International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 15, No. 1, March 2026, pp. 138~151
ISSN: 2252-8776, DOI: 10.11591/ijict.v15i1.pp138-151

A decision support system for mushroom classification using Naïve Bayesian algorithm

Vilchor G. Perdido[1], **Thelma D. Palaoag**[2]
[1]Department of Computer Science, College of Information Technology Education,
Nueva Vizcaya State University, Nueva Vizcaya, Philippines
[2]College of Information Technology and Computer Science, University of the Cordilleras, Baguio City, Philippines

Article Info

Article history:

Received Nov 28, 2024
Revised Jul 29, 2025
Accepted Oct 7, 2025

Keywords:

Area under the curve
Classification model
Cross-validation
Decision support system
KabuTeach
Naïve Bayes algorithm

ABSTRACT

Mushrooms are rich in vitamins and proteins, a well-known superfood, however, cases of harmful mushroom consumption worldwide result in hallucinations, illness, or death. A significant challenge is that some poisonous mushrooms closely resemble edible varieties, making it difficult for mushroom foragers to distinguish between them. This study introduced KabuTeach, a decision support system (DSS) designed to classify mushrooms based on their morphological characteristics using the Naïve Bayes (NB) algorithm. The classification model was applied to a real-world dataset of 8,124 instances from Kaggle, containing 23 attributes. Evaluation metrics, including accuracy, recall, precision, specificity, and F1-score, were used to assess the classifier's performance. Results indicated that the NB classification algorithm integrated into KabuTeach achieved a high accuracy level of 89.13%, using a 70:30 data split and 5-fold cross-validation approaches. The 0.98 AUC (area under the curve) value further concluded that the model was excellent in classifying between edible and poisonous mushrooms. These findings showed that KabuTeach is a reliable classification tool that aids mushroom foragers in differentiating mushrooms and promoting safer consumption practices. This innovation in agricultural technology could potentially reduce health risks by minimizing accidental ingestion of toxic mushrooms, ultimately contributing to public health safety.

Corresponding Author:

Vilchor G. Perdido
Department of Computer Science, College of Information Technology Education
Nueva Vizcaya State University
Bayombong, Nueva Vizcaya, Philippines
Email: vgperdido@nvsu.edu.ph

1. INTRODUCTION

Mushrooms are increasingly recognized as one of the healthiest foods due to their rich nutritional content, including calcium, phosphorus, vitamins, and proteins. They offer numerous health benefits, such as boosting immunity, aiding in weight loss, and combating cancer [1]. Despite their benefits, distinguishing between edible and poisonous mushrooms remains a significant challenge, as many species resemble one another, and incorrect identification can lead to severe health consequences [2].

The national poison data system in the United States recorded 133,700 cases of mushroom exposure between 1999 and 2016, with an additional 6,136 cases reported in 2017 [3]. In Germany, hospital data from 2000 to 2018 documented 4,412 hospitalizations and 22 fatalities caused by the toxic effects of mushroom consumption [4]. Distinguishing between edible and poisonous mushrooms is challenging and requires specialized knowledge. Since most mushrooms are inedible, consuming foraged mushrooms without proper identification is a serious mistake. The consequences of eating poisonous mushrooms could range from mild symptoms to death. As mushrooms become increasingly popular as a food source, the difficulty in visually distinguishing poisonous varieties from edible ones might explain the rising number of poisoning incidents [5].

The classification of objects is an important area within the field of data mining, and its application extends to a variety of areas. With advancements in emerging technologies, machine learning (ML) has become a powerful tool for learning from large, problem-specific training datasets, enabling the automation of intelligent model building and solving associated tasks [6]. ML has become a pivotal tool in classification tasks, enabling computers to learn from data and make predictions without explicit programming. Classification involves predicting categorical outcomes based on input features [7] and is widely used in image classification, predictive modeling, and data mining domains. For instance, when ML models are applied to mushroom classification, they could help identify edible and poisonous species based on morphological features [8]. With these massive collections of mushroom data available, classifying poisonous or toxic and edible mushrooms is important to address the global issue of mushroom poisoning [9], especially to the local communities. Several classification algorithms are commonly used in ML, each with its strengths and weaknesses. These include Bayesian networks, decision tree (DT) induction, K-nearest neighbor (KNN) classifiers, and support vector machines (SVM) [10], [11]. More advanced techniques like random forests (RF), extreme gradient boosting (XGBoost), and Naïve Bayes (NB) are frequently employed for their computational efficiency and accuracy in specific tasks [12], [13]. Additionally, with the help of decision support systems (DSS) and the integration of these various ML algorithms, it provides an interactive platform to assist users in making informed decisions. A DSS architecture mostly consists of the database (or knowledge base), the model or algorithm, and the user interface [14].

This study developed a DSS application to classify mushrooms depending on their morphological features or characteristics. The analysis was based on the end-user's interaction with the system. Then the mushroom was classified using the NB classification algorithm, and the results were presented as a final decision on whether it was an edible or poisonous mushroom. The study also would like to determine the performance of the system with the selected classifier model applied to a real-world dataset. Common evaluation measures sometimes referred to as performance metrics, were employed to measure the efficacy or effectiveness of the classification model.

2. RESEARCH METHOD

This section delves into the mushroom dataset and the methods used for classification. The goal of this study was the classification if mushrooms were edible or poisonous by integrating a machine-learning technique. This could be achieved by developing KabuTeach, a DSS that analyzed the input characteristics and concluded precise decisions that led to mushroom classification. The classification methods were divided into four stages: i) data source; ii) sampling; iii) mushroom classification; and iv) model performance evaluation. The general flow of the process is illustrated in the conceptual framework shown in Figure 1.

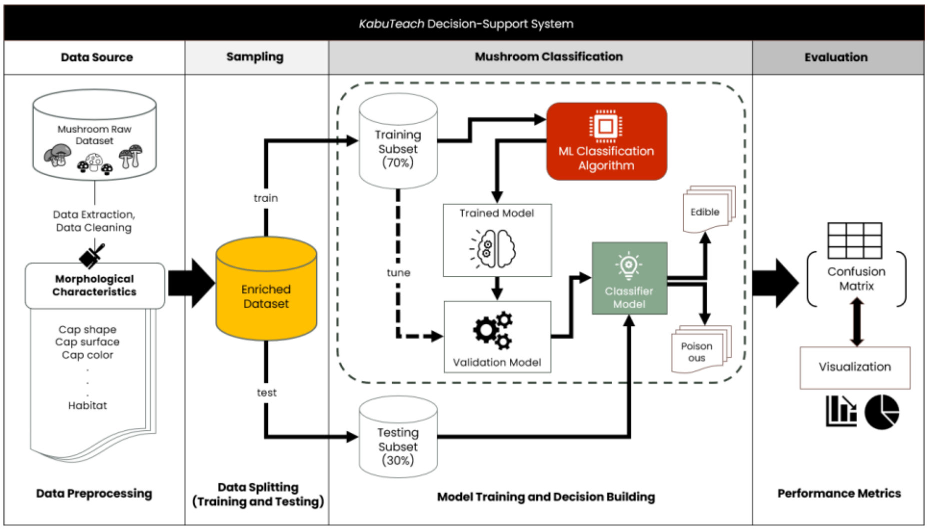


Figure 1. The conceptual framework of the study

The process began with the data preparation. The data source of the system used a mushroom dataset downloaded from an open-source mushroom repository on the internet. Data profiling and data cleaning activities are done in the first stage. The first activity is done to analyze the statistical features including the data size and type of the data. The latter is conducted to avoid data anomalies and irregularities in decision-making and ensure the reliability of data for more accurate predictive models. Next, the enriched data is uploaded into the system and split into training and testing datasets. These sample datasets are stored in a separate table in the database. In the mushroom classification, the training dataset is used to build and train the ML classification algorithm to come up with a trained model. In addition, a hyperparameter tuning mechanism is added to reveal whether the model is overfitting or underfitting. This validation model selects the best model configuration to improve the reliability of the model evaluation (classifier model). Once the final model is tuned up, then, the KabuTeach with the help of the classifier model learns some kind of patterns from the training dataset and applies them to the test dataset to build a decision in predicting or classifying whether an instance of the test data is edible or a poisonous mushroom. Performance metrics such as the confusion matrix, accuracy, precision, recall, specificity, and F1-score, including the area under curve (AUC) receiver operating characteristics (ROC) score are presented through a visualization page of the developed system.

