Comparison of Decision Tree and CatBoost for Stroke Risk Prediction Using the Healthcare Stroke Prediction Dataset: A Data Science Approach

Rahma Yuni Simanullang, Zulham Sitorus

Abstract

Stroke poses a significant health risk, necessitating an approach that can effectively predict risk based on an individual's health characteristics. This study falls within the field of Data Science specifically machine learning-based classification and aims to compare the performance of Decision Tree and CatBoost algorithms in predicting stroke risk using the Healthcare Stroke Prediction Dataset. This comparison was conducted because Decision Trees offer a simple, easily interpretable classification structure, whereas CatBoost employs a gradient boosting approach that potentially yields superior predictive capabilities. The dataset comprises 5,110 records, and the research methodology included data preprocessing, an 80:20 split for training and testing data, handling class imbalance via SMOTE, and hyperparameter optimization using GridSearchCV. Performance was evaluated using Accuracy, Precision, Recall, F1-Score, and ROC-AUC. The results indicate that the Decision Tree achieved an Accuracy of 92.37%, Precision of 18.18%, Recall of 16.00%, F1-Score of 17.02%, and ROC-AUC of 62.85%, while CatBoost achieved an Accuracy of 94.03%, Precision of 17.65%, Recall of 6.00%, F1-Score of 8.96%, and ROC-AUC of 78.39%. These findings demonstrate that CatBoost outperformed the Decision Tree in terms of Accuracy and ROC-AUC, whereas the Decision Tree performed better regarding Precision, Recall, and F1-Score. Thus, both algorithms exhibit distinct advantages in stroke risk prediction.

***Keywords**: CatBoost; Data Science; Decision Tree; Machine Learning; Stroke Prediction*

Introduction

Stroke is a cerebrovascular disorder that occurs when blood flow to the brain is disrupted so that it can cause brain tissue damage, disability, and even death. The risk of stroke is related to various demographic characteristics, health conditions, and lifestyle of a person, such as age, hypertension, heart disease, glucose levels, body mass index, and smoking habits. Therefore, patient health data can be used to identify patterns related to the likelihood of stroke to support early risk detection efforts .

The problem in stroke prediction is the number of interrelated factors and the imbalance in the amount of data between stroke and non-stroke patients. This condition can cause the process of conventional pattern identification to be less effective and the prediction model tends to recognize the majority class. One solution that can be applied is a Data Science approach through the use of machine learning to study patterns from health data and generate automatic stroke risk classification models.

Machine learning is an important part of Data Science that allows computers to learn patterns based on historical data to perform classifications and predictions [1],[2],[3]. In this study, the prediction process was carried out through the stages of data preprocessing, data sharing training and testing, handling class imbalances using the Synthetic Minority Over-sampling Technique (SMOTE), model formation, parameter optimization, and performance evaluation. This approach is used so that the processed data has the appropriate quality before being used in the process of forming a prediction model.

The algorithms compared in this study are Decision Tree and CatBoost. Decision Tree was chosen because it has a relatively simple and easy-to-interpret tree structure-based classification mechanism, while CatBoost is a gradient boosting-based algorithm that builds a number of trees gradually to improve predictive capabilities [1],[4]. A comparison was made to find out the difference in the performance of the single decision tree approach and the boosting approach. Both algorithms are optimized using GridSearchCV, while SMOTE is used to handle class imbalances in training data.

This study aims to compare the performance of Decision Tree and CatBoost in predicting stroke risk and find out the algorithm that provides the best performance characteristics based on Accuracy, Precision, Recall, F1-Score, and ROC-AUC. The data used is a Healthcare Stroke Prediction Dataset of 5,110 records, which contain demographic and health characteristics of patients. The entire process of data processing, algorithm implementation, model optimization, and evaluation is carried out using the Python programming language on Google Colaboratory [5].

Several previous studies have applied machine learning in stroke prediction. Werdiningsih et al. (2023) through a study entitled "Stroke Prediction Analysis Using Decision Tree Approach with Feature Selection and Neural Network" analyzed factors in health data using Decision Tree and Neural Network to support stroke prediction [6]. Furthermore, Kadhim and Radhi (2023) in the study "Machine Learning Prediction of Brain Stroke at an Early Stage" applied Random Forest, Logistic Regression, and Decision Tree accompanied by SMOTE and reported prediction accuracy of up to 99% [7]. Meanwhile, Aulia et al. (2024) through the study "Analysis of Stroke Prediction by Comparing Three Decision Tree, Naïve Bayes, and Random Forest Classification Methods" compared three classification methods to determine the appropriate model in stroke prediction [8].

Based on previous research, the use of machine learning for stroke prediction has been widely done, but a study that specifically compares Decision Tree and CatBoost in the Healthcare Stroke Prediction Dataset with the combination of SMOTE and GridSearchCV optimization is still a space that can be studied further. Therefore, the contribution of this study is to provide a comparative analysis between a single tree-based model and gradient boosting with the same data treatment and evaluation scheme. The comparison is not only based on Accuracy, but also Precision, Recall, F1-Score, and ROC-AUC so that it can provide a more comprehensive picture of the ability of the two algorithms to predict stroke risk.

Literature Review

Stroke risk prediction can be supported by machine learning techniques that identify patterns from patient health data. This study compares Decision Tree, which is simple and interpretable, with CatBoost, a gradient boosting algorithm capable of building more complex predictive models. SMOTE is applied to address class imbalance, while GridSearchCV is used to optimize model parameters [9],[10],[11].

