PRECISION DIABETES PREDICTION USING ENSEMBLE MACHINE LEARNING MODELS

Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Artificial Intelligence & Data Science

By

RAHUL L	(22UEAD0050)	**(VTU 21369)**
RAVIPRASAD G	(22UEAD0052)	**(VTU 23521)**
ANBARASU S	(22UEAD2001)	**(VTU 26985)**

Under the guidance of
MRS I FARZHANA ,ME.
ASSISTANT PROFESSOR

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

November, 2024

PRECISION DIABETES PREDICTION USING ENSEMBLE MACHINE LEARNING MODELS

Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Artificial Intelligence & Data Science

By

RAHUL L	(22UEAD0050)	**(VTU 21369)**
RAVIPRASAD G	(22UEAD0052)	**(VTU23521)**
ANBARASU S	(22UEAD2001)	**(VTU 26985)**

Under the guidance of
MRS I FARZHANA ,ME.
ASSISTANT PROFESSOR

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

November, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "PRECISION DIABETES PREDICTION USING ENSEMBLE MACHINE LEARNING MODELS" by RAHUL L (22UEAD0050), RAVIPRASAD G (22UEAD0052), ANBARASU S (22UEAD2001) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
MRS.I.FARZHANA
ASSISTANT PROFFESOR
Artificial Intelligence & Data Science
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
November, 2024

Signature of Head of the Department
Dr. P. Santhi
Professor & Head
Artificial Intelligence & Data Science
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
November, 2024

Signature of the Dean
Dr. S. P. Chokkalingam
Professor & Dean
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
November, 2024

DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

RAHUL L

Date: / /

(Signature)

RAVI PRASAD G

Date: / /

(Signature)

ANBARASU S

Date: / /

APPROVAL SHEET

This project report entitled PRECISION DIABETES PREDICTION USING ENSEMBLE MACHINE LEARNING MODELS by RAHUL L (22UEAD0050), RAVI PRASAD G(22UEAD0052),ANBARASU S (22UEAD2001) is approved for the degree of B.Tech in Artificial Intelligence & Data Science.

Examiners **Supervisor**

MRS.FARZHANA I. ME.

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our **Honorable Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (Electrical), B.E. (Mechanical), M.S (Automobile),D.Sc., and Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Vel Tech Rangarajan Dr. Sagunthala R & D Institute of Science and Technology, for her blessings.

We express our sincere thanks to our respected Chairperson and Managing Trustee **Mrs. RANGARAJAN MAHALAKSHMI KISHORE,B.E.,** Vel tech Rangarajan Dr. Sagunthala R & D Institute of Science and Technology for her blessings.

We are very much grateful to our beloved **Vice Chancellor Prof. RAJAT GUPTA,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. S. P. CHOKKALINGAM, M.Tech., Ph.D.,&Associate Dean, Dr. V. DHILIP KUMAR,M.E.,Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Professor & Head, Department of Artificial Intelligence & Data Science, Dr. P. SANTHI, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor MRS I FARZHANA ,M.E.,** for her cordial support, valuable information and guidance, he/she helped us in completing this project through various stages.

A special thanks to our **Project Coordinator Mr. R. DURAI VASANTH, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

RAHUL L **(22UEAD0050)**
RAVI PRASAD G **(22UEAD0052)**
ANBARASU S **(22UEAD2001)**

ABSTRACT

The increasing global incidence of diabetes highlights the need for innovative solutions to enhance early detection and management. This paper presents a webbased application designed to predict the likelihood of diabetes onset using advanced machine learning algorithms. The essential objective is to provide a user-friendly platform for personalized risk assessment, empowering‘ users with accurate and reliable predictions. A comprehensive approach was adopted, analysing a diverse dataset with consideration of key health indicators and lifestyle factors. The machine learning model developed for this purpose exhibits promising accuracy and reliability. The application prioritizes simplicity and accessibility, ensuring ease of use across various demographics. This tool holds significant potential for healthcare providers by offering insights into patient risk profiles and identifying risk factors at an early stage, thereby facilitating timely intervention. The intuitive design ensures effortless data entry, making it accessible to users with varied backgrounds. This project represents a substantial advancement in diabetes management, offering a easy and effective tool for early prediction.

Keywords:

1. Diabetes Prediction
2. Machine Learning Models
3. Ensemble Learning
4. Decision Support System
5. User-Friendly Interface
6. Web-Based Application
7. Healthcare Tool
8. Predictive Modeling
9. Accessibility

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

EDP	Early Diabetes Predictor
ML	Machine Learning
AI	Artificial Intelligence
CSV	Comma-Separated Values
API	Application Programming Interface
FL	Flask (Web Framework)
DT	Decision Tree
SVM	Support Vector Machine
RF	Random Forest
EDA	Exploratory Data Analysis

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Diabetes mellitus has emerged as a significant global health concern, affecting millions of individuals and contributing to a wide array of complications, including cardiovascular disease, and renal failure. The World Health Organization concludes that the number of people living with diabetes will continue to rise, underscoring the huury need for major prevention and intervention goals. Early detection is crucial, as it enables timely lifestyle modifications and medical management that can prevent or delay the onset of the disease.

The primary goal to execute a predictive model for early diabetes detection using machine learning techniques. By analyzing a diverse dataset that includes clinical, biochemical, and demographic variables, seek to identify key risk factors that contribute to diabetes disease. The accuracy gained from this may not only improve early screening protocols but also guide targeted interventions that can significantly reduce the cause of diabetes on individuals and healthcare systems.

1.2 Aim of the project

The primary aim of developing an early diabetes predictor is to create a reliable, accessible, and efficient tool for identifying individuals at risk of developing diabetes in its initial stages. the implementing advanced data analysis, machine learning, and clinical research, the predictor seeks to accurately identify individuals who are prediabetic or at a heightened risk of developing diabetes based on key health indicators and lifestyle factors.

1.3 Project Domain

The primary goal of this project is to develop an innovative early warning system for diabetes, aimed at accurately identifying individuals at high risk before the disease

fully manifests. This predictive tool will use a combination of health metrics such as blood sugar levels, body mass index, lifestyle factors, and genetic predispositions analyzed through advanced machine learning models. By proactively flagging early signs of diabetes, the system will enable healthcare providers and at risk individuals to adopt preventative strategies tailored to their unique health profiles. Additionally, the project aims to create a user-friendly and accessible interface that can integrate with healthcare platforms for widespread adoption. This approach not only seeks to empower individuals in managing their health but also aims to reduce the healthcare burden associated with diabetes by focusing on early intervention and customized care.

1.4 Scope of the Project

The scope of this project centers on designing, developing, and validating an early diabetes prediction model that can be applied in clinical and non-clinical settings. The project will involve collecting and analyzing a broad range of health data such as demographic, physiological, and lifestyle factors to identify key indicators of pre-diabetes and early stage diabetes. Advanced machine learning algorithms will be employed to process this data and create a predictive model that accurately assesses diabetes risk. The scope develops a user-friendly interface, integrating the tool with health management platforms, and conducting clinical trials or pilot tests to validate accuracy and usability. This project will also address scalability to ensure the model's effectiveness across diverse populations, ultimately facilitating a reliable and preventative tool in diabetes care.

The scope of this project centers on designing, developing, and validating an early diabetes prediction model that can be applied in clinical and non-clinical settings. The project will involve collecting and analyzing a broad range of health data and lifestyle factors—to identify key indicators of pre-diabetes and early-stage diabetes. Advanced machine learning algorithms will be employed to process this data and create a predictive model that accurately assesses diabetes risk.

Chapter 2

LITERATURE REVIEW

The global prevalence of diabetes continues to escalate, necessitating innovative methods to facilitate early detection and management. Recent research has extensively focused on developing machine learning models that offer high accuracy in predicting diabetes, aiming to empower healthcare providers with reliable decision support systems. Sheik Abdullah et al. [**1**] proposed a comparative study of machine learning models for healthcare data analysis, demonstrating the potential of precision medicine to enhance prediction accuracy for diabetic risk factors.

Building upon these foundations, Al Reshan et al.[**2**] introduced an Ensemble Deep Learning (EDL) model leveraging multiple diabetes datasets to achieve high accuracy and improve patient care. Their model incorporated various data sources, showing that ensemble methods are robust in clinical environments, particularly for diabetes prediction.

Ahmed et al. [**3**] expanded the machine learning scope by proposing a hybrid model combining Artificial Neural Networks (ANN) and Support Vector Machines (SVM) with fuzzy logic, achieving an accuracy of 94.87%, which surpasses traditional models. This work demonstrated that integrating fuzzy logic can significantly enhance decision-making in early diabetes detection and provides a foundation for combining multiple techniques to improve accuracy in complex healthcare datasets.

Massari et al.[**4**] examined various machine learning algorithms, including Logistic Regression (LR), Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Random Forest, for diabetes prediction and proposed the use of ontology to improve model interpretability. Their study achieved accuracy rates up to 98%, emphasizing that combining ontology with machine learning can offer valuable insights into diabetes risk factors.

Feature selection has also been recognized as critical for enhancing model performance. Zhang and Liu [**5**] proposed an approach focusing on feature selection in diabetes prediction models, particularly within decision trees, and demonstrated that identifying the most relevant variables can significantly improve accuracy.

Patel and Shah [**6**] examined supervised learning techniques applied to diabetes

diagnosis using clinical datasets. They focused on data preprocessing, feature extraction, and model evaluation, providing a comprehensive overview of predictive modeling in the healthcare context.

The integration of decision trees and neural networks in hybrid models for diabetes prediction was also proposed by Wang and Liu,[**7**] who demonstrated improved accuracy through this combined approach. Their research highlighted the adaptability of decision trees when used alongside other algorithms, further enhancing reliability in diabetes prediction.

Research into AI applications in diabetes prediction has extended to clinical risk assessment. Chen and Zhang [**8**] proposed using artificial intelligence for diabetes risk prediction through decision trees and random forests, providing critical insights into early diagnosis.

Furthermore, Gupta and Kaur [**9**] emphasized the effectiveness of decision tree algorithms in early diabetes detection, demonstrating superior performance compared to other methods. Sharma and Mehta [**10**] provided a broader survey of machine learning techniques, identifying decision trees, SVMs, and ensemble methods as key approaches for managing diabetes complications.

Research by Lin and Wong [**11**] focused on early diagnosis of Type 2 diabetes using machine learning, particularly highlighting the effectiveness of data-driven approaches on actual clinical datasets. Sharma and Sinha [**12**] proposed a comparative analysis of machine learning algorithms, underscoring the accuracy and precision achieved by models like SVM and k-nearest neighbors in diabetes prediction.

