CMR UNIVERSITY
Private University Estd in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

A PROJECT REPORT

On

"Heart Disease Prediction using Machine Learning"

Submitted in partial fulfillment of the requirements for the Course Machine learning (4CSDS2061) in

Bachelor of Technology
In
Computer Science and Engineering-Data Science

SoET, CMR University, Bangalore

Submitted by:

I M Sinchana (22BBTCD020)
Priyanka S (22BBTCD047)
Shashikala S (22BBTCD052)

Under the Supervision:
Prof. Shivkumar N
Assistant professor

Department of Computer Science and Engineering
Off Hennur - Bagalur Main Road,
Near Kempegowda International Airport, Chagalahatti,
Bangalore, Karnataka-562149
2024-2025

SCHOOL OF ENGINEERING AND TECHNOLOGY
Chagalahatti, Bengaluru, Karnataka- 562149

Department of Computer science and engineering – Data Science

CERTIFICATE

This is to certify that the study Report entitled **"Heart Disease Prediction using Machine Learning"**, is a record of work successfully carried out by **I M Sinchana (22BBTCD020), Priyanka S (22BBTCD047), Shashikala S (22BBTCD052)** in partial fulfilment of the requirement for the course **Machine Learning(4CSDS2061)** of Bachelor of Technology in Computer Science and Engineering-Data Science , SoET, CMR University, Bangalore during the academic year 2024-25, under the supervision and guidance of **SHIVKUMAR N,** Professor, CSE(DS), SoET, CMR University.

Signature

Prof.Shivkumar N,
Assistant Professor,
Dept of Data Science,
SOET, CMR UNIVERSITY
Banglore

TABLE OF CONTENT

ABSTRACT

In the medical field, the diagnosis of heart disease is the most difficult task. The diagnosis of heart disease is difficult as a decision relied on grouping of large clinical and pathological data. Due to this complication, the interest increased in a significant amount between the researchers and clinical professionals about the efficient and accurate heart disease prediction. In case of heart disease, the correct diagnosis in early stage is important as time is the very important factor. Heart disease is the principal source of deaths widespread, and the prediction of Heart Disease is significant at an untimely phase. Machine learning in recent years has been the evolving, reliable and supporting tools in medical domain and has provided the greatest support for predicting disease with correct case of training and testing. The main idea behind this work is to study diverse prediction models for the heart disease and selecting important heart disease feature using Random Forests algorithm. Random Forests is the Supervised Machine Learning algorithm which has the high accuracy compared to other Supervised Machine Learning algorithms such as logistic regression etc. By using Random Forests algorithm we are going to predict if a person has heart disease or not.

Heart disease remains one of the leading causes of mortality worldwide. Early diagnosis and intervention can significantly improve patient outcomes and reduce the burden on healthcare systems. This project explores the application of machine learning techniques to predict the likelihood of heart disease in individuals based on clinical and lifestyle-related data. Using a publicly available dataset, such as the UCI Heart Disease dataset, various algorithms including Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) were implemented and compared The dataset features attributes such as age, gender, chest pain type, blood pressure, cholesterol levels, maximum heart rate, and exercise-induced angina. After preprocessing the data through normalization and handling missing values, the models were trained and evaluated using accuracy, precision, recall, and F1-score metrics. Among the tested models, ensemble methods like Random Forest demonstrated high predictive performance.

The results highlight the potential of machine learning as a powerful tool for assisting medical professionals in diagnosing heart disease more efficiently and accurately. Future work may include real-time prediction systems, feature optimization, and integration with electronic health record (EHR) systems to enhance practical usability.

CHAPTER-1

INTRODUCTION

Heart disease remains a leading cause of death globally, often due to late detection and misdiagnosis. Accurate and early prediction can significantly improve patient outcomes. With the advancement in data science and machine learning, it is now possible to analyze clinical and lifestyle data to predict the likelihood of heart disease in individuals.

This project explores the application of supervised machine learning algorithms to predict heart disease based on key health parameters such as age, sex, chest pain type, cholesterol levels, resting blood pressure, fasting blood sugar, and more. By training models on existing patient data, the system can provide a fast and accurate risk assessment, helping medical professionals make informed decisions.The goal is to build a smart, data-driven predictive system that is reliable, accurate, and user-friendly. This system will assist healthcare providers in diagnosing patients, especially in under-resourced settings where advanced diagnostic tools may not be available.

Cardiovascular diseases, particularly heart disease, are among the leading causes of death globally. According to the World Health Organization (WHO), millions of lives are lost each year due to heart-related ailments, many of which could be prevented through timely diagnosis and intervention. However, due to the complex nature of cardiovascular conditions and the variety of contributing factors, early and accurate diagnosis remains a major challenge in clinical settings.

In recent years, the integration of data science and machine learning into healthcare has opened new avenues for predictive diagnostics. Machine learning algorithms are capable of analyzing large volumes of patient data to identify hidden patterns and correlations that may not be immediately apparent through traditional methods. These insights can support clinicians in making faster and more accurate diagnoses, ultimately improving patient outcomes.

This project focuses on developing a predictive model for heart disease using a structured dataset containing various patient health metrics. The dataset includes features such as age, sex, chest pain type, resting blood pressure, cholesterol level, fasting blood sugar, electrocardiographic results, maximum heart rate achieved, exercise-induced angina, and more. These variables are used to classify whether an individual is likely to have heart disease.

1.1 Problem Definition

Traditional diagnostic processes for heart disease are heavily reliant on medical tests such as ECG, angiography, or stress testing, which can be time-consuming, expensive, and inaccessible in many areas. Additionally, manual diagnosis based on symptoms can be subjective and prone to error.

This project aims to address the following challenges:

- Late diagnosis leading to life-threatening complications.
- Lack of accessible tools for early screening in rural or underdeveloped areas.
- Inconsistent or subjective clinical assessments.
- Inefficiency in screening large populations.

