*J-INTECH (Journal of Information and Technology)*
**Accredited Sinta 4 Ministry of Higher Education, Science and Technology**
**Republic of Indonesia SK No. 10/C/C3/DT.05.00/2025**
**E-ISSN: 2580-720X || P-ISSN: 2303-1425**

# Analysis of the Effectiveness of Traditional and Ensemble Machine Learning Models for Mushroom Classification

Neny Sulistianingsih[1*], Galih Hendo Martono[2]

*[1,2]Departement of Computer Science, Master Program, Universitas Bumigora, Ismail Marzuki St. Mataram, Indonesia*

***Keywords***
*Bagging; Ensemble Learning; K-Nearest Neighbors; Mushroom Classification; Random Forest; Stacking; Voting Classifier*

****Corresponding Author:***
*neny.sulistianingsih@universitasbumigora.ac.id*

***Abstract***
The classification of edible versus poisonous mushrooms presents a critical challenge in the domains of applied biology and public health, particularly due to the serious implications of misidentification. This research employs the UCI Mushroom Dataset to evaluate and compare the effectiveness of several machine learning models, including traditional algorithms like Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors and Naïve Bayes, as well as advanced ensemble techniques such as Stacking and Voting Classifier. Notably, both Random Forest and Stacking achieved flawless accuracy, reaching 100%, underscoring the high predictive capacity of these models in complex categorical scenarios. Conversely, Naïve Bayes exhibited significantly weaker performance—achieving only 59.8% accuracy—likely due to its underlying assumption of feature independence, which does not hold for this dataset. The ensemble learning approaches, including the combination of Stacking and Bagging, not only preserved but also enhanced model robustness and generalization. These methods effectively leverage the complementary strengths of individual learners to yield more accurate and stable predictions while mitigating overfitting risks. Comparative analysis with previous research confirms the consistency of these findings and reinforces the viability of ensemble strategies for handling intricate classification tasks. Overall, this study highlights the importance of algorithm selection tailored to data characteristics and supports the use of ensemble learning to boost predictive reliability.

## 1. Introduction

Mushrooms have long been an integral part of the dietary habits of the Indonesian people, serving both as a source of nutrition and as a key ingredient in a variety of traditional cuisines. With protein content ranging from 8.5 to 36.9 grams per 100 grams, mushrooms offer nutritional value comparable to meat and eggs and far exceed the protein content found in vegetables and grains [1]. Additionally, mushrooms are a rich source of B vitamins and have higher protein levels than most other vegetables. Beyond their high nutritional value, mushrooms also provide essential health benefits, such as reducing cancer risk, supporting weight loss programs, and strengthening the immune system [2]. However, out of approximately 14,000 identified mushroom species, only a small portion are safe for consumption, while around 3,000 species are known to be toxic [3], [4]. The presence of toxins in certain types of wild mushrooms poses a serious health threat, as they can cause severe poisoning and even death. Between 2010 and 2020, there were 76 reported cases of mushroom poisoning in Indonesia, involving more than 550 victims and resulting in at least nine deaths, most of which were attributed to genera such as Amanita, Chlorophyll, and Galerina [5], [6]. These incidents have occurred not only in rural areas but have also impacted urban populations. For example, in December 2024, 17 residents of Kampung Kebon Kalapa, Sukabumi, suffered poisoning after consuming wild mushrooms, with one individual in critical condition. Similarly, in March 2023, six residents in Sikka Regency, East Nusa Tenggara (NTT), were hospitalized after showing symptoms of poisoning due to wild mushroom consumption [7], [8].

Mushroom poisoning is not merely a public health issue but also intersects with education, food safety, and risk mitigation in consuming natural products. Amidst the growing trend of organic lifestyles and the use of non-timber forest products, limited knowledge about toxic mushroom species presents a new and emerging threat. Therefore, research that offers preventive solutions for identifying poisonous mushrooms is highly significant, particularly in tropical countries like Indonesia, which are rich in fungal biodiversity.

So far, most mushroom identification efforts have remained manual, relying on personal experience or visual recognition. Scientific approaches using machine learning technology are still rarely applied in the local context, particularly regarding datasets relevant to conditions in Indonesia.

Research on mushroom classification and edibility detection has been extensively developed using machine learning and ensemble learning approaches. [9] explored ensemble learning methods such as Bagging with Naïve Bayes, Boosting with AdaBoost, and Random Forest based on the CART method, where Bagging and Random Forest demonstrated near-perfect accuracy of 99.93%. In contrast, the study by [10] focused more on the morphological characteristics of mushrooms, such as cap color, shape, and habitat, to compare the performance of Naïve Bayes and K-Nearest Neighbors (KNN), finding that KNN (k=1) achieved perfect accuracy of 100%, outperforming Naïve Bayes, which reached only 90.2%. Meanwhile, [11] also used KNN to classify mushrooms as edible or poisonous, achieving a high model accuracy of 99% and a minimal error rate. On the other hand, [12], [13] extended the classification study into the domain of mushroom diseases by identifying six types of diseases using CNN for feature extraction and Random Forest for classification. The study reported precision, recall, and F1 scores in the 95%–97% range. [14] employed an ensemble learning approach using the UCI mushroom dataset and found that the Extra Trees method performed best, with 99.17% accuracy and a near-perfect ROC AUC score of 99.94%, while AdaBoost showed less optimal performance. Additionally, [15] applied Random Forest and Multiclass Support Vector Machine (M-SVM) for mushroom disease detection. The results showed that Random Forest produced higher accuracy (82%) than M-SVM (76%). Furthermore, machine learning has also been applied to monitor mushroom growth, as demonstrated by [16] using Yolov5 and Detectron2, which achieved an accuracy of 70%. In addition, [17], [18] employed CNN and YOLO for real-time detection of poisonous and edible mushrooms. These findings show that ensemble learning methods and traditional classification techniques can significantly enhance the accuracy of detecting mushroom edibility and diseases.

