INTERNATIONAL SCIENTIFIC AND VOCATIONAL JOURNAL (ISVOS JOURNAL)

Vol.: 8 Issue: 1 Date: 30.06.2024 Received: 06.03.2024 Accepted: 05.05.2024 Final Version: 29.06.2024

ISVOS Journal, 2024, 8(1): 22-32 – DOI: 10.47897/bilmes.1447878

Comparative Analysis of Diabetes Diagnosis with Machine Learning Methods

Tuğba Aktaş [a], İsmail Mert Temel [b], Ahmet Saygılı [c,1]

[a] Department of Computer Engineering, Tekirdağ Namık Kemal University, Tekirdağ, Turkey
ORCID ID: 0009-0005-0580-7502
[b] Department of Computer Engineering, Tekirdağ Namık Kemal University, Tekirdağ, Turkey
ORCID ID: 0009-0008-7989-9747
[c] Department of Computer Engineering, Tekirdağ Namık Kemal University, Tekirdağ, Turkey
ORCID ID: 0000-0001-8625-4842

Abstract

Diabetes is a disease that occurs when the body cannot regulate the level of sugar (glucose) in the blood. Early diagnosis of this disease is important in preventing more serious diseases that may arise later. Within the scope of this study, an attempt was made to optimize the diabetes data set for use by training it with different models. At the very beginning of the study, Logistic Regression, KNN, SVM (Support Vector Machine), CART (Classification and Regression Trees), RF (Random Forest), Adaboost, GBM (Gradient Boosting Machines), XGBoost (Extreme Gradient Boosting), LGBM (Light Gradient Boosting). Machine), CatBoost models were used. According to the results of the models, RF, LGBM, XGBoost accuracy, and f1 values were observed as the best models, respectively. As a result, in the Random Forest model, which produced the most successful results, Accuracy: 0.88, F1 Score: 0.84, and ROC AUC: 0.95 values were obtained, respectively.

Keywords: "Diabetes, Machine Learning, Random Forest."

1. Introduction

Diabetes arises from deficiencies, ineffectiveness, or inadequate production of the insulin hormone within the body, disrupting carbohydrate metabolism and elevating blood glucose levels. Characterized by symptoms like pronounced thirst, heightened hunger, and frequent urination, this condition, if left untreated, can lead to various complications, particularly impacting vascular health. The deleterious effects of prolonged elevated sugar levels can result in permanent damage to numerous organs including the eyes, kidneys, nerve endings, heart, brain, and lower limb vasculature [1].

Early diagnosis of diabetes is of critical importance in preventing these negative effects. However, diabetes can often progress without showing symptoms. Therefore, the diagnosis of diabetes is made by examinations by specialist doctors or by examining blood samples in a laboratory environment. The information understands the importance of diabetes worldwide that 537 million people have this disease as of 2021 and 6.7 million people died due to diabetes this year alone [2], early diagnosis of diabetes is of great importance to prevent possible complications and reduce the morbidity and mortality caused by this disease [3]. Diabetes is diagnosed through blood tests as well as symptoms of the disease. However, the disease can often progress without showing symptoms and may not be diagnosed even by specialist doctors [4]. Therefore, regular health checks and preventive screenings in individuals at risk are important for early diagnosis of diabetes.

With the development of technology, studies for the early diagnosis of diseases that affect most of the population in the world are increasing. Artificial intelligence and machine learning techniques put humanity at a plus point in terms of time and money in diagnosing many diseases.

The "Diabetes" dataset constitutes a subset of a broader dataset housed at the National Institutes of Diabetes-Digestive-Kidney Diseases in the United States. This dataset specifically pertains to diabetes research conducted on females of Pima Indian descent aged 21 and above, residing in Phoenix, the fifth most populous city in the state of Arizona, USA [5]. In our research, experimental outcomes were derived from the utilization of this dataset.

[1] Corresponding Author
E-mail Address: asaygili@nku.edu.tr

This study consists of three main headings: data preprocessing, feature determination, and classification. Studies were carried out to make the data ready for processing through data preprocessing. At the stage of determining the attributes, the attributes that most represented the data were determined. It has been observed that these processes contribute to the trained model giving the best results. In addition, creating a website using Streamlit with this model contributed to providing users with the opportunity to predict with the model [6]. The second part of our study includes materials and methods, the third part includes the findings, and the fourth and last part includes the results.

2. Material and Method

In this title, the data set used in the study and the methods that enable us to automatically detect diabetes using this series set will be mentioned.

2.1. Data Set

The data set used in the study [5] consists of 768 observations and 9 variables. 8 of the 9 variables are numerical variables, and the target variable (Outcome) is categorical. Out of 768 observations, 500 have a target variable (Outcome) value of 0, and 268 have a value of 1. Therefore, the data set does not show a normal distribution. This can also be seen in the histogram chart.

