EKSAKTA
Journal of Sciences and Data Analysis

E-ISSN: 2720-9326
P-ISSN: 2716-0459

DOI : 10.20885/EKSAKTA.vol7.iss1.art11

A Comparative Evaluation of XGBoost and LightGBM for Diabetes Mellitus Risk Prediction Using a Public Dataset and Web-Based Dashboard

Sischa Wahyuning Tyas[1,*] , Muhammad Al Hafidz[2]

[1] Department of Data Science, Faculty of Computer Science, Universitas Pembangunan Nasional "Veteran" Jawa Timur

[2] Information Systems, Faculty of Computer Science, Universitas Pembangunan Nasional "Veteran" Jawa Timur

[*] *Corresponding author: sischa_wahyuning.sada@upnjatim.ac.id*

Abstract: Diabetes mellitus is one which concerns the globe since most instances are often only diagnosed when there is a complication. Early diagnosis on diabetes mellitus is key in dealing with the problems resulting from such conditions both medically and financially. The objective of this research paper is to examine two models of machine learning that employ gradient boosting algorithm; XGBoost and LightGBM. This study uses a technique called RandomizedSearchCV to optimize the performance of the proposed machine learning models. This approach can be classified as a combination optimization algorithm, as the RandomizedSearchCV is coupled with a boosting algorithm in order to perform an efficient exploration of hyperparameter space. Model evaluation were considered in this study such as accuracy, precision, recall, F1 score, and ROC-AUC. The total dataset of 768 observations was split using an 80% proportion for training with 614 samples and 20% for testing with 154 samples. Accroding to the result, the LightGBM is a more efficient machine learning model compared to XGBoost. The accuracy, precision, and recall scores of the LightGBM were 77.3%, 71.1%, and 59.3%, respectively; however, the latter is similar to that of the XGBoost model. Moreover, the former has a higher F1 score of 64.6% and ROC-AUC of 83.0%, indicating that the LightGBM is a balanced model for classifying the target variables. The most machine learning model was embedded in a web-based application using a tool known as Streamlit. This integration further strengthens the novelty by bridging machine learning model development with practical and real-time healthcare application. The system is useful for early detection of diabetes mellitus and can be used to determine whether a patient is at risk of developing the disease using real-time prediction and user-friendly data input. The results of the study showed that gradient boosting machine learning models can be used to support preliminary risk assessment and detect early cases of diabetes mellitus.

Keywords: Diabetes, XGBoost, LightGBM, Risk Prediction, Streamlit

Introduction

Diabetes mellitus is a major global health concern that affects millions of people around the world. This condition exists as a result of the body's inability to use glucose appropriately. This occurs as a result of either an inadequate amount of insulin that the body requires or as a result of insulin resistance [1]. When left unchecked and without effective measures adopted to curb the situation, diabetes mellitus can result in various complications within one's future life. According to the World Health Organization [2], diabetes cannot be easily detected in its early stages as its symptoms are generally vague. As a result, an individual finds out that they are suffering from the condition after they have already suffered adverse effects from the complications that diabetes mellitus poses. This concern is redoubled by the International Diabetes Federation's 10th Diabetes Atlas

(2025), which reveals that approximately 43% of diabetes mellitus cases across the globe actually go undetected and consequently unnoticed [3].

Type 2 is the common form of diabetes, developing when the body isn't able to use insulin properly or doesn't make enough . In Indonesia, a consistent rise in diabetes is noted by the Ministry of Health. This drives up national health costs because so many people develop long-term complications requiring ongoing care. The Indonesian Health Survey of 2023 [4] showed that among people aged 15 years and above, diabetes prevalence based on blood sugar tests is 11.7%. This figure is higher than medical diagnosis, reflecting that many cases remain undetected at clinical practices.

Traditional screening for diabetes has significant limitations because it relies heavily on medical check-ups and healthcare access [5], [6]. This gap highlights the importance of data-driven prediction methods that enable earlier, more accurate, and more accessible risk identification. Machine learning (ML) [7], [8] is a powerful tool in detecting complicated nonlinear connections in the collected data. Unlike statistical analysis tools, ML shows greater flexibility and is capable of generating superior predictions in terms of classification, estimation, and detection in healthcare applications. Moreover, various ML algorithms have already been successfully used for making decisions when dealing with diagnosis and prediction of diabetes. With modern IT progress, this approach became even more efficient compared to traditional analytical methods.

The Extreme Gradient Boosting (XGBoost) is a popular machine learning algorithm characterized by excellent predictive ability. Furthermore, it integrates regularization, which prevents overfitting [8]. XGBoost has exhibited reliable predictive capability in disease prediction, especially diabetes, particularly after appropriate preprocessing, dealing with class imbalance, and hyperparameters' optimization [9]. Light Gradient Boosting Machine (LightGBM) is a recent development in gradient boosting characterized by a combination of fast training process and highly accurate predictions. From [10], it can be noted that LightGBM offers high sensitivity, which means that it facilitates efficient feature selection and thus is ideal for building effective cost-efficient screening systems.