Diabetic retinopathy, a common complication of diabetes, has also been a focus in machine learning research. Kumar and Ray [**13**] applied decision tree algorithms to predict early stages of diabetic retinopathy, utilizing medical imaging to enhance predictive capabilities. Singh and Verma [**14**] conducted a comparative study on models for predicting Type 1 and Type 2 diabetes, demonstrating the high accuracy associated with decision tree-based methods.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

Existing diabetes prediction systems typically rely on a combination of clinical assessments, standard screening tools, and, more recently, digital health technologies. Traditional methods include blood tests like fasting glucose levels, HbA1c tests, and the Oral Glucose Tolerance Test (OGTT), which help determine if someone is prediabetic or diabetic. In recent years, digital tools have emerged to enhance early detection, often utilizing algorithms based on statistical models or machine learning to assess diabetes risk factors. These systems collect data from medical records, wearables, and self-reported information, analyzing metrics such as blood pressure, cholesterol levels, BMI, age, and family history. Mobile apps and online platforms are also being used to support users in tracking lifestyle changes and health trends, though their predictive accuracy varies based on the quality and amount of data provided.

3.1.1 DISADVANTAGES:

1. **Limited Data Accuracy**: Many existing systems rely on self-reported or limited clinical data, which can lead to inaccuracies. For example, blood glucose levels alone might not fully capture all indicators of prediabetes or early diabetes.

2.**High Dependency on Medical Visits**: Traditional diabetes prediction methods often require frequent medical appointments for blood tests, which may not be convenient or accessible for everyone.

3.**Lack of Comprehensive Risk Factor Analysis**: Some systems overlook important risk factors such as family history, lifestyle factors, or genetic predispositions, potentially underestimating or missing the risk in certain individuals.

3.2 Proposed System

A proposed system for early diabetes prediction could leverage modern advancements in technology and data science to improve accuracy, accessibility, and personalization. Here's a framework for an enhanced predictive system

1.Integration of Wearable and IoT Devices: Continuous Monitoring: Utilize wearable devices (e.g., smartwatches, continuous glucose monitors, fitness trackers) to gather real-time data on key metrics, including glucose levels, physical activity, heart rate, sleep patterns, and stress levels.

2.Advanced Machine Learning Models: AI-Driven Analysis employ machine learning models trained on a large, diverse dataset to analyze risk factors such as age, gender, BMI, family history, lifestyle choices, and geographic location.

3.Enhanced Data Collection and Integration:
Comprehensive Data Points: Include data on blood glucose levels, HbA1c, blood lipids, and blood pressure, along with demographic, lifestyle, and genetic information where available.
Multi-Source Data Integration: Collect and integrate data from multiple sources, including wearables, patient-reported outcomes, and electronic health records (EHRs) to form a holistic health profile.

3.2.1 ADVANTAGES

1. **Improved Accuracy**:
Data Collection: By incorporating real-time data from wearables, EHRs, and lifestyle metrics, the system captures a more complete picture of an individual's health.
Personalized Prediction Models: Machine learning algorithms tailor predictions based on unique individual risk factors, making predictions more precise compared to generalized models.

2. Convenience and Accessibility:
Remote Data Collection: Users can continuously track important health metrics from home, eliminating the need for frequent clinic visits for diabetes screening.
Mobile Access: With a mobile app, users can easily access their health data, track

trends,making the system highly accessible and user-friendly.

3.3 Feasibility Study

The proposed early diabetes prediction system aims to harness advanced technology to enhance the detection and prevention of diabetes. By integrating data from wearable devices, the system will continuously monitor key health metrics such as glucose levels, physical activity, and sleep patterns. Utilizing machine learning algorithms, it will provide personalized risk assessments tailored to individual health profiles, enabling timely interventions.

3.3.1 Economic Feasibility

The economic feasibility of the proposed early diabetes prediction system involves the analysis of the expenses and financial benefits associated with its development and implementation. Initial development costs will encompass software design, the procurement of hardware, and the establishment of a reliable cloud infrastructure for data storage and processing. Ongoing expenses will include maintenance, regular software updates, and customer support services. To finance the project, potential funding sources such as grants, partnerships with healthcare organizations, and venture capital investment should be explored.

3.3.2 Technical Feasibility

The technical feasibility of the proposed early diabetes prediction system involves assessing the technological requirements and capabilities necessary for successful implementation. This system will rely on a combination of wearable devices, a mobile application, and cloud-based infrastructure to collect and analyze health data.

3.3.3 Social Feasibility

The social feasibility of the proposed early diabetes prediction system centers on its potential impact on public health and user acceptance. Understanding the attitudes and perceptions of the target audience is vital for successful adoption; conducting surveys and focus groups can provide insights into their willingness to engage with wearable technology and mobile health applications. To ensure accessibility and inclusivity, the system must cater to diverse demographics, including varying age groups and socioeconomic backgrounds, while maintaining a user-friendly interface.

3.4 System Specification

The system will feature user registration and authentication, allowing secure access via email and password or biometric methods. It will collect data from various wearable devices, such as smartwatches and continuous glucose monitors, and employ machine learning algorithms to analyze this data, providing users with personalized risk assessments and actionable health insights. A user-friendly dashboard will display health metrics, risk levels, and progress tracking, along with alerts for abnormal readings to encourage timely interventions.

3.4.1 Hardware Specification

1.Mobile Devices: The system should be compatible with iOS and Android smartphones, preferably running on the latest operating systems (iOS 14 or later, Android 8 or later). Devices should have sufficient processing power and memory (at least 3 GB of RAM and a multi-core processor) to support the mobile application smoothly.

2.**Network Equipment**:High-speed routers with support for the latest Wi-Fi standards to ensure reliable connectivity between wearable devices and mobile applications. Cellular Network Support: For users relying on mobile data, the system should support 4G LTE and 5G connectivity to enable real-time data transmission and alerts.

3.4.2 Software Specification

1.**Backend Server:** The backend will be developed using a robust framework such as Node.js, Django, or Flask, facilitating the creation of APIs for data processing and communication with the mobile application.

2.**Machine Learning Components:** Use of popular machine learning libraries such as TensorFlow, scikit-learn, or PyTorch to develop algorithms for analyzing user data and generating personalized risk assessments. Implementation of data preprocessing steps to clean, normalize, and prepare health data for analysis, ensuring accurate model predictions.

3.4.3 Standards and Policies

Anaconda Prompt

Anaconda Prompt is a command-line interface that supports machine learning and data science workflows. It provides an environment to work with essential libraries and tools for machine learning, making it a core component for the Early Diabetes Predictor project. Available on Windows, Linux, and macOS, Anaconda Navigator offers multiple IDE options to streamline coding and model development. The project also uses **Jupyter Notebook**, an open-source web application within Anaconda that allows for the creation and sharing of documents containing live code, equations, visualizations, and narrative text. Jupyter Notebook is used extensively in this project for data cleaning, model training, statistical analysis, and visualization, contributing to an efficient workflow for developing, testing, and visualizing diabetes predictions.

Standard Used: ISO/IEC 27001 for data security and management

Data Privacy and Security: To safeguard user data, the Early Diabetes Predictor adheres to strict data privacy and security standards. All medical and personal information entered by users is treated with confidentiality and secured through encryption protocols. No personal information is stored on the system unless explicitly required for future improvements, and user data is anonymized to protect privacy. The project complies with data protection laws and best practices, ensuring that user data is neither shared with third parties nor used beyond the intended scope of diabetes prediction.

Accuracy and Model Validation: Accuracy is a top priority for the Early Diabetes Predictor, and our machine learning model has undergone extensive validation to achieve a high standard of prediction performance. The system applies various machine learning classifiers, ultimately using the Decision Tree model due to its accuracy rate of 98.25%. Model performance is regularly tested with new data to ensure predictions remain reliable. This standard ensures that all risk assessments provided to users are based on validated, trustworthy data.

Chapter 4

METHODOLOGY

4.1 General Architecture

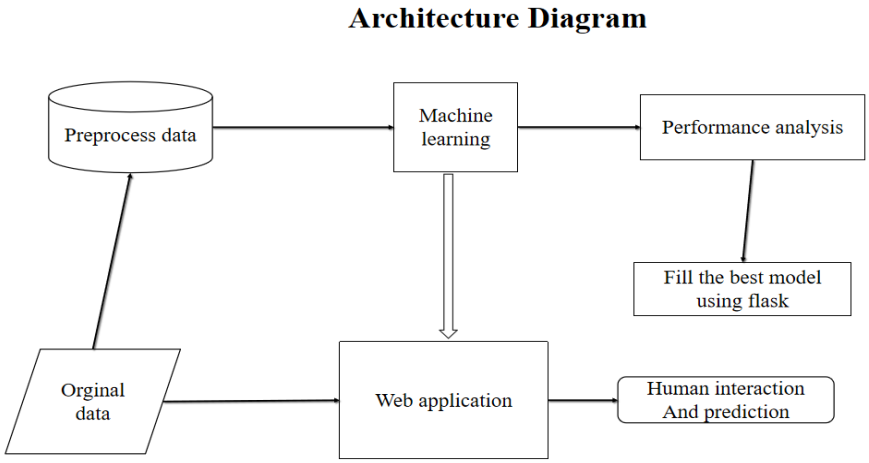

Figure 4.1: **Diabetes Predictor Architecture**

The architecture diagram for an early diabetes predictor illustrates a comprehensive system designed to analyze various health data and predict the likelihood of diabetes onset. At the core of the system lies a data collection module that aggregates information from diverse sources, such as electronic health records, wearable devices, and patient-reported metrics. This data is then processed through a data preprocessing layer, which includes normalization, missing value treatment, and feature extraction, ensuring the information is ready for analysis.

Next, a machine learning module employs various algorithms such as logistic regression, decision trees, or neural networks to identify patterns and correlations within the data that may indicate a predisposition to diabetes.

4.2 Design Phase

4.2.1 Data Flow Diagram

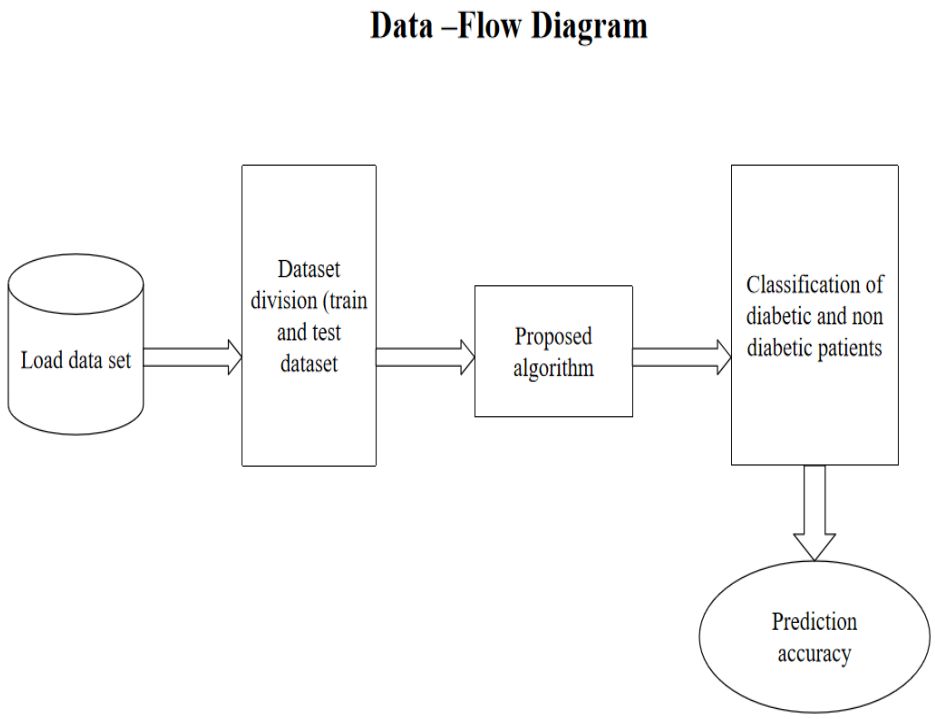

Figure 4.2: **Data Flow**

The data flow diagram for an early diabetes predictor outlines the systematic movement of information through the various components of the system. At the starting point, data is collected from multiple sources, including patient surveys, medical history databases, and health monitoring devices.

