Developing a Predictive Health Care System for Diabetes Diagnosis as a Machine Learning-Based Web Service

Rasha Ali Abdoul Raheem[1] and Ali Kadhum M. Al-Qurabat[2*]
1College of Science for Women, University of Babylon, rasha.talak.gsci29@student.uobabylon.edu.iq, Babylon, Iraq.
2College of Science for Women, University of Babylon, alik.m.alqurabat@uobabylon.edu.iq, Babylon, Iraq.
***Corresponding author email:** alik.m.alqurabat@uobabylon.edu.iq; mobile: 07802627087

تطوير نظام رعاية صحية تنبؤي لتشخيص مرض السكري كخدمة ويب قائمة على التعلم الآلي

رشا علي عبد الرحيم[١] ، علي كاظم محمد الغرابي[2*]
1 كلية العلوم للبنات، جامعة بابل، rasha.talak.gsci29@student.uobabylon.edu.iq، بابل، العراق
2 كلية العلوم للبنات، جامعة بابل، alik.m.alqurabat@uobabylon.edu.iq، بابل، العراق

Received:	31/12/2021	Accepted:	16/1/2022	Published:	1/3/2022

ABSTRACT
Background:
Diabetes is one of the dangerous and silent illnesses that cause sudden death. It can occur at any time and may cause great injury to the organs of the body or damage them completely. So, we must investigate this disease at the beginning of its appearance and before it gets hard to treat. With the fast advancement of Machine Learning (ML), these approaches enhanced the efficiency of decision processes in a wide range of applications, including medical diagnostics.
Materials and Methods:
In this paper, we chose a medical application field and used supervised machine learning algorithms to construct a high-accuracy prediction model for diabetes in humans at an early stage, before it progresses to the point of morbidity or fatality. The suggested model can extract hidden knowledge from diabetes-related data gathered from the Kaggle machine learning repository. We utilize Microsoft Azure ML Studio to model these ML algorithms.
Results:
This study will benefit the health industry by offering users an online tool (i.e., web service) that allows them to input data and receive results that predict whether or not the person has diabetes. As a result, prior knowledge and ongoing monitoring of their diabetic health state will lower the risk of complications, morbidity, and death caused by this illness. After running numerous experiments with the classifier models to evaluate the proposed system, several performance indicators, including Recall, Precision, Accuracy, and f1-score, are measured for comparison. Based on the classification output, it was determined that Decision Forest is a better strategy and produces better results than the other ML approaches.
Conclusion:
The suggested system's key contribution is to improve healthcare quality, minimize hospitalizations, and lower the high expenditures of healthcare and drugs.

Key words:
Diabetes, Classification, Machine Learning, Microsoft Azure ML Studio, Web service.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

الخلاصة

مقدمة:

مرض السكري من الأمراض الخطيرة والصامتة التي تسبب الموت المفاجئ. يمكن أن تحدث في أي وقت وقد تسبب إصابة كبيرة بأعضاء الجسم أو تتلفها بالكامل. لذلك يجب التحقيق في هذا المرض في بداية ظهوره وقبل أن يصعب علاجه. مع التقدم السريع في التعلم الآلي (ML) ، عززت هذه الأساليب كفاءة عمليات اتخاذ القرار في مجموعة واسعة من التطبيقات ، بما في ذلك التشخيص الطبي.

طرق العمل:

في هذا البحث ، اخترنا مجال تطبيق طبي واستخدمنا خوارزميات التعلم الآلي الخاضعة للإشراف لبناء نموذج تنبؤ عالي الدقة لمرض السكري لدى البشر في مرحلة مبكرة ، قبل أن يتطور إلى درجة المرض أو الوفاة. يمكن للنموذج المقترح استخلاص المعرفة المخفية من البيانات المتعلقة بمرض السكري التي تم جمعها من مستودع التعلم الآلي Kaggle. استخدمنا Microsoft Azure ML Studio لنمذجة خوارزميات ML.

الاستنتاجات:

تتمثل المساهمة الرئيسية للنظام المقترح في تحسين جودة الرعاية الصحية ، وتقليل الاستشفاء ، وخفض النفقات المرتفعة للرعاية الصحية والأدوية.

الكلمات المفتاحية:

مرض السكري ، التصنيف ، التعلم الآلي ، Microsoft Azure ML Studio ، خدمة الويب.

INTRODUCTION

Diabetes illness is a long-term illness characterized by high blood sugar. It can cause numerous complications. Thus, according to the increasing sickness during these years, in 2040, the world's diabetic patients will number 642 million, implying that one out of every ten people will be affected by diabetes [1]. Diabetes affects human function by causing major damage to the eyes, heart, kidneys, and nerves. Diabetes is a long-term illness causes weakness in the body besides it one of the 10 causes of death worldwide [2]. Insulin is a hormone that regulates glucose absorption from the blood into most cells. (Fundamentally Cells of muscle and fat, yet not central nervous system cells). In this way in all types of diabetes mellitus, a deficiency of insulin or the cold-heartedness of its receptors plays a key role [3]. However, risk factors such as heredity and lifestyle can play a role. Diabetes must be diagnosed early in order to live a healthy life [4]. And decrease the high costs of patient readmission [5]. Although diabetes is not curable in general, it can be managed with the right treatment [6]. Diabetes impacts organs such as the heart, kidney, nerves, eye, foot, and so on, making it hard for medical practitioners to detect early because of the intricate reliance on numerous elements.

Through the topic of prediction, several researches have been carried out for several diseases recently, to the level that some of today's clinicians now make use of machine learning models to predict different diseases. It is, therefore, imperative to design a diabetes classifier that is convenient, accurate and cost efficient. Artificial Intelligence techniques provide a wide range of ideas that are useful to human related fields of application like, medical diagnosis is a procedure in which a physician must consider a variety of elements before diagnosing diabetes, making the physician's job tough and time-consuming? Approaches to machine learning and data mining have a variety of health issues [7].

In recent times, many methods and algorithms have been discovered which can be used to find hidden information in biological datasets, such as supervised learning techniques like Neural networks (NNs), Averaged Perceptron (AP), Decision Forest (DF), Ambiguous Logic Systems, Boosted Decision Trees (BDT), Naïve Bayes, and logistic regression; unsupervised learning techniques like clustering analysis, pattern recognition and image analysis; and reinforcement

ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

algorithms which is applied in the field of game theory, control theory and decision theory. This project intends to develop a prediction model with high degree of accuracy for diabetes in people at an early stage, before it becomes escalated to a point of morbidity or mortality using some supervised learning algorithms. This project will also contribute to the health sector by providing people with accurate prior knowledge about their health status as related to diabetes hence, reducing the rate of complications, morbidity and mortality being caused by this disease.

The main contributions of this research are as follows:

1- Replace current diabetes diagnosis methods with modern technologies that save time. We intend to employ various machine learning algorithms, such as the Support Vector Machine and the Decision Tree, among others.
2- The research contributes to improving the body's health through early detection of diabetes, as well as saving the patient's life and prolonging his life away from the risk of disease development.
3- Contributes to predicting the diagnosis of diabetes without going to the doctor and losing money and saving time and effort.
4- Using Microsoft Azure Machine-Learning Studio, the proposed machine-learning algorithms are modeled and tested.
5- Comparing the performances of many algorithms and, in the end, using the most efficient model.
6- Deploying the proposed ML diabetic prediction model as a web service.

The rest of this paper is as follows: Section 2 describes the related works of predicting diabetics by machine learning. Sections 3 and 4 present the proposed system and its implementation and results. Conclusions and future work are presented in Section 5.

Introduction sample of introduction sample of introduction sample of introduction sample of introduction sample of introduction sample of introduction sample of introduction sample of introduction [1].

Related Works

As noted in the paragraph, this part will cover earlier works on the topic of diabetes mellitus prediction and diagnosis using machine learning.

Quan et al. (2018), To predict diabetes mellitus, researchers used Boosted decision. trees, random. forests, and neural. networks. A model was tested using five-fold cross validation in this study. They chose several methods with superior performance to conduct. independent test experiments in order to verify the methods' universal applicability. The results indicated that random forest prediction had the highest accuracy when all of the criteria were used. (ACC = 0.8084) [1]. Hasan et al. (2018), suggested a framework used for real-time diabetes expectation, application and monitoring (DPMA). The classification approaches for diabetes were reviewed in

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com | ISSN: 2312-8135 | Print ISSN: 1992-0652 info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

the study. The goal of this study is to create an optimal and efficient machine learning (ML) application that can accurately recognize and forecast diabetic symptoms. Five of the most important machine learning classification approaches for diabetes prediction were investigated in this study [8].

Aminul et al. (2017), investigated and compared various types of ML classification algorithms. The aim of this study is to use the results of machine learning. classification algorithms to detect the start of diabetes in diabetic patients [9]. Harleen Kaur (2018), Using the R data manipulation tool, hey created five distinct forecasting models and analyzed them. They employed supervised machine learning algorithms such as the linear kernel support vector machine (SVM-linear). Kernel support vector machine, Radial Basis Function (RBF), k-nearest neighbor (k-NN), Artificial Neural Network (ANN), and Multifactor Dimensionality Reduction (MDR). In the" Pima Indian diabetes dataset", the author used machine learning to create trends and find patterns with risk factors, as well as classify patients as diabetic or non-diabetic [4].

Jimmy et al. (2019), the intent of their study is to collect and use data of diabetic patients who have either been readmitted or not readmitted and feed the data into four ML algorithms to determine which algorithm performs the best. Comparisons about true/false positives, training scale and accuracy are performed on K-nearest. Neighbor, Boosted decision. tree, Logistic. regression, and neural. networks [5]. Ramana et al. (2018), proposed model has ability to extract the hidden information from a huge amount of diabetes-related data gleaned through Web services archive of data The assessment experiment consumes insulin and measures from dynamic, and it gives real-time blood glucose levels that are predicted based on numerous life events. Boosted tree, for example, is a circumstance. regularization, and algorithm. The prediction tool correctly predicts the proportion of blood glucose readings that exhibit diabetes positives to 90 percent accuracy. Otherwise, a percentage of patients with NO was determined using the true negative rate, which calculates the fraction of false negatives [3].

Clodagh (2019), resented predictive modeling. as a method for detecting patterns in historical data. It learns from these patterns so that it can generate predictions automatically when new data becomes available. The best-performing model is chosen to construct a web service that allows users to submit data and obtain scored results, thereby them to the data [10].

Materials and Methods

- **The Proposed System**

In this section, a Machine Learning model is proposed to illustrate the flow of research. To predict the outcomes for diabetic patients, several steps have to be taken to construct a good model for decision making. Figure 1 displays seven chief steps along with some sub-step to carry out to build the final model. The first step of this research was Data collection, which required to find and collect data set related to the problem. This project used an available diabetes dataset from

Machine Learning repository Kaggle named "Pima Indian diabetes". The following step is the second, the data was pre-processed since it can include many gaps, lost values, and outliers. The third step is, the dataset was analyzed in terms of description and split into two groups as per the model required. One for training the model and other for testing the model. In the fourth stage, different Machine Learning algorithms were executed on the training data set as per different parametric presentation. In the fifth stage, the execution was done in two coding environments for fine contrasts. In the sixth stage, the results were interpreted and analyzed for different algorithms in different programming environments in the seventh step, and in the eighth and final step, the model was ready to differentiate the conclusions for both the environments.

