

# Machine Learning Based Diabetes Status Classification and Web Based Health Assessment Framework with FastAPI Deployment

Lavanya R<sup>1</sup> , \*Nithyanandh S<sup>2</sup> , and Anita Sofia VS<sup>3</sup> 

<sup>1</sup> PG Scholar, Department of Computer Application, PSG College of Arts & Science, Coimbatore, Tamilnadu, India

<sup>2,3</sup> Associate Professor, Department of Computer Application, PSG College of Arts & Science, Coimbatore, Tamilnadu, India

\*Correspondence should be addressed to Nithyanandh S; [nknithu@gmail.com](mailto:nknithu@gmail.com)

**Received:** 21 Sep 2026; **Revised:** 3 Oct 2026; **Accepted:** 16 Oct 2026; **Published Online:** 26 Oct 2026

Copyright © 2026 \*Nithyanandh S et al. This is an open-access article distributed under the [Creative Commons Attribution 4.0 International License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**ABSTRACT-** The classification of diabetes from structured health records can be used to provide health awareness when model outputs are given with the right limitations. In this paper, we introduce the first Diabetes status assessment framework that integrates a comparative machine learning approach and an interactive application workflow, namely "DiaPredict". The original dataset contained 100,000 records. After removing 3,854 duplicate records, 96,146 unique records remained, comprising 87,664 non-diabetes records (91.18%) and 8,482 diabetes records (8.82%). All eight input attributes (gender, age, hypertension, heart disease, smoking history, body mass index (BMI), glycated hemoglobin (HbA1c), and blood glucose) are processed through a common pipeline and evaluated with Logistic Regression, Random Forest, Extra Trees, and XGBoost. Split of 80 to 20, with state 42 being the stratified portion and random. XGBoost obtains accuracy of 97.07 percent, precision of 95.43 percent, recall of 70.17 percent, F1 score of 80.87 percent and receiver operating characteristic area under the curve (ROC AUC) of 97.56 percent on the held-out test partition. Selected pipeline is coupled with a FastAPI backend and React / TypeScript frontend, SQLite persistence, assessment history, trend visualization, rule based educational insights and PDF reporting. This is a study on diabetes status classification and not future risk prediction as diabetes status was recorded in the target label and not when it occurs in the future. The application thus provides health awareness model probabilities as estimates rather than clinical diagnoses.

**KEYWORDS-** Diabetes Classification, FastAPI, Health Assessment, Machine Learning, XGBoost.

## I. INTRODUCTION

A diabetes mellitus is an ongoing condition of high blood sugar that leads to various problems over time. The evaluation usually includes various demographics, physiologies, and lifestyles that are relevant to it is well suited for machine learning based analyses. Recently, logistic regression models, boosting techniques, and tree-based ensemble methods are being studied for use in

diagnosing diabetes and predicting its risk factors; however, current research emphasizes issues like data quality problems, imbalance between classes, lack of proper validation procedures, and poor interpretability of these models [1], [2], [7], [12].

Performance of a model for predicting diabetes depends upon its data set properties; definition of targets; available features; and methodological approach used during experiments. There is a lot of variability among these models' predictions according to a recent systematic review and meta-analysis on diabetes prediction models; also, there are few studies that provide evidence for their effectiveness outside of clinical trials [3]. Recent research has shown good performance of XGBoost and other ensemble methods on structured diabetes related data sets [4] [5]. The study shows how important it is to compare several models within an identical experiment instead of just looking at their outcomes from separate experiments using various data sets and methods for evaluating them.

This study uses the publicly available Diabetes Prediction Dataset, which initially contained 100,000 records and nine columns, including eight input features and one target variable. During data cleaning, 3,854 duplicate records were identified and removed, leaving 96,146 unique records. The cleaned dataset contained 87,664 non-diabetes records (91.18%) and 8,482 diabetes records (8.82%), with no missing values [18]. One major difference between this research paper and other papers about predicting future cases of diabetes is how they define their target variables. Target variable is defined as an indicator for current state of diabetes diagnosis instead of predicting incidence rate at some point in time. Thus, this paper deals with a binary diabetic status prediction problem through machine learning techniques. While it shows models' probabilities via an estimated-risk classification for better comprehension by users, these results are not considered clinically accurate individual risks nor do they predict any future occurrence of diseases. For connecting comparison of models to real-world applications, DiaPredict incorporates an appropriate machine learning algorithm within its complete stack website service provider software product package. React

and TypeScript are used for developing a web application whereas Fast API serves as an endpoint to provide predictions from it. SQLite is an application of persistence while a machine learning model training process can be serialized to ensure consistency during predictions. An authenticated user may enter their medical history for evaluation purposes; they will be able to view past test results; they can also read about BMI, HbA1c levels and blood sugar levels; they have access to general health data; and they can produce a printable PDF report of this information. It is an application for health awareness and statistics rather than replacing complete medical evaluations by doctors.

Key findings and contributions to this paper include: (1) Comparative analysis between logistic regression, random forest, extra trees, and XGBoost models trained on same data set via same preprocessing procedure under hold out testing conditions; (2) Selection process for deployment model according to defined criteria such as ROC AUC, F1 score, and recall values; (3) Integration of chosen model with an online service by means of FastAPI technology; and (4) Combination of predictions with authentication records, descriptive statistics tables, educational details and PDF reports. This article is about what we did before writing it; our research proposal, methodological approach to study design and execution, experimentation procedure and outcome analysis, interpretation of data, conclusions drawn from that process.

## II. LITERATUR REVIEW

Machine learning research on diabetes involves gathering information from different types of datasets; defining objectives and selecting relevant variables or attributes for model building purposes. Variations make it difficult to compare results from different research projects since they highlight the importance of knowing exactly what methods were used during experimentation here. Masood et al. synthesized studies on AI-driven approaches to diabetes prediction and emphasized the importance of considering the data modality and intended application context [1]. Khokhar et al. identified several challenges in AI-based diabetes prediction, including data quality, class imbalance, interpretability, and validation [2]. Also, Li et al [3] showed a lot of differences among these models' accuracy for predicting diabetes risk between different groups and environments which shows that there are issues with how we measure their effectiveness. This study shows that differences among datasets and models can affect their performance significantly.

Also, recent research has been done on various techniques of machine learning to predict it. Rafie et al. [4] combined XGBoost and CatBoost with feature selection, class balancing, and explainability techniques for diabetes prediction. Zhang et al. [5] investigated focal active learning together with machine learning models to address minority-class learning and reported improved recall within their experimental setting. The research shows there are various methods of developing models such as preparing datasets, dealing with classes and choosing algorithms which makes it hard for us to compare them directly from independent experiments.

