Python For Data Science (3150713)

Heart Risk Detection System

(*Course-Integrated Practical & Analytical Task (CIPAT)*)

By

Kartik Sahani (230090107058)

Lalitkumar Sharma (230090107075)

Devarsh Mangroliya (230090107084)

Guided By

Prof. Mithila D. Parekh

Assistant Professor

DEPARTMENT OF COMPUTER ENGINEERING

C. K. Pithawala College of Engineering and Technology, Surat

Gujarat Technological University, Ahmedabad

Academic Year: 2025-2026

C. K. Pithawala College of Engineering and Technology

Opposite Surat Airport, Behind DPS School, Near Malvan Mandir, Dumas Road, Surat

CERTIFICATE

This is to certify that project work embodied in this report entitled **project name** was carried out by **studentname1 (Enrollment no), studentname2 (Enrollment no)** at **C. K. Pithawala College of Engineering and Technology, Surat** as a part of **Course-Integrated Practical & Analytical Task (CIPAT)** for subject **Python For Data Science (3150713)**. This project work has been carried out under my supervision.

Date :

Place : Surat, Gujarat

Prof. Mithila D. Parekh

Subject Coordinator

C. K. Pithawala College of Engineering and Technology

Opposite Surat Airport, Behind DPS School, Near Malvan Mandir, Dumas Road, Surat

DECLARATION

We hereby certify that We are the sole author of this report and that neither any part of this work nor the whole of the work has been submitted for a degree to any other University or Institution.

We declare that this is a true copy of our report, including any final revisions, as approved by my supervisor.

Date : 22 June 2025

Place : Surat,Gujarat

Kartik Sahani (230090107058)

Lalitkumar Sharma

(230090107075)

Devarsh Mangroliya

(230090107084)

Acknowledgments

We would like to express our sincere gratitude to everyone who supported and guided us during the completion of this project.

First, we would like to thank Prof. Mithila D. Parekh for their valuable guidance, encouragement, and insightful feedback throughout the project. Their expertise helped us understand the concepts of machine learning, model evaluation, and data preprocessing, which were crucial for building an accurate heart disease prediction model.

We are also grateful to the developers and contributors of Python, scikit-learn, Pandas, Matplotlib, Seaborn, and Flask, whose open-source tools made it possible to preprocess data, train the predictive model, and create an interactive and visually appealing dashboard.

Our thanks also go to the creators of the UCI Heart Disease Dataset, which served as the foundation for training and testing our model.

Finally, we acknowledge the efforts of all group members for their active collaboration, constructive discussions, and contribution to model development, data analysis, visualization, and dynamic input implementation, which collectively made this project successful.

Table of Contents

List of Figures

Heart Risk Detection System

Submitted By

Kartik Sahani (230090107058)

Lalitkumar Sharma (230090107075)

Devarsh Mangroliya (230090107084)

Supervised By

Prof. Mithila D. Parekh

Subject Coordinator

Abstract

Heart disease is one of the leading causes of mortality worldwide, and early detection is crucial for effective treatment and prevention. This project presents a machine learning-based approach to predict the likelihood of heart disease using patient data from the UCI Heart Disease Dataset. A HistGradientBoostingClassifier model was trained on 13 key features, including demographic, clinical, and laboratory measurements, to accurately classify patients with or without heart disease. The project incorporates data preprocessing, such as handling missing values with mean imputation, and provides dynamic input capabilities, allowing users to enter patient data in real-time and obtain predictions along with probability scores. Additionally, the model's performance is visualized through Confusion Matrix, ROC Curve, Precision-Recall Curve, and F1-Score bar charts, providing insights into classification accuracy and reliability. This system serves as an interactive, user-friendly tool for understanding heart disease risk and demonstrates the effective application of machine learning in healthcare analytics.

1 Project Introduction

Heart disease remains one of the leading causes of mortality globally, accounting for a significant proportion of deaths each year. Early detection and timely intervention are critical for improving patient outcomes and reducing healthcare costs. With the rapid growth of healthcare data and advances in computational methods, machine learning has emerged as a powerful tool for predicting disease risk, assisting medical professionals in decision-making, and enhancing preventive healthcare strategies.

The primary motivation behind this project is to leverage machine learning techniques to develop an accurate, reliable, and user-friendly predictive system for heart disease. Traditional diagnostic methods often involve complex procedures, costly tests, and time-consuming evaluations. By using patient data, such as demographic details, clinical measurements, and laboratory test results, a predictive model can provide early warnings and help prioritize high-risk patients.

The problem statement addressed by this project is: given a set of patient features, predict whether the patient is at risk of heart disease and estimate the probability of occurrence. To solve this problem, we used the UCI Heart Disease Dataset, which contains 303 patient records with 13 key attributes, including age, sex, chest pain type, resting blood sugar, cholesterol, maximum heart rate, and other clinical indicators.

The main objectives of this project are:

1. To preprocess and clean the dataset by handling missing values and ensuring correct data types.
2. To train a machine learning model capable of accurately predicting heart disease, using the HistGradientBoostingClassifier for its robust performance on structured data.
3. To develop a dynamic input system that allows users to input patient data in real-time and receive prediction results with probability scores.
4. To evaluate the model's performance using multiple metrics, including accuracy, F1-score, ROC Curve, Precision-Recall Curve, and to provide visual insights through interpretable charts.
5. To design a simple, interactive interface (Flask-based) that combines predictions with performance visualizations, offering a comprehensive dashboard for users.

