EVALUATION OF MACHINE LEARNING ALGORITHMS FOR GASTROENTEROLOGICAL DISEASES PREDICTION

[1]Yakhshiboyev E. Rustam, [2]Muminov B. Bahodir, [3]Eshmuradov E. Dilshod, [4]Kudratillaev B. Meyerbek

[1,4]Tashkent University of Information Technologies

[2] Head of the Department of Artificial Intelligence, Tashkent State University of Economics, Doctor of Technical Sciences, Professor

[3]Candidate of Technical Sciences, Tashkent University of Information Technologies

https://doi.org/10.5281/zenodo.8183162

Abstract. *This article presents a study and analysis of various artificial intelligence (AI) algorithms for their application in predicting gastroenterological diseases. Gastroenterological diseases pose a significant healthcare challenge, and early detection and accurate prognosis of these conditions can greatly improve treatment outcomes and impact patients' quality of life. The analysis of multiple AI algorithms is conducted in this work, with the aim of developing a hardware-software complex for predicting gastroenterological diseases. Particular attention is given to the k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), Artificial Neural Networks, and Random Forest algorithms applied to gastroenterological patient data.*

Various datasets from different sources and medical institutions were used for the research. The authors also discuss data preprocessing methods, such as normalization, feature selection, and outlier handling, to enhance the effectiveness of AI algorithms.

This work is a valuable study that advances the understanding of the applicability of AI algorithms in the field of gastroenterology. It may serve as a basis for further research and the development of innovative approaches to diagnosing and predicting gastroenterological diseases in modern medicine.

Keywords: *artificial intelligence, algorithm, prediction, gastroenterological diseases, hardware-software complex, k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), Artificial Neural Networks, Random Forest, initiation.*

I. INTRODUCTION

Currently, the development of artificial intelligence (AI) is progressing rapidly in all countries worldwide. In accordance with the "Digital Uzbekistan - 2030" strategy and with the aim of creating favorable conditions for the accelerated adoption of AI technologies and their widespread application in the country, ensuring accessibility and high-quality digital data, a presidential decree was issued on 26.08.2021, No. PP-5234, titled "Measures for the Implementation of a Special Regime for the Application of Artificial Intelligence Technologies" [1].

The special regime refers to the establishment of necessary organizational and legal conditions for legal entities and scientific organizations engaged in pilot projects involving AI, providing privileges in legal relationships arising during the testing and implementation of software products.

Artificial Intelligence is the science and technology of creating intelligent machines, especially intelligent computer programs. AI is associated with the similar task of using computers to understand human intelligence but is not necessarily limited to biologically plausible methods.

Preliminary diagnosis helps identify problems, determine enterprise weaknesses, develop a program for future changes, and must address questions such as whether identified problems can be solved and in what sequence they should be addressed. In the field of medicine, digital technologies can be widely used in diagnosing and treating various diseases and conditions of varying severity. With the aid of digital technologies, the work of medical personnel can be facilitated, while saving time on examinations and enhancing efficiency [2,3,4].

Within a short period, medical personnel can accurately establish a comprehensive diagnosis, where digital technologies can overcome controversial points. Digital technologies involve the use of artificial intelligence, neural networks, machine learning, and modern programming languages such as Python [8,9,10].

In this research, a significant object of study is human saliva. Saliva can predict the onset or detection of gastroenterological diseases. During the initiation period, the composition of saliva changes significantly. The composition of saliva serves as a parameter [20]. By altering the composition of saliva, a dataset for training AI algorithms can be created. Table No. 1 indicates the composition of a healthy individual [14,15,16].

Table №1.

Composition of Human Saliva.

№	***Composition of Saliva***	***COS (% and g/l):***
1	Water	99,4-99,5 %
2	Organic and inorganic components	0,5-0,6 %
3	Proteins	1,4-6,4 g/l
4	Mucin	0,8-6,0 g/l
5	Cholesterol	0,02-0,5 g/l
6	Glucose	0,1-0,3 g/l
7	Ammonium	0,01-0,12 g/l
8	Uric acid	0,005-0,03 g/l

In the article "Analysis of Algorithms for Predicting and Preliminary Diagnosis of Gastroenterological Diseases" [17, 18, 19], the authors utilized initial results based on data from 100 patients. In the subsequent investigation, the number of patients in the dataset was expanded to 1000. The parameters of the patients were obtained, and the dataset was trained using algorithms such as SVM and ANN based on these parameters.

