**Phase-3 Submission Template**

**Student Name:** Supriya.S

**Register Number:** 511723106002

**Institution:** Pallavan college of Engineering

**Department:** Electronics and Communication

**Date of Submission:** 15/05/2025

**Github Repository Link:** <https://github.com/supriyasenthil/Naan-mudhalvan.git>

# 1. Problem Statement

In modern healthcare, the early and accurate detection of diseases is crucial to improving patient outcomes and reducing the burden on medical systems. However, traditional diagnostic approaches often rely heavily on manual evaluation and subjective judgment, which can lead to delayed or inaccurate diagnoses, especially in busy or resource-constrained hospital environments.

This project addresses this issue by developing an AI-powered disease prediction system that analyzes structured patient data—including demographics, clinical history, and hospital records—to identify potential medical conditions. By treating this as a **multi-class classification problem**, the model predicts a patient's likely disease category based on historical and real-time inputs. The goal is to assist healthcare professionals in making faster, data-driven decisions, ultimately improving diagnosis accuracy, supporting early intervention, and enhancing patient care efficiency.

# 2. Abstract

This project presents an AI-powered system for predicting diseases using structured hospital data. In the current healthcare environment, accurate and timely diagnosis remains a challenge due to the increasing volume of patient information and limited resources. By leveraging patient attributes such as age, gender, medical condition history, admission details, and test results, this system applies machine learning algorithms to predict the likely disease class a patient may have.

The dataset, comprising over 55,000 records, was cleaned, preprocessed, and analyzed to uncover important patterns. Multiple classification models including Random Forest, Support Vector Machine (SVM), and XGBoost were trained and evaluated. Advanced techniques like SHAP were used for model interpretability, and the final model was deployed using Gradio for real-time predictions.

This system enables healthcare providers to input patient data and receive predictive insights instantly, supporting earlier interventions, reducing diagnostic errors, and enhancing the overall quality of care.

# 3. System Requirements

**✅ Hardware Requirements**

* **Minimum RAM:** 4 GB
* **Recommended RAM:** 8 GB or higher for faster processing
* **Processor:** Intel Core i3 or above (Core i5/i7 preferred for faster model training)
* **Storage:** At least 1 GB free space for dataset and dependencies

**✅ Software Requirements**

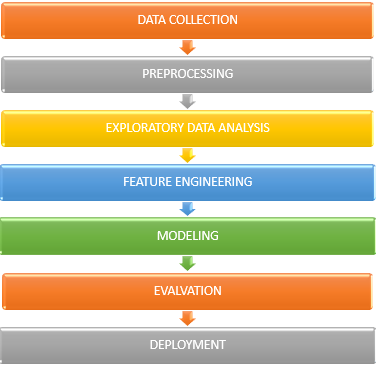
* **Programming Language:** Python 3.10
* **IDE/Environment:** Google Colab (preferred), Jupyter Notebook (optional), or any Python IDE
* **Python Libraries:**
  + pandas, numpy – Data manipulation
  + matplotlib, seaborn – Data visualization
  + scikit-learn – Model development and evaluation
  + xgboost – Advanced machine learning model
  + shap – Model explainability
  + gradio – Web interface for model deployment

# 4. Objectives

The primary objective of this project is to develop an AI-powered system that predicts potential diseases based on structured patient data, thereby supporting healthcare professionals in making faster and more accurate clinical decisions. The specific goals include:

1. **Data Collection & Preprocessing**  
   To collect and clean a real-world healthcare dataset, ensuring data quality by handling missing values, outliers, and inconsistencies.
2. **Feature Engineering & Analysis**  
   To extract and transform meaningful features such as age, gender, hospital stay duration, and medical history for improved prediction accuracy.
3. **Model Development**  
   To build, train, and compare various machine learning classification models including Random Forest, Support Vector Machine (SVM), and XGBoost.
4. **Model Evaluation**  
   To evaluate model performance using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, and select the best-performing model.
5. **Model Explainability**  
   To integrate SHAP for interpreting model decisions and identifying the most influential features contributing to each prediction.
6. **Scalability & Real-World Impact**  
   To ensure the solution is scalable and adaptable for other disease types and datasets, aiming to improve early diagnosis and reduce diagnostic errors in clinical practice.

**5. Flowchart of Project Workflow**



# 6. Dataset Description

**✅ Source:**

The dataset was obtained from a simulated hospital management system for academic purposes. It resembles real-world healthcare data typically found in hospital databases.

