KMANPUB | Health Nexus Vol. 2 No. 2 (2024): 103-111

Article history:
Received 11 January 2024
Revised 13 March 2024
Accepted 25 March 2024
Published online 01 April 2024

Prediction of Diabetes using Supervised Learning Approach

Nasim Khazouie[1*], Omid Rahmani Seryasat[2], Sadegh Moshrefzadeh[3]

[1] Assistant Professor, Department of Computer Engineering, Faculty of Technology and Engineering, Yasouj University, Yasouj, Iran
[2] Assistant Professor, Department of Electrical Engineering, Faculty of Technology and Engineering, Shams Higher Education Institute, Gorgan, Iran
[3] Department of Computer Engineering, Faculty of Technology and Engineering, Yasouj branch, Islamic Azad University, Yasouj, Iran

*** Corresponding author email address**: n.khozouie@yu.ac.ir

Article Info

Article type:
Original Research

How to cite this article:
Khazouie, N., Rahmani Seryasat, O., & Moshrefizadeh, S. (2024). Prediction of Diabetes using Supervised Learning Approach. *Health Nexus, 2*(2), 103-111.
https://doi.org/10.61838/kman.hn.2.2.12

ABSTRACT

This paper provides an in-depth evaluation of various supervised machine learning models used for predicting diabetes. It discusses the strengths and limitations of several algorithms, including Decision Trees, Random Forest, Rotation Forest, Ensemble Classifier, K-Star, Simple Bayes, Logistic Regression, Functional Tree, and Perceptron Neural Network. The study utilizes a publicly available diabetes dataset from chistio.ir, which includes 520 samples, comprising 200 diabetic patients and 320 non-diabetic patients, and assesses 16 features. Results are validated on the Weka 3.6 open-source platform, using metrics such as AUC, classification accuracy (CA), F1 score, precision, and recall.

Keywords: *diabetes prediction, diagnosis, data mining, algorithms.*

1. Introduction

Recent advances in healthcare have increasingly relied on various technologies to diagnose diseases and predict health outcomes based on clinical data (1, 2). One of the most promising technological advancements in this area is the application of machine learning (ML) techniques, which have significantly improved the accuracy of diabetes prediction (2). Machine learning enables computers to learn from experiences or inputs, such as clinical data, and predict outcomes like disease presence (3). This capability has revolutionized the way healthcare professionals approach disease prediction and management.

Machine learning techniques are generally categorized into three types: supervised, unsupervised, and reinforcement learning. In supervised learning, both the features and the target class are used as inputs for learning. This approach is particularly effective for classification problems, where the goal is to predict a specific category or class based on input data. On the other hand, unsupervised learning does not involve a target class. Instead, the input data is used to identify patterns and groupings based on similarity measures (4). Reinforcement learning, although not discussed in this paper, involves learning optimal actions through trial and error interactions with an environment.

In the context of diabetes prediction, the problem is often framed as a binary classification task, where the goal is to classify individuals as either diabetic or non-diabetic. Supervised learning algorithms are particularly well-suited for this type of problem. Various supervised learning algorithms have been utilized for diabetes prediction, including the Simple Bayes Algorithm (5 2012), Logistic Regression (6), Perceptron

Neural Network Algorithm (7), K-Star (8), Classification and Regression Trees (CART) algorithm (9), J48 Classification (4), Random Forest (10), Modified Rotation Forest Ensemble Classifier (11), Functional Tree (12), and Bagging Algorithm (13).

This paper focuses on the application of these supervised learning algorithms to predict diabetes. Sixteen different features present in the dataset are considered for prediction: Age, Gender, Polyuria, Polydipsia, Sudden Weight Loss, Weakness, Polyphagia, Genital Thrush, Visual Blurring, Itching, Irritability, Delayed Healing, Partial Paresis, Muscle Stiffness, Alopecia, and Obesity. The performance of the algorithms is compared in terms of classification accuracy (CA) and other metrics, such as the area under the curve (AUC), F1 score, precision, and recall, using the open-source platform Weka 3.6. The results indicate that Logistic Regression outperforms other algorithms in this context.

Several studies have demonstrated the effectiveness of machine learning algorithms in predicting diabetes. Dey et al. (3) used supervised machine learning algorithms, including SVM, KNN, Naive Bayes, and ANN with Min-Max scaling (MMS), on the Pima dataset. They found that the ANN with MMS achieved an accuracy of 82.35%, higher than the other algorithms. Alehegn et al. (4) utilized two datasets, PIDD (Pima Indian Diabetes Dataset) and a US hospital diabetes dataset, employing Random Forest, KNN, Naïve Bayes, and J48 techniques. Their ensemble approach resulted in an accuracy of 93.62% for PIDD and 88.56% for the US hospital dataset.