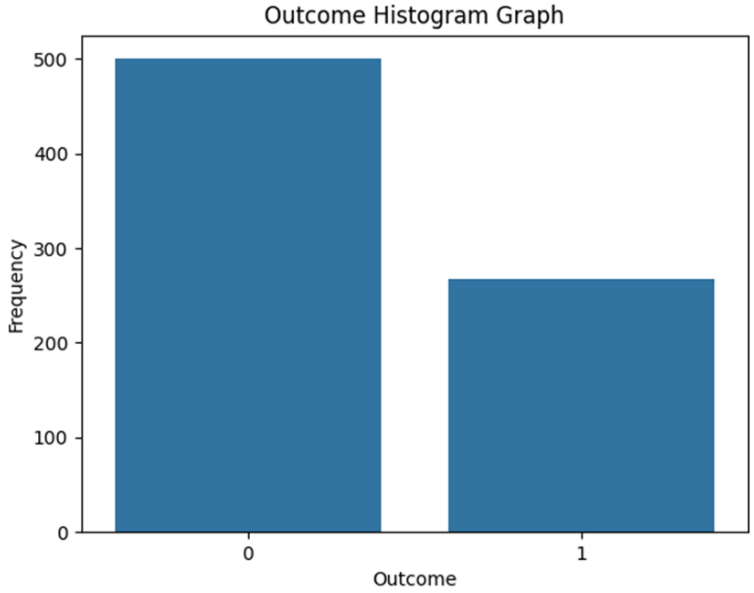

Fig. 1. Outcome Histogram Chart

It was determined that some values in the data set were empty. The distribution of null values is as follows; Glucose 5, Blood Pressure 35, Skin Thickness 227, Insulin 374, and BMI.

Table 1. Data set attributes table

Attribute Name	Attribute Description
Pregnancies	Number of pregnancies
Glucose	2-hour plasma glucose concentration in oral glucose tolerance test
Blood Pressure	Blood Pressure (diastolic blood pressure) (mm Hg)
Skin Thickness	Skin Thickness
Insulin	2-hour serum insulin (mu U/ml)
Diabetes Pedigree Function	Function (2-hour plasma glucose concentration in oral glucose tolerance test)
BMI	Body mass index
Age	Age (years)
Outcome	Patient (1) Healthy (0)

The data in the data set does not show a normal distribution. If it were a normally distributed data set, filling the empty values with the average values would be considered a more appropriate method. However, since the data set did not show a normal distribution, empty values were filled with the median. The median value of those with an Outcome value of 0 was assigned to

the blank values with an Outcome value of 0, and the median value of those with an Outcome value of 1 was assigned to the blank values with an Outcome value of 1.

Box-plot was used in outlier analysis, and larger outliers were observed in the insulin variable compared to others. Figure 2 shows the Box Plot graph where these outliers can be seen.

