

# SCHOOL OF TECHNOLOGY, DESIGN

# AND COMPUTER APPLICATION

# ADITYA SILVER OAK INSTITUTE OF TECHNOLOGY

# DEPARTMENT OF COMPUTER ENGINEERING

A Report on

**ML Model for Heart Disease Detection**

**Healthcare Innovation with AI**

Under Subject of

# PROBLEM BASED LEARNING - II

# Semester - VI

***Submitted by***

BHATIYA AARTI (2202030400015)

SUJAL JAIN (2202030400050)

GAURANG GOHIL (2202030400037)

**Prof. Dhenu Patel Dr. Jay Dave**

Guide Head of The Department

# 

**Aditya Silver Oak Institute of Technology**

Department of Computer Engineering

**CERTIFICATE**

This is to certify that the project entitled “ML Model for Heart Disease Detection” has been carried out by **“Aarti Bhatiya”** under my guidance in fulfilment of the Problem Based Learning-II (1010003392) Subject of Bachelor of Technology in **Computer Engineering** – 6th Semester of Silver Oak University, Ahmedabad during the academic year 2024- 2025.

**Prof. Dhenu Patel Dr. Jay Dave**

Guide Head of The Department

**Aditya Silver Oak Institute of Technology**

Department of Computer Engineering

**CERTIFICATE**

This is to certify that the project entitled “ML Model for Heart Disease Detection” has been carried out by **“Sujal Jain”** under my guidance in fulfilment of the Problem Based Learning-II (1010003392) Subject of Bachelor of Technology in **Computer Engineering** – 6th Semester of Silver Oak University, Ahmedabad during the academic year 2024- 2025.

**Prof. Dhenu Patel Dr. Jay Dave**

Guide Head of The Department

**Aditya Silver Oak Institute of Technology**

Department of Computer Engineering

**CERTIFICATE**

This is to certify that the project entitled “ML Model for Heart Disease Detection” has been carried out by **“Gaurang Gohil”** under my guidance in fulfilment of the Problem Based Learning-II (1010003392) Subject of Bachelor of Technology in **Computer Engineering** – 6th Semester of Silver Oak University, Ahmedabad during the academic year 2024- 2025.

**Prof. Dhenu Patel Dr. Jay Dave**

Guide Head of The Department

## ACKNOWLEDGEMENT

We are heartily thankful to our supervisor, **Prof. Dhenu Patel,** whose encouragement, supervision and support from the preliminary to the concluding level enabled me to develop an understanding of the subject. At the end, we offer my regards and blessings to all of those who supported us in any respect during the completion of the project and to our college for providing resources and materials.

We would like to extend my gratitude to **HOD Dr. Jay Dave** head of Computer Engineering Department, Aditya Silver Oak Institute of Technology, Ahmedabad, for his continuous encouragement and motivation.

Last but not the least we would like to mention here that we are greatly indebted to each and everybody who has been associated with our project at any stage but whose name does not find a place in this acknowledgement.

**Aarti Bhatiya, Sujal Jain, Gaurang Gohil**

Student name

# ABSTRACT

Cardiovascular diseases, particularly heart disease, represent a major global health concern, accounting for millions of deaths each year. Early detection and intervention are vital in reducing the risk of severe complications and improving patient outcomes. In recent years, machine learning (ML) has emerged as a powerful tool for enhancing diagnostic accuracy and supporting decision-making in the medical field. This project focuses on the development of a machine learning-based web application for the early detection of heart disease, aiming to provide a user-friendly, accessible platform that combines data science with healthcare.

The project begins with the use of a well-established heart disease dataset containing various clinical parameters such as age, sex, chest pain type, resting blood pressure, cholesterol levels, fasting blood sugar, electrocardiographic results, maximum heart rate achieved, and more. These features are preprocessed and analyzed to identify significant predictors of heart disease. Several machine learning algorithms—including Logistic Regression, Random Forest, and Support Vector Machines—are trained and evaluated to determine the most accurate model for prediction. The best-performing model is then serialized and integrated into a Flask-based web application.