Sonar and Jaya Malini (14) constructed a model to predict diabetes using machine learning algorithms such as Decision Tree, ANN, Naive Bayes, and SVM. The Decision Tree algorithm achieved an accuracy rate of 85%, outperforming the other algorithms. Similarly, Jain et al. (15) used various ML algorithms like Neural Network (NN), Fisher Linear Discriminant Analysis (FLDA), Random Forest, Chi-square Automatic Interaction Detection (CHAID), and SVM to predict diabetes. They found that the NN algorithm had the highest accuracy rate of 87.88%.

This study employs several supervised learning algorithms to predict diabetes, comparing their performance using a dataset that includes sixteen features related to the condition. The algorithms considered include the Simple Bayes Algorithm, Logistic Regression, Perceptron Neural Network Algorithm, K-Star, CART algorithm, J48 Classification, Random Forest, Modified Rotation Forest Ensemble Classifier, Functional Tree, and Bagging Algorithm. The dataset is analyzed using Weka 3.6, and the performance of each algorithm is evaluated based on classification accuracy (CA), area under the curve (AUC), F1 score, precision, and recall.

The analysis reveals that Logistic Regression outperforms other algorithms in terms of classification accuracy and other performance metrics. The results are consistent with previous studies that have shown the effectiveness of Logistic Regression in predicting diabetes (6). Other algorithms, such as Random Forest and ANN, also perform well, but Logistic Regression offers the highest accuracy and robustness across different datasets.

The application of machine learning algorithms has significantly enhanced the accuracy of diabetes prediction. This study demonstrates that supervised learning algorithms, particularly Logistic Regression, are highly effective in predicting diabetes based on clinical data. The findings are consistent with previous research and highlight the potential of machine learning techniques in healthcare. Future work could explore the integration of multiple algorithms to further improve prediction accuracy and the application of these techniques to other diseases.

In summary, the use of machine learning in healthcare is a promising approach to disease prediction and diagnosis. By leveraging clinical data and advanced algorithms, healthcare providers can make more accurate predictions and improve patient outcomes. This study adds to the growing body of evidence supporting the use of machine learning in healthcare and underscores the importance of continued research and development in this field.

2. Methods and Materials

In the methodology section, the experimental studies, dataset description, and algorithms for predicting diabetes are explained.

2.1. *Experimental Studies*

In this experimental study, three different Decision Tree-Based (DTB) classification algorithms (Random Tree, Rotation Forest, Bagging) were individually implemented on a real-world diabetes dataset to predict diabetes risk at an early stage.

2.1.1. *Dataset Description*

The experiments utilized a publicly available diabetes dataset from the site chistio.ir. This dataset includes 520 samples (patients), comprising 200 diabetic patients and 320 non-diabetic patients. Each sample reviews 16 features.

Table 1

Features of dataset (diabetes)

Feature	Name Type	Description
Age	Numerical	age of the reference
Gender	class (binary)	Gender
Polyuria	class (binary)	Excessive urination (yes/no)
Polydipsia	class (binary)	Fatigue or excessive thirst (yes/no)

WeightLoss	class (binary)	Rapid weight loss (yes/no)
Weakness	class (binary)	Weakness and disease state (yes/no)
Polyphagia	class (binary)	High appetite (yes/no)
GenitalThrush	class (binary)	Genital thrush (yes/no)
VisualBlurring	class (binary)	Blurred vision (yes/no)
Itching	class (binary)	itching (yes/no)
Irritability	class (binary)	Irritability and temper tantrums (yes/no)
DelayedHealing	class (binary)	Delayed recovery (yes/no)
PartialParesis	class (binary)	Local paralysis (yes/no)
MuscleStiffness	class (binary)	Muscle stiffness (yes/no)
Alopecia	class (binary)	Local baldness - alopecia (yes/no)
Obesity	class (binary)	Obesity (yes/no)
Class	class (binary)	Class (positive/negative)

2.2. *Algorithms for Prediction of Diabetes*

Machine learning is widely used today, especially in prediction (16-19). This section discusses various supervised learning algorithms for classifying diabetic and non-diabetic persons. These algorithms create training and testing datasets from the original dataset to classify or predict diabetes.

