Indonesian Journal of Data and Science

Volume 5 Issue 3
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v5i3.199

Research Article

Classification of Mushroom Edibility Using K-Nearest Neighbors: A Machine Learning Approach

Fadhila Tangguh Admojo [1,*], Made Leo Radhitya [2], Hamada Zein [3], Ahmad Naswin [4]

[1] Universiti Kuala Lumpur, Malaysia, fadhila.tangguh@s.unikl.edu.my
[2] Institut Bisnis dan Teknologi Indonesia, Bali, Indonesia, leo.radhitya@instiki.ac.id
[3] Universitas Muhammadiyah Kalimantan Timur, Indonesia, hz831@umkt.ac.id
[4] Universitas Megarezky Makassar, Makassar, Indonesia, ahmadnaswin@gmail.com

Correspondence should be addressed to Fadhila Tangguh Admojo; fadhila.tangguh@s.unikl.edu.my

Received 05 October 2024; Accepted 10 December 2024; Published 31 December 2024

Abstract:

This study investigates the use of the K-Nearest Neighbors (KNN) algorithm for the binary classification of mushroom edibility using a cleaned version of the UCI Mushroom Dataset. The dataset underwent pre-processing techniques such as modal imputation, one-hot encoding, z-score normalization, and feature selection to ensure data quality. The model was trained on 80% of the dataset and evaluated on the remaining 20%, achieving an overall accuracy of 99%. Evaluation metrics, including precision, recall, and F1-score, confirmed the model's effectiveness in distinguishing between edible and poisonous mushrooms, with minimal misclassification errors. Despite its high performance, the study identified scalability as a limitation due to the computational complexity of KNN, suggesting that future research should explore alternative algorithms for enhanced efficiency. This research underscores the importance of pre-processing and hyperparameter optimization in building reliable classification models for food safety applications.

Keywords: Binary Classification, Data Pre-processing, Food Safety, K-Nearest Neighbors, Mushroom Classification
Dataset link: https://www.kaggle.com/datasets/prishasawhney/mushroom-dataset

1. Introduction

Mushrooms have been a vital component of global diets due to their nutritional value and unique culinary applications. However, identifying whether a mushroom is edible or poisonous remains a significant challenge, as visual inspection and traditional identification methods are often unreliable and prone to human error [1], [2]. Misidentification can lead to severe health consequences, including fatal poisoning. With the advent of machine learning, there is an opportunity to develop automated, data-driven approaches to mushroom classification, which can provide a safer and more accurate alternative to traditional methods [3], [4].

The UCI Mushroom Dataset, a widely used resource for such studies, offers a comprehensive set of features describing various mushroom characteristics. However, the dataset's raw form presents challenges, such as missing values and categorical attributes, which can affect the performance of machine learning models if not properly addressed. Furthermore, the overlap in physical features between edible and poisonous mushrooms highlights the need for robust pre-processing techniques and careful algorithm selection. Addressing these challenges is crucial to enhance the reliability of automated classification systems.

This study explores the application of the K-Nearest Neighbors (KNN) algorithm for binary classification of mushroom edibility. KNN, a simple yet effective machine learning technique, classifies data points based on their proximity to neighboring points in the feature space [5]–[7]. The primary objective of this research is to pre-process the dataset using techniques such as modal imputation, one-hot encoding, and z-score normalization, and subsequently

train and optimize a KNN model to accurately classify mushrooms as either edible or poisonous. By focusing on binary classification, this research aims to demonstrate the effectiveness of KNN in a controlled environment while highlighting the impact of pre-processing on model performance [8]–[10].

The scope of this study is limited to the cleaned version of the UCI Mushroom Dataset, consisting of nine features selected for their relevance to classification. While the KNN algorithm is well-suited for small- to medium-sized datasets, its computational complexity for larger datasets remains a limitation. Despite these constraints, this research contributes to the field by providing a replicable methodology for mushroom classification and showcasing the potential of machine learning in addressing food safety challenges.

The remainder of this paper is organized as follows. The next section details the methodology, including dataset pre-processing, normalization, splitting, and model training. The results section presents the performance of the KNN model along with visualizations of key findings. This is followed by a discussion that compares the results with previous research and explores their practical implications. Finally, the conclusion summarizes the study's contributions and offers recommendations for future research directions.