Previous research has demonstrated the effectiveness of various classification algorithms in distinguishing between poisonous and edible mushrooms, particularly using publicly available datasets such as the UCI Mushroom Dataset. However, many of these studies remain limited to generalized scenarios and have not been tailored for practical application in specific local contexts, especially in regions with unique fungal biodiversity. Additionally, insufficient attention has been given to the interpretability of machine learning models, which poses challenges when communicating the reasoning behind classification results to non-technical stakeholders, such as local harvesters or public health practitioners. Addressing these gaps, this study aims not only to evaluate and compare the performance of traditional and ensemble classification algorithms but also to explore approaches that enhance the transparency, contextual relevance, and practical usability of predictive models in real-world mushroom identification and poisoning prevention.

## 2. Research Method

### 2.1 Research Design

This study adopts a quantitative approach with an experimental design based on supervised machine learning. Its primary focus is to develop a classification model that distinguishes poisonous and edible mushrooms using various traditional machine-learning methods and ensemble learning techniques. In general, the research consists of five main stages. First, data collection and understanding use a public dataset containing various mushroom characteristics. Second, data preprocessing is performed, including handling missing values with mean imputation, normalizing numerical attributes using MinMax Scaler, and encoding categorical attributes using One-Hot Encoding. Third, exploratory data analysis (EDA) is conducted to identify key data patterns and understand relationships between attributes and the target variable. The fourth stage involves model development, where traditional methods such as Logistic Regression, Decision Tree, Support Vector Machine, KNN, and Naïve Bayes are applied and combined using ensemble learning techniques such as Voting, Stacking, a combination of Bagging and Stacking with Logistic Regression as the final estimator, as well as other ensemble methods including Random Forest, Gradient Boosting, AdaBoost, and XGBoost. Finally, model evaluation uses five key metrics, accuracy, precision, recall, F1-score, and AUC, to assess model performance and generalization capability. Through this approach, the study aims to produce an accurate classification model. It also seeks to explore the effectiveness of ensemble learning strategies in enhancing the prediction performance of mushroom classification.

### 2.2 Dataset

This study utilizes the Mushroom Classification Dataset, which was obtained from the UCI Machine Learning Repository and accessible via Kaggle. The dataset provides information on various types of mushrooms, primarily focusing on determining whether a given mushroom species is edible or poisonous. It contains 8,124 samples with 22 attributes and one target attribute. Each mushroom is described based on several physical characteristics, such as cap shape, cap surface texture, cap color, presence of bruises, odor, gill attachment to the stalk, gill size and color, stalk shape, stalk root type, and the habitat in which the mushroom grows. The target label in this dataset is encoded as 'e' for edible mushrooms and 'p' for poisonous ones. An example from the dataset is presented in Table 1.

*Table 1. Example of the Dataset Used*

| Class | Cap Diameter | Cap Shape | Cap Surface | Cap Color | Does Bruise or Bleed | Gill Attachment | Gill Spacing | Gill Color | Stem Height | ... | Stem Root | Stem Surface |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| p | 15.26 | x | g | o | f | e | NaN | w | 16.95 | ... | s | y |
| p | 16.60 | x | g | o | f | e | NaN | w | 17.99 | ... | s | y |
| p | 14.07 | x | g | o | f | e | NaN | w | 17.80 | ... | s | y |
| p | 14.17 | f | h | e | f | e | NaN | w | 15.77 | ... | s | y |
| p | 14.64 | x | h | o | f | e | NaN | w | 16.53 | ... | s | y |

### 2.3 Data Preprocessing Steps

The preprocessing process is carried out to ensure the data is in optimal condition before entering the modeling phase, including:

- Handling Missing Values
  Nilai kosong pada dataset diidentifikasi dan diimputasi menggunakan nilai rata-rata (*mean*) untuk atribut numerik.
- Normalization
  Numerical attributes are normalized using MinMax Scaler to adjust the data scale to a range of 0 to 1.
- Categorical Data Encoding
  Categorical attributes are converted into numerical format through One-Hot Encoding.
- *Exploratory Data Analysis* (EDA)

Exploratory analysis is conducted to understand data distribution, correlations between attributes, and detect potential anomalies or outliers.