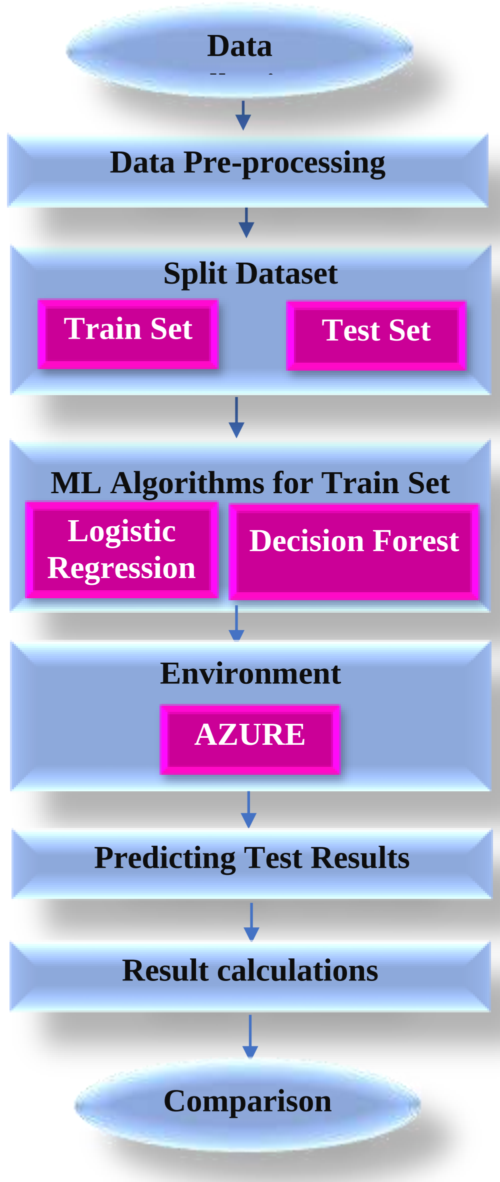


Figure 1. Proposed Model.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل

- **Selection of Datasets**

In today's world, there is a huge amount of data available through different resources like the Internet, surveys, and experiments, etc. There are different repositories available for Data Science such as UCI, Data World, and Kaggle, etc. which are well-known to provide datasets suitable for Machine Learning Algorithms in almost all areas. The most suitable and well-known dataset available on all three repositories UCI, Data World, and Kaggle for Machine Learning purposes was the "Pima Indian Diabetes (PID)" dataset. This dataset is broadly used to experimentally understand the data by the Machine Learning community from newbies to professionals. This dataset is standardly published in various research papers and studies for various Machine learning Algorithms.

The PID data set has been created by collecting the information among the Pima Indian female population near Phoenix, Arizona. The PID repository includes 768 Indian female patients. The conditional response factor takes two classes ' 0 ' or ' 1 ' values, where ' 1 ' is a positive diabetes test and ' 0 ' is a negative diabetes result. Class ' 1 ' has 268 (37.8%) cases and Class ' 0 ' has 500 (62.2%) cases.

In this data set eight medical features are evaluated to pick the occurrence of diabetes as set '1' or class '0' with no missing or outright liars reported. These medical conditions are linked to a higher risk of diabetes. Tables 1 and 2 list the eight factors or features, together with their data types and statistical analyses. Figure 2 illustrates the attributes of the dataset.

Table 1. Attributes of PID Dataset.

No.	List of Attributes	Range (Min – Max)	Description
1.	No. of Times Pregnant	0 – 17	Pregnancy frequency
2.	Serum Insulin	0 – 846	2 Hours Serum Insulin values obtained
3.	Tri-Skin-Fold thickness	0 – 99	Thickness Fold of Triceps Skin
4.	Diastolic BP	0 – 122	Diastolic Blood Pressure
5.	Plasma Glucose	0 – 199	Tolerance values of Oral Glucose Test
6.	AGE	21 – 81	Age
7.	Diabetes pedigree	0.078 – 2.42	Functions of Diabetes Pedigree
8.	BMI	0 – 67.1	Body Mass Index
9.	Outcome (class variables)	0 – 1	Positive and Negative

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل

Table 2. Statistical analysis of the Pima Indian dataset.

	Number of times Pregnancies	plasma Glucose concentration	diastolic Blood Pressure	Triceps Skin fold Thickness	2-hour serum Insulin	Body mass index	Diabetes Pedigree Function	Age	Outcomes
Count	768.00	768.00	768.00	768.00	768.00	768.00	768.00	768.00	768.00
Mean	3.84	120.89	69.10	20.53	79.79	31.99	0.471	33.24	0.348
Standard deviation	3.36	31.97	19.35	15.95	115.24	7.88	0.33	11.76	0.47

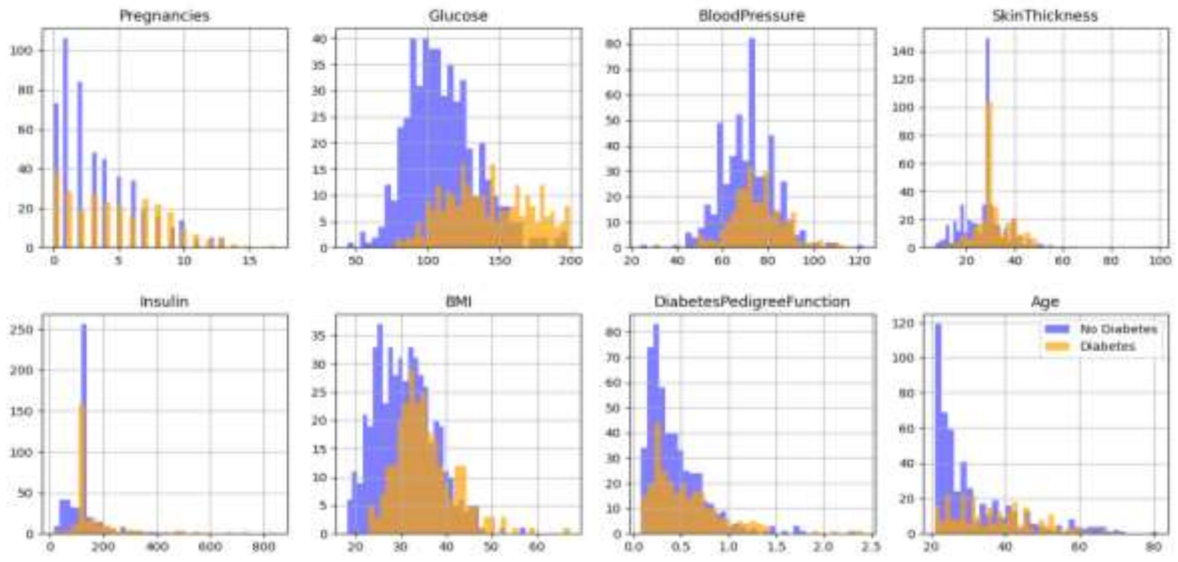


Figure 2. Illustrates the medical attributes of dataset.

- **Data Pre-processing**

Information pre-processing has an important role when a model is used to understand the data in the research study. Because most of the data to be analyzed contains outliers, null values, and special characters. If data comprises poor-quality input, regardless matter how effectively it is produced, the resulting output will always be poor. Hence, Data pre-processing is a significant stage in the data analysis and machine learning process.

Preprocessing greatly reduces the size of the data. Data cleaning is based on three main points

i. **Relevancy_** Each column (attributes) is linked to one another and to the result column
ii. **Connectedness_** Here are no or only a few missing data values.
iii. **Accuracy_** The data values in the uppermost rows are precise (also selectively accurate).

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

This dataset had many missing values, which were represented by zeros "0." Exploratory Data Analysis (EDA) was performed using the Pandas library from Python, and it was discovered from the histograms that there are many rows containing zero, which is impossible. E.g., Consider Blood Pressure; in some rows, it is 0 (impossible), and even the minimum Blood Pressure can never be zero, as shown in Figure 3.

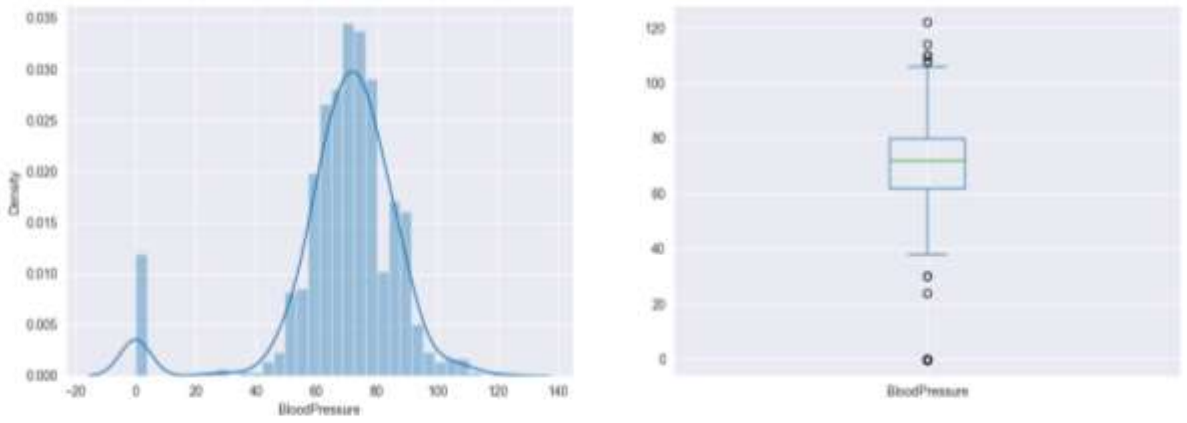


Figure 3. The Histogram for Blood Pressure.

- **Dataset Splitting**

In Machine Learning, splitting a dataset is an important step. Machine learning (in general) necessitates dividing dataset into two (non-equal) sets:

- Using training data, an ML algorithm creates the first model.
- Testing data has been utilized to estimate the produced model by assisting in the validation of its accuracy (Values in the test dataset versus expected values).

Machine Learning's primary goal is to circulating the model. The data it is trained on in other places. It is necessary to employ data that is not used in the training data in order to assess the model's ability. In addition, Because the values of future occurrences are unclear, it is difficult to assess their reliability. As a result, we must use part of the data from our Test Set, which has previously been used as a reference for future data.

The most appropriate technique to partition huge data into training and test subsets, usually with a 70-80% training percentage and a 20-30% test set percentage.

- **Training Set:** The train set includes almost 576 patient records, or 75 percent of the PID data set. Both the independence (X train matrix) and reliant (Y train matrix) variables are specified in the training set.
- **Test Set:** Remaining 25% of data in the PID data set is in a set of data, that also includes about 192 people’s records. Where X test matrix is given "a dependent variable" and Y test matrix is defined as "an independent variable" by the test set.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

- ML Algorithm

We must choose a certain machine learning algorithm that uses a training dataset and processes it iteratively to generate the model in order to train the model. The goal is to keep losses to a minimum. After importing and preprocessing data, the most crucial stage in any machine learning workflow is picking the algorithm to utilize. The type of problem to be solved and the type of data to be handled with must both be considered while selecting an algorithm.

A common question is "*Which machine learning algorithm should I use?*" The algorithm selection depends primarily on two different aspects of data science scenario:

- How will you use your relevant data? What particular business question do you hope to answer using your historical data?
- In your data science situation, what are the applications? What are the accuracy, training duration, linearity, grist of parameters, and grist of attributes specifics of your solvent?

This paper is concerned with classification using Decision Forest, Support Vector Machine, Average Perceptron, Boosted-Decision-Tree, Logistic Regression, and Neural Network algorithms as shown in Figure 4.