The application allows users to input their medical data through a simple interface and instantly receive a prediction indicating whether they are at risk for heart disease. This real-time feedback is powered by the trained machine learning model and includes all necessary preprocessing steps such as data normalization and transformation.

Beyond the technical implementation, the web application serves a broader goal: to make preventive healthcare more approachable and data-driven. By providing a tool that can assist both individuals and healthcare practitioners in assessing heart disease risk, the project demonstrates how artificial intelligence can be leveraged to support public health initiatives. The combination of ML techniques and web technologies creates a scalable solution that can be further enhanced and deployed in real-world medical environments.

This project exemplifies the intersection of machine learning, healthcare, and web development, offering a practical solution to a pressing global issue. It highlights the potential of digital health innovations in reducing diagnostic gaps, empowering users with knowledge, and ultimately contributing to the global fight against cardiovascular disease.

# LIST OF FIGURES

| **Figure No.** | **Title** |
| --- | --- |
| Fig 4.1.1. | Flow Chart |
| Fig 4.1.2. | Distribution of categorical features |
| Fig 4.1.3. | Numerical feature Corelation |
| Fig 4.1.4. | ROC curve |

# LIST OF TABLES

Table 1.1 Milestones and Deliverables............................................................................ 08

Table 1.2 Roles and Resposibilities................................................................................. 09

## TABLE OF CONTENT

| **Chapter No.** | **Title** | **Page No.** |
| --- | --- | --- |
|  | **ACKNOWLEDGEMENT** | **i** |
|  | **ABSTRACT** | **ii** |
|  | **LIST OF FIGURES** | **iii** |
|  | **LIST OF TABLES** | **iv** |
| **1** | **INTRODUCTION** | **1** |
|  | **Project Summary** | **2** |
|  | **Purpose** | **3** |
|  | **Scope** |  |
|  | **Technical and Literature Review** |  |
| **2** | **SYSTEM REQUIREMENT STUDY** |  |
|  | **User Characteristics** |  |

**Hardware and Software Characteristics**

Hardware Requirements Software Requirements

1. **SYSTEM ANALYSIS**

**Study of Current System**

**Problems in Existing Systems Requirement of New System** Functional Requirement

Non-functional Requirement

**Feasibility Study**

**Main Modules of New System**

1. **SYSTEM DESIGN**

**Flow charts**

**Samples of Forms, Screen Shots**

1. **LIMITATION AND FUTURE ENHANCEMENT**
2. **TIMELINE CHART**
3. **CONCLUSION REFERENCE**

## INTRODUCTION

### 1.1 Project summary

This project focuses on developing a machine learning-powered web application for the early detection of heart disease. The motivation behind this work is the alarming global impact of cardiovascular diseases, which continue to be a leading cause of mortality. By leveraging a structured dataset containing patient health parameters such as age, cholesterol levels, blood pressure, chest pain type, and more, this system aims to classify whether an individual is at risk of heart disease. Several machine learning models were trained and evaluated using performance metrics like accuracy, precision, recall, and AUC score. The best-performing model was chosen, fine-tuned, and then integrated into a lightweight, responsive web application built with Flask. This allows users to interactively input their medical information and receive immediate predictions on their heart health risk.

The system not only offers a quick and data-driven prediction but also serves as an awareness tool for users who may not have direct access to healthcare facilities. Behind the scenes, the app handles data preprocessing, scaling, and inference in real-time. The user interface is designed to be simple and accessible, promoting ease of use for people from all backgrounds. The deployment architecture supports both local and potential cloud hosting environments, making the solution scalable. Future enhancements could include integrating real-time user health data from wearable devices or electronic health records. Overall, this project exemplifies the use of AI in healthcare, aiming to support preventive diagnostics and promote early medical intervention for better outcomes.