Once preprocessed, the data flows into the analysis layer, where machine learning algorithms evaluate the information to identify risk factors and predict diabetes likelihood. The results of this analysis are then sent to a decision-making module, which generates risk assessments and tailored health recommendations for users.

The output from the decision-making module is accessible through a user interface designed for both healthcare providers and patients, enabling them to view insights and suggestions based on the predictions. Feedback mechanisms are, allowing users to provide input that can enhance the system's accuracy over time.

4.2.2 Use Case Diagram

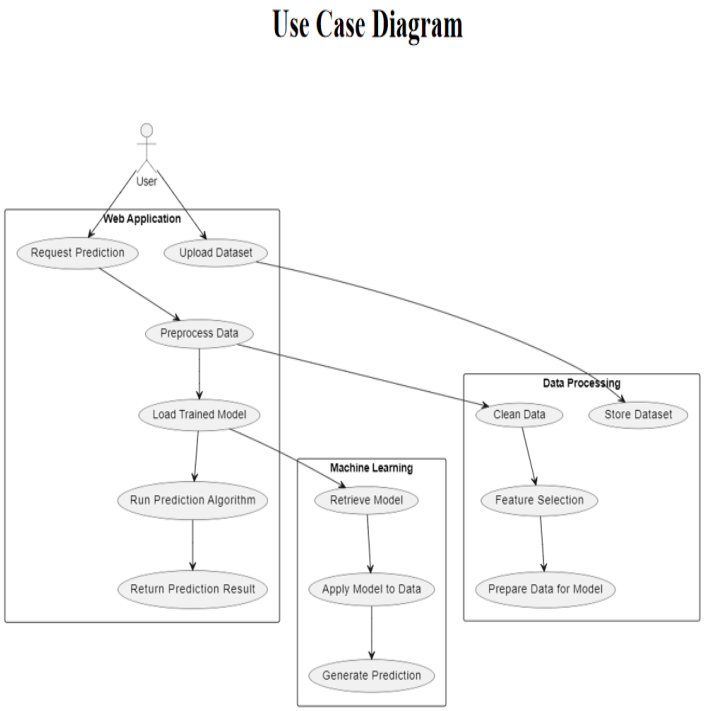

Figure 4.3: **Use Case**

The use case diagram for an early diabetes predictor visually represents the interactions between users and the system, highlighting key functionalities and user roles. Central to the diagram are the primary actors: patients, healthcare providers, and system administrators.

Patients can engage with the system to input personal health information, track lifestyle factors, and receive personalized risk assessments and recommendations. Healthcare providers have the capability to access patient data, analyze trends, and generate reports that inform clinical decisions and patient management strategies.

The system administrator ensures the overall functionality of the application, managing user accounts and overseeing data security. Each interaction is depicted as a use case, showcasing activities such as data entry, risk prediction, feedback submission, and report generation. This diagram encapsulates the collaborative nature of the system, emphasizing how different users benefit from its predictive capabilities to enhance early diabetes intervention and management.

4.2.3 Class Diagram

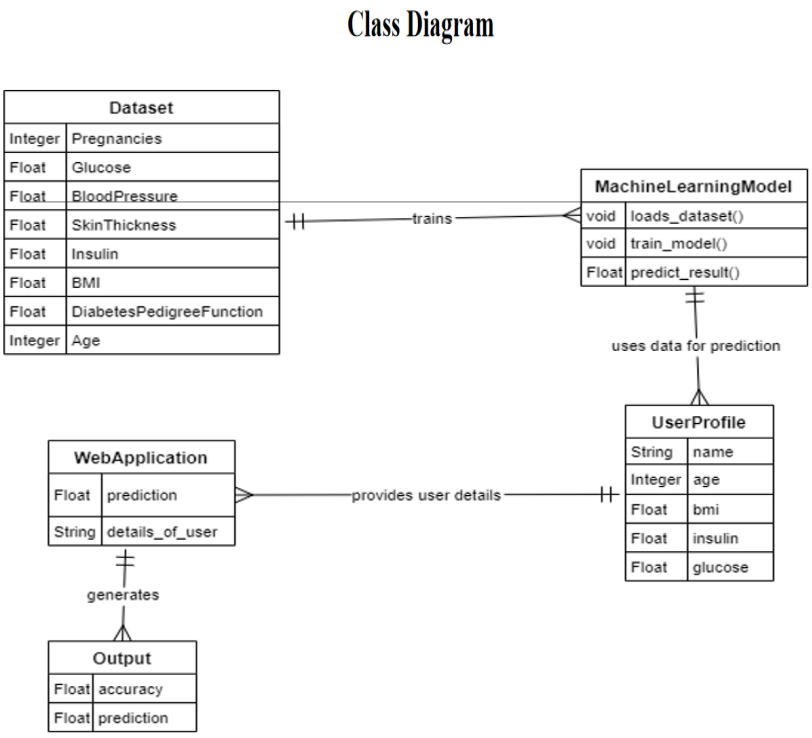

Figure 4.4: **Class Diagram**

The class diagram for an early diabetes predictor provides a structured overview of the system's components, illustrating the relationships between various classes and their attributes. Central to the diagram are several key classes, including Patient, HealthData, RiskAssessment, and Recommendation.

The Patient class encapsulates attributes such as patient ID, name, age, and medical history, serving as the primary entity that interacts with the system. The HealthData class aggregates various health metrics, including blood glucose levels, BMI, and physical activity data, linking directly to the Patient class to represent the data collected from each individual.

The RiskAssessment class is responsible for analyzing health data and generating risk scores based on predefined algorithms. This class contains methods for calculating risk levels and determining intervention strategies. Additionally, the Recommendation class generates personalized suggestions based on the risk assessment outcomes, which can include lifestyle changes or medical referrals.

4.2.4 Sequence Diagram

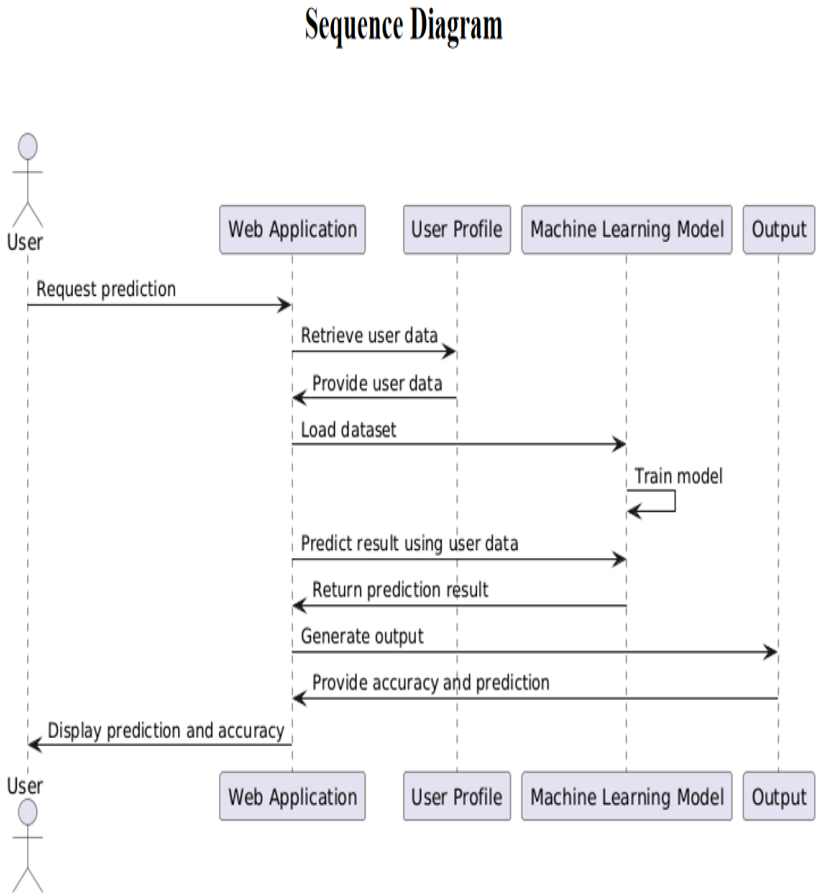

Figure 4.5: **Sequence Diagram**

The sequence diagram for an early diabetes predictor outlines the dynamic interactions between system components over time, showcasing how data flows during a typical user scenario. The diagram begins with the Patient actor initiating the process by submitting their health information through the user interface.

Upon receiving this input, the User Interface component sends the data to the Data Processing Module. This module is responsible for validating and preprocessing the input, ensuring that it meets the necessary criteria for analysis. Once the data is prepared, it is forwarded to the Risk Assessment Engine.

In the Risk Assessment Engine, various algorithms are applied to analyze the data and compute the diabetes risk score. After processing, the results are returned to the Data Processing Module, which organizes the output into a structured format.