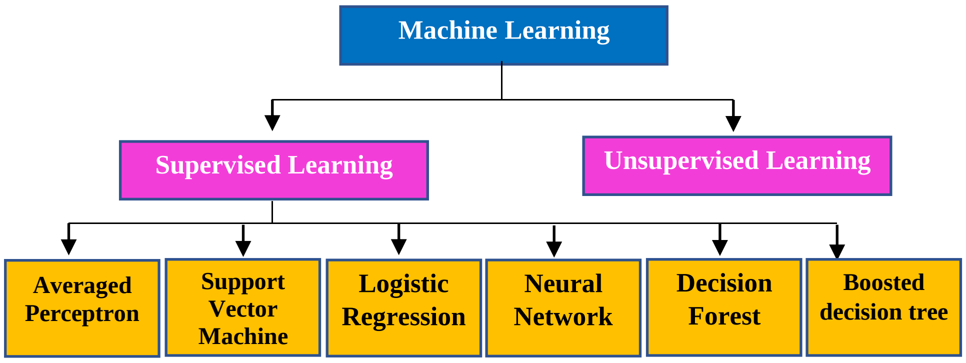


Figure 4. The ML used algorithms

❖ Logistic Regression (LR)

Logistic. regression is a kind of probabilistic statistical classification model used to analyze a dataset with one or more independent elements that influence the outcome [9]. Because our dataset only contains data coded as '1' (TRUE, success, infected, etc.) or '0' (failure, etc.), LR is a successful algorithm to employ with it (FALSE, failure, uninfected, etc.) [9].

❖ Artificial Neural Network (ANN)

Because of the vast amount of input data and only two possible outputs the ANN architecture is ideal for the dataset used in this study. The dataset already has the answer to the

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com | ISSN: 2312-8135 | Print ISSN: 1992-0652

diabetic question; therefore, the supervised training model will be used. The input layer, hidden layer, and output layer are all significant aspects of a neural network. as shown in Figure 5. The input layer is in charge of receiving. input data. The results can be obtained from the output layer. The hidden layer is the layer that sits between the input and output layers. Because they can't be seen from the outside. On the same layer, there is no connectivity between neurons [1, 5].

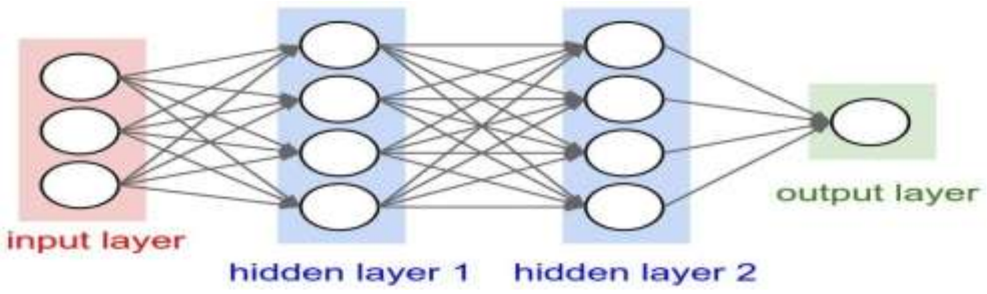


Figure 5. Shows the neural network algorithm.

❖ **Boosted Decision Tree (DT)**

Using tree structure techniques, a decision tree divides data into discrete categories. The primary goal of decision trees is to reveal the structural information in data. During the machine learning process, the decision tree method is a supervised machine learning methodology that generates a decision tree from a set of class labeled training samples. as shown in Figure 6. The training samples and their associated class labels are used to start the decision tree algorithm. This training set is recursively partitioned into subsets depending on feature value, with each subset's data being purer than the parent set's data. In a decision tree, each internal node represents an attribute (feature) test, each branch represents a test result, and each leaf node represents the class label. When a classifier. decision tree is used to determine the class label of an unknown sample, a path is traced from the root to the leaf node that contains the sample's class label [11].

Decision tree is indeed a machine learning predictive model that connects object notes with conclusions about the item's goal value. In decision trees, post-pruning techniques are often used. Following the pruning of decision trees with a validation set, classifiers are employed to evaluate their performance. Each node could be erased, and it can be allocated to the most common type of training set [12].

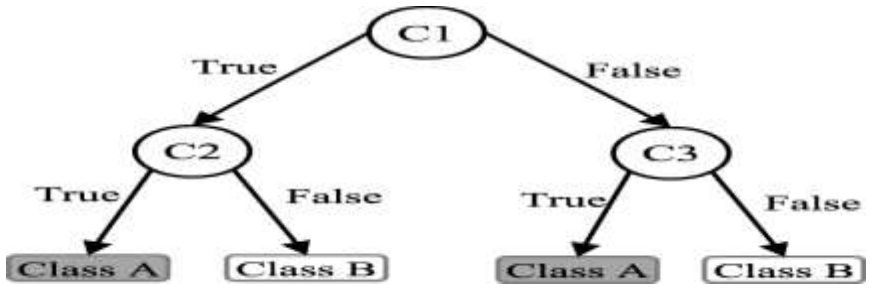


Figure 6. Shows Boosted decision tree algorithm

مجلـة جـامعة بـابـل للعلـــوم الصــرفة والتطـبيقية مـجلة جـامعة بـابـل للعلـــوم الصــرفة والتطـبيقية مـجلة جـامعة بـابـل

ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

❖ Averaged Perceptron (AP)

Perceptrons are also utilized in other known algorithms. Perceptron (as shown in Figure 7) can be succinctly defined as: “If X_1 through X_n are input feature values and w_1 through w_n are connection weights/prediction vector (typically real numbers in the interval [-1, 1]) then perceptron computes the sum of weighted inputs $\sum X_i w_i$ and output goes through an adjustable threshold: if the sum is above threshold, output is 1; else it is 0”. The perceptron approach has been most typically used to learn from a batch of training sample by frequently applying the algorithm across the training set until it finds a prediction vector that is correct across the time. The labels of the test set are then predicted using this prediction rule [13].

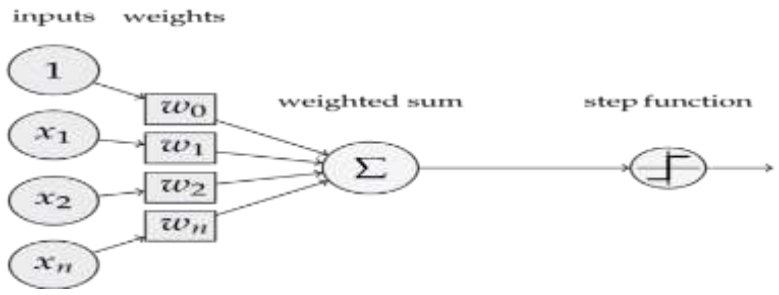


Figure 7. Shows the perceptron algorithm.

❖ Decision Forest (DF)

The goal of a decision forest is to improve a single decision tree's predictive performance by training several trees and aggregating their predictions [14]. Building a decision forest is one method to fully utilize the possibilities of decision trees. As the name implies, a decision forest is made up of numerous decision trees whose predictions are combined to form a single final forecast. By constructing a forest, a single decision tree's errors are mitigated by the forest's other decision trees [14]. A random forest (RF) is a combination classifier made up of many DTs, like to how a forest is made up of several trees. Figure 8 represent an illustration of the Random–Forest algorithm [15].

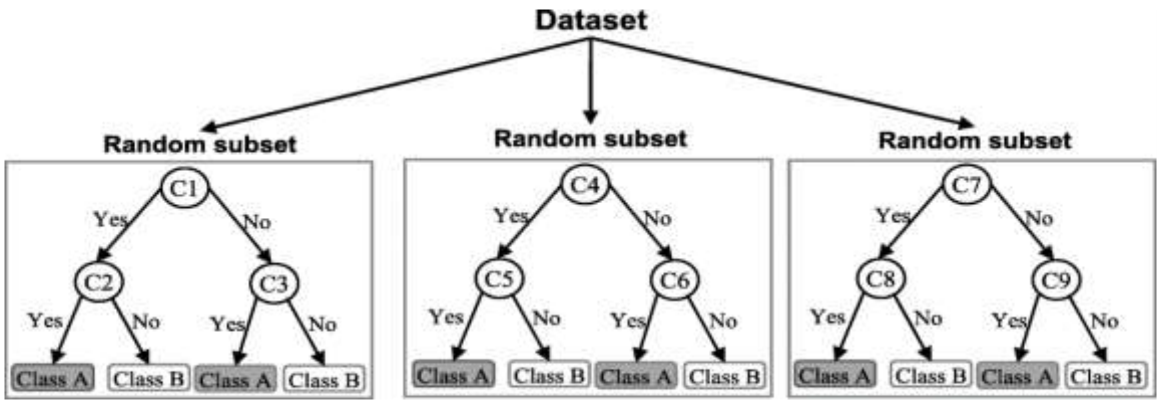


Figure 8. Decision forest.

❖ Support. Vector Machine (SVM)

The support vector machine (SVM) is utilized in each classification as well as regression. The SVM model assimilate the data items on a space and splits them into sets, with points with similar properties falling into the toke set. The given data set is viewed as a p-dimensional vector which can be subdivided into several as p-1 hyperplanes in linear SVM. For classification or regression problems, these planes partition the data space or define the boundaries among data groups, as shown in Figure 9. It splits a large number of hyperplanes based on the distance between the 2 classifications, the best hyperplane can be selected. The maximum-margin hyper-plane is the plane with the largest margin among the two classes [4].

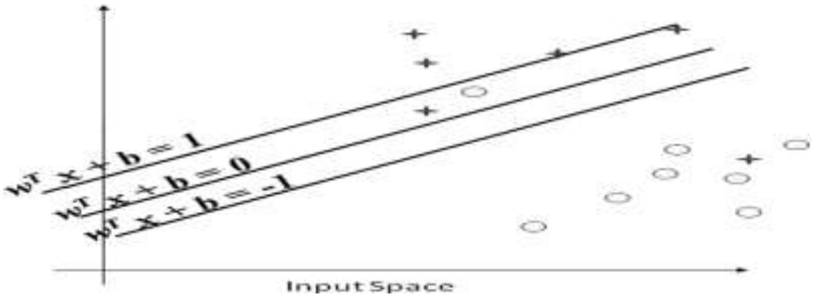


Figure 9. Support vector machine.

Figure 10 represents the flowchart of the algorithm (as an example, we take the decision forest here) we used and its diagram in Microsoft Azure Studio.

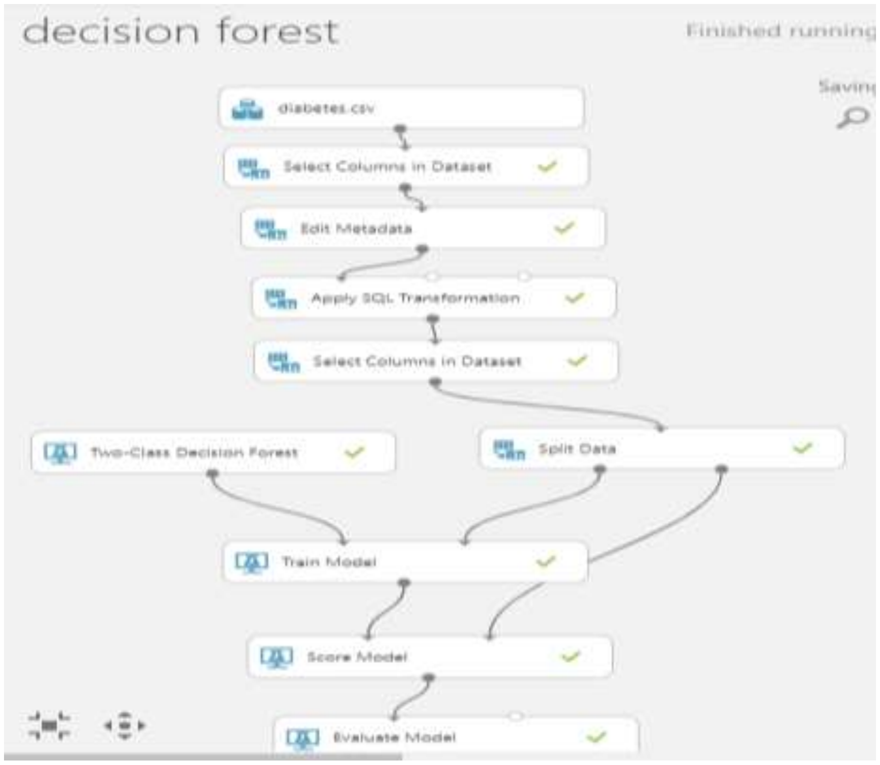


Figure 10. Decision Forest.