2.1. Data source

The dataset used in this study was collected from the UC irvine machine learning (UCI ML) repository through the Kaggle web portal, a popular online platform for data scientists and ML experts or professionals [15]. This dataset was donated to UCI ML by Jeffrey Schlimmer on April 27, 1987, including descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the *Agaricus* and *Lepiota* Family of mushrooms [16], [17].

Data preprocessing: morphological features are extracted and used in the training. These morphological features (attributes) as summarized in Table 1 were used in the classification of mushrooms. Data pre-processing involves preparing the data for the ML model. This data mining technique is used to convert raw data into a more interpretable and structured format intended for use as training data before the mining process [18], [19]. This study employed two data pre-processing stages. Data profiling was the first stage which was done by examining and analyzing instances of the collected mushroom dataset to collect statistics about its data content. There were 22 attributes that represented the morphological characteristics of the mushroom used as the basis for designing the classifier module of the DSS.

Table 1. Attributes of mushroom description in the dataset

No	Attribute	Description and values
1	cap shape	b=bell, c=conical, x=convex, f=flat, k=knobbed, s=sunken
2	cap surface	f=fibrous, g=grooves, y=scaly, s=smooth
3	cap color	n=brown, b=buff, c=cinnamon, g=gray, r=green, p=pink, u=purple, e=red, w=white, y=yellow
4	bruises	t=bruises, f=no
5	odor	a=almond, l=anise, c=creosote, y=fishy, f=foul, m=musty, n=none, p=pungent, s=spicy
6	gill attachment	a=attached, d=descending, f=free, n=notched
7	gill spacing	c=close, w=crowded, d=distant
8	gill size	b=broad, n=narrow
9	gill color	k=black, n=brown, b=buff, h=chocolate, g=gray, r=green, o=orange, p=pink, u=purple, e=red, w=white, y=yellow
10	stalk shape	e=enlarging, t=tapering
11	stalk root	b=bulbous, c=club, u=cup, e=equal, z=rhizomorphs, r=rooted, ?=missing
12	stalk surface above ring	f=fibrous, y=scaly, k=silky, s=smooth
13	stalk surface below ring	f=fibrous, y=scaly, k=silky, s=smooth
14	stalk color above ring	n=brown, b=buff, c=cinnamon, g=gray, o=orange, p=pink, e=red, w=white, y=yellow
15	stalk color below ring	n=brown, b=buff, c=cinnamon, g=gray, o=orange, p=pink, e=red, w=white, y=yellow
16	veil type	p=partial, u=universal
17	veil color	n=brown, o=orange, w=white, y=yellow
18	ring number	n=none, o=one, t=two
19	ring type	c=cobwebby, e= evanescent, f=flaring, l=large, n=none, p=pendant, s=sheathing, z=zone
20	spore print color	k=black, n=brown, b=buff, h=chocolate, g=green, o=orange, u=purple, w=white, y=yellow
21	population	a=abundant, c=clustered, n=numerous, s=scattered, v=several, y=solitary
22	habitat	g=grasses, l=leaves, m=meadows, p=paths, u=urban, w=waste, d=woods
23	class	This is the target variable that must predict or forecast; a value of 'e' denotes a mushroom is edible whereas a value of 'p' indicates a poisonous mushroom.

The attribute class was used by the NB algorithm to compute the probability and predict an instance (mushroom) whether it was an edible or poisonous class. Each attribute or character had values to select, and values were coded or represented by texts. The gill color attribute had the highest number of values with 12 values while bruises, gill size, stalk shape, veil type, and ring number attributes consist of two values. Data cleansing was the second stage and was conducted using Python libraries. This was done to prevent data anomalies by eliminating or removing missing (null) values and duplicated features in the mushroom dataset [20], [21]. Data transformation (categorical to numerical or vice versa) was no longer performed in this stage since values of the instances in the dataset were in categorical form. The NB algorithm is a classifier that was mostly applied in text classification, so performed well with categorical input variables compared to numerical values [22].

2.2. Sampling

The effectiveness of ML models could be tested using cross-validation (CV) techniques. This could be performed by splitting the whole data into training and testing datasets [23]. To get reliable findings from all of the data, the accuracy of each K-model's results is then averaged. The purpose of K-fold CV was to remove bias from the data. In this paper, a 5-fold CV method with a 70:30 balance ratio for training and testing was applied where the whole data was divided into five (5) folds and repeated five (5) times.

2.3. Mushroom classification using the Naïve Bayesian algorithm

There was no "best" ML algorithm and usually critical and difficult to choose, however, the correct selection was necessary [24]. Finding a suitable algorithm depends on the type of problem to solve [25] and on many factors such as the size, quality, and type or nature of datasets [26]. The NB algorithm was chosen as the classification model for several reasons. First, since the main objective of this study was to develop a DSS in training a model for prediction, therefore a supervised learning type of ML algorithm was needed. Second, the publicly available mushroom dataset contained 22 attributes (independent variable) which were used to classify a target variable (class) whether edible ('e') or poisonous ('p') category, thus a classification technique was necessary. In addition, datasets that had many attributes could be handled by the NB algorithm [16]. Third, instances (rows) in the dataset were organized by specific morphological features or attributes as shown in Table 1, each with categorical values (text) representing the physical characteristics of mushrooms. The NB algorithm was commonly applied in various applications which was often highly applied in text classification, spam filtering, sentiment analysis, medical diagnosis, and recommender systems. Lastly, with over 8,000 rows of data found in the dataset, this algorithm was known for its simplicity, efficiency, and effectiveness with high accuracy and speed in handling large datasets [17], [20].