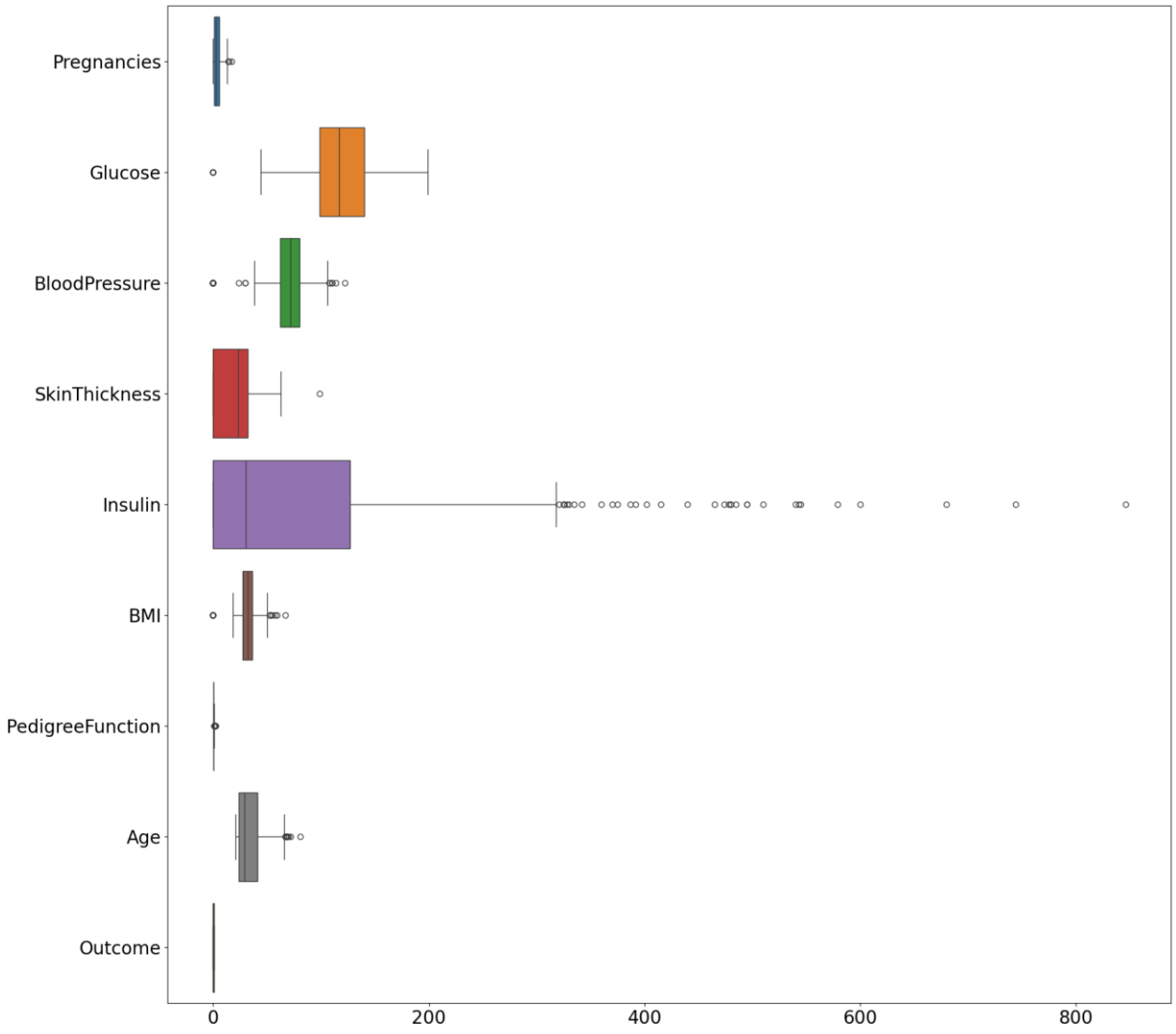

Fig. 2. Box plot graph

Based on this abnormal distribution, the 25th and 75th percentile values were used for the interquartile range. Low-up values of these percentiles were used to suppress outliers. During feature extraction processes, new features such as “NEW_AGE_CAT”, “NEW_BMI”, and “NEW_GLUCOSE” were produced from some features. Thanks to these new attributes, the values received were divided into certain categories. For example, the age group was divided into 3 groups: "young", "old" and "middle age". Thanks to this categorization, analysis by age groups was provided. As a result, the function returned X and Y values that can be tested in different models.

2.2. Classification Model

The concept of classification is simply distributing data among various classes defined on a data set. Classification algorithms learn this distribution shape from the given training set and then try to classify it correctly when test data whose class is not clear comes [7]. In practical implementation, classification algorithms undergo a two-stage process. Initially, the classification model is constructed through the analysis of a specific dataset labeled as training data (X). Subsequently, this derived classification model is employed to analyze a novel dataset, assessing the presence of identified classes within the data. This new dataset, wherein class labels are endeavored to be predicted and the predictive efficacy of the model is gauged, is referred to as test data.

The partitioning of the dataset into training and test data can be executed through various methodologies. For instance, there exist techniques where 60% of the dataset is designated for training purposes while 40% is reserved for testing, with random or

exponentially changing assignments for training and test sets. However, cross-validation more accurately evaluates the generalization performance of the model. It divides the data set into a limited number of pieces and creates multiple training and test sets by using these pieces sequentially. In this way, the performance of the model in different data sections is evaluated more comprehensively. In this study, models were trained with Random Forest, XGBoost, LGBM, CART, and Voting Classifier using the 10-fold cross-validation method, and the results were observed.

2.2.1. Cross Validation

In the k-fold cross-validation method, first, the training set to be used in the training process is mixed and divided into k subsets of equal size. These processes are repeated k times, and in each iteration, the next subset is removed from the training data set and used as the test set. Once the evaluation process is completed for all parts, the cross-validation model produces a performance measure and results for all data.

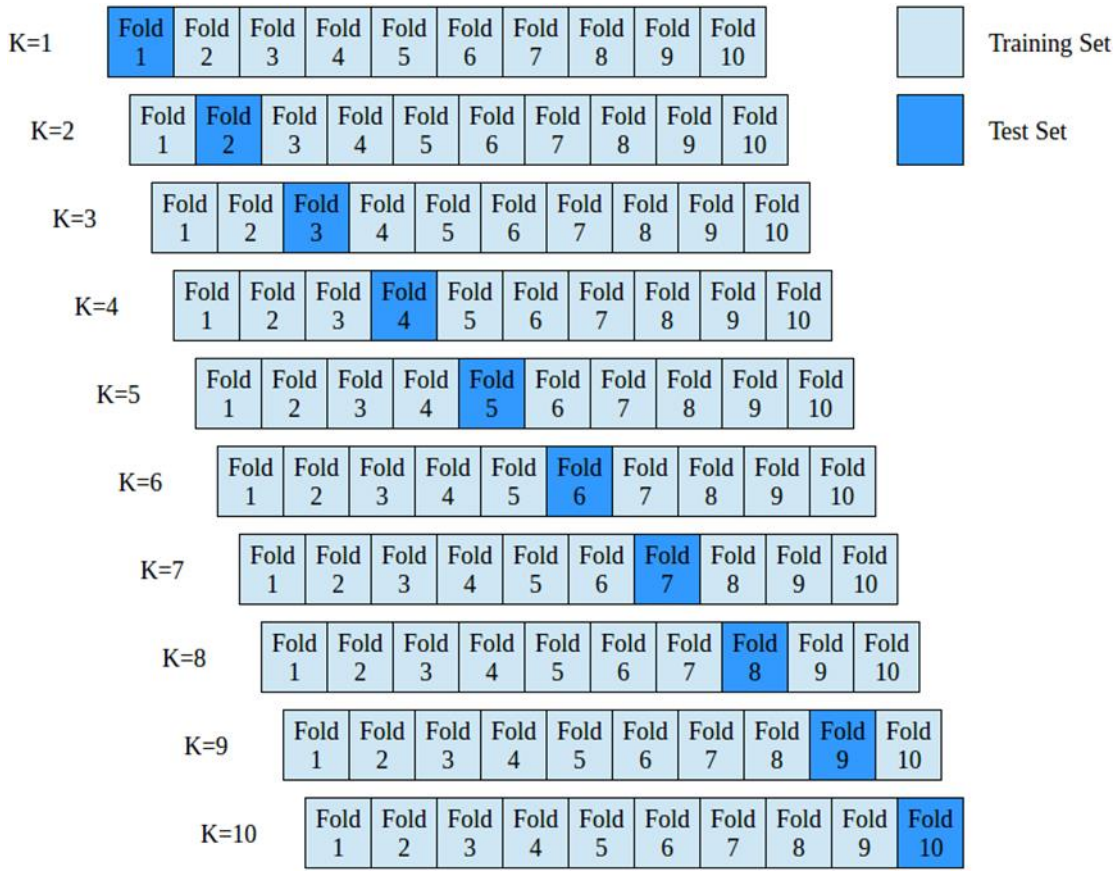

Fig. 3. Cross Validation

The success of the classification model is determined by comparing the number of samples assigned to the correct class and the number of samples assigned to the wrong class. Accuracy, F1 Score, and ROC AUC metrics were used to evaluate model success.