## 2.4 Modeling Stage

In this study, the modeling process is divided into two main phases: applying traditional models and applying ensemble learning-based models to optimize classification performance. In the first phase, various conventional machine-learning methods are applied. Some of the techniques used include Logistic Regression, which serves as a baseline for binary classification based on a linear model; Decision Tree Classifier, which allows decision visualization through an interpretable and straightforward tree structure; SVM, which is effective for determining the best-separating hyperplane between two classes; KNN, an instance-based method that leverages proximity between data points; and Naïve Bayes, a simple yet effective probabilistic-based method. The parameters used in each traditional method can be seen in Table 2.

*Table 2. Parameter of Traditional Method*

| Model | Main Parameters |
|---|---|
| Logistic Regression | max_iter=1000 |
| Decision Tree | default |
| Random Forest | default |
| Support Vector Machine | probability=True |
| K-Nearest Neighbors | default |
| Gradient Boosting | default |
| Naive Bayes | default |
| XGBoost | use_label_encoder=False, eval_metric='mlogloss' |
| Voting Classifier | voting='soft', |
| Stacking + Bagging | estimators= all method, final_estimator=LogisticRegression() |
| Bagging + Stacking | estimator=LogisticRegression(max_iter=1000), n_estimators=10 |
| Boosting (AdaBoost) | estimator=DecisionTreeClassifier(), n_estimators=50 |
| Model | Main Parameter used |
| Logistic Regression | max_iter=1000 |
| Decision Tree | default |

In the second phase, ensemble learning models are applied to improve the robustness and accuracy of the model against new data. First, the Voting Classifier is used, where the predictions from multiple models are combined based on the majority voting principle to determine the final class. Next, the Stacking Classifier is implemented, which is a method that combines predictions from various base learners and retrains them using Logistic Regression as a meta-learner to enhance generalization ability. Furthermore, this study also combines Bagging and Stacking techniques, where Bagging aims to reduce variance by building models from different data subsets. At the same time, Stacking strengthens accuracy through a meta-model approach, still using Logistic Regression as the final estimator. Additionally, several ensemble learning methods are applied, such as the Random Forest Classifier, a technique capable of reducing overfitting risk by combining multiple decision trees. Moreover, Gradient Boosting is used, an iterative technique that gradually corrects prediction errors; XGBoost, a boosting variant known for its speed and accuracy; and AdaBoost, a boosting method that combines several weak learners to form a strong learner. The ensemble method pipeline applied can be seen in Table 3.

In this study, model implementation was carried out using two primary libraries: Scikit-Learn and XGBoost. Scikit-Learn was utilized for all models except for the XGBoost model, which relied on its dedicated library. Scikit-Learn was chosen for its comprehensive collection of machine learning algorithms that are user-friendly and support efficient data preprocessing and evaluation. Meanwhile, XGBoost was employed separately to leverage its powerful boosting technique, which significantly enhances prediction accuracy, especially when dealing with complex and large datasets. By combining these two libraries, the research was able to optimize both the performance and efficiency of the model development process

*Table 3. Pipeline of Ensemble Method*

| Ensemble Method | Base Estimators | Final Estimator | Notes | Ensemble Method |
|---|---|---|---|---|
| Voting Classifier | Logistic, Tree, RF, SVM, KNN, GB, NB, XGBoost | Majority / Probabilistic | `voting='soft'` | Voting Classifier |
| Stacking + Bagging | Logistic, Tree, RF, SVM, KNN, GB, NB, XGBoost | LogisticRegression | Output base model → meta-model | Stacking + Bagging |
| Bagging + Stacking | LogisticRegression (as weak learner) | Aggregated | Ensemble of same model (bagging) | Bagging + Stacking |
| Boosting | DecisionTreeClassifier | Adaptive weights | AdaBoost strategy | Boosting |

## 2.5 Model Evaluation

In order to evaluate the performance of the developed model, this research employs five complementary evaluation metrics, offering a holistic understanding of the model's ability to distinguish between edible and poisonous mushrooms. Among these, accuracy serves as a fundamental metric, indicating the proportion of correct predictions relative to the total number of test instances. It provides an initial insight into how closely the model's classification outcomes match the actual labels. Next, precision is considered, which is the ratio of true positive predictions to the total number of positive predictions made by the model. This metric is critical in this study, as it shows how accurately the model can identify poisonous mushrooms without making many misclassifications of mushrooms that are safe to eat.

Next, recall is also evaluated, which reflects the model's ability to capture all cases of poisonous mushrooms in the dataset. Recall becomes crucial because, in mushroom detection, the failure to identify poisonous mushrooms can have fatal consequences. To balance precision and recall, the F1-Score is used, which is the harmonic mean of both metrics. The F1-Score provides a fair assessment, especially when the data is imbalanced or it is essential to optimize both aspects simultaneously.

Finally, an analysis is conducted on the AUC (Area Under the Curve). AUC measures the area under the ROC curve, and the closer the value is to 1, the better the model distinguishes between poisonous and edible mushrooms. Considering these five metrics, the model evaluation does not focus solely on one performance aspect. Still, it provides a more comprehensive assessment of the model's accuracy, sensitivity, balance, and discriminatory ability in classifying the data. Additionally, cross-validation was conducted after the initial evaluation phase to examine whether the model's exhibited overfitting with respect to the dataset. This step helps ensure the robustness and generalizability of the models applied.

