Solving diabetes diagnosis problems using machine learning

Donaxon Olimboyeva[1], *Davron* Ziyadullaev[2,*], *Dilnoz* Mukhamedieva[2], *Khosiyat* Khujamkulova[2], *Mukhammadyahyo* Teshaboyev[3], and *Gulchiroy* Ziyodullaeva[4]

[1]"Alfraganus University" is a non-state higher education institution, Tashkent, Uzbekistan
[2]National Research University "Tashkent Institute of Irrigation and Agricultural Mechanization Engineers institute", 100000 Tashkent, Uzbekistan
[3]Andijan State Medical Institute, 170100 Andijan, Uzbekistan
[4]Tashkent University of Information Technologies named after Muhammad Al-Khwarizmi, 100200 Tashkent, Uzbekistan

Abstract. This research is devoted to the study of the use of machine learning methods to solve the problem of diagnosing diabetes. The results of using machine learning in the context of diabetes are varied and depend on the methods of data analysis, the models used and the quality of the data provided. Experiments on the Diabetes dataset were conducted in the study using a Naive Bayes classifier model and a linear kernel SVM for a binary classification problem. Models are trained on the training dataset, standardizing features, and evaluated on the test set using confusion, precision, recall, F1-measure, and AUC-ROC metrics. The results obtained confirm that machine learning can improve the accuracy of diagnosing diabetes and classifying its type. This allows for customized treatment plans to be developed, considering the unique characteristics of each patient. Machine learning models are also successful in predicting the likelihood of complications, allowing for preventative measures to be taken. Their use facilitates the integration of data from various sources, enriching patient information. In conclusion, machine learning-based decision support systems assist physicians and patients in making informed decisions.

1 Introduction

Diabetes is a chronic disease that affects millions of people around the world. For early diagnosis and effective management of diabetes, the medical community is turning to machine learning techniques and data analysis. The relevance of machine learning in the context of diabetes cannot be underestimated. This is a current research and practical area that has a number of important aspects. Diabetes is an increasingly common disease and its epidemic is growing steadily. Machine learning provides tools to more effectively diagnose and manage disease. Early diagnosis of diabetes and its type is key to successful treatment and prevention of complications. Machine learning makes it possible to develop models for early diagnosis. Each patient is unique, and machine learning can help create personalized

* Corresponding author: dziyadullaev@inbox.ru

treatment plans based on genetic, clinical, and laboratory data. Medical data includes a variety of sources and formats. Machine learning can integrate and analyze this data, creating a more complete picture of patients' conditions. Complications of diabetes can be dangerous. Machine learning models can predict the likelihood of their development and help to take measures to prevent them [1-4].

The *Diabetes dataset* is a valuable resource for researchers and practitioners in the fields of medical data analytics and machine learning. It enables the development and testing of models to predict and monitor the progression of diabetes and may contribute to improved approaches to the diagnosis and management of this serious chronic disease. The *Diabetes dataset* includes 10 numeric features that describe the patient's conditions. These features represent important medical measurements and health characteristics. Below is a more detailed description of each of these features [5-6]:

Blood Sugar Level: This feature represents the level of glucose in the patient's blood. This is an important measurement for diagnosing diabetes and monitoring its control.

Blood Pressure: This feature shows the patient's blood pressure. High blood pressure may be associated with cardiovascular disease, which often accompanies diabetes.

Body Mass Index (BMI): BMI shows the ratio of a patient's weight to height and is used to determine whether someone is overweight or obese; this can be a risk factor for developing diabetes.

Serum Levels of Heart Disease Marker: This feature is an index of the level of a specific molecule that may be associated with the risk of developing cardiovascular disease in patients with diabetes.

Serum Low-Density Lipoprotein Cholesterol: This feature measures the level of "bad" cholesterol in the blood, which may also be associated with cardiovascular risk.

Serum High-Density Lipoprotein Cholesterol: This feature measures the level of "good" cholesterol in the blood and is an important factor in assessing cardiovascular health.

Natural Log of Three Insulin Parameters: This feature is the natural logarithm of three parameters related to insulin and metabolism.

Age: This feature represents the age of the patient at the time of observation. Age may be an important factor in the development and progression of diabetes.

