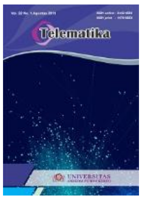

Available online at :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

Telematika

Accreditated SINTA "2" Kemenristek/BRIN, No. 85/M/KPT/2020

UNIVERSITAS AMIKOM PURWOKERTO

Optimization of the XGBoost Algorithm for Predicting Stroke Patient Care Outcomes

Puji Lestari[1], Imam Tahyudin[2], Ades Tikaningsih[3], Ade Nurhopipah[4]

[1,2] Department of Information System, Computer Science Faculty
[3,4] Department of Informatics, Computer Science Faculty
Universitas Amikom Purwokerto, Banyumas, Indonesia
E-mail: puji24318@gmail.com[1], imam.tahyudin@amikompurwokerto.ac.id[2], atikansh20@gmail.com[3], adenurhopipah@amikompurwokerto.ac.id[4]

ARTICLE INFO

History of the article:
Received December 7, 2023
Revised August 6, 2024
Accepted August 19, 2024

Keywords:
XGBoost
Machine Learning
Web-Based Application
Extreme Programming (XP)
Python Flask

Correspondece:
E-mail:
imam.tahyudin@amikompurwokerto.ac.id

ABSTRACT

Stroke is a critical health issue in Indonesia, contributing to high mortality rates. At Banyumas District Hospital, stroke is the fourth most common condition, presenting significant challenges in both clinical care and financial management. The purpose of this study is to enhance the quality of services and optimize treatment costs for stroke patients by developing a predictive model using the XGBoost algorithm. This study employs the XGBoost algorithm to develop predictive models, which are then implemented within a web-based machine learning application using the Python Flask framework. The models predict patient mortality and hospitalization duration. The results indicate that the XGBoost algorithm predicts patient mortality with 86% accuracy and hospitalization duration with 82% accuracy. The developed application significantly enhances care quality and resource management at Banyumas District Hospital by providing accurate predictions to support healthcare decision-making. The use of this application can significantly improve the management of stroke patient care at Banyumas District Hospital, thereby maximizing service quality and optimizing treatment costs. The application helps make better healthcare decisions by using accurate predictive modeling. This makes it easier to use resources wisely and get medical help when it's needed, which leads to better patient outcomes.

INTRODUCTION

Hospitals, as institutions of health care, have an important role to play in efforts to maintain public health and are expected to have the capacity to provide effective and efficient services (Putri & Sonia, 2021). A stroke is a clinical condition that occurs when there is local or generalized brain damage occurring over a period of more than 24 hours, can even lead to death, and is not caused by other factors than vascular problems (Mahmoud Hewedi et al., 2019). According to data from the World Stroke Organization, there are approximately 13.7 million strokes and about 5.5 million deaths each year. Currently, stroke ranks as the third leading cause of death globally, following coronary heart disease and cancer (Maskuri et al., 2022). Stroke, as a neurological condition caused by bleeding or congestion in a brain area that can lead to defects or even death, requires special attention in its treatment because of its impact that may threaten the life and quality of life of the patient (Setiawan, 2020).

In the medical world, strokes are divided into two main types, namely hemorrhagic strokes (blooding) and ischemic strokes (infra) (Indarto et al., 2020). Hemorrhagic strokes generally have a high mortality rate, while infrasound strokes have two possible outcomes, namely that the potential patient dies and the patient survives but requires long-term treatment due to residual symptoms. Although infrared strokes have a lower mortality rate than hemorrhagic strokes, a deeper understanding of the treatment of stroke patients is not only a clinical challenge but also has significant financial implications. Maintaining efficient and effective stroke care is becoming crucial to maximizing service quality and controlling treatment costs. Maintaining efficient and effective stroke patient care is becoming crucial to maximizing the quality of service. In addition to optimizing the quality of service, there is equal importance in controlling the cost of care for infarction stroke patients. Patients with residual symptoms requiring long-term care in hospitals can be a significant financial burden on the healthcare system, emphasizing the need for comprehensive management strategies within the healthcare system.

In this digital age, the use of information technology and web-based applications has become an effective tool in healthcare, especially in improving patient management and predicting treatment outcomes (Sittig et al., 2020). Drawing upon the aforementioned challenges, this research endeavors to optimize the quality of service and control the costs of treating stroke patients. In pursuit of this objective, a prediction model utilizing the XGBoost algorithm has been developed. This algorithm was selected due to its well-documented effectiveness in handling complex datasets and its ability to provide accurate predictions. Subsequently, a website-based machine learning application was created to implement this algorithm, enabling the prediction of potential deaths and length of stay for stroke patients at RSUD Banyumas.

Several earlier studies have provided an overview related to the prediction of strokes and the length of hospitalization for stroke patients. These studies cover significant aspects such as the methods used for analysis, key findings, and their potential impact on the treatment of stroke patients. For instance, one study successfully developed a web-based machine learning application for breast cancer prediction. This study utilized the Flask Python web framework and the XGBoost analytical model for classification. The resulting application demonstrated high accuracy in detecting breast cancer, achieving an accuracy of 98.24% on the dataset, thereby aiding early diagnosis and more effective treatment (Pawar et al., 2022).

Another study using the XGBoost algorithm model for diabetes detection reported an accuracy of 99% and a recall of 100% (Paleczek et al., 2021). In research focused on predicting long-term hospital stays for diabetic and hypertensive patients, the XGBoost model was identified as the best algorithm, with an R2 of 0.633, RMSE of 0.386, and MAE of 0.123 (Barsasella et al., 2022). Additionally, studies on predicting mortality in ICU patients with acute kidney injury demonstrated that the XGBoost algorithm performed exceptionally well (Liu et al., 2021). Realizing that the models used in previous studies have long response and retention times, as well as the limited focus of previous research solely on predicting length of stay and mortality among diabetes and hypertension patients without considering specific management efforts, and the need to develop predictive models for mortality capable of handling small and unbalanced datasets, which poses a challenge for single-machine learning approaches, are important gaps identified in the existing literature. The difference between this research and previous research is that previous studies have mainly focused on other diseases such as diabetes and breast cancer, or on general mortality prediction. However, this research specifically targets stroke patients and aims to improve the prediction accuracy by addressing the limitations of previous models, particularly in dealing with unbalanced datasets and optimizing interventions in stroke care management.

In addition to using a machine learning algorithm model, the study has also used the Python Flask framework in web-based application development. One of the previous studies using the Python Flask

Framework has been shown in research with the aim of building a web-based machine learning application for predicting chronic kidney cancer (Uddin et al., 2023). The use of the Flask Framework to integrate web applications with machine learning models has also been successfully implemented in previous research with the aim of conducting an early diagnosis of HIV disease (Saha et al., 2023). There is another study that has successfully implemented the Python Flask Framework, which aims to build a web-based machine learning application to diagnose liver failure (Amruthamathi & Devi, 2023). A website-based machine learning application for predicting the death and long-term hospitalization of stroke patients was created by implementing a machine learning algorithm model. The integration between websites and machine learning modeling is done using Python Flask. The application interface is created using HTML, CSS, and Bootstrap. System development method, namely Extreme Programming (XP), with system modeling designed using Unified Modeling Language (UML). The database management used in the creation of this application uses MySQL via a Web interface, namely phpMyAdmin.

The aim of this research is to develop a predictive model utilizing the XGBoost Classifier to accurately predict the potential for death and length of stay for stroke patients at Banyumas District Hospital. This model is then implemented in a web-based application to support better patient management and resource allocation. The novelty of this research is the development and integration of a machine learning model specifically tailored for predicting mortality and length of stay in stroke patients. The model’s implementation within a web-based application, using the Python Flask framework, represents a novel approach in the field of stroke patient management at Banyumas District Hospital. This research is an additional reference in the treatment of various diseases in Indonesia, especially stroke. The application, with the implementation of the website-based XGBoost algorithm developed in this research, designed to substantially enhance the quality of service for stroke patients at Banyumas District Hospital. The main contribution of this research lies in the development of a prediction model with high accuracy, implemented within web-based applications. This model enables Banyumas District Hospital to swiftly identify patients at high risk of death, facilitating timely medical interventions. Additionally, by providing accurate predictions, the application aids in resource allocation, minimizes unnecessary treatment duration, and enhances the overall efficiency of treatment cost management, thus contributing to the optimization of patient care.

