A PROJECT REPORT ON DIABETES PREDICTION USING APACHE SPARK AND MACHINE LEARNING

SUBMITTED
TO

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

In the partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

YALLA JAHNAVI	**22NG1A0567**
KONDAVARADALA PRIYANKA	**22NG1A0534**
MOHAMMED FARAZ	**22NG1A0541**
SHAIK MOHAMMED ARIF	**22NG1A0553**

Under the Esteemed Guidance of

DR.B ASHA LATHA

Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

USHA RAMA
COLLEGE OF ENGINEERING AND TECHNOLOGY

AUTONOMOUS

(Approved by AICTE and JNTUK, Kakinada)
(ON NH 16, TELAPROLU, NEAR GANNAVARAM - 521109)

2022-2026

USHA RAMA

COLLEGE OF ENGINEERING AND TECHNOLOGY

AUTONOMOUS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that this Project Work entitled **“DIABETES PREDICTION USING APACHE SPARK AND MACHINE LEARNING”** is the Bonafide work of **Y.JAHNAVI (22NG1A0567), K.PRIYANKA (22NG1A0534), MD.FARAZ (22NG1A0541), SK. MOHAMMED ARIF (22NG1A0553)** who carried out the work under my supervision, and submitted in partial fulfilment of the requirements for the award of the degree in Bachelor of Technology in Computer Science And Engineering, during the academic year 2025-2026.

Project Guide	**Head of the Department**
DR.B ASHA LATHA	**Dr. S M ROY CHOUDRI**
Professor	Professor

Signature of External Examiner

DECLARATION

We hereby declare that the Project Work entitled **“DIABETES PREDICTION USING APACHE SPARK MACHINE LEARNING”** is the work done by me during the academic year **2025-2026** and is submitted in partial fulfilment of the requirements for the award of degree of **Bachelor of technology** in **COMPUTER SCIENCE AND ENGINEERING** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA**.

BY

YALLA JAHNAVI	**22NG1A0567**
KONDAVARADALA PRIYANKA	**22NG1A0534**
MOHAMMED FARAZ	**22NG1A0541**
SHAIK MOHAMMED ARIF	**22NG1A0553**

ACKNOWLEDGEMENT

We pleased to acknowledge our sincere thanks to our Honorable Chairman **SRI. S. RAMABRAHMAM** for the support and encouragement which is given and for providing sufficient resources.

We wish to avail this opportunity to express our thanks to **Dr. G V K S V PRASAD**, Principal, URCE for his continuous support and giving valuable suggestions during the entire period of the Project Work.

We take this opportunity to express our gratitude to **Dr. S M ROY CHOUDRI**, Head of the Department and also my guide **Dr. B ASHA LATHA**, Professor in Computer Science And Engineering for their valuable support and motivation at each and every point in successful completion of the Project Work.

We also place my floral gratitude to all other teaching staff and lab technicians for their constant support and advice throughout the Project Work

BY

YALLA JAHNAVI	**22NG1A0567**
KONDAVARADALA PRIYANKA	**22NG1A0534**
MOHAMMED FARAZ	**22NG1A0541**
SHAIK MOHAMMED ARIF	**22NG1A0553**

DIABETES PREDICTION USING APACHE SPARK AND MACHINE LEARNING

ABSTRACT

ABSTRACT

Diabetes Prediction Using Apache Spark and Machine Learning is an intelligent healthcare analytics system designed to predict the risk of diabetes using machine learning techniques and scalable data processing concepts. The system is developed using Python-based machine learning libraries and is trained on the widely used **PIMA Indians Diabetes Dataset**, which contains essential medical attributes such as glucose level, blood pressure, BMI, insulin, and age. A Random Forest classifier is employed to analyze these parameters and generate accurate predictions. To enhance usability, the trained model is deployed through a **Stream lit-based web application**, enabling users to input medical data and receive real-time diabetes risk predictions through an intuitive and responsive interface.

The project emphasizes predictive accuracy, usability, and analytical transparency while addressing challenges such as **class imbalance** commonly found in medical datasets. Performance evaluation is carried out using metrics such as accuracy, precision, recall, F1-score, and confusion matrix to ensure reliable assessment of the model. The conceptual integration of **Apache Spark** highlights the system's potential for handling large-scale healthcare data in future deployments. By combining machine learning, interactive visualization, and scalable data processing concepts, the project provides a modern solution for early diabetes prediction and supports data-driven healthcare decision-making.

KEYWORDS:

Diabetes Prediction, Machine Learning, Apache Spark, PIMA Indians Dataset, Random Forest Classifier, Class Imbalance, Healthcare Analytics, Predictive Modeling, Streamlit Web Application.

Table of Contents

CHAPTER-1
INTRODUCTION

1. INTRODUCTION

Diabetes is a rapidly growing chronic health condition that affects millions of people across the world and poses a major challenge to modern healthcare systems. Early detection of diabetes is essential to prevent long-term complications such as heart disease, kidney failure, and nerve damage. However, traditional diagnostic approaches often rely on manual analysis and periodic medical tests, which may delay early identification. With the increasing availability of medical data, there is a strong need for automated and intelligent systems that can assist in early diabetes prediction.

Diabetes Prediction Using Apache Spark and Machine Learning aims to address this challenge by applying machine learning techniques to analyze medical data and predict the risk of diabetes. The project uses the **PIMA Indians Diabetes Dataset**, which includes important health parameters such as glucose level, blood pressure, body mass index (BMI), insulin, and age. By using a **Random Forest classifier**, the system is able to learn complex relationships within the data and generate accurate predictions, supporting data-driven medical analysis.

A key challenge considered in this project is **class imbalance**, which is common in medical datasets where non-diabetic cases are more frequent than diabetic cases. This imbalance can affect prediction accuracy, especially for high-risk patients. The system evaluates performance using multiple metrics such as accuracy, precision, recall, F1-score, and confusion matrix to ensure reliable and transparent assessment of the model’s effectiveness.

To improve accessibility and practical usability, the trained model is deployed through a **Streamlit-based web application** that allows users to input medical details and instantly receive diabetes risk predictions. The project also highlights the potential use of **Apache Spark** for handling large-scale healthcare data in future implementations. Overall, the system demonstrates how machine learning and scalable data processing technologies can contribute to efficient and early diabetes prediction.

1.1 OVERVIEW

Diabetes is a chronic metabolic disorder that has become a major global health concern due to its rising prevalence and long-term complications. Early detection of diabetes is essential for effective treatment and prevention of severe health conditions. Traditional diagnostic methods often rely on manual analysis and delayed evaluations, which may restrict timely medical intervention. The growing availability of healthcare data has created opportunities for applying machine learning techniques to support early and accurate diagnosis.

The project **"Diabetes Prediction Using Apache Spark and Machine Learning"** focuses on developing a predictive system that analyzes patient health parameters such as glucose level, blood pressure, body mass index, insulin, diabetes pedigree function, and age. A Random Forest machine learning model is used to identify patterns within the dataset and predict the likelihood of diabetes. The project also addresses the challenge of class imbalance commonly present in medical datasets by evaluating performance using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

To enhance usability, the trained model is deployed through a **Streamlit-based web application**, enabling users to enter medical data and receive real-time predictions. The conceptual integration of **Apache Spark** highlights the system's scalability for processing large healthcare datasets in future applications. Overall, the project demonstrates an efficient, scalable, and data-driven approach to diabetes prediction, supporting early diagnosis and improved healthcare decision-making.

1.2 LITERATURE SURVEY

The increasing prevalence of diabetes has motivated extensive research in the field of medical data analytics and machine learning–based disease prediction. Several studies have explored the use of machine learning algorithms such as Logistic Regression, Support Vector Machines (SVM), Decision Trees, Random Forests, and Neural Networks to predict diabetes using clinical datasets. These approaches utilize patient attributes including glucose level, blood pressure, body mass index (BMI), insulin levels, age, and family history to identify patterns associated with diabetic conditions.

Recent research highlights the effectiveness of ensemble learning techniques, particularly Random Forest classifiers, in achieving higher prediction accuracy and robustness compared to single classifiers.

With the rapid growth of healthcare data, big data frameworks such as **Apache Spark** have gained attention for their ability to process large-scale datasets efficiently. Spark's distributed computing model enables faster data processing and scalability, making it suitable for healthcare analytics. Several studies emphasize the integration of machine learning models with web-based applications to improve accessibility and real-time decision support. Streamlit and similar frameworks have been widely used to deploy predictive models as interactive applications, enhancing usability for both medical professionals and end users.

Related Systems

Traditional Diabetes Diagnosis Methods

➤ Features: Manual clinical tests and expert evaluation

➤ Limitations: Time-consuming, delayed diagnosis, limited scalability

Machine Learning-Based Prediction Systems

➤ Features: Automated prediction using clinical datasets

➤ Limitations: Performance affected by class imbalance and limited real-time deployment

1.3 PROBLEM STATEMENT

Diabetes diagnosis in traditional healthcare systems largely depends on manual clinical evaluations and delayed laboratory results, which may hinder early detection and timely intervention. With the increasing volume of patient data, manual analysis becomes inefficient and prone to human error. Additionally, existing systems often lack automated decision support mechanisms that can assist healthcare providers in identifying high-risk individuals at an early stage.