Model training and decision building: the NB model was not only simple but also easy to build. This classification algorithm operated on the principle of conditional independence, meaning it assumed that the value of each attribute within a specific class did not depend on the values of other attributes [27]. Implementing the NB algorithm to the DSS system involved several key steps that ensured accurate model training and effective classification:

1) Load the mushroom dataset for data profiling and cleaning.
2) Divide or split the dataset into two: training and testing subsets.
3) Apply the classification algorithm to train the training subsets.
4) Tune up the trained model using a validation technique to come up with a final model.
5) The final (classier) model classifies mushrooms using the following procedures:
   a. Calculate the prior probability for each class ('e' or 'p') by finding the proportion of each target class in the training data.
   b. For each feature in the dataset, determine the probability (likelihood) of each possible value occurring within each class.
   c. For a new instance, calculate the posterior probability using the formula in (1) derived from Bayes' Theorem.

$$P(A \mid B) = \frac{P(B \mid A)\,.\,P(A)}{P(B)} \tag{1}$$

The formula of NB classification with multiple features $X_1, X_2, X_3, \dots X_n$ could be extended in (2).

$$\underset{\substack{\downarrow\\ \textit{Posterior Probability}}}{P(Class \mid X_1, X_2, X_3, \dots X_n)} = \frac{\overset{\textit{Likelihood}}{P(X_1 \mid Class)\,.\,P(X_2 \mid Class)\,.\,P(X_3 \mid Class) \dots P(X_n \mid Class)}\,.\,\overset{\textit{Class Prior Probability}\ \nwarrow}{P(Class)}}{\underset{\substack{\downarrow\\ \textit{Predictor Prior Probability}}}{P(X_1, X_2, X_3, \dots X_n)}} \tag{2}$$

where, $P(Class|X_1, X_2, X_3, \dots X_n)$ was the posterior probability of a mushroom being in a specific class (either ‘e’ for ‘edible’ or ‘p’ for poisonous) given the features $X_1, X_2, X_3, \dots X_n$, $P(X_1|Class), P(X_2|Class), P(X_3|Class) \dots P(X_n|Class)$ were the conditional probabilities (likelihood) of how likely it was to observe each feature (mushroom attributes) if the mushroom belongs to that class, $P(Class)$ was the prior probability of the class, representing the likelihood of a mushroom being in that class (‘e’ or ‘p’) without considering any features, and $P(X_1|Class), P(X_2|Class), P(X_3|Class) \dots P(X_n|Class)$ was the total probability of observing the features X$X_1, X_2, X_3, \dots X_n$ across all classes.

6) Test the testing subsets into the classier model. The classification outcome (‘edible’ or ‘poisonous’) was determined by the class with the highest posterior probability.

2.4. Performance measures and evaluation

The final accuracy of the model was assessed through a confusion matrix (error matrix). Typically, the performance of classification algorithms was evaluated based on their overall results on the test dataset [17], [28]. In this case, the number of edible and poisonous mushrooms that were correctly and incorrectly classified by the classifier could be summarized and generated through a confusion matrix. Table 2 shows the confusion matrix, a tabular representation that illustrates the performance of an algorithm or model in classification tasks.

Table 2. Visualization of the confusion matrix

Classification		Value predictions	
		Edible	Poisonous
Actual value	Edible	True positive (TP)	False negative (FN)
	Poisonous	False positive (FP)	True negative (TN)

Based on the data shown in the confusion matrix, there were common evaluation metrics that could be used to measure the performance of the NB algorithm specifically to its accuracy, indicating the precision or correctness in classifying edible or poisonous mushrooms from the given dataset. Accuracy was computed using the formula in (3).

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \; x \; 100\% \tag{3}$$

Other metrics were also applied in this study. Precision could be computed in (4), recall was calculated in (5), and Specificity could be solved using the formula in (6).

$$Precision = \frac{(TP)}{(TP+FP)} \; x \; 100\% \tag{4}$$

$$Recall = \frac{(TP)}{(TP+ FN)} \; x \; 100\% \tag{5}$$

$$Specificity = \frac{(TP)}{(TN + FP)} \; x \; 100\% \tag{6}$$

On the other hand, the F1-score was the harmonic mean of precision and recall which can be computed using the formula in (7).

$$F1 - Score = \; 2 \; x \; \frac{(Precision \; x \; Recall)}{(Precision + Recall)} \; x \; 100\% \tag{7}$$

The performance of the model was further evaluated using the ROC curve and the AUC metrics, which were most used in ML to evaluate the effectiveness of a two-class classification model [29]. The ROC curve was created by plotting the true positive rate (TPR), also known as recall/sensitivity, against the false positive rate (FPR) at different threshold levels, while the AUC score was a single scalar value that gave an overall indication of how accurate the classifier can differentiate between classes [30].

The AUC score could be computed using the trapezoidal rule [31] after generating the ROC curve because the AUC represented the area beneath the curve [32]. The area of the trapezoid was calculated for each adjacent pair of points $(FPR_i,\ TPR_i)$ and $(FPR_{i+1},\ TPR_{i+1})$ using the formula shown in (8). To get the total AUC score, sum the area of each trapezoid across all adjacent points.

$$\text{Area}_i = \frac{(FPR_{i+1} - FPR_i) \times (TPR_{i+1} + TPR_i)}{2} \tag{8}$$

Where, $(FPR_{i+1} - FPR_i)$ represented the difference in FPR between two consecutive points, $(TPR_{i+1} + TPR_i)$ represented the difference in FPR between two consecutive points, $1/2$ averages the combined height of the trapezoid to account for the formula for the area of a trapezoid, Area_i denoted the calculated AUC for the interval between the i-th and $(i+1)$-th points.

In ML, the AUC score for a ROC curve was typically evaluated on a scale from 0 to 1 [31], [33], with different ranges often interpreted using a Likert-style rating as shown in Table 3. A score just above 0.5 showed the model has minimal predictive power, which was usually insufficient for practical applications. Generally, an ROC AUC score over 0.8 was regarded as good, and above 0.9 was considered excellent. This rating system was used to interpret AUC values to evaluate classifier effectiveness across different probability thresholds.

Table 3. AUC-ROC performance measurement

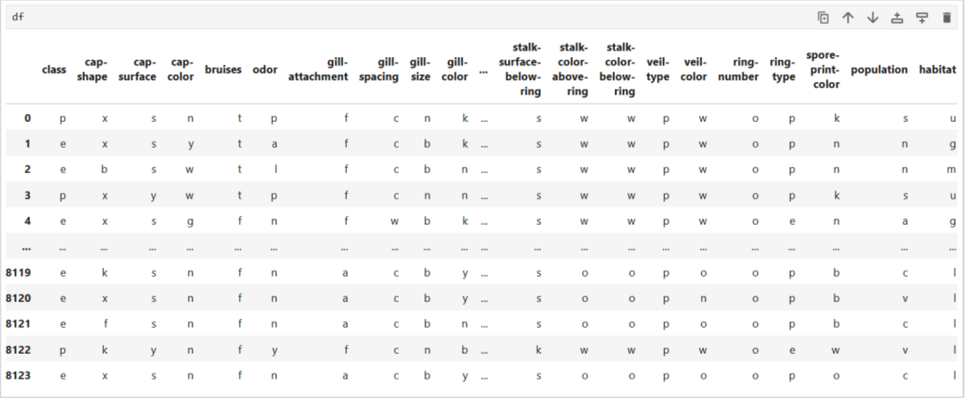
AUC range	Rating	Description
0.90 - 1.00	Excellent	Outstanding discrimination between classes.
0.80 - 0.90	Good	Strong classifier, reliable for most applications.
0.70 - 0.79	Fair	Moderate discrimination, useful in many scenarios.
0.60 - 0.69	Poor	Weak classifier; some improvement might be necessary.
0.50 - 0.59	Very poor	Barely better than random; generally unacceptable.
< 0.50	No discrimination	The model performs no better than random chance.