RESEARCH METHODS

The steps of this research include data description, data preprocessing, model building with an Extreme Gradient Boosting (XGBoost) classifier, hyperparameter tuning, performance evaluation, and web development. The research methodology is organized in a workflow diagram as shown in Figure 1.

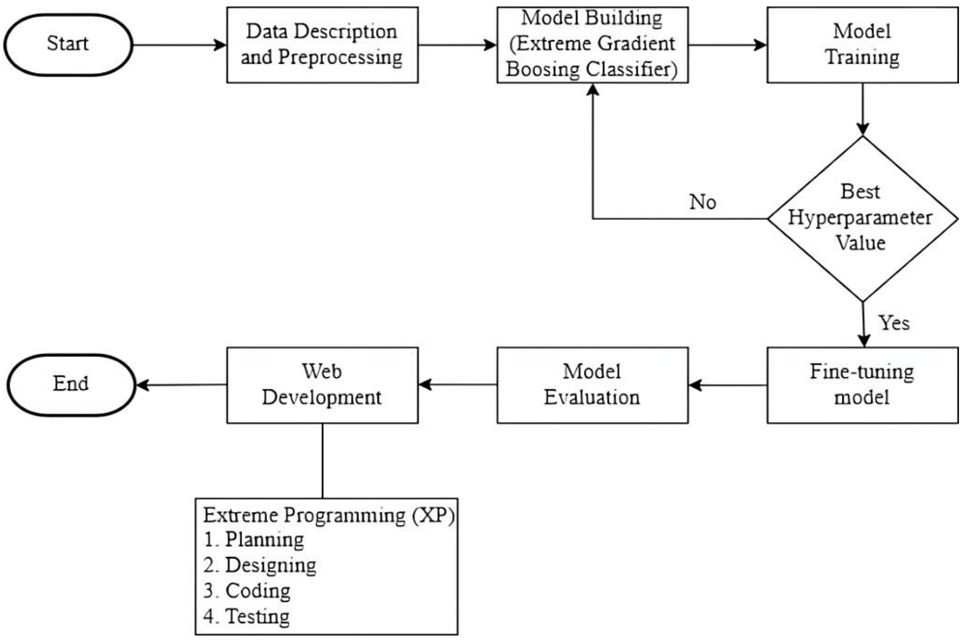

Figure 1. Research Method Flowchart

The detail of research construction of figure 1 is described detail as below:

1. **Data Description**

This study utilized data from 106 stroke patients treated at the Neurological Clinic of Banyumas Regional General Hospital (RSUD), Indonesia. Where the data consists of two categories, demographic data of stroke patients and medical record data of stroke patients. Working in conjunction with doctors and nurses, the research team at Banyumas Regional Hospital gathered data from January 2022 to May 2023. The data utilized in this study is original and can be accessed with the necessary permissions and ethical research approvals. A top priority is maintaining the confidentiality of patient identities, and measures have been taken to ensure data security and confidentiality. This research involves a series of data processing processes on the variables shown in Table 1.

Table 1. Stroke dataset variables

Categories	*Variables*	*Data Type*	*Description*
Pateinet demographic data	Age	Numerical (integer or float)	Represents the patient's age in years.
	Gender	Categorical (string)	Classification of the individual as male or female.
	Debtor	Categorical (string)	Status of a financial obligation or debt related to medical expenses.
Patient medical record data	History of Cardiovaskular Disease	Categorical (string)	Documentation of the patient's previous occurrences or experiences with cardiovascular diseases.
	Past medical history	Categorical (string)	Information about the patient's medical background, including previous illnesses.
	Previous stroke history	Categorical (string)	Details regarding any prior instances of stroke experienced by the patient.
	Hemoglobin (HB)	Numerical (integer or float)	Concentration of hemoglobin in the patient's blood.
	Hematokrit (HT)	Numerical (integer or float)	Proportion of blood that is composed of cellular components.
	Leukosit (LEU)	Numerical (integer or float)	Count or concentration of white blood cells in the patient's blood.
	Trombosit (TR)	Numerical (integer or float)	Count or concentration of blood platelets involved in clotting.

Categories	*Variables*	*Data Type*	*Description*
	Neutrophil-Lymphocyte Ratio (NLR)	Numerical (integer or float)	Ratio of neutrophils to lymphocytes in the blood, which can be indicative of inflammation or infection.
	Total Cholesterol (Chol Total)	Numerical (integer or float)	Total amount of cholesterol present in the patient's blood.
	High-Density Lipoprotein (HDL)	Numerical (integer or float)	Concentration of high-density lipoprotein in the blood.
	Triglisireda (TG)	Numerical (integer or float)	Levels of triglycerides, a form of fat, present in the patient's bloodstream.
	Low-Density Lipoprotein (LDL)	Numerical (integer or float)	Concentration of low-density lipoprotein in the blood.
	Stroke Location	Categorical (string)	Specific area or part of the brain affected by the stroke in a patient diagnosed with stroke disease.
	Patient Condition	Categorical (string)	Predicted overall health condition of the patient, including the likelihood of death and recovery, is based on consideration of various attributes.
	Length of Hospitalization	Numerical (integer)	Predicted duration of the patient's stay in the hospital, taking into account previous attributes and factors related to stroke.

2. Data Preprocessing

The data preprocessing process is carried out until it reaches a form that is ready to be used for analysis. The most important objective of this study is aimed at creating a predictive model addressing mortality and hospital stay duration in stroke patients, approached as a classification problem. In this research, there are 19 attributes that act as input variables and have been documented previously. The key target variable for predicting mortality in stroke patients is the 'patient condition' variable, which includes comprehensive information on the patient's health status, addressing both recovery and the risk of death. Similarly, the target variables for predicting the length of stay were categorized into two groups: standard and non-standard patients. This classification involves a detailed analysis to verify that each patient's length of stay aligns with the predefined standard range. Patients staying in the hospital for less than seven days will be designated as standard patients. Those admitted for over seven days or not meeting this duration will be classified as non-standard patients. The 'Length of Hospitalization' variable will be initially defined based on this categorization. Additionally, the target variable for predicting mortality is the patient's condition, which has two possible outcomes: recovery or death.

a. Handling Missing Value

This study detected missing values in various attributes such as 'History of Cardiovascular Disease,' 'Past Medical History,' and 'Previous Stroke History.' Properly addressing these missing values is crucial to ensure the data's accuracy and relevance prior to analysis. Filling in the missing values with a value of 0 was selected due to its significant medical relevance in predictive research concerning mortality and length of stay in stroke patients. In certain cases, a value of 0 can represent either the occurrence or non-occurrence of a specific event. Using a consistent value of 0 for numerical data allows for more seamless statistical analysis and reduces issues related to missing data (Huey Fern Tay, 2021).

b. Encoding Variable

This study utilizes two coding techniques: one-hot encoding and label encoding. One-hot encoding transforms each category of an attribute into a binary vector, generating extra columns that represent the sample's membership in each category. For instance, attributes like length of stay, gender, and patient condition are handled with one-hot encoding. On the other hand, label encoding converts distinct values in a column into integers, assigning sequential indices beginning from 0. This method is applied to attributes such as stroke location, cardiovascular disease history, past medical records, and previous stroke history.

3. Extreme Gradient Boosting (XGBoost) Classifier

The fundamental concept of boosting involves creating a more precise model by aggregating numerous simple tree models with low accuracy. In each iteration, an additional tree is incorporated into the model. The subsequent focus is on managing the complexity of these trees. The XGBoost algorithm is able to reduce computing time compared to conventional numerical models that need to handle solving several complicated partial differential equations (Tarwidi et al., 2023).

4. Hyperparameter Tuning

In the field of machine learning, hyperparameters play a vital role in optimizing the performance of a model. They are vital for fine-tuning algorithms and have a significant effect on the outcomes of different model assessments (Muslim Karo Karo, 2020). This research employed Optuna, a sophisticated framework for hyperparameter optimization, to refine the accuracy of mortality and length of stay predictions for stroke patients.

5. Performance Evaluation

In assessing the performance of the predictive model, this research conducted an analysis using various essential metrics to guarantee the precision and dependability of predictions concerning the condition of stroke patients and the duration of their hospital stay. The assessment encompasses precision, recall, F1 score, and accuracy (Yacouby & Axman, 2020).