2.2.1. *Simple Bayes Algorithm*

The Simple Bayes Algorithm, also known as Naive Bayes, is a classification method based on Bayes' theorem under the assumption of independence between predictors. Naive Bayes assumes that the presence of any attribute in a class has no implications for the presence of any other attribute. According to Naive Bayes, the calculation of posterior probability is represented by the following equation: p(c|x) = (p(x|c) * p(c)) / p(x), where:

- P(c|x) is the posterior probability of the class given the predictor.
- P(c) is the prior probability of the class.
- P(x|c) is the probability that the predictor's class has been given.
- P(x) is the prior probability of the predictor.

Advantages:

- Test data can be classified easily and swiftly.
- It is effective when the classes are more than two.
- Given the independence assumption, the Naive Bayes classifier is more efficient and requires less data to train compared to the logistic regression model.
- It is more effective if the inputs are categorized rather than numeric.

Disadvantages:

- If a class in the learning phase has no observed data, the classifier considers the probability of that class as zero, making it unable to classify it. A smoothing technique, such as the Laplace estimator, can be an alternative solution.
- Achieving conditional independence in the real world is nearly impossible (5).

2.2.2. *Logistic Regression Algorithm*

Logistic regression is a statistical model for binary dependent variables, such as disease presence or absence. It uses the logistic function as the link function and its error distribution follows a multinomial distribution. Initially used in the medical field to predict disease probability, logistic regression is now widely applied across various scientific fields. It is a special case of the general linear model and differs from linear regression in key ways, such as using a Bernoulli distribution for the conditional distribution and predicting probabilistic outcomes bounded between zero and one (6).

2.2.3. *Perceptron Neural Network Algorithm*

The Perceptron is a supervised machine learning algorithm designed for binary classification tasks. It predicts class membership based on a weighted sum of input features. Introduced in 1957 by Frank Rosenblatt, the Perceptron is one of the earliest artificial neural networks. It is a linear classifier that makes predictions using a weighted sum of inputs. The Perceptron is a binary classifier that maps its input x (a vector of real numbers) to an output $f(x)$ (a binary-valued scalar) calculated as:

$$1)\ f(x) = \begin{cases} 1\ if\ w.x + b > 0 \\ 0\ otherwise \end{cases}$$

2.2.4. *K-Star Algorithm*

The K-Star algorithm is a cluster analysis method aimed at dividing n observations into k clusters, so that each observation belongs to the cluster with the closest mean. The K-Star algorithm can be described as a model-based learning method that uses entropy theory as a distance measure. These methods maximize the possibility of extracting valuable information from the available data by providing a consistent approach to manage symbolic features and missing values. In this algorithm, the distance from one sample to another is described by the complexity of converting one sample to another. The K-Star algorithm uses an entropy distance function. Interpeak distance is used to obtain the most similar samples from the dataset. Suppose that a and b are the samples under consideration; then $P*$ can be described as the probability of each path from a to b. Therefore, the relationship p can be expressed as follows (8).

E-ISSN: 2981-2569

2) $P^*(b|a) = \sum_{t\in p:t(a)=b}^{N} p(t)$

where t represents the value of T (T is a set of data transformations) and P is a probability function. Considering that P* has the following conditions:

3)

$$\sum_{b}^{N} P^*(b|a) = 1, \qquad 0 \le P^*(b|a) \le 1$$

According to the above relationships, the k-star function is expressed as follows [19]:

4) $K^*(b|a) = -\log P^*(b|a)$

That

5)

$$K^*(b|a) \ge 0, \qquad k^*(b|a) + k^*(c|b) \ge K^*(c|a)$$

The above statements represent whole numbers and are rewritten as below for continuous numbers [19].

6)

$$P^*(b|a) = p^*(b|a) = P^*(i) = \frac{s}{\sqrt{2s-s^2}}\left(\frac{1-\sqrt{2s-s^2}}{1-s}\right), i = |a-b|$$

7) $K^*(b|a) = K^*(i)$

$$= \frac{1}{2} log\ (\sqrt{2s-s^2} - log(s) + i[log(1 - s) - log\ (1 - \sqrt{2s-s^2})$$

where s is the parameter of the model and is variable between zero and one. Having these records, it is possible to select the most appropriate sample for the desired data by using the calculated probability values (8).

2.2.5. Decision Tree-Based (DTB) Algorithms

Decision trees are commonly applied classification algorithms that are easy to interpret and create. In this approach, the classification process involves constructing a tree composed of a conjunction of rules. The tree consists of internal nodes, branches, and leaf nodes, representing attributes, attribute values, and classes in the dataset, respectively. In the tree structure, the output of an internal node—namely, a branch—is transferred as an input to another internal node.

