Uludağ University Journal of The Faculty of Engineering, Vol. 31, No. 2, 2026 RESEARCH

DOI: 10.17482/uumfd.1906798

THE MACHINE LEARNING-BASED DIABETES PREDICTION TOOL

Nurullah AYVALIK [*]
Fahri VATANSEVER [*]

Received: 10.03.2026; revised: 27.04.2026; accepted: 19.05.2026

Abstract: Machine learning methods are widely used in many areas, including engineering, healthcare, manufacturing, finance, transportation, and marketing because of their diversity, applicability, and high performances. Since diabetes mellitus is considered one of the most important health problems of our time, early diagnosis is extremely important to protect public health and reduce various complications caused by the disease. In this study, a new desktop-based clinical decision support software tool was developed to predict diabetes using numerous different machine learning algorithms. This feature-rich prediction tool aims to enable decision-makers to detect diabetes easily, quickly, and effectively with many machine learning models by generating individual risk probabilities.

Keywords: Machine learning, Deep learning, Diabetes, Prediction tool

Makine öğrenimi tabanlı diyabet tahmin aracı

Öz: Makine öğrenimi yöntemleri; çeşitlilikleri, uygulanabilirlikleri ve yüksek performansları nedeniyle mühendislik, sağlık, üretim, finans, ulaşım ve pazarlama gibi birçok alanda yaygın olarak kullanılmaktadırlar. Diyabet, günümüzün en önemli sağlık sorunlarından biri olarak kabul edildiğinden, halk sağlığını korumak ve hastalığın neden olduğu çeşitli komplikasyonları azaltmak için erken teşhis son derece önemlidir. Bu çalışmada, çok sayıda farklı makine öğrenimi algoritması kullanarak diyabet tahmini gerçekleştiren yeni bir masaüstü tabanlı klinik karar destek yazılım aracı geliştirilmiştir. Birçok özelliğe sahip bu tahmin aracı, bireysel risk olasılıkları üreterek karar vericilerin birçok makine öğrenimi modeli kullanarak diyabeti kolayca, hızlı ve etkili bir şekilde tespit etmelerini sağlamayı amaçlamaktadır.

Anahtar Kelimeler: Makine öğrenimi, Derin öğrenme, Diyabet, Tahmin aracı

1. INTRODUCTION

Machine learning (ML) is a field/subset/branch of artificial intelligence (AI). ML is concerned with the development of statistical algorithms and models that use data to enable machines and computers to learn in a similar way to humans, to perform tasks autonomously without explicit instructions/programming, and to improve their performance and accuracy with increasing amounts of data. In simple words, ML teaches algorithms and models to learn from data by iteratively optimizing based on error feedback, thus learning and understanding like humans. One of the most popular sub-branches of ML is deep learning (DL). For this reason, it is commonly used in many areas such as image or speech recognition, medical diagnosis, recommender systems, self-driving cars, fraud detection, social media, marketing and sales, traveling, healthcare, and smartphones.

[*] Department of Electrical-Electronics Engineering, Bursa Uludağ University, Bursa, Türkiye
Corresponding author: Fahri Vatansever (fahriv@uludag.edu.tr)

Diabetes mellitus is considered one of the most important health problems of our time. This disease is serious both because of its chronic nature and its increase worldwide. According to the Diabetes Atlas published by the International Diabetes Federation (IDF), 589 million adults aged 20-79 worldwide live with diabetes. This number is expected to reach 853 million by 2050 (IDF, 2025). Early diagnosis and treatment of diabetes are extremely important to protect public health and reduce various complications caused by the disease. Therefore, ML techniques are being used and developed intensively for high-accuracy diagnosis.

There are many studies in the literature using ML and DL in the field of diabetes (Alzboon et al., 2023; Feng et al., 2023; García-Jaramillo et al., 2024; Khan et al., 2021; Modak and Jha, 2024; Nayak et al., 2025; Sharma and Shah, 2021; Sun et al., 2025; Wee et al., 2024). These studies can be grouped mainly as detection, prediction, management and others. To further increase performance in these areas, different techniques and architectures are proposed and various decision-making and support units/software are developed. The most frequently used methods are neural networks (NN), artificial neural networks (ANN), convolutional neural networks (CNN), decision tree (DT), random forest (RF), support vector machines (SVM), naïve Bayes (NB), k-nearest neighbor (KNN), reinforcement learning (RL), multilayer perceptron (MLP), regression algorithms (RAs), support vector regression (SVR), transfer learning (TL), fuzzy neural networks (FNN), etc.

In this realized study, a new software tool was developed that predicts diabetes using different ML and DL models. The main contributions of this prediction tool are summarized below:

i. Having multiple trained models on a single platform,
ii. Including new models that have not been commonly used in this field in the literature,
iii. Allowing the addition of new models,
iv. Generating results for each model in diabetes prediction,
v. Presenting results both numerically and graphically,
vi. Performing detailed statistical analyses of the results (average, weighted average, etc.),
vii. Providing ease of use and flexibility with its user-friendly interface,
viii. Offering decision-makers a wide range of alternative outcomes to evaluate.

The organization of the paper is as follows: Section 2 summarizes the ML methods used in this study and describes the PIMA Indian Diabetes Dataset (PIDD) and its features, Section 3 presents the developed software tool and its applications, and Section 4 contains the conclusions.