Sex: This feature reflects the patient's gender (male or female) and may have an impact on the risk of developing diabetes.

C-Peptide Level: C-peptide is a molecule produced along with insulin. Its level may be related to pancreatic function and insulin levels.

These numerical features provide important information about patients' conditions and can be used to develop machine learning models that predict diabetes progression and assess risk [7].

The purpose of applying machine learning in the context of diabetes may vary depending on the specific task and scenario, but common goals include early diagnosis and risk prediction. The goal is to create models that can diagnose diabetes early or predict the risk of its development. This allows treatment to begin in the early stages of the disease and prevent its complications [8].

Machine learning can help develop personalized treatment and monitoring plans for each patient based on their unique characteristics and response to treatment. Machine learning can be used to monitor patient data to prevent complications such as hypoglycemia or diabetic ketoacidosis. The main goal of machine learning in the context of diabetes is to provide better quality of care to patients, reduce the risk of complications, and improve their quality of life [9].

Defining a machine learning problem in the context of diabetes depends on specific goals and scenarios [10].

Classification of diabetes type: The objective may be to classify the type of diabetes based on the clinical and laboratory data of the patients. Type 1 and type 2 diabetes have different characteristics and require different treatment approaches.

Diagnosis of diabetes: Development of a model for diagnosing diabetes in individuals suspected of having the disease. This may include analysis of blood glucose levels, anthropometric data, and other features.

Predicting the risk of developing diabetes: The goal may be to predict the risk of developing diabetes in individuals who do not currently have the disease. This helps to early identify patients at high risk and offer them preventive measures.

Predicting diabetes progression: Creating models to predict the rate of diabetes progression in patients who already have the disease. This can help clinicians optimize treatment and monitoring.

Treatment optimization: The goal may be to develop models that help clinicians select the best treatments for each patient based on patient characteristics and response to medications.

Monitoring and preventing complications: The goal may be to create monitoring systems that prevent complications such as hypoglycemia, diabetic ketoacidosis, or diabetic retinopathy.

Health Data Integration: The goal may be to integrate data from multiple sources to provide physicians with a better understanding of patients' conditions and support decision-making.

Research into new treatments: The goal may be to analyze clinical trial data to develop new treatments for diabetes.

Building decision support systems: Machine learning models can serve as decision support systems for doctors and patients, providing information about risks and possible treatment options.

Assessing the effectiveness of treatment: The task may be to assess how effectively a treatment is working for a particular patient and adjust the treatment plan if necessary.

Machine learning tasks in the field of diabetes lead to more effective diagnostics, treatment, and care of patients, and promote research and development of new methods to combat this chronic disease.

2 Methods

Various machine learning methods can be used to solve problems in the field of diabetes diagnosis, management, and research. Here are some of the most common methods [10-11]:

1. Linear regression can be used for a forecasting task such as predicting blood glucose levels based on various parameters.

Linear regression is a method for predicting the value of a dependent variable based on a linear combination of one or more independent variables. Here is the algorithm for training a linear regression model:

Data loading:

First, it is necessary to load a dataset containing the values of the independent (factor) and dependent variables. Generally, data is presented in the form of a table or an array.

Data preparation:

Check the data for zero values and outliers.

Divide the data into a training set and a test set to evaluate the model's performance.

Model Definition:

Linear regression models the dependent variable (Y) as a linear combination of independent variables (X) with weights (coefficients) and a constant term (intercept).

The model can be represented by equation: $Y = b_0 + b_1X_1 + b_2X_2 + \cdots + b_n * X_n$ is the intercept, and b_0, b2, ..., bn, $a, b_1, b_2, \ldots, b_n$are the coefficients.

Model training:

To train a linear regression model, the Least Squares method is used. This method determines the values of coefficients that minimize the sum of squared differences between the predicted and actual values of the dependent variable.

Python libraries such as *Scikit-learn* can be used to train a linear regression model.

Forecasting:

Once the model is trained, we can apply it to predict the values of the dependent variable used on new data.

Model evaluation:

Evaluate the model's performance on a test dataset using metrics such as mean squared error (MSE) and coefficient of determination (R-squared).

Interpretation of results:

Analyze the model coefficients to determine the independent variables that have the greatest impact on the dependent variable.