From previous research works carried out on public datasets, XGBoost and LightGBM are known to offer promising results in the context of disease prediction such as diabetes [11], [12], [13]. More specifically, XGBoost is known to exhibit superior predictive power and enhanced recall upon optimization to minimize false negatives in the medical context [11]. On the other hand, although similar in terms of predictive capability, LightGBM exhibits superior training efficiency and stable performance, particularly in the case of correlated data and class imbalance issues [14]. Earlier comparisons have highlighted XGBoost as having superior accuracy, recall, but LightGBM boasts comparable performance with faster execution without a significant loss of predictive performance, which warrants its usage regardless [15]. Despite all the positives, we still have several gaps. This is due to a lack of in-depth studies regarding the implications of hyperparameter optimization on the performance of not only XGBoost but also LightGBM. There is an overwhelming focus on accuracy without consideration of important values such as recall and AUC-ROC. Additionally, the transition from prediction models to accessible and usable applications is not a well-explored subject.

This work endeavors to fill this gap through a head-on comparison of the two models, i.e., XGBoost and LightGBM, and through the tuning of their hyperparameters via the RandomizedSearchCV technique. The models will also be validated through clinical metrics, recall, and AUC ROC, as well as through traditional performance tests.

Secondly, the best model will also be made available as a web app through the Streamlit library.

Materials and Methods

Materials

The research involved the use of open-source diabetes data set on Kaggle that originated from the National Institute of Diabetes and Digestive and Kidney Diseases. The data set had 768 instances with 8 predictor variables, namely Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age. In this analysis, the output variable was binary with diabetes indicated by a value of 1 and non-diabetes represented by 0. Some of the tools used in the process were Python version 3.10, scikit-learn, XGBoost, and LightGBM.

Once preprocessed, the data were split into 80% training and 20% testing sets before applying the initial models of XGBoost and LightGBM. The baseline models were evaluated based on standard performance metrics. Hyperparameter tuning was then carried out using RandomizedSearchCV to arrive at the optimal configurations. The best performing model from these optimizations was re-trained and then deployed in a web application using Streamlit, providing an interactive diabetes risk prediction interface. The detailed research workflow is presented in Figure 1.

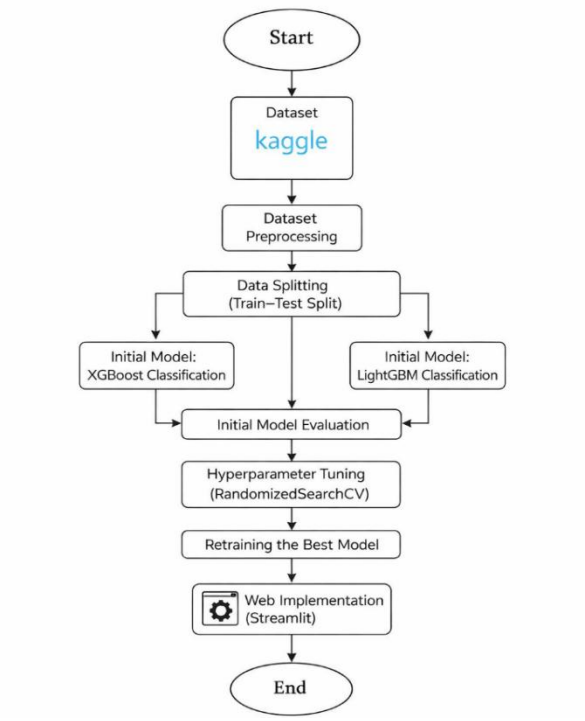

Figure 1. Research Flowchart

Research Methodology

The current study uses a quantitative approach to learn through classification using supervised learning algorithms. This statistical method of using numbers and data helps to ensure an evaluation of the algorithms' performance by quantitatively determining how successful they are using concrete evaluation criteria. Furthermore, there is significant

opportunity for the field of machine learning within the medical field to aid with the early detection of various forms of diseases. This includes treating the early stages of diabetes mellitus.

Additionally, ensemble methods of supervised learning, such as using multiple models within a single algorithm, have often proved to enhance stability and prediction accuracy. With this, ensemble methods have become very common within Indonesian research on healthcare.

Data Collection

For this specific task, the data can be retrieved from the Kaggle platform that originated from the National Institute of Diabetes and Digestive and Kidney Diseases. The data set had 768 instances with 8 specific features that are health-related, such as Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, and Outcome, wherein Outcome is used specifically to identify whether or not someone has diabetes [16].

Data Preprocessing

In the initial preprocessing stage, the dataset structure and attribute types were examined to ensure that all numerical variables were properly formatted. Invalid values were then identified, particularly zero values in several clinical attributes such as Glucose, BloodPressure, SkinThickness, Insulin, and BMI. From a medical perspective, these zero values do not represent realistic conditions and were therefore treated as missing values.

For handling missing values, median imputation technique was used since it is less affected by outliers and can be used on data with skewed distributions such as clinical data. In order to prevent data leakage, the imputation parameters were calculated using only the training set and applied on the testing set. Additionally, the feature scaling was done using the StandardScaler, but only after splitting the data into the training and testing sets in order to ensure that scaling parameters are not influenced by the testing set. It is worth noting that the chosen tree-based models (XGBoost and LightGBM) are usually invariant to feature scaling, but normalization was still applied for consistency.

The missing values were handled using appropriate statistical imputation techniques to prevent significant data reduction while preserving the overall data distribution. In addition, data normalization was performed to standardize the scale across variables, thereby preventing certain features from dominating the model training process. This preprocessing stage was designed to enhance input data quality, reduce potential bias, and support the optimal performance of the XGBoost and LightGBM classification algorithms in the subsequent modeling phase [17].

Data Splitting (Data Training and Data Testing)

Data splitting is the process of dividing the dataset into training and testing sets to enable model training and unbiased evaluation of its performance. This step aims to train the model on a portion of the data and evaluate its generalization capability on previously unseen data.