Research Methodology

This study applies a data science approach using the Healthcare Stroke Prediction Dataset consisting of 5,110 records. The research stages include problem identification, data collection, preprocessing, an 80:20 train-test split, class balancing using SMOTE, and model development using Decision Tree and CatBoost. Both models are optimized using GridSearchCV and evaluated using Accuracy, Precision, Recall, F1-Score, and ROC-AUC to compare their performance in predicting stroke risk [12],[13].

Results

3.1 Dataset

This study uses *the Healthcare Stroke Prediction Dataset* which consists of demographic data, health conditions, and individual characteristics related to the risk of stroke. Based on the results of the initial identification, the dataset has 5,110 records and 12 attributes. The stroke attribute is used as the target variable in the classification process, while the other attributes are used as the variables that support the prediction model formation process using the Decision Tree and CatBoost algorithms. A value of 0 on the stroke attribute indicates an individual who did not have a stroke, while a value of 1 indicates an individual who has had a stroke.

The initial stage of analysis is carried out by identifying the structure of the dataset which includes the name of the attribute, the data type, the amount of data available, and the presence of missing *values*. This identification is needed to determine the initial characteristics of the dataset before the preprocessing stage and the formation of the classification model are carried out. Based on the results of the examination, the dataset consists of numerical and categorical attributes. The characteristics of each attribute in the dataset are shown in Table 1.

Table 1. Characteristics of Healthcare Stroke Prediction Dataset

No.	Attributes	Data Type	Amount of Data	Missing Value	Remarks
1	id	Integer	5.110	0	Unique identity of individuals
2	Gender	Object	5.110	0	Gender
3	Age	Float	5.110	0	Individual age
4	hypertension	Integer	5.110	0	Hypertension status
5	Heart disease	Integer	5.110	0	Heart disease status
6	Ever married	Object	5.110	0	Marital status
7	Work type	Object	5.110	0	Type of job
8	Residence type	Object	5.110	0	Type of residence
9	Avg glucose level	Float	5.110	0	Average glucose levels
10	BMI	Float	4.909	201	Body mass index
11	Smoking status	Object	5.110	0	Smoking status
12	Stroke	Integer	5.110	0	Target classification

Based on Table 1, the dataset has a combination of numerical and categorical attributes. The results of the examination showed that most of the attributes had complete data, while the bmi attribute had 4,909 available values and 201 *missing values*. In addition, no duplicate data was found on the dataset. The id attribute only serves as an individual's unique identity so it does not represent health characteristics and is subsequently not used as a feature in model formation. Meanwhile, the stroke attribute is set as the classification target.
Furthermore, identification of the class distribution in the stroke target attribute was carried out. A class distribution analysis is necessary to find out the proportion between individuals who have had a stroke and individuals who have not had a stroke. The results of the target class distribution in the dataset are shown in Table 2.

Table 2. Distribution of Stroke Target Classes

Classes	Remarks	Amount of Data	Percentage
0	No Stroke	4.861	95,13%
1	Stroke	249	4,87%
Total		5.110	100%

Based on Table 2, as many as 4,861 data or 95.13% were included in the non-stroke class, while 249 data or 4.87% were included in the stroke class. The difference in proportions shows that the distribution of classes in the dataset is imbalanced, because the amount of data in the non-stroke class is much larger than that of the stroke class. This condition needs to be considered at the data processing and model evaluation stages so that the performance of Decision Tree and CatBoost is not only determined based on the level of accuracy, but also considers the model's ability to identify stroke classes.

3.2 Preprocessing Data Results

The data preprocessing stage is carried out to prepare the dataset before it is used in the process of forming the Decision Tree and CatBoost models. Based on the results of the initial identification, the dataset had 5,110 data with 12 attributes and found 201 missing values in the BMI attribute. The preprocessing stages carried out include the removal of irrelevant attributes, the handling of missing values, and the transformation of categorical attributes into numerical data.

The id attribute is removed from the dataset because it only serves as the unique identity of each individual and does not represent the characteristics used in predicting stroke risk. After the id attribute is removed, the missing value is handled on the BMI attribute. A total of 201 empty values in these attributes were handled using the median imputation technique. Based on the calculation results, a median BMI value of 28.1 was obtained which was then used to replace all blank BMI values. The results of missing value handling are shown in Table 3.

Table 3. Missing Value Handling Results

Attributes	Missing Value Before	Method	Median Value	Missing Value After
BMI	201	Median Imputation	28,1	0

Based on Table 3, as many as 201 *missing values* in the bmi attribute were successfully handled using a median value of 28.1 so that there were no more empty values in these attributes. The use of imputation techniques allows all 5,110 data to be preserved without deleting records. The next stage is to transform the categorical attributes. Based on the identification results, there are five categorical attributes, namely gender, ever married, work type, residence type, and smoking status. The five attributes are transformed into numerical representations using *the One-Hot Encoding* technique so that they can be used in the modeling process. The transformed categorical attributes are shown in Table 4.

