follow-up strain on limited LMIC resources. Specificity complements recall by measuring correct identification of non-HD cases. CV accuracy is reported as mean ± standard deviation across five folds to assess generalisation stability. The F1-score provides a single balanced measure of precision and recall, particularly informative in medical diagnosis where both error types carry meaningful consequences (Folorunso et al., 2022).

4. Results

4.1. Full Dataset

Across the full 15-feature dataset, all four classifiers delivered strong predictive performance, though with notable differences in reliability. DT achieved perfect scores across every metric, however its CV accuracy of 0.9988 ± 0.0025 indicates that this result reflects near-perfect memorisation of the training data rather than genuine generalisation. Notably, RF shares this identical CV accuracy and standard deviation yet produces more credible results precisely because its test accuracy of 0.990 falls marginally below perfect. This suggests RF generalises where DT merely memorises. RF's precision and specificity of 1.000, recall of 0.9744, and F1-score of 0.9870 further confirm it as the most reliable classifier on the full dataset. NB performed competitively given its simplicity, achieving 0.91 accuracy, a precision of 0.9545, and an F1-score of 0.8750, with a CV accuracy of 0.9075 ± 0.0083 indicating reliable generalisation. KNN was the weakest performer on the full dataset, with 0.875 accuracy and an F1-score of 0.8344, alongside the highest CV standard deviation of ±0.0199 among all full-dataset models, reflecting greater sensitivity to feature scale and local data distribution despite standardisation. (see Appendix, Evaluation Table)

4.2. Simplified Dataset

Removing the five clinically restricted features produced a consistent and measurable decline in performance across all models. RF retained the strongest overall results, achieving 88.0% accuracy and a perfect recall of 1.000, meaning it correctly identified every actual HD case in the test set, though at the cost of 24 false positives, as reflected in its reduced precision of 0.7647 and specificity of 0.8033. DT performed well on the simplified data, recording the highest F1-score among all simplified models at 0.8523, with a recall of 0.9615 and accuracy of 0.87, suggesting its tree-based logic adapts effectively to a reduced feature space. NB maintained reasonable performance at 0.835 accuracy with a recall of 0.8462, while KNN declined most substantially to 0.82 accuracy and an F1-score of 0.7632, indicating greater dependency on the removed clinical features. (see Appendix, Evaluation Table)

4.3. Full vs. Simplified Dataset: Comparative Evaluation

Comparing both datasets directly, the average accuracy reduction across all four models was approximately 9.25 percentage points. The most significant degradation was observed

in precision and specificity, reflecting an increased rate of false positives under the simplified feature set. Recall, by contrast, remained relatively robust, particularly for RF and DT, suggesting that the retained features carry sufficient predictive signal to identify positive HD cases even without laboratory or clinician-dependent inputs. CV accuracy on the simplified dataset ranged from 0.7488 (KNN) to 0.8575 (RF). The higher standard deviations observed for NB (±0.0272) and KNN (±0.0228), compared to RF (±0.0183) and DT (±0.0100), further indicate reduced model stability under the restricted feature set. This is consistent with the findings of Tr et al. (2022) regarding overfitting risks on reduced training data. (see Appendix, Evaluation Table)

4.4. Discussion

These findings carry direct implications for HD risk stratification in resource-constrained environments. In LMICs, where laboratory diagnostics, stress testing, and specialist assessment are frequently inaccessible, a simplified model achieving 88% accuracy with perfect recall represents a practically viable screening tool. The clinical priority in such settings is minimising false negatives, failing to identify a patient with HD carries far greater risk than a false positive triggering a follow-up referral. RF's perfect recall on the simplified dataset therefore makes it the most suitable algorithm for deployment in low-resource contexts, despite its reduced precision. This supports the broader argument advanced by Folorunso *et al.* (2022) and Shah *et al.* (2020) that ML-based HD prediction can remain clinically meaningful even when constrained to self-reportable or easily measurable variables, provided algorithm selection is guided by the specific diagnostic priorities of the deployment environment.