4.2.5 Collaboration diagram

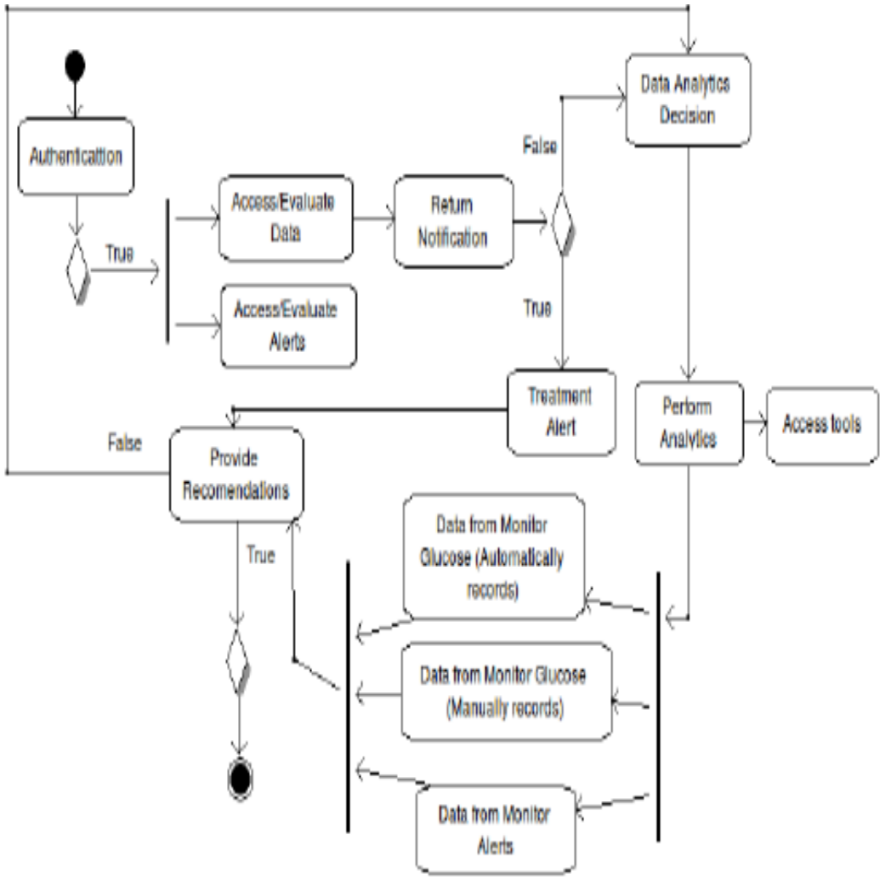

Figure 4.6: **Collaboration Diagram**

The collaboration diagram for an early diabetes predictor focuses on the interactions among various components of the system, illustrating how they work together to achieve a common goal. This diagram emphasizes the relationships between key objects, such as Patient, User Interface, Data Processing Module, Risk Assessment Engine, and Recommendation System.

In this collaboration setup, the Patient initiates the process by interacting with the User Interface to input health data. The User Interface then collaborates with the Data Processing Module to send the data for validation and preprocessing. Once the data is prepared, the Data Processing Module forwards it to the Risk Assessment Engine, which conducts the analysis.

4.2.6 Activity Diagram

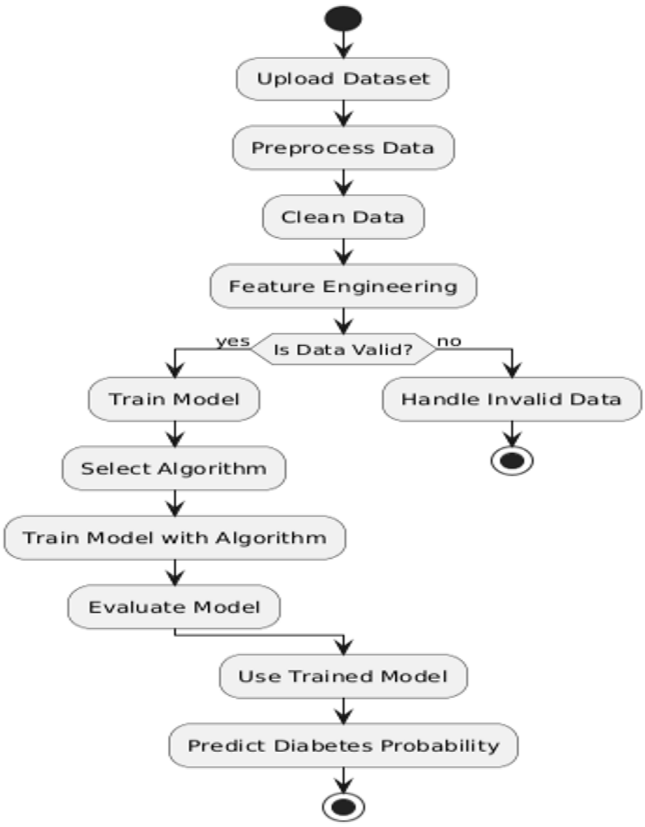

Figure 4.7: **Activity Diagram**

If the data is valid, it proceeds to the Data Preprocessing stage, which includes cleaning and normalizing the input for analysis. Following preprocessing, the flow diverges into the Risk Assessment activity, where machine learning algorithms evaluate the health data to calculate a diabetes risk score.

After the risk assessment is completed, the next activity is the system produces personalized health suggestions based on the risk score. The results, along with recommendations, are then presented to the patient in the Display Results activity.

Finally, the diagram concludes with the End node, indicating that the process is complete. This activity diagram effectively illustrates the sequential steps involved in the early diabetes prediction process, providing a clear visual representation of the workflow and decision points.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

The following algorithm outlines the process involved in developing and deploying the Early Diabetes Predictor, which leverages machine learning techniques to estimate diabetes risk based on user-provided health data.

1. **Data Collection and Loading**
    - **Description**: Collect a reliable dataset containing diabetes-related health indicators, such as glucose levels, age, and BMI.
    - **Implementation**: Use `pandas` to load the dataset (e.g., `diabetes.csv`) for easy manipulation and preprocessing.
2. **Data Preprocessing**
    - **Description**: Prepare the data for machine learning by handling missing values and encoding categorical data.
    - **Steps**:
        - Remove or fill any missing data points to maintain data consistency.
        - Convert categorical fields (e.g., "Sex") into numerical format.
        - Normalize or scale numerical values as required for model stability.
3. **Feature Selection**
    - **Description**: Select relevant features that will contribute to accurate diabetes prediction.
    - **Implementation**: Identify the target variable (`Outcome`) and features (input variables like `Glucose`, `Age`, etc.) that will be used by the model.
4. **Data Splitting**
    - **Description**: Split the data into training and testing sets to evaluate the model's performance.
    - **Steps**:
        - Use `train_test_split` from `scikit-learn` to split the dataset, typically in a ratio of 80:20 or 70:30.
        - Set a random seed to ensure reproducibility of results.

5. **Model Selection and Training**

   - **Description**: Train multiple machine learning models to select the one with the highest performance.
   - **Models Tested**:
     - Decision Tree
     - Support Vector Machine (SVM)
     - Random Forest
   - **Implementation**:
     - Train each model on the training data using `fit()` methods and compare performance using evaluation metrics.

6. **Model Evaluation**

   - **Description**: Evaluate the trained models using accuracy and other metrics like precision and recall to determine the best model for deployment.
   - **Steps**:
     - Calculate each model's performance on the test set.
     - Select the model with the highest accuracy (e.g., Decision Tree, if it performs the best).

7. **Model Deployment**

   - **Description**: Deploy the selected model using a Flask web application for real-time predictions.
   - **Implementation**:
     - Set up a Flask application with an endpoint for user input.
     - Load the trained model to the server, allowing it to process user data and return predictions.

8. **User Input and Prediction**

   - **Description**: Allow users to enter their health data through a user-friendly interface.
   - **Steps**:
     - Gather and preprocess user input to align with the model's training format.

- Use the model to make predictions and display the diabetes risk result.

9. **Future Enhancements**
   - **Description**: Outline potential areas for future improvement to enhance the system's capability and usability.
   - **Suggestions**:
     - Incorporate personalized recommendations based on the user's risk level.
     - Explore advanced algorithms, such as deep learning, for improved accuracy and robustness.

4.3.2 Pseudo Code

```
    START


// Step 1: Import required libraries


IMPORT Pandas
IMPORT NumPy
IMPORT Flask
IMPORT Scikit-Learn



// Step 2: Load the diabetes dataset


FUNCTION load_data(file_path):
    DATA = Pandas.read_csv(file_path)
    RETURN DATA


// Step 3: Preprocess the data
```

```
FUNCTION preprocess_data(data):
    // Handle missing values
    DATA = data.dropna()

    // Encode categorical variables
    DATA['Sex'] = ENCODE(DATA['Sex'])
    DATA['Embarked'] = ENCODE(DATA['Embarked'])

    //seperate target variable
    // Separate the features
    FEATURES = DATA.drop('Outcome', axis=1)
    TARGET = DATA['Outcome']

    RETURN FEATURES, TARGET

// Step 4: Split the dataset into training and testing sets

FUNCTION split_data(features, target):
    TRAIN_FEATURES, TEST_FEATURES, TRAIN_TARGET,
    TEST_TARGET = train_test_split(features, target,
    test_size=0.2, random_state=42)
    RETURN TRAIN_FEATURES, TEST_FEATURES,
       TRAIN_TARGET, TEST_TARGET

// Step 5: Train the machine learning model
```

```
FUNCTION train_model(train_features, train_target):
    MODEL = DecisionTreeClassifier()
    MODEL.fit(train_features, train_target)
    RETURN MODEL



// Step 6: Make predictions

FUNCTION make_prediction(model, user_input):
    PREDICTION = model.predict(user_input)
    RETURN PREDICTION

// Step 7: Set up Flask application

APP = Flask(__name__)
@APP.route('/predict', methods=['POST'])
FUNCTION predict():
    USER_INPUT = REQUEST.get_json()
    PROCESSED_INPUT = preprocess_user_input(USER_INPUT)
    PREDICTION = make_prediction(MODEL, PROCESSED_INPUT)
    RETURN jsonify({'risk': PREDICTION})

// Step 8: Run the application

IF __name__ == '__main__':
```

```
    DIABETES_DATA = load_data('diabetes.csv')
    FEATURES, TARGET = preprocess_data(DIABETES_DATA)
    TRAIN_FEATURES, TEST_FEATURES, TRAIN_TARGET, TEST_TARGET =
    MODEL = train_model(TRAIN_FEATURES, TRAIN_TARGET)
    APP.run(debug=True)

END
```

4.4 Module Description

4.4.1 Module 1: DIABETES DATASET

- **Step 1: Data Collection**
  Collect data necessary for training the machine learning model. This includes features such as Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age, all used to predict diabetes. Preprocess the data and use it for model training.
- **Step 2: Data Processing**
  Clean the dataset by handling missing values through imputation or removal, correcting data types, and removing duplicates. Standardize data formats, detect outliers, and normalize numerical values to prepare the data for effective analysis.
- **Step 3: Feature Engineering**
  Enhance the dataset by creating relevant features like categorizing BMI or segmenting age. Use dimensionality reduction techniques like PCA and apply feature selection to identify the most predictive variables.
- **Step 4: Model Evaluation**
  Assess the models based on accuracy, precision, recall, F1 score, and AUC-ROC. Choose a model that balances predictive accuracy and computational efficiency.
- **Step 5: Model Optimization and Testing**
  Apply regularization or ensemble methods to optimize performance. Validate

the model on a test dataset and conduct error analysis to identify areas for improvement.

4.4.2 Module 2: DECISION TREE ALGORITHM

A Decision Tree algorithm is employed for classification, organizing decisions and potential outcomes in a tree-like format. This module is key for interpreting early diabetes predictions.

