Comparison of Classification Results of Categorical Boosting and Extreme Gradient Boosting Methods in Obesity Class Diagnosis

Jonatan[1], Giska Auria[2], Anita Desiani[3], Ali Amran[4], Indri Ramayanti[5], Irmeilyana [6]

{jonatancy156@gmail.com[1], giskaauria89@gmail.com[2], anita_desiani@unsri.ac.id[3], ali_amran@mipa.unsri.ac.id[4], indri_ramayanti@um-palembang.ac.id[5], irmeilyana@unsri.ac.id[6]}

Department of Mathematics, Faculty of Mathematics and Natural Science, Universitas Sriwijaya[1,2,3,4,6], Department of Parasitology, Universitas Muhammadiyah Palembang[5]

Corresponding author: anita_desiani@unsri.ac.id

Abstract. Obesity, caused by a chronic energy imbalance in which calorie intake exceeds expenditure, is a growing global health problem that negatively impacts quality of life. Early detection is crucial for mitigation. This study compares the performance of the Extreme Gradient Boosting (XGBoost) and Categorical Boosting (CatBoost) algorithms for obesity status classification. The research methodology included data collection, pre-processing, and model evaluation using two validation strategies: percentage split and k-fold cross-validation. The results show the superiority of CatBoost. In the percentage split evaluation, CatBoost achieved an accuracy of 92.53%, which outperformed XGBoost at 91.67%. Similarly, using K-Fold Cross-Validation, CatBoost yielded an accuracy of 91.95% while XGBoost reached 90.46%. Additionally, CatBoost consistently produced superior average values for recall, F1-score, and precision. It is concluded that CatBoost provides a more accurate and robust model for obesity classification compared to XGBoost. Future research could explore feature engineering techniques or other algorithms to further enhance predictive accuracy.

Keywords: Boosting algorithm, CatBoost, classification, machine learning, XGBoost.

1 Introduction

Obesity is a condition in which the amount of fat in the body exceeds the normal weight limit [1]. Obesity occurs when there is excessive fat in adipose tissue, which can damage health [2]. Obesity occurs when the energy obtained from food exceeds the body's needs [1]. Lifestyle factors and low physical activity are associated with an increase in obesity prevalence [3]. The increase in obesity has a significant impact on health disorders and a decline in quality of life. To reduce the number of obesity cases, early detection is very important, especially in people who are at high risk. One way to detect early signs is to analyze data using machine learning models. Machine learning is a field of science that focuses on developing algorithms and

SICBAS 2025, November 06, Palembang, Indonesia

DOI 10.4108/eai.6-11-2025.2364304

statistical models. These models enable computer systems to perform tasks without explicit instructions and teach machines to process data more efficiently [4]. One application of machine learning is classification. Classification is a type of data analysis used to help determine the most suitable category for a piece of data or sample [5]. Some common algorithms used in machine learning are Extreme Gradient Boosting and Categorical Boosting. Extreme Gradient Boosting (XGBoost) is an algorithm developed with a gradient boosting approach that uses decision trees as the basis for classification [6]. This algorithm has good capabilities in processing unbalanced data and can identify the most important features in a data set [7]. Sukmawati *et al.* [8] applied the XGBoost algorithm for obesity classification and obtained an excellent accuracy of 92%. This study has limitations, namely the relatively small size of the dataset, which has the potential for overfitting.

Salsabil *et al.* [9] predicted diabetes using the XGBoost algorithm and achieved a good accuracy rate of 76%. The data used in this study has 9 attributes and 768 entries. Wicaksono *et al.* [10] applied the XGBoost algorithm for AIDS classification and achieved an accuracy rate of over 90%. This study only applied one algorithm without comparing it to other algorithms, making it difficult to determine whether XGBoost is the best choice for AIDS classification. The XGBoost algorithm is difficult to determine its parameters, and it takes a considerable amount of time. Categorical Boosting (CatBoost) is an algorithm based on GBDT (Gradient-Boosting Decision Tree) that can be used for classification, regression, and prediction [11]. The CatBoost algorithm has the advantage of being able to reduce bias caused by data leakage during training, improve model accuracy, reduce overfitting, and increase prediction speed (11). Darmawan *et al.* [12] Implemented CatBoost to predict diabetes and achieved an accuracy of 63.46%. This study only used 768 data points and 8 attributes, resulting in less than optimal accuracy.

Irfannandhy *et al.* [13] analyzed the performance of the CatBoost algorithm to predict the risk of diabetes and achieved an accuracy rate of over 80%. The dataset used in this study was relatively small, so it was necessary to add a variety of variables to improve the reliability of the algorithm. Haya and Ramme [14] applied the CatBoost algorithm to predict diabetes and achieved an accuracy of 76.04%. This study will compare the performance of the XGBoost and CatBoost algorithms in classifying obesity using a relevant dataset to obtain the best prediction results. The testing of both algorithms uses the k-fold cross-validation and percentage split methods. In both methods, the dataset is divided into 20% testing data and 80% training data, with k in k-fold cross-validation set to 7, and alternately tested as training and testing data 7 times. There are 4 labels used in this study, namely underweight, normal, overweight, and obesity. The performance of both algorithms will be evaluated based on precision, accuracy, F1-score, and recall metrics to determine the best algorithm for obesity classification.

