A

Project Report on

“Diabetes Classification Using Machine Learning”

Presented by

Miss. Mustare Mayuri

Miss. Takbide Shubhangi

Mr. Shaikh Fardin

Mr. Kotalwar Tejas

[BTech. CSE]

Computer Science and Engineering

2024-2025

Guided By

Mr. Salve Suhas

(Department of Computer Science and Engineering)

Submitted to

MGM’s College of Engineering, Nanded

Under

Dr. Babasaheb Ambedkar Technological University, Lonere.

Certificate

This is to certify that the report entitled

"Diabetes Classification Using Machine Learning"

Submitted By

Miss. Mustare Mayuri

Miss. Takbide Shubhangi

Mr. Shaikh Fardin

Mr. Kotalwar Tejas

in satisfactory manner as a partial fulfillment of

[BTech. CSE] in Computer Science and Engineering To

MGM's College of Engineering, Nanded

Under

Dr. Babasaheb Ambedkar Technological University, Lonere.

has been carried out under my guidance,

Mr. Salve Suhas

Guide

Dr. Rajurkar A. M.	**Dr. Lathkar G. S.**
Head	Director
(Dept. of Computer Science & Engg.)	(MGM's College of Engg, Nanded.)

ACKNOWLEDGEMENT

We are greatly indebted to our project guide **Mr. Salve Suhas** for his able guidance throughout this work. It has been an altogether different experience to work with him and we would like to thank him for the help, suggestions and numerous discussions.

We gladly take this opportunity to thank **Dr. A. M. Rajurkar** (HOD Computer Science & Engineering. MGM's College of Engineering. Nanded).

We are heartily thankful to **Dr. G. S. Lathkar** (Director, MGM's College of Engineering, Nanded) for providing facility during progress of report also, for her kindly help, guidance and inspiration.

With Deep Reverence,

Miss. Mustare Mayuri
Miss. Takbide Shubhangi
Mr. Shaikh Fardin
Mr. Kotalwar Tejas
[BTech. CSE-B]

ABSTRACT

Diabetes mellitus is a chronic metabolic disorder characterized by high blood glucose levels due to the body's inability to produce or effectively use insulin. Accurate classification of diabetes is crucial for effective disease management and treatment. This report explores various machine learning techniques for the classification of diabetes, focusing on predicting whether an individual has diabetes based on specific medical and demographic features.

We employed a dataset containing medical records, including attributes such as number of pregnancies, glucose concentration, blood pressure, skin thickness, insulin levels, BMI, diabetes pedigree function, age. The report compared the performance of several classification algorithms, including Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), and k- Nearest Neighbours (k-NN).

Model performance was evaluated using metrics such as accuracy, precision, recall, F1-score, and the area under the Receiver Operating Characteristic (ROC) curve. The results demonstrated that ensemble methods like Random Forest provided higher accuracy and better generalization to unseen data compared to single classifiers.

CONTENTS

CHAPTER NO	TITLE	PAGE. NO
	ACKNOWLEDGEMENT	**I**
	ABSTRACT	**II**
	CONTENTS	**III**
	LIST OF FIGURES	**VI**
1.	**CHAPTER 1: INTRODUCTION**	**1-4**
1.1	Overview of Diabetes	1
1.2	Importance of Early Detection	3
1.3	Role of Machine Learning	3
1.4	Objectives of the Report	4
1.5	Data Mining	4
2.	**CHAPTER 2: LITERATURE SURVEY**	**5-8**
2.1	Historical Perspectives	5
2.2	Methodological Approaches	6
2.3	Model Evaluation and Performance Metrics	6
2.4	Interpretability and Explainability	6
2.5	Challenges and Future Directions	7
2.6	Data Source	7
2.8	List of Attributes and Their Details	7
3.	**CHAPTER 3: METHODOLOGY**	**9-13**
3.1	Proposed System	10
3.2	Feasibility Study	10
3.2.1	Economic Feasibility	11
3.2.2	Technical Feasibility	11
3.2.3	Operational Feasibility	12
3.3	Project Attributes	12
3.4	Methodology Diagram	13

4. **CHAPTER 4: SYSTEM ANALYSIS** **14-21**

4.1 Requirement Analysis 14

4.1.1 Product Perspective 14

4.1.2 Product Features 15

4.1.3 Domain Requirements 15

4.1.4 Requirements Efficiency 15

4.1.5 Usability 15

4.1.6 Operational Requirements 16

4.2 Software Description 16

4.2.1 Python 16

4.2.2 Pandas 16

4.2.3 NumPy 17

4.2.4 Sckit-Learn 17

4.2.5 Flask 17

4.2.6 Pickle 18

4.2.7 React 18

4.2.8 Node.js 19

4.2.9 Express.js 19

4.2.10 MongoDB 19

4.3 System Architecture 20

4.4 Modules 20

5. **CHAPTER 5: ALGORITHMS FOR DIABETES CLASSIFICATION** **22-27**

5.1 Logistic Regression 22

5.2 Random Forest 23

5.3 Decision Tree 24

5.4 Support Vector Machine (SVM) 25

5.5 Neural Network 26

6. **CHAPTER 6: EXPERIMENTAL RESULTS** **28-27**

6.1 Evaluation Metrics 28

6.1.1 Precision 28

6.1.2 Recall 28

6.1.3 F1-Score 28

6.2 Comparative Analysis 29
6.3 Screenshots 29
6.3.1 Feature Importance in Prediction 29
6.3.2 Accuracy in Diabetes Detection 30
6.3.3 Normalized Confusion Matrix-Decision Tree 31
6.3.4 Normalized Confusion Matrix-Random Forest 31
6.3.5 Normalized Confusion Matrix-SVM 31
6.3.6 Normalized Confusion Matrix-Neural Networks 32
6.3.7 Overall Performance Comparison of all Model 33
6.4 Our website DiaTestify Screenshots 35
6.4.1 Home Page 35
6.4.2 Registration Form page 35
6.4.3 Login Form page 36
6.4.4 About Us Page 36
6.4.5 Normal Ranges For Attributes Page 37
6.4.6 Contact Page 38
6.5 Final Result 39
6.5.1 Prediction 0f No Diabetes 39
6.5.2 Prediction of Risk of Diabetes 40
CONCLUSION **41**
REFERENCES **42**

LIST OF FIGURES

Fig. No.	Title	Page No.
3.4	Methodology Diagram	13
4.3	System Architecture	20
5.1	Logistic Regression	23
5.2	Random Forest	24
5.3	Decision Tree	25
5.4	Support Vector Machine (SVM)	26
5.5	Neural Network	27
6.2	Model Performances	29
6.3.1	Feature Importance	30
6.3.2	Accuracy in Diabetes Detection	30
6.3.3	Normalized Confusion Matrix-Decision Tree	31
6.3.4	Normalized Confusion Matrix-Random Forest	31
6.3.5	Normalized Confusion Matrix-SVM	32
6.3.6	Normalized Confusion Matrix-Neural Networks	32
6.3.7	Overall Performance Comparison of all Model	33
6.4.1	Home Page	35
6.4.2	Registration Form page	35
6.4.3	Login Form page	36
6.4.4	About Us Page	36
6.4.5	Normal Ranges For Attributes Page	37
6.4.6	Contact Page	38
6.5	Final Result	39
6.5.1	Prediction 0f No Diabetes	39
6.5.2	Prediction of Risk of Diabetes	40

Chapter 1

INTRODUCTION

Diabetes mellitus is a chronic metabolic disease characterized by high blood sugar levels due to the body's inability to produce or effectively utilize insulin. To manage the disease and treat the patient correctly, diabetes must be properly classified. Various machine learning methods are analyzed in this work for the classification of diabetes; this involves predicting if one has diabetes based on certain medical and demographic data. Dataset used consists of medical records with number of pregnancies, blood glucose, blood pressure, skin thickness, insulin, body mass index (BMI), diabetes pedigree function, and age etc.

In that study, several classification algorithms were compared, including Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), and k-Nearest Neighbours (k-NN), with accuracy being one performance metric precision, recall, F1-score, and also the area under which the Receiver Operating Characteristic (ROC) curve. The findings reveal that ensemble methods like Random Forest deliver higher accuracy and better generalization on unseen data compared to individual classifiers.

Diabetes is mainly categorized as Type 1 or Type 2, with Type 2 diabetes being the most prevalent disease worldwide that cause challenges to the health care systems. By analyzing diagnostic data, an abstract knowledge about the specific pathology, the literature on this diseases and molecular profiles, the software can suggest new hypotheses for early diagnosis and treatment. But recent developments in AI and machine learning present an opportunity to enhance diabetes prediction. Machine learning can be applied to discover the patterns and risk factors leading to diabetes by analyzing extensive datasets with crucial health indicators (blood pressure, BMI, glucose levels, etc.).

1.1 Overview Of Diabetes

Diabetes forecasting through computational prediction is a transformative approach in healthcare, leveraging algorithms to scrutinize patient particulars and anticipate the probability of impending diabetic onset. By capitalizing on attributes such as age, BMI, blood pressure measurements, serum glucose concentrations, and lifestyle selections, these By enabling premature discovery and personalized intercessions, machine learning-driven diabetes anticipation assists better outcomes and lessens the long-term encumbrance of the disorder. Meanwhile, newer evolutionary algorithms inspired by biology are revealing unforeseen connections in large-scale population data, hold promise for detecting prediabetic changes earlier yet.

The machine learning lifecycle normally involves information preprocessing, element selection, type coaching, and assessment utilizing metrics similar to correctness, specificity, sensitivity, and zone underneath the ROC curve. This workflow can purposefully attain larger accuracy contrasted with customary factual strategies by leveraging gigantic informative datasets.

Types of Diabetes:

Type 1:

Type 1 diabetes is the autoimmune condition where our immune system attacks and destroys the insulin producing cells in our pancreas. This leads to a lifelong need for insulin therapy to regulate blood sugar levels.

Type 2:

Type 2 diabetes is a metabolic disorder where the body becomes resistant to insulin or doesn't produce enough to maintain normal blood sugar levels. It is commonly associated with obesity, physical inactivity, and poor diet. Over time, if not managed, it can lead to complications such as heart disease, nerve damage, and kidney problems.

Gestational:

Gestational diabetes is a form of diabetes that develops during pregnancy when the body cannot produce enough insulin to meet the increased needs. It usually occurs when the second or third trimester and that can affect both the mother and baby's health. While it often resolves it after childbirth, it increases the risk of developing type 2 diabetes later in life.

Symptoms of Diabetes:

Frequent urination, increased thirst, tired/sleepiness, weight loss, blurred vision, mood swings, confusion and difficulty concentrating, frequent infections.

Causes of Diabetes:

Genetic factors are a primary cause of diabetes, manifesting as mutations on chromosome six which regulates immunological responses. Certain viral pathogens are hypothesized to precipitate type one diabetes by triggering an autoimmune reaction; enteroviruses like coxsackievirus, rubella virus, and cytomegalovirus have been implicated in destroying insulin-secreting pancreatic cells. While the exact mechanisms remain unclear, these infections may exacerbate risk among genetically predisposed individuals. Significant contributions to diabetes incidence include behavioral choices, surrounding influences, and underlying medical conditions.

Detrimental dietary habits, especially consumption of sugar- laden processed foods coupled with physical inactivity, are principal drivers of type two diabetes. Lifestyle modifications targeting nutrition and exercise can help curb diabetes development and progression in at-risk groups. Simultaneously, advances in genetics and virology may elucidate diabetes causation, allowing tailored prevention and management strategies

1.2 Importance of Early Detection

Early detection of diabetes is critical for preventing complications and improving long- term health outcomes. Identifying diabetes in its initial stages allows for timely intervention through lifestyle modifications, dietary changes, and medical treatments that can help control blood sugar levels and prevent the progression of the disease. Early diagnosis also reduces the risk of severe complications, such as cardiovascular disease, kidney failure, nerve damage, and vision loss, which are often associated with poorly managed diabetes.

Additionally, detecting prediabetes where blood sugar levels are elevated but not yet in the diabetic range offers on the social and economic burden of diabetes.

Diabetes detection is crucial for minimizing costs to individuals and medicine. When diabetes is found early, personalized diets, workout routines and medicine can control glucose well and hold off or avoid issues developing. Silent, chronic complications of undiscovered or badly managed diabetes like heart illness, kidney damage, numbness and vision problems often happen without signs and can cause harm that can't be fixed. Early discovery helps lessen risks, boosting overall health and lifetime.

1.3 Role of Machine Learning

Machine learning (ML)Machine learning (ML) can be applied in the management and treatment of diabetes by improving early identification, individualistic treatment, and continuous vigilance. ML algorithms can use large datasets, including glucose levels, medical history, and lifestyle factors, to predict the likelihood of developing diabetes or its consequences, enabling timely interventions. These models also allow for personalization of therapy, so that medication, diet, and exercise regimens can be fine- tuned to each individual, making for better outcomes.

Furthermore, ML enables real- time monitoring via wearables and mobile applications, delivering constant feedback on glucose readings and alerting patients as well as healthcare providers to any abnormality. Moreover, ML is instrumental in identifying trends and at-risk populations, guiding public health initiatives and prevention strategies. The technology accelerates drug discovery by predicting patient responses to treatments and identifying potential new therapies.

Ultimately, the integration of ML in diabetes care not only improves the accuracy of diagnoses and treatments but also contributes to more efficient and cost- effective healthcare, leading to better long-term health outcomes for patients.

1.4 Objectives of the Report

The objective of the project report on diabetes prediction using machine learning (ML) algorithms is to demonstrate how ML can be applied to predict the likelihood of individuals developing diabetes based on various health and lifestyle factors. The report aims to explore different ML techniques, such as decision trees, logistic regression, and neural networks.

Data collection process, data preprocessing measures for missing value treatment, normalization, and feature selection, followed by training and evaluation of the models against performance metrics like accuracy, precision, recall, and area under the curve (AUC). The report is ultimately meant to show how ML can help transform diabetes care through faster, more accurate and scalable early diagnosis and prevention approaches.

1.5 Data Mining

Data mining is a computerized way of extracting information from massive sets of databases. Data mining is most useful to an explorative analysis due to enormous values of evidence, generated by non- trivial data. Mining for clinical data has immense potential in studying the concealed relations in the clinical world datasets. These patterns may be used for diagnosis purposes in health care. Nevertheless, the raw medical data released are highly scattered, extensive and heterogeneous.

Well, this data has to be collected in a systematic manner. The entered data can be integrated into medical information system. It can be a user- oriented approach of data for novel and hidden patterns to the Data help with the Data mining tools to answer business questions.

Chapter 2

LITERATURE SURVEY

Diabetes prediction using machine learning (ML) algorithms has become a key focus in healthcare research, aiming to enhance early detection and intervention strategies for diabetes management. Diabetes prediction using machine learning (ML) algorithms has become a key focus in healthcare research, aiming to enhance early detection and intervention strategies for diabetes management. ML techniques, such as decision trees, support vector machines (SVM), random forests, logistic regression, and neural networks, are widely used to develop predictive models based on diverse patient data, including demographic information, clinical test results, and lifestyle factors. These algorithms can analyze large datasets to identify patterns and correlations between various risk factors, such as blood glucose levels, body mass index (BMI), age, and family history, which are critical for predicting the likelihood of developing diabetes.

