RWANDA POLYTECHNIC

HUYE COLLEGE

SUMMATIVE OF MACHINE LEARNING(ITLML801)

ICT Department

Information Technology

Academic Year: 2025-2026

Level: Level 8, Year 4, B-tech

Names: NIYIKORIBITANGAZA Nicolas

Regno:25RP18183

16 January 2026

HEART DISEASE RISK PREDICTION SYSTEM

1. PROJECT OVERVIEW

This project implements an end-to-end Heart Disease Risk Prediction System for CHUB hospital. The system uses machine learning to predict a patient's heart disease risk level based on 13 clinical, demographic, and diagnostic features.

The prediction target has five classes:
1. No Disease, 2. Very Mild 3. Mild, 4. Severe, 5. Immediate Danger

The system includes data analysis, preprocessing, model training, evaluation, deployment using Flask, and a web-based frontend.

2. PROJECT STRUCTURE

```
ITML_801_S_A_25RP18183/
├── README.md
├── training_25RP18183.ipynb
├── app_25RP18183.py
├── templates/
│   └── index_25RP18183.html
├── deployment/
│   ├── heart_disease_model_25RP18183.pkl
│   ├── feature_columns.txt
│   └── class_names.txt
├── requirements.txt
├── project_report.pdf
└── demo_video.mp4
```

3. VIRTUAL ENVIRONMENT

All work was done inside a dedicated Python virtual environment named:

ITML_801_S_A_25RP18183

The environment contains:

- All required libraries
- The full codebase
- The Flask API file called app_25RP18183.py
- The frontend HTML file called index_25RP18183.html
- The deployment folder with model artifacts

DEPLOYMENT FOLDER

This Folder Contains all files required to run the model in production without retraining. **Files**

File	**Description**
heart_disease_model_25RP18183.pkl	Trained ML model
feature_columns.txt	Model input feature order
class_names.txt	Output class labels

-

4. DATA LOADING

Implemented in:

- training_25RP18183.ipynb This section includes:

- Loading the heart disease dataset
- Displaying the total number of samples
- Displaying the total number of features
- Displaying the first five records
- Displaying the total number of missing values

5. EXPLORATORY DATA ANALYSIS

Implemented in:

- training_25RP18183.ipynb Includes:

➢ Dataset shape and data types

➢ Detailed dataset information

➢ Descriptive statistics for numerical features

➢ Class distribution and class imbalance analysis

➢ **Visualizations:**

  - Bar plot of class distribution
  - Correlation heatmap for numerical features
  - Box plot of age vs heart disease class
  - Box plot of cholesterol vs heart disease class
  - Missing values analysis

6. DATA PREPROCESSING

Implemented in:

- training_25RP18183.ipynb Includes:
- Separation of features and target variable
- Train-test split (80/20) with stratification **Numerical preprocessing:**

- Missing value imputation
- Feature scaling using Standard Scaler

Categorical preprocessing:

Missing value imputation

One-Hot-Encoding with unknown handling

Combination of preprocessing pipelines using Column Transformer

Verification that no missing values remain after preprocessing

7. MODEL TRAINING AND EVALUATION

Implemented in:

- training_25RP18183.ipynb Includes:
- **Training multiple models:**

MLP

Random Forest

SVM

KNN

Gradient Boosting

➢ Hyperparameter tuning using GridSearchCV

➢ Model comparison table

➢ Overfitting and underfitting analysis

➢ Selection of the best-performing model

- **Detailed evaluation:**

➢ Classification report

➢ Confusion matrix

- Feature importance analysis

- **8. Model Saving and Verification**

- The best-performing model is saved in the deployment folder
- Feature names are saved in feature_columns.txt
- Class names are saved in class_names.txt
- The saved model is reloaded and verified using test samples and custom patient inputs

9. Flask API Implementation Implemented

in:

- app_25RP18183.py

Endpoints:

Endpoint	**Method**
/	GET
/api/health	GET
/api/predict	POST

Includes:

- Input validation
- Error handling
- Probability distribution output
- Risk level classification

10. FRONTEND IMPLEMENTATION

Implemented in:

- templates/index_25RP18183.html Features:
- Complete input form for 13 patient features
- Color-coded prediction results
- Confidence score and class probability distribution
- Responsive design for different screen sizes

11. HOW TO RUN THE PROJECT

1. Activate the virtual environment
2. Install dependencies:

```
pip install -r requirements.txt
```

3. Run the Flask application:

```
python app_25RP18183.py
```

4. Open a browser and visit:

http://localhost:5000