II. MAIN PART

The K-Nearest Neighbors (KNN) algorithm is a type of supervised machine learning algorithm that can be used for both classification and regression prediction tasks.

The KNN algorithm operates on the principle of "feature similarity" and is particularly employed for predicting the values of new data points. In other words, a new data point is assigned a value based on how closely it aligns with the points in the training dataset.

KNN falls into the category of supervised learning algorithms, which means that there is a dataset with labeled training measurements (x, y), and the objective is to discover the relationship between x and y. The goal is to find a function h:X→Y, so that given an unknown observation x, h(x) can positively predict the corresponding output y (1,2).

For distance metrics, the Euclidean metric will be utilized:

$$d(x, x') = \sqrt{(x_1 + x_1')^2} + \cdots + \sqrt{(x_n + x_n')^2} \quad (1)$$

the input x is assigned to the class with the highest probability.

$$P_{(y=j|X=x)} = \frac{1}{K}\sum_{i\in A} I(y^i = j) \quad (2)$$

For regression, the method will be the same; instead of neighbor classes, we will take target values and find the target value for an unseen data point by taking the average, mean, or any suitable function [3].

Random Forest algorithm is a machine learning algorithm proposed by Leo Breiman and Adele Cutler. It involves using an ensemble of decision trees, also known as a committee, to make predictions.

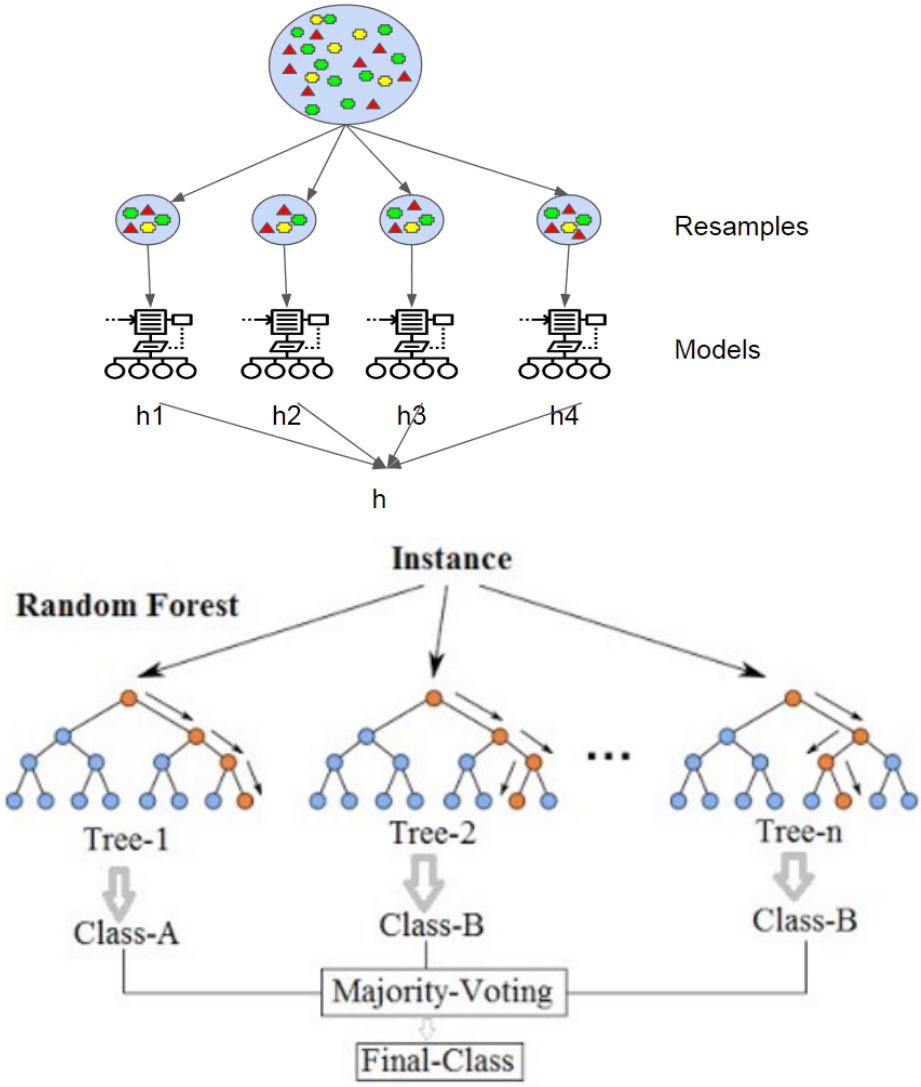

Fig 1. Algorithm K-Nearest Neighbors (KNN)

Random Forest performs sampling of rows and columns with decision trees as the basis. The models h1, h2, h3, h4 differ more from each other compared to using only bagging due to the sampling of columns.