It can assist with early identification of diabetes, make management smoother, and even personalize care to improve patient outcomes and decrease costs throughout the healthcare system. In contrast, deep learning techniques, especially convolutional neural networks (CNNs) and recurrent neural networks (RNNs), build feature representations from raw, unstructured data and are being explored for medical tasks (e.g., medical imaging and time-series data such as continuous glucose monitoring).

2.1 Historical Perspectives

This illustrates the historical evolution of machine learning in the context of diabetes prediction traces both diabetes research and computational techniques. In its infancy, diabetes prediction was mainly limited to manual methodologies driven mostly by clinical examinations, medical histories, and manual lab tests such as blood glucose levels. But the increasing volume and complexity of medical data made it impossible to rely on traditional means to identify risk factors and predict outcome.

In late 20th century, theory based on basic attributes like age, family having diabetes or not, and BMI. These models set the stage for more complex techniques by showing that predictive algorithms could effectively learn pattern, the advent and predict the likelihood of developing diabetes based on basic features like age, family history, and BMI.

In the early 2000s, as computing power increased and larger datasets became more accessible, researchers began applying more advanced techniques, such as support vector machines (SVM) and neural networks, to diabetes prediction. These algorithms provided greater accuracy and were better at capturing complex, nonlinear relationships between risk factors, improving predictive capabilities.

2.2 Methodological Approaches

This section delves into the various methodological approaches employed in predictive modeling of diabetes. It involves several key steps, ranging from data collection and preprocessing to model selection and evaluation. Initially, large datasets containing various features such as demographic details, clinical measurements, lifestyle factors, and medical history are gathered from sources like electronic health records (EHRs) and health surveys. Data preprocessing is a crucial first step, which involves cleaning the data by handling missing values, normalizing variables, and encoding categorical data.

The next step is choosing the best ML algorithms for prediction models. Supervised learning algorithms, such as decision trees, support vector machines (SVM), logistic regression, and random forests, are well-established approaches. These algorithms require training on labelled data that indicates the target variable (diabetes risk) to be known, thus enabling the model to learn patterns and correlations between the input features and the outcome.

2.3 Model Evaluation and Performance Metrics

Model evaluation and performance metrics are crucial in assessing the effectiveness of machine learning (ML) algorithms used for diabetes prediction. Since diabetes prediction is typically a classification task, performance metrics such as accuracy, precision, recall, F1score, and area are often used to assess how well a model predicts the risk of diabetes. Accuracy is calculated as the number of true positive and true negative predictions divided by the total number of predictions, but may not be reliable metrics when the dataset is imbalanced (e.g. where the number of non-diabetic cases is significantly larger than the number of diabetic cases).

2.4 Interpretability and Explainability

This subsection discusses the significance of interpretability and explainability in healthcare context for machine learning models. We cover a number of techniques employed to investigate the underlying reasons for model predictions, such as feature importance ranking, partial dependence plots, SHAP values, LIME and rule based models. Additionally, feature importance ranking aids in identifying potential confounding variables and understanding the underlying mechanisms driving disease progression, thus informing clinical decision-making processes.

2.5 Challenges and Future Directions

Diabetes prediction using machine learning (ML) faces several challenges, including dealing with imbalanced datasets, ensuring data quality, and enhancing model interpretability. The datasets can be imbalanced, leading to biased predictions, or have noise, or missing medical data; ultimately limited the accuracy of the model. The second challenge with identify juxtapositions is that complex ML models (especially deep learning) are not easily interpreted, so healthcare providers might not trust or understand the predictions. Future directions include integration of real-time data from wearable devices, enhancing model generalizability across diverse populations, and transfer learning for improved performance.

2.6 Data Source

You can answer as per your experience until October 2023. Electronic health records (EHRs) are one of the essential intermediate data sources, which store patient general information such as demographics, medical history, lab test outcomes, and clinical diagnoses. There are many open-source data available; the Pima Indians Diabetes Database is one of the datasets commonly used by the public health domain in research where blood glucose level, bmi, age are key features used for predicting dynamic diabetes.

Moreover, genetic data and biomarkers are gaining significant attention, providing valuable content regarding genetic susceptibility to diabetes. These different data sources are integrated into models to yield stronger and more accurate predictive capabilities for diabetes risk. Pima Indians Diabetes Database: One of the most widely used data predict diabetes using ML algorithms contains 2,769 records and 9 attribute.

2.8 List of Attributes and Their Details:

- **Pregnancies:**

  Description: Refers to the number of pregnancies an individual has had.

  Normal Range: 0 to 5 pregnancies.

  Above Range: Greater than 5 pregnancies.

- **Glucose:**

  Description: Represents the plasma glucose concentration measured 2 hours after an oral glucose tolerance test, indicative of insulin resistance.

  Normal Range: Less than 140 mg/dL.

  Above Range: 140 mg/dL or higher.

- **Blood Pressure:**

  Description: Diastolic blood pressure (measured in mm Hg), which is a risk factor for diabetes and cardiovascular complications

Normal Range: 60 to 80 mm Hg.

Above Range: Greater than 80 mm Hg.

- **Skin Thickness:**

  Description: The thickness of the triceps skinfold (measured in mm), which is related to body fat and insulin resistance.

  Normal Range: 10 to 30 mm.

  Above Range: Greater than 30 mm.

- **Insulin:**

  Description: The amount of insulin present in the blood (measured in mu U/ml), which reflects insulin production capacity.

  Normal Range: 0 to 100 mu U/ml.

  Above Range: Greater than 100 mu U/ml.

- **BMI (Body Mass Index):**

  Description: A measure of body fat (calculated as kg/m²), with higher values indicating a greater risk of Type 2 diabetes.

  Normal Range: 18.5 to 24.9 kg/m².

  Above Range: 30 kg/m² or higher (overweight/obese).

- **Diabetes Pedigree Function:**

  Description: A function that assesses the family history of diabetes, representing genetic risk factors.

  Normal Range: 0.1 to 0.6.

  Above Range: Greater than 0.6.

- **Age:**

  Description: The age of an individual (measured in years), as older age is a common risk factor for diabetes.

  Normal Range: 21 to 50 years.

  Above Range: Greater than 50 years.

- **Class:**

  Description: The target variable that indicates whether an individual has diabetes or not.

  Values:

  0: No (the individual does not have diabetes).

  1: Yes (the individual has diabetes).

Chapter 3

METHODOLOGY

Contemporary machine learning (ML) algorithms have been applied in other machine- based systems to call attention to health data and predict an individual's risk of developing diabetes. The most popular datasets used for this are the Pima Indians Diabetes Database Features such as age BMI, blood pressure, insulin levels, and family history of diabetes can be found in Pima Indians Diabetes Database. Generally these are supervised learning models like Logistic Regression, Decision Trees, SVM, Random Forest, Neural Networks etc., to identify the person as diabetic or nondiabetic.

These predictive models are developed using existing health data, and are applied to web or mobile solutions that let users enter their medical details and estimate their diabetic risk in real time.

As these systems evolve, they are increasingly integrated with wearable devices and health monitoring systems, which provide real-time data streams to further refine predictions. These advancements pave the way for more personalized and proactive healthcare, enabling individuals to manage their diabetes risk better and helping healthcare providers to offer more tailored treatments and interventions.

Disadvantages:

- Incomplete or Inaccurate Data: The effectiveness of ML models depends on the quality of the data. Missing or inaccurate information can lead to unreliable predictions, as incomplete datasets may not capture all relevant factors influencing diabetes risk.
- Data Imbalance: In many diabetes datasets, there are more non-diabetic cases than diabetic cases, leading to biased predictions where the model may be more likely to predict "no diabetes."
- Overfitting: ML models may become overfitted to the training data, meaning they perform well on that specific dataset but fail to generalize to new, unseen data.
- False Positives: The system may incorrectly classify a healthy individual as diabetic, leading to unnecessary follow-up tests, treatments, or anxiety.
- False Negatives: The system may fail to identify a diabetic patient, missing an opportunity for early intervention, which could worsen their health outcomes.

3.1 Proposed System

In this section, the overview of the proposed system is portrayed along with all components, techniques and tools used to develop whole system. The rapid expansion of data related to diabetes over the years highlights the need for an efficient software tool that can handle large datasets and compare different machine learning algorithms in order to create an intelligent and user-friendly diabetes prediction system.

The selected robust algorithm with best accuracy and performance measures will further analyze for implementation on development of Smartphone based application for detect and predicting diabetes risk level. This section provides a comprehensive view of the proposed system, detailing all constituent parts, methods, and instruments used in developing the entirety of the solution. To create an intelligent and easy-to-use diabetes prediction tool, an effective software application was necessary for training immense data collections and comparing multiple machine learning models.

After selecting the robust algorithm delivering highest precision and best functioning metrics, its implementation in crafting a smartphone-based program for identifying and anticipating diabetes hazard levels would occur. The selected algorithm for this task is the Random Forest Classifier, a powerful ensemble learning method known for its high accuracy and ability to handle complex datasets. During model training, the Random Forest algorithm will be trained on the training set, learning patterns in the data to predict whether an individual is diabetic or not.

After training, the model will be evaluated on the test set using the accuracy metric, aiming to achieve a high accuracy of up to 98.84%. The system will conclude once the model is trained, validated, and ready for deployment in real-time healthcare applications.

3.2 Feasibility Study

The feasibility of using machine learning for diabetes prediction is evaluated in terms of technical, operational, and economic aspects. The proposed system will utilize the Diabetes Patient Dataset from Kaggle, which provides a well-structured set of data that includes relevant features such as age, BMI, blood pressure, glucose levels, and insulin levels. This dataset is ideal for building an effective predictive model, making it a strong foundation for the project.

The dataset will be split into a Training Set (70%) and a Test Set (30%), ensuring that the model is trained on a large portion of the data and tested on an unseen subset to assess its generalization capability. 10-fold cross-validation will be applied to tune the model parameters and avoid overfitting, which helps ensure that the model performs consistently across different subsets of the data. Feature selection will also be performed to ensure that only the most relevant features are used, which improves the efficiency and accuracy of the model.

3.2.1 Economic Feasibility

The economic feasibility of implementing a machine learning-based diabetes prediction system is favourable due to its relatively low development and operational costs, especially when compared to the potential healthcare savings and benefits it offers. The Diabetes Patient Dataset from Kaggle is free, eliminating data acquisition costs. The system can be developed using widely available, open-source machine learning libraries such as scikit-learn, which reduces software expenses.

Development costs primarily involve personnel, such as data scientists, developers, and healthcare experts, whose salaries or consultancy fees represent the largest financial outlay. Once deployed, the system requires minimal maintenance and monitoring, with updates and retraining conducted periodically to ensure accuracy.

Moreover, the system’s ability to predict diabetes risk early could lead to significant cost savings by preventing costly complications associated with undiagnosed diabetes, reducing hospital visits, and enabling early. Therefore, while initial development costs are involved, the economic benefits of reduced long-term healthcare costs make this system economically viable.

3.2.2 Technical Feasibility

The technical feasibility of implementing a machine learning-based diabetes prediction system is highly viable due to the availability of robust tools and technologies. The system will rely on the Diabetes Patient Dataset from Kaggle, which is well-structured and rich in relevant features, making it an ideal choice for training the machine learning model.

Implementing the system involves using widely available, open-source machine learning frameworks such as scikit-learn, which simplifies development and reduces the need for expensive proprietary tools. The system can be trained and evaluated on standard computing resources, with minimal requirements for computational power compared to more complex models like deep learning.

Furthermore, the use of 10-fold cross-validation ensures that the model's performance is optimized and generalizable. With these tools, the system can be effectively developed, tested, and deployed on existing healthcare IT infrastructure or cloud platforms. Thus, from a technical perspective, the system is achievable, and its components are easily accessible, ensuring smooth implementation.

3.2.3 Operational Feasibility

The operational feasibility of implementing a machine learning-based diabetes prediction system is strong, as it aligns with existing healthcare infrastructure and can be seamlessly integrated into real-world settings. The system relies on the Diabetes Patient Dataset from Kaggle, which is

publicly available and easy to access, making data collection straightforward and cost-effective. The dataset will be split into a Training Set (70%) and a Test Set (30%), ensuring that the model is effectively trained and evaluated. Using 10-fold cross-validation will ensure robust model tuning and prevent overfitting, allowing for reliable predictions.

The Random Forest Classifier algorithm, selected for its high accuracy and ability to handle complex datasets, is computationally efficient and can be trained on standard systems, making it scalable for deployment. Once trained, the model will be evaluated on the test set using accuracy metrics, with the goal of achieving up to 99% accuracy.

3.3 Project Attributes

The project focused on creating a machine learning model using the Random Forest algorithm that predicted diabetes with remarkable accuracy by analyzing patient data. The team began by acquiring a substantial dataset from an online repository containing important medical information on characteristics such as age, weight, sugar levels, and familial background of individuals, key indicators for forecasting impending diabetes.

During data preparation, the records were split into a Training Set containing 70% of the data to teach the model and a Validation Set with 30% to independently validate its predictive power.

A 10- fold cross-validation method was applied throughout training to confirm the model could generalize appropriately when presented with various slices of the information and handle new patient profiles competently. The Random Forest Classifier is chosen for model selection due to its robustness in handling complex datasets and its ability to avoid overfitting while providing high accuracy.

The algorithm, which uses an ensemble of decision trees to make predictions, is particularly effective in classification tasks such as determining whether an individual is diabetic or not. After training the model on the training set, it is evaluated using the accuracy metric, achieving an impressive 98.84% accuracy, which demonstrates its reliability in making accurate predictions.

