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. INTRODUCTION

This project implements an end-to-end **Heart Disease Risk Prediction System** designed to support clinical decision-making at CHUB hospital. The system predicts a patient's heart disease risk level using routinely collected clinical and demographic data.

The objective of this work was to:

- Explore and analyze a heart disease dataset
- Train and evaluate multiple machine learning models
- Select the best-performing model
- Deploy the model using a Flask REST API
- Connect the API to a web-based frontend interface

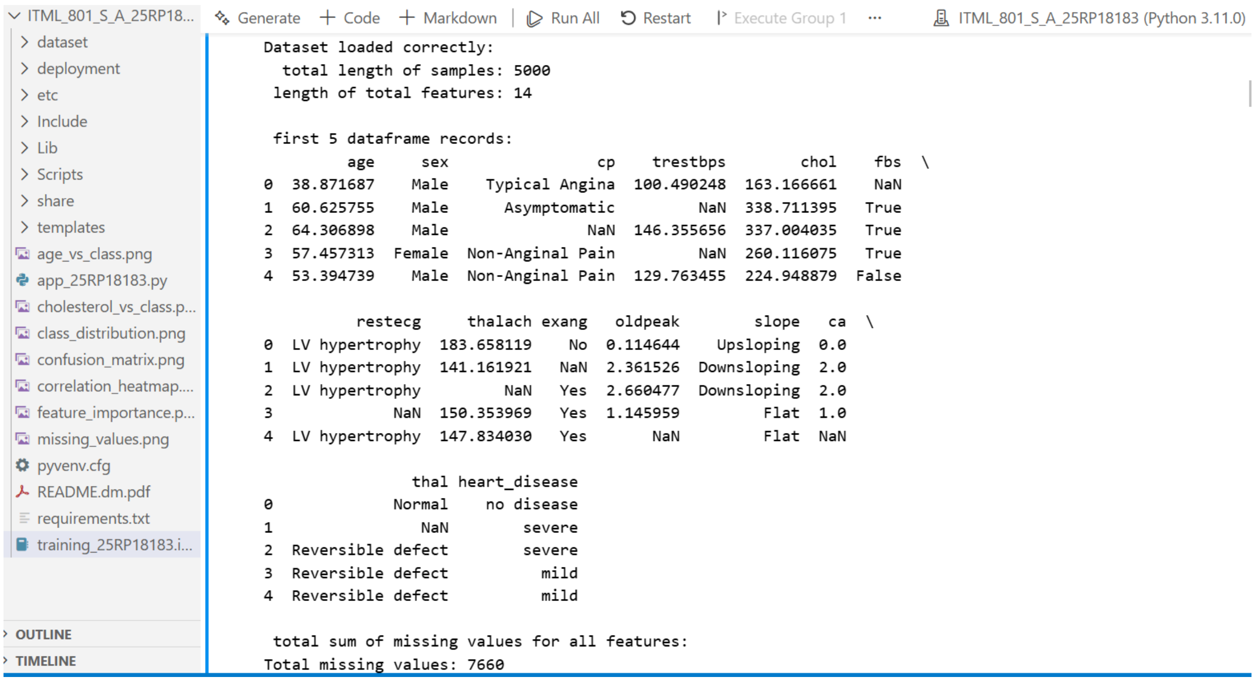
2. DATASET DESCRIPTION

The dataset consists of **5,000 patient records**, each described by **13 clinical features** including age, blood pressure, cholesterol, ECG results, exercise test outcomes, and other diagnostic indicators.

The target variable contains five heart disease risk classes:

- No Disease
- Very Mild
- Mild
- Severe
- Immediate Danger

The dataset was loaded and inspected using Python in the file training_25RP18183.ipynb.