- Evaluate Model

The post-initial model creation test data is used to evaluate the model and based on the following performance metrics.

A confusion matrix is a technique for emphasizing the performance of a classification algorithm. When there are more than two classes in a model, or when each class has an unequal number of examples, accuracy alone can be perplexing. One benefit of a confusion matrix is that it's simple to detect if the system is unable to differentiate between two categorizes. (i.e., frequently mislabeling one as another). Table 3 shows the confusion matrix of classification algorithm.

TABLE 3. Confusion Matrix of Classification Algorithm.

		Prediction	
		0	**1**
Actual	**0**	TN	FP
	1	FN	TP

- **True Positive (TP):** when the actual classification of the data element is marked true also the ones that were predicted are likewise true at the finale
- **True Negatives (TN):** when the data point's actual class is false and the anticipated ones are also false at the end.
- **False Positives (FP):** When the data point's real class is False, but the predicted ones are true in the end.
- **False Negatives (FN):** When the real class of a data item is true, but the expected classes are false in the final.

Sensitivity and specificity are two objective factors in measuring the performance of the classifiers which were also employed in the study. Sensitivity is also referred to as the rate of recall, and measures the proportion of actual positives predicted correctly as such; the percentage of people who are correctly predicted as having a disease. Specificity determines the number of negatives that are accurately predicted as the percentage of healthy people who are accurately classified as not having the disease. To identify the best classifier in diabetic's disease classification, some of the performance parameters need to be selected to get a better outcome. As far as calculating the best results, the higher the Accuracy value, the better the classifier is. The measurements below are used to define the performance metrics that widely used in machine learning.

- **The Accuracy:** Accuracy is determined as the number of correctly predicted instances divided by the total number of instances. This implies that accuracy is the percentage of correctly predicted Positive cases within the total cases.

ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100$$

- **The Precision:** Precision refers to the consistency or precision of a true positive class , so regarded as a predictor with a positive value This is the number of occurrences as a percentage. that accurately have classified x / Total forecast as class x. So, the accurate findings were reported essentially in high precision and it takes all related data but just returns the highest performance.

$$Precision = \frac{TP}{TP + FP} \times 100$$

- **Recall:** Recall provides problem responsiveness and it processes values or quantity or completeness of the component. It returned the most relevant documents or results that are closely related to the problem. In simple words, for all the positive classes, how many are accurately predicting. This should be as raised as practically possible.

$$Recall = \frac{TP}{TP + FN} \times 100$$

- **F1_Score :** The F1 score is a calculation that attempts to achieve a balance by combining the percentages of accuracy and recall.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100$$

- **AUC-Score:** AUC A curve is a representation of performance at various thresholds. for classification problems. AUC reflects the degree of separability measurement. It shows how much a model can differentiate between classes.

- Deploying a Trained-Model according Web Service

To check diabetes every time a patient has to go to the diagnostic center, give his/ her blood and wait for the result, which takes a very long time and is also a waste of money, as it needs to be checked regularly. The main aim of this project is to develop a system as a web service, which can predict the diabetes of a patient with higher accuracy by considering all the factors that cause diabetes. After completing the previous stages, now that we've training and testing a model, it's time to start putting it to work. We'll give end-user software apps the ability to programmatically forecast values by delivering it as a web service. Figure 11 shows the flowchart of the proposed web service.

ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل

.

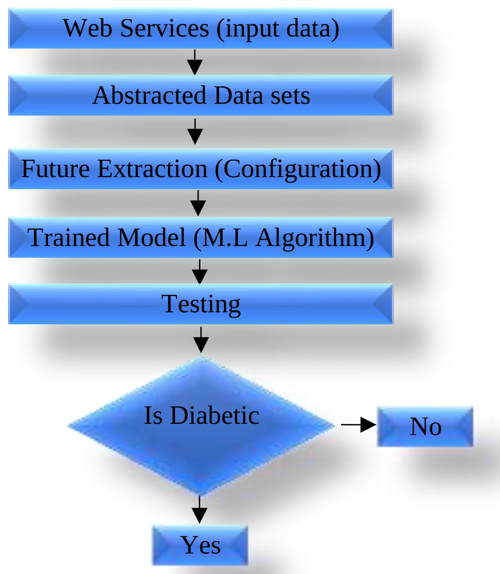

Web Services (input data)
Abstracted Data sets
Future Extraction (Configuration)
Trained Model (M.L Algorithm)
Testing
Is Diabetic
No
Yes

Figure 11. The flowchart of the proposed web service.

Results and Discussion

In this chapter, we show the implementation and performance evaluation as graphs and discussion for the proposed system outlined in Section “Materials and Methods”. The goal is twofold: first, to evaluate the performance of techniques with different performance metrics via a real diabetic dataset. Second, making a comparison among the results obtained with the used techniques.

- **The Experiment Environment**

ML applications can be created in a variety of ways. Someone, for example, can write code to implement a decision forest themselves, but this may cause a waste of time and it may be full of errors, hence it may be passive. Azure Machine-Learning Studio (AML) is one of the best ways for creating an end-to-end ML solution. AML Studio is a graphical interface for designing and implementing machine learning processes that use drag-and-drop functionality. AML Studio includes a large number of tried-and-true executions of tens of machine learning techniques for a wide range of problems.

- **The Experiment's Implementation**

It's time to demonstrate how to implement the proposed method for predicting diabetes using AML Studio, step by step, as follows:

  - **Creating an Account**

It's not necessary to have an Azure subscription to use AML Studio. It can quickly create a Microsoft Azure Studio account. After logging in to AML Studio, choose Blank Experiment, and a new screen appears, as in Figure 12 which can be saved to the workspace file for free that lets us use it as a Launchpad for our experiments. The workspace appears entirely empty during this stage.

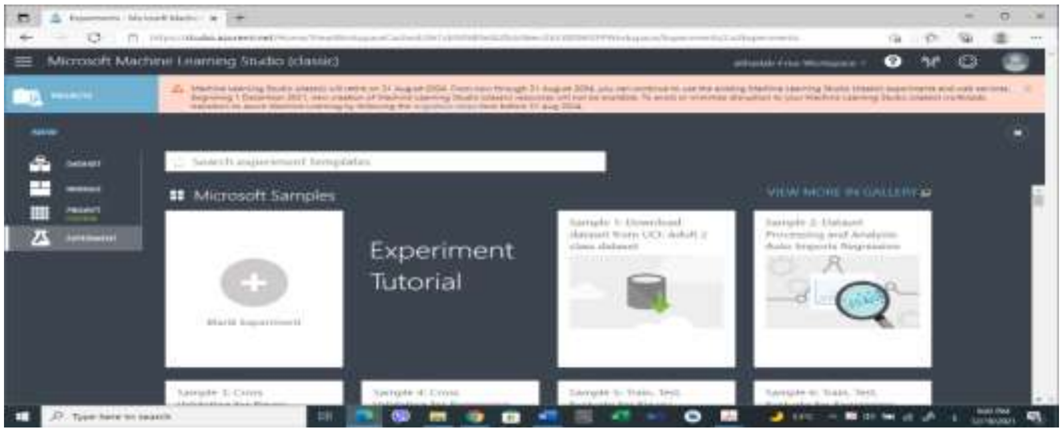


Figure 12. shows the creating of experimentation.

- **Importing Data**

A ML solution's most critical component is data. Importing data to assist in training our model is the first step. A dataset could be downloaded from the Internet and then imported into AML Studio, which accepts data in a variety of forms, including CSV, Excel, SQL, and others. We must download a dataset concerned with diabetes disease. Drag and drop the dataset onto the experiment canvas on the right-hand side, such as displayed in Figure 13.

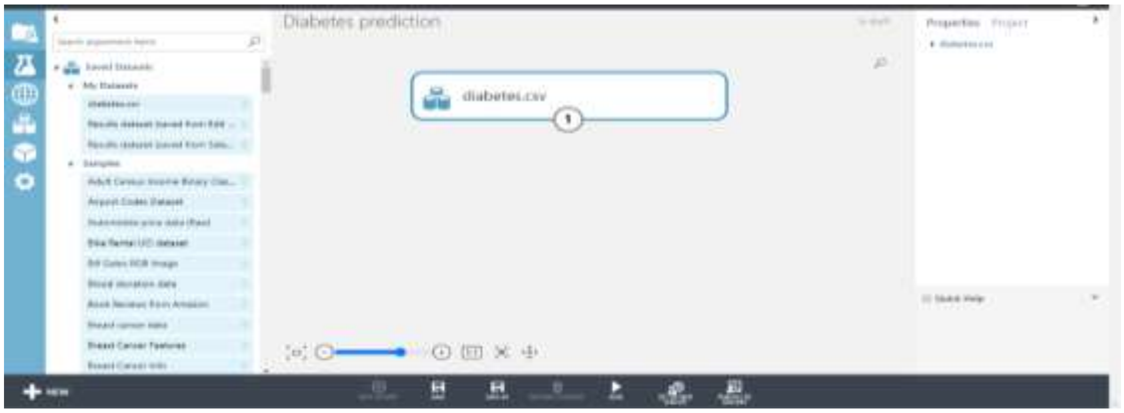


Figure 13. Shows the dataset in canvas.

- **Preprocessing Data**

The columns that are most significant to our experiment are as follows: "Glucose" (which denotes sugar level), "BloodPressure," "BMI," and "Age". The Category Variables column indicates whether or not the person has been estimated to have diabetes (1 or 0). With features obtained from other columns, this column will perform as our output (target). Examine a few cases to understand how different health

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

factors affect the outcome. When observing attentively, you will see that certain rows have zero values for columns where 0 makes absolutely no sense (e.g., SkinThickness, Insulin). These could be lost values that those sicker were unable to obtain. We can either use intelligent estimations to fill in all the missing values or eliminate them entirely to improve the dataset's quality. The former would necessitate a high level of data science expertise. For the sake of simplicity, let's remove all the rows with missing data. It's important to note that the Insulin and SkinThickness columns appear to have a large number of values that have vanished. To avoid a huge number of rows being lost while trying to clean our data, we must remove the whole column. We want to use as many different records as we can in order to enhance the accuracy of our trained model.

- **Removing Bad Columns**

From the sidebar, select "Data Transformation→Manipulation" and drag-and-drop Select Columns in Dataset onto the canvas, just beneath the dataset. Link the output port of the dataset to the input port of the newly added module. Drag the output port of the earlier toward the input port of the Choose Columns unit. This will establish a connection between the dataset and the unit as shown in Figure 14.