2. Method:

This section describes the detailed steps undertaken to pre-process the dataset, train the K-Nearest Neighbors (KNN) model, and evaluate its performance [11]. Each stage of the methodology was designed to ensure robust pre-processing and effective model training for the classification of mushroom edibility, show in **Figure 1**.

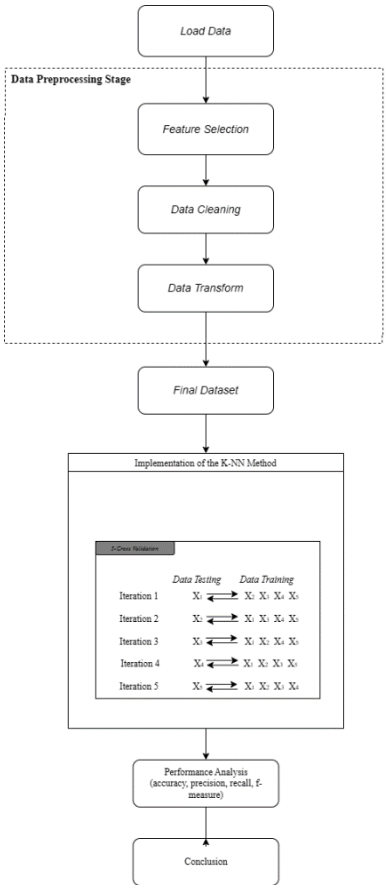

Figure 1. General Research Design Stages

Data Collection Process

The dataset used in this study is a cleaned version of the UCI Mushroom Dataset, specifically designed for binary classification. It contains nine features that describe various morphological characteristics of mushrooms, show in **Table 1** and **Figure 2**.

Table 1. Feature Descriptions

Feature Name	Data Type	Description	Example Values
Cap Diameter	Continuous	Diameter of the mushroom cap measured in centimeters.	3.5, 5.2, 7.1
Cap Shape	Categorical	Shape of the mushroom cap (e.g., bell, convex, flat).	Bell, Flat, Convex
Gill Attachment	Categorical	Describes how the gills are attached to the mushroom stem.	Free, Attached
Gill Color	Categorical	Color of the mushroom gills.	White, Brown, Pink
Stem Height	Continuous	Height of the mushroom stem measured in centimeters.	5.0, 7.3, 10.2
Stem Width	Continuous	Width of the mushroom stem measured in centimeters.	1.0, 1.5, 2.1
Stem Color	Categorical	Color of the mushroom stem.	White, Yellow, Brown
Season	Categorical	Season during which the mushroom is typically found.	Spring, Summer, Autumn
Target Class	Binary	Indicates whether the mushroom is edible (0) or poisonous (1).	0 (Edible), 1 (Poisonous)