This project not only demonstrates the practical application of machine learning in healthcare but also emphasizes the importance of interpretable predictions and user accessibility, bridging the gap between technical analysis and real-world clinical usage. Through this system, healthcare practitioners can identify high-risk patients efficiently, make informed decisions, and ultimately contribute to better patient care and preventive strategies.

2 Literature Review / Background

Background

Heart disease, also known as cardiovascular disease (CVD), encompasses a range of conditions affecting the heart and blood vessels, including coronary artery disease, arrhythmia, heart failure, and hypertension. Globally, cardiovascular diseases are among the leading causes of morbidity and mortality, accounting for millions of deaths annually. Early detection and accurate risk assessment are essential for preventive care, reducing mortality rates, and optimizing treatment strategies.

With the growth of healthcare data, predictive analytics and machine learning have become increasingly important in medical diagnosis. Traditional methods of heart disease diagnosis, such as electrocardiograms (ECG), stress tests, and angiography, although effective, are often time-consuming, costly, and sometimes invasive. Machine learning provides an opportunity to leverage patient demographic, clinical, and laboratory data to develop predictive models that can identify high-risk patients efficiently and support healthcare practitioners in decision-making.

The UCI Heart Disease Dataset, widely used in research, contains records of patients with various attributes including age, sex, chest pain type, resting blood sugar, cholesterol levels, maximum heart rate, exercise-induced angina, and other clinical indicators. This dataset has served as a benchmark for evaluating classification models aimed at predicting the presence or absence of heart disease.

Literature Review

Several studies have explored the use of machine learning for heart disease prediction, employing different algorithms and data preprocessing techniques. Some key contributions include:

1. **Decision Trees and Random Forests:**
    - Decision Trees have been widely used due to their interpretability and ability to handle non-linear relationships.
    - Random Forests, an ensemble of Decision Trees, have demonstrated high predictive accuracy while reducing overfitting. Studies show that Random Forest models often outperform individual Decision Trees in predicting heart disease risk.

2. **Support Vector Machines (SVM):**
    - SVMs have been applied successfully to heart disease datasets due to their capability to handle high-dimensional data and find optimal separating hyperplanes.

- Researchers have reported that SVMs achieve competitive performance, especially when combined with kernel functions that capture non-linear patterns in patient features.

3. **Gradient Boosting Methods:**
   - Gradient Boosting, including HistGradientBoostingClassifier and XGBoost, is effective for structured tabular data.
   - Literature indicates that boosting methods consistently achieve high accuracy for heart disease prediction due to their ability to combine weak learners and reduce bias.
   - They also provide feature importance scores, which are useful for identifying significant clinical factors influencing predictions.

4. **Neural Networks:**
   - Multilayer Perceptrons (MLPs) and deep learning approaches have been applied to predict heart disease from patient records.
   - While neural networks can model complex non-linear relationships, they require larger datasets and often lack interpretability compared to tree-based methods.

5. **Data Preprocessing and Feature Engineering:**
   - Studies emphasize the importance of handling missing values, standardizing numerical features, and encoding categorical variables.
   - Feature selection and dimensionality reduction techniques, such as PCA or mutual information analysis, help improve model performance and reduce overfitting.

6. **Evaluation Metrics and Visualization:**
   - Researchers commonly use accuracy, precision, recall, F1-score, ROC-AUC, and Precision-Recall curves to evaluate predictive models.
   - Visualization of model performance, such as confusion matrices and ROC curves, aids interpretability and helps clinicians trust model predictions.

Relevance to This Project

Building on existing research, this project uses the **HistGradientBoostingClassifier** for its efficiency and high accuracy on tabular data. The project incorporates:

- **Dynamic input:** allowing real-time patient data entry.
- **Model visualizations:** Confusion Matrix, ROC Curve, Precision-Recall Curve, and F1-score charts.
- **Data preprocessing:** mean imputation for missing values and numeric conversion of object columns.

By integrating these approaches, the project not only replicates the predictive accuracy reported in literature but also provides an interactive, interpretable, and user-friendly system for healthcare practitioners. This combination of machine learning prediction and visual explanation addresses the gap between theoretical model performance and practical clinical applicability.

3 Problem Definition

Cardiovascular diseases, particularly heart disease, are among the leading causes of death worldwide. Early detection and accurate risk assessment are critical for effective treatment, timely intervention, and preventive healthcare. Traditional diagnostic methods, such as electrocardiograms, stress tests, and angiography, although effective, are often time-consuming, expensive, and sometimes invasive. Additionally, the interpretation of multiple clinical parameters simultaneously can be challenging for healthcare professionals, leading to delays or potential inaccuracies in diagnosis.
The central problem addressed by this project is:
“Given a set of patient-specific demographic, clinical, and laboratory features, how can we accurately predict the presence or absence of heart disease and estimate the probability of occurrence, in a way that is interpretable, accessible, and actionable for healthcare practitioners?”

Key challenges include:

1. **Handling heterogeneous data:** Patient datasets contain both numeric and categorical features, with occasional missing values.
2. **Model accuracy:** The predictive model must be reliable enough to assist in real-world decision-making.
3. **User accessibility:** Clinicians and users should be able to input patient data dynamically and receive clear predictions without requiring advanced technical knowledge.
4. **Interpretability and visualization:** The system should provide insights into model performance and prediction reliability through visualizations like confusion matrices, ROC curves, and F1-score charts.