In this study, the 768 data were split using an 80% proportion for training and 20% for testing, a commonly applied ratio in machine learning research to balance model training and performance evaluation. Furthermore, stratified sampling based on the target variable (Outcome) was employed to maintain balanced proportions of diabetes and non-

diabetes cases in both the training and testing sets. Nevertheless, using only one train-test split may result in bias in the assessment since the results will be dependent upon the way the data is split. In that regard, the use of an alternative evaluation strategy, for instance, cross-validation, could help reduce the variance. In that regard, this problem is recognized in the current research paper.

Initial Model Implementation

The models were trained on the prepared training dataset using the default parameters of each algorithm, without any optimization process. The baseline models were implemented using two gradient boosting based classification algorithms such as XGBoost and LightGBM. This approach aimed to establish baseline performance benchmarks for subsequent comparison [18].

In detail, the baseline parameters are the random state parameter value was set to be 42 for the sake of repeatability, the accuracy measure served as the model's performance evaluation metric such as precision, recall, and F1-score measures were reported. The decision threshold was chosen to be 0.5 for both XGBoost and LightGBM.

The implementation of these initial models allows for the assessment of each algorithm's preliminary performance in classifying diabetes risk and provides a reference point for evaluating performance improvements after hyperparameter tuning.

XGBoost Model

One of the most widely used machine learning algorithms in dealing with classification and regression problems under the Gradient Boosting Decision Tree (GBDT) model is XGBoost. Although it is well-known for its stable and reliable performance, there have been some limitations identified by various researchers. One of the major challenges associated with XGBoost is related to the large number of hyperparameters, which makes it computationally intensive to determine optimal values [19].

As an extension of the GBDT model, XGBoost has included a regularization term in its objective function. This helps in controlling model complexity, thus avoiding the risk of overfitting. The objective function of XGBoost is designed in such a way that it can be represented as follows:

$$O = \sum_{i=1}^{n} L\left(y_i, F(x_i)\right) + \sum_{k=1}^{t} R(f_k) + C \quad (1)$$

where $L\left(y_i, F(x_i)\right)$ denotes the loss function, $R(f_k)$ represents the regularization term at iteration k and C is a constant. The regularization term $R(f_k)$ can be expressed as:

$$R(f_k) = \propto H + \frac{1}{2}\eta \sum_{j=1}^{H} w_j^2 \quad (2)$$

with $\propto$ denotes the leaf complexity parameter, H represents the number of leaves, η is the penalty parameter and w_j^2 corresponds to the output weight of each leaf node.

LightGBM Model

LightGBM is a machine learning based on the Gradient Boosting Decision Tree (GBDT) model. Its primary aim is to improve its efficiency in terms of computation, thereby making it more effective in solving large-scale prediction problems [14].

LightGBM has many advantages, such as faster training speed, memory usage, and predictive results, as well as large-scale data processing, learning, and GPU support, making it effective in solving many machine learning problems, such as classification, regression, and ranking [14].

Assume a raw dataset with observations $N = \{1,2,\dots,n\}$ and a LightGBM that generate $T = \{1,2,\dots,n\}$ trees. After iteration t, the final prediction is obtained by updating the previous prediction $(1-t)$ with the contribution of the newly constructed tree at iteration t. The iterative process can be expressed as follows :

$$y_i^{(t)} = y_i^{(t-1)} + f_i(x_i) \quad (3)$$

where $y_i^{(t)}$ denotes the predicted value for the i-th observation at iteration t, $y_i^{(t-1)}$ represents the prediction from the previous iteration, and $f_i(x_i)$ is the newly built tree model at iteration ttt.

Initial Model Performance Evaluation

The evaluation of the initial models was conducted to assess the classification performance of XGBoost and LightGBM prior to hyperparameter optimization. The metrics included accuracy, precision, recall, F1-score, and AUC-ROC for the evaluation of the classification potential of the models. This study is focused on recall, as it demonstrates its ability to identify individuals with diabetes and reduce false negatives, which are relevant to its clinical utilities. In genera based on [15], the model evaluation can be expressed as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall\ (Sensitivity) = \frac{TP}{TP+FN} \quad (6)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (7)$$

$$\text{AUC} = \int_0^1 TPR(FPR)dFPR \quad (8)$$

where the True Positive Rate (TPR) and False Positive Rate (FPR) serve as the basis for constructing the AUC value, which reflects the model's ability to distinguish between positive and negative cases across various threshold levels. An AUC value close to 1.0 indicates excellent discriminative ability, whereas a value near 0.5 suggests that the model performs no better than random guessing.

Hyperparameter Tuning

The hyperparameter tuning stage was conducted to improve the classification performance of the XGBoost and LightGBM models by identifying the optimal combination of parameter values. In this study, parameter optimization was performed using the RandomizedSearchCV method, which selects parameter combinations randomly from a predefined search space.

The tuning process incorporated a cross-validation scheme applied to the training data to ensure that the selected hyperparameters provide strong generalization performance and reduce the risk of overfitting [7]. Mathematically, the optimization process can be formulated as follows:

$$\theta^* = \arg\max_{\theta \in \Theta_s} \frac{1}{K} \sum_{k=1}^{K} M\left(f_\theta^{(k)}\right) \quad (9)$$

where:

- θ^* denotes the optimal hyperparameter combination;
- $\theta \in \Theta_s$ represents a set of hyperparameters randomly selected from the predefined search space;
- K indicates the number of folds in the cross-validation procedure;

- $f_\theta^{(k)}$ refers to the model with hyperparameters θ trained on the training data and evaluated on the k-th fold;
- M denotes the evaluation metric used, such as recall, AUC–ROC, or accuracy.

