

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA” Belagavi – 590018, Karnataka



MINI PROJECT REPORT ON

“DIABETES DIAGNOSIS USING MACHINE LEARNING”

Submitted in partial fulfillment for the award of the degree in

BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SUBMITTED BY

ESHWAR PAWAN P 1BH22AI014

G UDHBAV 1BH22AI016

SABARI GOVINDHAN 1BH22AI038

TEJAS HALEMANI 1BH22AI054

Under the Guidance of

Mrs. Dhivya C

Assistant Professor



Department of Artificial Intelligence And Machine Learning

BANGALORE TECHNOLOGICAL INSTITUTE

(An ISO 9001:2005 Certified Institute)

Kodathi Village, Varthoor Hobli, Banagalore East Tq, Bangalore Urban District Bangalore –

560035, Karnataka

2024 – 2025



BANGALORE TECHNOLOGICAL INSTITUTE

(An ISO 9001:2015 Certified Institute)

Kodathi Village, Varthoor Hobli, Bangalore East Tq, Bangalore Urban District,
Bangalore-560035, Karnataka.

principal@btibangalore.org

Phone: 7090404050

www.btibangalore.org

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

This is to certify that the mini project report on “**Diabetes Diagnosis Using Machine Learning**” carried out by Eshwar Pawan P [1BH22AI014], G Udhbav [1BH22AI016], Sabari Govindhan [1BH22AI038], Tejas Halemani [1BH22AI054] the Bonafide students of **Bangalore Technological Institute**, Bengaluru in partial fulfillment for the award of **Bachelor of Engineering in Artificial Intelligence and Machine Learning** of **Visvesvaraya Technological University, Belagavi** during the academic year **2024-2025**. Thus, it is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The Mini Project report has been approved, as it satisfies the academic requirements in the respect of the mini project report prescribed for the said degree.

Mrs. Dhivya C

Assistant prof, mini project guide

Dept. of AIML

Mrs. Dhivya C

Asst. Prof, Mini Project Coordinator

Dept. of AIML

Dr. G Gayatri Tanuja

HOD Dept. of AIML

Dr. H S Nanda

Principal



BANGALORE TECHNOLOGICAL INSTITUTE

(An ISO 9001:2015 Certified Institute)

Kodathi Village, Varthoor Hobli, Bangalore East Tq, Bangalore Urban District,
Bangalore-560035, Karnataka.

principal@btibangalore.org

Phone: 7090404050

www.btibangalore.org

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DECLARATION

We, the students of third year Artificial Intelligence and Machine Learning, Bangalore Technological Institute, Bangalore, hereby declare the mini project entitled “**Diabetes Diagnosis using Machine Learning**” has been independently carried out by us under the guidance of “**Mrs. Dhiyya C**”, Assistant Professor, Department of Artificial Intelligence and Machine Learning, Bangalore Technological Institute, Bengaluru and Submitted in partial fulfillment of the requirements for the award of the degree in **Bachelor of Engineering in Artificial Intelligence And Machine Learning** of the **Visvesvaraya Technological University**, Belagavi during the academic year **2024 – 2025**.

We also declare that to the best of our knowledge and believe the work reported here does not form part of any other mini project on the basis of which a degree or award was conferred on an early occasion of this by any other.

Date:

Place: Bengaluru

Student Name	USN	Signature
Eshwar Pawan P	1BH22AI014	_____
G Udhbav	1BH22AI016	_____
Sabari Govindhan	1BH22AI038	_____
Tejas Halemani	1BH22AI054	_____

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals and elders. We would like to take this opportunity to thank them all.

We heartily extend my words of gratitude to the Mini Project Coordinator and our Mini Project Guide **Mrs. Dhivya C**, for her valuable advice, encouragement and suggestion given us in the course of our mini project work.

We would like to express our immense gratitude to Head of Department **Dr. G Gayatri Tanuja**, for her unfailing encouragement and suggestion given to us in course of mini project work.

We would like to take opportunity to express our gratitude to the Principal **Dr. H S Nanda**, for giving us this opportunity to enrich our knowledge.

We are grateful to the President **Dr. Prabhakara Reddy A** and Secretary, **Sri. C L Gowda** for having provided us with a great infrastructure and well – furnished labs.

Finally, a note of thanks to the Department of Artificial Intelligence And Machine Learning, both teaching and non – teaching staff for their co-operation extended to us.

Last but not the least, we acknowledge the support and feedback of our parents, guardians and friends, for their indispensable help always.

Eshwar Pawan P	: 1BH22AI014
G Udhav	: 1BH22AI016
Sabari Govindhan	: 1BH22AI038
Tejas Halemani	: 1BH22AI054

ABSTRACT

The increasing prevalence of diabetes necessitates the development of effective diagnostic tools to facilitate early detection and intervention. Recent advancements in machine learning (ML) offer promising avenues for enhancing diabetes diagnosis through sophisticated predictive modeling. This abstract summarizes a comprehensive study that explores various ML techniques, including logistic regression, decision trees, support vector machines, and ensemble methods, to classify and predict diabetes risk based on clinical data. Utilizing datasets such as the Pima Indian Diabetes Database, which includes critical variables like blood glucose levels, body mass index (BMI), age, and family history, the study evaluates the performance of these models in terms of accuracy, precision, recall, and F1-score. Notably, results indicate that models like K-nearest neighbors (K-NN) and random forest exhibit superior predictive capabilities, achieving accuracies exceeding 90%. Furthermore, the integration of deep learning approaches, such as convolutional neural networks (CNNs) combined with long short-term memory (LSTM) networks, has demonstrated even higher accuracy rates in diagnosing diabetes. The findings underscore the potential of ML algorithms not only to improve diagnostic accuracy but also to facilitate personalized treatment plans by identifying at-risk individuals early. This research highlights the critical role of feature selection and data preprocessing in enhancing model performance and advocates for the implementation of ML-based diagnostic systems in clinical settings to mitigate the growing burden of diabetes-related complications.

INDEX

Acknowledgment	I
Abstract	II
Content	III
List of Figures	IV

CONTENT

Sl. No	Chapter	Page No.
01	Introduction	1 - 5
	1.1. Overview	1
	1.2. Problem Statement	2
	1.3. Objective	2
	1.4. Motivation	4
	1.5. Application	4
02	Literature Survey	6
03	System Analysis	7 - 9
	3.1. Existing Systems and their Drawbacks	7
	3.2. Proposed System	8
	3.3. Overcoming Limitations of Existing System	8
04	Objectives & Methodology	10 - 14
	4.1. Objectives	10
	4.2. Methodology	12
05	Requirement Analysis	15 - 20
	5.1. Functional Requirements	15
	5.2. Non-Functional Requirements	17
	5.3. Technical Requirements	18
	5.4. Hardware Requirements	20
06	Implementation	21 - 23
07	Conclusion	24 - 25
08	Future Enhancement	26 - 28
	References	29
	Appendix - A : Code	30 - 40
	Appendix - B : Snapshots	31 - 43
	Photo Gallery	44

LIST OF FIGURES

Fig. No	Title of the Figure	Page No.
4.1	Basic Methodology	12
4.2	Example Data Set	13
4.3	Neural Network	13
6.1	Implimentation Diagram	21

Chapter 1

INTRODUCTION

Diabetes is a chronic metabolic disorder characterized by persistently high blood sugar levels, primarily due to the body's inability to produce sufficient insulin or effectively utilize the insulin it produces. This condition can lead to severe health complications if not diagnosed and managed promptly. The increasing global prevalence of diabetes, particularly Type 2 diabetes mellitus (T2DM), underscores the urgent need for early diagnosis and effective management strategies. Early intervention is crucial in preventing complications such as cardiovascular diseases, neuropathy, and kidney failure, which can significantly impair quality of life and increase healthcare costs.

Traditional diagnostic methods for diabetes typically involve laboratory tests such as fasting blood glucose tests, oral glucose tolerance tests, and HbA1c measurements. While these methods are reliable, they can be time-consuming and may not always provide timely results. Patients often face delays in diagnosis due to the need for multiple visits to healthcare facilities for testing and follow-up consultations. This lag in diagnosis can result in missed opportunities for early intervention, allowing the disease to progress and lead to more severe health issues.

In recent years, the integration of machine learning (ML) techniques into diabetes diagnosis has emerged as a promising approach that addresses some of these challenges. Machine learning algorithms can analyze large datasets that include a variety of demographic, clinical, and lifestyle factors to identify patterns associated with diabetes risk. By leveraging these advanced analytical techniques, healthcare providers can achieve faster and more accurate assessments compared to traditional methods.

1.1 Overview

Machine learning (ML) involves the use of algorithms and statistical models that enable computers to perform specific tasks without explicit instructions, thereby allowing for automated analysis and decision-making. In the context of diabetes diagnosis, ML has emerged as a powerful tool capable of analyzing complex datasets to identify patterns and correlations that may not be immediately apparent through conventional diagnostic methods. This capability is particularly valuable given the multifactorial nature of diabetes, which can be influenced by a wide range of factors including genetics, lifestyle,

and environmental conditions.

One of the significant advantages of employing ML in diabetes diagnosis is its ability to utilize various data sources. These sources include electronic health records (EHRs), which provide comprehensive patient histories; genetic information that can reveal predispositions to diabetes; lifestyle factors such as diet and physical activity levels; and even real-time monitoring data from wearable devices like glucose monitors and fitness trackers. By integrating these diverse data streams, ML algorithms can create a more complete picture of an individual's health status, leading to more accurate predictions regarding their risk of developing diabetes.

1.2 Problem Statement

Diabetes is a chronic health condition that affects millions of people worldwide and can lead to serious complications if left untreated. Early detection of diabetes is critical for effective management and prevention of severe health outcomes.