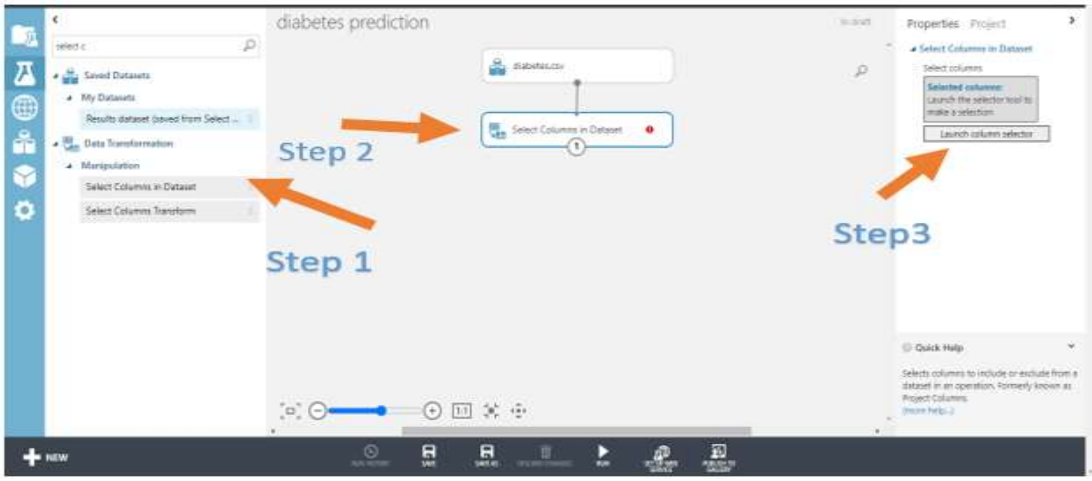


Figure 14. Steps of excluding bad columns in dataset.

- **Renaming Columns**

The original names in the dataset's collection of columns might be excessively lengthy; therefore, it is easier to simplify the column names and make them shorter. Therefore, the first thing we need to do is to search for the metadata editor and add it to the canvas at the bottom of the select_columns directly, and then connect it to the unit that is at the top of it. Then we move to the right part of the screen and click on the

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

Start Columns Selector and add the new names in the new column names field. We must leave all the other fields unchanged.

- **Removing Rows with Missing Values**

This is usually accomplished using the Clean Missing Data module, which can delete rows with no specified values in all or any of the columns. This module will not work for us because there are no particular missing data in our data set. We have to get a technique to filter out rows with "0" in specified columns (for example, glucose_level, bmi, so on). In the canvas, insert the Perform SQL Transformation module under Edit Metadata. Fill up the module's with the next query:

$$select * from\ t1\ where\ glucose\_level\ != 0\ and\ diastolic\_blood\_pressure\ != 0\ and\ bmi\ != 0\ and\ age\ != 0;$$

- **Defining Features**

Our data is now ready to be used for training. However, before we begin training, humans must manually select important columns from our data to utilize as attributes. In the canvas, insert a second select column and put it directly beneath the "Apply SQL Transformation button". Then pick "glucose_level, diastolic_blood_pressure, diabetes_pedigree_function, bmi, and age."

- **Splitting Data**

The dataset must be split into two portions: training and testing data. Though training data is employed to train the model, scoring data is employed to test and award a loss to the learned model. Insert the Split-Data-module, which is located directly below the second-Select-Columns-module, onto the canvas. Set the Fraction of Rows in the First Output Dataset to 0.75 on its properties page. The Split Data module's left output port can be used to review training data, while the right output port can be used to verify score data once it's completed. Both should be found in a 75-25% ratio.

- **Applying ML Algorithm**

Now we have to choose an algorithm that should be suitable for training the data as well as suitable for the required output. Algorithm modules do not have input ports, so we cannot connect them to any module at present.

- **Training the Model**

Add the Train Model module to the canvas, directly below the algorithm and Split Data components. Connect its left input port to the algorithm module and its right

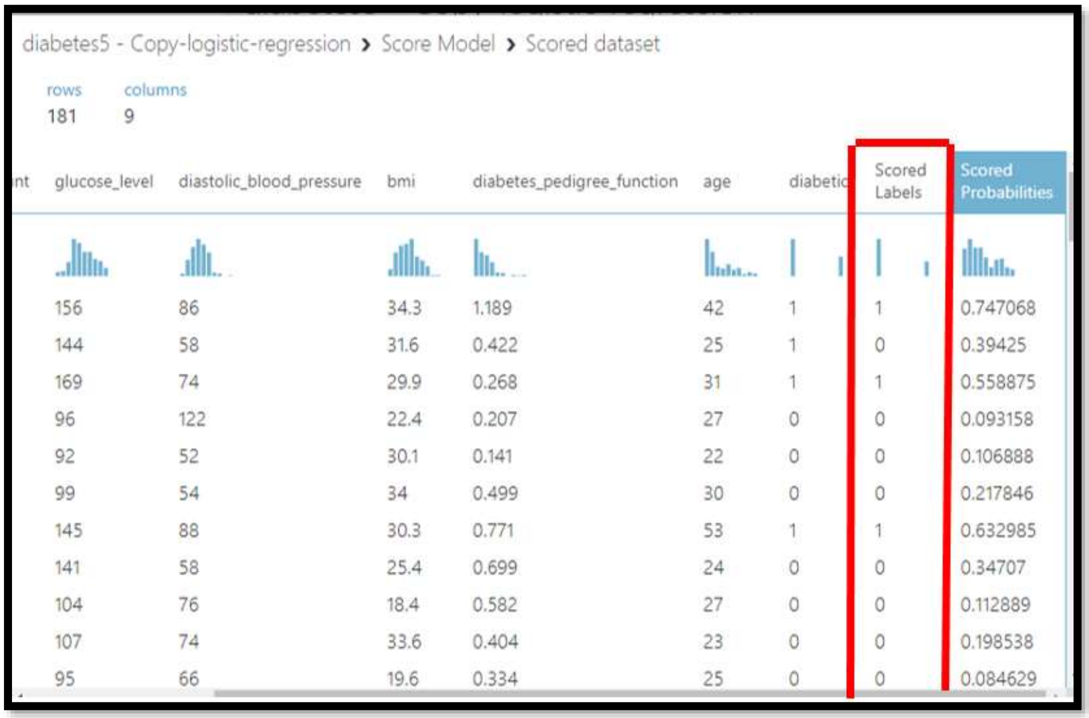
ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

port to Split Data's left output port. Carry out the experiment. If everything goes as planned, all units will receive a green checkmark.

▪ Scoring and Evaluating the Trained Model

Drag the Score Model subsystem beneath the Train Model subsystem onto the canvas. Link the Train-Model-module to the left input port and Split_Data to the right output port. Carry out the experiment, and the result will be as shown in Figure 15. The complete experiment must appear similar to Figure 16.


diabetes5 - Copy-logistic-regression › Score Model › Scored dataset

rows 181 columns 9

glucose_level	diastolic_blood_pressure	bmi	diabetes_pedigree_function	age	diabetic	Scored Labels	Scored Probabilities
156	86	34.3	1.189	42	1	1	0.747068
144	58	31.6	0.422	25	1	0	0.39425
169	74	29.9	0.268	31	1	1	0.558875
96	122	22.4	0.207	27	0	0	0.093158
92	52	30.1	0.141	22	0	0	0.106888
99	54	34	0.499	30	0	0	0.217846
145	88	30.3	0.771	53	1	1	0.632985
141	58	25.4	0.699	24	0	0	0.34707
104	76	18.4	0.582	27	0	0	0.112889
107	74	33.6	0.404	23	0	0	0.198538
95	66	19.6	0.334	25	0	0	0.084629

Figure 15. Visualizing scored data.

▪ Deploying a Trained Model as a Web Service

At this time, we have a trained and tested model, and it is ready to be put to actual use. We'll provide end-user software apps the ability to programmatically forecast values by delivering them as a web service. For example, we use the diabetes prediction model as a web service in the IoT solution backend or with the Azure Stream service for real-time diabetes prediction.

Select the "Set Up Web Service→Predictive Web Service option". The present experiment would be transformed into a forecasting experiment as a result of this change. As shown in Figure 17.

ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

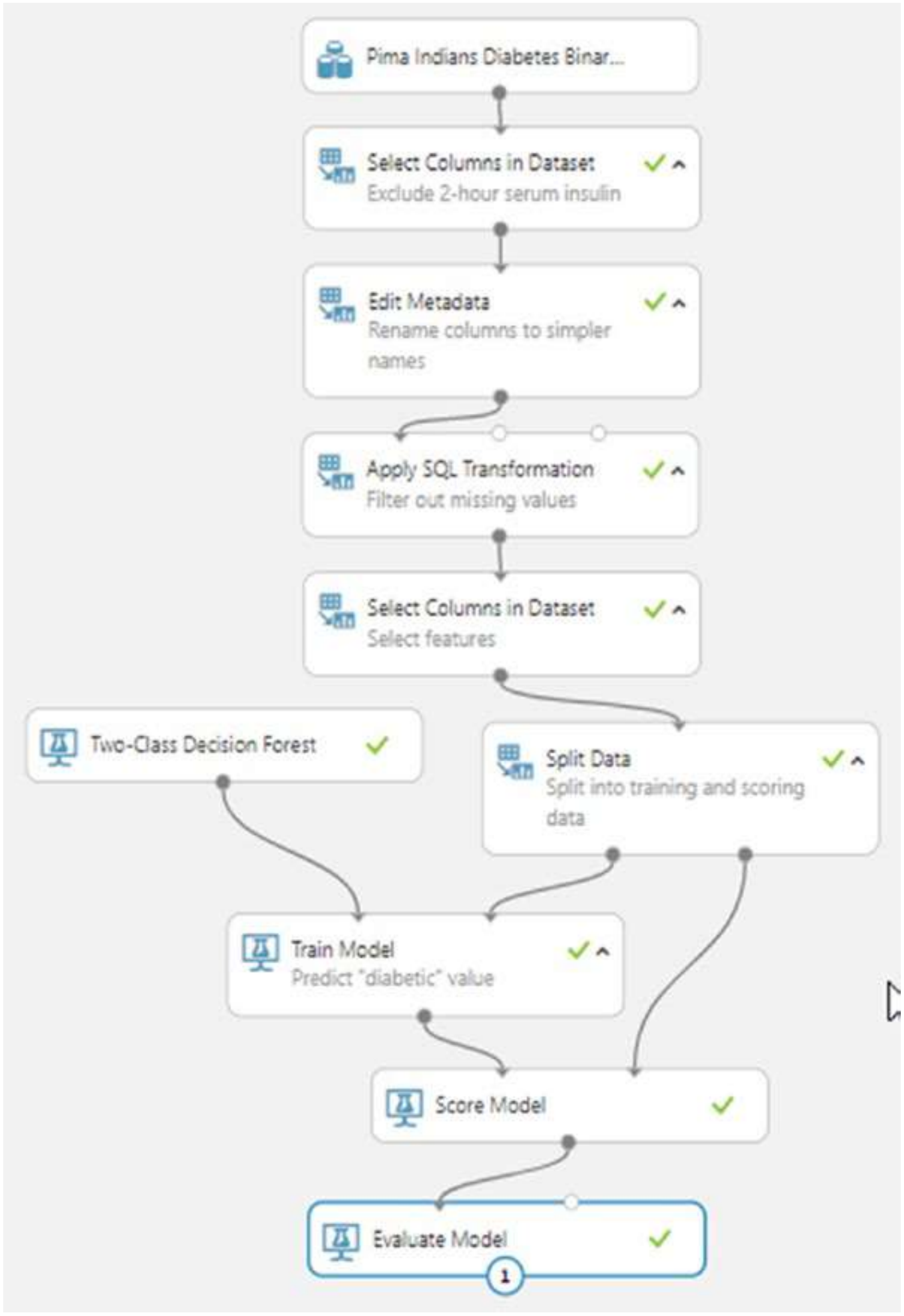

Pima Indians Diabetes Binar...
Select Columns in Dataset
Exclude 2-hour serum insulin
Edit Metadata
Rename columns to simpler names
Apply SQL Transformation
Filter out missing values
Select Columns in Dataset
Select features
Two-Class Decision Forest
Split Data
Split into training and scoring data
Train Model
Predict "diabetic" value
Score Model
Evaluate Model
1