3. RESULTS AND DISCUSSION

This study primarily focused on the development of a DSS highlighting the potential of an embedded algorithm for the precise classification task. The DSS is a web application designed for mushroom pickers or foragers, allowing them to interact with the system by selecting predefined morphological characteristics. The system then automatically classifies the mushroom as either edible or poisonous. Several studies have utilized DSS for mushroom classification; however, most of these were designed and deployed as mobile applications using image processing [34]–[37], while others were purely comparisons of algorithms with no actual application development [38]–[41]. The study followed the common components involved in building a DSS, which include data selection, the designing of user interfaces, and the training of the model and testing of its performance. The present study explains the results of these components, as it is detailed in the following three sub-sections.

3.1. Data preparation and preprocessing

A raw data of mushrooms obtained from Kaggle [17], an online data source, was used as the dataset of this study. This raw data is originally stored in a comma-separated values (CSV) file and 374 KB of file size. This study utilized Python to create a dataset from the raw data commanded in the Jupyter Notebook. It had a total number of 8,124 data samples [42], [43] with 23 columns as shown in Figure 2. These columns were the 22 morphological characters (attributes) of mushrooms [8], [16], [18] which were important to classify one (1) target feature (class), either an edible ('e') or a poisonous ('p') type of mushroom. It was found that there were 4,208 (51%) instances belonging to the edible category while a total number of 3,916 (49%) instances were included in the poisonous category as shown in Figure 3.



df

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p	w	o	p	k	s	u
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p	w	o	p	n	n	g
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p	w	o	p	n	n	m
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p	w	o	p	k	s	u
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p	w	o	e	n	a	g
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8119	e	k	s	n	f	n	a	c	b	y	...	s	o	o	p	o	o	p	b	c	l
8120	e	x	s	n	f	n	a	c	b	y	...	s	o	o	p	n	o	p	b	v	l
8121	e	f	s	n	f	n	a	c	b	n	...	s	o	o	p	o	o	p	b	c	l
8122	p	k	y	n	f	y	f	c	n	b	...	k	w	w	p	w	o	e	w	v	l
8123	e	x	s	n	f	n	a	c	b	y	...	s	o	o	p	o	o	p	o	c	l

Figure 2. Structure and features of the data source

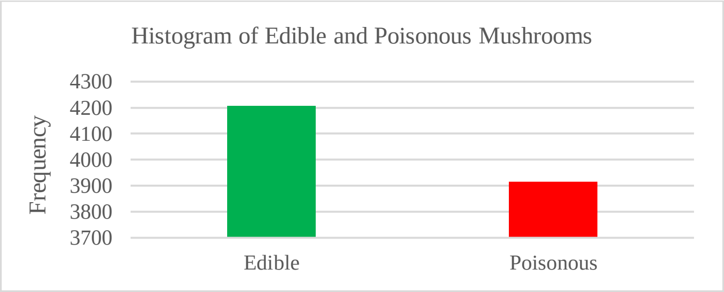
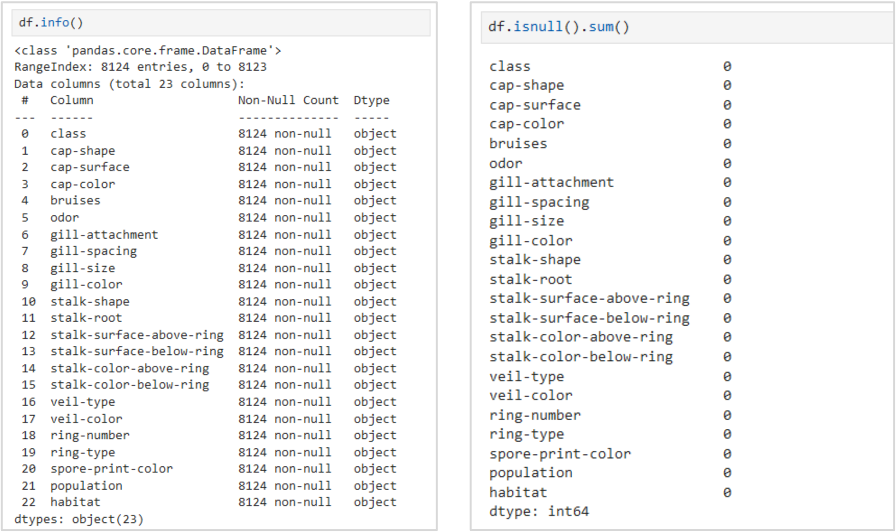


Figure 3. Count of the edible and poisonous classes

Additionally, see Figure 4, the data type of these columns was found objects as shown in Figure 4(a), where values contained texts, thus the expected type of data for each instance (row) of the dataset was in categorical form. As mentioned, the NB algorithm was highly applied in text classification. In data cleaning, identifying missing was necessary in this study prior to the training and testing of the classification algorithm. By using built-in functions in Python, it found out that there were no null values shown in Figure 4(b), thus the dataset had good data quality.



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8124 entries, 0 to 8123
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   class                     8124 non-null   object
 1   cap-shape                 8124 non-null   object
 2   cap-surface               8124 non-null   object
 3   cap-color                 8124 non-null   object
 4   bruises                   8124 non-null   object
 5   odor                      8124 non-null   object
 6   gill-attachment           8124 non-null   object
 7   gill-spacing              8124 non-null   object
 8   gill-size                 8124 non-null   object
 9   gill-color                8124 non-null   object
 10  stalk-shape               8124 non-null   object
 11  stalk-root                8124 non-null   object
 12  stalk-surface-above-ring  8124 non-null   object
 13  stalk-surface-below-ring  8124 non-null   object
 14  stalk-color-above-ring    8124 non-null   object
 15  stalk-color-below-ring    8124 non-null   object
 16  veil-type                 8124 non-null   object
 17  veil-color                8124 non-null   object
 18  ring-number               8124 non-null   object
 19  ring-type                 8124 non-null   object
 20  spore-print-color         8124 non-null   object
 21  population                8124 non-null   object
 22  habitat                   8124 non-null   object
dtypes: object(23)
```

(a)