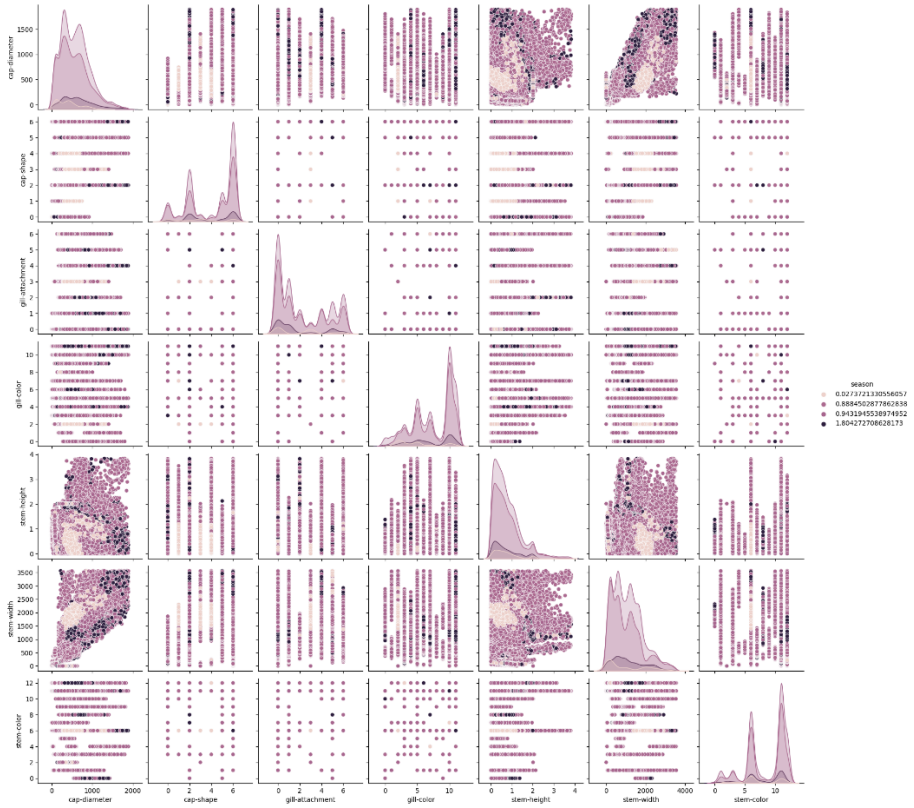

Figure 2. Scatter plots

Pre-processing Steps

Proper data pre-processing was critical to ensure the dataset was well-prepared for classification using KNN. The steps included:

1. Handling Missing Values: Missing categorical data was imputed using modal imputation, where missing values were replaced by the most frequently occurring value in the respective column.

2. One-Hot Encoding: Categorical variables, such as Gill Color and Cap Shape, were converted into numerical format using one-hot encoding. This transformation allowed the KNN algorithm to interpret categorical features during distance calculations.
3. Z-Score Normalization: Continuous features, such as Cap Diameter, Stem Height, and Stem Width, were normalized using the z-score formula [12]–[14]:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

Where:

x is the raw feature value

μ is the mean of the feature

σ is the standard deviation

This step ensured all features had equal weight in the distance computation used by KNN, **Figure 3** show distribution of dataset.

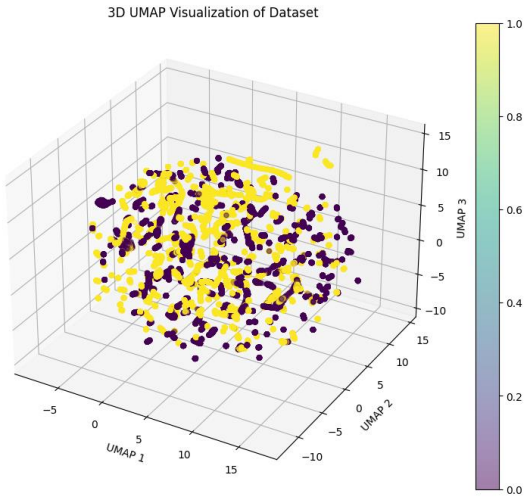

Figure 3. Visualization of Dataset

4. Feature Selection: Feature importance was analysed, and nine key features were retained for the model. This step reduced noise in the data and improved model interpretability, results of feature selection show in **Figure 4**.

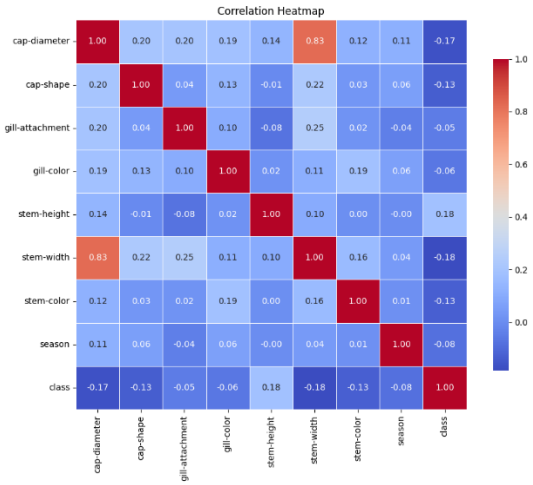

Figure 4. Correlation Heatmap

Data Analysis Methods

The pre-processed dataset was split into training and testing subsets:

- Training Set: 80% of the dataset for model training.
- Testing Set: 20% of the dataset for evaluating performance.

The split was stratified to maintain the proportion of classes (edible vs. poisonous) in both subsets. This approach ensured a balanced evaluation of the model, detailed show in **Figure 5**.

