Iraqi Journal of Statistical Sciences

www.stats.mosuljournals.com

Classification of Diabetes Data Set from Iraq via Different Machine Learning Techniques

Dilshad Altalabani1[1] Fevzi Erdogan[2]

[1]Department of Computer Science, Directorate of Information Technology and Statistics,Sulaimani Polytechnic University, Garmian University, Kalar, Iraq.
[2]Department of Statistics, University of Van Yuzuncu Yil, Van, Turkey.

Article information

Article history:
Received March 13 ,2024
Accepted April 26 ,2024
Available online June 1, 2024

Keywords:
Machine Learning, Classification, Diabetes

Correspondence:
Dilshad Altalabani
dilshad.saeed20@gmail.com

Abstract

Diabetes has become one of the most prevalent diseases in Iraq and is listed as one of the leading causes of death. Machine learning provides effective information extraction results by creating predictive models from diagnostic medical datasets collected from diabetes patients in Iraq.

In this study, we applied machine learning classification to compare and contrast the performances of classification and regression trees (CART), support vector machines (SVM), random forests (RF), linear discrimination analysis (LDA), and K-nearest neighbors (KNN). We sought to design a model that can predict with maximum accuracy the probability that a person has, is healthy, or is expected to develop diabetes in the future using the two scales of accuracy and kappa.

Based on the results obtained from the algorithms, it showed that the accuracy and sequence of the algorithms concerning the training data were Random Forest (RF), Classification and Regression Trees (CART), Support Vector Machine (SVM), Linear Discrimination Analysis (LDA), and K-Nearest Neighbors (KNN). While the test data results showed some differences, the sequence of the algorithms was as follows: SVM, RF, CART, LDA, and KNN were the highest, respectively. The training data set refers to the samples that were used to construct the model, whereas the testing data set is used to evaluate the model's performance.

Based on the assessment criteria discussed above, we chose the best machine learning approach to predict diabetes mellitus in Iraq to achieve high performance. All of the strategies listed above are approximated using a supervised diabetes testing dataset. The approach that achieves the maximum performance in terms of accuracy and kappa is regarded as the best option. Based on the results, it can be seen that the SVM and RF algorithms predicted diabetes with more accuracy.

DOI 10.33899/IQJOSS.2024.183258 , ©Authors, 2024, College of Computer Science and Mathematics University of Mosul.

1. Introduction

International Diabetes Federation (IDF) (2017) data shows that hundreds of millions live with diabetes worldwide. Diabetes now routinely tops lists of the leading causes of death worldwide. Over the past 30 years, based on World Health Organization (WHO) (2018) data, diabetes prevalence has increased rapidly, especially in low- and middle-income countries.[1] The International Diabetes Federation (IDF) (2017) reported an 8.8% (425 million people) prevalence among adults in 2017. The Middle East and North Africa (MENA) region has the second highest rate at 9.2%. The MENA region is projected to grow 110 percent between 2017 and 2045, from 329 million to 629 million (IDF) (2017). Diabetes is a significant illness, with a 10.7% death rate in adults aged 20–79. The MENA Region, including Iraq, has the highest rate of fatalities due to diabetes in individuals under 60, ranking second among IDF regions (IDF) (2017). However, Only 2.9% of the world's diabetes investment is directed towards researching the development and consequences of the disease, leaving a significant knowledge gap (WHO) (2018). Iraq faces 1.4 million diabetes cases, but insufficient epidemiological studies and RCTs make it difficult to understand the prevalence and effective therapies for the population as described by Mansour *et al.* (2014).

Khanam and Simon (2021) state that diabetes identification is one of the most difficult challenges in healthcare. Baran (2020) the rapid increase of so-called data sources gives diversity and importance to studies in machine learning. The development of technology has led to the introduction of multi-label classification for increasing datasets. Alan (2020) choosing the optimal classifier is one of the most important difficulties when developing a model in machine learning. Parthiban and Srivatsa (2012) data classification is a typical job in machine learning. Data mining is critical for extracting knowledge from huge datasets. Keskin (2018), Nahzat and Yaanolu (2021), in recent years, several academics have discussed their experiences with various machine learning methods, including Decision Tree, Naive Bayes, Random Forest, and K-Nearest Neighbour Support Vector Machine. Research has shown that machine learning algorithms can predict outcomes for a variety of diseases with a high degree of accuracy. The power of machine learning algorithms comes from their capacity to handle vast amounts of data, mix data from many sources, and incorporate fundamental knowledge into their research.

The focus of this study is to develop prediction models using diagnostic and interventional datasets from diabetic patients in Iraq. We employed various machine learning techniques while considering their features and performance, and compared them to obtain the best disease prediction. We explored multiple supervised learning algorithms in the R programming language. Our study employs machine learning classification algorithms to predict the likelihood of diabetes. We evaluated the performance of all algorithms across multiple measures and found that the Support Vector Machine and Random Forest machine learning classification algorithms achieved perfect accuracy.

2. Materials and methods

2.1. Diabetes dataset

The data for this study was initially collected from the laboratories of Medical City Hospital and the Specialized Centre for Endocrinology and Diabetes Al-Kindy Teaching Hospital in Baghdad, Iraq's capital, and contains 1000 records of diabetes patients of all ages. Attributes of the dataset (Gender, AGE, Urea, Cr (Creatinine ratio), HBA1c (Haemoglobin A1c Test), Chol. (Cholesterol), TG (Triglycerides), HDL (High-Density Lipoprotein, or good cholesterol), LDL (Low-Density Lipoprotein, sometimes called bad cholesterol), VLDL (Very Low-Density Lipoproteins), BMI (body mass index), and CLASS (Diabetic=Positive (P), Non-Diabetic=Negative (N), or Predict-Diabetic=Y). The data set was acquired from a particular location. https://data.mendeley.com/datasets/wj9rwkp9c2/1

2.2.Methods

Supervised machine learning algorithms give critical and high-accuracy results for prediction. Classification and Regression Trees (CART), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), and Random Forest (RF) were utilized. Rebala *et al.*, (2019) showed the algorithms were evaluated based on metrics like accuracy and kappa to determine the best model for predicting. Resampling was used to ensure compatibility between the training scheme and models. Rating metrics for each algorithm were stored, and the model was tested using the test dataset. The prediction confusion matrix was created to confirm the results. The error rate for the best model is then determined to optimize the prediction. An extensive collection of data points with responses, also known as a labeled data set, was delivered to the learning algorithm. As a result, the algorithm should be able to predict the result and respond correctly the next time it is presented with a new data point based on essential qualities. Where the model learns about different types of inputs. The models were trained using a labeled dataset in guided learning.

2.3.R Studio

R is an open-source statistical computing and graphics language that supports various statistical methods like linear and nonlinear modeling, statistical testing, time series analysis, classification, and clustering. Ramasubramanian and Singh (2019) showed that it is easy to learn and robust, making it suitable for academics and those with little programming experience. R's easy calculation of statistical features makes it a popular tool for data analysts and statisticians.

2.4.Model Diagram

The architecture of the Proposed Approach in Figure 1 below in the form of model diagram.

2.5.Classification and Regression Trees (CART)

The first algorithm we modeled was CART. Breiman *et al.*(1984) showed the theoretical foundations and practical applications of CART were first presented by Breiman, Friedman, Olshen, and Stone. CART has evolved into a powerful exploratory method for data analysis since then as computer power has increased. Although many statisticians refer to CART as a single statistical method, it refers to two different analytic methods: classification trees (CT) and regression trees (RT), depending on the dependent variable's measurement nature. CART is a heuristic tree method that unravels the relationships between a dependent variable and a set of predictors in general (independent variables). Genuer and Poggi (2020) showed the root is at the top of a CART tree, which is an upside-down tree. The tree's leaves are nodes with no descendants, while the others are nonterminal nodes with two child nodes. As a result, the tree is described as binary. A condition (a question) labels nonterminal nodes, while a class label or a response variable value label leaves. When given a tree, it is simple to use for prediction. Indeed, going through the only path from the root to a leaf by answering the sequence of questions posed by the successive splits and reading the value of the dependent variable labeling the reached leaf suffices to determine the predicted value of the dependent variable for a given independent variable. When traversing the tree, the rule is to go to the left node if the condition is verified and to the right if it is not.