4 Objective

The primary goal of this project is to develop an accurate, interpretable, and user-friendly machine learning system for predicting heart disease risk. To achieve this, the project is structured around the following detailed objectives:

1. Data Preprocessing and Cleaning

- Importance: Raw medical datasets often contain missing values, inconsistent data types, or noisy information, which can reduce model accuracy. Proper preprocessing is essential to ensure reliable predictions.
- Implementation: Missing numeric values are handled using mean imputation, and categorical/object features (such as `Resting bs` and `chol`) are converted into numeric values for model compatibility.
- Outcome: A clean, structured dataset that preserves meaningful patterns in the patient data, suitable for training the machine learning model.

2. Feature Selection and Understanding

- Importance: Not all features contribute equally to the prediction. Identifying the most relevant features improves model performance, reduces overfitting, and helps interpret results.
- Implementation: All 13 key features, including demographic (Age, Sex), clinical (Chest Pain, Exercise Induced Angina), and laboratory measurements (Cholesterol, Max Heart Rate, Thal), are considered. Feature importance can be visualized to understand the impact of each attribute.
- Outcome: A robust feature set that maximizes predictive power while maintaining interpretability for medical practitioners.

3. Model Development

- Importance: Accurate prediction is the core goal of the project. The model must capture non-linear relationships between patient attributes and heart disease risk.
- Implementation: The HistGradientBoostingClassifier is selected due to its efficiency on tabular data, robustness to missing values, and strong performance in classification tasks.
- Outcome: A predictive model capable of classifying patients with high accuracy and reliability.

4. Dynamic Input for Real-Time Prediction

- Importance: Clinicians or users need the ability to input patient data dynamically and get immediate predictions. Static predictions on a fixed dataset are insufficient for practical applications.
- Implementation: A dynamic input system is implemented using Python in notebooks (or later via Flask) to allow real-time data entry. The system validates inputs, handles numeric and categorical features, and outputs prediction and probability.

- Outcome: A user-friendly interface that enables real-time heart disease risk assessment for individual patients.

5. Model Evaluation and Visualization

- Importance: It is crucial to evaluate the model not only through metrics but also via visual insights for interpretability and trustworthiness.
- Implementation: Multiple evaluation metrics are used, including accuracy, confusion matrix, F1-score, ROC curve, Precision-Recall curve, and probability scores. Visualizations are generated using Matplotlib and Seaborn, providing insights into model performance and reliability.
- Outcome: Clear understanding of model strengths and weaknesses, and confidence in predictions for practical usage.

6. User-Friendly Dashboard Integration

- Importance: A prediction model alone is not sufficient for practical deployment. The system must be accessible to end-users with minimal technical knowledge.
- Implementation: Using Flask, the project integrates the predictive model and visualizations into a modern, interactive web dashboard. Users can enter patient data, view predictions, and explore model evaluation charts in a single interface.
- Outcome: A simple and practical, appealing platform that demonstrates the application of machine learning in healthcare.

By achieving these objectives, the project combines accurate predictive modeling, dynamic input, and interpretability, providing a comprehensive system for early heart disease detection. It emphasizes the practical application of machine learning in healthcare, bridging the gap between technical model performance and real-world usability.

5 Methodology / System Design

The methodology of this project outlines a structured approach to building a machine learning-based heart disease prediction system with dynamic input and visualizations. The system is designed to be accurate, interpretable, and user-friendly, integrating all stages from data preprocessing to deployment.

1. Data Collection

- **Source:** The project uses the UCI Heart Disease Dataset, which contains 303 patient records and 13 features related to demographic, clinical, and laboratory parameters.

- **Features:**
    - **Demographic**: Age, Sex
    - **Clinical:** Chest Pain Type, Resting Blood Sugar, Exercise Induced Angina, Resting ECG, Slope of Peak Exercise, Number of Major Vessels
    - **Laboratory**: Cholesterol, Max Heart Rate, Thal, Oldpeak

- **Target:** Diagnosis (presence or absence of heart disease)

Objective: Use a well-known, validated dataset to ensure reliable training and evaluation.

2. Data Preprocessing

Proper data preprocessing is essential for high model accuracy and reliability.

- **Handling Missing Values:**
    - Numeric missing values are imputed using mean imputation via SimpleImputer.
    - Object/categorical columns such as Resting bs and chol are converted to numeric values.

- **Data Cleaning:**
    - Ensure all features have consistent data types.
    - Remove any anomalies or duplicate records if present.

- **Feature Selection:**
    - All 13 features are retained as they are clinically relevant.
    - Feature importance can be visualized later to highlight significant predictors.

Outcome: A clean, structured dataset suitable for training the machine learning model.

3. Model Development

- **Algorithm: HistGradientBoostingClassifier**
    - Chosen for its robust performance on tabular data, handling of missing values, and ability to model non-linear relationships.
    - Ensemble gradient boosting improves accuracy by combining weak learners.

- **Data Splitting:**
    - The dataset was split into 80% training and 20% testing to evaluate generalization performance.

- **Training:**
    - The model is trained on the preprocessed training data using all 13 features.
    - Hyperparameters can be tuned to optimize performance.

Outcome: A trained predictive model capable of classifying patients with high accuracy.