As the number of base learners (k) increases, the variance decreases. When reducing k, the variance increases, while the bias remains constant throughout the process. k can be determined using cross-validation. [4]

Implementation in Scikit-learn. For each decision tree in Scikit-learn, the importance of nodes is calculated using the Gini importance, assuming only two child nodes (binary tree):

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

• ni sub(j) = importance of node j
• w sub(j) = weighted number of samples reaching node j
• C sub(j) = impurity value of node j
• left(j) = left child node split at node j
• right(j) = right child node split at node j

Then, the importance of each feature in the decision tree is calculated as:

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ nodes} ni_k}$$

• fi sub(i) = importance of feature i
• ni sub(j) = importance of node j

These values can then be normalized to values ranging from 0 to 1 by dividing by the sum of all feature importances:

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j}$$

The final feature importance at the Random Forest level is the average value across all trees. It is calculated by summing the feature importance values for each tree and dividing by the total number of trees:

$$RFfi_i = \frac{\sum_{j \in all\ trees} normfi_{ij}}{T}$$

- RFfi sub (i) = feature importance of object i computed from all trees in the Random Forest model
- normfi sub(ij) = normalized feature importance of object i in tree j
- T = total number of trees

Implementation in Spark:

For each decision tree in Spark, the feature importance of an object is computed by summing the gain coefficient scaled by the number of samples passing through the node:

$$fi_i = \sum_{j:nodes\ j\ splits\ on\ feature\ i} s_j C_j$$

- fi sub (i) = importance of feature i
- s sub (j) = number of samples reaching node j
- C sub(j) = impurity value of node j

To compute the final feature importance at the Random Forest level, the feature importance values for each tree are first normalized with respect to the tree:

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j}$$

- norm f_i sub(i) = normalized feature importance of object i
- f_i sub (i) = importance of feature i

$$RFfi_i = \frac{\sum_j normfi_{ij}}{\sum_{j \in all\ features, k \in all\ trees} normfi_{jk}}$$

Then, the importance values from each tree are summed and normalized:

- RFfi sub (i) = feature importance of object i computed from all trees in the Random Forest model
- Norm f_i sub(i_j) = normalized feature importance of object i in tree j

Support Vector Machines (SVM). Machine Learning Algorithm Support Vector Machines (SVM) are a set of supervised learning methods used for classification, regression, and outlier detection. These tasks are common in the field of machine learning. They can be used to detect cancer cells based on millions of images or predict future driving routes with a well-tuned regression model.

The key point to remember is that these methods are simply mathematical equations tuned to provide the most accurate answer as quickly as possible. SVMs differ from other classification algorithms in how they choose the decision boundary by maximizing the distance from the nearest data points of all classes. The decision boundary created by SVM is referred to as the maximum margin classifier or maximum hyperplane.

How does the algorithm work? The simple linear classifier of the Support Vector Machines (SVM) method operates by constructing a straight line between two classes. This means that all data points on one side of the line will belong to one category, while data points on the other side of the line will be assigned to another category. It implies that there are an infinite number of lines to choose from.

What makes the linear SVM algorithm superior to some other algorithms, such as the k-nearest neighbors method, is that it selects the best line for classifying your data points. It chooses the line that separates the data and is as far as possible from the nearest data points.

An example on a two-dimensional plane helps to grasp all the machine learning terminologies. Imagine you have some data points on a grid. You are attempting to separate these data points into categories, but you want to avoid misclassifying data. This means you are trying to find a line between two closest points that will divide the other data points.

Thus, two closest data points give you support vectors that you will use to find this line. This line is called the decision boundary.

