Application of ensemble machine learning methods for diabetes diagnosis

Davron Ziyadullaev[1,*], *Dilnoz* Muhamediyeva[1], *Komil* Madazimov[2], *Madamin* Madazimov[2], *Pulat* Temirov[2], and *Dilmurod* Abdukadirov[2]

[1]National Research University "Tashkent Institute of Irrigation and Agricultural Mechanization Engineers institute", 100000 Tashkent, Uzbekistan
[2]Andijan State Medical Institute, 170100 Andijan, Uzbekistan

Abstract. Ensemble machine learning techniques provide a powerful tool for improving the diagnostic accuracy of diabetes mellitus, one of the most common chronic diseases. The use of ensemble methods such as Random Forest, Gradient Boosting and Bagging for diagnosing diabetes mellitus are considered in the article and their advantages and challenges are analyzed. Ensemble methods help to increase diagnostic accuracy and reduce false positives and false negatives. They allow us to operate with heterogeneous data, provide resistance to overfitting, and give information about the importance of features. Overall, ensemble techniques of machine learning represent a promising tool for improving diabetes diagnosis and may contribute to more effective detection and management of this chronic disease. Further research and development in this area may lead to more accurate and reliable methods for diagnosing and treating diabetes.

1 Introduction

The relevance of machine learning in the field of diabetes diagnosis and management cannot be overestimated. Machine learning makes it possible to create models that can analyze a set of patient medical data and predict the risk of developing diabetes. This allows doctors and patients to take measures for early diagnosis and prevention of the disease and allows them to analyze the characteristics and response of each patient to treatment. This helps develop personalized diabetes treatment and management plans, which improve the effectiveness of therapy. Monitoring systems for glucose and other biometric parameters are becoming increasingly available and can help analyze this data in real time and warn of potential problems or the need for treatment adjustments. Machine learning facilitates the integration of data from various sources, such as medical records, laboratory data, medication, and monitoring information, allowing doctors and researchers to gain a more complete understanding of patients' conditions. Automated systems and machine learning tools help doctors make decisions that are more informed and provide more accurate and personalized care for patients with diabetes. All these aspects highlight the importance of machine learning in the field of diabetes, and make it a relevant and promising tool for medical practice and research [1].

[*] Corresponding author: dziyadullaev@inbox.ru

Machine learning makes it possible to develop classification models that can predict whether a patient has diabetes or not and, if necessary, classify the type of diabetes (type 1 or type 2). This is done using datasets with clinical and laboratory indicators such as blood glucose levels, body mass index (BMI), blood pressure and other risk factors. Machine learning models can help in the early diagnosis of diabetes and in assessing the risk of developing diabetes in individuals with a predisposition to the disease. They can analyze large arrays of patient data and alert doctors to the possibility of diabetes, allowing timely treatment. Several machine learning datasets are used for the task of diagnosing diabetes and predicting its progression. Here are some of the most known datasets [2]:

Pima Indians Diabetes Database: This dataset contains information about patients from Pima Indian tribes, which includes blood glucose, blood pressure, BMI, and others. It is often used for diabetes classification tasks.

UCI Diabetes Dataset: The UCI Diabetes Dataset contains patient information including 8 features such as age, BMI, glucose level and others. This dataset is also widely used in machine learning for diabetes classification tasks.

Diabetes dataset (Scikit-Learn): This is a diabetes dataset included in the *Scikit-learn* library. It contains information about patients and their symptoms such as glucose levels, blood pressure and others.

Indian Diabetes Dataset: This dataset contains information about patients from India and includes features such as age, gender, number of pregnancies, and others.

Debrecen Diabetes Retinopathy Dataset: This data is focused on the diagnosis of diabetic retinopathy, assessing the extent of its development. It includes images of the patient's retina.

These datasets provide different types of information about patients and their conditions and can be used to train machine learning models to diagnose diabetes and predict its progression.

The *Diabetes* dataset is one of the most common and widely used datasets for machine learning and statistics tasks. This dataset provides information about patients with diabetes and their health characteristics. It contains information about 442 patients. Each patient is described by one record in the dataset. This dataset contains 10 numeric features that characterize the conditions of patients. The most important features are:

Blood sugar level.

Blood pressure.

Body mass index (BMI).

Serum levels of a heart disease marker.

These features are important for diagnosing and managing diabetes.