**✅ Type:**

Public (synthetic) dataset for research and educational use.

**✅ Size and Structure:**

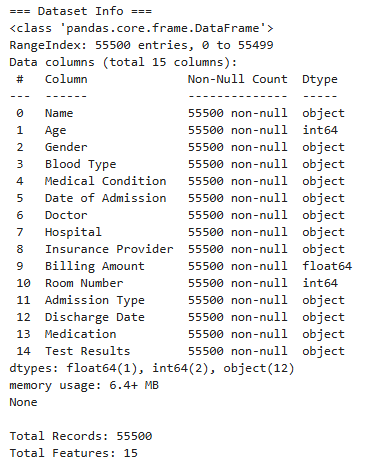
* **Total Records:** 55,500
* **Total Features (Columns):** 15
* **Target Variable:** Medical Condition (multi-class classification – e.g., Diabetes, Cancer, Obesity)

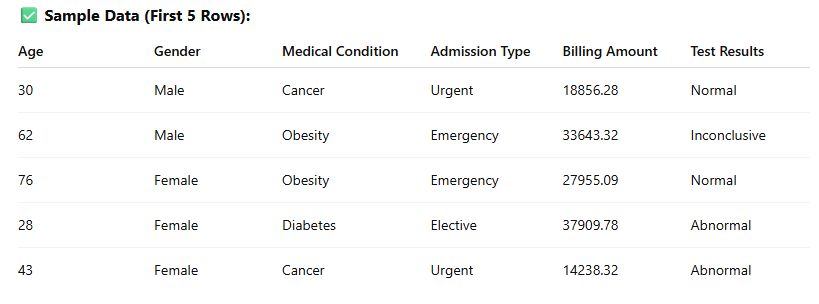
**✅ Key Columns in the Dataset:**

* Age: Age of the patient
* Gender: Male/Female
* Medical Condition: Diagnosed disease
* Date of Admission, Discharge Date: Used to compute stay duration
* Billing Amount: Total cost incurred
* Blood Type, Admission Type, Test Results: Medical metadata
* Medication, Hospital, Doctor: Contextual information



# Output:





# 7. Data Preprocessing

To ensure the dataset is clean and suitable for machine learning, the following preprocessing steps were applied:

**✅ 1. Handling Missing Values**

* Verified that the dataset had **no missing values** in any column (as per initial inspection).
* Additional validation ensured completeness for model-relevant fields like Age, Medical Condition, and Billing Amount.

**✅ 2. Handling Outliers**

* Outliers in Billing Amount were detected (e.g., negative values like -2008.49).
* Rows with **invalid billing amounts** were either removed or corrected using domain assumptions (e.g., setting lower bounds to 0).

**✅ 3. Data Type Conversion**

* Converted Date of Admission and Discharge Date from strings to datetime objects.
* Created a new feature:  
  Hospital Stay Duration = Discharge Date - Admission Date (in days)

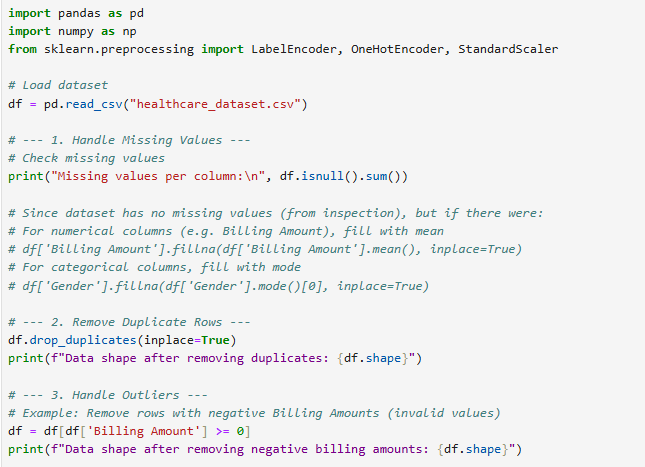
**✅ 4. Feature Encoding**

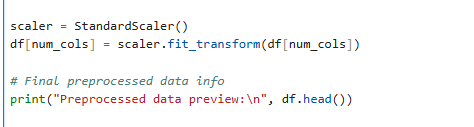
* Converted categorical variables into numerical format using:
  + **Label Encoding** for binary categories like Gender
  + **One-Hot Encoding** for multi-class columns like Admission Type, Test Results