2 Literature Review

2.1 Extreme gradient boosting

Extreme Gradient Boosting is an ensemble algorithm that focuses on decision trees and uses gradient boosting techniques [15]. The XGBoost algorithm can perform calculations with great precision and accuracy, which is why it is widely used in computing environments due to its ability to model new attributes and classify data. The XGBoost algorithm works very quickly because it is capable of performing many tasks simultaneously, discarding unnecessary data

structures, and is specifically designed to be compatible with the way modern computers work. This algorithm uses three types of gradient boosting, including gradient boosting for machine learning, stochastic gradient boosting with subsampling, and regular gradient boosting with L1 and L2 regularization. Previous research applied the XGBoost algorithm in classifying obesity is Sukmawati *et al.* [8]. The study used a dataset obtained from Kaggle and presented in CSV format with a total of 2111 data points consisting of 16 attributes and 1 target attribute. This study produced an accuracy, precision, and recall of 92%. Abdurrahman *et al.* [16] also applied the XGBoost algorithm using Hyperparameter Gridsearch and Random Search to classify diabetes. This study uses the Pima Indian Diabetes dataset obtained from UCI Machine Learning. The dataset used consists of 768 data records. The accuracy of XGBoost performance with hyperparameters reaches 75%, while using random search results, the algorithm performance reached 95%.

2.2 Categorical boosting

Categorical Boosting is a decision tree-based algorithm that has proven to be very effective in handling machine learning involving categorical and heterogeneous data [17]. CatBoost has been widely used in various research studies to address classification problems, including in the fields of finance, medicine, failure prediction, fraud detection, psychology, anomaly detection, cybersecurity, materials engineering, biochemistry, biology, and many other fields [17]. Several previous studies have applied the CatBoost algorithm in classification. Purbolingga *et al.* [18] applied the CatBoost algorithm to classify heart disease and achieved an accuracy rate of over 85%. This study utilized a dataset obtained from the UCI Machine Learning Repository, comprising 11 independent features and 1 dependent feature, with a total of 918 data points. The study was conducted by Sabili *et al.* [19] also applied the CatBoost algorithm to classify diabetes, and the results of the study showed a fair. high accuracy rate of 98.63%. This study developed a prediction model for diabetes classification using the CatBoost algorithm, with a dataset of 1460 samples divided into 20% for testing and 80% for training.

3 Methodology

3.1 Data collection

This research leverages the 'Obesity Dataset', a public repository accessible via Kaggle. Comprising 1,610 instances with 15 distinct variables, the dataset provides a comprehensive overview of various weight categories, ranging from underweight to obesity. This dataset contains 1,610 data points and has 15 attributes. There are 4 label classes in this dataset, namely underweight, normal, overweight, and obesity. The attributes used can be seen in **Table 1**.

Table 1. Dataset attribute information

Attribute Name	Dataset Attribute Information	
	Attribute Type	*Value*
Sex	Boolean	1. Male (712) 2. Female 898
Age	Numerical	Integers
Height	Numerical	Integers (cm)

Attribute Name	Dataset Attribute Information	
	Attribute Type	*Value*
Genetic Factors of Obesity.	Boolean	1. Positive (266) 2. Negative (11174)
Dietary Habits: Fast Food	Boolean	1. Positive (436) 2. Negative (1174)
Intensity of Vegetable Consumption.	Ordinal	1. Rarely (400) 2. Sometimes (708) 3. Always (502)
Daily Food Quantity	Ordinal	1. 1 and 2 (444) 2. 3 (928) 3. > 3 (238)
Eating Patterns Outside of Main Meal Times	Ordinal	1. Rarely (346) 2. Sometimes (564) 3. Usually (417) 4. Always (283)
The habit of Smoking.	Boolean	1. Positive (492) 2. Negative (1118)
Amount of Water Consumed	Ordinal	1. Less than 1 liter (456) 2. Between 1 and 2 liters (523) 3. More 2 liters (631)
Calculation of Energy Intake	Boolean	1. Positive (286) 2. Negative (1324)
Exercise Routine	Ordinal	1. No activity (206) 2. Range of 1-2 days (290) 3. Range of 3-4 days (370) 4. Range of 5-6 days (358) 5. More than 6 days (386)
Time for the use of Technology	Ordinal	1. Low usage : < 2 hours (382) 2. Moderate used : 3 – 5 hours (826) 3. High use : > 5 hours (402)
Vehicle Type	Nominal	1. Automobile (660) 2. Motorbike (94) 3. Bike (116) 4. Public transportation (602) 5. Walking (138)
Body Mass Index Classification	Ordinal	1. Underweight (73) 2. Normal (658) 3. Overweight (592) 4. Obesity (287)