Hyperparameters search space is defined as follows. In case of XGBoost, the following parameters and values have been considered: n_estimators (100, 200), max_depth (3, 4), and learning_rate (0.05, 0.1). In the case of LightGBM, the following parameters and values have been considered: n_estimators (100, 200), num_leaves (31, 63), and learning_rate (0.05, 0.1).
RandomizedSearchCV algorithm's parameters are defined as follows. Number of iterations is equal to 4, cross-validation configuration includes three-folds for training data, scoring criteria include roc_auc metric, and random_state is set to 42.

Final Retraining of the Best Model

The retraining of the best model was done after the identification of the best model based on the results of the RandomizedSearchCV tuning algorithm. At this stage, the models based on the XG Boost algorithm and the LightGBM algorithm were retrained only using the training dataset with the identified best parameters, without including any data from the test set. Based on [20], this ensured the development of a model with enhanced performance.

The aim of this phase was to guarantee that the model fully incorporates all the essential patterns from the data, ensuring enhanced generalization for the model. The test set was kept fully independent and was not used during the training or tuning process, and it was only utilized for the final evaluation of the model. The model developed during this phase was used to make predictions, becoming the best model for the purposes of the final evaluation and for the development of the diabetes risk prediction system.

Final Model Evaluation

The final model evaluation was conducted to evaluate the performance of the retrained models with the optimized hyperparameters. At this stage, both models, i.e., XGBoost and LightGBM, were evaluated using the same test dataset to assess their performance with respect to the results obtained during the initial model evaluation stage.

The performance of the models was assessed using the same parameters, i.e., accuracy, precision, recall, F1 measure, and AUC-ROC, which helped to make an objective assessment of the improvements made during the model development stage prior to the hyperparameter tuning phase.

Model Implementation into a Web-Based System

The implementation stage of the proposed system focused on integrating the best model from the final model evaluation stage into a web-based system to enable users to access the diabetes risk prediction system. For this study, the model was implemented into a web-based system to enable users to interact with the system.

The system was developed using Streamlit [21], [22], a Python-based web application development library, which allows developers to create web applications for data

visualization and model-based predictions. The developed web application allows users to input their clinical parameters, i.e., Glucose, BMI, and Age, which are used to generate the diabetes risk automatically.

Results and Discussions

In this study, the technique used was a secondary data approach with eight independent variables: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age, and one dependent variable: Outcome. The source of the data was the Kaggle website.

In the initial preprocessing of the data, the structure of the dataset and the type of the variables were checked to ensure proper formatting of the entire dataset, especially the numerical variables. The dataset was also checked for any impossible values, especially the values equal to zero, as they appeared in the variables Glucose, BloodPressure, SkinThickness, Insulin, and BMI. After dealing with the problems mentioned above, the feature scaling process was performed to normalize the values of the numeric features.

Splitting of the dataset was performed and the data was divided into a training set and a testing set in the ratio of 80 over 20. The evaluation of the model was performed by comparing the performance of XGBoost model and LightGBM model. The models was evaluated by metrics of accuracy, precision, recall, F1-score, and AUC-ROC. The results were analyzed to show the performance of the models and the effectiveness of the models in predicting the risk of a person suffering from diabetes mellitus. Moreover, a clinical interpretation aspect was taken into account to ensure that the results generated by the model will be consistent with the medical facts, especially when analyzing significant factors like glucose level, BMI, and age in connection with the risk of developing diabetes mellitus.

The best model was chosen and used as the basis of creating a web application for prediction using Streamlit. In the discussion, it is essential to emphasize that future studies need to include the SHAP technique for explanation to improve clinical transparency.

Evaluation Results of the XGBoost Model Before Hyperparameter Tuning

At this stage, the initial implementation of the XGBoost algorithm was carried out by splitting the dataset into 80% training data and 20% testing data. The model was constructed using the XGBClassifier() function from the XGBoost library.

Model evaluation was performed using eval_metric='logloss' and the parameter random_state=42 was specified to ensure reproducibility and consistent results. The model was then trained on the training dataset (X_train and y_train) and its performance was evaluated on the testing dataset (X_test and y_test).

Table 1. Evaluation Results of the XGBoost Model Without Hyperparameter Tuning

Evaluation Metric	Value
Accuracy	73,4%
Precision	62,3%
Recall	61,1%
F1-Score	61,7%
ROC AUC	80,5%

Based on Table 1. the initial XGBoost model without hyperparameter tuning achieved an accuracy of 73.4% which indicates that the model correctly predicted approximately three out of four cases. In terms of precision the model obtained a value of 62.3% meaning that when the model predicted a positive outcome about 62 out of 100 predictions were correct. Meanwhile a recall of 61.1% indicates that the model was able to identify approximately 61% of all actual positive cases while the F1-score of 61.7% reflects a balance between precision and recall so the model can be considered reasonably consistent.

Furthermore the ROC–AUC value of 80.5% indicates that the model demonstrates good ability in distinguishing between positive and negative classes.

Evaluation Results of the LightGBM Model Before Hyperparameter Tuning

The LightGBM model was trained using the LGBMClassifier function from the LightGBM library. The training procedure followed the same approach applied to XGBoost where the dataset was divided into 80 percent training data and 20 percent testing data. After the model was constructed its performance was evaluated using five main metrics which include accuracy precision recall F1-score and AUC–ROC.