We constructed a tree for our model CLASS (dependent variable) with all other independent variables for training data. As we can see, there are 15 nodes in this tree. As shown in Figure 2, the root is at the top, the leaf is at the bottom, and the most crucial variable in the prediction model (BMI) is at the top. If we have a new patient whose BMI at node 1 is less than 25, we go to node 2, the following most influential variant in the model: HbA1c. If it is less than 5.6, we go to node 3, the Chol variable, and if it is less than 4.9, we go to node 4, the BMI variable; if it is less than 23, it will reach a terminal node or decision node. We look at the probability that the response variable equals one within class N (Non-Diabetic=Negative) with a probability up to 1; otherwise, if it is greater than 23, it will be within class N with a probability of about 0.9. In node number 7, if the variable TG is less than 1.9, we go to node 8, and it will be within class N with a probability of more than 0.6; otherwise, it will be within class Y with a probability of up to 1. In node number 10, if the variable Hb1Ac is less than 6.4 and we go to node 11, it will be within class P (Diabetic=Positive) with a probability up to 1; otherwise, it will be within class Y with a probability up to 1. In node number 13, if the variable age is less than 43, we go to node 14, and then it will be within class Y (Predict-Diabetic=Y) with a probability greater than 0.8; otherwise, it will be within class Y with a probability up to 1.

The first algorithm we modeled was CART for data consisting of the response variable class, which consists of three classes: N (Non-Diabetic=Negative), P (Diabetic=Positive), and Y (Predict Diabetic) with ten features. The data were divided into training data at 80% and test data at 20%. We then obtained a model with an accuracy of 0.9823 and a Kappa of 0.9381. We obtained more confirmed results in the confusion matrix, showing the number of correct classifications at 778 out of 792, 88 N, 35 P, and 655 Y versus 14 missed classifications, 10 in N, and 4 in Y in the prediction model. The model's results were compared with the test data to determine the accuracy of the models. We obtained an accuracy of 0.976 and a Kappa of 0.905 with a confidence interval of 0.9448 to 0.9921, considered high accuracy. We obtained more confirmed results in the confusion matrix, showing the number of correct classifications at

203 out of 208, 15 N, 12 P, and 176 Y versus five missed classifications, 3 in N and 2 in Y in the prediction model. The confusion matrix for the training and test data is shown in Table 1.

2.6. Support Vector Machines (SVM)

The second algorithm we modeled was SVM. Rebala *et al.* (2019) illustrated Support vector machines (SVM) were introduced in the early 1990s and have proven effective in real-world classification and regression problems. They can work with sparse data and are a compromise between parametric and non-parametric approaches. Support vector machines as Suthaharan (2016) showed are a classic machine-learning technique that can classify large amounts of data. It uses a simple mathematical model and manipulates it to allow linear domain division. There are two types of support vector machines: linear and nonlinear. Nonlinear support vector machines are used when the data domain cannot be divided linearly but can be transformed into a feature space using a kernel function. The goal of the SVM as Brown *et al.* (1999) illustrated, is to find a memorable line known as the hyperplane, that separates the classes and is the furthest away from both classes. The test data are included in the class on whichever side of the boundary it lies on after the hyperplane is found using the training data. It provides maximum sample discrimination by linearly decomposing the nonlinear hyperplane sample space. A support vector machine locates the best-separating hyperplane between members and non-members of a particular class in an abstract space.

In a simple case, the feature vector will be x, a linear classifier that creates a hyperplane. All y values are more significant than are in class 1, whereas all other y values are in class 2. The feature vector is expected to be a linearly represented table in this linear equation. In Figure 3, the red lines represent the valid class boundaries of a, b, and c. All of these boundaries correctly divide the data points into two groups. Line b, on the other hand, gives both classes the most room to maneuver. SVM seeks out boundaries that maximize the data points' margins. The points closest to lines a and c are the support vectors that provide the class boundary lines.

We then obtained the model by applying the SVM algorithm with an accuracy of 0.954 and a kappa of 0.825. In the confusion matrix, we obtained the correct classification of 954 out of 1000 for N 96, P 29, and Y 830, while the missed classification was N 17, P 3, and Y 26 in the prediction model. We use hyperparameter optimization (tuning), which helps us select the best model by tuning the function. We use the support vector machine model and cost capture to capture the constraint violation. The accuracy of the models was determined by comparing the results of model optimization with the test data. We obtained an accuracy of 0.999 and a Kappa of 0.996, considered perfect high accuracy. In the confusion matrix, we use the model on the entire dataset to obtain more results, showing the number of correct classifications at 999 out of 1000, 103 N, 53 P, and 843 Y versus one missed classification in the P class in the prediction model. As shown in Table 2.

based on the confusion matrix, misclassification error, and accuracy for hyperparameter optimization (tuning), the model is considered to be the best based on the results. The darker sections indicate better outcomes, which translate into reduced misclassification error, lower cost, and varied epsilon values, as shown in Figure 4. The summary of the tuning of the support vector machine model gives us the following parameters: sampling method: 10-fold cross-validation, the best parameters' epsilon = 0, the best parameters epsilon cost = 256, and best performance error: 0.042, as shown in Table 3.

2.7. Random forest (RF)

The bagging method, first described by Breiman, is a historical example of a random forest method, a machine learning algorithm used for classification and regression. Random forests as Genuer and Poggi (2020) illustrated, consisting of decision trees, have excellent predictive capacity and versatility, making them widely used in various applications. They enable simultaneous assessment of qualitative and quantitative explanatory elements without preprocessing, making them suitable for analyzing both traditional data with a higher number of observations and high-dimensional data with a higher number of variables. As a result, statisticians and data scientists increasingly focus on random forests as a preferred methodology. *Random forests* as Rebala *et al.* (2019) illustrated are a successful and intuitive classification and regression model based on decision tree structures. These supervised machine learning techniques outperform single decision trees in prediction, providing a clear route to a solution. Random forests simplify identifying characteristics contributing to regression or classification and their relevance to the conclusion. These supervised techniques are essential for enhancing the effectiveness of machine learning techniques in various fields. A Random Forest algorithm uses numerous decision trees on distinct subsets of a dataset to raise the predicted accuracy by averaging them. Rather than relying on a single decision tree, the random forest collects the predictions from each tree and predicts the final output based on the majority votes of the predictions, as shown in Figure 5. The more trees in a forest, the more accurate a model will be, avoiding overfitting.

.

Random Forest is used because it takes less time to train than other algorithms. The program then predicts output with excellent accuracy and sprints even with a vast dataset. Finally, accuracy can be maintained even when significant data is absent. The random forest was formed in two phases: the first was to combine N decision trees to build the random forest, and the second was to make predictions for each tree created in the first phase. In the following stages, we are used to demonstrating the working process:

Step 1: Pick a random K data point from the training set.
Step 2: Construct decision trees for the subsets of data points you have chosen.
Step 3: Choose the value of N, which represents the number of decision trees generated.
Step 4: Repetition of Steps 1 and 2.
Step 5: Find the forecasts of each decision tree for new data points and assign them to the category with the most votes.

The third algorithm we modeled is RF. We divided the data into training data at 80% and test data at 20%. To get the model, we use the Random Forest function. Next to the confusion matrix, we have a class error matrix that includes errors for each class. Then, we use our model on training data to get a confusion matrix that gives us perfect accuracy for the model with an error of 0.02352041, 0.12195122, and 0.00591715 for classes N, P, and Y, respectively. From the model from the training data, we will get overall statistics that give us perfect results, such as Accuracy 1 and Kappa 1. We get more results in the confusion matrix for train data, showing the number of correct classifications at 802 out of 802, 85 N, 41 P, and 676 Y without missing classifications in the prediction model. By using the random forest model (RF) for test data, we will have a confusion matrix with prediction data and compare the test data with our model. The accuracy is 0.9899, and the Kappa is 0.9623. In the confusion matrix for test data, we get more results, showing several correct classifications of 196 out of 198, 18 N, 11 P, and 167 Y, with just two missed classifications, 1 in N and 1 in Y, in the prediction model shown in Table 4. By plotting the (RF) model, we discover that as the number of trees grows, the out-of-bag error initially drops and becomes more or less constant, so we cannot improve this error after about 400 trees, as shown in Figure 6. To get the most negligible error for (OOB) and optimize this parameter, we use the (tuneRF) function, and the result and graph show that it is equal to 6, as shown in Figure 7. Using a plot for variable importance, we can determine which variable plays a vital role in the model. In the random forest model, we make a plot for our model RF that shows us the error rate. Show graph every variable importance in the random forest in the mean decrease accuracy and mean decrease Gini in Table 7, Figure 8.

2.8. Linear Discriminant Analysis (LDA)

The fourth algorithm we modeled was LDA. Discriminant Analysis and Other Linear Classification Models: In general, discriminant or classification strategies aim to group samples based on predictor features, and the way to accomplish this varies by methodology and follows a mathematical path. Kuhn and Johnson (2016) illustrated the roots of linear discriminant analysis date back to Fisher in 1936 and Welch in 1939. **For the linear discriminant analysis,** Croux (2008) showed **the sample mean and covariance matrices were taken from different groups of the training sample**. Linear Discriminant Analysis (LDA) is a method for lowering the dimension of supervised classification applications. It depicts group distinctions, such as separating two or more classes. It is a method of projecting the properties of a higher-dimensional object onto a lower-dimensional surface. We have two courses, for example, and need to separate them properly. Classes contain a wide range of characteristics.