2. MATERIALS AND METHODS

2.1. The Used Dataset

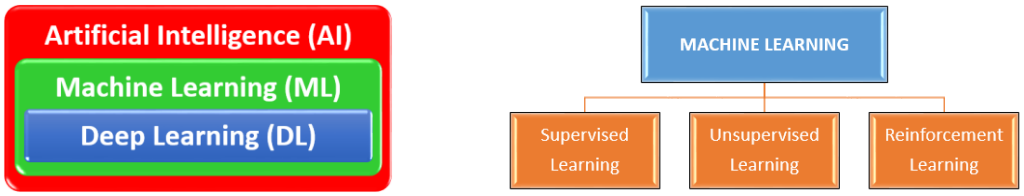
In this study, the PIDD (PIDD, 2025), which is popular in medical and ML research, was used. All of 768 participants/instances in this dataset are female, which 268 samples were identified as diabetic and 500 were non-diabetics and at least 21 years old. The eight most effective features (attributes, variables) and their descriptions included in the dataset and used in diabetes prediction are given in Table 1.

Table 1. The features of the used PIDD

Feature	Description	Type	Values	
			Min	*Max*
Pregnancies	Number of pregnancies	Integer	0	17
Glucose	Plasma glucose concentration a 2 hours in an oral glucose tolerance test	Integer	0	199
Blood pressure	Diastolic blood pressure (mmHg)	Integer	0	122
Skin thickness	Triceps skin fold thickness (mm)	Integer	0	99
Insulin	2-Hour serum insulin (μU/ml)	Integer	0	846
BMI	Body mass index (kg/m²)	Real	0	67.1
Diabetes pedigree function	Diabetes risk prediction based on age and family history	Real	0.08	2.42
Age	Age of participants (years)	Integer	21	81
Outcome	Class variable; 268 of 768 are 1, the others are 0	Binomial	0	1

2.2. The Basic Machine Learning Methods

AI is a technology that enables digital systems (computers, machines, etc.) to simulate human abilities such as learning, problem-solving, and decision-making. ML is a subset of AI that allows these systems to learn and improve autonomously from large amounts of data without being detailed and explicitly programmed. This ML process, which involves creating algorithms and statistical models, enables computers to make decisions and predictions. A subset of ML that uses ANNs to process and analyze information is DL (Figure 1a). ML basically comes in three types: supervised learning, unsupervised learning, and reinforcement learning (Figure 1b). In addition, there are two further types: semi-supervised and self-supervised learning. Supervised learning is a type of model where the input-output relationship is known and labeled training data (structured data) is used. Here, the model is trained on the data with a known output. Unsupervised learning is a type of model where the output is not known beforehand, and unlabeled data (unstructured data) is used. Here, the model learns from this data. Reinforcement learning is a type that uses a reward and penalty system to train models. Here, the model learns by doing through many trial-and-error experiments. Semi-supervised learning is a combination of supervised and unsupervised learning, and is a type of model that uses both data types (labeled and unlabeled). In this model, a small number of labeled data guides the learning process for a larger number of unlabeled data. Self-supervised learning is a ML model that uses unlabeled data to train itself. The model automatically creates its own pseudo-labels by learning from some of the unlabeled data. Thus, learning changes from unsupervised to supervised.

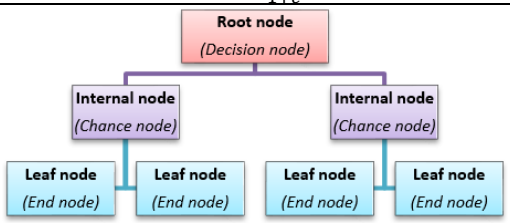
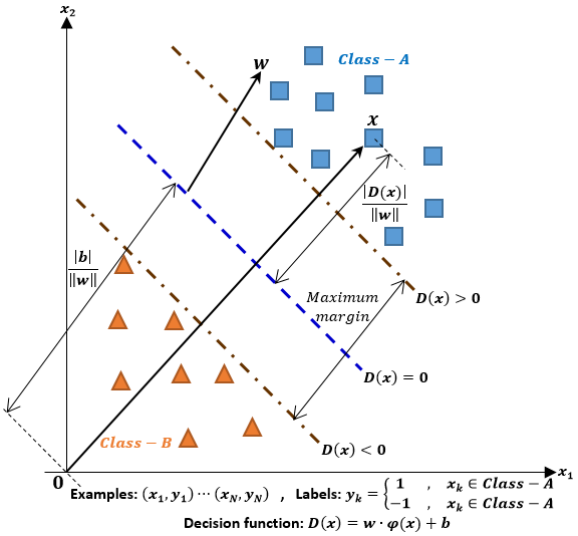


a. The AI, ML, and DL **b.** The ML types

Figure 1:
The AI, ML, DL and ML types

The basic ML methods in this study are summarized in Table 2 (Alpaydin, 2010; Boser et al., 1992; Breiman, 2001; Dhawan and Bansal, 2025; Ho, 1995; Hochreiter and Schmidhuber, 1997; Hosmer et al., 2013; Joshi, 2020; Li et al., 2022; Mienye and Sun, 2022; Peterson, 2009; Van Houdt et al., 2020; Wickramasinghe and Kalutarage, 2021; Wlodarczak, 2020).