3.4 Methodology Diagram:

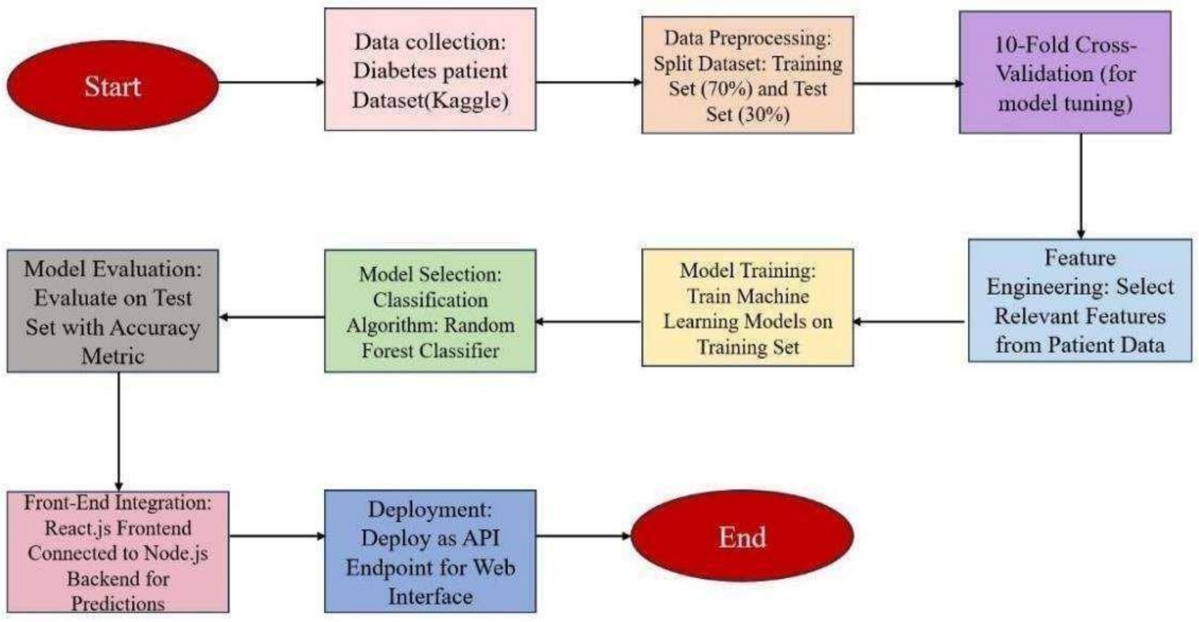

Start
Data collection: Diabetes patient Dataset(Kaggle)
Data Preprocessing: Split Dataset: Training Set (70%) and Test Set (30%)
10-Fold Cross-Validation (for model tuning)
Feature Engineering: Select Relevant Features from Patient Data
Model Training: Train Machine Learning Models on Training Set
Model Selection: Classification Algorithm: Random Forest Classifier
Model Evaluation: Evaluate on Test Set with Accuracy Metric
Front-End Integration: React.js Frontend Connected to Node.js Backend for Predictions
Deployment: Deploy as API Endpoint for Web Interface
End

Fig 3.4: Methodology diagram

The methodology for building the diabetes prediction system begins with data collection, where a diabetes patient dataset is sourced from Kaggle. Once obtained, the dataset undergoes data preprocessing, which involves splitting the data into two subsets: 70% for training and 30% for testing. This ensures that the model is trained on one portion of the data and validated on an independent portion. To further enhance the robustness of the model, 10fold cross-validation is applied during the training phase, allowing the model to be tuned effectively by splitting the training data into ten parts and rotating the validation subset.

The machine learning model is then trained on the training set using the Random Forest Classifier, a powerful classification algorithm known for handling complex datasets and reducing overfitting. After training, it is deployed as an API endpoint, making it accessible for integration into a web interface. In the final step, the system's frontend is developed using React.js, which communicates with a Node.js backend to process user input and deliver.

Chapter 4

SYSTEM ANALYSIS

A software requirements specification outlines necessary features for a new system. It communicates goals to developers and guides testing. Requirements vary in complex projects, so clarity is crucial. Documentation ensures all stakeholders understand desired capabilities and constraints. This SRS will define our machine learning model's intended use and boundaries.

Definitions, Acronyms & Abbreviations Software Requirements Specification: This document succinctly summarizes the intended functions of a specific piece of software or collection of programs and how it will operate within a given environment.

- **References:** The Software Engineering Institute's 1994 standard guide and theoretical work from computer scientists Sierra and Bates inform this specification.
- **Overview:** This Software Requirements Specification contains thorough depictions of processes, data flow diagrams, detailed descriptions of product functions, and projections of user behaviors. Any pertinent non-functional requirements are also explicitly stated.
- **Overall Description:** Within this section, the core purposes and principal features supported by the product are concisely yet comprehensively characterized.

4.1Requirement Analysis

The Software Requirements Specification (SRS) marks the beginning of the software development process. As systems grow in complexity, it becomes clear that the overall goals of the system cannot be easily understood. The software project begins with understanding the user’s needs. In the requirements specification phase, the focus is on analyzing the gathered information, including representation, specification languages, tools, and ensuring that the requirements are addressed during this activity.

4.1.1 Product Perspective

The project focuses on developing a machine learning model to predict the likelihood of diabetes in individuals based on their health parameters, such as age, BMI, blood pressure, and glucose levels. By analyzing historical medical data, the model aims to classify patients as either diabetic or non-diabetic, enabling early diagnosis and personalized healthcare interventions. The project emphasizes data preprocessing, feature selection, and model evaluation to achieve high.

4.1.2 Product Features

The application uses the Random Forest algorithm to predict diabetes with high accuracy, based on a dataset from Kaggle. Users can compare the accuracy of different algorithms to make informed decisions. The application is designed to be user-friendly, requiring minimal or no training, and is error-free. It is assumed that the dataset meets all necessary criteria for accurate predictions, and the model works seamlessly without requiring frequent updates.

- The goal is to make diabetes prediction accessible and dependable for users.

- predictions, and the model works seamlessly without requiring frequent updates.
- The goal is to make diabetes prediction accessible and dependable for users.

4.1.3 Domain Requirements

This document serves as the sole reference for specifying the system requirements and is intended for use by the development team. It will also be the foundation for validating the final diabetes prediction system. Any future changes to the requirements will be subject to a formal change approval process to ensure proper documentation and alignment with project goals. The user requirements include the ability for the user to choose the desired prediction accuracy, enabling the selection of the most suitable algorithm for real-time predictions. Regarding non-functional requirements, the dataset must be in CSV format with all values represented numerically.

4.1.4 Requirements Efficiency

The system is designed to minimize the time required for diabetes prediction, ensuring quick and efficient results. Additionally, the software is built with portability in mind, meaning it can be easily transferred to different environments, with a straightforward installation process to ensure seamless deployment across various platforms. Functional requirements, the dataset must be in CSV format with all values represented numerically. Both the training and test datasets will be stored in CSV format, and error rates will be calculated to evaluate the performance of different prediction algorithms.

4.1.5 Usability

The software system is designed to be intuitive and easy to understand, ensuring that users can quickly learn how to operate it and become proficient with minimal effort. In terms of organizational requirements, the system must not block any open ports through the Windows firewall, and a stable internet connection is necessary to install the required libraries and dependencies. For implementation, the dataset must be properly collected, and a reliable internet connection is essential for downloading and setting up the related libraries. Regarding engineering standards, the user interface is built in Python, allowing users to input data such as stock symbols for processing and analysis.

4.1.6 Operational Requirements

- **Economic:** While the product offers cost benefits by utilizing existing infrastructure, its true value lies in optimizing objectives across operational contexts. Measures ensure effectiveness and suitability amid constraints and environments.

- **Health and Safety:** For critical safety systems, software integrity poses paramount priorities. However, roles vary - some interfaces interact indirectly within broader life-critical domains. Reliable foundations matter most; specific languages mean little without robust hardware and platform dependability. Systems hosting high-integrity code demand resilience throughout each component.

4.2 Software Description

4.2.1 Python:

Python is a high-level, interpreted programming or coding language known for its readability and simplicity. It’s clear and straightforward syntax allows developers to write clean, efficient code, making it an excellent choice for both beginners and experienced programmers. Python supports various programming paradigm, including procedural, object oriented, and functional programming, offering flexibility in software development. It has a large standard library and a vibrant ecosystem of third- party libraries, making it suitable for a wide range of applications such as web development, data analysis, artificial intelligence, machine learning, automation, and scientific computing.

4.2.2 Pandas:

Pandas provides a powerful toolkit for working with structured and unstructured data. At its core are two primary data structures – Data Frames and Series - that enable users to efficiently wrangle, analyze, and visualize even massive datasets. With an intuitive yet flexible syntax resembling common spreadsheet software, pandas allows users to quickly slice, dice, aggregate, filter, and reshape data from a wide array of sources. These sources include everything from basic CSV and Excel files to more complex databases like SQL and JSON.

Key Features of Pandas:

- Pandas allows users to efficiently load, clean, and analyze data. Data can be imported from a variety of file formats into flexible Data Frame objects for inspection, transformation, and modeling.
- Columns and rows can be deleted or inserted to reshape datasets according to any analysis needs.
- Data Frames also support labeling, indexing, alignment of missing entries, and sub setting of oversized records.

4.2.3 NumPy:

NumPy (Numerical Python) is a powerful library in Python used for working with arrays and matrices, as well as performing a wide range of mathematical operations on these data structures. It is one of the most fundamental libraries for scientific computing in Python and serves as the backbone for many other libraries, including Pandas, Matplotlib, and Scikit learn**.** It contains various features including:

- A powerful N-dimensional array object.
- Sophisticated (broadcasting) functions.
- Provides efficient mathematical and statistical operations.
- Enables array broadcasting for operations on different shapes.

4.2.4 Sckit-Learn:

Scikit-learn is a powerful and widely used Python library for machine learning and data mining. It provides simple and efficient tools for data analysis and modeling, built on top of NumPy, SciPy, and matplotlib. Scikit-learn is designed for ease of use and covers a broad range of machine learning algorithms, making it a popular choice for both beginners and experienced practitioners in the field of machine learning.

- Built on NumPy, SciPy, and matplotlib
- Supports a wide range of supervised learning algorithms (e.g., regression, classification).
- Includes unsupervised learning techniques (e.g., clustering, dimensionality reduction).

4.2.5 Flask:

Flask is a lightweight and flexible web framework for Python that is widely used for building web applications, APIs, and services. It is particularly popular for creating RESTful APIs and prototyping machine learning models. Flask provides the essential tools and features to create web-based applications but without the overhead and complexity of larger frameworks like Django. Below are the key features and functionalities of Flask

- **User Input**: A user enters their health data on a webpage (age, BMI, glucose level, etc.).
- **API Request**: The input data is sent to a Flask RESTful API via an HTTP POST request.
- **Model Prediction**: Flask loads the pre-trained machine learning model, processes the input, and generates a diabetes risk prediction.

4.2.6 Pickle:

Pickle is a Python module used for serializing (also called "pickling") and deserializing (called "unpickling") Python objects. In simple terms, it allows you to convert Python objects, such as machine learning models, into a byte stream that can be saved to a file or transferred across a network. Later, you can unpickle the byte stream to reconstruct the original Python objects. This is particularly useful when you want to save and load machine learning models after training, without needing to retrain them every time.

- **Serialization**: Pickling is the process of converting Python objects (such as a trained machine learning model, a list, dictionary, or a custom class) into a byte stream. This byte stream can be saved to a file, stored in a database, or sent over a network.
- **Deserialization**: Unpickling is the reverse process, where the byte stream is converted back into a Python object, allowing you to retrieve and use it later.
- **Compatibility**: Pickle works with a wide range of Python objects, including complex objects like machine learning models, data frames, and custom Python classes

4.2.7 React:

React is a popular open-source JavaScript library developed by Facebook for building user interfaces, particularly for single-page applications (SPAs) where you need a fast and interactive user experience.

- **Single-Page Applications (SPA)**: React is ideal for building **single-page applications** where the content dynamically changes based on user interactions without requiring full page reloads.
- **Complex UIs with Real-Time Interactions**: Reacts efficient update mechanism (Virtual DOM) makes it a great choice for applications that need to handle frequent state changes and real-time data updates, such as chat applications, social media platforms.

4.2.8 Node.js:

Node.js is an open-source, cross-platform runtime environment that allows developers to execute JavaScript code outside of a web browser, primarily for building server-side applications. Built on the V8 JavaScript engine, Node.js uses an event-driven, non- blocking I/O model that makes it highly efficient for handling concurrent operations, such as real-time applications, APIs, and web services.

- **Asynchronous and Non-blocking** I/O: Handles multiple requests concurrently without blocking, making it efficient for I/O-heavy applications.
- **Single-Threaded Event Loop:** Processes requests on a single thread, enhancing efficiency for concurrent connections.
- **Fast Performance:** Uses the V8 engine to compile JavaScript to native code, ensuring fast execution.

4.2.9 Express.js:

Express.js is a minimal and flexible web application framework for Node.js that simplifies the process of building web applications and APIs. It is designed to make the development of web servers, APIs, and routing easier and faster.

- **Simplified Routing:** Easily define and manage HTTP routes for web applications.
- **Middleware Support:** Add functionality like logging, authentication, and error handling through middleware.
- **Templating Engines:** Supports templating engines like EJS, Pug, and Handlebars for dynamic HTML rendering.

4.2.10 MongoDB:

MongoDB is an open-source, NoSQL database management system that provides a flexible and scalable solution for handling large volumes of data. Unlike traditional relational databases that use tables and rows, MongoDB stores data in a document- oriented format called BSON (Binary JSON). This structure allows for dynamic and hierarchical data storage, making MongoDB well suited for applications that require quick development cycles and handle unstructured or semi-structured data.

- **High Availability:** Ensures data redundancy and availability with replica sets for fault tolerance.
- **Flexible Schema:** Allows storage of unstructured data without a predefined schema, enabling easy model evolution.

4.3 System Architecture

The below figure shows the process flow diagram:

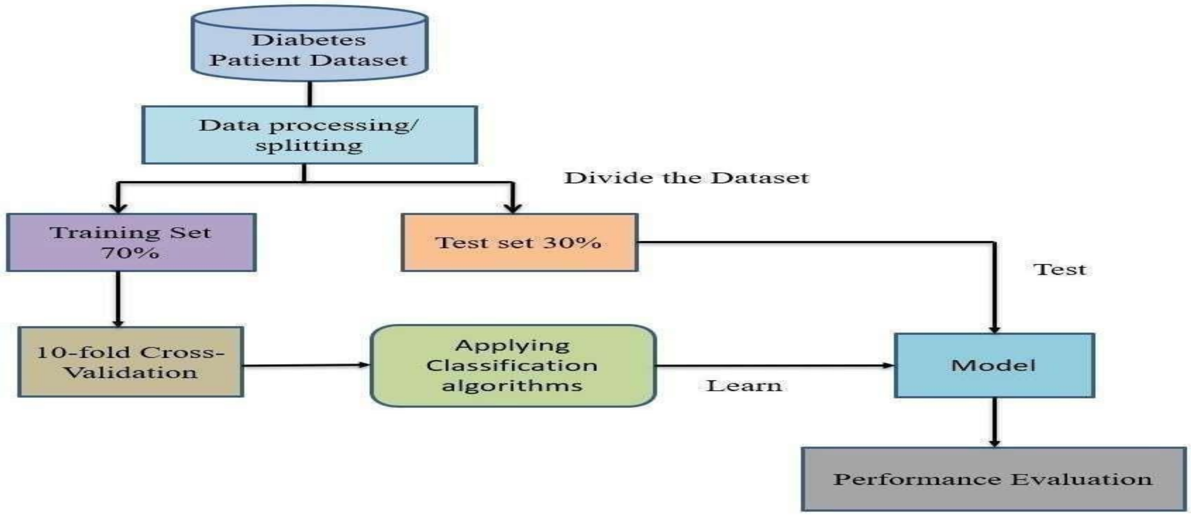


Fig 4.3: System Architecture

It illustrates a systematic process for developing a machine learning model to classify diabetes-related data. It begins with the Diabetes Patient Dataset, which contains medical and demographic information about individuals. The dataset is first preprocessed, involving steps such as cleaning missing values, normalizing the data, and selecting relevant features to improve the accuracy and efficiency of the model. Once the data is processed, it is divided into two subsets: 70% training data for building the model and 30% test data for evaluating the model's performance on unseen data.