Using only one characteristic as James *et al.* (2021) illustrated to categorize them may result in some overlap. As a result, the number of attributes necessary for accurate categorization will continue to increase. Assume we have two sets of data to classify, each of which belongs to a distinct class. There is no straight line on a 2D graph that can entirely separate the two groups of data points. As a result, in this instance, LDA is used to reduce the two-dimensional graph to a one-dimensional graph, increasing the separability of the two classes in Figure 9. A new axis (in red) is drawn in the 2D graph to optimize the distance between the two classes' meanings while minimizing variance within each class. In alternative terms, the recently established axis increases the disparity between the data points belonging to the two distinct groups. All data points from the classes are displayed on this new axis when the criteria mentioned above are applied. Linear discriminant analysis, in this case, utilizes both axes (X and Y) to establish a new axis and projects data onto it to maximize the separation of the two categories, reducing the 2D graph to a 1D graph. LDA uses two criteria to construct a new axis.

1. Maximize the difference between the two classes' means.
2. Reduce the amount of variety within each class.

Linear discriminant analysis (LDA) is a statistical technique employed to classify subjects into distinct groups, with each category being precisely defined. It helps to find the linear combination of the original variables that provides the best possible separation between the groups. The essential purpose is to estimate the relationship between a single categorical dependent variable and a set of independent variables. We want to carry out a discriminant analysis that will help find the linear combination of these ten variables, giving us the best possible separation among these three groups or three different classes. We will randomly partition the data set into training and test data sets by sampling with replacement. We will make our model by function LDA for linear discriminant analysis for the respondent variable class as a function of all the other ten variables for training data. In the below table, we have the coefficients of linear discriminants for LD. The first discriminant function is a linear combination of the ten variables, and the discriminant functions are scaled LD1 and LD2 for every ten variables in Table 6 and Figure 10.

We have the proportion of trees that can tell us the percentage separation. The first discriminant function's achieved percentage separation is 0.9785, which is exceptionally high. The second discriminant function, in contrast, achieved a relatively low percentage separation of 0.0215, indicating that it is relatively difficult to distinguish between the first and the second categories. The dataset was partitioned into two subsets: the training dataset, which accounted for 80% of the data, and the test dataset, which accounted for the remaining 20%. Subsequently, the model was acquired, exhibiting a misclassification error rate of 0.1013767, an accuracy level of 0.898, and a Kappa coefficient of 0.64. In the confusion matrix, we obtained the correct classification of 718 out of 799 for N 70, P 13, and Y 635, while the missed classification was N 33, P 18, and Y 30 in the prediction model. The model's results were compared with the test data to determine the model's accuracy. We got a miss classification error of 0.06965174, an accuracy of 0.93, and a Kappa of 0.75, which is considered outstanding accuracy. The numbers on the diagonal indicate correct classification, while the off-diagonal numbers indicate misclassification. In the confusion matrix, we get more results, showing the number of correct classifications at 187 out of 208, 20 N, 3 P, and 164 Y versus 14 missed classifications, 4 in N, 6 in P, and 4 in Y in the prediction model, as shown by the confusion matrix in Table 7 below.

2.9. K-Nearest Neighbors (K-NN)

The fifth algorithm we modeled was K-NN. Breiman *et al.* (1984) initially introduced the theoretical underpinnings and pragmatic implementations of the Classification and Regression Trees (CART) algorithm. CART has evolved into a powerful exploratory method for data analysis as computer power has increased. Kramer (2013) illustrated that the nearest neighbor classification, also known as K-nearest neighbors (KNN), is based on the premise that the patterns closest to a target pattern for which we are looking for a label provide important information. KNN labels the most K-nearest patterns in data space with a class label. To do this, we must be able to define a similarity measure in the data space.

Rebala *et al.* (2019) showed that regression and classification problems can be solved with the KNN technique. Finding the K-nearest neighbors for a given data item is a straightforward concept. Similar objects are closer together, according to the notion. In this case, the term near refers to a distance metric as fundamental as the Euclidean distance between two places. By locating the largest class of items close to the test data, we can deduce that it belongs to the largest class. We search the database for the K points closest to a new data point for which we need to predict the classification. The nearest neighbors of the new data point are assigned to the same class or cluster. Take the average of the label values for these K-nearest neighbors in order to solve regression problems. When categorizing questions, we consider the majority. The distance between the test data and all training data points will be determined, and KNN will assist in determining which test data class to utilize. Then, choose the number of K points most similar to the test data. The K-nearest neighbor method analyzes the probability of the test data belonging to each training data class K being evaluated, and the one with the highest probability is preferred. In regression, the value is the average of the chosen K training score. Suppose there are two classes, A and B, and we have a new data point, x_1. We want to determine which classes will be the data points. We will require the K-NN method to address this type of problem. We can quickly define the class or classes of a data set using K-NN. Consider Figure 11.

We will partition the data set into training and testing data sets to two independent sample sizes with a replacement of 80% for training data, which will take 792 observations, and 20% for testing data, which will take 208 observations. A majority vote of its neighbors determines the classification of an item. If k = 1, the item is assigned to the class of the object's single nearest neighbor. The outcome is determined by whether k-NN is used for classification or regression. It has widespread application in the medical field.

Before making the k-nearest neighbor model, we will perform resampling to find the best model by choosing the tuning parameters' values using train control, and this will specify the resampling scheme. TrControl: The caret package specifies the resampling scheme used for cross-validation to find the optimal tuning parameters. We will use repeat CV (cross-validation) to develop the model. For several recent iterations, ten and repeat, which are several complete sets of folds to repeat, this cross-validation is 3. That means it repeats the whole thing three times. Then, we use the function that has a repeatable outcome. We fitted the model, and we are using training data. Our response variable is class, and we create the model by training data between classes versus all the independent variables, which are HbA1c, BMI, AGE, TG, VLDL, Chol, Urea, Cr, HDL, and LDL. As it shows, we have 792 samples, ten predictors, and three classes. In the resampling cross-validation, we have ten folds and repeat them three times, indicating that each cross-validation training data set is split into ten parts or ten folds; nine are used for creating the model, and the remaining one is used for assessing the model. The model accuracy and kappa values have been assessed, and we get listed for various values of k. Accuracy is used to select the optimal model using the most significant value we get when k = 7, as shown in Table 8 and Figure 12.

Out of ten values for all variables, the critical values are spread between 0 (zero) and 100, and we can recognize that variables Hb1Ac and BMI are the most important. In contrast, variable LDL turns out to be the least important. Table 9 displays the classes of variables in order of maximum importance. The fifth algorithm we modeled was k-NN. The data was divided into training data at 80% and test data at 20%. Then, we obtained the model from training data, and the model results had a classification error of 0.094, an accuracy of 0.906, and a Kappa of 0.658, considered good accuracy. In the confusion matrix, we get more results, showing the number of correct classifications at 718 out of 792: 59 N, 19 P, and 640 Y versus 40 missed classifications: 27 in N, 14 in P, and 33 in Y in the prediction model. The model's results were compared with the test data to determine the model's accuracy. We got a classification error of 0.101, an accuracy of 0.899, and a Kappa of 0.6187, which is considered good accuracy. In the confusion matrix, we get more results, showing the number of correct classifications at 187 out of 208, 11 N, 6 P, and 170 Y versus 21 missed classifications, 10 in N, 6 in P, and 5 in Y in the prediction model, as shown in Table 10.

2.10. Comparison of Machine Learning Algorithms

The goal of comparing machine learning algorithms is to discover the strength and clarity of the algorithms in giving better results, predictive models that last for long periods, are easy to evaluate, and give solid and multiple statistical indicators. Showing these practical qualities gives clear indications, which makes machine learning algorithms extremely important. We believe that comparing machine learning algorithms is significant in and of itself. There are several key advantages to successfully comparing multiple trials. The fundamental goal of model comparison and selection is to achieve unquestionably improved machine learning solution performance. The other goal is identifying the optimal algorithms that meet the data and business needs. At their most basic, machine learning models are statistical equations that run at high speeds on several data points to arrive at a result. As a result, statistical tests on the algorithms are crucial for fine-tuning them and determining if the model's equation best matches the dataset at hand. In the previous stages, we also presented machine learning algorithms and how to process data in multiple ways, gave an idea of the data structure and characteristics, and used five important algorithms in dealing with data to analyze and represent them graphically, showing the essential indicators among them in predicting diabetes in terms of whether a person is infected or not or is expected to be infected. After each algorithm shows its results, we will directly compare the results of the five algorithms with each other.