Table 2. The brief explanations of basic ML methods

Method	Explanation
Logistic Regression (LogReg)	The logistic regression is a popular statistical and ML method with three main approaches: binary, ordinal, and multinomial. This analysis, which calculates the predicted/estimated values of the dependent variable as probabilities, uses the odds ratio (OR) ($Odds = \frac{P}{1-P}$). The OR is the ratio of two odds and shows the relationship between two variables. Since OR is asymmetric, it is converted to symmetric by taking its natural logarithm ($logit(P) = Ln(OR)$). For $z = \beta_0 + \beta_1 X_1$ the logistic regression model $logit(P) = Ln\left(\frac{P}{1-P}\right) = Ln(e^z) = z$, where $P = \frac{e^z}{1+e^z}$.
Decision Tree (DT)	 Decision trees are one of the most effective methods for data mining. Due to their ease of use, this non-parametric supervised learning method is widely used in many areas (regression, classification, diagnosis, prediction, data manipulation, etc.). It basically has a hierarchical tree structure consisting of nodes (root and internal), branches and leaves. Nodes are points where decisions are made or outcomes are shown. The root or decision node is the starting point representing the whole dataset; internal or chance nodes are the points where decisions are made based on data features; and leaf or end nodes (leaves) are the end points where the final decision/outcome or prediction is made. Branches connect nodes and represent available choices (chance outcomes or occurrences). In this algorithm, the decision made at each node leads to the next branch or leaf, thus reaching the solution. Algorithms such as Classification and Regression Tree (CART), Chi-Squared Automatic Interaction Detector (CHAID), Iterative Dichotomiser (ID3), C4.5, and QUEST are available for building these decision trees.
Random Forest (RF)	The random forest, proposed by Ho and extended by Breiman, is an ensemble learning algorithm created by combining many completely independent tree-structured classifiers (decision trees). Here, the performance of decision tree models is improved using a divide-and-conquer approach.
Support Vector Machine (SVM)	 Support vector machines, introduced in the 1990s, are supervised ML models used in many areas. This algorithm, based on the fundamental concepts of hyperplane (decision boundary separating classes), support vectors (points closest to the hyperplane), and margin (distance between the hyperplane and the nearest support vectors), aims to maximize the difference (maximum margin) between the training data and the decision boundary.

Naïve Bayes (NB)

The naive Bayes algorithm, used in many problems for data classification due to its simplicity and accuracy, is based on Bayes' theorem:

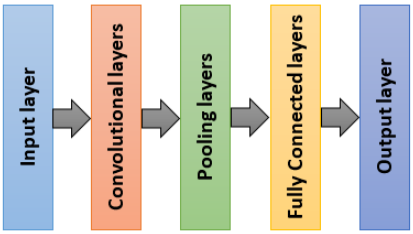
$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)}$$

where $P(C_i|x)$, $P(x|C_i)$, $P(C_i)$ and $P(x)$ are the probability of C_i given data x, the probability of x given C_i, the prior probability of C_i and the prior probability of x, respectively. NB, a fast and efficient probabilistic classifier algorithm, applies Bayes' Theorem with the assumption that variables are independent. This algorithm, which is based on the probability that a given instance belongs to a specific class, also has different variants such as semi-Naïve and weighted-Naïve.

K-Nearest Neighbors (KNN)

K-Nearest Neighbors is a non-parametric and supervised ML method, commonly based on the Euclidean distance between a test and training samples. In this classifier, the distance between target nodes is used to assign them to the class generated by their nearest neighbors (K neighbors) in the pre-classified training data. In other words, it works by comparing a new data point with stored samples and determining its label or value based on its nearest neighbors. The Euclidean distance between n samples x_i and x_j with property p is defined as $d(x_i, x_l) = \sqrt{\sum_{k=1}^{p}(x_{ik} - x_{jk})^2}$.

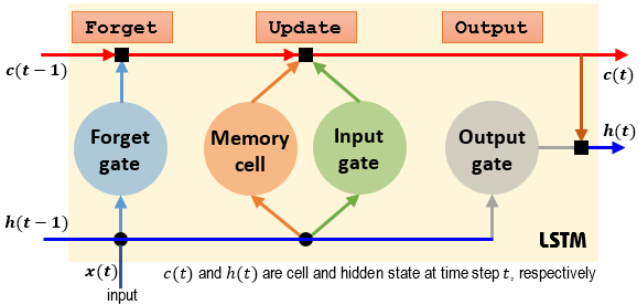
Convolutional Neural Networks (CNN)



Layer	Process
Input	Acquiring raw data
Convolutional	Filtering for local pattern detection
Pooling	Reducing the dimensionality and making the model more efficient and robust
Fully connected	Combining the extracted features
Output	Converting final scores into probabilities

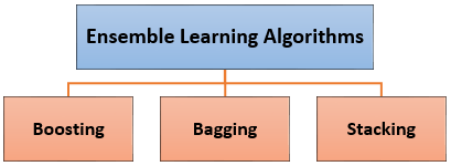
Convolutional Neural Networks, one of the most important networks in the field of DL, are used in many areas, primarily natural language processing (NLP) and computer vision. A CNN, a type of feedforward neural network, is a DL model that extracts/learns features through filters/kernels implemented in convolutional layers. CNNs incrementally extract features through convolution and pooling, and then classify them using fully connected layers.

Long Short-Term Memory (LSTM)



Component	Process
Memory cell	The long-term memory that manages and stores information
Forget gate	Decides what information to removed (forget, discarded) from memory cell
Input gate	Updates the Cell State by deciding what new information will store
Output gate	Decides what information is passed to the next step

The Long Short-Term Memory, was proposed by Hochreiter and Schmidhuber in 1997, is an artificial Recurrent Neural Network (RNN) architecture used in DL, designed to overcome the vanishing gradient problems that limit traditional RNNs.

Ensemble Learning Algorithms	 Ensemble learning techniques that improve application performance by combining predictions from multiple models can be generally categorized into three types: boosting, bagging, and stacking. Boosting techniques such as AdaBoost, GradientBoosting, and XGBoost increase the capabilities of learners. For example, AdaBoost achieves this by adjusting the weights of misclassified samples; GradientBoosting uses gradient descent in optimizing the loss function. Bagging (bootstrap aggregating) techniques enhance the performance of ML models via combining randomly generated training sets' predictions. On the other hand, stacking (stacked generalization) combines models in a layered architecture to improve performance.
Transformer-based Algorithms	One type of DL architecture is the transformer architecture. This architecture primarily consists of an "encoder" that processes input data, "attention layers" that calculate relationships between tokens, "feedforward layers" that transform the related information into representations, and a "decoder" that produces output. Transformer-based methods include algorithms such as TabNet, TabPFN (Transformer-based Prior-data Fitted Network), and FT-Transformer (Feature Tokenizer+Transformer)

2.3. The Performance Metrics

The basic performance metrics in ML models are given in Table 3.