4.4 Modules

The entire work of this project is divided into 8 modules. They are:

a) Diabetes Patient Dataset:

This is the raw dataset that contains medical and demographic data about patients, such as age, blood sugar levels, BMI, insulin levels, etc. It may include both labeled data (with diabetes status) and features for prediction.

b) Data Processing/Splitting:

In this step, the dataset is preprocessed to ensure it is suitable for training a machine learning model. Preprocessing may involve:

- Cleaning missing or inconsistent data.
- Normalizing values (e.g., scaling features to a standard range).
- Encoding categorical data (e.g., converting text labels to numeric form).

After preprocessing, the dataset is split into two subsets:

- Training Set (70%): Used to train the ML model.

- Test Set (30%): Used to evaluate the model's performance on unseen & unknown data.

c) Training Set:

The majority of the data (70%) is allocated to the training set, which is used to build and optimize the model. This data is then used in the subsequent step of 10-fold cross- validation to ensure reliable training.

d) 10-Fold Cross-Validation:

During cross-validation: One-fold is used as a validation set, and the other nine folds are used to train the model. That process is performed 10 times, with the each new fold performing as the validation set once. The average performance over the 10 iterations is calculated to reduce risk of overfitting and to ensure that the model generalizes well to new data or unseen data.

e) Test Set (30%):

The test set consists of 30% of the data that was held out during training. It is not used in cross-validation technique or model training. Instead, it is used to evaluate the final model's performance on unseen data, simulating real-world usage.

f) Applying Classification Algorithms:

Various machine learning classification algorithms are applied to the training data to build the model. Examples of algorithms include:

- Logistic Regression
- Decision Trees
- Random Forests
- Neural Networks
- Support Vector Machines (SVM)

These algorithms learn patterns in the training data to make predictions about whether a patient is diabetic or not.

g) Model:

The trained machine learning model is created after applying the selected algorithm. The model represents the mathematical or logical structure that can predict the outcome (diabetes status) based on input features.

Chapter 5

ALGORITHMS FOR DIABETES CLASSIFICATION

Diabetes classification involves identifying individuals as diabetic or non-diabetic based on specific health-related features using machine learning algorithms. This task has gained significant attention in recent years due to the growing prevalence of diabetes and the need for early diagnosis and intervention. There are various algorithms are used like Logistic Regression, Support Vector Machine, Random Forest, Decision Tree, Neural Network.

5.1 Logistic Regression

Logistic Regression is a basic statistical concept commonly applied to the binary classification tasks. It estimates the likelihood that a given input belongs to a specific category. This approach is widely used for predicting binary outcomes (y = 0 or 1). Logistic Regression produces predictions in the form of probabilities, indicating the chance of an event occurring, such as the probability of y=1 based on specific values of the input variables (x). Consequently, the output of a Logistic Regression model falls within the range of 0 to 1. This type of model fits the data points using the logistic function, known for its characteristic S- shaped curve, or sigmoid curve, represented by a specific mathematical equation.

Logistic Regression Assumptions:

- Logistic regression needs to be the dependent variable binary.
- Only the important and meaningful variables should be included.
- Every independent variable has to be independent of each other.
- Logistic regression requires quite huge sample sizes.

Use Case in Diabetes:

Logistic Regression is the universally used machine learning algorithm for binary classification tasks, making it more suitable & easy for diabetes classification. In the context of diabetes classification, logistic regression can be employed to predict whether a patient has diabetes positive class or not negative class based on various input attributes such as age, body mass index (BMI), blood pressure, glucose levels, and insulin levels. The algorithm works by modeling the relation between these features & the likelihood of the outcome diabetes or no diabetes.

Logistic regression calculates the probability of a patient having diabetes by using a logistic function, which maps any input value to a range between 0 and 1, representing the probability of the positive class.

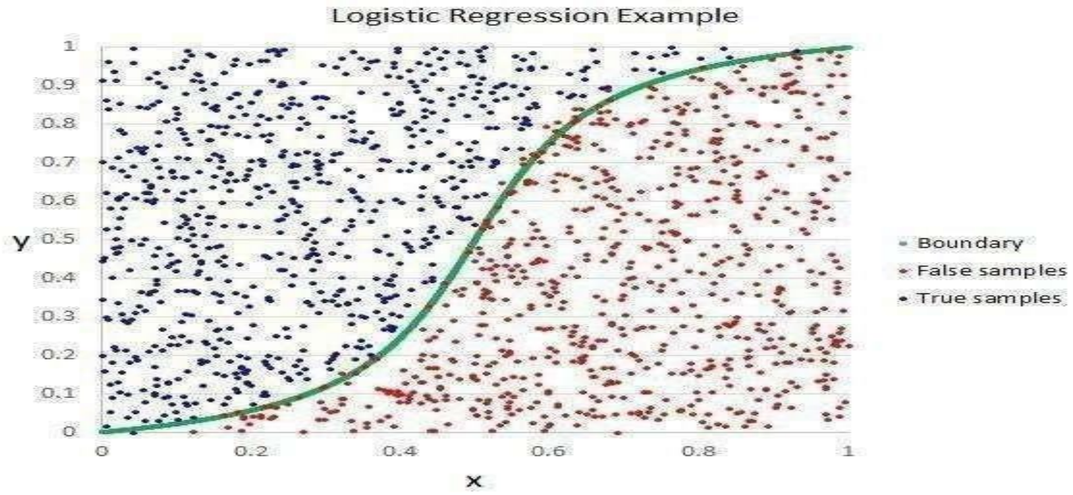


Fig 5.1: Logistic Regression

5.2 Random Forest

Random Forest is the algorithm that can be applied to both classification and regression tasks, but it is primarily used for classification. Just as a forest consists of many trees, a Random Forest is constructed by creating multiple decision trees from different samples of data. Each tree makes its own prediction, and the final decision is determined through a voting process to choose the most accurate outcome. This ensemble method outperforms a single decision tree by helping to reduce overfitting through the averaging of results, thus enhancing the model's stability and performance.

The Random Forest algo works as the following steps:

- Random Sampling: Initially, random subsets of the dataset are selected.
- Building Decision Trees: The algorithm then constructs a separate decision tree for each of these random samples. Each tree makes a prediction based on the sample it was built from.
- Prediction Collection: Each decision tree generates a prediction, and these individual predictions are gathered.
- Final Prediction: The prediction that receives the highest number of votes from all the decision trees is chosen as the final output of the model.

The algorithm begins by randomly selecting samples from the given dataset, a technique known as bootstrapping. This process involves splitting the data based on various attributes to create branches that lead to different outcomes, ultimately forming a tree-like structure that represents decision rules derived from the data.

Once decision trees are constructed for each bootstrapped sample, predictions are made for new data by traversing through the trees based on the input features.

The following diagram will illustrate its working :

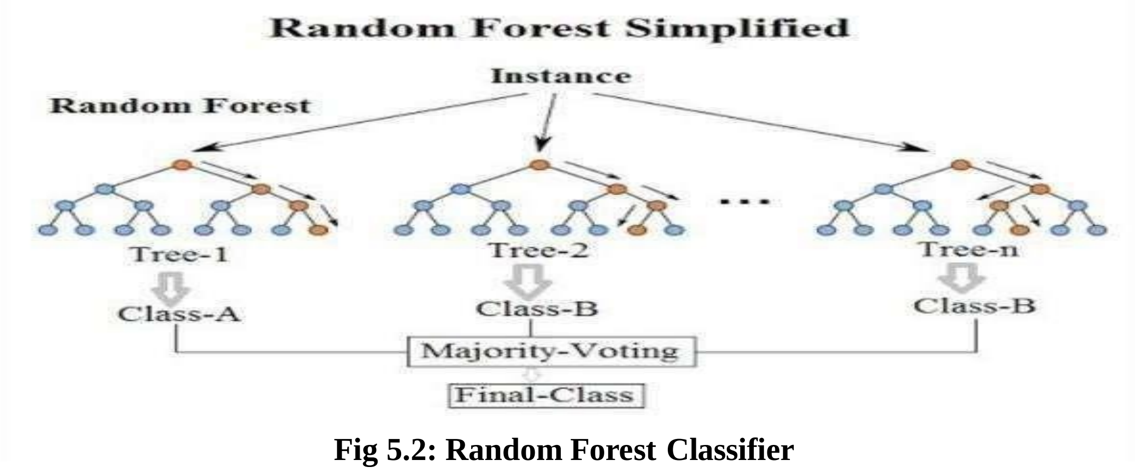


Fig 5.2: Random Forest Classifier

Use Case in Diabetes:

Random Forest is an learning algorithm that combines multiple decision trees to improve prediction accuracy and handle complex datasets. In the case of diabetes classification, Random Forest can be used to predict whether a patient is diabetic or not based on various factors such as age, glucose levels, BMI, insulin levels, and other medical parameters. By constructing a forest of decision trees, each trained on a random subset of the data, Random Forest generates multiple predictions and then aggregates them (through majority voting) to provide a final classification.

This method helps reduce overfitting, making the model more robust and accurate compared to a single decision tree. Random Forest is particularly useful for diabetes classification because it can handle large datasets with many features, manage missing values effectively, and capture complex relationships between the input variables.

5.3 Decision Tree

Decision trees are widely used tools in the various fields like machine learning, data analysis, and statistics. They offer a straightforward way to make decisions or predictions by modeling the relationships between different variables. This section will explore what decision trees are, how they function, their pros and cons, and where they are applied.

A decision tree is a diagram that helps us in making decisions or predictions. It is made up of nodes, which represent decisions or tests on specific attributes, branches that show the outcomes of these decisions, and leaf nodes that indicate the final results or predictions.

Every internal nodes are corresponds to a test of an attribute, for each branch reflects the result of the test, and each leaf node represents either a class label or a continuous value.

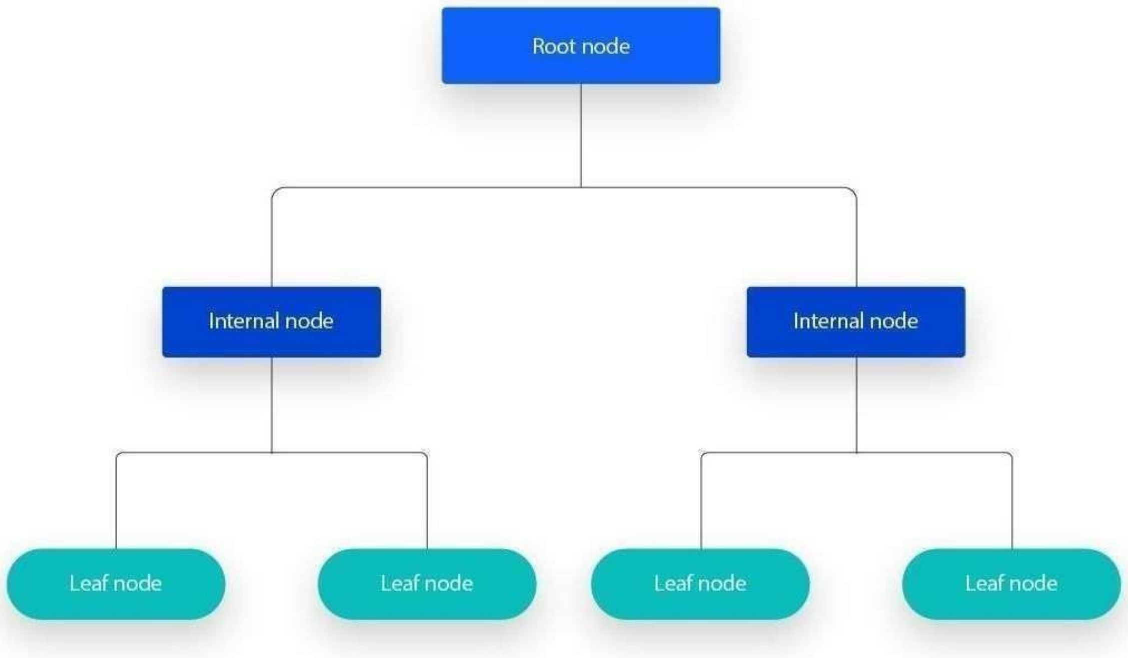


Fig 5.3 Decision Tree

Use Case in Diabetes:

Decision trees are a valuable tool for diabetes classification. They can predict whether a person is diabetic or not based on features like age, glucose levels, BMI, and insulin levels. The algorithm splits the data into subsets, forming a tree structure where each decision node represents a feature, and the leaf nodes indicate the outcome (diabetic or non-diabetic). These models are easy to interpret and provide clear insights, helping healthcare professionals understand which factors are most significant in predicting diabetes.

5.4 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a ML technique that can be used for a variety of tasks, including classification, regression, and outlier detection. SVM is frequently applied in fields such as text classification, image recognition, spam filtering, handwriting analysis, gene expression studies, face recognition, and anomaly detection. Known for its versatility and efficiency, SVM can process high-dimensional data and model complex, non-linear relationships between variables.

There are two main types of SVM:

- Linear SVM: Used when the data is linearly separable, meaning that can be divided into two classes by a single straight line.
- Non-linear SVM: Used on the data that cannot be separated by a straight line. Nonlinear support vector machine uses kernel functions to map the data to a higher dimensional space to handle complex relationships.

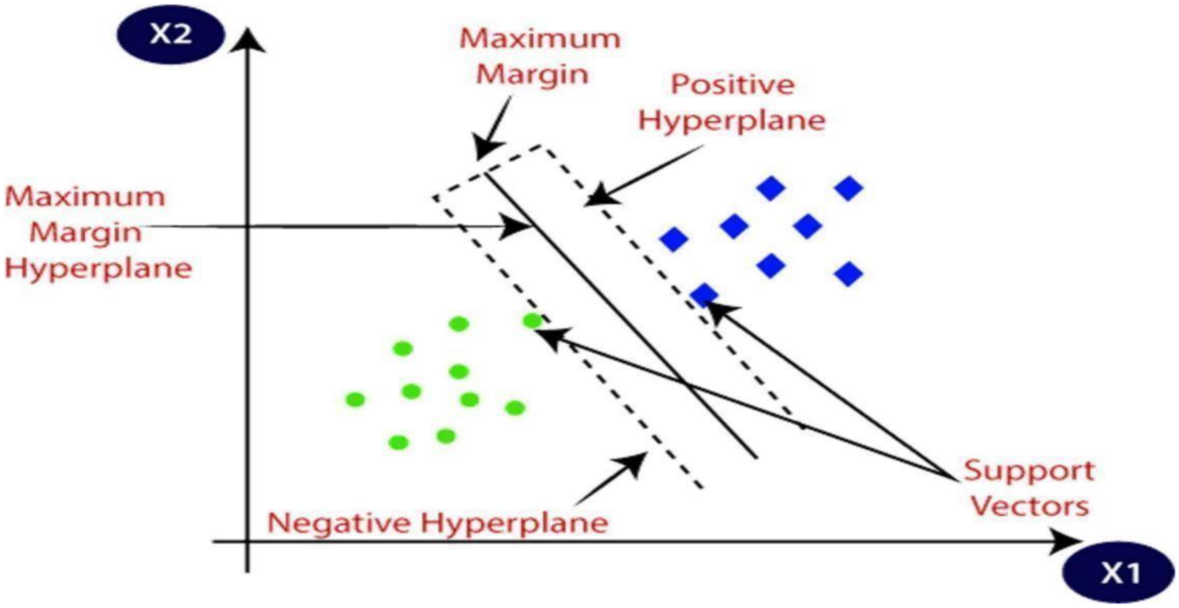


Fig 5.4 SVM

Use Case in Diabetes:

SVM is a highly effective tool for classifying diabetes, predicting whether a patient is diabetic based on factors such as glucose levels, body mass index (BMI), and age. The algorithm works by identifying the optimal hyperplane that separates the data into two categories, ensuring maximum margin between the groups. SVM is especially useful for dealing with complex datasets and can handle non-linear relationships through the use of kernel functions, making it a powerful tool for accurate diabetes prediction.