### 1.2 PURPOSE

The purpose of this project is to develop a machine learning-based web application capable of predicting the likelihood of heart disease in individuals based on key medical attributes. With heart disease being one of the most prevalent and life-threatening conditions globally, early detection and risk assessment can significantly improve treatment outcomes and save lives. This project aims to leverage data-driven techniques to build a reliable and accessible diagnostic tool that empowers users to monitor their heart health. By utilizing clinical features such as age, cholesterol levels, blood pressure, and chest pain type, the system provides quick and accurate predictions to support early intervention. The web application is designed to be user-friendly and accessible to both healthcare professionals and the general public, promoting health awareness and informed decision-making. In addition, the project demonstrates how artificial intelligence can be effectively applied in the healthcare domain to assist with preventive care. It also provides a foundation for further research, enhancements, and real-world deployment in clinical settings.

### 1.3 OBJECTIVES

1. Data Acquisition & Preprocessing
   * Collect a reliable dataset related to heart disease (e.g., UCI Heart Disease dataset).
   * Clean, preprocess, and normalize data (handle missing values, encode categorical variables, etc.).
2. Exploratory Data Analysis (EDA)
   * Analyze the distribution and relationship between features.
   * Identify important features correlated with heart disease using visualizations and statistical methods.
3. Feature Selection & Engineering
   * Select relevant features that contribute most to the prediction.
   * Perform feature scaling and transformation as needed.
4. Model Development
   * Implement various ML algorithms (e.g., Logistic Regression, Decision Tree, Random Forest, SVM, etc.).
   * Train models on the processed dataset.
5. Model Evaluation & Comparison
   * Evaluate models using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
   * Compare model performances to select the best one.
6. Hyperparameter Tuning
   * Optimize model performance using techniques like Grid Search or Random Search with cross-validation.
7. Deployment (Optional, if included)
   * Create a simple UI or API to demonstrate the model in action.
   * Deploy the model using a platform like Flask, Streamlit, or FastAPI (if applicable).
8. Conclusion & Future Scope
   * Summarize findings, model performance, and real-world applicability.
   * Suggest potential improvements or extensions (e.g., using deep learning, larger datasets, or real-time prediction).

### 1.4 SCOPE

### The scope of this project is to develop a machine learning model capable of predicting the presence of heart disease based on clinical and demographic features such as age, blood pressure, cholesterol levels, and chest pain type. The project involves data preprocessing, model training, evaluation, and comparison of different algorithms to identify the most accurate approach. While the model aims to assist in early detection and risk assessment, it is not intended to replace professional medical diagnosis. The scope is limited to structured datasets and serves as a foundation for potential integration into decision-support systems in healthcare.

### 1.5 Technology and Literature Review

Technology:

1. Programming Language:
   * Python – for data handling, modeling, and visualization.
2. Libraries & Tools:
   * Pandas and NumPy – for data manipulation and analysis.
   * Matplotlib and Seaborn – for data visualization.
   * Scikit-learn – for building and evaluating machine learning models.
   * Jupyter Notebook – for development and interactive coding.
   * (Optional) Streamlit/Flask – for creating a simple web interface for the model.
   * (Optional) Google Colab – for cloud-based model development and training.

Literature Review:

Several studies and research papers have explored the application of machine learning in detecting heart disease, demonstrating promising results in improving early diagnosis and risk assessment. The UCI Heart Disease dataset has been widely used in many of these studies due to its rich and structured features. Techniques such as Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines have been employed to classify the presence of heart disease, with accuracy levels typically ranging from 80% to 90%. Some studies also utilized feature selection methods like Recursive Feature Elimination to enhance performance. In recent years, ensemble methods and neural networks have also shown improved predictive capability. However, a common challenge highlighted across literature is the need for more diverse and real-time datasets **to increase model generalizability. This project builds upon these existing approaches to evaluate and identify the most suitable algorithm for accurate heart disease prediction using standard clinical data.**

## SYSTEM REQUIREMENTS STUDY

### 2.1 USER CHARACTERISTICS

User characteristics encompass diverse roles, including clinicians, researchers, and healthcare professionals, with varying expertise levels in neuroimaging and software utilization.

1. Clinicians:
   * Require intuitive interfaces and accurate results for clinical decision-making.
   * Varying levels of technical expertise, from novice to advanced. - Need seamless integration with existing clinical workflows.