Table 3. The performance metrics in ML models

Confusion matrix			
		Predicted	
		Positive	*Negative*
Actual	*Positive*	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Accuracy	**Precision**	**Recall**	**F1 score**
$\frac{TP+TN}{TP+TN+FP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$	$2\frac{Precision \cdot Recall}{Precision+Recall}$

3. THE DEVELOPED PREDICTION TOOL

The diabetes prediction tool presented in this study is a desktop-based clinical decision support application developed in Python 3.12, primarily using the CustomTkinter graphical user interface library, with NumPy for numerical computation, Matplotlib for data visualization, and Pillow for images. The main components/libraries used in the design are given in Figure 2 (CustomTkinter, 2025; Keras, 2025; Matplotlib, 2025; NumPy, 2025; Pillow, 2025; Python, 2025; PyTorch, 2025; scikit-learn, 2025; SciPy, 2025; TensorFlow, 2025).

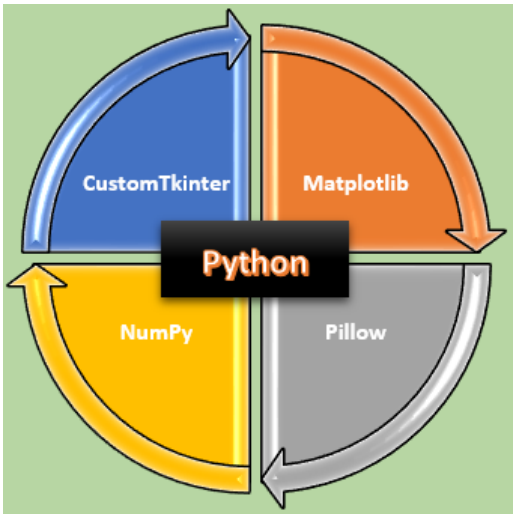


Figure 2:
The main components used in the design

Table 4. *The brief explanations of components used in the software tool*

Python	Python is a high-level, object-oriented programming language widely used in many areas such as programming (desktop and mobile applications, web and game development), data science, and machine learning etc. Its main features include simple syntax, readability, modularity, flexibility, ease of learning, open source, interpreted language, dynamic type system, cross-platform compatibility, extensive libraries, and community support.
CustomTkinter	CustomTkinter is a hardware-accelerated GUI (Graphical User Interface) framework used to create user-friendly and interactive interfaces, modernizing Python's standard Tkinter library. It is a modern and customizable Python UI (user interface) library with essential features such as ease of use, platform compatibility, customization, themes, modern widgets, and community support.
NumPy	NumPy (Numerical Python) is a fundamental and powerful open-source library/package for numerical and scientific computing within the Python ecosystem. Utilizing a C-based infrastructure, this package offers more efficient memory usage, enabling direct and rapid execution of many mathematical and algebraic operations, especially with multi-dimensional arrays.
Matplotlib	Matplotlib is a fundamental and comprehensive library/package for data visualization. With the many functions (command) included in this library, static, animated, and interactive visualizations can be created easily and quickly in Python.
PyTorch	PyTorch is an open-source optimized tensor library for deep learning using GPUs and CPUs. It's widely used for building and training neural networks.

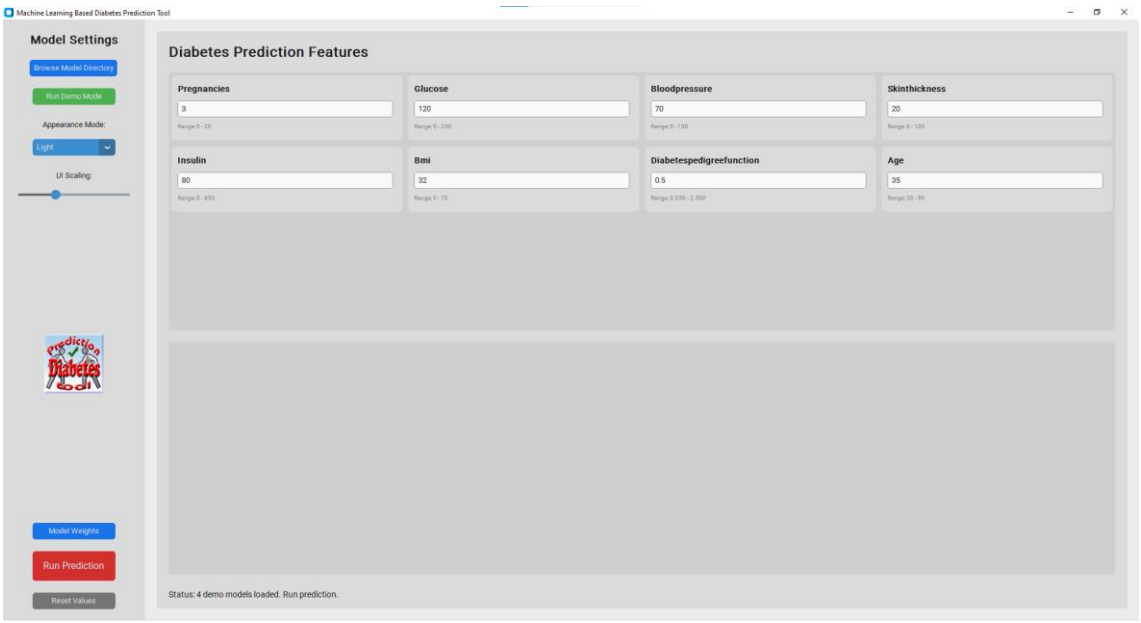
TensorFlo	TensorFlow is a popular open-source machine learning and deep learning library. It facilitates the creation, training, and deployment of models that can run in any environment by simplifying complex mathematical operations with dataflow graphs.
Keras	Keras is an open-source, high-level deep learning library. Its user-friendly API, multiple-backend approaches (TensorFlow, PyTorch, JAX), integration, and cross-platform deployment enable easy and rapid the building and training of neural network models.
SciPy	SciPy is an open-source library/package in Python for scientific computing widely used in different areas. It provides many algorithms and functions for various scientific problems such as solving equations, integration, interpolation, statistics, optimization, etc.
scikit-learn	Scikit-learn is an open source machine learning library in Python. It provides various tools (data preprocessing, model selection, model fitting, model evaluation, etc.) for machine learning applications and supports supervised and unsupervised learning.
Pillow	Pillow is a Python library that provides support for many different image file formats, powerful basic and advanced image processing capabilities, and efficient internal representation.