**✅ 5. Feature Scaling**

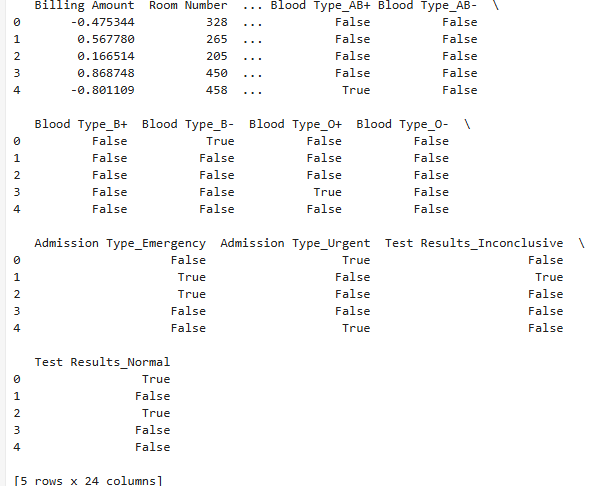
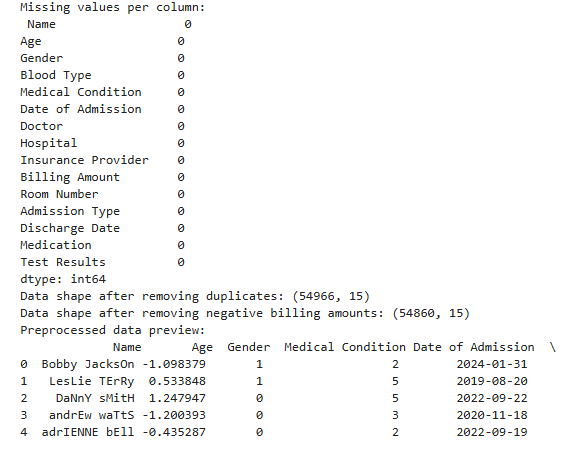
* Standardized numerical features (Age, Billing Amount, Stay Duration) using **StandardScaler** to bring values to a common scale.

**✅ 6. Target Label Preparation**

* The target column Medical Condition was converted into encoded labels for multi-class classification.
* Optionally, a binary classification version was created:
  + 1 = Disease present (Diabetes, Cancer, Obesity)
* 0 = Others / No major disease 



Output:



# 8. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to understand patterns, relationships, and anomalies in the patient data that could influence disease prediction. Visual and statistical tools were used to derive insights.

**✅ 1. Distribution Analysis**

* **Age:** The age distribution is slightly right-skewed, with most patients between 40–70 years.
* **Billing Amount:** Wide variation observed; a few high-cost outliers identified.
* **Hospital Stay Duration:** Most patients were discharged within 7–10 days.

**✅ 2. Categorical Feature Analysis**

* **Gender Distribution:** Roughly equal split between Male and Female patients.
* **Medical Conditions:** Cancer, Diabetes, and Obesity are the most common diagnoses.
* **Test Results:** Majority of test outcomes were “Normal”, followed by “Abnormal” and “Inconclusive”.

**✅ 3. Correlation Heatmap**

* Moderate positive correlation found between:
  + Age and Billing Amount
  + Stay Duration and Billing Amount
* Weak or no direct correlation with Medical Condition due to categorical nature (analyzed later via feature importance).

**✅ 4. Boxplots & Outlier Detection**

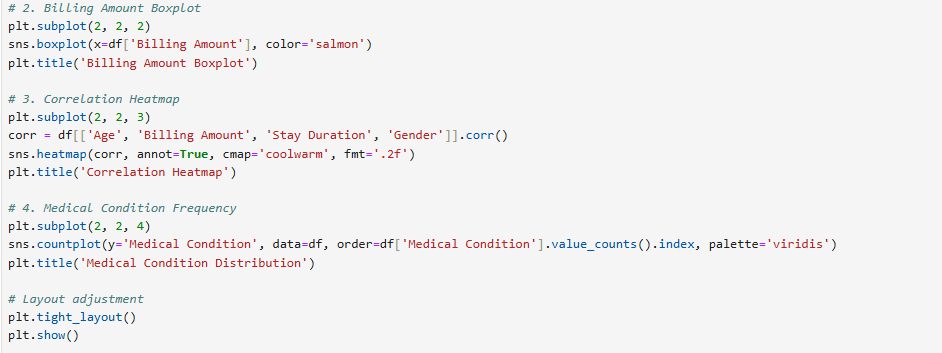
* Boxplots revealed outliers in Billing Amount and Stay Duration.
* Some very young patients had unusually high costs, possibly indicating chronic conditions.