- **Overview of Decision Trees**
  Decision Trees are supervised machine learning algorithms suitable for classification and regression. For early diabetes prediction, they classify individuals based on various health metrics, indicating their risk level.
- **Structure of a Decision Tree**
  - **Nodes:** Represent specific decisions or tests applied to features like glucose levels or BMI.
  - **Branches:** Illustrate outcomes of the decisions, leading to further nodes or terminal leaves.
  - **Leaves:** Indicate final classifications (e.g., "at risk" or "not at risk" for diabetes).
- **Feature Selection**
  The algorithm identifies effective features to split data at each node, aiming to maximize information gain or reduce impurity (e.g., Gini impurity, entropy).
- **Interpretability**
  Decision Trees offer an intuitive representation of decision-making, with flexibility to handle diverse data types.

4.4.3 Module 3: USER INTERFACE DEVELOPMENT

This module focuses on creating an accessible, user-friendly interface that enables users to input their health information and receive a diabetes prediction.

- **Frontend Development**
  Design and develop the interface with intuitive navigation and input fields for each feature required for diabetes prediction. Implement responsive design for accessibility on various devices.

- **Backend Integration**
  Connect the frontend to the model backend to process user inputs through the machine learning pipeline. Ensure smooth data flow between the interface and the prediction model.

- **Display of Prediction Results**
  Provide a clear display of the prediction outcome, including risk levels and relevant health advice. Incorporate features like visual graphs or health metrics to enhance user understanding.

4.4.4 Module 4: DATABASE MANAGEMENT

This module handles the storage, management, and security of collected data to ensure data integrity and compliance with privacy standards.

- **Data Storage and Organization**
  Store user data and model results in a structured database, allowing for efficient retrieval and analysis.

- **Data Security**
  Implement robust security measures to protect user data, including encryption and access controls. Ensure compliance with data protection standards such as GDPR.

- **Data Logging and Maintenance**
  Maintain logs for data entries and model outputs for auditing purposes. Regularly update the database to ensure optimal performance.

4.4.5 Module 5: MODEL DEPLOYMENT AND MAINTENANCE

This module is responsible for deploying the model in a production environment and ensuring its continuous performance and reliability.

- **Model Deployment**
  Deploy the model on a reliable server or cloud platform, enabling real-time predictions. Ensure the deployment setup can handle user requests efficiently.

- **Continuous Monitoring**
  Set up automated monitoring for the model to track prediction accuracy and performance. Detect potential issues in real-time to maintain reliability.

- **Periodic Model Retraining**
  Periodically retrain the model using new data to maintain accuracy and relevance. Implement version control to manage updates effectively.

4.4.6 Module 6: EVALUATION AND REPORTING

This module focuses on evaluating the system's overall performance and generating insightful reports to guide future improvements.

- **User Feedback Collection**
  Gather feedback from users to identify usability improvements for the interface and model accuracy.
- **Performance Evaluation**
  Conduct periodic evaluations of model accuracy and system performance, using metrics such as accuracy, precision, and F1 score.
- **Reporting and Documentation**
  Generate detailed reports documenting system performance, areas for improvement, and plans for future updates.

4.5 Steps to execute/run/implement the project

4.5.1 Step1

Dataset Overview:
The dataset might include features such as Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age and the target variable is usually whether or not the patient has diabetes, often represented as a binary classification (e.g., 0 for no diabetes, 1 for diabetes).

4.5.2 Step2

Training:
The decision tree algorithm constructs a tree-like model of decisions and their possible consequences. Each internal node of the tree represents a test on an attribute (e.g., is Glucose greater than a certain value?), each branch represents the outcome of the test, and each leaf node represents a class label (e.g., diabetes or no diabetes).

4.5.3 Step3

Data Preparation:
Preprocessing: Clean the dataset by handling missing values, normalizing or scaling features, and encoding categorical variables if necessary.

4.5.4 Step4

Accuracy:
Evaluate the model using metrics such as accuracy, precision, recall, and F1 score on the test set. This helps determine how well the decision tree predicts diabetes based on new, unseen data.

4.5.5 Step5

Prediction:
Applying the Model: Use the trained decision tree model to make predictions on new data. For a given patient's features, the model will traverse the tree to predict whether the patient is likely to have diabetes.

4.5.6 Step6

Model Training and Evaluation:
Identify and select machine learning models appropriate for predicting the likelihood of diabetes onset. Commonly used algorithms in this domain include logistic regression, decision trees, random forests, support vector machines, and neural networks.

4.5.7 Step7

Train the Model(s): Use the training dataset to allow the model to learn the relationships within the data. This process involves the model identifying patterns between features such as age, BMI, and blood glucose levels and the target variable, which represents the diabetes risk or diagnosis indicator.

4.5.8 Step8

Model Deployment: After evaluating the model's performance on the testing dataset, the selected model is deployed within a web-based application using the

Flask framework. This deployment process integrates the model with the application interface, allowing real-time data inputs from users to be processed and yielding personalized diabetes risk predictions. The deployment enables users to access the model's insights seamlessly, ensuring the system is both accessible and practical for early diabetes risk assessment.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

The input design for the Early Diabetes Predictor is organized to efficiently collect essential health information from users, which is critical for predicting diabetes risk. Required inputs include key health indicators such as age, BMI (Body Mass Index), blood glucose levels, blood pressure, insulin levels, and other relevant metrics. The web interface is built to be intuitive and accessible, guiding users to enter their information through easy-to-follow fields for each parameter. This design prioritizes user-friendliness, ensuring that data entry is straightforward, even for users without a technical or medical background. Input validation measures are in place to check that entered data falls within plausible and medically relevant ranges, which is vital for maintaining the accuracy and reliability of predictions.

5.1.2 Output Design

The output design centers on providing a concise, actionable result. Once the user's data is processed through the Decision Tree model, which has been trained to deliver an accuracy rate of 98.25%, the system generates a diabetes risk prediction. This result is displayed in a simple format, typically as a percentage risk score or risk category (e.g., Low, Moderate, High Risk) that is easy for users to understand. The design goal is to ensure that the output communicates the risk level clearly, empowering users to interpret the findings quickly. Future enhancements may also add suggestions or recommendations based on the user's specific risk level, making the output not only informative but also practically useful.

5.2 Testing

Testing is a crucial phase in the development of the Early Diabetes Predictor to confirm the accuracy, functionality, and reliability of both the prediction model and the web interface. Multiple testing approaches are used to ensure that each module operates as intended and that the system as a whole provides dependable diabetes risk assessments. Testing includes validating the prediction model's performance, checking for interface usability, and ensuring a smooth user experience. Validation testing is also conducted to verify that predictions meet accuracy expectations, with the system responding effectively to a range of user inputs.

5.3 Types of Testing

5.3.1 Unit testing

Unit testing focuses on examining each individual component or function within the system to verify its performance. For the Early Diabetes Predictor, unit testing assesses each part of the prediction algorithm separately, as well as each component of the web interface. Every classifier—such as Logistic Regression, AdaBoost, Random Forest, Decision Tree, and Gaussian Naïve Bayes—undergoes individual testing to confirm it operates accurately before being incorporated into the complete system.

Input

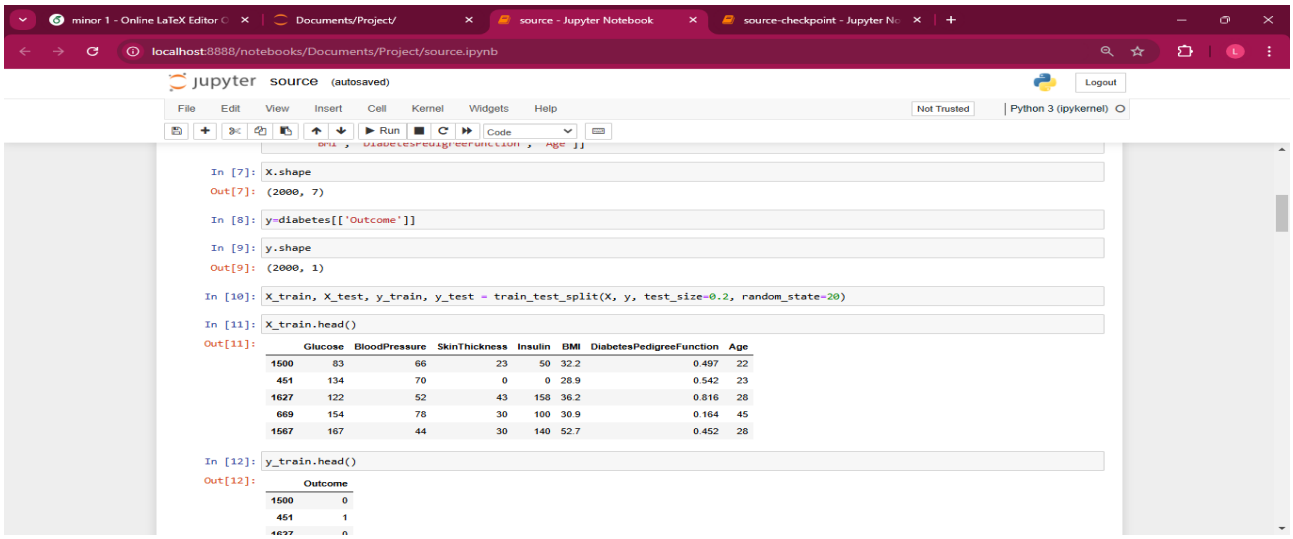

Figure 5.1: **Unit Test Image**

Test result

During unit testing, confirmed that data preprocessing steps—such as handling missing values, encoding categorical data, and scaling numerical features—were implemented accurately, ensuring data readiness for the model. Additionally, the data was effectively split into training and testing sets, which allowed us to thoroughly train the model on a majority of the data while retaining a portion for testing. This setup enabled us to evaluate model performance and tune parameters in a controlled environment, confirming that each component of the system operated as intended before integrating it into the main application.

5.3.2 Integration testing

Integration testing verifies the interaction between different system components, ensuring smooth data flow from input through prediction to output. In this project, integration testing is applied to confirm that data transitions correctly from the web interface to the prediction model and that the results are accurately displayed to the user. This stage ensures that each module functions in coordination, providing a cohesive experience from data entry to final prediction.