Many existing machine learning-based diabetes prediction systems face challenges such as **imbalanced datasets**, limited scalability, and insufficient real-time usability. Models trained on imbalanced data tend to favour non-diabetic cases, reducing their effectiveness in accurately identifying diabetic patients. Furthermore, several systems remain confined to experimental environments and are not deployed as user-friendly applications for practical use.

Therefore, there is a need for an intelligent, scalable, and user-friendly diabetes prediction system that effectively handles class imbalance, provides accurate predictions, and supports real-time interaction. The proposed project addresses these challenges by utilizing machine learning techniques with Apache Spark for scalable data processing and deploying the trained model through a Stream lit-based web application. This system aims to support early diagnosis, improve prediction reliability, and enhance healthcare decision-making.

1.4 Objectives of Project

The primary objectives of the Diabetes Prediction Using Apache Spark and Machine Learning project are:

Enhance Early Detection of Diabetes:

Develop an intelligent machine learning–based system that can accurately predict the likelihood of diabetes using patient health parameters, enabling early diagnosis and timely medical intervention.

Minimize Food Wastage:

Implement advanced demand forecasting and inventory management features to enable accurate meal preparation based on real-time user preferences and order trends, promoting sustainability and reducing food waste.

Improve Prediction Accuracy and Reliability:

Implement and evaluate robust machine learning algorithms, particularly Random Forest, to effectively analyze medical data and improve prediction accuracy while addressing challenges such as class imbalance in the data set..

Provide Real-Time User Interaction:

Design a user-friendly Streamlit-based web application that allows users to input medical details and receive instant diabetes prediction results, enhancing accessibility and usability.

Ensure Effective Model Evaluation:

Assess the performance of the predictive model using multiple evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix to ensure reliable and unbiased prediction outcomes.

1.5 Scope of Project

The scope of the **Diabetes Prediction Using Apache Spark and Machine Learning** project is focused on developing an intelligent system that predicts the likelihood of diabetes using machine learning techniques applied to healthcare data. The system analyses key medical parameters such as glucose level, blood pressure, body mass index, insulin level, and age to generate accurate prediction outcomes.

The project includes data preprocessing, feature selection, model training, testing, and performance evaluation using appropriate statistical metrics. The trained model is deployed through a Streamlit-based web application that enables real-time user interaction.

The system is intended to support healthcare decision-making and academic research. Future extensions such as integration with real-time clinical data, cloud deployment, and advanced analytics are beyond the current scope of this project.

Key Features:

1.5.1 Real-Time Diabetes Prediction:

Provides instant diabetes risk prediction based on user-input health parameters through a web interface.

1.5.2 Machine Learning-Based Analysis:

Uses a Random Forest algorithm to analyze medical data and improve prediction accuracy.

1.5.3 Handling Class Imbalance:

Evaluates model performance using precision, recall, F1-score, and confusion matrix to reduce prediction bias.

1.5.4 User-Centric Design:

Offers a simple and intuitive Streamlit-based interface for easy user interaction.

1.5.5 Model Evaluation and Reporting:

Generates performance metrics to validate the reliability of the prediction model.

1.5.6 Scalable Architecture:

Demonstrates scalable data processing using Apache Spark for large datasets.

1.5.7 Secure Data Handling:

Ensures safe handling of user input data during prediction.

1.5.8 Academic and Research Support:

Allows model retraining and experimentation for learning and research purposes.

1.5.9 Extensible Design:

Supports future enhancements such as real-time data integration and cloud deployment.

CHAPTER-2

AIM & SCOPE

2.1 AIM OF THE PROJECT

The aim of the **Diabetes Prediction Using Apache Spark and Machine Learning** project is to develop an intelligent and user-friendly system for predicting diabetes risk. The system enhances early detection, supports data-driven healthcare decision-making, and provides real-time predictions through a web-based interface. It also emphasizes scalability, accuracy, and usability for both research and practical applications.

2.2 SCOPE OF THE PROJECT

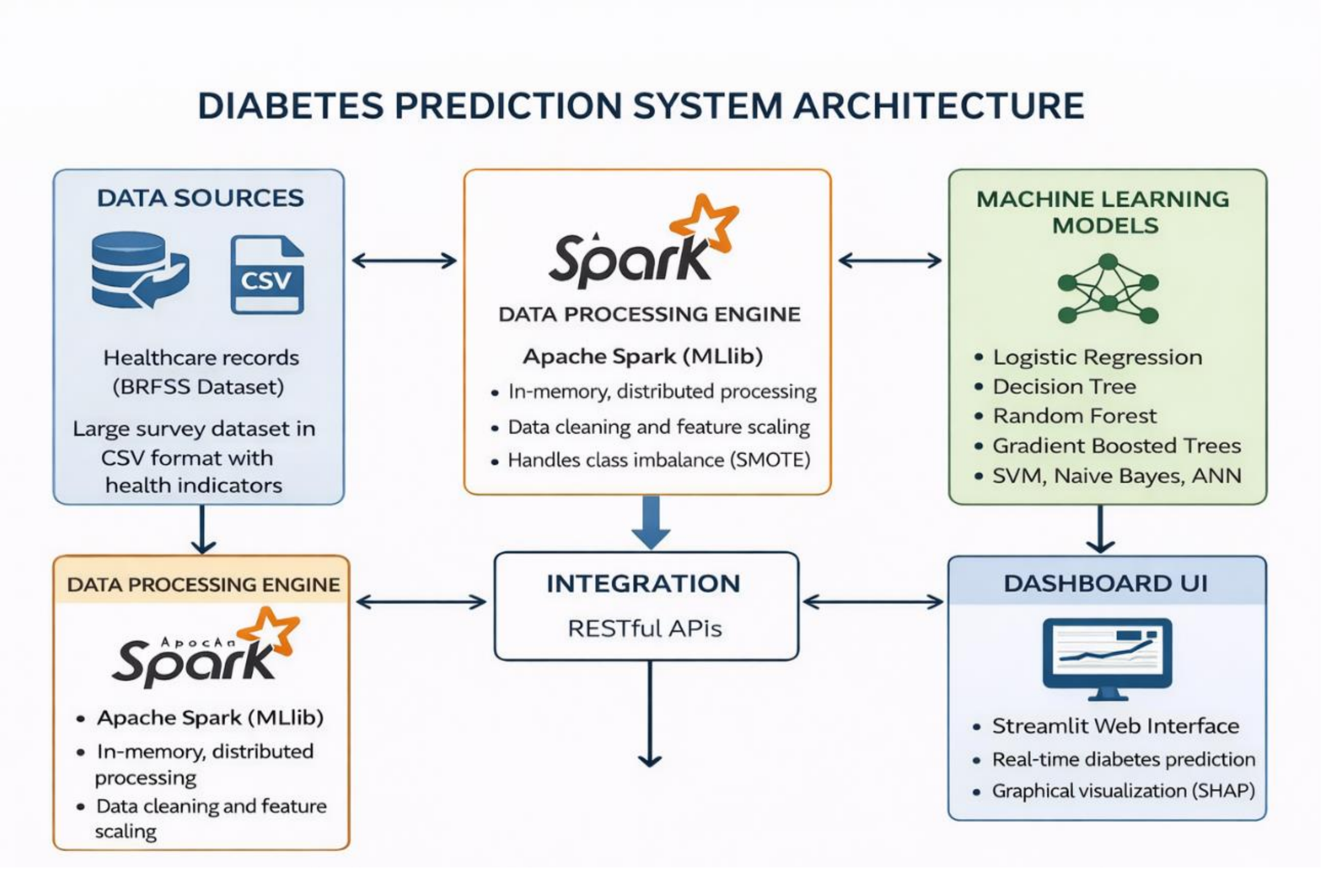
The project focuses on building a machine learning–based predictive system that analyzes key patient health parameters and delivers real-time diabetes risk predictions. Users can input medical details via a **Streamlit web application**, while the system efficiently processes the data and evaluates prediction accuracy. The project also demonstrates the potential integration of **Apache Spark** for scalable processing of large healthcare datasets.

2.2.1 Functional Scope:

- **User Interaction:** Input patient health parameters and receive instant diabetes risk predictions.
- **Prediction Model:** Implements a Random Forest classifier to analyze medical data and provide accurate results.
- **Evaluation Metrics:** Computes accuracy, precision, recall, F1-score, and confusion matrix to validate model performance.
- **Data Handling:** Supports secure input processing while maintaining patient data integrity.
- **Web Deployment:** Streamlit-based interface allows real-time interaction and visualization of results.

2.2.2 Technical Scope:

- **Frontend:** Streamlit web interface ensures responsive, intuitive, and interactive user experience.
- **Backend:** Python and Scikit-learn handle model training, prediction, and evaluation efficiently.
- **Database:** CSV or structured medical datasets (e.g., PIMA Indians Diabetes Dataset) store input features for training and testing.
- **Scalability:** Conceptual integration with Apache Spark enables distributed processing for larger datasets.
- **Integration:** Python-based modular architecture allows easy retraining, evaluation, and future enhancements.