3. Exploratory Data Analysis (EDA)

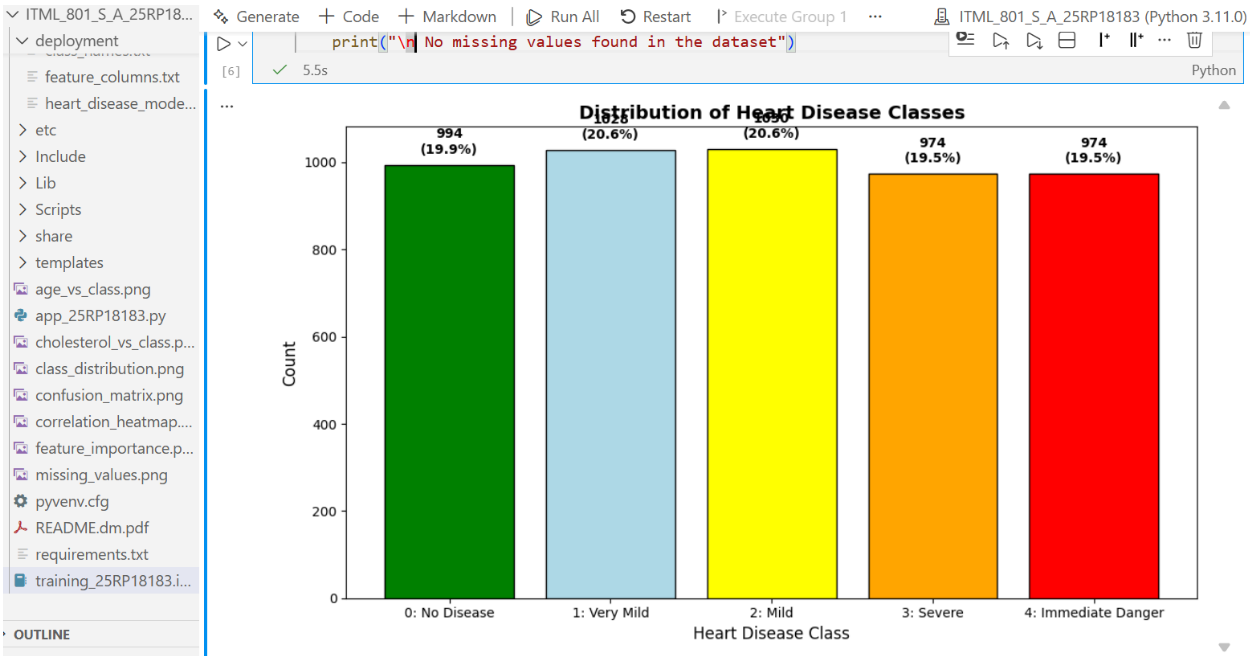
Exploratory Data Analysis was performed to understand the structure and characteristics of the dataset. This included:

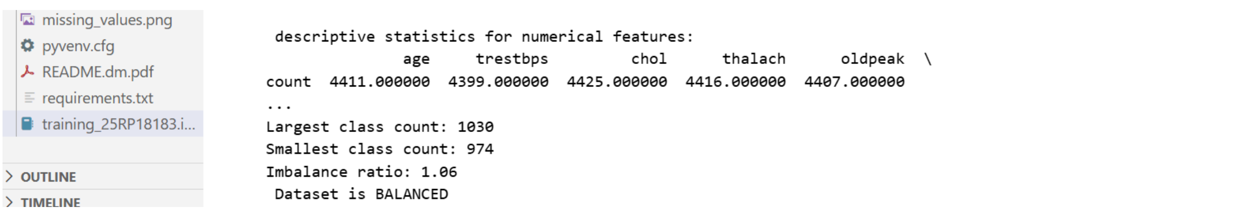
- ➢ Displaying dataset shape, data types, and summary statistics
- ➢ Analyzing class distribution and balance
- ➢ Checking missing values
- ➢ Visualizing feature relationships

The following visualizations were generated:

- Bar chart of heart disease class distribution
- Correlation heatmap for numerical features
- Box plots comparing age and cholesterol across disease classes

These analyses helped identify feature patterns and confirmed that stratified sampling was required.






```
 descriptive statistics for numerical features:
                 age     trestbps         chol      thalach      oldpeak  \
count  4411.000000  4399.000000  4425.000000  4416.000000  4407.000000
...
Largest class count: 1030
Smallest class count: 974
Imbalance ratio: 1.06
 Dataset is BALANCED
```

4. Data Preprocessing

The dataset was split into training and testing sets using an **80/20 stratified split** to preserve class balance.