Traditional diagnostic methods can be time-consuming and often rely on invasive procedures. In recent years, machine learning has emerged as a powerful tool in predictive healthcare, offering the potential to predict diabetes at an early stage based on various health indicators.

This project aims to develop a machine learning model to accurately predict the likelihood of an individual developing diabetes. The model will be trained on medical data, including factors such as age, body mass index (BMI), glucose levels, insulin levels, and other relevant features. By leveraging advanced algorithms, the goal is to improve the accuracy and efficiency of diabetes diagnosis, facilitating earlier interventions and better patient outcomes.

1.3 Objective

The primary objective of employing machine learning (ML) in diabetes diagnosis is to develop predictive models that can effectively identify individuals at risk of diabetes or those already diagnosed with the condition. This innovative approach aims to transform the landscape of diabetes care by leveraging advanced computational techniques to enhance diagnostic processes and patient management. The specific goals of integrating ML into diabetes diagnosis can be elaborated as follows:

Enhancing Diagnostic Accuracy

One of the foremost goals of utilizing machine learning in diabetes diagnosis is to

improve the precision of diagnoses by minimizing false positives and false negatives. Traditional diagnostic methods often rely on subjective clinical assessments and standardized laboratory tests, which can sometimes lead to misdiagnoses. In contrast, ML algorithms can analyze vast amounts of data from electronic health records, genetic information, and lifestyle factors to identify subtle patterns that may indicate diabetes risk. For example, studies have shown that machine learning models can achieve accuracies ranging from 71% to 94% in predicting new-onset diabetes, outperforming conventional statistical methods. By enhancing diagnostic accuracy, healthcare providers can ensure that patients receive timely and appropriate interventions.

Early Detection

Facilitating timely intervention through early identification of at-risk individuals is another critical goal of employing ML in diabetes diagnosis. Early detection is essential in preventing the progression of diabetes and its associated complications, such as cardiovascular diseases, neuropathy, and kidney failure. Machine learning models can analyze data from various sources—including demographic information, medical history, and real-time monitoring from wearable devices—to predict the likelihood of developing diabetes before clinical symptoms manifest. For instance, research has demonstrated that models trained on comprehensive datasets can identify prediabetes with high sensitivity, allowing for proactive lifestyle modifications and medical interventions. This proactive approach not only improves patient outcomes but also reduces long-term healthcare costs associated with managing advanced diabetes complications.

Personalized Treatment Plans

Another significant advantage of integrating machine learning into diabetes care is the ability to tailor treatment strategies based on individual risk profiles derived from ML analysis. By considering a patient's unique combination of genetic predispositions, lifestyle factors, and clinical history, healthcare providers can develop personalized treatment plans that address specific needs. For example, machine learning algorithms can recommend individualized dietary plans or exercise regimens based on predicted postprandial glucose responses. This personalization enhances the effectiveness of interventions and empowers patients to take an active role in managing their health.

Resource Optimization

The implementation of machine learning models in diabetes diagnosis also aims to optimize healthcare resources by streamlining diagnostic processes. Traditional

diagnostic workflows often involve multiple appointments for testing and follow-up consultations, which can strain healthcare systems and lead to delays in treatment. By automating data analysis and risk assessment through ML algorithms, healthcare providers can expedite the diagnostic process, allowing for quicker decision-making and resource allocation. This efficiency not only alleviates the burden on healthcare professionals but also improves patient satisfaction by reducing wait times for diagnoses and treatment plans

1.4 Motivation

The motivation behind utilizing machine learning for diabetes diagnosis stems from several factors:

Rising Diabetes Incidence:

The World Health Organization (WHO) estimates that diabetes will be the seventh leading cause of death by 2030. The increasing incidence demands innovative solutions for effective management.

Limitations of Traditional Methods:

Conventional diagnostic techniques can be slow and may require multiple visits to healthcare facilities. Machine learning offers a more efficient alternative.

Data Availability:

The proliferation of health data from electronic health records and wearable devices provides a rich resource for training ML models.

Technological Advancements:

Recent advancements in computational power and algorithm development have made it feasible to implement sophisticated ML techniques in clinical settings.

1.5 Application

Machine learning applications in diabetes diagnosis encompass various domains:

Predictive Modeling:

Algorithms can analyze patient data to predict the likelihood of developing type 2 diabetes based on factors such as age, body mass index (BMI), family history, and lifestyle choices.

Classification Systems:

ML models can classify patients into different categories (e.g., diabetic vs. non-diabetic)

based on their clinical data, enabling targeted interventions.

Risk Assessment Tools:

Development of web-based or mobile applications that utilize ML algorithms to provide users with personalized risk assessments based on their input data.

Continuous Monitoring:

Integration with wearable devices allows for real-time monitoring of glucose levels and other vital signs, with ML algorithms providing insights into potential health risks.

Research and Development:

Machine learning facilitates large-scale studies that can uncover new biomarkers for diabetes or identify novel therapeutic targets.

Chapter 2

LITERATURE SURVEY

Diabetes affects millions worldwide and can lead to severe complications if undiagnosed. Early prediction is crucial for effective management, and machine learning (ML) has emerged as a powerful tool in this domain. ML models analyze large datasets, identifying patterns in factors like age, BMI, glucose levels, and family history to predict diabetes risk, offering more accurate results compared to traditional methods.

1. ML in Healthcare

Kavakiotis et al. (2017) found that ML algorithms like decision trees and neural networks excel in predicting diabetes by analyzing health data, significantly improving diagnostic accuracy .

2. Model Comparison

Sisodia and Sisodia (2018) demonstrated that decision trees and random forests achieve over 75% accuracy in diabetes prediction, surpassing logistic regression .

3. Feature Selection

Krishnan et al. (2020) showed that using key health features enhances model performance, reducing diagnostic errors .

4. Real-World Use

Contreras et al. (2020) highlighted the integration of ML in wearable devices, improving real-time diabetes monitoring and early intervention.

Chapter 3

SYSTEM ANALYSIS

3.1 Existing Attendance Systems and Their Drawbacks

1. Traditional Clinical Methods:

Description: These methods include fasting blood glucose tests, oral glucose tolerance tests, and HbA1c tests, which are commonly used to diagnose diabetes.

Drawbacks:

Time-Consuming: Patients often need multiple visits for testing and results, delaying diagnosis.

Subjectivity: Interpretation of results can vary among healthcare providers, leading to inconsistencies.

Limited Predictive Capability: These tests primarily assess current glucose levels and may not effectively predict future risk.

2. Wearable Health Devices:

Description: Devices that monitor glucose levels in real-time and provide data for analysis.

Drawbacks:

Cost and Accessibility: High costs can limit access for many patients, particularly in low-income regions.

User Compliance: Continuous monitoring requires patient adherence, which can be challenging due to lifestyle factors.

Data Overload: Patients may receive excessive data without clear guidance on how to interpret it effectively.

3. Mobile Health Applications:

Description: Apps that allow users to track their glucose levels, diet, and physical activity while providing insights through analytics.

Drawbacks:

Privacy Concerns: Users may be hesitant to share sensitive health information due to privacy risks.

Variable Quality of Apps: The effectiveness of these apps can vary widely based on the underlying algorithms and user interface design.

Limited Clinical Integration: Many apps operate independently of healthcare providers, which can hinder coordinated care.

3.2 Proposed System

The proposed system for diabetes diagnosis using machine learning focuses on developing an intelligent application that leverages various data inputs to accurately predict diabetes risk and facilitate timely intervention. The system utilizes a web-based interface built with Flask, allowing users to input their medical history, demographic information, and lifestyle factors. At its core, the application employs machine learning algorithms, particularly the K-Nearest Neighbors (KNN) classifier, to analyze patient data and classify individuals as diabetic or non-diabetic based on established patterns in the training dataset.

The process begins with data collection, where users can enter relevant health metrics through a user-friendly form. This data is then preprocessed and fed into the KNN model, which has been trained on a comprehensive dataset containing various attributes associated with diabetes. The model's predictions are displayed in real-time, providing users with immediate feedback on their diabetes risk. Additionally, the system incorporates features for continuous monitoring by integrating with wearable devices that track glucose levels and other vital signs, enhancing the predictive capabilities of the model.

To ensure accuracy and reliability, the system includes functionalities for updating and retraining the model as new data becomes available. This adaptability allows the application to remain effective in a dynamic healthcare environment. By combining machine learning with user-centric design, this proposed system aims to revolutionize diabetes diagnosis, making it more accessible and efficient while empowering individuals to take proactive steps in managing their health.

3.3 Overcoming Limitations of Existing Systems

The proposed machine learning-based system addresses several limitations of existing diabetes diagnosis methods:

Speed and Efficiency:

Traditional diagnostic processes can be slow, involving multiple visits for tests and consultations. The proposed system streamlines this process by providing quick risk assessments based on input data, reducing the time needed for diagnosis.

Accuracy and Precision:

Existing methods may suffer from subjective interpretations and human error. By relying on objective data analysis through machine learning, the proposed system enhances diagnostic accuracy, minimizing false positives and negatives.

Scalability:

Traditional diagnostic approaches may not be scalable in resource-limited settings. The proposed system can be deployed in various healthcare environments, including rural clinics and telehealth platforms, making it accessible to a broader population.

Personalization:

Unlike conventional methods that often apply a one-size-fits-all approach, the machine learning model can provide personalized risk assessments tailored to individual patient profiles, leading to more effective prevention strategies.

Integration with Technology:

The proposed system capitalizes on advancements in technology by incorporating data from wearable devices and mobile health applications, enabling continuous monitoring of patients' health metrics.