Variable selection and interpretation are also important factors when conducting research about diabetes patient

statistics analysis for predictive modeling. Navarro-Cerdán et al. [6] investigated diabetes diagnosis and prognosis using tailored heterogeneous feature groups and showed that different feature configurations can produce different predictive settings and performance characteristics. Nie et al. [7] developed a model for early diabetes detection and risk stratification using National Health and Nutrition Examination Survey (NHANES) data, demonstrating the application of population-level health information to diabetes-related prediction. Barman et al. [8] investigated interpretable machine learning for diabetes risk prediction using India's National Family Health Survey 5 (NFHS-5) dataset and emphasized the importance of interpretability when applying machine learning to large-scale population health data. The research shows how feature selection, demographic attributes, and model explainability affect building up to interpreting diabetic data science algorithms' results.

Predictions differ from other types of studies by prediction task. Matboli et al. [9] investigated machine learning-based stratification across healthy, prediabetes, and type 2 diabetes-related states, demonstrating the application of machine learning to multiple disease-related categories and progression-oriented classification. Li et al. [10] used data from the China Health and Nutrition Survey (CHNS) to investigate prediabetes risk and reported XGBoost as an effective classifier among the models evaluated in their experimental setting. The research shows these words like "diabetes prediction," "risk of diabetes prediction," "diagnosis for diabetes", "classification by disease type", and "progression to complications from diabetes.". Therefore, models' effectiveness can be measured by their correlation to some specific outcome measure and time frame for prediction purposes if necessary. Classification models are used by doctors to diagnose current levels of diabetes while prediction is done by predicting future risk factors for developing this disease among patients.

Research on diabetes prediction is based upon various kinds of biological data as well as demographics. Khan et al. [11] examined the use of genetic biomarkers together with machine learning techniques for predicting diabetes, illustrating how diabetes-related prediction research can incorporate biological information beyond conventional demographic and clinical variables. Lugner et al. [12] used machine learning analysis of UK Biobank data to identify important predictors of type 2 diabetes, further demonstrating the relevance of feature characteristics when developing diabetes prediction models. The findings show that these items need not necessarily be treated equally for each model since they may vary depending on what you are trying to predict from them.

One major issue is to move from experimental machine learning model development towards real-world application usage by humans. Dharmarathne et al. [13] demonstrated a diabetes diagnosis system incorporating a self-explainable interface. Huang et al.'s [14] research involves using both genetics and medical imaging data to assess the risk of developing type 2 diabetes. Srinivasu et al.'s [15] work is a model of explainable AI to predict diabetes. The research shows there are various types of datasets used by these models for predicting patients' health status due to diabetes along with their interface design features such as explanation mechanisms among others. Thus, practical application should also take into

account user interaction with models' results besides accuracy rate of prediction.

Katiyar et al. [16] reviewed recent machine learning and deep learning approaches for diabetes detection and demonstrated the wide range of computational methods explored in this field. Mohsen et al. [17] conducted a scoping review of artificial intelligence-based methods for diabetes risk prediction and highlighted substantial variation in datasets, machine learning approaches, and evaluation strategies across existing studies. These papers are also useful for understanding how to interpret findings from various research on predicting diseases through models based upon data sets, objectives of predictions, features used by algorithms etc.

This paper [18] uses a public Diabetes Prediction Dataset that contains information about demographics, health statuses, lifestyles and medical measurements. This paper uses a dataset to classify patients' current state as either having or not having diabetes instead of predicting their likelihood of developing it later on.

Therefore, there are many studies on how machine-learning algorithms can be applied for detecting, classifying and predicting risk factors of diabetes as well as their classification and interpretability models. However, it does not propose a new diagnostic method or claim that it would be useful to doctors. However, this article focuses more on a neutral analysis of logistic regression models such as random forests, extra-trees and xgboost using same dataset for training purposes before applying them to new problems via different methods like cross validation techniques or other statistical tests. Once we choose a classifier to use for classifying data on our site that includes security features such as secure assessments, archiving historical documents, showing trend analysis statistics related to healthcare topics, providing medical education materials through web pages in PDF format etc. then it would become available here. Comparative machine-learning evaluation is linked with practical application of software engineering rather than being confused by statistics for making decisions about medicine. Therefore, DiaPredict serves primarily to

provide information on patient health status through statistics instead of being used by clinicians for making diagnoses or forecasting diseases in the future.

### III. PROPOSED METHODOLOGY

#### A. System Architecture

DiaPredict is an integrated software program for medical diagnosis which combines artificial intelligence algorithms to analyse patient data through its interface on this website. It has four levels like a model; Presentation layer, APIs Layer and Application-level Machine Learning Level Data Reporting. It separates out user interactions from requests for processing by models of predictions on datasets; it also allows these components to function separately. The presentation layer is implemented using React and TypeScript and provides the user-facing interface for authentication, health assessment, dashboard visualization, assessment history, results, and analysis. Validation on client side gives instantaneous response to incorrect/out of range data entries. FastAPI provides an interface to this service which includes a web-based login form, forms of evaluation submissions, access logs, analytics reports on dashboards, analytical tools for reporting purposes. Validation backend is a second level of verification for models after they have been validated by their respective APIs.

Machine Learning Layer is a serialized pre-processing pipeline along with an XGBoost model. Preprocessing transforms and classifiers are combined for consistency between training and testing phases of model development process. Eight inputs are fed to this model for making predictions of whether they belong to class 'positive' or not. SQLite is used by Data and Reporting Layer to store users' information as well as assessments while Report Lab generates reports in PDF format. Every evaluation belongs to an authenticated account where it saves measured data along with its results from a model. It enables retrieval of past data as well as a description for an existing model without having to run it again.

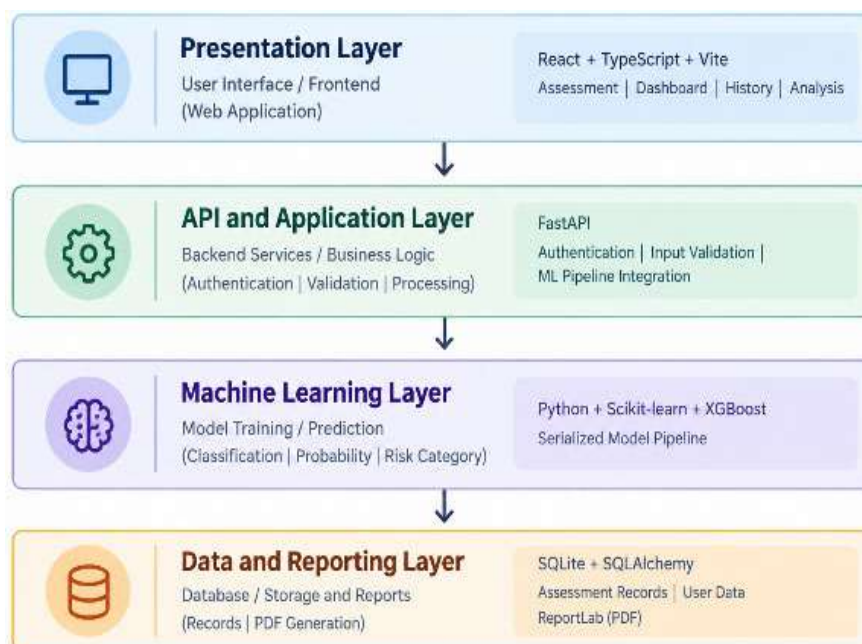


Figure 1: Layered architecture of DiaPredict

In above Figure 1 illustrates the layered architecture of DiaPredict, showing how the React and TypeScript frontend communicates with the FastAPI backend, which connects to the machine-learning pipeline and the data storage and reporting components. This modular architecture separates user interaction, request processing, model inference, and data management.