Precision is a metric used to assess the proportion of correctly predicted positive data points among all the points identified as positive by the model. It is determined by dividing the number of true positives (accurate positive predictions) by the total of true positives and false positives (incorrect positive predictions). The higher the precision value, the fewer false positives the model produces. The formula for precision is shown in equation 1.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

Recall is a performance metric that gauges the proportion of actual positive instances that the model correctly identifies. It is determined by dividing the count of true positives (accurate positive predictions) by the total of true positives and false negatives (incorrect negative predictions). The higher the recall value, the fewer false negatives the model produces. Recall is also known as "sensitivity" in some contexts. Equation 2 shows the formula for recall.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

The F1 Score, as shown in equation 3, is an evaluation metric that combines precision and recall into a single value that reflects the balance between them. The higher the F1 Score, the better the balance between precision and recall of the model. The F1 Score is useful when we want to consider both false positives and false negatives in the evaluation of a classification model.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Accuracy is a metric used to evaluate the proportion of correct predictions (including both true positives and true negatives) relative to the total number of instances. As shown in Equation 4, accuracy is determined by dividing the combined count of true positives and true negatives by the overall number of instances. Higher accuracy indicates that the model is more effective at accurately predicting both positive and negative cases.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives} \quad (4)$$

Additionally, this study demonstrates the model's performance through a Receiver Operating Characteristic (ROC) curve, a two-dimensional metric used to evaluate classification accuracy. The AUC of the ROC curve functions as a scalar measure assessing one aspect of classification model performance. The AUC increases non-linearly with enhanced predictive accuracy, with values approaching 1 indicative of a substantial proportion of predictions exhibiting satisfactory quality (Marzban, 2004).

6. Web Development

The web development approach employed in this study is Extreme Programming (XP), a widely used model within the Agile methodology for system development (Javed et al., 2010). The Extreme Programming (XP) system development method is one of the fastest, most adaptive, and most flexible system development methods in the development process (Sudarsono, 2020). XP consists of four main stages, namely planning, design, coding, and testing (Suryantara & Andry, 2018). The flow of development with XP is shown in Figure 2.

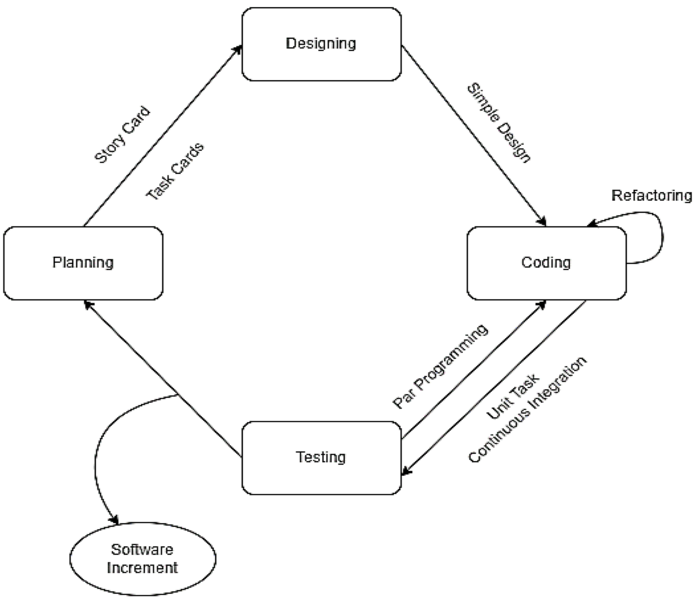

Figure 2. Extreme Programming (XP) Lifecycle

The process of extreme programming development starts with planning and ends with testing for the applications that have been created, as shown in Figure 2.

a. Planning

Planning is the initial stage of application development with Extreme Programming. The primary objective of the planning phase is to identify and comprehend thoroughly the requirements required by the user, which can be accurately documented (Sudarsono, 2020).

b. Designing

The next stage is the design phase, at which stage a design model is made of the application to be created. The modeling tools and languages used to document and describe the architecture, structure, and behavior of software systems are UML (Unified Modeling Language). UML is a flexible modeling language that combines various techniques from data, business, object, and component modeling, applicable throughout the software development process and across different technology platforms (Siau, 2010). An activity diagram, a type of UML diagram, is used to depict the flow of information within a system. Besides, activity diagrams are also useful in modeling system behavior and are very useful in describing business processes and scenarios of system usage (Ahmad et al., 2019).

c. Coding

The coding phase in Extreme Programming (XP) is an important stage in the software development process. At this stage, the source code is written to create the features and functionality that have been planned. This research utilizes Python and the Flask framework for coding. The Python Flask Framework is a microframework that has a simple, easy-to-use language because functions, components, libraries, and extensions are not installed by default so that application developers can adapt to the needs of the created application (Saputra & Susetyo, 2022). To deploy a machine learning model on a local server, the already-created model must first be stored in a suitable format. In this study, the model of machine learning is saved in a Python joblib. Joblib is used for the purpose of increasing the execution speed of the model implemented (Rojas & Bellinger, n.d.). Using the joblib.load() function, the model can be used inside the flask, and it may be used for the prediction of death and the long hospitalization of stroke patients.

There are 17 inputs available on the data input page to predict the death of stroke patients, namely: Length of Hospitalization, Age, Gender, Debtor, History of Cardiovascular Disease, Past Medical History, Previous Stroke History, Hemoglobin (HB), Hematocrit (HT), Leucocyte (LEU), Thrombocytes (TR), Neutrophil-Lymphocyte Ratio (NLR), Total Cholesterol (Chol Total), High-Density Lipoprotein (HDL), Triglisireda (TG), Low-density Lipoproteins (LDL), and Stroke Location.

Similarly to the prediction of death, there are 17 inputs available on the data input page to predict the length of hospitalization, including Patient Condition, Age, Gender, Debtor, History of Cardiovascular Disease, Past Medical History, Previous Stroke History, Hemoglobin (HB), Hematocrit (HT), Leucocyte (LEU), Thrombocytes (TR), Neutrophil-Lymphocyte Ratio (NLR), Total Cholesterol (Chol Total), High-Density Lipoproteins (HDL), Triglisireda (TG), Low-density Lipoprotein (LDL), and Stroke Location.

Making machine learning application interfaces based on website prediction of death and long-term hospitalization of stroke patients in this study uses HTML as web page structure, CSS as web interface styling, and Framework Bootstrap helps in making the web more responsive. To get the value entered by the user into the flask application, the user can use the request.form() function. By creating a data frame of the values entered by the user, the application can proceed to the method predict () of the joblib, which will give the result of predicting the death and duration of hospitalization of the stroke patient.

d. Testing

The next step after the coding phase is Testing. In this phase, an evaluation of the results of the applications that have been created. Testing is done to understand to what extent the application is in line with the goals and expectations planned and to evaluate the performance and effectiveness of the application that has been made (Sudarsono, 2020). The test method applied to web-based machine learning applications for the prediction of death and hospitalization for stroke patients at RSUD Banyumas is Blackbox Testing. Blackbox testing is a system testing technique that prioritizes the assessment of functional requirements in a system or program (Nidhra, 2012). The Blackbox testing implemented on a web-based machine learning application predicting death and long hospitalization of stroke patients includes testing on functional requirements announced at the planning stage. It's designed so that the system that's built can properly help the user and fit the needs of the user. The test plan will be carried out in several available forms, including: (i) Registration form testing: This form is used as a form for registering a new user account. (ii) Login form testing: This form is used as a form for login and application access. (iii) Data input form testing: This form is used as a form for new data input on all attributes used as predictions of death and long hospitalization. (iv) Results form testing: This form is used to display predictions of death and hospitalization length based on newly entered data.

RESULTS AND DISCUSSION

In machine learning, the data is split into two primary subsets: training data and testing data. The training data is utilized to build and train the model, allowing it to learn from the existing data and adjust its internal parameters to perform specific tasks. Conversely, testing data is used to evaluate the trained model to ensure it works well and provides accurate results on previously unseen data (Medar et al., 2017). In this research, we applied an 80:20 split approach, where 80% of the data is designated for training purposes and the remaining 20% is reserved for testing. This approach ensures that the model has sufficient data to learn patterns while reserving a portion for evaluating its performance.