The goal of applying ensemble methods in machine learning to diabetes diagnosis is to improve the diabetes diagnosis process by increasing the accuracy, reliability, and efficiency of detecting this chronic disease [3].

Machine learning tasks in the context of diabetes can be varied and cover different aspects of diagnosis, management, and research of this disease. Here are some of the key problems that machine learning can solve in the field of diabetes [4-7]:

Classification of Diabetes: One of the main challenges is to classify the type of diabetes (type 1 or type 2) or different stages of diabetes. Machine learning models can analyze clinical and laboratory data obtained from patients to make correct classifications.

Early diagnosis: Creating models for early diagnosis of diabetes and predicting the risk of its development. This allows treatment to begin early, when it is most effective.

Predicting progression: The goal may be to predict the rate of diabetes progression and determine which patients are prone to more rapid deterioration.

Treatment optimization: Machine learning can help doctors optimize treatment plans for each patient based on patient characteristics and his/her response to medications.

Monitoring and prevention of complications: The goal may be to establish monitoring systems to prevent complications such as hypoglycemia, diabetic ketoacidosis and diabetic retinopathy.

Health data integration: Machine learning can help integrate data from various sources so that doctors and researchers can gain a better understanding of patients' conditions.

Researching new treatments: Machine learning can be used to analyze clinical trial data and develop new procedures for diabetes treatments.

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: Machine learning can help assess how effectively a treatment is working for a particular patient and, if necessary, adjust the treatment plan.

Improving the quality of care: The main goal is to improve the quality of care for patients with diabetes, reduce the risk of complications and improve their quality of life.

These diabetes machine learning tasks lead to more effective diagnosis, treatment, and care for patients, and advance research and development of new methods to combat this chronic disease.

2 Methods

Application of ensemble methods in machine learning for diabetes diagnosis can improve the accuracy and reliability of models. Ensemble methods combine several base models to reduce overfitting, increase generalization ability, and improve the quality of predictions [8].

1. *Random Forest* is a powerful ensemble method in machine learning used for classification and regression problems. It is based on the concept of creating multiple decision trees and combining their results to improve the accuracy and robustness of the model [9].

Random forest consists of many decision trees that are trained on different subsets of data. Each tree can be a weak model, but their combination creates a strong model. When constructing each tree, the random forest randomly selects a subset of features to split the data. This helps reduce the correlation between trees and increase model diversity. To train each tree, a bootstrap sample (sample with repetition) from the original data is used. This allows for the diversity of data to be taken into account. After training all the trees, a random forest aggregates their predictions. In a classification problem, majority voting is most often used, and in a regression problem, averaging of predictions is used.

Random forest often demonstrates high classification and regression accuracy. Thanks to the random selection of features and the use of bootstrap sampling, the random forest is resistant to overfitting. We can use random forests to process data with a large number of features. Random forest allows us to evaluate the importance of each feature for a task. The method is applicable to a variety of data types.

2. *Gradient boosting* is an ensemble in which each tree is tuned to the errors of the previous tree. This improves the quality of predictions and allows us to work with different types of data. Examples include *Gradient Boosting Trees* and libraries such as *XGBoost, LightGBM,* and *CatBoost*. Gradient Boosting is a powerful ensemble method in machine learning also used for classification and regression problems. Unlike bagging, where base models are trained independently, gradient boosting builds an ensemble of models sequentially, with each new model tuning to the errors of the previous model. Gradient boosting builds an ensemble of models in a sequence. The new model is trained to correct the errors of the previous model. The gradient descent method is used to tune model parameters. It allows us to optimize the loss function. Weak models, such as decision trees of small depth, are often used as base models in gradient boosting. This makes the method

more resistant to overfitting. Gradient boosting pays special attention to the errors of the previous model, which is used in training the next model. This allows the creation of models that focus on complex examples [10].

Gradient boosting usually provides high prediction accuracy. Iterative training and attention to the errors of previous models make gradient boosting resistant to overfitting. Gradient boosting can be used with a variety of data types, including numeric and categorical features. It allows us to evaluate the importance of each feature for the task. Gradient boosting has found applications in various areas of machine learning, including data mining competitions.