Several DTB classifiers are available in the literature, including CART, J48 Classification, Random Forest, and Modified Rotation Forest Classifier, Functional Tree, which are used to predict diabetes. These algorithms are executed individually and as base learners for bagging and boosting methods on the diabetes dataset in this experimental study.

2.2.5.1 Classification and Regression Trees (CART) Algorithm

The CART algorithm's classification process involves dividing the training set into progressively smaller subsets. The ideal result is to ensure that the same label exists for the leaf samples, thereby generating the tree. The criterion for selecting tree regression nodes is to minimize the impurity of nodes as much as possible. The lowest Gini coefficient for each feature is used as a standard for selecting test features in tree regression (9).

2.2.5.2 J48 Classification

J48 is a successor to the ID3 algorithm. Additional features of J48 include handling missing values, decision tree pruning, continuous feature value ranges, and rule derivation. In the WEKA data mining tool, J48 is implemented as the C4.5 algorithm with an open-source Java implementation. WEKA provides several options related to tree pruning, which can be used as a screening tool if there is a high probability of pruning. In other algorithms, the classification is done recursively until each leaf is pure, meaning the data classification should be as complete as possible. This algorithm creates rules, from which the specific identity of that data is generated. The goal is to gradually generalize a decision tree until a balance of flexibility and accuracy is achieved (4).

2.2.5.3 Random Forest

In Random Forest algorithms, more than one decision tree is constructed using randomly selected samples from the original dataset. Among all the equally probable constructed trees, a random one is selected (10).

2.2.5.4 Rotation Forest Algorithm

Rotation forests are classifier algorithms similar to Random Forests but address a major weakness of Random Forests—connected models. Decision trees can only partition feature space directly (along side axes). Random Forest, which predicts common results from n trees trained on bootstrap samples, can still face issues due to the network-like decision structure of the base trees. Rotational forestry minimizes this by randomly "visiting" every tree in the forest. This is done by randomly partitioning the sample components into several partitions, taking samples from these partitions, and applying PCA to obtain a "rotation matrix" (eigenvector matrix). The model's parts are rotated by multiplying the initial feature splits and the rotation matrix. The randomness provided by random sampling and splitting means that each tree in the forest "points in a different direction" in feature space, allowing the cluster to compute curves more efficiently than Random Forests (11).

2.2.5.5 Functional Tree

The functional tree algorithm provides a general framework for learning functional trees, which are multivariate classification or regression trees that use a combination of features in decision nodes, leaf nodes, or both. This algorithm uses a standard top-down recursive partitioning strategy to build the decision tree. The split at each node is univariate but considers both the original features in the data and newly constructed features using a feature constructor function. Multiple linear regression in the regression setting and linear discriminants or multiple logistic regression in the classification setting are used. The value of each new feature is the prediction of the generating function for each instance that reaches the node. In classification mode, a new feature is created for each class, and probability values are predicted. In regression mode, a new feature is created. Thus, the algorithm considers oblique divisions based on a combination of features in addition to the standard axis-parallel divisions based on the main features. To choose the dividing point, information gain in classification mode and variance reduction in regression mode are used. Once a tree has grown, it is pruned again using a bottom-up method. At each non-leaf node, three possibilities are considered: performing no pruning (leaving the subtree at the node in place), replacing the node with a leaf that predicts a constant, or replacing it with a leaf that predicts it. Constructor function that is created during the construction of the tree on the node. The error-based criterion C4.5 is used to make the decision. Prediction of a test sample is done using a functional tree by traversing the tree from the root to a leaf. At each decision node, a local constructor function is used to expand the feature set, and the decision test determines the path the sample will follow. Once a leaf is reached, the sample is classified using a constant or constructor function at that leaf, depending on what was put in place during the pruning process (12).