# 3. Result

## 3.1 *Preprocessing*

Before building the classification model, a preprocessing phase is conducted to ensure the quality and readiness of the data to be used. This phase addresses common issues in raw data, such as missing values, inconsistent attribute scales, and categorical data types that machine learning methods cannot directly process.

First, missing values are handled. The analysis revealed that several attributes contain missing values, such as class, cap diameter, cap shape, cap color, does-bruise-or-bleed, gill color, stem height, stem width, stem color, ring, habitat, and season. On the other hand, other columns such as cap-surface, gill-attachment, gill-spacing, stem-root, stem-surface, veil-type, veil-color, ring-type, and spore-print-color have values. Attributes with missing values were imputed using the mean value of the corresponding attribute. This approach is used to maintain the balance of the data distribution without introducing significant bias.

Next, a normalization technique using Min-Max Scaler is applied for the numerical attributes. This process transforms the range of numerical values to a scale of [0,1], thereby speeding up the model training process and preventing attributes with larger values from dominating the classification results.

One Hot Encoding technique is applied for categorical attributes, which converts categorical data into binary form so that machine learning methods can recognize and process it optimally. Table 4 provides an example of the One Hot Encoding results.

*Table 4. One Hot Encoding Results*

| | cap-diameter | stem-height | stem-width | cap-shape_b | cap-shape_c | cap-shape_f | cap-shape_o | cap-shape_p | cap-shape_s | cap-shape_x |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.240 | 0.499 | 0,164 | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE |
| 1 | 0.261 | 0.530 | 0,175 | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE |
| 2 | 0.220 | 0.524 | 0,170 | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE |
| 3 | 0.222 | 0.464 | 0,153 | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE |
| 4 | 0.230 | 0.487 | 0,165 | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE |

After the data cleaning and transformation stage is completed, initial data exploration is also conducted to understand the class distribution relationships between attributes and to detect essential patterns within the dataset. Examples of the attribute distribution used, such as the cap-shape and cap-surface attributes, can be seen in Figure 1.

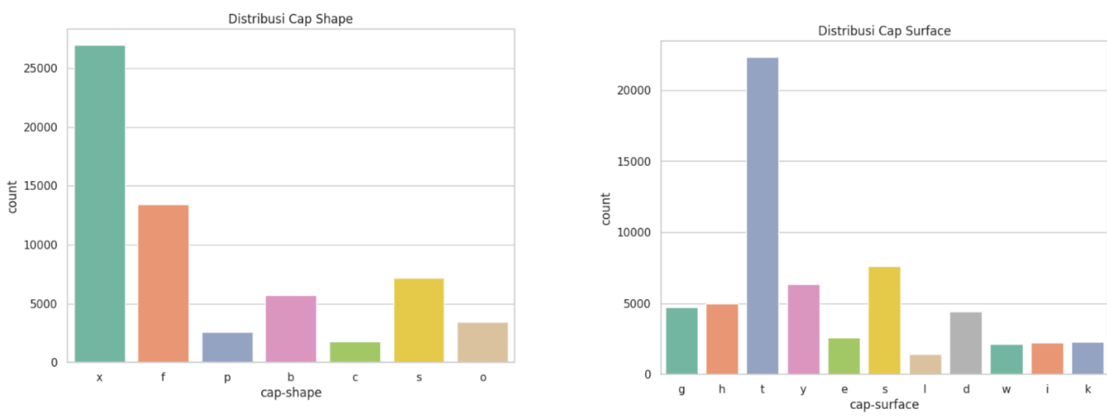


*Figure 1. Attribute Distribution*

Figure 1 presents a visual representation of the frequency distribution for various categories under the **cap-shape** and **cap-surface** features in the mushroom dataset. The x-axis displays categorical codes that correspond to different cap shapes, such as "g" for grooves, "h" for fibrous, among others, while the y-axis indicates the number of occurrences for each category within the dataset. The chart reveals that the "x" category (convex cap shape) is the most prevalent, appearing in over 27,000 samples. The "f" category (flat) also appears frequently, with approximately 13,500 entries, whereas the "c" (conical) shape is the least represented. For the **cap-surface** feature, the "t" category (scaly texture) dominates the distribution, exceeding 22,000 occurrences, suggesting that this texture is the most commonly observed. Other textures such as "s" (silky), "y" (shiny), and "e" (smooth) show moderate frequencies, ranging from 5,000 to 8,000 samples. This distribution highlights the diversity of cap textures and shapes, offering valuable insights that can support more precise mushroom classification or species identification. Figure 2 shows the EDA visualization related to the correlations between attributes in the dataset.


*Figure 2. Correlation between attributes in the dataset used*

Figure 2 displays a heatmap of the Pearson correlation matrix, illustrating the linear relationships among three numerical features in the mushroom dataset: cap diameter, stem height, and stem width. The colors on the heatmap represent the strength of these correlations, with red indicating a strong correlation and blue indicating a weaker one. The results reveal a relatively strong positive correlation of 0.70 between cap diameter and stem width, suggesting that mushrooms with larger caps tend to have thicker stems. Meanwhile, the correlation between cap diameter and stem height is 0.42, and between stem height and stem width is 0.44, both indicating moderate positive relationships. The diagonal values display perfect correlation (value of 1) between each feature and itself. These findings highlight the interdependence among features within the dataset, which is crucial to consider—especially when applying algorithms that assume feature independence, such as Naïve Bayes. If not properly addressed, these correlations could impact model accuracy.