3. *Bagging* is a technique in which multiple models are trained independently on different subsets of data and then their predictions are averaged. Bagging (*Bootstrap Aggregating*) is an ensemble method in machine learning, in which multiple models (such as decision trees or any other underlying models) are trained independently on different subsets of data, and then their predictions are averaged or combined. This method produces an ensemble of models that is often more stable and accurate than individual models. When using bagging, each model is trained on a random sample (with repetitions) from the original data. This means that some examples may be included in the sample more than once, while others may not be included at all. After all models are trained, their predictions are averaged to produce the final prediction. In a classification problem, majority voting is most often used, and in a regression problem, averaging of predictions is used. Bagging reduces overfitting because each model is trained on a different subset of data, which contributes to model robustness [11].

By averaging the predictions of multiple models, bagging reduces error variance. Bagging helps create more stable and generalizable models. It is applicable to different base models: Bagging can be used with different types of base models. Since the data is randomly selected, bagging is less sensitive to outliers in the data. Models can be trained independently, allowing for parallel computing.

Examples of bagging include *Random Forest* and *Bagged Decision Trees*. These methods are widely used in various fields of machine learning and data analytics to improve the quality and reliability of models.

4. *Adaptive Boosting* or *AdaBoost* is a machine learning algorithm used to improve the performance of classification models. It works by combining multiple weak classifiers into one strong model. AdaBoost uses weak classifiers, which may be models that are not powerful enough to solve the classification problem on their own. Each sample in the training set is assigned a weight, and it is equal at the start of training. The algorithm focuses on samples that are misclassified by previous models. AdaBoost adapts sample weights at each training stage. Classification errors increase the weights of misclassified samples, allowing subsequent models to focus on those samples. It combines the predictions of all weak models, weighting them based on their performance. This creates a strong model that can be more accurate than each of the weak models can individually. The AdaBoost algorithm is executed in several iterations (stages), and at each stage, a new weak model is added that focuses on the errors made by previous models [12].

5. Model mixing *(Model Stacking).* Model Stacking is a technique in which multiple models are combined and their predictions become input to the final model. This can be useful for combining different types of models.

Model mixing, also known as "*Model Stacking*" or "*Ensemble Stacking*", is an ensemble technique in machine learning in which the results of several different models are combined to improve overall performance. This method is widely used to solve complex classification and regression problems [13-14].

First, several different underlying models (or classifiers) are created using different algorithms or parameters. Each underlying model is trained on the same training dataset.

They can use different features or parameters, which increase the variety of models. An additional model (called a "stacking model" or "meta-model") is developed that takes the predictions of the underlying models as input features. The meta-model is trained on the outputs of the underlying models, predicting the final result. This allows the meta-model to combine the predictions of the underlying models and take into account their strengths and weaknesses.

Once the meta-model is trained, it can be used for classification or regression on new data. The benefits of the stacking model include increased generalization power, improved performance, and high robustness of predictions. However, staking requires more complex setup and management and may require more computing resources.

Examples of stacking models include using different machine learning algorithms such as decision trees, nearest neighbors, logistic regression, and even neural networks as underlying models and then using a stacking model such as Gradient Boosting or Random Forest to combine their results.

3 Results

To apply ensemble methods to the diagnosis of diabetes, we need to follow the following steps:

Collect and prepare data: Obtain reliable and clean patient records, including features such as age, gender, BMI (body mass index), blood glucose, blood pressure, and others. Split the data into training and test sets.

Select an ensemble method: Decide which ensemble method is best suited for the diabetes classification task.

Train Ensemble: Train the selected ensemble on the training dataset. Fine-tuning settings can improve performance.

Evaluate performance: Use metrics such as accuracy, precision, recall, F1-score and AUC-ROC to evaluate the model's performance on the test dataset.

Validation and Tuning: Perform cross-validation and parameter tuning to ensure that the model performs well and does not overfit.

Application of the model: Use the trained model to diagnose diabetes in new patients.