2.2.5.6 *Bagging Algorithm*

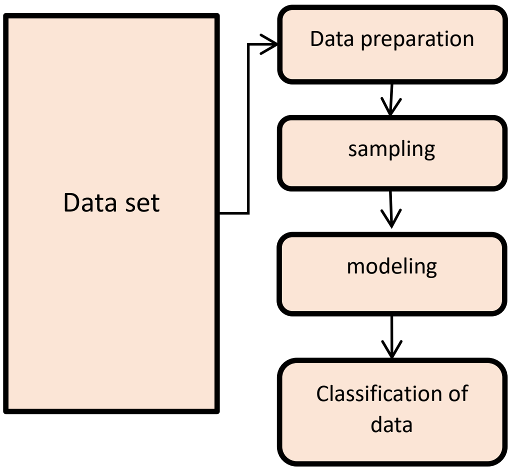
The Bagging algorithm, an ensemble classifier whose name is derived from the term "bootstrap aggregating," was introduced by Breiman in 1992. Hybrid classifiers combine multiple categories, each building its model on the data and saving this model. Finally, votes are taken for classification among these categories, and the class that gets the most votes is the final class. In the Bagging method, a subset of the original dataset is given to each of the classifiers. Each classifier observes a part of the dataset and builds its model based on that portion. Research has shown that the Bagging algorithm can be useful for algorithms like neural networks or decision trees that may generate different classes with slight variations in the samples. The Bagging algorithm is implemented with tree classifiers as the objective function of evolutionary algorithms (13).

3. Findings and Results

In this study, we use steps for implementation, which are shown in Figure 1.

Figure 1

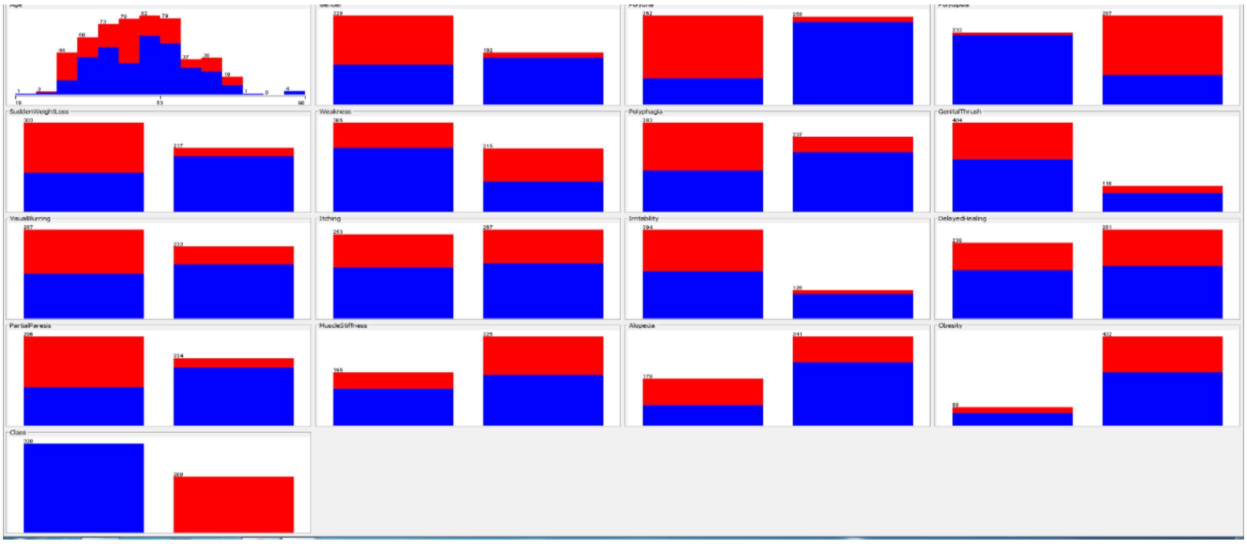
Flowchart of the implementation steps Supervised Learning Approach algorithm



In data preparation step, we uploaded a data set from the site /chistio.ir, which includes 520 samples (patients) and these samples have 200 diabetic patients and 320 non-diabetic patients, each sample 16 is reviewed in table1. In Figure 2, the data set in the Weka software in the form of a diagram are shown.

Figure 2

The data set in the Weka software



In the target class step, we specify the type of class to check all the features. During the sampling step, the number of training and test datasets is selected, with seed set to 1 and fold set to 10. In the modeling step, we choose the type of algorithm to use. Finally, in the data classification step, the classification accuracy is estimated based on the percentage of test samples or test datasets that are classified. The implementation results are shown in Table 2.