1. Researchers:
   * Seek customizable parameters for experimentation and validation.
   * Prefer flexibility and control over segmentation algorithms and parameters.
   * Require robust documentation and support for reproducibility in research studies.

1. Radiologists:
   * Demand high-quality segmentation results for accurate diagnosis.
   * Focus on efficiency and speed in processing large volumes of MRI scans. - Require user-friendly interfaces for ease of use in clinical settings.

1. Software Developers:
   * Need access to well-documented APIs for integration into existing systems.
   * Prefer modular and scalable architectures for future enhancements and updates.
   * Seek comprehensive testing and validation procedures for reliable performance.

1. Data Analysts:
   * Require access to segmented data for further analysis and interpretation.
   * Seek compatibility with data visualization tools for insightful data exploration.
   * Need support for handling large datasets and managing computational resources effectively.

### 2.2 HARDWARE AND SOFTWARE REQUIREMENTS

#### 2.2.1 Software requirements

Code Language: Python

Deep Learning Framework: PyTorch or TensorFlow

Web Development Framework: Flask or Django (fordeployment)

Front End: HTML, CSS, JavaScript (for web interface)

#### 2.2.2 Hardware Requirements

RAM: 8GB or higher

Processor: Intel Core i7 or equivalent

Graphics Processing Unit (GPU): NVIDIA GPU with CUDA support (for accelerated training, if available)

Storage: 500GB SSD or higher (for storing datasets, model checkpoints, and intermediate results)

Display: 15.6” Full HD monitor or larger

## SYSTEM ANALYSIS

### 3.1 STUDY OF CURRENT SYSTEM

Existing Software Solutions:

Identify any current software solutions or systems used for hippocampal segmentation in neuroimaging research or clinical practice.

Workflow Analysis:

Analyze the workflow and processes involved in hippocampal segmentation tasks, including data acquisition, preprocessing, segmentation, and analysis.

Tools and Technologies:

Evaluate the tools, technologies, and methodologies currently employed for hippocampal segmentation, including software platforms, algorithms, and hardware resources.

User Feedback:

Gather feedback from users, such as clinicians and researchers, regarding the usability, effectiveness, and limitations of the current system.

Integration Challenges:

Identify any challenges or limitations in integrating the current system with other software platforms or clinical workflows.

Performance Assessment:

Assess the performance of the current system in terms of segmentation accuracy, processing speed, and scalability to handle large datasets.

Gap Analysis:

Conduct a gap analysis to identify areas where the current system may fall short or lack functionality, guiding the development of the proposed software solution.

### 3.2 PROBLEMS AND WEAKNESSES OF CURRENT SYSTEM

The current system for hippocampal segmentation in neuroimaging research and clinical practice exhibits several shortcomings. Firstly, existing software solutions may lack accuracy, leading to inconsistencies in segmentation results. Secondly, the workflow analysis reveals inefficiencies in data preprocessing, segmentation, and analysis, resulting in time-consuming processes. Additionally, outdated tools and technologies may limit the system's capabilities, hindering advancements in segmentation accuracy and performance. User feedback indicates dissatisfaction with usability and integration challenges with other software platforms. Performance assessments reveal issues with segmentation accuracy, particularly in handling complex anatomical structures and variations in MRI images. Overall, the current system's weaknesses pose significant challenges in achieving accurate and reliable hippocampal segmentation, highlighting the need for a more robust and efficient software solution to address these shortcomings and improve outcomes in neuroimaging research and clinical diagnosis.

### 3.3 FEASIBILTY STUDY

A feasibility study assesses the viability of a proposed software project, considering four key dimensions. Technical feasibility evaluates if the project can be implemented using available technology and resources. Economic feasibility determines the cost- effectiveness of development and deployment. Operational feasibility examines if the software aligns with organizational processes and user needs. Schedule feasibility ensures the project can be completed within the specified timeframe. By analyzing these dimensions, stakeholders can make informed decisions regarding the project's potential success and resource allocation.