1. According to the developed program using the Random Forest algorithm, the following results were obtained:

```
Confusion Matrix:
[[37 12]
 [13 27]]
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: 0.8125

The metrics presented in the classification and confusion matrix report evaluate the performance of a machine learning model on a classification task where there are two classes: "Class 0" and "Class 1". Let us consider each of these metrics:

Confusion Matrix:

The first row of the confusion matrix represents the true values of Class 0, and the second row represents the true values of Class 1.

The first column of the matrix shows how many objects were predicted to be class 0, and how many of them actually belong to class 0.

The second column shows how many objects were predicted to be class 0 but actually belong to class 1.

The third column shows how many objects were predicted to be class 1, and how many of them actually belong to class 0.

The fourth column shows how many objects were predicted to be class 1, and how many of them actually belong to class 1.

Accuracy:

Accuracy estimates the proportion of correctly classified objects relative to all objects. In this case, the accuracy is 0.7191, which means that the model correctly classified 71.91% of the objects.

Precision:

Precision measures the proportion of true positive predictions relative to all positive predictions. In this case, the accuracy for "Class 0" is 0.74, and for "Class 1", it is 0.69.

Recall:

Recall measures the proportion of true positive predictions relative to all true positive objects. In this case, the completeness for "Class 0" is 0.76, and for "Class 1", it is 0.68.

F1-Score:

The F1-score is the harmonic mean between precision and recall. This is a numerical estimate of the balance between precision and recall. In this case, the F1-score for "Class 0" is 0.75, and for "Class 1", it is 0.68.

AUC-ROC (Area under the ROC Curve):

AUC-ROC is a model quality measure that evaluates the ability of a model to split classes and is dependent on a probability threshold. A value of 0.8125 means that the model splits the classes well, and the AUC-ROC close to 1 indicates that the model's performance is good.

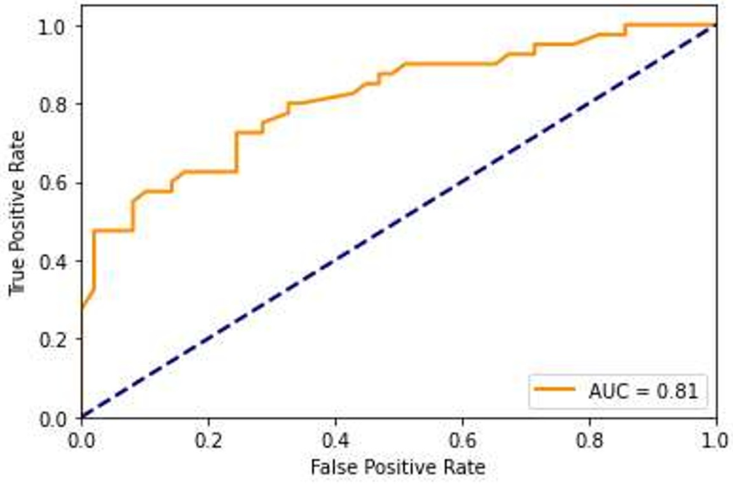

Fig. 1. ROC Curve.

In general, metrics evaluate the performance of a model in a classification task. Precision, recall, and F1-score evaluate the quality of classification for each class, while AUC-ROC provides an overall assessment of the quality of class splitting.

2. The following results were obtained with the developed program using the Gradient Boosting algorithm:

Confusion Matrix:
[[37 13]
 [14 25]]
Classification Report:

	precision	recall	f1-score	support
0	0.73	0.74	0.73	50
1	0.66	0.64	0.65	39
accuracy			0.70	89
macro avg	0.69	0.69	0.69	89
weighted avg	0.70	0.70	0.70	89

Accuracy: 0.6966292134831461
Precision: 0.6578947368421053
Recall: 0.6410256410256411
F1-Score: 0.6493506493506495
AUC-ROC: 0.7605128205128205

The metrics presented in the classification and confusion matrix report evaluate the performance of a machine learning model on a binary classification task. In this case, we have two classes: "0" and "1". Let us figure out what these metrics mean:

Confusion Matrix:

The first row of the confusion matrix represents the true values of class "0", and the second row represents the true values of class "1".