Table 2

Results of supervised learning algorithms

Evaluation criteria	k-star	Rotation Forest	Random Forest	Bagging	Logistic Regression	Simple Bayes	Perceptron neural network algorithm	J48	Functional tree
(Correctly)	95.7692 %	%97.1154	97.1154%	93.4615%	92.3077%	87.1154%	96.3462%	%95.9615	93.6538%
Correctly number of samples	498	505	505	486	480	453	501	499	487
(Incorrectly)	4.2308 %	2.8846%	2.8846%	%6.5386	7.6923%	12.8846%	3.6538%	4.0385%	6.3462%
Inaccurate number of samples	22	15	15	34	40	67	19	21	33
(Kappa)	0.9123%	%0.9393	0.939%	0.8621%	0.8378%	0.734%	0.923%	0.9156%	0.8673%
(Mean absolute error)	%0.0536	%0.0509	0.0531%	0.113%	0.1114%	0.149%	0.0398%	0.0549%	0.0841%
(Root mean squared error)	0.1711%	0.1429%	0.1468%	0.2243%	0.2521%	0.3184%	0.1638%	0.1975%	0.2382%
(Relative absolute error)	11.3161 %	10.749%	11.2098%	23.8723%	23.5292%	31.4737%	8.4109%	11.5905%	17.7568%
(Root relative squared error)	35.1777%	%29.3744	%30.1662	46.1059%	51.815%	65.4532%	33.6789%	40.5926%	48.9642%
(Total Number of Instances)	520	520	520		520	520	520	520	520

The number of samples that are correctly classified are shown, and the detection accuracy of the supervised algorithms is shown in Table 2. Equation (8) is used to calculate accuracy:

$$8)\ Accuracy = \frac{TP^{*}+TN^{\dagger}}{TP+TN+FP^{\ddagger}+FN^{\S}}$$

Accordingly, the number of wrongly classified samples and the classification error is shown. The amount of error is calculated based on equation (9):

$$9)\ Error = 100 - \frac{TP+TN}{TP+TN+FP+FN}$$

Table 3

Results of supervised learning algorithms

Evaluation criteria	k-star	Rotation Forest	Random Forest	Bagging	Logistic Regression	Simple Bayes	Perceptron neural network algorithm	J48	Functional tree
(Correctly)	95.7692 %	%97.1154	97.1154%	93.4615%	92.3077%	87.1154%	96.3462%	%95.9615	93.6538%

[*] - True Positive
[†] - True Negative
[‡] - False Positive
[§] - False Negative

E-ISSN: 2981-2569

Correctly number of samples	498	505	505	486	480	453	501	499	487
(Incorrectly)	4.2308 %	2.8846%	2.8846%	%6.5386	7.6923%	12.8846%	3.6538%	4.0385%	6.3462%
Inaccurate number of samples	22	15	15	34	40	67	19	21	33
(Kappa)	0.9123%	%0.9393	0.939%	0.8621%	0.8378%	0.734%	0.923%	0.9156%	0.8673%
(Mean absolute error)	%0.0536	%0.0509	0.0531%	0.113%	0.1114%	0.149%	0.0398%	0.0549%	0.0841%
(Root mean squared error)	0.1711%	0.1429%	0.1468%	0.2243%	0.2521%	0.3184%	0.1638%	0.1975%	0.2382%
(Relative absolute error)	11.3161 %	10.749%	11.2098%	23.8723%	23.5292%	31.4737%	8.4109%	11.5905%	17.7568%
(Root relative squared error)	35.1777%	%29.3744	%30.1662	46.1059%	51.815%	65.4532%	33.6789%	40.5926%	48.9642%
(Total Number of Instances)	520	520	520		520	520	520	520	520

The first column (TP Rate): shows the correctness of the classification for each type of class.

The second column (FP Rate): the amount of incorrect data classification.

The third column (Precision): indicates the accuracy of the classification of each of the classes in the data set, which is calculated based on equation (10).

10) $Precision = \frac{TP}{TP+FP}$

The fourth column (Recall): the ratio of the number of correct items classified by the algorithm from one class to the number of items in the said class, which is calculated based on the relationship (11).

11) $ReCall = \frac{TP}{TP+FN}$

The fifth column (F-Measure): According to the calculations made for the Precision and Recall criteria, at this stage, the value of the F-Measure weighted quantity can be calculated. It is between the two quantities Precision and Recall. For a classification algorithm in ideal conditions, the value of this quantity is equal to one and in the worst case it is equal to zero. This parameter is calculated using equation (12).

12) $\mathrm{F-Measure} = 2 * \frac{Precision*ReCall}{Precision+ReCall}$

The sixth column (ROC Area): expresses the degree of correctness and incorrectness of the classification according to ROC.

As shown in Table 4 Accuracy rate in classifier algorithms for diabetes prediction are compared.