### B. Proposed Model

The proposed machine learning model performs binary classification using eight health-related input variables: gender, age, hypertension, heart disease, smoking history, BMI, HbA1c level, and blood glucose level. Categorical variables are transformed using one-hot encoding, while numerical variables are standardized before being supplied to the classifier. For a numerical feature  $x_j$ , standardization is performed as,

$$x'_j = \frac{x_j - \mu_j}{\sigma_j} \quad (1)$$

where  $x_j$  is the original feature value,  $\mu_j$  is the training-set mean, and  $\sigma_j$  is the training-set standard deviation. The transformation parameters are learned only from the training data and retained within the serialized preprocessing pipeline. The transformed feature vector is then processed by the XGBoost classifier. The model produces a decision score  $\hat{z}$ , which is converted into the positive-class probability using the sigmoid function:

$$P(y = 1|x) = \frac{1}{1 + e^{-\hat{z}}} \quad (2)$$

where,  $P(y = 1|x)$  represents the estimated probability of the positive class for the input vector  $x$ . The XGBoost model is trained as an additive ensemble of decision trees. At iteration  $t$  is the objective can be expressed as,

$$Obj^{(t)} = \sum_{i=1}^n L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (3)$$

where,  $L$  represents the classification loss and  $f_t$  represents the newly added tree. The regularization term is expressed as,

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (4)$$

where,  $T$  represents the number of leaves,  $w$  represents leaf weights, and  $\gamma$  and  $\lambda$  control model complexity. The classifier output is converted into application-level presentation categories. For a positive prediction, a probability of at least 0.70 is displayed as Higher Estimated Risk, while a probability below 0.70 is displayed as Moderate Estimated Risk. For a negative prediction, a probability of at least 0.35 is displayed as Moderate Estimated Risk, while a probability below 0.35 is displayed as Lower Estimated Risk. These thresholds are application-defined presentation rules and are not clinically validated decision boundaries. This separation ensures that the machine learning model produces the statistical classification output, while the application layer is responsible for presenting that output in an understandable form. The probability and resulting category are therefore not interpreted as a clinical diagnosis or independently validated individual risk estimate.

### C. Assessment Workflow

The assessment workflow begins after successful user authentication. The user provides these eight parameters to a trained model: gender, age, hypertension, heart disease, smoking history, BMI, HbA1c level, and blood glucose level. The client side validates immediately upon receiving an error message from a server whereas FastAPI is

responsible to validate data on its own before making predictions. Validation is done by sending an assessment request from this site to a FastAPI model server for processing purposes. Back-end links a request to an authenticated user for submitting feature requests that are forwarded by them through the inference engine. Serialization of a pre-processing process is used for transforming an input prior to feeding into an XGBoost model. Prediction results along with a positive class likelihood will be passed on by this model for use at an application level.

Assessment records, measurement data, predictions, probabilities at an application level of users' associations along with timestamps will be saved to a database called SQLite. Afterwards it will be available for viewing from an evaluation report, a dashboard, and analytical tools. Analysis module produces descriptive trend data for BMI, HbA1c levels of blood sugar, probability score and risk category scores. The graphs shown here are just a record of what we have observed; they do not imply causality nor indicate any stage of an illness process.

Nutrition and Health Insights module is separate to Prediction Service for providing general education information. In this way, it stores reports as files in PDF format through report lab while not changing any of them from their current state.

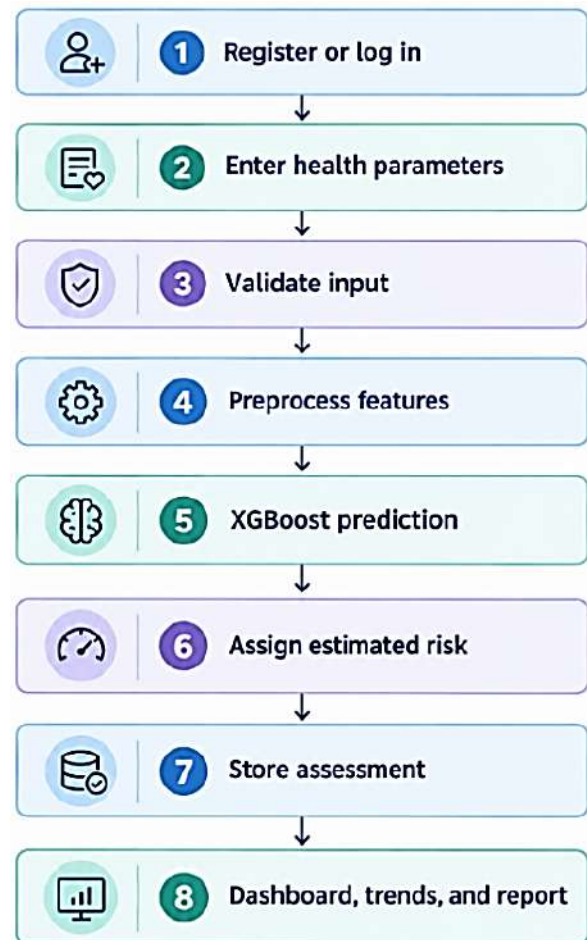


Figure 2: End to end assessment workflow.

Figure 2 presents the end-to-end assessment workflow, beginning with user authentication and health-parameter entry, followed by input validation, preprocessing, and

XGBoost inference. The predicted class and probability are then used to determine the application-level presentation category, and the assessment details are stored for subsequent history viewing, dashboard analysis, and report generation.

#### D. Algorithm

##### Algorithm 1: DiaPredict Assessment and Prediction

###### Input:

User health parameters  $X$   
 Trained preprocessing pipeline  $P$   
 Trained XGBoost classifier  $M$

###### Output:

Prediction  $Y$   
 Probability  $P$   
 Application-level risk category  $R$

1. Authenticate user.
2. Accept the eight health parameters  $X$ .
3. Validate the input values.
4. If validation fails, return validation error.
5. Apply preprocessing pipeline  $P$  to  $X$ .
6. Pass the transformed features to XGBoost model  $M$ .
7. Obtain binary prediction  $Y$ .
8. Obtain positive-class probability  $P$ .
9. Assign application-level risk category  $R$ .  
 If  $Y = 1$  and  $p \geq 0.70$  :  $R$  = Higher Estimated Risk  
 Else if  $Y = 1$  and  $p < 0.70$ :  $R$  = Moderate Estimated Risk  
 Else if  $Y = 0$  and  $p \geq 0.35$ :  $R$  = Moderate Estimated Risk  
 Else:  
      $R$  = Lower Estimated Risk
10. Store  $X, Y, P, R, userID$ , and *timestamp*.
11. Return the assessment result.
12. Make the stored assessment available to history, dashboard, analysis, and reporting modules.