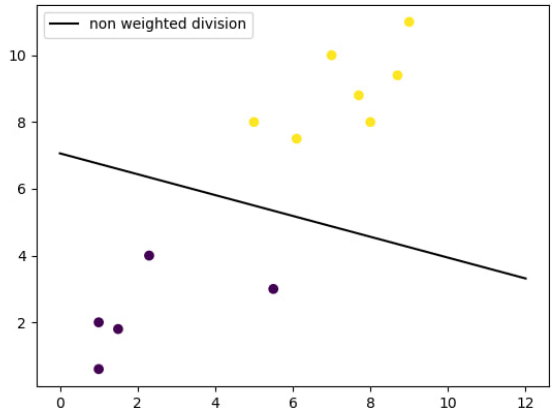

Fig 2. Linear SVM (Support Vector Machine)

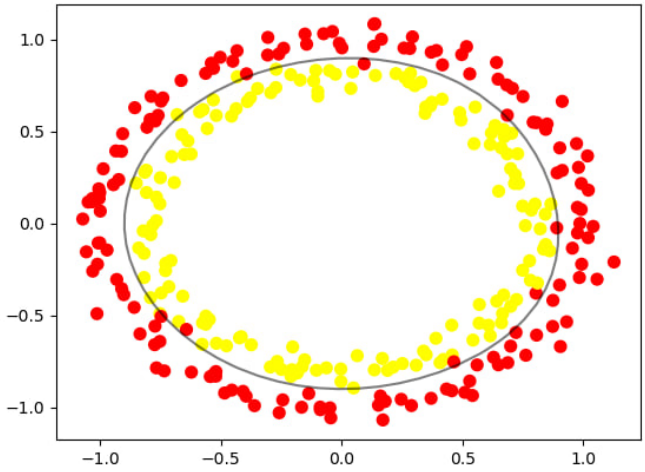

Fig 3. Nonlinear SVM using the RBF kernel.

The goal of the algorithm used in SVM: In other words, "The goal is to maximize the minimum distance." For distance, it is defined as:

$$d_{H(\varphi(x_0))} = \frac{|w^T(\varphi(x_0)) + b|}{||w||_2}$$

$$w^* = arg_w \max[min_n d_H(\varphi(x_n))]$$

Now that the goal is clear, let's consider the predictions for the training data, which have been separated into two groups: positive and negative. If we substitute a point from the positive group into the hyperplane equation, we get a value greater than zero from a mathematical perspective, like:

$$\mathbf{w}^T(\boldsymbol{\varphi}(\mathbf{x})) + \mathbf{b} > \mathbf{0}$$

And predictions from the negative group in the hyperplane equation would yield a negative value as:

$$\mathbf{w}^T(\boldsymbol{\Phi}(\mathbf{x})) + \mathbf{b} < \mathbf{0}.$$

However, these signs are related to the training data, i.e., how we train our model. It is for the positive class; give a positive sign, and for the negative class, give a negative sign.

But when testing this model on test data, if we correctly predict the positive class (positive sign or value greater than zero) as positive, then two positives will result in a positive outcome, which is greater than zero. The same applies if we correctly predict the negative class since two negatives will again lead to a positive outcome.

However, if the model's error classifies the positive group as negative, then one positive and one negative make a negative, thus overall less than zero.

Summarizing the above concept: The product of the predicted and actual label will be greater than 0 (zero) for correct predictions; otherwise, it will be less than 0.

$$y_n[w^T\varphi(x) + b] = \begin{cases} \geq 0 \; if \; correct \\ < 0 \; if \; incorrect \end{cases}$$

For perfectly separable datasets, the optimal hyperplane correctly classifies all points, and the optimal values are substituted into the weight equation.

arg max is a shorthand for arguments of maxima, which are essentially points in the function's domain where the function's values are maximized.

Additionally, extracting the independent term of the weight outwards gives:

$$w^* arg_w max \frac{1}{||w||_2}[min_n y_n |w^T(\varphi(x) + b)|]$$

The inner term (min_n y_n | w T φ(x) + b |) essentially represents the minimum distance from a point to the decision boundary and the nearest point to the decision boundary H.

Rescaling the distance to the nearest point to 1, i.e., (min_n y_n | w T φ(x) + b |) = 1. Here, the vectors remain in the same direction, and the hyperplane equation remains unchanged. This is similar to changing the scale of an image; objects expand or contract, but the directions remain the same, and the image remains unchanged.