1. Performance evaluation of XGBoost

Table 2. XGBoost Performance Without Hyperparameter Tuning for Mortality Prediction

	Precission	*Recall*	*F-1 Score*	*Accuracy*
Recovery (0)	75	83	77	73%
Mortal (1)	71	60	67	

In Table 2, the performance results of XGBoost in predicting mortality without hyperparameter tuning are shown. The evaluation includes several critical metrics, such as precision, recall, F-1 score, and accuracy, which provide deep insight into the model's effectiveness. First, in the “Recovery (0)” category, XGBoost showed a precision of 75%, indicating that most of the recovery predictions were completely accurate. The recall rate of 83% indicates the model's ability to identify the majority of patients who actually recovered. The F-1 Score, as a combined value of precision and recall, reached 77%, reflecting a balance between accuracy and completeness of predictions. Overall accuracy was 73%, indicating an overall rate of correct predictions for both recoveries and deaths. Second, in the “Mortality (1)” category, XGBoost achieved a precision of 71%, indicating the model's ability to provide correct mortality predictions 71% of the time. However, the recall rate of 60% suggests that the model may have challenges in identifying the small proportion of patients who actually die. Nevertheless, the F-1 score reached 67%, reflecting a good balance between precision and recall. The overall accuracy for predicting death and length of stay in stroke patients was 73%.

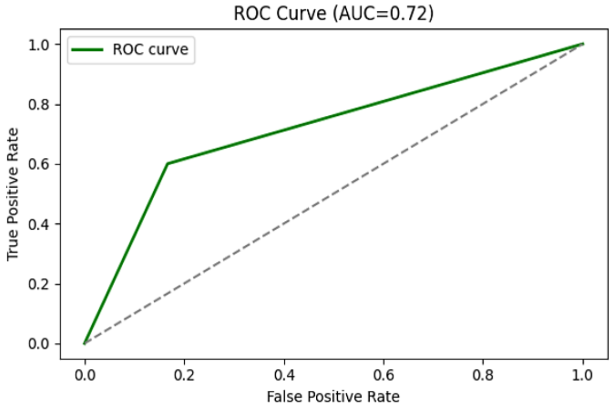

Figure 3. Untuned ROC Curve for Mortality Prediction

The examination through ROC graphs presents a unique perspective on assessing the effectiveness of models in predicting mortality among stroke patients. Particularly focusing on mortality prediction, the ROC AUC assumes a pivotal role as it gauges the model's capacity to differentiate between patients at risk of mortality and and patients with a chance of recovery. With the XGB model's ROC AUC value of 0.72, as delineated in Figure 3, it underscores satisfactory performance, affirming its capability to discern between patient groups based on mortality risk. Thus, the utilization of the AUC ROC graph not only furnishes deeper insights but also facilitates a more precise comparison of the model's predictive prowess.

Table 3. XGBoost Performance: Untuned Length of Hospitalization Prediction

	Precission	*Recall*	*F-1 Score*	*Accuracy*
Standard (0)	69	85	76	68%
Non-Standard (1)	67	44	53	

Table 3 describes the performance of XGBoost in predicting the length of stay of stroke patients in hospitals without adjusting for hyperparameters. In the “Standard (0)” category, XGBoost demonstrated a precision rate of 69%, indicating that most predictions regarding length of stay for patients in the standard category were correct. The high recall rate, reaching 85%, indicates that the model is able to identify the majority of patients who actually experience the standard length of stay. An F-1 score of 76% reflects a good balance between precision and recall. Meanwhile, in the “Non-Standard (1)” category, XGBoost achieved a precision rate of 67%, indicating that most predictions regarding the length of stay of patients in the non-standard category were correct. However, the lower recall rate, at 44%, indicates that the model may have difficulty identifying the portion of patients who actually experienced non-standard lengths of stay. The F-1 score of 53% indicates a balance between precision and recall within this category. Overall, the evaluation results of XGBoost for predicting length of stay in stroke patients show good performance. The overall accuracy of 68% reflects the accuracy of the model in providing predictions of patient length of stay in the hospital without hyperparameter tuning.

The XGBoost model's ROC AUC value of 0.65, as shown in Figure 4, further accentuates its performance in distinguishing between stroke patients with “Standard” and “Non-standard” lengths of hospitalization. This metric provides valuable insight into the model's ability to discriminate between these patient groups based on their hospitalization durations.

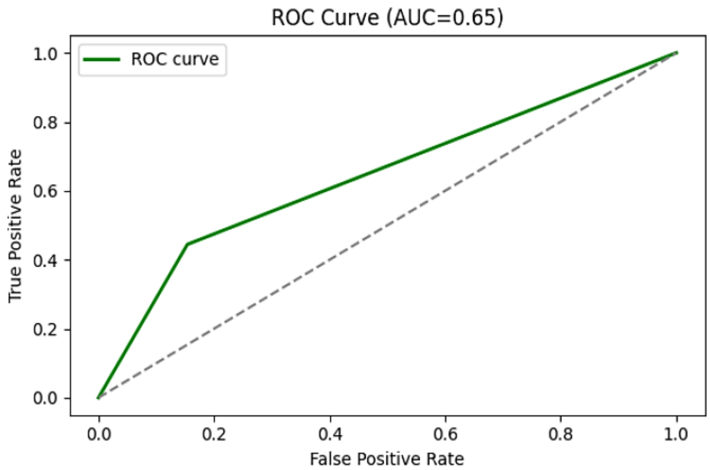

Figure 4. Untuned ROC Curve for Length of Hospitalization Prediction

The use of the algorithm model in this study is consistent with the findings of (Abujaber et al., 2024), which also reported high results for the use of the XGBoost algorithm model after comparative analysis with other algorithm models in predicting early mortality, improving risk assessment, and personalized care in stroke patients.

2. Model Performance with Hyperparameter Tuning

To enhance the performance of the previously discussed XGBoost model, one approach involves optimizing hyperparameters. This optimization is carried out using Optuna, an advanced tool that automates the search for optimal hyperparameter values. During this process, multiple iterations are conducted, and the most effective hyperparameter values are identified and applied to improve the accuracy of mortality and length of stay predictions for stroke patients. The results of this optimization show that the best parameters found involve n_estimator values of 267, max_depth of 12, learning_rate of 0.82, subsample of 0.92, colsample_bytree of 0.92, gamma of 0.28, and reg_alpha of 0.0.

Table 4. XGBoost Performance of Mortality Prediction After Tuning

	Precission	*Recall*	*F-1 Score*	*Accuracy*
Recovery (0)	82	90	86	86%
Mortal (1)	91	83	87	

Based on the results in Table 4, the tuned XGBoost model has better performance in predicting patient mortality. This is indicated by an accuracy value of 86%, which means the model is able to correctly predict 86% of all patient data. In more detail, the XGBoost model was able to correctly predict 82% of patients who recovered (recovery) and 91% of patients who died (mortality). The precision and recall values for patients who recovered were 82% and 90%, respectively, which means the model was able to correctly predict 82% of patients who recovered and 90% of patients who should have recovered. The precision and recall values for patients who died were 91% and 83%, respectively, which means the model was able to correctly predict 91% of patients who died and 83% of patients who should have died. In general, the tuned XGBoost model has quite good performance in predicting patient mortality. A high accuracy value indicates that the model is able to correctly predict most of the patient data. The high precision and recall values for both data classes indicate that the model is able to correctly predict the majority of patients who should recover and die.

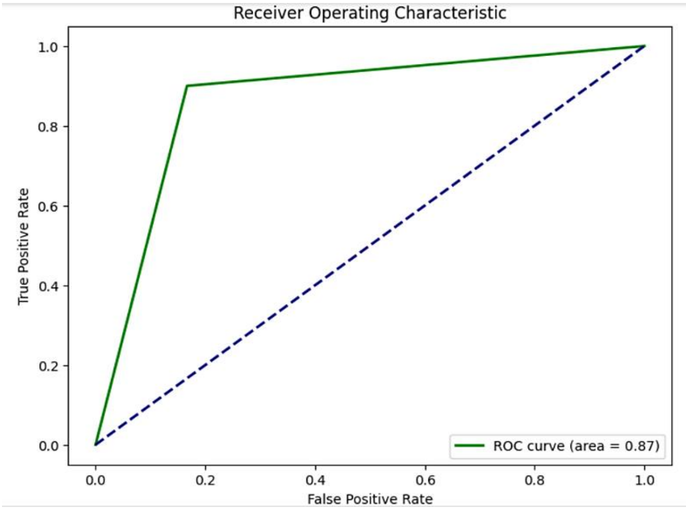

Figure 5. Tuned ROC curve for mortality prediction

After adjustment, the XGBoost model demonstrates improved performance in predicting stroke patient mortality as seen from the ROC AUC value of 0.87 shown in Figure 5. This indicates the model's strong capability in distinguishing between patients with different outcomes, particularly in predicting mortality. It suggests that the model effectively discriminates between patients at high risk of mortality and and patients with a chance of recovery, providing valuable insights into its enhanced predictive model performance of mortality with hyperparameter tuning.