5. Conclusion

This study evaluated four ML classification algorithms across a full and a simplified clinical dataset to assess whether HD prediction remains viable when restricted to easily obtainable features. Results demonstrate that all four models delivered strong performance on the full dataset, with RF emerging as the most reliable classifier. On the simplified dataset, performance declined by an average of 9.25 percentage points in accuracy, yet RF maintained perfect recall, and DT achieved the highest F1-score of 0.8523, indicating that meaningful risk stratification remains achievable without specialist-dependent variables.

Several limitations must be acknowledged. The dataset of 1,000 records is relatively small, increasing the risk of overfitting, as evidenced by DT's perfect full dataset scores. The

Kaggle dataset originates from a single source and may not reflect the clinical diversity of LMIC populations, limiting external validity. Furthermore, the feature set was simplified on the basis of assumptions regarding equipment accessibility, the variability of which may be significant across different low-resource settings. In addition, the incorporation of self-reported features (e.g. stress levels, smoking status, alcohol intake) introduces measurement error that is not addressed by the models.

In practice, the simplified RF model represents a feasible entry point for community-level HD screening in LMICs, where its perfect recall ensures that high-risk patients are consistently flagged for further assessment. Deployment could take the form of a lightweight decision-support tool operated by non-specialist healthcare workers, requiring only self-reported and basic physiological inputs. Future research should validate these findings on larger, geographically diverse datasets and explore model calibration to reduce the false positive rate that accompanies RF's aggressive positive prediction behaviour.

Declaration of Authorship

DCU Business School
Assignment Submission

Student Names and Student Numbers: Tom Blume (46557)
Programme: Bachelor of Arts in Global Business (Germany)
Project Title: Evaluating Machine Learning Classification Algorithms for Heart Disease Prediction: Performance on Full and Reduced Clinical Datasets
Module Code and Title: BAA1027 Data Analytics: Machine Learning & Advanced Python
Lecturer: Dr. Michael Farayola
Project Due Date: 20.04.2026

Declaration

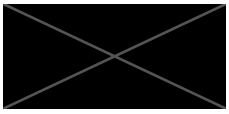
I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying is a grave and serious offence in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion, or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references.

I have not copied or paraphrased an extract of any length from any source without identifying the source and using quotation marks as appropriate. Any images, audio recordings, video or other materials have likewise been originated and produced by me or are fully acknowledged and identified.

This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. I have read and understood the referencing guidelines found at https://www.dcu.ie/library/citing-referencing and/or recommended in the assignment guidelines.

I understand that I may be required to discuss with the module lecturer/s the contents of this submission.

I/me/my incorporates we/us/our in the case of group work, which is signed by all of us.

Signed: 

Reference List

Folorunso, S.O. *et al.* (2022) 'Heart Disease Classification Using Machine Learning Models', in S. Misra *et al.* (eds) *Informatics and Intelligent Applications*. Cham: Springer International Publishing (Communications in Computer and Information Science), pp. 35–49. Available at: https://doi.org/10.1007/978-3-030-95630-1_3.

Khan, Y. *et al.* (2019) 'Machine Learning Techniques for Heart Disease Datasets: A Survey', *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*. *ICMLC '19: 2019 11th International Conference on Machine Learning and Computing*, Zhuhai China: ACM, pp. 27–35. Available at: https://doi.org/10.1145/3318299.3318343.

Mohan, S. *et al.* (2019) 'Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques', *IEEE Access*, 7, pp. 81542–81554. Available at: https://doi.org/10.1109/ACCESS.2019.2923707.

Rinty, M.R. *et al.* (2022) 'A prospective interoperable distributed e-Health system with loose coupling in improving healthcare services for developing countries', *Array*, 13, p. 100114. Available at: https://doi.org/10.1016/j.array.2021.100114.

Santangelo, G. *et al.* (2023) 'The Global Burden of Valvular Heart Disease: From Clinical Epidemiology to Management', *Journal of Clinical Medicine*, 12(6), p. 2178. Available at: https://doi.org/10.3390/jcm12062178.

Shah, D. *et al.* (2020) 'Heart Disease Prediction using Machine Learning Techniques', *SN Computer Science*, 1(6), p. 345. Available at: https://doi.org/10.1007/s42979-020-00365-y.

