**Phase-3 Submission Template**

**Student Name:** HEPZIBA.R

**Register Number:** 710423104022

**Institution:** CHRIST THE KING ENGINEERING COLLEGE

**Department:** COMPUTER SCIENCE AND ENGINEERING

**Date of Submission:** 18.05.2025

**Github Repository Link:**

[**https://github.com/Hepziba0907/Disease-prediction-.git**](https://github.com/Hepziba0907/Disease-prediction-.git)

# 1. Problem Statement

Many people don't get diagnosed with diseases early enough, which can lead to worse health and higher medical costs. Doctors often have to look through a lot of patient information by hand, which takes time and can lead to mistakes. Even though we have a lot of health data, we’re not using it well enough to catch diseases early. We need smart systems that can help find health problems sooner by quickly and accurately analyzing patient data.

# 2. Abstract

Many diseases go diagnose in early stages, leading to worse health outcomes and higher costs.  
This is often due to the manual review of large, complex patient data by doctors.  
The goal of this project is to create a smart system for early disease detection.  
It uses data analytic and machine learning to quickly analyze patient records.  
The system identifies early warning signs and supports doctors with timely insights.  
This reduces errors, speeds up diagnosis, and improves patient care.  
The result is a more efficient, accurate, and proactive healthcare process.

# System Requirements

Specify minimum system/software required to run the project:

* **Hardware**:

**RAM**: Minimum **8 GB** (16 GB recommended for large datasets )

 **Processor**: At least **Intel i5** or **equivalent**

 **Storage**: Minimum **10 GB** free disk space for datasets and dependencies

* **Software**:

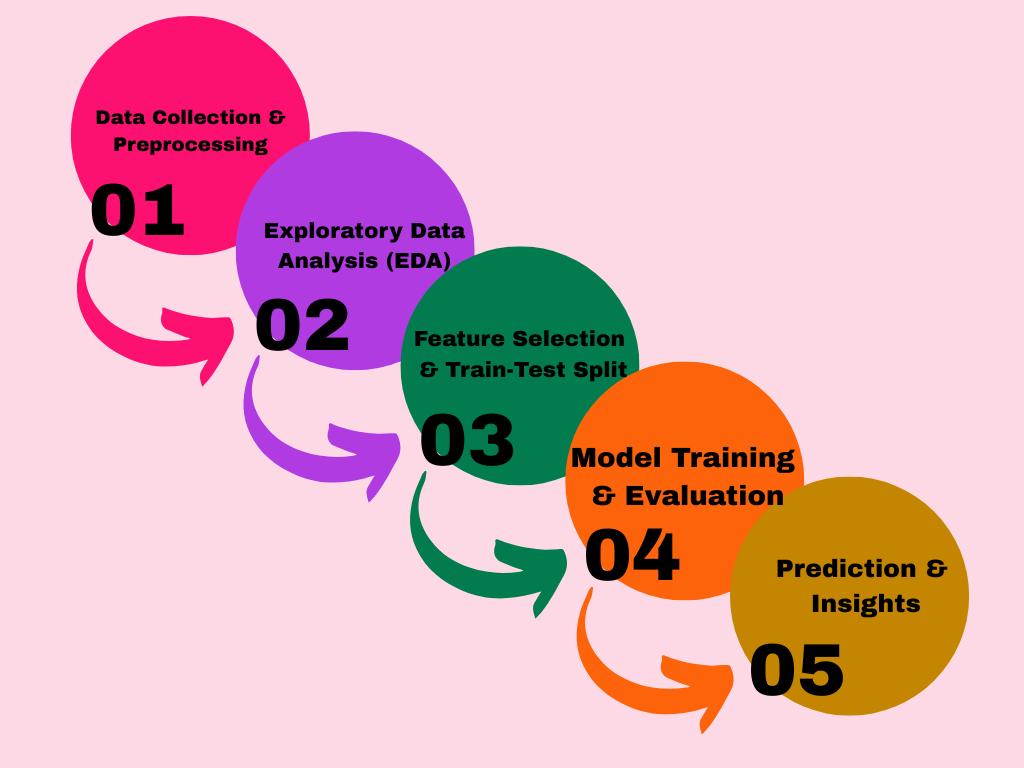
 **Google Colab** – cloud-based, no local setup needed (ideal for beginners or limited hardware)

