நான் முதல்வன்

ORACLE

AdroIT Technologies®
Innovative Solutions Pvt LTD

Phase-3 Submission Template

Student Name: C.Malavika

Register Number: 620123106061

Institution: AVS Engineering College

Department: ECE

Date of Submission: 17.05.2025

Github Repository Link: https://github.com/Malavika09-04/Malavika.git

---

Transforming Healthcare with AI-Powered Disease Prediction Based on Patient Data

1. Problem Statement

Traditional diagnostic systems in healthcare rely heavily on manual evaluations by medical professionals, which can be time-consuming, inconsistent, and prone to human error. There is a growing need for automated, accurate, and early disease prediction systems that can assist doctors and healthcare providers by analyzing patient data efficiently.

2. Abstract

This project aims to develop an AI-powered system capable of predicting diseases based on patient medical data. Using machine learning models trained on historical healthcare datasets, the system can identify patterns and risk factors associated with various diseases. The tool is designed to assist healthcare providers with early diagnosis and treatment planning, ultimately improving patient outcomes and reducing workload on medical staff.

3. System Requirements

Hardware Requirements:
- Processor: Intel i5 or higher
- RAM: 8 GB minimum
- Storage: 100 GB
- GPU (for deep learning models): Optional

Software Requirements:
- Python 3.x
- Jupyter Notebook/VS Code
- Libraries: NumPy, Pandas, scikit-learn, Matplotlib, Seaborn, TensorFlow/PyTorch (optional), Flask/Streamlit (for deployment)

4. Objectives

- To collect and preprocess patient medical data.
- To explore and analyze the data for insights.
- To build machine learning models that predict diseases.
- To evaluate model performance with appropriate metrics.
- To deploy the model in a user-friendly interface.
- To assist healthcare professionals in early diagnosis.

5. Flowchart of Project Workflow

Patient Data → Data Preprocessing → EDA → Feature Engineering → Model Building → Model Evaluation → Deployment

6. Dataset Description

You can use publicly available datasets like:
- UCI Machine Learning Repository (e.g., Heart Disease, Diabetes datasets)
- Kaggle Datasets (e.g., Stroke Prediction, Liver Disease)

Example fields:
- Age, Gender

- Blood Pressure
- Glucose Level
- Cholesterol
- Heart Rate
- Previous Diagnosis
- Symptoms

7. Data Preprocessing

- Handling missing values
- Encoding categorical variables
- Normalization/Standardization
- Removing duplicates
- Outlier detection

8. Exploratory Data Analysis (EDA)

- Statistical summary (mean, median, std deviation)
- Visualization: histograms, boxplots, correlation heatmaps
- Disease incidence by age, gender, lifestyle factors

9. Feature Engineering

- Feature selection using correlation or feature importance
- Creating new features like BMI, risk scores
- Dimensionality reduction (PCA, if needed)

10. Model Building

Models used can include:

- Logistic Regression
- Random Forest
- Support Vector Machines
- XGBoost
- Neural Networks (if dataset is large enough)

11. Model Evaluation

- Accuracy, Precision, Recall, F1-score

- Confusion Matrix - ROC-AUC Curve - Cross-validation results

12. Deployment

Deploy the final model using: - Flask or FastAPI (for API) - Streamlit or Dash (for interactive UI) - Host on Heroku, Render, or local server

13. Source Code

Organize code as: /disease_prediction

```
├──── data/
├──── preprocessing.py
├──── eda.ipynb
├──── model.py
├──── evaluation.py
├──── app.py (for deployment)
└──── requirements.txt
```

14. Future Scope

- Integrate more comprehensive datasets including genetic and lifestyle data.
- Extend prediction to multiple diseases.
- Use deep learning for medical image-based diagnosis.
- Integrate wearable device data for real-time predictions.
- HIPAA/GDPR compliance for real-world healthcare use.

15. Team Members and Roles

- Project Lead: Oversees the project and ensures timely delivery.
- Data Engineer: Handles data collection and preprocessing.
- ML Engineer: Builds and evaluates predictive models.
- Frontend Developer: Designs UI for model interaction.

- Backend Developer: Integrates the model into the application.
- Research Analyst: Performs EDA and feature selection.