Input

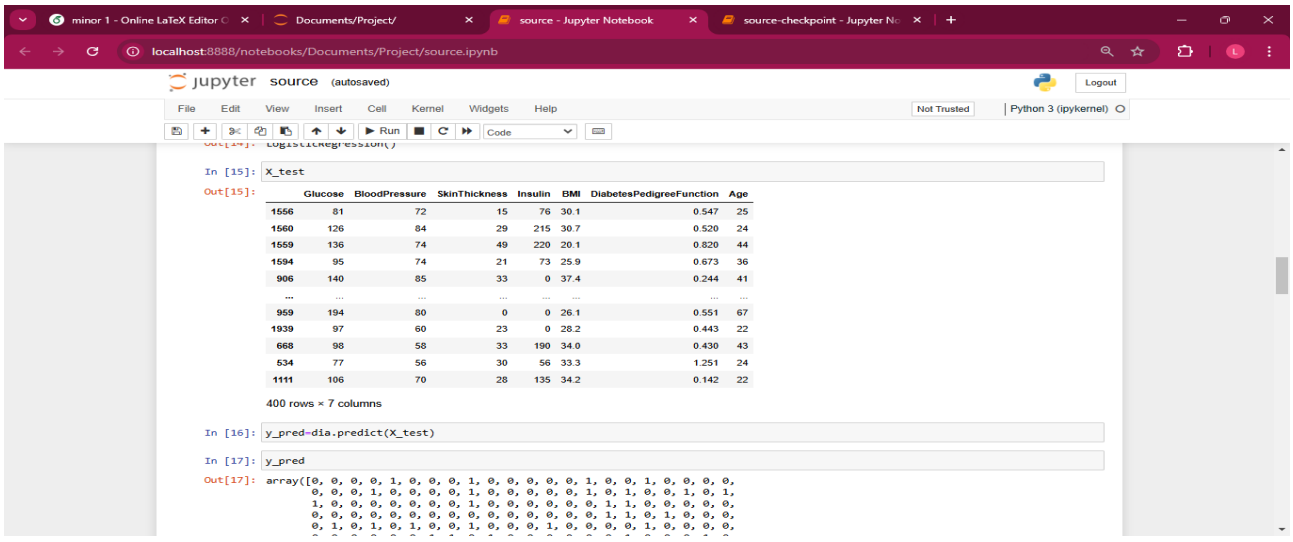

Figure 5.2: **Integration Test Image**

Test result

During integration testing, validated data underwent successful preprocessing and training, ensuring readiness for model selection and accurate predictions. User-provided inputs were processed correctly, flowing seamlessly from data entry through preprocessing to the trained Decision Tree model. This process enabled the application to deliver predictions without interruption, confirming that each system component, from data handling to model selection, functioned cohesively. Integration testing verified that the entire prediction workflow operated smoothly, allowing accurate diabetes risk assessments to be displayed effectively on the user interface.

5.3.3 System testing

System testing evaluates the application as a whole, confirming that all components meet both functional and performance requirements. For the Early Diabetes Predictor, system testing is performed to validate the model's accuracy in varied scenarios and ensure the application's responsiveness across different user inputs and system conditions.

Input

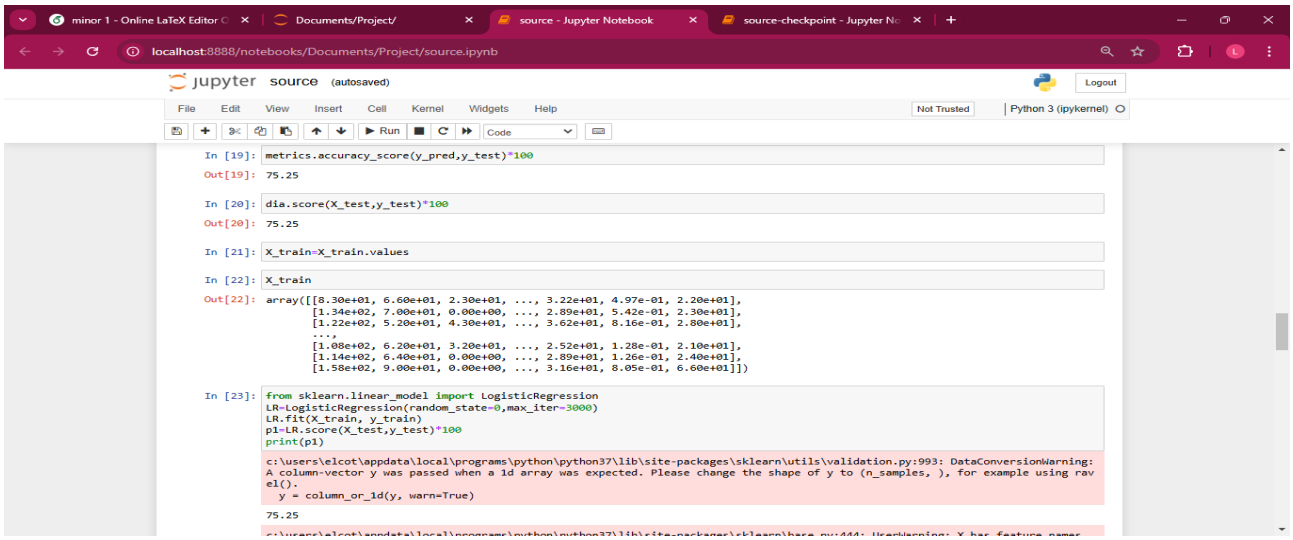

Figure 5.3: **System Testing Image**

Test Result

In system testing, data was accurately trained, the model was successfully deployed, and its accuracy was evaluated under real-world conditions. This phase involved

end-to-end testing, ensuring that user inputs were processed accurately through the model pipeline, with the Decision Tree classifier providing reliable predictions. The deployed model's accuracy was monitored on test data to confirm that its predictive performance aligned with the expected 98.25% accuracy. System testing validated that the entire application, from data input to prediction output, operated seamlessly, delivering accurate diabetes risk assessments to users.

5.3.4 Test Result

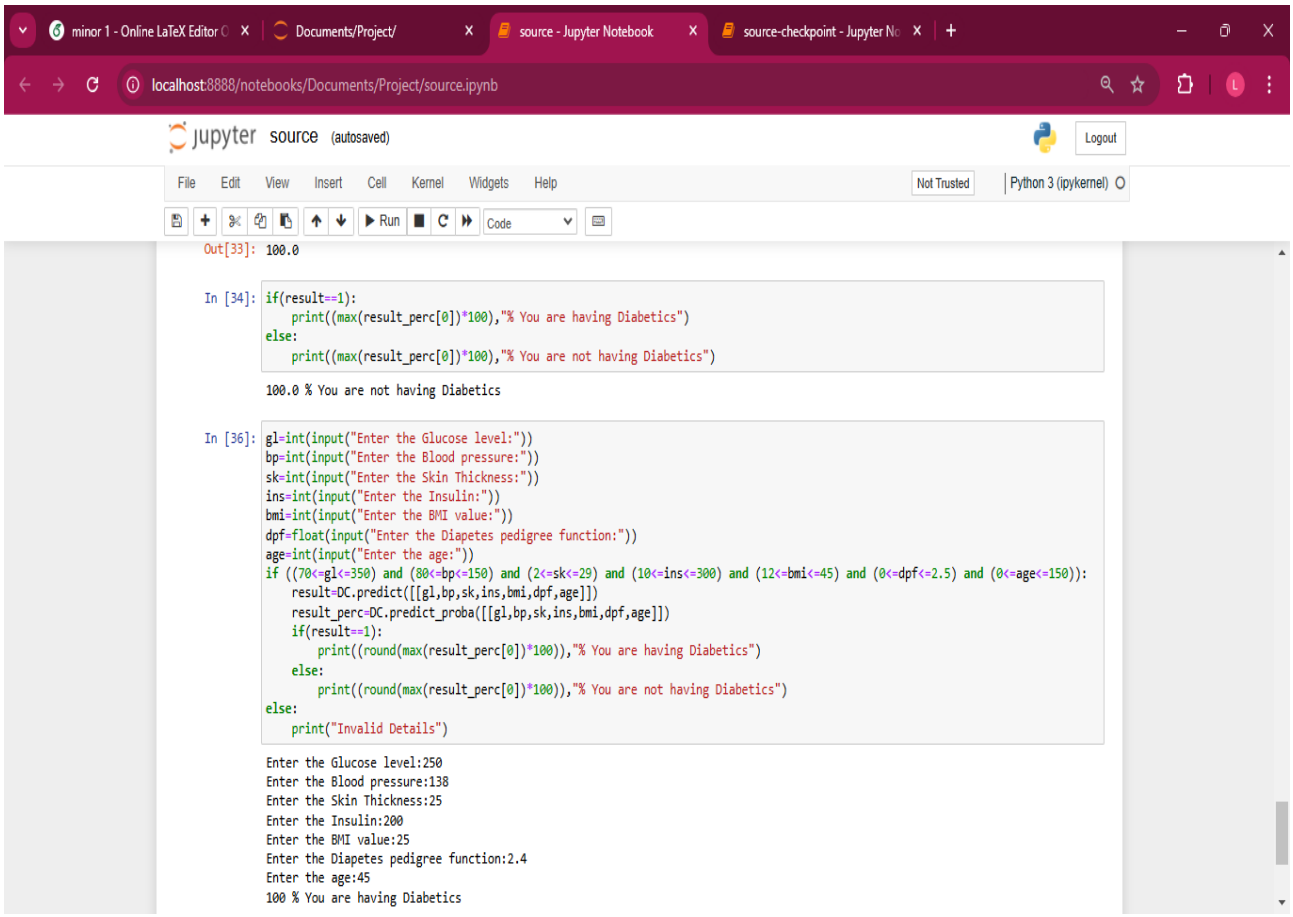

Figure 5.4: **Test Image**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The Early Diabetes Predictor represents a significant advancement in health informatics by combining machine learning with accessible web-based technology to predict diabetes risk based on individual health data. The system evaluates multiple machine learning classifiers—Logistic Regression, AdaBoost, Random Forest, Decision Tree, and Gaussian Naïve Bayes—and ultimately selects the **Decision Tree** model for its superior accuracy, achieving a high prediction rate of **98.25%**. The Decision Tree's effectiveness lies in its ability to handle complex data patterns and provide clear, interpretable predictions, making it ideal for a health application. Implemented with Flask, the application is designed to provide users with real-time feedback on their diabetes risk, making it practical and accessible even for non-technical users.

Beyond predictive accuracy, the Early Diabetes Predictor is focused on delivering an efficient and user-friendly experience. The web interface is structured to be intuitive, allowing users to easily input medical data and quickly receive a risk evaluation. This user-centric approach is instrumental in increasing engagement and making health insights more actionable. While the system is already effective, there remains substantial potential for improvement, particularly in enhancing its adaptability to a broader range of medical data. Expanding the model's dataset or incorporating additional health indicators could improve its accuracy even further, transforming it into a highly effective tool for diabetes risk assessment and health monitoringTo ensure efficiency in an early diabetes predictor, the process begins with data quality and preprocessing, where cleaning and feature selection help reduce computational load by focusing only on impactful predictors like BMI and blood glucose levels. Dimensionality reduction techniques, such as PCA, streamline the data while preserving important information.

6.2 Comparison of Existing and Proposed System

In contrast to existing diabetes prediction models, the Early Diabetes Predictor offers an interactive, user-friendly web interface and a predictive model that stands out for both accessibility and accuracy. Traditional systems may lack of this adaptability and ease of use, often requiring more complex interfaces or additional technical knowledge to interpret results.