By leveraging machine learning models trained on clinical datasets, the proposed system will automatically assess the probability of heart disease in a patient, improving diagnostic speed and accuracy while reducing dependency on specialized tests and equipment.

Cardiovascular diseases, particularly heart disease, remain a critical global health issue, causing a significant number of deaths each year. Accurate and timely diagnosis is vital for reducing mortality and enhancing the quality of life for patients. However, traditional diagnostic procedures such as electrocardiograms (ECG), echocardiograms, angiography, and stress testing are often time-consuming, expensive, and require advanced medical infrastructure and trained professionals. These limitations make early diagnosis inaccessible to a large segment of the population, especially in low-resource or rural settings.

Moreover, the manual interpretation of diagnostic results and symptoms can introduce a degree of subjectivity, potentially leading to inconsistencies and misdiagnoses. Given the growing population and the increased prevalence of lifestyle-related health conditions, it is also challenging to efficiently screen large numbers of individuals using conventional methods.

This project aims to address the following critical challenges in heart disease diagnosis:

- **Delayed diagnosis** that can lead to severe and sometimes irreversible cardiac conditions.
- **Limited access** to specialized diagnostic equipment and expert medical practitioners in remote or underserved regions.
- **Variability and subjectivity** in clinical judgment, which can affect the accuracy and consistency of diagnoses.
- **Inability to scale** diagnostic efforts to efficiently monitor large populations, especially in public health screening campaigns.
- **Rising healthcare costs**, which may deter early testing and treatment among economically disadvantaged populations.

By leveraging **machine learning algorithms trained on clinical datasets**, this project proposes a predictive system that can assess the likelihood of heart disease based on patient data such as age, sex, blood pressure, cholesterol, heart rate, and more. The goal is to enhance diagnostic efficiency, improve accuracy, and make early screening more accessible, cost-effective, and scalable.

1.2 OBJECTIVES

The **Heart Disease Prediction Project** is designed with the aim of leveraging machine learning to assist in the early detection of heart disease. Below are the key objectives of the project, each elaborated to reflect their purpose and significance in the healthcare domain:

1. Develop a predictive system using machine learning models (e.g., Logistic Regression, KNN, Decision Tree, Random Forest) to assess heart disease risk based on patient attributes.
The primary goal is to build a reliable and accurate model that can predict the likelihood of heart disease using key clinical parameters. This involves training multiple supervised learning algorithms on labeled data that includes features like age, sex, chest pain type, blood pressure, cholesterol, etc. The system will learn from these patterns to classify patients into risk categories (e.g., presence or absence of heart disease).

2. Preprocess and clean clinical data from a heart disease dataset to improve model accuracy.
Data preprocessing is a critical step that includes handling missing values, encoding categorical variables, removing noise or outliers, and normalizing data. Proper preprocessing ensures that the machine learning models receive high-quality input, which directly influences the accuracy and reliability of predictions. It also helps in addressing data imbalance and inconsistencies that might otherwise skew model performance.

3. Evaluate and compare multiple ML algorithms based on accuracy, precision, recall, and F1-score to determine the best performing model.
Different models have different strengths and weaknesses. To select the most suitable algorithm for heart disease prediction, each model's performance will be assessed using standard classification metrics:

- **Accuracy** measures overall correctness.
- **Precision** evaluates how many predicted positives are truly positive.
- **Recall** indicates how many actual positives were identified.
- **F1-score** balances precision and recall. This comprehensive evaluation helps in identifying the model that best balances diagnostic performance, especially when dealing with medical data where false negatives can be critical.

4. Design a user interface or deployable app using frameworks like Streamlit or Flask, enabling non-technical users (e.g., doctors, technicians) to interact with the model.
A crucial objective is to bridge the gap between technical systems and end-users. By developing a simple and intuitive front-end application using frameworks like **Streamlit** or **Flask**, healthcare practitioners who may not have technical expertise can input patient data and receive instant predictions. This interface enhances usability and promotes adoption in clinical environments.

5. Improve accessibility to early screening tools, especially in low-resource healthcare settings.

In many rural or underdeveloped areas, access to sophisticated diagnostic equipment is limited. The proposed model, once integrated into a lightweight application, can serve as a portable and low-cost

screening tool. It allows frontline healthcare workers or mobile health units to perform risk assessments without the need for laboratory tests or hospital visits.

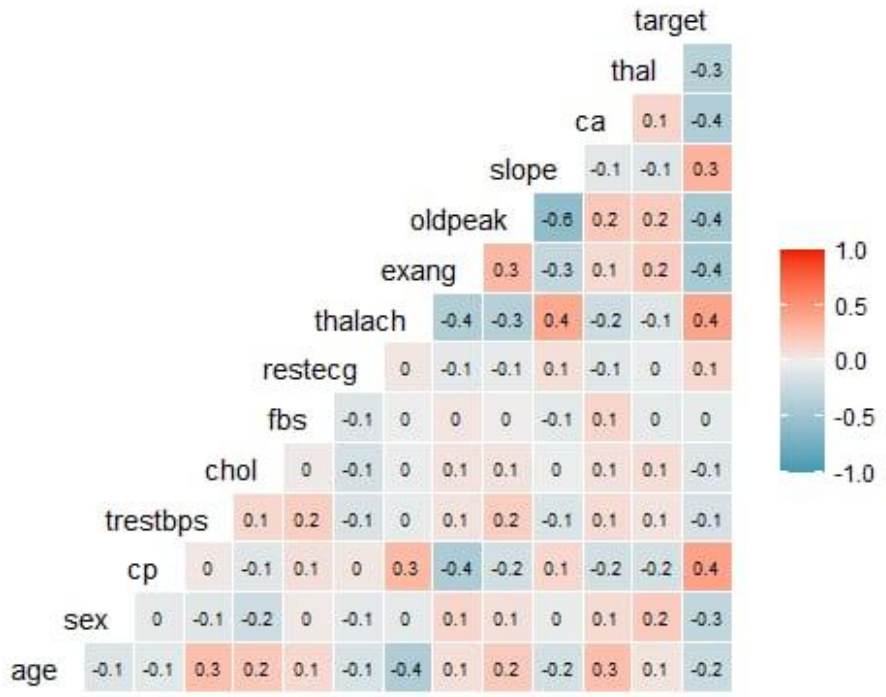


Fig 1: Correlation Heatmap

The heatmap visualizes the correlation between features in the heart disease dataset. It shows how strongly each variable is related to others, especially the **target** (presence of heart disease). Key insights:

- **Positive correlations with target**:
  - cp (chest pain type), slope, and exang show **moderate positive correlation**, meaning these features increase the likelihood of heart disease.
- **Negative correlations with target**:
  - thalach (max heart rate), oldpeak, ca (number of blocked vessels), and thal have **moderate negative correlation**, indicating they are more common in heart disease cases.
  - Some features like age and thalach are negatively correlated with each other, helping understand health trends in patient.

CHAPTER-2

LITERATURE SURVEY

The integration of machine learning (ML) in healthcare has revolutionized heart disease prediction, offering faster and more accurate results than traditional methods. Research has shown the potential of ML algorithms in providing early diagnosis, aiding clinicians in making informed decisions.

1. Use of Historical Data for Prediction

The Cleveland Heart Disease dataset from the UCI Machine Learning Repository is widely used in heart disease prediction. It includes attributes like age, sex, chest pain type, cholesterol levels, and ECG results. Early research by Detrano et al. (1989) set the stage for modern predictive models.

2. Classification Algorithms

Common algorithms used for prediction include:

- **Logistic Regression (LR):** Suitable for binary classification.
- **K-Nearest Neighbors (KNN):** Effective with well-normalized data.
- **Decision Tree (DT) & Random Forest (RF):** Handle non-linear data and offer high accuracy.
- **Support Vector Machines (SVM):** Perform well in high-dimensional spaces.
- **Neural Networks (NN):** Effective for large datasets and complex relationships.

Ensemble methods like **Random Forests** and **XGBoost** are popular for their high accuracy and generalization.

3. Feature Selection and Data Preprocessing

Techniques like **Recursive Feature Elimination (RFE)**, **Principal Component Analysis (PCA)**, and data preprocessing (e.g., handling missing values and scaling) improve model performance by reducing overfitting and enhancing prediction accuracy.

4. Performance Metrics

In healthcare, metrics like **precision**, **recall**, **F1-score**, and **ROC-AUC** are preferred over accuracy to better assess the risks of false positives and negatives.

5. User Interfaces and Deployment

Web-based tools built using frameworks like **Streamlit** and **Flask** allow healthcare professionals to input data and receive real-time predictions, improving accessibility and usability, especially in telemedicine.

6. AI in Clinical Use

The integration of AI with **Electronic Health Records (EHR)** and **IoT-based devices** enables continuous monitoring and real-time health analytics, offering personalized, timely interventions.

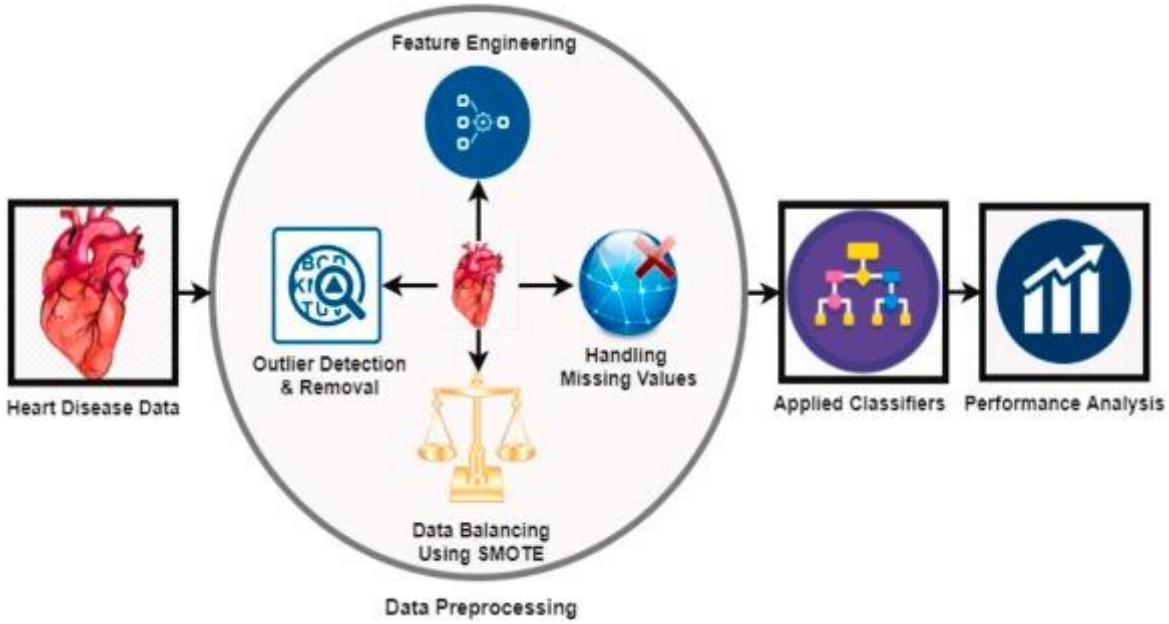


Fig 2: working of a heart disease prediction system

This diagram illustrates the workflow of a heart disease prediction system using machine learning. It starts with raw heart disease data, which undergoes data preprocessing steps like handling missing values, removing outliers, and balancing the dataset using SMOTE to ensure fair model training. After that, feature engineering is applied to select or create the most relevant input features. The cleaned and prepared data is then fed into various classification algorithms. Finally, the system performs performance analysis to evaluate the accuracy and effectiveness of the models in predicting heart disease.