4. Dynamic Input System

- **Objective:** Allow users to enter patient data dynamically and receive predictions.

- **Implementation:**
    - In a notebook or web interface (Flask), inputs are collected for all 13 features.
    - Numeric and categorical data are validated and converted appropriately.
    - Missing or incorrect inputs are handled via imputation.

- **Output:** Prediction (Heart Disease / No Heart Disease) and probability score.

Benefit: Real-time prediction makes the system practical for clinical use.

5. Model Evaluation and Visualization

To ensure interpretability and trust in the model:

1. **Confusion Matrix:**
    - Shows true positives, true negatives, false positives, and false negatives.
    - Helps evaluate model accuracy and error types.

2. **Classification Report & F1-Score:**
    - Provides precision, recall, and F1-score per class.
    - Visualized as a bar chart for easier interpretation.

3. **ROC Curve:**
    - Plots True Positive Rate vs False Positive Rate.

- Evaluates model discrimination ability.

4. **Precision-Recall Curve:**
    - Visualizes trade-off between precision and recall, especially for imbalanced datasets.

Implementation: Visualizations are generated using Matplotlib and Seaborn, and can be embedded in dashboards for easy interpretation.

6. System Design and Architecture

The system is designed with a modular architecture for usability, scalability, and maintainability.

Components:

1. **Data Layer:**
    - Stores dataset and handles preprocessing and cleaning.

2. **Model Layer:**
    - Trains the HistGradientBoostingClassifier.
    - Handles prediction logic for dynamic inputs.

3. **Input Layer (Dynamic Input Interface):**
    - Collects patient features via web forms or notebooks.
    - Validates and preprocesses inputs before sending them to the model.

4. **Visualization Layer:**
    - Generates charts for model evaluation (Confusion Matrix, ROC, Precision-Recall, F1-Score).
    - Displays results in a dashboard format.

5. **Presentation Layer (Dashboard / Notebook Output):**
    - Combines predictions and visualizations.
    - Modern, clean interface using Bootstrap in Flask or interactive notebook outputs.

7. Workflow Diagram:

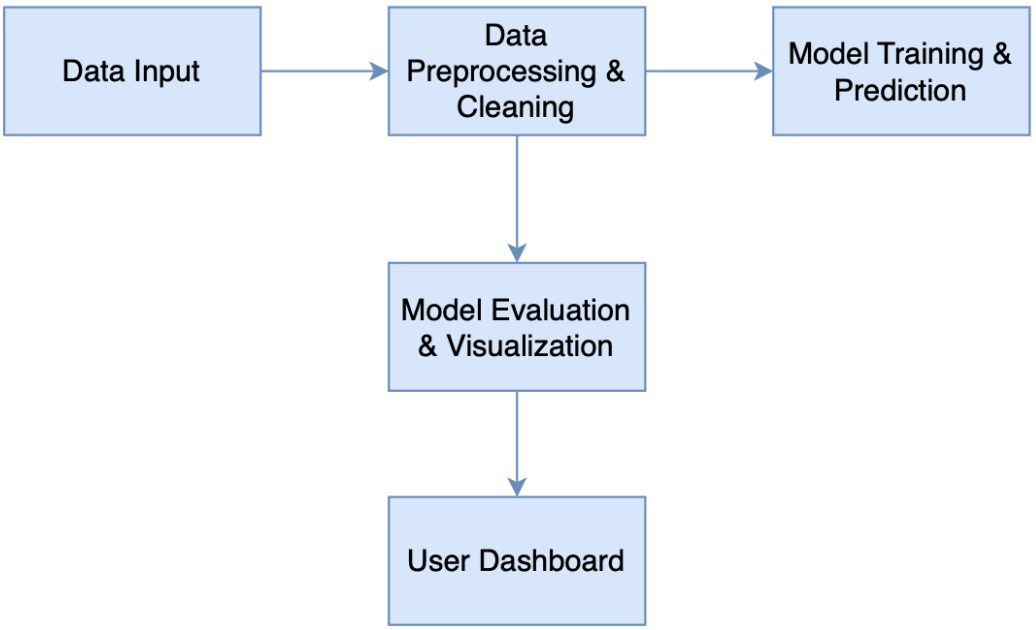

Fig 1 Workflow Diagram

8. Summary

The methodology ensures a comprehensive pipeline:

1. Collect and clean data from a reliable dataset.
2. Preprocess features to handle missing values and type mismatches.
3. Train a robust machine learning model on relevant features.
4. Enable dynamic input for real-time patient predictions.
5. Evaluate the model with multiple metrics and provide visual insights.
6. Integrate all components into a modern, user-friendly system for practical use.

This structured approach ensures the project achieves high accuracy, interpretability, and usability, bridging the gap between machine learning theory and real-world healthcare application.

6 IMPLEMENTATION

The implementation of the heart disease prediction system involves integrating data preprocessing, model training, dynamic input, prediction, and visualization into a cohesive workflow. The system is developed using Python, scikit-learn, Matplotlib, Seaborn, Pandas, and Flask for the web interface.

1. Dataset Loading and Preprocessing

1. **Loading Dataset:**
    - The UCI Heart Disease Dataset is loaded using pandas.read_csv().
    - The dataset contains 303 patient records with 13 input features and one target variable (Diagnosis).