Figure 16. A final preview of an ML solution for two-class classification.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

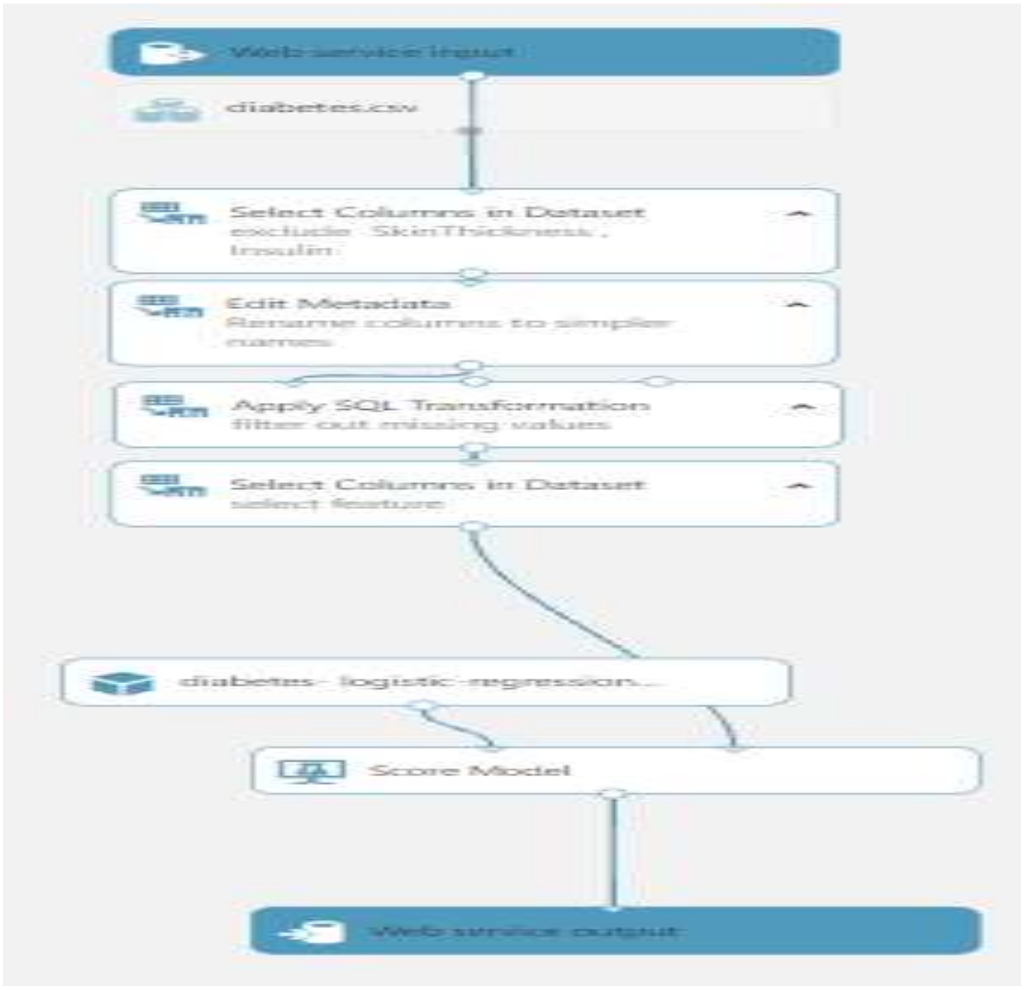

Web service input
diabetes.csv
Select Columns in Dataset
exclude: SkinThickness, Insulin
Edit Metadata
Rename columns to simpler names
Apply SQL Transformation
filter out missing values
Select Columns in Dataset
select feature
diabetes- logistic-regression...
Score Model
Web service output

Figure 17. Illustrate the converting of canvas as predictive experiment.

After publishing the system as a web service, users can access it and enter their medical data to diagnose their health condition. As an example, for a person who uses this service to find out whether he has diabetes or not (see Figure 18), by entering the following information as in Table 4.

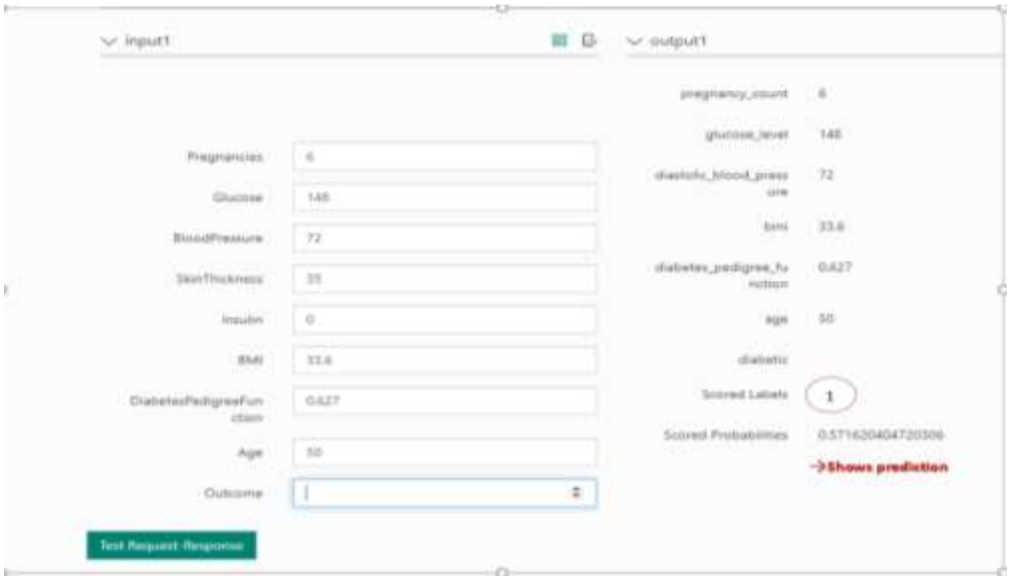
Table 4. The patient information.

Information	Value
Pregnancy – count	6
Glucose – level	148
Diastolic – blood – pressure	72
bmi	35
insulin	Null
Diabetes – pedigree – function	33.6
Age	50

Diabetic is the item we want to predict. The result is Scored Probabilities = 1 (diabetic).

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل



Pregnancies	6
Glucose	148
BloodPressure	72
SkinThickness	35
Insulin	0
BMI	33.6
DiabetesPedigreeFunction	0.627
Age	50
Outcome	

pregnancy_count	6
glucose_level	148
diastolic_blood_pressure	72
bmi	33.6
diabetes_pedigree_function	0.627
age	50
diabetic	
Scored Labels	1
Scored Probabilities	0.571620404720306

→Shows prediction

Figure 18. Shows the implementation of the prediction operation.

- **Experimental Results**

We will present results for prediction and efficiency of each algorithm depending on $true\ positive\ (TP)$, $false\ negative\ (FN)$, $false\ positive\ (FP)$, $true\ negative\ (TN)$, $accuracy$, $precision$, $recall$, and AUC. Throughout all the experiments, 70% of the data was used for training and 30% for testing. The six ML models were applied to the PIMA diabetes dataset in the following experiments, and one attribute was chosen as "Label" and utilized for instance categorization.

- **Logistic Regression (LR)**

The following findings of execution LR were acquired in this experiment and after the learning process. From the results, we can notice the following:

- The ROC (Receiver Operating Characteristic) curve depicts the trade-off between the True Positive Rate (the proportion of positive instances that are properly diagnosed) and the False Positive Rate (positive cases incorrectly classified). This curve is to the left and above the bright 45-degree line as shown in Figure 19. This shows that the classifier is more successful than guessing.
- Table 5 shows the confusion matrix generated by this experiment.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com ISSN: 2312-8135 | Print ISSN: 1992-0652

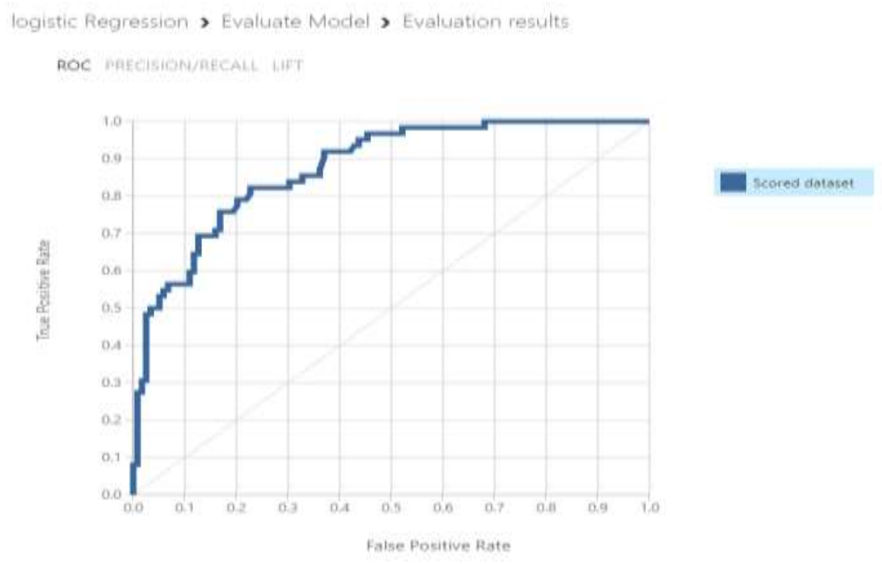


Figure 19. Receiver Operating Characteristic of LR.

Table 5. Confusion matrix Using LR algorithm.

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	111	8
Diabetes (1)	27	35

The LR produced a reasonable confusion matrix, indicating that of the 30% of diabetic datasets used for testing, (35) were true positive, i.e., they were correctly predicted to be diabetic, and (27) were false negative, i.e., they were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that 30% of non-diabetic datasets were used for testing. Eight of them were false positives, which means they were incorrectly projected to be non-diabetic, while (111) were true negatives, which means they were properly predicted to be non-diabetic.

- The **AUC** is greater than 0.87, suggesting that the classifier is practically good.
- The **Accuracy** is currently more than 80%.
- The **Recall** rate is presently close to 0.57.
- The **Precision** of 0.814 indicates that False Positives have been considerably decreased.
- The **F1** score is 0.667.

- **Decision Forest (DF):**

The following findings of execution DF were acquired in this experiment and after the learning process. From the results, we can notice the following:

- The ROC curve is to the left and above the bright 45-degree line as shown in Figure 20. This shows that the classifier is more successful than guessing.

ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

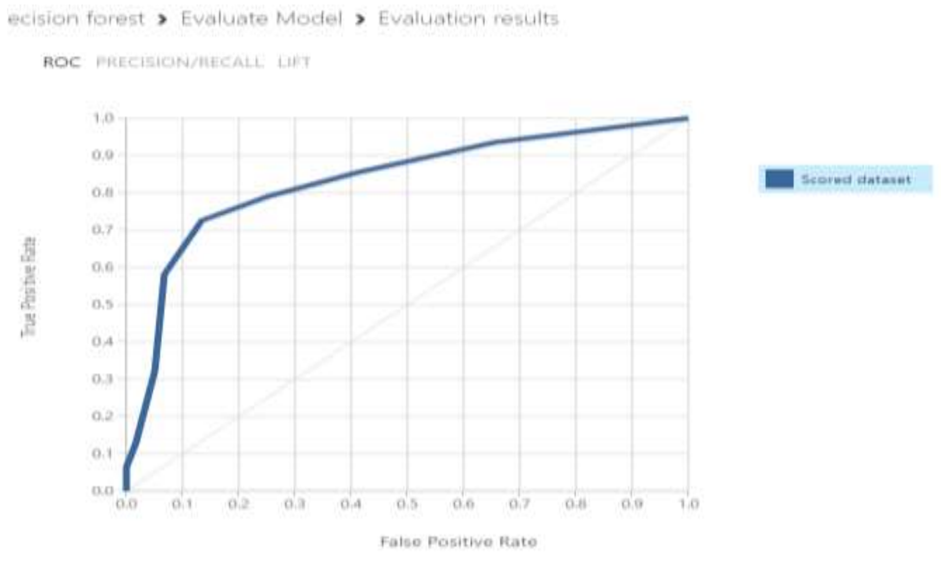


Figure 20. Receiver Operating Characteristic of DF.