3.2 Pre-processing data

The pre-processing stage begins with handling outliers by removing deviating observations in each column to reduce noise. Next, numerical features are normalized using the Max-Abs Scaler

into the range [0, 1] to ensure model stability. To avoid estimation bias and data leakage, the dataset is first divided into testing and training data using the K-Fold Cross Validation and Percentage Split methods. The Synthetic Minority Over-sampling Technique (SMOTE) is then applied only to the training data to address class imbalance in the four obesity categories. This step produces a balanced class distribution with 435 samples per category in the training data, without contaminating the test data with synthetic samples.

3.3 Data splitting and handling imbalance

The model evaluation process was carried out by dividing the dataset using the percentage split method to separate the data into training and test datasets. In this study, the division was carried out with a ratio of 80:20, where 80% of the data was randomly selected to train the model and the remaining 20% was allocated specifically for testing to objectively assess the predictive ability of the algorithm. After the division, performance validation was reinforced by applying the K-Fold Cross-Validation method. Using a value of k=7, the dataset was divided into seven parts, where six parts were used as training data and one part was used as test data alternately. This layered approach ensured that the resulting accuracy was reliable and able to represent the generalization of the model on different data subsets.

3.4 Synthetic minority over-sampling technique (SMOTE)

To mitigate the bias inherent in the unequal distribution of the four weight classes, specifically the minority categories, data augmentation was performed using the Synthetic Minority Over-sampling Technique (SMOTE). To ensure the accuracy of the results, the SMOTE procedure was performed exclusively on the training data set immediately after the data splitting stage was completed. This approach effectively balanced the class distribution to achieve 435 samples per category without affecting the purity of the test data. By limiting the data augmentation process to the training phase, the risk of data leakage and estimation bias can be fully mitigated, so that the high accuracy achieved by the model is a true reflection of its generalization ability to real data that has never been seen before.

3.5 XGBoost algorithm implementation

XGBoost is implemented by sequentially constructing an ensemble of decision trees to optimize predictive accuracy [20]. The process begins by initializing the prediction probability Pr_i^1 for all samples $i = 1,2,\dots,n$. In each iteration t, The algorithm calculates the residual, defined as the difference between the actual target Y_i and the current predicted probability Pr_i^t as shown in Equation (1).

$$Residual_i^t = Y_i - Pr_i^t , \tag{1}$$

where $Residual_i^t$ denotes the residual value of the i-th sample at iteration t, Y_i represent the actual target value of the i-th sample, and Pr_i^t denotes the predicted probability at iteration t.

To determine the optimal split points, XGBoost evaluates the attribute coverage $Cover(A)$ using Equation (2), which measures the variation in predicted probabilities. Simultaneously, a similarity score (SS) is computed for each node Equation (3) to assess the quality of the split.

The gain from a specific attribute is then derived by comparing the similarity scores of the left and right child nodes against the parent node Equation (4).

$$Cover(A) = \sum_{i=1}^{n}(Pr_i^t(1 - Pr_i^t)) , \quad (2)$$

where Cover(A) denotes the attribute coverage of attribute A, Pr_i^t represents the predicted probability of the i-th sample at iteration t, and n is the total number of samples.

$$SS_{node} = \frac{(\sum_{i=1}^{n} Residual_i)^2}{\sum_{i=1}^{n}\left(Pr_i^t\,(1-Pr_i^t)\right)+\lambda} , \quad (3)$$

where SS_{node} denotes the similarity score of a node, $Residual_i$ represents the residual value of the i-th sample, Pr_i^t denotes the predicted probability at iteration t, n is the total number of samples, and λ is the regularization parameter.

$$Gain\ (A) = SS_{left} + SS_{right} - SS_{root} , \quad (4)$$

where $Gain\ (A)$ represents the gain obtained from splitting attribute A, SS_{left} and SS_{right} denote the similarity scores of the left and right child nodes, respectively, and SS_{root} denotes the similarity score of the parent (root) node.