Rescaling the distance is done by replacing:

$$w \to cw, \qquad b \to cb$$

$$(cw)^T \varphi(x_n) + (cb) = c(w^T \varphi(x_n) + b) = 0$$

Now, the equation becomes (describing that each point is at a distance of at least 1/||w||_2 from the hyperplane):

$$w^* = arg_w max \frac{1}{||w||_2}, s.t. min_n y_n [w^T \varphi(x_n) + b] = 1$$

This maximization problem is equivalent to the following minimization problem, which is scaled by a constant since they do not affect the results.

The artificial neural network algorithm is a mathematical model, a software or hardware embodiment, for the organization and construction of neural networks like a living organism. The artificial neural network is not programmed but learned. The learning process involves finding the coefficients of connections between neurons. The ability to learn is one of the advantages over algorithms.

The artificial neural network algorithm is also used for forecasting. This occurs after the training, meaning the neural network can predict future values in a sequence based on several previous values and the current value.

Forecasting happens when previous actual values override future ones. For instance, predicting various types of diseases.

- The learning process happens in two ways:
- Supervised learning
- Semi-supervised learning

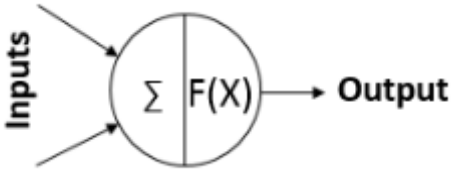

Fig 4. Simple Neuron

Activation of the incoming signal using the function F(X). The activation function can be sigmoid, ReLU, tanh, etc. In this example, the sigmoid activation function is used in the nodes of the layers.

$$F(X) = \frac{1}{1 + e^{-x}}$$

In Figure 4, a simple neuron is given. Now, the task will be solved through the function. The value in the hidden layer 1 = (1 * 0.1) + (1 * 0.1) + (1 * 0.1) = 0.3.

$$Y_{in} = \sum X_i * W1_{ij}$$

RESULTS

The dataset example is provided in Table No.2. It includes the parameters and the composition of human saliva. An analysis of K-Nearest Neighbors (KNN) and Random Forest algorithms was performed with 100 and 1000 patients, respectively. The corresponding results were obtained [9,10,11].

TABLE №2.

COMPOSITION OF HUMAN SALIVA

Parameters of the Dataset	Name of Saliva Composition
Parameter_1	Proteins
Parameter_2	Mucin
Parameter_3	Cholesterol
Parameter_4	Glucose
Parameter_5	Ammonium
Parameter_6	Uric acid

The first training process was conducted using 100 patients in the dataset. (Figures 5,6)

	Patient	Parameter_1	Parameter_2	Parameter_3	Parameter_4	Parameter_5	Parameter_6
0	1	1.4	0.8	0.02	0.10	0.01	0.005
1	1	1.5	0.9	0.03	0.11	0.02	0.006
2	1	1.6	1.0	0.04	0.12	0.03	0.007
3	1	1.7	1.1	0.05	0.13	0.04	0.008
4	1	1.8	1.2	0.06	0.14	0.05	0.009
...	...	...	...	...	...	...	...
94	3	10.9	10.2	0.97	1.04	0.95	0.099
95	3	11.0	10.3	0.98	1.05	0.96	0.100
96	3	11.1	10.4	0.99	1.06	0.97	0.101
97	3	11.2	10.5	1.00	1.07	0.98	0.102
98	3	11.3	10.6	1.01	1.08	0.99	0.103

99 rows × 7 columns

Fig 5. Dataset of 100 Patients

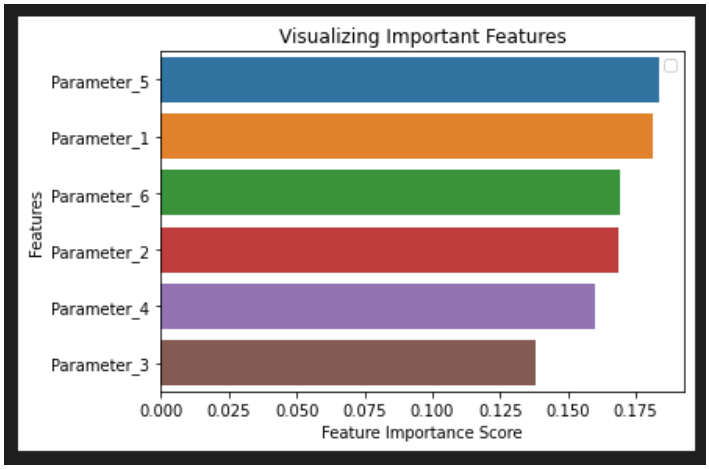

Fig 6. Importance of Parameters from Dataset (100 Patients)