```
df.isnull().sum()
```

```
class                       0
cap-shape                   0
cap-surface                 0
cap-color                   0
bruises                     0
odor                        0
gill-attachment             0
gill-spacing                0
gill-size                   0
gill-color                  0
stalk-shape                 0
stalk-root                  0
stalk-surface-above-ring    0
stalk-surface-below-ring    0
stalk-color-above-ring      0
stalk-color-below-ring      0
veil-type                   0
veil-color                  0
ring-number                 0
ring-type                   0
spore-print-color           0
population                  0
habitat                     0
dtype: int64
```

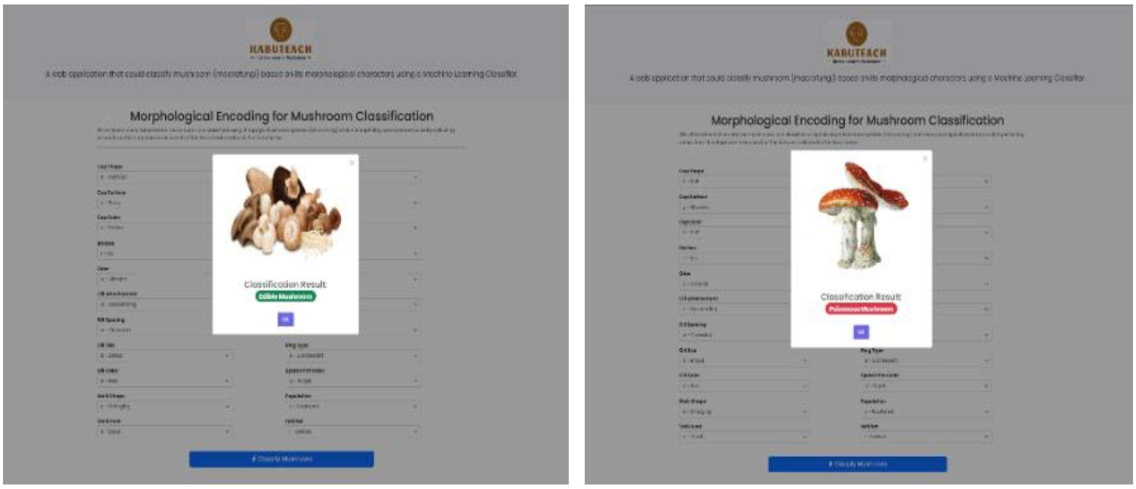
(b)

Figure 4. The extracted mushroom variables from the dataset to be used in classification consist of (a) object data types and (b) non-null values

3.2. The DSS for mushroom classification using Naïve Bayes algorithm

The developed DSS was known as KabuTeach, designed to differentiate or classify a mushroom, either edible (eatable) or poisonous (toxic) based on its morphological features as the inputs. The functionalities were mainly developed using Laravel 11, an open-source PHP framework, while user interfaces were generally designed in Bootstrap 5. The MariaDB database was used to store the datasets used during the training and testing of the NB classification algorithm.

The classifier module of the system was used by the forages (mushroom pickers) to test the classification of a collected and characterized mushroom. All morphological characters were encoded by carefully selecting values in the combo boxes. After selecting all the necessary characters, the system matched these characters with all the instances (records) from the historical dataset in the database. If none of the examples matches, then the NB classification algorithm classifies them in Figure 5, as edible (see Figure 5(a)) or poisonous (see Figure 5(b)), depending on which category has the highest posterior probability, and stores them in the test data set in the database.

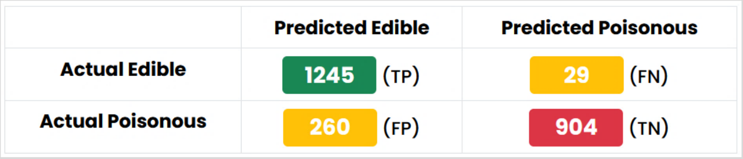


(a) (b)

Figure 5. The classification of mushrooms using the Bayes' Theorem integrated into the classifier module of KabuTeach predicts two classes (a) edible and (b) poisonous

3.3. Performance results

To measure the efficacy of the performance of the NB classification algorithm (model), KabuTeach provided a module where it could train and test the model. The splitting of data into two datasets and K-fold cross-validation approaches were applied in this study. This study applied the 70:30 ratio [43] random splitting mechanism, where 70% (5,686) of the data were used in training the NB classification model. The remaining 30% (2,438) were employed to test or evaluate its performance on the trained model. The evaluation metrics were derived from the resulting confusion matrix of the testing data fed to the algorithm shown in Figure 6 generated by the KabuTeach system.



	Predicted Edible	**Predicted Poisonous**
Actual Edible	1245 (TP)	29 (FN)
Actual Poisonous	260 (FP)	904 (TN)

Figure 6. The confusion matrix

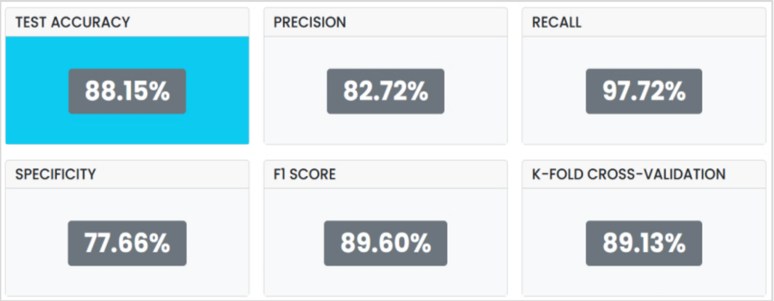
From this confusion matrix, the actual number of edible mushrooms is 1,274, while the number of poisonous mushrooms was 1,164, both stored in the testing dataset. In this figure, the NB classification model correctly classified 97.72% (TP) of the edible mushrooms, with 2.28% (FN) classified incorrectly. On the other hand, 77.66% (TN) of the poisonous mushrooms were correctly identified. However, it misclassified 22.34% (FP) as edible, which is a significant number given the risk of consuming poisonous mushrooms. With these figures indicated from the confusion matrix, the NB classifier performed well with edible mushrooms but struggled with poisonous ones, which could lead to potential health risks. While the TP rate for edible mushrooms is promising, the FP rate for poisonous mushrooms highlights a risk area that needs improvement. This study suggests that the model could improve the model's accuracy, especially for identifying poisonous mushrooms, which could involve refining features, gathering more data, or exploring other classification models.

Moreover, the overall model's Accuracy rate derived from the confusion matrix and computed using the formula in (9).

$$Accuracy = \frac{(1245 + 904)}{(1245 + 904 + 260 + 29)} \; x\; 100$$
$$Accuracy = \frac{2149}{2438} \; x\; 100 = 88.15$$
$$Accuracy = 0.8814602132895816 \; x\; 100 = 88.15\% \text{ (rounded to two decimal places)} \qquad (9)$$

Based on this computation, the model yielded a high accuracy rate of 88.15% correct classification. This indicated that the classifier performed effectively and reliably in distinguishing between edible and poisonous mushrooms. This level of accuracy implied that the model served its intended purpose and was good enough to be used in real-world applications, following the evaluation standards in Table 3 applied in the study of Cruz [27].

On the other hand, the accuracy of the NB classification model in classifying mushrooms was enhanced with the integration of the 5-fold CV technique. The result obtained an 89.13% high accuracy rate. Several studies have used the NB classification model in mushroom classification [16], [18], [44], which closely resembles the accuracy rate achieved in this study, good enough to be applied in classifying the types of poisonous and edible mushrooms. The results obtained from the cross-validation accuracy and the testing accuracy implied that the model performed consistently across different subsets of the data and was well-suited for practical applications. Other metrics, such as precision, recall, specificity, F1-score, [44], [45] and K-fold cross-validation, were automatically computed by KabuTeach, as shown in Figure 7.



TEST ACCURACY	PRECISION	RECALL
88.15%	82.72%	97.72%
SPECIFICITY	**F1 SCORE**	**K-FOLD CROSS-VALIDATION**
77.66%	89.60%	89.13%

Figure 7. The performance metrics of the NB classification model

In terms of precision, the model correctly identified 82.72% edible mushrooms as shown in (10) when it predicted a mushroom to be edible. A lower precision indicated some FP (poisonous mushrooms classified as edible), which could be risky.

$$Precision_{edible} = \frac{(1245)}{(1245 + 260)}\ x\ 100$$

$$Precision_{edible} = \frac{(1245)}{(1505)}\ x\ 100 = (0.8272425249169435\ x\ 100) = 82.72\% \text{ (rounded to two decimal places)} \quad (10)$$