Table 5. XGBoost Performance of Length of Hospitalization Prediction After Tuning

	Precission	*Recall*	*F-1 Score*	*Accuracy*
Standard (0)	80	92	86	82%
Non-Standard (1)	86	67	75	

Based on Table 5, it shows the performance results of the XGBoost model after tuning the prediction of length of hospitalization. Model evaluation is carried out through a number of performance metrics, including precision, recall, F-1 score, and accuracy. For the standard category (0), the model achieved a precision level of 80%, recall of 92%, F-1 score of 86%, and accuracy of 82%. Meanwhile, for the non-standard category (1), the model shows precision of 86%, recall of 67%, and an F-1 score of 75%. Although the precision in the non-standard category is higher than the standard category, the lower recall indicates that the model tends to be better at identifying standard cases compared to non-standard cases. These results provide a comprehensive picture of the model's performance in predicting length of hospitalization after going through the tuning process of the XGBoost algorithm.

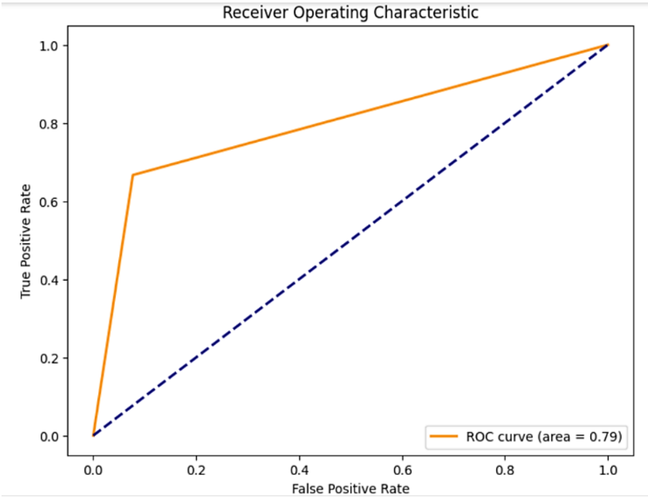

Figure 6. Tuned ROC Curve for Length of Hospitalization Prediction

The improvement in the Receiver Operating Characteristic (ROC) curve's Area Under the Curve (AUC) to 0.79 following parameter tuning, as shown in Figure 6, indicates a significant improvement in the XGBoost model's ability to predict the duration of hospitalization for stroke patients. A higher ROC AUC implies that the model has a greater ability to differentiate between patients with standard and non-standard lengths of hospitalization. Therefore, the model tuning has successfully enhanced the accuracy and reliability of the model's predictions in determining the duration of hospitalization for stroke patients.

These findings align with the work of (Bhaskar et al., 2024), who observed improvements for accuracy after further improved via hyperparameter tuning strategies their machine learning models for healthcare predictions. On the other hand, this model can be further enhanced by exploring other hyperparameter optimization techniques or refining the search space for Optuna. Consider running longer optimization processes or more iterations to find better hyperparameters, thereby offering strong reliability in assessing mortality risk and providing precise estimates concerning patient hospitalization duration.

3. Web Development Implements the XGBoost algorithm

a. Planning

An important aspect of planning when building a system is to identify the functional requirements of the system. The functional requirements of machine learning applications for predicting the death and long hospitalization of stroke patients can be described as follows: (i) Registered users must be able to log in with the correct username/email and password. (ii) The system must display an error message if the user inputs incorrect username and password credentials during the login attempt. (iii) The system must provide a registration form that allows new users to enter registration data such as name, email address, password, and user level. (iv) The system should send a message if the user tries to sign up with the email data already in the system. (v) Once logged in, the user should be able to access two prediction buttons, namely death and hospitalization duration, which will open the new data input form of the stroke patient to make a prediction of death and hospital duration. (vi) The system must be able to receive new data inputs to be used in the predictive model. (vii) The system must be able to produce predictions of death and hospitalization time based on data entered by the user and display the results to the user. (viii) As long as the user does not log out before exiting the application, then the user doesn't need to log in again when accessing the application.

b. Designing

The application design is made from the system's point of view using the UML, which is an activity diagram. Figure 7 Activity login and register diagrams explains the login and registration process of a web-based machine learning application for predicting death and length of stay for stroke patients produced in this study. When a user first accesses the web application, they will be directed to the login page. When the user already has a previous account, they only need to input the username in the form of an email and the password that has been createdIf the user lacks an account, they will need to create one in order to access the application. Users will be directed to input data in the form of username, email, password, and user level, whether the user is an admin or an officer. If the user has successfully registered and has an account, then return to the login page and input the username and password to enter the application. The system will automatically validate the username and password. If the username and password entered by the user are correct, then when the user clicks the login button, they are directed to the web application dashboard for predicting death and length of stay for stroke patients.

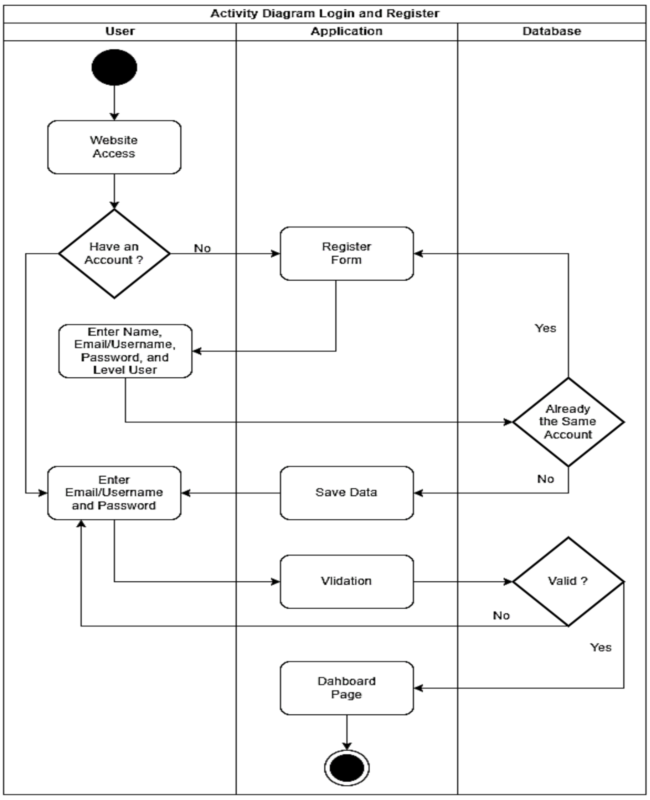

Figure 7. Activity Diagram Login and Register

To predict death and length of stay for stroke patients, the user must input stroke patient data. Figure 8 shows the activity diagram for inputting stroke patient data to produce prediction output. Figure 8 shows the process of inputting stroke patient data needed to predict death and length of stay for stroke patients. In the resulting application, there are two different input forms, namely, a data input form predicting death and a data input form predicting length of stay in the hospital. For the first step, the user must click the prediction button available on the application dashboard to access the required data input form.