Chapter 4

OBJECTIVE & METHODOLOGY

4.1 Objectives

The integration of machine learning (ML) into diabetes diagnosis is driven by several key objectives aimed at enhancing patient care and improving clinical outcomes. These objectives can be expanded upon as follows:

Enhance Diagnostic Accuracy

The primary goal of employing machine learning in diabetes diagnosis is to develop predictive models that significantly improve the accuracy of identifying diabetes cases. By analyzing complex patterns in patient data, ML algorithms can reduce the incidence of false positives and negatives. Traditional diagnostic methods often rely on fixed thresholds and subjective interpretations, which can lead to misdiagnoses. In contrast, ML models, such as support vector machines and neural networks, can learn from vast datasets that include various health indicators—such as blood glucose levels, body mass index (BMI), and family history—to make more nuanced predictions. For example, recent studies have shown that advanced models can achieve accuracy rates exceeding 90%, thereby providing healthcare professionals with reliable tools for making informed decisions about patient care.

Facilitate Early Detection

Another critical objective is to enable early identification of individuals at risk of developing diabetes through predictive analytics. Early detection is vital for preventing the progression of diabetes and its associated complications, such as cardiovascular diseases and neuropathy. Machine learning can analyze historical health data alongside real-time monitoring from wearable devices to identify risk factors early on. By flagging at-risk individuals promptly, healthcare providers can implement timely interventions—such as lifestyle modifications or pharmacological treatments—that significantly reduce the likelihood of developing diabetes. This proactive approach not only enhances patient outcomes but also alleviates the long-term burden on healthcare systems.

Personalized Treatment Plans

Utilizing insights gained from machine learning models allows for the creation of tailored treatment plans based on individual risk factors and health profiles. Each patient's response to treatment can vary significantly based on their unique characteristics; thus, a one-size-fits-all approach may not be effective. Machine learning enables the analysis of diverse datasets to identify which interventions are most effective for specific patient

profiles. For instance, algorithms can recommend personalized dietary plans or exercise regimens that align with a patient's metabolic response. This personalized approach enhances patient engagement and adherence to treatment plans, ultimately leading to better management of diabetes.

Streamline Data Processing

The automation of data processing and analysis workflows is another essential objective of integrating machine learning into diabetes diagnosis. Traditional diagnostic processes can be labor-intensive and time-consuming, often requiring multiple appointments for testing and follow-up consultations. By automating these workflows with ML algorithms, the time required for diagnosis can be minimized significantly. This efficiency allows healthcare providers to focus more on direct patient care rather than administrative tasks, improving overall healthcare delivery.

Support Healthcare Professionals

Providing healthcare professionals with reliable decision-support tools is crucial for enhancing their ability to diagnose diabetes accurately and efficiently. Machine learning models serve as adjuncts to clinical judgment, offering evidence-based recommendations that can guide practitioners in their decision-making processes. For example, an ML-driven tool could analyze a patient's data in real-time during a consultation and provide insights into potential diagnoses or treatment options based on established patterns from similar cases. This support not only enhances diagnostic accuracy but also boosts healthcare professionals' confidence in their clinical decisions.

Improve Patient Outcomes

Ultimately, the overarching aim of integrating machine learning into diabetes diagnosis is to contribute to better health outcomes for patients through proactive management of diabetes risk factors and personalized care strategies. By facilitating early detection, enhancing diagnostic accuracy, and tailoring treatment plans to individual needs, ML has the potential to transform how diabetes is managed in clinical settings. Improved patient outcomes are expected not only in terms of glycemic control but also in reducing the incidence of complications associated with poorly managed diabetes.

Utilize Diverse Data Sources

To enhance the predictive capabilities of machine learning models, it is essential to integrate various data sources into the analysis process. These sources may include electronic health records (EHRs), genetic information, lifestyle data (such as diet and physical activity), and real-time monitoring from wearable devices. By synthesizing this diverse array of information, ML algorithms can develop a more comprehensive

understanding of the factors contributing to an individual's risk of developing diabetes. This holistic view enables more accurate predictions and better-informed clinical decisions.

Monitor Model Performance

Finally, implementing mechanisms for continuous monitoring and updating of machine learning models is crucial to ensure their relevance and effectiveness as new data becomes available. The healthcare landscape is constantly evolving; therefore, models must adapt to changes in population health trends, emerging research findings, and advancements in medical knowledge. Regularly updating ML models based on new data ensures that they remain accurate over time and continue to provide valuable insights for diabetes diagnosis and management.

4.2 Methodology

The methodology for diagnosing diabetes using machine learning involves a systematic approach shown in **Fig. 4.1** that encompasses data collection, pre-processing, model training, evaluation, and deployment. Below is a detailed breakdown of the steps involved in this process:

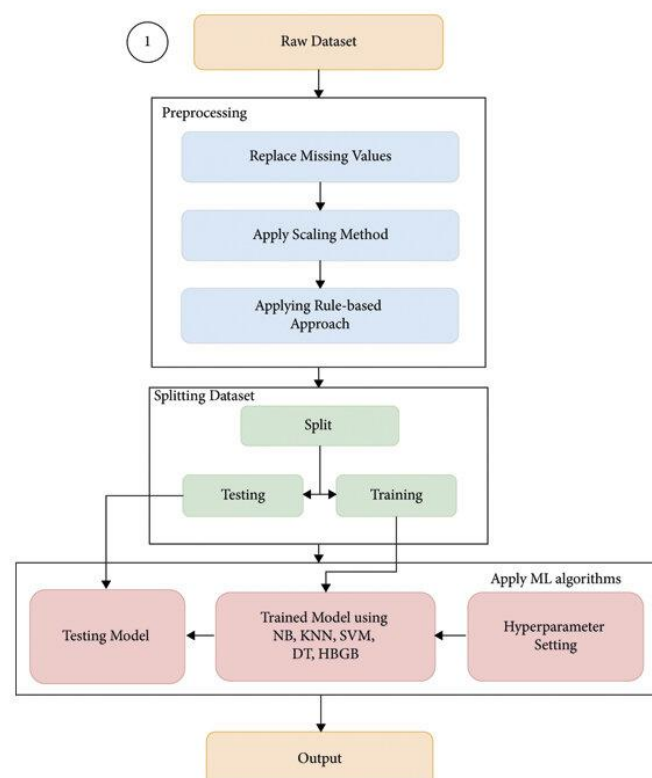


Fig. 4.1 : Basic Methodology

Data Collection:

Gather relevant datasets as shown in **Fig. 4.2** that contain features associated with diabetes diagnosis. Common sources include public health databases such as the Pima

Indians Diabetes Database, which includes attributes like age, BMI, blood pressure, glucose levels, and family history.

Male	68	0	1	former	30.4	6.8	160	1
Female	6	0	0	No Info	16.2	5.8	85	0
Female	39	0	0	No Info	37.6	6.5	155	0
Female	68	0	0	never	25.5	6.5	80	0
Male	44	0	0	never	27.53	6.2	90	0
Male	14	0	0	No Info	21.82	6	90	0
Male	25	0	0	No Info	27.32	6.5	80	0
Female	23	0	0	current	17.22	4	140	0
Female	22	0	0	never	27.32	4.8	145	0
Male	78	0	1	never	25.68	6	145	1
Female	16	0	0	No Info	27.32	6.5	130	0
Male	48	0	0	not curren	37.44	8.8	159	1
Female	62	0	0	former	27.32	4.5	145	0
Female	22	0	0	current	21.32	3.5	85	0
Male	46	1	0	former	27.33	6	140	0
Female	70	0	0	never	38.94	5.8	155	0
Female	41	0	0	No Info	34.77	6.6	140	0

Fig. 4.2 : Example Data Set

Data Preprocessing:

Data Cleaning: Handle missing values and outliers to ensure data quality. Techniques may include imputation for missing values and removing or correcting outliers.

Feature Selection: Identify the most relevant features that contribute to diabetes prediction. This can be done using techniques such as correlation analysis or feature importance from tree-based models.

Data Normalization: Scale the data to ensure that all features contribute equally to the model's performance. Common methods include Min-Max scaling or Standardization.

Splitting the Dataset:

Divide the dataset into training and testing subsets (commonly a 70-30 or 80-20 split). The training set will be used to train the machine learning model, while the testing set will evaluate its performance.

Model Selection and Training:

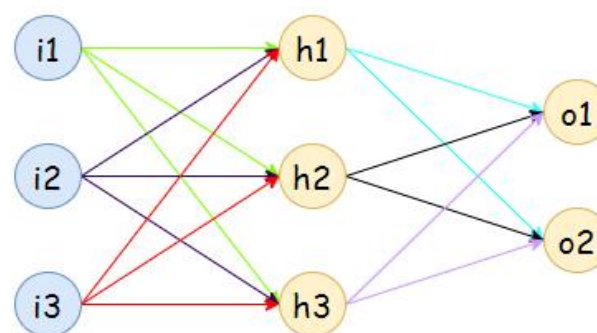


Fig 4.3 : Neural Network

Choose appropriate machine learning algorithms for classification tasks. Common

choices for diabetes diagnosis include:

Logistic Regression

Decision Trees

Random Forest

Support Vector Machines (SVM)

K-Nearest Neighbors (KNN)

Train the selected models using the training dataset. This involves fitting the model to learn patterns in the data that correlate with diabetes diagnosis.

Model Evaluation:

Assess the performance of the trained models using the testing dataset. Key evaluation metrics include:

Accuracy: The proportion of correctly predicted instances.

Precision: The ratio of true positive predictions to the total predicted positives.

Hyperparameter Tuning:

Optimize model performance by adjusting hyperparameters using techniques like Grid Search or Random Search. This step helps in finding the best combination of parameters for improved accuracy.

Deployment:

Once a satisfactory model is achieved, deploy it in a user-friendly application interface (e.g., a web application using Flask) where healthcare professionals can input patient data and receive predictions regarding diabetes risk.

Ensure that the application includes functionality for continuous learning, allowing it to update its model with new data over time.

Monitoring and Maintenance:

Continuously monitor the model's performance in real-world scenarios to identify any degradation in accuracy over time.