2.2.2. Model performance evaluation metrics.

In this study, the F1 Score, ROC AUC, and Accuracy performance measures were taken into consideration. F1 Score shows the harmonic average of Precision and Recall values. The reason for having a harmonic average instead of a simple average is that we should not ignore extreme cases.

$$\text{F1} - \text{score: } 2 \text{ x } \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1)$$

ROC AUC (Receiver Operating Characteristic Area Under the Curve) is a metric utilized to evaluate the performance of classification models. The ROC curve itself is a graphical representation of the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR). TPR is plotted on the Y-axis, while FPR is on the X-axis as seen in Figure 4. The area under the ROC curve, denoted as AUC quantifies the degree of separability between different classes. A higher AUC value indicates better discrimination performance between the classes.

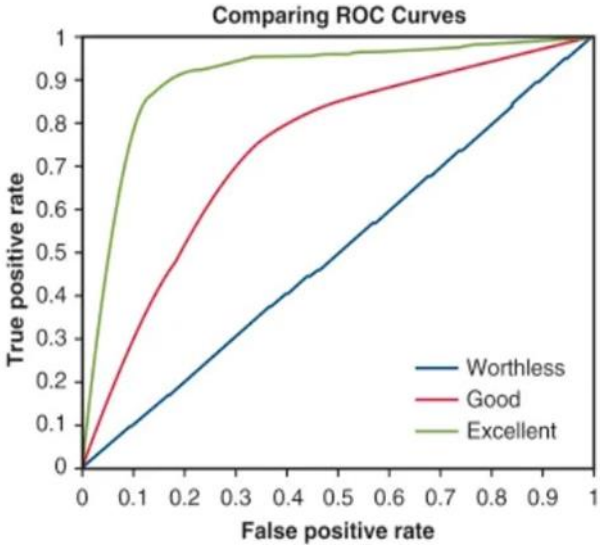

Fig. 4. ROC Curves chart

Accuracy serves as a prevalent metric for evaluating model performance, yet it is not exhaustive in isolation. Computed as the ratio of correctly predicted areas within the model to the entire dataset, accuracy provides a snapshot of predictive success.

$$\text{Accuracy:}\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}\text{x}\,100 \tag{2}$$

2.3. Classification Methods

For the classification problem addressed in the study, machine learning algorithms existing in the literature were used. The model performances of Logistic Regression, kNN, SVM, and Decision tree were evaluated. The classification algorithms including Decision Tree, Random Forest, Ada Boost, Gradient Boosting, XGBoost (XGB), LightGBM (LGBM), and CatBoost were assessed using the dataset without preprocessing. In this way, the difference in performance measures before and after data pre-processing was examined.

2.3.1. Logistic Regression

Logistic Regression, despite its name containing regression, functions primarily as a classification algorithm, commonly utilized in binary classification tasks [8, 9]. It aims to estimate the probability of an event occurring between two distinct classes. Notably, the divergence between logistic regression and linear regression lies in their methodologies for delineating the boundary between classes. While Linear Regression employs the Least Squares Method to determine the optimal boundary line, Logistic Regression utilizes the Maximum Likelihood approach. Logistic Regression implements the Sigmoid (Logistic) function for classification purposes. This function is characterized by an S-shaped curve, facilitating the conversion of real numbers from the range (-∞, ∞) to the range (0, 1).

2.3.2. K-Nearest Neighborhood (K-NN)

The K-Nearest Neighbor (K-NN) algorithm is a simple machine learning algorithm used for classification and regression problems. K-NN is one of the supervised learning algorithms structured for different analyses with special operations on various data sets. Essentially, it performs classification or regression by examining the K-Nearest Neighbors around an example.

K-Nearest Neighborhood works on the principle of “belonging to similar classes” [9]. Determines the number of neighbors to use to classify one sample or predict another. Usually, the K value is selected by the user. If the number K is 1, then it is included in the class of its nearest neighbor. In this algorithm, the choice of the number k is of critical importance in determining the result. In addition to the K value, the distance calculation method also affects the performance of the algorithm. The most used among these distance calculation methods is the method known as the Minkowski distance calculation function. Therefore, a comparative analysis of the results obtained using different distance calculation methods can be made.

2.3.3. SVM (Support Vector Machine)

SVM (Support Vector Machine) stands as a prominent machine learning algorithm employed predominantly in classification tasks. It endeavors to delineate the boundary between classes through the utilization of various parameters. This boundary is represented by a plane known as the hyperplane, which effectively segregates the feature space into distinct regions. A Hyper

Plane is a plane that divides the feature space. The SVM method selects and optimizes this hyperplane. SVM is generally applied to nonlinear classification problems. In this case, Kernel functions are used to create nonlinear boundaries in the feature space. Support Vectors are used to maximize the margin between classes and maintain this margin. To prevent overlearning, the C Parameter is used to help these processes. As the C value increases, the model tries to fit more training data sets, which can lead to overlearning [10-12].