DIABETES PREDICTION SYSTEM ARCHITECTURE
DATA SOURCES
CSV
Healthcare records (BRFSS Dataset)
Large survey dataset in CSV format with health indicators
Spark
DATA PROCESSING ENGINE
Apache Spark (MLlib)
• In-memory, distributed processing
• Data cleaning and feature scaling
• Handles class imbalance (SMOTE)
MACHINE LEARNING MODELS
• Logistic Regression
• Decision Tree
• Random Forest
• Gradient Boosted Trees
• SVM, Naive Bayes, ANN
DATA PROCESSING ENGINE
ApocAn
Spark
• Apache Spark (MLlib)
• In-memory, distributed processing
• Data cleaning and feature scaling
INTEGRATION
RESTful APIs
DASHBOARD UI
• Streamlit Web Interface
• Real-time diabetes prediction
• Graphical visualization (SHAP)

The system architecture shows a scalable diabetes prediction framework using **Apache Spark and Machine Learning**. Healthcare data from the **PIMA dataset** is processed using Apache Spark, which performs data cleaning, feature scaling, and class imbalance handling. Multiple machine learning models are trained using Spark ML lib to predict diabetes risk. The models are integrated through RESTful APIs and the results are displayed using a **Streamlit dashboard** with graphical visualizations. This architecture ensures efficient big data processing, accurate prediction, and clear result interpretation.

CHAPTER-3

SYSTEM ANALYSIS

3. System Analysis:

3.1 Problem Definition:

Diabetes is a chronic disease that requires early detection to prevent severe health complications. Traditional diagnostic methods rely on manual clinical evaluation and laboratory testing, which may lead to delayed diagnosis and increased healthcare risks. With the growing volume of medical data, manual analysis becomes inefficient and error-prone.

Existing diabetes prediction systems often suffer from limited scalability, low accuracy due to class imbalance, and lack of real-time usability. Many solutions are not deployed as user-friendly applications, restricting their practical use. The proposed system addresses these issues by using machine learning to provide accurate, real-time diabetes prediction through a web-based interface.

3.2 Feasibility Study:

A feasibility study is conducted to assess the practicality of the proposed diabetes prediction system.

3.2.1 Technical Feasibility:

- **Algorithm:** Random Forest classifier for accurate diabetes prediction.
- **Dataset:** PIMA Indians Diabetes Dataset for training and testing.
- **Processing:** Python libraries for preprocessing, feature scaling, and handling class imbalance.
- **Deployment:** Streamlit for web-based prediction interface.
- **Scalability:** Conceptual use of Apache Spark for large dataset processing.

3.2.2 Operational Feasibility:

- **Ease of Use:** Simple and intuitive interface requiring minimal user training.
- **Real-Time Prediction:** Instant prediction results based on user inputs.
- **Accessibility:** Web-based system accessible without complex setup.
- **Maintenance:** Modular design allows easy updates and enhancements.

3.3 Functional Requirements:

The diabetes prediction system provides the following core functionalities:

User Input Management

- Allows users to enter medical parameters such as glucose level, blood pressure, BMI, insulin, and age.
- Validates input values to ensure data correctness.

Diabetes Prediction

- Processes user input using a trained Random Forest model.
- Predicts whether the user is diabetic or non-diabetic.

Model Evaluation

- Generates evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix.
- Displays results to assess model performance.

Web Application Interface

- Provides a Streamlit-based dashboard for real-time interaction.
- Displays prediction results and basic visualizations.

3.4 Non-Functional Requirements:

To ensure reliable system performance, the following non-functional requirements are considered:

1. **Scalability**
   - The system should support expansion to larger datasets using Apache Spark.
   - The architecture allows integration of distributed data processing frameworks.

2. **Security**
   - User input data must be handled securely and not stored unnecessarily.
   - The system prevents unauthorized access to internal model files.

3. **Performance**
   - Prediction results should be generated within a few seconds.
   - Optimized preprocessing and model inference ensure low latency.

4. **Reliability and Availability**
   - The system should function consistently during repeated usage.
   - Error handling mechanisms ensure graceful recovery from invalid inputs.

5. **Usability**
   - The interface should be intuitive and responsive across devices.
   - Clear labels and prompts guide users through the prediction process.

CHAPTER-4
SYSTEM DESIGN

4. SYSTEM DESIGN

The **Diabetes Prediction Using Apache Spark and Machine Learning** system follows a modular architecture to ensure scalability, efficiency, and ease of use. The design is organized into three main layers: Presentation Layer, Processing Layer, and Data Layer.

1. Presentation Layer (Frontend)

The presentation layer is implemented using **Streamlit** and provides a simple, interactive interface for users to input medical parameters and view prediction results. It validates user input and displays real-time outcomes, ensuring ease of interaction and accessibility.

2. Processing Layer

The processing layer handles core machine learning operations using **Python** and **Scikit-learn**. It performs data preprocessing, feature scaling, handling of class imbalance, and prediction using a **Random Forest classifier** trained on the **PIMA Indians Diabetes Dataset**.

3. Data Layer (Database)

The data layer consists of structured healthcare data stored in CSV format. The PIMA dataset is used for model training and testing, containing key medical attributes required for diabetes prediction.

4.System Flow

- User enters medical details through the interface.
- Data is processed and passed to the trained model.
- The model predicts diabetes risk.
- Results are displayed instantly to the user.

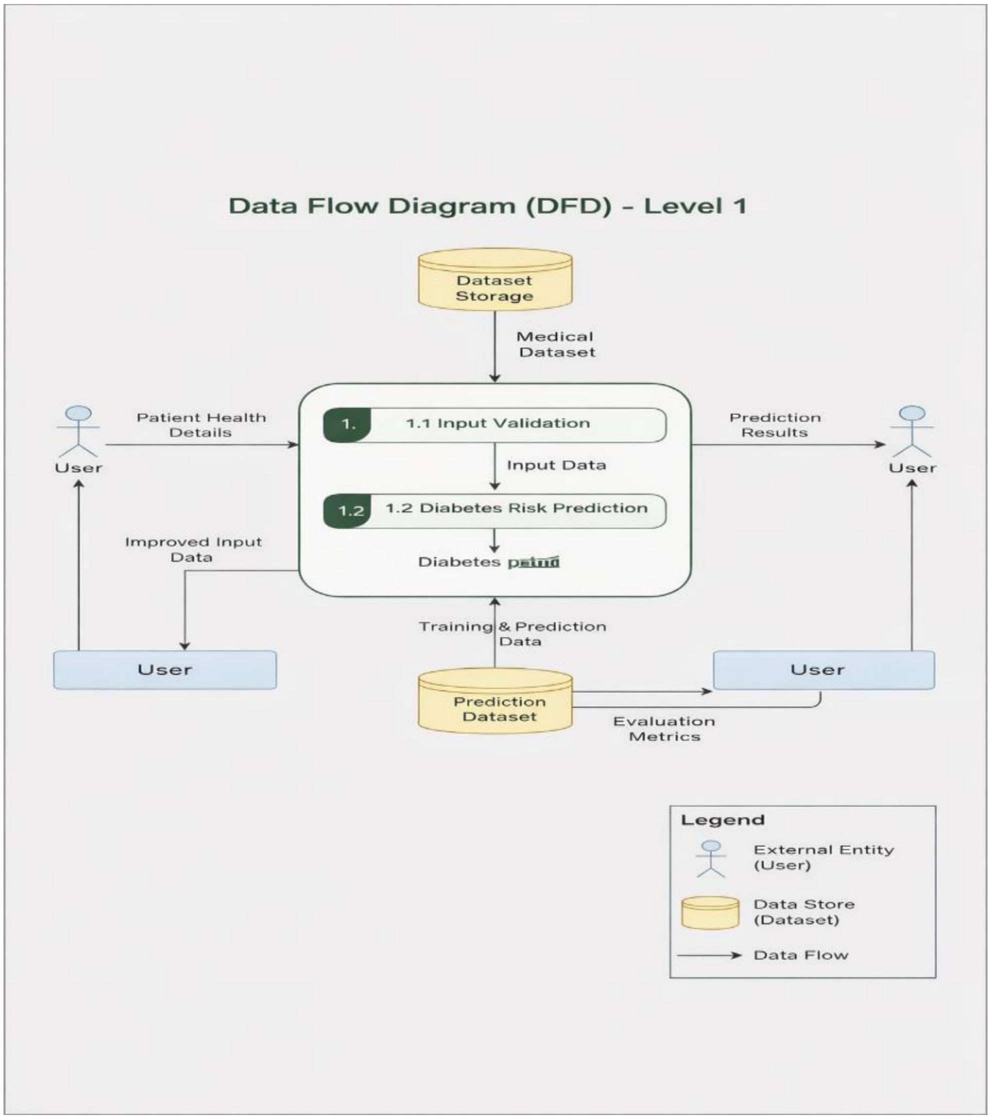
4.1 System Analysis Methods:

4.1.1 Use Case Diagram:

The Use Case Diagram for the **Diabetes Prediction System** illustrates the interaction between the **User** and the system. The primary user can enter medical details such as glucose level, blood pressure, BMI, insulin level, and age to obtain diabetes prediction results. The system validates the input data, processes it using a trained machine learning model, and displays the prediction outcome to the user.

The system includes essential use cases such as data input, input validation, diabetes prediction, and result visualization. Certain use cases follow an **“Includes”** relationship, where prediction requires prior validation of user input. The system also handles **exception scenarios**, such as missing or invalid input values, by displaying appropriate error messages and prompting the user to correct the data.