The first column of the matrix shows how many objects were predicted to be class "0" and how many of them actually belong to class "0".

The second column shows how many objects were predicted to be class "0" but actually belong to class "1".

The third column shows how many objects were predicted to be class "1" and how many of them actually belong to class "0".

The fourth column shows how many objects were predicted to be class "1" and how many of them actually belong to class "1".

Accuracy:

Accuracy estimates the proportion of correctly classified objects relative to all objects. In this case, the accuracy is 0.6966, which means that the model correctly classified 69.66% of the objects.

Precision:

Precision measures the proportion of true positive predictions relative to all positive predictions. In this case, the accuracy for class "0" is 0.73, and for class "1", it is 0.66.

Recall:

Recall measures the proportion of true positive predictions relative to all true positive objects. In this case, the recall for class "0" is 0.74, and for class "1", it is 0.64.

F1-Score:

The F1-score is the harmonic mean between precision and recall. This is a numerical estimate of the balance between precision and recall. In this case, the F1-score for class "0" is 0.73, and for class "1", it is 0.65.

AUC-ROC (Area under the ROC Curve):

AUC-ROC is a model quality measure that evaluates the ability of a model to split classes and is dependent on a probability threshold. A value of 0.7605 means that the model splits the classes well, and the AUC-ROC close to 1 indicates that the model performs well.

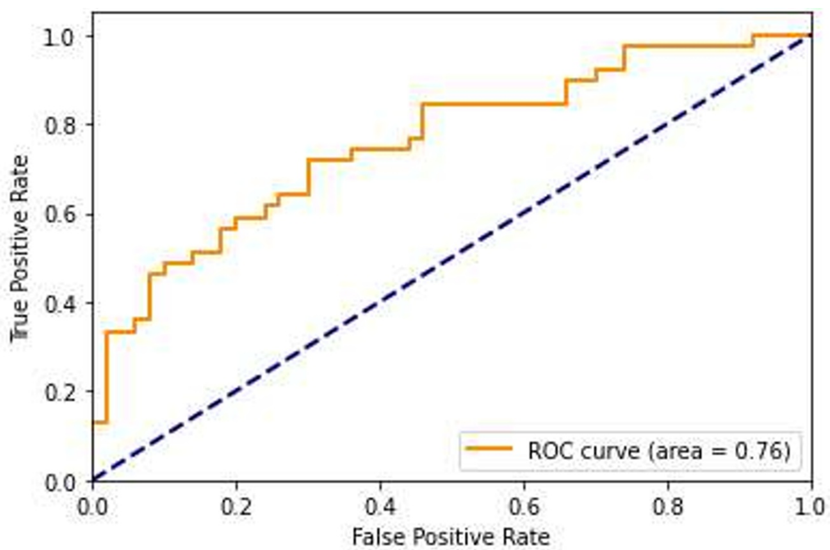

Fig. 2. Receiver Operating Characteristic.

3. Bagging result:

In this program, we use the bagging method with a base classifier, which is a Decision Tree Classifier. We split the data into training and test sets, train the bagging model and evaluate its performance, compute the confusion matrix, and present a classification report and five basic classification metrics.

Confusion Matrix:
[[39 11]
 [15 24]]

Classification Report:

	precision	recall	f1-score	support
0	0.72	0.78	0.75	50
1	0.69	0.62	0.65	39
accuracy			0.71	89
macro avg	0.70	0.70	0.70	89
weighted avg	0.71	0.71	0.71	89

Accuracy: 0.7078651685393258
Precision: 0.6857142857142857
Recall: 0.6153846153846154
F1-Score: 0.6486486486486486
AUC-ROC: 0.7892307692307693

Accuracy estimates the proportion of correctly classified objects relative to all objects. In this case, the accuracy is 0.7079, which means that the model correctly classified 70.79% of the objects.

Precision measures the proportion of true positive predictions relative to all positive predictions. In this case, the accuracy for class "0" is 0.72, and for class "1", it is 0.69.

Recall measures the proportion of true positive predictions relative to all true positive objects. In this case, the recall for class "0" is 0.78, and for class "1", it is 0.62.

The F1-score is the harmonic mean between precision and recall. This is a numerical estimate of the balance between precision and recall. In this case, the F1-score for class "0" is 0.75, and for class "1", it is 0.65.

AUC-ROC is a model quality measure that evaluates the ability of a model to split classes and is dependent on a probability threshold. The value of 0.7892 means that the model splits the classes well and the AUC-ROC close to 1 indicates that the classification model has good performance.