Table 4. Categorical Attributes in the Encoding Process

No.	Attributes	Transformation Techniques
1	Gender	One-Hot Encoding
2	Ever married	One-Hot Encoding
3	work_type	One-Hot Encoding
4	Residence type	One-Hot Encoding
5	Smoking status	One-Hot Encoding

Based on Table 4, all categorical attributes are transformed using *One-Hot Encoding*. The process converts each category into binary attributes with values of 0 and 1. Meanwhile, numerical attributes such as age, hypertension, heart_disease, avg_glucose_level, and bmi were retained. The stroke attribute also does not undergo the encoding process because it has been represented in numerical form, namely a value of 0 for the non-stroke class and a value of 1 for the stroke class. Based on all the preprocessing processes that have been carried out, the dataset has changed the number of attributes due to the removal of id attributes and the formation of new attributes through *the One-Hot Encoding* process. Nevertheless, the number of records was maintained as many as 5,110 data. A summary of the dataset changes at each stage of preprocessing is shown in Table 5.

Table 5. Summary of Preprocessing Data Results

Stages	Results
Initial Dataset	5,110 data and 12 attributes
ID removal	The id attributes were removed
BMI Imputation	201 missing value filled median 28.1
One-Hot Encoding	5 categorical attributes transformed
Final Dataset	5,110 data and 22 attributes
Separation of Features and Targets	21 features and 1 target

Based on Table 5, the preprocessing process produced a final dataset of 5,110 data with 22 attributes. After the stroke attribute is separated as the target variable, 21 predictor features (X) and one classification target (y) are obtained. The fixed target distribution consisted of 4,861 non-stroke class data and 249 stroke class data. These results show that the preprocessing process does not change the amount of data or the distribution of the target class. The dataset of preprocessing results is then used in the stage of data sharing, training and testing, as well as handling class imbalances.

3.3 Data Sharing and Handling Class Imbalances

After the preprocessing stage is completed, the dataset is then divided into training data and testing data with an 80:20 ratio. Data training is used in the process of forming Decision Tree and CatBoost models, while data testing is used to evaluate the model's ability to predict data that is not used during the training process. Data distribution was carried out using stratified sampling based on the target stroke class so that the proportion of stroke and non-stroke classes was maintained in both data groups.

Based on the process of sharing 5,110 data, 4,088 data were obtained as training data and 1,022 data as testing data. The results of the complete dataset distribution are shown in Table 6.

Table 6. Data Sharing Training and Testing

Data Type	Percentage	Amount of Data
Data Training	80%	4.088
Data Testing	20%	1.022

Total	**100%**	**5.110**

Based on Table 6, as many as 4,088 data or 80% of the total dataset were used as training data, while 1,022 data or 20% were used as data testing. The results *of stratified sampling* showed that the training data consisted of 3,889 non-stroke class data and 199 stroke class data. Meanwhile, the testing data consisted of 972 non-stroke class data and 50 stroke class data. The distribution shows that the proportion of minority classes is maintained in training data and data testing.

Although the data sharing has been carried out in a stratified manner, the class distribution in the training data still shows a fairly high imbalance. Of the 4,088 training data, 3,889 data included the non-stroke class and only 199 data included the stroke class. This imbalance has the potential to cause models to be more dominant in recognizing the majority class and less optimal in identifying individuals who have had a stroke. Therefore, class imbalance is handled using *the Synthetic Minority Over-sampling Technique* (SMOTE).

The implementation of SMOTE is carried out only on training data after the dataset sharing process. The testing data does not undergo an oversampling process and still retains its original distribution. This approach is carried out to prevent *data leakage*, so that synthetic data produced by SMOTE is not included in data testing. The results of the comparison of class distribution on training data before and after the implementation of SMOTE are shown in Table 7.

Table 7. Distribution of Training Data Classes Before and After SMOTE

Classes	Remarks	Before SMOTE	After SMOTE
0	No Stroke	3.889	3.889
1	Stroke	199	3.889
Total		**4.088**	**7.778**

Based on Table 7, before the implementation of SMOTE, the training data had 3,889 non-stroke class data and 199 stroke class data. After SMOTE was implemented, the number of stroke classes increased to 3,889 data so that the number of both classes became balanced. Thus, the training data after the SMOTE process amounted to 7,778 data consisting of 3,889 non-stroke class data and 3,889 stroke class data. The addition of data in the stroke class was the result of the formation of a synthetic sample by SMOTE, not the addition of new patient data. To show the change in class distribution more clearly, the comparison of the number of classes in the training data before and after the implementation of SMOTE is visualized in Figure 2 and Figure 3.