## IV. METHODOLOGY AND MODULE IMPLEMENTATION

### A. Dataset and Preprocessing

This study uses the publicly available Diabetes Prediction Dataset [18]. The original dataset contains 100,000 records and nine columns, comprising eight input features and one binary target variable. The input features are gender, age, hypertension, heart disease, smoking history, body mass index (BMI), glycated haemoglobin (HbA1c) level, and blood glucose level. The target variable represents recorded diabetes status, with class 0 denoting non-diabetes and class 1 denoting diabetes. The dataset was examined for its dimensions, data types, missing values, duplicate records, and class distribution before model development. No missing values were identified in the nine columns. However, 3,854 duplicate records were detected and removed. After duplicate removal, 96,146 unique records remained, comprising 87,664 class-0 records (91.18%) and 8,482 class-1 records (8.82%). These

statistics indicate a class imbalance, with the non-diabetes class accounting for the majority of observations. The feature types, roles, and representations are summarized in Table 1.

Preprocessing was selected according to the characteristics of each feature. Gender and smoking history are categorical variables and were transformed using one-hot encoding. Numerical variables, including age, BMI, HbA1c level, and blood glucose level, were standardized. Hypertension and heart disease were retained as binary input features. The standardization of a numerical feature is expressed as,

$$z = \frac{x - \mu}{\sigma} \quad (5)$$

where,  $(x)$  represents the original feature value,  $(\mu)$  is the mean,  $(\sigma)$  is the standard deviation calculated from the training data, and  $(z)$  is the standardized value. This transformation places numerical features on a comparable scale. To maintain consistency and prevent information leakage, the preprocessing operations were incorporated into the machine-learning pipeline. The preprocessing parameters were learned from the training data and then applied to the test data and subsequent application inputs using the same fitted pipeline. This ensures that the feature transformations used during evaluation and deployment remain consistent across the evaluated classifiers.

Table 1: Feature and Target Description

| Variable            | Type        | Role   | Representation   |
|---------------------|-------------|--------|------------------|
| Gender              | Categorical | Input  | One-hot encoding |
| Age                 | Numerical   | Input  | Standardized     |
| Hypertension        | Binary      | Input  | Standardized     |
| Heart disease       | Binary      | Input  | Standardized     |
| Smoking history     | Categorical | Input  | One-hot encoding |
| BMI                 | Numerical   | Input  | Standardized     |
| HbA1c level         | Numerical   | Input  | Standardized     |
| Blood glucose level | Numerical   | Input  | Standardized     |
| Diabetes            | Binary      | Target | Class 0/1        |

The resulting preprocessing configuration therefore provides a consistent numerical representation of all eight input variables while preserving the binary classification structure of the target. The same configuration is subsequently used for all candidate models so that differences in performance are primarily attributable to the classification algorithms rather than differences in data preparation.

### B. Data Preparation and Train-Test Split

After removing duplicates and preprocessing, we divided our data set for training and testing purposes by splitting it 80-20 (stratified) with a random seed value of 42. There were 76,916 observations for the result and 19,230 for testing. The stratification method is applied for maintaining similar proportions of classes between these two groups.

This dataset is solely utilized to learn about parameter estimation through pre-processing transformation modelling and model fitting on candidates' classifications. The held-out test set is not used while training a model; it is kept aside to evaluate its performance at the end of modelling process. The separation gives us an unbiased way to compare these models based upon data collected from experiments which are independent of their training process.

There is an extreme class imbalance where class 0 occurs much more often than class 1. Thus, depending solely upon precision may not be enough to describe a good model's performance well. The model may achieve good performance through mostly predicting the most common category but fails at identifying an important percentage of positives. For addressing it, we evaluate these models by means of accuracy, precision, recall, F1 score and ROC-AUC. There is no artificial over-sample data used for this model development process. For example, techniques like SMOTE are used for artificial over-representation of a minority class. Therefore, these models are tested on a natural occurrence to clean data set distributions. A constant value for 42 helps improve reproducibility because it ensures that the same observations are assigned to both training and test sets during each run on this data set with identical settings.

### C. Model Development and Selection

#### 1. Development of Candidate Models-

Logistic regression, random forest algorithm extra trees XGBoost was used to develop a machine learning model. Logistic regression is an appropriate starting point since it serves as a benchmark against which we can compare our models' results. This study provides benchmarks to compare performance of various ensembles against each other in terms of accuracy rate or precision/recall ratio between them. A Random Forests algorithm for modelling non-linear relationship among inputs by using multiple decision trees as a single predictive model. Multiple Decision Trees are combined for better accuracy by combining them' s results together. To assess its effectiveness for evaluating this model we used an algorithm as part of our study. Extra Trees is an extra-tree-based model which adds more noise to each node while building a single decision tree. The extra randomness gives more variability versus bias so that we can compare it against an algorithm called Random Forests.

Boosting based XGBoost is used for this study.

This is an iterative process for building sequences of decision trees that attempt to enhance prediction accuracy compared to their predecessors. Since XGBoost is appropriate to handle structured tabular datasets like those found here, it will be selected among other possible options during model selection process.

For an accurate comparison between these models we need to train them on identical inputs, datasets for both training and test purposes, pre-processing procedures as well as scoring criteria. Pre-processing steps are part of a machine learning workflow where they're done once for all datasets to be used by an algorithm when it makes predictions on them. Thus, each model was evaluated on an identical experiment rather than being trained separately from a separate dataset/train/test split. The consistency of this

design allows for comparison between different models' performances.

#### 2. Model Selection

Selection model is done through an evaluation hierarchy where ROC-AUC is used for primary criteria, then F1 score and recall are considered. ROC AUC is chosen for this purpose since it measures how well a classifier separates positive from negative instances at various threshold levels. The F1 score is an important metric for evaluating models since it combines precision and recall into one value. The recall is an important factor for evaluating how well a model can detect instances belonging to certain classes. Thus, a selection tree may be shown by means of this below mentioned flow.

$$ROC - AUC \rightarrow F1Score \rightarrow Recall \quad (6)$$

The hierarchy is an example of selection rules to build models from prototypes. This is not meant to be used for making medical decisions nor does it imply any such thing about those measurements having set priority levels.

Comparative analysis led to choosing XGBoost over other models for use case implementation. All four model details' performance metrics can be found under section v.

#### D. Model Deployment and Inference

Post-selection of models for training purposes, we serialize all data cleaning steps together with an XGBoost classifier through Joblib so that it can be deployed later on a computer system. The serialized version of a trained model along with its corresponding pre-processing settings can be saved for future use instead of having to train it again. While running an app on a server-side using FastAPI framework it is loaded by machine learning services for processing pipelines. An authenticated user submitting a test will provide these to the system for processing by its eight models of inputs:

- Gender
- Age
- Hypertension
- Heart disease
- Smoking history
- BMI
- HbA1c level
- Blood glucose level

Before sending them for prediction purposes, they have been checked out by me. Then it will be organized according to its desired features before being fed back into those models that were trained on them. A transformed feature representation is provided for use by a serialized XGBoost model. Binary classification models produce predictions for classes as well as probabilities of being in those classes. Probability is given by this formula.

$$P(y = 1|x) = \frac{1}{1+e^{-z}} \quad (7)$$

where,  $z$  is a representation of the model's decision value for an input observation and  $P(y = 1|x)$  denotes an estimate of its likelihood to be in class 1. Prediction Service provides an output of classifications along with probabilities for use by applications. Thusly it gives a rough estimate of risk at the application level from those results obtained by us.