On the other hand, a high precision of 96.89% computed in (11) for poisonous mushrooms meant that the model was generally accurate when it predicted a mushroom as poisonous. This high precision reduced the likelihood of edible mushrooms being falsely classified as poisonous, which was generally preferable. This result implies that the higher precision for poisonous mushrooms suggested that the classifier was conservative, prioritizing safety by leaning towards classifying mushrooms as poisonous unless it was highly confident. This was a beneficial trait for health-sensitive applications, as it minimized the chance of poisonous mushrooms being misclassified as edible.

$$Precision_{poisonous} = \frac{(904)}{(904 + 29)}\ x\ 10$$

$$Precision_{poisonous} = \frac{(904)}{(933)}\ x\ 100\ 0.9689174705251876\ x\ 100 = 96.89\% \text{ (rounded to two decimal places)} \quad (11)$$

The recall result computed in (12) was very high with 97.72%, indicating that nearly all edible mushrooms were correctly identified by the model. This reduced the chance of edible mushrooms being misclassified as poisonous, which helped foragers avoid mistakenly discarding safe mushrooms.

$$Recall = \frac{(1245)}{(1245 + 29)}\ x\ 100$$

$$Recall = \frac{(1245)}{(1274)}\ x\ 100 = (0.9772370486656201)\ x\ 100 = 97.72\% \text{ (rounded to two decimal places)} \quad (12)$$

Specificity, or the true negative rate (TNR) computed in (13), indicates that the classifier correctly identifies poisonous mushrooms 77.66% of the time. This is a significant metric because it reflects the model’s effectiveness in rejecting dangerous mushrooms. While the result is moderately high, enhancing specificity could further ensure user safety by minimizing the number of poisonous mushrooms classified as edible.

$$Specificity = \frac{(904)}{(904 + 260)}\ x\ 100$$
$$Specificity = \frac{(904)}{(1164)}\ x\ 100 = (0.7766323024054983\ x\ 100) = 77.66\% \text{ (rounded to two decimal places)} \tag{13}$$

The F1-score calculated using the formula in (14), which was the harmonic mean of recall and precision yielded 89.60%, highlighted that both metrics performed well and were balanced in the model’s classifications. This further explained that high recall was particularly critical because misclassifying a poisonous mushroom as edible could result in serious health risks. The high recall (97.7%) indicates that the model could minimize such risks. On the other hand, precision (82.7%) was good but could be improved to ensure fewer edible mushrooms were incorrectly classified as poisonous.

$$F1 - Score = \left(2\ x\ \frac{(82.72\ x\ 97.72)}{(82.72 + 97.72)}\right)\ x\ 100$$
$$F1 - Score = \left(2\ x\ \frac{(8083.3984)}{(180.44)}\right)\ x\ 100$$
$$F1 - Score = (2\ x\ 44.79826202615828\ )\ x\ 100$$
$$F1 - Score = (89.59652405231656\ )\ x\ 100 = 89.60\% \text{ (rounded to two decimal places)} \tag{14}$$

To further determine the degree of performance or effectiveness of the NB algorithm, KabuTeach plotted a chart of the ROC curve applied to the test data showing the TPR in the y-axis against the FPR in the x-axis for the different thresholds as shown in Figure 8. This helped the ability of the algorithm to differentiate between edible and poisonous classes in the dataset.

From this ROC curve presented, the AUC [46]–[48] was summarized, computed using the Trapezoidal rule, and obtained a score of 0.98. The effectiveness of a model using this metric was measured from 0 to 1. A higher AUC score (close to 1) indicated better model performance, like the study of [49] which all the four models used exhibited a high AUC score of above 0.90. In this study, with the AUC value computed, it clearly showed that the NB classification model was “excellent” in classifying between edible and poisonous mushrooms as indicated in Table 3. Furthermore, the achieved AUC score means that there is a very high probability (98%) that the NB classification model correctly distinguishes a randomly chosen edible mushroom and a randomly chosen poisonous one. Given this high AUC score, this study strongly agrees with the model’s ability to correctly classify mushrooms. However, the present study still suggests further validation and potential improvements for even greater reliability, especially in real-world applications.

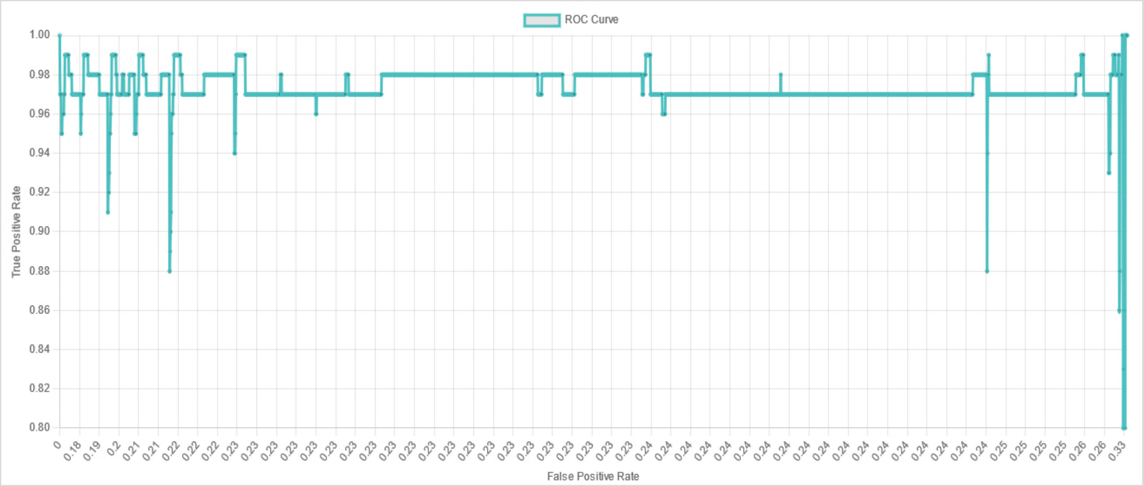


Figure 8. The testing data plotted in the ROC curve

To sum up, this study implemented the NB algorithm with a 70:30 data split and 5-fold cross-validation mechanisms. With all the results presented through the confusion matrix, the mushroom classification model demonstrates a strong performance with high recall, precision, and an excellent AUC score. Moreover, the model is particularly effective at identifying edible mushrooms, as indicated by the high recall rate. However, it is suggested that there is a need to enhance the specificity to reduce the risk of misclassifying poisonous mushrooms and help achieve a more balanced and reliable mushroom classification model.

4. CONCLUSION

This paper presented a DSS designed using the Laravel 11 framework to classify mushrooms as edible or poisonous based on their physical characteristics. The system achieved a high testing accuracy of 88.15%, which improved to 89.13% with 5-fold CV. The developed DSS, named KabuTeach, effectively utilized the NB algorithm to distinguish between edible and poisonous mushrooms. With a recall rate of 97.72%, the model accurately identified almost all edible mushrooms, reducing the risk of discarding safe ones. Furthermore, its high precision of 82.72% for poisonous mushrooms highlighted its safety-oriented approach, minimizing the likelihood of misclassifying toxic mushrooms as edible. The balanced F1-score of 89.60% and an AUC value of 0.90 confirmed the model’s excellence in classification tasks.

The development of KabuTeach promoted innovation in agricultural technology and food safety. The study concluded that the NB classifier is a highly effective model for mushroom classification, particularly when safety is prioritized. The DSS proved to be a practical tool for mushroom foragers and could be extended to other applications requiring categorical data classification. KabuTeach enhanced safe access to edible mushrooms, ensuring people could safely forage for mushrooms to reduce hunger and malnutrition, especially in rural and low-income communities where wild mushrooms are an accessible and sustainable food option. Additionally, KabuTeach is a reliable classification system that could reduce health risks associated with mushroom poisoning and minimize the likelihood of accidental consumption of toxic mushrooms, which could cause illness or death.