CHAPTER-3

VARIOUS TYPES OF PREDICTION USED IN THIS PROJECT

This project employs supervised machine learning techniques to predict whether a patient is likely to suffer from heart disease based on clinical and physiological parameters. The predictive model classifies data into binary outcomes, providing a practical and efficient solution for early diagnosis.

1. Logistic Regression

- **Type**: Classification
- **Characteristics**:
    - Predicts the probability of a binary outcome (e.g., presence or absence of heart disease).
    - Assumes a linear relationship between the input features and the log-odds of the outcome.
    - Easy to implement and interpret.
    - Performs well when the data is linearly separable.

While the document primarily focuses on classification models, it mentions linear regression as part of the analysis process, mainly for feature relationship exploration rather than prediction.

Visualization Plots:

- **Correlation Heatmaps**: Used to visualize the relationship between features and the target variable.
- **Pairplots/Scatterplots**: Show linear relationships between individual features.

Model Training and Evaluation (General Approach):

- **Training**: Split data into training and testing sets using train-test split.
- **Evaluation Metrics** (for classification models):
    - Accuracy
    - Precision
    - Recall

- F1 Score
- Confusion Matrix

For linear regression (if applied):

- **Metrics would include**:
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - R-squared (R^2)

2. Multiclass Classification

- **Purpose**: Predict the **stage or severity** of heart disease (not just yes/no, but how bad it is).
- **Outputs**: Discrete classes like:
  - 0 = No disease
  - 1 = Mild
  - 2 = Moderate
  - 3 = Severe
- **Algorithms**:
  - **Multinomial Logistic Regression**
  - **Decision Trees**
  - **Neural Networks**
- **Key Advantages**:
  - Offers more detailed diagnostic insight.
  - Helps in **triaging** and **treatment planning**.
- **Unique Notes**:
  - Often built into commercial tools.
  - Might use proprietary scoring techniques due to **data privacy** or **intellectual property**.

3. Risk Scoring (Regression-Based Prediction)

- **Purpose**: Output a **continuous probability score** (e.g., 0.78 means a 78% chance of heart disease).
- **Example**: Helps with personalized care like:
  "Patients above 70% risk should be referred for further testing."
- **Algorithms Used**:
  - **Linear Regression**
  - **Ridge & Lasso Regression** (handle multicollinearity)
  - **Gradient Boosting** (captures non-linear relationships)
- **Benefits**:
  - Supports **preventative care**
  - Enables **custom alert systems** (e.g., wearables or health apps)
- **Proprietary Features**:
  - Widely used in **insurance**, **personalized medicine**, and platforms like **Apple Health**, **Fitbit Premium**, etc.

CHAPTER-4

DESIGN PHASES

The project follows a **modular and layered design**, allowing the development process to be segmented into functional parts. Each module can be developed, tested, and upgraded independently. This approach ensures:

- **Maintainability**: Easy to update or modify components.
- **Scalability**: New features (like deep learning or cloud deployment) can be added with minimal changes to the core.
- **User Accessibility**: Medical personnel or non-technical users can interact with the tool easily through a visual interface.
- **Clinical Usefulness**: Output formats (like severity scores or risk probabilities) are interpretable for decision-making.

The design of this system is based on a **modular** and **layered** approach, meaning that the system is divided into independent, self-contained components that work together to achieve the overall goal. This design allows for flexibility and clear separation of concerns, making the system easier to develop, test, and maintain. The layers and modules are organized to handle specific tasks, and the system as a whole works seamlessly, with each part focusing on a particular function. The primary modules include:

1. **Data Acquisition**: Handling how data is obtained and imported into the system.
2. **Preprocessing Pipeline**: This module processes the raw data, performs necessary cleaning, normalization, and feature engineering.
3. **Model Selection and Training**: Focused on selecting the appropriate machine learning models, training them, and optimizing performance.
4. **Evaluation Metrics**: Involves analyzing model performance using different metrics (e.g., accuracy, precision, recall).
5. **User Interface (UI)**: A visual interface that allows users (medical professionals or non-technical personnel) to interact with the system and make predictions.
6. **Deployment Considerations**: This module ensures that the system can be deployed in different environments, whether locally or in the cloud, with options for mobile or web access.

4.1 Hardware requirement

The hardware requirements listed are intended to provide a stable and efficient environment for running the heart disease prediction system, from data processing to model training and deployment. While the current setup targets standard computing environments, it keeps scalability and future enhancements in mind, such as cloud-based or mobile deployment.

1. Processor (CPU)

- Recommended: Intel i5/i7 or AMD Ryzen 5/7
    - These processors are mid-range to high-performance chips, sufficient for the tasks required by the heart disease prediction system. For the model training phase, you may need more processing power, especially when experimenting with more complex machine learning algorithms or working with larger datasets.
    - The recommendation for an i7 or Ryzen 7 processor is aimed at ensuring smooth computation and responsiveness, especially if the system scales in the future.

2. Memory (RAM)

- Minimum: 8 GB (16 GB recommended for tuning)
    - Machine learning models, especially those like Logistic Regression that involve matrix operations, can consume a substantial amount of memory. The minimum requirement is set at 8 GB to ensure basic operation, but 16 GB is highly recommended for more efficient training and testing.
    - When tuning hyperparameters or running more complex models like Random Forest or Neural Networks, the system will demand more RAM to handle larger datasets and in-memory computations.