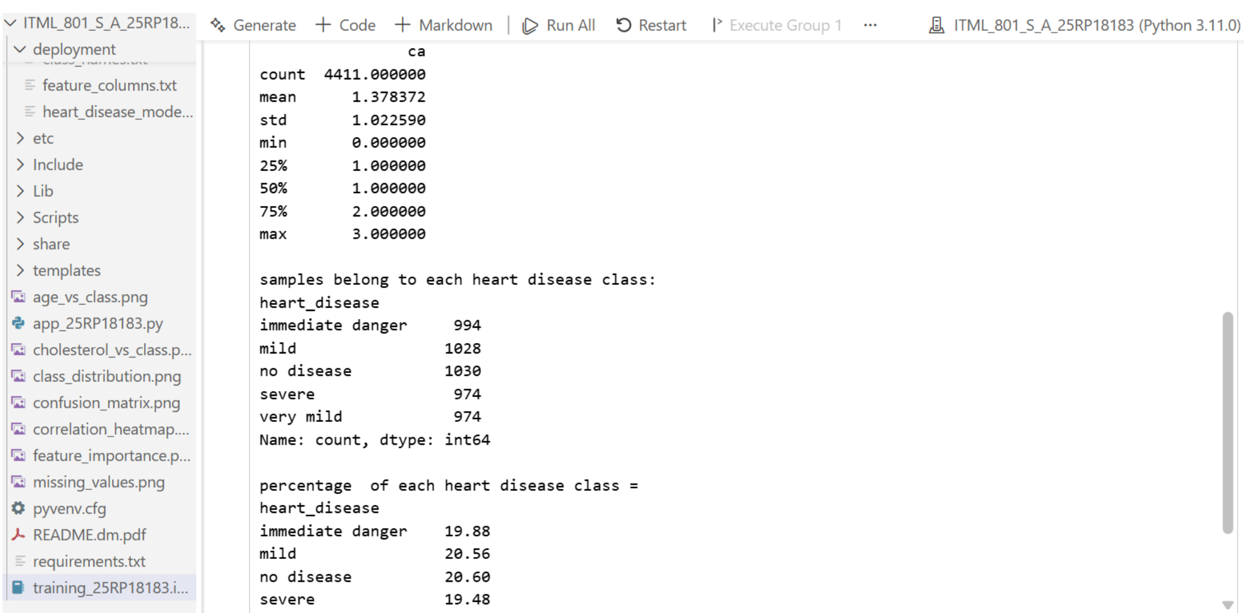
Preprocessing pipelines were created as follows:

- **Numerical features:** missing value imputation and standard scaling
- **Categorical features:** most frequent imputation and one-hot encoding

A Column Transformer was used to combine preprocessing steps.
After preprocessing:

- No missing values remained
- All features were numeric
- Training and testing shapes were verified

```
1(a) Train-test split completed (80/20, random_state=42, stratified)
1(b) Training samples: 4000 (80.0%)
     Testing samples: 1000 (20.0%)

1(c) Stratification Verification Table:
                  Original (%)  Train (%)  Test (%)
heart_disease
no disease               20.60     20.600      20.6
mild                     20.56     20.575      20.5
immediate danger         19.88     19.875      19.9
severe                   19.48     19.475      19.5
very mild                19.48     19.475      19.5

2(a) Numerical features: ['age', 'trestbps', 'chol', 'thalach', 'oldpeak', 'ca']
2(b) Number of numerical features: 6

3(a) Categorical features: ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'thal']
3(b) Number of categorical features: 7

5(a) Transformed training set shape: (4000, 25)
5(b) Transformed testing set shape: (1000, 25)

6(a) Training set contains no missing values: True
6(b) Testing set contains no missing values: True
6(c) All transformed features are numeric: True
```