Model deployment:

If the model meets our requirements, we can deploy it for use in real-world applications.

This is a general algorithm for linear regression. The implementation of specific steps may vary depending on the software and programming language used.

2. Model of naive Bayes classifier [1].

We load the *Diabetes* data and convert the problem into a binary classification based on the median value.

We divide the data into training and test sets and standardize the features (naive Bayes does not require standardization, but we do this for consistency with previous examples).

We create and train a naive Bayes classifier model.

We make a forecast on a test data set and calculate metrics.

We derive the confusion matrix, classification report, and AUC-ROC plot.

3. Support Vector Machines (SVM) [2]. SVM can be used for both classification and regression problems. It is especially useful in cases where the data is separated non-linearly.

The Support Vector Machine (SVM) algorithm is a machine learning technique used for classification and regression problems. Here is a general SVM algorithm for a binary classification problem:

Data loading:

First, load a dataset containing a set of objects (feature vectors) and the corresponding class labels. In a binary classification problem, there are two classes: positive (1) and negative (0).

Data preparation:

Check the data for zero values and perform preprocessing if necessary. Scale the features so they have the same range.

Kernel selection:

SVM can use different kernels such as linear, polynomial, or radial basis function (RBF) kernels. Choose the appropriate kernel for your problem.

Model training:

Train the SVM on the training dataset using the selected kernel. The main goal is to find the optimal separating hyperplane that maximizes the margins between classes.

Parameter optimization:

Tune SVM parameters such as regularization coefficient (C) and kernel parameters to achieve better model performance.

Forecasting:

Use the trained model to predict class labels for new data.

Model evaluation:

Evaluate model performance using metrics such as precision, recall, F1-score, and confusion matrix in case of classification.

Cross-validation:

Use cross-validation to more reliably evaluate model performance and avoid overfitting.

Interpretation of results:

Analyze your results and feature weights to understand which features contribute the most to your classification.

Regularization and tuning:

Depending on the results, you can regularize the model or tune other parameters to improve performance.

Model deployment:

If the model meets your requirements, you can deploy it for use in real-world applications.

This is a general SVM algorithm for the binary classification problem. For a regression problem, the principles of the algorithm are similar, but the goal is to predict continuous values rather than class labels.

4. Decision trees and random forests [12-14]. Decision trees can be used for classification and regression. Random forests are an ensemble of decision trees and can improve the predictive ability of a model.

We load the *Diabetes* data and convert the problem into a binary classification based on the median value.

We divide the data into training and test sets and standardize the features (standardization is not required, but we do it for consistency with previous examples).

We create and train a decision tree model.

We make a forecast on a test data set and calculate metrics.

We derive the confusion matrix, classification report, and AUC-ROC plot.

These are just a few examples of machine learning techniques that can be applied in the context of diabetes. The choice of a particular method depends on the nature of the problem and the available data. In addition, ensembles of methods are often used to improve the quality of forecasts [15-17].

3 Results

The results of machine learning in the field of diabetes can be varied and depend on the specific task, the methods used, and the data available. Here are some of the typical results that can be obtained using machine learning in the context of diabetes:

1. To solve a regression problem on the *Diabetes* dataset from the *Scikit-learn* library in *Python*, we can use the following algorithm:

Mean Squared Error: 2900.1732878832318

R-squared: 0.452606602161738

Mean Squared Error (MSE) and coefficient of determination (R-squared) are the two main metrics used to evaluate the performance of a linear regression model. The values of MSE and R-squared we use, tell us how well our model fits the data and predicts the dependent variable.

Mean Square Error (MSE):

MSE measures the standard deviation between the actual values of the dependent variable (in our case, probably blood glucose) and the predicted values calculated by our model.

Our MSE value is 2900.17. The lower the MSE, the better. This means that our model is, on average, wrong by the square root of this amount when predicting values.

Coefficient of determination (R-squared):

R-squared measures how well our model explains the variability in the dependent variable. It ranges from 0 to 1, where 0 means that the model does not explain the variability and 1 means that the model fits the data perfectly.

The R-squared value is 0.4526. This means that our model explains approximately 45% of the variability in the data. That is, the model explains less than half of the variation in the dependent variable, which could be improved by using more complex models or additional features.