### 3.2 Evaluation of Modeling Results

At this stage, the study presents a comprehensive explanation of the results obtained from various classification experiments. Each machine learning algorithm is assessed using five essential evaluation metrics—accuracy, precision, recall, F1-score, and AUC—to provide a complete picture of the model's ability to distinguish between poisonous and edible mushrooms. In addition to these quantitative metrics, confusion matrices are included to offer a clearer breakdown of classification outcomes, specifically in terms of true positives, false positives, true negatives, and false negatives. This study specifically presents the confusion matrices for the Random Forest and Naive Bayes models as illustrative examples of two algorithms with distinctly different performance levels. The confusion matrices can be found in Figure 3. Including these visualizations highlights the contrasting strengths and weaknesses of each model and serves as a foundational reference for deeper discussion regarding model selection and its practical implications in real-world biological classification scenarios.

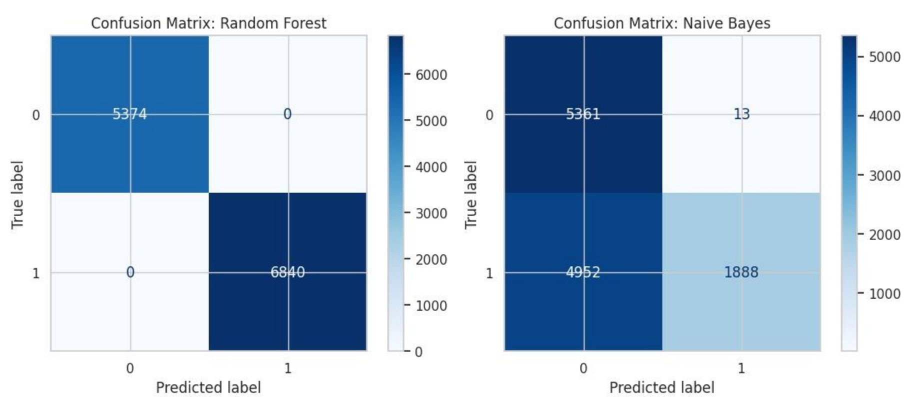


*Figure 3. Example of Confusion Matrix*

The results of applying traditional modeling methods can be seen in Table 5.

*Table 5. Classification Results with Traditional Methods*

| Model | Accuracy | Precision | Recall | F1 Score | AUC |
|---|---|---|---|---|---|
| Logistic Regression | 0.841 | 0.842 | 0.841 | 0.842 | 0.913 |
| Decision Tree | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 |
| Support Vector Machine | 0.995 | 0.995 | 0.995 | 0.995 | 0.999 |
| K-Nearest Neighbors | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| Naive Bayes | 0.598 | 0.787 | 0.598 | 0.550 | 0.835 |

Based on the performance evaluation results in Table 3, from the five machine learning algorithms applied—Logistic Regression, Decision Tree, SVM, KNN, and Naïve Bayes—there is a significant variation in the accuracy and precision of the models. Logistic Regression achieved an accuracy of 84.18% with fairly balanced precision and recall, around 84.24% and 84.18%, respectively, and an F1-score of 84.20%. Although its AUC value reached 0.91361, its performance still lagged behind other models. On the other hand, the Decision Tree showed nearly perfect results, with all metrics—accuracy, precision, sensitivity, and F1-score—at 99.87% and an almost perfect AUC value of 0.998. The SVM model also performed competitively, with an accuracy of 99.53% and other metrics at similar levels, plus an AUC of 0.999, indicating its excellent ability to classify the two classes. KNN even showed the best performance, with all evaluation values reaching 100%, reflecting the data's suitability for the distance-based approach.

In contrast, Naïve Bayes exhibited the lowest performance among all models, with an accuracy of only 59.87% and an F1-score of 55.01%, although its precision was 78.70%. Its AUC value of only 0.835 further demonstrates the limitations of this model in understanding the complex relationships between features. Figure 4 shows the training and validation accuracy of the traditional method.

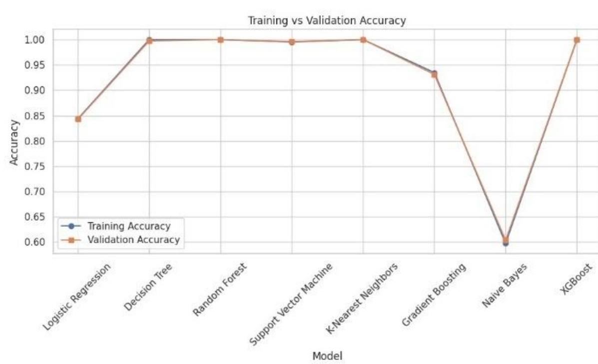


*Figure 4. Training and Validation Accuracy of Traditional Method*

Fig 4. illustrates a comparison of several classification algorithms based on their performance across training and validation datasets. Models such as Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and XGBoost exhibit outstanding results, achieving near-perfect accuracy on both training and validation sets. This suggests that these models effectively learn from the data without significant overfitting. While Logistic Regression performs slightly lower with around 84% accuracy, its consistent results across both datasets indicate reliable and stable performance. A minor drop in validation accuracy is noticeable in the Gradient Boosting model, pointing to mild overfitting that still remains within acceptable bounds. In contrast, the Naïve Bayes classifier shows a significant drop in performance, hovering around 60% accuracy, which may stem from its strong assumption of feature independence—an assumption that doesn't align well with the complex relationships present in the dataset. Overall, the chart emphasizes the importance of choosing the right algorithm based on data characteristics to achieve robust and accurate classification outcomes.