Update and retrain the model periodically with new patient data to maintain its relevance and effectiveness in diagnosing diabetes.

Chapter 5

REQUIREMENT ANALYSIS

5.1 Functional Requirements

Data Input:

The system must accept input data from various sources, including electronic health records (EHRs), patient questionnaires, and laboratory results. This multifaceted approach ensures that the model has access to a comprehensive view of each patient's health status. The input data should include relevant features such as age, body mass index (BMI), glucose levels, blood pressure, and family history of diabetes. By integrating diverse data sources, the system can leverage a holistic dataset that captures both clinical and lifestyle factors influencing diabetes risk. This comprehensive data input is crucial for training machine learning models that accurately reflect the complexities of diabetes diagnosis.

Data Preprocessing:

Data preprocessing is a critical step in preparing the input data for analysis. The system should include functionalities to clean and preprocess the data effectively, which involves handling missing values, normalizing numerical features, and encoding categorical variables. Missing values can be addressed through imputation techniques or by removing incomplete records to ensure data integrity. Normalization of numerical features is essential to bring different scales to a common scale, enhancing the model's performance during training. Additionally, categorical variables should be encoded using methods such as one-hot encoding or label encoding to convert them into a format suitable for machine learning algorithms. This preprocessing phase ensures that the data fed into the models is clean, consistent, and ready for analysis.

Model Training:

The system must allow for the selection and training of multiple machine learning algorithms (e.g., Logistic Regression, Decision Trees, Random Forests) to predict diabetes risk based on the input data. By providing options for various algorithms, healthcare professionals can choose the most appropriate model based on their specific needs and the characteristics of their dataset. The training process involves splitting the dataset into training and testing subsets, allowing for validation of model performance. Techniques such as cross-validation can be employed to ensure that the model generalizes

well to unseen data. This flexibility in model selection empowers users to experiment with different algorithms and find the optimal solution for their diagnostic needs.

Model Evaluation:

To ensure that the machine learning models are performing effectively, the system should provide mechanisms to evaluate model performance using metrics such as accuracy, precision, recall, F1 score, and ROC-AUC (Receiver Operating Characteristic - Area Under Curve). These metrics offer insights into how well the models are predicting diabetes risk and help identify areas for improvement. For instance, accuracy measures the overall correctness of predictions, while precision and recall provide insights into false positives and false negatives respectively. The F1 score offers a balance between precision and recall, making it particularly useful in scenarios with class imbalance—common in medical datasets where healthy individuals often outnumber those with diabetes. ROC-AUC provides a comprehensive view of model performance across different thresholds.

User Interface:

A user-friendly web-based or desktop interface must be developed to allow healthcare professionals to input patient data easily and receive predictions regarding diabetes risk. The interface should prioritize usability by featuring intuitive navigation elements, clear instructions, and accessible visualization tools that present predictions in an understandable manner. For example, interactive dashboards could display patient risk profiles alongside recommended interventions based on ML predictions. Ensuring that the interface accommodates users with varying levels of technical expertise is crucial for widespread adoption among healthcare providers.

Reporting:

The system should generate comprehensive reports summarizing the predictions made by the machine learning models along with evaluations of model performance for review by healthcare providers. These reports can serve as valuable documentation for clinical decision-making processes, offering insights into patient risk levels and suggested management strategies. Additionally, visualizations such as graphs or charts can enhance understanding by illustrating trends or correlations within patient data. Providing clear reporting mechanisms not only aids healthcare professionals in their decision-making but also facilitates communication with patients regarding their health status.

5.2 Non-Functional Requirements

Performance:

The system should provide predictions in real-time or near real-time to facilitate timely decision-making in clinical settings. This capability is crucial for healthcare professionals who need immediate insights to make informed decisions about patient care. Real-time predictions allow clinicians to respond promptly to changes in a patient's condition, such as adjusting treatment plans based on current glucose levels or other health indicators. For instance, a real-time monitoring hybrid deep learning model can analyze incoming data from wearable devices and provide alerts or recommendations, thereby enabling proactive interventions that can significantly improve patient outcomes and reduce the risk of complications associated with diabetes.

Scalability:

Scalability is essential for ensuring that the system can handle increasing amounts of data and user requests without significant degradation in performance. As more patients are monitored and more data is collected over time, the system must be capable of processing this information efficiently. This involves designing the architecture to support horizontal scaling, where additional resources can be added to manage increased loads. Utilizing cloud-based solutions can enhance scalability, allowing for dynamic allocation of computing resources based on demand. Ensuring that the system remains responsive and efficient as it scales is vital for maintaining user satisfaction and operational effectiveness.

Usability:

The user interface should be intuitive and easy to navigate for healthcare professionals with varying levels of technical expertise. A well-designed UI is critical for ensuring that users can quickly access the information they need without extensive training. Features such as clear navigation menus, straightforward data entry forms, and easily interpretable visualizations contribute to a positive user experience. Additionally, incorporating user feedback during the design process can help identify pain points and areas for improvement, ultimately leading to a more effective tool for clinicians.

Security:

Patient data must be stored securely to comply with regulations such as HIPAA (Health Insurance Portability and Accountability Act). This includes implementing encryption protocols for data at rest and in transit, ensuring that sensitive information is protected

from unauthorized access. Access control measures should also be established to limit data access to authorized personnel only, thereby safeguarding patient confidentiality. Regular security audits and compliance checks are necessary to maintain adherence to regulatory standards and protect against potential data breaches.

Reliability:

The system should be robust and capable of handling errors gracefully, ensuring that predictions can still be made even in the event of minor issues. This reliability is crucial in clinical settings where accurate information is needed consistently. Implementing failover mechanisms, such as backup servers or redundant systems, can help maintain functionality during outages or technical difficulties. Additionally, the system should include error logging and reporting features to facilitate troubleshooting and continuous improvement.

5.3 Technical Requirements

Programming Language:

Python is recommended for its extensive libraries for data analysis and machine learning. For the front-end we have used HTML the framework and CSS for Styling the front-end.

Libraries and Frameworks:**1. Pandas: For Data Manipulation and Preprocessing**

Pandas is a powerful library designed for data manipulation and analysis, providing flexible data structures like Series and DataFrames that facilitate the handling of large datasets. It excels in data cleaning, which involves managing missing values through functions like `dropna()` to remove incomplete entries or `fillna()` to substitute them with appropriate values. Additionally, Pandas enables users to identify and eliminate duplicates using `drop_duplicates()`, ensuring data integrity. The library also supports data transformation techniques such as normalization and feature engineering, where numerical data can be scaled using `StandardScaler` from Scikit-learn for better model performance. Moreover, Pandas offers intuitive methods for filtering and selecting data based on specific criteria, enhancing the efficiency of data preprocessing workflows.

2. NumPy: For Numerical Operations

NumPy is a fundamental package for numerical computing in Python, providing support for large multi-dimensional arrays and matrices along with a collection of mathematical functions to operate on these arrays. Its core feature is the N-dimensional array object,

which allows for efficient storage and manipulation of numerical data. NumPy's performance is optimized for large datasets, making it suitable for scientific computing tasks. It includes functions for mathematical operations such as linear algebra, statistical analysis, and Fourier transforms, which are crucial in preparing data for machine learning models. Furthermore, NumPy seamlessly integrates with other libraries like Pandas and Scikit-learn, enhancing its utility in data analysis pipelines.

3. Scikit-learn: For Implementing Machine Learning Algorithms

Scikit-learn is a robust library that provides simple and efficient tools for predictive data analysis. It encompasses a wide range of supervised and unsupervised learning algorithms including regression, classification, clustering, and dimensionality reduction techniques. Scikit-learn's user-friendly API allows for easy integration with other libraries like NumPy and Pandas, facilitating seamless data preprocessing steps such as feature scaling with StandardScaler or MinMaxScaler. Additionally, it includes utilities for model evaluation through cross-validation techniques and metrics that help assess model performance. The library also supports hyperparameter tuning through GridSearchCV or RandomizedSearchCV to optimize model accuracy.

4. Flask For Developing the Web Application Interface

Flask and Django are popular web frameworks in Python that enable developers to build robust web applications efficiently. Flask is a lightweight framework that provides the essentials for web development while allowing flexibility in design choices. It is particularly suited for small to medium-sized applications where simplicity is key. On the other hand, Django is a high-level framework that follows the "batteries-included" philosophy, offering built-in features such as authentication, ORM (Object-Relational Mapping), and an admin panel out of the box. Both frameworks can be utilized to create interactive web interfaces that allow users to input data and visualize results generated by machine learning models developed using libraries like Scikit-learn.

5. Matplotlib: For Data Visualization and Reporting

Matplotlib is a widely used plotting library that provides extensive capabilities for creating static, animated, and interactive visualizations in Python. It offers fine-grained control over plot elements such as axes, labels, and legends. Seaborn builds on Matplotlib by providing a high-level interface for drawing attractive statistical graphics with ease. It simplifies complex visualizations like heatmaps or violin plots while integrating seamlessly with Pandas DataFrames. Both libraries are essential for visualizing data

distributions, relationships between variables, and results from machine learning models. Effective visualization aids in understanding patterns within the data and communicating insights clearly in reports.

Development Tools:

An Integrated Development Environment (IDE) such as PyCharm or Jupyter Notebook for coding and testing. Version control using Git for collaborative development.

5.4 Hardware Requirements

We need 1 machine with following minimal requirements

CPU	: Intel 2.1 GHZ
GPU	: RTX (for high accuracy in low time)
Memory	: 6GB
Disk	: 1GB

Chapter 6

IMPLEMENTATION

The implementation of diabetes diagnosis through machine learning (ML) involves several key steps as shown in **Fig. 6.1**, including data collection, preprocessing, model selection, training, and evaluation. Below is a detailed explanation of the process based on recent studies and methodologies.

