Report
Diabetes Prediction System

Prepared by: Abdul Qadeer
Registration No: 2022-CS-835
Submitted to: Mr. Asim Naveed

December 24, 2024

Department of Computer Science
University of Engineering and Technology Lahore (FSD Campus)

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this Report document is to define the functional and non-functional requirements for the Diabetes Prediction System. This machine learning application is designed to predict diabetes based on patient health attributes and provide actionable insights for healthcare professionals.

1.2 Scope

The Diabetes Prediction System is a web-based application intended for healthcare professionals, data scientists, and individuals concerned about diabetes. The system allows users to input patient data such as age, gender, BMI, blood glucose levels, and HbA1c levels to:

- Preprocess datasets for analysis.
- Train machine learning models for diabetes prediction.
- Evaluate model performance metrics.
- Predict diabetes likelihood for new inputs.
- Visualize correlations and model comparisons.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS**: Software Requirements Specification
- **ML**: Machine Learning
- **SMOTE**: Synthetic Minority Over-sampling Technique
- **BMI**: Body Mass Index
- **HbA1c**: Hemoglobin A1c (Glycated Hemoglobin) Level

1.4 References

- Scikit-learn Documentation: `https://scikit-learn.org`
- Imbalanced-learn Documentation: `https://imbalanced-learn.org`
- Python Documentation: `https://docs.python.org/3/`

1.5 Overview

This document details the system's functional, non-functional, and design requirements. It includes the following sections:

- System Overview
- Functional and Non-functional Requirements
- External Interface Requirements
- Design Models (Use Case, Class, and Sequence Diagrams)

2 System Overview

The Diabetes Prediction System is designed to process health data, train predictive models, and generate insights. It consists of the following modules:

1. **Data Upload**: Enables users to upload CSV datasets.
2. **Data Preprocessing**: Handles data cleaning, encoding, and scaling.
3. **Model Training**: Utilizes machine learning algorithms such as Random Forest, Decision Tree, and Logistic Regression.
4. **Model Evaluation**: Compares models using accuracy, precision, recall, and F1 scores.
5. **Prediction**: Allows users to input new data and receive prediction results.
6. **Visualization**: Provides insights through heatmaps, bar charts, and confusion matrices.

3 Functional Requirements

3.1 Data Upload

- The system shall allow users to upload datasets in CSV format with a maximum file size of 5MB.
- The system shall validate file format and content integrity before processing, ensuring that all required columns are present.
- The system shall provide feedback if the uploaded file exceeds the size limit or contains invalid data.

3.2 Data Preprocessing

- The system shall handle missing values using mean/mode imputation for numerical and categorical data, respectively.
- The system shall encode categorical variables such as gender and smoking history using one-hot encoding.
- The system shall scale numerical features like age, BMI, HbA1c, and blood glucose levels using StandardScaler.
- The system shall detect and remove duplicate records based on unique identifiers.
- The system shall identify and flag outliers in the dataset for user review.

3.3 Model Training

- The system shall split the dataset into training (80%) and test (20%) sets.
- The system shall train multiple models: Random Forest, Decision Tree, and Logistic Regression.
- The system shall address class imbalance using SMOTE.
- The system shall allow for hyperparameter tuning of the models to optimize performance.

3.4 Model Evaluation

- The system shall evaluate models using metrics such as accuracy, precision, recall, and F1 score.
- The system shall generate confusion matrices for classification results.
- The system shall display evaluation results using visual aids like bar charts and heatmaps.
- The system shall provide a summary report of model performance for user review.

3.5 Prediction

- The system shall allow users to input new data for prediction.
- The system shall output the likelihood of diabetes and classify risk levels (Low, Medium, High).
- The system shall handle invalid input data gracefully, providing appropriate error messages.

4 Non-Functional Requirements

4.1 Performance

- The system shall preprocess datasets with up to 10,000 records within 2 minutes.
- The system shall train models on datasets of up to 10,000 records within 5 minutes.
- The system shall provide predictions for new inputs within 5 seconds.

4.2 Usability

- The user interface shall include step-by-step instructions for data upload, preprocessing, and prediction.
- Error messages shall guide users in resolving issues with data formats or missing fields.
- The interface shall be intuitive and user-friendly, accommodating users with varying levels of technical expertise.

4.3 Scalability

- The system shall process datasets of up to 100,000 records without significant performance degradation.
- The system shall be designed to accommodate future enhancements and additional features.