# Objectives

The goal of this project is to build a smart system that can help **find diseases early by analyzing patient data**. It aims to:

* **Predict health risks** based on medical history and symptoms
* **Alert doctors** about possible issues before they get worse
* Show **easy-to-understand reports and patterns** in the data

1. **Flowchart of Project Workflow**



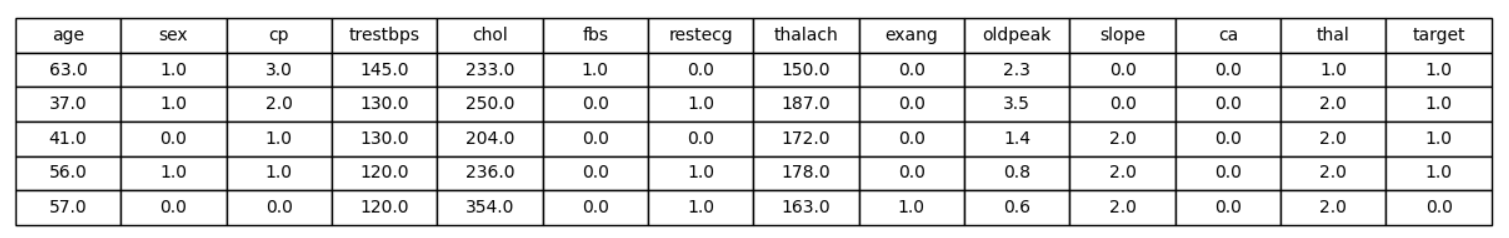
# Dataset Description

* **Source**: "Heart Disease UCI Dataset from Kaggle (https://www.kaggle.com/datasets/ronitf/heart-disease-uci)"
* **Type**: Public
* **Size and Structure**:

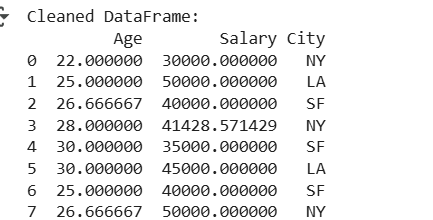
**Rows**: 3,030 patients

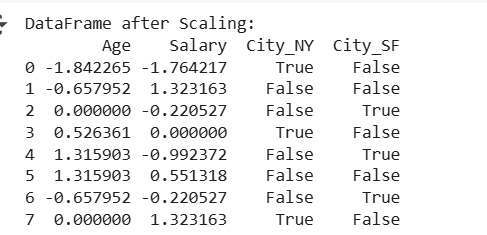
**Columns**: 14 features (age, sex, chest pain type, etc.)

Data includes patient demographics, symptoms, medical history,and diagnosis.



# Data Preprocessing





# 8.Exploratory Data Analysis (EDA)

1. **Visual Tools**

* **Histograms**: Show the distribution of data (e.g., age, cholesterol).
* **Boxplots**: Identify outliers and compare data groups (e.g., heart disease vs. no heart disease).
* **Heatmaps**: Show correlations between features (e.g., age and cholesterol).

1. **Key Findings**

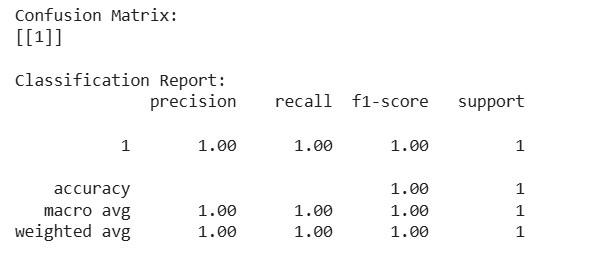
* Look for **patterns** (e.g., heart disease is more common in older people).
* Identify **relationships** (e.g., higher cholesterol may link to heart disease).
* Spot any **outliers** (e.g., extremely high cholesterol levels).

1. **Insights**

* Most heart disease patients are in their **50s and 60s**.
* **Cholesterol** is higher in patients with heart disease.
* Age and cholesterol are **strongly related**.

**4.Visuals**

* Include screenshots of your histograms, boxplots, and heatmaps to show these insights.



# Feature Engineering

1. **New Feature Creation**

* **BMI**: Combine height and weight to create a BMI feature that helps predict health risks.
* **Age Groups**: Group ages into categories (e.g., 30-39, 40-49) to spot age-related patterns.