We frequently end up with several good models when working on a machine learning project. Each model will have its own set of performance characteristics. We may evaluate how accurate each model is on unseen data using resampling approaches such as cross-validation. We should examine the predicted accuracy of our machine learning algorithms in various ways before deciding on one or two to finish. We may do this by displaying the average accuracy, variance, and other aspects of the distribution of model accuracies using various visualization approaches.

We will compare models using repeated cross-validation with ten folds and three repetitions, a popular standard design. Accuracy and kappa are the assessment metrics. After training, the models are added to a list, and resamples run on the list of models. This function ensures that the models are similar and were trained using the same method of train control configuration. This object holds the evaluation metrics for each fold and the repeat of each method to be tested. Table 4.26 presents the output from the collected results for comparison between models using repeated cross-validation with ten folds and three repetitions. For example, in attempts 1 to 3 in fold 1, we repeat three times and note that RF, CART, and SVM models have the best accuracy, respectively. If we check extensively in the table, we note that this thing is repeated in the rest of Table 11.

Table 12, Figure 13, Figure 14, and Figure 15 summarize the results obtained from the previous five algorithms with a comparison function of machine learning algorithms. It shows that the accuracy sequence of the algorithms concerning the training data is utterly identical to the results of the total comparison because the latter depends mainly on the results from the training data RF, CART, SVM, LDA, and KNN. As for the test data results, they showed some differences in the accuracy sequence of the algorithms, where SVM, RF, CART, LDA, and KNN were the highest, respectively.

3. Conclusion

Based on the results obtained from the previous five algorithms with a comparison function for machine learning algorithms, it shows that the sequence of the accuracy of the algorithms concerning the training data is utterly identical to the results of the overall comparison because the latter mainly depends on the results of the RF, CART, SVM, LDA, and KNN training data. The test data results showed some differences in the accuracy sequence of the algorithms shown, with SVM, RF, CART, LDA, and KNN being the highest, respectively. The training data set refers to the samples used to build the model, while the test or validation data set is used to check performance.

Building a model that understands underlying data patterns is vital to providing long-lasting predictions with little retraining. At their most basic, machine learning models are statistical equations that run at high rates on several data points. As a result, statistical tests on the algorithms are essential for fine-tuning them and verifying whether the model's equation best fits the dataset at hand. We often generate multiple viable models when working on a machine learning project. Each model will have its own set of performance attributes. Using resampling techniques like cross-validation, we can determine how accurate each model is on unseen data. We must be able to use the estimations to choose one or two of the best models from our array of models.

This research will contribute to a scientific addition to past studies in this field of knowledge, which must conducted in various sectors. Focusing on authentic data in all areas, notably health, because it is directly related to human life, which is at the heart of all life on Earth. It is vital to emphasize the importance of data from its sources and urge governments to open data centers, particularly in countries such as Iraq. In this pilot project, we used data from scientific research in Baghdad, Iraq's capital, and five machine learning algorithms. In the future, we hope that data will be available in a variety of disciplines so that we may give service to future generations a better living chance.

After carefully considering the assessment above criteria, we have selected the most optimal machine learning methodology to predict diabetes mellitus in Iraq, with the objective of attaining superior performance. The techniques above are estimated with a supervised dataset for diabetes testing. The optimal choice is the technique that attains the highest level of performance in terms of accuracy and kappa. The findings indicate that the Support Vector Machine (SVM) and Random Forest (RF) algorithms exhibited higher accuracy in predicting diabetes.

The study's findings might help healthcare providers in Iraq avoid diabetes earlier and make better clinical decisions to control it, perhaps saving lives. Our future study will include considering and evaluating new features for further investigation.

Reference

1. International Diabetes Federation. Chapter 3. The global picture. In: *Diabetes Atlas*. 8th ed. Brussels, Belgium: International Diabetes Federation; 2017. https://idf.org/e-library/epidemiology-research/diabetes-atlas/134-idf-diabetes-atlas-8th-edition.html. Accessed March (2019).
2. World Health Organization. ; (2018). *Diabetes*. Geneva, Switzerland: World Health Organization https://www.who.int/news-room/fact-sheets/detail/diabetes. Updated October 30, 2018. Accessed March (2019).
3. Mansour AA, Al-Maliky AA, Kasem B, Jabar A, Mosbeh KA. (2014). Prevalence of diagnosed and undiagnosed diabetes mellitus in adults aged 19 years and older in Basrah, Iraq. Diabetes Metab Syndr Obes. ;7:139-144.

4. Khanam, j.j., Simon, Y.F. (2021). Comparison of machine learning algorithms for diabetes prediction, Science Direct, ICT Express: 7, 432–439.
5. Baran, M. (2020). Classification of multi-label data with machine learning methods (Master thesis). Sivas Cumhuriyet University, Sivas.
6. Alan, A. (2020). Evaluation of performance metrics and test techniques on various data sets in machine learning classification methods (Master thesis). Firat University, Fen Bilimleri Enstitusu, Elazig.
7. Parthiban, G., Srivatsa, S.K. (2012). Applying Machine Learning Methods in Diagnosing Heart Disease for Diabetic Patients. International Journal of Applied Information Systems (IJAIS). Foundation of Computer Science FCS, Volume 3– No.7, 25-30.
8. Keskin, A.K. (2018). Investigation of Machine Learning Classification Algorithms (Master thesis). SINOP UNIVERSITY, Fen Bilimleri institute, Sinop.
9. Nahzat, S., Yaganoglu, M. (2021). Diabetes Prediction Using Machine Learning Classification Algorithms. European Journal of Science and Technology, (24), 53-59.
10. Rebala, G., Ravi, A., Churiwala, S. (2019). *An Introduction to Machine Learning*. Springer, Gewerbestrasse 11, 6330 Cham, Switzerland. 9-11.
11. Ramasubramanian, K., Singh, A. (2019). Machine Learning Using R with Time Series and Industry-Based Use Cases in R. Second Edition. Apress Media, LLC California LLC, USA. 3.
12. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C.J. (1984). Classification and regression trees. First Edition. Chapman & Hall/CRC, NW, USA. 41.
13. Genuer, R., Poggi, J. (2020). USE R Random Forests with R. First Edition. Springer Nature Switzerland 6330 Cham, Switzerland. 10-12.
14. Rebala, G., Ravi, A., Churiwala, S. (2019). *An Introduction to Machine Learning*. Springer, Gewerbestrasse 11, 6330 Cham, Switzerland. 58-80.
15. Suthaharan, SH. (2016). Machine Learning Models and Algorithms for Big Data Classification. Volume 36. Springer Science & Business Media, New York 2016, USA. 7.
16. Brown, M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet,C. ve Haussler, D. (1999). *Support Vector Machine Classification of Microarray Gene Expression Data*, Special Work, Department of Computer Science & Biology University of California, Department of Engineering Mathematics, University of Bristol, Bristol, UK. 1-10.
17. Genuer, R., Poggi, J. (2020). USE R Random Forests with R. First Edition. Springer Nature Switzerland 6330 Cham, Switzerland. 43-107.
18. Rebala, G., Ravi, A., Churiwala, S. (2019). *An Introduction to Machine Learning*. Springer, Gewerbestrasse 11, 6330 Cham, Switzerland. 77-91.
19. Kuhn, M., Johnson K. (2016). *Applied Predictive Modeling*. Fifth Edition. Springer Science Business Media, New York. USA. 275-300.
20. Croux, C., Filzmoser, P., Joossens, K. (2008). Classification Efficiency for Robust Linear Discriminant Analysis, *Statistica Sinica*, 18 (1): 581-599.
21. James, G., Witten, D., Hastie, T., Tibshirani, (2021). *An Introduction to Statistical, Learning, with Applications in R*. Second Edition. Springer, NY 10004, USA. 132- 153.
22. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C.J. (1984). Classification and regression trees. First Edition. Chapman & Hall/CRC, NW, USA. 15-17.
23. Kramer, O. (2013). Dimensionality Reduction with Unsupervised Nearest Neighbors. Volume 51. Springer, USA. 14-15.
24. Rebala, G., Ravi, A., Churiwala, S. (2019). *An Introduction to Machine Learning*. Springer, Gewerbestrasse 11, 6330 Cham, Switzerland. 72-76.