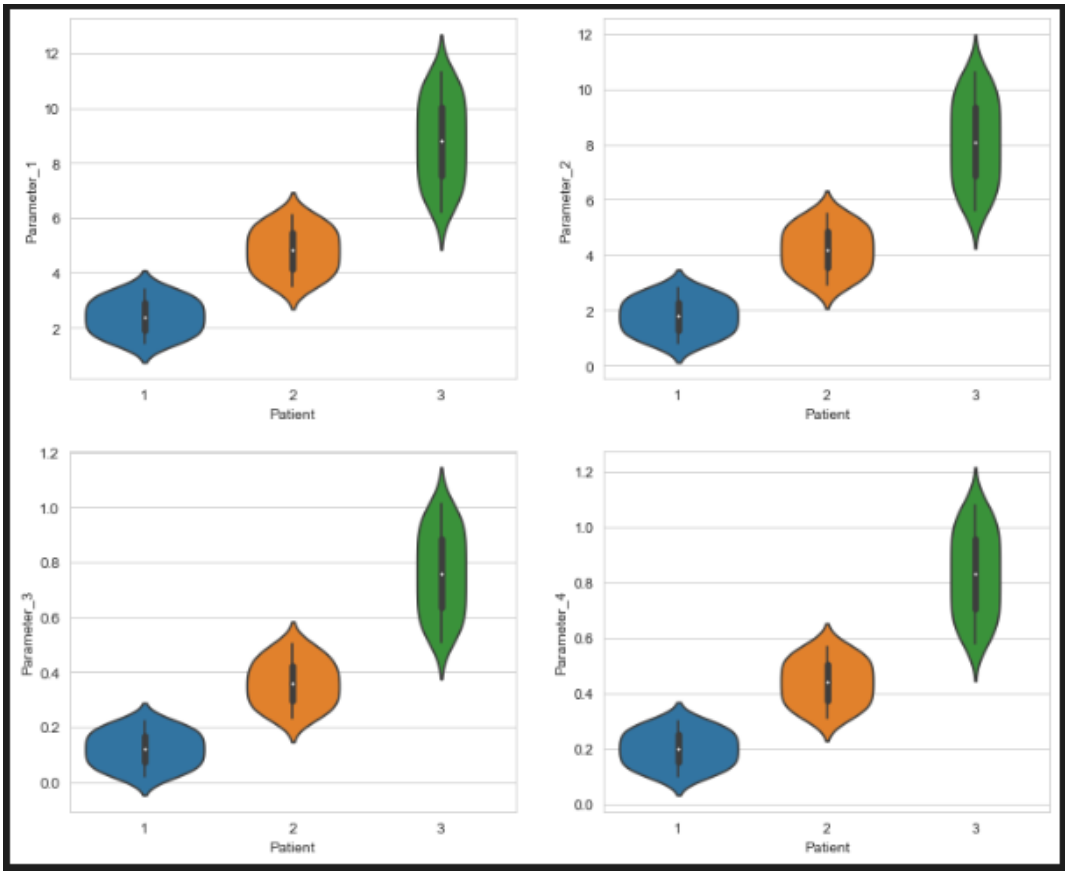

Fig 7. Determination of parameter importance and prediction of patient disease probability using K-Nearest Neighbors (KNN).

In this algorithm, the results are represented using three colors. Accordingly, the results of the training can be determined based on these colors:

• Purple - highest probability of disease
• Green - lower probability of disease
• Pistachio - healthy

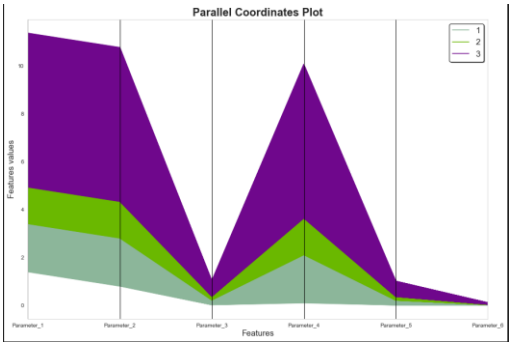

Fig 8. Determination of Parameter Importance and Prediction of Patient Disease Probability using Random Forest