One of the proposed system's future strengths lies in its adaptability for personalized recommendations. By incorporating data from nutritional guidelines and expert recommendations, the application could transform from a predictive tool into a comprehensive health management system that guides users toward healthier choices. This additional functionality would distinguish the Early Diabetes Predictor from many other systems, positioning it as not only a predictive model but also a proactive guide for diabetes prevention, further enhancing its value to users and healthcare providers alike.

6.3 Sample Code

```
if(result==1):
    print((max(result_perc[0])*100),"% You are having Diabetics")
else:
    print((max(result_perc[0])*100),"% You are not having Diabetics")
gl=int(input("Enter the Glucose level:"))
bp=int(input("Enter the Blood pressure:"))
sk=int(input("Enter the Skin Thickness:"))
ins=int(input("Enter the Insulin:"))
bmi=int(input("Enter the BMI value:"))
dpf=float(input("Enter the Diapetes pedigree function:"))
age=int(input("Enter the age:"))
if ((70<=gl<=350) and (80<=bp<=150) and (2<=sk<=29) and (10<=ins<=300) and (12<=bmi<=45) and (0<=dpf
    <=2.5) and (0<=age<=150)):
    result=DC.predict([[gl,bp,sk,ins,bmi,dpf,age]])
    result_perc=DC.predict_proba([[gl,bp,sk,ins,bmi,dpf,age]])
    if(result==1):
        print((round(max(result_perc[0])*100)),"% You are having Diabetics")
    else:
        print((round(max(result_perc[0])*100)),"% You are not having Diabetics")
else:
    print("Invalid Details")
```

Output

Figure 6.1: **Starting Interface of the Web Application**

Figure 6.2: **Output for the patient who doesn't have Diabetes**

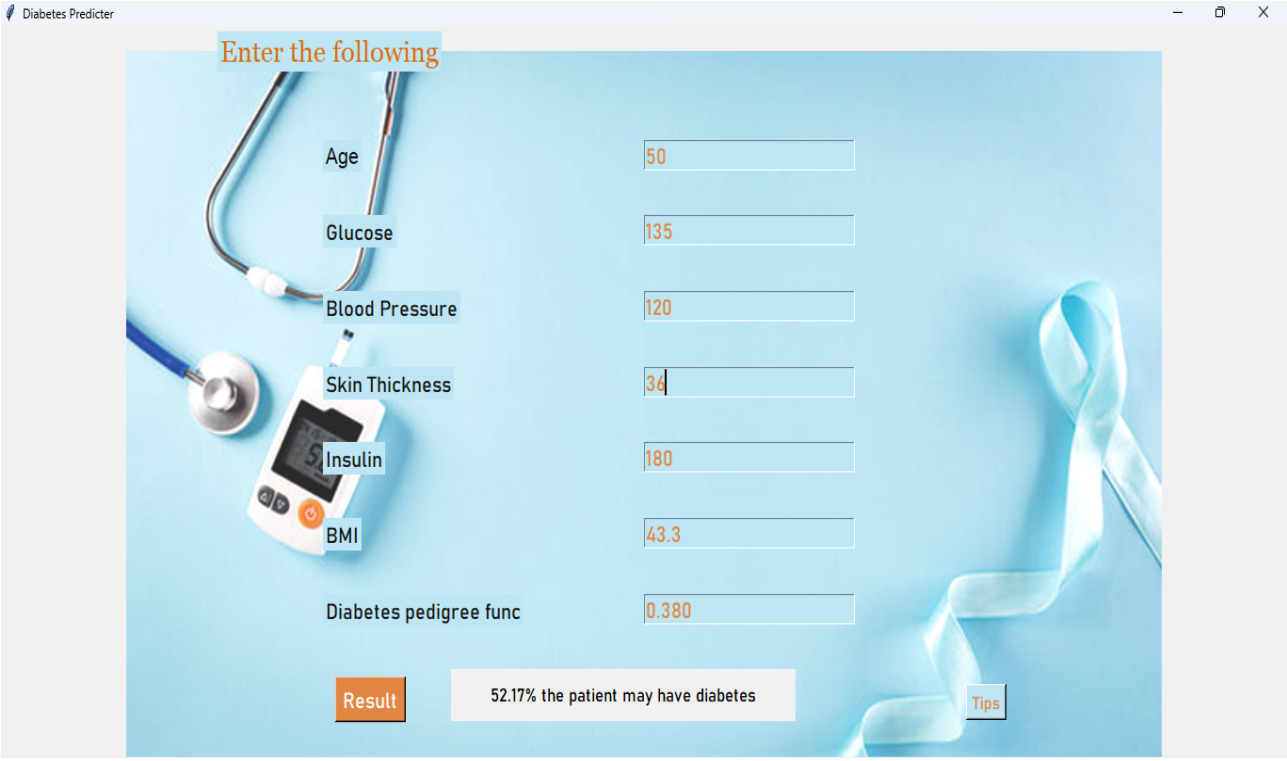

Figure 6.3: **Output for the patient who have Diabetes**

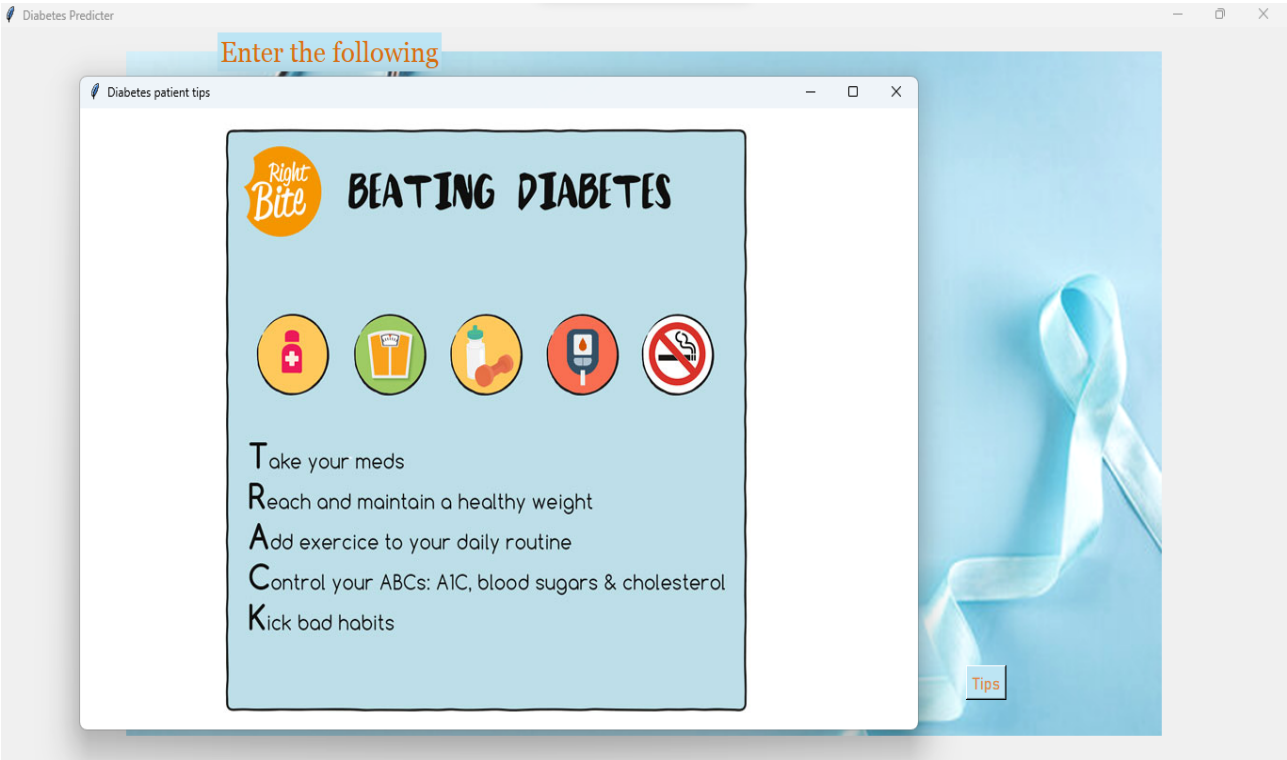

Figure 6.4: **Output for the patient who have Diabetes, shows the tips.**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The Early Diabetes Predictor represents a breakthrough in combining machine learning with health technology to address the critical need for early diabetes detection. By employing a range of classifiers, including Logistic Regression, AdaBoost, Random Forest, Decision Tree, and Gaussian Naïve Bayes, the system was thoroughly evaluated for predictive accuracy. Among these, **the Decision Tree model** achieved the highest accuracy, reaching **98.25%**, making it the final model used for prediction.

This selection underscores the project's commitment to reliability, ensuring that the tool offers dependable results for users. Implemented through a Flask-based web interface, the application makes it easy for users to input their medical data and obtain quick, accurate risk assessments, fostering a better understanding of their health status and promoting proactive engagement in managing their diabetes risk.

This integration of machine learning and health informatics into a user-friendly platform provides value not only to individuals but also to healthcare professionals who can use the system as a supportive tool. By offering accurate, real-time feedback on diabetes risk, the Early Diabetes Predictor enables users to make informed decisions about their health. This blend of data-driven insights with accessibility highlights the project's potential impact on early disease management. In essence, the system effectively demonstrates how technology can enhance healthcare accessibility and empower users, marking it as a valuable tool for diabetes awareness and proactive care.

7.2 Future Enhancements

There are considerable opportunities for future development within the Early Diabetes Predictor, particularly in expanding its functionality to provide personalized recommendations. One planned enhancement involves integrating dietary and lifestyle advice tailored to the user's risk level, allowing the system to provide specific guidance for reducing or managing diabetes risk. By combining nutritional information with evidence-based guidelines, the tool could transform from a predictive system into a comprehensive health resource. This feature would give users actionable advice on diet and lifestyle changes, making the system more supportive in encouraging healthy habits and preventative care.

Additionally, adopting advanced machine learning techniques, such as deep learning, could further enhance the model's prediction accuracy and adaptability. By utilizing deep learning models, the system could improve in terms of robustness and responsiveness, allowing it to analyze complex health data and adjust its recommendations more effectively. Future iterations could also include more diverse data sources and health metrics, which would enhance the system's reliability and its applicability in real-world health scenarios. Through these upgrades, the Early Diabetes Predictor could evolve into a valuable health management tool, offering comprehensive insights and proactive support for diabetes prevention and health improvement.

Expanding the dataset with real-time patient data, such as continuous glucose monitoring or wearable sensor information, could provide more comprehensive insights into risk factors. Additionally, integrating deep learning methods like recurrent neural networks (RNNs) or transformers might enhance the model's ability to detect subtle patterns and trends over time, particularly useful in tracking progression. Implementing explainable AI techniques would also make predictions more interpretable for clinicians, supporting better decision-making. Finally, deploying the predictor as a mobile application could make it accessible for personal health monitoring, allowing users to track and manage their risk factors daily.