5. Model Training and Evaluation

Multiple machine learning models were trained and tuned using **GridSearchCV**, including:

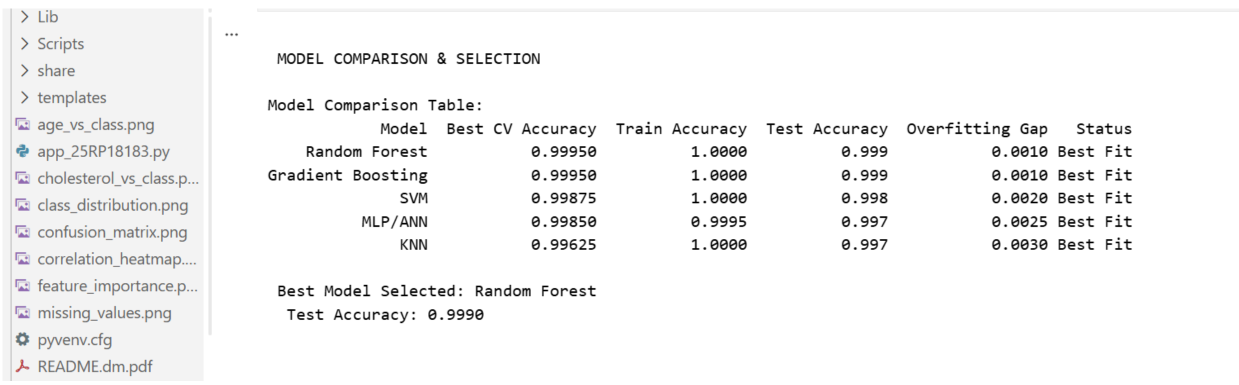
- Artificial Neural Network (MLP)

- Random Forest
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Gradient Boosting

For each model:

- Best hyperparameters were identified
- Cross-validation accuracy was recorded
- Training and testing accuracy were compared
- Overfitting gaps were analyzed

A comparison table was created to select the best-performing model based on test accuracy and generalization performance.