#### 3.3.1 Technical Feasibility

Technical feasibility evaluates the practicality of implementing the software project using existing technology and resources. It assesses if the proposed solution can be developed within the constraints of available hardware, software, and expertise. Factors such as compatibility with existing systems, scalability, and performance requirements are considered. Additionally, technical feasibility examines if the necessary tools and infrastructure are readily accessible to support the development and deployment of the software. By addressing these considerations, stakeholders can determine if the project is technically feasible and identify any potential challenges or limitations that may need to be addressed during implementation.

#### 3.3.2 Time Schedule Feasibility

Time Schedule Feasibility assesses if the project can be completed within the allocated timeframe. It considers resource availability, dependencies, and potential risks to create a realistic project schedule. By evaluating these factors, stakeholders can ensure that the project meets its deadlines and aligns with organizational goals.

**3.3.3 Operational Feasibility**

Operational feasibility evaluates the practicality of implementing the software within the organization's existing processes. It assesses user acceptance, ease of integration with current systems, and potential impacts on daily operations. By considering these factors, stakeholders can determine if the software solution aligns with operational objectives and user needs.

#### 3.3.4 Implementation Feasibility

Implementation feasibility assesses the practicality of executing the software project. It evaluates the availability of resources, expertise, and infrastructure required for development and deployment. By analyzing these factors, stakeholders can determine if the project can be successfully implemented within the organization's constraints and objectives.