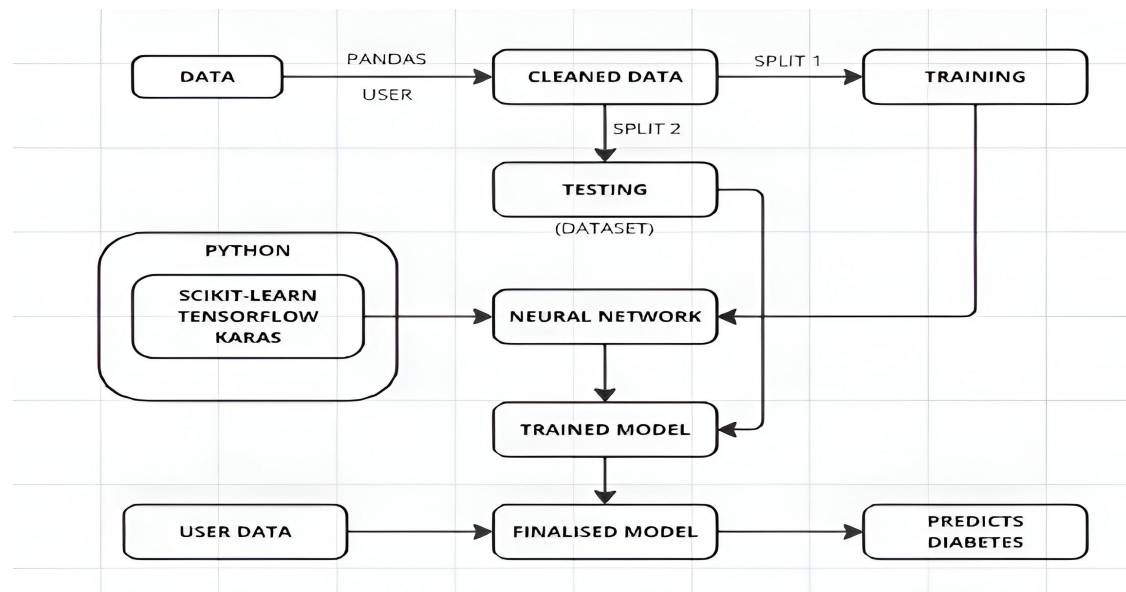


Fig. 6.1 : Implimentation Diagram

Data Collection:

Data collection is foundational in developing a machine learning model for diabetes diagnosis. Datasets typically include various health parameters such as:

- **Demographic Information:** Age, gender, ethnicity.
- **Clinical Measurements:** Fasting blood glucose levels, body mass index (BMI), blood pressure.
- **Symptoms:** Polyuria, polydipsia, weight loss.
- **Medical History:** Family history of diabetes and other comorbidities.

For instance, studies have utilized datasets like the PIMA Indian Diabetes Database and the National Health and Nutrition Examination Survey (NHANES) to gather relevant patient data.

Data Preprocessing

Preprocessing is critical to ensure the quality and usability of the data. Common preprocessing steps include:

- **Handling Missing Values:** Techniques such as imputation or removal of incomplete records are employed to maintain dataset integrity.
- **Normalization and Standardization:** Scaling features to a uniform range helps improve model performance by ensuring that no single feature dominates due to its scale.
- **Feature Selection:** Identifying the most relevant features that contribute to diabetes prediction can enhance model accuracy. Techniques like recursive feature elimination or using algorithms like Random Forest for feature importance are common.

Model Selection:

Various machine learning algorithms can be applied to predict diabetes. The choice of model often depends on the characteristics of the dataset and the specific requirements of the diagnosis task. Commonly used models include:

- **Decision Trees (DT):** Useful for their interpretability and ability to handle both numerical and categorical data.
- **Random Forest (RF):** An ensemble method that improves prediction accuracy by combining multiple decision trees.
- **Support Vector Machines (SVM):** Effective in high-dimensional spaces and for datasets where classes are not linearly separable.
- **Neural Networks (NN):** Particularly beneficial for capturing complex relationships in large datasets.

Training the Model

Once the model is selected, it undergoes training using a portion of the dataset. This involves:

- **Splitting Data:** The dataset is typically divided into training and testing subsets (e.g., 80/20 split).
- **Hyperparameter Tuning:** Techniques such as grid search or Bayesian optimization are employed to find the best hyperparameters that maximize model performance.
- **Cross-validation:** Implementing k-fold cross-validation helps assess how the results will generalize to an independent dataset.

Evaluation Metrics

Evaluating the performance of the ML models is crucial. Common metrics include:

- **Accuracy:** The proportion of true results among the total cases examined.
- **Precision:** The ratio of correctly predicted positive observations to the total predicted positives.

Results and Interpretations

Recent studies have shown that ML models can achieve high accuracy in predicting diabetes. For example, one study reported an average accuracy of 98.87% using an ensemble approach that combined multiple classifiers. Another study highlighted that Random Forest outperformed other classifiers in specific datasets.

Challenges

While ML shows promise in diabetes diagnosis, challenges remain:

- **Data Imbalance:** Many datasets may have more non-diabetic than diabetic cases, leading to biased models. Techniques such as oversampling or undersampling can be used to mitigate this issue 1.
- **Generalizability:** Models trained on specific populations may not perform well across diverse demographic groups. Future research should focus on developing more generalized models through diverse dataset integration.

Chapter 7

CONCLUSION

The implementation of machine learning (ML) for diabetes diagnosis represents a significant advancement in healthcare, offering the potential for improved accuracy, efficiency, and early detection of this prevalent condition. As diabetes, particularly Type 2 diabetes mellitus (T2DM), continues to rise globally, leveraging advanced technologies like ML can play a crucial role in combating this public health challenge. By utilizing diverse datasets that encompass a wide range of demographic and clinical variables, researchers can develop models that are not only accurate but also reflective of real-world populations.

The success of ML in diabetes diagnosis hinges on the application of various algorithms—such as decision trees, random forests, support vector machines (SVM), and neural networks—which enable the identification of complex patterns and relationships within the data. These algorithms allow for the extraction of meaningful insights from large datasets, facilitating the recognition of risk factors associated with diabetes. For instance, studies have shown that deep learning approaches, particularly those utilizing convolutional neural networks (CNNs), can effectively analyze medical imaging data to detect diabetic retinopathy, a common complication of diabetes, thereby improving early diagnosis and intervention strategies².

This adaptability allows healthcare professionals to tailor diagnostic approaches to individual patient profiles, enhancing personalized medicine. Furthermore, the importance of robust evaluation metrics cannot be overstated; employing techniques such as cross-validation and hyperparameter tuning ensures that models are not only accurate but also generalizable across different patient populations. Metrics like precision, recall, and F1 score provide a comprehensive view of model performance, particularly in scenarios where class imbalance exists. Addressing data imbalance is crucial since many datasets exhibit a predominance of non-diabetic cases over diabetic ones, potentially skewing model predictions. Advanced techniques such as oversampling minority classes or undersampling majority classes are being explored to create more balanced datasets that enhance model performance.

Despite the promise of ML in diabetes diagnosis, challenges such as data imbalance and generalizability must be addressed. Future research should focus on developing strategies to create more balanced datasets and ensuring that models are validated across diverse

populations to enhance their applicability in clinical settings. Additionally, the integration of ML tools into clinical workflows can facilitate timely interventions and improve patient outcomes. For example, predictive models can alert healthcare providers about patients at high risk for developing diabetes or its complications, enabling proactive management strategies.

Advancements in explainable AI (XAI) will be essential to gain trust among healthcare professionals and patients alike, ensuring that these models are effective and transparent. XAI techniques help demystify the decision-making processes of ML algorithms, allowing clinicians to understand how specific inputs influence predictions. This transparency is vital for integrating ML into routine clinical practice and ensuring adherence to ethical standards.

In conclusion, the integration of machine learning into diabetes diagnosis is a transformative step toward more proactive healthcare management. As technology continues to evolve, the potential for ML to enhance diagnostic accuracy and facilitate early intervention will be pivotal in reducing the burden of diabetes worldwide. By embracing these innovations while addressing inherent challenges such as data quality and model interpretability, we can pave the way for a healthier future where diabetes is detected earlier and managed more effectively. The collaboration between technology and medicine holds great promise for improving patient care and outcomes in the fight against diabetes.

Chapter 8

FUTURE ENHANCEMENT

The future of diabetes diagnosis is poised for significant advancements through the integration of advanced machine learning (ML) and deep learning (DL) algorithms, utilization of big data, and the development of innovative tools and methodologies.

Integration of Advanced Algorithms:

Future developments in diabetes diagnosis should focus on integrating advanced machine learning and deep learning algorithms. For instance, combining convolutional neural networks (CNNs) with long short-term memory (LSTM) networks can enhance feature extraction and improve diagnostic accuracy. CNNs are particularly effective in processing spatial data, such as images, while LSTMs excel in handling sequential data, making this hybrid approach suitable for analyzing complex datasets that include both imaging and temporal health records. This integration has shown promising results, achieving accuracy rates as high as 95.7% in diagnosing diabetes through intricate data analysis. By harnessing the strengths of both algorithms, healthcare providers can develop more robust predictive models that capture a wider array of risk factors.

Utilization of Big Data

Leveraging big data analytics will be crucial for enhancing the robustness and generalizability of diabetes diagnostic models. By incorporating vast amounts of health data from diverse populations, researchers can develop models that are more representative and capable of performing well across different demographic groups. This approach addresses current limitations related to model performance on unseen data by ensuring that the models are trained on comprehensive datasets that reflect various genetic, environmental, and lifestyle factors influencing diabetes risk. The ability to analyze large-scale datasets also enables the identification of rare patterns and correlations that may be overlooked in smaller studies.