Furthermore, the results of applying ensemble learning models for mushroom classification can be seen in Table 6.

*Table 6. Classification Results with Esemble Methods*

| *Model* | *Accuracy* | *Precision* | *Recall* | *F1 Score* | *AUC* |
|---|---|---|---|---|---|
| *Voting Classifier* | 0.999 | 0.999 | 0.999 | 0.999 | 1.000 |
| *Stacking dan Bagging* | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| *Bagging dan Stacking* | 0.841 | 0.841 | 0.841 | 0.841 | 0.913 |
| *Random Forest* | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| *Gradient Boosting* | 0.934 | 0.935 | 0.934 | 0.934 | 0.984 |
| *XGBoost* | 0.999 | 0.999 | 0.999 | 0.999 | 0.999 |
| *AdaBoost* | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 |

Based on the evaluation results of the ensemble learning models shown in Table 4, it can be concluded that most of the algorithms performed very highly, even approaching perfect scores on all evaluation indicators. Stacking, Bagging, and Random Forest models scored 1.000 for accuracy, precision, recall, F1-score, and AUC. This indicates that these approaches could optimally integrate the strengths of various base models to solve the task of classifying poisonous and edible mushrooms. The Voting Classifier model also displayed nearly perfect performance, with all metrics approaching a value of 0.999. This suggests that the voting mechanism among models can provide highly reliable prediction results.

Meanwhile, the XGBoost and AdaBoost models delivered very competitive results, with evaluation values above 0.998 and AUC nearing 1, demonstrating the effectiveness of the boosting approach in handling classification complexities. However, not all ensemble learning model combinations provided optimal performance. The combination of Bagging and Stacking only achieved an accuracy of 84.13%, with other metric values also being in a similar range and an AUC of 0.913. This indicates that the success of the ensemble learning approach heavily depends on the selection and integration of base models. The Gradient Boosting model also showed decent results, with accuracy and other metrics ranging around 93% and an AUC of 0.984. Although it was not superior to XGBoost or AdaBoost, this model remains a solid alternative. Overall, ensemble learning approaches such as Random Forest, Stacking and Bagging, and Voting Classifiers proved to be the superior choices for classifying this dataset.

In the results section, it was found that the model achieved a very high accuracy rate, both on the training data and the testing data. This condition highlights the risk of overfitting, which occurs when a model becomes overly adapted to the training data and consequently performs poorly on unseen data. Overfitting is often a significant concern in machine learning model evaluation because, although training accuracy can reach 100%, the true performance is assessed by its generalization ability. However, despite the training accuracy reaching perfect scores in this study, the nearly ideal testing accuracy shows that the model could still maintain strong generalization. Therefore, the results demonstrate that the model is reliable on known data and effective in predicting unseen data. This can be seen from the overfitting results presented in Table 7.

*Table 7. Overfitting Evaluation*

| | |
|---|---|
| *CV Accuracy Scores* | [1, 0,99988304 1, 1, 1,] |
| *Mean CV Accuracy* | *0,999* |
| *Training Accuracy* | *1* |
| *Testing Accuracy* | 0,999 |

Based on the results obtained, it can be seen that the model demonstrates awe-inspiring performance. The CV Accuracy Scores, ranging from 99.99% to 100% across five-fold cross-validation, indicate that the model has high consistency in learning patterns from the training data. The average cross-validation accuracy reaches

0.999, meaning the model can maintain outstanding performance across various validation scenarios. During training, the model achieved 100% accuracy on the training data, showing that all data were predicted without errors. While perfect training accuracy usually raises concerns about overfitting, this was refuted by the evaluation results of the testing data. The model achieved an accuracy of 99.99% on the testing data, proving that its ability to generalize remains exceptionally well-maintained. Overall, this performance confirms that the built Random Forest model is highly reliable, stable, and suitable for making predictions on similar data in the future.

## 4. Discussions

The experiment results show that Random Forest, KNN, and the combination of Stacking and Bagging achieved perfect performance with accuracy, precision, recall, F1-score, and AUC values 1.0. This condition indicates that all three models can separate the poisonous and edible mushroom classes without errors. The impressive performance of Random Forest can be explained by its method, which builds multiple decision trees and combines the predictions of each tree to reduce the risk of overfitting and improve accuracy [19]. Meanwhile, KNN operates based on the distance between data points, and in a mushroom dataset with well-defined attribute patterns, this method can provide very accurate predictions. This finding aligns with research by [20], which showed that KNN and Decision Trees excel in handling large datasets with complex class distributions. The combination of stacking and bagging approaches also produced perfect results, achieving 100%. This result is consistent with the study by [21] which showed that ensemble learning methods such as Bagging and Boosting are generally more accurate than single models. However, boosting performance can be influenced by dataset characteristics. However, not all ensemble learning combinations yielded the best results, as seen in the combination of Bagging and Stacking, which performed more similarly to the Logistic Regression model than to Random Forest or KNN. Overall, these findings support the existing literature emphasizing that when combined and optimized correctly, ensemble learning can be an effective solution to enhance accuracy and model robustness against overfitting, particularly in complex categorical data classification tasks like poisonous mushroom identification.