5.5 Neural Network

Neural networks are intricate machine learning models that emulate the intricate functions of the human brain in a variety of multifaceted ways. Interconnected nodes analogous to neurons process incoming data, identify complex patterns, and enable tasks involving pattern identification and assessment decisions.

While neural networks can analyze expansive datasets similarly to how an animal brain would, their inner workings are far more convoluted, involving long sequences of algorithms that simulate brain activity to pinpoint interconnections between huge amounts of information. Practitioners utilize these systems in diverse financial applications, including anticipating trends, researching markets, identifying fraudulent behavior, and evaluating risks - helping to sift through reams of data with both speed and acuity.

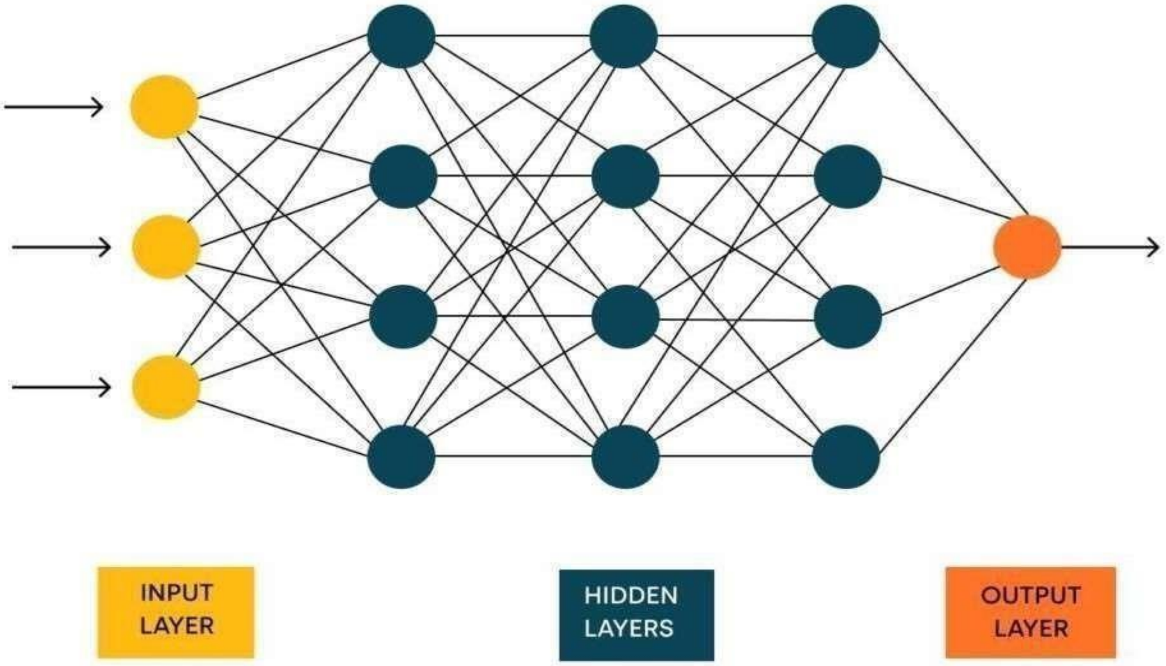


Fig 5.5 Neural Network

Use Case in Diabetes:

Neural networks can be used in diabetes classification to predict whether a patient is diabetic or not based on various features like age, glucose levels, BMI, and insulin levels. By learning complex patterns from large datasets, neural networks can identify intricate relationships between these factors, enabling high-accuracy predictions.

In addition to the points mentioned, neural networks are particularly advantageous for diabetes classification due to their ability to handle high-dimensional data and capture subtle patterns that may not be apparent to traditional models. For example, they can effectively process interactions between multiple features, such as insulin levels and glucose concentrations, that may have a combined effect on diabetes diagnosis.

Chapter 6

EXPERIMENTAL RESULTS

6.1 Evaluation Metrics

Evaluation metrics play a very crucial role in assessing the effectiveness of the machine learning models. They provide quantitative insights that help determine how accurately a model predicts outcomes, and they support model selection and optimization for improved performance.

$$\text{Accuracy} = \frac{TP + TN}{(TP+TN+FP+FN)} \qquad \text{....Eq (6.1)}$$

Where:

TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives

Accuracy is useful when the classes are balanced. However, in cases of imbalanced datasets, it can be misleading, as it may not reflect the model's true performance, particularly when one class is much more frequent than the other.

6.1.1 Precision

Precision, also known as Positive Predictive Value, represents the proportion of positive predictions that are actually correct. It is especially very important when the impact of false positives is significant.

$$\text{Precision} = \frac{TP}{(TP + FP)} \qquad \text{..Eq No.(6.1.1)}$$

6.1.2 Recall

Recall, also called Sensitivity or True Positive Rate, measures the proportion of actual positive instances that the model correctly identifies. It is very vital when the cost of false negatives is high.

$$\text{Recall} = \frac{(TP)}{(TP + FN)} \qquad \text{...Eq No.(6.1.2)}$$

6.1.3 F1-Score

Harmonic mean is used to calculate Precision and Recall. It provides a balance between the two metrics and is especially useful when the class distribution is uneven, or when both false negatives and false positives carry significant consequences.

6.2 Comparative Analysis

The image is a bar chart comparing the performance of different machine learning models used for diabetes prediction.

- Neural Networks show a balanced performance, with both accuracy and F1- macro being relatively close but slightly leaning towards higher accuracy.
- SVM has similar behaviour, with accuracy slightly exceeding F1-macro.
- Random Forest demonstrates excellent performance, with its accuracy being among the highest, closely followed by a high F1-macro score.
- Decision Tree shows a higher gap between the metrics, with accuracy being noticeably higher than F1-macro.
- Logistic Regression achieves the least favourable performance compared to others, as both metrics are lower, although accuracy slightly surpasses F1- macro.

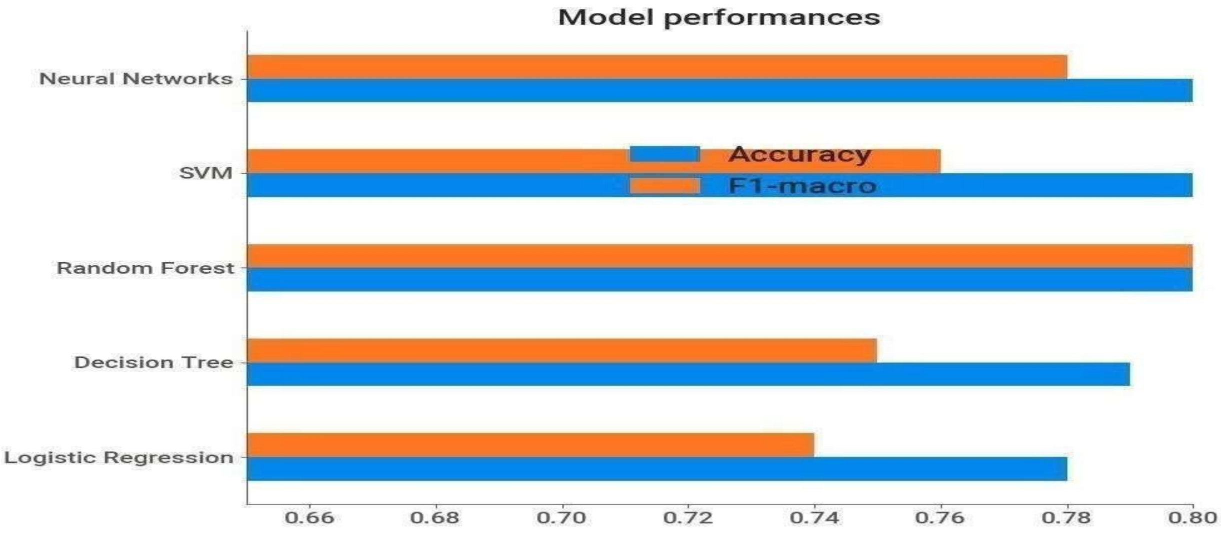


Fig 6.2 Model Performances

6.3 Screenshots

6.3.1 Feature importance in prediction:

- Glucose is the most significant feature, contributing the highest importance to the model.
- BMI (Body Mass Index) and Age are also highly influential features, following glucose in importance.
- Blood Pressure, Skin Thickness, and Diabetes Pedigree Function show moderate contributions.

- Insulin and Pregnancies have relatively lower importance, indicating less influence on the model's decision-making process.

.

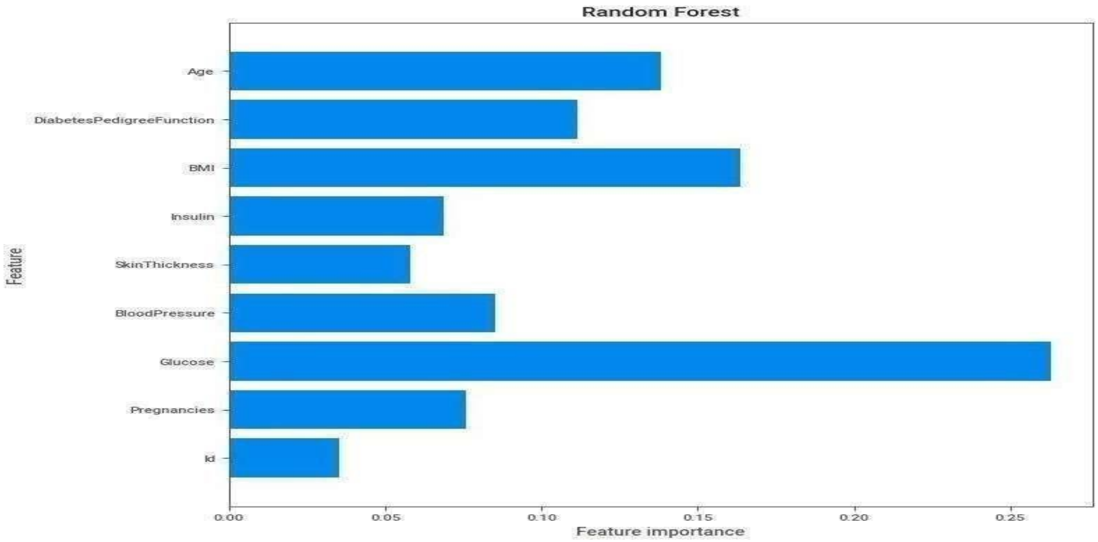


Fig 6.3.1 Feature importance

6.3.2Accuracy in diabetes detection:

- Random Forest achieves the highest accuracy among the models, showcasing its effectiveness in diabetes detection.

- SVM and Decision Tree follow closely, indicating strong predictive capabilities but slightly lower accuracy compared to Random Forest.

- Neural Networks perform moderately well but are outperformed by SVM and Random Forest.

- Logistic Regression has the lowest accuracy, suggesting it is less effective than the other models for this task.

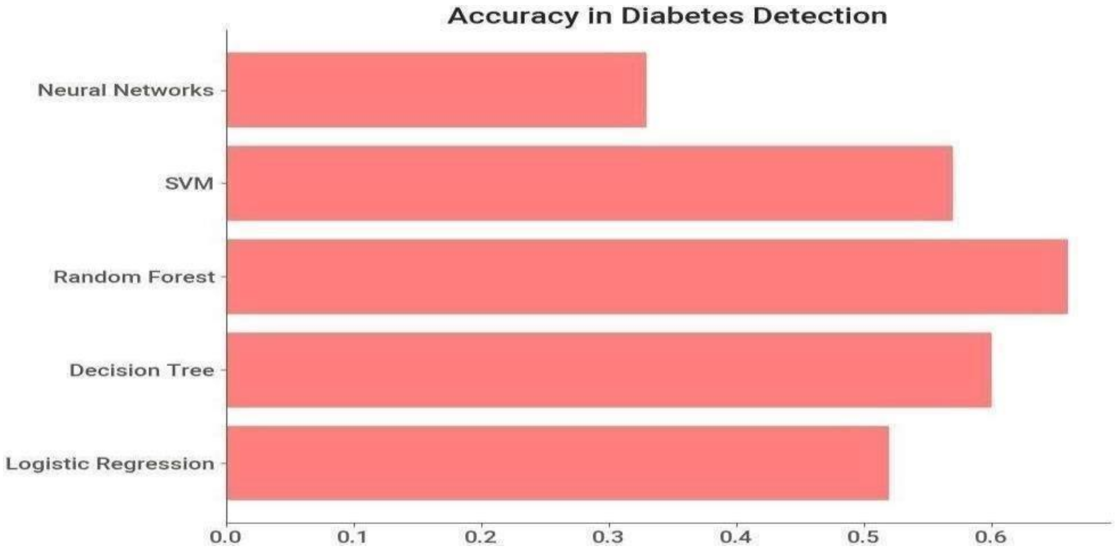


Fig 6.3.2 Accuracy in Diabetes Detection

6.3.3 Normalized Confusion Matrix- Decision Tree:

The matrix indicates the performance of the model in predicting whether a patient has diabetes (label 1) or does not have diabetes (label 0).

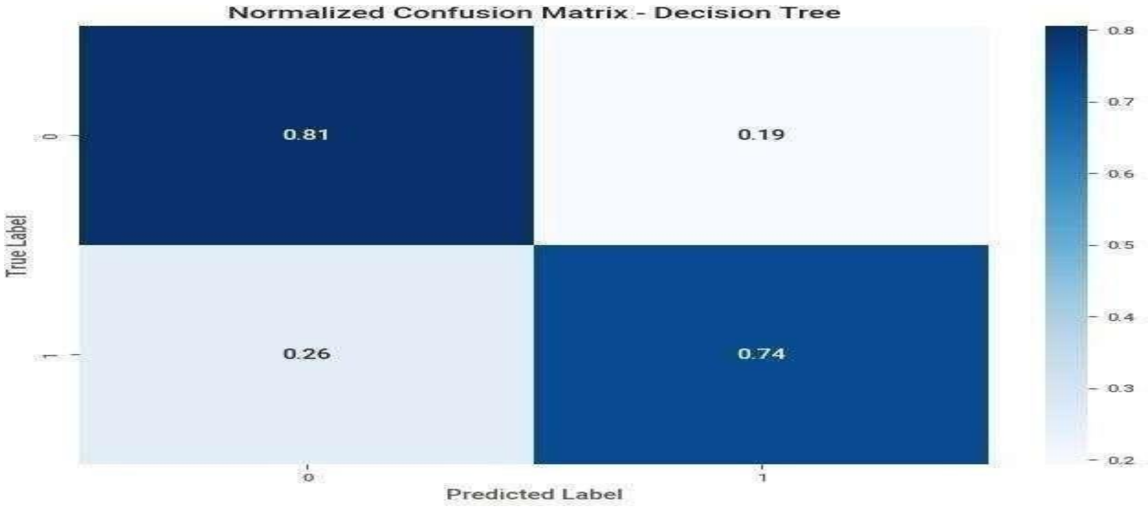


Fig 6.3.3 Normalized Confusion Matrix- Decision Tree

6.3.4 Normalized Confusion Matrix- Random Forest:

This matrix demonstrates the high accuracy of the model, as 99% of non-diabetic cases (true label 0) were correctly classified as non-diabetic, and 99% of diabetic cases (true label 1) were correctly identified as diabetic.