Automated Risk Assessment Tools:

The development of automated risk assessment tools that utilize real-time data can significantly enhance early detection capabilities. These tools can analyze various parameters, including lifestyle factors, genetic predispositions, and clinical measurements, to provide personalized risk assessments for diabetes onset. For example, integrating wearable technology with machine learning algorithms can allow continuous monitoring of vital signs and lifestyle choices, enabling healthcare providers to intervene proactively

when risk levels rise. Such proactive measures will facilitate timely interventions and management strategies tailored to individual patient needs.

Enhanced Feature Engineering Techniques:

Future research should emphasize sophisticated feature engineering techniques to extract relevant predictors from complex datasets. This includes using advanced statistical methods and domain knowledge to identify key variables that influence diabetes risk, thus improving model accuracy and interpretability. Effective feature engineering can enhance the predictive power of ML models by ensuring that they focus on the most informative aspects of the data. Techniques such as dimensionality reduction, interaction terms, and polynomial features can be employed to refine the input data further.

AI-Driven Diabetic Retinopathy Screening:

Integrating AI technologies in diabetic retinopathy screening alongside diabetes diagnosis can create a comprehensive care model. By analyzing retinal images with high precision using deep learning algorithms, AI can facilitate earlier detection of complications associated with diabetes, ensuring timely treatment and better patient outcomes. This dual approach not only addresses the immediate needs of diabetes management but also incorporates preventive care strategies by identifying complications before they progress to more severe stages.

Personalized Medicine Approaches:

The future of diabetes diagnosis will likely involve personalized medicine, where ML models are tailored to individual patient profiles based on genetic, environmental, and lifestyle factors. This customization can lead to more accurate predictions and targeted treatment plans that address the unique needs of each patient. For instance, machine learning could help identify which patients are likely to respond best to specific medications or lifestyle interventions based on their genetic makeup or previous health outcomes.

Collaboration Between Disciplines:

Enhancing diabetes diagnosis through ML will require collaboration between healthcare professionals, data scientists, and AI experts. Such interdisciplinary partnerships can foster the development of innovative solutions that combine clinical insights with advanced technological capabilities. Collaborative efforts can lead to the creation of user-friendly tools that integrate seamlessly into clinical workflows while providing actionable insights based on robust data analysis.

Ethical Considerations and Regulatory Compliance:

As AI technologies advance in healthcare applications, addressing ethical considerations

and ensuring regulatory compliance will be paramount. Developing transparent algorithms that prioritize patient privacy and data security will be essential in gaining trust among users and stakeholders in the healthcare system. Moreover, establishing clear guidelines for the ethical use of AI in clinical settings will help mitigate potential biases in model predictions and ensure equitable access to care.

Real-Time Monitoring Systems:

Implementing real-time monitoring systems powered by AI can provide continuous assessment of patients' metabolic states. Wearable devices that collect biometric data—such as glucose levels, heart rate variability, and physical activity—can feed into ML algorithms to predict potential diabetes-related complications before they manifest clinically. This capability allows for immediate intervention when necessary, significantly improving patient outcomes through proactive management.

Longitudinal Studies for Model Validation:

Conducting longitudinal studies will be critical for validating the effectiveness of ML models in predicting diabetes onset and progression over time. Such studies can provide insights into the long-term performance of these models and their impact on patient outcomes. By tracking patients over extended periods, researchers can assess how well predictive models adapt to changes in individual health status and refine them accordingly based on real-world evidence.

REFERENCES

Research Papers:

- [1] R Shouval, O Bondi, H Mishan, A Shimoni, R Unger and A Nagler, “Application of machine learning algorithms for clinical predictive modeling: a data-mining approach in SCT”, 2014.
- [2] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects”, 2015.

Books:

- 1. Machine Learning for Healthcare: Methods and Applications by Sharmila M. K., S. R. S. Prakash.
- 2. Data Science for Medical Decision Making by David A. Cohn

Websites:

- 1. IEEE Xplore
 - <https://ieeexplore.ieee.org>
 - for Research papers and journals.
- 2. Google Scholar
 - <https://scholar.google.com>
 - for filtering and accesing of Research papers and journals.
- 3. GitHub
 - <https://github.com>
 - Repositories for open source ML projects and data sources.

APPENDIX – A

CODE:

MODEL TRAINING:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import pylab
import scipy.stats as stat
from scipy.stats import ttest_ind, chi2_contingency
import statsmodels.api as sm
from statsmodels.formula.api import ols
from category_encoders import TargetEncoder
from sklearn.preprocessing import LabelEncoder, StandardScaler, MinMaxScaler,
RobustScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import GridSearchCV
import os

os.chdir("C:\\Users\\Nilesh\\Documents\\GitHub\\01 Diabetes Prediction")
df = pd.read_csv("Diabetes-dataset_FT.csv")

X = df.iloc[:, :-1].values

Y = df.iloc[:, -1].values

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, train_size=0.66, shuffle=True)

from sklearn import metrics
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(random_state=16)
```

```

lr.fit(X_train, Y_train)
Y_pred = lr.predict(X_test)
confusion_matrix = metrics.confusion_matrix(Y_test, Y_pred)
print("-"*70)
print("Report")
print("-"*70)
print("Confusion Matrix:")
print(str(confusion_matrix))
target_names = ['without diabetes', 'with diabetes']
acc=(confusion_matrix[0][0] +
confusion_matrix[1][1])/(confusion_matrix[0][0]+confusion_matrix[0][1]+confusion_ma
trix[1][0]+confusion_matrix[1][1])
print("Accuracy by confusion matrix: "+str(acc))
print("\n")
print(classification_report(Y_test, Y_pred, target_names=target_names))
print("-"*70)

```

rn.metrics import classification_report

HYPER-PARAMETER TRAINING:

```

from sklearn.model_selection import RepeatedStratifiedKFold
model = LogisticRegression()
solvers = ['newton-cg', 'lbfgs', 'liblinear']
penalty = ['l2']
c_values = [100, 10, 1.0, 0.1, 0.01]
# define grid search
grid = dict(solver=solvers,penalty=penalty,C=c_values)
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
grid_search = GridSearchCV(estimator=model, param_grid=grid, n_jobs=-1, cv=cv,
scoring='accuracy',error_score=0)
grid_result = grid_search.fit(X, Y)
# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']

```

for mean, stdev, param in zip(means, stds, params):

print("%f (%f) with: %r" % (mean, stdev, param))

```
lr = LogisticRegression(random_state=16,C=0.1,penalty='l2',
    dual=False,
    tol=0.0001,
    fit_intercept=True,
    intercept_scaling=1,
    class_weight=None,
    solver='liblinear',
    max_iter=100,
    multi_class='auto',
    verbose=0,
    warm_start=False,
    n_jobs=None,
    l1_ratio=None,)
```

```
lr.fit(X_train, Y_train)
```

```
Y_pred = lr.predict(X)
```

```
confusion_matrix = metrics.confusion_matrix(Y, Y_pred)
```

```
print("-"*70)
```

```
print("Report")
```

```
print("-"*70)
```

```
print("Confusion Matrix:")
```

```
print(str(confusion_matrix))
```

```
target_names = ['without diabetes', 'with diabetes']
```

```
acc=(confusion_matrix[0][0] +
```

```
confusion_matrix[1][1])/(confusion_matrix[0][0]+confusion_matrix[0][1]+confusion_ma
trix[1][0]+confusion_matrix[1][1])
```

```
print("Accuracy by confusion matrix: "+str(acc))
```

```
print("\n")
```

```
print(classification_report(Y, Y_pred, target_names=target_names))
```

```
print("-"*70)
```

```
model.save()
```

APP.PY:

```
from flask import Flask, render_template, request
import pickle
import numpy as np
import pandas as pd

app = Flask(__name__)

# te = pickle.load(open('Address_te.pkl','rb'))
# scaler=pickle.load(open('scaler.pkl','rb'))
# lr=pickle.load(open('lr.pkl','rb'))

@app.route('/')
def hello_world():
    return render_template("index.html")

@app.route('/result', methods=['POST'])
def result():

    Age = request.form.get("Age")
    Age = int(Age)

    Glucose = request.form.get("Glucose")
    Glucose = int(Glucose)

    BloodPressure = request.form.get("BloodPressure")
    BloodPressure = int(BloodPressure)

    Insulin = request.form.get("Insulin")
    Insulin = int(Insulin)

    BMIs = request.form.get("BMI")
```



```
BMI = float(BMI)

SkinThickness = request.form.get("SkinThickness")
SkinThickness = int(SkinThickness)

DiabetesPedigreeFunction = request.form.get("DiabetesPedigreeFunction")
DiabetesPedigreeFunction = float(DiabetesPedigreeFunction)

temp_arr=list()
temp_arr=temp_arr+[Glucose, BloodPressure, SkinThickness, Insulin, BMI,
DiabetesPedigreeFunction, Age]

data=np.array([temp_arr])
temp_sc=scaler.transform(data)
pred=lr.predict(temp_sc)[0]
pred=round(pred, 2)
print(temp_arr)
print(temp_sc)
print(pred)

if pred==0:
    res="does not indicate"
if pred==1:
    res="indicates"

return render_template('result.html', prediction=res)

if __name__ == '__main__':
    app.run(debug=True)
```

FRONTEND:**INDEX.HTML:**

```
from flask import Flask, render_template, request
import pickle
import numpy as np
```