The application is designed to simultaneously load and execute multiple pre-trained classification models saved in various serialization formats, including Pickle (.pkl), Joblib (.joblib), HDF5 (.h5), and PyTorch (.pt/.pth), thereby supporting models built with Scikit-Learn-compatible machine learning libraries (such as XGBoost and LightGBM) as well as DL frameworks such as TensorFlow/Keras and PyTorch. The interface follows a two-panel layout: a sidebar for system controls (model directory browsing, appearance mode selection, UI scaling, and model weight configuration) and a main workspace for clinical data entry and result visualization. Users entering eight clinical input features (Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, and Age) consistent with the PIDD, namely, where each has real-time boundary validation. Upon execution, the tool generates individual diabetes risk probability scores (ranging from 0 to 1) from each loaded model and computes both an arithmetic average and a user-configurable weighted average across all models. The results are presented through a three-tab interface: a “Table View” listing each model's prediction in ranked order, a “Chart View” displaying a color-coded bar chart with gradient risk coloring overlaid with average reference lines, and a “Summary Dashboard” featuring donut charts for aggregate risk visualization alongside a horizontal bar chart identifying the highest and lowest predicting models. Additional features include a built-in demo mode, dynamic light/dark theme switching, and adjustable UI scaling from 50% to 200%.

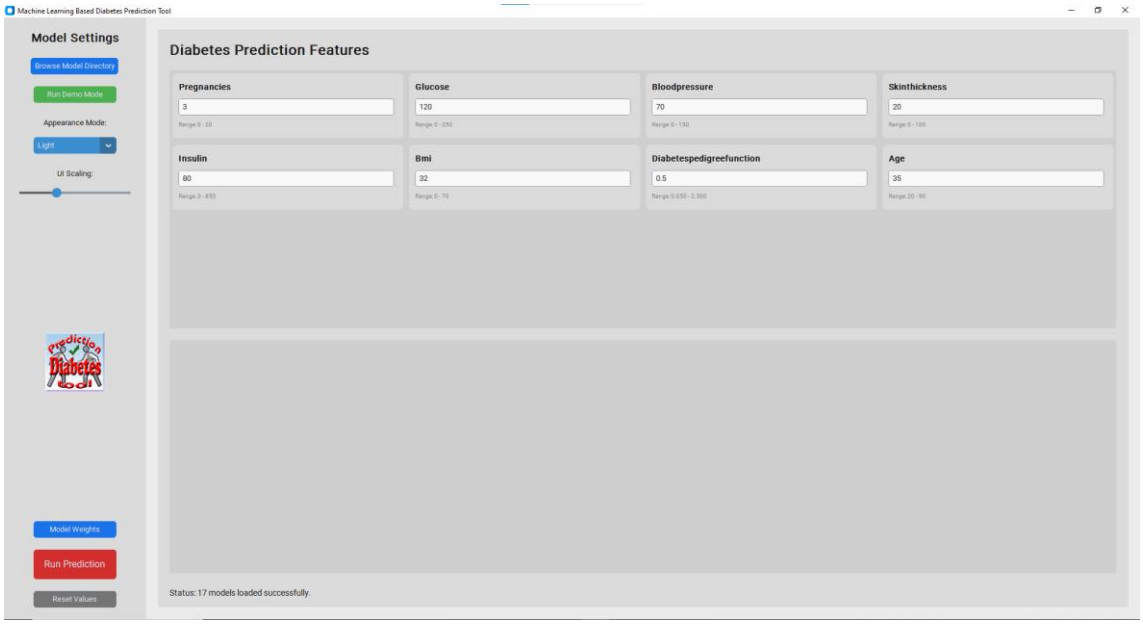
In the data preprocessing phase of this study, biological variables that cannot logically take a value of zero were initially replaced with NaN values. To rigorously prevent data leakage, the imputation process was executed strictly after the train-test split, addressing missing values by utilizing class-specific means derived exclusively from the training set. Subsequently, to preserve model integrity and prevent any statistical influence from the test set, feature scaling was applied to the dataset using a scaler fitted solely on the training data.

The basic process of the developed prediction tool is summarized in Figure 3. The main screen of the designed prediction tool is given in Figure 4a. The prediction process follows these steps: After clicking the “Browse Model Directory” button, the models in this directory are loaded into the program (Figure 4b). The relevant values are then entered using this interface to predict the person's diabetes. Additionally, model weights for the weighted average can be configured from the "Adjust Model Weights" window that opens when clicking the "Model Weights" button if necessary. The software tool uses the training results of the loaded models as default weights (Figure 4c). Finally, the results are displayed using the "Run Prediction" button. The results screen consists of three types: table view, chart view, and summary. In the table view, the prediction results for each model are listed in tabular form with descending order (Figure 4d). In chart view, these results are presented as a bar graph (Figure 4e). In the summary view, the average and weighted average of the prediction results are presented as donut graphs, and the models that produced the minimum and maximum values are shown as bar graphs (Figure 4f).