Appendix A:

Table 1. The confusion matrix for training and test data

		Actual					
		Training data			Test data		
		N	P	Y	N	P	Y
Prediction	N	88	0	10	15	0	3
	P	0	35	0	0	12	0
	Y	0	4	655	0	2	176

Table 2. SVM and hyper parameter optimization SVM

		Actual					
		SVM			hyper parameter optimization		
		N	P	Y	N	P	Y
Prediction	N	95	6	11	103	0	0
	P	0	29	3	0	53	1
	Y	8	18	830	0	0	843

Table 3. Detailed performance results

	epsilon	cost	error	dispersion		epsilon	cost	error	dispersion
1	0	4	0.064	0.017764	45	0	64	0.044	0.018379
2	0.1	4	0.064	0.017764	46	0.1	64	0.044	0.018379
3	0.2	4	0.064	0.017764	47	0.2	64	0.044	0.018379
4	0.3	4	0.064	0.017764	48	0.3	64	0.044	0.018379
5	0.4	4	0.064	0.017764	49	0.4	64	0.044	0.018379
7	0.6	4	0.064	0.017764	51	0.6	64	0.044	0.018379
6	0.5	4	0.064	0.017764	50	0.5	64	0.044	0.018379
8	0.7	4	0.064	0.017764	52	0.7	64	0.044	0.018379
9	0.8	4	0.064	0.017764	53	0.8	64	0.044	0.018379
10	0.9	4	0.064	0.017764	54	0.9	64	0.044	0.018379
11	1	4	0.064	0.017764	55	1	64	0.044	0.018379
12	0	8	0.058	0.018135	56	0	128	0.043	0.014181
13	0.1	8	0.058	0.018135	57	0.1	128	0.043	0.014181
14	0.2	8	0.058	0.018135	58	0.2	128	0.043	0.014181
15	0.3	8	0.058	0.018135	59	0.3	128	0.043	0.014181
16	0.4	8	0.058	0.018135	60	0.4	128	0.043	0.014181
17	0.5	8	0.058	0.018135	61	0.5	128	0.043	0.014181
18	0.6	8	0.058	0.018135	62	0.6	128	0.043	0.014181
19	0.7	8	0.058	0.018135	63	0.7	128	0.043	0.014181
20	0.8	8	0.058	0.018135	64	0.8	128	0.043	0.014181
21	0.9	8	0.058	0.018135	65	0.9	128	0.043	0.014181
22	1	8	0.058	0.018135	66	1	128	0.043	0.014181
23	0	16	0.052	0.02044	67	0	256	0.042	0.013166
24	0.1	16	0.052	0.02044	68	0.1	256	0.042	0.013166
25	0.2	16	0.052	0.02044	69	0.2	256	0.042	0.013166
26	0.3	16	0.052	0.02044	70	0.3	256	0.042	0.013166
27	0.4	16	0.052	0.02044	71	0.4	256	0.042	0.013166
28	0.5	16	0.052	0.02044	72	0.5	256	0.042	0.013166
29	0.6	16	0.052	0.02044	73	0.6	256	0.042	0.013166

30	0.7	16	0.052	0.02044	74	0.7	256	0.042	0.013166
31	0.8	16	0.052	0.02044	75	0.8	256	0.042	0.013166
32	0.9	16	0.052	0.02044	76	0.9	256	0.042	0.013166
33	1	16	0.052	0.02044	77	1	256	0.042	0.013166
34	0	32	0.047	0.01567	78	0	512	0.043	0.014944
35	0.1	32	0.047	0.01567	79	0.1	512	0.043	0.014944
36	0.2	32	0.047	0.01567	80	0.2	512	0.043	0.014944
37	0.3	32	0.047	0.01567	81	0.3	512	0.043	0.014944
38	0.4	32	0.047	0.01567	82	0.4	512	0.043	0.014944
39	0.5	32	0.047	0.01567	83	0.5	512	0.043	0.014944
40	0.6	32	0.047	0.01567	84	0.6	512	0.043	0.014944
41	0.7	32	0.047	0.01567	85	0.7	512	0.043	0.014944
42	0.8	32	0.047	0.01567	86	0.8	512	0.043	0.014944
43	0.9	32	0.047	0.01567	87	0.9	512	0.043	0.014944
44	1	32	0.047	0.01567	88	1	512	0.043	0.014944

Table 4. Prediction with train and test data

		Actual					
		Train data			Test data		
		N	P	Y	N	P	Y
Prediction	N	85	0	0	18	0	1
	P	0	41	0	0	11	0
	Y	0	0	676	0	1	167

Table 5. Variable importance in random forest

	N	P	Y	Mean Decrease Accuracy	Mean Decrease Gini
Urea	3.906245	0.94913	3.658897	5.529994	1.856987
Cr	2.072084	4.594404	3.754678	5.62601	2.766652
HbA1c	136.7808	56.92705	20.41938	87.89449	93.43137
Chol	23.26642	6.2411	20.51264	30.00659	14.15726
TG	14.05844	5.361861	11.04533	18.3316	8.54632
HDL	-0.60844	1.452021	5.253631	4.729278	2.307927
LDL	-0.47808	4.044575	6.951772	6.510779	3.422808
VLDL	12.41205	5.001586	5.707	13.67666	7.401924
BMI	113.4069	19.21873	16.70639	56.30501	71.42615
AGE	3.775568	17.36448	15.94896	20.69322	15.88178

Table 6. The coefficients value for each class in the training data.

	LD1	LD2
Urea	0.03823	0.05407
Cr	-0.0003	-0.0038
HbA1c	0.25217	-0.2378
Chol	0.16504	-0.058
TG	-0.0099	-0.2394
HDL	-0.1428	0.46892
LDL	-0.0581	0.2392
VLDL	0.01206	0.06841
BMI	0.14684	-0.0043

AGE	0.04041	0.09947

Table 7. The actual classification and the predicted classification LDA

		Actual					
		Train data			Test data		
		N	P	Y	N	P	Y
Prediction	N	70	8	25	20	1	3
	P	3	13	15	4	3	2
	Y	6	24	635	0	4	164

Table 8. Accuracy for different values for k

k	Accuracy	Kappa	k	Accuracy	Kappa
5	0.877946	0.55643	25	0.871264	0.478
7	0.878763	0.551541	27	0.867888	0.459466
9	0.86909	0.50088	29	0.8658	0.449493
11	0.86698	0.480685	31	0.867477	0.447494
13	0.860303	0.458736	33	0.867877	0.451045
15	0.863219	0.470735	35	0.864138	0.430092
17	0.86409	0.46503	37	0.867471	0.436519
19	0.866216	0.4677	39	0.869127	0.434543
21	0.867044	0.462562	41	0.869122	0.428844
23	0.870014	0.471428	43	0.87207	0.424599

Table 9. Variable importance

	N	P	Y
HbA1c	100	100	94.743
BMI	92.992	82.8	92.992
AGE	70.392	82.736	82.736
TG	48.907	47.647	48.907
VLDL	41.161	26.893	41.161
Chol.	28.226	24.212	28.226
Urea	10.113	12.353	12.353
Cr	10.717	10.717	9.613
HDL	6.037	3.488	6.037
LDL	2.464	2.464	0

Table 10. Confusion matrix and statistics- testing data

		Actual					
		Train data			Test data		
		N	P	Y	N	P	Y
Prediction	N	59	9	18	11	6	4
	P	7	19	7	1	6	5
	Y	22	11	640	3	2	170

Table 11. Evaluation metrics for each fold

	CART	LDA	SVM	KNN	RF	Resample
1	0.949495	0.871287	0.919192	0.868687	0.979798	Fold01.Rep1
2	0.98	0.89	0.96	0.88	1	Fold01.Rep2
3	0.97	0.85	0.91	0.88	0.99	Fold01.Rep3
4	0.950495	0.888889	0.930693	0.881188	0.990099	Fold02.Rep1
5	0.959596	0.909091	0.949495	0.878788	0.989899	Fold02.Rep2