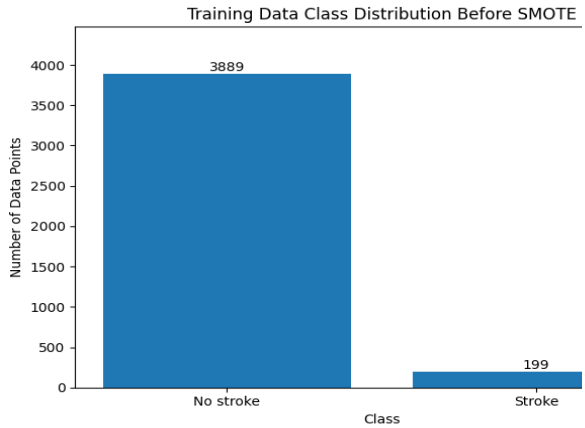

Figure 3. Distribution of Training Data Classes Before SMOTE

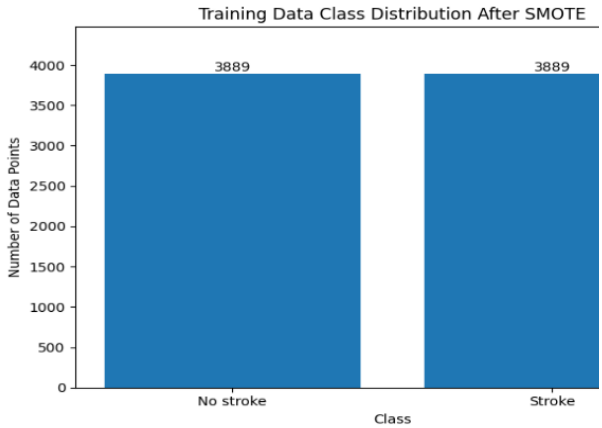

Figure 3. Distribution of Training Data Classes After SMOTE

Based on Figure 2 and Figure 3, the implementation of SMOTE has succeeded in balancing the distribution of classes in training data. Before SMOTE, the non-stroke class dominated with 3,889 data compared to 199 stroke data. After the oversampling process, the stroke class increased to 3,889 data so that both classes had the same number. The SMOTE results data were then used as training data in the process of forming the Decision Tree and CatBoost models, while the evaluation of the two models was still carried out using 1,022 original testing data that did not undergo the SMOTE process.

3.4 Implementation of Decision Tree Algorithm

Before the training process is carried out, the Decision Tree Algorithm is first optimized using the GridSearchCV method. Optimization is done to obtain a combination of parameters that results in the best performance based on the cross-validation process. Optimized parameters include criterion, max depth, min sample split, and min sample leaf. The results of the Decision Tree parameter optimization are shown in Table 8.

Table 8. Best Parameters of Decision Tree

Parameters	Best Value
Criterion	Entropy
Max Depth	None
Min Samples Split	5
Min Samples Leaf	2

Based on Table 8, the best parameters obtained through the GridSearchCV process use the entropy criterion, max depth = None, min samples split = 5, and min samples_leaf = 2. In addition to producing the best combination of parameters, the optimization process also obtained an average F1-Score Cross Validation score of 94.80%, which shows that the model has consistent performance in the data training validation process. After the Decision Tree model is successfully built using the best parameters, a prediction process is carried out on the data testing. The model's performance was then evaluated using the Accuracy, Precision, Recall, F1-Score, and ROC-AUC metrics. The results of the evaluation of the Decision Tree model are shown in Table 9.

Table 9. Decision Tree Evaluation Results

Metrics	Value (%)
Accuracy	92,37
Accuracy	18,18
Recall	16,00
F1 Score	17,02
ROC-AUC	62,85

Based on Table 9, the Decision Tree algorithm obtained an Accuracy of 92.37%. However, the Precision value of 18.18%, the Recall of 16.00%, and the F1-Score of 17.02% indicate that the model still has limited ability to identify stroke classes. In addition, the ROC-AUC value of 62.85% indicates that the model's ability to distinguish stroke and non-stroke classes is still at a moderate level. These results show that although the model has a high level of accuracy, its performance against minority classes still needs to be improved. To find out the distribution of prediction results in each class, an analysis was carried out using *the Confusion Matrix*. The *results of the Confusion Matrix* are shown in Figure 4.

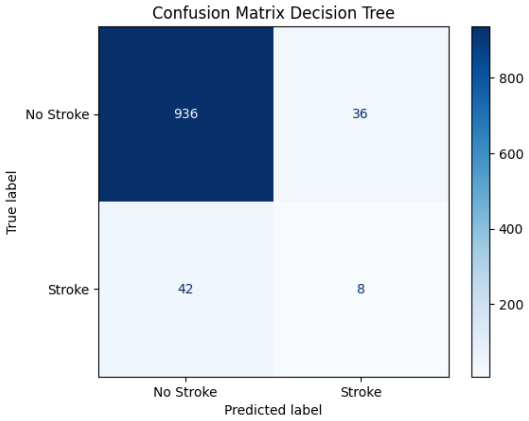

Figure 4. Confusion Matrix Decision Tree

Based on Figure 4, the Decision Tree model successfully classified 936 non-stroke data as non-stroke (*True Negative*) and 8 stroke data as stroke (*True Positive*). On the other hand, there were 36 non-stroke data predicted as stroke (*False Positive*) and 42 stroke data predicted as non-stroke (*False Negative*). A greater number of *False Negatives* than *True Positives* suggests that the model still has difficulty recognizing patients who have actually had a stroke. This condition affects the low Recall value in the stroke class. A more detailed analysis of the model's performance was carried out using *a Classification Report* that presented the Precision, Recall, F1-Score, and data values for each class. The results of *the Classification Report* are shown in Table 10.

Table 10. Classification Report Decision Tree

Classes	Accuracy (%)	Recall (%)	F1 Score (%)	Support
No Stroke	95,71	96,30	96,00	972
Stroke	18,18	16,00	17,02	50

Based on Table 10, the Decision Tree model shows excellent performance in classifying the Non-Stroke class, with a Precision value of 95.71%, Recall of 96.30%, and F1-Score of 96.00%. In contrast, in the Stroke class, the model only obtained a Precision of 18.18%, a Recall of 16.00%, and an F1-Score of 17.02%. The difference in values shows that the model is easier to recognize the majority class than the minority class, even though the training data has been balanced using SMOTE.

In addition to using evaluation metrics, the model's ability to distinguish between the two classes was also analyzed using ROC curves. The ROC curve of the Decision Tree model test is shown in Figure 5.