## SYSTEM DESIGN

### 4.1 SYSTEM ARCHITECTURE DESIGN

#### 4.1.1 Flow Chart

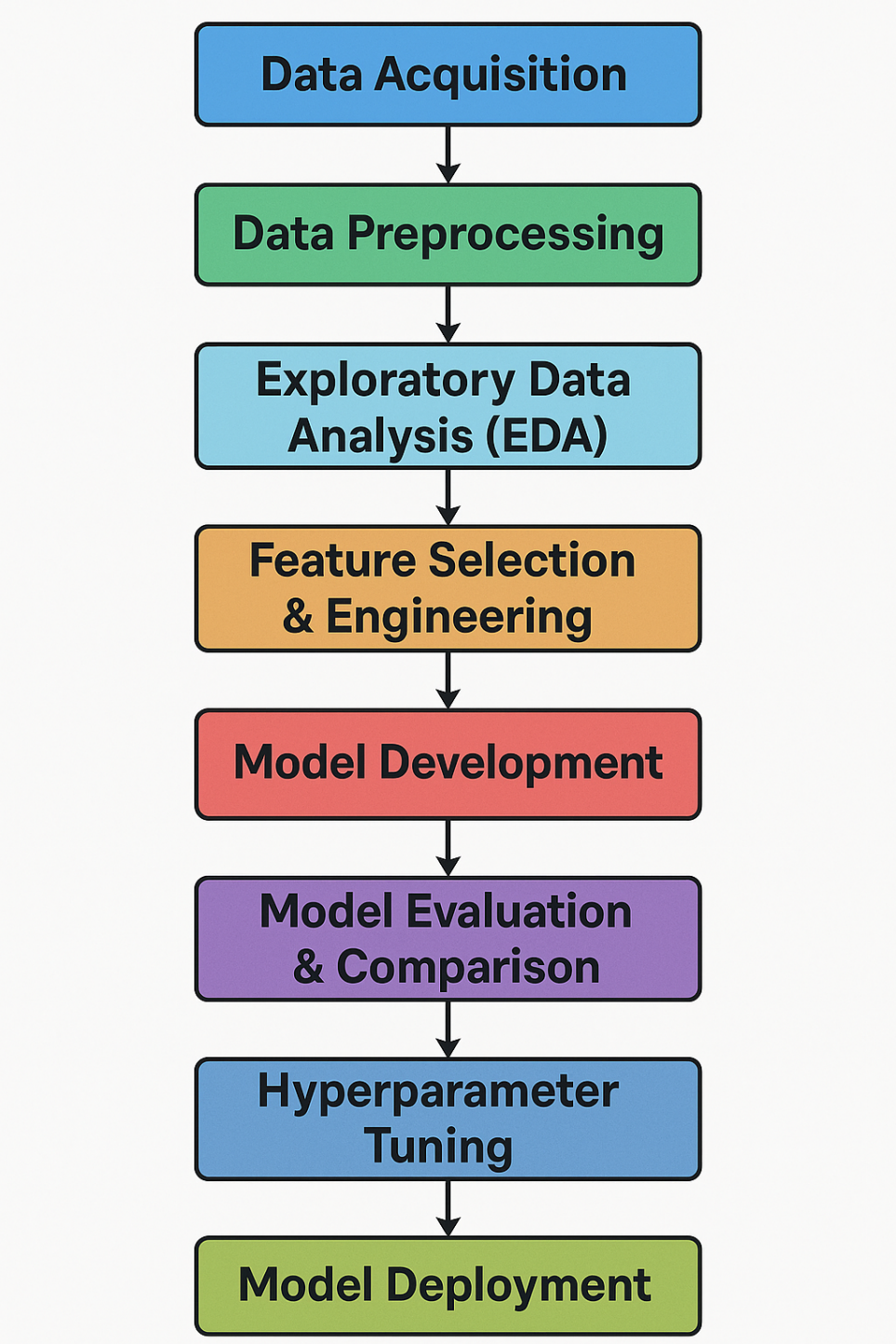


Fig 6.1 Flow Chart