The implemented mapping is:

- **Prediction = 1 and probability  $\geq 0.70$ :** Higher Estimated Risk
- **Prediction = 1 and probability  $< 0.70$ :** Moderate Estimated Risk
- **Prediction = 0 and probability  $\geq 0.35$ :** Moderate Estimated Risk
- **Prediction = 0 and probability  $< 0.35$ :** Lower Estimated Risk

These thresholds are used only to provide an understandable presentation of the statistical model output within the application. They are application-level presentation rules and are not clinically validated thresholds. Similarly, the model probability is not presented as a medically calibrated individual risk probability. Following prediction, the application stores the submitted measurements, predicted class, probability, estimated-risk category, timestamp, and authenticated user association in the SQLite database. This enables each assessment to be retrieved later without requiring the model to be executed again.

The stored information supports the application's history and dashboard functionality. Users can review previous assessments and observe descriptive changes in BMI, HbA1c, and blood glucose values across multiple assessments. These visualizations describe the recorded assessment history and are not intended to represent clinically established disease progression. The prediction service is separated from other application services so that model inference remains independent from analytics, nutrition education, authentication, and PDF report generation. This modular design allows the machine-learning component to be updated without requiring the complete application architecture to be redesigned.

#### E. Reproducibility Considerations

Reproducibility was considered throughout the model development and deployment process. The use of a fixed random state of 42 for the train-test split ensures that the same training and testing partitions can be reproduced when the same dataset and experimental configuration are used. The preprocessing transformations and selected classifier are stored together as a serialized pipeline. This is important because deploying only the classifier without its associated preprocessing configuration could result in differences between the representation used during training and the representation supplied during inference.

The implementation also maintains model-related metadata describing the training configuration, selected model, input features, preprocessing structure, and evaluation results. A dependency manifest is maintained to identify the software libraries required by the application and machine-learning environment. For a more complete experimental record, the dataset version, dataset checksum, software versions, random seeds, model configuration, preprocessing configuration, training date, and evaluation results can be preserved alongside the serialized model. These records provide a basis for reproducing the experimental environment and tracing the deployed model to its training configuration. The complete methodology therefore establishes a consistent pipeline from dataset inspection and cleaning to preprocessing, model comparison, model selection, serialization, API-based inference, database

storage, and user-facing health assessment. This integration allows the experimental machine-learning component to be evaluated independently while also demonstrating its implementation within a functional full-stack application.

## V. EXPERIMENTAL SETUP AND DISCUSSIONS

The experimental evaluations were conducted on Logistic Regression, Random Forest, Extra Trees and XGBoost methods with the same preprocessing and evaluation conditions. Following a duplicate removal, there were 96,146 unique observations, of which 76,916 were used for training and 19,230 were held out for testing. Classifier performance was measured by accuracy, precision, recall, F1 score and ROC AUC. These measures were performed in conjunction with each other due to the imbalanced distribution of the target. Performance metrics were calculated from the confusion matrix, which consists of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. Accuracy measures the proportion of correctly classified observations among all observations and is calculated as,

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

Precision measures the proportion of observations predicted as positive that are actually positive. It is given by,

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

Recall measures the proportion of actual positive observations that are correctly identified by the classifier. It is calculated as,

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

The F1 score combines precision and recall into a single measure by calculating their harmonic mean. It is expressed as,

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (11)$$

Specificity measures the proportion of actual negative observations that are correctly identified as negative. Although it is not included in the comparison table, it can be derived from the confusion matrix as,

$$Specificity = \frac{TN}{TN+FP} \quad (12)$$

Table 2 compares the performance of Logistic Regression (LR), Random Forest (RF), Extra Trees (ET), and XGBoost using accuracy, precision, recall, F1-score, and ROC-AUC. Among the four classifiers, XGBoost (XGB) achieved the highest accuracy (97.07%), precision (95.43%), F1-score (80.87%), and ROC-AUC (97.56%). It was therefore selected for deployment based on the model-selection hierarchy described in this study.

Figure 3 & Table 2 shows the graphical representation of performance metrics results of all models.

Table 2: Comparative Performance of the Evaluated Models (%)

| Metrics   | LR    | RF    | ET    | XGB   |
|-----------|-------|-------|-------|-------|
| Accuracy  | 95.96 | 96.90 | 96.50 | 97.07 |
| Precision | 86.92 | 94.21 | 89.62 | 95.43 |
| Recall    | 63.86 | 69.04 | 68.22 | 70.17 |
| F1-Score  | 73.62 | 79.69 | 77.47 | 80.87 |
| ROC-AUC   | 95.99 | 95.52 | 94.69 | 97.56 |

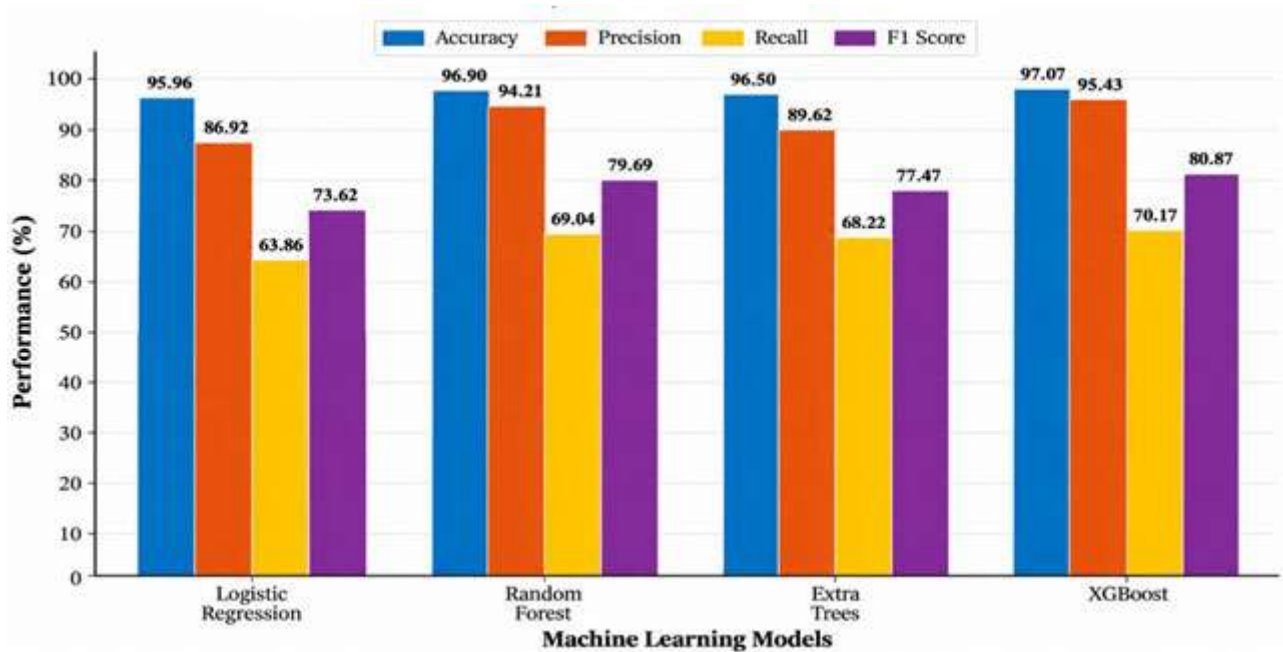


Figure 3: Metric Comparison of the evaluated models

The best performing models were XGBoost (XGB), and the model chosen for deployment was the XGBoost model.