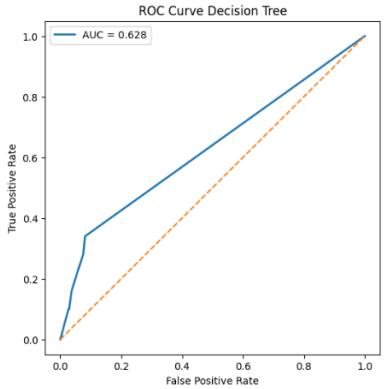

Figure 5. ROC Curve Decision Tree

Based on Figure 5, the Decision Tree algorithm obtained an Area Under Curve (AUC) value of 0.6285 or 62.85%. These values show that the model has sufficient ability to

distinguish stroke and non-stroke classes, but has not yet achieved the optimal level of discrimination. Furthermore, an analysis was carried out on the level of importance of each attribute (*feature importance*) to determine the most influential factors in the classification process. The ten attributes with the highest *feature importance* values are shown in Figure 6.

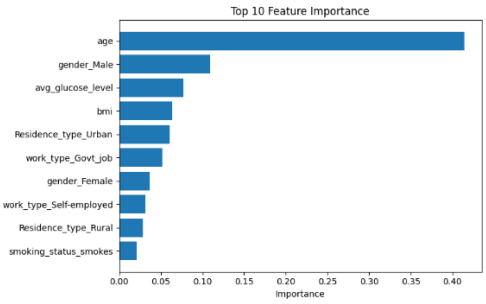

Figure 6. Top 10 Feature Importance Decision Tree

Based on Figure 6, the age attribute is the most dominant factor in the classification process with a *feature importance* value of 0.414466. It was followed by the gender attributes of Male (0.109076), avg glucose level (0.077241**),** bmi (0.063410), and Residence type Urban (0.060747). These results showed that age was the most influential factor in predicting stroke risk in the dataset used, while other attributes contributed less to the modeling process.

Based on all the results of the evaluation that has been carried out, the Decision Tree algorithm is able to build a classification model with an Accuracy level of 92.37%. However, the values of Precision, Recall, F1-Score, and ROC-AUC in the stroke class were still relatively low, indicating that the model's ability to detect stroke cases was not optimal. Therefore, in the next subchapter, the implementation of the CatBoost algorithm will be carried out to find out if the algorithm is able to provide better performance, especially in identifying stroke classes.

3.5 CatBoost Algorithm Implementation

After implementation using the Decision Tree algorithm, this study then implemented the CatBoost algorithm as a comparison method. CatBoost is a decision tree-based *gradient boosting* algorithm that builds a number of trees gradually to improve prediction performance. In this Research Stage, the CatBoost model was trained using SMOTE training data and evaluated using the same data testing as the Decision Tree algorithm so that the results of the two models could be objectively compared. Before the training process is carried out, the CatBoost model is first optimized using *the GridSearchCV* method to obtain the best combination of parameters. Optimized parameters include *depth*, *learning rate*, *iterations*, and *l2 leaf reg*. The results of parameter optimization are shown in Table 11.

Table 11. Best Parameters of CatBoost

Parameters	Best Value
Depth	8
Learning Rate	0,1
Iterations	200
L2 Leaf Reg	1

Based on Table 11, the best parameters obtained from the *GridSearchCV* process are depth = 8**,** learning rate = 0.1**,** iterations = 200**,** and L2 leaf regularization = 1. In addition, the optimization process resulted in an average F1-Score Cross Validation score of 96.52%, which shows that the CatBoost model has consistent performance in the data training validation process. Furthermore, the CatBoost model with the best parameters is used to predict the testing data. Model performance was evaluated using the Accuracy, Precision, Recall, F1-Score, and ROC-AUC metrics. The results of the CatBoost model evaluation are shown in Table 12.

Table 12. CatBoost Evaluation Results

Metrics	Value (%)
Accuracy	94,03
Accuracy	17,65
Recall	6,00
F1 Score	8,96
ROC-AUC	78,39

Based on Table 12, the CatBoost algorithm obtained an Accuracy of 94.03%, higher than the results obtained by the Decision Tree algorithm. However, the Precision value of 17.65%, the Recall of 6.00%, and the F1-Score of 8.96% indicate that the model still has difficulty in recognizing stroke classes. On the other hand, the ROC-AUC value of 78.39% shows that CatBoost's ability to distinguish stroke and non-stroke classes is better than Decision Tree. To find out the distribution of prediction results in each class, an analysis was carried out using *the Confusion Matrix*. The *results of the Confusion Matrix* are shown in Figure 7.

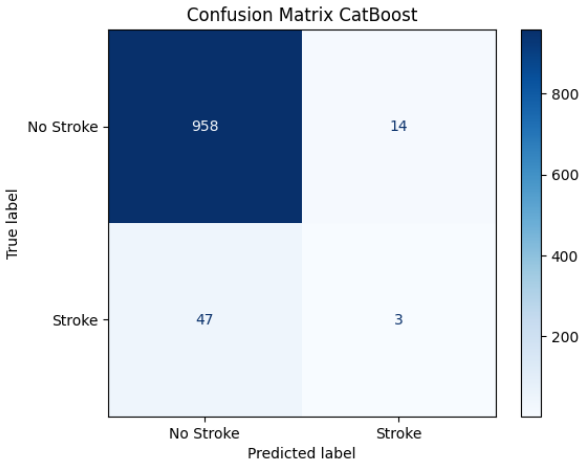

Figure 7. CatBoost Matrix Confusion

Based on Figure 7, the CatBoost model successfully classified 958 non-stroke data as non-stroke (*True Negative*) and 3 stroke data as stroke (*True Positive*). In addition, there were 14 non-stroke data predicted as stroke (*False Positive*) and 47 stroke data predicted as non-stroke (*False Negative*). The results showed that the model had a low error rate in the non-stroke class, but still had difficulty detecting patients who actually had a stroke. This condition causes the Recall value in the stroke class to be low.