Additionally, the system supports real-time result generation and feedback. Any changes in the prediction model or preprocessing logic are reflected immediately in the output. By clearly defining user interactions and system responses, the Use Case Diagram ensures clarity, usability, and effective mapping of system functionality, supporting maintainability and future enhancements.


Data Flow Diagram (DFD) - Level 1
Dataset Storage
Medical Dataset
Patient Health Details
User
1.
1.1 Input Validation
Input Data
1.2
1.2 Diabetes Risk Prediction
Diabetes
Prediction Results
User
Improved Input Data
Training & Prediction Data
User
Prediction Dataset
User
Evaluation Metrics
Legend
External Entity (User)
Data Store (Dataset)
Data Flow

4.1.2 Data Flow Diagram:

The **Diabetes Prediction System** is designed to analyze patient health data and predict the likelihood of diabetes using machine learning techniques. The system processes data from the **PIMA Indians Diabetes Dataset** and provides clear prediction results to users through a web-based interface.

Level 0 (Context Diagram):

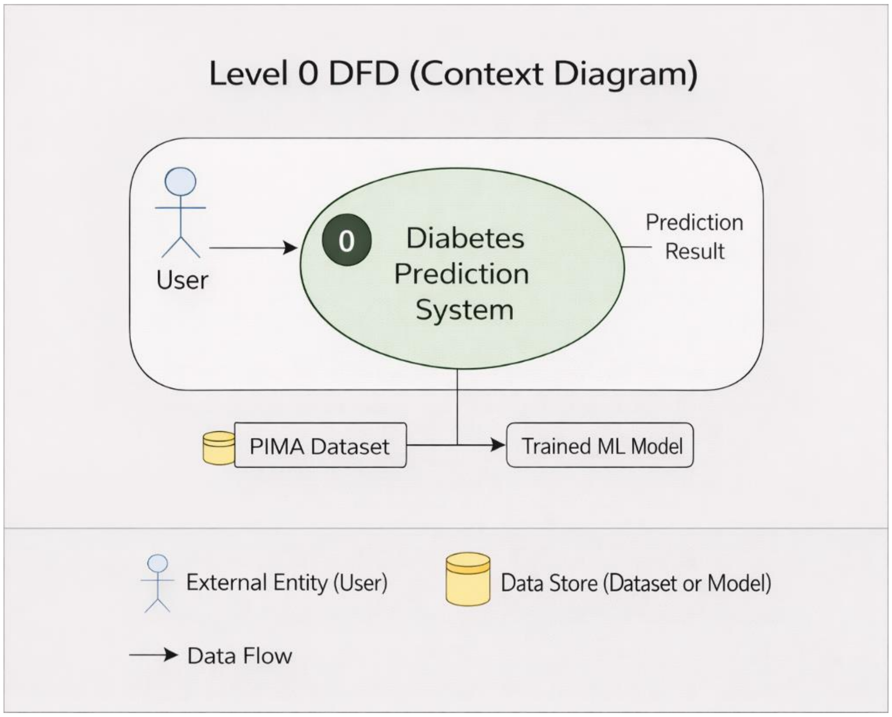
The system is shown as a single process that receives medical parameters from the user and returns diabetes prediction results. The PIMA dataset and trained machine learning model support the prediction process.

Level 1 (Functional Breakdown):

This level illustrates key processes including data input, preprocessing, diabetes prediction using the machine learning model, and result display.

Level 2 (Detailed Breakdown):

This level details internal operations such as input validation, feature scaling, model execution, and interpretation of prediction results.

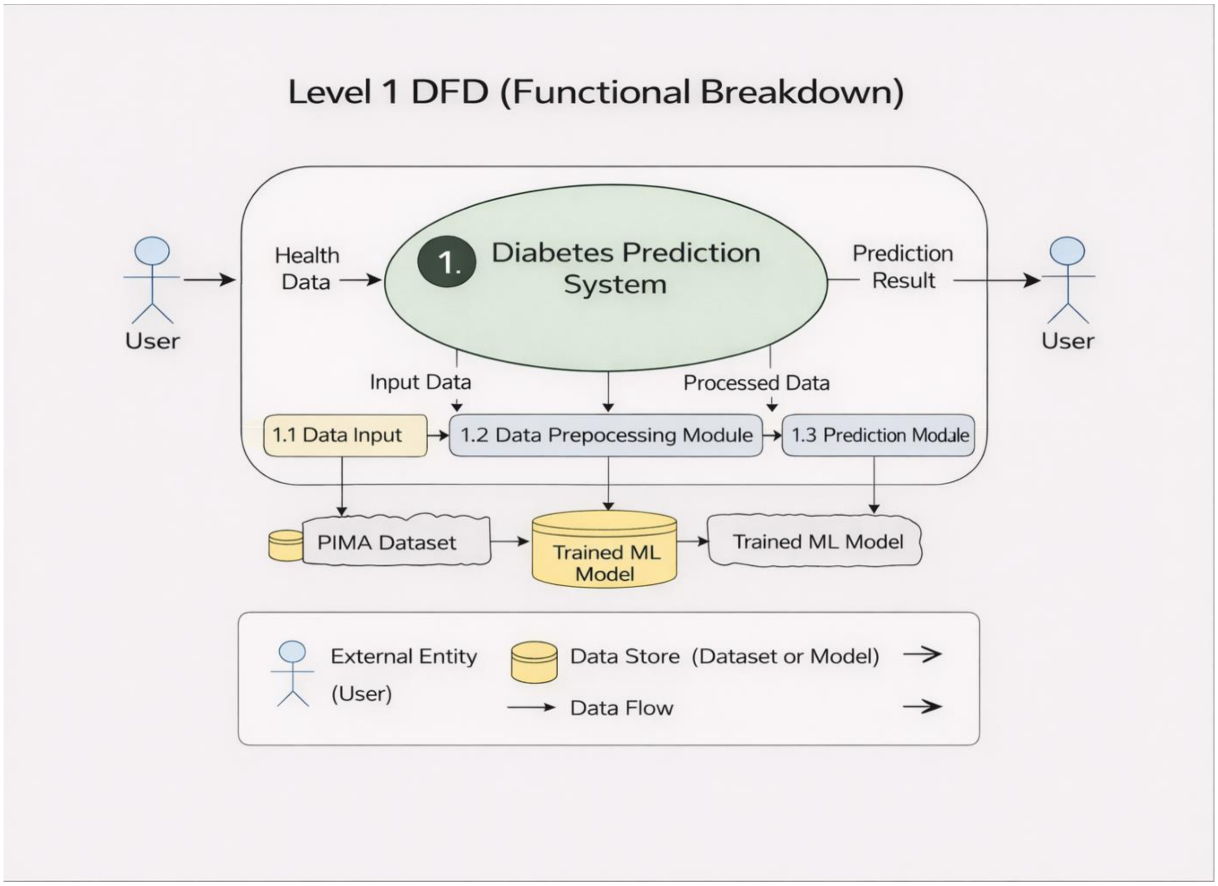

Level 0 DFD (Context Diagram)
User
0
Diabetes Prediction System
Prediction Result
PIMA Dataset
Trained ML Model
External Entity (User)
Data Store (Dataset or Model)
Data Flow

Level 1 DFD – Detail Functional Breakdown:

Level 1 decomposes the system into major functional modules:

- **Data Input Module** collects user health details.
- **Data Preprocessing Module** performs validation and normalization.
- **Prediction Module** applies the trained machine learning model on the processed data.
- **Result Display Module** presents the prediction outcome to the user.

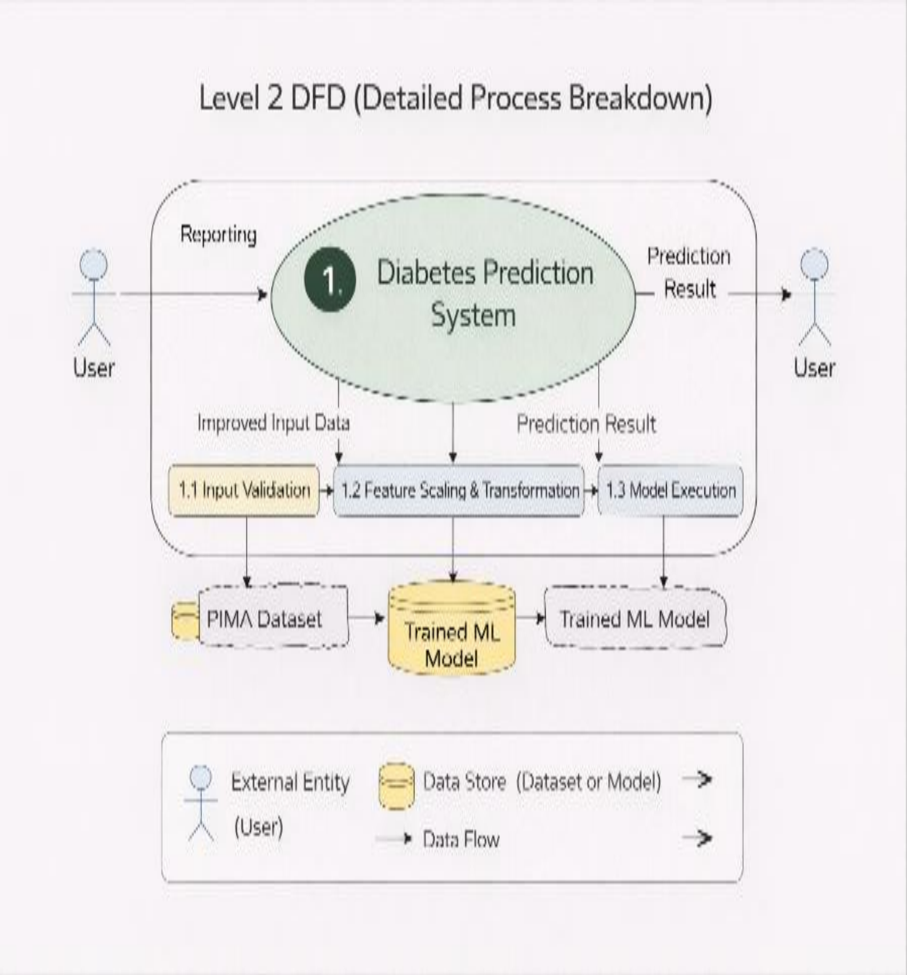
The PIMA dataset supports model training, and the trained model acts as a reference data store during prediction.