Its performance was 97.07% accuracy, 95.43% precision, 70.17% recall, 80.87% F1 score, and 97.56% ROC AUC.

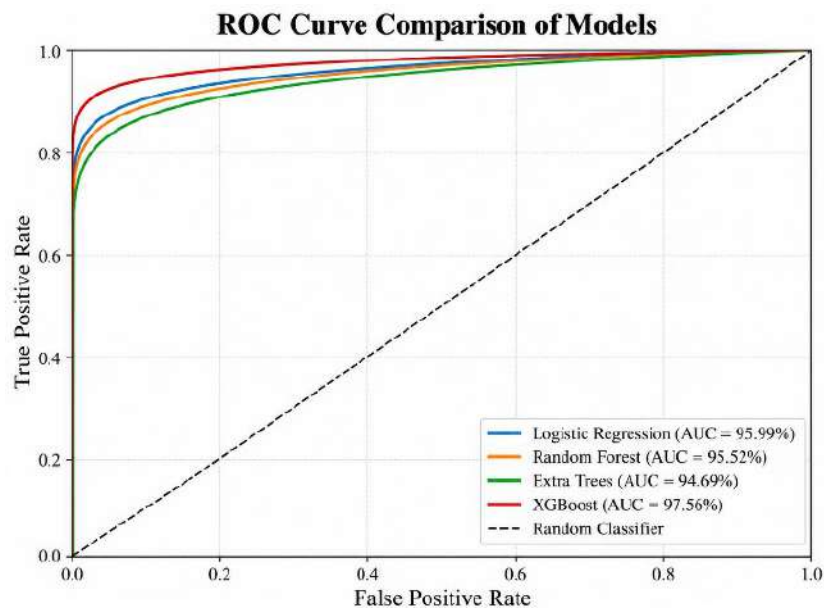


Figure 4: ROC-AUC comparison of the evaluated models

The comparison of models is done on a single held-out test partition created from the available data. So, the reported differences should be viewed as evidence of the experimental results for this evaluation setting, and not as evidence of generalization to external populations. Performance would have to be assessed using an independent set of data before external validation.

Table 3 presents the confusion matrix for the selected XGBoost model on the 19,230-record test set. The model correctly classified 17,477 negative observations and 1,190 positive observations, while 57 negative observations were incorrectly classified as positive and 506 positive observations were incorrectly classified as negative. These results provide the basis for verifying the model's accuracy, precision, recall, and F1-score. Figure 5 presents the graphical representation of confusion matrix.

Table 3: Confusion Matrix of the Selected XGBoost Model

|                 | Predicted Negative | Predicted Positive |
|-----------------|--------------------|--------------------|
| Actual Negative | 17,477             | 57                 |
| Actual Positive | 506                | 1,190              |

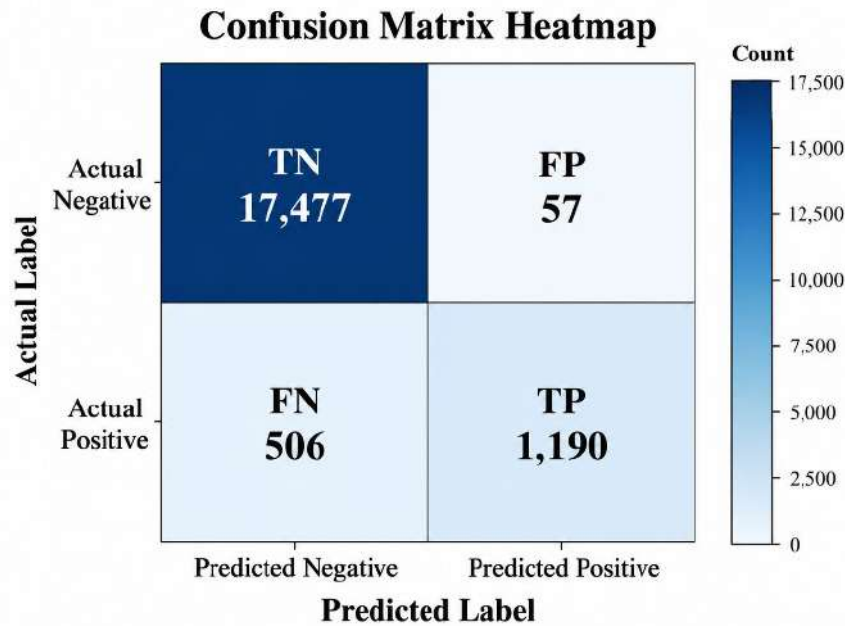


Figure 5: Confusion Matrix of XGBoost Model

The confusion matrix of the selected XGBoost model contains 17,477 true negatives, 506 false negatives, 57 false positives, and 1,190 true positives, covering all 19,230 observations in the held-out test set. Based on these results, the model achieved an accuracy of 97.07%, precision of 95.43%, recall of 70.17%, and F1-score of 80.87%. The relatively low number of false positives contributes to the high precision, while the 506 false negatives reduce the recall for the positive class. Since the dataset is imbalanced, accuracy alone does not fully describe the classifier's performance. Therefore, precision, recall, F1-score, and ROC-AUC should also be considered when evaluating the model.

The receiver operating characteristic (ROC) curve evaluates the classifier across different classification thresholds by relating the true positive rate to the false positive rate. The true positive rate is equivalent to recall, while the false positive rate is calculated as,

$$FPR = \frac{FP}{FP+TN} \quad (13)$$

The true positive rate used in the ROC curve can be expressed as,

$$TPR = \frac{TP}{TP+FN} \quad (14)$$

ROC-AUC represents the area under the receiver operating characteristic curve and summarizes the classifier's ability to distinguish between positive and negative classes across classification thresholds. It can be represented as,

$$ROC-AUC = \int_0^1 TPR(FPR) d(FPR) \quad (15)$$

The model (XGBoost) that was selected according to the model-selection hierarchy described in the use case was used as the deployed model, with ROC AUC being the main criterion, followed by F1 score and recall.