6	0.950495	0.89899	0.930693	0.920792	0.970297	Fold02.Rep3
7	0.980392	0.910891	0.941176	0.872549	1	Fold03.Rep1
8	0.960396	0.841584	0.910891	0.851485	0.980198	Fold03.Rep2
9	0.989899	0.88	0.939394	0.89899	0.989899	Fold03.Rep3
10	0.949495	0.891089	0.949495	0.888889	0.989899	Fold04.Rep1
11	0.96	0.881188	0.89	0.84	0.98	Fold04.Rep2
12	0.95	0.95	0.93	0.92	0.97	Fold04.Rep3
13	0.960396	0.861386	0.910891	0.851485	1	Fold05.Rep1
14	0.979798	0.89899	0.969697	0.909091	1	Fold05.Rep2
15	0.96	0.91	0.93	0.89	1	Fold05.Rep3
16	0.930693	0.888889	0.910891	0.891089	0.960396	Fold06.Rep1
17	0.96	0.92	0.92	0.93	0.99	Fold06.Rep2
18	0.93	0.929293	0.89	0.84	1	Fold06.Rep3
19	0.959596	0.919192	0.949495	0.888889	0.989899	Fold07.Rep1
20	0.960396	0.891089	0.960396	0.920792	0.990099	Fold07.Rep2
21	0.989899	0.910891	0.949495	0.868687	0.989899	Fold07.Rep3
22	0.98	0.920792	0.96	0.92	0.99	Fold08.Rep1
23	0.95	0.929293	0.92	0.88	0.98	Fold08.Rep2
24	0.940594	0.88	0.930693	0.881188	0.980198	Fold08.Rep3
25	0.969697	0.868687	0.909091	0.89899	0.989899	Fold09.Rep1
26	0.94	0.89	0.89	0.87	1	Fold09.Rep2
27	0.96	0.878788	0.9	0.88	0.99	Fold09.Rep3
28	0.969697	0.89899	0.959596	0.868687	1	Fold10.Rep1
29	0.96	0.88	0.95	0.86	0.98	Fold10.Rep2
30	0.96	0.843137	0.94	0.88	1	Fold10.Rep3

Table 12. Summaries results algorithms

	train data			test data		
algorithm	accuracy	kappa	best	accuracy	kappa	best
CART	0.9823	0.9381	2	0.976	0.905	3
LDA	0.898	0.64	4	0.93	0.75	4
SVM	0.954	0.825	3	0.999	0.996	1
KNN	0.899	0.6187	5	0.899	0.6187	5
RF	1	1	1	0.9899	0.9623	2

Appendix B:

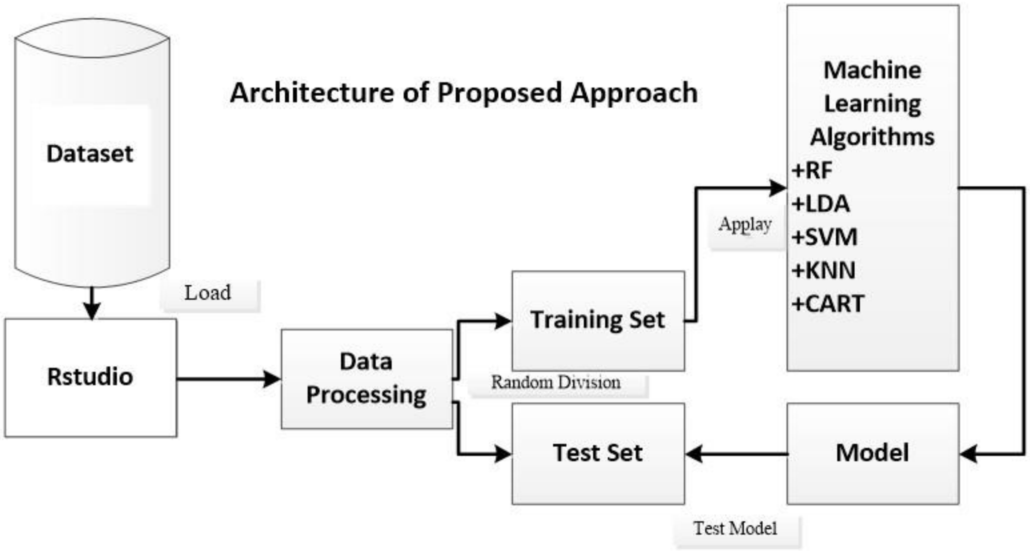


Figure 1. The architecture of the Proposed Approach

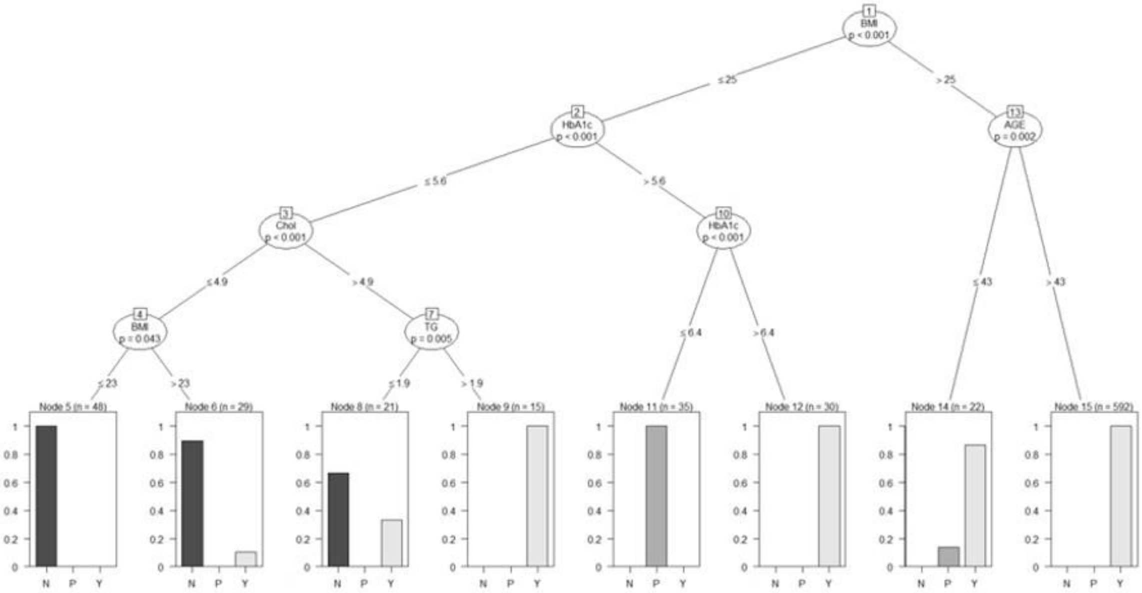


Figure 2. CART for diabetes.

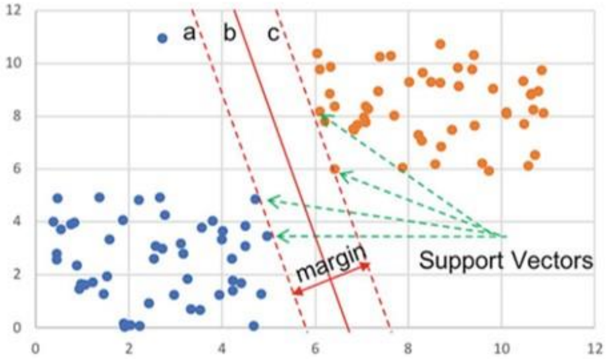


Figure 3 Support vectors and margin representation.

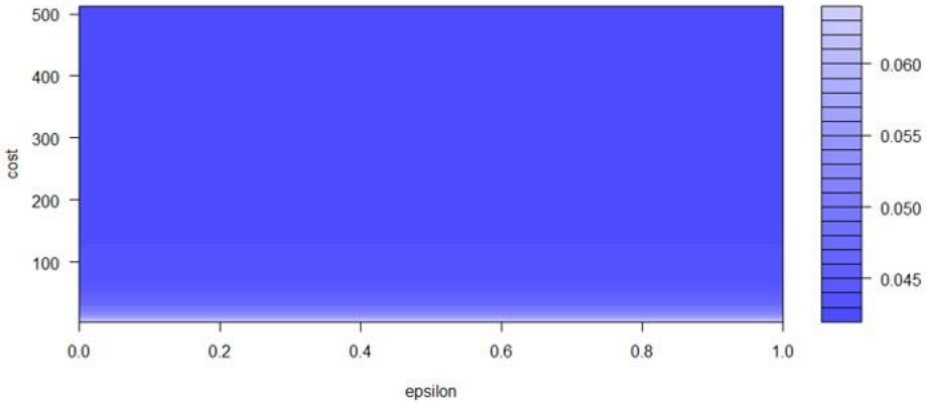


Figure 4. Performance for SVM

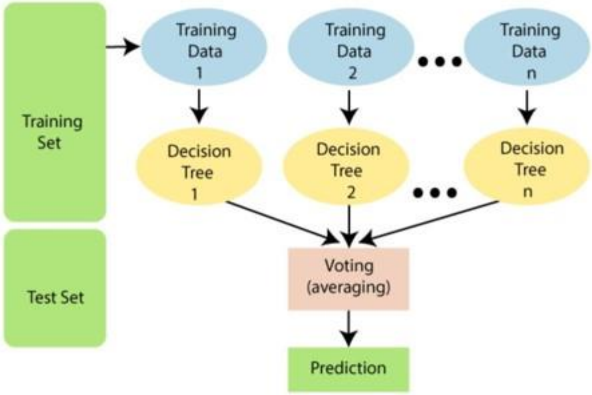


Figure 5. Explains the working of the random forest algorithm.