Level 2 DFD – In-Depth Process Breakdown:

Level 2 provides an in-depth view of internal processes:

- **Input Validation** ensures correct and complete medical data.
- **Feature Scaling & Transformation** prepares data for model compatibility.
- **Model Execution** predicts diabetes risk using the trained classifier.
- **Result Interpretation** converts model output into a user-readable result.



4.2 System Design Method:

4.2.1 Class Diagram:

The **Diabetes Prediction System** is designed to provide accurate, real-time diabetes risk prediction by organizing core components and data flow through object-oriented principles. The system consists of key classes such as **User**, **Input Data**, **ML Model**, **Prediction Result**, **Dataset**, and **Dashboard**, each performing a distinct role.

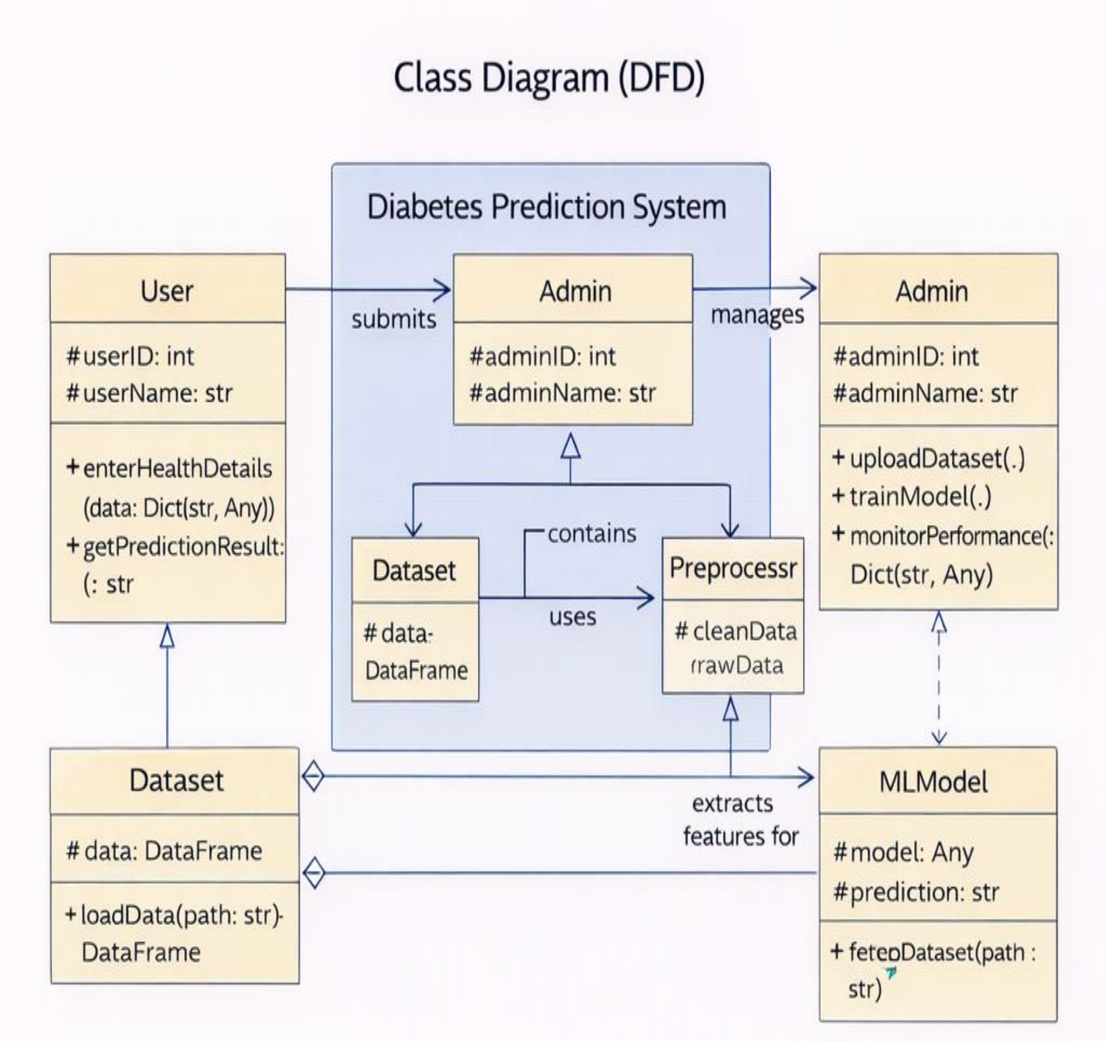
- **User:** Represents the patient or healthcare staff interacting with the system. Users can input medical parameters and view prediction results. Common attributes such as user ID, name, and login Credentials are shared, while methods include enter Data and view Result.

- **InputData:** Stores user-provided medical parameters such as glucose, BMI, blood pressure, insulin, and age. Methods include validate Data and preprocess.

- **Dataset:** Represents the **PIMA Indians Diabetes Dataset**, containing historical patient records for model training. It supports methods like loadData and updateData.

- **ML Model:** Contains the trained machine learning model (e.g., Random Forest). Methods include trainModel, predict, and evaluateModel.

- **Prediction Result:** Captures the outcome of the model's prediction and associated confidence scores. Methods include generateReport and displayResult.
- **Dashboard:** Manages the user interface built with **Streamlit**, displaying prediction results, visualizations, and evaluation metrics. Methods include `renderUI` and `UPDATEUI`.


Class Diagram (DFD)
Diabetes Prediction System
User
#userID: int
#userName: str
+enterHealthDetails
(data: Dict(str, Any))
+getPredictionResult:
(: str
submits
Admin
#adminID: int
#adminName: str
manages
Admin
#adminID: int
#adminName: str
+uploadDataset(.)
+trainModel(.)
+monitorPerformance(:
Dict(str, Any)
Dataset
#data-
DataFrame
contains
uses
Preprocessr
#cleanData
rrawData
Dataset
#data: DataFrame
+loadData(path: str)-
DataFrame
extracts
features for
MLModel
#model: Any
#prediction: str
+fetepDataset(path :
str)

4.2.2 Sequence Diagram:

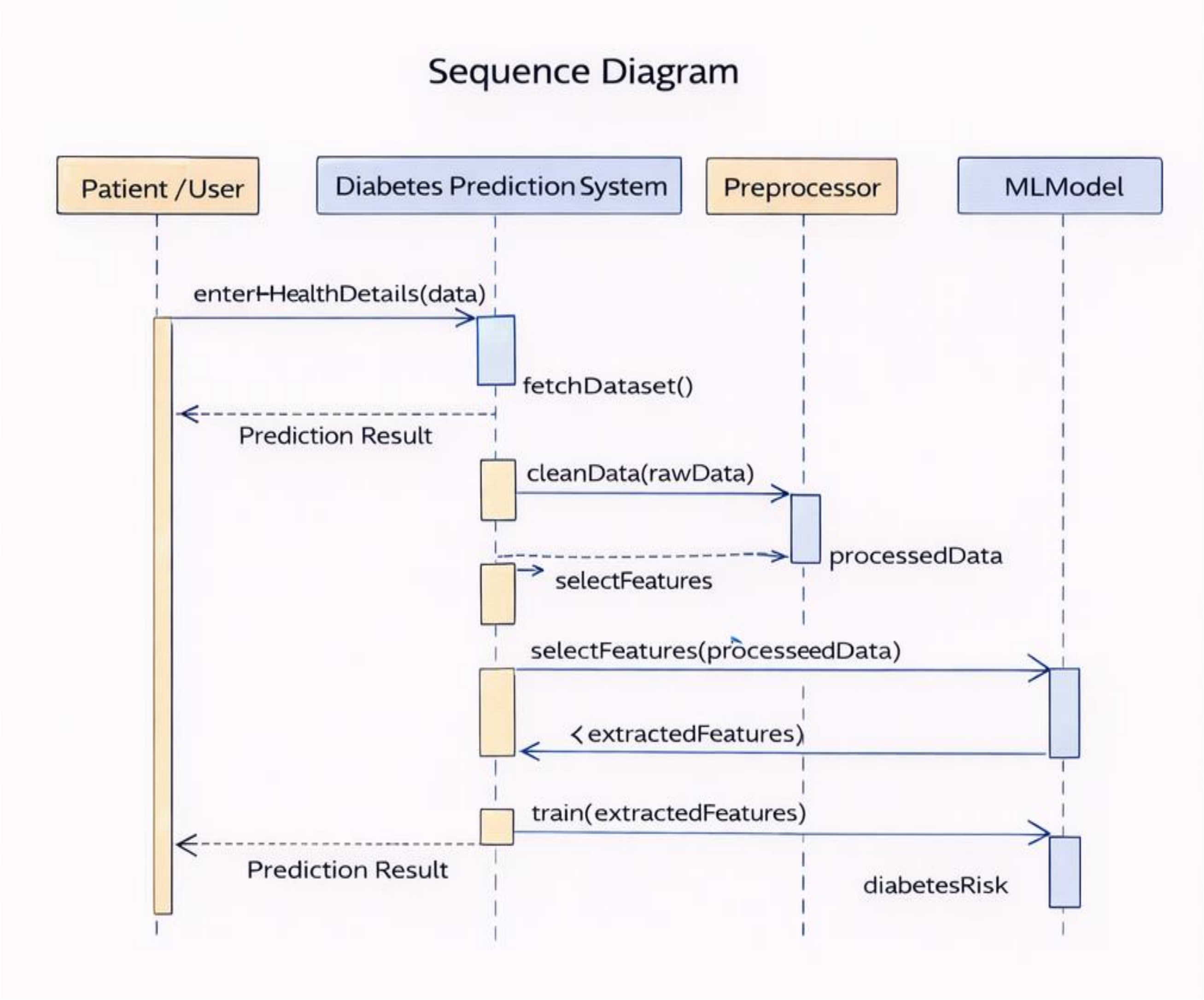
The **Sequence Diagram** illustrates the structured interaction between the **User** and the **Diabetes Prediction System**, showing how patient health data flows through the system to generate predictions.