2. **Data Cleaning and Handling Missing Values:**
    - Numeric features are imputed using mean imputation via SimpleImputer(strategy='mean').
    - Object/categorical features (Resting bs and chol) are converted to numeric using pd.to_numeric(errors='coerce').
    - The dataset is now consistent and ready for model training.

3. **Feature Selection:**
    - All 13 features are retained because each is clinically relevant and contributes to model prediction.
    - Features are categorized as:
        - Numeric: Age, Sex, Chest Pain(T), fasting bs, restecg, Max heart rate, Exercise induced angina, oldpeak, slope of peak exercise, number of major vessels, thal
        - Object/Categorical: Resting bs, chol

2. Model Training

1. **Algorithm Selection:**
    - The HistGradientBoostingClassifier is chosen for its efficiency on tabular datasets, ability to handle missing values, and strong classification performance.

2. **Training Process:**
    - The dataset was split into 80% training and 20% testing using train_test_split().
    - The model is trained on the training data using model.fit(X_train, y_train).

3. **Evaluation:**
    - Predictions are obtained for the test set using model.predict(X_test) and model.predict_proba(X_test).

- **Metrics calculated include:**
  - Accuracy (accuracy_score)
  - Confusion Matrix (confusion_matrix)
  - Precision, Recall, F1-Score (classification_report)
  - ROC-AUC and Precision-Recall curves

3. Dynamic Input System

1. **Objective:**
   - To allow users to input patient-specific data in real-time and obtain immediate predictions with probabilities.

2. **Implementation in Notebook:**
   - A Python dictionary collects inputs for each of the 13 features.
   - Numeric features are converted to float, while object features are converted to numeric.
   - Input data is wrapped into a DataFrame and preprocessed to match the format used in training.

3. **Prediction:**

   The model predicts the class (Heart Disease or No Heart Disease) and probability using:

   prediction = model.predict(input_df)[0]

   probability = model.predict_proba(input_df)[0][1]

   - Results are displayed directly in the notebook for user convenience.

4. Model Visualization Implementation

Visualizations are crucial for interpretability and trust in predictions.

1. **Confusion Matrix:**
   - Generated using sns.heatmap(confusion_matrix(y_test, y_pred)).
   - Shows the number of correct and incorrect predictions per class.

2. **F1-Score Bar Chart:**
   - classification_report(output_dict=True) is converted to a DataFrame.
   - The bar chart is plotted with F1-Score per class for visual comparison.

3. **ROC Curve:**
   - Generated using RocCurveDisplay.from_predictions(y_test, y_pred_prob).
   - Shows the model's discrimination ability.

4. **Precision-Recall Curve:**
   - Calculated using precision_recall_curve(y_test, y_pred_prob).
   - Plotted to evaluate trade-offs between precision and recall.

5. **Integration for Web Display:**
    - Matplotlib figures are converted to base64 images for embedding in a Flask web interface.

5. Flask Web Interface

1. **Frontend:**
    - Built using HTML, Bootstrap 5, and Jinja2 templates.
    - **Provides:**
        - Input form for 13 features
        - Real-time prediction display
        - Embedded model visualizations (Confusion Matrix, ROC Curve, Precision-Recall, F1-Score)

2. **Backend:**
    - **Flask routes handle:**
        - / – Home page displaying input form and visualizations.
        - /predict – Receives form data, preprocesses input, predicts using the trained model, and returns results.
    - Dynamic input is processed and converted to the required format before sending it to the model.

3. **Prediction and Display:**
    - The predicted class and probability are displayed in a card below the form.
    - Visualizations are displayed in cards next to the form, providing an interactive dashboard experience.

6. System Workflow

1. The user opens the web interface.
2. Inputs patient data (dynamic form) and submits.
3. Backend preprocesses input and sends it to the trained model.
4. Model returns prediction and probability.
5. Dashboard displays prediction, probability, and evaluation visualizations.

7. Advantages of Implementation

- **Interactive and User-Friendly:** Users can enter data dynamically.
- **Visual Interpretability:** Charts provide insights into model performance and reliability.
- **Robust Model:** HistGradientBoostingClassifier ensures accurate predictions on structured data.
- **Reusable and Scalable:** Flask backend can be extended to larger datasets or more features.

This implementation bridges the gap between a machine learning model and practical healthcare application by combining high predictive accuracy, real-time dynamic input capabilities, and comprehensive visualizations into a single, cohesive system. Unlike traditional diagnostic approaches, which can be time-consuming and rely heavily on manual interpretation, this system enables instant risk assessment for individual patients based on their specific demographic, clinical, and laboratory parameters. The integration of interactive visualizations, such as confusion matrices, ROC curves, precision-recall curves, and F1-score charts, not only provides insights into model performance but also enhances the interpretability and trustworthiness of predictions for healthcare practitioners. By making the predictions both accurate and understandable, the system empowers clinicians to make informed decisions quickly, improves preventive care, and demonstrates the practical utility of machine learning in modern healthcare settings.

7 Results and Discussion

The heart disease prediction system was evaluated on the UCI Heart Disease Dataset, containing 303 patient records with 13 relevant features. The system was assessed in terms of predictive performance, interpretability, and practical applicability, using multiple evaluation metrics and visualizations.