```
import pandas as pd
```

```
app = Flask(__name__)
```

```
# te = pickle.load(open('Address_te.pkl','rb'))
```

```
scaler=pickle.load(open('scaler.pkl','rb'))
```

```
lr=pickle.load(open('lr.pkl','rb'))
```

```
@app.route('/')
```

```
def hello_world():
```

```
    return render_template("index.html")
```

```
@app.route('/result', methods=['POST'])
```

```
def result():
```

```
    Age = request.form.get("Age")
```

```
    Age = int(Age)
```

```
    Glucose = request.form.get("Glucose")
```

```
    Glucose = int(Glucose)
```

```
    BloodPressure = request.form.get("BloodPressure")
```

```
    BloodPressure = int(BloodPressure)
```

```
    Insulin = request.form.get("Insulin")
```

```
    Insulin = int(Insulin)
```

```
    BMIs = request.form.get("BMI")
```

```
    BMIs = float(BMIs)
```

```
    SkinThickness = request.form.get("SkinThickness")
```

```
    SkinThickness = int(SkinThickness)
```

```

DiabetesPedigreeFunction = request.form.get("DiabetesPedigreeFunction")
DiabetesPedigreeFunction = float(DiabetesPedigreeFunction)

temp_arr=list()
temp_arr=temp_arr+[Glucose, BloodPressure, SkinThickness, Insulin, BMIs,
DiabetesPedigreeFunction, Age]

data=np.array([temp_arr])
temp_sc=scaler.transform(data)
pred=lr.predict(temp_sc)[0]
pred=round(pred, 2)
print(temp_arr)
print(temp_sc)
print(pred)

if pred==0:
    res="does not indicate"
if pred==1:
    res="indicates"

return render_template('result.html', prediction=res)

if __name__ == '__main__':
    app.run(debug=True)

```

RESULTS.HTML:

```

<!DOCTYPE html>
<html>
<head>
    <title>Prediction</title>
    <link rel="stylesheet" type="text/css"
href="{{ url_for('static',filename='css/style.css')}}">
    <link href="https://fonts.googleapis.com/css?family=Quicksand&display=swap"

```

```

rel="stylesheet">
    <!-- <script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script> -->
    <meta name="viewport" content="width=device-width,initial-scale=1.0,minimum-
scale=1.0,maximum-scale=1.0">
</head>
<body>
    <div class="container">
        <div class="contact-box">
            <div class="left"></div>
            <div class="right">
                <h1 style="padding-top: 165px; padding-bottom: 130px;"><b>Your report
{{prediction}} the presence of diabetes.</b></h1>
                <button class="btn" onclick="location.href =
'http://127.0.0.1:5000/'; "><b>Predict Again</b></button>
                <div style="padding-bottom: 0px;"></div>
            </div>
        </div>
    </div>
</body>
</html>

```

STYLE.CSS:

```

*{
    padding: 0;
    margin: 0;
    box-sizing: border-box;
    font-family: 'Quicksand', sans-serif;
}

body{
    height: 100vh;
    width: 100%;
}

.container{

```

```
position: relative;
width: 100%;
height: 100%;
display: flex;
justify-content: center;
align-items: center;
padding: 20px 100px;
}

.container:after{
  content: "";
  position: absolute;
  width: 100%;
  height: 100%;
  left: 0;
  top: 0;
  background: url("../img/bg.png") no-repeat center;
  background-size: cover;
  filter: blur(50px);
  z-index: -1;
}

.contact-box{
  max-width: 850px;
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  justify-content: center;
  align-items: center;
  text-align: center;
  background-color: #fff;
  box-shadow: 0px 0px 19px 5px rgba(0,0,0,0.19);
}

.left{
  background: url("../img/bg.png") no-repeat center;
  background-size: cover;
  height: 100%;
```

```
}  
.right{  
  padding: 25px 40px;  
}  
  
h2{  
  position: relative;  
  padding: 0 0 10px;  
  margin-bottom: 10px;  
}  
  
h2:after{  
  content: "";  
  position: absolute;  
  left: 50%;  
  bottom: 0;  
  transform: translateX(-50%);  
  height: 4px;  
  width: 50px;  
  border-radius: 2px;  
  background-color: #be1558;  
}  
  
.field{  
  width: 100%;  
  border: 2px solid rgba(0, 0, 0, 0);  
  outline: none;  
  background-color: rgba(230, 230, 230, 0.6);  
  padding: 0.5rem 1rem;  
  font-size: 1.1rem;  
  margin-bottom: 22px;  
  transition: .3s;  
}  
.field:hover{  
  background-color: rgba(0, 0, 0, 0.1);
```

```
}
```

```
textarea{  
  min-height: 150px;  
}
```

```
.btn{  
  width: 100%;  
  padding: 0.5rem 1rem;  
  background-color: #be1558;  
  color: #fff;  
  font-size: 1.1rem;  
  border: none;  
  outline: none;  
  cursor: pointer;  
  transition: .3s;  
}
```

```
.btn:hover{  
  background-color: #be1558;  
}
```

```
.field:focus{  
  border: 2px solid #3B0208;  
  background-color: #fff;  
}
```