Despite its reputation for being fast and computationally efficient, the Naïve Bayes algorithm exhibited significantly lower performance in this study compared to other models, with an accuracy of only 59.8%. This underperformance stems not only from its fundamental assumption that features are conditionally independent but also from the nature of the dataset itself. In the mushroom dataset used, several attributes—such as odor, spore print color, gill color, and cap shape—are not only highly informative but also strongly correlated with one another. Since Naïve Bayes fails to capture such interdependencies, it produces inaccurate estimations of class probabilities, leading to frequent misclassifications. Furthermore, the presence of imbalanced distributions in specific attribute values, such as the "foul" odor which predominantly appears in poisonous mushrooms, causes the model to overemphasize certain features while ignoring subtler patterns that are equally important for accurate prediction. These limitations make Naïve Bayes unsuitable for complex classification tasks involving rich categorical data with overlapping feature relationships. The performance contrast among models in this study highlights important considerations for algorithm selection. More advanced models like Random Forest, K-Nearest Neighbors (KNN), and ensemble techniques such as Stacking and Bagging proved far more capable of handling datasets with intricate dependencies among attributes. In contrast, simpler models like Naïve Bayes may still be effective when applied to data with more clearly separated and independent features, but fall short in capturing the complexity found in real-world biological classification problems such as mushroom toxicity prediction.

Thus, this study provides an additional contribution to strengthen the empirical evidence that in the mushroom dataset, models with the ability to capture attribute complexity (such as Random Forest, XGBoost, and KNN) outperform models based on simple assumptions (such as Naïve Bayes). Furthermore, the use of a combination of traditional models and ensemble learning also opens up new opportunities to enhance classification accuracy in the domain of applied biology.

## 5. Conclusions and Future Works

This study successfully demonstrates that traditional classification models, such as KNN, can achieve perfect performance in distinguishing between poisonous and edible mushrooms. These results emphasize the effectiveness of models that capture complex relationships between attributes in a categorical dataset. On the other hand, simpler methods such as Naïve Bayes are less able to deliver optimal performance due to their independence assumption, which does not align with the interdependent characteristics of mushroom data. Applying ensemble learning methods, such as Random Forest and the combination of Stacking and Bagging, has proven to preserve and even enhance model performance with very high consistency in evaluation. Overall, this research reinforces the evidence that ensemble learning approaches can improve models' stability, accuracy, and generalization ability, particularly in complex data-based classification tasks.

Based on the results obtained, it is recommended that future research explore more varied combinations of ensemble learning models, including boosting-based methods such as AdaBoost or Gradient Boosting, to assess further the potential for improvements in accuracy and model robustness. Additionally, testing these methods on more diverse mushroom datasets or other real-world datasets would provide deeper insights into the generalization ability of the models. Although this study has successfully demonstrated the capabilities of traditional and ensemble learning methods in classifying mushrooms using structured tabular data, there remain several opportunities for future exploration. One important direction is to incorporate Explainable Artificial Intelligence (XAI) approaches, such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations). These interpretability tools would help reveal the influence of each individual feature on the model's predictions, thereby enhancing transparency and making the models more trustworthy, especially in safety-critical applications like food consumption and public health.

Another valuable extension would be to test the models on alternative data types, such as real-world images of mushrooms or descriptive textual inputs. Such experimentation would assess how well the current models generalize beyond structured data, which is crucial when applying machine learning in real-world environments where tabular formats are not always available. Broadening the input modalities not only validates model robustness but also paves the way for developing more flexible, multi-modal classification systems capable of handling diverse data sources.

## 6. References


[1] Z. Yan, H. Liu, J. Li, and Y. Wang, "Application of Identification and Evaluation Techniques for Edible Mushrooms: A Review," 2023, *Taylor and Francis Ltd.* doi: 10.1080/10408347.2021.1969886.

[2] R. Gürfidan and Z. Akçay, "Real-time Deep Learning Based Mobile Application for Detecting Edible Fungi: Mushapp," *International Journal of Intelligent Systems and Applications*, vol. 16, no. 5, pp. 1–9, Oct. 2024, doi: 10.5815/ijisa.2024.05.01.

[3] S.-T. Chang and S. P. Wasser, "The Role of Culinary-Medicinal Mushrooms on Human Welfare with a Pyramid Model for Human Health," *Int J Med Mushrooms*, vol. 14, no. 2, pp. 95–134, 2012, doi: 10.1615/IntJMedMushr.v14.i2.10.