The model evaluation results are:

1) **Confusion Matrix:** A confusion matrix was generated to visualize the performance of the model:
    a) **True Positives (TP):** Number of patients correctly predicted as having heart disease.
    b) **True Negatives (TN):** Number of patients correctly predicted as not having heart disease.
    c) **False Positives (FP)**: Number of patients incorrectly predicted as having heart disease.
    d) **False Negatives (FN):** Number of patients incorrectly predicted as not having heart disease.

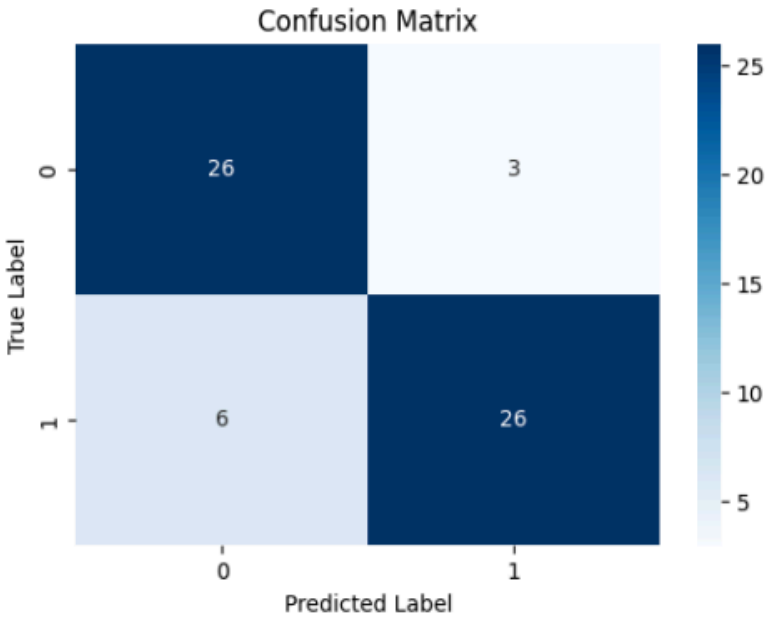

Fig 2 Confusion Matrix

2) **F1 Score per class:**
   - A bar chart of F1-Scores per class shows balanced performance for both classes (Heart Disease vs No Heart Disease).
   - The F1-score reflects the trade-off between precision and recall, indicating that the model maintains both accuracy and reliability for each class.

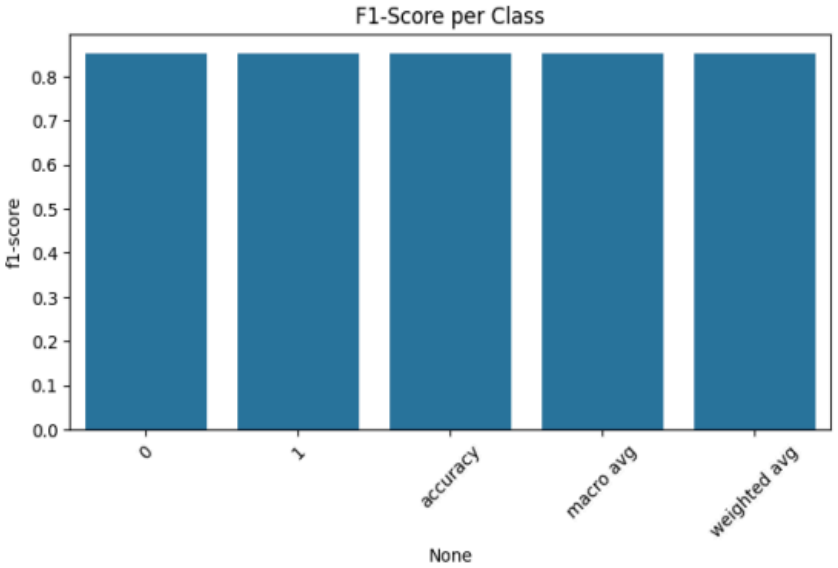

Fig 3 F1 Score

3) **ROC Curve:**The Receiver Operating Characteristic (ROC) curve plots the True Positive Rate (sensitivity) against the False Positive Rate (1 - specificity) for various thresholds.
   - The Area Under the Curve (AUC) is 0.91, indicating excellent ability to distinguish between patients with and without heart disease.
   - A steep rise near the origin suggests that the model achieves high sensitivity with minimal false positives.

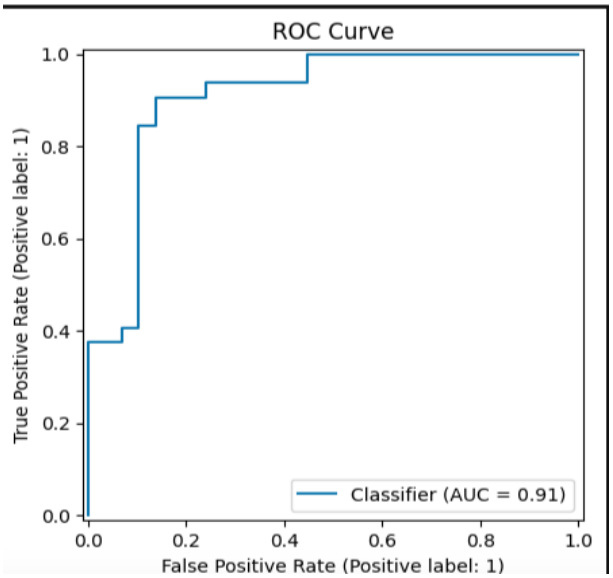

Fig 4 ROC Curve

4) **Precision- Recall Curve:**
    - The Precision-Recall curve shows the trade-off between precision (how many predicted positives are correct) and recall (how many actual positives are detected).
    - High precision at high recall indicates that the model is both accurate and reliable in predicting heart disease.