Therefore, the model has average prediction quality (MSE is not too high) but it explains only part of the variability in the data (R-squared is not very close to 1). We may need to further analyze the data, add new features, or choose a different modeling method to improve the model's performance.

2. Naive Bayes classifier model

Confusion Matrix:

[[37 12]
[13 27]]

Element (0, 0) (upper left corner) represents the number of true negative (TN) examples.
Element (0, 1) (upper right corner) represents the number of false positive (FP) examples.
Element (1, 0) (lower left corner) represents the number of false negative (FN) examples.
Element (1, 1) (lower right corner) represents the number of true positive (TP) examples.

Classification Report:

	precision	recall	f1-score	support
Class 0	0.74	0.76	0.75	49
Class 1	0.69	0.68	0.68	40
accuracy			0.72	89
macro avg	0.72	0.72	0.72	89
weighted avg	0.72	0.72	0.72	89

Accuracy: 0.7191
Precision: 0.6923
Recall: 0.6750
F1-Score: 0.6835
AUC-ROC: 0.8260

Classification Report provides detailed metrics for each class (Class 0 and Class 1) and average values (macro avg and weighted avg). In our case:

Precision measures how many of the objects that the model predicts as positive are actually positive. Precision for Class 0 is 0.74 and for Class 1, it is 0.69.

Recall measures how many of all actual positive objects the model correctly classified. Recall for Class 0 is 0.76 and for Class 1, it is 0.68.

F1-Score is the harmonic mean between precision and recall. The F1-Score for Class 0 is 0.75 and for Class 1, it is 0.68.

Accuracy shows the proportion of correctly classified examples relative to the total number of examples. Accuracy is 0.7191, which means that the model correctly classified 71.91% of all examples.

AUC-ROC (Area under ROC Curve):

AUC-ROC measures the area under the ROC (Receiver Operating Characteristic) curve, which represents the model's performance at various classification thresholds. An AUC-ROC value close to 1 (in our case, the value is 0.8260) indicates a good ability of the model to separate classes.

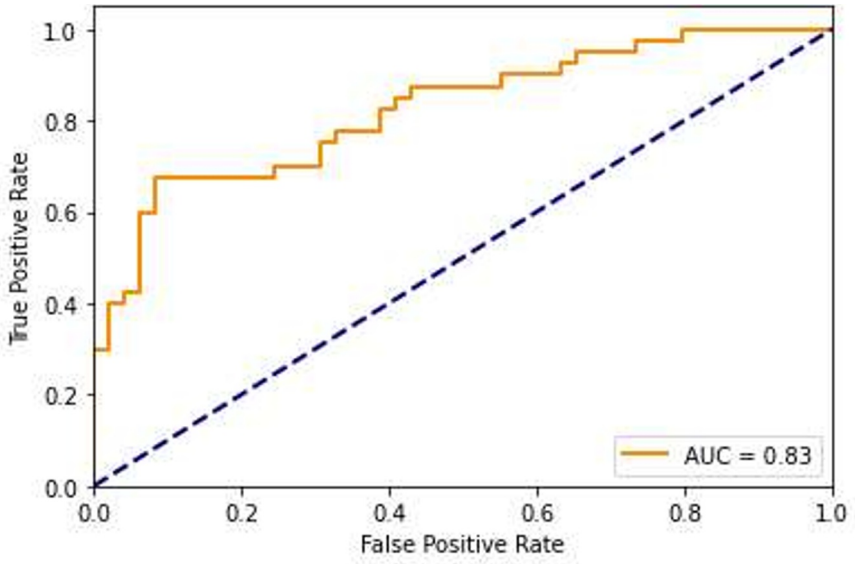

Fig. 1. ROC Curve.

These metrics evaluate the performance of a classification model, specifically, the model's ability to separate classes, the balance of precision and recall, and the overall accuracy of the model. In our case, the model showed average results in the classification evaluation.

3. SVM with linear kernel.

Load *Diabetes* data and convert the problem to a binary classification (based on the median value).

Divide data into training and test sets, and standardize features.

Create and train an SVM model with a linear kernel.

Make a forecast on a test data set and calculate confusion metrics, accuracy, precision, recall, F1-score, and AUC-ROC.

