Phase-2 Submission

Student Name: Jeeviksri R

Register Number: 410723104029

Institution: Dhanalakshmi college of engineering

Department: Computer Science and Engineering

Date of Submission: 07.05.2025

GitHub Repository Link: Jeevika_repository

AI – POWERED DISEASE PREDICTION

1. Problem Statement

- **Refined Problem Statement:**
  The goal is to use patient data such as demographics, medical history, and test results to predict the likelihood of specific diseases using AI, enabling early detection and intervention.

- **Type of Problem:**
  This is a classification problem, where patients are categorized based on the presence or risk of disease.

- **Why It Matters:**
  Early prediction improves patient outcomes, supports preventive care, reduces healthcare costs, and enables personalized treatment.

2. Project Objectives

As we transition into practical implementation, the project aims to build an AI-based disease prediction model using patient data.

- **Key Technical Objectives:**

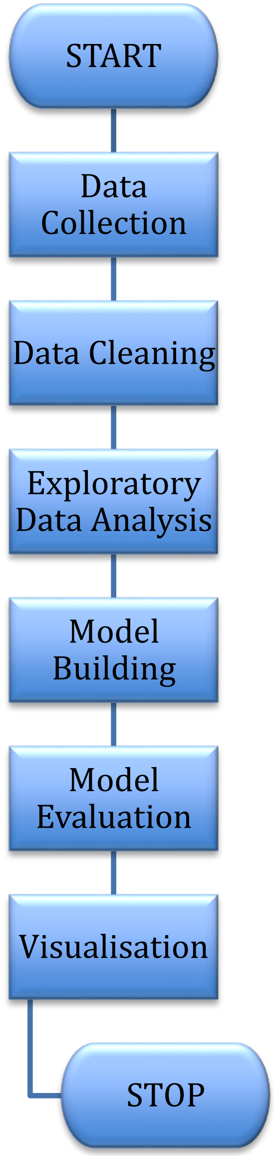
i. Preprocess and clean the dataset for optimal model performance.
ii. Train and evaluate classification algorithms to predict disease presence or risk.
iii. Optimize model performance using techniques like feature selection, hyperparameter tuning, and cross-validation.

- **Model Goals:**

i. Achieve high **accuracy** and **precision** in disease prediction.
ii. Ensure **interpretability** so healthcare professionals can trust and understand predictions.
iii. Maintain **real-world applicability** by handling imbalanced data and unseen patient cases effectively

- **Evolved Understanding:**
  After exploring the data, the focus has shifted slightly from general prediction to emphasizing early detection and risk classification, as this has higher clinical relevance and impact.

3. Flowchart of the Project Workflow



4. Data Description

- Dataset name : Healthcare Dataset
- Dataset source : Kaggle (dataset)
- Type of data : Structured
- Number of records : 769
- Number of features: 09
- Type : Static
- Target variable : Diabetes

5. Data Preprocessing

- Missing values: No missing values were found in the dataset.

  Code: `data.isnull().sum()`

- Duplicate records: No duplicate values is present in the dataset.
  Code: `data.drop_duplicates(inplace=True)`

- Outliers: There is no outliers.

- Data Types: All features are numeric. No conversion is needed.

- Encode categorical variables: Not required as all features are already numerical.
  Code:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
data["Glucose"]=encoder.fit_transform(data["Glucose"])
```

6. Exploratory Data Analysis (EDA)

- **Univariate Analysis**:
  ○ Histogram of Glucose, Age, and BMI to understand distribution of key health indicators
  ○ Boxplots for variables like Glucose, Insulin, and BMI to detect outliers and spread
  ○ Count plot for the Outcome variable to observe class distribution (diabetic vs. non-diabetic)

- **Bivariate & Multivariate Analysis**:
  ○ Correlation matrix shows strong positive correlation between Glucose and

Outcome
○ Scatter plots of Glucose vs Outcome and BMI vs Outcome show higher values linked to diabetes
○ Grouped bar charts reveal increased diabetes prevalence with higher age and BMI categories

- **Key Insights**:
○ Glucose level is the strongest indicator of diabetes
○ Higher BMI and age are associated with increased diabetes risk
○ Dataset contains outliers in Glucose, Insulin, and BMI that may affect model performance

7. Feature Engineering

● **Created binary feature**: is_obese = 1 if BMI ≥ 30, else 0 – based on standard obesity threshold
● **Binned glucose levels** into categories: low, normal, high – to simplify model interpretation
● **Created interaction feature**: glucose_bmi_ratio = Glucose / BMI – captures combined effect on diabetes risk
● **Removed zero-value entries** in features like Insulin and Skin Thickness where 0 is medically implausible
● **Scaled numeric features** using Standard Scaler to normalise ranges for model input

8. Model Building

● **Algorithms Used**:
○ Logistic Regression: for interpretable baseline classification
○ Random Forest Classifier: to capture non-linear relationships and rank feature importance

● **Model Selection Rationale**:
○ Logistic Regression: simple, fast, and well-suited for binary classification (diabetes: yes/no)
○ Random Forest: handles imbalanced data well, resistant to overfitting, and works with non-linear patterns

● **Train-Test Split**:
○ 80% training, 20% testing
○ Used train_test_split with stratify=Outcome to maintain class balance
○ Set random_state for reproducibility

● **Evaluation Metrics** (Classification):
○ **Accuracy**: Overall correctness of predictions
○ **Precision**: Focus on correct positive predictions (important to avoid false positives)
○ **Recall**: Critical for identifying actual diabetic cases (minimize false negatives)
○ **F1-score**: Balanced metric for imbalanced data

9. Visualization of Results & Model Insights

● **Feature Importance**:
○ Visualized using bar plot from Random Forest Classifier
○ Glucose ranked highest in importance, followed by BMI, Age, and Insulin

● **Model Comparison**:
○ Plotted Accuracy, Precision, Recall, and F1-score for both models
○ Random Forest outperformed Logistic Regression across all metrics, especially Recall

● **Confusion Matrix & ROC Curve**:
○ Confusion matrix showed fewer false negatives with Random Forest (important for medical diagnosis)
○ ROC curves plotted to compare model AUC – Random Forest had a higher AUC, indicating better classification ability

● **Model Explainability**:
○ Used feature importance to interpret key health factors influencing diabetes prediction
○ Glucose and BMI were the most impactful features, aligning with medical understanding

10. Tools and Technologies Used

- **Programming Language:** Python
- **IDE/Notebook:** Jupyter

- **Libraries:** pandas, numpy, seaborn, matplotlib

11. Team Members and Contributions

S.No	NAME	ROLE
1	Agnes Selestina S	Documentation and Reporting
2	Christina Ryka S	Model Development
3	Jeevikasri R	Exploratory Data Analysis (EDA), Feature Engineering
4	Keerthana R	Data Cleaning