RWANDA POLYTECHNIC

HUYE COLLEGE

Heart Disease Prediction System Report

ICT Department

Information Technology

Academic Year: 2025-2026

Level: Level 8, Year 4, B-tech

Course: **ITLML 801**

`

Augustin KALISA 25RP21655

Date: **16th January 2026**

1. Introduction

Heart disease is one of the leading causes of mortality worldwide. Early and accurate diagnosis can significantly reduce fatal outcomes. This project focuses on building a machine learning-based heart disease prediction system capable of classifying patients into five severity levels.

2. Objectives

- ✓ Perform exploratory data analysis
- ✓ Preprocess numerical and categorical features
- ✓ Train and tune multiple ML models
- ✓ Select the best-performing model
- ✓ Deploy the model using Flask

3. Dataset Description

The dataset contains 13 clinical features including age, sex, chest pain type, cholesterol, resting blood pressure, and others. The target variable consists of five heart disease severity classes.

4. Data Preprocessing

- ✓ Missing values handled using mean/median for numerical features
- ✓ Categorical features encoded using OneHotEncoder
- ✓ StandardScaler applied to numerical features
- ✓ Stratified 80/20 train-test split

5. Model Training

The following models were trained and tuned using GridSearchCV:

- ✓ Random Forest
- ✓ Gradient Boosting
- ✓ Support Vector Machine
- ✓ K-Nearest Neighbors
- ✓ Multi-Layer Perceptron

6. Model Evaluation

Random Forest achieved the best test accuracy of **99.9%** and train accuracy of **100%** was selected as the final model as best-fit status.

7. Model Serialization

The trained model and scaler were saved as "**heart_disease_best_model.pkl**", "**feature_columns.txt**" and "**class_names.txt**". These files are loaded by the Flask application to ensure prediction consistency.

8. Deployment

The final model was deployed using Flask with REST API with endpoints for prediction, feature listing, and health checks.

A Flask REST API was created with endpoints to:

- ✓ Expose model information
- ✓ Accept patient data through '/api/predict'
- ✓ Return predicted class, confidence score, and per-class probabilities

9. Web Interface

A single-page HTML interface allows medical staff to input patient data. The interface sends data to the API using JavaScript (fetch API) and displays results dynamically with color-coded risk levels.

Conclusion

The system demonstrates high accuracy and robustness, making it suitable for clinical decision-support applications for heart disease risk assessment.