Plot an ROC curve to visualize model performance.

Confusion Matrix:

[[35 14]
[10 30]]

Classification Report:

	precision	recall	f1-score	support
Class 0	0.78	0.71	0.74	49
Class 1	0.68	0.75	0.71	40
accuracy			0.73	89
macro avg	0.73	0.73	0.73	89
weighted avg	0.73	0.73	0.73	89

Accuracy: 0.7303
Precision: 0.6818
Recall: 0.7500
F1-Score: 0.7143
AUC-ROC: 0.8398

The confusion matrix shows the number of correctly and incorrectly classified examples for each class.

Classification Report:

Classification Report provides detailed metrics for each class (Class 0 and Class 1) and average values (macro avg and weighted avg). In our case:

Precision measures how many of the objects that the model predicts as positive are actually positive. Precision for Class 0 is 0.78 and for Class 1, it is 0.68.

Recall measures how many of all actual positive objects the model correctly classified. Recall for Class 0 is 0.71, and for Class 1, it is 0.75.

F1-Score is the harmonic mean between precision and recall. The F1-Score for Class 0 is 0.74 and for Class 1, it is 0.71.

Accuracy shows the proportion of correctly classified examples relative to the total number of examples. Accuracy is 0.7303, which means that the model correctly classified 73.03% of all examples.

AUC-ROC (Area under ROC Curve):

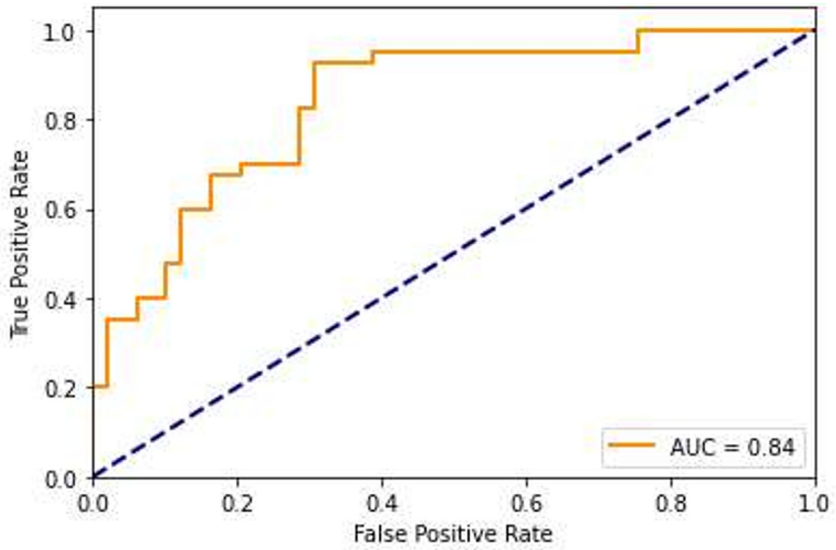

Fig. 2. ROC Curve.

AUC-ROC (Area under ROC Curve): ROC (Receiver Operating Characteristic) is a graph that displays the performance of a model at various classification thresholds. The area under the ROC curve (AUC-ROC) measures the overall performance of the model. An AUC-ROC value close to 1 (in our case, the value is 0.8398) indicates a good ability of the model to separate classes.

4. Decision tree results

Confusion Matrix:

[[35 14]
 [15 25]]

Classification Report:

	precision	recall	f1-score	support
Class 0	0.70	0.71	0.71	49
Class 1	0.64	0.62	0.63	40

accuracy			0.67	89
macro avg	0.67	0.67	0.67	89
weighted avg	0.67	0.67	0.67	89

Accuracy: 0.6742
Precision: 0.6410
Recall: 0.6250
F1-Score: 0.6329
AUC-ROC: 0.6696

The confusion matrix displays the number of correctly and incorrectly classified examples for each class.

Classification Report:

Classification Report provides detailed metrics for each class (Class 0 and Class 1) and average values (macro avg and weighted avg). In our case:

Precision measures how many of the objects that the model predicts as positive are actually positive. Precision for Class 0 is 0.70 and for Class 1, it is 0.64.

Recall measures how many of all actual positive objects the model correctly classified. Recall for Class 0 is 0.71, and for Class 1, it is 0.62.