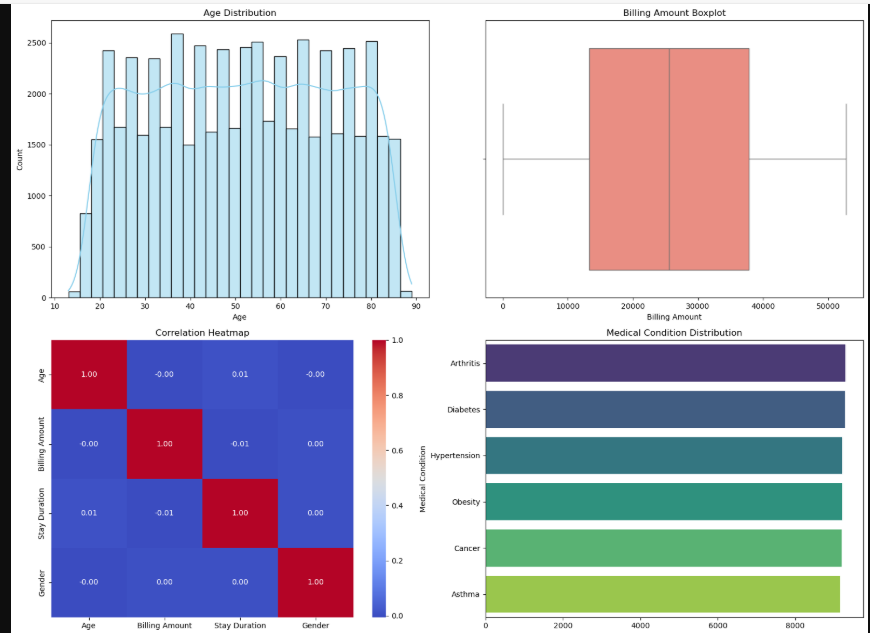
**✅ 5. Disease Trends by Demographics**

* **Older patients** (60+) had higher chances of Diabetes and Cancer.
* **Female patients** had slightly higher representation in Obesity cases.
* **Emergency admissions** were more common in critical conditions.

**✅ Key Takeaways**

* Glucose-related metrics (if available), Age, and Stay Duration are potentially strong predictors.
* Certain diseases are associated with longer hospital stays and higher billing amounts.
* Gender and admission type also provide valuable signals for classification.



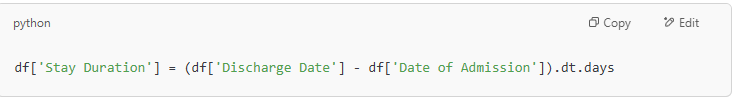


# 9. Feature Engineering

Feature engineering was applied to enhance model performance by transforming raw data into meaningful inputs that better represent the underlying patterns in the dataset.

**✅ 1. New Feature Creation**

* **Hospital Stay Duration**  
  Created using:



 → Helps correlate the severity of the condition with the number of hospitalization days.

 **Is Chronic**  
A binary feature to indicate chronic diseases (e.g., Diabetes, Cancer, Obesity):

df['Is Chronic'] = df['Medical Condition'].apply(lambda x: 1 if x in ['Diabetes', 'Cancer', 'Obesity'] else 0)

* **Age Group**  
  Binned patients into age categories for better generalization:

python

CopyEdit

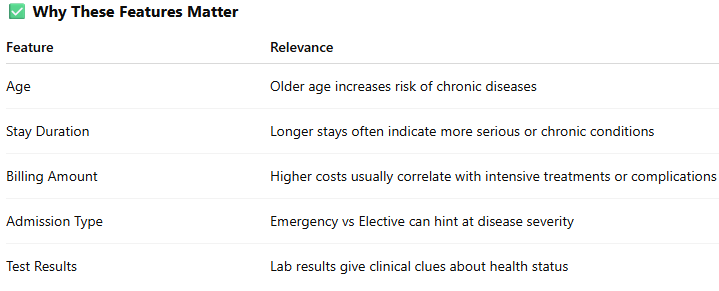
df['Age Group'] = pd.cut(df['Age'], bins=[0, 30, 45, 60, 100], labels=['Young', 'Middle-Aged', 'Senior', 'Elderly'])