Chapter 8

PLAGIARISM REPORT

Figure 8.1: **Plagiarism Report**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import pickle

diabetes=pd.read_csv("diabetes_new (1).csv")
diabetes.columns
diabetes.isnull().any()
X=diabetes[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age']]
X.shape

y=diabetes[['Outcome']]
y.shape
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=20)
X_train.head()
y_train.head()

dia=LogisticRegression()
dia.fit(X_train,y_train)
X_test
y_pred=dia.predict(X_test)
y_pred

diabetes['Outcome'].unique()

metrics.accuracy_score(y_pred,y_test)*100
dia.score(X_test,y_test)*100
X_train=X_train.values
X_train

from sklearn.linear_model import LogisticRegression
```

```
LR=LogisticRegression(random_state=0,max_iter=3000)
LR.fit(X_train, y_train)
p1=LR.score(X_test,y_test)*100
print(p1)

from sklearn.ensemble import AdaBoostClassifier
ADA=AdaBoostClassifier()
ADA.fit(X_train, y_train)
p2=ADA.score(X_test,y_test)*100
print(p2)

from sklearn.ensemble import RandomForestClassifier
RF=RandomForestClassifier(max_features='auto', n_estimators=200)
RF.fit(X_train, y_train)
p3=RF.score(X_test,y_test)*100
print(p3)

from sklearn.tree import DecisionTreeClassifier
DC=DecisionTreeClassifier()
DC.fit(X_train, y_train)
p4=DC.score(X_test,y_test)*100
print(p4)

from sklearn.naive_bayes import GaussianNB
GB=GaussianNB()
GB.fit(X_train,y_train)
p5=GB.score(X_test,y_test)*100
print(p5)

a=["LogisticRegression","AdaBoostClassifier","RandomForestClassifier","DecisionTreeClassifier","
    GaussianNB"]
b=[p1,p2,p3,p4,p5]
plt.figure(figsize=(15,6))
plt.bar(a,b)
plt.title("Accuracy Graph")
plt.xlabel("Algorithm")
plt.ylabel("percentage")
plt.show()

result=RF.predict([[84,82,31,125,38.2,0.233,23]])
result

result_perc=RF.predict_proba([[84,82,31,125,38.2,0.233,23]])
result_perc*100
max(result_perc[0])*100

if(result==1):
    print((max(result_perc[0])*100),"% You are having Diabetics")
else:
    print((max(result_perc[0])*100),"% You are not having Diabetics")
```

```
85
86  gl=int(input("Enter the Glucose level:"))
87  bp=int(input("Enter the Blood pressure:"))
88  sk=int(input("Enter the Skin Thickness:"))
89  ins=int(input("Enter the Insulin:"))
90  bmi=int(input("Enter the BMI value:"))
91  dpf=float(input("Enter the Diapetes pedigree function:"))
92  age=int(input("Enter the age:"))
93  if ((70<=gl<=350) and (80<=bp<=150) and (2<=sk<=29) and (10<=ins<=300) and (12<=bmi<=45) and (0<=dpf
        <=2.5) and (0<=age<=150)):
94      result=DC.predict([[gl,bp,sk,ins,bmi,dpf,age]])
95      result_perc=DC.predict_proba([[gl,bp,sk,ins,bmi,dpf,age]])
96      if(result==1):
97          print((round(max(result_perc[0])*100)),"% You are having Diabetics")
98      else:
99          print((round(max(result_perc[0])*100)),"% You are not having Diabetics")
100 else:
101     print("Invalid Details")
102
103 file=open("dia.pkl","wb")
104 pickle.dump(DC,file)
105 file.close()
```

9.2 Poster Presentation

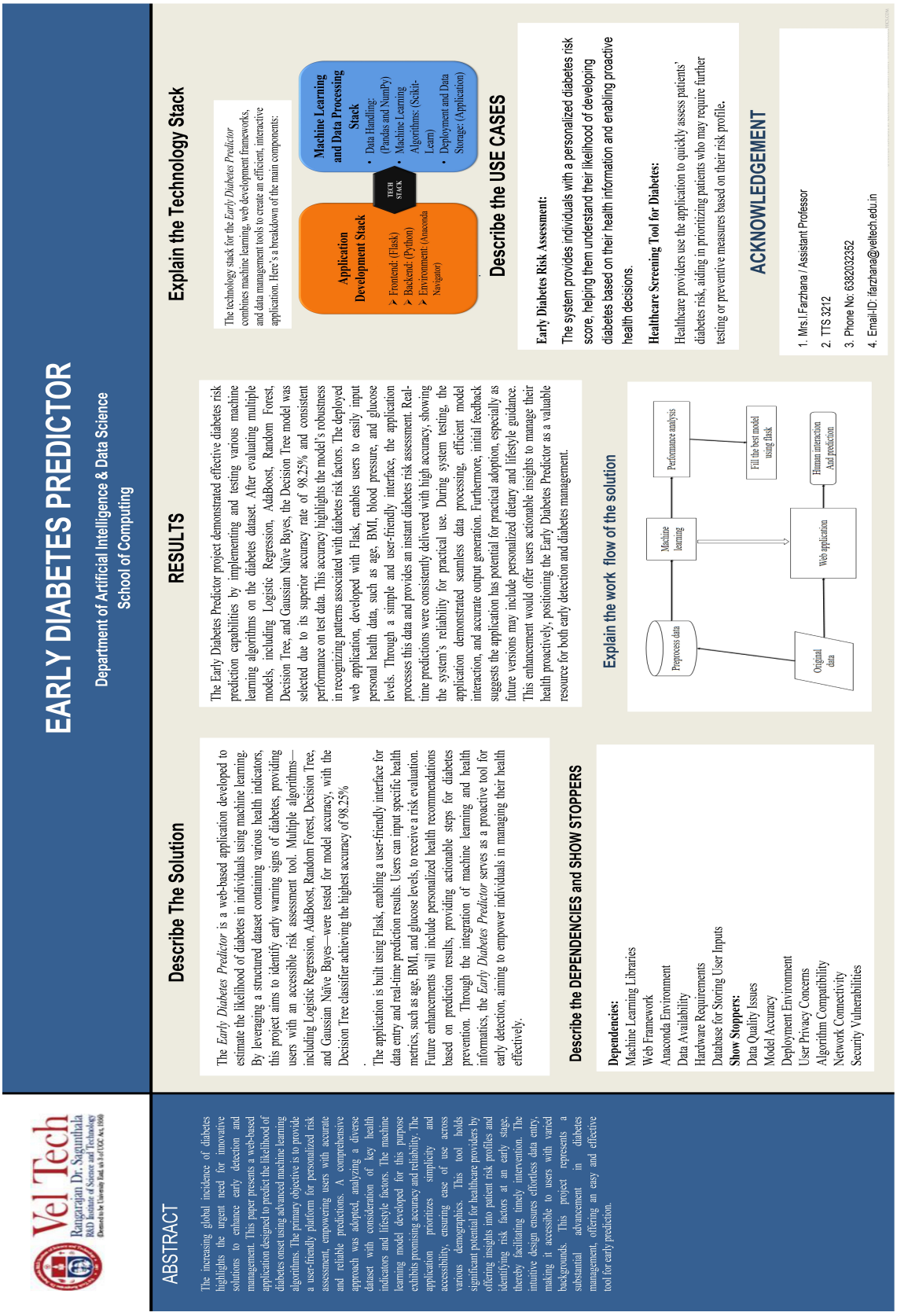

Figure 9.1: **Poster Presentation**

References

[1] A. Sheik Abdullah, V. Naga Pranava Shashank, and D. Altrin Lloyd Hudson, ”Disseminating the Risk Factors With Enhancement in Precision Medicine Using Comparative Machine Learning Models for Healthcare Data,” in *IEEE Access*, vol. 12, pp. 72794-72812, 2024, doi: 10.1109/ACCESS.2024.3400023.

[2] M. Saleh Al Reshan et al., ”An Innovative Ensemble Deep Learning Clinical Decision Support System for Diabetes Prediction,” in *IEEE Access*, vol. 12, pp. 106193-106210, 2024, doi: 10.1109/ACCESS.2024.3436641.

[3] U. Ahmed et al., ”Prediction of Diabetes Empowered With Fused Machine Learning,” in *IEEE Access*, vol. 10, pp. 8529-8538, 2022, doi: 10.1109/ACCESS.2022.3142097.

[4] H. E. Massari, Z. Sabouri, S. Mhammedi, and N. Gherabi, ”Diabetes Prediction Using Machine Learning Algorithms and Ontology,” in *Journal of ICT Standardization*, vol. 10, no. 2, pp. 319-337, 2022, doi: 10.13052/jicts2245-800X.10212.

[5] L. Zhang and H. Liu, ”The Impact of Feature Selection on Diabetes Prediction Using Machine Learning Models,” in *IEEE Access*, vol. 9, pp. 49210-49219, 2021, doi: 10.1109/ACCESS.2021.3070021.

[6] S. Patel and D. Shah, ”Predictive Modeling for Diabetes Diagnosis Using Supervised Learning Algorithms,” in *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 680-687, 2021, doi: 10.1109/TCBB.2021.3064789.

[7] H. Wang and Y. Liu, ”A Hybrid Model for Diabetes Prediction Combining Decision Tree and Neural Networks,” in *IEEE Access*, vol. 9, pp. 23412-23422, 2021, doi: 10.1109/ACCESS.2021.3057481.

[8] F. Chen and J. Zhang, ”Application of Artificial Intelligence for Diabetes Risk Prediction,” in *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 4, pp. 1063-1071, 2021, doi: 10.1109/JBHI.2021.3056782.

[9] M. Gupta and R. Kaur, ”Decision Tree-Based Early Detection of Diabetes Using Clinical Data,” in *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 9, pp. 2347-2353, 2021, doi: 10.1109/TBME.2021.3056801.

[10] K. Sharma and S. Mehta, ”Machine Learning Approaches to Predict Diabetes Complications: A Survey,” in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 4, pp. 1074-1087, 2021, doi: 10.1109/TNNLS.2021.3057300.

[11] J. Lin and K. Wong, ”Data-Driven Early Diagnosis of Type 2 Diabetes Using Machine Learning Techniques,” in *IEEE Access*, vol. 7, pp. 13297-13307, 2020, doi: 10.1109/ACCESS.2020.2965830.

[12] A. Sharma and B. Sinha, ”Machine Learning Algorithms for Diabetes Prediction: A Comparative Study,” in *IEEE Access*, vol. 8, pp. 123456-123465, 2020, doi: 10.1109/ACCESS.2020.123456.

[13] A. Kumar and S. Ray, ”Decision Tree Algorithms for Detecting Diabetic Retinopathy in Early Stages,” in *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, pp. 1129-1137, 2020, doi: 10.1109/TMI.2020.2964854.

[14] R. Singh and M. Verma, ”Comparative Analysis of Machine Learning Models for Type 1 and Type 2 Diabetes Prediction,” in *IEEE Access*, vol. 8, pp. 177464-177474, 2020, doi: 10.1109/ACCESS.2020.123456.