2.3.4. CART (Classification & Regression Trees)

CART (Classification & Regression Trees) is an algorithm rooted in decision trees, serving applications in both classification and regression analyses. Notably, it serves as the foundational framework for Random Forest. The primary objective of CART is to streamline intricate structures within datasets into straightforward decision structures. This entails the segmentation of heterogeneous datasets into homogeneous subgroups based on a designated target variable. Decision trees refer to the tree structure used to represent a data set and predict a target variable by splitting the data set by making decisions within it. The CART algorithm creates and refines these decision trees. At each internal node, a decision is made using a feature and a threshold value. Partitioning aims to divide the data set into homogeneous or "purer" subsets. While doing this, division criteria such as the Gini coefficient and Mean Squared Error can be used [13].

2.3.5. Random Forest

Random Forest is a stronger decision tree created by combining many decision trees. Since multiple decision trees are brought together, each tree is modeled using a different subset of data or features in training. These trees are then combined. It is generally used in classification and regression processes. In the Random Forest method, random samples and random features are used when training each tree. At the same time, each tree is trained using a specific subset of features. A randomly selected subset of features is used for each tree. Another feature of the random forest model is that it shows us how important the features are. The most appreciated point about the algorithm is that it allows you to re-explore your data set more deeply by creating various models on it [10, 14].

2.3.6. AdaBoost (Adaptive Boosting)

AdaBoost, short for Adaptive Boosting, is an algorithm designed to construct a robust classifier by amalgamating weaker classifiers. Its operational principle revolves around iteratively refining the classifier by accentuating the significance of misclassified instances from preceding stages. Throughout this iterative process, the weights attributed to misclassified examples are augmented, thereby intensifying the model's attention toward rectifying its deficiencies. This strategic emphasis aims to enhance the model's precision in classification tasks by fortifying its focus and mitigating weaknesses.

2.3.7. Gradient Boosting

The basis of Gradient Boosting is models such as Gradient Boosted Tree (GBM). The basic idea is to add new estimators by trying to correct the errors of previous estimators. In AdaBoost, each example is learned with a weighted emphasis based on the predictions of preceding models. The two biggest advantages of Gradient Boosting are high predictive power, resistance to overfitting, and flexibility. Disease diagnosis and medical diagnosis can be shown as an application area [15].

2.3.8. XGBoost

XGBoost (Extreme Gradient Boosting) is one of the Gradient Boosting algorithms. It is known for its generalization ability, especially in classification and regression tasks. XGBoost includes many improvements to Gradient Boosting. XGBoost works on decision trees. It makes a name for itself with its high performance and high efficiency. The first step in XGBoost is to make the first estimate (Base Score). The model works iteratively. The goal is to ensure that the sum of all models produces a prediction that comes as close as possible to the actual output. This process is accomplished by stepping the loss function along its gradient [16].

2.3.9. LightGBM (Light Gradient Boosting Machine)

LightGBM (Light Gradient Boosting Machine) is an open-source Gradient Boosting framework that provides high performance on large data sets. Many features distinguish LightGBM from others. Providing faster and higher efficiency. Better performance with less memory usage. Justification to better righteousness. Ability to process big data. Supports parallel and GPU learning. These can be considered features that make LightGBM stand out. It offers the flexibility to choose defined custom loss functions and evaluation metrics. LightGBM is optimized to deal with large data sets. Thanks to its parallel computing capabilities and scalable structure, shorter training times can be achieved by reducing training times [17].

2.3.10. CatBoost

CatBoost especially in classification and regression tasks with both numerical, categorical, and text data. It is an open-source Gradient Boosting method designed to provide high performance and ease of use with GPU support and visualization options. CatBoost maintains a row balance as its trees grow. This ensures that when splitting during each feature, other features are less affected. CatBoost automatically adjusts the learning rate and also allows user customization, which distinguishes itself from other gradient-boosting methods [18].

Parameter values of the classifiers used in the study are given in Table 2.

Table 2. Parameter Values of Classifiers

Method	Parameters
Logistic Regression	Default Values
kNN	Number of neighbors: 5 Leaf size: 30 Distance metric: Euclidean
SVM	Default values
CART	Default values
Random Forest	The number of trees in the forest: 100 The function to measure the quality of a split: Gini
AdaBoost	The maximum number of estimators: 50 The learning rate: 1.0
Gradient Boosting	The number of boosting stages to perform: 100 The learning rate: 0.1
XGBoost	Default Values
LightGBM	Maximum tree leaves for base learners:31 Boosting Type: Gradient Boosting Decision Tree Max Depth: -1 The learning rate: 0.1 Number of boosted trees to fit: 100
Cat Boost	Default Values

Table 3 lists the results of machine learning methods before data preprocessing. According to the average of these results, it can be seen from the table that the most successful methods are Cat Boost, Logistic Regression, and Random Forest.

Table 3. Performance Values Before Feature Extraction and Data Preprocessing

Method	ROC_AUC	F1-Score	Accuracy
Logistic Regression	0.83	0.63	0.77
kNN	0.75	0.57	0.72
SVM	0.82	0.58	0.76
CART	0.67	0.57	0.70
Random Forest	0.83	0.61	0.77
AdaBoost	0.82	0.64	0.76
Gradient Boosting	0.82	0.63	0.76
XGBoost	0.79	0.62	0.74
LightGBM	0.80	0.62	0.75
Cat Boost	0.83	0.64	0.77

In the graphs in Figures 5, 6, and 7, we see the accuracy, f1 score, and roc-auc values of the base models before data pre-processing, respectively. In the Accuracy values chart in Figure 5, it is seen that the most successful results were obtained with logistic regression, random forest, and CatBoost methods.

Figure 6 shows the success of F1 Score values according to the methods. When the figure is examined, it is seen that the CatBoost and AdaBoost methods give the highest values.