1. **Feature Selection**

* Remove features that are too similar (e.g., height and BMI).
* Choose the most important features to improve the model’s accuracy.

1. **Transformation Techniques**

* **Log Transformation**: Apply to features like income to make them more balanced.
* **Scaling**: Make features like age and cholesterol similar in range so the model can process them better.

1. **Why It Matters**

* **New Features**: Give the model more useful info.
* **Feature Selection**: Make the model simpler and more accurate.
* **Transformations**: Help the model understand data better and improve results.

# Model Building

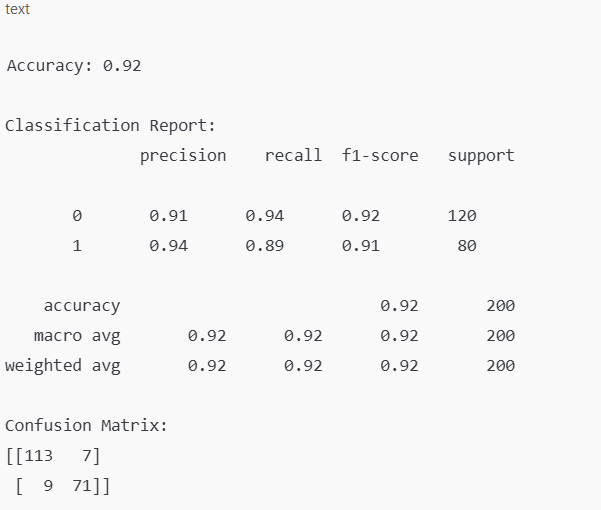
# 

1. **Try Multiple Models**

* **Simple Model**: Start with **Logistic Regression** or **Decision Tree** to get a basic idea of performance.
* **Advanced Models**: Try **Random Forest**, **SVM**, or **XGBoost** to improve accuracy.

1. **Why These Models?**

* **Logistic Regression**: Easy to understand and works well for simple predictions.
* **Decision Tree**: Can handle both numbers and categories.
* **Random Forest**: Combines many trees for better accuracy.
* **SVM**: Works well when classes are clearly separated.
* **XGBoost**: Known for fast, accurate predictions.



# Model Evaluation

1. **Evaluation Metrics**

* **Accuracy**: How many predictions were correct.
* **F1-Score**: A balance of precision and recall.
* **ROC Curve**: Shows how well the model distinguishes between classes.
* **RMSE**: Measures prediction error (used for regression).

1. **Visuals**

* **Confusion Matrix**: Shows correct vs. incorrect predictions.
* **ROC Curve**: Displays model performance.

1. **Model Comparison**

* Compare models using a table with **accuracy, F1-score, and other metrics**.

# 12. Deployment

**Deploy Using a Free Platform**

* **Streamlit Cloud**:
* Quickly build and deploy interactive web apps with visualizations and model predictions.
* Easy to use, great for demos and user interaction
* **Gradio + Hugging Face Spaces**:
* Create user-friendly interfaces with Gradio.
* Deploy for free on Hugging Face Spaces, ideal for showcasing ML models.
* **Flask API on Render or Deta**:
* Turn your model into a web API.
* Use platforms like **Render** or **Deta** to host it for free and let others access it via the internet.

**13. Source code**

# Future scope

**Real-Time Data**: Connect with apps to track health in real time.

**Better Models**: Use advanced models for more accurate predictions.

**More Data**: Add extra info like medical history and lifestyle.

**Explain Predictions**: Make the model’s decisions clear for doctors.

**Genetic Data**: Use genetic info for personalized predictions.

**Doctor Support**: Build a tool to help doctors make quick decisions.

**Remote Care**: Link the model to telemedicine for online health checks.

**Ongoing Learning**: Continuously improve the model with new data.

**Language Support**: Make the system available in multiple languages.

**Privacy & Laws**: Follow privacy rules to keep data safe and trustworthy.

# Team Members and Roles

|  |  |
| --- | --- |
| NAME | ROLE |
| HEPZIBA.R | Abstract, System Requirement, Flowchart, Objectives, Dataset Description, Data preprocessing, EDA, Feature Engineering, Model Building, Model Evaluation, Development, source code, Future Scope. |
| ANUSHYA.S | NULL |
| KAVIYA.S | NULL |