To obtain more detailed information about the performance of the model in each class, the analysis was carried out using *the Classification Report*. The results *of the Classification Report* are shown in Table 13.

Table 13. CatBoost Classification Report

Classes	Accuracy (%)	Recall (%)	F1 Score (%)	Support
No Stroke	95,32	98,56	96,91	972
Stroke	17,65	6,00	8,96	50

Based on Table 13, the CatBoost algorithm showed excellent performance in the non-stroke class, with Precision of 95.32%, Recall of 98.56%, and F1-Score of 96.91%. In contrast, in the stroke class**,** the model only obtained a Precision of 17.65%, a Recall of 6.00%, and an F1-Score of 8.96%. These results show that CatBoost is more likely to classify the data as non-stroke, so the ability to recognize stroke classes is still low. In addition to using evaluation metrics, the model's ability to distinguish between the two classes was analyzed using ROC curves. The ROC curve of the CatBoost model test is shown in Figure 8.

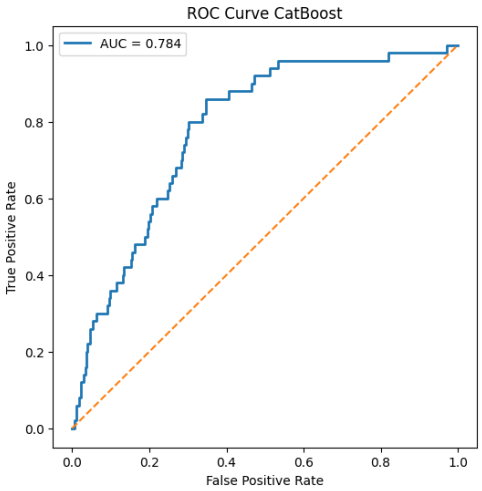

Figure 8. ROC Curve CatBoost

Based on Figure 8, the CatBoost algorithm obtained an Area Under Curve (AUC) value of 0.7839 or 78.39%. This value shows that CatBoost has a better ability to distinguish stroke and non-stroke classes compared to the Decision Tree algorithm which obtained an AUC value of 62.85%. Furthermore, an analysis of the level of *feature importance* was carried out to determine the most influential factors in the classification process. The ten attributes with *the highest feature importance* values are shown in Figure 9.

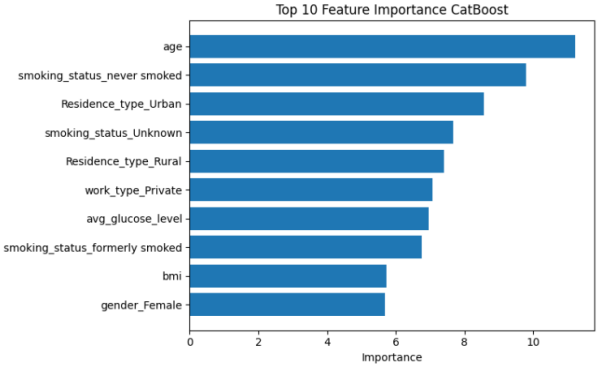

Figure 9. Top 10 Feature Importance of CatBoost

Based on Figure 9, the age attribute is the factor that has the greatest contribution to the classification process with *a feature importance* value of 11.22. Furthermore, it was followed by the attributes of smoking status never smoked (9.79), Residence type Urban (8.56), smoking status Unknown (7.68), Residence type Rural (7.41), work type Private (7.07), avg glucose level (6.95), smoking status formerly smoked (6.76), bmi (5.72), and gender Female (5.67). These results show that age and smoking status are factors that make a major contribution to the process of predicting stroke risk using the CatBoost algorithm.

Based on the results of implementation and evaluation, the CatBoost algorithm obtained an Accuracy of 94.03%, higher than Decision Tree. In addition, the ROC-AUC value of 78.39% indicates better discriminating ability in distinguishing stroke and non-stroke classes. However, the Recall and F1-Score values in the stroke class are still low, so the model is not able to detect most stroke cases optimally. Therefore, in the next subchapter, a thorough comparison between the Decision Tree and CatBoost algorithms is carried out to determine the algorithm that provides the best performance based on all evaluation metrics.

3.6 Comparison of Decision Tree Algorithm and CatBoost Algorithm

After the implementation using the Decision Tree and CatBoost algorithms, the next step is to compare the performance of the two algorithms based on several evaluation metrics, namely Accuracy, Precision, Recall, F1-Score, and ROC-AUC. Comparisons are carried out

using the same testing data so that the results of the evaluation can be compared objectively. The results of the performance comparison of the two algorithms are shown in Table 14.