In Figure 7, we see the performance values of the methods according to ROC Auc values. From this figure as in other metrics, logistic regression, random forest, and CatBoost methods are the most successful.

Figures 8, 9, and 10 show the results obtained after feature extraction and preprocessing. As seen in the graphs in Figures 8, 9, and 10, better results were obtained when classification was performed after data preprocessing and feature engineering

processes. It is seen that there is an increase in all accuracy, f1 score, and roc-auc values. This increase reveals the importance of preprocessing and feature engineering.

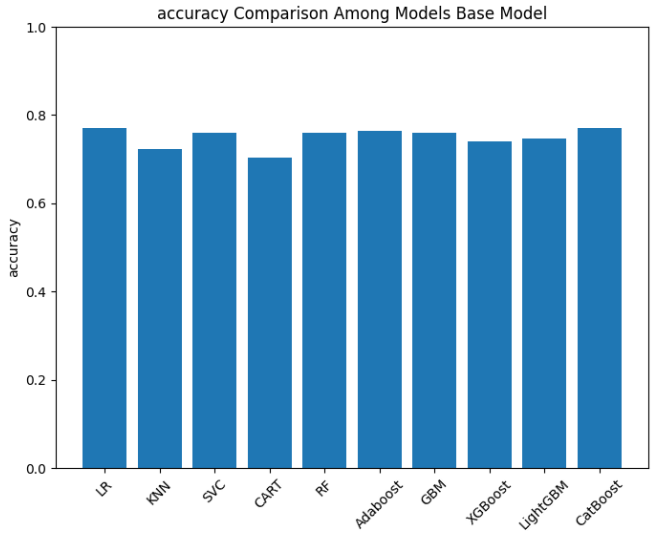

Fig. 5. Accuracy Values for Base Model

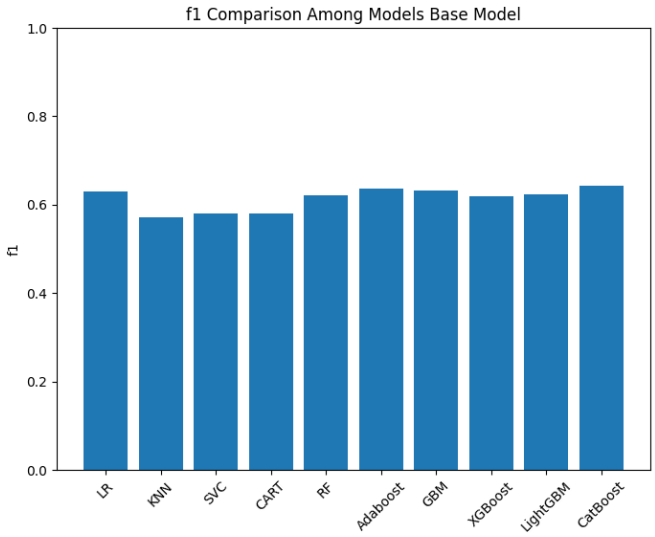

Fig. 6. F1-Score Values for Base Model

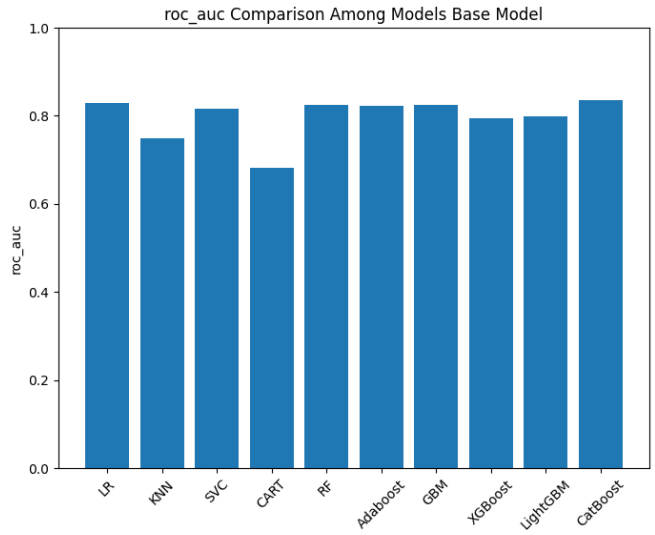

Fig. 7. Roc-AUC Values for Base Model

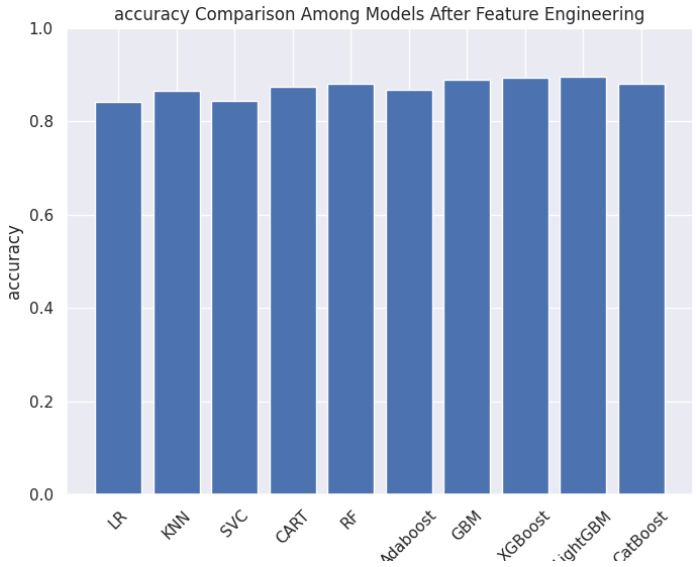

Fig. 8. Accuracy Values after Feature Engineering

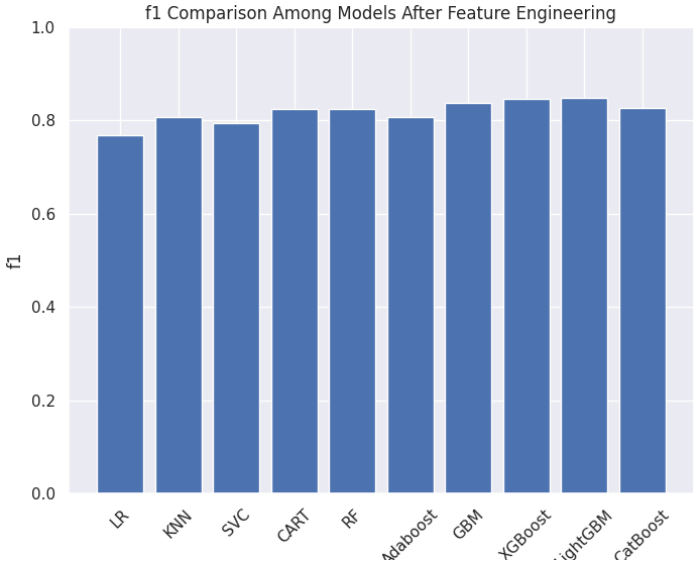

Fig. 9. F1 Values after Feature Engineering

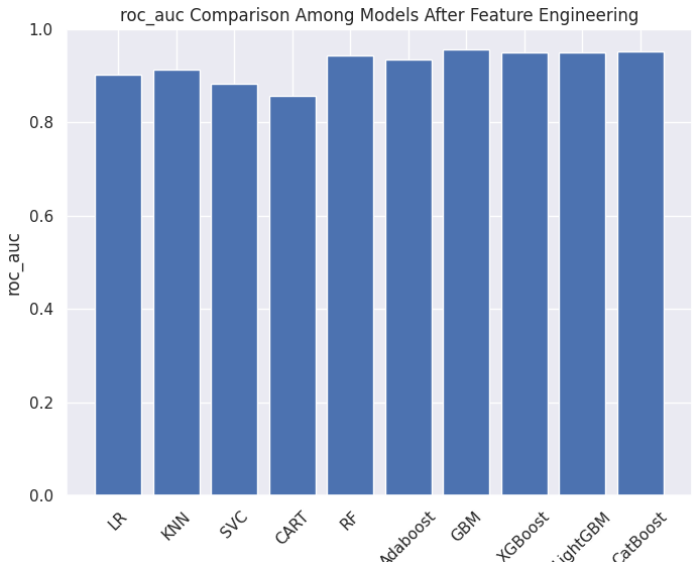

Fig. 10. Roc AUC Values after Feature Engineering

3. Experimental Results