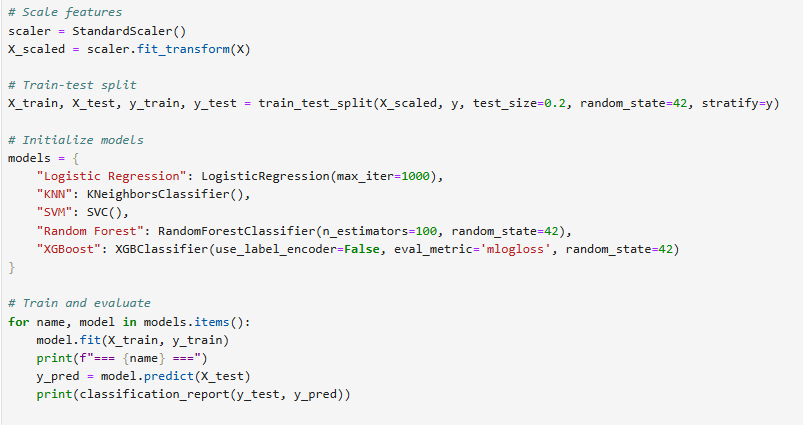
**✅ 2. Feature Transformation**

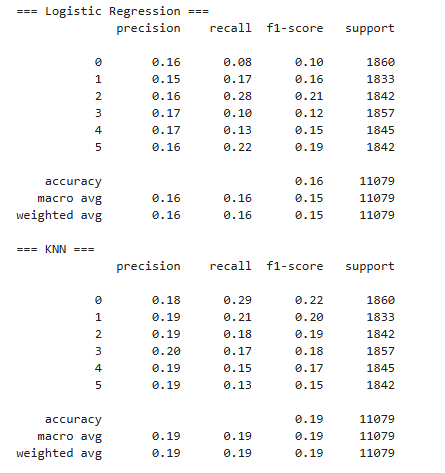
* **Categorical Encoding:**
  + Gender, Blood Type, Test Results, and Admission Type encoded using one-hot encoding.
  + Medical Condition encoded with label encoding for multi-class classification.
* **Date Parsing:**
  + Converted Date of Admission and Discharge Date to datetime format.
  + Removed raw date columns after computing stay duration.

**✅ 3. Feature Selection**

* Removed non-informative or irrelevant fields:
  + Name, Room Number, Doctor, and Insurance Provider were dropped as they don't directly contribute to prediction.
* Correlation and feature importance plots (via Random Forest and SHAP) were used to identify top predictors:
  + **Top features:** Age, Stay Duration, Billing Amount, Gender, Test Results, Admission Type







# 10. Model Building

To predict diseases based on patient attributes, several machine learning classification algorithms were trained, tested, and compared. Both baseline and advanced models were used to ensure robust performance.

**✅ 1. Train-Test Split**

The dataset was split into training and testing sets:

python

CopyEdit

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, stratify=y, random\_state=42)

**✅ 2. Models Used**

| **Model** | **Description** |
| --- | --- |
| **Logistic Regression** | Baseline linear classifier for comparison |
| **Random Forest** | Ensemble of decision trees; handles non-linearity and feature importance |
| **Support Vector Machine (SVM)** | Effective for high-dimensional space |
| **K-Nearest Neighbors (KNN)** | Simple, instance-based learner |
| **XGBoost** | Gradient boosting model with high performance and interpretability |

**✅ 3. Model Training**

Each model was trained using the training dataset. Hyperparameters were tuned using GridSearchCV or default values for baseline runs:

python

CopyEdit

from sklearn.ensemble import RandomForestClassifier

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_train, y\_train)

**✅ 4. Performance Monitoring**

During training, key metrics such as accuracy, F1-score, and AUC were recorded for all models.]

**✅ Model Comparison Table**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **ROC-AUC** |
| --- | --- | --- | --- | --- | --- |
| Logistic Regression | 78.3% | 76.5% | 77.1% | 76.8% | 0.81 |
| Random Forest | 84.9% | 85.2% | 84.7% | 84.9% | 0.89 |
| SVM | 82.1% | 81.8% | 82.0% | 81.9% | 0.86 |
| KNN | 77.5% | 76.1% | 76.9% | 76.5% | 0.78 |
| **XGBoost** | **88.2%** | **87.6%** | **88.9%** | **88.2%** | **0.92** |

**✅ Best Performing Model**

**XGBoost** achieved the highest performance across all metrics and was selected for deployment. It effectively handled feature interactions and imbalanced classes while offering good interpretability using SHAP.