While the results are promising, future work could focus on improving the model’s specificity and further reducing the misclassification of poisonous mushrooms as edible for enhanced safety. To enhance the model, particularly in terms of specificity, it is recommended to balance the dataset to ensure an equal representation of edible and poisonous mushrooms. Additionally, exploring new features or transformations through feature engineering could help the model better distinguish between the two categories. Experimenting with different hyperparameters or algorithms might also improve performance. Implementing ensemble methods, such as RF or gradient boosting, can combine the strengths of multiple models for better results. Finally, applying regularization techniques will help prevent overfitting, ensuring the model generalizes well to new data. These steps will contribute to a more accurate and reliable mushroom classification model. Incorporating other classification models and techniques to compare performance using different evaluation metrics could yield the best results.

ACKNOWLEDGEMENTS

Our thanks to the University of the Cordilleras, especially to Dr. Thelma Domingo-Palaoag, our research teacher, for her invaluable support, knowledge, and guidance in pushing me to complete this research. I also would like to thank UCI ML and Kaggle for providing me with the available data source for free. Special thanks also to my loved ones who are always there to support me all the way, and most especially to our Almighty Father.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Vilchor G. Perdido	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Thelma D. Palaoag	✓	✓		✓	✓		✓			✓	✓	✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in the UCI ML repository via Kaggle at: https://www.kaggle.com/datasets/uciml/mushroom-classification/data.

REFERENCES

[1] D. M. Ba *et al.*, “Association of mushroom consumption with all-cause and cause-specific mortality among American adults: prospective cohort study findings from NHANES III,” *Nutrition Journal*, vol. 20, no. 1, p. 38, Dec. 2021, doi: 10.1186/s12937-021-00691-8.

[2] K. Kousalya, B. Krishnakumar, S. Boomika, N. Dharati, and N. Hemavathy, “Edible mushroom identification using machine learning,” in *2022 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2022, pp. 1–7, doi: 10.1109/ICCCI54379.2022.9741040.

[3] M. Beaumier, J.-P. Rioult, M. Georges, I. Brocheriou, T. Lobbedez, and A. Lanot, “Mushroom poisoning presenting with acute kidney injury and elevated transaminases,” *Kidney International Reports*, vol. 4, no. 6, pp. 877–881, Jun. 2019, doi: 10.1016/j.ekir.2019.02.016.

[4] R. Wennig, F. Eyer, A. Schaper, T. Zilker, and H. Andresen-Streichert, “Mushroom poisoning,” *Deutsches Ärzteblatt international*, Oct. 2020, doi: 10.3238/arztebl.2020.0701.

[5] V. Vaibhav, R. Meshram, Y. S, N. Jha, and G. Khorwal, “Mushroom poisoning: a case series with a literature review of cases in Asia Region,” *Cureus*, May 2023, doi: 10.7759/cureus.39550.

[6] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” *Electronic Markets*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/s12525-021-00475-2.

[7] S. Ray, “A quick review of machine learning algorithms,” in *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*, Feb. 2019, pp. 35–39, doi: 10.1109/COMITCon.2019.8862451.

[8] K. Tutuncu, I. Cinar, R. Kursun, and M. Koklu, “Edible and poisonous mushrooms classification by machine learning algorithms,” in *2022 11th Mediterranean Conference on Embedded Computing, MECO 2022*, Jun. 2022, pp. 1–4, doi: 10.1109/MECO55406.2022.9797212.

[9] S. Govorushko, R. Rezaee, J. Dumanov, and A. Tsatsakis, “Poisoning associated with the use of mushrooms: a review of the global pattern and main characteristics,” *Food and Chemical Toxicology*, vol. 128, pp. 267–279, Jun. 2019, doi: 10.1016/j.fct.2019.04.016.

[10] P. C. Sen, M. Hajra, and M. Ghosh, “Supervised classification algorithms in machine learning: a survey and review,” in *Advances in Intelligent Systems and Computing*, vol. 937, 2020, pp. 99–111.

[11] A. F. A. H. Alnuaimi and T. H. K. Albaldawi, “An overview of machine learning classification techniques,” *BIO Web of Conferences*, vol. 97, p. 00133, Apr. 2024, doi: 10.1051/bioconf/20249700133.

[12] A. E. Maxwell, T. A. Warner, and F. Fang, “Implementation of machine-learning classification in remote sensing: an applied review,” *International Journal of Remote Sensing*, vol. 39, no. 9, pp. 2784–2817, May 2018, doi: 10.1080/01431161.2018.1433343.

[13] H. Tan, “Machine learning algorithm for classification,” *Journal of Physics: Conference Series*, vol. 1994, no. 1, p. 012016, Aug. 2021, doi: 10.1088/1742-6596/1994/1/012016.

[14] Z. Zhai, J. F. Martínez, V. Beltran, and N. L. Martínez, “Decision support systems for agriculture 4.0: survey and challenges,” *Computers and Electronics in Agriculture*, vol. 170, p. 105256, Mar. 2020, doi: 10.1016/j.compag.2020.105256.

[15] M. Reddy and M. S. Saravanan, “Prediction of insufficient accuracy for mushroom classification whether poisonous or eatable food using k-nearest neighbour by comparing Naïve Bayes training to improve accuracy,” *Journal of Pharmaceutical Negative Results*, vol. 13, no. SO4, Jan. 2022, doi: 10.47750/pnr.2022.13.s04.058.

[16] R. Hamonangan, M. B. Saputro, and C. B. S. D. K. Atmaja, “Accuracy of classification poisonous or edible of mushroom using naïve bayes and k-nearest neighbors,” *Journal of Soft Computing Exploration*, vol. 2, no. 1, Mar. 2021, doi: 10.52465/joscex.v2i1.26.

[17] C. M. Wati, A. C. Fauzan, and H. Harliana, “Performance comparison of mushroom type classification based on multi-scenario dataset using decision tree C4.5 and C5.0,” *Jurnal Riset Informatika*, vol. 4, no. 3, pp. 247–258, Jun. 2022, doi: 10.34288/jri.v4i3.383.

[18] D. Rianasari, M. N. Triana, M. R. Dewi, Y. Astutik, and R. Wirawan, “The classification of mushroom types using naïve bayes and principal component analysis,” *JISA(Jurnal Informatika dan Sains)*, vol. 5, no. 2, pp. 124–130, Dec. 2022, doi: 10.31326/jisa.v5i2.1380.

[19] S. A. N. Alexandropoulos, S. B. Kotsiantis, and M. N. Vrahatis, “Data preprocessing in predictive data mining,” *Knowledge Engineering Review*, vol. 34, p. e1, Jan. 2019, doi: 10.1017/S026988891800036X.

[20] G. Lawton, “data preprocessing,” 2022. https://www.techtarget.com/searchdatamanagement/definition/data-preprocessing (accessed Sep. 09, 2024).

[21] A. Meiriza, E. Lestari, P. Putra, A. Monaputri, and D. A. Lestari, “Prediction graduate student use Naïve Bayes classifier,” 2020, doi: 10.2991/aisr.k.200424.056.

[22] S. Ray, "Naïve Bayes classifier explained with practical problems," *Analytics Vidhya*, 2024. https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ (accessed Sep. 09, 2024).

[23] N. Darapureddy, N. Karatapu, and T. K. Battula, "Research of machine learning algorithms using k-fold cross validation," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6 Special issue, pp. 215–218, Sep. 2019, doi: 10.35940/ijeat.F1043.0886S19.

[24] K. Naminas, "Machine learning algorithm," 2024. [Online]. Available at: https://labelyourdata.com/articles/how-to-choose-a-machine-learning-algorithm (accessed Sep. 09, 2024).