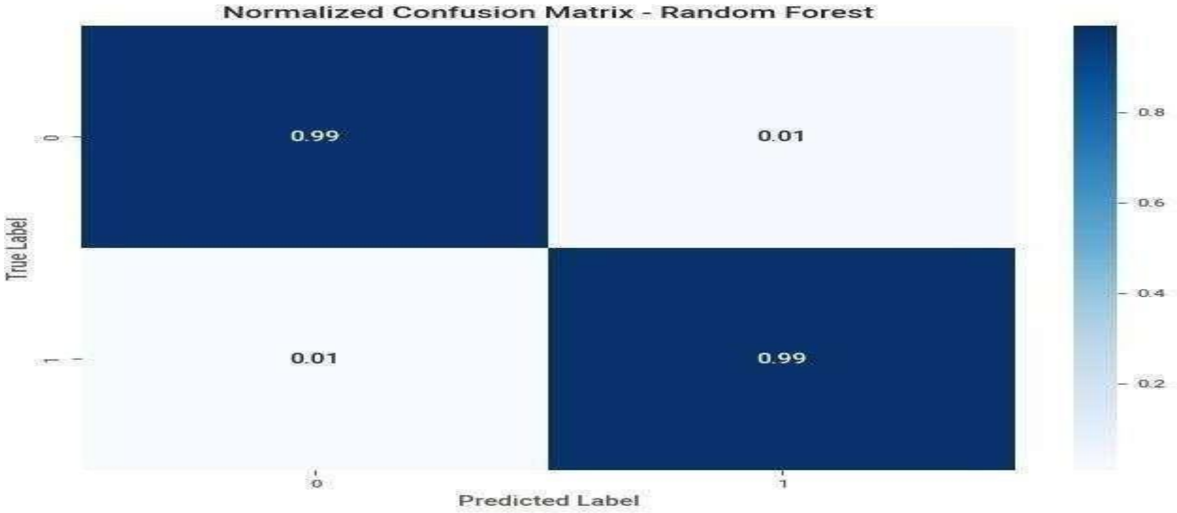


Fig 6.3.4 Normalized Confusion Matrix- Random Forest

6.3.5 Normalized Confusion Matrix- SVM:

The matrix indicates that the model correctly classifies 81% of non-diabetic cases (true label 0) and 74% of diabetic cases (true label 1).

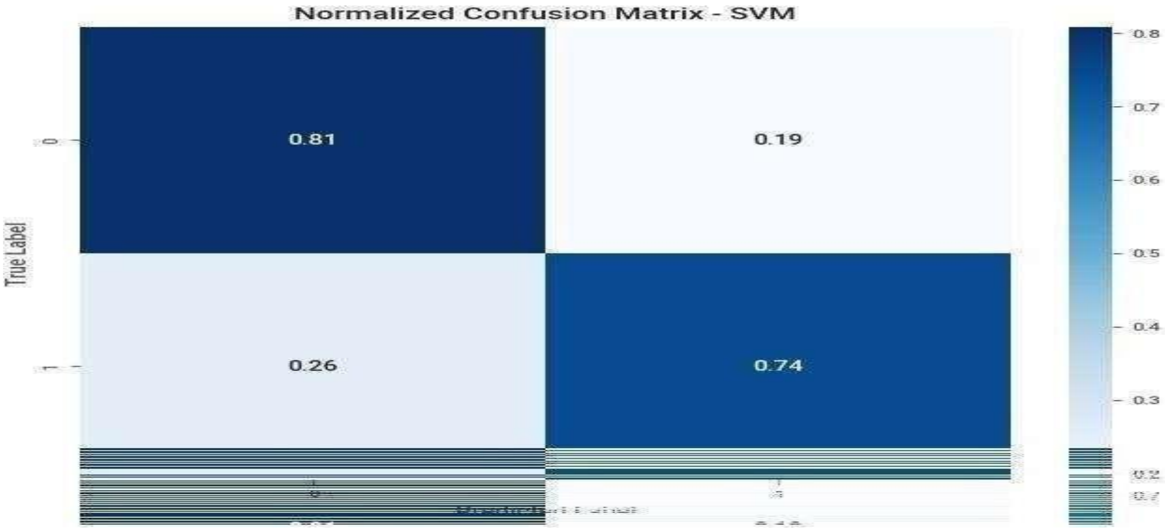


Fig 6.3.5 Normalized Confusion Matrix- SVM

6.3.6 Normalized Confusion Matrix- Neural Networks:

- True Negatives (TN): 88% of the actual non-diabetic patients (true negative class) were correctly predicted as non-diabetic. This indicates a high specificity.

- False Positives (FP): 12% of the actual non-diabetic patients were incorrectly classified as having diabetes.

- False Negatives (FN): 32% of the actual diabetic patients were mistakenly predicted as non-diabetic.

- True Positives (TP): 68% of the actual diabetic patients were correctly classified as diabetic.

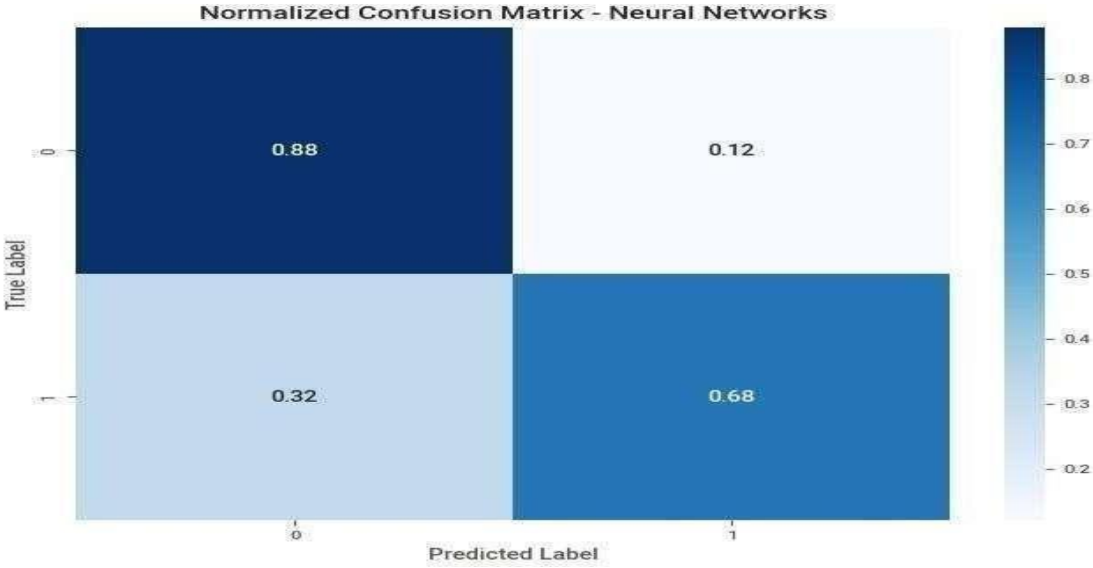
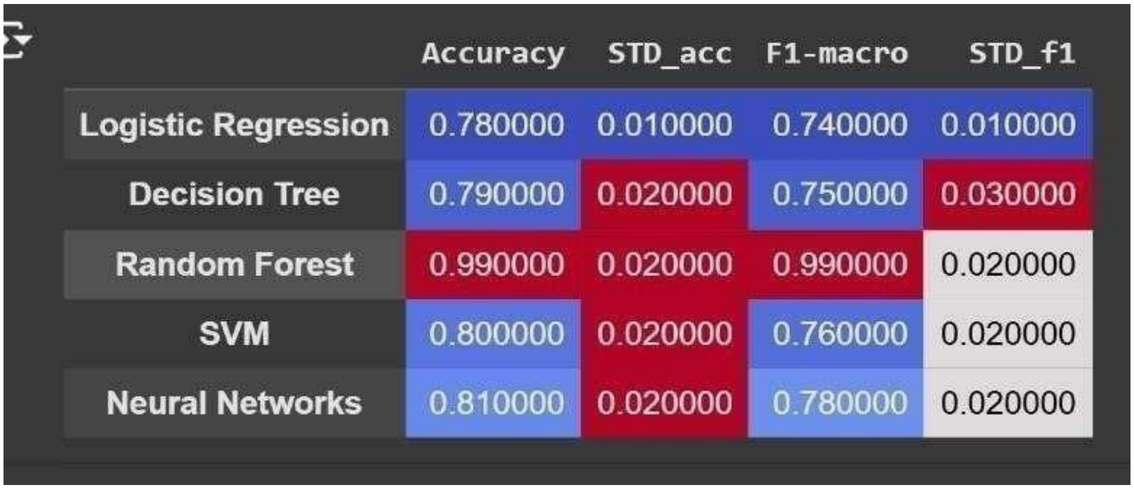


Fig 6.3.6 Normalized Confusion Matrix- Neural Networks

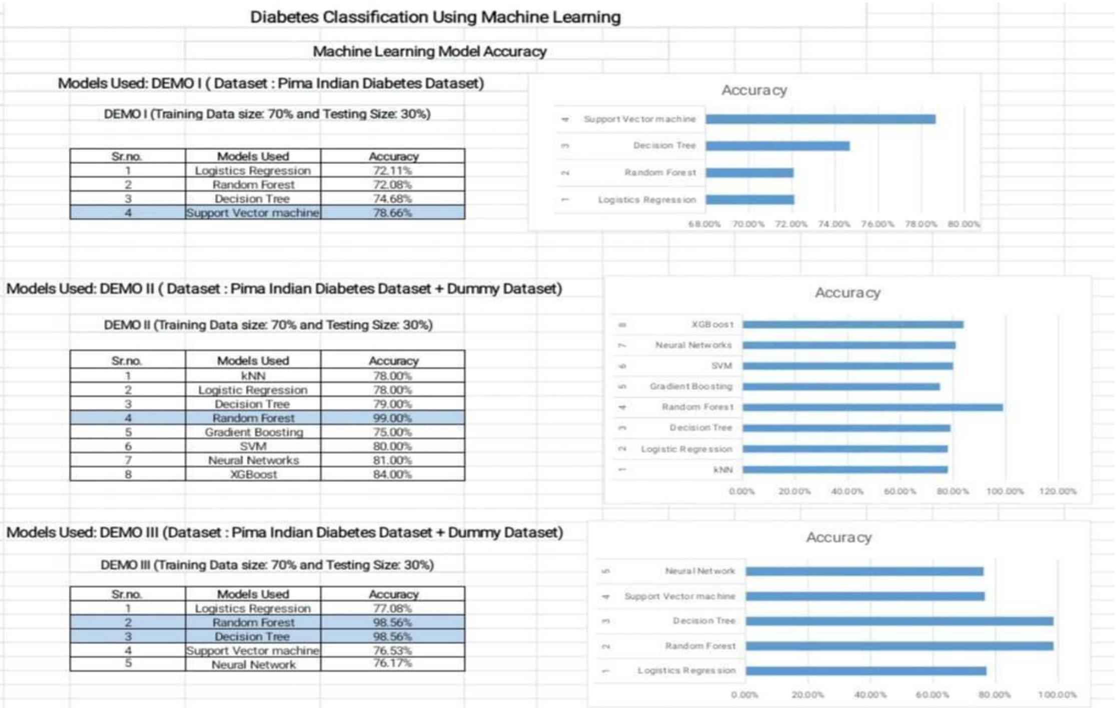
6.3.7 Overall Performance Comparison of all Model:

Random Forest appears with the highest accuracy (0.99) and F1-macro score (0.99). Neural Networks also show results with an accuracy of 0.81 and F1-macro of 0.78.



	Accuracy	STD_acc	F1-macro	STD_f1
Logistic Regression	0.780000	0.010000	0.740000	0.010000
Decision Tree	0.790000	0.020000	0.750000	0.030000
Random Forest	0.990000	0.020000	0.990000	0.020000
SVM	0.800000	0.020000	0.760000	0.020000
Neural Networks	0.810000	0.020000	0.780000	0.020000

Fig 6.3.7 Overall Performance Comparison of all Model



Diabetes Classification Using Machine Learning

Machine Learning Model Accuracy

Models Used: DEMO I (Dataset : Pima Indian Diabetes Dataset)

DEMO I (Training Data size: 70% and Testing Size: 30%)

Sr.no.	Models Used	Accuracy
1	Logistics Regression	72.11%
2	Random Forest	72.08%
3	Decision Tree	74.68%
4	Support Vector machine	78.66%

Models Used: DEMO II (Dataset : Pima Indian Diabetes Dataset + Dummy Dataset)

DEMO II (Training Data size: 70% and Testing Size: 30%)

Sr.no.	Models Used	Accuracy
1	kNN	78.00%
2	Logistic Regression	78.00%
3	Decision Tree	79.00%
4	Random Forest	99.00%
5	Gradient Boosting	75.00%
6	SVM	80.00%
7	Neural Networks	81.00%
8	XGBoost	84.00%

Models Used: DEMO III (Dataset : Pima Indian Diabetes Dataset + Dummy Dataset)

DEMO III (Training Data size: 70% and Testing Size: 30%)

Sr.no.	Models Used	Accuracy
1	Logistics Regression	77.08%
2	Random Forest	98.56%
3	Decision Tree	98.56%
4	Support Vector machine	76.53%
5	Neural Network	76.17%

Fig 6.3.7.1 Overall Performance Comparison of all Model

DEMO I: Dataset: Pima Indian Diabetes Dataset

- Models Tested:

    1. Logistic Regression: 72.11%
    2. Random Forest: 72.08%

3. Decision Tree: 74.68%
4. Support Vector Machine (SVM): 78.66%

- Findings: SVM achieved the highest accuracy (78.66%), while Random Forest had the lowest (72.08%).

DEMO II: Dataset: Pima Indian Diabetes Dataset + Dummy Dataset

- Models Tested:

1. K-Nearest Neighbors (KNN): 75.00%
2. Logistic Regression: 79.00%
3. Decision Tree: 79.00%
4. Random Forest: 81.00%
5. Gradient Boosting: 81.00%
6. Support Vector Machine (SVM): 81.00%
7. Neural Networks: 81.00%
8. XGBoost: 81.00%

- Findings: Many models (Random Forest, Gradient Boosting, SVM, Neural Networks, XGBoost) achieved the same high accuracy of 81%, with KNN being the least accurate (75%).