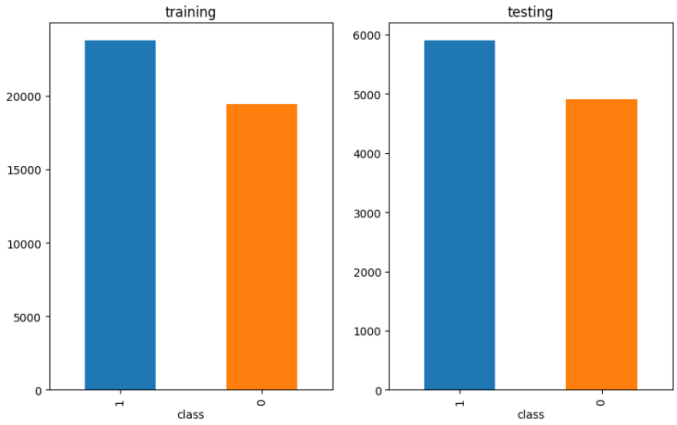

Figure 5. Split Data

Model Training

1. Algorithm Choice: The K-Nearest Neighbors (KNN) algorithm was selected for its simplicity and effectiveness in classification tasks. The algorithm predicts the class of a data point based on the majority class of its k-nearest neighbors in the feature space [6], [15]–[17].
2. Distance Metric: The Euclidean distance was used to measure the similarity between data points:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2}$$

   Where x_i and y_i represent feature value of two data points, and n is the number of features.
3. Optimizing k: The optimal value of k was determined by training the model with values of k ranging from 1 to 20. For each value, the model's accuracy on the validation set was evaluated, and the k yielding the highest accuracy was selected.
4. Cross-Validation: A 5-fold cross-validation strategy was employed to ensure the reliability of the model's performance metrics [18], [19].

$$\mathrm{CV}_{(K)} = \frac{1}{K}\sum_{i=1}^{K}\mathrm{Error}_i \tag{3}$$

Performance Evaluation

- **Accuracy:** The ratio of correctly predicted instances to the total instances [20]:

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

- **Precision:** The ratio of true positive predictions to the total positive predictions [21]:

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

- **Recall:** The ratio of true positive predictions to the total actual positives [22]:

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

- **F1-Score:** The harmonic mean of precision and recall [23]:

$$F - measure = \frac{2(presisi \times recall)}{(presisi + recall)} \quad (7)$$

Where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative. These metrics provided a comprehensive understanding of the model's performance, highlighting its strengths and areas of improvement.

3. Results and Discussion

The performance of the KNN algorithm in classifying mushrooms as edible or poisonous was highly effective, as demonstrated by the evaluation metrics. The model achieved an impressive accuracy of 99%, indicating that the majority of predictions were correct. Precision values for edible mushrooms (class 0) and poisonous mushrooms (class 1) were 98% and 99%, respectively, suggesting that the model made very few false positive errors. Similarly, the recall values were 98% and 99%, showing the model's ability to correctly identify both edible and poisonous mushrooms with high reliability. The F1-scores for both classes were nearly 99%, reflecting a strong balance between precision and recall across the dataset.

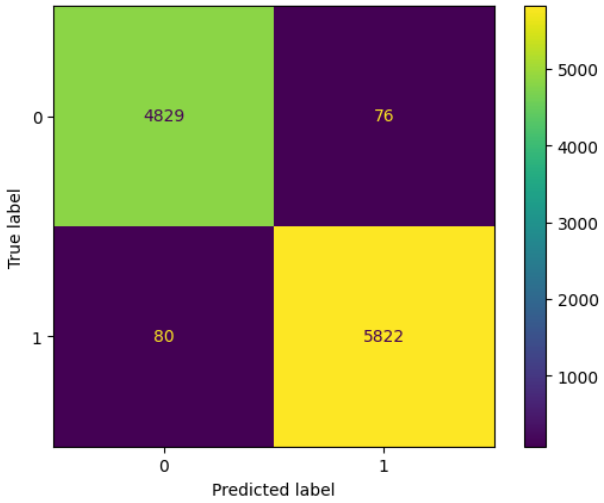

Figure 6. Confusion Matrix

Figure 6 provides a clear visualization of the model's performance. Out of 10,807 instances, the model correctly classified 4,829 edible mushrooms and 5,822 poisonous mushrooms. Only 76 edible mushrooms were misclassified as poisonous, and 80 poisonous mushrooms were misclassified as edible. These results highlight the robustness of the KNN model in distinguishing between the two classes with minimal errors.

The high performance of the KNN model can be attributed to several factors. The pre-processing steps, including one-hot encoding for categorical features and z-score normalization for continuous features, ensured that the input data was well-suited for distance-based calculations. Feature selection further contributed by focusing on nine relevant features, reducing noise and improving the model's interpretability. Moreover, optimizing the k-value allowed the model to strike a balance between underfitting and overfitting, which is crucial for achieving high accuracy.