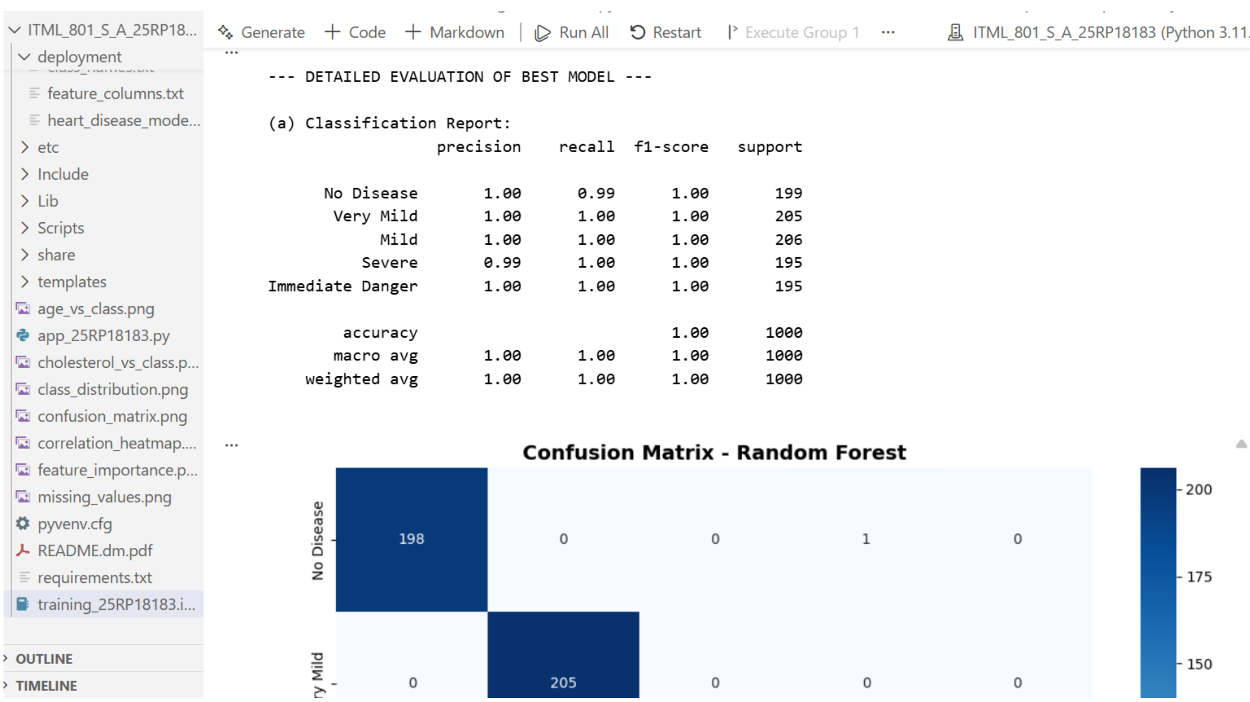



```
MODEL COMPARISON & SELECTION

Model Comparison Table:
            Model  Best CV Accuracy  Train Accuracy  Test Accuracy  Overfitting Gap   Status
    Random Forest           0.99950          1.0000          0.999           0.0010 Best Fit
Gradient Boosting           0.99950          1.0000          0.999           0.0010 Best Fit
              SVM           0.99875          1.0000          0.998           0.0020 Best Fit
          MLP/ANN           0.99850          0.9995          0.997           0.0025 Best Fit
              KNN           0.99625          1.0000          0.997           0.0030 Best Fit

 Best Model Selected: Random Forest
  Test Accuracy: 0.9990
```

6. Best Model Analysis

The Random Forest model was selected as the best-performing model and was evaluated comprehensively:

- ➢ Classification Report: A detailed report was generated for all five classes, providing precision, recall, and F1-scores.
- ➢ Confusion Matrix: The confusion matrix was computed and visualized to assess misclassifications and overall accuracy.
- ➢ Per-Class Interpretation: Clinical interpretation was performed for each class to understand the model's predictions in the context of disease severity.
- ➢ Feature Importance: Features contributing most to predictions were analyzed, highlighting key factors influencing model decisions



7. Model Saving and Verification

The final trained model, including preprocessing steps, was saved in the deployment/ directory along with:

- Feature column names
- Class names

Two verification steps were performed:

1. Predictions on random test samples
2. Predictions on custom patient inputs

Both verification steps confirmed consistent and correct model behavior after reloading.

```
(a) Model saved: deployment/heart_disease_model_25RP18183.pkl
(b) Feature columns saved: deployment/feature_columns.txt
(c) Class names saved: deployment/class_names.txt

(d) Verification 1 - Random Test Samples:
 Sample     Actual Class  Predicted Class  Match
      1             mild             mild   True
      2 immediate danger immediate danger   True
      3        very mild        very mild   True
      4       no disease       no disease   True
      5       no disease       no disease   True

(e) Verification 2 - Custom Patient Samples:
 Sample  Predicted Class Class Name  Confidence  P(No Disease)  P(Very Mild)  P(Mild)  P(Severe)  P(Immedi
      1             mild  Very Mild        1.00           0.00           1.0      0.0       0.00
      2           severe     Severe        0.99           0.01           0.0      0.0       0.99
      3 immediate danger No Disease        1.00           1.00           0.0      0.0       0.00
```

8. Flask API Development

The trained model was deployed using **Flask**.
The API provides:

- A health check endpoint
- A /api/predict endpoint for predictions

The API performs:

- Input validation
- Error handling
- Probability estimation for each class

The application runs successfully using:

python app_25RP18183.py

```
(ITML_801_S_A_25RP18183) PS C:\Users\PC\OneDrive\Desktop\25RP18183\ITML_801_S_A_25RP18183> python app_25RP181
83.py
Model loaded successfully from: C:\Users\PC\OneDrive\Desktop\25RP18183\ITML_801_S_A_25RP18183\deployment\hear
t_disease_model_25RP18183.pkl
13 feature columns loaded
Class names loaded: ['No Disease', 'Very Mild', 'Mild', 'Severe', 'Immediate Danger']
=== HEART DISEASE RISK PREDICTION API RUNNING ===
```

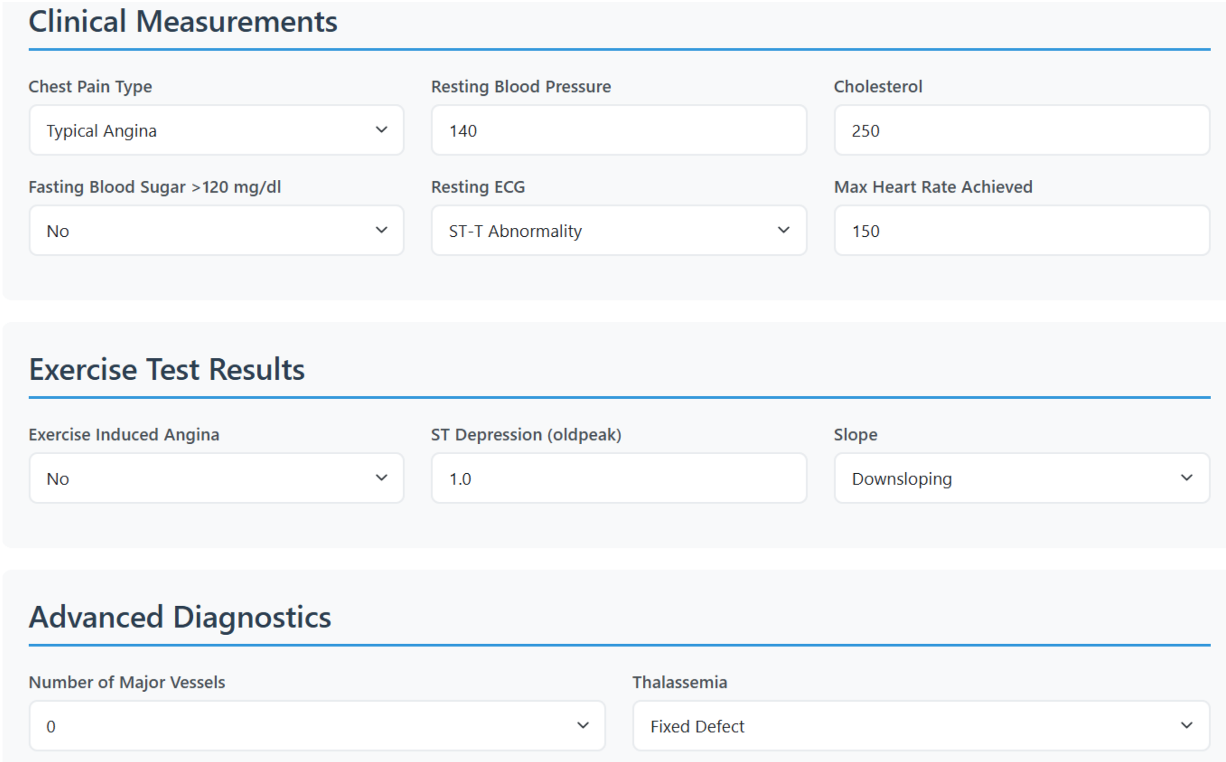
```
(ITML_801_S_A_25RP18183) PS C:\Users\PC\OneDrive\Desktop\25RP18183\ITML_801_S_A_25RP18183> python app_25RP181
83.py

Patient Input:
age: 55
sex: Male
cp: Typical Angina
trestbps: 140
chol: 250
fbs: No
restecg: ST-T Abnormality
thalach: 150
exang: No
oldpeak: 1.0
slope: Downsloping
ca: 0 vessels
thal: Fixed Defect
Predicted Class: Mild (46.0%)
Class Probabilities:
  No Disease: 2.0%, Very Mild: 12.0%, Mild: 46.0%, Severe: 8.0%, Immediate Danger: 32.0%
```