```
@media screen and (max-width: 880px){  
  .contact-box{  
    grid-template-columns: 1fr;  
  }  
  .left{  
    height: 200px;  
  }  
}
```

APPENDIX – B

SNAPSHOTS:

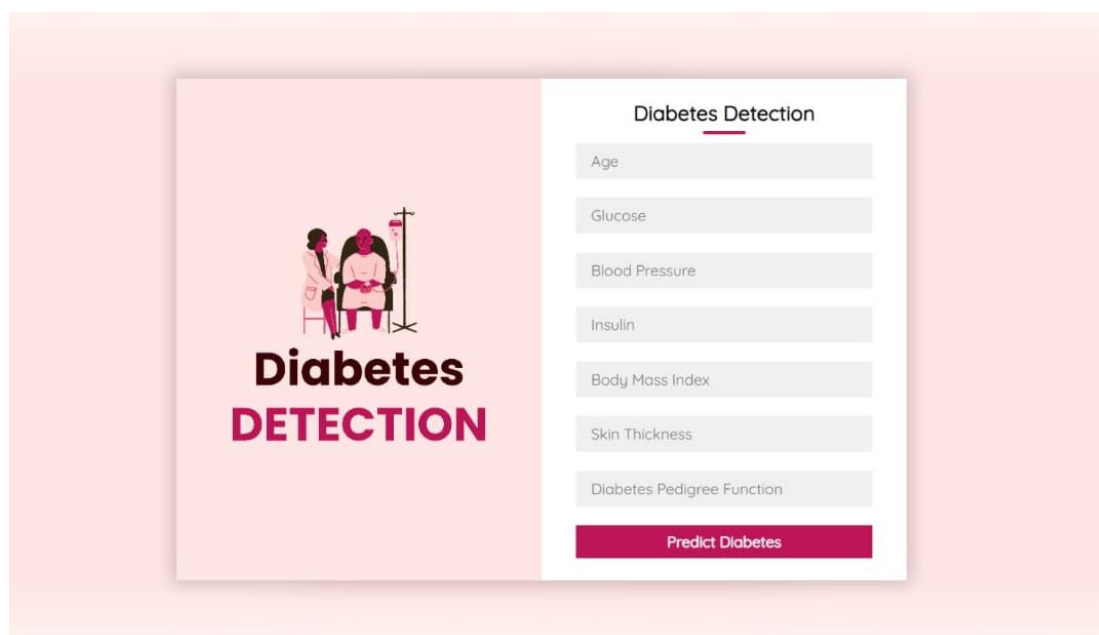
User Interface:

The user interface (UI) of our project has been meticulously designed to provide an intuitive and engaging experience for users. At its core, we have extensively utilized HTML (HyperText Markup Language) to create the structural framework of the web page. HTML serves as the backbone of our UI, allowing us to define various elements such as headings, paragraphs, forms, buttons, and navigation menus.

To enhance the visual aesthetics and overall user experience, we have employed CSS (Cascading Style Sheets) for styling our HTML elements. CSS enables us to control the layout, colors, fonts, and spacing of the webpage, creating a cohesive and attractive design.

Here is the User view of the project, where the user has to enter the following data

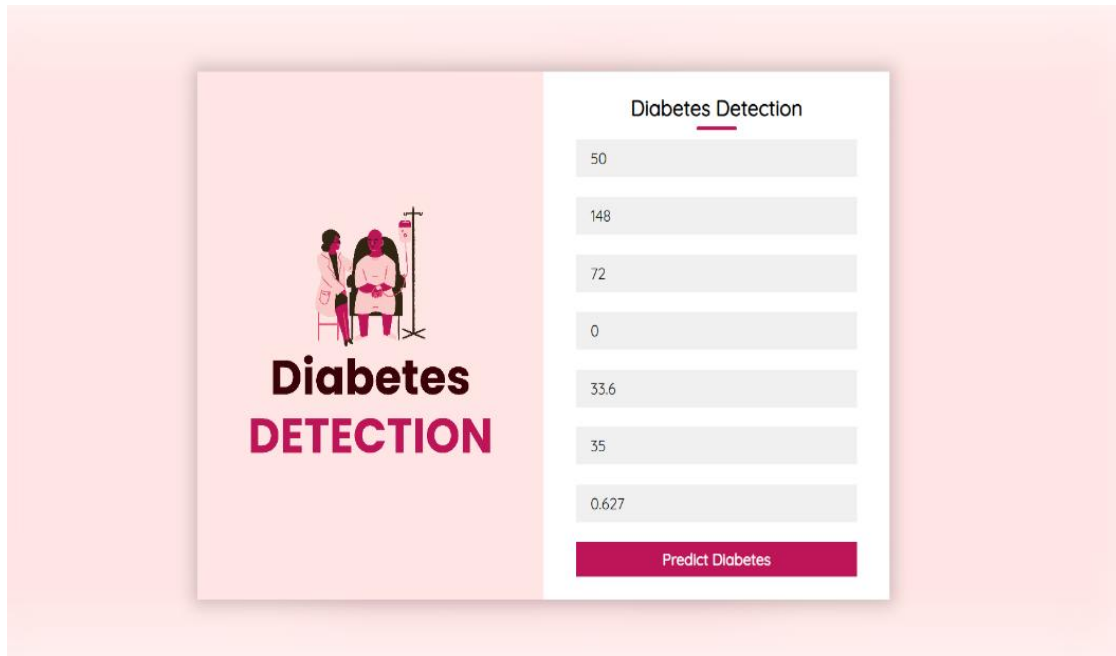
- Age
- Glucose level
- Blood pressure
- Insulin level
- Body Mass Index
- Skin Thickness
- Diabetic Pedegree function



The screenshot displays the user interface of the 'Diabetes Detection' application. On the left, there is a pink square graphic containing an illustration of a doctor in a white coat and a patient in a pink dress, with the text 'Diabetes DETECTION' in bold black and pink letters. To the right of this graphic is a white rectangular form titled 'Diabetes Detection' in black text. The form contains seven input fields, each with a light gray border and placeholder text: 'Age', 'Glucose', 'Blood Pressure', 'Insulin', 'Body Mass Index', 'Skin Thickness', and 'Diabetes Pedegree Function'. Below these fields is a prominent pink button with the text 'Predict Diabetes' in white.

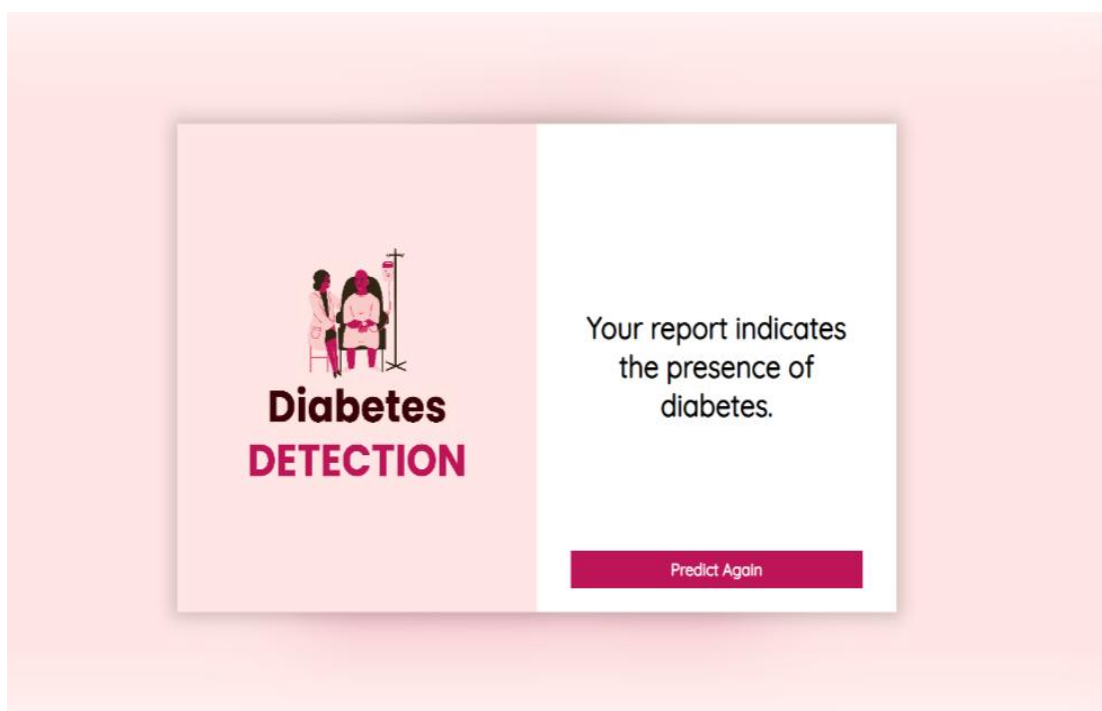
Positive result:

In the below snapshot, we are entering the data of a diabetic person,



The screenshot shows a web application titled "Diabetes Detection". On the left, there is an illustration of a doctor examining a patient, with the text "Diabetes DETECTION" below it. On the right, there is a form with the title "Diabetes Detection" and a list of input fields containing the following values: 50, 148, 72, 0, 33.6, 35, and 0.627. At the bottom of the form is a red button labeled "Predict Diabetes".

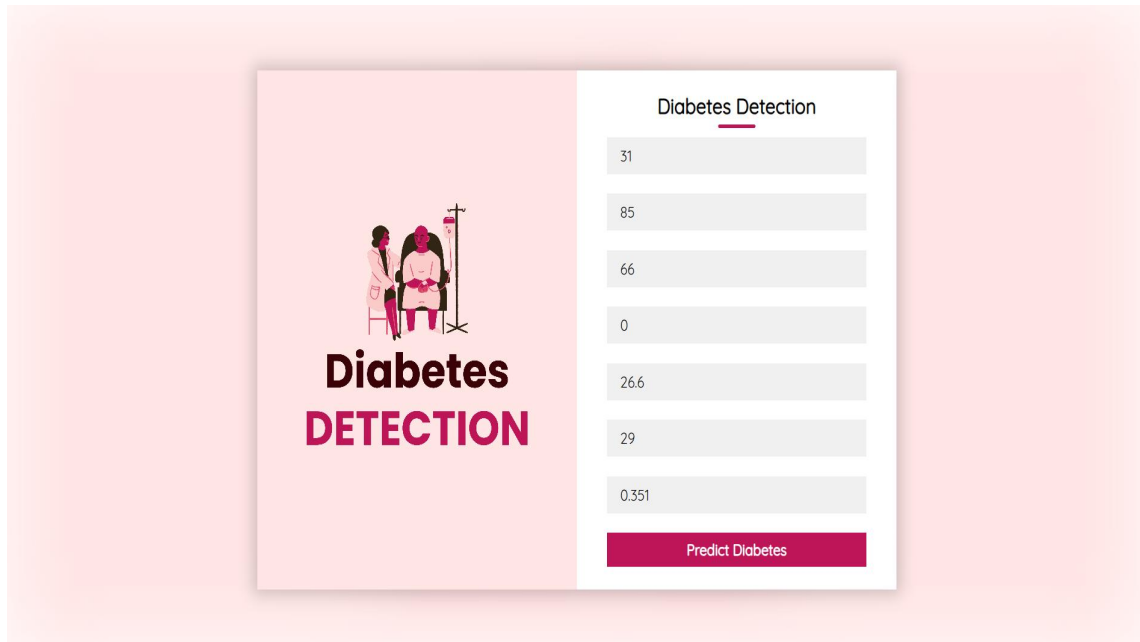
after entering all the data we click the “Predict Diabetes” button, which will show you the result that the model indicates the presence of diabetes



The screenshot shows the result screen of the "Diabetes Detection" application. On the left, there is the same illustration of a doctor examining a patient, with the text "Diabetes DETECTION" below it. On the right, there is a white box with the text "Your report indicates the presence of diabetes." and a red button labeled "Predict Again" at the bottom.

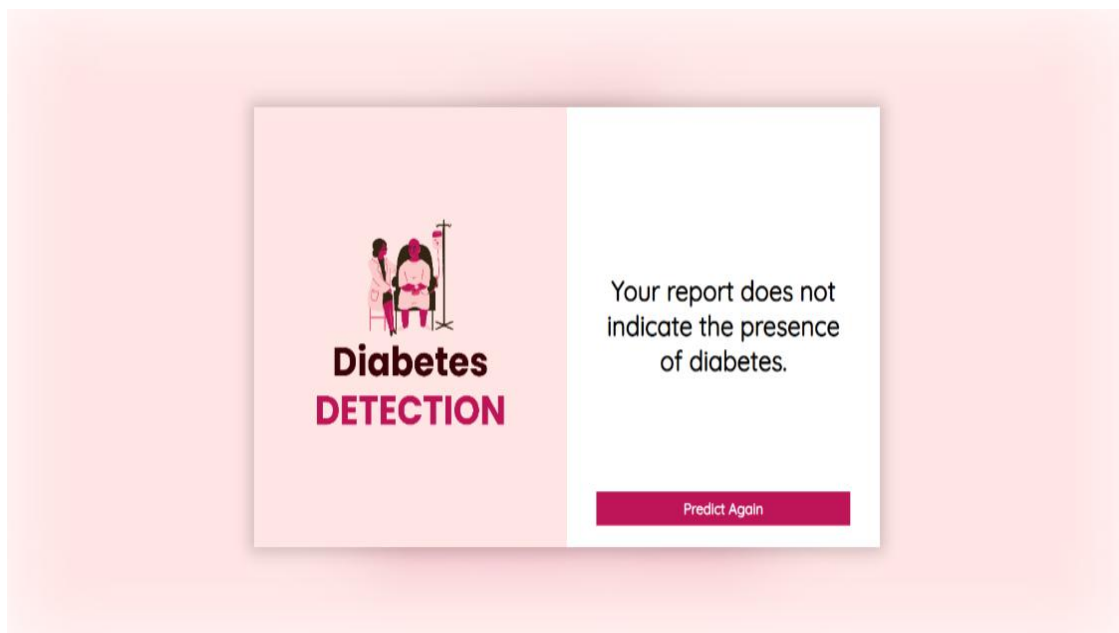
Negative Result:

In the below snapshot, we are entering the data of a non-diabetic person



The screenshot shows a web application titled "Diabetes Detection". On the left, there is an illustration of a doctor examining a patient. On the right, there is a form with the title "Diabetes Detection" and a list of input fields. The fields are filled with the following values: 31, 85, 66, 0, 26.6, 29, and 0.351. Below the fields is a red button labeled "Predict Diabetes".

after entering all the data we click the “Predict Diabetes” button, which will show you the result that the model does not indicate the presence of diabetes



The screenshot shows the result page of the "Diabetes Detection" application. On the left, there is the same illustration of a doctor examining a patient. On the right, there is a white box with the text "Your report does not indicate the presence of diabetes." and a red button labeled "Predict Again".

PHOTO GALLERY



We would like to extend our heartfelt gratitude to our project guide and project coordinator, Mrs. Dhivya C, for her unwavering support and guidance throughout the course of our final project.