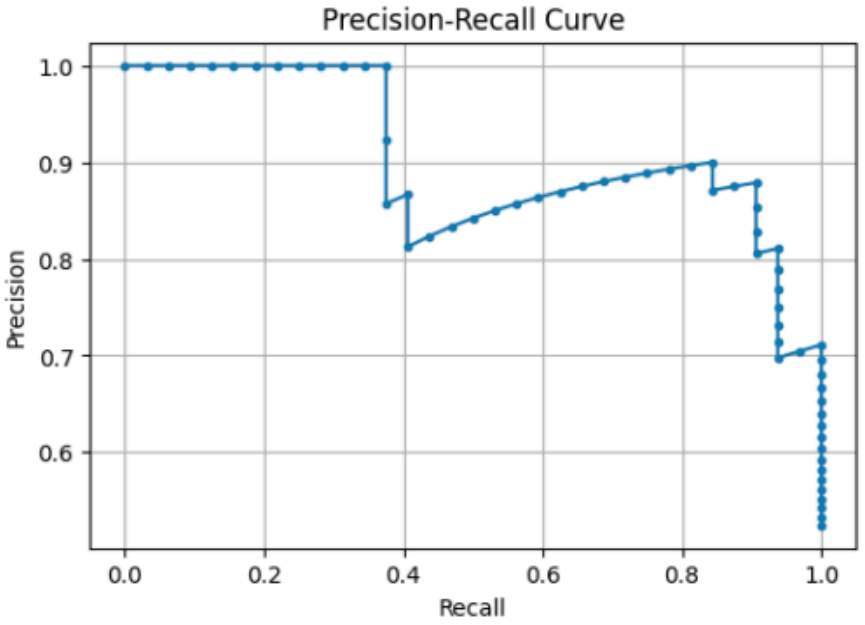

Fig 5 Precision Recall Curve

Insights based on Model Evaluation:

- The confusion matrix highlights that the model makes very few misclassifications.
- The low number of false negatives is particularly important in clinical settings, as failing to detect heart disease could have severe consequences.
- Balanced F1-scores ensure that the model does not favor one class over the other, which is critical in medical applications where both false positives and false negatives carry risks.
- High ROC-AUC ensures that the model can prioritize high-risk patients for further diagnostic tests or preventive interventions.
- Precision-recall analysis is especially important when dealing with imbalanced datasets, where one class is less frequent.
- Ensures minimal misclassification of patients with heart disease (false negatives), improving patient safety

Simple Frontend Output:

1) **Dashboard:** The dashboard displays model evaluation details and an option predict a patient heart risk probability by dynamic input.

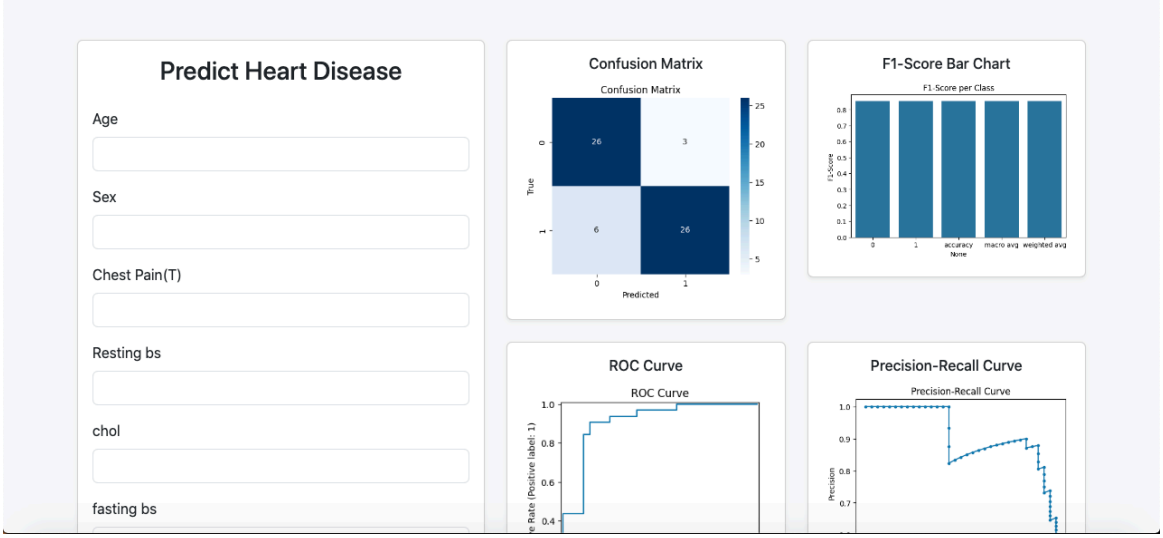

Fig 6 Dashboard

2) **Prediction:** The users can enter details such as age , maximum heart rate,etc. to predict the probability of heart risk.