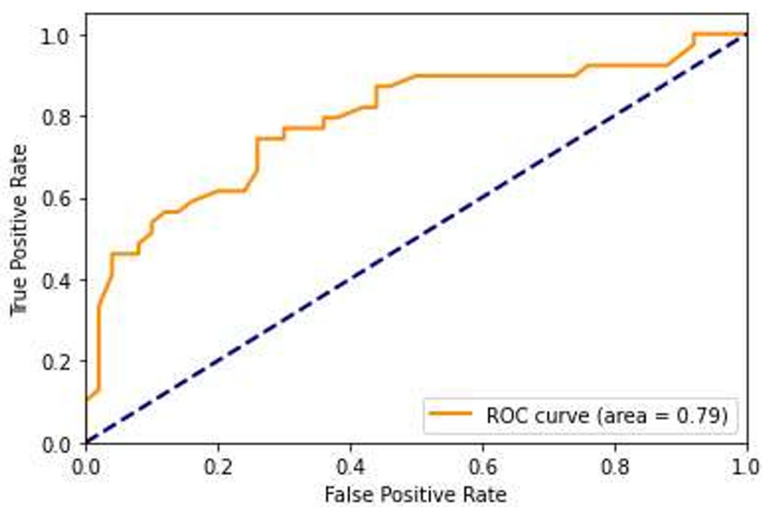

Fig. 3. Receiver Operating Characteristic.

4. Adaptive Boosting (AdaBoost) result.

In the program developed, we use the AdaBoost method with basic Decision Tree Classifier. We split the data into training and test sets, train the AdaBoost model and evaluate its performance, compute the confusion matrix, and present a classification report and five basic classification metrics.

Confusion Matrix:

[[42 8]
 [18 21]]

Classification Report:

	precision	recall	f1-score	support
0	0.70	0.84	0.76	50
1	0.72	0.54	0.62	39
accuracy			0.71	89
macro avg	0.71	0.69	0.69	89
weighted avg	0.71	0.71	0.70	89

Accuracy: 0.7078651685393258
Precision: 0.7241379310344828
Recall: 0.5384615384615384
F1-Score: 0.6176470588235294
AUC-ROC: 0.698974358974359

Accuracy estimates the proportion of correctly classified objects relative to all objects. In this case, the accuracy is 0.7079, which means that the model correctly classified 70.79% of the objects.

Precision measures the proportion of true positive predictions relative to all positive predictions. In this case, the accuracy for class "0" is 0.70, and for class "1", it is 0.72.

Recall measures the proportion of true positive predictions relative to all true positive objects. In this case, the recall for class "0" is 0.84, and for class "1", it is 0.54.

The F1-score is the harmonic mean between precision and recall. This is a numerical estimate of the balance between precision and recall. In this case, the F1-score for class "0" is 0.76, and for class "1", it is 0.62.

AUC-ROC (Area under the ROC Curve):

AUC-ROC is a model quality measure that evaluates the ability of a model to split classes and is dependent on a probability threshold. A value of 0.6989 means that the model splits the classes but with some uncertainty and the AUC-ROC close to 0.5 indicates that the classification model's performance is average.

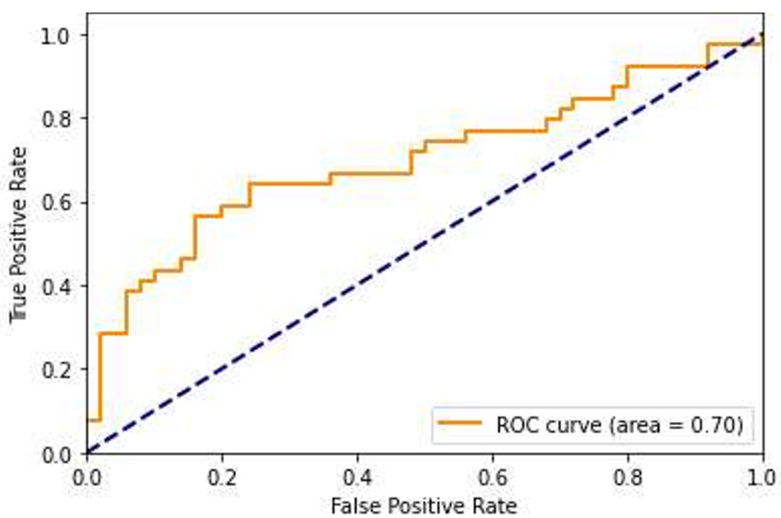

Fig. 4. Receiver Operating Characteristic.

5. Model Stacking Result:

This method shows how Model Stacking can be developed for a diabetes diagnosis problem using logistic regression and random forest as underlying models and gradient boosting as a meta-model. We can change the basic models and parameters to suit our aims and data.

Accuracy estimates the proportion of correctly classified objects relative to all objects. In this case, the accuracy is 0.764, which means that the model correctly classifies 76.4% of the objects.

Precision measures the proportion of true positive predictions relative to all positive predictions. In this case, the accuracy for class "0" is 0.71, and for class "1", it is 0.95.

Recall measures the proportion of true positive predictions relative to all true positive objects. In this case, the recall for class "0" is 0.98, and for class "1", it is 0.49.

The F1-score is the harmonic mean between precision and recall. This is a numerical estimate of the balance between precision and recall. In this case, the F1-score for class "0" is 0.82, and for class "1", it is 0.64.

AUC-ROC (Area under the ROC Curve):

AUC-ROC is a model quality measure that evaluates the ability of a model to split classes and is dependent on a probability threshold. The value of 0.7756 means that the model splits the classes with good performance, and the AUC-ROC close to 0.5 indicates an average performance of the classification model.