- ❖ Table 6 shows the confusion matrix generated by this experiment.

Table 6. Confusion matrix Using DF algorithm

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	111	8
Diabetes (1)	26	36

The DF produced a reasonable confusion matrix, indicating that (36) were correctly predicted to be diabetic, and (26) were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that (8) of them were incorrectly projected to be non-diabetic, while (111) were were properly predicted to be non-diabetic.

- ❖ The **AUC** is greater than 0.83, suggesting that the classifier is practically good.
- ❖ The **Accuracy** is currently more than 81%.
- ❖ The **Recall** rate is presently close to 0.58.
- ❖ The **Precision** of 0.818 indicates that False Positives have been considerably decreased.
- ❖ The **F1** score is 0.679.

- ▪ **Averaged Perceptron (AP)**

The following findings of execution AP were acquired in this experiment and after the learning process. From the results, we can notice the following:

- ❖ The ROC curve is to the left and above the bright 45-degree line as shown in Figure 21. This shows that the classifier is more successful than guessing.

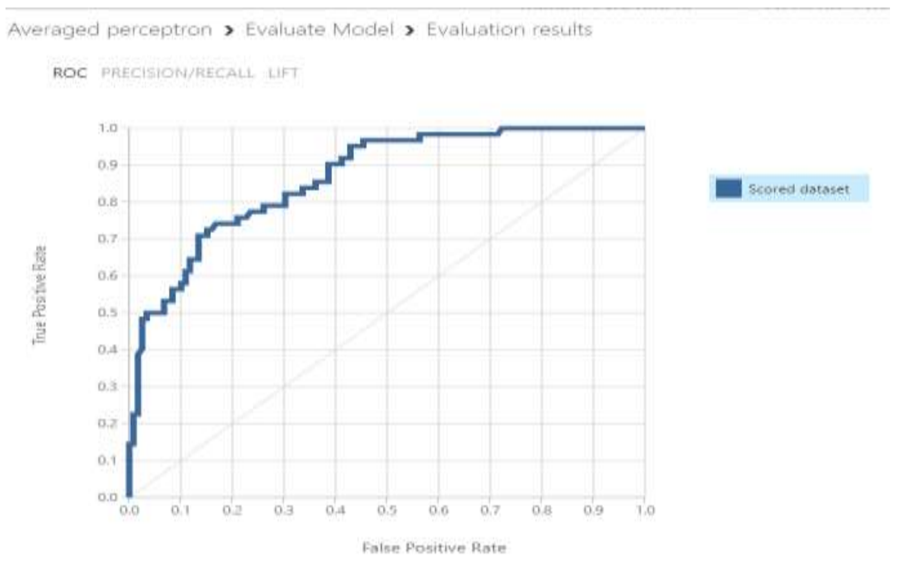


Figure 21. Receiver Operating Characteristic of AP.

❖ Table 7 shows the confusion matrix generated by this experiment.

Table 7. Confusion matrix Using AP algorithm

	Non-Diabetes (0)	**Diabetes (1)**
Non-Diabetes (0)	106	13
Diabetes (1)	26	36

The AP produced a reasonable confusion matrix, indicating that (36) were correctly predicted to be diabetic, and (26) were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that (13) of them were incorrectly projected to be non-diabetic, while (106) were properly predicted to be non-diabetic.

❖ The **AUC** is greater than 0.86, suggesting that the classifier is practically good.
❖ The **Accuracy** is currently more than 78%.
❖ The **Recall** rate is presently close to 0.58.
❖ The **Precision** of 0.735 indicates that False Positives have been considerably decreased.
❖ The **F1** score is 0.649.

▪ **Neural Network (NN)**

The following findings of execution NN were acquired in this experiment and after the learning process. From the results, we can notice the following:

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

- The ROC curve is to the left and above the bright 45-degree line as shown in Figure 22. This shows that the classifier is more successful than guessing.

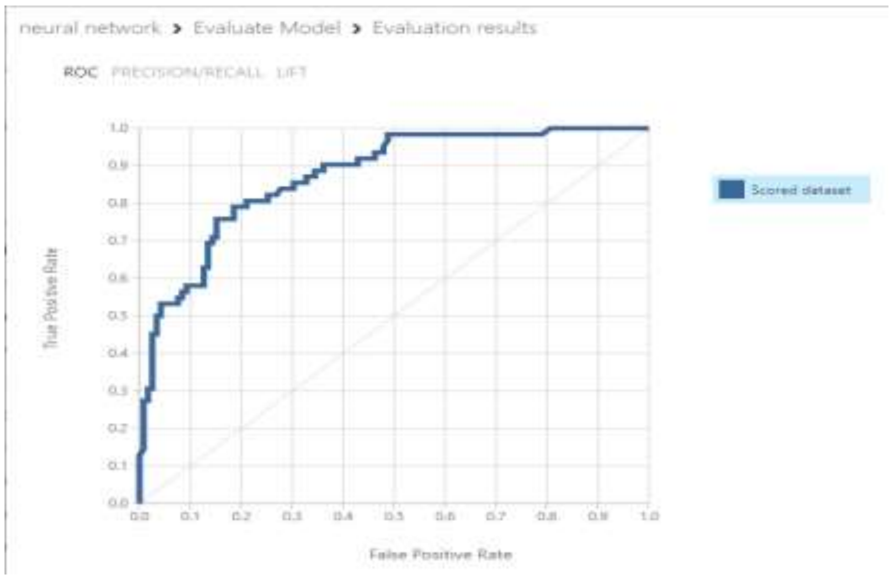


Figure 22. Receiver Operating Characteristic of NN.

- Table 8 shows the confusion matrix generated by this experiment.

Table 8. Confusion matrix Using NN algorithm

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	106	13
Diabetes (1)	26	36

The NN produced a reasonable confusion matrix, indicating that (36) were correctly predicted to be diabetic, and (26) were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that (13) of them were incorrectly projected to be non-diabetic, while (106) were properly predicted to be non-diabetic.

- The **AUC** is greater than 0.87, suggesting that the classifier is practically good.
- The **Accuracy** is currently more than 78%.
- The **Recall** rate is presently close to 0.58.
- The **Precision** of 0.735 indicates that False Positives have been considerably decreased.
- The **F1** score is 0.649.

- **Support Vector Machine (SVM)**

The following findings of execution SVM were acquired in this experiment and after the learning process. From the results, we can notice the following:

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل للعلوم الصرفة والتطبيقية مجلة جامعة بابل

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

- ❖ The ROC curve is to the left and above the bright 45-degree line as shown in Figure 23. This shows that the classifier is more successful than guessing.

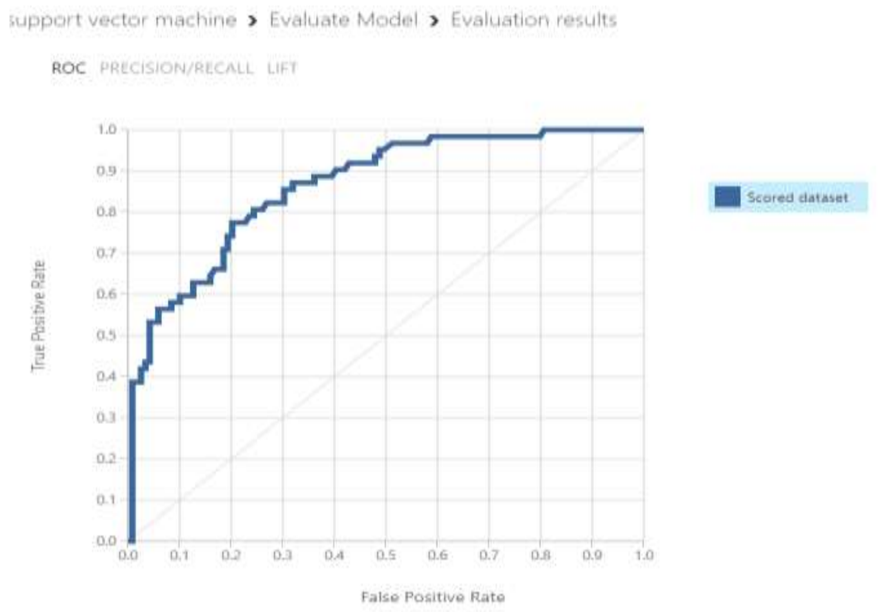


Figure 23. Receiver Operating Characteristic of SVM.

- ❖ Table 9 shows the confusion matrix generated by this experiment.

Table 9. Confusion matrix Using NN algorithm

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	107	12
Diabetes (1)	26	36

The SVM produced a reasonable confusion matrix, indicating that (36) were correctly predicted to be diabetic, and (26) were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that (13) of them were incorrectly projected to be non-diabetic, while (106) were properly predicted to be non-diabetic.

- ❖ The **AUC** is greater than 0.86, suggesting that the classifier is practically good.
- ❖ The **Accuracy** is currently more than 79%.
- ❖ The **Recall** rate is presently close to 0.58.
- ❖ The **Precision** of 0.735 indicates that False Positives have been considerably decreased.
- ❖ The **F1** score is 0.655.

- **Boosted Decision Tree (BDT)**

The following findings of execution BDT were acquired in this experiment and after the learning process. From the results, we can notice the following:

❖ The ROC curve is to the left and above the bright 45-degree line as shown in Figure 24. This shows that the classifier is more successful than guessing.

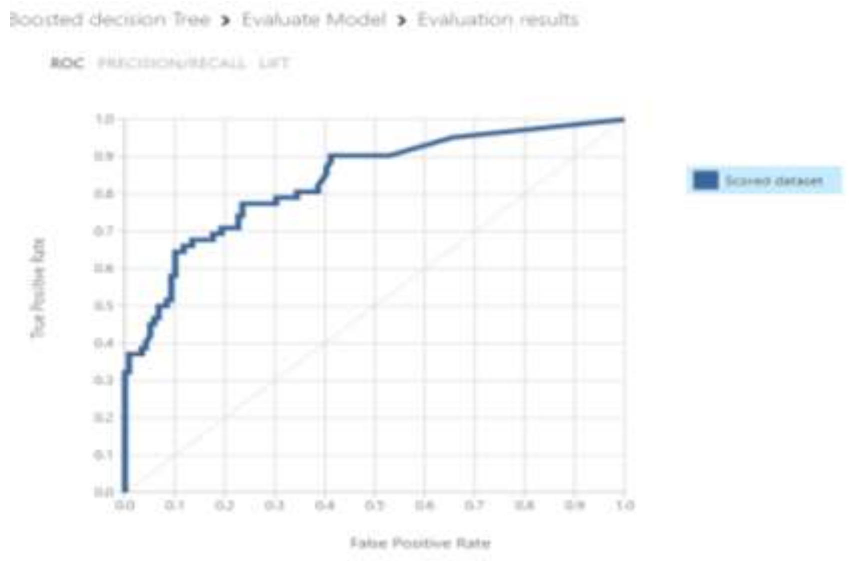


Figure 24. Receiver Operating Characteristic of BDT.