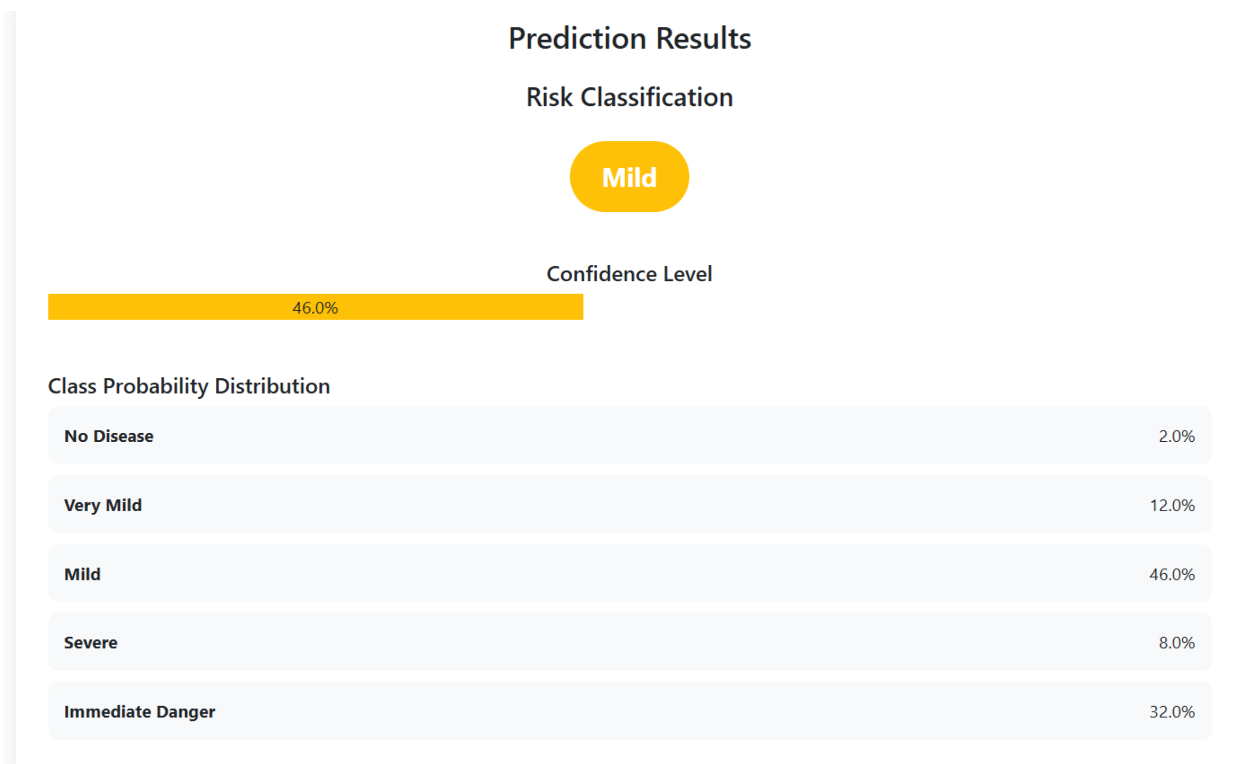
9. Frontend Interface

A responsive HTML frontend was developed to allow medical staff to:

- Enter patient data (13 features)
- Submit data to the API
- View predicted risk level, confidence, and class probabilities

Predicted risk levels are displayed using color-coded indicators for clarity.





10. Conclusion

This project successfully delivered a complete heart disease risk prediction system, from data analysis and model training to deployment and user interaction. The system demonstrates how machine learning can support clinical decision-making through accurate predictions and accessible interfaces.