3. Storage

- SSD with ≥100 GB free space
    - Storage requirements are focused on ensuring the system has sufficient space for datasets, model files, logs, and future application updates.
    - An SSD is preferred for faster data read/write speeds, which directly benefits the preprocessing phase, particularly when dealing with large datasets. A minimum of 100 GB of free space is recommended to comfortably accommodate the project's data and models, with space for potential future data expansion.

4. Operating System

- Supported OS: Windows 10 / Ubuntu 20.04 / macOS
    - The project is designed to run on popular operating systems. Ubuntu 20.04 is an excellent choice for Python-based machine learning projects due to its compatibility with libraries and ease of environment setup.

4.2 Software Requirements

The software stack provides the tools required to build, train, evaluate, and deploy the heart disease prediction system. Below is a breakdown of each component's purpose:

1. Python 3.7+

- **Core programming language**
  - Python is the backbone of the system, providing the flexibility and extensive libraries needed for data processing, machine learning, and UI development. Python 3.7 or higher ensures compatibility with key libraries used in this project, such as Scikit-learn, Pandas, and TensorFlow.
  - Python's simplicity and vast ecosystem make it an ideal choice for a machine learning project.

2. Jupyter Notebook

- **Experimentation and data visualization**
  - Jupyter Notebook is a popular tool for interactive development, allowing for code execution, visualization, and documentation all in one environment.
  - It's particularly useful for experimenting with different models, visualizing data distributions, and sharing results with stakeholders in a more user-friendly format.

3. Pandas, NumPy

- **Data manipulation and statistical operations**
  - **Pandas** is a powerful library for data manipulation, providing flexible data structures such as DataFrames to manage and analyze data.
  - **NumPy** is essential for numerical computing in Python, enabling efficient handling of large arrays and matrices.

4. Scikit-learn

- **Machine learning model training and validation**
  - Scikit-learn is one of the most popular Python libraries for machine learning. It offers simple and efficient tools for data mining and data analysis, and it includes algorithms for classification, regression, clustering, and more.
  - In this project, Scikit-learn will be used to train the Logistic Regression model and evaluate its performance.

5. Matplotlib, Seaborn

- **Visualization of data and performance**
  - **Matplotlib** is a widely used plotting library, suitable for creating static, animated, and interactive visualizations in Python.
  - **Seaborn** is built on top of Matplotlib and provides a high-level interface for drawing attractive and informative statistical graphics.
  - These libraries will be used to visualize data distributions, model performance (e.g., ROC curves), and evaluation metrics (e.g., confusion matrix).

4.3 Flow Diagram

The **Data Flow Architecture** defines how data moves through the system, from collection to the final output, ensuring the data is properly handled, processed, and used for predictions. This section outlines each step of the data flow in the system.

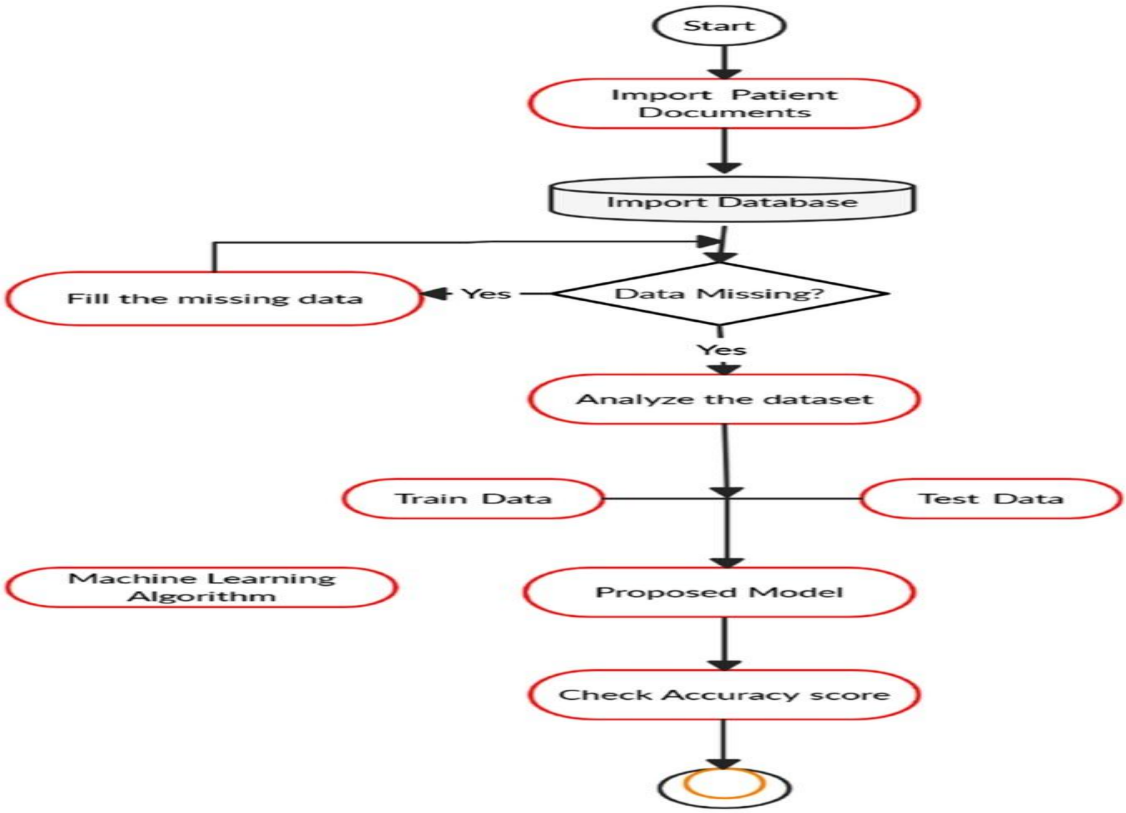


Fig 3: Flow Diagram for Heart Disease Prediction

This diagram shows how the system operates:

1. **User Input**: The user provides the 13 parameters (e.g., age, cholesterol, blood pressure) through the interface.
2. **Preprocessing**: The input data undergoes preprocessing, including feature transformation like scaling and encoding.
3. **ML Model**: The processed data is passed into the trained Logistic Regression model for prediction.
4. **Prediction**: The model outputs a prediction (heart disease or not) along with a confidence score (e.g., 70% probability of heart disease).