Python programming language and PyCharm were used in this study. The observations in the data set show an unbalanced distribution and action can be taken for this. The outcome value of 0 was made more balanced by removing 250 of the 500 values. It was concluded successfully. However, training the model in its unstable state continued. After data preprocessing and feature extraction, the data set was retrained with the specified classification models. The results have improved noticeably. The models with the highest F1 score and accuracy values are RF, LGBM, and XGBoost. The model was trained using a Random Forest. For the Random Forest model observed after data preprocessing and feature extraction, the Roc_Auc score value was 0.95, the f1 score value was 0.84, and the accuracy score value was 0.88.

Table 3. Performance comparison of the studies

Author	Method(s)	Accuracy	F1 Score	ROC AUC
Mehmet Bilal ER [19]	ESA+LSTM	0,86	0.88	-
Güneş HARMAN [20]	DVM	0,88	0.87	-
Hassan ve Shaheen [21]	Random Forest	0,84	-	-
Başer [1]	Random Forest	-	-	0.91
This Study	Random Forest	0.88	0.84	0.95

Streamlit interface was developed with the model. Information was given to the user by making explanations about diabetes and the data set [6]. The interface of the program is seen in Figure 11.

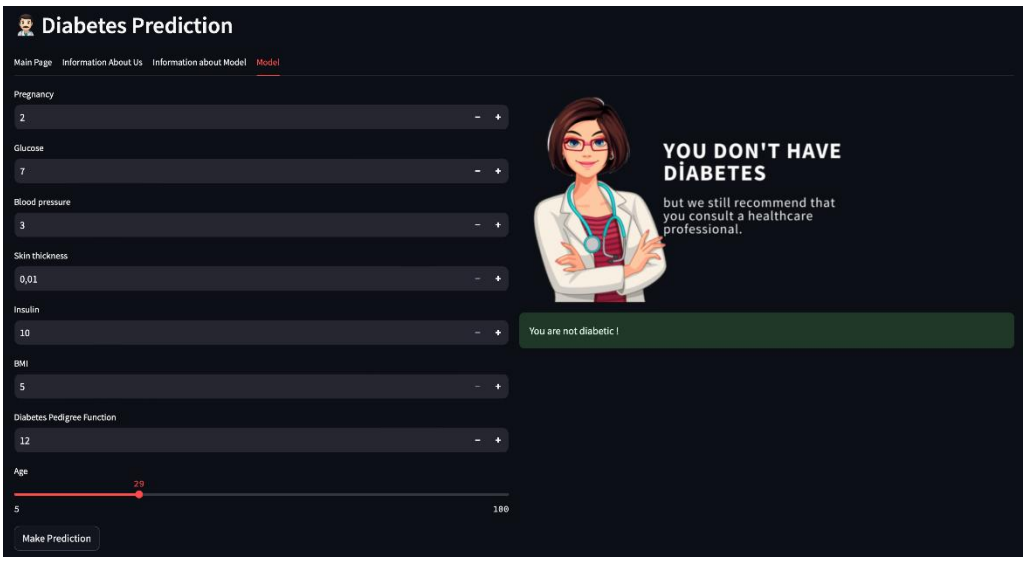

Fig. 11. Streamlit Interface [6]