F1-Score is the harmonic mean between precision and recall. The F1-Score for Class 0 is 0.71 and for Class 1, it is 0.63.

Accuracy shows the proportion of correctly classified examples relative to the total number of examples. Accuracy is 0.6742, which means that the model correctly classified 67.42% of all examples.

AUC-ROC (Area under ROC Curve):

AUC-ROC measures the area under the ROC (Receiver Operating Characteristic) curve, which represents the model's performance at various classification thresholds. An AUC-ROC value close to 1 (in our case, the value is 0.6696) indicates the ability of the model to separate classes, but it is not very close to 1, which may indicate a relatively low ability of the model.

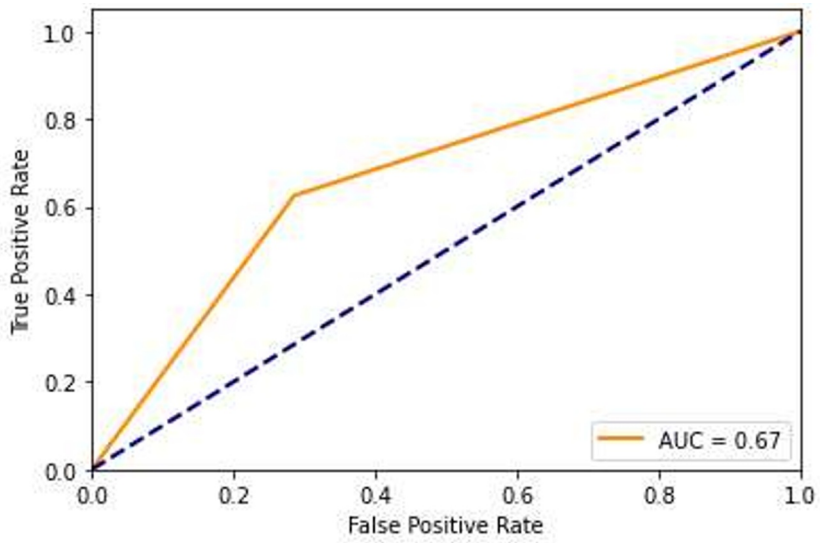

Fig. 3. ROC Curve.

These metrics evaluate the performance of a classification model, specifically, the model's ability to separate classes, the balance of precision and recall, and the overall accuracy of the model. The model showed average results in the classification evaluation.

4 Discussion

Machine learning can improve the accuracy of diabetes diagnosis and the classification of its type. It allows us to create individual treatment plans, taking into account the unique characteristics of each patient. Machine learning models can predict the likelihood of complications, allowing taking timely measures to prevent them. It promotes the integration of data from various sources, enriching patient information. Machine learning-based decision support systems help doctors and patients make decisions that are more informed.

The acquisition and analysis of medical data raises issues of privacy and security of patient data. Machine learning requires access to large, high-quality data, which can be a challenge in medical practice. Many machine learning models, especially neural networks, can be difficult to interpret, making it difficult to explain decisions to patients and doctors. Machine learning (to be applied correctly) requires specialized expertise and training for medical personnel.

It is important to develop rules and regulations governing the use of machine learning in medicine to ensure transparency and responsibility. Patient data must be protected and confidentiality standards must be respected. An important aspect of the discussion is how machines can work with doctors and help them make decisions rather than replace them. Training of medical staff and machine learning specialists is becoming increasingly important to ensure that the technology is used correctly.

5 Conclusion

It is important to properly balance the use of machine learning, taking into account ethical and practical aspects. Machine learning makes it possible to diagnose diabetes with high accuracy and identify the risks of developing the disease at early stages. This facilitates early initiation of treatment and prevention of complications.

Machine learning models can predict the likelihood of complications and warn doctors and patients. This facilitates more proactive monitoring and care. Machine learning is a powerful tool for diabetes management, and its applications will continue to evolve, creating new opportunities to improve patient care and research in this area. Technological advances must be combined with high standards of ethics and safety to ensure maximum benefit to patients and society.