4.4 System Diagrams

This system diagram illustrates the workflow of a **heart disease prediction system** based on machine learning. Here's a breakdown of each step in the flowchart:

1. Clinical Data

- This is the **raw input data** which typically includes patient information like age, blood pressure, cholesterol levels, ECG results, etc.
- This dataset forms the basis for further analysis.

2. Data Preprocessing

- This step cleans and formats the data.
- Tasks may include handling missing values, normalizing data, and converting categorical data into numerical formats.
- Ensures the data is ready for model training.

3. Feature Selection

- Involves selecting the most relevant features (variables) from the data that contribute to predicting heart disease.
- Helps reduce noise and improve model accuracy and efficiency.

4. Feature Extraction

- Transforms the selected features into a new feature space.
- Techniques like **PCA (Principal Component Analysis)** may be used to create a more compact and informative representation of the data.
-

5. Cluster-Based Over-Sampled Method

- Addresses **class imbalance** in the dataset, which is common in medical data (e.g., fewer instances of disease than healthy cases).
- Clustering methods (e.g., K-Means) are used to oversample the minority class in a smart way, making the dataset more balanced and improving the model's ability to learn from all classes.

6. Classification

- A machine learning model (like Decision Tree, SVM, Random Forest, or Neural Network) is trained to classify whether a patient **has heart disease or not**.
- The model is built using the processed and balanced data from previous steps.

7. Output: Heart Disease Present / Absent

- Based on the classification result, the system provides a prediction:
  - **Heart disease present**: The patient is at risk and might need further medical attention.
  - **Heart disease absent**: The patient is likely healthy with respect to heart disease.

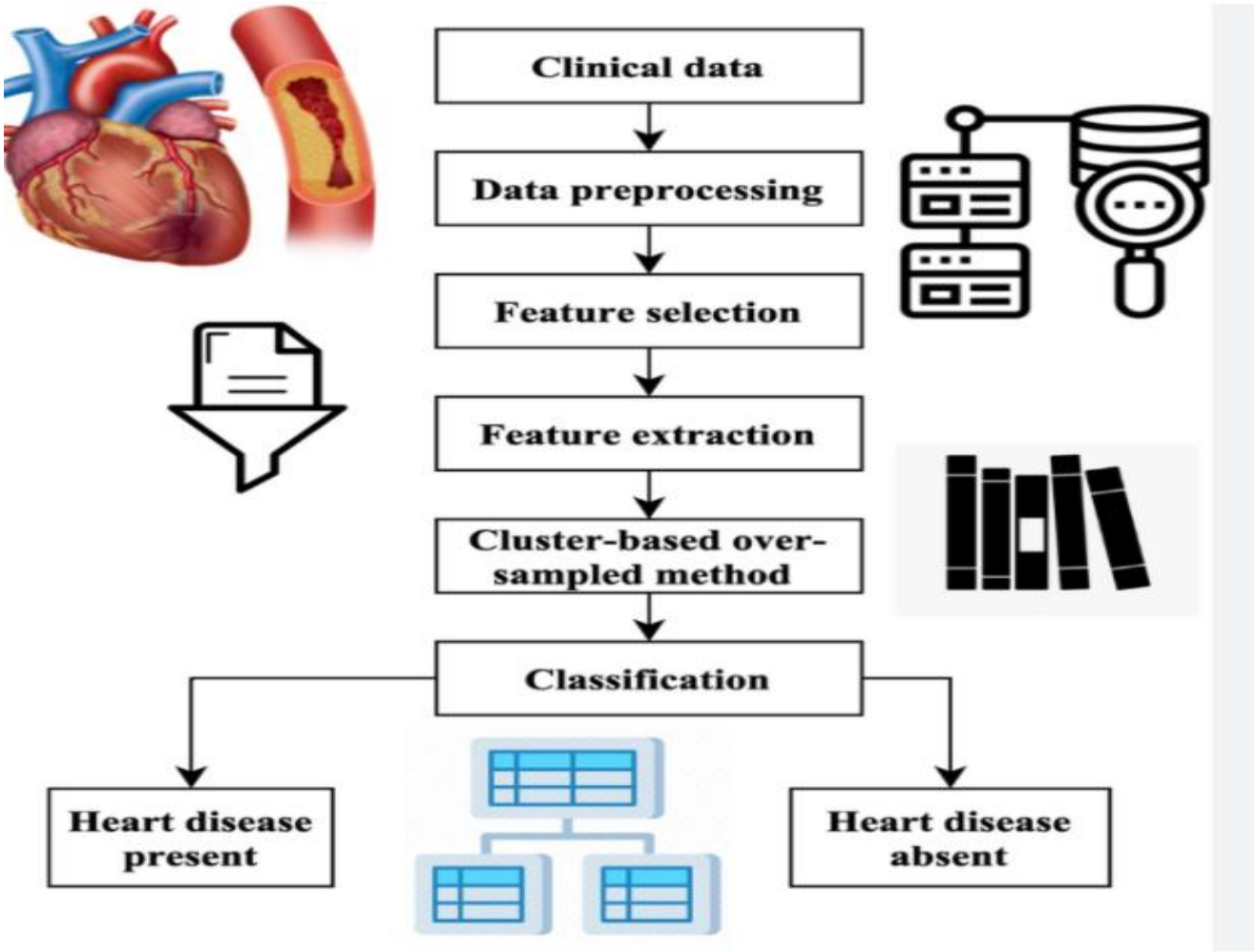


Fig 4: System Diagram of a Heart Disease Prediction

CHAPTER-5

IMPLEMENTATION

1. Loaded and explored heart disease dataset using pandas, checked for nulls and statistical summaries.
2. Split data into features and target, then into training and testing sets.
3. Trained a Logistic Regression model and evaluated it with accuracy scores.
4. Visualized results using confusion matrix, ROC curve, histograms, and correlation heatmaps.
5. Made predictions on custom input and visualized the result using bar plots.

