PHASE- 2

Project Title: Transfroming healthcare with AI- Powered disease prediction based on patient data.

Github: https://github.com/nithyapriya- 136/transforming- healthcare- with- AI- powered- disease- prediction- based- on- patient- data.git

1.Problem Statement

- Healthcare systems often struggle to predict diseases early due to lack of integrated data analysis. Manual diagnosis can be time- consuming and error- prone. By leveraging AI to predict diseases using patient data, healthcare providers can identify conditions early, improve treatment plans, and enhance patient outcomes.

- This project focuses on building a machine learning model to predict the presence or risk level of a specific disease using patient demographic, lifestyle, and clinical attributes.

- Significance: Enables preventive care, reduces healthcare costs, and improves quality of life.

2.Project Objectives

- Develop a predictive model that identifies potential diseases based on patient data.

- Improve diagnostic efficiency using machine learning algorithms.

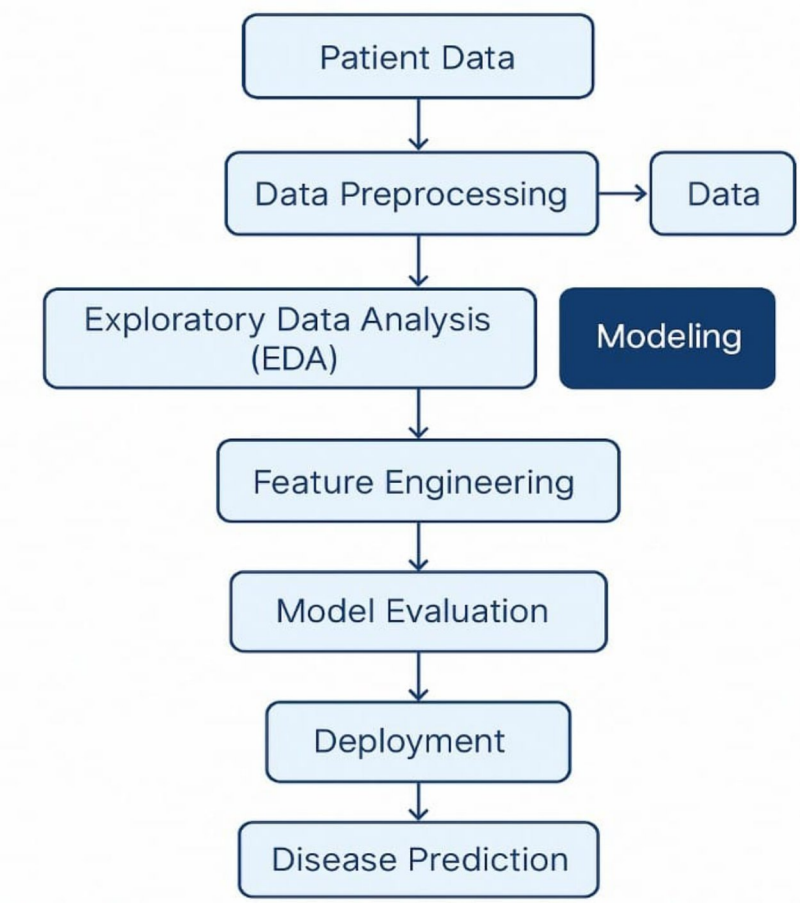
- Deliver actionable insights to assist healthcare professionals in decision- making.

- Enhance early detection and prevention strategies.

- Build an accurate AI model for disease prediction.

- Identify the key features influencing disease risk.
- Interpret model outputs to guide clinical decisions.
- Deploy a user- friendly web interface for real- time predictions.
- Ensure ethical AI practices and patient data privacy

3.Flowchart of the Project Workflow



4.Data Description

- Dataset Name:
  Could be sourced from open healthcare datasets.
- Source:
  Public repositories like UCI, Kaggle, or hospital collaborations.
- Data Type:
  Structured tabular (e.g., CSV).
- Records and Features:
  1,00– 100,0 records with 10– 50 features.
- Target Variable:

Presence/absence or probability of diseases.(e.g.,stroke)

- Features:
  id,Age, gender, hypertension,heart disease,work type,bmi,even married,smoking status ,stroke,etc..
- Dataset link:
https://www.kaggle.com/code/manarmohamed11/stroke- prediction- eda?scriptVersionId=236663015&cellId=2

5.Data Preprocessing

- Handle missing/null values.
- Convert categorical variables via one- hot encoding or label encoding.
- Scale numerical features using StandardScaler/MinMaxScaler.
- Remove or cap outliers (using z- scores or IQR).
- Address imbalanced data using SMOTE, oversampling, or class weights.

6.Exploratory Data Analysis (EDA)

- Univariate:
  Distribution of disease status, age, BMI, etc.
- Bivariate:
  Compare features (e.g., glucose vs. disease outcome).
- Multivariate:
  Correlation heatmaps, risk factor clusters.
- Key Insights:
  - High cholesterol, older age, and sedentary lifestyle may correlate with higher disease risk.
  - Gender- specific trends may exist in disease occurrence.

7.Feature Engineering

- Create composite indicators (e.g., BMI from weight & height).

- Derive binary flags (e.g., smoker = yes/no).
- Remove redundant or highly correlated features.
- Encode interaction effects (e.g., age × cholesterol).

8.Model Building

- Algorithms:
  - Logistic Regression: baseline model.
  - Random Forest / XGBoost: for non- linear patterns
    Neural Networks (optional for large data).
- Train- Test Split: 80/20 split using train_test_split.
- Cross- Validation: 5- fold or 10- fold to ensure robustness.
- Evaluation Metrics:
  - Accuracy
  - Precision, Recall, F1- score
  - AUC- ROC Curve
  - Confusion Matrix

9.Visualization of Results & Model Insights

- Feature Importance:
  - Visualized via bar charts (Random Forest, SHAP values)
  - Identify top 5– 10 contributing factors
- Model Performance:
  - Compare evaluation metrics across models
  - ROC curves and confusion matrices for final model
- Residual Analysis:
  - Check for prediction bias (e.g., gender, ethnicity)

10.Deployment & Interface

- Tool:
  Gradio or Streamlit

- Features:
    - Input patient data via sliders/forms
    - Get real- time disease prediction and risk score
    - Display risk explanation (via SHAP or LIME)

11.Tools and Technologies Used

- Language:
  Python

- Environment:
  Google Colab, Jupyter Notebook

- Libraries:
    - pandas, numpy – Data handling
    - matplotlib, seaborn, plotly – Visualization
    - scikit- learn, xgboost, shap – Modeling
    - gradio, streamlit – UI deployment

12.Team Members and Contributions

- Clearly mention who worked on:

    - V.SANGEETHA:
        - Data cleaning
    - C.NITHYAPRIYA:
        - EDA (Exploratory Data Analysis)
    - S.K.LAKSHMIPRIYA:
        - Feature engineering
    - S.MAHALAKSHMI
        - Model development
        - Documentation and reporting