Table 14. Decision Tree and CatBoost Performance Comparison

Metrics	Decision Tree (%)	CatBoost (%)
Accuracy	92,37	94,03
Accuracy	18,18	17,65
Recall	16,00	6,00
F1 Score	**17,02**	8,96
ROC-AUC	62,85	78,39

Based on Table 14, the CatBoost algorithm obtained an Accuracy of 94.03%, higher than Decision Tree which obtained 92.37%. In addition, CatBoost also has a ROC-AUC of 78.39%, higher than Decision Tree which obtained 62.85%, thus showing a better ability to distinguish stroke and non-stroke classes overall.

On the other hand, Decision Tree obtained a Precision score of 18.18%, a Recall of 16.00%, and an F1-Score of 17.02%, which is higher than CatBoost. These results suggest that Decision Tree has a better ability to detect stroke classes in testing data, although the accuracy rate is lower.

Next, the algorithm that provides the best performance in each evaluation metric is identified. The results of the comparison are shown in Table 15.

Table 15. Best Algorithm Based on Evaluation Metrics

Metrics	Best Algorithm
Accuracy	CatBoost
Accuracy	Decision Tree
Recall	Decision Tree
F1 Score	Decision Tree
ROC-AUC	**CatBoost**

Based on Table 15, CatBoost excels in the Accuracy and ROC-AUC metrics, while Decision Tree excels in the Precision**,** Recall**,** and F1-Score **metrics.** These results show that the two algorithms have different characteristics. CatBoost is better at generating correct predictions overall and has better class discrimination capabilities, whereas Decision Tree is more effective at detecting stroke cases on test datasets.

In addition to comparing the results of the evaluation on the testing data, this study also compared the average value of F1-Score Cross Validation obtained during the parameter optimization process using *GridSearchCV*. The results of the comparison are shown in Table 16.

Table 16. Comparison of Cross Validation Values

Algorithm	F1-Score Cross Validation (%)
Decision Tree	94,80
CatBoost	96,52

Based on Table 16, CatBoost obtained an average F1-Score Cross Validation score of 96.52%, higher than Decision Tree which obtained 94.80%. These results show that CatBoost has a more stable performance during the training and validation process using SMOTE training data. Furthermore, a comparison of attributes that have the greatest contribution to the classification process is carried out based on *feature importance* values. The top five attributes of each algorithm are shown in Table 17.

Table 17. Feature Importance Comparison

Ranking	Decision Tree	Catboost

1	Age	Age
2	Gender Male	Smoking Status Never Smoked
3	Avg Glucose Level	Residence Type Urban
4	BMI	Smoking Status Unknown
5	Residence Type Urban	Residence Type Rural

Based on Table 17, both algorithms place the age attribute as the most influential factor in the process of predicting stroke risk. However, there is a difference in the next attribute. Decision Tree placed more emphasis on the attributes of gender, avg glucose level**,** and bmi, while CatBoost contributed more to attributes related to smoking status and type of residence. These differences show that each algorithm has a different mechanism in forming classification patterns.

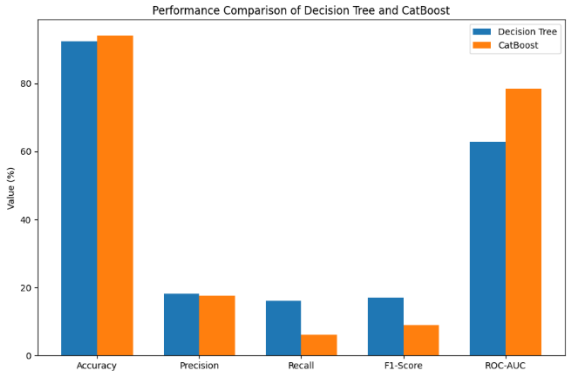

Figure 10. Decision Tree and CatBoost Performance Comparison

Based on Figure 10, it can be seen that CatBoost has higher Accuracy and ROC-AUC values than Decision Tree. In contrast, Decision Tree earns higher Precision, Recall, and F1-Score scores. The visualization makes it clear that there is no single algorithm that excels in all evaluation metrics.

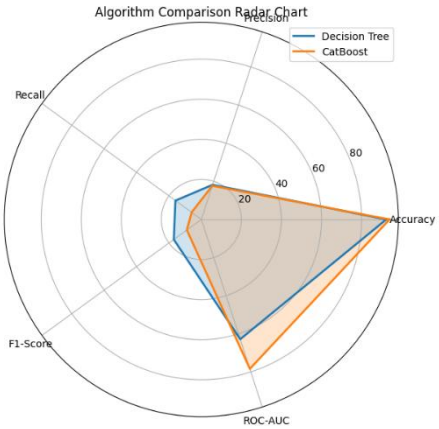

Figure 11. Radar Chart Comparison Algorithm

Based on Figure 11, the performance profiles of the two algorithms show different patterns. The CatBoost area is larger on the Accuracy and ROC-AUC metrics, while the Decision Tree has a larger area on the Precision, Recall, and F1-Score metrics. This shows that there is a difference in performance characteristics between the two algorithms in predicting stroke risk.