Table 4

Comparison of supervised learning algorithms

Algorithm	Algorithm Evaluation Criterion (Accuracy Rate)
Bagging	93.4615%
Rotation Forest	97.1154%
Random Forest	97.1154%
Simple Bayes	87.1154%
k-star	95.7692%
Perceptron Neural network	96.3462%
Logistic Regression	92.3077%
J48	93.6538%
Functional tree	95.9415%

4. Discussion and Conclusion

The study conducted provides a comprehensive evaluation of various supervised machine learning (ML) algorithms for predicting diabetes. The findings highlight the effectiveness of these algorithms in accurately classifying individuals as diabetic or non-diabetic based on clinical data. This section will discuss the strengths and limitations of the applied algorithms, comparing them with previous research, and suggest areas for future improvement.

The application of machine learning in healthcare, particularly in predicting diabetes, has shown remarkable success. The supervised learning algorithms utilized in this study, including Logistic Regression, Random Forest, and Neural Networks, have proven to be effective tools. Logistic Regression, in particular, demonstrated superior performance with high classification accuracy (CA), area under the curve (AUC), F1 score, precision, and recall (6). This algorithm's robustness and ease of interpretation make it a reliable choice for binary classification problems in medical diagnostics.

Random Forest and Neural Networks also performed well, aligning with the findings of Dey et al. (3), who reported an accuracy of 82.35% for an Artificial Neural Network (ANN) with Min-Max scaling (MMS) on the Pima dataset. Alehegn et al. (2019) also highlighted the effectiveness of ensemble approaches, with their Random Forest model achieving an accuracy of 93.62% for the PIDD dataset. The use of ensemble methods, such as Random Forest and Bagging, enhances the prediction accuracy by combining multiple models, thereby reducing the risk of overfitting and improving generalization (10).

The results of this study are consistent with those of previous research, indicating that machine learning algorithms can significantly improve the accuracy of diabetes prediction. For instance, Sonar and Jaya Malini (14) achieved an 85% accuracy rate with a Decision Tree algorithm, while Jain et al. (15) reported an accuracy rate of 87.88% with a Neural Network. These findings underscore the potential of machine learning to revolutionize diabetes diagnosis and management by providing reliable, data-driven predictions.

However, the study also revealed some limitations. Despite the high accuracy rates of the algorithms, there is still room for improvement in terms of handling diverse datasets and integrating more complex features. For example, the Simple Bayes Algorithm, while effective, assumes independence between features, which is rarely the case in real-world data. This assumption can limit the algorithm's performance, particularly when dealing with highly interdependent clinical features (5).

Moreover, the dataset used in this study, sourced from chistio.ir, includes 520 samples with 16 features. While this dataset is sufficient for initial evaluations, larger and more diverse datasets would provide a more robust assessment of the algorithms' performance. Future research should consider incorporating datasets from multiple sources and with more varied features to enhance the generalizability of the findings.

The study also highlighted the importance of feature selection and preprocessing. The features considered in this study, such as Age, Gender, Polyuria, and Polydipsia, are critical indicators of diabetes. Proper preprocessing, including scaling and normalization, is essential to ensure that the features contribute effectively to the model's predictive power. Techniques such as Min-Max scaling and normalization have been shown to improve model performance, as demonstrated by Dey et al. (3).

The study also identified several areas for future research. Firstly, there is a need to evaluate the performance of these algorithms on larger and more diverse datasets to improve their generalizability. Secondly, integrating more complex features and employing advanced preprocessing techniques can further enhance the accuracy and reliability of the predictions. Finally, exploring the use of ensemble methods and deep learning algorithms could provide additional insights and improvements in diabetes prediction.

Authors' Contributions

M.M.H.S. was primarily responsible for conceptualizing the study, designing the trial, and overseeing the intervention delivery. He also managed participant recruitment and data collection. S.A.A., the corresponding author, led the data analysis, interpreted the results, and was the primary contributor to writing and revising the manuscript. Both authors collaboratively developed the intervention materials and ensured the ethical conduct of the study. They have both read and approved the final manuscript for publication.

Declaration

In order to correct and improve the academic writing of our paper, we have used the language model ChatGPT.

Transparency Statement

Data are available for research purposes upon reasonable request to the corresponding author.

Acknowledgments

We would like to express our gratitude to all individuals helped us to do the project.

Declaration of Interest

The authors report no conflict of interest.

Funding

According to the authors, this article has no financial support.

Ethics Considerations

The study placed a high emphasis on ethical considerations. Informed consent obtained from all participants, ensuring they are fully aware of the nature of the study and their role in it. Confidentiality strictly maintained, with data anonymized to protect individual privacy. The study adhered to the ethical guidelines for research with human subjects as outlined in the Declaration of Helsinki.