Once the tree structure is determined, the leaf output values are calculated using Equation (5) to update the model. These values are transformed into log-odds in Equation (6), which are then updated with a learning rate η Equation (7). Finally, the updated log-odds are converted back into probabilities using the sigmoid function Equation (8) to produce the final classification result.

$$Output\ (A)_i = \frac{\sum_{i=1}^{n} Residual_i}{\sum_{i=1}^{n}(Pr_i\,(1-Pr_i))+\lambda} , \quad (5)$$

where $Output\ (A)_i$ denotes the output value of the leaf node corresponding to attribute A for the i-th sample, $Residual_i$ represents the residual value of the i-th sample, Pr_i denotes the predicted probability of the i-th sample, n is the total number of samples, and λ is the regularization parameter.

$$\log odds_i^t = \log\left(\frac{Pr_i^t}{1-Pr_i^t}\right), \quad (6)$$

where $\log odds_i^t$ denotes the lod-odds value of the predicted probability for the i-th sample at iteration t, and Pr_i^t represents the predicted probability at iteration t.

$$Pr_i^{t+1} = \log odds_i^t + (\eta \times output\ (A)_i) , \quad (7)$$

where Pr_i^{t+1} denotes the updates predicted probability for the i-th sample at iteration $t+1$, $\log odds_i^t$ represents the log-odds value at iteration t, η denotes the learning rate, and $output\ (A)_i$ represents the leaf output value corresponding to attribute A.

$$Sigmoid\ (Pr_i^{t+1}) = \frac{exp^{Pr_i^{t+1}}}{1+ exp^{Pr_i^{t+1}}}\ , \tag{8}$$

where $\boldsymbol{Sigmoid}(.)$ denotes the sigmoid activation function, and $\boldsymbol{Pr_i^{t+1}}$ represents the updated log-odds value for the $\boldsymbol{i}$-th sample.

3.6 CatBoost algorithm implementation

Categorical Boosting is a modification of the Gradient Boosting algorithm commonly used in classification problems. Gradient Boosting works by gradually building predictive models through the combination of several simple prediction models, usually in the form of decision trees. The goal is to minimize the loss function that can be systematically derived (differentiable loss function) [21]. The main steps in Gradient Boosting are as follows:

1. Initialize the initial model by minimizing the loss function.
2. Perform iterations starting from $i = 1$ to $j = M$ with the following steps.
   a. Calculate the residual (prediction error) by minimizing the loss function using the following Equation (9).

$$r_{ij} \ = \ - \left[\frac{\partial L(y_i, F(\boldsymbol{x_i}))}{\partial F(\boldsymbol{x}_i)}\right] F(\boldsymbol{x_i}) = F_{j-1}(\boldsymbol{x_i})\ . \tag{9}$$

   b. Using pseudo-residuals to train the base model $h_j(\boldsymbol{x})$.
   c. Solve the following optimization problem to calculate the multiplier γ_j to minimize losses using the following Equation (10).

$$\gamma_j = \arg\min \sum_{i=1}^{n} L\ (y_i, F_{j-1}(\boldsymbol{x_i}) \ + \ \gamma h_j(x_i))\ . \tag{10}$$

   d. Updating the model $F_j(x)$ by adding the results of the new weak learner using the following Equation (11).

$$F_j(\boldsymbol{x}) = \ F_{j-1}(\boldsymbol{x}) \ + \gamma_j h_j(\boldsymbol{x})\ . \tag{11}$$

3. Final model from the combined results of iterative weak learners $F_j(\boldsymbol{x})$.

One of CatBoost's approaches is Ordered Boosting. Ordered Boosting is an efficient modification of standard Gradient Boosting. The outline of the Ordered Boosting algorithm is as follows.

1. Perform randomization or random permutation to select the order of training data.
2. Model $M1 \ldots Mi$, each M_i model is formed using only the first sample in the permutation sequence.
3. Use the Mj model to obtain the j-th residual (error difference).

3.7 Evaluation of results (confusion matrix)

Model validation was conducted using a confusion matrix analysis to visualize the discrepancy between predicted labels and actual data classes [22]. This method maps the classification

performance into four quadrants: True Positives (TP) and True Negatives (TN) represent the model's ability to correctly identify the positive and negative classes, respectively.

In terms of misclassification, the model distinguishes between False Positives (FP) where negative data is wrongly predicted as positive and False Negatives (FN), where positive data is incorrectly classified as negative. These values are essential for calculating the model's overall Accuracy using Equation (12), as well as other specific metrics such as precision and recall [23].

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+TN+FN+FP}}. \quad (12)$$

Precision is the degree to which a system provides answers that match the information requested by the user. The Equation for calculating precision can be seen in Equation (13) [24].

$$Precision = \frac{\text{TP}}{\text{TP+FP}}. \quad (13)$$

Recall is the degree of success of a system in retrieving relevant data or information. The Equation for calculating recall can be seen in Equation (14) [25].

$$Recall = \frac{\text{TP}}{\text{TP+FN}}. \quad (14)$$

F1-Score is the result of calculating a combination of precision and recall values, which is then referred to as a measurement value. The Equation for calculating F1-Score can be seen in Equation (15) (26).

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision \pm Recall}. \quad (15)$$

3.8 Software and computational environment

The research was implemented using Python on Google Colab. Key libraries included Pandas and NumPy for data processing, Scikit-learn for preprocessing and model evaluation. Class balancing utilized Imbalanced-learn (SMOTE), while the classification models were built using the specific XGBoost and CatBoost libraries.