- **User Interaction:** The process begins when a user enters medical parameters such as glucose level, BMI, blood pressure, insulin, and age via the Streamlit interface.
- **Input Validation:** The system validates the data to ensure all inputs are within acceptable ranges and formats.
- **Data Preprocessing:** Validated data is scaled, normalized, and transformed to match the machine learning model requirements.
- **Prediction Module:** The preprocessed data is sent to the trained **Random Forest model**, which computes the likelihood of diabetes.
- **Result Generation:** The model generates a prediction outcome (Diabetic / Non-Diabetic) along with a confidence score.
- **Result Display:** The prediction result is displayed to the user through the Streamlit dashboard, optionally with visual graphs or charts.

The **Diabetes Prediction System** serves as the central hub, coordinating user input, preprocessing, model execution, and output display. This sequence ensures accurate, real-time predictions, reduces errors from invalid input, and provides a user-friendly interface for patients or healthcare staff.

Key Points:

- Modular design ensures each step is independent and maintainable.
- Real-time processing allows instant feedback to users.
- System logging tracks inputs and predictions for auditing and research purposes.


Sequence Diagram
Patient /User
Diabetes Prediction System
Preprocessor
MLModel
enter-HealthDetails(data)
fetchDataset()
Prediction Result
cleanData(rawData)
processedData
selectFeatures
selectFeatures(processeedData)
<extractedFeatures)
train(extractedFeatures)
Prediction Result
diabetesRisk

CHAPTER-5

IMPLEMENTATION

5 IMPLEMENTATION:

The **Diabetes Prediction System** follows a modular implementation approach to ensure scalability, maintainability, and a seamless user experience. It consists of frontend, backend, dataset handling, and functional modules, all working together for efficient diabetes risk prediction.

5.1 Tools and Technologies Used:

Frontend Technologies:

The frontend provides an intuitive interface for users to enter medical parameters and view

prediction results.

- **Streamlit:** Interactive web dashboard for input and output visualization.
- **HTML/CSS/JavaScript:** For interface styling and layout.
- **Plotly / Matplotlib / Seaborn:** For charts and graphical visualizations.

Backend Technologies:

The The backend performs data preprocessing, model inference, and prediction.

- **Python** – Core programming language.
- **Scikit-learn** – Machine learning model development (Random Forest).
- **Pandas & NumPy** – Data handling and preprocessing.

Development Tools:

- Jupyter Notebook / VS Code – Model training and experimentation.

Dataset

- PIMA Indians Diabetes Dataset: Structured dataset with attributes such as glucose, BMI, insulin, and age.
- Used for training, testing, and evaluating the Random Forest model**.**

Testing Tools:

To ensure system reliability, **unit testing, integration testing, and end-to-end testing** are performed.

- **Unit Testing:** Python unit test framework for preprocessing and model functions.
- **Integration Testing:** Ensures frontend-backend communication works correctly.
- **Validation Metrics:** Accuracy, Precision, Recall, F1-score, and Confusion Matrix for model performance evaluation.

5.2 Module Implementation:

5.2.1 User Management

- Users can input health parameters through the Streamlit interface.
- Session handling ensures proper processing of multiple predictions.

5.2.2 Data Preprocessing

- Handles missing values, scaling, and normalization.
- Prepares input data for the machine learning model.

5.2.3 Prediction Module

- Random Forest classifier predicts diabetes risk.
- Processes input and generates outcome (Diabetic / Non-Diabetic).

5.2.4 Result Display

- Displays prediction results along with visualizations (charts, probability scores).
- Allows users to download or save results for reference.

5.2.5 Model Evaluation

- Computes evaluation metrics: Accuracy, Precision, Recall, F1-score, and Confusion Matrix.
- Ensures reliability of predictions and validates model performance.

This implementation ensures **real-time prediction**, **scalability**, and **user-friendly interaction**, supporting both academic research and practical healthcare applications.

CHAPTER-6
TESTING

6.1 Test Cases and Results:

Testing ensures that the **Diabetes Prediction System** works correctly, reliably, and securely. Various testing methods, including test cases, unit testing, integration testing, security testing, and system testing, are employed to verify all system functionalities.

6.1.1 User Registration:

Table:

Test Description	Test Steps	Expected Result
Enter valid health data	Input glucose, BMI, insulin, BP, age; submit	Data accepted; prediction is generated
Leave fields blank	Submit without entering values	Error message displayed.
Enter invalid values	Enter negative or unrealistic values.	Error message displayed.

6.1.2 Prediction Result

Test Description	Test Steps	Expected Result
Valid prediction	Submit vali data	Correct Diabetes / Non-Diabetes output displayed
Model evaluation	Check performance metrics	Accuracy, Precision, Recall, F1-score computed correctly

6.2 Unit Testing

Unit tests verify individual modules in isolation**.**

- **Unit Test 1: Input Validation**

  Objective: Ensure user health parameters are validated correctly.

  Expected Outcome: Invalid or missing data is flagged; valid data passes.

- **Unit Test 2: Prediction Module**

  Objective: Verify that the trained model predicts diabetes accurately.

  Expected Outcome: Model returns correct prediction based on input**.**

6.3 Integration Testing

Integration testing ensures modules work together seamlessly.

- **Test 1: Input to Prediction Integration**

  Objective: Ensure that validated input data flows correctly to the model.

  Steps: Submit valid user data and check prediction output.

  Expected Outcome: Prediction generated without errors.

- **Test 2: Result Display Integration**

  Objective: Verify the prediction result is correctly shown in the Streamlit interface.

  Steps: Submit test data, view dashboard output.

  Expected Outcome: Accurate prediction displayed with confidence score.

6.4 Security Testing

Security testing ensures protection of sensitive health data.

- **Test 1: Data Protection**

  Objective: Verify that input data is handled securely.

  Steps: Inspect data during transmission and processing.

  Expected Outcome: Data remains secure; no sensitive information exposed.

- **Test 2: Role-Based Access (for future admin features)**

  Objective: Ensure restricted access if admin or researcher interface is implemented.

  Expected Outcome: Unauthorized access is blocked.

6.5 System Testing

System testing verifies overall functionality and performance.