**4.1.2 Distribution of categorical features**

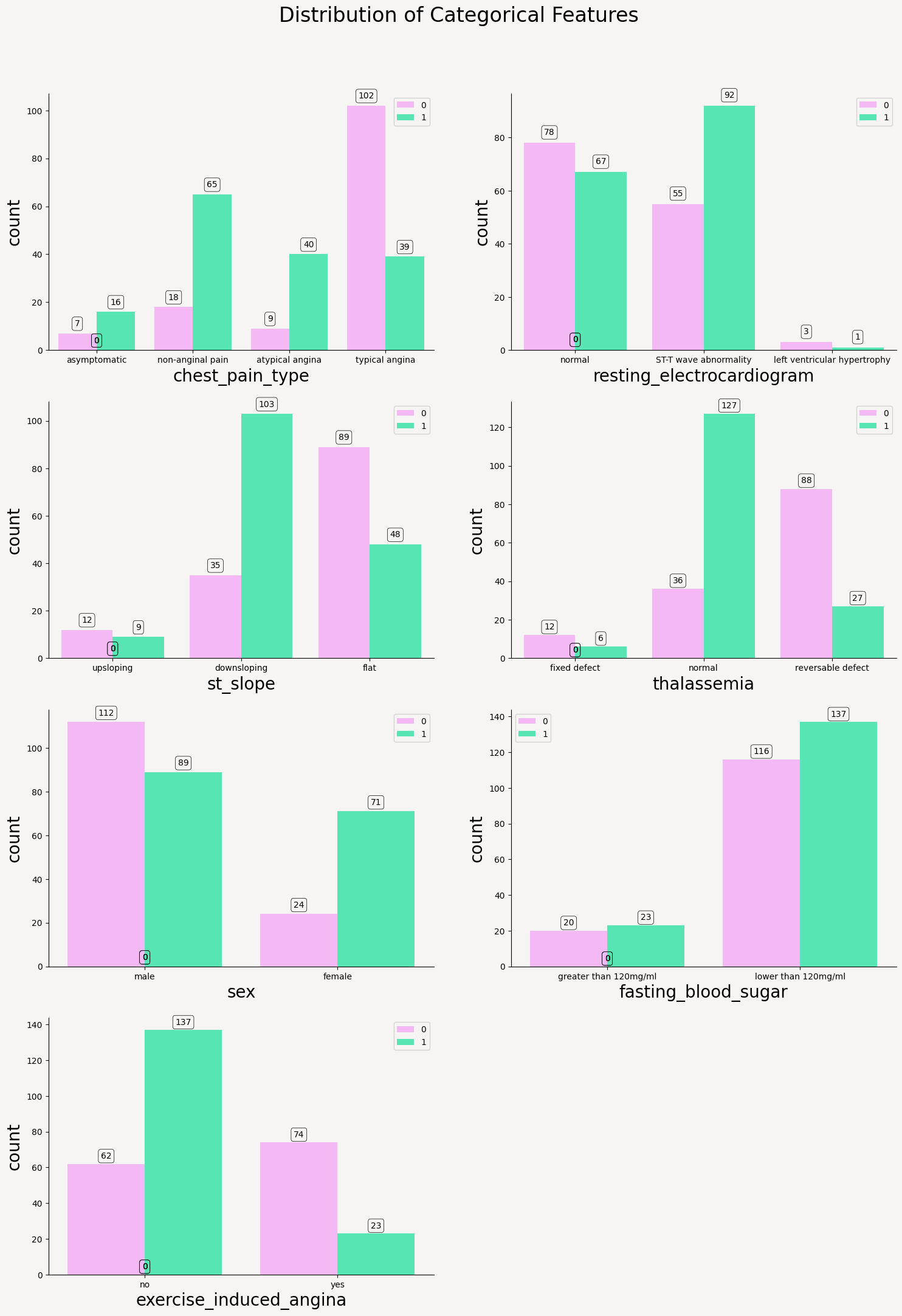
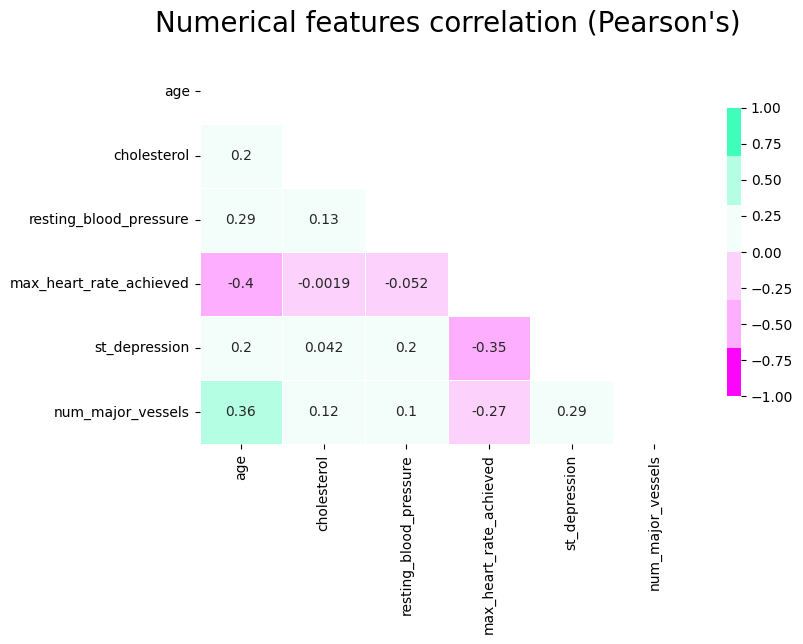
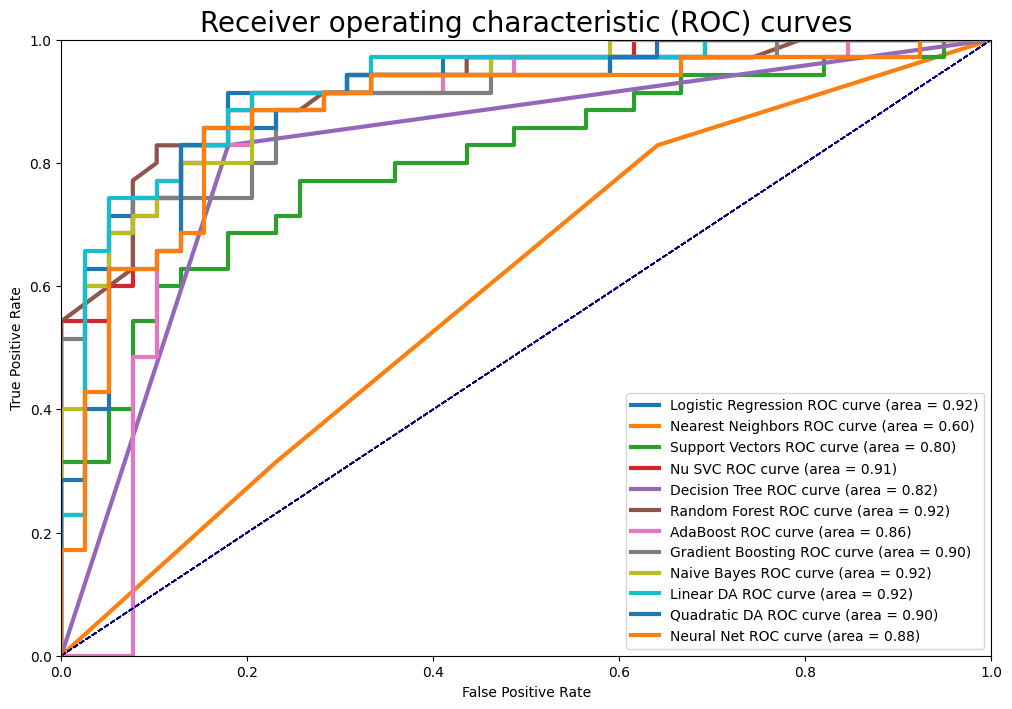