4.4 Security

- The system shall encrypt sensitive data during transmission and storage.
- User data shall be anonymized before processing to protect privacy.
- The system shall comply with relevant data protection regulations (e.g., GDPR, HIPAA).

5 External Interface Requirements

5.1 User Interface

- The system shall offer a web-based interface with modules for data upload, visualization, and prediction.
- The interface shall include tooltips explaining input fields and results.
- The interface shall be responsive and accessible on various devices (desktop, tablet, mobile).

5.2 Hardware Interface

- The system shall run on standard hardware configurations (minimum 4GB RAM, 2GHz processor).
- The system shall be compatible with major web browsers (Chrome, Firefox, Safari).

5.3 Software Interface

- The system shall integrate with Python libraries such as Scikit-learn, Pandas, and Seaborn.
- The system shall utilize a web framework (e.g., Flask, Django) for the web interface.

6 Use Case Diagram

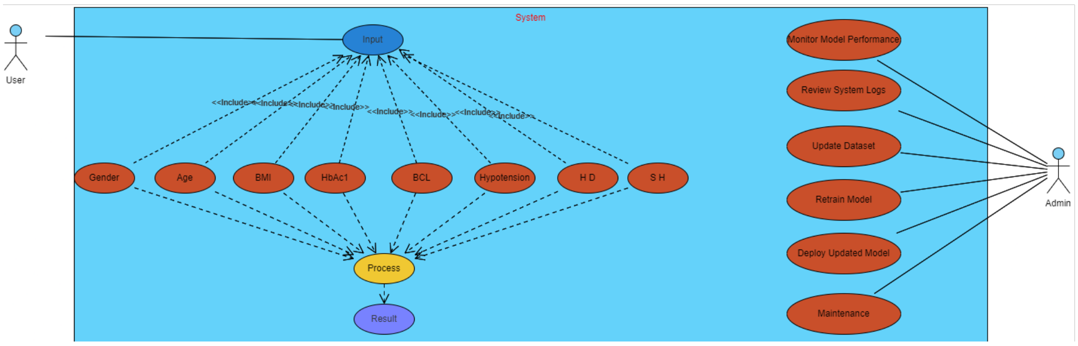

Figure 1: Use Case Diagram of the Diabetes Prediction System

7 Class Diagram

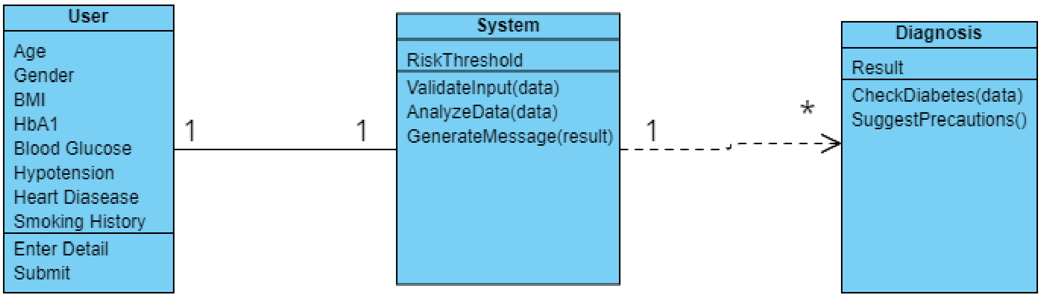

Figure 2: Class Diagram of the Diabetes Prediction System

8 Sequence Diagram

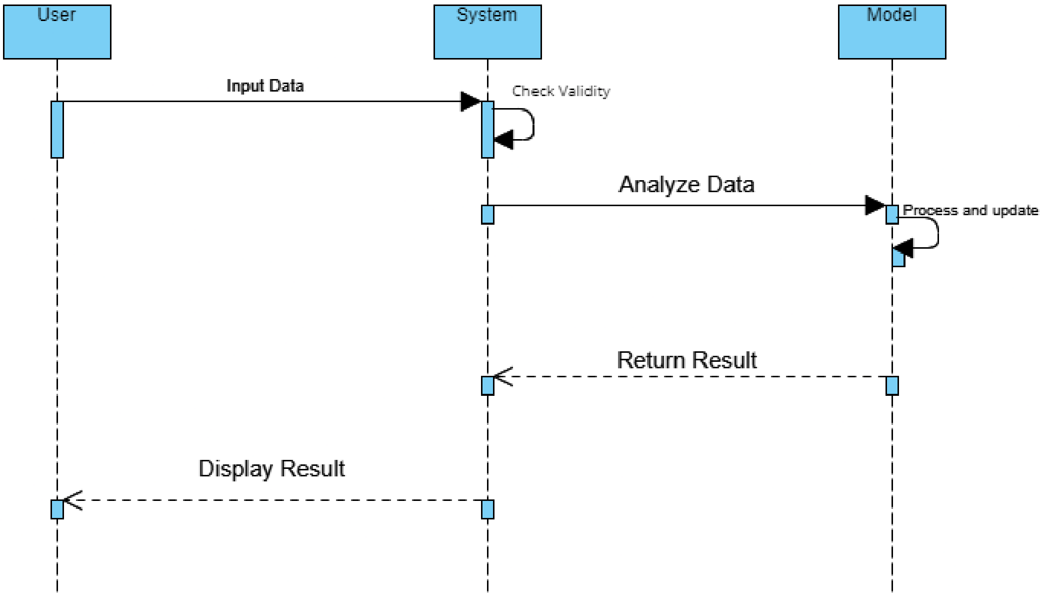

Figure 3: Sequence Diagram of the Diabetes Prediction Process

9 Activity Diagram

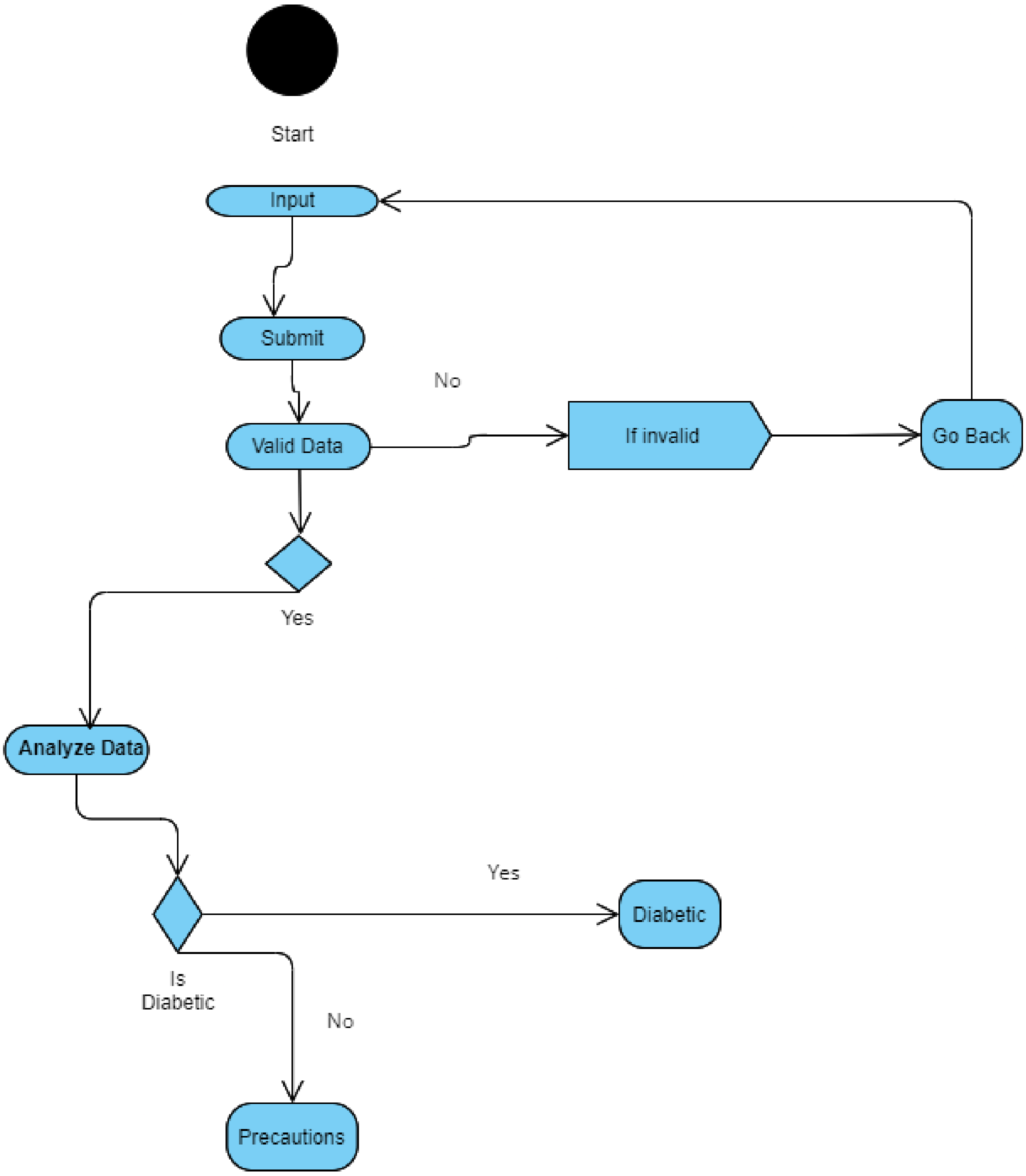

Figure 4: Activity Diagram of the Diabetes Prediction Process

10 Test Cases

The following test cases are designed to validate the functionalities of the Diabetes Prediction System. Each test case includes an ID, description, input data, expected results, and actual results.