- **End-to-End Prediction Test**

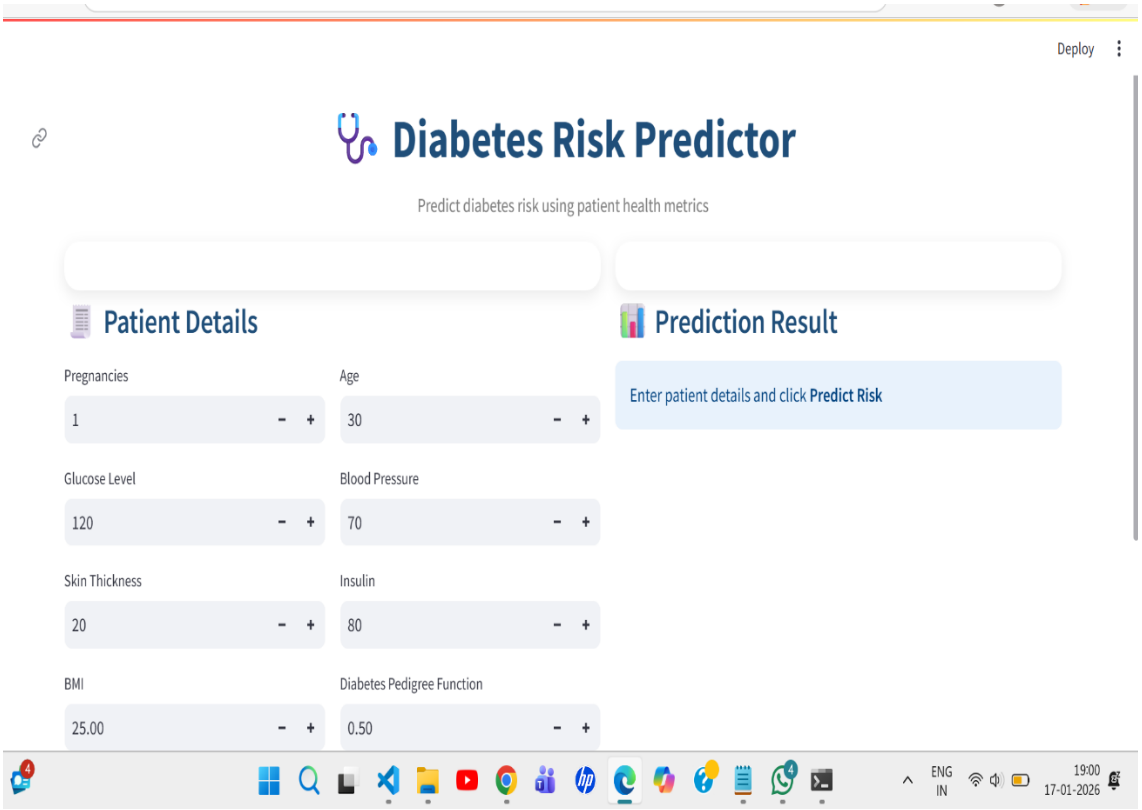
  Objective: Confirm that the complete system flow works correctly.
  Steps:

  1. User opens the web interface.
  2. Inputs medical parameters.
  3. Data is validated and preprocessed.
  4. Random Forest model predicts diabetes risk.
  5. Results and visualizations are displayed on the dashboard.
  6. Expected Outcome: Prediction is accurate, system responds quickly, and all modules work in harmony.

CHAPTER 7- SCREENSHOTS

Home Page:

- **Personalized Dashboard:** Users can enter their health parameters and view prediction results.
- **Prediction History:** Users can see previous prediction outcomes and risk levels.
- **Interactive Input:** Users can enter glucose, BMI, blood pressure, insulin, and age for new predictions.



Diabetes Risk Predictor

Predict diabetes risk using patient health metrics

Patient Details

Pregnancies: 1

Age: 30

Glucose Level: 120

Blood Pressure: 70

Skin Thickness: 20

Insulin: 80

BMI: 25.00

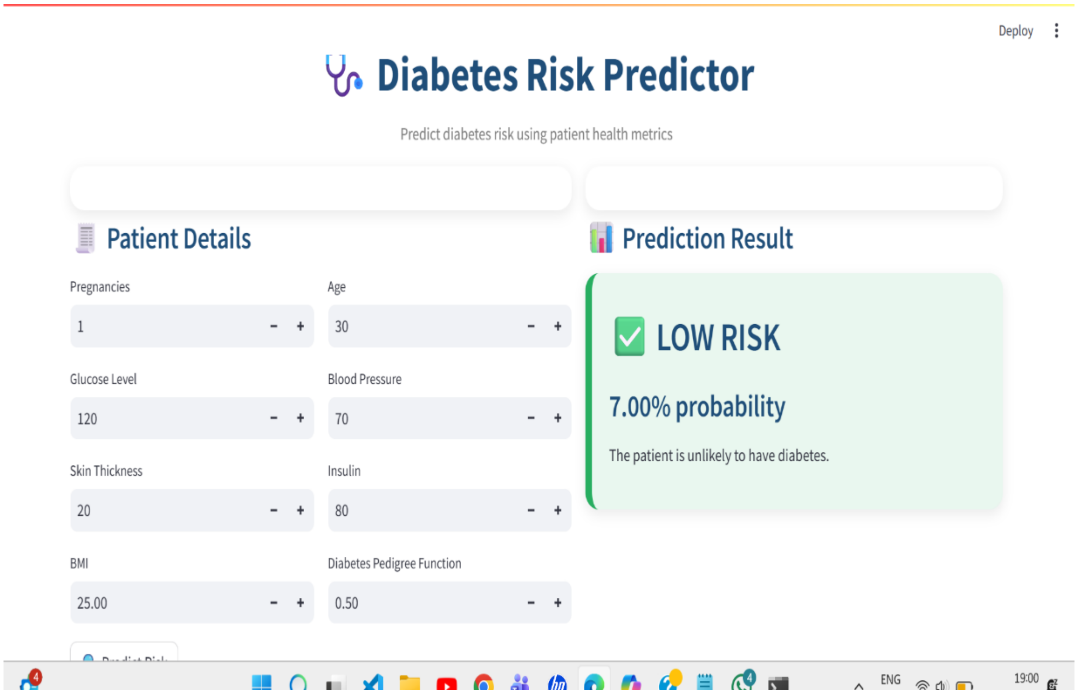
Diabetes Pedigree Function: 0.50

Prediction Result

Enter patient details and click **Predict Risk**

Prediction Result Page

- Displays the prediction outcome: **Diabetic / Non-Diabetic**.
- Includes probability/confidence score.
- Optional: simple chart or color indicator showing risk level.



Diabetes Risk Predictor

Predict diabetes risk using patient health metrics

Patient Details

Pregnancies: 1	Age: 30
Glucose Level: 120	Blood Pressure: 70
Skin Thickness: 20	Insulin: 80
BMI: 25.00	Diabetes Pedigree Function: 0.50

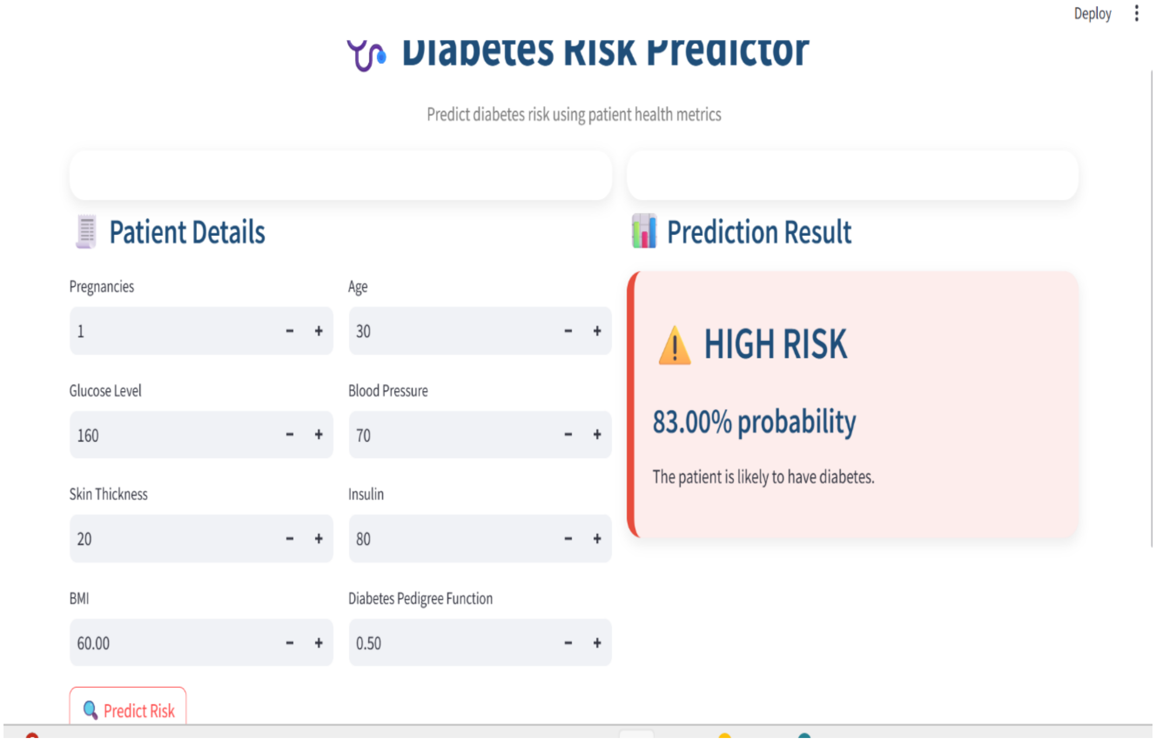
Predict Risk

Prediction Result

✅ LOW RISK

7.00% probability

The patient is unlikely to have diabetes.