DEMO III:

Dataset: Pima Indian Diabetes Dataset + Dummy Dataset

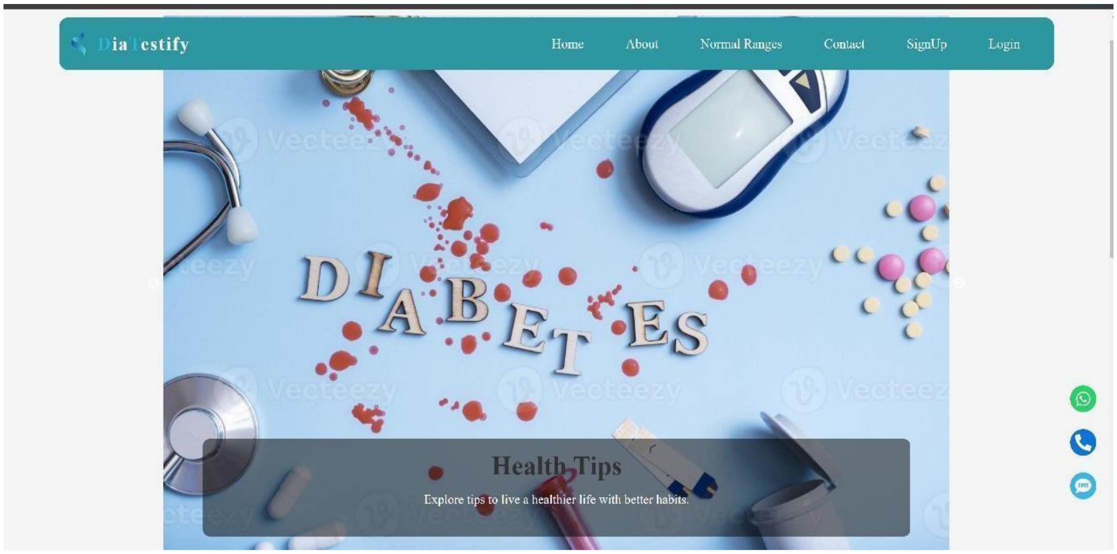
- Models Tested:

1. Logistic Regression: 77.08%
2. Random Forest: 98.84%
3. Decision Tree: 98.56%
4. Support Vector Machine (SVM): 86.71%
5. Neural Network: 76.17%

- Findings: The Random Forest achieved the highest accuracy (98.84%), while Neural Network had the lowest accuracy (76.17%)

6.4 Our website DiaTestify Screenshots:

6.4.1 Home Page:



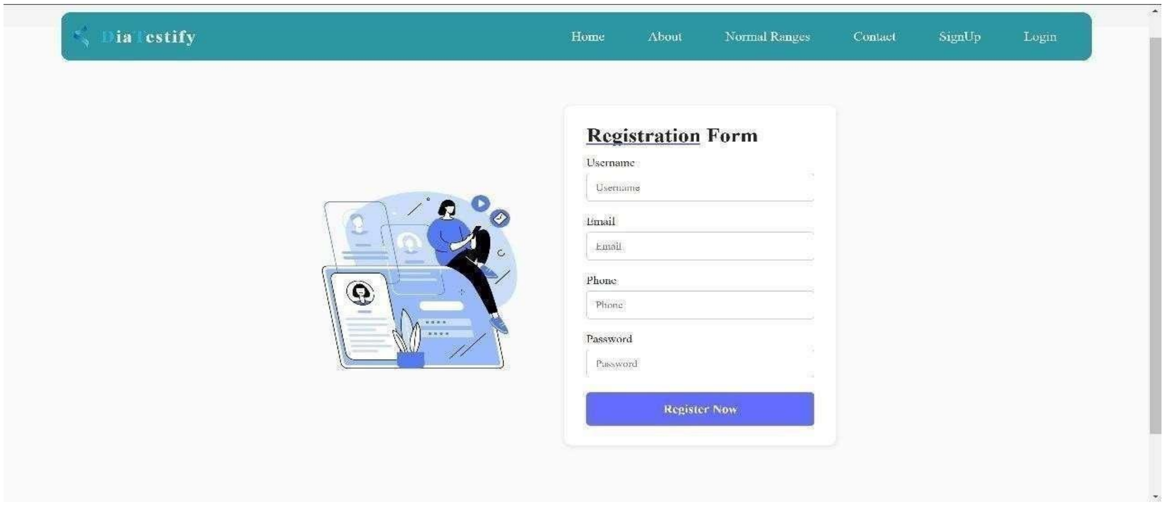
Health Tips

Explore tips to live a healthier life with better habits.

Fig 6.4.1 Home Page

The Home Page of the DiaTestify website serves as the central hub for users seeking information and tools related to diabetes awareness and prediction. The design incorporates an intuitive interface with a clean layout, showcasing visually engaging elements like medical-themed images and a welcoming title to ensure ease of navigation. The homepage provides users with direct access to key features, such as health tips aimed at promoting a healthier lifestyle, information about normal blood sugar ranges, and options for signing up or logging into their personalized accounts.

6.4.2 Registration Form Page:



Registration Form

Username: Username

Email: Email

Phone: Phone

Password: Password

Fig 6.4.2 Registration Form Page

The Registration Form Page in the DiaTestify website allows new users to create their accounts seamlessly. The form collects essential information, such as username, email, phone number, and password, ensuring a secure and personalized experience for each user. With a user-friendly design and clear labels, the registration process is streamlined, enabling individuals to quickly set up their profiles and access the platform's features.

6.4.3 Login Form Page:

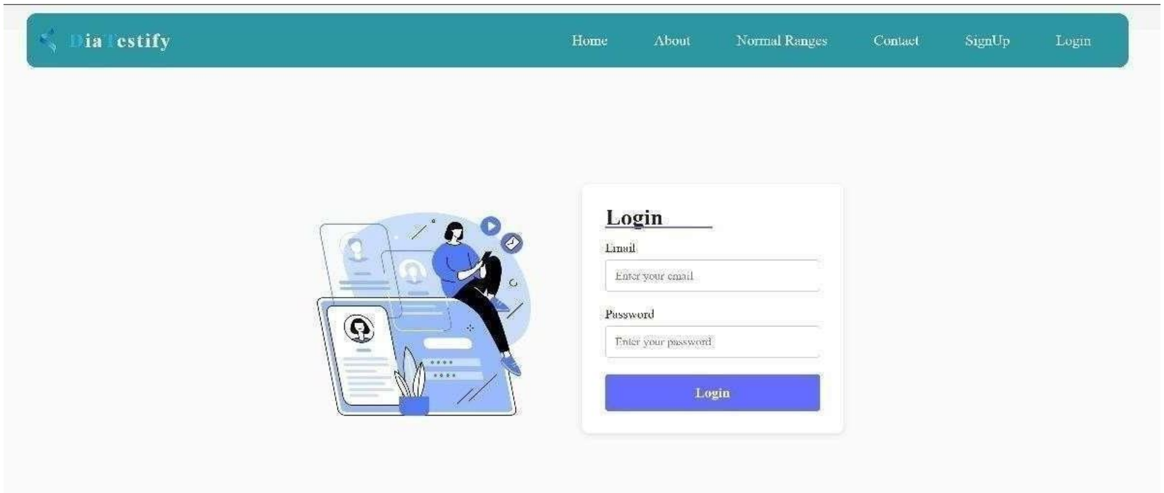
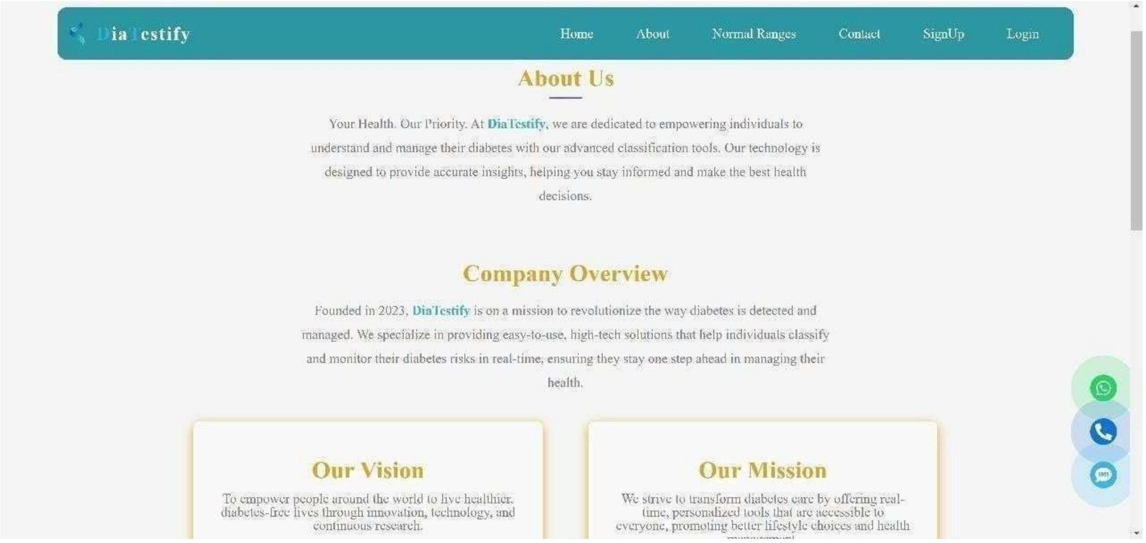


Fig 6.4.3 Login Form Page

The Login Form Page provides a secure gateway for registered users to access their accounts. Featuring input fields for username and password, the page ensures authentication before granting access to the platform. The straightforward design prioritizes user convenience, enabling swift and secure logins to explore the website's tools and services. This page reflects the platform's commitment to ensuring a safe and hassle-free user experience.

6.4.4 About Us Page:



About Us

Your Health. Our Priority. At DiaTestify, we are dedicated to empowering individuals to understand and manage their diabetes with our advanced classification tools. Our technology is designed to provide accurate insights, helping you stay informed and make the best health decisions.

Company Overview

Founded in 2023, DiaTestify is on a mission to revolutionize the way diabetes is detected and managed. We specialize in providing easy-to-use, high-tech solutions that help individuals classify and monitor their diabetes risks in real-time, ensuring they stay one step ahead in managing their health.

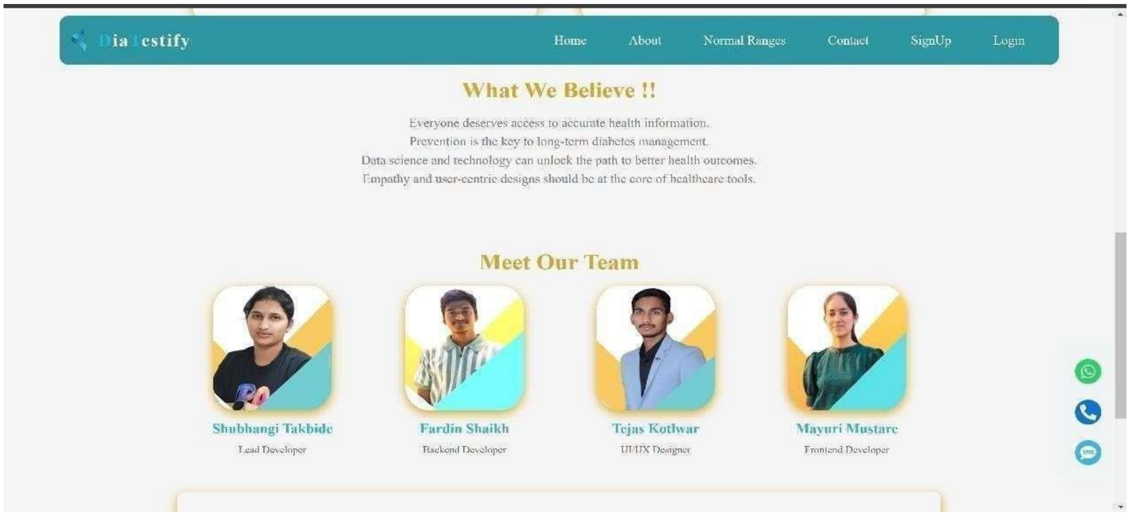
Our Vision

To empower people around the world to live healthier, diabetes-free lives through innovation, technology, and continuous research.

Our Mission

We strive to transform diabetes care by offering real-time, personalized tools that are accessible to everyone, promoting better lifestyle choices and health

Fig 6.4.4.1 About Us Page



What We Believe !!

Everyone deserves access to accurate health information.
Prevention is the key to long-term diabetes management.
Data science and technology can unlock the path to better health outcomes.
Empathy and user-centric designs should be at the core of healthcare tools.

Meet Our Team

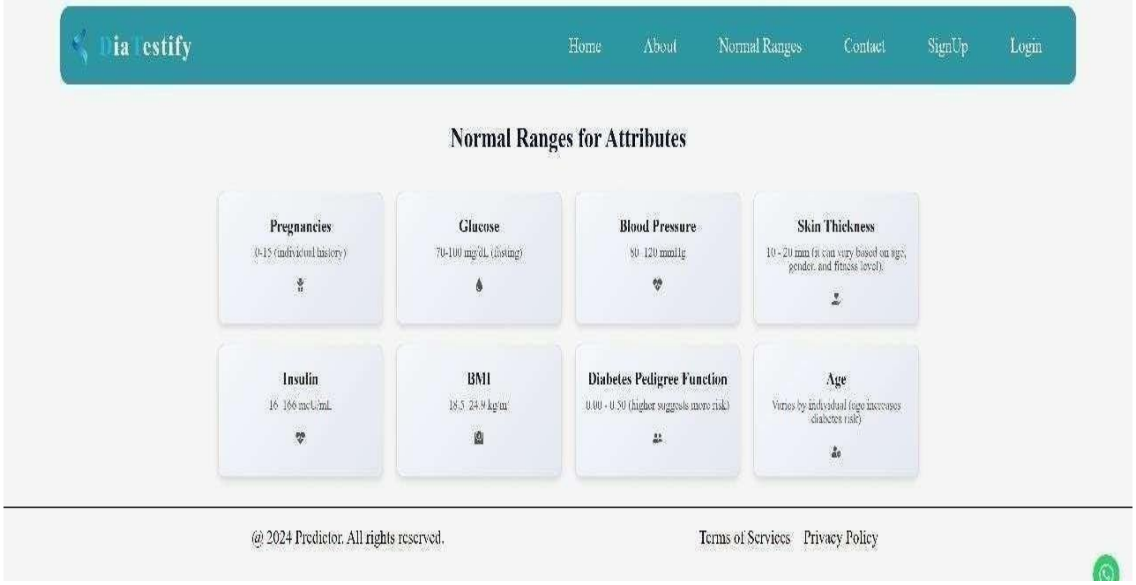
Shubhangi Takbide	Fardin Shaikh	Tejas Kotlwar	Mayuri Mustare
Lead Developer	Backend Developer	UI/UX Designer	Frontend Developer

Fig 6.4.4.2 About Us Page

The About Us Page of the DiaTestify website provides a comprehensive overview of the platform's mission, vision, and commitment to empowering individuals in managing their diabetes effectively. It highlights the website's dedication to using advanced classification tools and real-time solutions for identifying and monitoring diabetes risk.

DiaTestify aims to revolutionize the approach to diabetes detection and management by offering high-tech tools that keep users informed and proactive about their health. The page outlines the company's Vision of enabling people to lead healthier, diabetes-free lives through cutting-edge technology and innovative approaches. Additionally, the Mission emphasizes delivering personalized care and accurate predictions, helping users make confident health decisions.