[25] D. Theng and K. K. Bhoyar, "Feature selection techniques for machine learning: a survey of more than two decades of research," *Knowledge and Information Systems*, vol. 66, no. 3, pp. 1575–1637, Mar. 2024, doi: 10.1007/s10115-023-02010-5.

[26] D. Breton, "A full guide on choosing the right machine learning algorithm," 2022. https://medium.com/@davidbreton03/a-full-guide-on-choosing-the-right-machine-learning-algorithm-5fa282a0b2a1 (accessed Sep. 09, 2024).

[27] N. N. dela Cruz, "Decision support system for predicting cardiovascular diseases using naïve bayesian algorithm," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3178–3183, Jun. 2020, doi: 10.30534/ijatcse/2020/106932020.

[28] A. Özdemir, U. Yavuz, and F. A. Dael, "Performance evaluation of different classification techniques using different datasets," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 3584–3590, Oct. 2019, doi: 10.11591/ijece.v9i5.pp3584-3590.

[29] R. S. Kleiman and D. Page, "AUCμ: A performance metric for multi-class machine learning models," *36th International Conference on Machine Learning, ICML 2019*, 2019. https://proceedings.mlr.press/v97/kleiman19a.html (accessed Sep. 16, 2024).

[30] V. Dey, "Understanding the AUC-ROC curve in machine learning classification," 2021. https://analyticsindiamag.com/ai-mysteries/understanding-the-auc-roc-curve-in-machine-learning-classification/ (accessed Sep. 16, 2024).

[31] "Classification: ROC and AUC," 2024. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

[32] V. Chugh, "AUC and the ROC curve in machine learning | DataCamp," *DataCamp*, 2024. https://www.datacamp.com/tutorial/auc (accessed Sep. 16, 2024).

[33] S. K. Agrawal, "Metrics to evaluate your classification model to take the right decisions," *Analytics Vidhya*, 2021. https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/ (accessed Sep. 19, 2024).

[34] J. J. Lee, M. C. Aime, B. Rajwa, and E. Bae, "Machine learning-based classification of mushrooms using a smartphone application," *Applied Sciences (Switzerland)*, vol. 12, no. 22, p. 11685, Nov. 2022, doi: 10.3390/app122211685.

[35] P. M. Jacob, J. Moni, S. Sunil, A. Johnson, J. M. Mathews, and A. M, "An intelligent system for cultivation and classification of mushrooms using machine vision," in *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, Apr. 2023, vol. 12, no. 22, pp. 264–270, doi: 10.1109/CISES58720.2023.10183464.

[36] A. Samantara, T. Nayak, and B. Parida, "Mushroom quality prediction using machine learning classification," *International Journal of Computer Applications*, vol. 185, no. 9, pp. 14–23, May 2023, doi: 10.5120/ijca2023922746.

[37] 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.

[38] M. A. Ottom, N. A. Alawad, and K. M. O. Nahar, "Classification of mushroom fungi using machine learning techniques," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 5, pp. 2378–2385, Oct. 2019, doi: 10.30534/ijatcse/2019/78852019.

[39] N. R. Seelam, K. Sirisha, H. N. Akunuru, D. Ballipara, and G. A. Prabath, "Mixed breed ensemble method for mushroom classification," in *Proceedings - International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2022*, May 2022, pp. 1071–1075, doi: 10.1109/ICAAIC53929.2022.9793221.

[40] M. S. Ahmed *et al.*, "Comparative analysis of interpretable mushroom classification using several machine learning models," in *Proceedings of 2022 25th International Conference on Computer and Information Technology, ICCIT 2022*, Dec. 2022, pp. 31–36, doi: 10.1109/ICCIT57492.2022.10055555.

[41] I. A. Essa and R. Dhanalakshmi, "Machine learning-based classification of edible and poisonous mushrooms: a performance comparison," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 5, pp. 1364–1370, May 2023, doi: 10.22214/ijraset.2023.51772.

[42] R. Sahu, S. Pandey, R. Verma, and P. Pandey, "Ensemble learning based classification of edible and poisonous agaricus mushrooms," in *2024 4th International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies, ICAECT 2024*, Jan. 2024, pp. 1–7, doi: 10.1109/ICAECT60202.2024.10469539.

[43] S. Metlek and H. Çetiner, "Classification of poisonous and edible mushrooms with optimized classification algorithms," *International Conference on Applied Engineering and Natural Sciences*, vol. 1, no. 1, pp. 408–415, Jul. 2023, doi: 10.59287/icaens.1030.

[44] N. Chitayae and A. Sunyoto, "Performance comparison of mushroom types classification using k-nearest neighbor method and decision tree method," in *2020 3rd International Conference on Information and Communications Technology, ICOIACT 2020*, Nov. 2020, pp. 308–313, doi: 10.1109/ICOIACT50329.2020.9332148.

[45] N. Chumuang *et al.*, "Mushroom classification by physical characteristics by technique of k-nearest neighbor," in *Proceedings - 2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing, iSAI-NLP 2020*, Nov. 2020, pp. 1–6, doi: 10.1109/iSAI-NLP51646.2020.9376820.

[46] D. Wagner, D. Heider, and G. Hattab, "Mushroom data creation, curation, and simulation to support classification tasks," *Scientific Reports*, vol. 11, no. 1, p. 8134, Apr. 2021, doi: 10.1038/s41598-021-87602-3.

[47] S. Upadhyay, A. Dwivedi, A. Verma, and V. Tiwari, "Heart disease prediction model using various supervised learning algorithm," in *Proceedings - 2023 12th IEEE International Conference on Communication Systems and Network Technologies, CSNT 2023*, Apr. 2023, pp. 197–201, doi: 10.1109/CSNT57126.2023.10134595.

[48] S. Ozsari *et al.*, "Deep learning-based classification of macrofungi: comparative analysis of advanced models for accurate fungi identification," *Sensors*, vol. 24, no. 22, p. 7189, Nov. 2024, doi: 10.3390/s24227189.

[49] C. M. Bhatt, P. Patel, T. Ghetia, and P. L. Mazzeo, "Effective heart disease prediction using machine learning techniques," *Algorithms*, vol. 16, no. 2, p. 88, Feb. 2023, doi: 10.3390/a16020088.

BIOGRAPHIES OF AUTHORS

Vilchor G. Perdido is a faculty member in the Computer Science Department at Nueva Vizcaya State University, Bayombong, Nueva Vizcaya, Region 02, and is a passionate programmer and researcher in the field of information technology. His research interests include web/mobile development, IoT, and machine learning. He is currently pursuing his Doctor of Information Technology (DIT) at the University of the Cordilleras. His commitment to advancing IT research is complemented by his academic pursuits. He can be contacted at email: vgperdido@nvsu.edu.ph.

Thelma D. Palaoag is the Director of the Innovation and Technology Transfer Office of the University of the Cordilleras, Baguio City, Philippines. She is a visionary leader, accomplished researcher, and trailblazer in the field of information technology (IT). With an unwavering commitment to advancing technological frontiers, she has dedicated her career to pushing the boundaries of IT research and fostering innovation within academic institutions. She can be contacted at email: tdpalaoag@uc-bcf.edu.ph.