Soni, J. *et al.* (2011) 'Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction', *International Journal of Computer Applications*, 17(8), pp. 43–48. Available at: https://doi.org/10.5120/2237-2860.

Tr, R. *et al.* (2022) 'Predictive Analysis of Heart Diseases with Machine Learning Approaches', *Malaysian Journal of Computer Science*, pp. 132–148. Available at: https://doi.org/10.22452/mjcs.sp2022no1.10.

Tsao, C.W. *et al.* (2023) 'Heart Disease and Stroke Statistics—2023 Update: A Report From the American Heart Association', *Circulation*, 147(8). Available at: https://doi.org/10.1161/CIR.0000000000001123.

Vincent Paul, S.M. *et al.* (2022) 'Intelligent Framework for Prediction of Heart Disease using Deep Learning', *Arabian Journal for Science and Engineering*, 47(2), pp. 2159–2169. Available at: https://doi.org/10.1007/s13369-021-06058-9.

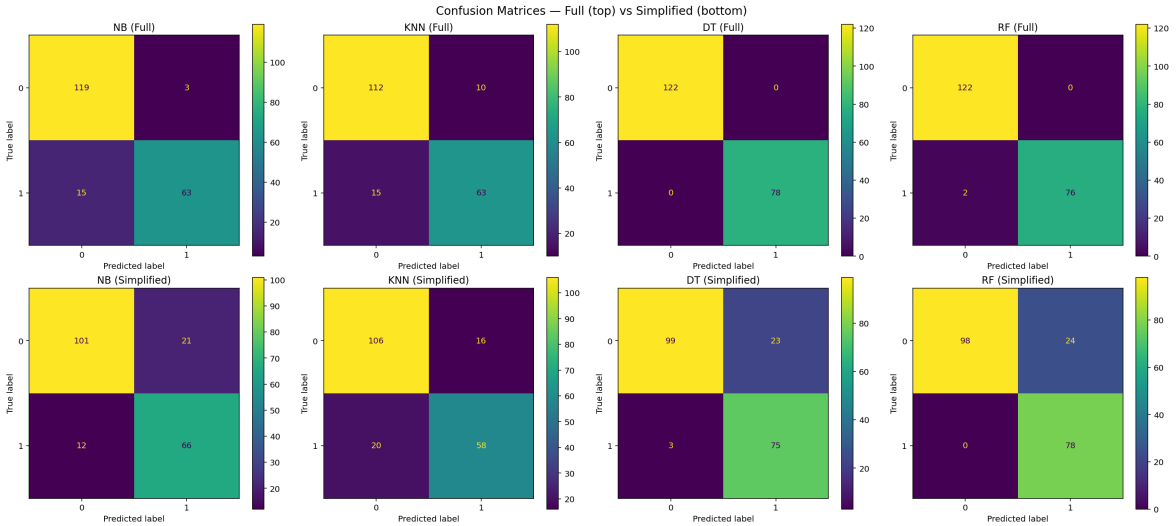
Appendix

Dataset

Variable	Type	Values / Range	Category
Age	**Numerical**	Discrete integer (years)	Demographic
Cholesterol	**Numerical**	Discrete integer (mg/dL)	Physiological
Blood Pressure	**Numerical**	Discrete integer (mmHg)	Physiological
Heart Rate	**Numerical**	Discrete integer (bpm)	Physiological
Exercise Hours	**Numerical**	Discrete integer (hours/week)	Behavioural
Blood Sugar	**Numerical**	Discrete integer (mg/dL)	Physiological
Gender	**Categorical**	Male, Female	Demographic
Chest Pain Type	**Categorical**	Typical Angina, Atypical Angina, Non-anginal Pain, Asymptomatic	Physiological
Smoking	**Ordinal**	Never < Former < Current	Behavioural
Alcohol Intake	**Ordinal**	None < Moderate < Heavy	Behavioural
Stress Level	**Ordinal**	Scale 1–10	Behavioural
Family History	**Binary**	0 = Absent, 1 = Present	Demographic
Diabetes	**Binary**	0 = Absent, 1 = Present	Physiological
Obesity	**Binary**	0 = Absent, 1 = Present	Physiological
Exercise Induced Angina	**Binary**	0 = Absent, 1 = Present	Physiological
Heart Disease (Target)	**Binary**	0 = Not Present, 1 = Present	Target Variable