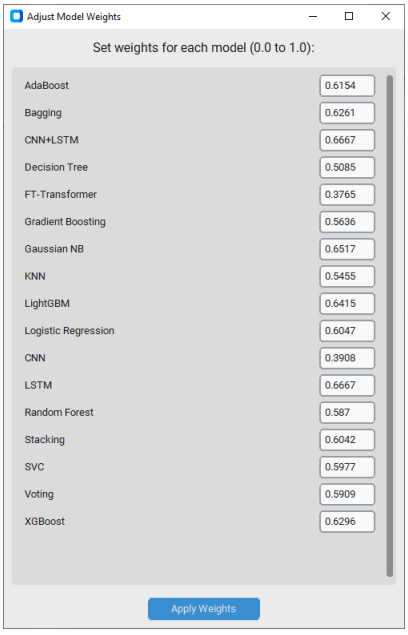
Figure 3:
The main steps in designed prediction tool



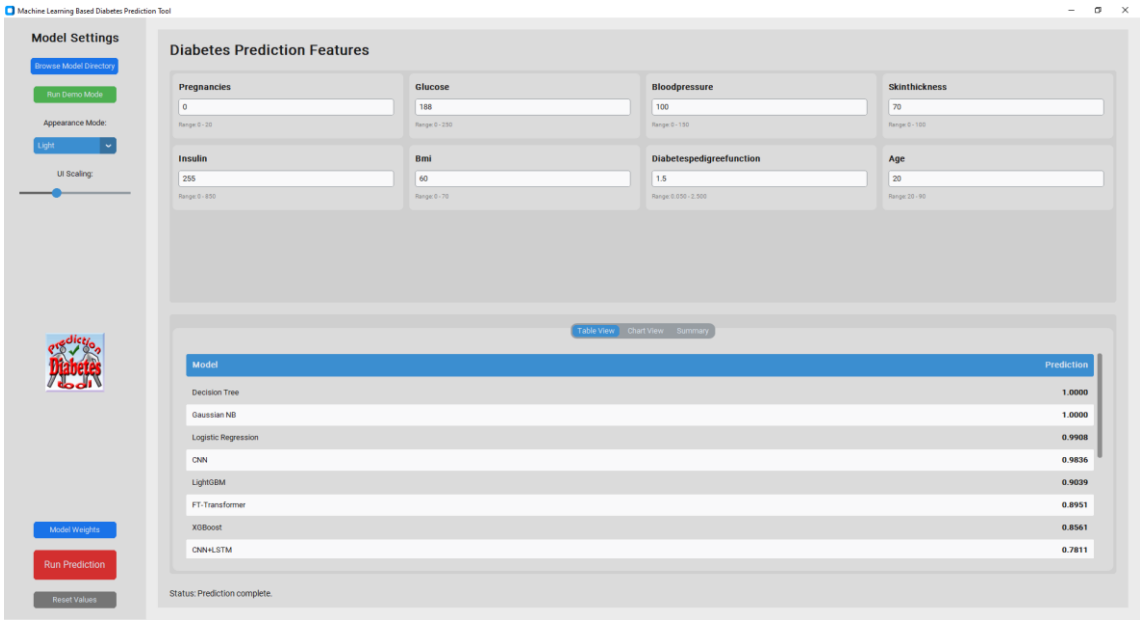
(a)


Machine Learning Based Diabetes Prediction Tool
Model Settings
Browse Model Directory
Run Demo Mode
Appearance Mode:
Light
UI Scaling:
Model Weights
Run Prediction
Reset Values
Diabetes Prediction Features
Pregnancies
3
Glucose
120
Bloodpressure
70
Skinthickness
20
Insulin
80
Bmi
32
Diabetespedigreefunction
0.5
Age
35
Status: 17 models loaded successfully.

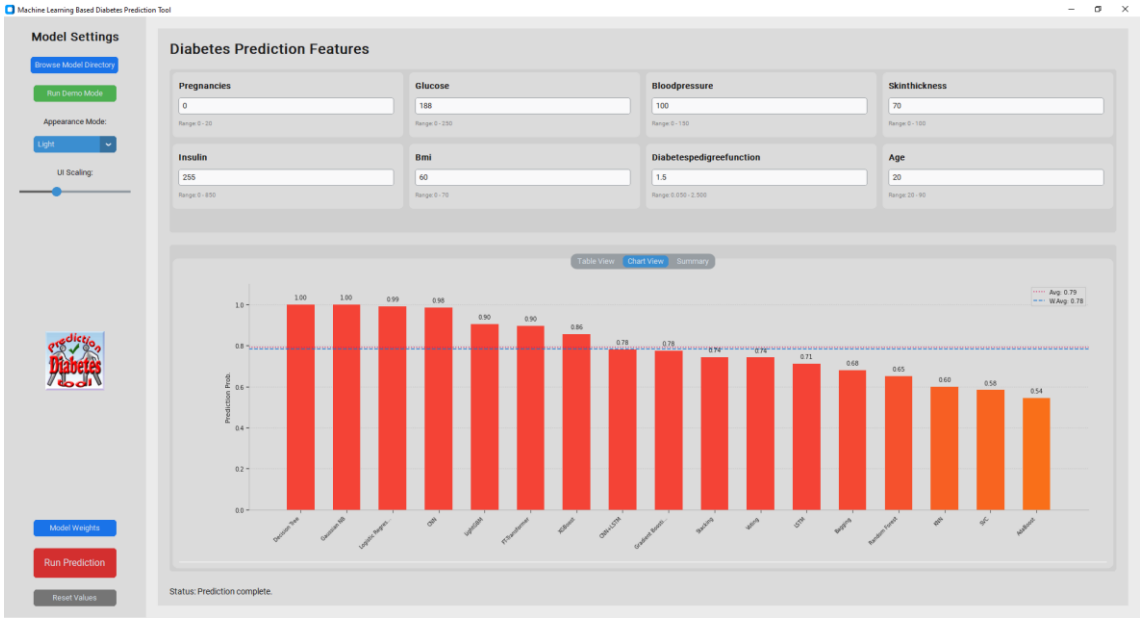
(b)


