Phase-1

Student Name: Geetharani C

Register Number: 71772317112

Institution: Government College of Technology, Coimbatore

Department: Computer Science and Engineering

Date of Submission: 13-04-2025

1.Problem Statement

In many healthcare systems, disease diagnosis is primarily reactive, occurring after noticeable symptoms emerge. This often results in delayed treatments, increased healthcare costs, and poorer patient outcomes. The lack of predictive tools for early disease detection is especially problematic for chronic and life-threatening conditions such as heart disease, diabetes, and cancer. The aim of this project is to leverage AI technologies to predict potential diseases using patient data after some early symptoms . This proactive approach can drastically improve early intervention, personalize treatments, and ultimately save lives.

2.Objectives of the Project

- Develop a machine learning model capable of predicting the likelihood of specific diseases based on patient data.
- Identify and analyze key health indicators contributing to the early onset of diseases.
- Provide actionable insights to healthcare providers and patients to support early intervention and preventive care.
- Build a simple, user-friendly interface (optional) for users to input data and receive disease risk predictions.

3.Scope of the Project

Features to Analyze /Build:

- Patient demographics (age, gender, etc.)
- Vital signs (blood pressure, glucose levels, etc.)
- Medical history and lifestyle factors (smoking, physical activity)
- Predictive models for common diseases (e.g., diabetes, heart disease)

Constraints:

- Use of publicly available datasets
- Limited to offline prediction models (no real-time data streaming)

4.Data Sources

- Dataset Source: Kaggle (e.g., Heart Disease UCI, Diabetes Dataset)
- Availability: Public
- Type: Static (downloaded and used as-is)
- Additional Notes: May use synthetic data to supplement specific attributes if necessary.

5.High-Level Methodology

Data Collection

Data will be sourced primarily from public repositories such as **Kaggle** (e.g., Heart Disease UCI dataset, Pima Indians Diabetes dataset). These datasets are well-structured and widely used in research. Additional data may be synthetically generated using Python libraries like scikit-learn's make_classification() to simulate missing variables or balance class distributions.

Data Cleaning

Key insights and predictions will be visualized through:

- Confusion matrices
- ROC curves
- Feature importance plots
- Dashboards or Jupyter notebooks with interactive visualizations using **Plotly** or **Streamlit** (optional)

Deployment

If feasible, the final model will be deployed as a simple **Streamlit web app** or a **Jupyter Notebook-based interactive report**. This will allow users to input test data and view predictions directly, demonstrating practical applicability.

6.Tools and Technologies

- **Programming Language:**
  Python, HTML, CSS
- **Notebook/IDE:**
  Google Colab and Jupyter Notebook (for development, testing, and visualization)
- **Libraries:**
  - **Data Processing:** pandas, numpy
  - **Visualization:** matplotlib, seaborn, plotly
  - **Modeling:** scikit-learn, xgboost, statsmodels
  - **Others (optional):** imbalanced-learn (for handling class imbalance), joblib or pickle (for model saving)
- **Tools for Deployment:**
  - **Streamlit** – For building an interactive web app to demonstrate model predictions
  - **Flask** – If a lightweight API is needed for backend model serving
  - **Gradio** – For rapid prototyping of model interfaces

Additional Frontend Technologies

HTML – For custom templates (e.g., Flask + Jinja)
CSS – For UI styling

7.Team Members and Roles

Dharani A - Data Collection & Research

- Collect relevant datasets from Kaggle or UCI
- Analyze and clean the dataset (handle missing values, duplicates)
- Perform Exploratory Data Analysis (EDA) to identify patterns
- Create visual reports on data insights

Deliverables: Cleaned dataset, EDA report, documentation of key findings

Sibitha S - Model Development

- Train and tune machine learning models (e.g., logistic regression, randomforest, XGBoost)
- Perform feature selection/engineering
- Save the final trained model using pickle or joblib
- Evaluate model using metrics (accuracy, precision, recall, ROC-AUC)

Deliverables: Final ML model, evaluation results, model.pkl file

Geetharani C - Frontend & UI Developer

- Build the frontend of the web app using Streamlit (or Gradio)
- Create user input forms for patient health data
- Display prediction results and visual insights clearly
- Style the app with user-friendly design and layout

Deliverables: app.py (main Streamlit app file), UI screenshots

Anusha S - Integration & Backend Developer

Responsibilities:

- Integrate the trained model with the app
- Handle input preprocessing and output formatting in the app
- Ensure input validation and manage exceptions (e.g., empty fields)
- Prepare API endpoint (if using Flask or FastAPI instead of Streamlit)

Deliverables: working integrated app, app logic scripts

Devisri V - Deployment & Documentation

- Deploy the final app on a public platform (Streamlit Cloud / Hugging Face Spaces)
- Write final project documentation (overview, problem statement, methodology, tools, etc.)
- Prepare a final presentation (PowerPoint / Google Slides)
- Record a demo video (optional, if required for submission)

Deliverables: hosted app link, documentation PDF, presentation deck