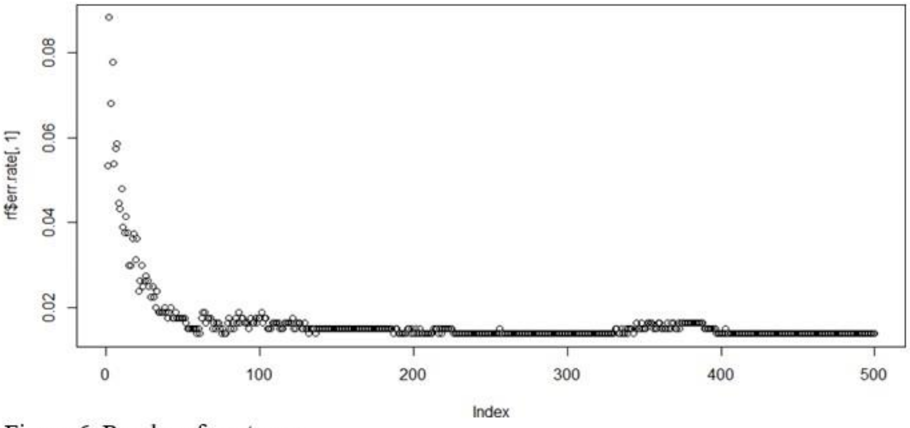


Figure 6. Random forest error.

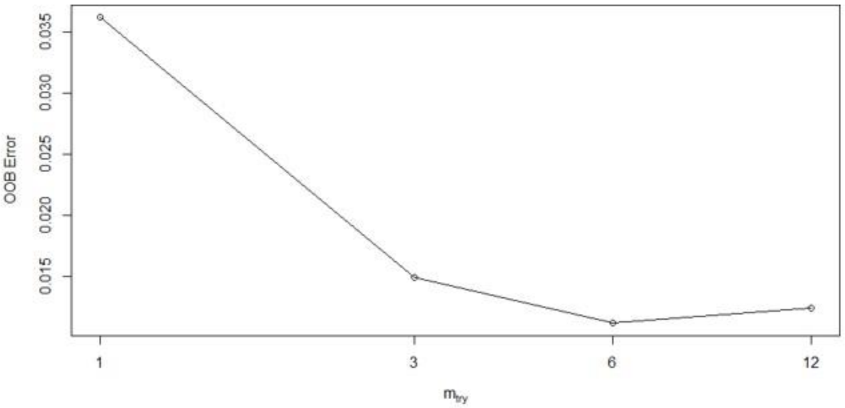


Figure 7. Least error for OOB and optimize parameter tuneRF function.

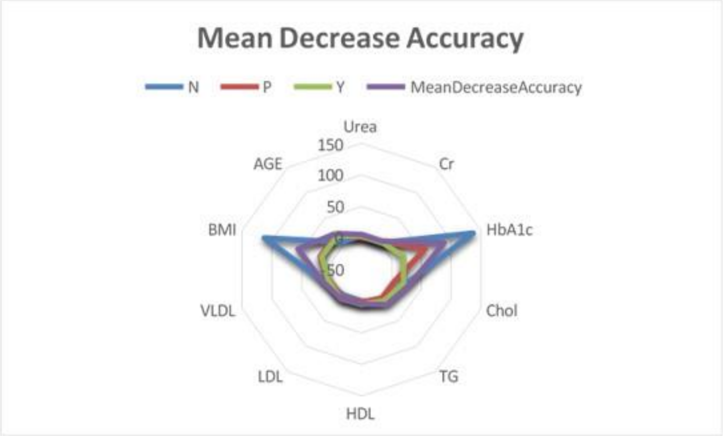


Figure 8. Mean decrease accuracy.

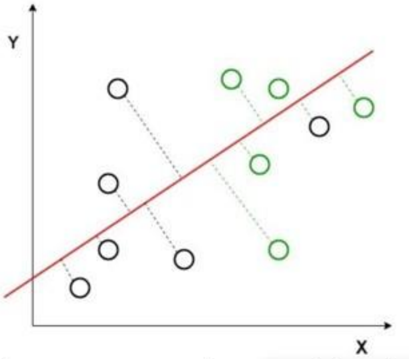


Figure 9. Constructed a new axis in red.

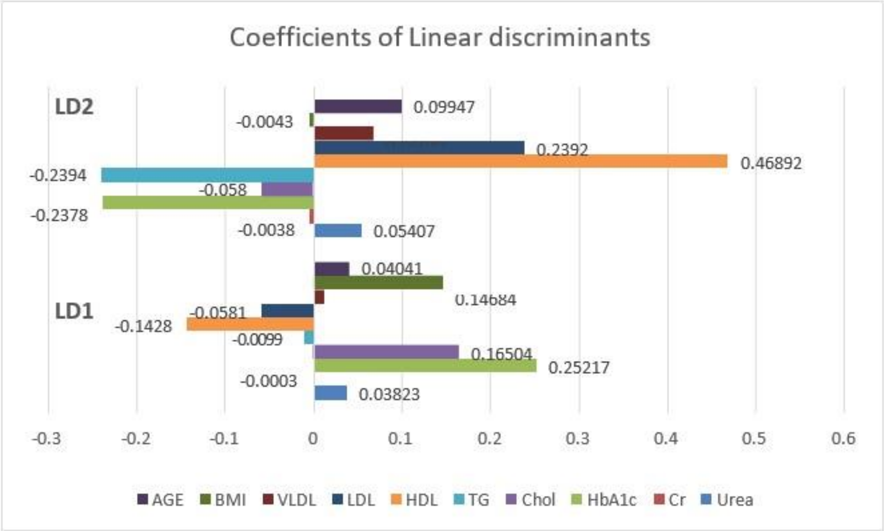


Figure 10. Coefficients of linear discriminants.

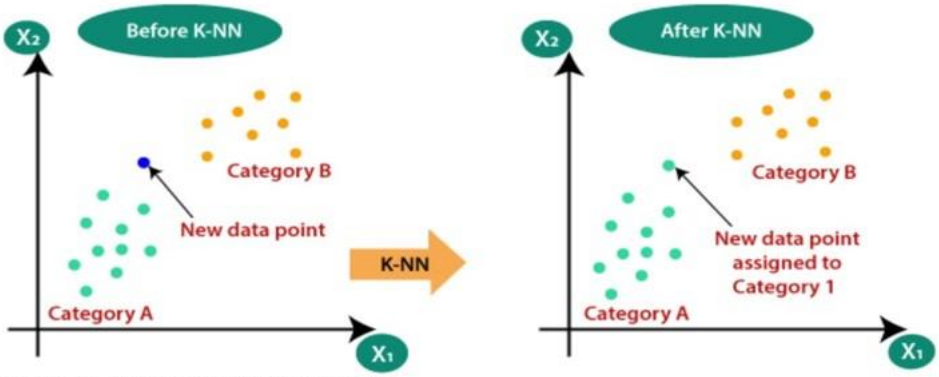


Figure 11. Class category using K-NN.