Table 2. Evaluation Results of the LightGBM Model Without Hyperparameter Tuning

Evaluation Metric	Value
Accuracy	74,7%
Precision	65,3%
Recall	59,3%
F1-Score	62,1%
ROC AUC	81,8%

Based on Table 2, the LightGBM model without hyperparameter tuning achieved an accuracy of 74.7% which indicates that the model correctly classifies about three out of four cases. In terms of precision the model obtained a value of 65.3% which means that when the model predicted a positive outcome about 65 out of 100 predictions were truly positive.

At the same time, a recall of 59.3% indicates that the model identified approximately 59% of all actual positive cases while the F1-score of 62.1% demonstrates a balance between precision and recall and therefore reflects relatively consistent performance. In addition the ROC–AUC value of 81.8% shows that the model has good discriminative ability in distinguishing between positive and negative classes.

Evaluation Results of the XGBoost Model After Hyperparameter Tuning Using RandomizedSearchCV

The initial XGBoost model was optimized using the RandomizedSearchCV method to identify the most appropriate combination of hyperparameters. The tuning process focused on several key parameters including the number of estimators set to 300 the learning rate set to 0.1.

Table 3. Evaluation Results of the XGBoost Model with RandomizedSearchCV

Evaluation Metric	Value
Accuracy	75,3%
Precision	66,7%

Recall	59,3%
F1-Score	62,7%
ROC AUC	82,7%

Based on Table 3 the model achieved an accuracy of 75.3% which indicates that it correctly predicted nearly three out of four cases. In terms of precision the model obtained a value of 66.7% which means that when the model predicted a positive outcome about 67 out of 100 predictions were truly positive. Meanwhile a recall of 59.3% indicates that the model identified nearly 59% of all actual positive cases.

The F1-score of 62.7% reflects a fairly good balance between precision and recall and therefore suggests that the model demonstrates stable predictive performance. In addition the ROC–AUC value of 82.7% indicates that the model has strong discriminative ability in distinguishing between positive and negative classes and can reliably differentiate between the two categories.

Evaluation Results of the LightGBM Model After Hyperparameter Tuning Using RandomizedSearchCV

The initial LightGBM model was optimized using the RandomizedSearchCV method to identify the most appropriate combination of hyperparameters. The tuning process focused on several key parameters including the number of estimators set to 300 the learning rate set to 0.1.

Table 4. Evaluation Results of the LightGBM Model with RandomizedSearchCV

Evaluation Metric	Value
Accuracy	77,3%
Precision	71,1%
Recall	59,3%
F1-Score	64,6%
ROC AUC	83,0%

Based on Table 4., the accuracy of the tuned LightGBM model stands at 77.3%, meaning the model correctly classifies nearly three out of every four cases. The precision of the model is at 71.1%, meaning the model correctly makes around 71 out of every 100 positive predictions. The recall is at 59.3%, meaning the model correctly classifies around 59% of all positive cases.

Although the overall hrl_F1 is only 64.6%, the level of stability between precision and recall increases over the baseline, indicating more dependable predictions. The ROC-AUC at 83.0% indicates that the model is doing a decent job of differentiating between positive and negative data points.

In summary, after tuning, the parameters of LightGBM improve significantly, mainly in precision and F1 score, making its results accurate.

Comparison of XGBoost and LightGBM Models to Determine the Best Model

The best model was determined based on the performance evaluation results of the XGBoost and LightGBM models. Each model was compared using several key evaluation metrics to assess the level of accuracy, prediction precision, and the ability to recognize

patterns in the data. The model that demonstrated the most optimal and consistent performance was then selected as the best model to be used in the subsequent stage.

Table 5. Comparison of the Evaluation Results of XGBoost and LightGBM Models

Method	Accuracy	Precision	Recall	F1 Score	ROC AUC
XGBoost	75,3%	66,7%	59,3%	62,7%	82,7%
LightGBM	77,3%	71,1%	59,3%	64,6%	83,0%

Based on Table 5, it is known that there was an improvement in the performance of both the XGBoost and LightGBM models after the tuning process.The XGBoost results indicate that the accuracy is 75.3%, while the F1 score is 62.7%. In the case of LightGBM, we can notice a significant improvement compared to XGBoost because the accuracy is 77.3%.

The precision of the LightGBM model was also higher at 71.1%, whereas the recall of both models was the same at 59.3%, indicating that their ability to capture actual positive cases was equivalent. The ROC AUC value of the LightGBM model was slightly higher at 83.0%, demonstrating a better ability to distinguish between positive and negative classes. The comparison results of the two models' performance are also visualized in Figure 2 as follows.

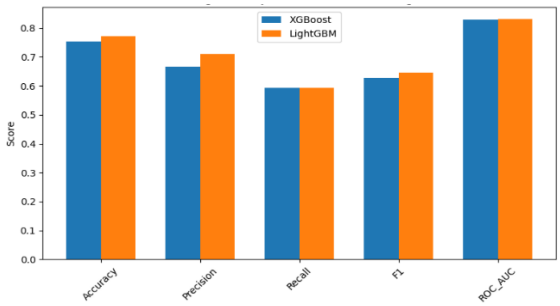

Figure 2. Performance Comparison of XGBoost and LightGBM Models

Based on Figure 2. we know that the performance of the LightGBM model is superior to that of the XGBoost model. Overall, the tuned LightGBM model was selected as the best model because it demonstrated better and more stable overall performance for prediction. Furthermore, the best model will be used for further analysis.