❖ Table 10 shows the confusion matrix generated by this experiment.

Table 10. Confusion matrix Using BDT algorithm

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	107	12
Diabetes (1)	25	37

The BDT produced a reasonable confusion matrix, indicating that (37) were correctly predicted to be diabetic, and (25) were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that (12) of them were incorrectly projected to be non-diabetic, while (107) were properly predicted to be non-diabetic.

❖ The **AUC** is greater than 0.83, suggesting that the classifier is practically good.
❖ The **Accuracy** is currently more than 79%.
❖ The **Recall** rate is presently close to 0.6.
❖ The **Precision** of 0.755 indicates that False Positives have been considerably decreased.
❖ The **F1** score is 0.667.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

- **Comparing the Performance of ML Algorithms**

In this section, a comparison between the performances of the six algorithms that were used in the prediction process for the diagnosis of diabetes is made. The six algorithms are, namely, DF, BDT, LR, SVM, NN, and AP. The highest accuracy recorded was 81.2 and 80.7 for each of the DF and LR algorithms, respectively. The algorithms that scored the highest AUC are LR, equal to 87.5%, NN, equal to 87.3%, and AP, equal to 87.6%. And the algorithm that scored the highest precision was DF, which was 81.8%, as shown in Table 11.

Table 11. Performance evaluation of ML algorithms

Algorithm	Accuracy	True-Positive	False-Negative	Precision	False-Positive	True-Negative	Recall	F1_score	AUC
SVM	79 %	36	12	0.75	12	107	0.581	0.655	0.863
LG	80.7%	35	27	0.814	8	111	0.565	0.667	0.875
DF	81.2 %	36	26	0.818	8	111	0.581	0.679	0.833
AP	78.5 %	36	26	0.735	13	106	0.581	0.649	0.867
NN	78.5 %	36	26	0.735	13	106	0.581	0.649	0.873
BDT	79.6 %	37	25	0.755	12	107	0.597	0.667	0.832

- **The Result of Prediction using Web Service**

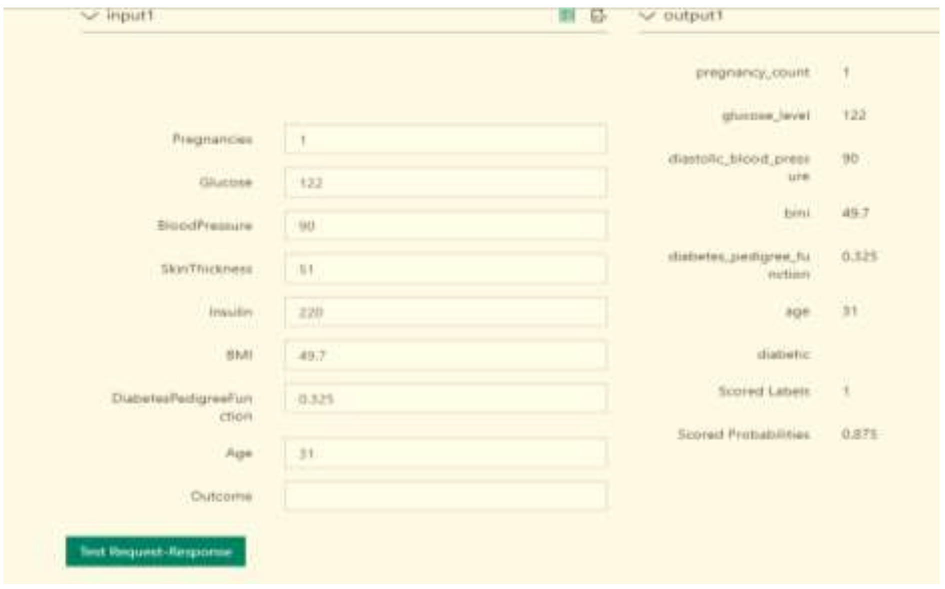
This section displays the results of applying the proposed system after converting it into a web service and based on the Decision Forest ML algorithm. Figures 25 and 26 show the results obtained after entering the required information.

In Figure 25, it appears the patient is diabetic. When the scored labels get to (1), it means that a diabetic is found as there are only two cases, infected 1 and uninfected 0, and the scored probability equals 0.875, which is greater than 0.500, then the diabetes is found.

In Figure 26, it appears the patient is free of diabetes. When the scored labels get to (0), it means that a diabetic is not found as there are only two cases, infected 1 and uninfected 0, and the scored probability equals 0.125. Surely, it's a number less than 0.500 as we mentioned earlier, it means diabetes-free.

ISSN: 2312-8135 | Print ISSN: 1992-0652 info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

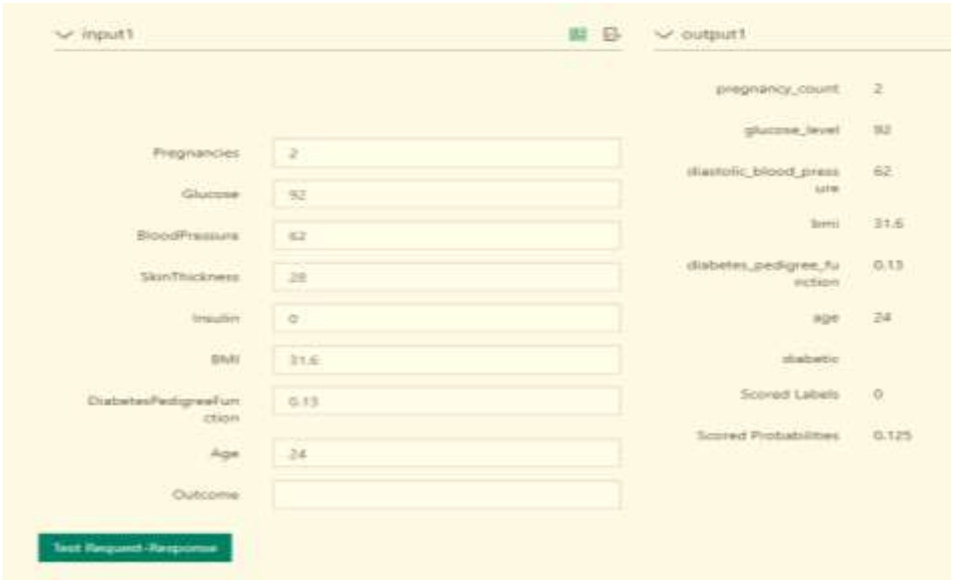
ISSN: 2312-8135 | Print ISSN: 1992-0652



Pregnancies	1
Glucose	122
BloodPressure	90
SkinThickness	51
Insulin	220
BMI	49.7
DiabetesPedigreeFunction	0.325
Age	31
Outcome	

pregnancy_count	1
glucose_level	122
diastolic_blood_pressure	90
bmi	49.7
diabetes_pedigree_function	0.325
age	31
diabetic	
Scored Labels	1
Scored Probabilities	0.875

Figure 25. Illustrates the prediction of diabetes in Web service.



Pregnancies	2
Glucose	92
BloodPressure	62
SkinThickness	28
Insulin	0
BMI	31.6
DiabetesPedigreeFunction	0.13
Age	24
Outcome	

pregnancy_count	2
glucose_level	92
diastolic_blood_pressure	62
bmi	31.6
diabetes_pedigree_function	0.13
age	24
diabetic	
Scored Labels	0
Scored Probabilities	0.125

Figure 26. Shows predicting the absence of diabetes.

Conclusion

ML aids in the creation of prediction methods for diabetes and associated consequences. In our proposed system machine learning technology was used with Microsoft Azure to make a system to predict diabetes. Six machine learning algorithms were used, and the best accuracy was achieved by the decision forest algorithm, where the accuracy reached more than 81%, which supports the use of machine learning methods to predict the occurrence of diabetes, as the result depends on the outcome column, which consists of only two cases, either 0 indicates that the patient is free of diabetes, or 1 indicates that the patient has diabetes. Our project can contribute to the integration of individualized care into diabetes control. People appear to have greater control over their personal health, and doctors can use technological platforms to deliver timely and targeted interventions. Since data is possible to be obtained remotely and virtual monitoring replaces periodic clinic visits, these advancements save time and money. The proposed system is easy and can be applied for free.

As a future work, we intend to investigate additional attributes that could be utilized as predictors. As a result, this research can be broadened to aid in the fully automated prediction of diabetes. We can't forecast the type of diabetes, so in the future, we'll try to predict the kind of diabetes and look at the percentage of each indicator to see if we can enhance the accuracy of diabetes prediction. Also, investigating the possibility of linking the system with the IoT network to fetch information quickly and in real-time in order to predict the patient's condition more efficiently, accurately, and quickly.

Conflict of interests.

There are non-conflicts of interest.

References

[1] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, and H. Tang, "Predicting Diabetes Mellitus With Machine Learning Techniques," *Front. Genet.*, vol. 9, 2018, doi: 10.3389/fgene.2018.00515.

[2] L. Ismail and H. Materwala, "IDMPF: intelligent diabetes mellitus prediction framework using machine learning," *Appl. Comput. Informatics*, vol. ahead-of-print, no. ahead-of-print, 2021, doi: 10.1108/aci-10-2020-0094.

[3] D. Ramana Kumar and S. K. Rao, "Health care System: Stream Machine Learning Classifier for features Prediction in Diabetes Therapy," 2018.

[4] M. A. Sarwar, N. Kamal, W. Hamid, and M. A. Shah, "Prediction of diabetes using machine learning algorithms in healthcare," 2018, doi: 10.23919/IConAC.2018.8748992.

[5] R. Forsman, and J. Jönsson, "Artificial intelligence and Machine learning: a diabetic readmission study," 2019.

[6] S. Patel and V. N. Khedkar, "Diabetes Prediction using Machine learning: A Bibliometric Analysis," *Libr. Philos. Pract.*, vol. 2021, 2021.

[7] T. N. Joshi, and P. P. M. Chawan, "Diabetes prediction using machine learning techniques," *Ijera*, vol. 8, no. 1, pp.9-13, 2018.

[8] S. M. Hasan Mahmud, M. A. Hossin, M. Razu Ahmed, S. R. H. Noori, and M. N. I. Sarkar, "Machine learning based unified framework for diabetes prediction," 2018, doi: 10.1145/3297730.3297737.

[9] M. Aminul and N. Jahan, "Prediction of Onset Diabetes using Machine Learning Techniques," *Int. J. Comput. Appl.*, vol. 180, no. 5, 2017, doi: 10.5120/ijca2017916020.

[10] C. Reid, Diabetes "Diagnosis and Readmission Risks Predictive Modelling: USA," (Doctoral dissertation, Dublin, National College of Ireland), 2019.

[11] H. Bhavsar and A. Ganatra, "A Comparative Study of Training Algorithms for Supervised Machine Learning," *Int. J. Soft Comput. Eng.*, vol. 2, no. 4, 2012.

[12] F. Y. Osisanwo, J. E. T. Akinsola, O. Awodele, J. O. Hinmikaiye, O. Olakanmi and J. Akinjobi, "Supervised machine learning algorithms: classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)*, vol. *48, no.* 3, pp.128-138, 2017.

[13] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica (Ljubljana)*, vol. 31, no. 3. 2007, doi: 10.31449/inf.v31i3.148.

ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

[14] L. Rokach, “Decision forest: Twenty years of research,” *Inf. Fusion*, vol. 27, 2016, doi: 10.1016/j.inffus.2015.06.005.

[15] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, “Comparing different supervised machine learning algorithms for disease prediction,” *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 1, 2019, doi: 10.1186/s12911-019-1004-8.

[16] N. Pathak and A. Bhandari, *IoT, AI, and Blockchain for .NET*. 2018.

ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com