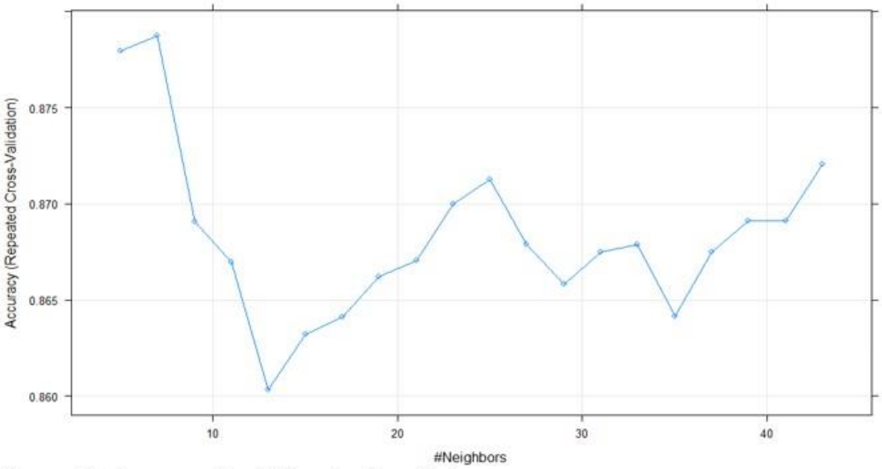
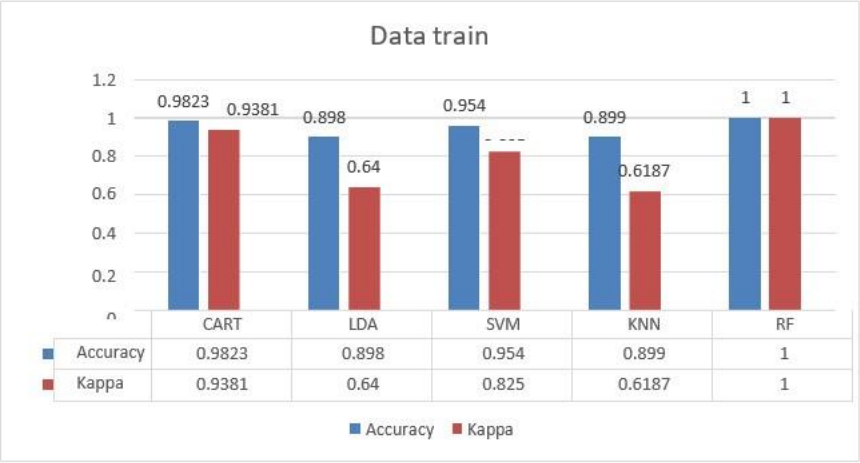
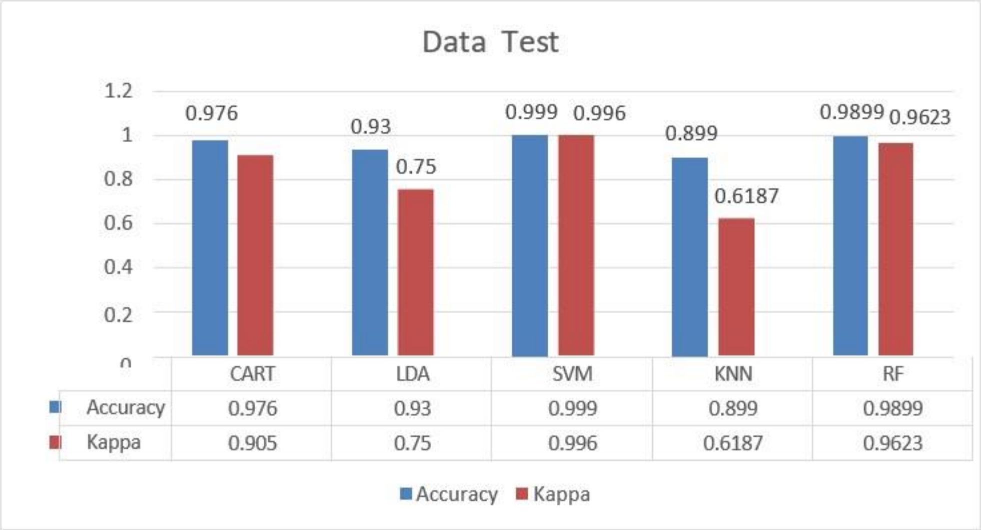


Figure 12. Accuracy for different values for k.



	CART	LDA	SVM	KNN	RF
🟦 Accuracy	0.9823	0.898	0.954	0.899	1
🟥 Kappa	0.9381	0.64	0.825	0.6187	1

Figure 13. Compare accuracy algorithms data train.



	CART	LDA	SVM	KNN	RF
🟦 Accuracy	0.976	0.93	0.999	0.899	0.9899
🟥 Kappa	0.905	0.75	0.996	0.6187	0.9623

Figure 14. Compare accuracy algorithms data validate.

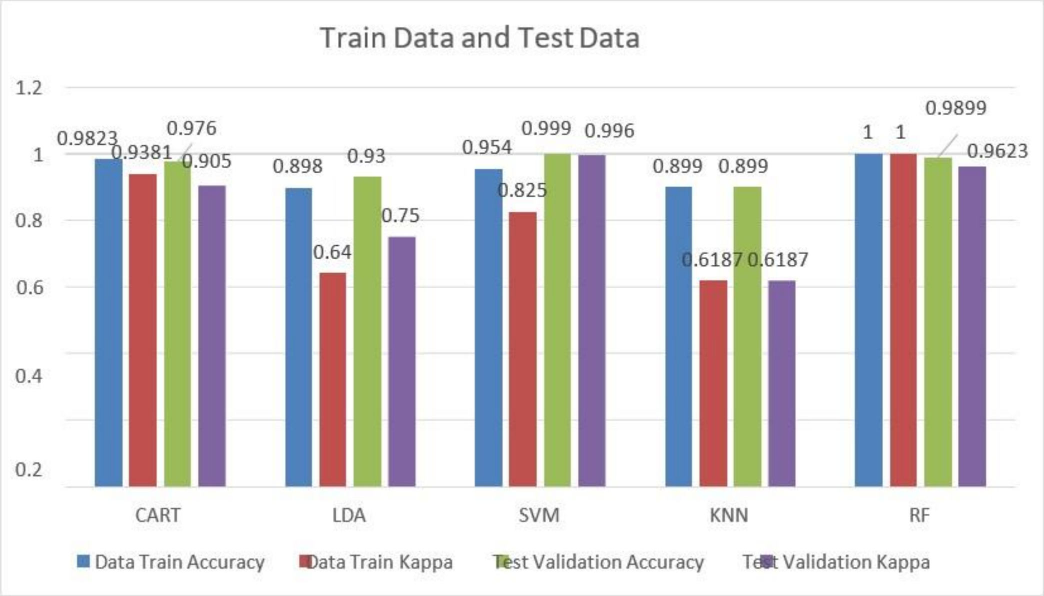


Figure 15. Compare accuracy algorithms data train and validation.

تصنيف مجموعة بيانات مرض السكري عبر تقنيات التعلم الآلي المختلفة

دلشاد محمد سعيد الطلباني [1] و فيفزي اردوغان[2]

[1] قسم علوم الحاسوب ، مديرية تكنولوجيا المعلومات والإحصاء ، جامعة السليمانية بوليتكنيك ، جامعة قرميان ، كالار ، العراق.

dilshad.saeed20@gmail.com ، [2]قسم الإحصاء ، جامعة فان يوزونكو ييل ، فان ، تركيا. fevzier@gmail.com

الخلاصة: أصبح مرض السكري من أكثر الأمراض انتشارا في العراق وهو مدرج كأحد الأسباب الرئيسية للوفاة. يوفر التعلم الآلي نتائج فعالة لاستخراج المعلومات من خلال إنشاء نماذج تنبؤية من مجموعات البيانات الطبية التشخيصية التي تم جمعها من مرضى السكري في العراق.

في هذه الدراسة ، طبقنا تصنيف التعلم الآلي لمقارنة ومقارنة أداء أشجار التصنيف والانحدار (العربة) ، وآلات ناقلات الدعم (سفم) ، والغابات العشوائية (رف) ، وتحليل التمييز الخطي (لدا) ، وأقرب الجيران (كن). سعينا إلى تصميم نموذج يمكن أن يتنبأ بأقصى قدر من الدقة باحتمالية إصابة الشخص بمرض السكري أو أنه يتمتع بصحة جيدة أو من المتوقع أن يصاب به في المستقبل باستخدام مقياسي الدقة وكابا.

واستنادا إلى النتائج التي تم الحصول عليها من الخوارزميات ، أظهرت أن دقة وتسلسل الخوارزميات المتعلقة ببيانات التدريب كانت الغابات العشوائية (رف) ، وتصنيف وأشجار الانحدار (العربة) ، ودعم آلة ناقلات (سفم) ، وتحليل التمييز الخطي (لدا) ، و ك-أقرب الجيران (كن). في حين أظهرت نتائج بيانات الاختبار بعض الاختلافات ، وكان تسلسل الخوارزميات على النحو التالي: سفم ، رف ، عربة ، لدا ، و كن كانت أعلى ، على التوالي. تشير مجموعة بيانات التدريب إلى العينات التي تم استخدامها لبناء النموذج ، بينما تستخدم مجموعة بيانات الاختبار لتقييم أداء النموذج.

بناء على معايير التقييم التي نوقشت أعلاه ، اخترنا أفضل نهج للتعلم الآلي للتنبؤ بمرض السكري في العراق لتحقيق أداء عال. يتم تقريب جميع الاستراتيجيات المذكورة أعلاه باستخدام مجموعة بيانات اختبار مرض السكري الخاضعة للإشراف. يعتبر النهج الذي يحقق أقصى قدر من الأداء من حيث الدقة وكابا الخيار الأفضل. بناء على النتائج ، يمكن ملاحظة أن خوارزميات سفم و رف تنبأت بمرض السكري بمزيد من الدقة.

الكلمات المفتاحية: التعلم الآلي ، التصنيف ، مرض السكري.