References

1. Deberneh HM, Kim I. Prediction of Type 2 Diabetes Based on Machine Learning Algorithm. International Journal of Environmental Research and Public Health. 2021;18(6):3317. [PMID: 33806973] [PMCID: PMC8004981] [DOI]
2. Ahmed U, Issa GF, Khan MA, Aftab S, Khan MF, Said RAT, et al. Prediction of Diabetes Empowered With Fused Machine Learning. IEEE Access. 2022;10:8529-38. [DOI]
3. Dey SK, Hossain A, Rahman MM, editors. Implementation of a web application to predict diabetes disease: an approach using machine learning algorithm. 2018 21st international conference of computer and information technology (ICCIT); 2018: IEEE. [PMCID: PMC6334885] [DOI]
4. Alehegn M, Joshi RR, Mulay P. Diabetes analysis and prediction using random forest, KNN, Naïve Bayes, and J48: an ensemble approach. Int J Sci Technol Res. 2019;8(9):1346-54.
5. Ameri H, Alizadeh S, Barzegari A. Extracting knowledge from the data of diabetic patients using decision tree method C5. Health Management. 2012;16(53):58-72.
6. Park H-A. An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to

Nursing Domain. J Korean Acad Nurs. 2013;43(2):154-64. [PMID: 23703593] [DOI]

7. Gandomi M, Dolatshahi Pirooz M, Varjavand I, Nikoo MR. Application of Multilayer Perceptron Neural Network and Support Vector Machine for Modeling the Hydrodynamic Behavior of Permeable Breakwaters with Porous Core. Journal Of Marine Engineering. 2019;15(29):167-79.

8. Ahmadi F, Maddah MA. Development of Wavelet-Kstar Algorithm Hybrid Model for the Monthly Precipitation Prediction (Case Study: Synoptic Station of Ahvaz). Iranian Journal of Soil and Water Research. 2021;52(2):409-20.

9. Zhu F, Tang M, Xie L, Zhu H. A classification algorithm of CART decision tree based on MapReduce attribute weights. International journal of performability engineering. 2018;14(1):17. [DOI]

10. Moshrefzadeh S, Rahmani Seryasat O, Ravaei B. Intelligent intrusion Detection of computer networks using Random Forest Algorithm. Transactions on Machine Intelligence. 2019;2(1):48-58.

11. Ani R, Jose J, Wilson M, Deepa OS, editors. Modified Rotation Forest Ensemble Classifier for Medical Diagnosis in Decision Support Systems. Progress in Advanced Computing and Intelligent Engineering; 2018 2018//; Singapore: Springer Singapore. [DOI]

12. Alpaydin E. Introduction to machine learning: MIT press; 2020.

13. Taser PY. Application of Bagging and Boosting Approaches Using Decision Tree-Based Algorithms in Diabetes Risk Prediction. Proceedings [Internet]. 2021; 74(1). [DOI]

14. Sonar P, JayaMalini K, editors. Diabetes Prediction Using Different Machine Learning Approaches. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC); 2019 27-29 March 2019. [DOI]

15. Jain B, Ranawat N, Chittora P, Chakrabarti P, Poddar S. WITHDRAWN: A machine learning perspective: To analyze diabetes. Materials Today: Proceedings. 2021. [PMID: 35155131] [PMCID: PMC8820461] [DOI]

16. Salah MM, Shekari E, Hassani H, Salehi A. Prediction of Marital Conflicts Based on Mindfulness in Couples Facing Extra-Marital Relationships Attending Counseling Centers in Fars Province. Transactions on Data Analysis in Social Science. 2024;6(1):1-13.

17. Moslehi Z, Robat Milli S. The Purpose of Determining Prediction of Quality of Life Based on the Feeling of Psychological Coherence and Tolerance of Distress in Students. Transactions on Data Analysis in Social Science. 2023;5(2):104-10.

18. Ghayoumi Zadeh H, Fayazi A, Rahmani Seryasat O, Rabiee H. A Bidirectional Long Short-Term Neural Network Model to Predict Air Pollutant Concentrations: A Case Study of Tehran, Iran. Transactions on Machine Intelligence. 2022;5(2):63-76.

19. Saleh B, Hasanpour H. Diabetes Diagnosis from Big Data using Fuzzy-Neural Chaotic Tree. Transactions on Machine Intelligence. 2023;6(2):104-13.