Adjust Model Weights
Set weights for each model (0.0 to 1.0):
AdaBoost 0.6154
Bagging 0.6261
CNN+LSTM 0.6667
Decision Tree 0.5085
FT-Transformer 0.3765
Gradient Boosting 0.5636
Gaussian NB 0.6517
KNN 0.5455
LightGBM 0.6415
Logistic Regression 0.6047
CNN 0.3908
LSTM 0.6667
Random Forest 0.587
Stacking 0.6042
SVC 0.5977
Voting 0.5909
XGBoost 0.6296
Apply Weights

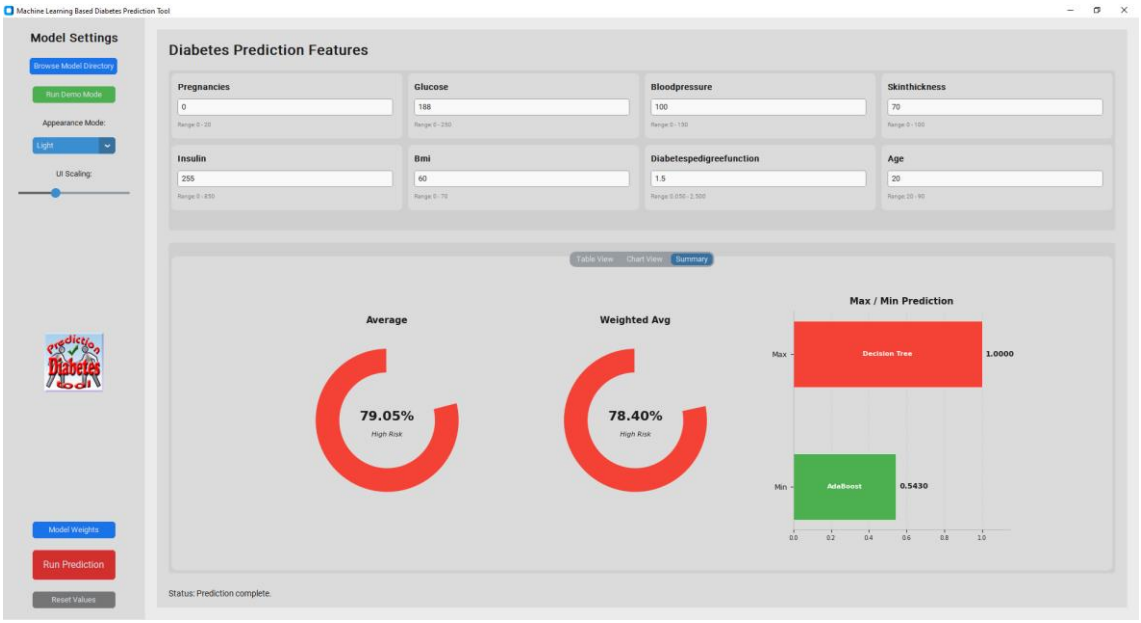
(c)


Machine Learning Based Diabetes Prediction Tool
Model Settings
Browse Model Directory
Run Demo Mode
Appearance Mode:
Light
UI Scaling:
Diabetes Prediction Features
Pregnancies
0
Glucose
188
Bloodpressure
100
Skinthickness
70
Insulin
255
Bmi
60
Diabetespedigreefunction
1.5
Age
20
Table View
Chart View
Summary
Model
Prediction
Decision Tree 1.0000
Gaussian NB 1.0000
Logistic Regression 0.9908
CNN 0.9836
LightGBM 0.9039
FT-Transformer 0.8951
XGBoost 0.8561
CNN+LSTM 0.7811
Model Weights
Run Prediction
Reset Values
Status: Prediction complete.

(d)


Machine Learning Based Diabetes Prediction Tool
Model Settings
Browse Model Directory
Run Demo Mode
Appearance Mode:
Light
UI Scaling:
Diabetes Prediction Features
Pregnancies
0
Glucose
188
Bloodpressure
100
Skinthickness
70
Insulin
255
Bmi
60
Diabetespedigreefunction
1.5
Age
20
Table View
Chart View
Summary
Avg: 0.79
W.Avg: 0.78
Prediction Prob.
1.00
1.00
0.99
0.98
0.90
0.90
0.86
0.78
0.78
0.74
0.74
0.71
0.68
0.65
0.60
0.58
0.54
Model Weights
Run Prediction
Reset Values
Status: Prediction complete.

(e)



(f)

Figure 4:

The screenshots of the designed prediction tool

4. CONCLUSIONS

In this study, a new diabetes prediction tool with a user-friendly interface was developed using different ML and DL models. This software tool, which is extremely simple and easy to use, presents healthcare decision-makers with results obtained from numerous different models, both numerically and graphically. It also allows for the addition of new models and the adjustment of weights, thereby improving the variety and accuracy of the results. Thus, it assists decision-makers in their evaluations by offering many alternatives.

CONFLICT OF INTEREST

The authors confirm that there are no known conflicts of interest and no shared interests with any institution, organization, or individual.

AUTHOR CONTRIBUTION

All authors have contributed equally.