Fig 4.2 Categorical features

4.1.3 Numerical feature Corelation

4.1.4 ROC curves



## FUTURE ENHANCEMENT

### 5.1 IMPLEMENTATION ENVIRONMENT

For implementation we have used:

1. Python Env. and VS code setup
2. Load DL model and UI setUp

### 5.2 CODING STANDARDS

establishing coding standards is crucial to maintain consistency, readability, and scalability across the codebase. Adhering to these standards enhances collaboration among developers, facilitates code maintenance, and ensures the long-term viability of the software solution. Firstly, consistent naming conventions for variables, functions, and classes improve code clarity and understanding. Descriptive and meaningful names aid in comprehending the purpose and functionality of different components. Secondly, maintaining uniform formatting guidelines, including indentation, spacing, and line length, promotes code readability and organization. Consistent formatting reduces cognitive load and makes the codebase more approachable for developers.Thorough documentation, including docstrings and comments, is essential for explaining complex logic, function parameters, and usage examples. Well-documented code accelerates onboarding for new developers and facilitates code reviews and collaboration among team members. Effective error handling mechanisms ensure the robustness and reliability of the application. Properly managed errors with meaningful error messages enhance user experience and facilitate troubleshooting and debugging. Lastly, integrating version control systems like Git and implementing automated testing practices ensure code quality and stability throughout the development lifecycle. By adhering to these coding standards, the project can achieve a high level of maintainability, scalability, and overall software quality.

## CONCLUSION

In this project, we successfully developed and evaluated machine learning models for the detection of heart disease based on clinical and demographic data. Through data preprocessing, feature selection, and comparative analysis of multiple algorithms, we were able to identify a model that offers high accuracy and reliability in predicting the presence of heart disease. The results demonstrate the potential of machine learning as a supportive tool for early diagnosis and risk assessment in the healthcare sector. While the current model shows promising performance, further improvements can be made by incorporating larger, more diverse datasets and integrating real-time clinical data. This project serves as a foundation for future research and development of AI-driven health diagnostic systems.

## REFERENCES

**[1] S. Palaniappan and R. Awang, “Heart Disease Prediction Using Data Mining Techniques,” International Journal of Computer Science and Network Security (IJCSNS),vol. 8, no. 4, pp. 343–350, 200 [2] M. A. Jabbar, B. L. Deekshatulu, and P. Chandra, “A Comparative Study of Machine Learning Algorithms for Predicting Heart Disease,” in International Conference on Advanced Computing and Communication Systems, 2013, pp. 1–5. [3] R. K. Kavitha, G. S. S. Krishnan, and S. S. Kumar, “An Efficient Heart Disease Prediction System Using Machine Learning Techniques,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 6, no. 2, pp. 123–128, 2017. [4] A. K. Dwivedi, “A Deep Learning Approach for Heart Disease Prediction Using Convolutional Neural Networks,” Journal of Medical Systems, vol. 44, 2020.**