Evaluation Graphics

Confusion Matrix



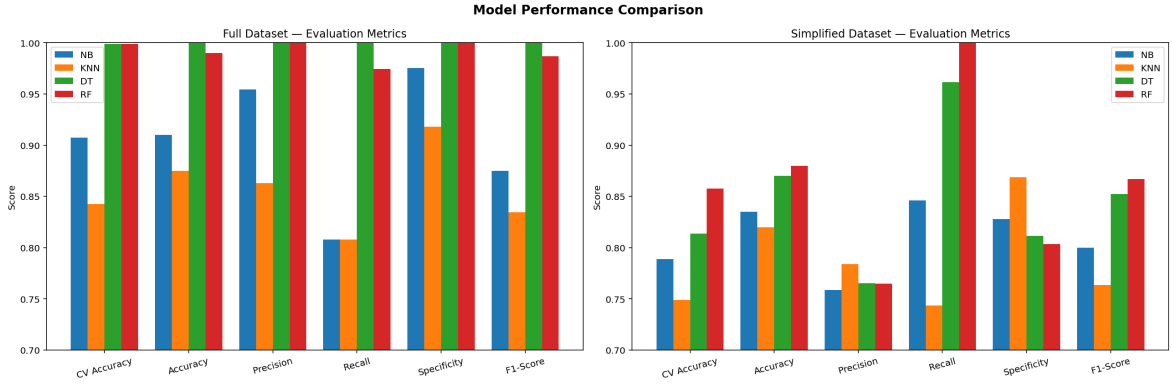
Evaluation Table

=== Full Dataset ===							
Algorithm	CV Accuracy	CV Accuracy SD	Accuracy	Precision	Recall	Specificity	F1-Score
NB	0,9075	0,0083	0,9100	0,9545	0,8077	0,9754	0,8750
KNN	0,8425	0,0199	0,8750	0,8630	0,8077	0,9180	0,8344
DT	0,9988	0,0025	1,0000	1,0000	1,0000	1,0000	1,0000
RF	0,9988	0,0025	0,9900	1,0000	0,9744	1,0000	0,9870