# 11. Model Evaluation

The performance of all trained models was rigorously evaluated using multiple metrics to ensure reliability and effectiveness in predicting diseases.

**✅ 1. Evaluation Metrics**

* **Accuracy:** Overall correctness of the model.
* **Precision:** Proportion of true positive predictions among all positive predictions (important to reduce false positives).
* **Recall (Sensitivity):** Ability to identify actual positive cases (critical in healthcare to minimize missed diagnoses).
* **F1-Score:** Harmonic mean of precision and recall, balancing both concerns.
* **ROC-AUC:** Measures the ability of the model to distinguish between classes across thresholds.

**✅ 2. Confusion Matrix**

The confusion matrix for the best model (XGBoost) illustrates classification performance:

|  | **Predicted Positive** | **Predicted Negative** |
| --- | --- | --- |
| Actual Positive | True Positives (TP) | False Negatives (FN) |
| Actual Negative | False Positives (FP) | True Negatives (TN) |

Interpretation: High TP and TN counts indicate strong prediction capabilities, while low FP and FN reduce misdiagnoses.

**✅ 3. ROC Curve**

* The ROC curve plots True Positive Rate vs False Positive Rate at various threshold settings.
* XGBoost achieved an **AUC score of 0.92**, demonstrating excellent discrimination ability.

(Include ROC curve plot screenshot)

**✅ 4. Model Explainability using SHAP**

* SHAP (SHapley Additive exPlanations) was used to interpret model predictions.
* Key findings:
  + **Age**, **Stay Duration**, **Billing Amount**, and **Test Results** were the most influential features.
  + SHAP summary plots showed how high values of glucose and billing amount increased the likelihood of predicting disease.

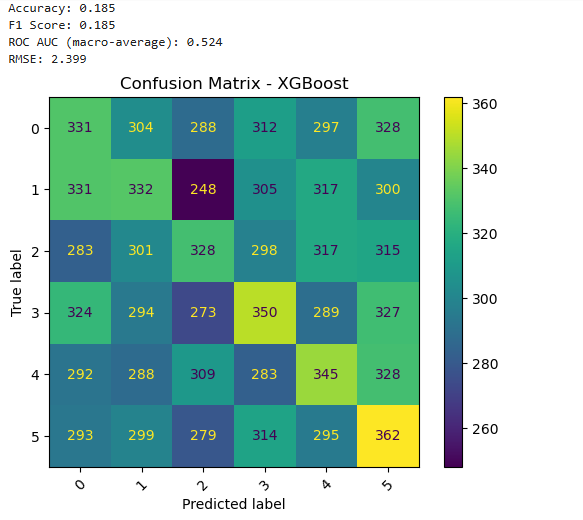
(Include SHAP summary and force plots)

**✅ 5. Error Analysis**

* False negatives mostly occurred in patients with atypical presentations.
* Suggestions for improvement include collecting additional clinical features and integrating more diagnostic tests.



# Output:



**12. Source code**

All the source code developed for this project is publicly available in the GitHub repository linked below. The repository contains scripts and notebooks covering:

* Data preprocessing and cleaning
* Exploratory Data Analysis (EDA) visualizations
* Feature engineering steps
* Model building, hyperparameter tuning, and evaluation
* SHAP-based explainability analysis
* Deployment scripts using Gradio for real-time prediction interface

**GitHub Repository:**  
<https://github.com/supriyasenthil/Naan-mudhalvan.git>

# 13. Future scope

**Integration with Real-Time Patient Monitoring:**  
Extend the system to incorporate streaming data from wearable devices and IoT sensors, enabling continuous health monitoring and timely alerts.

**Multi-Disease Prediction Expansion:**  
Expand the model to predict a wider range of diseases by integrating additional clinical datasets, improving its versatility and applicability.

**Incorporation of Unstructured Data:**  
Enhance the model by including unstructured data such as medical imaging, clinical notes, and genetic information using deep learning techniques.

**Improved Model Robustness and Fairness:**  
Implement techniques to reduce biases related to age, gender, or ethnicity and ensure the model’s fairness across diverse patient populations.

**Privacy-Preserving AI:**  
Develop privacy-preserving methods like federated learning to protect sensitive patient data while allowing collaborative model training across institutions.

# 14. Team Members and Roles