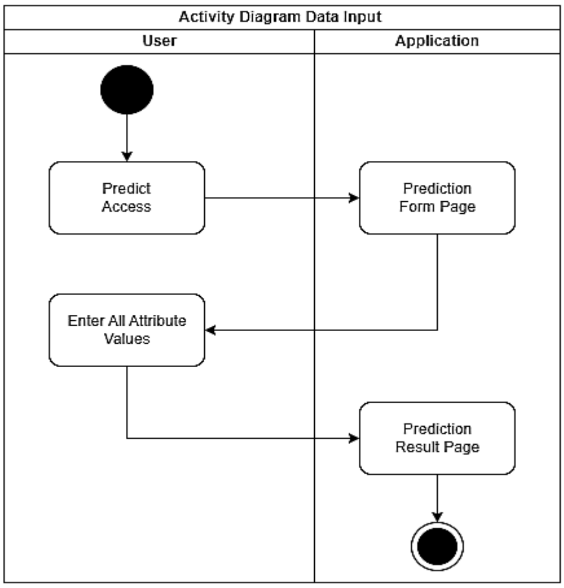

Figure 8. Activity Diagram Data Input

Next, the application will display a form that is used by the user as a place to input new stroke patient data. Users are required to input the required stroke patient data on all attributes to produce predictions of death and length of stay. There are 17 attributes that users must input with numeric fields in each data input form. After the data has been inputted, the user then clicks the predict button to see the resulting output, which is in the form of a prediction of death and length of stay for stroke patients on the results page in the application.

c. Coding

After obtaining the best model accuracy performance of XGBoost, the machine learning model is saved in Python joblib format to be implemented in the web application created. After the model is trained and saved in joblib form, the next stage in coding for a web-based machine learning application for predicting death and length of stay is to design a web application where the user will enter all the attribute values and the data will be passed to the machine learning model based on the training provided. Once the model has been carried out, it will predict the results, namely in the form of predictions of death and length of stay for stroke patients.

To be able to access this web-based machine learning application, a login and registration form are required. The login and registration forms are saved in a folder called templates in the formats 'login.html' and'registrasi.html'. The login page display is shown in Figure 9, and the registration page display is shown in Figure 10.

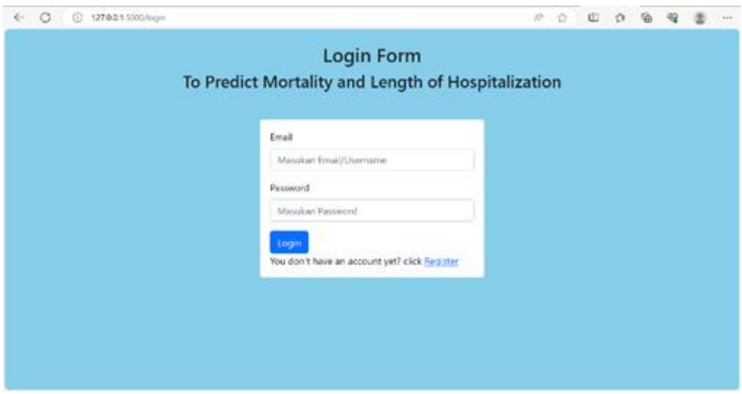

Figure 9. Login Page of Application

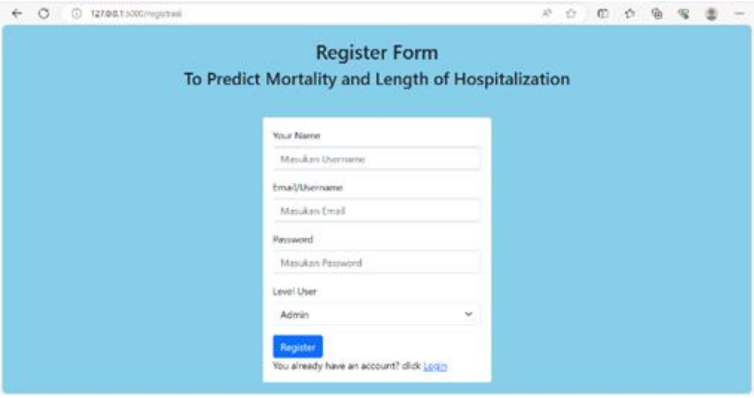

Figure 10. Register Page of Application

The first page displayed after a user logs in is the main page. This page was created in the 'index.html' format, which is saved in the templates folder. This main page contains a button for prediction. The page view is shown in Figure 11.

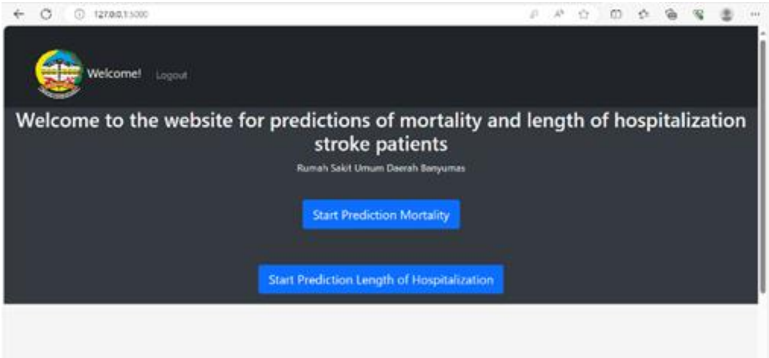

Figure 11. Index Page of Application

The application will collect new data or attribute values that are used in predicting death and length of stay for stroke patients and then use a machine learning model to predict whether the patient will die or recover and the standard or non-standard treatment time that will be displayed to the user. An HTML form is needed to collect this data, which will contain all input attribute values for all prediction variables. There were four HTML forms created in this research. The first and second forms are used to enter independent variable values, which are saved in the 'prediction.html' format for mortality prediction and the format 'predik_lama_rawat_inap.html' for predicting length of stay.

The third and fourth forms are used to display prediction results, which are made in the'result.html' format to display death prediction results and the format'result_lama_rawat_inap.html' to display the results of predicting the length of stay for stroke patients. The four folders are stored in the templates folder. Apart from the HTML form, to read the application display code in the template folder, you need a script that contains the flask function. The flask function is used as a Python link that contains a predictive machine learning model with an application display created so that prediction results can be displayed to the user. This script is saved in the format'main.py'. After creating all the required script code, the flask script code can be run on localhost. The script can be run by activating the flask main.py script in the terminal so that a notification will be displayed that the flask has been running on localhost, as seen in Figure 12.

```
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 136-734-846
```

Figure 12. Flask Script Code for Running

The program code can be accessed via the GitHub repository: https://github.com/pujilestari21/stroke_app. The display of the new data input form is shown in Figure 13, with the prediction results displayed on the results page as shown in Figure 14.