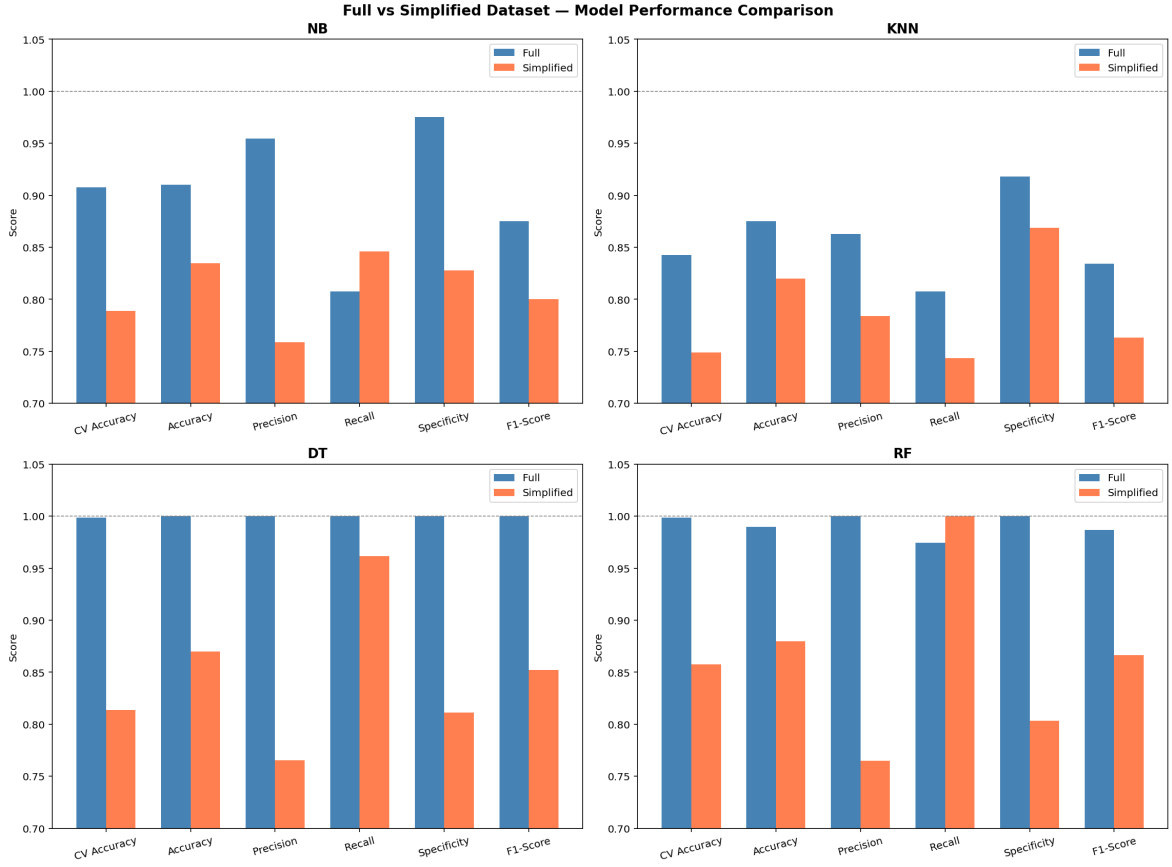
=== Simplified Dataset ===							
Algorithm	CV Accuracy	CV Accuracy SD	Accuracy	Precision	Recall	Specificity	F1-Score
NB	0,7888	0,0272	0,8350	0,7586	0,8462	0,8279	0,8000
KNN	0,7488	0,0228	0,8200	0,7838	0,7436	0,8689	0,7632
DT	0,8138	0,0100	0,8700	0,7653	0,9615	0,8115	0,8523
RF	0,8575	0,0183	0,8800	0,7647	1,0000	0,8033	0,8667

=== Performance Trend ===							
Algorithm	CV Accuracy	CV Accuracy SD	Accuracy	Precision	Recall	Specificity	F1-Score
NB	-11,87%	1,89%	-7,50%	-19,59%	3,85%	-14,75%	-7,50%
KNN	-9,37%	0,29%	-5,50%	-7,92%	-6,41%	-4,91%	-7,12%
DT	-18,50%	0,75%	-13,00%	-23,47%	-3,85%	-18,85%	-14,77%
RF	-14,13%	1,58%	-11,00%	-23,53%	2,56%	-19,67%	-12,03%
Overall	-13,47%	1,13%	-9,25%	-18,63%	-0,96%	-14,55%	-10,36%

ML Model Performance per Dataset Comparison



Dataset Performance per ML Model Comparison



Python code

1. Imports

All libraries imported at the top of the file to avoid duplication and ensure a clear dependency overview.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelEncoder, OrdinalEncoder, StandardScaler
from sklearn.model_selection import train_test_split, cross_val_score,
StratifiedKFold
from sklearn.metrics import (accuracy_score, precision_score, recall_score,
                             f1_score, confusion_matrix, ConfusionMatrixDisplay)
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