Despite the promising results, the study faced some limitations. The computational cost of KNN increases with larger datasets, as the algorithm requires distance calculations for every training instance during prediction. Additionally, although the misclassification rate was low, certain edge cases with overlapping features, such as similarities in gill or cap characteristics, posed challenges for the model. These limitations suggest the need for exploring more advanced algorithms, such as Support Vector Machines (SVM) or ensemble methods like Random Forest, to further enhance classification accuracy and scalability.

4. Conclusion

This study explored the application of the K-Nearest Neighbors (KNN) algorithm for the binary classification of mushrooms as either edible or poisonous. By utilizing a cleaned and pre-processed version of the UCI Mushroom Dataset, the research emphasized the importance of pre-processing steps such as modal imputation, one-hot encoding, z-score normalization, and feature selection in enhancing the performance of machine learning models. The dataset was split into training and testing subsets, and the KNN model was trained with optimal hyperparameters to achieve maximum accuracy. The findings revealed that the KNN model performed exceptionally well, achieving an overall accuracy of 99%, with high precision, recall, and F1-scores across both classes. The confusion matrix confirmed that the model effectively distinguished between edible and poisonous mushrooms, with only a small number of misclassifications. These results highlight the algorithm's robustness and reliability in handling this classification task, making it a viable solution for real-world applications in food safety and mushroom identification.

However, the study also identified certain limitations. The computational cost of KNN increases with larger datasets due to the reliance on distance-based calculations, and certain edge cases with overlapping features led to minor misclassifications. These challenges suggest opportunities for further research to explore advanced classification algorithms, such as Support Vector Machines (SVM) or ensemble approaches like Random Forest, which could address scalability and improve classification accuracy. Additionally, incorporating domain knowledge into feature engineering may further refine the model's performance. In conclusion, this research demonstrates that the KNN algorithm, when combined with robust pre-processing techniques, is highly effective for the binary classification of mushroom edibility. The methodology and findings provide a strong foundation for future studies aiming to enhance food safety and automate mushroom classification processes. Future research should focus on optimizing scalability and exploring alternative algorithms to build even more reliable and efficient classification models.

References:

[1] S. K. Pal, “Mushroom Classification Model to Check Edibility using Machine Learning,” *Proceedings of the 17th INDIACom; 2023 10th International Conference on Computing for Sustainable Global Development, INDIACom 2023*. pp. 214–217, 2023, doi: 10112407.

[2] S. Verma, “A Comprehensive Study on the Classification of the Edibility of Mushrooms,” *Proceedings of the 2023 12th International Conference on System Modeling and Advancement in Research Trends, SMART 2023*. pp. 7–13, 2023, doi: 10.1109/SMART59791.2023.10428619.

[3] M. S. Morshed, “Predicting Mushroom Edibility with Effective Classification and Efficient Feature Selection Techniques,” *International Conference on Robotics, Electrical and Signal Processing Techniques*, vol. 2023. pp. 1–5, 2023, doi: 10.1109/ICREST57604.2023.10070049.

[4] M. S. Devi, “Dimensionality Reduction Based Component Discriminant Factor Implication for Mushroom Edibility Classification Using Machine Learning,” *Lecture Notes in Networks and Systems*, vol. 311. pp. 1–15, 2022, doi: 10.1007/978-981-16-5529-6_1.

[5] C. Budak, “Classification of the Ionospheric Disturbances Caused by Geomagnetic and Seismic Activity with K-Nearest Neighbors Algorithm,” *Wirel. Pers. Commun.*, vol. 134, no. 3, pp. 1551–1569, 2024, doi: 10.1007/s11277-024-10965-z.

[6] R. A. Dharmmesta, “Classification of Foot Kicks in Taekwondo Using SVM (Support Vector Machine) and KNN (K-Nearest Neighbors) Algorithms,” *Proceedings of the 2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 2022*. pp. 36–41, 2022, doi: 10.1109/IAICT55358.2022.9887475.

[7] M. Mentari, “Classification of Siam Orange Ripeness Level using K-Nearest Neighbors Algorithm and Features Gray Level Run Length Matrix,” *Proceeding - COMNETSAT 2023: IEEE International Conference on Communication, Networks and Satellite*. pp. 272–277, 2023, doi: 10.1109/COMNETSAT59769.2023.10420620.