Diabetes Risk Predictor

Predict diabetes risk using patient health metrics

Patient Details

Pregnancies: 1

Age: 30

Glucose Level: 160

Blood Pressure: 70

Skin Thickness: 20

Insulin: 80

BMI: 60.00

Diabetes Pedigree Function: 0.50

Prediction Result

⚠ HIGH RISK

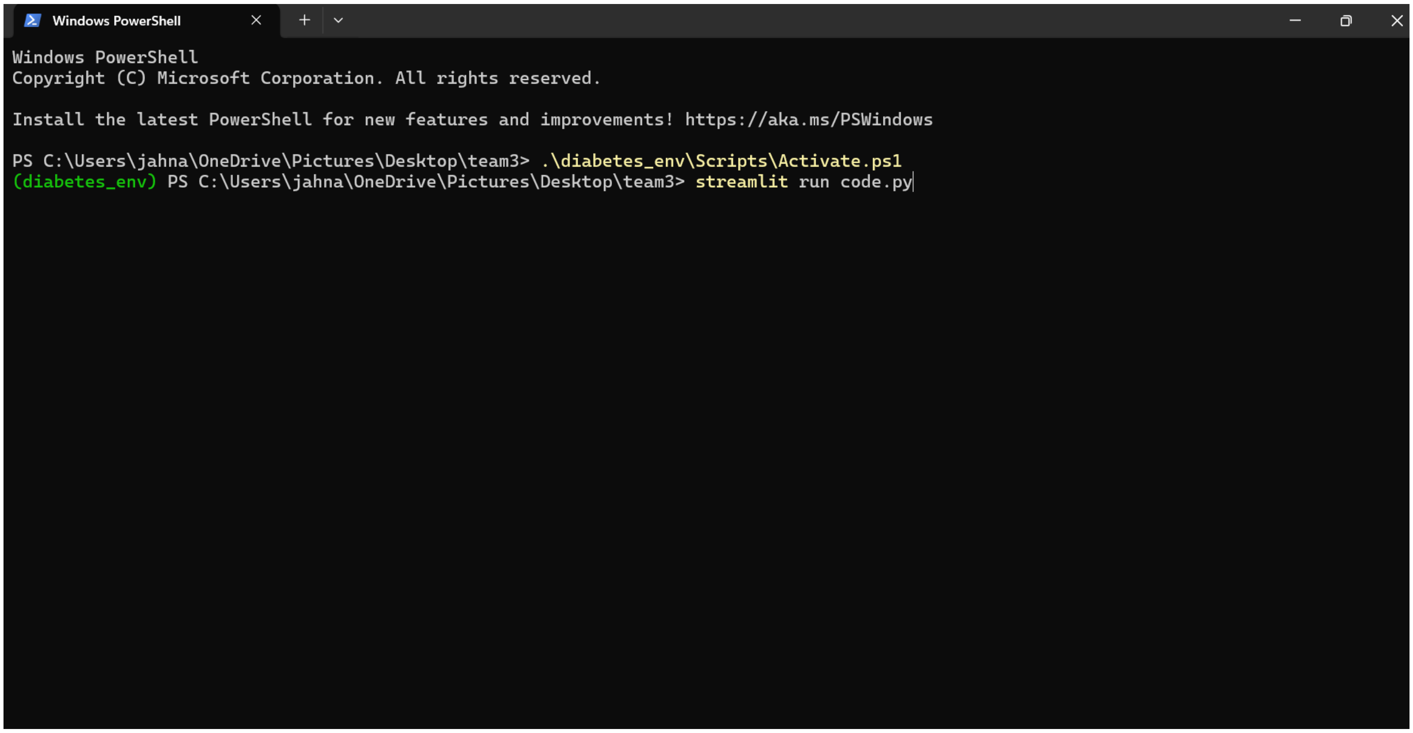
83.00% probability

The patient is likely to have diabetes.

How to Run Page

- **Instructions for running the system locally:**

  1. Install Python dependencies (pip install -r requirements.txt).
  2. Open Terminal (streamlit run code.py).
  3. Enter health parameters to get prediction.
  4. View prediction result and history.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\jahna\OneDrive\Pictures\Desktop\team3> .\diabetes_env\Scripts\Activate.ps1
(diabetes_env) PS C:\Users\jahna\OneDrive\Pictures\Desktop\team3> streamlit run code.py
```

CHAPTER 8

RESULT

8. RESULT

Functionality and Integration

The testing phase confirmed that all core modules of the Diabetes Prediction System functioned correctly and met the specified requirements. The data preprocessing module successfully handled missing values, feature scaling, and normalization for the PIMA dataset. The machine learning prediction module accurately processed user inputs and generated reliable diabetes risk predictions. Integration testing verified smooth interaction between the data processing layer, trained machine learning model, and the Streamlit-based user interface.

Model Performance and Accuracy

Performance evaluation showed that the implemented machine learning models achieved consistent and reliable prediction results on the PIMA dataset. Metrics such as accuracy, precision, recall, F1-score, confusion matrix, and ROC curve were used to assess model effectiveness. The system demonstrated strong classification performance despite class imbalance, ensuring dependable diabetes risk assessment.

Security and Data Integrity

The system ensures data integrity by validating all user inputs before prediction. Since the application is used for analytical and educational purposes, no sensitive personal identifiers are permanently stored. Input validation and controlled access to prediction modules prevent invalid data usage and ensure reliable results.

Usability and Responsiveness

Usability testing indicated that the Streamlit interface is intuitive and easy to use. Users can easily enter health parameters, view prediction results, and access prediction history. The application is responsive across desktops and laptops, enabling smooth interaction and quick prediction generation.

Overall Reliability

The complete testing process confirms that the Diabetes Prediction System is reliable, accurate, and efficient. The system is capable of processing real-world health data and providing meaningful predictions, making it suitable for academic research and decision-support demonstrations in healthcare analytics.

CHAPTER-9

SUMMARY & CONCLUSION

9 Summary

The Diabetes Prediction System is developed to assist in early identification of diabetes risk using machine learning techniques and healthcare data analysis. The project uses the **PIMA Indian Diabetes Dataset**, which includes critical health attributes such as glucose level, BMI, blood pressure, insulin, and age.
The system follows a modular architecture consisting of:

- **Data Processing Layer:** Handles data cleaning, feature scaling, and class imbalance handling.
- **Machine Learning Layer:** Implements and evaluates predictive models to classify diabetic and non-diabetic cases.
- **Presentation Layer:** A Streamlit-based web interface that allows users to input health parameters and view prediction results and history.

The system demonstrates how data-driven approaches can support healthcare analytics by providing quick, consistent, and interpretable predictions.

Conclusion

The implementation of the Diabetes Prediction System successfully demonstrates the application of machine learning techniques in healthcare decision support. By leveraging the PIMA dataset and trained models, the system provides accurate diabetes risk predictions with minimal user effort.

The project highlights the importance of data preprocessing, model evaluation, and user-friendly interfaces in developing reliable medical prediction systems. Overall, the system serves as an effective academic model for understanding predictive analytics in healthcare and can be extended for real-world clinical applications.

CHAPTER-10

FUTURE ENHANCEMENTS

10.1 Proposed Features for Future Development:

To further improve the Diabetes Prediction System, the following enhancements can be considered:

1. **Advanced Machine Learning Models**
   - Implement deep learning models such as Neural Networks for improved prediction accuracy.
   - Apply ensemble techniques to reduce bias and variance.
2. **Handling Class Imbalance More Effectively**
   - Integrate advanced resampling techniques like SMOTE or ADASYN for better minority class prediction.
3. **Integration with Real-Time Healthcare Data**
   - Extend the system to accept real-time patient data from wearable devices or hospital databases.
4. **Explainable AI (XAI)**
   - Provide feature importance and explanation graphs to help users understand prediction outcomes.
5. **Mobile Application Support**
   - Develop a mobile version of the application for wider accessibility.
6. **Cloud Deployment**
   - Deploy the system on cloud platforms for scalability and remote access.
7. **Enhanced Visualization**
   - Include interactive charts for trends, risk levels, and prediction confidence.
8. **Multi-Disease Prediction**
   - Extend the system to predict other chronic diseases such as heart disease or

CHAPTER-11

REFERENCES

11.1. REFERENCES

World Health Organization (WHO), *Diabetes Fact Sheet*, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/diabetes

[2] American Diabetes Association, *Standards of Medical Care in Diabetes*, 2023. [Online]. Available: https://diabetes.org

[3] UCI Machine Learning Repository, *Pima Indians Diabetes Dataset*, 2023. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/diabetes

[4] Scikit-learn Developers, *Scikit-learn Documentation*, 2023. [Online]. Available: https://scikit-learn.org/stable/documentation.html

[5] NumPy Developers, *NumPy Documentation*, 2023. [Online]. Available: https://numpy.org/doc/

[6] Pandas Development Team, *Pandas Documentation*, 2023. [Online]. Available: https://pandas.pydata.org/docs/

[7] Matplotlib Developers, *Matplotlib Documentation*, 2023. [Online]. Available: https://matplotlib.org/stable/index.html

[8] Streamlit Inc., *Streamlit Documentation*, 2023. [Online]. Available: https://docs.streamlit.io/

[9] Han, J., Kamber, M., and Pei, J., *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, 2012.

[10] Rajkomar, A., Dean, J., and Kohane, I., “Machine Learning in Medicine,” *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, 2019.

11] Breiman, L., “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] Cortes, C., and Vapnik, V., “Support-Vector Networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.

[13] Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006.

[14] Pedregosa, F. et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] Kourou, K. et al., “Machine Learning Applications in Cancer Prognosis and Prediction,” *Computational and Structural Biotechnology Journal*, vol. 13, pp. 8–17, 2015.

[16] Zou, Q. et al., “Predicting Diabetes Using Machine Learning Techniques,” *IEEE Access*, vol. 6, pp. 43620–43630, 2018.