4. Conclusion

Processing data and obtaining information in the field of health plays a major role in the early diagnosis and treatment of diseases. Machine learning techniques show very successful results in the analysis and diagnosis of these diseases. Diabetes is a serious disease and if not diagnosed early, it can cause unavoidable consequences. Therefore, early diagnosis of such diseases is important. In this study, various machine learning methods were used to predict whether a person has diabetes or not. The results obtained were compared with other studies in the literature and it was observed that current studies are ongoing. After LightGBM, XGBoost, and Random Forest, data preprocessing, and feature engineering, the highest outputs were obtained as f1 score and accuracy value. The most successful results were obtained with the Random Forest modeling method. The results mentioned are a Roc_Auc score of 0.95, an f1 Score of 0.84, Accuracy Score of 0.88, respectively. Since the "Outcome" values of the data set do not have a normal distribution and the decrease in the number of observations in the data set when normalization is attempted may cause the model to memorize, the non-normally distributed version was used. To obtain better results in future studies, a solution will be sought to organize the data and reduce the number of observations.

If a better data pre-processing process is applied and the data is trained in a more balanced way; A mobile application can be created using this forecasting model and easily presented to the user. Thus, an accessible model is developed. Users are aware of their diabetes risks and can be directed to the nearest healthcare facility.

References

[1] B. Ö. Başer, M. Yangın, and E. S. Sarıdaş, "Makine öğrenmesi teknikleriyle diyabet hastalığının sınıflandırılması," *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi,* vol. 25, no. 1, pp. 112-120, 2021.

[2] W. W. H. Organization. " “Diabetes.”." https://www.who.int/news-room/fact-sheets/detail/diabetes (accessed Feb. 12, 2024).

[3] H. Zhou *et al.*, "A computer simulation model of diabetes progression, quality of life, and cost," *Diabetes care,* vol. 28, no. 12, pp. 2856-2863, 2005.

[4] U. Köse, "Zeki optimizasyon tabanlı destek vektör makineleri ile diyabet teşhisi," *Politeknik Dergisi,* vol. 22, no. 3, pp. 557-566, 2019.

[5] A. D. Khare. "“Diabetes Dataset.”." https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset/data (accessed Feb. 1, 2024).

[6] T. A. a. İ. M. Temel. "“Diagnosing Diabetes Streamlit Web Page.”." https://github.com/tubaaktas/DiabetesPred (accessed Feb. 1, 2024).

[7] G. Bonaccorso, "Machine learning algorithms Packt Publishing Ltd," ed: Packt Publishing Ltd, 2017.

[8] E. Dağdevir and M. Tokmakçı, "The Role of Feature Selection in Significant Information Extraction from EEG Signals," *International Scientific and Vocational Studies Journal,* vol. 5, no. 1, pp. 1-6, 2021.

[9] J. P. Mueller and L. Massaron, *Machine learning for dummies*. John Wiley & Sons, 2021.

[10] A. Saygılı, "Classification and Diagnostic Prediction of Breast Cancers via Different Classifiers," *International Scientific and Vocational Studies Journal* pp. 48-56, 2018.

[11] A. Saygılı and S. Varlı, "Automated diagnosis of meniscus tears from MRI of the knee," *International Scientific and Vocational Studies Journal,* vol. 3, no. 2, pp. 92-104, 2019.

[12] S. Suthaharan and S. Suthaharan, "Support vector machine," *Machine learning models and algorithms for big data classification: thinking with examples for effective learning,* pp. 207-235, 2016.

[13] W. Y. Loh, "Classification and regression trees," *Wiley interdisciplinary reviews: data mining and knowledge discovery,* vol. 1, no. 1, pp. 14-23, 2011.

[14] G. Biau and E. Scornet, "A random forest guided tour," *Test,* vol. 25, pp. 197-227, 2016.

[15] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics,* vol. 7, p. 21, 2013.

[16] T. Chen *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2,* vol. 1, no. 4, pp. 1-4, 2015.

[17] D. D. Rufo, T. G. Debelee, A. Ibenthal, and W. G. Negera, "Diagnosis of diabetes mellitus using gradient boosting machine (LightGBM)," *Diagnostics,* vol. 11, no. 9, p. 1714, 2021.

[18] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," *Advances in neural information processing systems,* vol. 31, 2018.

[19] M. B. Er and İ. Işık, "LSTM tabanlı derin ağlar kullanılarak diyabet hastalığı tahmini," *Türk Doğa ve Fen Dergisi,* vol. 10, no. 1, pp. 68-74, 2021.

[20] G. Harman, "Destek vektör makineleri ve naive bayes sınıflandırma algoritmalarını kullanarak diabetes mellitus tahmini," *Avrupa Bilim ve Teknoloji Dergisi,* no. 32, pp. 7-13, 2021.

[21] F. Hassan and M. E. Shaheen, "Predicting diabetes from health-based streaming data using social media, machine learning and stream processing technologies," *International Journal of Engineering Research and Technology,* vol. 13, no. 8, pp. 1957-1967, 2020.