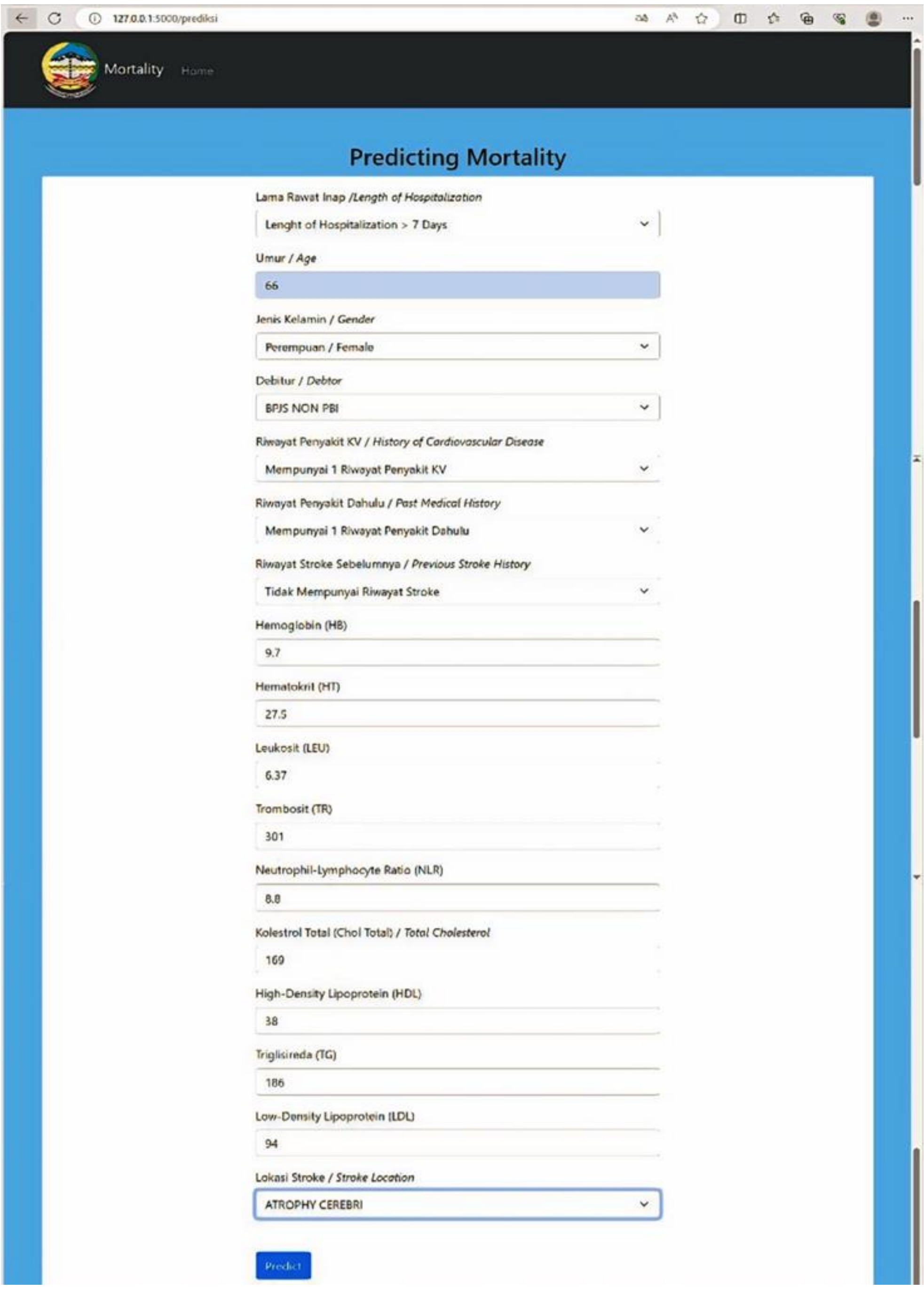

Figure 13. Data Input Page of Mortality Prediction

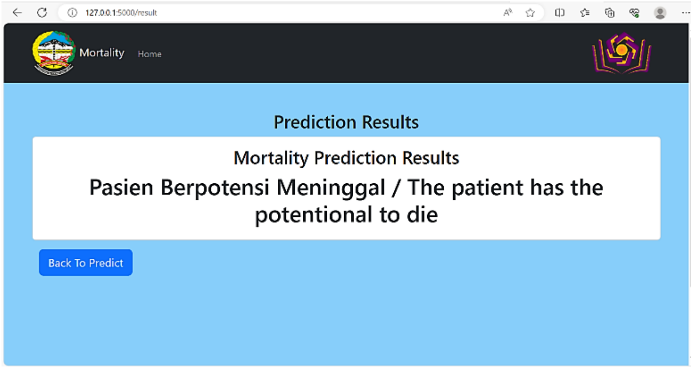

Figure 14. Result Page of Mortality Prediction

The research implemented a web application using the XGBoost algorithm. It has been proven to be very effective in predicting the duration of hospitalization and death for stroke patients. The objective of developing this application is to design an XGB model that is seamlessly integrated into a web-based platform utilizing the Python Flask framework, enabling the medical team to estimate a patient's condition and hospital stay duration. Hopefully, this application will facilitate a more efficient medical decision-making process.

This web application processes data entered by users through the input form, as depicted in Figure 9. his data is fed into a machine learning model trained with the XGBoost algorithm. The training involves historical data from stroke patients, enabling the model to identify patterns related to mortality and hospital stay duration. Once user input is received, the XGBoost model will assess the attributes and forecast the duration of hospitalization for stroke patients based on the given data.

The predicted results will be delivered to the user through the web interface, presented as an output form, as illustrated in Figure 10. This report not only provides information but also provides a more detailed understanding of the patient's condition and length of stay for stroke patients. Patient condition data is presented in the categories of potential recovery (0) and potential death (1). The prediction result of healing potential (0) shows that the patient has great potential to recover and undergo outpatient treatment. On the other hand, the results of predicting potential death (1) show that patients have a high potential for death. This information can also be used in making treatment decisions to reduce the risk of death in stroke patients. Data predicting patient length of stay are classified into standard (0) and non-standard (1) categories. A standard prediction (0) suggests that the patient is likely to be treated within the hospital's designated standard duration (<7 days). Conversely, a non-standard prediction (1) indicates a tendency for the patient to require hospitalization beyond the standard timeframe (>7 days).

The effectiveness of this application is closely linked to the quality of health services and financial management for stroke patients. The application’s predictions offer insights into the probability of mortality and the duration of treatment for stroke patients, as well as valuable information on the potential financial implications. As a result, this application not only supports patient health monitoring but also offers a holistic and strategic perspective on financial management related to stroke patient care.

d. Testing

Testing of the web-based machine learning application for predicting death and length of stay for stroke patients was carried out by prioritizing the functional requirements of the system, which had been outlined at the planning stage. Testing using black box testing on the application is described in Table 6.

Table 6. Black box testing

Testing Description	*Expected Results*	*Actual test results*	*Conclusion*
Log in using the valid username and password.	The system will give users access to the application	The system provides access to the application	According to expectations
Login with the wrong username or password	The system will display an error message, either a username error or a password error	The system successfully displays an input error message on the username and password	According to expectations
Register by filling in your name, email, password, and user level	The system can accept user registration and display a registration success message	The system successfully received the registration and successfully delivered the registration success message	According to expectations
Register by filling in the email data that was previously registered	The system will display an error message that the email data already exists and has been registered	The system successfully displays an error message when registering with an existing or previously registered email	According to expectations
Click the death prediction and length of stay button on the dashboard	The system will display a new data input page for predictions of death and length of stay	The system successfully displays the data input form page	According to expectations
Enter new data in the data input form	The system can receive new data input	The system successfully received new data input	According to expectations
Click the prediction button on the data input form	The system can display prediction results	The system successfully displays the prediction results page	According to expectations
Exit the application without clicking logout	Users can access the application without having to log in again	Users can successfully access the application without having to log in again	According to expectations
Exit the application by clicking logout	Users must log in again to access the application	Users re-login when they want to access the application	According to expectations

From the results of tests carried out on each system's functional requirements, it was found that the web-based machine learning application for predicting death and length of stay in the hospital as a whole has system functionality that meets user expectations and needs for predicting death and length of stay for stroke patients. The XGBoost model was incorporated into a web application aimed at forecasting both mortality and the duration of hospital stay for stroke patients. The application leverages a customized machine learning model to provide healthcare professionals with accurate predictions that can aid in decision making. This implementation follows best practices in applying machine learning models to real-world applications, as discussed in (Vani et al., 2024) which provides an integrated platform for predicting health conditions. Also, this study can fill the gap in previous studies by using stroke patient data to produce a high level of accuracy in stroke disease predictions.

CONCLUSIONS AND RECOMMENDATIONS

This research successfully addressed the pressing issue of stroke management at RSUD Banyumas by developing a predictive model using the XGBoost algorithm. The study utilized a dataset consisting of 106 patient records to train and validate the model. The study demonstrated the model's remarkable accuracy in forecasting patient mortality, achieving an 86% accuracy rate. Furthermore, the XGBoost algorithm exhibited excellent performance in predicting the length of hospitalization, with an accuracy of

82%. The execution of this model into a web-based machine learning application using the Python Flask framework represents a significant leap toward enhancing the quality of care for stroke patients. The successful integration of the XGBoost algorithm into the application holds promise for maximizing service quality and controlling treatment costs at RSUD Banyumas. The application offers immediate predictions on patient outcomes, enabling healthcare providers to make better-informed decisions about patient management, resource distribution, and treatment strategies. By leveraging these predictive insights, RSUD Banyumas can optimize its operational efficiency and ensure that patients receive timely and appropriate interventions. As this tool enters the decision-making process at RSUD Banyumas, it is anticipated that the application will substantially contribute to the overall improvement of stroke patient care. By providing accurate predictions and actionable insights, the application empowers healthcare professionals to deliver more personalized and effective care, ultimately leading to better clinical outcomes and enhanced patient satisfaction.