3.6 Confusion Matrix for the LightGBM Model

The prediction results of the selected best-performing model, LightGBM, are presented in the form of a confusion matrix. This visualization facilitates a clear understanding of the distribution of correct and incorrect predictions and highlights the classification error patterns for patients with diabetes and non-diabetes. Furthermore, by analyzing the confusion matrix, the model's ability to distinguish between the two classes can be systematically evaluated and the most frequent types of prediction errors can be identified.

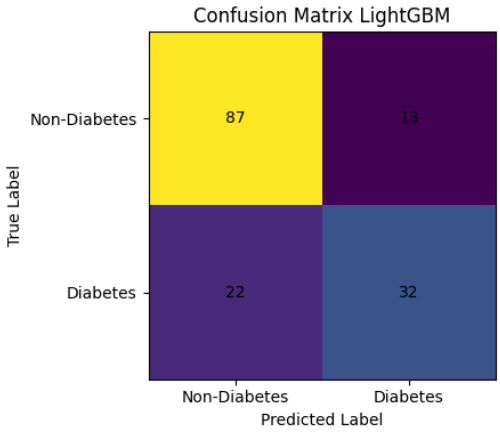

Figure 3. Confusion Matrix of The Best Model

Based on the confusion matrix presented in Figure 3, the model was able to classify 87 non-diabetes patients and 32 diabetes patients accurately. But the model also misclassified 13 non-diabetes patients as diabetes (false positives) and 22 diabetes patients as non-diabetes (false negatives). Despite the high effectiveness of the model in terms of classification, the appearance of 22 false negatives suggests that certain cases of diabetes are not being identified. This is certainly a significant disadvantage since failure to diagnose a case could delay treatment for the patient.Nevertheless, the model still provides a solid predictive baseline, and its performance can be further enhanced in future research.

Web-Based Model Implementation

The best model obtained through the training and tuning process is kept in a *.joblib* file to maximize the efficiency of storage and minimize the time required to deploy the model. The model is then integrated into a web-based application developed using Python with the help of the Streamlit library. This allows the development of a web-based platform for the prediction of the risk of developing diabetes mellitus. In the web-based application, users are required to input a number of important factors for the development of the disease, including the number of Pregnancies, Glucose level in mg/dL, Blood Pressure, Skin Thickness, Insulin, BMI as a measurement of adipose tissue, Diabetes Pedigree Function as a measurement of genetic risk, and Age. The user then proceeds to the prediction step by clicking the "Predict Risk" button after all the required information has been entered. The web-based application then proceeds to the pre-processing step as was done during the training process, and the pre-trained model kept in the *.joblib* file is used for the prediction with the help of the LightGBM algorithm. The predicted result will be a binary outcome of either "At Risk of Diabetes" or "Not at Risk of Diabetes" along with the probability of the outcome, depicting the confidence level of the model.

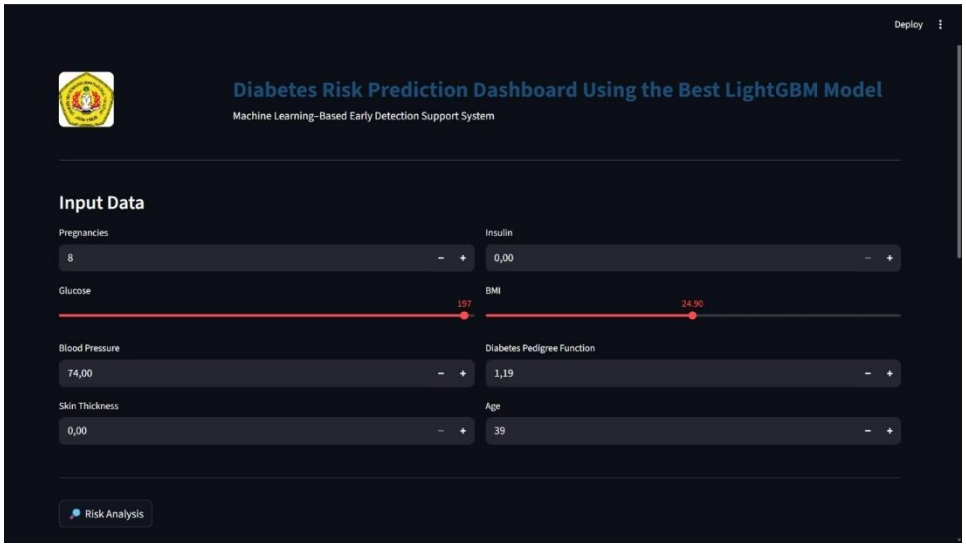

Figure 4. Diabetes Prediction Website Interface (Input Page)