References

1. G. A. Pethunachiyar, “Classification of diabetes patients using kernel based support vector machines”, in *2020 International Conference on Computer Communication Informatics (ICCCI)* (2020)
2. S. Gupta, H. K. Verma, D. Bhardwaj, “Classification of diabetes using naïve bayes and support vector machine as a technique’, in *Operations Management and Systems Engineering* (Springer, Singapore, 2021)
3. C. Rashka, *Python and machine learning* (DMK Press, 2017)
4. A. Khattak, A. Habib, M. Z. Asghar, F. Subhan, I. Razzak, A. Habib, Soft Comput. **25**, 2191–220 (2021)

5. M. Mohri, A. Rostamizadeh, A. Talwalkar, *Foundations of Machine Learning* (MIT Press, USA, Massachusetts, 2012)
6. R. Miotto, F. Wang, S. Wang, X. Jiang, J. T. Dudley, Brief Bioinform. **19**, 1236–46 (2018)
7. P. Flach, *Machine learning. The science and art of constructing algorithms that extract knowledge from data* (DMK Press, 2015)
8. L. Chen, D. J. Magliano, P. Z. Zimmet, Nat Rev Endocrinol **8**, 228-236 (2011)
9. D. S. Char, N. H. Shah, D. Magnus, New England Journal of Medicine **378(11)**, 981-983 (2018)
10. U. M. Butt, S. Letchmunan, M. Ali, F. H. Hassan, A. Baqir, H. H. R. Sherazi, J Healthcare Eng. **2021**, 9930985 (2021)
11. E. Alpaydin, *Introduction to Machine Learning* (The MIT Press, London, 2010)
12. The Checkup. Diabetes Statistics: Read the Facts (2020), https://www.singlecare.com/blog/news/diabetes-statistics/
13. Y. K. Qawqzeh, A. S. Bajahzar, M. Jemmali, M.M. Otoom, A.Thaljaoui, Biomed Res Int. **2020**, 3764653 (2020)
14. D. K. Choubey, M. Kumar, V. Shukla, S. Tripathi, V. K. Dhandhania, Curr Diabetes Rev. **16**, 833–50 (2020)
15. H. Ahmad, M. U. Asghar, M. Z. Asghar, A. Khan, A. H. Mosavi, IEEE Access **9**, 146214–32 (2021)
16. D. Sh. Ziyadullayev, D. T. Mukhamedieva, G. E. Ziyodullaeva, Z. J. Ibadullaeva, JARDCS **10(14)** (2018)
17. D. Sh. Ziyadullaev, D. T. Mukhamedieva, M. G. Teshaboyev, Sh. G'. To'ychiev, M. O. Kamolov, Yu. Sh. Bakhramova, G. E. Ziyodullaeva, E3S Web of Conferences **419**, 02004 (2023)
18. D. Alghazzawi, O. Bamasaq, H. Ullah, M. Z. Asghar, Appl Sci. **11**, 11634 (2021)
19. A. Mujumdar, V. Vaidehi, Proc Comput Sci. **165**, 292–9 (2019)
20. D. Ziyadullaev, D. Mukhamedieva, M. Teshaboyev, G. Ziyodullaeva, D. Abduraimov, BIO Web of Conferences **67**, 02009 (2023)
21. D.S.Ziyadullaeva, D.T.Mukhamedieva, G.E.Ziyodullaeva, Z.J.Ibadullaeva 2018 Develop the student model. Journal of Advanced Research in Dynamical and Control Systems – JARDCS Vol. 10(14) http://www.jardcs.org/backissues/archives-special.php?year=2018&issue=14.
22. D.S.Ziyadullaeva, D.T.Mukhamedieva, G.E.Ziyodullaeva 2018. Development of mathematical model of lesson schedule formation system. Journal of Advanced Research in Dynamical and Control Systems – JARDCS Vol. 10(14).– P. 1850 – 1854.
23. Z. Abdullaev, D.Sh. Ziyadullaev, D.T. Muhamediyeva. The task of assessing the risk in the operation of a complex free formal system. INFORMATION LETTER on holding an international conference “Efficiency of application of innovative technologies in agriculture and water management” HIRM-2021 IOP Publishing Journal of Physics: Conference Series 2176 (2022) 012071 doi:10.1088/1742-6596/2176/1/012071. https://iopscience.iop.org/article/10.1088/1742-6596/2176/1/012071/pdf