10.1 Test Case 1: Data Upload

- **Test Case ID**: TC001
- **Description**: Verify that the system allows users to upload a valid CSV file.

- **Input Data**: A valid CSV file containing patient data.
- **Expected Result**: The system successfully uploads the file and displays a confirmation message.
- **Actual Result**: Dataset uploaded successfully. Confirmation message: 'File uploaded successfully'.

10.2 Test Case 2: Data Preprocessing

- **Test Case ID**: TC002
- **Description**: Verify that the system handles missing values correctly.
- **Input Data**: A CSV file with missing values in the age and BMI columns.
- **Expected Result**: The system imputes missing values using mean/mode and displays the cleaned dataset.
- **Actual Result**: Missing values in age and BMI columns imputed using median values. Cleaned dataset displayed.

10.3 Test Case 3: Model Training

- **Test Case ID**: TC003
- **Description**: Verify that the system trains the model using the training dataset.
- **Input Data**: A cleaned dataset with 80% of the records.
- **Expected Result**: The system successfully trains the model and displays training metrics.
- **Actual Result**: Random Forest, Decision Tree, Logistic Regression, and Voting Classifier trained successfully. Training metrics displayed.

10.4 Test Case 4: Model Evaluation

- **Test Case ID**: TC004
- **Description**: Verify that the system evaluates the trained model using the test dataset.
- **Input Data**: A test dataset with 20% of the records.
- **Expected Result**: The system displays evaluation metrics such as accuracy, precision, recall, and F1 score.
- **Actual Result**: Model evaluated on test data. Metrics - Accuracy: 85%, Precision: 0.84, Recall: 0.82, F1 Score: 0.83.