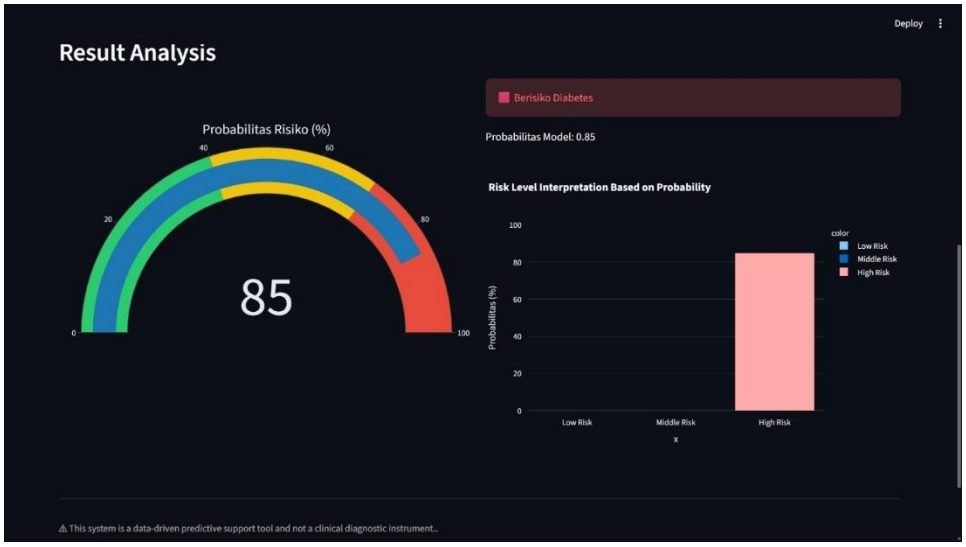

Figure 5. Diabetes Prediction Website Interface (Output Page)

In order to check the precision of the prediction system designed in the project, the evaluation was carried out on six different samples of the test dataset using the LightGBM algorithm with a split ratio of 80:20 after parameter tuning. The purpose of the evaluation was to check the extent of agreement between the predictions made by the system and the actual labels in the test dataset, thereby measuring the efficiency of the system in accurately classifying the data.

Table 6. System Predictions

Features Used	Expected Prediction	System Output	Conclusion
Pregnancies: 8 Glucose: 105 Blood Pressure: 100 Skin Thickness: 36 Insulin: 0 BMI: 43.3 DPF: 0.239 Age: 45	Diabetes	Non-Diabetes (Prob: 0.13)	The system failed to predict correctly according to the actual label

Pregnancies: 8 Glucose: 197 Blood Pressure: 74 Skin Thickness: 0 Insulin: 0 BMI: 25.9 DPF: 1.191 Age: 39	Diabetes	Diabetes (Prob: 0.85)	The system successfully predicted correctly
Pregnancies: 2 Glucose: 100 Blood Pressure: 64 Skin Thickness: 23 Insulin: 0 BMI: 29.7 DPF: 0.368 Age: 21	Non-Diabetes	Non-Diabetes (Prob: 0.11)	The system successfully predicted correctly
Pregnancies: 0 Glucose: 113 Blood Pressure: 76 Skin Thickness: 0 Insulin: 0 BMI: 33.3 DPF: 0.278 Age: 23	Non-Diabetes	Diabetes (Prob: 0.12)	The system failed to predict correctly according to the actual label
Pregnancies: 0 Glucose: 137 Blood Pressure: 70 Skin Thickness: 38 Insulin: 0 BMI: 33.2 DPF: 0.17 Age: 22	Non-Diabetes	Non-Diabetes (Prob: 0.42)	The system successfully predicted correctly
Pregnancies: 0 Glucose: 131 Blood Pressure: 0 Skin Thickness: 0 Insulin: 0 BMI: 43.2 DPF: 0.27 Age: 26	Diabetes	Diabetes (Prob: 0.84)	The system successfully predicted correctly

The testing was carried out on six samples of the test data set using the LightGBM model after parameter tuning with an 80:20 split of the data. Out of the results obtained, four were consistent with the actual labels, while two were not. The accuracy of the model was 77.3%, which shows that there is an incomplete level of consistency in the sample set. However, the ROC AUC of 83.0% shows that there is a high level of discrimination between people at risk of diabetes mellitus and those not at risk. Overall, the results show that the system has high potential as an interactive tool for early detection of diabetes mellitus.

Conclusion

In conclusion, based on the research procedures and empirical findings, the LightGBM model demonstrates superior performance compared with XGBoost in classifying diabetes mellitus cases. LightGBM achieved a higher accuracy of 77.3% and a precision of 71.1%, while both models obtained the same recall value of 59.3%. Moreover,

the F1-score of 64.6% and the ROC-AUC of 83.0% indicate that LightGBM provides a more balanced prediction performance and stronger class discrimination ability. For these reasons, LightGBM was selected as the final model and implemented in a web-based application using Streamlit. The implementation through Streamlit results in a responsive and user-friendly system that can be easily operated. This application enables non-technical users to conduct an initial assessment of diabetes risk in a simple and informative manner without requiring advanced technical expertise. In addition, the interactive interface, efficient data input process, and real-time prediction output enhance usability and practical value.

Despite the model shows promising performance, its recall (59.3%) indicates that some diabetes cases may be missed, limiting its effectiveness for early detection. Futhermore, the use of a limited dataset highlights the need for further validation on larger and more diverse populations to ensure robustness. Overall, this study highlights the potential of machine learning based web applications to support early detection of diabetes mellitus. Future research should focus on exploring ensemble or deep learning approaches to further improve predictive performance and optimize early detection capabilities.

References

[1] Khurin, I. Wahyuni, A. A. Prayitno, and Y. I. Wibowo, "Efektivitas Edukasi Pasien Diabetes Mellitus Tipe 2 Terhadap Pengetahuan dan Kontrol Glikemik Rawat Jalan di RS Anwar Medika," *Jurnal Pharmascience*, vol. 06, no. 01, pp. 1–9, 2019, [Online]. Available: https://ppjp.ulm.ac.id/journal/index.php/pharmascience

[2] Gojka. Roglic, *Global report on diabetes*. World Health Organization, 2016.