Conclusion

This study successfully compared the performance of Decision Tree and CatBoost algorithms in predicting stroke risk using the Healthcare Stroke Prediction Dataset through the stages of data preprocessing, data sharing using the stratified train-test split method, handling

class imbalances using SMOTE, and parameter optimization with GridSearchCV. The results of the preprocessing produced a dataset of 5,110 data with 21 predictor attributes and 1 target attribute which were then used in the training and testing of the model. Based on the test results, the Decision Tree algorithm obtained an Accuracy score of 92.37%, Precision of 18.18%, Recall of 16.00%, F1-Score of 17.02%, and ROC-AUC of 62.85%. Meanwhile, the CatBoost algorithm obtained Accuracy of 94.03%, Precision of 17.65%, Recall of 6.00%, F1-Score of 8.96%, and ROC-AUC of 78.39%. These results show that CatBoost performs better in terms of Accuracy and ROC-AUC, while Decision Tree outperforms Precision, Recall, and F1-Score, making it better at detecting stroke classes in the test dataset. In addition, the feature importance analysis showed that the age attribute was the most influential factor in both algorithms in the stroke risk prediction process. The next important attribute difference shows that each algorithm has different characteristics in forming a classification pattern for patient data. Based on the overall comparison results, CatBoost is more suitable for use when the primary goal is to achieve a higher level of accuracy and class discrimination capability, while Decision Tree is more appropriate if the primary focus of the study is to improve the detection of stroke cases, as demonstrated by better Recall and F1-Score scores. Thus, the selection of algorithms should be adjusted to the needs of the implementation of the stroke risk prediction system.

References

[1] Z. S. Iswadi Hamzah, "Analisa Classification Decision Tree C45 dan Naïve Bayes Pada Indikasi Penyakit Diabetes Menggunakan Rapid Miner," *J. Nas. Teknol. Komput.*, vol. 4, no. 1, pp. 25–33, 2024.

[2] Z. Sitorus, E. Hariyanto, and F. Kurniawan, "Analysis of artificial intelligence machine learning technology for mapping and predicting flood locations in Pahlawan Batu Bara village," *Int. J. Comput. Sci. Math. Eng.*, vol. 2, no. 2, pp. 281–288, 2023.

[3] M. Sunjaya, Z. Sitorus, M. Iqbal, and A. P. U. Siahaan, "Analysis of machine learning approaches to determine online shopping ratings using naïve bayes and svm," *Int. J. Comput. Sci. Math. Eng.*, vol. 3, no. 1, pp. 7–16, 2024.

[4] Z. Sitorus and M. Indra, "Analysis of Room Allocation Based on Age in Nursing Homes Using the C4 . 5 Decision Tree Method," vol. 1, no. 2, pp. 153–157, 2024, doi: 10.61306/jitcse.v1i2.

[5] H. Rodhiy and Z. Sitorus, "Data Mining Menggunakan Algoritma Apriori Dalam Menentukan Tarif Pajak Penghasilan Di Oenity," *Bull. Inf. Technol.*, vol. 4, no. 2, pp. 198–204, 2023.

[6] I. Werdiningsih *et al.*, "Analisis Prediksi Stroke Menggunakan Pendekatan Decision Tree dengan Seleksi Fitur dan Neural Network," *J. Sist. Cerdas*, vol. 6, no. 3, pp. 213–221, 2023.

[7] M. A. Kadhim and A. M. Radhi, "Machine Learning Prediction of Brain Stroke at an Early Stage," *Iraqi J. Sci.*, vol. 64, no. 12, pp. 6596–6610, 2023, doi: 10.24996/ijs.2023.64.12.39.

[8] Y. Aulia, A. Andriyansyah, S. Suharjito, and S. W. Nensi, "Analisis prediksi stroke dengan membandingkan tiga metode klasifikasi decision tree, naïve bayes, dan random forest," *J. Ilmu Komput. Dan Inform.*, vol. 3, no. 2, pp. 89–98, 2024.

[9] I. O. Herlistiono and S. Violina, "Model Prediksi Risiko Stroke Menggunakan Machine Learning," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 7, no. 4, pp. 1230–1238, Jul. 2024, doi: 10.31539/INTECOMS.V7I4.10942.

[10] K. Wirastuti, N. S. Riasari, D. Djannah, and M. Silviana, "Upaya Pencegahan Stroke melalui Skrining Skor Risiko Stroke dengan Intervensi Penyuluhan dan Pemeriksaan Faktor Risiko Stroke di Kelurahan Bojong Salaman Kecamatan Pusponjolo Selatan Semarang Barat," *J. ABDIMAS-KU J. Pengabdi. Masy. Kedokt.*, vol. 2, no. 1, pp. 23–

29, Jan. 2023, doi: 10.30659/abdimasku.2.1.23-29.

[11] C. Paramita, C. S. Simbolon, A. S. Pamungkas, J. M. Triono, E. P. Widi Utomo, and E. R. Subhiyakto, "Analisis Pengaruh SMOTE terhadap Kinerja Model KNN untuk Prediksi Risiko Stroke," *J. Inform. J. Pengemb. IT*, vol. 10, no. 4, pp. 978–988, Sep. 2025, doi: 10.30591/JPIT.V10I4.8809.

[12] L. Amalia, "Clinical significance of Platelet-to-White Blood Cell Ratio (PWR) and National Institute of Health Stroke Scale (NIHSS) in acute ischemic stroke," *Heliyon*, vol. 6, no. 10, 2020, doi: 10.1016/j.heliyon.2020.e05033.

[13] M. R. Firmansyah and Y. P. Astuti, "Stroke Classification Comparison with KNN through Standardization and Normalization Techniques," *Adv. Sustain. Sci. Eng. Technol.*, vol. 6, no. 1, p. 02401012, Jan. 2024, doi: 10.26877/ASSET.V6I1.17685.