In addition to its impact on clinical care, the application also offers significant benefits in terms of financial management. By helping to identify patients at high risk of adverse outcomes, the application enables RSUD Banyumas to allocate resources more efficiently, reduce unnecessary treatment costs, and optimize its budget allocation. This research thus highlights the potential of machine learning technologies to drive meaningful improvements in both clinical and financial aspects of healthcare delivery. In conclusion, this research marks a crucial step in leveraging machine learning to address healthcare challenges and underscores the potential for innovative solutions to enhance patient outcomes and healthcare system efficiency. Moving forward, further research and development in this area hold great promise for revolutionizing the delivery of healthcare services and improving the lives of patients. It is hoped that future research can use a larger dataset, so that it can provide more optimal prediction results regarding the potential for death and length of stay. Additionally, this model can be improved by exploring other hyperparameter optimization techniques or refining the search space for Optuna. Running longer optimization processes or more iterations can help find better hyperparameters, enhancing reliability in identifying mortality risk and estimating patient length of hospitalization accurately.

REFERENCES

Abujaber, A. A., Albalkhi, I., Imam, Y., Nashwan, A., Akhtar, N., & Alkhawaldeh, I. M. (2024). Machine learning-based prognostication of mortality in stroke patients. *Heliyon*, *10*(7), e28869. https://doi.org/10.1016/j.heliyon.2024.e28869

Ahmad, T., Iqbal, J., Ashraf, A., Truscan, D., & Porres, I. (2019). Model-based testing using UML activity diagrams: A systematic mapping study. *Computer Science Review*, *33*(July), 98–112. https://doi.org/10.1016/j.cosrev.2019.07.001

Amruthamathi, A., & Devi, D. (2023). *Diagnosis of liver failure using flask web framework*. *050016*.

Barsasella, D., Bah, K., Mishra, P., Uddin, M., Dhar, E., Suryani, D. L., Setiadi, D., Masturoh, I., Sugiarti, I., Jonnagaddala, J., & Syed-Abdul, S. (2022). A Machine Learning Model to Predict Length of Stay and Mortality among Diabetes and Hypertension Inpatients. *Medicina (Kaunas, Lithuania)*, *58*(11). https://doi.org/10.3390/medicina58111568

Bhaskar, P., Kumar Burramsetty, V. S. R., Pinnaka, B., Teja Kalapala, B., & Kumar Tanguturi, V. S. S. (2024). Advancing Healthcare Predictions: Harnessing Machine Learning for Accurate Health Index Prognosis. *International Journal of Innovative Science and Research Technology (IJISRT)*, *9*(4), 934–941. https://doi.org/10.38124/ijisrt/ijisrt24apr709

Huey Fern Tay. (2021). *When is it ok to impute missing values with a zero?* Towardsdatascience.Com. https://towardsdatascience.com/when-is-it-ok-to-impute-missing-values-with-a-zero-6d94b3bf1352

Indarto, Utami, E., & Suwanto, R. (2020). Prediksi Risiko Kematian Pasien Stroke Pendarahan Dengan Menggunakan Teknik Klasifikasi Data Mining. *Jurnal Informasi Interaktif*, *5*(2), 86–91.

Javed, M., Ahmad, B., Hussain, S., & Ahmad, S. (2010). Mapping The Best Practices of XP and Project Management: Well defined approach for Project Manager. *Journal of Computing*, *2*(3), 2151–9617.

Liu, J., Wu, J., Liu, S., Li, M., Hu, K., & Li, K. (2021). Predicting mortality of patients with acute kidney injury in the ICU using XGBoost model. *PLoS ONE*, *16*(2 February), 1–11. https://doi.org/10.1371/journal.pone.0246306

Mahmoud Hewedi, K., El Din Tharwat Mohamed, M., & Awad Yousef Abo Sleem, M. (2019). Pre Hospital and in Hospital Delays After the Onset of Acute Ischemic Stroke in a Sample of Egyptian Patients. *Al-Azhar Medical Journal*, *48*(4), 311–322. https://doi.org/10.21608/amj.2019.64940

Marzban, C. (2004). The ROC curve and the area under it as performance measures. *Weather and Forecasting*, *19*(6), 1106–1114. https://doi.org/10.1175/825.1

Maskuri, M. N., Harliana, Sukerti, K., & Bhakti, R. M. H. (2022). Penerapan Algoritma K-Nearest Neighbor (KNN) untuk Memprediksi Penyakit Stroke. *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, *4*(1), 130–140.

Medar, R., Rajpurohit, V. S., & Rashmi, B. (2017). Impact of Training and Testing Data Splits on Accuracy of Time Series Forecasting in Machine Learning. *2017 International Conference on Computing, Communication, Control and Automation, ICCUBEA 2017*, 1–6. https://doi.org/10.1109/ICCUBEA.2017.8463779

Muslim Karo Karo, I. (2020). Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan. *Journal of Software Engineering, Information and Communication Technology*, *1*(1), 11–18.

Nidhra, S. (2012). Black Box and White Box Testing Techniques - A Literature Review. *International Journal of Embedded Systems and Applications*, *2*(2), 29–50. https://doi.org/10.5121/ijesa.2012.2204

Paleczek, A., Grochala, D., & Rydosz, A. (2021). Artificial breath classification using xgboost algorithm for diabetes detection. *Sensors*, *21*(12). https://doi.org/10.3390/s21124187

Pawar, S., Shukla, P., Bagal, P., Dawkhar, A., & Jadhav, R. (2022). Development of Web Application for Breast Cancer Detection using Machine Learning Classifier. *13th International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT 2022*, *8*(December), 552–556.

Putri, A. K., & Sonia, D. (2021). *Efektivitas Pengembalian Berkas Rekam Medis Rawat Inap Dalam Menunjang Kualitas Laporan di Rumah Sakit Bhayangkara Sartika Asih Bandung*. *2*(3).

Rojas, A. P., & Bellinger, E. P. (n.d.). *Engineering a Web-based Interface to Predict the Unknown Physical Properties of Solar-like Stars*. 2–5.

Saha, R., Malviya, L., Jadhav, A., & Dangi, R. (2023). Early stage HIV diagnosis using optimized ensemble learning technique. *Biomedical Signal Processing and Control*, *89*(July 2023), 105787. https://doi.org/10.1016/j.bspc.2023.105787

Saputra, J. A., & Susetyo, Y. A. (2022). Development of Price Distribution Module on Merchandise Application Using Flask Framework At Pt Xyz. *Jurnal Teknik Informatika (Jutif)*, *3*(5), 1293–1300. https://doi.org/10.20884/1.jutif.2022.3.5.351

Setiawan, P. A. (2020). Diagnosis Dan Tatalaksana Stroke Hemoragik. *Jurnal Medika Utama*, *02*(01), 402–406.

Siau, K. (2010). An analysis of unified modeling language (UML) graphical constructs based on BWW ontology. *Journal of Database Management*, *21*(1).

Sittig, D. F., Wright, A., Coiera, E., Magrabi, F., Ratwani, R., Bates, D. W., & Singh, H. (2020). Current challenges in health information technology–related patient safety. *Health Informatics Journal*, *26*(1), 181–189. https://doi.org/10.1177/1460458218814893

Sudarsono, B. G. (2020). Using an Extreme Programming Method for Hotel Reservation System Development. *International Journal of Emerging Trends in Engineering Research*, *8*(6), 2223--2228. https://doi.org/10.30534/ijeter/2020/01862020

Suryantara, I. G. N., & Andry, J. F. (2018). Development of Medical Record With Extreme Programming SDLC. *International Journal of New Media Technology*, *5*(1), 47–53. https://doi.org/10.31937/ijnmt.v5i1.706

Tarwidi, D., Pudjaprasetya, S. R., Adytia, D., & Apri, M. (2023). An optimized XGBoost-based machine learning method for predicting wave run-up on a sloping beach. *MethodsX*, *10*(December 2022), 102119. https://doi.org/10.1016/j.mex.2023.102119

Uddin, K. M. M., Nahid, M. N. H., Ullah, M. M. H., Mazumder, B., Khan, M. S. I., & Dey, S. K. (2023). Machine Learning-Based Chronic Kidney Cancer Prediction Application: A Predictive Analytics Approach. *Biomedical Materials & Devices*, *0123456789*. https://doi.org/10.1007/s44174-023-00133-5

Vani, G., Karunakar, K., Reddy, C. V., Reddy, S., & Reddy, V. (2024). Health Monitoring and Disease Detection Using Machine Learning. *International Journal for Innovative Engineering and Management Research*, *09*(June).

Yacouby, R., & Axman, D. (2020). *Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models*. 79–91. https://doi.org/10.18653/v1/2020.eval4nlp-1.9