#### A. Application-Level Experimental Validation

The application implemented was tested in addition to its model level evaluation, through its main functional workflow. The assessment interface takes in the eight model input variables and validates the variables' range on

the client side before sending the request to the FastAPI backend. The model prediction is done in the backend with the processed model and XGBoost pipeline.

The successful prediction results in the output of the predicted class, the estimated probability, the application-level risk category, and selected health indicators. The resulting assessment is saved in the SQLite database with the input values provided by the user, the prediction output, the probability value, the risk category, the user association and the timestamp. This permits to get the same assessment result at a later stage without re-running the model. The authenticated history interface fetches authenticated history records of the currently authenticated user and the dashboard shows the most recent assessment and available history records. When there are multiple assessment records, the analysis interface includes a descriptive comparison of the latest assessment to the previous assessment. It shows the changes in the BMI, HbA1c, blood glucose, estimated probability and the category of application-level risk.

Repeated assessment records are used to create a trend visualization, which visualizes the trend of selected measurements over time. These visualizations are descriptive not clinical progression modelling. Additionally, the reporting module has been included in the assessment workflow. The system produces a PDF summary of the assessment context, measurements selected, result, observations, and health-awareness instructions when the assessment is completed.

The report generated makes it clear that it's a statistical estimation, and not a medical diagnosis. In general, the experimental results show both comparative performance of the tested machine-learning models and the incorporation of the selected model within a complete working application. For the well-defined experimental setup, XGBoost showed the best performance and was chosen as the prediction model for DiaPredict. Application-level implementation is the extension of the model beyond individual prediction, to include authentic storage, assessment history, trend analysis with description, educational information, and downloadable reporting.

## VI. CONCLUSION AND LIMITATIONS

This paper introduces DiaPredict which is an end-to-end model used to classify patients' diabetic conditions through web-based health assessments. This tool is an integrated solution for comparing machine learning models through data analysis tasks such as submitting patient records, obtaining statistics on them, storing past results of evaluations, analysing patterns from descriptions, generating organized documents etc. Development started by systematically preparing an open-source database for diabetes research purposes. There are 3,854 exact duplicates that have been removed from this dataset to produce a total of 96,146 unique entries for use by models during training and testing phases. There are eight features for this data set which include demographics, physiology and health related information where 'diabetes' is a binary outcome variable.

Logistic regression, random forest, extra trees, and XGBoost are four classifiers that will be tested on this dataset through standardization techniques with an 80-20

ratio for training versus testing purposes. Accuracy, precision, recall, F1 score and ROC AUC were evaluated for better understanding of model performance under imbalanced data conditions. XGBoost was the best among all of them for this task as it achieved an accuracy rate of 97.07%, a precision rate of 95.43%, a recall rate of 70.17%, an F1 score of 80.87%, and an ROC AUC value of 97.56%. According to a predefined selection hierarchy of ROC AUC, followed by F1 score and recall, XGBoost is chosen for use with applications. This paper shows that it is possible to use an ensemble-based approach towards predicting a patient's health status as a binary variable from this particular data set.

One major benefit to this research project was integrating it within an overall system instead of just analysing models individually for classification purposes alone. React and TypeScript is used by DiaPredict to display information; it has an API built on top of FastAPI that handles machine learning models' predictions, a database called SQLite where data can be stored persistently, and Report Lab which generates reports as PDFs. Authentication of submissions to an app for evaluation purposes; Input Validation Prediction History Dashboards Descriptive Analysis Educational Information Downloadable Reporting. The assessment record contains both submitted health parameter data and its associated model outputs so that past predictions can be fetched from them instead of running a new prediction process each time. Separation of model inference from data persistence is described as an interface to allow further development or improvement later on by separating them into different modules like description, education and report generation.

Interpretation is an issue that needs to be addressed by some important conditions. Target variable for this data set is an indicator of current state of being diabetic; hence it should be treated as a binary classifier instead of predicting future cases of diabetes development. Also, there is no independence of probabilities for this model from its classification results. Risk levels shown hereby are defined at an interface level rather than being a clinical validation of those risks. Therefore, it will be used to raise awareness about health issues and statistics instead of making a diagnosis by doctors, prescribing medicines, or replacing them with other healthcare providers' opinions. Descriptive trends are shown by this tool as a graph of measured values over time for each test; they don't model disease progression.

Also, there are some problems with this research paper. The first step is that we are doing an experiment on this paper using just one public dataset and one holdout test set; thus, its results do not necessarily generalize well for other datasets or models. Also, there was no external verification or independent probability calibration done here. Also, thirdly it has an inadequate amount of data on demographics physiology health related factors as well as no lab genetics device usage statistics for this study. Fourthly, there are no advanced explanation methods like SHAP based ones nor any details on subgroups within this model. The constraints limit our ability to interpret this model on a larger scale and use it more effectively in high-stakes situations.

Thus, future work may be concerned with external validation through an independent dataset; probability calibration; subgroup-based performance assessment; and

systematized model versions. Explanation of AI models would help us understand more about what affects their results. Longitudinal data would be useful for further detailed study on repeat tests; it may also provide additional details through integration with other medical records databases. Improvements in visualizations, reports, reproducibility, and ongoing evaluations would help improve this system's design for better performance testing on new models.

In general, DiaPredict shows that it is possible to combine comparative machine learning evaluation techniques with an actual software product used by people for monitoring their condition related to diabetes. This paper is a comparative analysis between common machine learning algorithms for classification purposes as well as its integration to connect models' predictions with real-world information storage systems like databases or spreadsheets; it also includes descriptive statistics on these datasets along with relevant educational materials and reports. Given these constraints on data availability and experimental setup, our findings show that it is feasible to use an integrated method but also highlight its need for verification from outside sources through calibration processes; interpretation capabilities should be considered as well as clear separation between statistical machine learning output versus medical decisions made by clinicians.

### CONFLICTS OF INTEREST

The authors declare that they have no known financial, professional, or personal conflicts of interest that could have influenced the research work, analysis, interpretation, or presentation of the results reported in this study.

### ACKNOWLEDGMENT

The first author sincerely expresses her gratitude to Dr. Nithyanandh Selvam, Associate Professor & Project Supervisor, Department of Computer Applications (PG-MCA), PSG College of Arts and Science, Coimbatore, for his continuous guidance, valuable suggestions, academic support, and encouragement throughout the research work. The authors also acknowledge the Department of Computer Applications (PG), PSG College of Arts and Science, Coimbatore, India, for providing the necessary academic environment, facilities, and institutional support for carrying out this research successfully.