4 Results and Discussion

To determine the most effective algorithm and method for classifying obesity, a performance comparison between the XGBoost and CatBoost algorithms is necessary. This comparison is carried out by analyzing the confusion matrix, accuracy, precision, recall, and F1-score of each algorithm. The evaluation is performed using two techniques, name K-Fold cross-validation and Percentage Split, to obtain objective and comprehensive results.

4.1 XGBoost algorithm

The results of obesity diagnosis using the XGBoost algorithm can be tested using a confusion matrix, which forms the basis for calculating accuracy, precision, recall, and F1-score as indicators of the algorithm's success. The confusion matrix is obtained from the classification process by comparing the actual class labels with the predicted results. In this study, the confusion

matrices were generated using two validation approaches, namely the Percentage Split and K-Fold Cross-Validation methods. The classification results produced by both methods are presented in **Table 2**.

Table 2. Confusion matrix of XGBoost with percentage pplit and K-Fold cross-validation method

Confusion Matrix Of XGBoost Algorithms						
			Prediction			
	Class		0	1	2	3
Percentage Split	Actual	0	86	2	0	0
		1	4	79	9	1
		2	0	8	80	1
		3	0	0	2	77
			Prediction			
	Class		0	1	2	3
K-Fold Cross Validation	Actual	0	432	2	1	0
		1	13	366	56	0
		2	3	51	372	9
		3	0	0	5	430

From **Table 2**, it can be seen that there are 348 data points predicted using the percentage split testing technique. Data correctly predicted as label 0 amounted to 86 data. Data correctly predicted as label 1 amounted to 79 data. Data correctly predicted as label 2 amounted to 80 data. Data correctly predicted as label 3 amounted to 77 data. In the k-fold cross-validation testing technique, it is known that there are 1,740 data points tested. The number of data points correctly predicted as label 0 is 432. The number of data points correctly predicted as label 1 is 366. The number of data points correctly predicted as label 2 is 372. The number of data points correctly predicted as label 3 is 430.

4.2 CatBoost algorithm

The results of obesity diagnosis using the CatBoost algorithm can be tested using a confusion matrix, which forms the basis for calculating precision, recall, F1-score, and accuracy as indicators of the algorithm's success. The results of the XGBoost algorithm classification performed using the Percentage Split and K-Fold Cross-Validation methods is given in **Table 3**.

Table 3. Confusion matrix of CatBoost with percentage split and K-Fold vross-validation methods

Confusion Matrix of CatBoost Algorithms						
			Prediction			
	Class		0	1	2	3
Percentage Split	Actual	0	86	2	0	0
		1	4	77	10	1
		2	0	11	78	0
		3	0	0	1	78
			Prediction			
K-Fold Cross Validation	Class		0	1	2	3

Confusion Matrix of CatBoost Algorithms					
Actual	0	423	11	1	0
	1	19	351	65	0
	2	5	57	370	3
	3	0	0	5	430

From **Table 3**, it can be seen that there are 348 data points predicted using the percentage split testing technique. Data correctly predicted as label 0 amounted to 86 data. Data correctly predicted as label 1 amounted to 77 data. Data correctly predicted as label 2 amounted to 78 data. Data correctly predicted as label 3 amounted to 78 data. In the k-fold cross-validation testing technique, it is known that there are 1,740 data points tested. There are 423 data points correctly predicted as label 0. There are 351 data points correctly predicted as label 1. There are 370 data points correctly predicted as label 2. There are 430 data points correctly predicted as label 3.

4.3 Comparison of the two algorithms

The performance assessment of XGBoost and CatBoost in obesity classification demonstrates robust capabilities, with accuracy values consistently exceeding 90%. While the numerical difference in top accuracy between CatBoost 92.53% and XGBoost 91.67% using the Percentage Split method is narrow, 0.86%, and similarly for K-Fold Cross-Validation, 91.95% and 90.46%, a holistic view of the metrics confirms the stability of CatBoost.Addressing the minimal margins, it is crucial to observe the consistency of the results. CatBoost did not merely achieve a higher peak accuracy but consistently outperformed XGBoost across all evaluation metrics in both validation strategies (**Table 4**). Specifically, in the K-Fold Cross-Validation, which provides a more statistically robust estimate by averaging performance over 7 different data subsets, CatBoost maintained a lead of approximately 1.5%.

This consistency across multiple folds suggests that the performance advantage of CatBoost, albeit small in magnitude, is not a product of random data splitting but reflects a genuine improvement in handling the dataset's features. The use of Ordered Boosting in CatBoost likely contributes to this stability by reducing prediction shift, making it a slightly more reliable choice for this specific obesity classification task compared to XGBoost. Furthermore, the exceptionally high performance on minority labels can be attributed to the SMOTE process applied during training. By synthetically balancing the classes, the model was able to learn robust decision boundaries for underrepresented groups.