2. Dataset and Data Preparation

Loads the CSV, encodes binary/ordinal/categorical variables, performs 80–20 train-test split, standardises numerical features, and creates the simplified dataset by removing the five clinically restricted attributes.

```
# --- Load Dataset ---
df = pd.read_csv('/Users/tom.bme/Projects/heart_disease_dataset.csv',
                 sep=';',
                 keep_default_na=False)

print(df.shape)
print(df.isnull().sum())
print(df.dtypes)

# --- Define Features & Target ---
X = df.drop(columns=['Heart Disease'])
y = df['Heart Disease']

# --- Encode Binary Variables (0/1) ---
binary_cols = ['Gender', 'Family History', 'Diabetes', 'Obesity', 'Exercise
Induced Angina']
le = LabelEncoder()
for col in binary_cols:
    X[col] = le.fit_transform(X[col])

# --- Encode Ordinal Variables (preserve order) ---
oe_smoking = OrdinalEncoder(categories=[['Never', 'Former', 'Current']])
X[['Smoking']] = oe_smoking.fit_transform(X[['Smoking']])

oe_alcohol = OrdinalEncoder(categories=[['None', 'Moderate', 'Heavy']])
X[['Alcohol Intake']] = oe_alcohol.fit_transform(X[['Alcohol Intake']])

# Stress Level is already numeric (1-10) — no encoding needed

# --- Encode Categorical Variables (one-hot) ---
X = pd.get_dummies(X, columns=['Chest Pain Type'])

# --- Clean up dtypes ---
chest_pain_cols = ['Chest Pain Type_Asymptomatic', 'Chest Pain Type_Atypical
Angina',
                   'Chest Pain Type_Non-anginal Pain', 'Chest Pain Type_Typical
Angina']
X[chest_pain_cols]    = X[chest_pain_cols].astype(int)
X['Smoking']          = X['Smoking'].astype(int)
X['Alcohol Intake']   = X['Alcohol Intake'].astype(int)
```

```
# --- Full Dataset: Train-Test Split & Standardisation ---
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

numerical_cols = ['Age', 'Cholesterol', 'Blood Pressure', 'Heart Rate',
                  'Exercise Hours', 'Blood Sugar']

scaler = StandardScaler()
X_train[numerical_cols] = scaler.fit_transform(X_train[numerical_cols])
X_test[numerical_cols]  = scaler.transform(X_test[numerical_cols])

# --- Simplified Dataset: Remove features requiring professional equipment ---
cols_to_drop = ['Cholesterol', 'Blood Sugar', 'Diabetes',
                'Exercise Induced Angina',
                'Chest Pain Type_Asymptomatic',
                'Chest Pain Type_Atypical Angina',
                'Chest Pain Type_Non-anginal Pain',
                'Chest Pain Type_Typical Angina']

X_simple = X.drop(columns=cols_to_drop)
numerical_simple = ['Age', 'Blood Pressure', 'Heart Rate', 'Exercise Hours']

X_train_s, X_test_s, y_train_s, y_test_s = train_test_split(
    X_simple, y, test_size=0.2, random_state=42, stratify=y
)

scaler_s = StandardScaler()
X_train_s[numerical_simple] = scaler_s.fit_transform(X_train_s[numerical_simple])
X_test_s[numerical_simple]  = scaler_s.transform(X_test_s[numerical_simple])
```

3. Classification Algorithms and Evaluation

Defines NB, KNN, DT, and RF with their tuning parameters. Applies 5-fold stratified CV and computes all six evaluation metrics for both the full and simplified datasets.

```
# --- Define Models ---
models = {
    'NB':  GaussianNB(),
    'KNN': KNeighborsClassifier(n_neighbors=5, metric='euclidean'),
    'DT':  DecisionTreeClassifier(max_depth=5, min_samples_split=10,
random_state=42),
    'RF':  RandomForestClassifier(n_estimators=100, max_depth=5, random_state=42),
}

cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# --- Evaluation Function ---
def evaluate(models, X_tr, X_te, y_tr, y_te, cv):
    results = {}
    for name, model in models.items():
        cv_scores = cross_val_score(model, X_tr, y_tr, cv=cv, scoring='accuracy')
        model.fit(X_tr, y_tr)
        y_pred = model.predict(X_te)
        tn, fp, fn, tp = confusion_matrix(y_te, y_pred).ravel()
        results[name] = {
            'CV Accuracy':       round(cv_scores.mean(), 4),
            'CV Accuracy SD':    round(cv_scores.std(), 4),
            'Accuracy':          round(accuracy_score(y_te, y_pred), 4),
            'Precision':         round(precision_score(y_te, y_pred), 4),
            'Recall':            round(recall_score(y_te, y_pred), 4),
            'Specificity':       round(tn / (tn + fp), 4),
            'F1-Score':          round(f1_score(y_te, y_pred), 4),
        }
    return pd.DataFrame(results).T

# --- Run Evaluation ---
results_full_df   = evaluate(models, X_train,   X_test,   y_train,   y_test,   cv)
results_simple_df = evaluate(
```

```
    {name: type(m)(**m.get_params()) for name, m in models.items()},
    X_train_s, X_test_s, y_train_s, y_test_s, cv
)

print("=== FULL DATASET ===")
print(results_full_df)
print("\n=== SIMPLIFIED DATASET ===")
print(results_simple_df)
```