### REFERENCES

- [1] S. Masood, M. A. Al-Bashrawi, M. A. Khan, and Y. K. Dwivedi, "From data to diagnosis: A systematic review on AI-driven approaches to diabetes prediction," *Artificial Intelligence Review*, vol. 59, no. 3, Art. no. 103, 2026. Available from: <https://doi.org/10.1007/s10462-025-11485-3>
- [2] P. B. Khokhar, C. Gravino, and F. Palomba, "Advances in artificial intelligence for diabetes prediction: Insights from a systematic literature review," *Artificial Intelligence in Medicine*, vol. 164, Art. no. 103132, 2025. Available from: <https://doi.org/10.1016/j.artmed.2025.103132>
- [3] M. Li, A. Tan, L. Song, X. Qiu, L. Zhou, and Z. Wen, "Performance of diabetes risk prediction models: A systematic review and meta-analysis," *Endocrine Connections*, vol. 14, no. 11, Art. no. e250353, 2025. Available from: <https://doi.org/10.1530/EC-25-0353>
- [4] Z. Rafie, M. S. Talab, B. E. Z. Koor, A. Garavand, C. Salehnasab, and M. Ghaderzadeh, "Leveraging XGBoost and explainable AI for accurate prediction of type 2 diabetes," *BMC Public Health*, vol. 25, Art. no. 3688, 2025. Available from: <https://doi.org/10.1186/s12889-025-24953-w>
- [5] W. Zhang, Z. Xia, G. Cai, J. Wang, and X. Dong, "Enhancing diabetes risk prediction through focal active learning and machine learning models," *PLOS ONE*, vol. 20, no. 7, Art. no. e0327120, 2025. Available from: <https://doi.org/10.1371/journal.pone.0327120>
- [6] J. R. Navarro-Cerdán, P. Pons-Suñer, *et al.*, "A machine learning approach for type 2 diabetes diagnosis and prognosis using tailored heterogeneous feature subsets," *Medical & Biological Engineering & Computing*, vol. 63, no. 9, pp. 2733–2752, 2025. Available from: <https://doi.org/10.1007/s11517-025-03355-5>
- [7] Nie, X. Song, and W. Chen, "DRPM: An advanced predictive model for early diabetes detection and risk stratification," *Molecular Therapy—Nucleic Acids*, vol. 36, no. 3, Art. no. 102576, 2025. Available from: <https://doi.org/10.1016/j.omtn.2025.102576>
- [8] B. Barman, H. K. Choudhury, and B. Jajodia, "Interpretable machine learning for diabetes risk prediction: A large-scale analysis of Indian national survey data," *Discover Public Health*, vol. 22, Art. no. 832, 2025. Available from: <https://doi.org/10.1186/s12982-025-01236-8>
- [9] M. Matboli, A. Khaled, M. F. Ahmed, M. Y. Ahmed, R. Khaled, G. M. Elmakromy, A. M. Abdel Ghani, M. M. El-Shafei, M. R. M. Abdelhalim, and A. M. Abd El Gwad, "Machine learning-based stratification of prediabetes and type 2 diabetes progression," *Diabetology & Metabolic Syndrome*, vol. 17, Art. no. 227, 2025. Available from: <https://doi.org/10.1186/s13098-025-01786-6>
- [10] X. Li, F. Ding, L. Zhang, S. Zhao, Z. Hu, Z. Ma, F. Li, Y. Zhang, Y. Zhao, and Y. Zhao, "Interpretable machine learning method to predict the risk of pre-diabetes using a nationwide cross-sectional data: Evidence from CHNS," *BMC Public Health*, vol. 25, Art. no. 1145, 2025. Available from: <https://doi.org/10.1186/s12889-025-22419-7>
- [11] S. Khan, F. Mohsen, and Z. Shah, "Genetic biomarkers and machine learning techniques for predicting diabetes: Systematic review," *Artificial Intelligence Review*, vol. 58, Art. no. 41, 2025. Available from: <https://doi.org/10.1007/s10462-024-11020-w>
- [12] M. Lugner, A. Rawshani, E. Helleryd, and B. Eliasson, "Identifying top ten predictors of type 2 diabetes through machine learning analysis of UK Biobank data," *Scientific Reports*, vol. 14, Art. no. 2102, 2024. Available from: <https://doi.org/10.1038/s41598-024-52023-5>
- [13] Dharmarathne, M. R. Bogawaththa, P. N. Meddage, and U. Rathnayake, "A novel machine learning approach for diagnosing diabetes with a self-explainable interface," *Healthcare Analytics*, vol. 5, Art. no. 100301, 2024. Available from: <https://doi.org/10.1016/j.health.2024.100301>
- [14] Y.-J. Huang, C.-H. Chen, and H.-C. Yang, "AI-enhanced integration of genetic and medical imaging data for risk assessment of type 2 diabetes," *Nature Communications*, vol. 15, Art. no. 4230, 2024. Available from: <https://doi.org/10.1038/s41467-024-48618-1>
- [15] P. N. Srinivasu, S. Ahmed, M. Hassaballah, and N. Almusallam, "An explainable artificial intelligence software system for predicting diabetes," *Heliyon*, vol. 10, no. 16, Art. no. e36112, 2024. Available from: <https://doi.org/10.1016/j.heliyon.2024.e36112>
- [16] N. Katiyar, H. K. Thakur, and A. Ghatak, "Recent advancements using machine learning and deep learning approaches for diabetes detection: A systematic review," *e-Prime—Advances in Electrical Engineering, Electronics and Energy*, vol. 9, Art. no. 100661, 2024. Available from: <https://doi.org/10.1016/j.prime.2024.100661>

- [17] F. Mohsen, H. R. H. Al-Absi, N. A. Yousri, N. El Hajj, and Z. Shah, "A scoping review of artificial intelligence-based methods for diabetes risk prediction," *npj Digital Medicine*, vol. 6, Art. no. 197, 2023. Available from: <https://doi.org/10.1038/s41746-023-00933-5>
- [18] Kaggle, "Diabetes prediction dataset." Available from: <https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset>. Accessed: Sep. 2, 2026.

## ABOUT THE AUTHORS



**Lavanya R**, PG-MCA Student, PSG College of Arts and Science, Coimbatore, India. An enthusiastic learner who has mastering skills in Machine Learning, Deep Learning and Application Development.



**Dr. Nithyanandh Selvam**, Associate Professor in the Department of Computer Applications, PSG College of Arts & Science, Coimbatore, with 17 years of teaching and 10 years of research experience. He holds a Ph.D. in Computer Science & Applications. His research interests include Artificial Intelligence, Machine Learning, Deep Learning, Advanced Networking, Sensor Networks, Image Processing, and Disease Prediction. He is a Multidisciplinary AI Researcher who collaborates and do research with all streams. He has published 35 research articles and 16 book chapters, with 6 Industrial Design recognitions, 7 Copyrights, and 1 Utility Patent Publication. He is an editor for various books and reviewer for Many Scopus, SCIE, Springer, Elsevier and IEEE Journals.



**Dr. V. S. Anita Sofia**, Associate Professor in the Department of Computer Science at PSG College of Arts & Science, Coimbatore, with over 25 years of teaching and research experience. She holds a Ph.D. in Computer Applications, an M.Phil. in Computer Science, and an MCA. Her research interests include Machine Learning, Artificial Intelligence, Deep Learning, Wireless Sensor Networks, and IoT. Her contributions include books, research publications, book chapters, a patent in wearable healthcare technology, and TNSCST-funded student projects. She has also served as Head of the Department of Computer Technology at Sri Krishna Arts and Science College and contributed to research supervision, curriculum development, and academic administration.