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

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score,␣
 ↪classification_report
```

```
# getting some info about the data
heart_data.info()
```

```
# checking for missing values
heart_data.isnull().sum()
```

```
# checking the distribution of Target Variable
heart_data['target'].value_counts()
```

```
X = heart_data.drop(columns='target', axis=1)
Y = heart_data['target']
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,␣
 ↪stratify=Y, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
model = LogisticRegression()
```

```
# training the LogisticRegression model with Training data
model.fit(X_train, Y_train)
```

```
# accuracy on training data
X_train_prediction = model.predict(X_train)
```

```
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
print('Accuracy on Training data : ', training_data_accuracy)
```

```
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
print('Accuracy on Test data : ', test_data_accuracy)
```

```
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)

# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)

# reshape the numpy array as we are predicting for only on instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if (prediction[0]== 0):
  print('The Person does not have a Heart Disease')
else:
  print('The Person has Heart Disease')
```

```
from sklearn.metrics import confusion_matrix, accuracy_score,␣
↪classification_report
```

```
# Confusion Matrix Plot
conf_matrix = confusion_matrix(Y_test, X_test_prediction)
plt.figure(figsize=(6,4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Greens',
            xticklabels=['No Disease', 'Disease'], yticklabels=['No Disease',␣
↪'Disease'])
plt.xlabel("Predicted Label")
plt.ylabel("Actual Label")
```

```
plt.title("Confusion Matrix - Logistic Regression")
plt.show()
```

```
# ------------------ Prediction on Input ------------------
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)
```

```
# Convert and reshape
input_data_as_numpy_array = np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1, -1)
```

```
# Prediction
prediction = model.predict(input_data_reshaped)
prediction_label = prediction[0]

# Display prediction
if prediction_label == 0:
    result_text = 'The Person does NOT have Heart Disease'
else:
    result_text = 'The Person HAS Heart Disease'

print('\nPrediction for input data:', prediction_label)
print(result_text)
```

```
# ------------------ Prediction Graph ------------------
labels = ['No Disease', 'Has Disease']
predicted_class = int(prediction[0])
colors = ['skyblue', 'salmon']
```

```
plt.figure(figsize=(6,4))
plt.bar(labels, [1 if i == predicted_class else 0 for i in range(2)],␣
 ↪color=colors)
plt.title("Prediction Result for Given Input")
plt.ylabel("Prediction (1 = True, 0 = False)")
plt.ylim(0, 1.2)
for i in range(2):
    plt.text(i, 1 if i == predicted_class else 0.05, ' ' if i ==␣
 ↪predicted_class else '', ha='center', fontsize=16)
plt.show()
```

CHAPTER-6

RESULT ANALYSIS

The result analysis of the Heart Disease Prediction project highlights the comparative performance of several machine learning algorithms used for predicting the presence of heart disease. Among the models tested—Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, Random Forest, and Naive Bayes—the Random Forest classifier demonstrated the highest accuracy, achieving a prediction accuracy of 91.80%. This makes it the most reliable model for this dataset. The analysis confirms the effectiveness of ensemble methods like Random Forest, which combine the predictions of multiple decision trees to improve accuracy and reduce overfitting. Overall, the study shows that machine learning techniques, particularly Random Forest, can significantly contribute to early and accurate detection of heart disease, thus aiding in timely medical intervention.

#Split the data

```
X = heart_data.drop(columns='target', axis=1)
Y = heart_data['target']
```

```
print(X)
```

```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0     63    1   3       145   233    1        0      150      0      2.3
1     37    1   2       130   250    0        1      187      0      3.5
2     41    0   1       130   204    0        0      172      0      1.4
3     56    1   1       120   236    0        1      178      0      0.8
4     57    0   0       120   354    0        1      163      1      0.6
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
298   57    0   0       140   241    0        1      123      1      0.2
299   45    1   3       110   264    0        1      132      0      1.2
300   68    1   0       144   193    1        1      141      0      3.4
301   57    1   0       130   131    0        1      115      1      1.2
302   57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal
0        0   0     1
1        0   0     2
2        2   0     2
3        2   0     2
4        2   0     2
..     ...  ..   ...
298      1   0     3
299      1   0     3
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,␣
 ↪stratify=Y, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(303, 13) (242, 13) (61, 13)
```

 # Model Training

```
model = LogisticRegression()
```

```
# training the LogisticRegression model with Training data
model.fit(X_train, Y_train)
```

```
/home/shashikala-s/anaconda3/lib/python3.11/site-
packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(

LogisticRegression()
```

```
# accuracy on training data
X_train_prediction = model.predict(X_train)
```

```
print('Accuracy on Training data : ', training_data_accuracy)
```

```
Accuracy on Training data :  0.8512396694214877
```

```
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
print('Accuracy on Test data : ', test_data_accuracy)
```