From a computational evaluation perspective, the comparative analysis is reinforced through the use of confusion matrix–based metrics, which offer a detailed representation of model behavior beyond overall accuracy. Metrics such as precision, recall, and F1-score enable class-level performance assessment by explicitly capturing the balance between false positives and false negatives, which is particularly important in multi-class obesity classification. Given that these performance indicators directly quantify predictive effectiveness and error characteristics at the class level, additional statistical significance testing is not required for model comparison (27) . By examining these metrics across both Percentage Split and K-Fold Cross-Validation strategies, the evaluation highlights not only predictive correctness but also the consistency, stability, and reliability of each model’s predictions. This comprehensive assessment framework

provides a practical and application-oriented comparison of CatBoost and XGBoost, emphasizing classification quality and robustness across validation settings.

Table 4 comprehensively presents the key performance metrics, including precision, recall, f1-score, support, and overall accuracy, for the CatBoost and XGBoost algorithms tested using two distinct data sampling techniques: the Percentage Split method and the K-Fold Cross-Validation method. Analysis of this data clearly indicates that the Percentage Split method yields superior results when compared to the K-Fold Cross-Validation approach. Specifically, the highest overall accuracy achieved using Percentage Split was 92.53% (with CatBoost), which is marginally better than the highest accuracy of 91.95% recorded under K-Fold Cross-Validation. Furthermore, across both testing methodologies, the CatBoost algorithm consistently demonstrated a higher final accuracy value than the XGBoost algorithm, reinforcing its better predictive capability for this dataset. The full comparative results of these two algorithms, illustrating their performance under both the Percentage Split and K-Fold Cross-Validation methods, are visually presented in detail in **Figure 1**.

Table 4. Precision, recall, f1-score, support, and accuracy of CatBoost algorithms and K-Fold Cross-Validation with percentage split and K-Fold Cross-Validation methods

Method	Algorithm	Label	Precision	Recall	F1-Score	Support	Accuracy (%)
Percentage Split	XG Boost	0	0.96	0.98	0.97	88	91.67
		1	0.86	0.84	0.85	92	
		2	0.88	0.88	0.88	89	
		3	0.99	0.99	0.99	79	
	CatBoost	0	0.96	0.98	0.97	88	92.53
		1	0.89	0.86	0.87	92	
		2	0.88	0.90	0.89	89	
		3	0.99	0.97	0.98	79	
K-Fold Cross Validation	XG Boost	0	0.95	0.97	0.96	435	90.46
		1	0.84	0.81	0.82	435	
		2	0.84	0.85	0.84	435	
		3	0.99	0.99	0.99	435	
	CatBoost	0	0.96	0.99	0.98	435	91.95
		1	0.87	0.84	0.86	435	
		2	0.86	0.86	0.86	435	
		3	0.98	0.99	0.98	435	

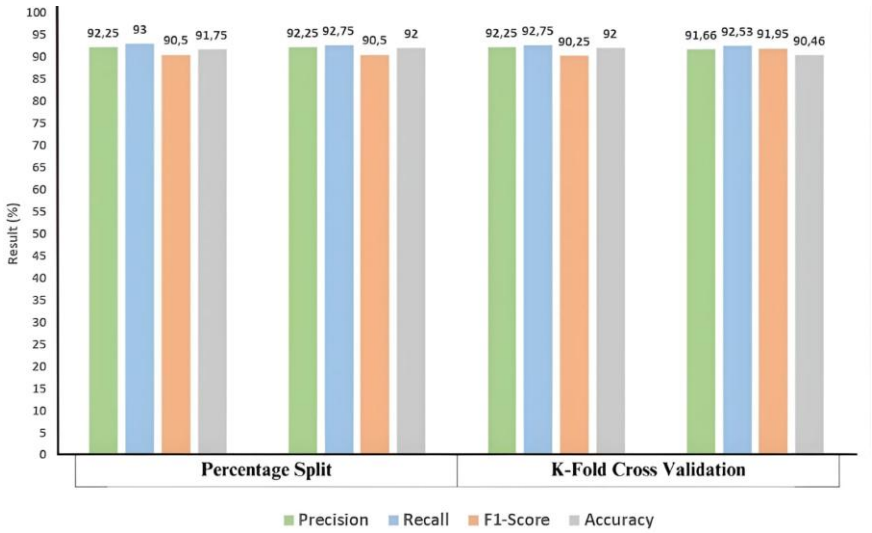

Fig. 1. Comparative graph of precision, recall, f-1 score, and accuracy values in XGBoost and CatBoost algorithms using percentage split and K-Fold Cross-Validation techniques

Based on the evaluation results in **Figure 1**, CatBoost and XGBoost show very effective classification performance with consistent accuracy above 90%. In the percentage split method, CatBoost achieved an accuracy of 92.53% compared to XGBoost's 91.67%, while in k-fold cross-validation, CatBoost obtained 91.95% and XGBoost 90.46%. It should be emphasized that these two algorithms are not expansions of each other; CatBoost uses Ordered Boosting to handle categorical data, while XGBoost relies on split-finding efficiency and λ regulation to control complexity. Validation through k-fold cross-validation ensures that these high results represent good model generalization capabilities and are not indicative of overfitting.