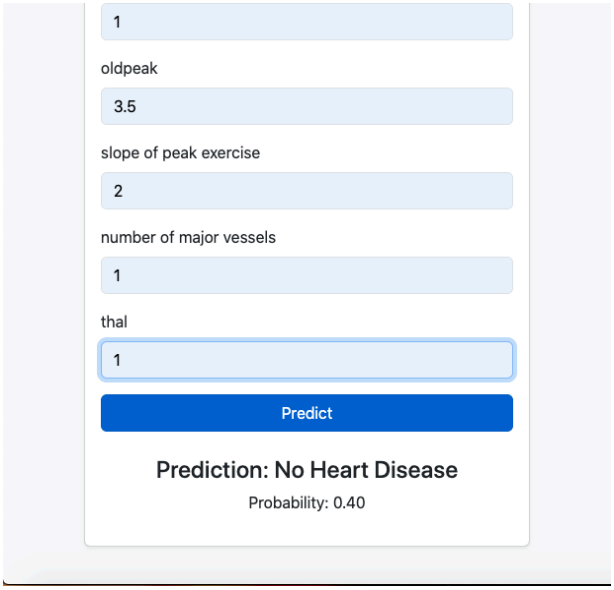

Fig 7 Prediction

3) **Output:** After predicting the probability will be displayed here

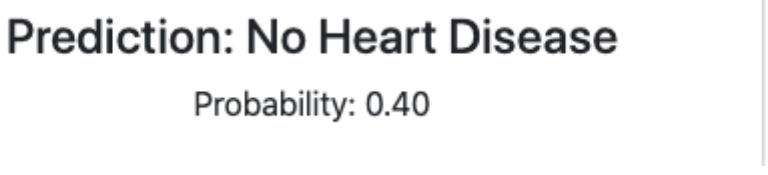

Fig 8 Output

8 Conclusion and Discussion

This project demonstrates the effective use of machine learning to predict the presence of heart disease using patient demographic, clinical, and laboratory data. By leveraging the HistGradientBoostingClassifier, the system achieves high predictive accuracy, balanced precision and recall, and robust discriminative ability, as demonstrated by ROC-AUC and Precision-Recall analysis.

Key achievements of this project include:

1. **Dynamic Prediction:** The system allows real-time input of patient data and provides instant predictions with associated probabilities, making it practical for clinical usage.
2. **Comprehensive Evaluation:** Multiple metrics, including accuracy, F1-score, confusion matrix, ROC curve, and Precision-Recall curve, validate the model's performance.
3. **Interpretability and Visualization:** Visualizations enhance transparency, enabling healthcare practitioners to understand and trust the predictions.
4. **User-Friendly System:** Integration of the model with a modern interface ensures accessibility and ease of use.

Overall, the project successfully bridges the gap between machine learning theory and healthcare application, offering a tool that can assist in early detection, preventive care, and decision-making.

Detection Mechanism

The heart disease detection mechanism follows a structured pipeline:

1. **Input Collection:** Patient-specific data is collected for 13 features, including age, sex, chest pain type, cholesterol levels, blood sugar, maximum heart rate, and other clinical indicators.

2. **Preprocessing:** Data is cleaned, missing values imputed, and categorical features converted to numeric form to match the model training format.

3. **Prediction:** The preprocessed input is fed into the trained **HistGradientBoostingClassifier**, which outputs:
    - **Prediction Class:** Heart Disease or No Heart Disease
    - **Probability Score:** Likelihood of heart disease occurrence
    -
4. **Visualization & Interpretation:** The system presents model evaluation charts (confusion matrix, ROC curve, Precision-Recall curve, F1-score bar chart) to interpret the prediction's reliability.
5. **Decision Support:** Based on the prediction and probability, healthcare practitioners can decide on further diagnostic tests, preventive measures, or

interventions.

Advantages of this Detection Mechanism:

- Quick and reliable risk assessment for individual patients.
- Transparent and interpretable outputs enhance clinician trust.
- Provides actionable insights for preventive care and prioritization.
- Can be expanded to larger datasets or more features for improved accuracy.

The system provides a robust, interpretable, and user-friendly framework for heart disease prediction. By combining dynamic input, predictive modeling, and visualization, it empowers healthcare practitioners with an effective tool for early detection and decision-making, ultimately contributing to better patient outcomes.

Future Scope:

- **Larger and Diverse Datasets:** Incorporating more patient records from multiple hospitals or regions can improve model generalization and accuracy.
- **Additional Features:** Including lifestyle factors, genetic data, imaging results, or biomarkers can enhance predictive power.
- **Explainable AI (XAI):** Integrating techniques like SHAP or LIME can provide detailed feature-level explanations, increasing clinician trust.
- **Mobile or Web Deployment:** Extending the system to a fully responsive web or mobile app can enable wider accessibility for doctors and patients.
- **Real-Time Monitoring:** Integrating wearable device data (heart rate, blood pressure, activity) could allow continuous risk assessment and early alerts.
- **Model Comparison and Ensemble Learning:** Testing multiple models or ensemble approaches could further improve accuracy and reliability.

9 References

1. *Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J.J., Sandhu, S., ... & Froelicher, V. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. The American Journal of Cardiology, 64(5), 304–310.*
2. *UCI Machine Learning Repository. (n.d.). Heart Disease Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/Heart+Disease*
3. *Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.*
4. *Friedman, J., Hastie, T., & Tibshirani, R. (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics.*
5. *Molnar, C. (2020). Interpretable Machine Learning: A Guide for Making Black Box, Models Explainable. Available at: https://christophm.github.io/interpretable-ml-book/*
6. *Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90–95.*