[4] Y. Zhang, C. Venkitasamy, Z. Pan, and W. Wang, "Recent developments on umami ingredients of edible mushrooms – A review," *Trends Food Sci Technol*, vol. 33, no. 2, pp. 78–92, Oct. 2013, doi: 10.1016/j.tifs.2013.08.002.

[5] I. P. Putra, "kasus-Kasus Keracunan Jamur Liar di Indonesia," *Jurnal Ekologi Kesehatan*, vol. 20, no. 3, pp. 215–230, Mar. 2022, doi: 10.22435/jek.v20i3.4943.

[6] I. P. Putra, "Laporan kasus keracunan Chlorophyllum cf. molybdites di Surabaya, Indonesia," *Jurnal Agercolere*, vol. 3, no. 1, pp. 1–6, Feb. 2021, doi: 10.37195/jac.v3i1.120.

[7] S. Jehadu and Krisiandi, "Kadinkes Sebut 6 Warga Keracunan Usai Makan Jamur di Sikka Sudah Sembuh Artikel ini telah tayang di Kompas.com dengan judul 'Kadinkes Sebut 6 Warga Keracunan Usai

Makan Jamur di Sikka Sudah Sembuh’, Klik untuk baca: https://regional.kompas.com/read/2023/03/02/150839078/kadinkes-sebut-6-warga-keracunan-usai-makan-jamur-di-sikka-sudah-sembuh. Kompascom+ baca berita tanpa iklan: https://kmp.im/plus6 Download aplikasi: https://kmp.im/app6,” *Kompas*, Sikka, Mar. 02, 2023.

[8] A. Rohman, “17 warga Kebon Kalapa Sukabumi keracunan jamur,” *ANTARA*, Sukabumi, Dec. 25, 2024.

[9] N. J. Pinky, S. M. M. Islam, and R. S. Alice, “Edibility Detection of Mushroom Using Ensemble Methods,” *International Journal of Image, Graphics and Signal Processing*, vol. 11, no. 4, pp. 55–62, 2019, doi: 10.5815/ijigsp.2019.04.05.

[10] R. Hamonangan, M. Bagus Saputro, C. Bagus, S. Dinata, and K. Atmaja, “Accuracy of classification poisonous or edible of mushroom using naïve bayes and K-nearest neighbors.”

[11] F. T. Admojo, M. L. Radhitya, H. Zein, and A. Naswin, “Classification of Mushroom Edibility Using K-Nearest Neighbors: A Machine Learning Approach,” *Indonesian Journal of Data and Science*, vol. 5, no. 3, pp. 243–250, Dec. 2024, doi: 10.56705/ijodas.v5i3.199.

[12] T. A. Mir, D. Banerjee, R. Chauhan, and H. S. Pokhariya, “A Novel Framework for Automated Mushroom Disease Diagnosis using CNN and Random Forest,” in *2024 4th Asian Conference on Innovation in Technology, ASIANCON 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ASIANCON62057.2024.10838126.

[13] M. S. Tuna and A. Kristianto, “Klasifikasi Cuaca Berbasis Citra dengan Model CNN LeNet-5 yang Dimodifikasi,” *J-Intech : Journal of Information and Technology*, no. 204, pp. 401–410, 2022.

[14] M. Arslan *et al.*, “A Comparative Study of Machine Learning Methods for Optimizing Mushroom Classification”, doi: 10.56979/801/2024.

[15] Y. Rakesh Kumar, V. Chandrashekar, and N. Vemula, “Mushroom Disease Detection and Classification Using Machine Learning Techniques,” in *IEEE International Conference on Data Engineering and Communication Systems, ICDECS 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICDECS59733.2023.10503469.

[16] V. Moysiadis, G. Kokkonis, S. Bibi, I. Moscholios, N. Maropoulos, and P. Sarigiannidis, “Monitoring Mushroom Growth with Machine Learning,” *Agriculture (Switzerland)*, vol. 13, no. 1, Jan. 2023, doi: 10.3390/agriculture13010223.

[17] W. Werapan, U. Suksawatchon, S. Srikamdee, and J. Suksawatchon, “DeepThaiMush: A Deep Learning Approach to Classify Poisonous and Edible Mushrooms using YOLOv8,” in *7th International Conference on Information Technology, InCIT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 75–80. doi: 10.1109/InCIT60207.2023.10413176.

[18] Y. Wang, L. Yang, H. Chen, A. Hussain, C. Ma, and M. Al-Gabri, “Mushroom-YOLO: A deep learning algorithm for mushroom growth recognition based on improved YOLOv5 in agriculture 4.0,” in *IEEE International Conference on Industrial Informatics (INDIN)*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 239–244. doi: 10.1109/INDIN51773.2022.9976155.

[19] L. Breiman, “Random Forest,” *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[20] J. E. Simarmata, G.-W. Weber, and D. Chrisinta, “Performance Evaluation of Classification Methods on Big Data: Decision Trees, Naive Bayes, K-Nearest Neighbors, and Support Vector Machines,” *Jurnal Matematika, Statistika dan Komputasi*, vol. 20, no. 3, pp. 623–638, May 2024, doi: 10.20956/j.v20i3.32970.

[21] D. Opitz and R. Maclin, “Popular Ensemble Methods: An Empirical Study,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, Aug. 1999, doi: 10.1613/jair.614.