6.4.5 Normal Ranges For Attributes Page:



Normal Ranges for Attributes

Pregnancies	Glucose	Blood Pressure	Skin Thickness
0-15 (individual history)	70-100 mg/dL (fasting)	80-120 mmHg	10 - 20 mm (it can vary based on age, gender, and fitness level).

Insulin	BMI	Diabetes Pedigree Function	Age
16-166 mcU/mL	18.5-24.9 kg/m²	0.00 - 0.50 (higher suggests more risk)	Varies by individual (age increases diabetes risk)

@ 2024 Predictor. All rights reserved.

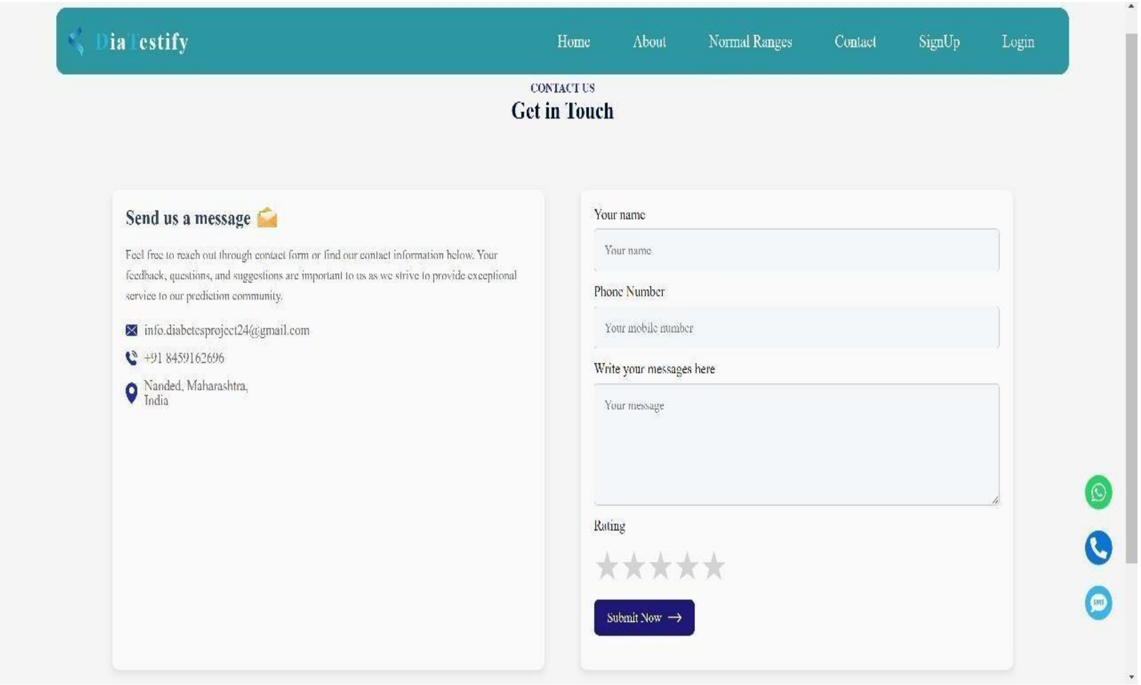
Terms of Services Privacy Policy

Fig 6.4.5 Normal Ranges For Attributes Page

"Normal Ranges for Attributes" along with individual cards for each health parameter. These include Pregnancies (0-15, based on individual history), Glucose (70-100 mg/dL for fasting), Blood Pressure (80-120 mmHg), Skin Thickness (10-20 mm, varying by age, gender, and fitness level), Insulin (16-166 mU/mL), BMI (18.5-24.9 kg/m²).

Diabetes Pedigree Function (0.00-0.50, where higher values suggest greater risk), and Age, which varies based on individual factors, with older age increasing diabetes risk.

6.4.6 Contact Page:



CONTACT US

Get in Touch

Send us a message

Feel free to reach out through contact form or find our contact information below. Your feedback, questions, and suggestions are important to us as we strive to provide exceptional service to our prediction community.

info.diabetesproject24@gmail.com

+91 8459162696

Nanded, Maharashtra,
India

Your name: Your name

Phone Number: Your mobile number

Write your messages here: Your message

Rating

Fig 6.4.6 Contact Page

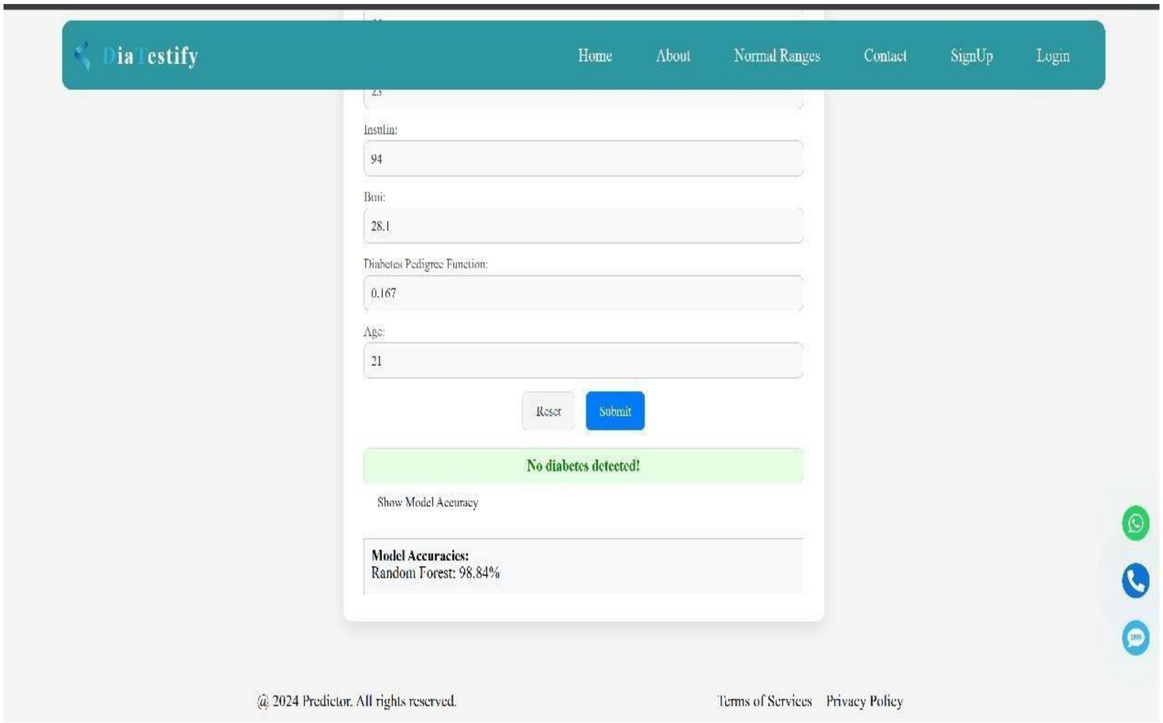
The image depicts a Contact Us page for a platform called DiaTestify, which appears to focus on diabetes-related services or awareness. The page is designed to facilitate easy communication between users and the platform. It is divided into two sections. The left section, titled "Send us a message," provides essential contact details, including an email address (info.diabetesproject24@gmail.com), a phone number (+91 8459162696), and a location (Nanded, Maharashtra, India).

A brief message encourages users to reach out for queries, feedback, or suggestions. The right section features a user-friendly form where visitors can input their name, phone number, and a message, along with a five-star rating system to evaluate the service. Additionally, the page includes a "Submit Now" button to send the message.

6.5 Final Result:

The DiaTestify website is an innovative platform designed to assist users in assessing their risk of diabetes through a simple and interactive interface. By leveraging the Pima Indian Diabetes Dataset, a reputable dataset widely used in machine learning for diabetes prediction, the platform ensures robust model training and reliable predictions. Users are required to input key health metrics such as glucose levels, blood pressure, BMI, insulin levels, age, and other relevant parameters. Once submitted, the machine learning model processes these inputs to classify the likelihood of diabetes.

6.5.1 Prediction of No Diabetes:



Insulin: 94

Bmi: 28.1

Diabetes Pedigree Function: 0.167

Age: 21

No diabetes detected!

Show Model Accuracy

Model Accuracies:
Random Forest: 98.84%

@ 2024 Predictor. All rights reserved.

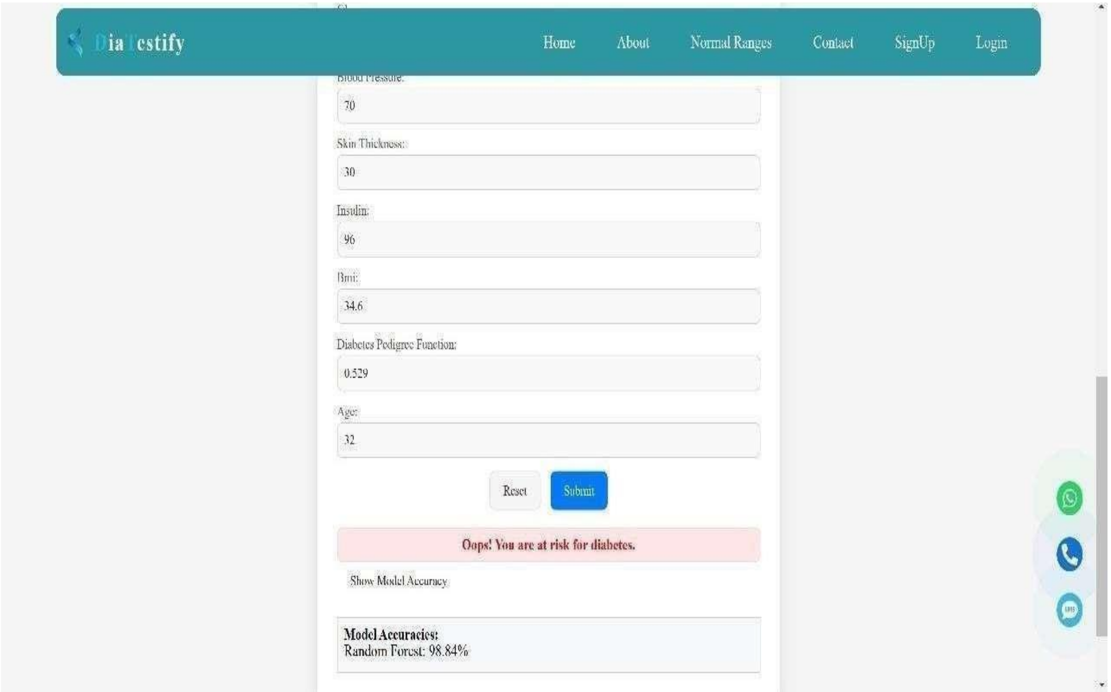
Terms of Services Privacy Policy

Fig 6.5.1 Prediction of No Diabetes

The image shows a diabetes prediction interface on the DiaTestify platform, where users can input medical data to check for the likelihood of diabetes. The page includes a form for entering key health metrics such as glucose levels, blood pressure, insulin levels, BMI, diabetes pedigree function, and age. At the bottom of the form, there are two buttons: Reset, which clears the input fields, and Submit, which processes the input data.

Upon submission, the system displays a result indicating whether diabetes is detected or not; in this case, it shows "No diabetes detected!" in a green-highlighted box, indicating a positive outcome for the user. Below this, the model's performance accuracy is displayed, stating that the Random Forest model used has an accuracy of 98.84%.

6.5.2 Prediction of Risk of Diabetes:



Blood Pressure:
70

Skin Thickness::
30

Insulin:
96

Bmi:
34.6

Diabetes Pedigree Function:
0.529

Age:
32

Oops! You are at risk for diabetes.

Model Accuracies:
Random Forest: 98.84%

Fig 6.5.2 Prediction of Risk of Diabetes

The screenshot displays a web application interface named DiaTestify, which is designed to predict the risk of diabetes based on user inputs. Below this, there is a form where users can input health-related parameters such as: Blood Pressure, Skin Thickness, Insulin, BMI (Body Mass Index), Diabetes Pedigree Function, Age.

After filling out the form, users can click on the Submit button to receive a prediction. The example in the screenshot shows a warning message: "Oops! You are at risk for diabetes," indicating that the inputted data suggests a high likelihood of diabetes.

Additionally, there is an option to view the model's accuracy, displayed as "Model Accuracies: Random Forest: 98.84%", suggesting that the Random Forest algorithm is being used for prediction and has high accuracy.

The interface also includes a Reset button to clear the form and floating icons for quick communication via WhatsApp, email, or SMS. The design appears user-friendly and informative, catering to individuals seeking a quick assessment of their diabetes risk.

CONCLUSION

By analyzing various classification techniques proves insightful for diabetes prediction. Among the evaluated methods - Random Forest, Logistic Regression, Support Vector Machines, Neural Networks, and Decision Trees - Random Forest emerged most accurate relative to Decision Trees. Nonetheless, further optimizing Random Forest intrigues us. By filtering out nonessential, noninformative attributes and retaining solely the most diagnostic for classification, we aim to enhance its performance. A targeted selection of distinguishing features could potentially sharpen predictions and refine understanding of this health condition. A revised model incorporating only meaningfully discriminating variables may uncover nuanced relationships and subtle patterns previously obscured. Such refinement hopefully illuminates new perspectives on this important topic.

To bolster the effectiveness of our diabetes classification system, we are integrating feature selection techniques like Recursive Feature Elimination (RFE) and also used the advanced evaluation techniques like Confusion Matix and K-Fold Cross Validation. These methodologies aid in pinpointing and retaining the most pertinent attributes, consequently trimming down the datasets dimensionality and enhancing the model performance. By honing in on the most relevant features, our objective is to mitigate overfitting, enhance model interpretability, and ultimately achieve heightened accuracy and resilience in diabetes classification.

REFERENCES

[1] N.Kumaravelu, R.Subaramanian, “Medical Decision Making: Machine Learning for Diabetes Prediction”, CRC Press; 1st edition (2021), ISBN-10:0367769134.

[2] Dr. Sachi Nandan Mohanty, “Healthcare Analytics Using Machine Learning and Deep Learning”, Wiley; 1st edition (2021), ISBN-10:1119792260.

[3] Dr. M. S. Roobini, C. A. Daphine Desona Clemency, Aishwarya D, “Machine Learning for Type 2 Diabetes Classification”, Lambert Academic Publishing, ISBN-13:978-620-7-44767-1.

[4] G. G. Rajput, Ashvini Alashetty, “Diabetes Classification Using ML Algorithms”, Springer, ISBN-978-981-99-1623-8.

[5] A. K. Maurya, S. K. Singh, P. K. Gupta, “Classification of Diabetic Disorder using Machine Learning Approaches”, 2023 International Conference on Artificial Intelligence and Smart Systems (ICAIS), ISBN-979-8-3503-9784-0.

[6] S. S. R. K. Prasad, K.V. S. Sairam, K. S. R. Anjaneyulu, “Diabetes Prediction using Machine Learning Classification Algorithm”, 2022 International Conference on Smart Technologies and System for Next Generation computing (ICSTSN), ISBN-978-1-6654-7884-7.

[7] Kaggle, https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database,“Pima Indians Diabetes Dataset”.

[8] Analytics Vidhya, https://www.analyticsvidhya.com., "Diabetes Prediction using Machine Learning".

[9] YouTube, https://www.youtube.com., "Video Tutorials on Converting Flask to Node.js".