4. Evaluation Graphics

Generates three figures: confusion matrices (full vs simplified), grouped metrics bar charts per dataset, and side-by-side full vs simplified comparison per model.

```
# --- Retrain models for plotting ---
trained_full, trained_simple = {}, {}
for name, model in models.items():
    model.fit(X_train, y_train)
    trained_full[name] = model
    model_s = type(model)(**model.get_params())
    model_s.fit(X_train_s, y_train_s)
    trained_simple[name] = model_s

# --- Graphic 1: Confusion Matrices ---
fig, axes = plt.subplots(2, 4, figsize=(20, 9))
fig.suptitle('Confusion Matrices — Full (top) vs Simplified (bottom)',
fontsize=14, fontweight='bold')
for idx, name in enumerate(models.keys()):
    ConfusionMatrixDisplay.from_estimator(trained_full[name],   X_test,   y_test,
ax=axes[0, idx])
    ConfusionMatrixDisplay.from_estimator(trained_simple[name], X_test_s,
y_test_s, ax=axes[1, idx])
    axes[0, idx].set_title(f'{name} (Full)')
    axes[1, idx].set_title(f'{name} (Simplified)')
plt.tight_layout()
plt.savefig('confusion_matrices.png', dpi=150)
plt.show()

# --- Graphic 2: Metrics Comparison (per dataset) ---
metrics      = ['CV Accuracy', 'Accuracy', 'Precision', 'Recall', 'Specificity',
'F1-Score']
model_names  = list(models.keys())
x, width     = np.arange(len(metrics)), 0.2

fig, axes = plt.subplots(1, 2, figsize=(18, 6))
fig.suptitle('Model Performance Comparison', fontsize=14, fontweight='bold')
for i, name in enumerate(model_names):
    axes[0].bar(x + i * width, [results_full_df.loc[name, m]   for m in metrics],
width, label=name)
    axes[1].bar(x + i * width, [results_simple_df.loc[name, m] for m in metrics],
width, label=name)
for ax, title in zip(axes, ['Full Dataset', 'Simplified Dataset']):
    ax.set_title(title)
    ax.set_xticks(x + width * 1.5)
    ax.set_xticklabels(metrics, rotation=15)
    ax.set_ylim(0.7, 1.05)
    ax.set_ylabel('Score')
    ax.legend()
    ax.axhline(y=1.0, color='grey', linestyle='--', linewidth=0.8)
plt.tight_layout()
plt.savefig('metrics_comparison.png', dpi=150)
plt.show()

# --- Graphic 3: Full vs Simplified per Model ---
fig, axes = plt.subplots(2, 2, figsize=(16, 12))
fig.suptitle('Full vs Simplified Dataset — Model Performance Comparison',
fontsize=14, fontweight='bold')
width = 0.35
for idx, name in enumerate(model_names):
    ax = axes[idx // 2, idx % 2]
```

```
    ax.bar(x - width/2, [results_full_df.loc[name, m]   for m in metrics], width,
label='Full', color='steelblue')
    ax.bar(x + width/2, [results_simple_df.loc[name, m] for m in metrics], width,
label='Simplified', color='coral')
    ax.set_title(name, fontsize=13, fontweight='bold')
    ax.set_xticks(x)
    ax.set_xticklabels(metrics, rotation=15)
    ax.set_ylim(0.7, 1.05)
    ax.set_ylabel('Score')
    ax.legend()
    ax.axhline(y=1.0, color='grey', linestyle='--', linewidth=0.8)
plt.tight_layout()
plt.savefig('full_vs_simplified.png', dpi=150)
plt.show()
```