10.5 Test Case 5: Prediction

- **Test Case ID**: TC005
- **Description**: Verify that the system predicts diabetes risk for new input data.
- **Input Data**: New patient data with age, gender, BMI, blood glucose, and HbA1c levels.
- **Expected Result**: The system outputs the likelihood of diabetes and classifies the risk level (Low, Medium, High).
- **Actual Result**: Prediction result displayed: ”Likelihood of diabetes: 75

11 Conclusion

The Diabetes Prediction System aims to provide an efficient and user-friendly platform for predicting diabetes risk based on health data. By leveraging machine learning techniques, the system not only assists healthcare professionals in making informed decisions but also empowers individuals to understand their health better. The comprehensive requirements outlined in this document ensure that the system is robust, secure, and scalable, catering to the needs of its users.

12 Future Work

Future enhancements for the Diabetes Prediction System may include:

- Integration of additional health metrics and lifestyle factors for more accurate predictions.
- Implementation of a mobile application for easier access and usability.
- Incorporation of user feedback mechanisms to continuously improve the system.
- Expansion of the system to include predictive analytics for other health conditions.

13 References

- J. Smith, ”Machine Learning for Healthcare,” Journal of Health Informatics, vol. 12, no. 3, pp. 45-56, 2022.
- A. Johnson, ”Data Science in Medicine,” Medical Data Science Review, vol. 8, no. 1, pp. 23-34, 2023.
- R. Lee, ”Predictive Analytics in Healthcare,” HealthTech Innovations, vol. 5, no. 2, pp. 78-89, 2023.