[3] "IDF Diabetes Atlas 10th edition 537 million people worldwide have diabetes." [Online]. Available: www.diabetesatlas.org

[4] B. Kebijakan Pembangunan, K. Kementerian, and K. Ri, "Dalam Angka Tim Penyusun Ski 2023 Dalam Angka Kementerian Kesehatan Republik Indonesia."

[5] I. F. Amri, F. H. N. Rohim, M. I. Nurul Azka, and M. S. Rakhmawati, "Analysis of Gallstone Incidence Factors Using a Logistic Regression Model.," *EKSAKTA: Journal of Sciences and Data Analysis*, Oct. 2025, doi: 10.20885/eksakta.vol6.iss2.art3.

[6] A. Purwo Wicaksono and I. Muhimmah, "Eksakta: Jurnal Imu-Ilmu MIPA Application Of Logistic Regression In Analysis Of Factors That Affect Implementation Of Electronic Medical Record".

[7] M. Bhavsar and M. Patel, "Predicting Cardiovascular Disease with Machine Learning Algorithms: A Review," *ITM Web of Conferences*, vol. 65, p. 03011, 2024, doi: 10.1051/itmconf/20246503011.

[8] R. A. Salasa, H. Rahman, and D. Andiani, "Faktor Risiko Diabetes Mellitus Tipe 2 Pada Populasi Asia: A systematic Review," *Jurnal BIOSAINSTEK*, vol. 1, no. 1, 2019, doi: 10.52046/biosainstek.v1i01.306.95-107.

[9] D. Zhang and Y. Gong, "The Comparison of LightGBM and XGBoost Coupling Factor Analysis and Prediagnosis of Acute Liver Failure," *IEEE Access*, vol. 8, pp. 220990–221003, 2020, doi: 10.1109/ACCESS.2020.3042848.

[10] A. Dwi Novika and A. Mujhidi, "Cost-Sensitive Learning with LightGBM for Class Imbalance in Intrusion Detection Systems," vol. 7, no. 2, pp. 147–153, 2025, doi: 10.21512/emacsjournal.v6.

[11] A. Brahmandjati, A. Mizwar, A. Rahim, and F. Asharudin, "Optimasi Prediksi Diabetes Dengan Algoritma XGBoost Dan Teknik Preprocessing Data." [Online]. Available: https://www.kaggle.com/datasets/mathchi/diabetes-data-set,

[12] X. Y. Fu *et al.*, "Development and validation of LightGBM algorithm for optimizing of Helicobacter pylori antibody during the minimum living guarantee crowd based gastric cancer screening program in Taizhou, China," *Prev. Med. (Baltim).*, vol. 174, Sep. 2023, doi: 10.1016/j.ypmed.2023.107605.

[13] F. Caroline and N. Rachmat, "Comparison of XGBoost and LightGBM Algorithms in Predicting Heart Disease," *Brilliance: Research of Artificial Intelligence*, vol. 5, no. 2, pp. 1232–1239, Dec. 2025, doi: 10.47709/brilliance.v5i2.7505.

[14] P. Septiana Rizky, R. Haiban Hirzi, U. Hidayaturrohman, U. Hamzanwadi Selong Jl TGKH Muhammad Zainuddin Abdul Madjid Pancor, and L. Timur, "Perbandingan Metode LightGBM dan

XGBoost dalam Menangani Data dengan Kelas Tidak Seimbang," 2022. [Online]. Available: www.unipasby.ac.id
[15] E. R. Susanto and A. Cahyana, "Penerapan Algoritma XGBoost untuk Prediksi Diabetes: Analisis Confusion Matrix dan ROC Curve," *Fountain of Informatics Journal*, vol. 10, no. 1, pp. 40–50, May 2025, doi: 10.21111/fij.v10i1.14311.
[16] J. Khatib Sulaiman, D. Wijayanto, B. Pilu Hartato, and U. Amikom Yogyakarta, "Analisis Perbandingan Performa Algoritma XGBoost dan LightGBM pada Klasifikasi Kanker Payudara," *Indonesian Journal of Computer Science*.
[17] W. Zhou, "analysis of Diabetes Prediction Models Based on XgBoost and LightgBM".
[18] R. G. Wardhana, G. Wang, and F. Sibuea, "PENERAPAN MACHINE LEARNING DALAM PREDIKSI TINGKAT KASUS PENYAKIT DI INDONESIA," 2023.
[19] O. P. Handayani, Purwono, I. A. Ashari, and R. Ardianto, "Systematic Literature Review: Penerapan Machine Learning dalam Diagnosis dan Prediksi Penyakit Diabetes," *Komputa : Jurnal Ilmiah Komputer dan Informatika*, vol. 14, no. 2, pp. 108–118, Nov. 2025, doi: 10.34010/komputa.v14i2.16642.
[20] W. Liang, S. Luo, G. Zhao, and H. Wu, "Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms," *Mathematics*, vol. 8, no. 5, May 2020, doi: 10.3390/MATH8050765.
[21] A. Putranto, N. L. Azizah, I. Ratna, and I. Astutik, "Web-based Heart Disease Prediction System Using SVM Method and Streamlit Framework [Sistem Prediksi Penyakit Jantung Berbasis Web Menggunakan Metode SVM dan Framework Streamlit]." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/heart+disease
[22] M. Al Hafidz and P. M. Effendi, "Aplikasi Penentuan Kebutuhan Pelatihan Berbasis Kompetensi Untuk Peningkatan Kinerja Staf Analis Laboratorium," *Teknika*, vol. 12, no. 2, pp. 129–137, Jun. 2023, doi: 10.34148/teknika.v12i2.622.