5 Conclusion

Based on the findings of this study, it is conclusively demonstrated that both the CatBoost and XGBoost algorithms exhibit robust and highly effective performance in the complex multi-class classification of obesity categories. The high level of discriminatory power achieved by these models is substantiated by the evaluation metrics, where precision, recall, F1-score, and accuracy consistently surpassed the 90% threshold across all configurations. This result affirms the suitability of advanced gradient boosting techniques for accurately modelling and predicting non-linear health status classifications such as obesity. A detailed analysis of the performance metrics reveals clear differentiation between the tested methodologies. Specifically, the XGBoost algorithm delivered a competitive accuracy of 91.67% when evaluated using the percentage split validation technique, and a marginally lower accuracy of 90.46% with the K-fold cross-validation approach. Conversely, the CatBoost algorithm consistently yielded superior classification outcomes; it achieved an accuracy of 92.53% using the percentage split technique and maintained a high performance of 91.95% under K-fold cross-validation. The minor difference between the two validation methods across both algorithms suggests good

generalization capability. Nonetheless, the percentage split method combined with the CatBoost algorithm is confirmed to provide the optimal and most decisive performance, achieving the highest recorded accuracy of 92.53% for obesity class prediction in this study.

Acknowledgment. The authors gratefully acknowledge the technical assistance and support provided by the staff of the Computational Laboratory, Faculty of Mathematics and Natural Sciences, Sriwijaya University, during the execution of this research.

References

[1] Septiyanti, S., Seniwati, S.: Obesity and Central Obesity in Indonesian Urban Communities. Jurnal Ilmiah Kesehatan (JIKA). vol. 2, pp. 118–27 (2020)

[2] Kinansi, R. R., Shaluhiyah, Z., Kartasurya, M. I., Sutiningsih, D., Adi, M. S., Widjajanti, W.: Pengetahuan, Sikap dan Perilaku tentang Obesitas pada Wanita Usia Produktif di Dukuh Gamol, Wilayah Kerja Kecamatan Mangunsari, Kota Salatiga. Jurnal Kesehatan Masyarakat. vol. 11, pp. 318–33 (2023)

[3] Arifani, S., Setiyaningrum, Z.: Faktor Perilaku Berisiko yang Berhubungan Dengan Kejadian Obesitas Pada Usia Dewasa di Provinsi Banten Tahun 2018. Jurnal Kesehatan. vol. 14, pp. 160–8 (2021)

[4] Baharuddin, F., Tjahyanto, A.: Peningkatan Performa Klasifikasi Machine Learning Melalui Perbandingan Metode Machine Learning dan Peningkatan Dataset. Jurnal Sisfokom (Sistem Informasi Dan Komputer). vol. 11, pp. 25–31 (2022)

[5] Fahmi Idris, J., Ramadhani, R., Malik Mutoffar, M.: Klasifikasi Penyakit Kanker Paru Menggunakan Perbandingan Algoritma Machine Learning. Jurnal Media Akademik (JMA)2. vol. 2 (2024)

[6] Kusuma, E. J., Nurmandhani, R., Aryani, L., Pantiawati, I., Shidik, G. F.: Optimasi Model Extreme Gradient Boosting Dalam Upaya Penentuan Tingkat Risiko Pada Ibu Hamil Berbasis Bayesian Optimization (BOXGB). Jurnal Teknologi Informasi Dan Ilmu Komputer. vol. 12, pp. 111–20 (2025)

[7] Siringoringo, R., Perangin Angin, R., Rumahorbo, B.: Model Klasifikasi Genetic-XGBoost Dengan T-Distributed Stochastic Neighbor Embedding Pada Peramalan Pasar. vol. 11 (2022)

[8] Emilia Sukmawati, C., Fitri Nur Masruriyah, A., Ratna Juwita, A., Damaiarta Tejayanda, R., Nurmayanti, T., *et al.*: Efektivitas algoritma AdaBoost dan XGBoost pada dataset obesitas populasi dewasa. Jambura Journal of Informatics. vol. 6, pp. 101–11 (2024)

[9] Salsabil, M., Azizah, N. L., Eviyanti, A.: Implementasi Data Mining Dalam Melakukan Prediksi Penyakit Diabetes Menggunakan Metode Random Forest Dan Xgboost. Jurnal Ilmiah Komputasi. vol. 23, pp. 51–8 (2024)