Random Forest showed the best results.

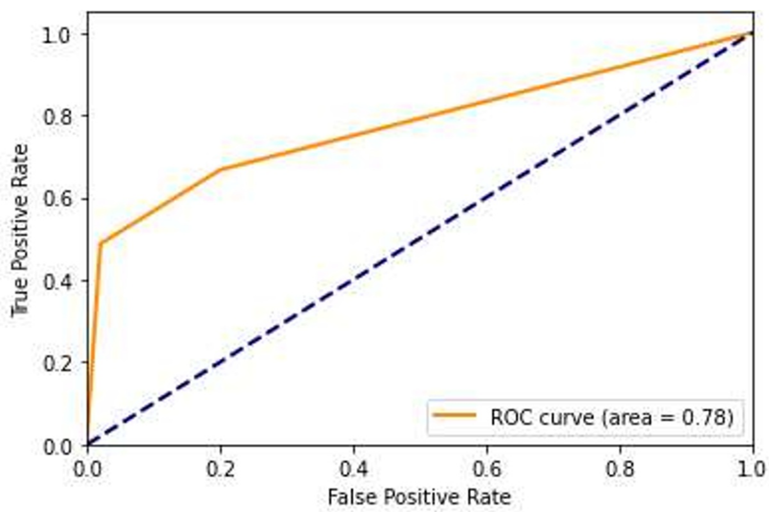

Fig. 5. Receiver Operating Characteristic.

4 Discussion

The application of ensemble methods in machine learning for the diagnosis of diabetes mellitus represents an important area of research and practical application in medicine.

However, it should be noted that the application of ensemble methods also comes with some risks and issues, such as the need for more data to train complex models and high computational resource capacities. It is also important to ensure the security and privacy of medical data when used in machine learning algorithms.

Overall, ensemble methods in machine learning play an important role in improving the diagnosis of diabetes and may lead to more efficient and accurate methods for detecting this disease.

5 Conclusion

Ensemble methods in machine learning for the diagnosis of diabetes mellitus represent promising and important fields of research and practical application in medicine. During our discussion, we covered a number of key points related to the use of ensemble methods for this purpose.

It is important to highlight that ensemble methods in machine learning such as Random Forest, Gradient Boosting, Bagging, and Adaptive Boosting provide significant benefits in diagnosing diabetes. Ensemble methods combine predictions from multiple models, leading to improved diagnostic accuracy. This is especially important in the medical field, where even small improvements can save patients' lives. These methods allow us to reduce the number of false-positive and false-negative diagnostic results, which is critical for the effective treatment of patients. Ensemble methods can combine information from various sources, such as clinical data, biochemical parameters, and images, making diagnosis more reliable. They have good resistance to overfitting, which is especially important when working with limited volumes of medical data. Some ensemble methods, such as Random Forest, can provide information about the importance of features, which helps doctors and researchers better understand the factors that influence diabetes diagnosis. Therefore, the application of ensemble methods in machine learning for diabetes diagnosis has enormous potential to improve healthcare and the quality of life of patients. Further research and development in this area will contribute to the development of effective tools for the diagnosis and treatment of diabetes.

References

1. H. Ahmad, M. U. Asghar, M. Z. Asghar, A. Khan, A. H. Mosavi, IEEE Access **9**, 146214–32 (2021)
2. D. Alghazzawi, O. Bamasaq, H. Ullah, M. Z. Asghar, Appl Sci. **11**, 11634 (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. D. Ziyadullaev, D. Muhamediyeva, Z. Abdullaev, S. Aynaqulov, K. Kayumov, E3S Web of Conferences **365**, 01019 (2023)
14. Y. K. Qawqzeh, A. S. Bajahzar, M. Jemmali, M. M. Otoom, A. Thaljaoui, Biomed Res Int. **2020**, 3764653 (2020)
15. D. K. Choubey, M. Kumar, V. Shukla, S. Tripathi, V. K. Dhandhania, Curr Diabetes Rev. **16**, 833–50 (2020)
16. 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.

17. 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.
18. 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