In this algorithm, the results are represented using three colors. Accordingly, the results of the training can be determined based on these colors:

- Purple - highest probability of disease
- Green - lower probability of disease
- Pistachio - healthy

The first training process was conducted using 1000 patients in the dataset. (Figures 9,10)

	Patient	Parameter_1	Parameter_2	Parameter_3	Parameter_4	Parameter_5	Parameter_6
0	1	1.40	0.80	0.020	0.10	0.010	0.0050
1	1	1.41	0.81	0.021	0.11	0.011	0.0051
2	1	1.42	0.82	0.022	0.12	0.012	0.0052
3	1	1.43	0.83	0.023	0.13	0.013	0.0053
4	1	1.44	0.84	0.024	0.14	0.014	0.0054
...	...	...	...	...	...	...	...
994	3	11.34	10.74	1.014	10.04	1.004	0.1044
995	3	11.35	10.75	1.015	10.05	1.005	0.1045
996	3	11.36	10.76	1.016	10.06	1.006	0.1046
997	3	11.37	10.77	1.017	10.07	1.007	0.1047
998	3	11.38	10.78	1.018	10.08	1.008	0.1048

Fig 9. Dataset of 1000 Patients

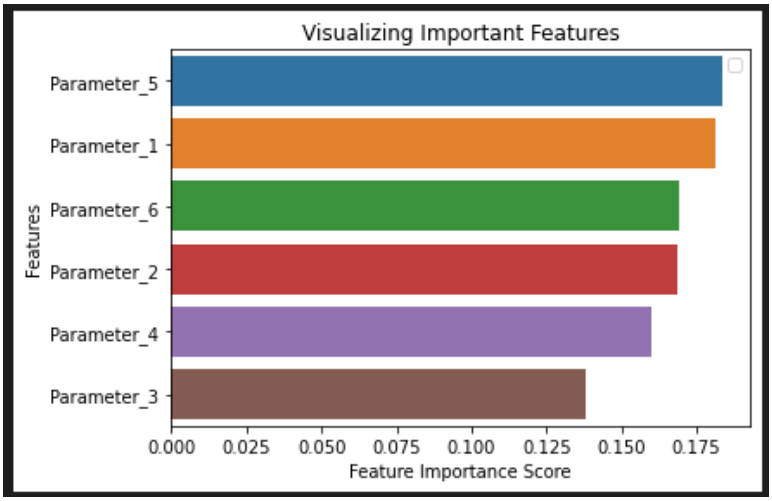

Fig 10. Importance of Parameters from Dataset (1000 Patients)

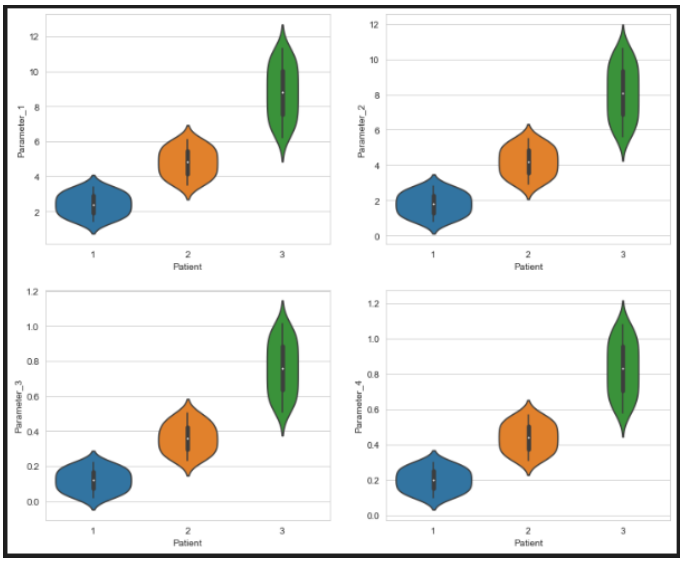

Fig 11. Determination of parameter importance and prediction of patient disease probability using K-Nearest Neighbors (KNN).