[8] H. Oumarou and N. Rismayanti, "Automated Classification of Empon Plants: A Comparative Study Using Hu Moments and K-NN Algorithm," *Indones. J. Data ...*, 2023, doi: 10.56705/ijodas.v4i3.115.

[9] D. Ratnasari, "Comparison of Performance of Four Distance Metric Algorithms in K-Nearest Neighbor Method on Diabetes Patient Data," *Indones. J. Data Sci.*, 2023, doi: 10.56705/ijodas.v4i2.71.

[10] I. G. I. Sudipa, R. A. Azdy, I. Arfiani, and ..., "Leveraging K-Nearest Neighbors for Enhanced Fruit Classification and Quality Assessment," *Indones. J. ...*, 2024, doi: 10.56705/ijodas.v5i1.125.

[11] H. Azis, F. Fattah, and P. Putri, "Performa Klasifikasi K-NN dan Cross-validation pada Data Pasien Pengidap Penyakit Jantung," *Ilk. J. Ilm.*, vol. 12, no. 2, pp. 81–86, 2020, doi: 10.33096/ilkom.v12i2.507.81-86.

[12] M. Sholeh, "Comparison of Z-score, min-max, and no normalization methods using support vector machine algorithm to predict student's timely graduation," *AIP Conference Proceedings*, vol. 3077, no. 1. 2024, doi: 10.1063/5.0202505.

[13] L. Peng, "Dual-Structure Elements Morphological Filtering and Local Z-Score Normalization for Infrared Small Target Detection against Heavy Clouds," *Remote Sens.*, vol. 16, no. 13, 2024, doi: 10.3390/rs16132343.

[14] D. Geem, "Progression of Pediatric Crohn's Disease Is Associated With Anti–Tumor Necrosis Factor Timing and Body Mass Index Z-Score Normalization," *Clin. Gastroenterol. Hepatol.*, vol. 22, no. 2, pp. 368–376, 2024, doi: 10.1016/j.cgh.2023.08.042.

[15] R. Thakur, "Classification Performance of Land Use from Multispectral Remote Sensing Images using Decision Tree, K-Nearest Neighbor, Random Forest and Support Vector Machine Using EuroSAT Data," *Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 1, pp. 67–77, 2022.

[16] P. Suksomboon, "Performance Comparison Classification using k-Nearest Neighbors and Random Forest Classification Techniques," *2022 3rd International Conference on Big Data Analytics and Practices, IBDAP 2022*. pp. 43–46, 2022, doi: 10.1109/IBDAP55587.2022.9907218.

[17] I. Budiman, "Classification of Bird Species using K-Nearest Neighbor Algorithm," *2022 10th International Conference on Cyber and IT Service Management, CITSM 2022*. 2022, doi: 10.1109/CITSM56380.2022.9936012.

[18] E. Najwaini, T. E. Tarigan, and F. P. Putra, "Application of the K-Nearest Neighbors (KNN) Algorithm on the Brain Tumor Dataset," *... Artif. Intell. ...*, 2023, doi: 10.56705/ijaimi.v1i1.85.

[19] A. Aisyah and S. Anraeni, "Analisis penerapan metode K-Nearest Neighbor (K-NN) pada dataset citra penyakit malaria," *Indones. J. Data Sci.*, 2022, doi: 10.56705/ijodas.v3i1.22.

[20] F. T. Admojo and N. Rismayanti, "Estimating Obesity Levels Using Decision Trees and K-Fold Cross-Validation: A Study on Eating Habits and Physical Conditions," *Indones. J. Data ...*, 2024, doi: 10.56705/ijodas.v5i1.126.

[21] I. Alwiah, U. Zaky, and A. W. Murdiyanto, "Assessing the Predictive Power of Logistic Regression on Liver Disease Prevalence in the Indian Context," *... J. Data Sci.*, 2024, doi: 10.56705/ijodas.v5i1.121.

[22] A. P. Wibowo, M. Taruk, T. E. Tarigan, and ..., "Improving Mental Health Diagnostics through Advanced Algorithmic Models: A Case Study of Bipolar and Depressive Disorders," *Indones. J. ...*, 2024, doi: 10.56705/ijodas.v5i1.122.

[23] I. P. A. Pratama, E. S. J. Atmadji, and ..., "Evaluating the Performance of Voting Classifier in Multiclass Classification of Dry Bean Varieties," *Indones. J. ...*, 2024, doi: 10.56705/ijodas.v5i1.124.