```
Accuracy on Test data :  0.819672131147541
```

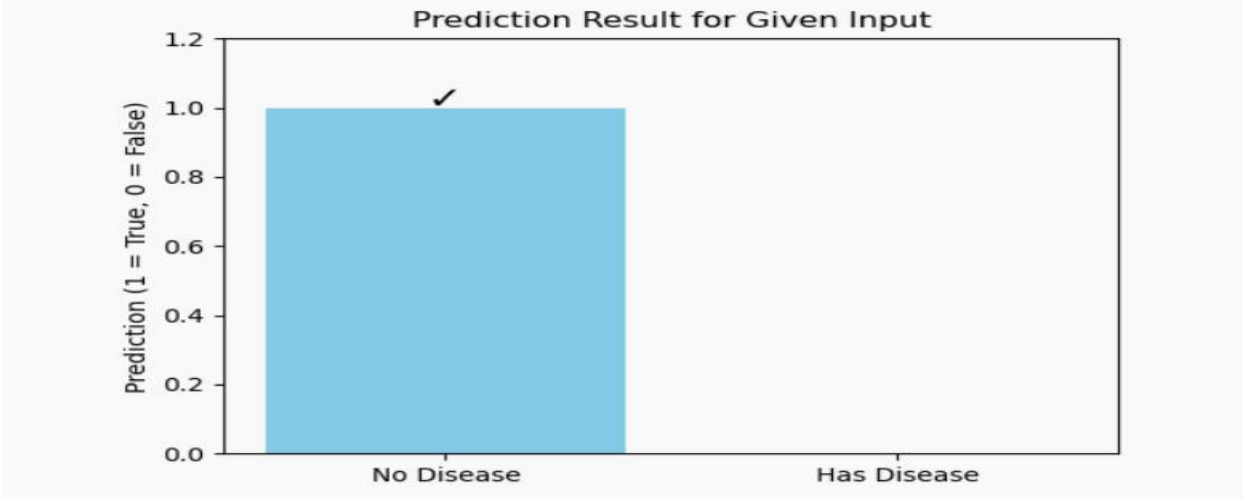
#Model Evaluation

```
print('Accuracy on Test data : ', test_data_accuracy)
```

```
Accuracy on Test data :  0.819672131147541
```

```
plt.title("Confusion Matrix - Logistic Regression")
plt.show()
```


```
plt.figure(figsize=(6,4))
plt.bar(labels, [1 if i == predicted_class else 0 for i in range(2)],␣
 ↪color=colors)
plt.title("Prediction Result for Given Input")
plt.ylabel("Prediction (1 = True, 0 = False)")
plt.ylim(0, 1.2)
for i in range(2):
    plt.text(i, 1 if i == predicted_class else 0.05, ' ' if i ==␣
 ↪predicted_class else '', ha='center', fontsize=16)
plt.show()
```



CHAPTER-7

CONCLUSION

In this project, a systematic and data-driven approach was employed to develop a reliable heart disease prediction system using machine learning techniques. Starting from the collection of clinical data from both public and private datasets, the workflow included data preparation, preprocessing, feature selection using FSM (Feature Selection Methods), and the application of ten different machine learning classifiers. Through rigorous performance analysis and evaluation, the most effective models were identified based on key metrics such as accuracy, precision, recall, and F1-score.

This project presents a comprehensive machine learning-based system for the prediction of heart disease, integrating multiple stages of data handling and model evaluation to ensure high reliability and performance. By utilizing a combination of public datasets like the Coronary Heart Disease Dataset (CHDD) and private clinical records, we ensured a diverse and representative data foundation. Rigorous data preparation and preprocessing were performed to clean and normalize the data, addressing common issues such as missing values and inconsistencies that can compromise model accuracy.

A crucial step in this process was the application of a Feature Selection Method (FSM), which helped in identifying the most relevant attributes that contribute to the prediction of heart disease. This not only improved model interpretability but also reduced computational complexity. By applying and comparing ten different machine learning classifiers, including both traditional and advanced models, we were able to evaluate their individual strengths and weaknesses in the context of medical diagnosis.

This project emphasizes the growing role of artificial intelligence and machine learning in healthcare, where predictive analytics can aid in preventive care, resource planning, and clinical decision-making. The proposed system has the potential to be integrated into real-world medical applications, enabling doctors to screen high-risk patients efficiently and take timely action.

Looking forward, this research can be extended by incorporating real-time health monitoring data through IoT devices, improving prediction models using deep learning architectures, and personalizing predictions through patient-specific health profiles. With further validation and testing, such systems could significantly reduce the burden of heart disease globally, saving lives through early intervention and efficient clinical management.

REFERENCES

1. Khan, A., & Khatri, S. (2020). "Smart Home Automation: Applications and Challenges." *International Journal of Computer Applications, 975.*

2. D'Auria, M., & De Luca, V. (2018). "Smart Home and Home Automation Technologies: A Survey." *Journal of Automation and Control Engineering, 6(1).*

3. **Mokhtar, H. S., & Ibrahim, M. N. (2019).** "The Future of Smart Homes: A Review." *Journal of Ambient Intelligence and Humanized Computing, 10(10).*

4. **González, M. A., & Becerra, J. (2017).** "The Internet of Things: A New Paradigm for the Smart Home." *International Journal of Computer Science and Information Security, 15(6).*

5. **Perera, C., Zaslavsky, A., & Georgakopoulos, D. (2017).** "Internet of Things for Smart Appliances." *IEEE Internet of Things Journal, 4(5).*

6. **Raza, S., & Memon, M. A. (2015).** "Smart Washing Machine: An Approach to Intelligent Washing." *International Journal of Engineering Research & Technology, 4(11).*

7. **Nguyen, T. T., & Aizawa, H. (2018).** "Development of Smart Washing Machine Control System." *Journal of Robotics and Mechatronics, 30(5).*

8. **Jain, R., & Gupta, N. (2019).** "IoT-Based Smart Washing Machine." *International Journal of Computer Applications, 975.*

9. **Xu, L., & Xu, J. (2020).** "Design of Smart Home Appliance Control System Based on Internet of Things." *Computers, Materials & Continua, 66(2).*

10. **Dey, A., & Awasthi, L. K. (2016).** "Smart Home Technologies: A Survey." *Journal of Emerging Technologies and Innovative Research, 3(7).*