REFERENCES

1. Alpaydin, E. (2010) *Introduction to Machine Learning*, 2nd ed., MIT Press, England.

2. Alzboon, M.S., Al-Batah, M.S., Alqaraleh, M., Abuashour, A., Bader, A.F.H. (2023) Early diagnosis of diabetes: A comparison of machine learning methods, *International Journal of Online and Biomedical Engineering*, 19(15), 144-165. doi: 10.3991/ijoe.v19i15.42417

3. Boser, B.E., Guyon, I.M., Vapnik, V.N. (1992) A training algorithm for optimal margin classifiers, *Proceedings of 5th annual workshop on Computational learning theory COLT'92*, Pittsburgh, PA, 144-152. doi: 10.1145/130385.130401

4. Breiman, L. (2001) Random forests, *Machine Learning*, 45, 5-32. doi: 10.1023/A:1010933404324

5. Dhawan, L., Bansal, A. (2025) Beyond Machine Learning: Exploring Deep Learning and Transformers for Software Defect Prediction. In: Swaroop, A., Virdee, B., Correia, S.D., Polkowski, Z. (eds) *Proceedings of Data Analytics and Management. ICDAM 2024*, Lecture Notes in Networks and Systems, vol 1301, Springer, Singapore. doi: 10.1007/978-981-96-3372-2_7

6. Feng, X., Cai, Y., Xin, R. (2023) Optimizing diabetes classification with a machine learning-based framework, *BMC Bioinformatics*, 24, 428. doi: 10.1186/s12859-023-05467-x

7. García-Jaramillo, M., Luque, C., León-Vargas, F. (2024) Machine learning and deep learning techniques applied to diabetes research: A bibliometric analysis, *Journal of Diabetes Science and Technology*, 18(2), 287-301. doi: 10.1177/1932296823121

8. Ho, T.K. (1995) Random decision forests, *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, Canada, 278-282. doi: 10.1109/ICDAR.1995.598994

9. Hochreiter, S., Schmidhuber, J. (1997) Long short-term memory, *Neural Computation*, 9(8), 1735-1780. doi: 10.1162/neco.1997.9.8.1735

10. Hosmer, D.W., Lemeshow, S., Sturdivant, R.X. (2013) *Applied Logistic Regression*, 3rd ed., Wiley.

11. https://customtkinter.tomschimansky.com/, Accessing date: Accessing date: 2025, *CustomTkinter*.

12. https://keras.io/, Accessing date: 2025, *Keras*.

13. https://matplotlib.org/, Accessing date: 2025, *Matplotlib*.

14. https://numpy.org/, Accessing date: 2025, *NumPy*.

15. https://pypi.org/project/pillow/, Accessing date: 2025, *Pillow*.

16. https://pytorch.org/, Accessing date: 2025, *PyTorch*.

17. https://scikit-learn.org/stable/, Accessing date: 2025, *scikit-learn*.

18. https://scipy.org/, Accessing date: 2025, *SciPy*.

19. https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database, Accessing date: 2025, *Pima Indians Diabetes Database*.

20. https://www.python.org/, Accessing date: 2025, *Python 3.12.0*.

21. https://www.tensorflow.org/, Accessing date: 2025, *TensorFlow*.

22. International Diabetes Federation (2025) *IDF Diabetes Atlas*, 11th edn. Brussels, Belgium. Available at: https://diabetesatlas.org

23. Joshi, A.V. (2020) *Machine Learning and Artificial Intelligence*, Springer.

24. Khan, F.A., Zeb, K., Al-Rakhami, M., Derhab, A., Bukhari, S.A.C. (2021) Detection and prediction of diabetes using data mining: A comprehensive review, *IEEE Access*, 9, 43711-43735. doi: 10.1109/ACCESS.2021.3059343

25. Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J. (2022) A survey of Convolutional Neural Networks: Analysis, applications, and prospects, *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 6999-7019. doi: 10.1109/TNNLS.2021.3084827

26. Mienye, I.D., Sun, Y. (2022) A Survey of ensemble learning: Concepts, algorithms, applications, and prospects, *IEEE Access*, 10, 99129-99149. doi: 10.1109/ACCESS.2022.3207287

27. Modak, S.K.S., Jha, V.K. (2024) Diabetes prediction model using machine learning techniques, *Multimedia Tools and Applications*, 83, 38523-38549. doi: 10.1007/s11042-023-16745-4

28. Nayak, P., Jyothi, J.S.N., Harika, V., Swaraja, K., Sai Hanuman, A. (2025) Diabetes monitoring and prediction using computational intelligence techniques: A systematic review, *SN Computer Science*, 6, 267.doi: 10.1007/s42979-025-03802-y

29. Peterson, L.E. (2009) K-nearest neighbor, *Scholarpedia*, 4(2): 1883. doi: 10.4249/scholarpedia.1883

30. Sharma, T., Shah, M. (2021) A comprehensive review of machine learning techniques on diabetes detection, *Visual Computing for Industry, Biomedicine and Art*, 4, 30. doi: 10.1186/s42492-021-00097-7

31. Sun, Q., Cheng, X., Han, K., Sun, Y., Ren, H., Li, P. (2025) Machine learning-based assessment of diabetes risk, *Applied Intelligence*, 55, 106. doi: 10.1007/s10489-024-05912-1

32. Van Houdt, G., Mosquera, C., Nápoles, G. (2020) A review on the long short-term memory model, *Artificial Intelligence Review*, 53, 5929–5955. doi: 10.1007/s10462-020-09838-1

33. Wee, B.F., Sivakumar, S., Lim, K.H., Wong, W.K., Juwono, F.H. (2024) Diabetes detection based on machine learning and deep learning approaches, *Multimedia Tools and Applications*, 83, 24153-24185. doi: 10.1007/s11042-023-16407-5

34. Wickramasinghe, I., Kalutarage, H. (2021) Naïve Bayes: Applications, variations and vulnerabilities: a review of literature with code snippets for implementation, *Soft Computing*, 25, 2277-2293. doi: 10.1007/s00500-020-05297-6

35. Wlodarczak, P. (2020) *Machine Learning and its Applications*, CRC Press.