[10] Wicaksono, D. F., Basuki, R. S., Setiawan, D.: Peningkatan Performa Model Machine Learning XGBoost Classifier melalui Teknik Oversampling dalam Prediksi Penyakit Aids. Jurnal Media Informatika Budidarma. vol. 8, p. 736 (2024)

[11] Putri, E. R., Arianto, D. B.: Perbandingan Performa Algoritma Metode Bagging dan Boosting pada Prediksi Konsentrasi PM 10 di Jakarta Utara. Jurnal Nasional Teknologi Dan Sistem Informasi. vol. 10 (2024)

[12] Darmawan, A., Kartini, D., Hamonangan Saragih, T., Adi Nugraha, R.: Implementasi Catboost Menggunakan Hyper-Parameter Tuning Bayesian Search Untuk Memprediksi Penyakit Diabetes. Jurnal Komputasi. vol. 11, pp. 148–56 (2023)

[13] Irfannandhy, R., Handoko, L. B., Ariyanto, N.: Analisis Performa Model Random Forest dan CatBoost dengan Teknik SMOTE dalam Prediksi Risiko Diabetes. Edumatic: Jurnal Pendidikan Informatik. vol. 8, pp. 714–23 (2024)
[14] Haya, A. N., Ramme, M. Y.: Penerapan Algoritma Stacking Ensemble Machine Learning Berbasis Pohon untuk Prediksi Penyakit Diabetes. Seminar Nasional Sains Data. pp. 954–61 (2024)
[15] Arif Ali, Z., Abduljabbar, Z. H., Tahir, H. A., Bibo Sallow, A., Almufti, S. M.: Exploring the Power of EXtreme Gradient Boosting Algorithm with Machine Learning: a Review. Academic Journal of Nawroz University. vol. 12, pp. 320–34 (2023)
[16] Abdurrahman, G., Oktavianto, H., Sintawati, M.: Optimasi Algoritma XGBoost Classifier Menggunakan Hyperparameter Gridesearch dan Random Search Pada Klasifikasi Penyakit Diabetes. vol. 7 (2022)
[17] Hadianto, A., Utomo, W. H.: CatBoost Optimization Using Recursive Feature Elimination. Jurnal Online Informatika. vol. 9, pp. 169–78 (2024)
[18] Purbolingga, Y., Marta Putria, D., Rahmawatia, A., Wajhi Akramunnas, B.: Perbandingan Algoritma CatBoost dan XGBoost dalam Klasifikasi Penyakit Jantung. Jurnal Artikel Ilmiah Aplikasi Teknologi. vol. 15, pp. 126–33 (2023)
[19] Sabili, N. L., Fajri, R., Umbara, M.: Klasifikasi Penyakit Diabetes Menggunakan Algoritma Caterogical Boosting Dengan Faktor Risiki Diabetes. Jurnal Mahasiswa Teknik Informatika. vol. 8 (2024)
[20] Dwinanda, M. W., Satyahadewi, N., Andani, W.: Classification Of Student Graduation Status Using XGBoost Algorthm. Barekeng: Jurnal Ilmu Matematika Dan Terapan. vol. 17, pp. 1785–94 (2023)
[21] Joy, R. A.: An Interpretable Catboost Model to Predict the Power of Combined Cycle Power Plants. 2021 International Conference on Information Technology, ICIT 2021 - Proceedings. pp. 435–9 (2021)
[22] Batubara, G. M. C., Desiani, A., Amran, A.: Klasifikasi Jamur Beracun Menggunakan Algoritma Naïve Bayes dan K-Nearest Neighbors. Jurnal Ilmu Komputer Dan Informatika. vol. 3, pp. 33–42 (2023)
[23] Muzakir, A., Desiani, A., Amran, A.: Klasifikasi Penyakit Kanker Prostat Menggunakan Algoritma Naïve Bayes dan K-Nearest Neighbor. Komputika : Jurnal Sistem Komputer. vol. 12, pp. 73–9 (2023)
[24] Desiani, A.: Perbandingan Implementasi Algoritma Naïve Bayes dan K-Nearest Neighbor Pada Klasifikasi Penyakit Hati. SIMKOM. vol. 7, pp. 104–10 (2022)
[25] Iffah'da, A. N., Anita Desiani.: Implementasi Algoritma K-Nearest Neighbor (K-NN) dan Single Layer Perceptron (SLP) Dalam Prediksi Penyakit Sirosis Biliari Primer. Jurnal Ilmiah Informatika. vol. 7, pp. 65–74 (2022)
[26] Clara, S., Laksmi Prianto, D., Al Habsi, R., Friscila Lumbantobing, E., Chamidah, N., Kom, S.: Implementasi Seleksi Fitur Pada Algoritma Klasifikasi Machine Learning Untuk Prediksi Penghasilan Pada Adult Income Dataset. Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA) (2021)
[27] James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning with Applications in R. Springer. 2nd ed. (2021)