In this algorithm, the results are represented using three colors. Accordingly, the results of the training can be determined based on these colors:

- Purple - highest probability of disease
- Green - lower probability of disease
- Pistachio - healthy

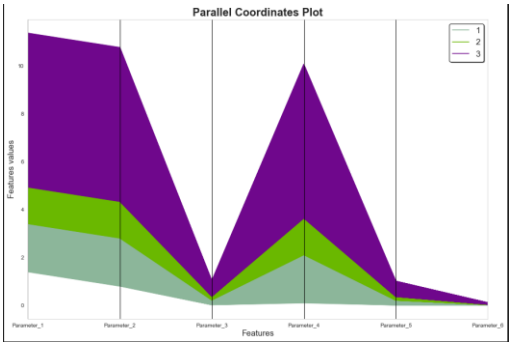

Fig 12. Determination of Parameter Importance and Prediction of Patient Disease Probability using Random Forest

In this algorithm, the results are represented using three colors. Accordingly, the results of the training can be determined based on these colors:

- Purple - highest probability of disease
- Green - lower probability of disease
- Pistachio - healthy

ACKNOWLEDGMENTS

This work was supported by the Tashkent Medical Academy and the Department of Biomedical Engineering, Informatics, and Biophysics.

CONCLUSION

As a result, a 96% accuracy was achieved in training the KNN and Random Forest algorithms. In further research, the patient dataset will be examined using SVM and ANN algorithms. The number of patients and parameters will gradually increase, and each time, they will be tested for accuracy.

REFERENCES

1. http://lex.uz//docs/5297051 О мерах по созданию условий для ускоренного внедрения технологий искусственного интеллекта
2. Балашова, А. Фейки и роботы: какими будут главные технологические тренды 2019.
3. А. Балашова, А. Посыпкина, Е. Баленко // РБК. – 2018. – 3 дек.
4. Christopher Bishop, Pattern Recognition and Machine Learning, 2006
5. Leon Stenneth, Philip S. Yu, Monitoring and mining GPS traces in transit space, SIAM International Conference on Data Mining
6. Ganesh J., Gupta M., Varma V.Interpretation of Semantic Tweet Representations // arXivpreprint arXiv:1704.00898. — 2017.
7. Zhang A.,Culbertson B.,Paritosh P.Characterizing Online Discussion Using CoarseDiscourse Sequences // Proceedings of the International AAAI Conference on Web andSocial Media. — 2017
8. Hastie, T., Tibshirani R., Friedman J. Chapter 15. Random Forests // The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — 2nd ed. — Springer-Verlag, 2009. — 746 p.

9. Eshmuradov D., Ismailov O., Magrupova M. METHODS AND MEANS OF DIGITAL PROCESSING OF BIOELECTRIC SIGNALS //Science and innovation. – 2023. – T. 2. – №. A2. – C. 84-88.
10. Эшмурадов Д. Э., Магрупова М. Т., Неъматова Д. Х. ОПТИМАЛЬНЫЕ МЕТОДЫ ЦИФРОВОЙ ОБРАБОТКИ БИОЭЛЕКТРИЧЕСКИХ СИГНАЛОВ //Теория и практика современной науки. – 2023. – №. 1 (91). – С. 276-282.
11. Raxmonov E. S. et al. IRON DEFICIENCY ANEMIA MED ANDROID APP OPERATING TECHNOLOGY //CENTRAL ASIAN JOURNAL OF EDUCATION AND COMPUTER SCIENCES (CAJECS). – 2023. – T. 2. – №. 2. – C. 25-28.
12. Кудратиллаев М., Яхшибоев Р. ТЕЛЕМЕДИЦИНА–НОВОЕ НАПРАВЛЕНИЕ СОВРЕМЕННОЙ МЕДИЦИНЫ //Innovations in Technology and Science Education. – 2023. – T. 2. – №. 9. – C. 222-238.
13. Yaxshiboyev R. Development of a software and hardware complex for primary diagnostics based on deep machine learning //Central asian journal of education and computer sciences (CAJECS). – 2022. – T. 1. – №. 4. – C. 20-24.
14. Yakhshiboyev R. DEVELOPMENT OF A “SALIVA” HARDWARE-SOFTWARE COMPLEX MODULES FOR THE PRIMARY DIAGNOSIS OF GASTROINTESTINAL DISEASES //Science and innovation. – 2023. – T. 2. – №. A2. – C. 27-34.
15. Kudratillaev M. B., Yakhshiboev R. E. ANALYSIS OF INNOVATIVE EQUIPMENT FOR THE DIAGNOSIS OF GASTROENTEROLOGICAL DISEASES //Open Access Repository. – 2023. – T. 4. – №. 03. – C. 13-23.